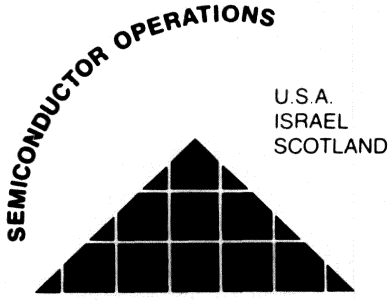
digital

1987

Semiconductor Databook

Volume 1



Confidential and Proprietary

▪ Foreword

We at Semiconductor Operations (SCO) are committed to provide excellence in integrated circuit technologies, products, and services to support our customers, the Digital Systems Groups.

Our primary objective is to optimize Digital's competitive market position by developing leadership system performance at the lowest possible cost and within the appropriate time constraints.

The execution of programs designed to achieve this objective has resulted in the technologies and products described in the *1987 Semiconductor Databook Volumes 1 and 2*. While the basic charter of SCO is to provide strategic and tactical management of all integrated circuit requirements, the VLSI design and manufacturing function of SCO has become the focal point for unique and complex circuits that have contributed significantly to the success of many new Digital products. A strategic investment has been made in CMOS technology and in the design tools necessary to take advantage of this technology. Increased circuit densities and performance have resulted, and capabilities have been extended from full-custom design for maximum performance to semi-custom design for fast time-to-market application. CAD tools are continually being developed to further enhance design and design methodology.

SCO is continually expanding its facilities to provide you with better service. While Hudson and Andover, Massachusetts are the nucleus of the engineering and manufacturing operations, supplemental design facilities are available in Israel and Japan and additional manufacturing capacity is being planned in Scotland. In addition, a new 6-inch wafer pilot fabrication line has been approved for construction in Hudson to aid in the state-of-the-art development of the advanced CMOS devices.

During the past year, many new integrated circuits have been developed and released. Although some are application-specific, the circuits that are suitable for general use are described in Volumes 1 and 2 of this databook. Volume 1 is a revision to the 1986 Databook and includes the latest revisions and changes. Volume 2 contains information related to the new CMOS products that have been recently developed for general use. We encourage you to become familiar with these products and to use them in the design of Digital's systems products when possible. We are ready to assist you in your design process and in support of your production needs.

Our ultimate goal is to ensure that Digital's systems continue to maintain significant competitive advantage through the use of SCO services and products.

• Functional Product Classification

Name	Number	Description
Microprocessors		
78032	21-20887-01	MicroVAX 32-bit CPU
78132	21-22797-01	MicroVAX Floating-point Unit
78532	21-24329-01	MicroVAX Direct Memory Access
78516	21-24330-01	MicroVAX Vectored Interrupt Controller
78584	21-23864-01	MicroVAX Dynamic RAM Controller
ADVICE		MicroVAX Incircuit Emulation Unit
DC327	21-20852-AA	V-11 ROM/RAM
DC328	21-20851-AA	V-11 Instruction/Execution Logic
DC329	21-20850-AA	V-11 Memory Management Logic
DC330	21-20849-AA	V-11 Floating-point Accelerator Logic
DCJ11		DCJ11 16-bit Microprocessor
FPJ11	21-21858-00	FPJ11 Floating-point Accelerator
DCT11	21-17311-01	DCT11 16-bit Microprocessor
Video Controllers		
78610	21-24941-01	Programmable Sprite Cursor
78680	21-25011-01	Video Processor (VIPER)
78690	21-21553-01	Video Control (ADDER)
Communications		
78808	21-23458-01	Octal ART
DC319	21-17312-00	DLART
Bus Support		
78701	19-22110-01	VAXBI Clock Driver
78702	19-22111-01	VAXBI Clock Receiver
78732	21-21689-00	VAXBI BIIC
78733	21-23839-01	VAXBI BCI3
78743	21-23838-01	VAXBI BCAI
DC003	19-12730-00	Dual-interrupt Circuit
DC004	19-12729-00	Register Selector (Protocol Chip)
DC005	19-13040-00	4-Bit Transceiver
DC006	19-14035-00	Word Count/Bus Address Logic
DC010	19-14038-00	Direct Memory Access Logic
DC013	19-14438-00	UNIBUS Request Logic
DC021	19-19015-00	Octal Bus Transceiver
General Purpose		
DC022	19-17871-00	16-word by 4-bit Register File
DC102	19-13888-00	Equals Checker
DC301	21-12623-00	Dual Baud Rate Generator
Mass Storage Support		
DC018	19-19015-00	Serializer/Deserializer Logic
DC024	19-20116-01	Encoder/Decoder Logic
DC309	21-15102-00	Reed Solomon Generator

• Part Identification Codes

The following identification codes are used with the devices in this databook.

780 Series

78xyz - xx

0 = Processors	5 = Controllers	- xx
1 = Coprocessor	6 = Graphic devices	GA = Gullwing
2 = Memories	7 = Bus interfaces	FA = Straight
3 = I/O devices	8 = Communications devices	PA = Pin grid array
4 = Reserved	9 = Reserved	

DC Series

DCxyz

0 = Custom bipolar devices	3 = MOS devices
1 = Custom bipolar devices	5 = MOS devices

• Cross-referencing of Semiconductor Products

Part Name	Part Number	Purchase Number	Description
DC003	DC003	19-12730-00	Dual-interrupt Circuit
DC004	DC004	19-12729-00	Register Selector (Protocol) Logic
DC005	DC005	19-13040-00	4-bit Transceiver
DC006	DC006	19-14035-00	Word Count/Bus Address Logic
DC010	DC010	19-14038-00	Direct Memory Access
DC013	DC013	19-14438-00	UNIBUS Request Logic
DC018	—	19-17043-00/1	Serializer/Deserializer
DC021	DC021	19-19015-00	Octal Bus Transceiver
DC022	—	19-17871-00	16-Word by 4-bit Register File
DC024	—	19-20116-01	Encoder/Decoder Logic
DC028	78701	19-22110-01	VAXBI Clock Driver
DC029	78702	19-22111-01	VAXBI Clock Receiver
DC102	—	19-13888-00	Equals Checker
DC301	—	21-12623-00	Dual Baud Rate Generator
DC309	—	21-15102-00	Reed Solomon Generator
DC310	DCT11	21-17311-01	DCT11 16-bit Microprocessor
DC319	DC319	21-17312-00	DLART
DC321	FPJ11	21-21858-00	FPJ11 Floating-point Accelerator
DC502	78680-GA	21-25011-01	Video Processor (VIPER)
DC323	78690-GA	21-21553-01	Video Control (ADDER)
DC324	78732-PA	21-21689-00	VAXBI BIIC
DC327	—	21-20852-AA	V-11 ROM/RAM
DC328	—	21-20851-AA	V-11 Instruction/Execution Logic
DC329	—	21-20850-AA	V-11 Memory Management Logic
DC330	—	21-20849-AA	V-11 Floating-point Accelerator Logic
DC333	78032-GA	21-20887-01	MicroVAX 32-bit CPU
DC335	DCJ11	21-17679-00	DCJ11 16-bit Microprocessor
DC337	78132-GA	21-22797-01	MicroVAX Floating-point Unit
DC343	78743-PA	21-23838-01	VAXBI BCAI

Part Name	Part Number	Purchase Number	Description
DC344	78733-PA	21-23839-01	VAXBI BCI3
DC349	78808-GA	21-23458-01	Octal ART
DC357	78584-GA	21-23864-01	Dynamic RAM Controller (DYRC)
DC358	78532-GA	21-24329-01	MicroVAX Direct Memory Access (DMA)
DC503	78610-GA	21-24941-01	Programmable Sprite Cursor
DC506	78516-GA	21-24330-01	MicroVAX Vectored Interrupt Controller (VIC)
ADVICE	ADVICE	—	MicroVAX Incircuit Evaluation/Emulation Unit

Contents

Section 1 • Microprocessors and Support Devices

MicroVAX 78032 32-bit Central Processing Unit 1-1
MicroVAX 78132 Floating-point Unit 1-59
MicroVAX 78516 Vectored Interrupt Controller 1-79
MicroVAX 78532 Direct Memory Access 1-109
MicroVAX 78584 Dynamic RAM Controller 1-175
MicroVAX Incircuit Emulator Unit 1-199
Introduction to the V-11 Chipset 1-207
DC327 V-11 Processor ROM/RAM Logic 1-209
DC328 V-11 Processor Instruction/Execution Logic 1-217
DC329 V-11 Processor Memory Management Logic 1-227
DC330 V-11 Processor Floating-point Accelerator Logic 1-241
DCJ11 16-bit Microprocessor 1-249
FPJ11 Floating-point Accelerator 1-323
DCT11 16-bit Microprocessor 1-349

Section 2 • Video Processors and Controllers

78680 Video Processor (VIPER) 2-1
78690 Video Control (ADDER) 2-31
DC503 Programmable Sprite Cursor 2-75

Section 3 • Communications Devices

78808 Eight-channel Asynchronous Receiver/Transmitter 3-1
DC319 DL11 Compatible Asynchronous Receiver/Transmitter 3-27

Section 4 • Bus Support Devices

VAXBI 78732 General Purpose Bus Interconnect Interface 4-1
VAXBI 78743 BCI Adapter Interface 4-135
VAXBI 78733 BCI Bus to MicroVAX II Bus Interface 4-159
VAXBI 78701 Clock Driver 4-191
VAXBI 78072 Clock Receiver 4-203
DC003 Dual-interrupt Logic 4-213
DC004 Register Select (Protocol) Logic 4-223
DC005 4-Bit Transceiver 4-233
DC006 Word Count/Bus Address Logic 4-243
DC010 Direct Memory Access Logic 4-255
DC013 UNIBUS Request and Control Logic 4-269
DC021 Octal Bus Transceiver 4-283

Section 5 ▪ Mass-storage Devices

DC018 Serializer/Deserializer Logic 5-1
DC024 Encoder/Decoder Logic 5-19
DC309 Reed Solomon Generator for ECC 5-49

Section 6 ▪ General Purpose Devices

DC022 16-Word by 4-bit Register File 6-1
DC102 Eight-channel Equals Checker 6-23
DC301 Dual Baud Rate Generator 6-31

Appendixes

Appendix A ▪ MicroVAX 78032 CPU and MicroVAX 78132 FPU Instruction Set A-1
Appendix B ▪ DCT11 and DCJ11 Microprocessors Instruction Set B-1
Appendix C ▪ dc Specification Test Circuits C-1
Appendix D ▪ Input/Output Voltage Waveform Parameters D-1
Appendix E ▪ Mechanical Specifications E-1

Section 1—Microprocessor and Support Devices

The microprocessors and support devices provide a low-cost means to implement the power and versatility of the PDP-11 and VAX computers into system designs.

MicroVAX 32-bit Microprocessor

MicroVAX 78032 Central Processing Unit—The MicroVAX CPU is a 32-bit high-performance microprocessor that contains the architecture and functions of a VAX minicomputer. The MicroVAX 78032 implements a subset of the VAX instruction set and full VAX-11 memory management. It is fabricated in ZMOS (double-metal NMOS) and is contained in a single 68-pin package.

MicroVAX 78132 Floating-point Unit—The MicroVAX FPU is a high-performance cooperative processor used to accelerate the floating-point instructions of the MicroVAX 78032 CPU. It supports floating-point add, subtract, multiply, divide, and convert and other VAX-11 floating-point operations. The FPU is fabricated in ZMOS and is contained in a 68-pin package.

MicroVAX 78516 Vectored Interrupt Controller—The VIC is a programmable interrupt controller that is fully compatible with the MicroVAX 78032 CPU. The 78516 VIC services up to 16 interrupt sources, resolves interrupt priorities, drives the IRQ lines of the CPU, and provides a programmable 16-bit interrupt to the CPU. It is fabricated in high-speed CMOS and is contained in a 68-pin package.

MicroVAX 78532 Direct Memory Access Controller—The 78532 DMA is a high-performance dual-ported four channel virtual memory DMA controller that enables high-speed data transfers between I/O subsytems and peripheral devices and the MicroVAX 78032 CPU bus. It contains dual ports and four channels that are independently programmable. The 132-pin device is fabricated in CMOS.

MicroVAX 78584 Dynamic RAM Controller—The MicroVAX DYRC provides an interface between the MicroVAX CPU and up to 4 Mbytes of dynamic random access memory (DRAM). The 78584 DYRC operates at two speeds to support 256K by 1-bit DRAMs that operate at different speeds. It is contained in an 84-pin package and is fabricated in CMOS.

Advanced Development VAX Incircuit Emulator—The ADVICE is contained on a single module and provides a full-speed incircuit emulator of the MicroVAX 78032 CPU and MicroVAX 78132 FPU. It is used for the development of hardware and software products using the MicroVAX CPU and FPU.

V-11 32-bit VAX Processor

The V-11 processor chip set consists of four custom VLSI chips that were developed for use with the Scorpio CPU module which is a single module VAX system.

DC327 ROM/RAM Logic—The ROM/RAM chip is a 44-pin cerquad device that provides the microcode control store function for the V-11 processor.

DC328 Instruction/Execution Logic—The I/E chip is a 132-pin PGA device that functions as the main data path and contains the microsequencer, minitranslation buffer, and instruction buffer.

DC329 Memory Management Logic—The M chip is a 132-pin PGA device that provides most of the memory management logic and includes a tag store for cache memory, four UARTS, and a 512-entry backup translation buffer.

DC330 Floating-point Accelerator Logic—The F chip is a 132-pin PGA device used to decrease the execution time of F , D , and G floating-point instructions and some integer multiply and divide instructions.

PDP-11 16-bit Processors

DCJ11 Microprocessor—The DCJ11 microprocessor is a 60-pin CMOS DIP device that implements the full PDP-11 instruction set and has a performance comparable to the PDP-11/44 miniprocessor.

FPJ11 Floating-point Accelerator—The FPJ11 FPA is a 40-pin DIP that implements in hardware all the floating-point instructions of the DCJ11 thereby significantly improving the performance of floating-point instructions.

DCT11 Microprocessor—The DCT11 microprocessor is a 40-pin DIP device that contains the essential elements of the PDP-11 architecture.

• Features

- High performance
  - 32-bit internal and external data path
  - Pipelined architecture
  - Instruction prefetch
- 4 gigabyte virtual address space
- 1 gigabyte physical address space
  - 512 Mbyte physical memory space
  - 512 Mbyte I/O space
- VAX memory management
  - Full memory protection
  - Four privilege modes
  - Process and system space mapped
- Subset of the VAX instruction set
  - 245 instructions
  - 21 address modes
  - 9 data types
- Sixteen 32-bit general purpose registers
- 22 interrupt levels
  - 15 software
  - 7 hardware
- Vectored software and hardware interrupts
- Industry compatible external interface
- Single 5 Vdc power supply

• Description

The MicroVAX 73082 is a high-performance single-chip microprocessor that provides the architecture and functions of the VAX minicomputer in a single 68-pin package. Fabricated in ZMOS (double-metal MOS), the MicroVAX 78032 implements a full 32-bit architecture that can directly access 4 Gbytes of virtual memory and 1 Gbyte of physical memory. Figure 1 is a block diagram of the MicroVAX 78032 microprocessor.

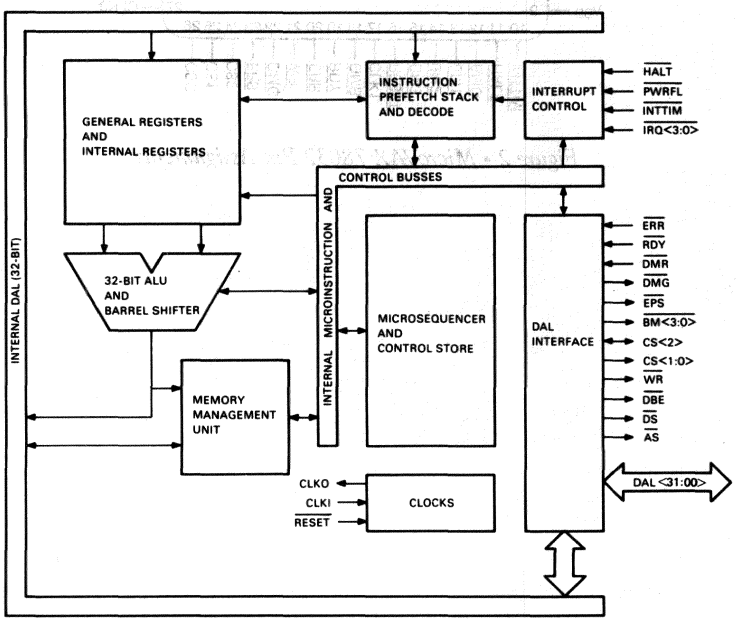


Figure 1 • MicroVAX 78032 Microprocessor Block Diagram

The MicroVAX 78032 uses a single 5 Vdc power supply, requires no special support logic, and is easily interfaced with industry standard peripheral chips. It is ideal for use as a single-board computer, personal computer and workstation, and as a low-end system.

• Pin and Signal Descriptions

This section provides a brief description of the input and output signals and power and ground connections of the MicroVAX 78032 68-pin package. The pin assignments are identified in Figure 2 and the signals are summarized in Table 1.

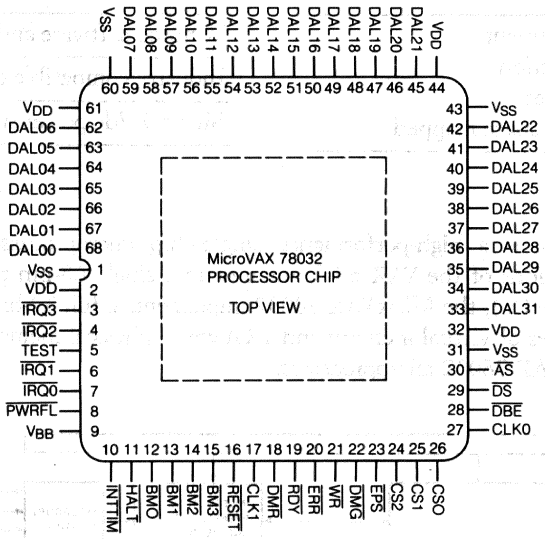


Figure 2 • MicroVAX 78032 Pin Assignments

Table 1 • MicroVAX 78032 Pin and Signal Summary

Pin	Signal	Input/Output*	Definition/Function
33–42 45–59 62–68	DAL<31:00>	input/output	Data/Address lines—Time multiplexed, bidirectional data and address bus.
30	$\overline{AS}$	output	Address strobe—System address strobe.
29	$\overline{DS}$	output	Data strobe—System data strobe.
12–15	$\overline{BM}$<3:0>	input/output	Byte masks—Identifies the bytes of the DAL bus that contain valid data.
21	$\overline{WR}$	output	Write—Provides read/write control for the bus.
28	$\overline{DBE}$	output	Data buffer enable—Provides control of the DAL bus transceivers.
19	$\overline{RDY}$	input	Ready—Provides status of the data transfer during a memory or interrupt bus cycle.
20	$\overline{ERR}$	input	Error—Indicates a bus or memory error.
16	$\overline{RESET}$	input	Reset—Starts the CPU initialization process.
11	$\overline{HALT}$	input	Halt—Halts the execution of macroinstructions (nonmaskable interrupt).
24–26	CS<2:0>	input/output	Control status—Indicates the type of bus cycle.
3,4, 6,7	$\overline{IRQ}$<3:0>	input/output	Interrupt request—Interrupt lines for device interrupts.
8	$\overline{PWRFL}$	input	Powerfail—Indicates an external power fail condition.
10	$\overline{INTTIM}$	input	Interval timer—Indicates an external interval timer condition.
18	$\overline{DMR}$	input	DMA request—Requests the bus for DMA transfers.
22	$\overline{DMG}$	output	DMA grant—Grants bus for DMA transfers.
23	$\overline{EPS}$	output	External processor strobe—Coordinates external processor transactions.
2,32, 44,61	V_{DD}	input	Supply voltage—5 Vdc supply.
1,31, 43,60	V_{SS}	input	Ground—Ground reference.
17	CLKI	input	Clock in—Clock input for chip timing.
27	CLKO	input	Clock out—Clock output at half the frequency of CKLI.
9	V_{BB}	output	Back-bias—For manufacturing use.
5	TEST	input	Test—For manufacturing use.

*All signals are TTL levels except for pin 23 ($\overline{EPS}$) which is CMOS.

Data and Address Bus

Data and address bus (DAL<31:00>)—The data and address bus is a time-multiplexed bidirectional bus that transfers address, data, and other information during bus cycles. For a detailed description of DAL<31:00> bus, refer to the *MicroVAX 78032 32-Bit Central Processing Unit User's Guide. (Document No. EK-78032-UG)*

Bus Control

Address strobe ($\overline{AS}$)—This signal indicates that valid address information is available on the DAL<29:02> bus and valid status information is on the $\overline{BM<3:0>}$, CS<2:0>, and $\overline{WR}$ lines. The leading edge of this signal can be used to latch the address.

External processor strobe ($\overline{EPS}$)—This signal is used by the CPU to coordinate external processor transactions. It is used with the following transactions:

- Transaction between an external processor controlled by the CPU, such as the MicroVAX 78132 Floating-Point Unit.
- Transactions between logic that implements a register or registers that are defined as a part of the MicroVAX internal processor register set.

Data strobe ($\overline{DS}$)—This signal indicates that the DAL bus is free to receive data during a CPU read cycle or that valid data is on the DAL bus during a CPU write cycle.

Byte masks ($\overline{BM<3:0>}$)—These signals are used to indicate which bytes of the DAL bus contain valid data as listed in Table 2. For a read cycle, they indicate which bytes of the DAL bus must have data driven onto them. For a write cycle, they indicate which bytes of the DAL bus contain valid data. Bits $\overline{BM<3:0>}$ are valid when the $\overline{AS}$ signal is asserted.

Table 2 • MicroVAX 78032 Byte Mask Data Selection

Byte Mask	Valid Data Byte
$\overline{BM3}$	DAL<31:24>
$\overline{BM2}$	DAL<23:16>
$\overline{BM1}$	DAL<15:08>
$\overline{BM0}$	DAL<07:00>

Write ($\overline{WR}$)—This signal specifies the direction of data transfer on the DAL bus for the current bus cycle. When asserted, the CPU is performing a write operation. When not deasserted, the CPU is performing a read operation. The $\overline{WR}$ signal is valid when the $\overline{AS}$ or $\overline{EPS}$ signal is asserted.

Data buffer enable ($\overline{DBE}$)—This signal is used with the $\overline{WR}$ signal to the control transceivers that may be between the CPU and the DAL bus.

Ready ($\overline{RDY}$)—This signal is asserted by external logic to indicate the completion of the current bus cycle. When not asserted, it extends the current bus cycle for a slower memory or peripheral device. The $\overline{RDY}$ or $\overline{ERR}$ signal must be asserted to end the current bus cycle.

Error ($\overline{ERR}$)—This signal is asserted by external logic to indicate that an error, associated with the current bus cycle, has occurred (e.g., bus timeout or parity error) and to end the bus cycle. The $\overline{ERR}$ or $\overline{RDY}$ signal must be asserted to end the bus cycle.

System Control

Reset ($\overline{RESET}$)—This input signal is used to initialize the CPU to a known state.

Control status (CS<2:0>)—These lines are used with either the $\overline{AS}$ or the $\overline{EPS}$ and $\overline{WR}$ signals to define the type of operation in progress for the current bus cycle lines. Lines CS<2:0> are valid when the $\overline{AS}$ or the $\overline{EPS}$ signal is asserted.

During a read, write, or interrupt-acknowledge cycle ($\overline{AS}$ asserted), the $\overline{WR}$ and CS<2:0> lines select the bus cycles indicated in Table 3.

Table 3 • MicroVAX 78032 Bus Cycle Selection

Write $\overline{WR}$	Control Status CS<2:0>	Bus Cycle
H	LLL	reserved
H	LLH	reserved
H	LHL	reserved
H	LHH	interrupt acknowledge
H	HLL	read (instruction)
H	HLH	read lock
H	HHL	read (data, modify intent)
H	HHH	read (data, no modify intent)
L	LLL	reserved
L	LLH	reserved
L	LHL	reserved
L	LHH	reserved
L	HLL	reserved
L	HLH	write unlock
L	HHL	reserved
L	HHH	write (data)

At the beginning of an external processor read, write, or response cycle ($\overline{EPS}$ asserted), the CS2 signal is high, and the $\overline{WR}$ and CS<1:0> signals select the bus cycles indicated in Table 4.

Table 4 • MicroVAX 78032 External Register Bus Cycle

Write $\overline{WR}$	Control Status CS<1:0>	Bus Cycle
H	LL	reserved
H	LH	read data
H	HL	reserved
H	HH	response enable
L	LL	write command (FPU)
L	LH	write data
L	HL	write command (non-FPU)
L	HH	reserved

During a response enable cycle the CS<2> signal may be pulled low by the external logic. Refer to the External Processor Cycle section for a description of a response enable cycle.

Interrupt Control

Interrupt request ($\overline{IRQ}$<3:0>)—These lines are used by the external logic to generate interrupt requests to the CPU. The lines are sampled by the CPU every microcycle. Table 5 lists the interrupt level assignments.

Table 5 • MicroVAX 78032 Interrupt Request Assignments

IRQ Line	Interrupt Level
$\overline{IRQ\ 3}$	IPL 17
$\overline{IRQ\ 2}$	IPL 16
$\overline{IRQ\ 1}$	IPL 15
$\overline{IRQ\ 0}$	IPL 14

Powerfail ($\overline{PWRFL}$)—This line allows the external logic to notify the CPU of a powerfail condition. It is sampled by the CPU every microcycle. The $\overline{PWRFL}$ signal generates an interrupt at IPL 1E (hexadecimal). This interrupt is internally acknowledged by the CPU and does not use an interrupt acknowledge bus cycle.

Interval timer ($\overline{INTTIM}$)—This line provides system timing information of the interval timer and is sampled every microcycle. The $\overline{INTTIM}$ signal generates an interrupt at IPL 16 (hexadecimal). This interrupt is internally acknowledged by the CPU and does not use an interrupt acknowledge bus cycle.

Halt ($\overline{HALT}$)—This signal results in an interrupt used to halt the execution of macroinstructions, and is sampled every microcycle. At the conclusion of the current macroinstruction the CPU executes an external processor write cycle (CS<1:0> = 10 and DAL<05:00> = 111111) and then enters the restart process. The restart process sets the CPU to a known state and then passes control to user code beginning at physical address 20040000 (hexadecimal). For a description of the restart process, refer to the *MicroVAX 78032 User's Guide.*

DMA Control

DMA request ($\overline{DMR}$)—This signal is used by the external logic to take control of the DAL bus and its related control signals.

DMA grant ($\overline{DMG}$)—This signal indicates that the CPU has granted the use of the DAL bus and its related control signals.

Clock Signals

Clock in (CLKI)—A TTL input that provides the basic clock timing to the clock generator on the MicroVAX 78032.

Clock out (CLKO)—A timing signal output at half the frequency of basic clock (CLKI) to be used for system timing.

Miscellaneous Signals

Test (TEST)—Reserved. This pin *must* be connected to ground.

Power Supply Connections

Power (V_{DD})—5 Vdc supply.

Ground (V_{SS})—Ground reference.

Back-bias generator (V_{BB})—Reserved. This pin *must not* be connected.

Architecture Summary

The MicroVAX 78032 architecture shown in Figure 3 is grouped into two main areas. One area is used by the application programmer and contains general registers, pointer registers, and the processor status word. The remaining area is used by the system programmer and contains process control registers, memory management registers, interrupt registers, and the processor status longword.

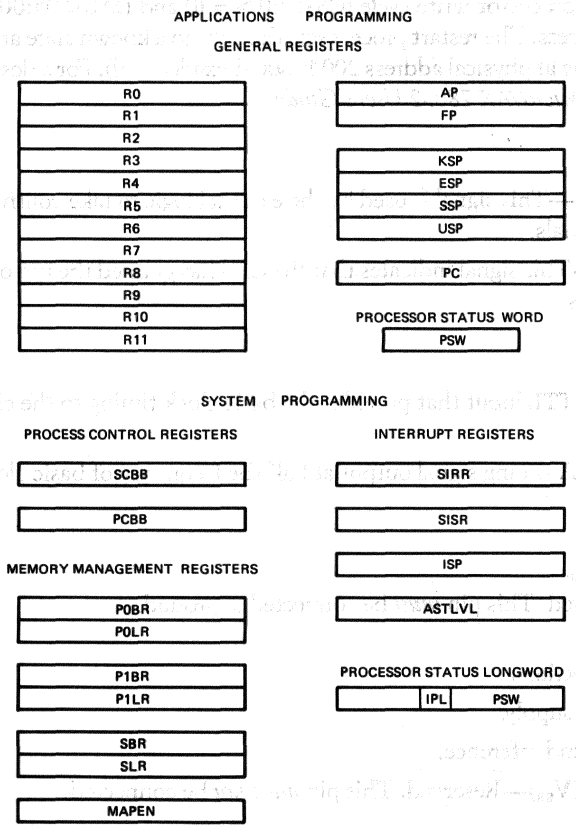


Figure 3 ▪ MicroVAX 78032 Programming Model

General Registers

Sixteen 32-bit general registers are included and can be used for temporary storage, as accumulators, as base registers, and index registers. The registers used for specific functions are the stack pointer (SP), argument pointer (AP), frame pointer (FP), and program counter (PC).

Stack pointer—The stack pointer (SP) contains the address of the processor defined stack. There are five stack pointers, one for each of the four operating modes of the processor and one for use by the system for handling interrupts. The stack pointer in use is determined by the operating mode of the processor.

Argument pointer—The VAX procedure call convention uses an argument list data structure. The argument pointer (AP) contains the address of the base of this structure.

Frame pointer—The VAX procedure call convention builds a data structure on the stack frame. The frame pointer contains the address of the base of this structure.

Program counter—The program counter (PC) contains the address of the next byte of the program. Therefore, the PC is not used as an accumulator, index, or temporary register.

Processor Status Word

The processor status word (PSW), the lower 16 bits of the PSL, contains the condition codes and trap enable flags. The PSW is the user accessible portion of the processor status longword (PSL) and is shown in Figure 4 and described in Table 6.

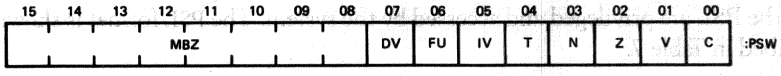


Figure 4 ▪ MicroVAX 78032 Processor Status Word Format

Table 6 ▪ MicroVAX 78032 Processor Status Word Descriptions

Bit	Description
15:08	MBZ (Must be zero).
07:04	Trap Enable Flags—These bits cause traps to occur under the following conditions. DV (Decimal overflow)—Used by the emulation software in the emulation of decimal instructions. FU (Floating underflow)—When set, this bit causes a floating underflow fault after the execution of any instruction which produced a floating result too small in magnitude to be represented. IV (Integer overflow)—When set, this bit causes an integer overflow trap after an instruction that produced an integer result that could not be correctly represented in the space provided. T (Trace)—When set, this bit causes a trace trap to occur after the execution of the next instruction.
03:00	Condition Codes—These bits contain information related to the result of the last CPU arithmetic or logical operation. The bits are defined as follows: N = 1 if the result was negative. Z = 1 if the result was zero. V = 1 if the operation resulted in an arithmetic overflow. C = 1 if the operand resulted in a carry-out-of or borrow-into the MSB (Most-Significant Bit).

System Registers

The system registers are privileged registers that are accessed by the operating system. These registers are used in context switching, memory management, exception and interrupt handling, and processor control.

System control block base register—The system control block base register (SCBB) contains the base address of the system control block (SCB). The SCB contains the vectors used for servicing interrupts and exceptions.

Process control block base register—The processor control block base register (PCBB) contains the base address of the process control block (PCB). The PCB contains the hardware context of the current process.

Memory management registers—These registers are used by the system to control the memory management unit of the MicroVAX 78032 and to access the page-table entries in memory used to translate virtual addresses into physical addresses. The function of each of these registers is described in the memory management section.

Interrupt registers—These registers are used to control the interrupt system of the processor by storing interrupt requests, current interrupt priority level, and the interrupt stack pointer. The function of each of these registers is described in the Exception and Interrupt section.

Processor status longword—The processor status longword (PSL) contains the processor status information. The lower 16 bits of the PSL are the user accessible processor status word. The upper 16 bits of the PSL are privileged and accessed by the system. The PSL format is shown in Figure 5 and described in Table 7.

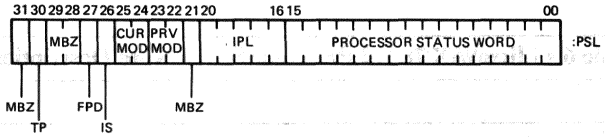


Figure 5 • MicroVAX 78032 Processor Status Longword Format

Table 7 • MicroVAX 78032 Processor Status Longword Descriptions

Bit	Descriptions
31	MBZ (Must be zero).
30	TP (Trace pending)—Forces a trace trap when set at the beginning of any instruction. Set by the processor if the T bit in the PSW is set at the beginning of an instruction.
29:28	MBZ (Must be zero).
27	FPD (First part done)—Set when an exception or interrupt occurs during an instruction that can be suspended. If FPD is set when the processor returns from an exception or interrupt, it resumes the interrupted operation where it left off, rather than restarting the instruction.
26	IS (Interrupt stack)—Set when the processor is executing on the interrupt stack.
25:24	CUR MOD (Current access mode)—The access mode of the currently executing process as follows: 0 = Kernel 1 = Executive 2 = Supervisor 3 = User
23:22	PRV MOD (Previous access mode)—Loaded from CUR MOD by exceptions and Change Mode instructions, cleared by interrupts, and restored by REI.
21	MBZ (Must be zero).
20:16	IPL (Interrupt priority level)—Contains the current processor priority in the range 0 to 1F hexadecimal. The processor will accept interrupts only on levels greater than its current IPL.
15:00	PSW (Processor status word)—Contains the processor status accessible by the user.

▪ Data Types

The architecture of the MicroVAX 78032 supports nine data types: byte, word, longword, quadword, character string, variable-length bit field and, through the optional floating-point unit, F_floating, D_floating, and G_floating. Figures 6 and 7 show the organization of the data types.

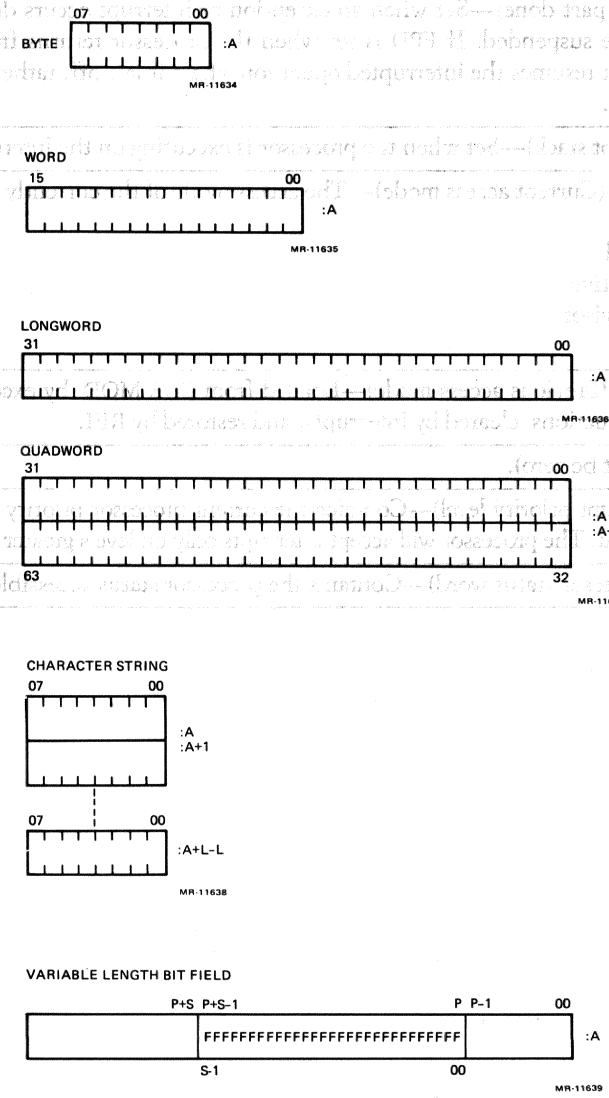


Figure 6 ▪ MicroVAX 78032 Integer, Character-string, and Bit-field Data Types

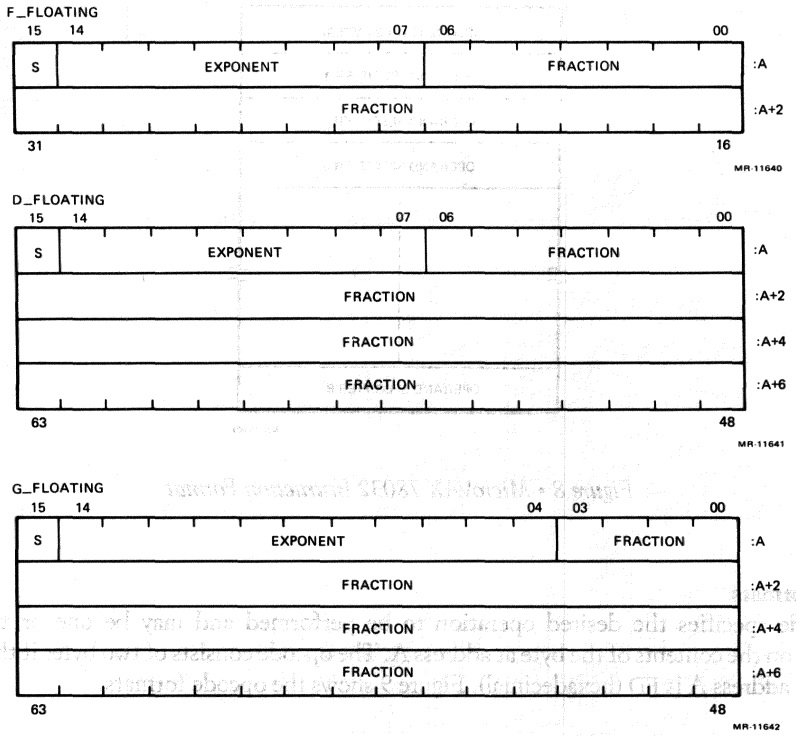


Figure 7 ▪ MicroVAX 78032 Floating-point Data Types

▪ Instruction Formats

The VAX instruction set has a variable length instruction format that may be one byte or more depending on the type of instruction. The general format of a VAX instruction is shown in Figure 8. Each instruction consists of an opcode followed by as many as six operand specifiers. The number and type of operand specifiers depends on the opcode. All operand specifiers are of the same format: an address mode plus additional information used to locate the operand. This additional information contains up to two register designators and addresses, data, or displacements. The use of the operand is determined from the opcode and is called the operand type. It includes both the access type and the data type.

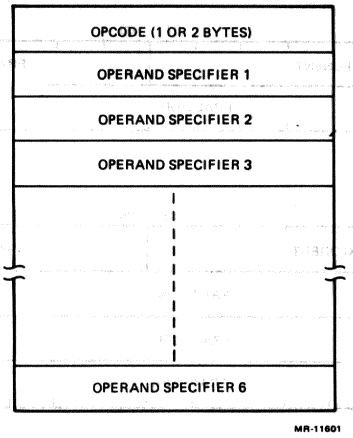


Figure 8 ▪ MicroVAX 78032 Instruction Format

Opcode Formats

The opcode specifies the desired operation to be performed and may be one or two bytes, depending on the contents of the byte at address A. The opcode consists of two bytes if the value of the byte at address A is FD (hexadecimal). Figure 9 shows the opcode formats.

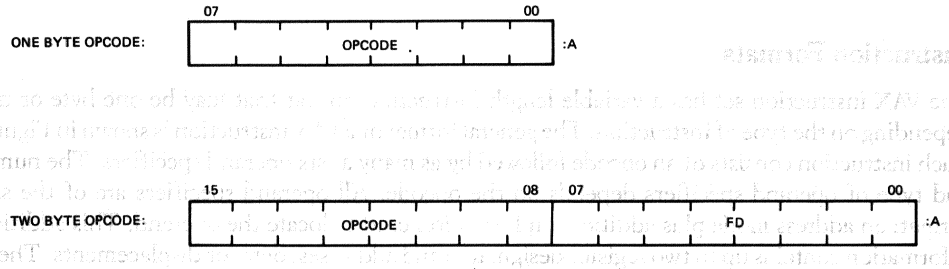


Figure 9 ▪ MicroVAX 78032 Opcode Formats

• Addressing Modes

A summary of the addressing modes used by the MicroVAX 78032 is listed in Table 8 with a brief description of each mode.

Table 8 • MicroVAX 78032 Register Addressing Modes

General Register Addressing Mode

Hexadecimal	Name	Assembler	Access r	m	w	a	v	PC	SP	Indexable?
0–3	literal	S^ #literal	y	f	f	f	f	–	–	f
4	index	i (Rx)	y	y	y	y	y	f	y	f
5	register	Rn	y	y	y	f	y	u	uq	f
6	register deferred	(Rn)	y	y	y	y	y	u	y	y
7	autodecrement	–(Rn)	y	y	y	y	y	u	y	ux
8	autoincrement	(Rn)+	y	y	y	y	y	p	y	ux
9	autoincrement deferred	@(Rn)	y	y	y	y	y	p	y	ux
A	byte displacement	B^ d(Rn)	y	y	y	y	y	p	y	y
B	byte displacement deferred	@B^ d(Rn)	y	y	y	y	y	p	y	y
C	word displacement	W^ d(Rn)	y	y	y	y	y	p	y	y
D	word displacement deferred	@W^ d(Rn)	y	y	y	y	y	p	y	y
E	longword displacement	L^ d(Rn)	y	y	y	y	y	p	y	y
F	longword displacement deferred	@L^ d(Rn)	y	y	y	y	y	p	y	y

Program Counter Addressing Mode

Hexadecimal	Name	Assembler	Access r	m	w	a	v	Indexable?
8	immediate	I^ #constant	y	u	u	y	y	y
9	absolute	@#address	y	y	y	y	y	y
A	byte relative	B^ address	y	y	y	y	y	y
B	byte relative deferred	@B^ address	y	y	y	y	y	y
C	word relative	W^ address	y	y	y	y	y	y
D	word relative deferred	W^ address	y	y	y	y	y	y
E	longword relative	L^ address	y	y	y	y	y	y
F	longword relative deferred	L^ address	y	y	y	y	y	y

Table 8 • MicroVAX 78032 Register Addressing Modes (Cont.)

Addressing Legend

Access:	Syntax:
r = read	i = any indexable address mode
m = modify	d = displacement
w = write	Rn = general register, n = 0 to 15
a = address	Rx = general register, x = 0 to 14
v = field	

Results:

y = yes, always valid address mode
f = reserved address mode fault
– = logically impossible
p = program counter addressing
u = unpredictable
uq = unpredictable for quad, D_/G_floating, or field if pos + size > 32
ux = unpredictable if index reg = base reg

General Register Addressing

The general register address modes use one or more general registers, depending on the instruction and data type, or information required to locate the operand(s) to be used by the specified instruction.

Register mode—The operand is contained in one of the general registers (Rn).

Register deferred mode—Register Rn contains the address of the operand.

Autoincrement mode—Register Rn contains the address of the operand. After the operand address is determined, the size of the operand in bytes is determined by its data type and is added to the contents of register Rn and the result is placed in register Rn.

Autoincrement deferred mode—Register Rn contains a longword address that points to the operand address. After the operand address has been determined, the number four is added to the contents of Rn and the result is so tred in Rn.

Autodecrement mode—The size of the operand in bytes is determined by its data type and is subtracted from the contents of Rn and the result is stored in Rn. The updated content of register Rn is the address of the operand.

Literal mode—Literal mode addressing provides an efficient means of specifying integer constants in the range of from 0 to 63 (decimal). In addition to short integer literals, this mode can be used to specify floating-point literals. The value is contained in the operand specifier.

Displacement mode—The displacement contained in the operand specifier, after being sign-extended to 32 bits if it is a byte or word, is added to the contents of register Rn, and the result is the operand address.

Displacement deferred mode—The displacement contained in the operand specifier, after being sign-extended to 32 bits if it is a byte or word, is added to the contents of register Rn, and the result is the operand address.

Index mode—The operand specifier consists of two bytes or more, a primary operand specifier and a base operand specifier. The primary operand specifier, contained in bits 0 through 7, includes the index register (Rx) and a mode specifier of 4. The address of the primary operand is determined by multiplying the contents of index register Rx by the size of the primary operand in bytes which is determined by operand type. This value is then added to the address specified by the base operand specifier (bits 15 through 8), and the result is the primary operand address.

Program Counter Addressing

Register 15 is used as the program counter (PC). It can also be used in the addressing modes. The processor increments the program counter as the opcode, operand specifier, and immediate data or addresses of the instruction are evaluated. The amount that the PC is incremented is determined by the opcode, number of operand specifiers and other values. The PC can be used with all of the VAX addressing modes, except register, index, register deferred, or autodecrement.

Immediate mode—This mode is an autoincrement mode and the PC is used as the general register. The contents of the location following the addressing mode contain immediate data.

Absolute mode—This mode is an autoincrement deferred mode using the PC as the general register. The contents of the location following the addressing mode are taken as the operand address. This is interpreted as an absolute address that is an address that remains constant in the memory location where the assembled instruction is executed.

Relative mode—This mode is a displacement mode and the PC is used as the general register. The displacement that follows the operand specifier is added to the PC and the sum is the address of the operand.

Relative deferred mode—This mode is similar to the relative mode except that the displacement, which follows the addressing mode, is added to the PC and the sum is the longword address of the operand.

Branch Addressing

During branch displacement addressing, the byte or word displacement is sign-extended to 32 bits and added to the updated content of the PC. The updated content of the PC is the address of the first byte beyond the operand specifier.

Instruction Set

A summary of the VAX instructions implemented by the MicroVAX 78032, the floating-point instructions supported by the floating-point unit, and the emulated instructions that are assisted by the microcode are listed in Appendix A.

Operand Type

The operand type specifies the use of the operand that is associated with an instruction. The opcode includes the data type of each operand and the method of access as follows:

1. Read—The specified operand is ready-only.
2. Write—The specified operand is write-only.
3. Modify—The specified operand is read, may be modified, and is written.
4. Address—Address calculation occurs until the address of the operand is obtained. In this mode, the data type indicates the operand size to be used in address calculation. The specified operand is not accessed directly; however, the instruction may use the address to access that operand.

5. Variable bit field base address—If only register R[n] is specified, the field is in general register R[n] or in R[n+1]'R[n] (i.e., R[n+1] concatenated with R[n]). If R[n] is not specified, an address calculation occurs until the actual address of the operand is obtained. This address specifies the base to which the field position (offset) is applied.
6. Branch—No operand is accessed. The operand specifier is the branch displacement and the data type indicates the size of the branch displacement.

• Memory Management

The memory management unit provides a flexible and efficient virtual memory programming environment. Memory management and the operating system provide paging with user control and swapping. It also provides four hierarchical modes—kernel, executive, supervisor, and user, with read/write access control for each mode.

The VMS Virtual Memory System provides a large address space and allows programs to run with small memory configurations. Programs are executed in a process environment. Each process can operate with an address space of 4-billion bytes.

Virtual Address Space

Memory management divides the virtual address space into two spaces of equal size—the system space and the process space. The process space is divided into P0 and P1 regions. Figure 10 shows the virtual address space assignments.

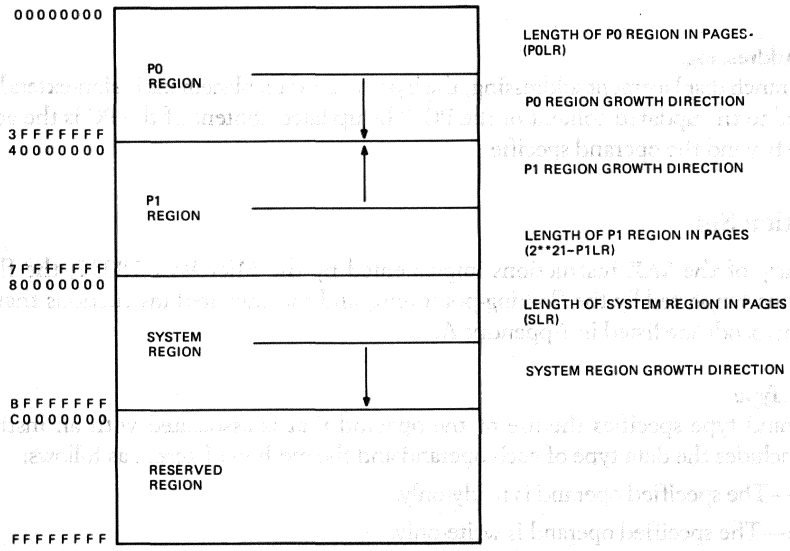


Figure 10 • MicroVAX 78032 Virtual Address Space Assignments

Virtual address format—A 32-bit virtual address is generated for each instruction and operand in memory. As the process is executed, the processor translates each virutal address into a physical address. The format of a virtual address is shown in Figure 11 and described in Table 9.

Page protection—Independent of its location in virtual address space, a page of 512 bytes may be protected according to its use. A program may generate any address; however, the program may be prevented from modifying or accessing portions of the shared system space. A program may also be prevented from accessing or modifying portions of process space.

Virtual address space layout—Access to the P0, P1, and System regions is controlled by the (P0LR, P1LR, and SLR) length registers. Within the limits set by the length registers, the access is controlled by a page table that specifies the validity, access requirements, and location of each page in the region.

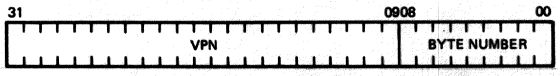


Figure 11 • MicroVAX 78032 Virtual Address Format

Table 9 • MicroVAX 78032 Virtual Address Descriptions

Bit	Descriptions
31:09	VPN (Virtual Page Number)—This field specifies the virtual page to be referenced. Virtual address space contains 8,388,608 pages of 512 bytes each. Bits <31:30> of the VPN select the region of virtual address space being referenced as follows: Value of Bits <31:30> — Region Referenced 0 — P0 1 — P1 2 — System 3 — Reserved
08:00	Byte number—This field specifies the number of the byte within the page.

Access Control

The access control function determines whether a read or write memory reference will be allowed to a memory page. Every page in memory is assigned a protection code to prevent illegal access to memory information.

Mode—The four hierarchical modes used by the MicroVAX 78032 in the order of most to least privileged are

0 Kernel—used by the kernel of the operating system for page management, scheduling, and I/O drivers.

1 Executive—used for many of the operating system service calls.

2 Supervisor—used for services such as command interpretation.

3 User—used for user level code, utilities, compilers, debuggers, etc.

The current processor mode is stored in the current mode field of the processor status longword (PSL).

Protection code—A protection code, located in the page table entry for that page, defines the accessibility of the page for each mode. These codes are described in Table 10.

Table 10 • MicroVAX 78032 Protection Code Assignments

Decimal Code	Binary Code	Mnemonic	Current Mode K	E	S	U	Comment
0	0000	NA	—	—	—	—	no access
1	0001	*	*	*	*	*	reserved
2	0010	KW	RW	—	—	—	
3	0011	KR	R	—	—	—	
4	0100	UW	RW	RW	RW	RW	all access
5	0101	EW	RW	RW	—	—	
6	0110	ERKW	RW	R	—	—	
7	0111	ER	R	R	—	—	
8	1000	SW	RW	RW	RW	—	
9	1001	SREW	RW	RW	R	—	
10	1010	SRKW	RW	R	R	—	
11	1011	SR	R	R	R	—	
12	1100	URSW	RW	RW	RW	R	
13	1101	UREW	RW	RW	R	R	
14	1110	URKW	RW	R	R	R	
15	1111	UR	R	R	R	R	

Legend:

— = no access
* = unpredictable
R = read only
RW= read/write
W = write

K = Kernel
E = Executive
S = Supervisor
U = User

• Memory Management Control

Three registers are used to control memory management. One register is used to enable and disable memory management and the other two are used to control the address translation buffer.

Memory Management Enable

The map enable register (MAPEN) determines whether the memory management functions is disabled or enabled. The format of the map enable register is shown in Figure 12 and described in Table 11.

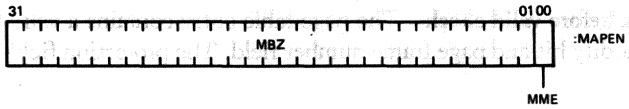


Figure 12 • MicroVAX 78032 Map Enable Register Format

Table 11 • MicroVAX 78032 Map Enable Register Description

Bit	Descriptions
31:01	MBZ (Must be zero).
00	MME (Memory management enable)—used to enable and disable memory management as follows: 1 = MME enabled 0 = MME disabled

Translation Buffer

The translation buffer stores frequently used memory page references. The translation buffer stores eight entries that contain page table entries (PTE) for successful virtual address translations. It is controlled by the translation buffer invalidate single (TBIS) register and the translation buffer invalidate all (TBIA) register.

The TBIS register invalidates single PTE entries in the translation buffer. This is accomplished by the system software by writing a virtual address into the TBIS register shown in Figure 13. The MicroVAX 78032 will invalidate the translation buffer entry that maps to the page in virtual memory accessed by the virtual address written into the register.

The TBIA register clears the translation buffer by invalidating all the PTEs in the translation buffer. This is accomplished when the system software writes a 0 into the TBIA register shown in Figure 13. When a 0 is written into the TBIA register, all the PTEs in the translation buffer will be invalidated.

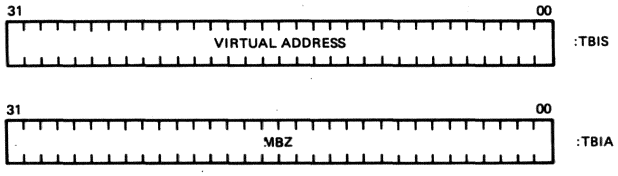


Figure 13 • MicroVAX 78032 Translation Buffer Registers

Address Translation

Translating a virtual address into a physical address by the memory management unit is controlled by the memory mapping enable (MME) bit in the MAPEN register. When MME = 0, the memory mapping is disabled, bits <29:00> of the virtual address become the physical address, and access is allowed in all modes. When MME = 1, the memory mapping is enabled and the virtual address is mapped to a physical address by the memory management. The address translation process when memory management is enabled is as follows.

Page table entry—All virtual addresses are translated to physical addresses by a page table entry (PTE) shown in Figure 14 and described in Table 12.

Protection check before valid check—The page table entry contains a valid bit that controls the validity of the modify bit and page frame number field. The protection field is always valid and is checked first.

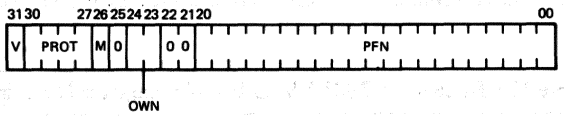


Figure 14 • MicroVAX 78032 Page-table Entry Format

Table 12 • MicroVAX 78032 Page Table Entry Descriptions

Bit	Descriptions
31	V (Valid bit)—Governs the validity of the M modify bit and the page frame number (PFN) field. V = 1 for valid; V = 0 for not valid.
30:27	PROT (Protection field)—Describes the protection for the page. This field is always valid and is used by the hardware even when V = 0.
26	M (Modify)—This bit is set (= 1) if the page has already been recorded as modified. M = 0 if the page has not been recorded as modified. Used only if V = 1.
25	0 (Zero)—reserved.
24:23	OWN (Owner)—reserved.
22:21	0 (Zero)—reserved.
20:00	PFN (Page frame number)—The upper 21 bits of the physical address of the base of the page. Used if V = 1.

System Space Address Translation

A virtual address with bits 31 and 30 equal to 2 is an address in the system virtual address space that is mapped by the system page table (SPT). The SPT is located in physical memory and its location and length are defined by the system base register (SBR) and the system length register (SLR), Figure 15. The SBR contains the physical address of the system page table. The SLR contains the size of the SPT in longwords that is the number of page table entries. The page table entry pointed to by the SBR maps to the first page of system virtual address space that is virtual byte address 80000000 (hexadecimal).

Figure 16 shows the translation of a system virtual address to a physical address.

The algorithm used to generate a physical address from a system region virtual address is

SYS_PA=(SBR+4*SVA<29:9>)<20:00> SVA <08:00>

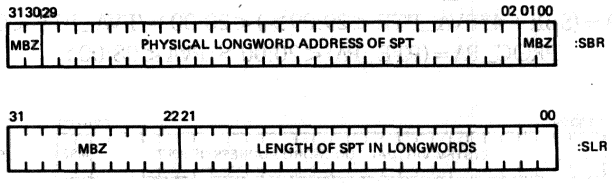


Figure 15 • MicroVAX 78032 System Mapping Register Formats

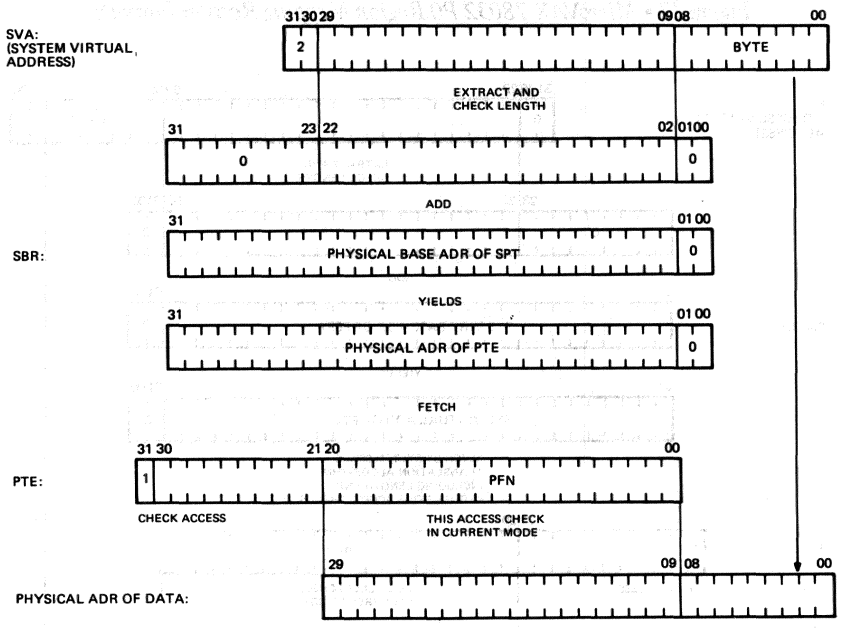


Figure 16 • MicroVAX 78032 System Virtual-to-Physical Address Translation

Process Space Address Translation

A virtual address with bit 31 set to 0 is an address in the process virtual address space. The process space is divided into two equal sized, separately mapped regions. If virtual address bit 30 is set to 0, the address is in region P0. If virtual address bit 30 is set to 1, the address is in region P1.

P0 region address translation— The P0 region of process address space is specified by the P0 page table (P0PT). The P0PT is located in system virtual address and its location and length are defined by the P0 base register (P0BR) and the P0 length register (P0LR), Figure 17. The P0BR contains the system virtual address of the P0 page table. The P0LR contains the size of the P0PT in longwords, that is, the number of page table entries. The page table entry pointed to by the P0 base register maps the first page of the P0 region of the virtual address space, that is, virtual byte address 0.

Figure 18 shows the translation of a P0 virtual address into a physical address.

The algorithm used to generate a physical address from a P0 region virtual address is

$$PVA_PTE = P0BR + 4*PVA<29:09>$$
$$PTE_PA = (SBR + 4*PVA_PTE<29:09>)<20:00> \text{'} PVA_PTE<08:00>$$
$$PROC_PA = (PTE_PA)<20:00> \text{'} PVA<08:00>$$

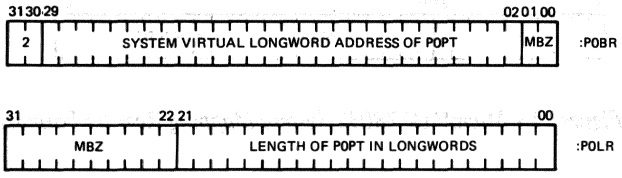


Figure 17 • MicroVAX 78032 P0 Region Mapping Register Formats

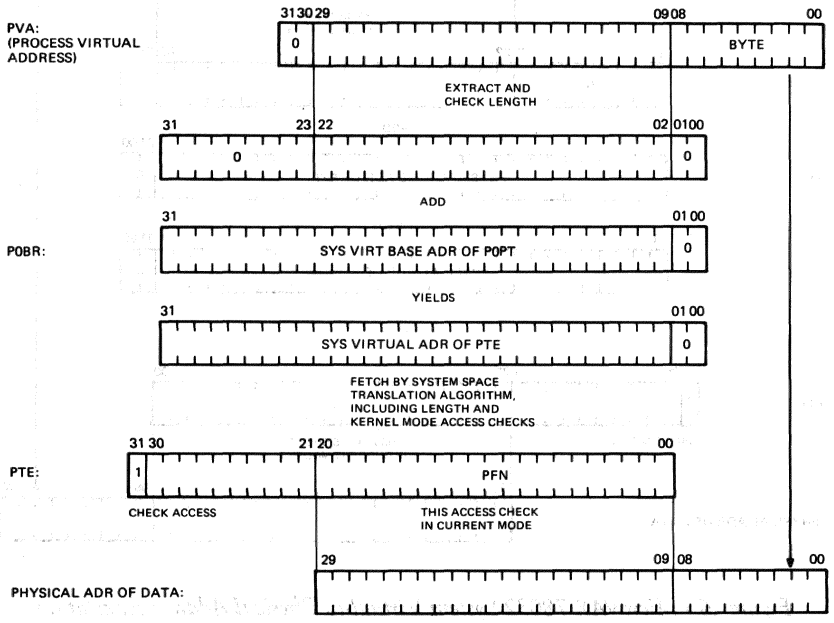


Figure 18 • MicroVAX 78032 P0 Virtual-to-physical Address Translation

P1 region address translation—The P1 region of the address space is specified by the P1 page table (P1PT). The P1PT is located in the system virtual address space and its location and length are defined by the P1 base register (P1BR) and the P1 length register (P1LR), shown in Figure 19. Because the P1 space expands toward smaller addresses, and a consistent hardware interpretation of the base and length registers is desirable, the P1BR and P1LR contain the portion of the P1 space that is not accessible. Note that P1LR contains the number of nonexistent PTEs. P1BR contains the system virtual address of what would be the PTE for the first page of P1, which is the virtual byte address 40000000 (hexadecimal). The address in the P1BR may not be a valid system virtual address; however, all the addresses of PTEs must be valid system virtual addresses. Figure 20 shows the P1 virtual address to physical address translation.

The algorithm used to generate a physical address from a P1 region virtual address is

$$PVA_PTE = P1BR + 4*PVA<29:09>$$
$$PTE_PA = (SBR + 4*PVA_PTE<29:09>)<20:00> \text{'} PVA_PTE<08:00>$$
$$PROC_PA = (PTE_PA)<20:00> \text{'} PVA<08:00>$$

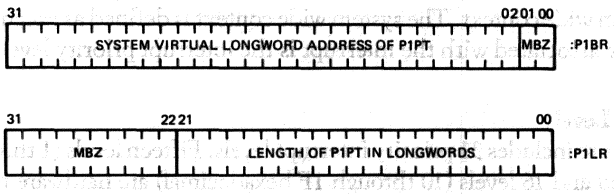


Figure 19 • MicroVAX 78032 P1 Region Mapping Register Formats

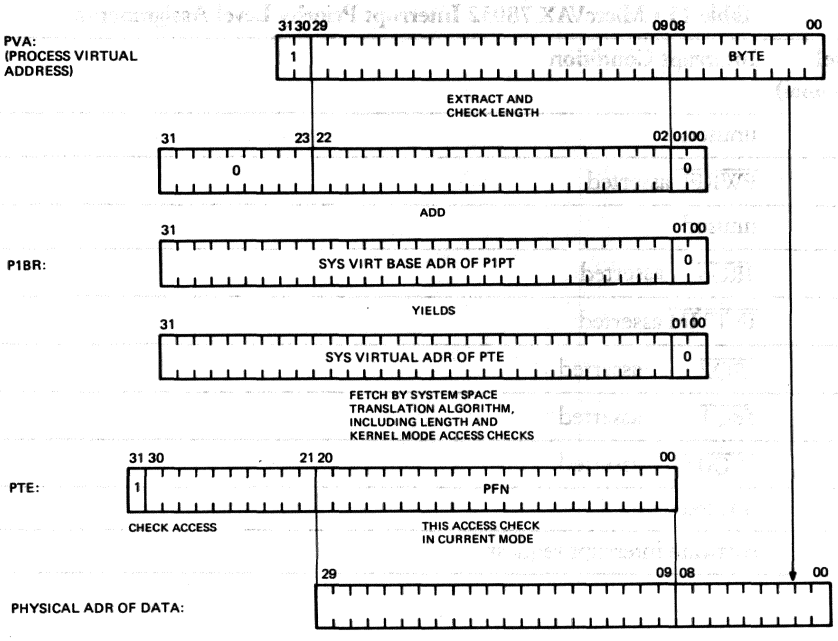


Figure 20 • MicroVAX 78032 P1 Virtual-to-physical Address Translation

Memory Management Faults

The two types of faults associated with memory mapping and protection are translation not valid and access control violation. An access control violation fault exists when the protection field of the PTE indicates that the intended page reference in the specified access mode is illegal. A translation not-valid fault exists when a read or write reference is attempted through an invalid PTE. If an access control violation and a translation not-valid faults occur, the access control takes precedence.

• Exceptions and Interrupts

During system operation, events that are not related to the current process can require service. These events cause the processor to interrupt the process being executed and transfer control to a program that will service the event.

An exception is the notification of an event that is relevant to the currently executing process and normally invokes a program in the context of the executing process.

An interrupt is the notification of an event that is relevant to other processes or to the system and is serviced in a system wide context. The system wide context is defined as executing on the interrupt stack. The priority associated with the interrupt is the interrupt priority level. (IPL).

Interrupt Priority Levels

The VAX architecture includes 31 priority interrupt levels. Fifteen levels (1 through F hexadecimal) are software related and 16 levels (10 through 1F hexadecimal) are hardware related. Table 13 lists the interrupt priority level assignments for the MicroVAX 78032.

Table 13 • MicroVAX 78032 Interrupt Priority Level Assignments

IPL Level (hexadecimal)	Interrupt Condition
1F	unused
1E	$\overline{\text{PWRFL}}$ asserted
18–1D	unused
17	$\overline{\text{IRQ3}}$ asserted
16	$\overline{\text{INTTIM}}$ asserted
16	$\overline{\text{IRQ2}}$ asserted
15	$\overline{\text{IRQ1}}$ asserted
14	$\overline{\text{IRQ0}}$ asserted
10–13	unused
01–0F	software interrupt request

Interrupt Requests

Interrupt requests are serviced during the execution of long interactive instructions such as string instructions and at the completion of an instruction.

Urgent interrupts—Interrupt level 1E (hexadecimal) indicates a powerfail condition and requires immediate service in the MicroVAX 78032.

Device interrupts—Interrupts 14 through 17 (hexadecimal) are assigned to the peripheral devices operating with the MicroVAX 78032.

Software interrupts—Interrupts 1 through F are used by the MicroVAX 78032 system to generate software controlled interrupts.

Interrupt Registers

The interrupt system is controlled by the interrupt priority level register (IPL), the software interrupt request register (SIRR), and the software interrupt summary register (SISR).

Software interrupt summary register—The software interrupt summary register (SISR), shown in Figure 21, is a privileged register that records pending software interrupts. A 1 is set in the bit position corresponding to levels on which software interrupts are pending.

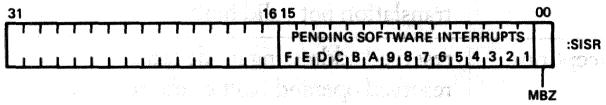


Figure 21 • MicroVAX 78032 Software Interrupt Summary Register Format

Software interrupt request register—The software interrupt request register (SIRR), shown in Figure 22, is a write-only, 4-bit privileged register used for making a software interrupt request. The software requests an interrupt by writing the appropriate interrupt level to the SIRR. Once a software interrupt request is made, the corresponding bit in the SISR is set. The processor will clear the bit in the SISR when the interrupt has been acknowledged.

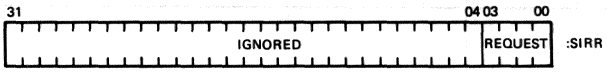


Figure 22 • MicroVAX 78032 Interrupt Request Register Format

Interrupt priority level register—Writing to the IPL register, shown in Figure 23, loads the processor priority field in the processor status longwood (PSL).

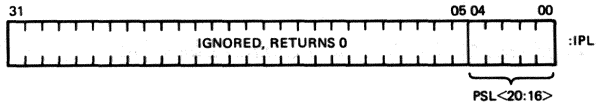


Figure 23 • MicroVAX 78032 Interrupt Priority Level Register Format

• Exceptions

An exception is an event that is the direct result of executing a specific instruction. Exceptions also include errors automatically detected by the processor, such as improperly formed instructions.

The MicroVAX 78032 recognizes the six classes of exceptions summarized in Table 14.

Table 14 • MicroVAX 78032 Classes of Exceptions

Exception Class	Condition
arithmetic traps/faults	integer overflow trap integer divide by zero trap subscript range trap floating overflow fault floating divide by zero fault floating underflow fault
memory management exceptions	access control violation fault translation not valid fault
operand reference exceptions	reserved addressing mode fault reserved operand fault or abort
instruction execution exceptions	reserved/privileged instruction fault emulated instruction fault extended function fault breakpoint fault
tracing exception	trace trap
system failure exceptions	memory read error abort memory write error abort kernel stack not valid abort interrupt stack not valid abort machine check abort

System Control Block

The system control block (SCB) is a page in physical memory that contains the vectors for servicing interrupts and exceptions. Table 15 shows the type and location of the vectors. The SCB is pointed to by the system control block base register (SCBB), Figure 24.

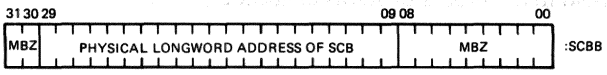


Figure 24 • MicroVAX 78032 System Control Block Base Register Format

Table 15 • MicroVAX 78032 System Control Block Vectors

Vector Address (hexadecimal)	Vector Name	Type
00	unused	abort
04	machine check	abort
08	kernel stack not valid	interrupt
0C	powerfail	fault
10	reserved/privileged instruction	fault
14	extended instruction	fault/abort
18	reserved operand	fault
1C	reserved addressing mode	fault
20	access control violation	fault
24	translation not valid	fault
28	trace pending (TP)	fault
2C	breakpoint instruction	—
30	unused	trap/fault
34	arithmetic	—
38–3C	unused	trap
40	CHMK	trap
44	CHME	trap
48	CHMS	trap
4C	CHMU	—
50–80	unused	interrupt
84	software level 1	interrupt
88	software level 2	interrupt
8C	software level 3	interrupt
90–BC	software levels 4–15	interrupt
C0	interval timer	—
C4	unused	fault
C8	emulation start	fault

Table 15 • MicroVAX 78032 System Control Block Vectors (Cont.)

Vector Address (hexadecimal)	Vector Name	Type
CC	emulation continue	—
D0–FC	unused	interrupt
100–1FC	adapter vectors*	interrupt
200–3FC	device vectors*	

*Used by the MicroVAX 78032 to directly vector interrupts from the external bus. The vector is determined from bits <9:2> of the value supplied by external hardware. If bit <0> of the offset is 1, then the new IPL is forced to 17 hexadecimal. Only device vectors in the range of 100 to 3FC hexadecimal should be used, except by devices emulating console storage and terminal devices.

• Process Structures

A process is the basic entity scheduled by the system software. The context of the current process is contained in the process control block (PCB) shown in Figure 25. The PCB is located in physical memory and is pointed to by the process control block base register (PCBB) shown in Figure 26.

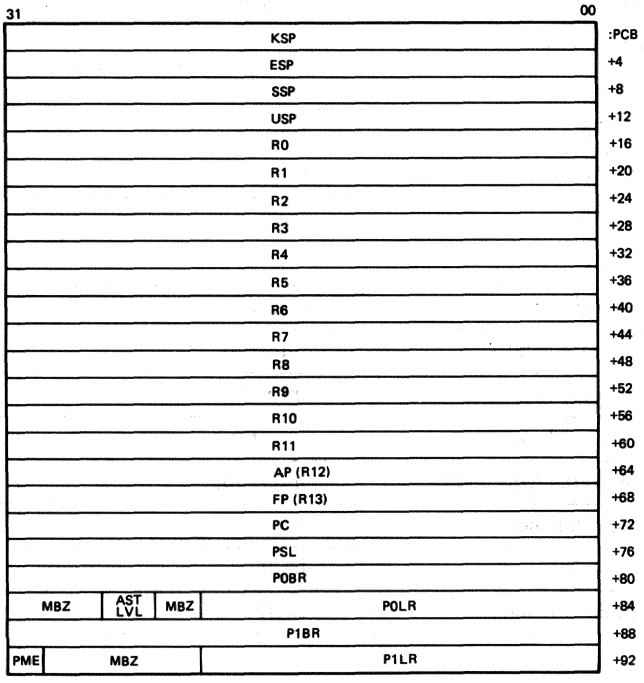


Figure 25 • MicroVAX 78032 Process Control Block Assignments

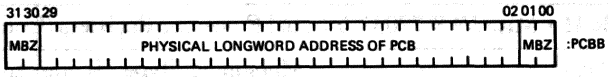


Figure 26 • MicroVAX 78032 Process Control Block Base Register Format

• Processor Registers

The MicroVAX 78032 processor contains many registers that are accessible to the user. These registers are listed in Table 16 and are groups described by the following categories.

1 = Registers implemented by the MicroVAX 78032 as specified by the MicroVAX Architecture.

2 = Registers implemented only by the MicroVAX 78032.

3 = Registers passed to the external logic via the external processor register protocol. If not implemented externally, they are read as zero and result in no operation during a write cycle.

4 = Register access is not allowed (reserved operand fault).

Table 16 • MicroVAX 78032 Internal Processor Registers

Number	Register Name	Mnemonic	Type	Scope	Initialize	Category*
0	Kernel Stack Pointer	KSP	RW	PROC	—	1
1	Executive Stack Pointer	ESP	RW	PROC	—	1
2	Supervisor Stack Pointer	SSP	RW	PROC	—	1
3	User Stack Pointer	USP	RW	PROC	—	1
4	Interrupt Stack Pointer	ISP	RW	CPU	—	1
5	reserved	—	—	—	—	4
6	reserved	—	—	—	—	4
7	reserved	—	—	—	—	4
8	P0 Base Register	P0BR	RW	PROC	—	1
9	P0 Length Register	P0LR	RW	PROC	—	1
10	P1 Base Register	P1BR	RW	PROC	—	1
11	P1 Length Register	P1LR	RW	PROC	—	1
12	System Base Register	SBR	RW	CPU	—	1
13	System Length Register	SLR	RW	CPU	—	1
14	reserved	—	—	—	—	4
15	reserved	—	—	—	—	4
16	Process Control Block Base	PCBB	RW	PROC	—	1

*Refer to Processor Register description.

Table 16 • MicroVAX 78032 Internal Processor Registers (Cont.)

Number	Register Name	Mnemonic	Type	Scope	Initialize	Category*
17	System Control Block Base	SCBB	RW	CPU	—	1
18	Interrupt Priority Level	IPL	RW	CPU	yes	1
19	AST Level	ASTLVL	RW	PROC	yes	1
20	Software Interrupt Request	SIRR	W	CPU	—	1
21	Software Interrupt Summary	SISR	RW	CPU	yes	1
22	Interprocessor Interrupt	IPIR	RW	CPU	—	4
23	CMI Error Register	CMIERR	R	CPU	—	4
24	Interval Clock Control	ICCS	RW	CPU	yes	2
25	Next Interval Count	NICR	W	CPU	—	3
26	Interval Count	ICR	R	CPU	—	3
27	Time Of Year	TODR	RW	CPU	—	3
28	Console Storage Receiver Status	CSRS	RW	CPU	—	3
29	Console Storage Receiver Data	CSRD	R	CPU	—	3
30	Console Storage Transmitter Status	CSTS	RW	CPU	—	3
31	Console Storage Transmitter Data	CSTD	W	CPU	—	3
32	Console Receiver Status	RXCS	RW	CPU	—	3
33	Console Receiver Data	RXDB	R	CPU	—	3
34	Console Transmitter Status	TXCS	RW	CPU	—	3
35	Console Transmitter Data	TXDB	W	CPU	—	3
36	Translation Buffer Disable	TBDR	RW	CPU	—	3
37	Cache Disable	CADR	RW	CPU	—	3
38	Machine Check Error Summary	MCESR	RW	CPU	—	3
39	Cache Error	CAER	RW	CPU	—	3
40	Accelerator Control/Status	ACCS	RW	CPU	—	4
41	Console Saved Interrupt Stack Pointer	SAVISP	R	CPU	—	2
42	Console Saved PC	SAVPC	R	CPU	—	2
43	Console Saved PSL	SAVPSL	R	CPU	—	2
44	WCS Address	WCSA	RW	CPU	—	4
45	WCS Data	WCSD	RW	CPU	—	4
46	reserved	—	—	—	—	4
47	reserved	—	—	—	—	4

*Refer to Processor Register description.

Table 16 • MicroVAX 78032 Internal Processor Registers (Cont.)

Number	Register Name	Mnemonic	Type	Scope	Initialize	Category*
48	SBI Fault/Status	SBIFS	RW	CPU	—	3
49	SBI Silo	SBIS	R	CPU	—	3
50	SBI Silo Comparator	SBISC	RW	CPU	—	3
51	SBI Maintenance	SBIMT	RW	CPU	—	3
52	SBI Error Register	SBIER	RW	CPU	—	3
53	SBI Timeout Address	SBITA	R	CPU	—	3
54	SBI Quadword Clear	SBIQC	W	CPU	—	3
55	IO Bus Reset	IORESET	W	CPU	—	3
56	Memory Management Enable	MAPEN	RW	CPU	yes	1
57	Trans. Buf. Invalidate All	TBIA	W	CPU	—	1
58	Trans. Buf. Invalidate Single	TBIS	W	CPU	—	1
59	Translation Buffer Data	TBDATA	RW	CPU	—	3
60	Microprogram Break	MBRK	RW	CPU	—	3
61	Performance Monitor Enable	PMR	RW	PROC	—	3
62	System Identification	SID	R	CPU	—	1
63	Translation Buffer Check	TBCHK	W	CPU	—	1
64:127	reserved	—	—	—	—	4

*Refer to Processor Register description.

• Interfacing Requirements

The MicroVAX 78032 connects to memory, to external circuits, and to the power source through the connection pins on the package. The following paragraphs define the power, reset, and bus connections and describe the timing considerations for bus operation.

Power Connections

The MicroVAX 78032 requires a single 5 Vdc power supply. Eight pins are provided for power connections; four V_{DD} pins and four V_{SS} pins. The V_{DD} pins connect to 5 V and the V_{SS} pins connect to ground. The power decoupling and grounding is important. Decoupling the power supply is implemented by connecting a capacitor between each V_{DD} pin and its associated V_{SS} pin as shown in Figure 27. The recommended capacitor type is 10 µf tantalum, +1, −10%. The ground pins (V_{SS}) should be connected to the common ground for the power supply at the chip.

The MicroVAX 78032 internally generates the required negative voltage that is externally available on the V_{BB} pin. This voltage does not require filtering and the V_{BB} pin *must not* be connected either to ground or to 5 V.

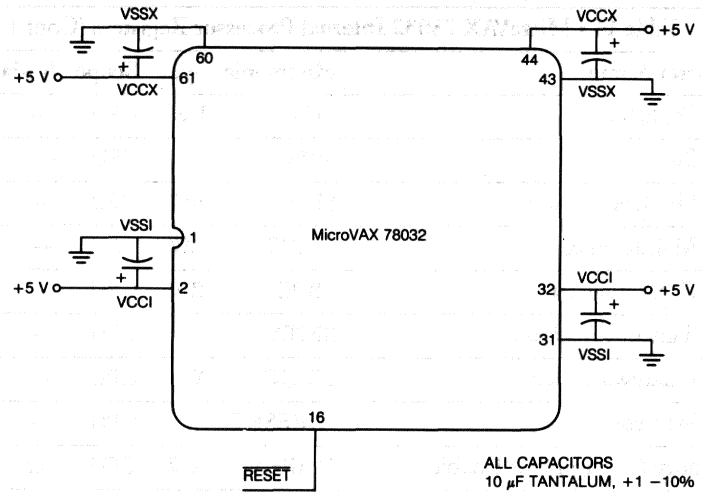


Figure 27 • MicroVAX 78032 Power and Reset Connections

Reset and Powerup Requirements

The MicroVAX 78032 is reset by the following conditions.

1. When power is first applied, the $\overline{\text{RESET}}$ level must be held low for a minimum of 3.0 ms after V_{DD} has reached 4.75 V. To ensure that the internal voltages are stable before an operation begins.
2. The $\overline{\text{RESET}}$ level must be held low for a minimum of 3.0 μs if the $\overline{\text{RESET}}$ level is asserted after V_{DD} has been at 4.75 V for more than 3.0 ms.

When $\overline{\text{RESET}}$ level is asserted, the MicroVAX 78032 stops executing instructions and enters the restart process. The restart process sets the CPU to a known state and then passes control to user code beginning at physical address 20040000 (hexadecimal). For a description of the restart process, refer to the *MicroVAX 78032 Central Processing Unit User's Guide.*

Bus Connections

Figure 28 shows a typical interface configuration of the MicroVAX 78032 and includes control signals and bus connections. The directions of the input and output signal are indicated by the arrows on the lines.

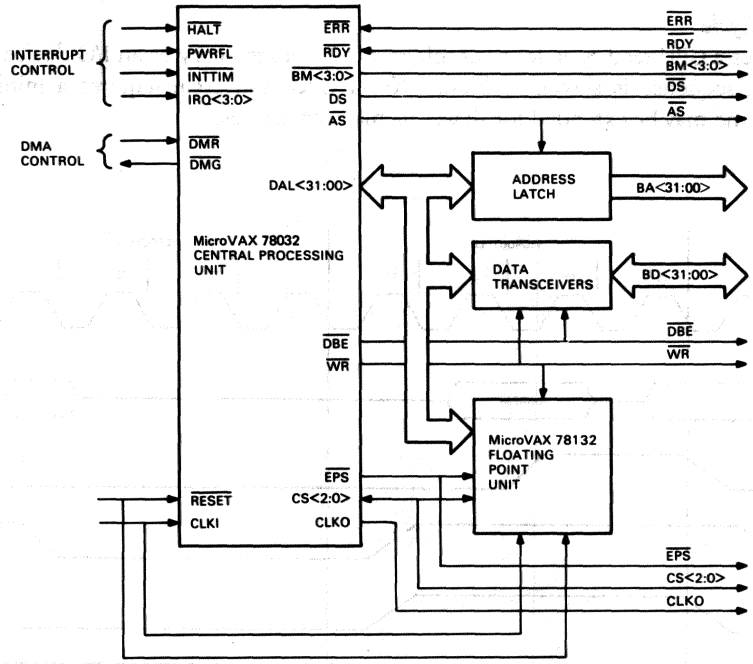


Figure 28 • MicroVAX 78032 Typical Interface Configuration

• Bus Cycles

A bus cycle will be initiated by one of the following conditions:

A microcycle is the basic timing unit for a bus cycle. A microcycle is shown in Figure 29 and is defined as four cycles of CLKO (T1 through T4).

- Reading or writing information from or to memory or a peripheral device.
- Acknowledging an interrupt by reading the device interrupt vector.
- Transferring information from or to an external processor.

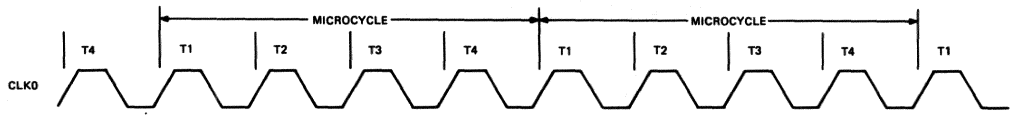


Figure 29 • MicroVAX 78032 Microcycle

CPU Read Cycle

The CPU uses a CPU read cycle to input information from memory or an I/O device. A CPU read cycle timing sequence is shown in Figure 30. A CPU read cycle requires a minimum of 2.0 microcycles and may be extended for slower memory or devices.

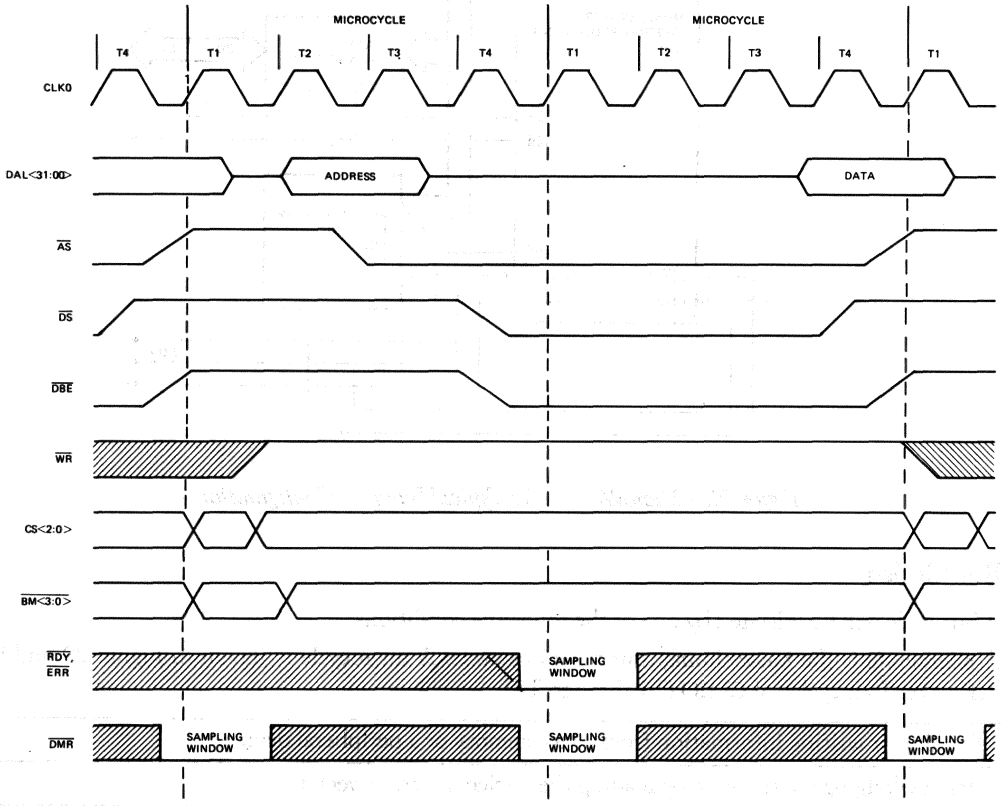


Figure 30 ▪ MicroVAX 78032 CPU Read Cycle Timing Sequence

The first microcycle of a CPU read operation is used to transfer the address and control information and the data is latched into the CPU during the last microcycle.

The sequence of events for a CPU read operation follows:

1. The physical (longword) address is driven onto DAL <29:02> and the memory operand length onto DAL<31:30> by the CPU.
2. The $\overline{WR}$ signal is unasserted and CS<2:0> are asserted as required to indicate the type of bus cycle being performed.
3. The $\overline{BM}$<3:0> lines are asserted as required.
4. The $\overline{AS}$ signal is asserted to indicate that the address is valid and can be latched for demultiplexing and to qualify CS<2:0> and $\overline{BM}$<3:0> information.
5. The $\overline{DS}$ signal is asserted to indicate that the bus is free to receive the requested information. The $\overline{DBE}$ signal is also asserted at this time and can be used to control the DAL bus transceivers.
6. If the requested data is valid, it can be placed on the bus during T3 of the next microcycle, the external logic asserts the $\overline{RDY}$ signal, and the microcycle that follows is the last for this bus cycle. If the $\overline{RDY}$ signal is not asserted by the end of the current microcycle, the bus cycle will be extended by one microcycle.

   If a bus error occurs, external logic responds by asserting the $\overline{ERR}$ signal. If $\overline{ERR}$ is asserted during a data read, the CPU ignores the data on DAL<31:00>, extends the bus cycle by one microcycle, and initiates a machine check. If the $\overline{ERR}$ signal is asserted during an instruction read with CS<2:0> = 100, the CPU stops prefetching and when the instruction buffer is empty, the CPU will attempt to fetch the next instruction byte with a data read cycle. The $\overline{ERR}$ signal takes precedence over the $\overline{RDY}$ signal. The assertion of either $\overline{RDY}$ or the $\overline{ERR}$ signals results in the completion of the current bus cycle.
7. The requested data is latched into the CPU and the $\overline{DS}$ signal is deasserted.
8. The $\overline{AS}$ and $\overline{DBE}$ signals are deasserted to end the bus cycle.

CPU Write Cycle

The CPU uses a CPU write cycle to transfer information to memory or to an I/O device. A CPU write cycle, shown in Figure 31, requires a minimum of 2 microcycles and may be extended for slower memory or devices.

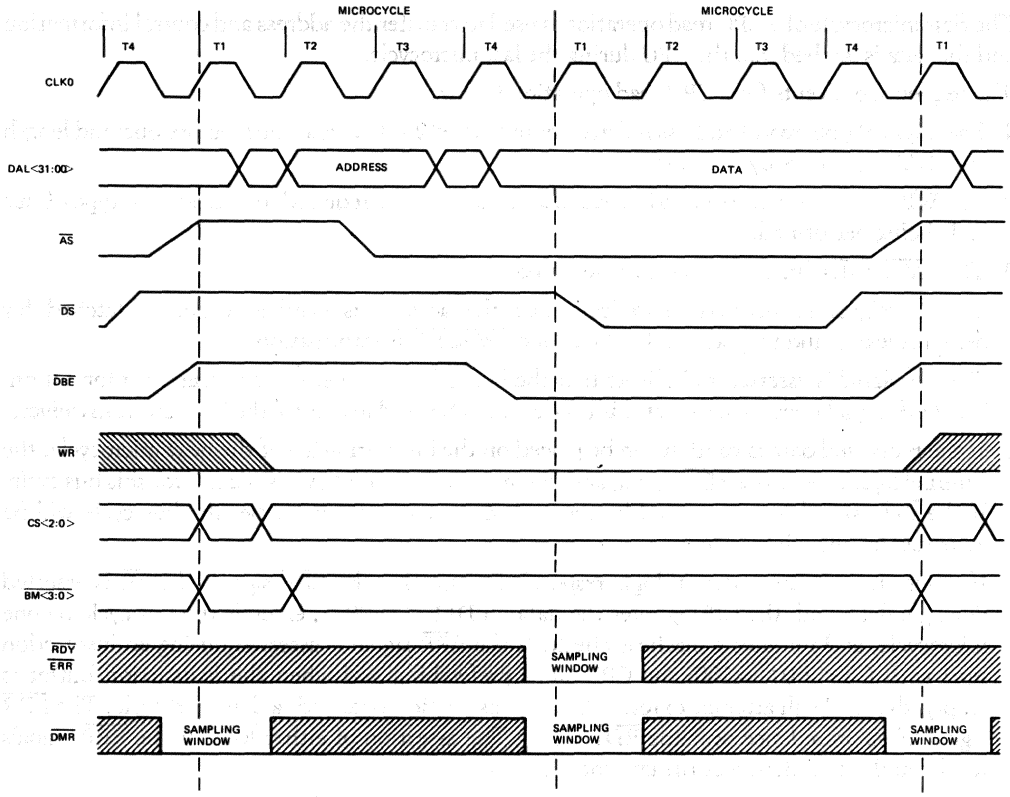


Figure 31 ▪ MicroVAX 78032 CPU Write Cycle Timing Sequence

The first microcycle of a CPU write operation is used to transfer the address and control information and the valid data is written during the second microcycle.

The sequence of events for a CPU write operation follows:

1. The physical (longword) address is driven onto DAL<29:02> and the memory operand length is onto DAL<31:30> by the CPU.
2. The $\overline{WR}$ signal is asserted and CS<2:0> lines are asserted as required.
3. The $\overline{BM}$<3:0> lines are asserted as required.
4. The $\overline{AS}$ signal is asserted to indicate that the address is valid and can be latched for demultiplexing and to qualify the CS<2:0> and $\overline{BM}$<3:0> information.
5. The $\overline{DBE}$ signal is asserted and can be used to control the DAL bus transceivers.
6. The CPU drives data onto the DAL bus and asserts the $\overline{DS}$ signal to indicate that the data is valid.
7. If the data can be read during the next microcycle, the external logic asserts the $\overline{RDY}$ signal and the following microcycle is the last for this bus cycle. If the $\overline{RDY}$ signal is not asserted by the end of the current microcycle, the bus cycle will be extended by one microcycle.

   If a bus error occurs, external logic responds by asserting the $\overline{ERR}$ signal and the CPU initiates a machine check. The $\overline{ERR}$ signal takes precedence over the $\overline{RDY}$ signal.

   The assertion of either the $\overline{RDY}$ or $\overline{ERR}$ signals results in the completion of the current bus cycle.

8. The $\overline{DS}$ signal is deasserted to indicate that the data will be removed from the DAL bus by the CPU.
9. The $\overline{AS}$ and $\overline{DBE}$ signals are deasserted to end the bus cycle.

Interrupt Acknowledge Cycle

An interrupt acknowledge cycle is used to acknowledge an interrupt request from an I/O device, and to read a vector. The structure of this cycle is the same as a CPU read cycle shown in Figure 30.

The first microcycle of an interrupt acknowledge cycle is used to transfer the interrupt priority level (IPL) that is being acknowledged and the interrupt vector from the interrupting device is latched into the CPU during the last microcycle.

The sequence of events for an interrupt acknowledge cycle is as follows:

1. The CPU places the IPL of the interrupt being acknowledged on DAL<04:00>. DAL<29:05> lines are zero and the DAL<31:30> lines are=10.
2. Lines CS<2:0> are asserted to indicate an interrupt acknowledge cycle.
3. Lines $\overline{BM<3:0>}$ are all asserted and the $\overline{WR}$ signal is unasserted.
4. The $\overline{AS}$ signal is asserted to indicate that the IPL level on the DAL<04:00> lines is valid.
5. The $\overline{DS}$ is asserted to indicate that the bus can receive incoming data. The $\overline{DBE}$ signal is also asserted at this time and can be used to control the DAL bus transceivers.
6. If no error occurs, the external logic responds by placing the interrupt vector on the DAL<09:02> lines the normal Q-bus processing flag on DAL<00>, and by asserting the $\overline{RDY}$ signal. The DAL<15:10,01> lines *must* be a high or low level in accordance with the setup times specified in the timing diagrams.
7. If an error occurs, the external logic asserts the $\overline{ERR}$ signal and the CPU cancels the cycle and ignores the data on the DAL bus.
8. The interrupt vector is latched into the CPU and the $\overline{DS}$ signal is deasserted.
9. The $\overline{AS}$ and $\overline{DBE}$ signals are deasserted to end the bus cycle.

DMA Cycle

A DMA cycle shown in Figure 32, is used by the CPU to relinquish control of the DAL bus and related control signals upon request from a DMA device or another CPU.

The sequence of events for a DMA cycle is

1. The DMA device requests use of the bus by asserting the $\overline{DMR}$ signal.
2. The CPU samples the $\overline{DMR}$ line for a DMA request during each microcycle unless the current bus cycle is a read lock cycle.
3. The CPU causes the DAL<31:00>, $\overline{AS}$, $\overline{DS}$, $\overline{DBE}$, $\overline{WR}$, $\overline{BM<3:0>}$, and CS<2:0> lines to become a high-impedance and asserts the $\overline{DMG}$ line to grant the DMA device use of the DAL bus.
4. When the requesting device is finished using the bus, it deasserts the $\overline{DMR}$ signal, and the CPU takes control of the bus.

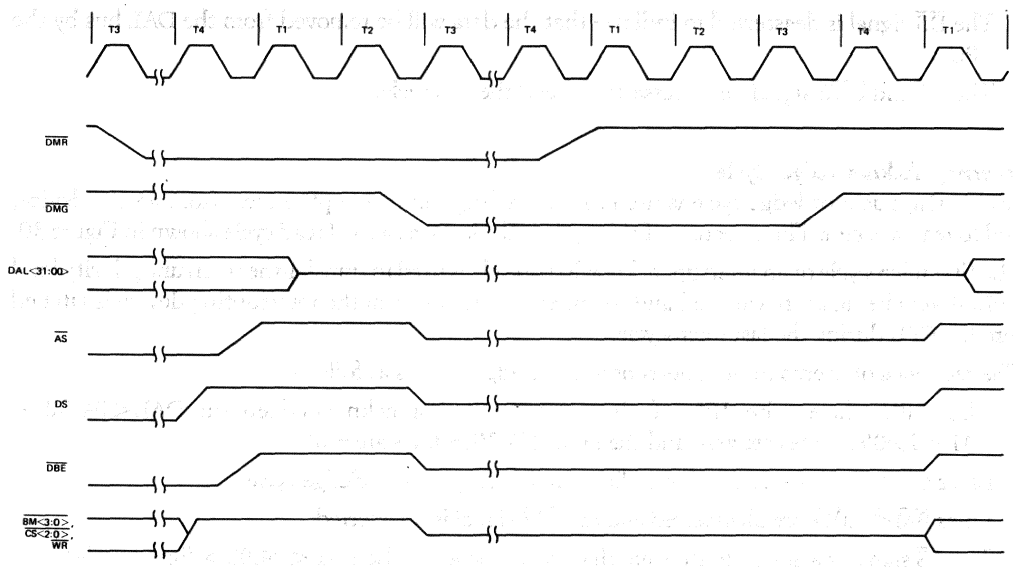


Figure 32 • MicroVAX 78032 DMA Cycle Timing Sequence

• External Processor Cycles

The CPU uses external processor cycles to communicate with external processors and external processor registers.

External Processor Read Cycle

The external processor read cycle shown in Figure 33 is used to transfer information from an external processor or external processor register to the CPU. An external processor read cycle requires one microcycle.

The sequence of events for an external processor read cycle is:

1. The CS<1:0> lines are asserted as required and the CS2 line is sustained at a high level.
2. The $\overline{WR}$ signal is not asserted for a read cycle.
3. The $\overline{EPS}$ signal is asserted to indicate that an external processor bus cycle is in process and to qualify the CS<2:0> lines.
4. The external processor places the requested information on the DAL.
5. The requested information is latched into the CPU and the $\overline{EPS}$ line is deasserted.
6. The external processor removes its information from the DAL bus to end the bus cycle.

External Processor Response Cycle

The external processor response cycle shown in Figure 33 is used to transfer information and a completion or confirmation signal from an external processor or external processor register to the CPU. An external processor response cycle requires one microcycle.

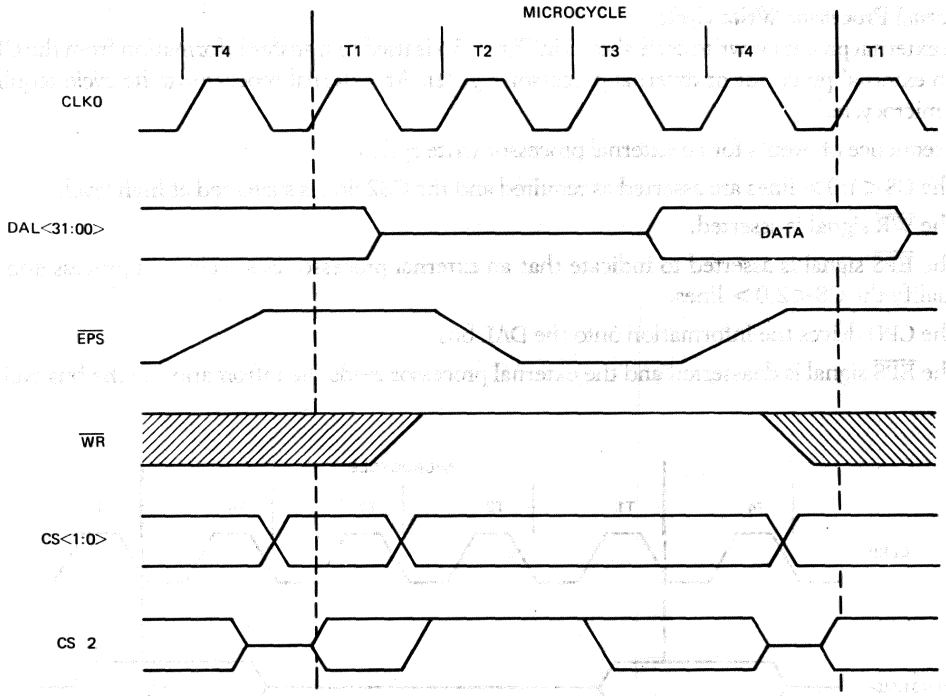


Figure 33 • MicroVAX 78032 External Processor Read/Response Cycle Timing Sequence

The sequence of events for an external processor response cycle is:

1. The CS<1:0> lines are asserted as required and the CS2 line is sustained at high level.
2. The $\overline{WR}$ signal is not asserted for a read cycle.
3. The $\overline{EPS}$ signal is asserted to indicate that an external processor bus cycle is in process and to qualify the CS<2:0> lines.
4. The external processor places the requested information on the DAL bus and optionally drives the CS2 line low.
5. The requested information is latched into the CPU and the $\overline{EPS}$ signal is deasserted.
6. The external processor removes its information from the DAL bus and deasserts CS2, if asserted, to end the bus cycle.

External Processor Write Cycle

The external processor write cycle shown in Figure 34 is used to transfer information from the CPU to an external processor or external processor register. An external processor write cycle requires one microcycle.

The sequence of events for an external processor write cycle is:

1. The CS<1:0> lines are asserted as required and the CS2 line is sustained at high level.
2. The $\overline{WR}$ signal is asserted.
3. The $\overline{EPS}$ signal is asserted to indicate that an external processor bus cycle is in process and to qualify the CS<2:0> lines.
4. The CPU drives the information onto the DAL bus.
5. The $\overline{EPS}$ signal is deasserted and the external processor reads the infromation to the bus cycle.

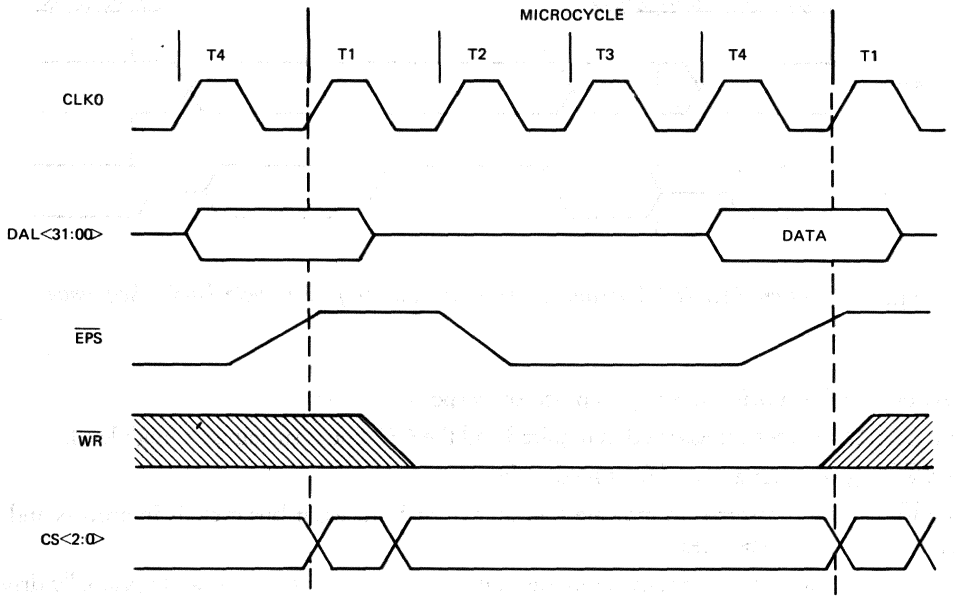


Figure 34 ▪ MicroVAX 78032 External Processor Write Cycle Timing Sequence

▪ Memory Access Protocol

The 28-bit address provided by the MicroVAX 78032 on DAL<29:02> is a longword address that uniquely identifies one of up to 268,435,456 32-bit memory locations. The chip provides four-byte masks, BM<3:0>, to select byte accesses within the 32-bit memory locations. No restrictions exist on data alignment. The data may start at any memory address except for the aligned operands of ADAWI instruction and the interlocked queue instructions.

The memory consists of four parallel 8-bit banks, each of which receive the longword address on the DAL<29:02> lines in parallel. Each bank reads or writes one byte of the data bus (DAL<31:00>), when its byte mask signal is asserted as shown in Figure 35.

CPU read or write operations are grouped into one of the following categories—byte access, word access within a longword, word access across longwords, aligned longword access, and unaligned longword access. Quadword accesses are treated as two successive longword accesses, with no optimization. Byte accesses, word accesses within a longword, and aligned longword accesses require one bus cycle. Unaligned longword accesses and word accesses that cross a longword boundary require two bus cycles.

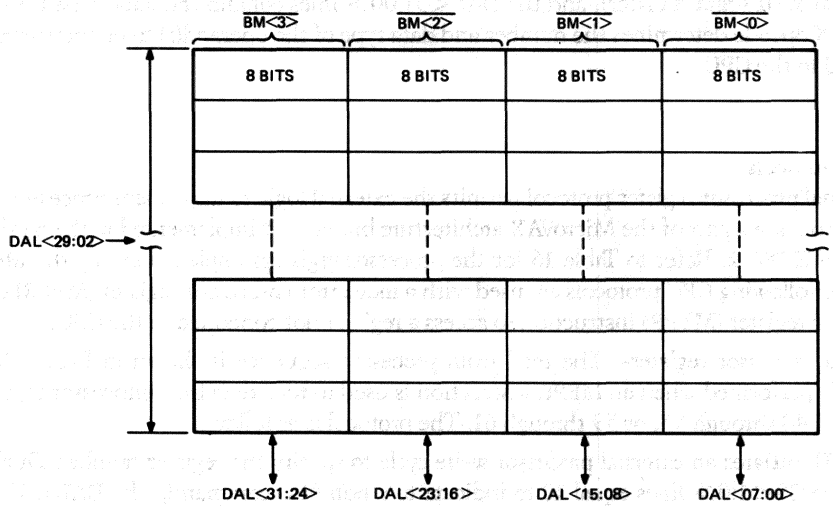


Figure 35 • MicroVAX 78032 Memory Organization

• External Processor Protocols

External processor protocols allow the MicroVAX 78032 to communicate efficiently with one or more external processors. Two external processor protocols exist—one for communicating with the optional floating-point unit and the second for communicating with processor register logic.

Floating-Point Unit Protocols

The optional floating-point unit (FPU) is controlled by the CPU. When the CPU receives a floating-point instruction, it passes the opcode and operands to the FPU for processing. The CPU waits for the FPU to complete the operation and then requests status information and the processing results. The FPU protocol is as follows:

1. Command transfer—The CPU performs an external processor write cycle to transmit a command to the FPU. During this cycle, the CS<1:0> contains 00 indicating a FPU command and the opcode of the floating-point instruction is placed on the DAL<08:00> lines.
2. Operand transfer—The VAX opcode determines the number and data type of operands to be transferred from the CPU to the FPU. The CPU performs one or more external processor write cycles to transfer the operands. During these cycles, the CS<1:0> lines are equal to 01 (data transfer), and the DAL<31:00> lines contain the data to be transferred.

3. Operand processing—While the FPU is processing the operands, the CPU checks to determine where the operation is completed by executing external processor response enable cycles.
4. Status transfer—When the FPU has finished processing the operands, it responds to the next external processor response enable cycle by placing status information on the DAL<05:00> lines and by driving the CS2 bus low. The CPU responds to the CS2 low signal by reading the status information on the DAL<05:00> lines.
5. Result transfer—After reading the status code, the CPU may initiate one or more external processor read cycles to transfer the result operand(s). During these cycles, the CS<1:0> lines are equal to 01 (data transfer), and the DAL<31:00> lines contain the data to be transferred. The VAX opcode determines the number and data type of the operand(s) to be transferred from the FPU to the CPU.

Register Protocols

The external processor register protocol permits the external logic to implement processor register functions that are a part of the MicroVAX architecture but are not implemented in the hardware of the MicroVAX 78032. Refer to Table 16 for the processor registers implemented by the MicroVAX 78032. The following CPU protocols are used with a move from processor register (MFPR) or move to processor register (MTPR) instruction to access a register not contained in the CPU.

Read from processor register—The read from processor sequence is shown in Figure 36. This sequence is performed when an MFPR instruction is used to read data from processor registers 25 through 39, 48 through 55, or 59 through 61. The protocol is as follows:

1. The CPU initiates an external processor write cycle to specify the register number. During this cycle, the CS<1:0> lines equal 10 to indicate to a non-FPU command, the DAL<31> lines equal 1 (read register), and the DAL<05:00> lines contain the register number specified by the MFPR instruction.
2. The CPU waits one cycle and then executes an external processor response cycle to read the register data. If the CS2 line is driven low by the external logic, the data on the DAL<31:00> lines is the result of the MFPR instruction. If the CS2 line is high, the CPU returns zero as the result.

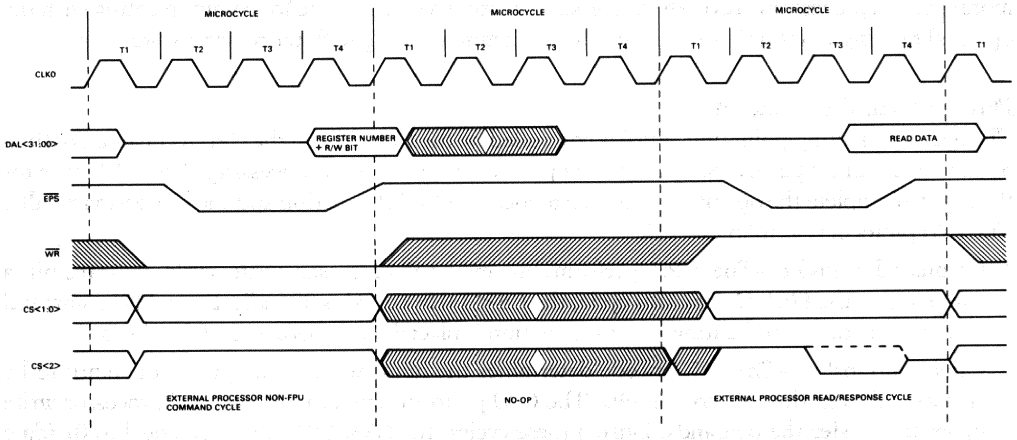


Figure 36 • MicroVAX 78032 Read from Processor Register Timing Sequence

Write to processor register—The write to processor register sequence is shown in Figure 37. This sequence is performed when an MTPR instruction is used to write data to processor registers 25 through 39, 48 through 55, or 59 through 61. The protocol is as follows:

1. The CPU initiates an external processor write cycle to specify the register number. During this cycle, the CS<1:0> lines equal 10 indicating a non-FPU command, the DAL31 lines equal 0 (write register), and the DAL<05:00> lines contain the register number specified by the MTPR instruction.
2. The CPU executes an external processor write cycle to write the register data. During this cycle, the CS<1:0> lines equal 01 (write data), and the DAL<31:00> lines contain the data specified in the MTPR instruction.
3. The next cycle is not an external processor cycle.

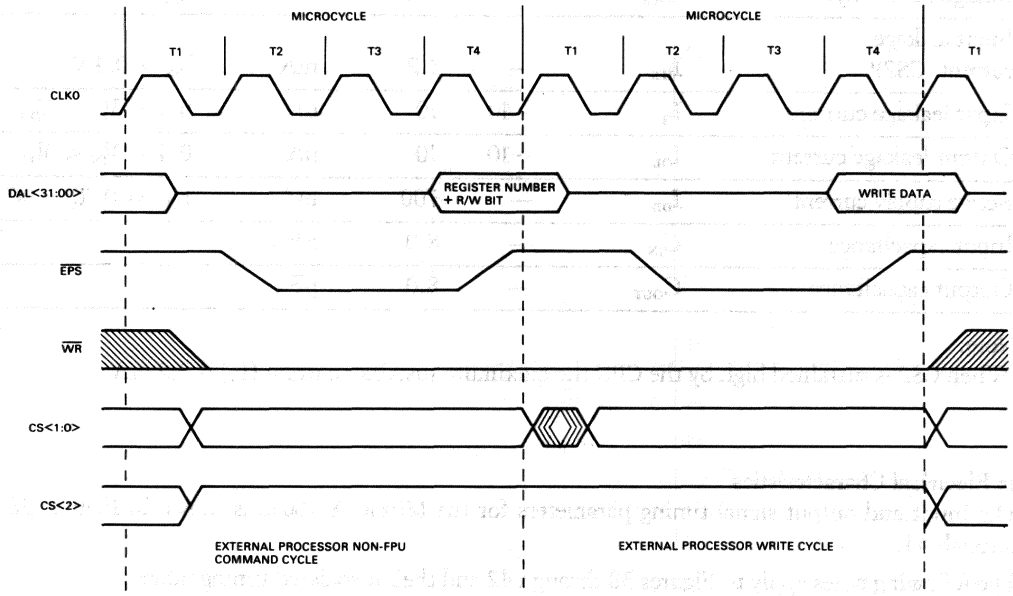


Figure 37 • MicroVAX 78032 Write to Processor Register Timing Sequence

dc Electrical Characteristics

The dc electrical characteristics of the MicroVAX 78032 for the operating voltage and temperature ranges specified are listed in Table 17.

Table 17 • MicroVAX 78032 dc Input and Output Parameters

Parameter	Symbol	Requirements min	Requirements max	Units	Test Condition
High-level input voltage	V_{IH}	2.0		V	
Low-level input voltage	V_{IL}		0.8	V	
High-level output voltage	V_{OH}	2.4	—	V	$I_{OH} = -400\ \mu A$
Low-level output voltage	V_{OL}		0.4	V	$I_{OL} = 2.0$ mA
High-level output voltage (EPS only)	V_{OHE}	2.6	—	V	$I_{OH} = -100\ \mu A$
Low-level output voltage (EPS only)	V_{OLE}	—	0.2	V	$I_{OL} = 1.0$ mA
Input leakage current (CS2)[1]	I_{ILS}	—	3.2	mA	$V_{IN} = 0.4$ V
Input leakage current	I_{IL}	−10	10	μA	$0.4 < V_{IN} < V_{DD}$
Output leakage current	I_{OL}	−10	10	μA	$0.4 < V_{IN} < V_{DD}$
Active supply current	I_{DD}	—	700	mA	$I_{OUT} = 0$, $T_A = 0C$
Input capacitance	C_{IN}	—	8.0	pF	
Output capacitance	C_{OUT}	—	8.0	pF	

Note:
[1]When CS2 is sustained high by the CPU the maximum sustainer current (I_{IL}) is 3.2 mA.

ac Electrical Characteristics

The input and output signal timing parameters for the MicroVAX 78032 is shown in Figures 38 through 43.

The following notes apply to Figures 38 through 42 and their associated timing tables.

1. Formulas for the timing parameters are stated in terms of the CLKI period. CLKI period = t_{CIP} = P.
2. All times are in nanoseconds except where noted.
3. The ac characteristics are measured with a purely capacitive load of 100 pF. Times are valid for loads of up to 100 pF on all pins.
4. ac high levels are measured at 2.0 volts and ac low levels at 0.8 volts except for the $\overline{EPS}$ and TEST signals.
5. An ac high level for the $\overline{EPS}$ and TEST signals are measured at 2.2 volts and an ac low level at 0.6 volts.
6. S = the number of microcycles slipped during a bus cycle.
7. The sampling window is used to sample the following asynchronous signals: $\overline{RDY}$, $\overline{ERR}$, and $\overline{DMR}$. The $\overline{RDY}$ and $\overline{ERR}$ signals are qualified when $\overline{AS}$ is asserted. The $\overline{DMR}$ signal is qualified by the $\overline{AS}$ signal being deasserted. The effect of these signals on the current bus cycle is as follows:

- The bus cycle will conclude at the end of the current microcycle if the $\overline{RDY}$ signal is asserted and the $\overline{ERR}$ signal is not asserted throughout the sampling window while the $\overline{AS}$ signal is asserted.
- If the $\overline{ERR}$ signal is asserted throughout the sampling window while the $\overline{AS}$ signal is asserted, the current microcycle becomes an extension cycle and the bus cycle ends after the next microcycle.
- If the $\overline{RDY}$ or $\overline{ERR}$ signals go through a transition during the sampling window while the $\overline{AS}$ signal is asserted, the result is indeterminate.
- The $\overline{DMR}$ signal is sampled at every microcycle boundary.
- If the $\overline{DMR}$ signal is asserted throughout the sampling window and the $\overline{AS}$ signal is not asserted, and the CPU has not locked the bus, the next microcycle will be the beginning of a DMA cycle.
- The first microcycle after the end of the current bus cycle will begin a DMA cycle if the $\overline{DMR}$ signal is asserted throughout the sampling window, the $\overline{AS}$ signal is asserted, and the CPU has not locked the bus.
- A DMA cycle concludes at the end of the current microcycle if the $\overline{DMR}$ signal, is deasserted throughout the sampling window.

8. There are no internal pull-up circuits on the $\overline{IRQ<3:0>}$, $\overline{PWRFL}$, $\overline{INTTIM}$, and $\overline{HALT}$ lines.

Specifications

The mechanical, electrical, and environmental characteristics and specifications for the MicroVAX 78032 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Operating temperature (T_A): 70°C
- Ground reference (V_{SS}): 0 V
- Supply voltage (V_{DD}): 4.75 V

Mechanical Configuration

The physical dimensions of the MicroVAX 78032 68-pin cerquad package are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Supply voltage (V_{DD}): –0.5 V to 7.0 V
- Input or output voltage applied: –0.5 V to 7.0 V
- Active temperature (T_A): 0°C to 70°C
- Storage temperature (T_S): –55°C to 125°C
- Power dissipation: 3.5 watts (maximum)

Recommended Operating Conditions

- Supply voltage (V_{DD}): 4.75 V to 5.25 V
- Active supply current: (I_{DD}): 700 mA (maximum)
- Temperature (T_A): 0°C to 60°C
- Relative humidity: 10% to 95% (noncondensing)
- Minimum airflow over chip: 250 linear feet/minute

Clock Input Timing

Figure 38 shows the timing specifications for the CLKI input clock signal and Table 18 lists the timing parameters indicated on the diagram.

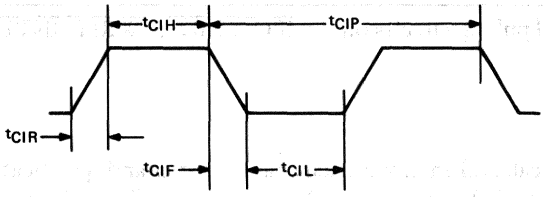


Figure 38 • MicroVAX 78032 CLK1 Timing Waveform

Table 18 • MicroVAX 78032 CLKI Timing Parameters

Timing Symbol	Signal Definition	Requirements (ns) min	max
t_{CIF}	Clock in fall time	—	4.5
t_{CIH}	Clock in high	8	—
t_{CIL}	Clock in low	8	—
t_{CIP}	Clock period	25	50
t_{CIR}	Clock in rise time	—	4.5

CPU Read and Write Cycle Timing

Figure 39 shows the timing sequence for the CPU read cycle and Figure 40 shows the timing sequence for the CPU write cycle. The parameters for the CPU read and write cycles are listed in Table 19.

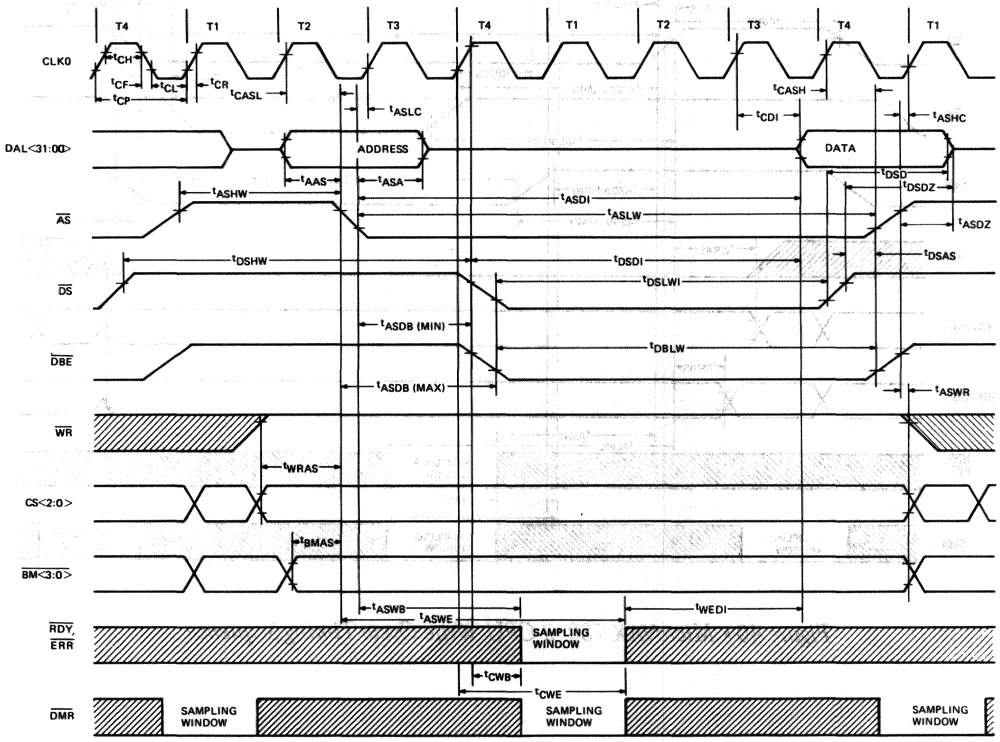


Figure 39 • MicroVAX 78032 CPU Read Cycle Timing Sequence

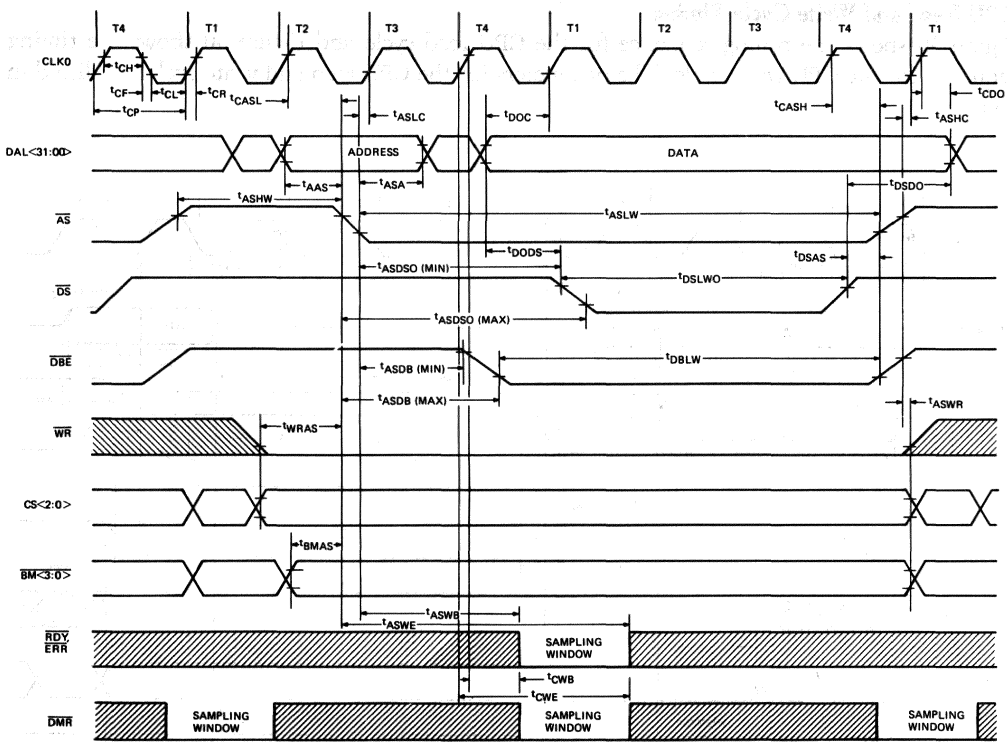


Figure 40 • MicroVAX 78032 CPU Write Cycle Timing Sequence

Table 19 • MicroVAX 78032 CPU and Write Cycle Parameters

Timing Symbol	Signal Definition	Requirements (ns) min	max
t_{AAS}	Address set up time to $\overline{AS}$ assertion	2P–28	—
t_{ASA}	Address hold time after $\overline{AS}$ assertion	2P–15	—
t_{ASHC}	$\overline{AS}$ rising through 2.0 V to CLKO rising through 0.8 V	P–23	—
t_{ASLC}	$\overline{AS}$ falling through 0.8 V to CLKO rising through 0.8 V	P–20	—
t_{ASDB}	$\overline{AS}$ assertion to $\overline{DBE}$ and $\overline{DS}$ (read) assertion	3P–15	3P+20
t_{ASDI}	$\overline{AS}$ assertion to read data valid[1]		11P–30+8PS
t_{ASDSO}	$\overline{AS}$ assertion to $\overline{DS}$ assertion (write)	5P–15	5P+20
t_{ASDZ}	$\overline{AS}$ and $\overline{DBE}$ deassertion to data three-state	—	2P–20
t_{ASHW}	$\overline{AS}$ deassertion width	3P	—
t_{ASLW}	$\overline{AS}$ assertion width	12P–15+8PS	
t_{ASWB}	$\overline{AS}$ assertion to beginning of $\overline{RDY}$, $\overline{ERR}$, and $\overline{DMR}$ sampling window[2]	—	(6P–45)+8PS
t_{ASWE}	$\overline{AS}$ assertion to end of $\overline{RDY}$, $\overline{ERR}$, and $\overline{DMR}$ sampling window[3]	6P+10+8PS	—
t_{ASWR}	$\overline{WR}$, $\overline{BM<3:0>}$, CS<2:0> hold time from $\overline{AS}$ deassertion	P–20	—
t_{BMAS}	$\overline{BM<3:0>}$ set up time before $\overline{AS}$ assertion	2P–25	—
t_{CASH}	CLKO rising through 2.0 V to $\overline{AS}$ rising through 0.8 V	P–7	P+15
t_{CASL}	CLKO rising through 2.0 V to $\overline{AS}$ falling through 2.0 V	P–9	P+16
t_{CDI}	CLKO rising through 2.0 V to read data valid	—	P–5
t_{CDO}	Write data hold time from CLKO rising through 2.0 V	P–15	—
t_{CF}	CLKO fall time	—	12.5
t_{CH}	CLKO high	(2P–25) x .5	—
t_{CL}	CLKO low	(2P–25) x .5	—
t_{CP}	CLKO period	50	100
t_{CR}	CLKO rise time	—	12.5
t_{CWB}	T4 CLKO rising through 2.0 V to beginning of $\overline{RDY}$, $\overline{ERR}$, and $\overline{DMR}$ sampling window[2]	—	3P–45
t_{CWE}	T4 CLKO rising through 0.8 V to end of $\overline{RDY}$, $\overline{ERR}$, and $\overline{DMR}$ sampling window[3]	3P+15	—

Table 19 • MicroVAX 78032 CPU and Write Cycle Parameters (Cont.)

Timing Symbol	Signal Definition	Requirements (ns) min	max
t_{DBLW}	$\overline{DBE}$ assertion width	9P–20+8PS	—
t_{DOC}	Write data set-up time to CLKO rising through 0.8 V	3P–42	—
t_{DODS}	Write data set-up time to $\overline{DS}$ assertion	3P–30	—
t_{DSAS}	$\overline{DS}$ deassertion to $\overline{AS}$ and $\overline{DBE}$ deassertion	P–15	—
t_{DSD}	Read data hold time after $\overline{DS}$ deassertion	0	—
t_{DSDI}	$\overline{DS}$ assertion to read data valid[1]	—	8P–35+8PS
t_{DSDO}	Write data hold time from $\overline{DS}$ deassertion	3P–20	—
t_{DSDZ}	$\overline{DS}$ deassertion to read data high impedence	—	3P–20
t_{DSHW}	$\overline{DS}$ deassertion width	6P	—
t_{DSLWI}	$\overline{DS}$ assertion width (read)	8P–20+8PS	—
t_{DSLWO}	$\overline{DS}$ assertion width (write)	6P–20+8PS	—
t_{WEDI}	Sampling window end to read data valid	—	5P–25
t_{WRAS}	$\overline{WR}$, CS<2:0> set up time before $\overline{AS}$ assertion	3P–35	—

Notes:

[1] Read data is valid early enough if t_{ASDI} or t_{DSDI} or t_{CDI} is satisfied.

[2] Requirements for the beginning of the sampling window are satisfied if either t_{ASWB} or t_{CWB} is satisfied.

[3] Requirements for the end of the sampling window are satisfied if either t_{ASWE} or t_{CWE} is satisfied.

Direct Memory Access Cycle Timing

Figure 41 shows the timing sequence for direct memory access (DMA) transfers and Table 20 lists the timing parameters for the symbols referenced on the diagram.

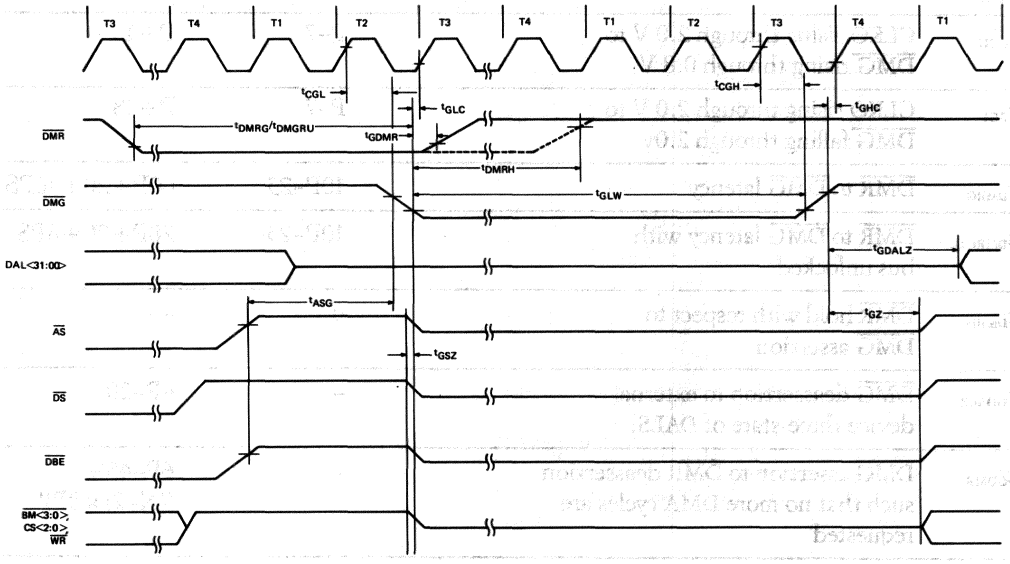


Figure 41 ▪ DMA Timing Sequence

Table 20 • MicroVAX 78032 DMA Cycle Timing Parameters

Timing Symbol	Signal Definition	Requirements (ns) min	max
t_{ASG}	$\overline{AS}$ and $\overline{DBE}$ deassertion to $\overline{DMG}$ assertion	4P–25	—
t_{CGH}	CLKO rising through 2.0 V to $\overline{DMG}$ rising through 0.8 V	P–7	P+16
t_{CGL}	CLKO rising through 2.0 V to $\overline{DMG}$ falling through 2.0v	P–7	P+18
t_{DMRG}	$\overline{DMR}$ to $\overline{DMG}$ latency	10P–25	60P+20+16PS
t_{DMRGU}	$\overline{DMR}$ to $\overline{DMG}$ latency with bus unlocked	10P–25	28P+20+8PS
t_{DMRH}	$\overline{DMR}$ hold with respect to $\overline{DMG}$ assertion	0	—
t_{GDALZ}	$\overline{DMG}$ deassertion to external device three-state of DALS.	—	4P–20
t_{GDMR}	$\overline{DMG}$ assertion to $\overline{DMR}$ deassertion such that no more DMA cycles are requested	—	6P–45+ ((N–2)×8P)[1]
t_{GHC}	$\overline{DMG}$ rising through 2.0 V to CLKO rising through 0.8 V	P–25	—
t_{GLC}	$\overline{DMG}$ falling through 0.8 V to CLKO rising through 0.8 V	P–23	—
t_{GLW}	$\overline{DMG}$ minimum assertion width	10P–25+ ((N–2)×8P)[1]	—
t_{GSZ}	$\overline{DMG}$ assertion to three-state of $\overline{AS}$, $\overline{DS}$, $\overline{DBE}$, $\overline{WR}$. CS<2:0> and $\overline{BM}$ <3:0>	–10	0
t_{GZ}	$\overline{DMG}$ deassertion to external device of three-state of $\overline{AS}$, $\overline{DS}$, $\overline{DBE}$, $\overline{WR}$, CS<2:0><3:0> and $\overline{BM}$ <3:0>		3P–20[2]

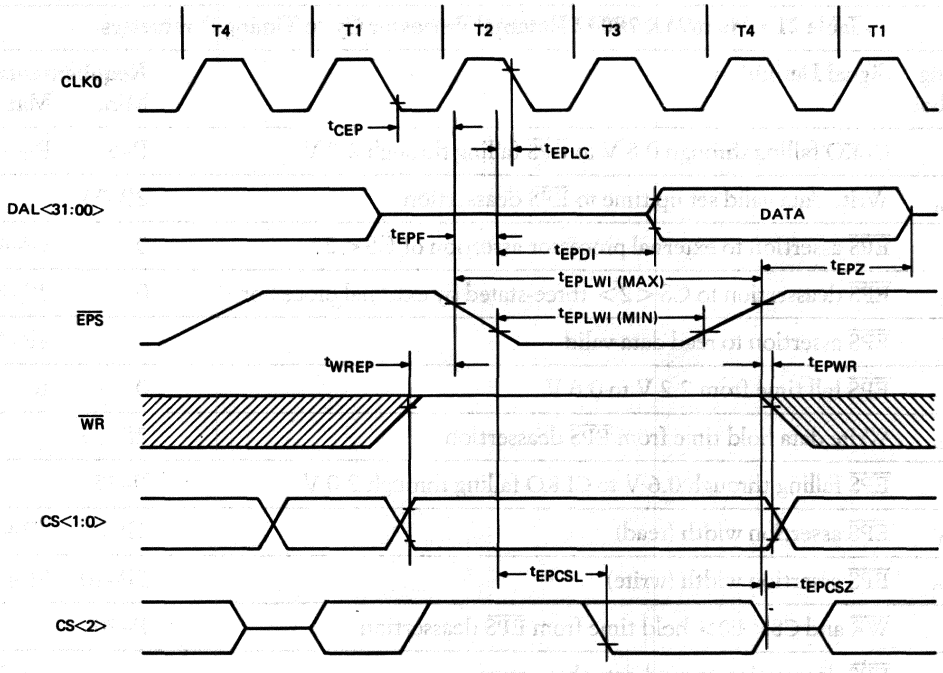
Notes:

[1] The number of microcyles that occur during a DMA grant. A DMA grant is issued for a minimum of two microcycles.

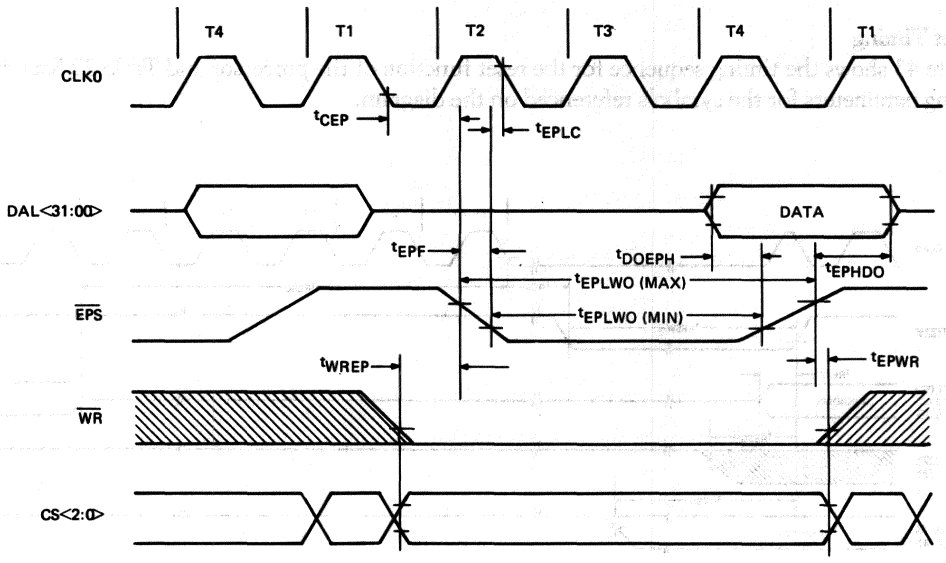
[2] At the conclusion of a $\overline{DMA}$ grant the external logic must deassert the $\overline{AS}$, $\overline{DS}$, and $\overline{DBE}$ signals before the external bus drivers become a high impedance.

External Processor Cycle Timing

Figure 42 shows the timing sequence for the external processor read and response timing and for the external processor write command timing. Table 21 lists the timing parameters for the symbols referenced on the diagrams.



External Processor Read/Response Timing



External Processor Write/Command Timing

Figure 42 ▪ MicroVAX 78032 External Processor Cycle Timing Sequence

Table 21 • MicroVAX 78032 External Processor Cycle Timing Parameters

Timing Symbol	Signal Definition	Requirements Min.	Max.
t_{CEP}	CLKO falling through 0.8 V to $\overline{EPS}$ falling through 2.2 V	P–5	P+19
t_{DOEPH}	Write data valid set up time to $\overline{EPS}$ deassertion	2P–35	
t_{EPCSL}	$\overline{EPS}$ assertion to external processor assertion of CS<2>	0	3P–40
t_{EPCSZ}	$\overline{EPS}$ deassertion to CS<2> three-stated by external processor	0	2P–20
t_{EPDI}	$\overline{EPS}$ assertion to read data valid		4P–40
t_{EPF}	$\overline{EPS}$ fall time from 2.2 V to 0.6 V	0	10
t_{EPHDO}	Write data hold time from $\overline{EPS}$ deassertion	2P–25	
t_{EPLC}	$\overline{EPS}$ falling through 0.6 V to CLKO falling through 2.0 V	P–25	
t_{EPLWI}	$\overline{EPS}$ assertion width (read)	4P–20	4P+20
t_{EPLWO}	$\overline{EPS}$ assertion width (write)	5P–20	5P+20
t_{EPWR}	$\overline{WR}$ and CS<1:0> hold time from $\overline{EPS}$ deassertion	P–20	
t_{EPZ}	$\overline{EPS}$ deassertion to read data three-state		3P–20
t_{WREP}	$\overline{WR}$ and CS<1:0> set up time before $\overline{EPS}$ assertion.	2P–35	

Reset Timing

Figure 43 shows the timing sequence for the reset function of the processor and Table 22 lists the timing parameters for the symbols referenced on the diagram.

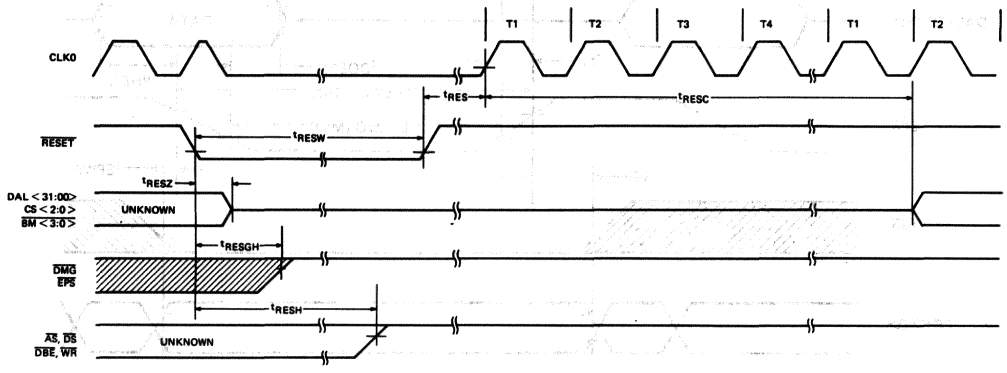


Figure 43 • MicroVAX 78032 Reset Timing Sequence

Table 22 • MicroVAX 78032 Reset Timing Parameters

Timing Symbol	Signal Definition	Requirements (ns) min	max
t_{RES}	$\overline{RESET}$ deassertion to first CLKO pulse if $\overline{RESET}$ is deasserted synchronously	3P+10	3P+85
t_{RESC}	Number of CLKO periods from RESET deassertion until first DAL activity	32 periods	—
t_{RESGH}	$\overline{RESET}$ assertion to DMG, EPS deassertion[1]	—	150
t_{RESH}	$\overline{RESET}$ assertion to $\overline{AS}$, $\overline{DS}$, $\overline{DBE}$, $\overline{WR}$ deassertion[2]	—	1.0 µs
t_{RESW}	$\overline{RESET}$ assertion width after V_{DD} = 4.75 V	3.0 ms	—
t_{RESWB}	$\overline{RESET}$ assertion width if V_{DD} has already been at 4.75 V for 3 ms when $\overline{RESET}$ is asserted	3.0 µs	—
t_{RESZ}	$\overline{RESET}$ assertion to DAL<31:00>, $\overline{BM<3:0>}$, CS<2:0> high impedence[3]		100

Notes:
[1] When the $\overline{RESET}$ level is asserted, the $\overline{DMG}$ and $\overline{EPS}$ signals become high and remain high.
[2] When $\overline{RESET}$ is asserted, $\overline{AS}$, $\overline{DS}$, $\overline{DBE}$ and $\overline{WR}$ outputs become a high impedance state and the levels become high by low current internal pull-ups.
[3] When the $\overline{RESET}$ level is asserted the $\overline{BM<3:0>}$ lines and CS<2:0> lines become high-impedance.

• Mechanical Specifications

The dimensions of the MicroVAX 78032 68-pin cerquad surface and socket mount packages are shown Appendix E.

• Features

- High performance
  —Accelerates by 50 times the execution of MicroVAX floating-point instructions
  —Accelerates by two times the execution of MicroVAX integer multiply and divide
- Subset (70 instructions) of the VAX floating-point instruction set
- Operates with standard VAX integer data types
  —byte, word, longword, and quadword
- Operates with standard VAX floating-point data types
  —single-precision (F_floating)
  —double-precision (D_floating)
  —extended range double-precision (G_floating)
- Arithmetic error checking and reporting
- High-speed ZMOS technology
- Single 5 Vdc power supply

• Description

The MicroVAX 78132 Floating-Point Unit (FPU), contained in a 68-pin cerquad package, is a high-performance cooperative processor that extends the data paths of the MicroVAX 78032 central processing unit (CPU). Its primary function is to execute MicroVAX floating-point instructions to eliminate the emulation of floating-point instructions in software. Figure 1 is a general block diagram of the MicroVAX 78132 FPU.

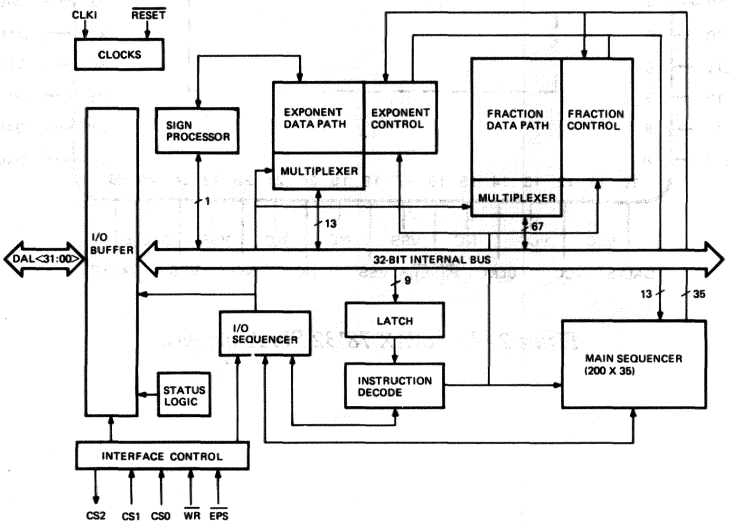


Figure 1 • MicroVAX 78132 General Block Diagram

The MicroVAX FPU handles the F_floating (single-precision), D_floating (double-precision), and G_floating (extended range double-precision) VAX floating-point data types. It supports several VAX floating-point operations, including floating-point add, subtract, multiply, divide, and convert.

The MicroVAX FPU also accelerates the execution of integer multiply and divide operations. The FPU supports signed integer multiply and unsigned integer divide operations.

• Pin and Signal Descriptions

This section provides a brief description of the input and output signals and power and ground connections of the MicroVAX 78132 68-pin package. The pin assignments are identified in Figure 2 and the signals are summarized in Table 1.

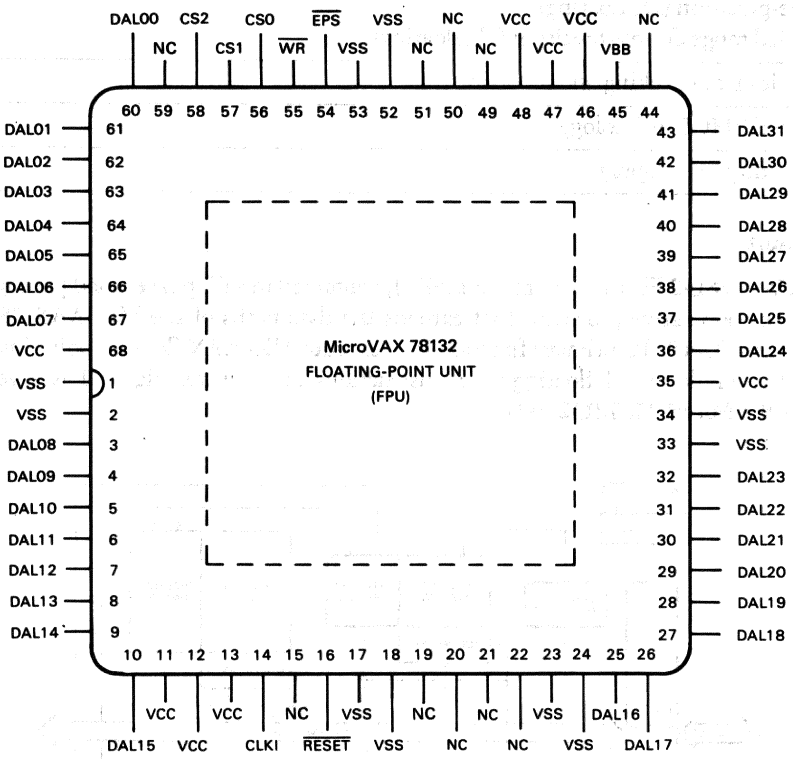


Figure 2 • MicroVAX 78132 Pin Assignments

Table 1 • MicroVAX 78132 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
43-36, 32-25, 10-3, 67-61	DAL<31:00>	input/output	Data/address lines—Transfers data, status, and control information between the FPU and CPU.
57-56	CS<1:0>	input	Control status <1:0>—Indicates the type of information being transferred to or from the FPU (commands, data, or response enable). Valid when $\overline{EPS}$ is asserted.
58	CS2	output	Control status 2—Asserted during an external processor response enable bus cycle when the FPU has completed the commanded operation.
55	$\overline{WR}$	input	Write—Input from the MicroVAX CPU that indicates data flow direction. When asserted, indicates data flow from the CPU to the FPU. Valid when $\overline{EPS}$ is asserted.
54	$\overline{EPS}$	input	External processor strobe—Asserted by the MicroVAX CPU to qualify all communication between the FPU and CPU.
16	$\overline{RESET}$	input	Reset—Asserted by external logic to resynchronize the FPU with the CPU.
14	CLKI	input	Clock input—Basic clock timing input to the FPU. Has the same frequency as the MicroVAX CPU clock (CLKI) input.
45	V_{BB}	output	Back-bias voltage.
11-13,46-48, 35,68	V_{CC}	input	Voltage—Power supply dc voltage
1,2,17,18, 23,24,33,34, 52,53	V_{SS}	input	Ground—Common ground reference
15,19-22, 44,49-51,59	NC		No connection—All unused pins should be left floating and not pulled up or grounded.

Data and Address Bus

Data and address bus (DAL<31:00>)—The data and address bus is a time-multiplexed bidirectional bus that transfers address, data, status, and control information between the FPU and the CPU. The MicroVAX CPU is always the bus master and communicates with the FPU according to the protocol outlined in the "CPU/FPU General Protocol" discussion.

Bus Control

External processor strobe ($\overline{EPS}$)—The $\overline{EPS}$ signal is used to initiate all CPU/FPU bus cycles that are external processor bus cycles. It indicates to the FPU that the information on the CS<1:0> and $\overline{WR}$ lines is valid. Refer to the "Bus Cycles Description" discussion for more information on CPU/FPU bus cycles.

For reset operations, the FPU uses the first assertion of the $\overline{EPS}$ cycle following the assertion of the $\overline{RESET}$ signal to synchronize itself with the CPU.

Write ($\overline{WR}$)—The $\overline{WR}$ signal is an input from the MicroVAX CPU that specifies the direction of data flow between the CPU and FPU on the DAL<31:00> lines. When $\overline{WR}$ is asserted, data is being transferred from the CPU to the FPU.

System Control

Reset ($\overline{RESET}$)—External logic asserts the $\overline{RESET}$ signal to synchronize the FPU and the MicroVAX CPU. The assertion of $\overline{RESET}$ causes the CPU to perform a number of EPS cycles. The first external processor (EPS) cycle synchronizes the FPU and CPU. The remaining EPS cycles are used to verify the presence of the FPU in the system.

Control Status (CS<2:0>)—The CS lines provide status information about the current bus cycle. The CS<1:0> lines are inputs to the FPU that specify the type of information being transferred. CS<1:0> are valid inputs only when the $\overline{EPS}$ signal is asserted. The $\overline{WR}$ signal further qualifies the type of bus cycle, as summarized in Table 2. Refer to the "Bus Cycle Descriptions" section for more information on the types of bus cycles.

Table 2 • MicroVAX 78132 Bus Cycle Types

CS line 1	CS line 0	Write ($\overline{WR}$)	Bus Cycle Type
0	0	0	External processor command write
0	0	1	External processor data read
0	1	0	External processor data write
1	0	0	Non-FPU command write
1	1	1	External processor response enable

CS2 is an open-drain output that is asserted when the current bus cycle is an external processor response enable cycle and the FPU gas completed the current commanded operation.

Clock Signal

Clock In (CLKI)—This signal is the basic timing input provided to both the FPU and CPU from an external clock source.

Power Supply Connections

Power (V_{CC})—These inputs are used to supply 5 Vdc power to the FPU.

Ground (V_{SS})—These inputs are used as a ground reference for the chip.

Back-Bias Generator (V_{BB})—This is a back-bias voltage that is either bypassed to ground with a capacitor (typically 0.01 μF) or attached to a back-bias supply (typically −2.5 volt ± 10% at 10 mA).

▪ Architecture Summary

The MicroVAX 78132 FPU architecture, shown in Figure 3, consists of three separate processors—a 1-bit sign processor, a 13-bit exponent processor, and a 67-bit fraction processor. The extra bits in the fraction data path accommodate extended precision instructions such as EMODx. A microsequencer containing 200 35-bit control words controls operation of the three processors. Data enters and leaves the FPU through the I/O buffer. Control information passes to and from the FPU through the interface control logic.

The following paragraphs describe the FPU architecture accessible to the user.

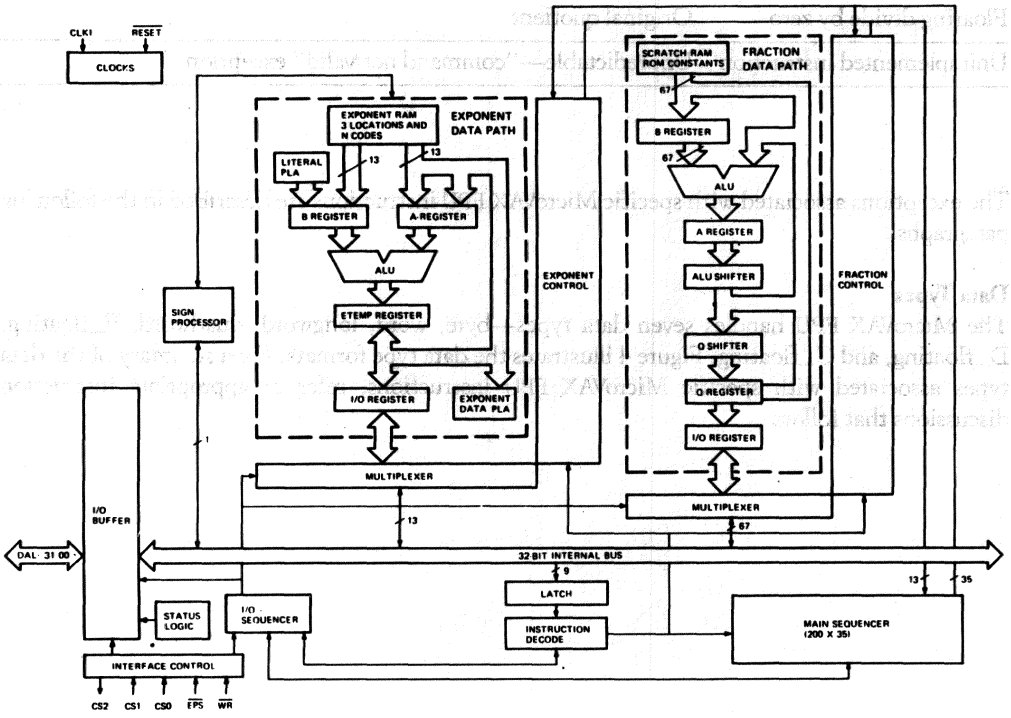


Figure 3 ▪ MicroVAX 78132 Detailed Block Diagram

Visible State

The MicroVAX FPU does not contain user-accessible internal registers or mode bits. The MicroVAX CPU contains the floating-point general registers and condition codes. Commands sent from the CPU to the FPU determine the operational modes (round or truncate) and the data types (for example, F_floating or D_floating).

Exception Detection and Reporting

Table 3 lists the exceptions that the MicroVAX FPU detects and reports.

Table 3 • MicroVAX 78132 Reported Exceptions

Condition	Result Returned to CPU
Reserved operand	Unpredictable—CPU unconditionally faults
Integer overflow	Low order 8, 16, or 32 bits of the true result
Floating overflow	Unpredictable—CPU unconditionally faults
Floating underflow	Floating zero
Floating divide by zero	Original quotient
Unimplemented instruction	Unpredictable—"command not valid" exception

The exceptions associated with specific MicroVAX FPU instructions are described in the following paragraphs.

Data Types

The MicroVAX FPU handles seven data types—byte, word, longword, quadword, F_floating, D_floating, and G_floating. Figure 4 illustrates the data type formats. For a summary of the data types associated with specific MicroVAX FPU instructions, refer to appropriate instruction discussions that follow.

TYPE	LENGTH	USE
BYTE	8 BITS	SIGNED OR UNSIGNED INTEGER
WORD	16 BITS	SIGNED OR UNSIGNED INTEGER
LONGWORD	32 BITS	SIGNED OR UNSIGNED INTEGER
QUADWORD	64 BITS	SIGNED OR UNSIGNED INTEGER
F_FLOATING	32 BITS	FLOATING POINT
D_FLOATING	64 BITS	FLOATING POINT
G_FLOATING	64 BITS	FLOATING POINT

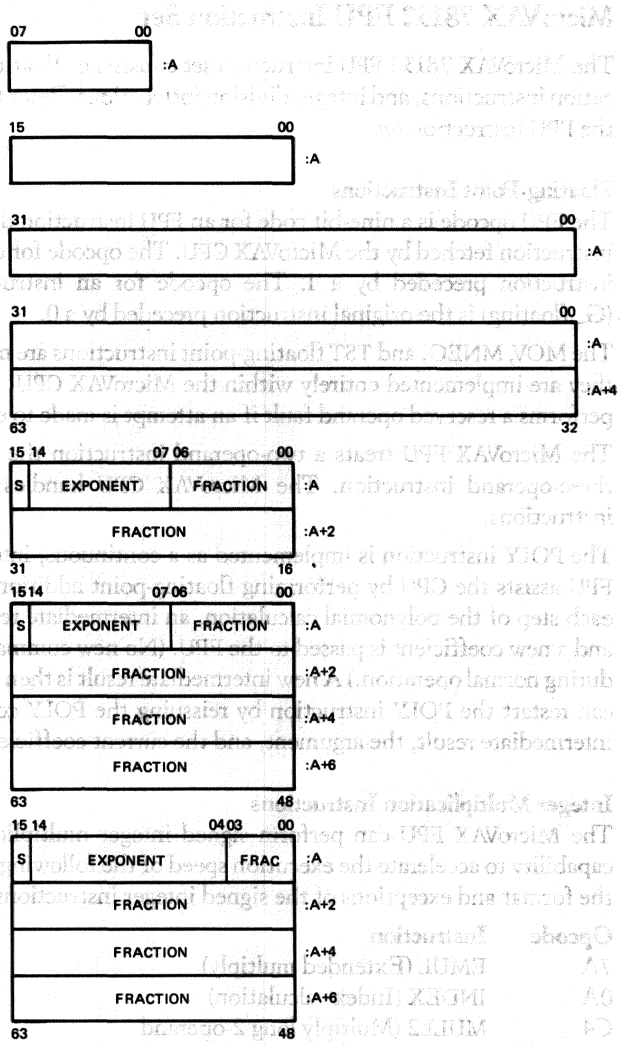


Figure 4 ▪ MicroVAX 78132 Data Types

• MicroVAX 78132 FPU Instruction Set

The MicroVAX 78132 FPU instruction set consists of floating-point instructions, integer multiplication instructions, and integer division instructions. Refer to Appendix A for a summary listing of the FPU instruction set.

Floating-Point Instructions

The FPU opcode is a nine-bit code for an FPU instruction derived from the opcode of the original instruction fetched by the MicroVAX CPU. The opcode for a G_floating instruction is the original instruction preceded by a 1. The opcode for an instruction that is not of extended range (G_floating) is the original instruction preceded by a 0.

The MOV, MNEG, and TST floating-point instructions are marked with an asterisk to indicate that they are implemented entirely within the MicroVAX CPU. In a system without an FPU, the CPU performs a reserved operand fault if an attempt is made to execute these instructions.

The MicroVAX FPU treats a two-operand instruction the same way it treats the corresponding three-operand instruction. The MicroVAX CPU handles the differences in processing these instructions.

The POLY instruction is implemented as a continuous, interruptible instruction. The MicroVAX FPU assists the CPU by performing floating-point addition and multiplication operations. After each step of the polynomial calculation, an intermediate result is returned to the MicroVAX CPU and a new coefficient is passed to the FPU. (No new command is issued to process this coefficient during normal operation.) A new intermediate result is then computed. If the CPU is interrupted, it can restart the POLY instruction by reissuing the POLY command and sending the most recent intermediate result, the argument, and the current coefficient.

Integer Multiplication Instructions

The MicroVAX FPU can perform signed integer multiplication. The MicroVAX CPU uses this capability to accelerate the execution speed of the following instructions. Refer to Appendix A for the format and exceptions of the signed integer instructions executed by the MicroVAX FPU.

Opcode	Instruction
7A	EMUL (Extended multiply)
0A	INDEX (Index calculation)
C4	MULL2 (Multiply long 2-operand
C5	MULL3 (Multiply long 3-operand)

During MicroVAX FPU integer multiplication

- The FPU performs a 32 by 32-bit signed multiplication.
- The FPU in all cases returns a 64-bit result.
- The exception code is zero and the condition codes are unpredictable.

Integer Division Instructions

The MicroVAX FPU can perform unsigned integer division. The MicroVAX CPU uses this capability to accelerate the execution speed of the following instructions. Refer to Appendix A for the format and exceptions of the unsigned integer division instructions executed by the MicroVAX CPU.

Opcode	Instruction
C6	DIVL2 (Divide long 3-operand)
C7	DIVL3 (Divide long 3-operand)
7B	EDIV (Extended divide)

During MicroVAX FPU integer division

- A 64-bit dividend is divided by a 32-bit divisor.
- The operands are unsigned.
- The operands are guaranteed not to cause an integer overflow. (The MicroVAX CPU checks this condition before activating the FPU.)
- The FPU returns a 32-bit result and a 32-bit remainder.
- The exception code is zero and the condition codes are unpredictable.

Interfacing Requirements

The MicroVAX 78132 FPU is designed as a coprocessor for the MicroVAX 78032 CPU. Therefore, all interfacing considerations are made with respect to the MicroVAX CPU. The FPU/CPU interface—including bus cycles, the FPU/CPU protocol, and a typical FPU/CPU interconnection scheme—is described in the following paragraphs.

Bus Cycle Descriptions

The MicroVAX FPU recognizes five types of bus cycles—external processor command write, external processor data read, external processor data write, non-FPU command write, and external processor response enable. The following paragraphs briefly describe these bus cycles. Refer to the "MicroVAX 78032 32-bit Central Processor Unit" section for the corresponding MicroVAX CPU bus cycles. Figures 9 and 10 are detailed bus-cycle diagrams.

External Processor Command Write—An external processor command write cycle is performed when the CPU has a command (typically an instruction opcode) for the FPU to read and execute. This type of cycle lasts four clock (CLKO) periods or one microcycle. The sequence of events is

- The CPU drives cycle-status information on lines CS<1:0> (CS<1:0>=00 for this type of cycle), drives the command on the DAL<31:00> bus, and asserts the $\overline{\text{EPS}}$ and $\overline{\text{WR}}$ signals.
- The FPU reads the opcode on the DAL.
- The CPU deasserts the $\overline{\text{EPS}}$ and $\overline{\text{WR}}$ signals, and the cycle ends.

External processor data read—An external processor data read cycle is performed when the FPU has data for the CPU to process. This type of cycle lasts four clock periods. The sequence of events is

- The CPU drives cycle-status information on lines CS<1:0> (CS<1:0>=01 for this type of cycle) and asserts the $\overline{\text{EPS}}$ signal. The $\overline{\text{WR}}$ signal is not asserted because this is not a write cycle.
- The FPU responds by driving the DAL<31:00> bus with data.
- The CPU reads the data.
- The CPU deasserts the $\overline{\text{EPS}}$ signal, and the cycle ends.

External processor data write—An external processor data write cycle is performed when the CPU has data to write to the FPU. This type of cycle lasts four clock periods. The sequence of events is

- The CPU drives cycle-status information on lines CS<1:0> (CS<1:0>=01 for this type of cycle) and asserts the $\overline{EPS}$ and $\overline{WR}$ signals.
- The CPU drives the data on the DAL<31:00> bus and deasserts the $\overline{EPS}$ and $\overline{WR}$ lines.
- The FPU responds to the deassertion of the $\overline{EPS}$ signal by reading the data on the DAL<31:00> bus, and the cycle ends.

Non-FPU command write—A non-FPU command write cycle is performed when the CPU has an instruction or command for a processing unit other than the FPU. The sequence of events is

- The CPU drives cycle-status information on lines CS<1:0> (CS<1:0>=10 for this type of cycle) and asserts the $\overline{EPS}$ and $\overline{WR}$ signals.
- The FPU initializes itself, suspends its operation, and disables its outputs so that it cannot respond to an external processor data read cycle or an external processor response enable cycle until an external processor command write cycle for the FPU has been initiated.

External processor response enable—An external processor response enable cycle is performed when the CPU is ready to accept the result of some operation from the CPU. The sequence of events is

- The CPU drives cycle-status information on lines CS<1:0> (CS<1:0>=11 for this type of cycle), asserts the $\overline{EPS}$ signal, and puts the CS2 line in the high-impedance state. The $\overline{WR}$ signal is not asserted because this is not a write cycle.
- When the FPU completes its current instruction, it drives status information on the DAL bus and pulls line CS2 low.
- The CPU reads the information and deasserts the $\overline{EPS}$ signal to end the cycle.

CPU/FPU General Protocol

The communication protocol between the CPU and the FPU is grouped into five categories—command transfer, operand transfer, operand processing, status transfer, and result transfer. The following paragraphs describe each transfer.

Command transfer—The CPU initiates an interaction with the FPU by performing an external processor command write cycle. The CPU drives a command (an instruction opcode) on the DAL bus, drives a status code on lines CS<1:0>, and asserts the $\overline{EPS}$ and $\overline{WR}$ signals.

Although the DAL<31:00> bus is driven with a command (in this case an instruction opcode) during the external processor command write cycle, only DAL<08:00> is significant to the FPU. Figure 5 shows the FPU command format and Table 4 describes the bit functions.

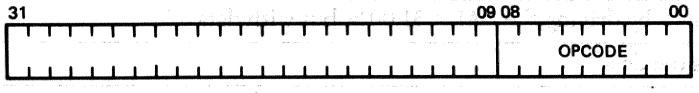


Figure 5 • MicroVAX 78132 Command Format

Table 4 • MicroVAX 78132 Command Description

DAL Line	Description
<31:09>	Not used
<08:00>	Contains the opcode of the instruction that the FPU is to execute or assist in executing.

Operand transfer—The instruction opcode specifies the operation(s) to be performed by the FPU and the number and data type of operands involved. The CPU fetches the required operand(s) and transfers them to the FPU by performing one or more external processor data write cycles. During these cycles, lines CS<1:0> =01 and the DAL<31:00> bus contains the transferred data.

The opcode of the instruction to be executed determines the number and data type of the operands transferred from the CPU to the FPU. The following rules apply to the transfer of operands from the CPU to the FPU:

- Integer operand—An integer operand is transferred in one external processor data write cycle. If the integer is a byte, it appears on DAL<07:00> with leading zeroes on DAL<31:08>. If the integer is a word, it appears on DAL<15:00> with leading zeroes on DAL<31:16>.
- Floating-point operands—An F_floating operand is transferred in one external processor data write cycle. A D_floating or G_floating operand is transferred in two consecutive external processor data write cycles; bits <31:00> are transferred during the first cycle and bits <63:32> during the second.
- Multiple operands—In operations requiring two operands, the second operand is transferred first. In operations requiring three or more operands, the order of operand transfer is determined by the instruction being executed.
- POLY instruction—Execution of the POLY instruction involves an indeterminate number of operands and results. During a POLY instruction, the FPU need only receive a coefficient to calculate the next intermediate result. The FPU continues to accept coefficients and return results until the CPU sends the FPU another command.

Table 5 summarizes the order in which operands and results are transferred for all instructions recognized by the FPU. The table uses the following conventions:

- An "x" in a mnemonic indicates that the FPU processes the two-operand and three-operand versions of the instruction in exactly the same way
- The "#" next to some of the operands and results of a POLY instruction indicates that the number of these operands and results depends on the size of the coefficient table.
- A result may be rounded [R], truncated [T], or exact [E].

The *MicroVAX FPU User's Guide* provides complete descriptions of the operation of these instructions.

Table 5 • MicroVAX78132 Operand Transfer

VAX Mnemonic	FPU Opcode	First Transfer	Second Transfer	Third Transfer	Operation	Result 1	Result 2
ACBD	06F	limit.d	add.d	index.d	(index+add):limit	(index.d[R])	—
ACBF	04F	limit.f	add.f	index.f	(index+add):limit	(index.f[R])	—
ACBG	14F	limit.g	add.g	index.g	(index+add):limit	(index.g[R])	—
ADDDx	060, 061	add2.d	add1.d	—	add1+add2	sum.d[R]	—
ADDFx	040, 041	add2.f	add1.f	—	add1+add2	sum.f[R]	—
ADDGx	140, 141	add2.g	add1.g	—	add1+add2	sum.g[R]	—
CMPD	071	src2.d	src1.d	—	src1-src2	—	—
CMPF	051	src2.f	src1.f	—	src1-src2	—	—
CMPG	151	src2.g	src1.g	—	src1-src2	—	—
CVTBD	06C	src.b	—	—	flt cvrt	d_float[E]	—
CVTBF	04C	src.b	—	—	flt.cvrt	f_float[E]	—
CVTBG	14C	src.b	—	—	flt cvrt	g_float[E]	—
CVTDB	068	src.d	—	—	int cvrt	byte[T]	—
CVTDF	076	src.d	—	—	flt change	f_float[R]	—
CVTDL	06A	src.d	—	—	int cvrt	longword[T]	—
CVTDW	069	src.d	—	—	int cvrt	word[T]	—
CVTFB	048	src.f	—	—	int cvrt	byte[T]	—
CVTFD	056	src.f	—	—	flt change	d_float[E]	—
CVTFG	199	src.f	—	—	flt change	d_float[E]	—
CVTFL	04A	src.f	—	—	int cvrt	longword[T]	—
CVTFW	049	src.f	—	—	int cvrt	word[T]	—
CVTGB	148	src.g	—	—	int cvrt	byte[T]	—
CVTGF	133	src.g	—	—	flt change	f_float[R]	—
CVTGL	14A	src.g	—	—	int cvrt	longword[T]	—
CVTGW	149	src.g	—	—	int cvrt	word[T]	—
CVTLD	06E	src.l	—	—	flt cvrt	d_float[E]	—
CVTLF	04E	src.l	—	—	flt cvrt	f_float[R]	—
CVTLG	14E	src.l	—	—	flt cvrt	g_float[E]	—
CVTWD	06D	src.w	—	—	flt cvrt	d_float[E]	—
CVTWF	04D	src.w	—	—	flt cvrt	f_float[E]	—
CVTWG	14D	src.w	—	—	flt cvrt	g_float[E]	—
CVTRDL	06B	src.d	—	—	mid int cvrt	longword[R]	—
CVTRFL	04B	src.f	—	—	mid int cvrt	longword[R]	—
CVTRGL	14B	src.g	—	—	mid int cvrt	longword[R]	—
DIVDx	066, 067	divd.d	divr.d	—	divd/divr	quo.d[R]	—
DIVFx	046, 047	divd.f	divr.f	—	divd/divr	quo.f[R]	—
DIVGx	146, 147	divd.g	divr.g	—	divd/divr	quo.g[R]	—
EMODD	074	muirx.b	muir.d	muld.d	muir*(muir'muirx)	int.l	fract.d[R]
EMODF	054	muirx.b	muir.f	muld.f	muir*(muir'muirx)	int.l	fract.f[R]
EMODG	154	muirx.w	muir.g	muld.g	muir*(muir'muirx)	int.l	fract.g[R]

VAX Mnemonic	FPU Opcode	First Transfer	Second Transfer	Third Transfer	Operation	Result 1	Result 2
MULDx	064, 065	muir.d	muld.d	—	muir*muld	prod.d[R]	—
MULFx	044, 045	muir.f	muld.f	—	muir*muld	prod.f[R]	—
MULGx	144, 145	muir.g	muld.g	—	muir*muld	prod.g[R]	—
POLYD	075	arg.d	int1.d	#coeff.d	(arg*int1)+coeff	#int2.d[R]	#int3.d[R] ...
POLYF	055	arg.f	int1.f	#coeff.f	(arg*int1)+coeff	#int2.f[R]	#int3.f[R] ...
POLYG	155	arg.g	int1.g	#coeff.g	(arg*int1)+coeff	#int2.g[R]	#int3.g[R] ...
SUBDx	062, 063	min.d	sub.d	—	min-sub	diff.d[R]	—
SUBFx	042, 043	min.f	sub.f	—	min-sub	diff.f[R]	—
SUBGx	142, 143	min.g	sub.g	—	min-sub	diff.g[R]	—
EMUL	07A	muir.ri	muld.ri	—	muir*muld	prod.wq[E]	—
INDEX	00A	(muir-ri)	size-ri	—	muir*size	indexout.wq[E]	—
MULLx	0C4, 0C5	muir-ri	muld.ri	—	muir*muld	prod.wq[E]	—
DIVLx	0C6, 0C7	divr.rl	divd.rq	—	divd/divr	quo.w[E]	rem.w[E]
EDIV	07B	divr.rl	divd.rq	—	divd/divr	quo.w[E]	rem.w[E]

Note: The integer divide instructions require that the lower 32-bits of the dividend be transferred first, and then the upper 32-bits.

Operand processing—After the CPU transfers the last operand, it continuously performs external processor response enable cycles and waits for a response from the FPU. For each of these cycles, the CPU asserts the appropriate status code on lines CS<1:0> (CS<1:0> = 11) and asserts the $\overline{EPS}$ signal.

FPU operand processing is completely invisible to the CPU and may not be altered by the user.

Status transfer—When the FPU is ready to pass a result to the CPU, it responds to the next external processor response enable cycle by asserting the CS2 signal and simultaneously driving status information onto the DAL bus. The CPU responds to the assertion of the CS2 signal by performing one additional external processor response enable cycle, during which the CPU reads the FPU status information again.

The format of the status information is shown in Figure 6 and defined in Table 6. When the FPU asserts the CS2 signal, it places 32 bits of status onto the DAL<31:00> bus. The CPU examines bits <05:00>.

Figure 6 • MicroVAX 78132 Status Format

Table 6 • MicroVAX 78132 Status Description

DAL Line	Description
31:06	Not used
05	FN (Function negative)—Indicates the status as follows: FN = 1 if result LSS is 0 FN = 0 if LSS not 0
04	FZ (Function zero)—Indicates the status as follows: FZ = 1 if the result EQL is 0 FZ = 0 if result EQL is not 0
03	BR (Branch)—Indicates the status as follows: BR = 1 if ABCx should branch BR = 0 if ABCx should not branch
02:00	EXC CODE (Exception code)—Indicates that the following events have occurred: **Code** — **Description** 0 — reserved 1 — floating divide by zero 2 — integer overflow 3 — floating overflow 4 — floating underflow 5 — reserved operand detected 6 — operation completed normally

If the EXC CODE is not 7 after a floating-to-integer conversion instruction, the condition codes associated with the instruction are unpredictable and must be determined by the CPU. Integer multiply and divide instructions always return unpredictable condition codes (that is, in the FN, FZ, and BR fields) and set the EXC CODE field to 7.

Result transfer—The CPU performs one or more external processor data read cycles to read a result from the FPU. During one of these cycles, lines CS<1:0> = 01 and the DAL<31:00> bus contains the read data. After the result transfer, the CPU and FPU are free for the next transaction.

The following rules apply to result transfers from the FPU to the CPU:

- Integer results—An integer result is transferred in one external processor data read cycle. If the integer is a byte, it appears on DAL<07:00> with unpredictable data on DAL<31:08>. If the integer is a word, it appears on DAL<15:00> with unpredictable data on DAL<31:16>. For integer multiplication, two 32-bit transfers are necessary to return the entire result.
- Floating-point results—An F_floating operand is transferred in one external processor data read cycle. A D_floating or G_floating operand is transferred in two consecutive external processor data read cycles; bits <31:00> are transferred during the first cycle; and bits <63:32>, during the second cycle.
- Overflow and underflow—In integer overflow cases, the FPU always returns the low-order bits of the true integer result; in floating underflow cases, the FPU always returns a zero result.
- CMPx instruction results—CMPD, CMPF, and CMPG do not cause a result to be generated by the FPU, but the CPU will request one. The FPU returns a meaningless result that the CPU ignores.

Typical FPU/CPU Interconnection

Figure 7 illustrates a typical FPU/CPU hardware configuration. In the example, a DAL transceiver and latch is included in the design for the benefit of the external logic that communicates with the FPU/CPU via the DAL. Also included for completeness are several MicroVAX 78032 CPU control signals that must be interpreted or generated by external logic. Refer to the "MicroVAX 78032 CPU" section for information on these control signals.

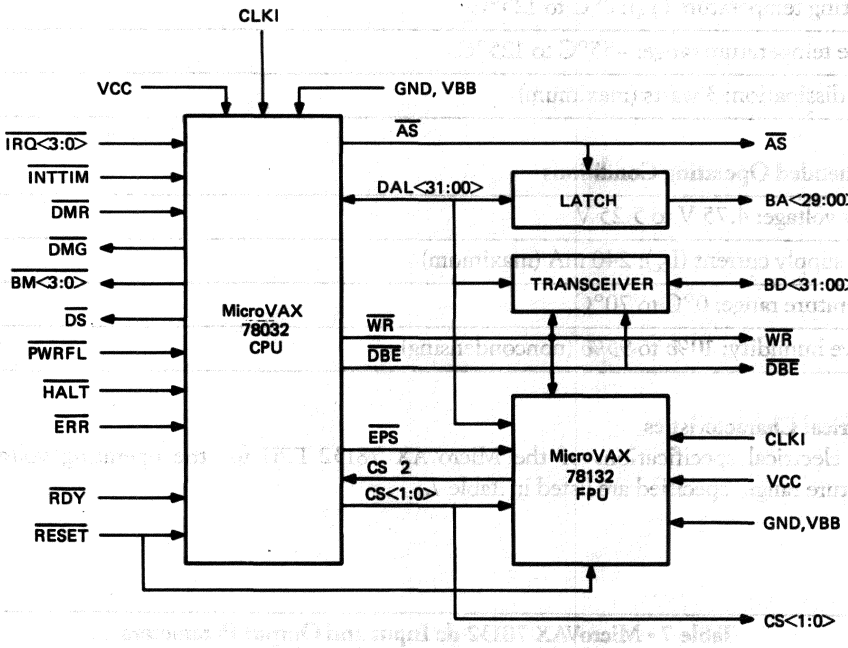


Figure 7 • MicroVAX 78132 Typical FPU/MicroVAX 78032 CPU Interconnection

• Specifications

The mechanical, electrical, and environmental characteristics and specifications for the MicroVAX 78132 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Operating temperature range (T_A): 0°C to 70°C
- Ground reference (V_{SS})
- Supply voltage (V_{CC}): 4.75 V

Mechanical Configuration

The physical dimensions of the MicroVAX 78132 68-pin CERQUAD package are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Supply voltage (V_{CC}): –0.5 V to 7.0 V
- Input or output voltage applied: –1.0 V to 10 V
- Operating temperature (T_A): 0°C to 125°C
- Storage temperature range: –55°C to 125°C
- Power dissipation: 3 watts (maximum)

Recommended Operating Conditions

- Supply voltage: 4.75 V to 5.25 V
- Active supply current (I_{CC}): 240 mA (maximum)
- Temperature range: 0°C to 70°C
- Relative humidity: 10% to 95% (noncondensing)

dc Electrical Characteristics

The dc electrical specifications of the MicroVAX 78132 FPU for the operating voltage and temperature ranges specified are listed in Table 7.

Table 7 • MicroVAX 78132 dc Input and Output Parameters

Symbol	Parameter	Requirements Min.	Requirements Max.	Units	Test Conditions
V_{IH}	High-level input voltage	2.0	—	V	
V_{IL}	Low-level input voltage	—	0.8	V	
V_{IHE}	High-level input $\overline{EPS}$ signal	2.2	—	V	
V_{ILE}	Low-level input $\overline{EPS}$ signal	—	0.6	V	
V_{OH}	High-level output voltage	2.4	—	V	I_{OH} = –400 µA
V_{OL}	Low-level output voltage	—	0.4	V	I_{OL} = 2.0 mA
V_{OLS}	Low-level output voltage CS2	—	0.4	V	I_{OL} = 5.2 mA

Symbol	Parameter	Requirements Min.	Max.	Units	Test Conditions
I_{IL}	Input leakage current	-10	10	µA	$0 < V_{in} < V_{DD}$
I_{OL}	Output leakage current	-10	10	µA	$0.4 < V_{in} < V_{DD}$
I_{CC}	Active supply current	—	700	mA	$I_{out} = 0$, $t_A = 0°C$
C_{in}	Input capacitance (except $\overline{EPS}$)	10	—	pF	
	$\overline{EPS}$ input capacitance	30	—	pF	

ac Electrical Characteristics

Figures 8 is the timing waveform for the clock input (CLKI) and Table 8 lists the clock timing parameters. The formulas for the timing parameters are stated in terms of clock periods where a period $P = t_{CIP}$.

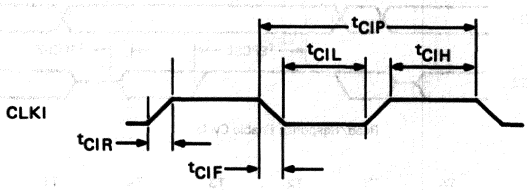


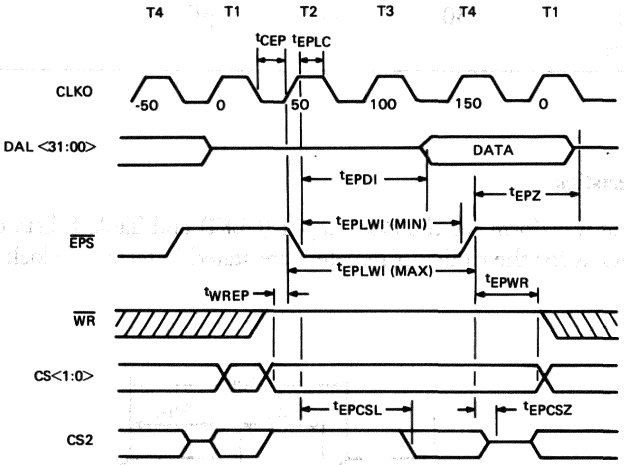
Figure 8 • MicroVAX 78132 CLKI Timing Waveform

Table 8 • MicroVAX 78132 CLKI Timing Parameters

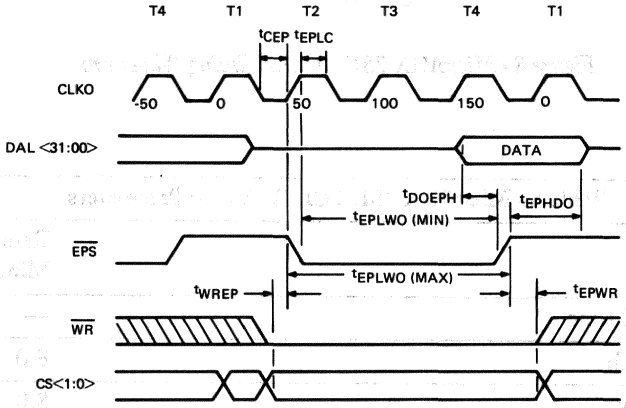
Symbol	Definition	Requirements (ns) Min.	Max.
t_{CIF}	Clock in fall time	—	4.5
t_{CIH}	Clock in high	8.0	—
t_{CIL}	Clock in low	8.0	—
t_{CIP}	Clock in period	25	250
t_{CIR}	Clock in rise time	—	4.5

Figure 9 shows the external processor data timing sequence for a read/response enable cycle and a write/command write cycle. Table 9 lists the signal timing parameters. The following notes apply to the signal measurements.

- ac characteristics are measured with a purely capacitive load of 100 pF and the parameters are valid for loads up to 100 pF.
- ac high levels are measured at 2.0 V and low levels at 0.8 V except for the $\overline{EPS}$ signal.
- The ac high level of the $\overline{EPS}$ signal is measured at 2.2 V and the ac low level is measured at 0.6 V.
- $P = t_{CIP}$



Read/Response Enable Cycle



Write/Command Write Cycle

Figure 9 • MicroVAX 78132 External Processor Data Transaction Timing

Table 9 • MicroVAX78132 External Processor Data Transaction Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{CEP}	CLKO falling through 0.8 V to $\overline{EPS}$ falling through 2.2 V	P + 1	P + 19
t_{DOEPH}	Write data valid setup time to $\overline{EPS}$ deassertion	2P - 35	—
t_{EPCSL}	$\overline{EPS}$ assertion to external processor assertion of CS<2>	0	3P - 40
t_{EPCSZ}	$\overline{EPS}$ deassertion to CS<2> three-stated by external processor	0	2P - 20
t_{EPDI}	$\overline{EPS}$ assertion to read data valid	4P - 40	—
t_{EPHDO}	Write data hold time from $\overline{EPS}$ deassertion	2P - 25	—
t_{EPLC}	$\overline{EPS}$ falling through 0.6 V to CLKO rising through 2.0 V	P - 25	—
t_{EPLWI}	$\overline{EPS}$ assertion width (read)	4P - 20	4P + 20
t_{EPLWO}	$\overline{EPS}$ assertion width (write)	5P - 20	5P + 20
t_{EPWR}	WR and CS<1:0> hold time from $\overline{EPS}$ deassertion	P - 20	—
t_{EPZ}	$\overline{EPS}$ deassertion to read data three-state	3P - 20	—
t_{WREP}	WR and CS<1:0> set up time before $\overline{EPS}$ assertion	2P - 35	—

Features

- Fully compatibile with the MicroVAX 78032 CPU, CVAX 78034 CPU and rtVAX 78R32 CPU
- 16 Peripheral Interrupt Request (PIRQ) lines
- Edge or level triggering for each PIRQ line with individually selected priorities
- 16 Programmable vector addresses
- Optional external vector generation
- Fixed or round robin priority modes
- Uses a daisychain interrupt-enable scheme for cascading
- High-speed, low-power CMOS technology
- Single 5-Vdc power supply

Description

The MicroVAX 78516 Vectored Interrupt Controller (VIC)* is a low-cost, programmable interrupt controller that is fully compatible with the MicroVAX 78032 CPU, CVAX 78034 CPU, and rtVAX 78R32 CPU. The VIC manages as many as 16 interrupt sources, resolves interrupt priorities, drives the interrupt request (IRQ) lines of the CPU, and provides a programmable 13-bit interrupt vector to the CPU. Users can choose either the fixed or round robin interrupt priority mode. Using a daisychain scheme, the VIC is cascadable. Figure 1 is a general block diagram of the MicroVAX 78516 VIC.

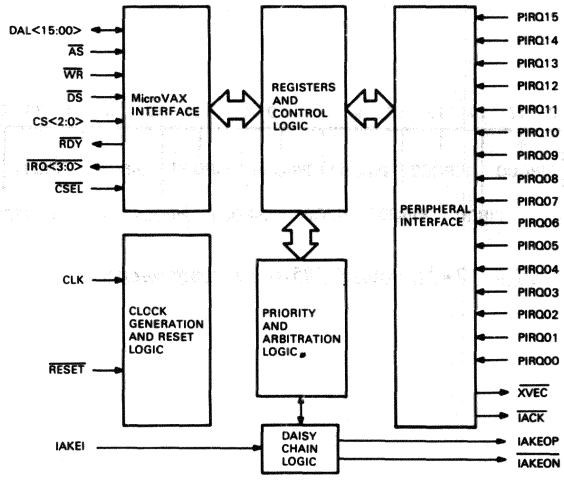


Figure 1 • MicroVAX 78516 VIC General Block Diagram

*The MicroVAX 78516 VIC and the rtVAX 78516 VIC are physically and functionally identical and are assigned the same Digital part number.

• Pin and Signal Descriptions

The input and output signals and power and ground connections for the 68-pin device are shown in Figure 2. Table 1 contains a summary of the signals and describes their functions. Detailed descriptions of the signal functions are contained in paragraphs that follow.

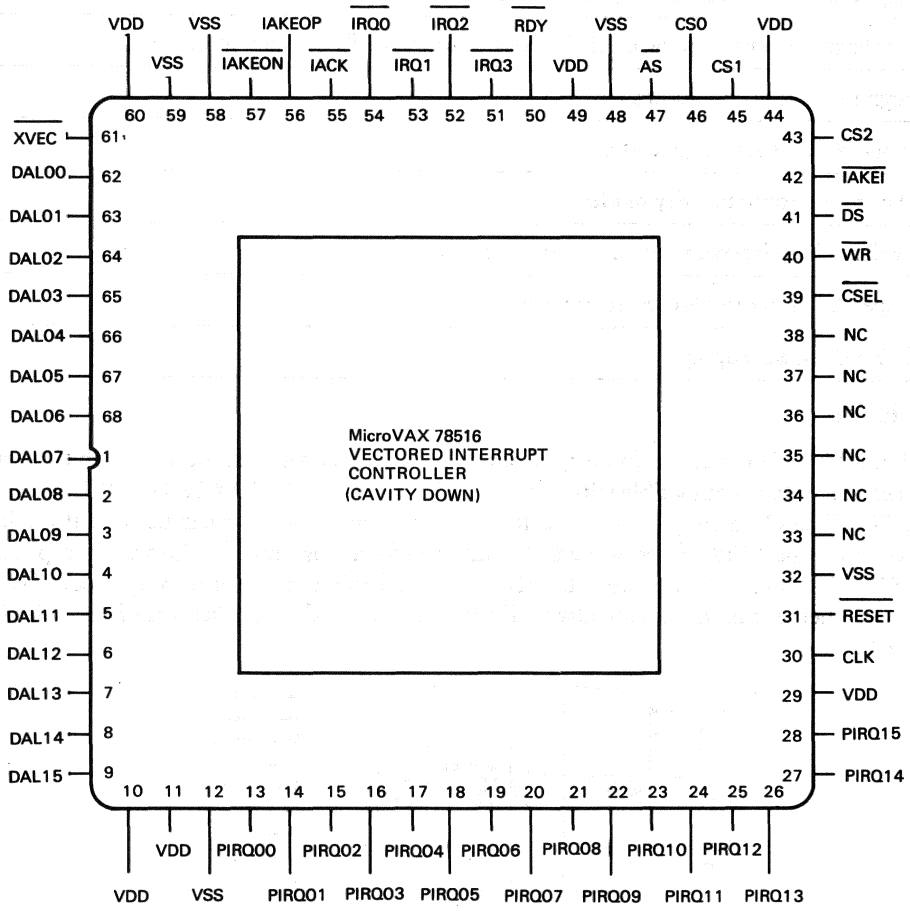


Figure 2 • MicroVAX 78516 Pin Assignments

Table 1 • MicroVAX 78516 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
9-1,68-62	DAL<15:00>	input/output	Data/address lines <15:0>—Time-multiplexed lines used to transfer address, data, and interrupt information between the VIC and the CPU.
47	$\overline{AS}$	input	Address strobe—Latches the state of the VIC into internal registers.
41	$\overline{DS}$	input	Data Strobe—When asserted during a CPU read or interrupt acknowledge cycle, it indicates that DAL<15:00> lines are available to receive data. When asserted during a CPU write cycle, it latches the data on the DAL<15:00> lines into the internal registers.
40	$\overline{WR}$	input	Write—Indicates the direction of data transfer on the DAL<15:00> lines.
43,45,46	CS<2:0>	input	Control status—Used to decode the bus cycle type.
50	$\overline{RDY}$	output	Ready—Synchronizes data transfers between the VIC and the CPU.
28-13	PIRQ<15:00>	inputs	Peripheral interrupt requests <15:00>—Used by peripheral devices to request an interrupt.
51-54	$\overline{IRQ<3:0>}$	output	Interrupt request <3:0>—Used to notify the CPU of any pending interrupts. These lines are maskable by the CPU.
55	$\overline{IACK}$	output	Interrupt acknowledge—Indicates that the current bus cycle edge is an interrupt acknowledge cycle.
61	$\overline{XVEC}$	output	External vector—Indicates that the interrupt request is being acknowledged and the peripheral device must supply a vector to the CPU.
42	$\overline{IAKEI}$	input	Interrupt acknowledge enable in—Daisychain control signal that indicates the VIC can respond to the current interrupt acknowledge cycle.
56	IAKEOP	output	Interrupt acknowledge enable out P—An active high pullup output that connects together with the with the $\overline{IAKEON}$ output to the $\overline{IAKEI}$ input of the next device in the daisychain.

Pin	Signal	Input/Output	Definition/Function
57	$\overline{\text{IAKEON}}$	output	Interrupt acknowledge enable out N—An active low pulldown output that connects together with the IAKEOP output to the $\overline{\text{IAKEI}}$ input of the next device in the daisychain or connects to the $\overline{\text{ERR}}$ input of the CPU.
39	$\overline{\text{CSEL}}$	input	Chip select—Enables read/write operations to the internal registers.
31	$\overline{\text{RESET}}$	input	Reset—Sets the VIC to a known initial state.
30	CLK	input	Clock—Used to generate the internal time states of the VIC.
10,11,29,44,60	V_{DD}	input	Voltage—Power supply voltage.
12,32,48,58,59	V_{SS}	input	Ground—Ground reference

MicroVAX Bus Interface Signals

Data/Address lines (DAL<15:00>)—These lines are bidirectional and are used to transfer address and data between the VIC and the CPU. During internal VIC register access cycles, when the $\overline{\text{CSEL}}$ line is asserted, the DAL<15:00> lines transfer data to and from the internal registers. During interrupt acknowledge cycles, if the $\overline{\text{IAKEI}}$ input is asserted and the VIC has a pending interrupt at the level being acknowledged, the VIC places one of its interrupt vector registers on the DAL<15:00> lines or assert the external vector control signal ($\overline{\text{XVEC}}$). When the $\overline{\text{XVEC}}$ signal is asserted, the interrupting device must then supply a vector. During interrupt acknowledge cycles, the DAL<15:00> lines are driven only when the $\overline{\text{IAKEI}}$ input is asserted, the $\overline{\text{IAKEON}}$ output is deasserted, and the XVE bit in the interrupt vector register is cleared. The DAL<15:00> lines are otherwise in a high-impedance state.

Address strobe ($\overline{\text{AS}}$)—When asserted, this signal latches the information on the DAL<06:00>, CS<2:0>, and the $\overline{\text{WR}}$ lines into the VIC. This information is used internally to latch the PIRQ<15:00> line information for the duration of a read or interrupt acknowledge bus cycle that accesses the VIC.

Data strobe ($\overline{\text{DS}}$)—This signal is used by the VIC for data timing during internal register access cycles and interrupt acknowledge cycles. When writing to one of the internal registers, the assertion of this signal strobes the DAL<15:00> line data into the selected register. When reading an internal register, the assertion of this signal is used to transfer the contents of the selected register onto the DAL<15:00> lines. When responding to an interrupt acknowledge cycle, the assertion of this signal is used to transfer the contents of the appropriate interrupt vector register onto the DAL<15:00> lines.

Write ($\overline{\text{WR}}$)—This signal indicates whether the current bus cycle is a read or a write cycle. This signal is used with the CS<2:0> inputs to decode the type of bus cycle in progress and to access internal registers to determine whether the operation is a read or write operation. The $\overline{\text{WR}}$ input is asserted for write cycles and is deasserted for read or interrupt acknowledge bus cycles.

Control status (CS<2:0>)—These lines and the $\overline{\text{WR}}$ input are decoded to determine the presence of a read, write, or interrupt acknowledge bus cycle. The bus cycle selections are listed in Table 2.

Table 2 • MicroVAX 78516 Bus Cycle Decoding*

CS Line 2	1	0	$\overline{WR}$	$\overline{CSEL}$	Bus Cycle
H	X	X	H	L	Read
H	X	H	L	L	Write
L	H	H	H	X	Interrupt acknowledge

*H = high level, L = low level, X = either high or low level.

Ready ($\overline{RDY}$)—This signal is asserted by the VIC when its internal registers are accessed during a read or write cycle or during an interrupt acknowledge (IACK) cycle when the VIC is providing an interrupt vector. During IACK cycles, at least one ready slip will be generated to allow an interrupt acknowledge enable signal ($\overline{IAKEI}$, IAKEOP, or IAKEON) to propagate through the daisychain. The total number of ready slips that occur depends on the length of the daisychain. This is an open drain (pulldown) output capable of sinking 16 mA.

Interrupt Interface Signals

Peripheral interrupt request (PIRQ <15:00>)—These input lines are used by peripheral circuits to request an interrupt. When one or more of these lines are asserted and the interrupts are enabled, the VIC will assert the appropriate IRQ line(s). Mapping between each PIRQ line and the IRQ line is programmable by software though the IRQ Map registers. The interrupt request can be sensed by a signal level or edge or by the signal polarity. The sensing is programmable by the user. Unused PIRQ lines must be connected to a valid logic level.

Interrupt request ($\overline{IRQ<3:0>}$)—One or more of these lines will be asserted by the VIC when a PIRQ line is asserted and the interrupts are enabled. The IRQ Map registers determine which IRQ line is asserted for a particular PIRQ line. An IRQ line will be deasserted when all pending interrupts mapped to that IRQ line have been serviced. These are open drain (pulldown) outputs that require external pullup resistors.

Interrupt acknowledge ($\overline{IACK}$)—This signal is a result of decoding the CS<2:0> and the $\overline{WR}$ lines and will be asserted for all interrupt acknowledge cycles. The signal is not affected by the interrupt acknowledge daisychain signals. It allows the external logic to disable the memory transceivers during an interrupt acknowledge cycle.

External vector enable ($\overline{XVEC}$)—This signal is asserted if the XVE bit is set in the interrupt vector register for the PIRQ being acknowledged. The requesting device must supply its own vector and the VIC places the DAL<15:00> lines in the high-impedance state. The hardware supplying the vector is required to assert the $\overline{RDY}$ signal at the correct time.

Daisychain Interface Signals

Interrupt acknowledge enable in ($\overline{IAKEI}$)—This input allows more than one VIC and other peripheral chips to be connected together in a daisychain. When this input is asserted, the VIC can respond to the current interrupt acknowledge bus cycle. This signal should be connected to a ground reference if the VIC is the highest priority device in the daisychain.

Interrupt acknowledge enable out high (IAKEOP)—This output and the $\overline{\text{IAKEON}}$ output are connected to the $\overline{\text{IAKEI}}$ pin of the next lowest device in the interrupt daisychain. The IAKEOP output is normally an active high pullup. However, when the $\overline{\text{IAKEI}}$ signal is asserted and the VIC has no pending interrupts at the level being acknowledged, the IAKEOP output is a high-impedance. If the VIC is the lowest-priority device in the daisychain, this line is not connected to another device. This is an open drain, pullup output that cannot be pulled low. For daisychain operation, the pulldown function is performed by the $\overline{\text{IAKEON}}$ line.

Interrupt acknowledge enable out low/error low ($\overline{\text{IAKEON}}$)—This is an open drain pulldown output that is used to either pull down the next $\overline{\text{IAKEI}}$ level in a daisychain application, or pull down the $\overline{\text{ERR}}$ input to the CPU if this VIC is the last (or only) device in the daisychain. The $\overline{\text{IAKEON}}$ line is normally in a high-impedance state. However, when the $\overline{\text{IAKEI}}$ signal is asserted and the VIC has no pending interrupts at the level being acknowledged, the $\overline{\text{IAKEON}}$ signal is asserted. The VIC asserts this signal if PIRQ line is in level mode and the interrupting device removes its request before the interrupt is acknowledged. The level sensitive inputs are not stored by the VIC. Therefore, the PIRQ line that was asserted cannot be used to determine the vector to return to the CPU. This output is a high current open drain output.

Miscellaneous Signals

Chip select ($\overline{\text{CSEL}}$)—This signal, when asserted, enables read/write operations to the internal registers.

Reset ($\overline{\text{RESET}}$)—This signal, when asserted, sets all the internal registers to a known value except for the interrupt vector (IVEC) and IRQ map (IMAP) registers. The contents of the IVEC and IMAP registers are unknown. The interrupts are disabled and the DAL<15:00> lines become a high impedance.

Clock (CLK)—This signal is used to generate the internal time states within the VIC. Any oscillator that meets the input requirements of CLK signal may be used.

Power and Ground Connections

Power supply voltage (V_{DD})—Power supply 5 Vdc.

Ground (V_{SS})—Ground reference.

Functional Description

The VIC may be connected directly to the CPU bus or to a buffered I/O bus. It accepts up to 16 priority interrupt requests (PIRQ<15:00>) from peripheral devices and it drives an associated IRQ line to the CPU. The mapping between the VIC PIRQ lines and CPU IRQ lines is programmable. The VIC decodes the presence of a CPU interrupt acknowledge (IACK) cycle on the bus and monitors the interrupt priority level of the interrupt being acknowledged. It will respond to the IACK cycle by transferring the appropriate user programmed vector on lines DAL<15:00>. The CPU uses the vector as an offset into the system control block (SCB) to locate the starting address of the interrupt routine.

A daisychain wiring scheme enables the user to connect more than one VIC together so as to expand the interrupt handling capability from that of a single VIC. This scheme is compatible with the daisychain scheme used by the other peripheral interfaces.

A peripheral device requests service by asserting one of the PIRQ lines. When the VIC detects the PIRQ line that has been enabled, it reflects the assertion of the line in the pending summary register (PSR) bit that corresponds to that PIRQ line. The IRQ output, programmed by the user for that PIRQ, will also be asserted to indicate to the CPU the interrupt condition at the specified IPL

level. The CPU will respond with an interrupt acknowledge cycle that contains the priority level of the interrupt being acknowledged. The VIC then decodes the IACK cycle and IPL line information and if the VIC generated the interrupt and the $\overline{\text{IAKEI}}$ (daisychain input) signal is asserted, it selects the vector of the next PIRQ to be serviced for that IPL level. It then places that vector on the DAL<15:00> lines. If the VIC did not request the interrupt, it asserts the $\overline{\text{IAKEON}}$ (daisychain output) signal to allow the next devices in the daisy chain to be serviced. When the VIC is responding to an interrupt, it holds the $\overline{\text{IAKEON}}$ line from being asserted to prevent devices in the daisychain that have a lower priority from responding.

Registers

The VIC contains 16 interrupt vector registers and 9 interrupt control registers that allow each request to be individually configured by software. The internal VIC registers, shown in Figure 3, are accessible by the CPU and are used by software to configure the operation of the VIC. Each register consists of 16-bits and is located on a longword boundary. The base address is determined by external address decode logic. Direct access to the VIC registers is enabled when the $\overline{\text{CSEL}}$ signal is asserted and the VIC decodes the address on the DAL<06:00> lines to select the register to be accessed.

NOTE: Only word access to the lower 16-bits of the longword are allowed to transfer data between the CPU and the VIC. Byte accesses and longword accesses are not allowed. Longword access may result in the CPU reading the incorrect data or lost data during a write cycle.

ADDRESS	15 — 00
BASE	POLARITY REGISTER
BASE+4	LEVEL/EDGE REGISTER
BASE+8	PENDING SUMMARY REGISTER
BASE+12	INTERRUPT ENABLE REGISTER
BASE+16	IRQ MAP REGISTER 0
BASE+20	IRQ MAP REGISTER 1
BASE+24	IRQ MAP REGISTER 2
BASE+28	IRQ MAP REGISTER 3
BASE+32	ROUND ROBIN REGISTER
BASE+36	• ADDRESSES (BASE+36) TO (BASE+60) ARE NOT INTERNALLY DECODED BY THE VIC •
BASE+64	INTERRUPT VECTOR REGISTER 0
BASE+68	• •
BASE+124	INTERRUPT VECTOR REGISTER 15

Figure 3 ▪ MicroVAX 78516 Register Address and Descriptions

Polarity register—The polarity (POL) register selects the polarity of the input used to assert a PIRQ<15:00> line. When a bit is set, the corresponding line is asserted by a low-to-high transition or by a high level. When a bit is clear, the corresponding line is asserted by a high-to-low transition or by a low level. The register format is shown in Figure 4.

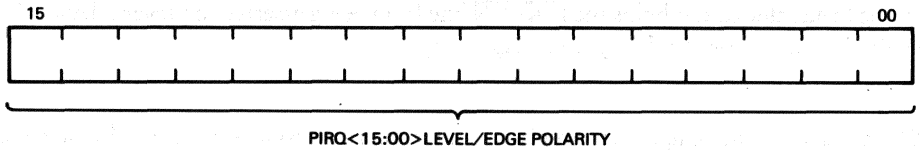


Figure 4 ▪ MicroVAX 78516 Polarity Register Format

The POL register is used with the level/edge (LE) register to configure each PIRQ input. A PIRQ input may be configured to respond to a rising edge, a falling edge, a high level, or a low level signal. Table 3 shows the bit selections of the POL and LE registers and the resulting state of a PIRQ line. When the $\overline{\text{RESET}}$ line is asserted, the POL register is cleared.

Table 3 ▪ MicroVAX 78516 PIRQ Input Line Configurations

POL Bit	LE Bit	PIRQ Asserted State
0	0	Falling edge
1	0	Rising edge
0	1	Low level
1	1	High level

Level/Edge register—The level/edge (LE) register is used to select the way in which a PIRQ <15:00> line detects an interrupt request. It allows the user to select either level or edge sensitive triggering. When a bit is set, the corresponding PIRQ line is level sensitive. When a bit is clear, the corresponding PIRQ line is edge sensitive. The polarity of the PIRQ line input is selected by the polarity register (POL). Figure 5 shows the register format.

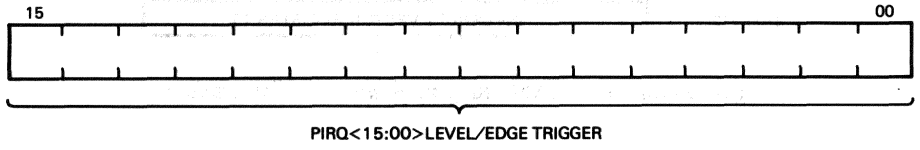


Figure 5 ▪ MicroVAX 78516 Level/Edge Register Format

Level-sensitive inputs allow more than one device to be connected to a single PIRQ line by using a wired NOR structure. Once the correct polarity level is detected by the VIC, the corresponding interrupt pending bit is set in the pending summary register (PSR). The interrupt pending bit will remain set until the PIRQ line is cleared. Therefore, an interrupt acknowledge cycle from the CPU will not clear the interrupt pending bit in the PSR register until the PIRQ line is deasserted. If a wired NOR structure is used, a external pullup resistors is required on the PIRQ line.

Edge sensitive inputs detect either a high-to-low (falling edge) or low-to-high (rising edge) transition. When the correct transition is detected, the corresponding bit in the PSR register will be set. The VIC will clear the bit when the interrupt is serviced and will not recognize another interrupt request on this line until the proper transition occurs. When the $\overline{\text{RESET}}$ line is asserted, the LE register is cleared.

Pending Summary register—The pending summary register (PSR) provides a summary of the internal interrupt pending flags. When a bit is set, an interrupt request is pending for the corresponding PIRQ line. When a bit is clear, no interrupt is pending for the corresponding PIRQ line. The contents of the PSR register are latched during a read and IACK cycle. The register format is shown in Figure 6.

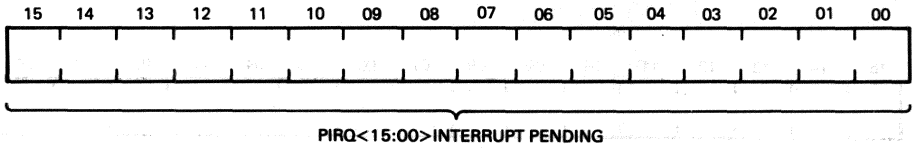


Figure 6 • MicroVAX 78518 Pending Summary Register Format

The VIC manages the setting and clearing the PSR register bits for level and edge sensitive PIRQ inputs as follows. When the $\overline{\text{RESET}}$ input is asserted, the PSR register is cleared.

- For level sensitive PIRQ inputs, the corresponding PSR bit will be set when the PIRQ line is asserted and cleared when line is deasserted.
- For edge sensitive PIRQ inputs, the corresponding PSR bit is set on the asserting edge of the PIRQ input. The PSR bit for a PIRQ input will be cleared by an interrupt acknowledge cycle that acknowledges the interrupt request of the corresponding PIRQ line, when the software clears the PSR bit by writing a zero into the appropriate bit, or when information is written into the LE register.

Interrupt Enable register—The interrupt enable (IEN) register is used to enable or disable the reporting of interrupts to the CPU by each PIRQ line. When a bit is set, it allows an interrupt request from the associated PIRQ line to generate an interrupt to the CPU. When a bit is clear, the associated PIRQ line is prevented from generating an interrupt to the CPU. The register format is shown in Figure 7. .

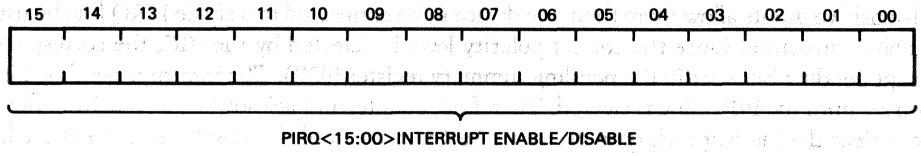


Figure 7 • MicroVAX 78516 Interrupt Enable Register Format

The IEN register enables or disables the generating of an interrupt to the CPU and does not affect the detection of interrupts by the VIC. When a PIRQ line is asserted, the corresponding bit in the PSR register is set regardless of the state of the IEN bit for the PIRQ line. The IEN register provides the support for a software interrupt polling scheme. The register is cleared when the $\overline{\text{RESET}}$ input is asserted.

IRQ Map registers (0-3)—The interrupt request map registers (IMAP0 through IMAP3) are used to select the IRQ line to be asserted by the VIC when a PIRQ line is asserted. When a bit in one of the IMAP registers is set, the corresponding PIRQ line is mapped to the associated IRQ line. The register format is shown in Figure 8. Each register corresponds to one of the IRQ outputs as defined in Table 4.

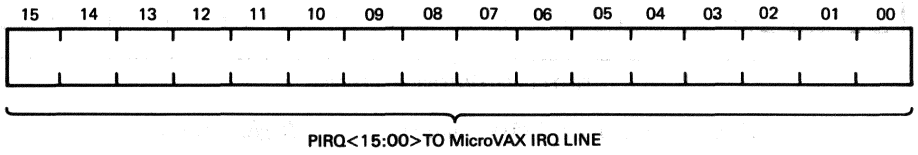


Figure 8 • MicroVAX 78516 IRQ Map Registers (0-3) Format

Table 4 • MicroVAX 78516 IMAP Register to IRQ Mapping

Register	Line
IMAP3	$\overline{\text{IRQ3}}$
IMAP2	$\overline{\text{IRQ2}}$
IMAP1	$\overline{\text{IRQ1}}$
IMAP0	$\overline{\text{IRQ0}}$

Example: If bit 3 of the IMAP1 register is set when the PIRQ3 line is asserted and the IEN register bit is set for this line, line IRQ1 will be asserted.

The IMAP registers are not initialized when the $\overline{\text{RESET}}$ line is asserted and the contents will be undefined until programmed by software.

Round Robin register—The round robin (ROBIN) register is used to select either fixed or round robin priority mode of operation for each IRQ level. More than one bit may be set in this register at a time and the register controls only the PIRQ lines for the asscociated VIC. The register is cleared when the $\overline{\text{RESET}}$ input is asserted. The register format is shown in Figure 9. Table 5 describes the function of each bit.

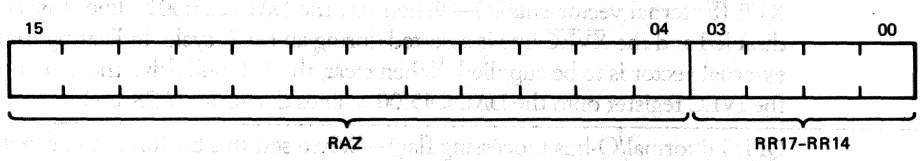


Figure 9 • MicroVAX 78516 Round Robin Register Format

Table 5 • MicroVAX 78516 Round Robin Register Description

Bit	Description
15:04	RAZ (Read as zeros)—Not used
03:00	RR17-RR14 (ROUND ROBIN IPL17-IPL14)—These bits select the priority mode for all interrupts mapped to lines $\overline{\text{IRQ<3:0>}}$. RR17 selects IRQ3 etc. When set, the round robin mode is selected. When cleared, the fixed mode is selected.

Interrupt Vector registers (0-15)—Each of the 16 interrupt vector (IVEC0 through IVEC15) registers contains a fully programmable 16-bit vector. There is an IVEC register for each PIRQ line. The register format is shown in Figure 10 and Table 6 describes the function of each bit.

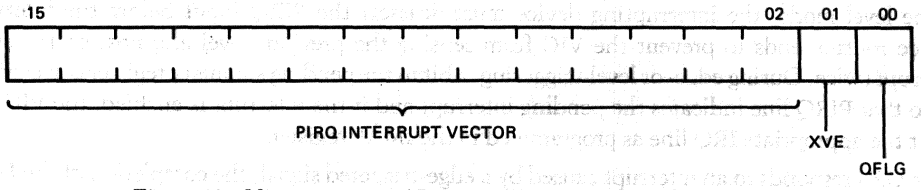


Figure 10 • MicroVAX 78516 Interrupt Vector Registers (0-15) Format

Table 6 • MicroVAX 78516 Interrupt Vector Registers (0-15) Description

Bit	Description
15:02	VECTOR (PIRQ interrupt vector)—This vector is the offset into the system control block (SCB) for the location of the interrupt routine.
01	XVE (External vector enable)—When set, the DAL<15:00> line drivers are disabled and the $\overline{\text{XVEC}}$ line is asserted during an IACK cycle, indicating that an external vector is to be supplied. When clear, the VIC will drive the contents of the IVEC register onto the DAL<15:00> lines during an IACK cycle.
00	QFLG (Normal/Q-bus processing flag)—When set, this bit forces the interrupt priority line of the CPU to priority IPL17 when servicing the interrupt. When clear, the CPU will service the interrupt normally.

These registers are not initialized when the $\overline{\text{RESET}}$ input is asserted and the contents of the register is undefined until programmed by software.

Interrupt Level Triggering and Edge Triggering

The sensing of an interrupt condition by the VIC may be programmed for each PIRQ input by the LE register. Each PIRQ line can be set to respond to either a signal level or to a signal transition (edge). The polarity of the sensed condition is also programmable.

In the edge-triggered mode, either a high-to-low or low-to-high transition on the PIRQ line will cause the VIC to latch the PIRQ line information. Further transitions on this PIRQ line will have no effect. After the acknowledgment of the latched assertion by the CPU, the VIC resets the latching mechanism allowing the user to again assert the interrupt with a proper transition on the PIRQ line. A latched PIRQ assertion may be cleared by writing to the LE register or by writing a zero to the corresponding bit of the pending status register.

In the level mode, the interrupting device must deassert the PIRQ input before the interrupt service routine ends to prevent the VIC from sensing the previous level and posting the same interrupt twice. During edge- or level-triggering, a bit in the pending summary register corresponding to that PIRQ line indicates the pending interrupt and if the interrupt is enabled, the VIC will assert the appropriate IRQ line as programmed in the IMAP register.

If the CPU responds to an interrupt caused by a edge-triggered signal, the completion of the IACK cycle will cause the VIC to clear the corresponding PSR register bit. If level-triggered mode was selected, the PSR bit would continue to reflect the PIRQ status.

Fixed and Round Robin Priority

The two priority modes available to the user are fixed and round robin. Each PIRQ line has a fixed priority with respect to the other PIRQ lines with line PIRQ15 as the highest priority and line PIRQ0 as the lowest.

In fixed priority mode, the highest pending PIRQ for the IRQ level being recognized by the CPU will be serviced first. In round robin mode, the highest pending PIRQ for the IRQ level being recognized by the CPU will be serviced and then prevented from requesting another interrupt until all other pending interrupts for that IRQ level have been serviced. When all pending interrupts assigned to an IRQ level have been serviced, the VIC will enable all the PIRQ lines assigned to that IRQ level and the round robin process will start again. The round robin mode operates only within the PIRQ lines of a specific VIC.

The VIC accepts as many as 16 interrupts from peripheral devices and drives an interrupt request (IRQ) line of the CPU, as determined by the user. The VIC decodes the presence of an interrupt acknowledge cycle on the bus, monitors the interrupt priority line (IPL) being recognized, and sends a 13-bit vector to the CPU. Each of the 16 (IRQ) lines is configured by software as follows:

- triggering mode and polarity
- IRQ mapping to the CPU
- enabling/disabling of interrupt request
- an interrupt vector

External Vector Generation

External devices can generate their own vector under control of a bit in the IVEC register. The vector generation sequence is as follows:

1. The VIC provides the external vector enable ($\overline{\text{XVEC}}$) signal to the external logic that generates the vector.
2. The $\overline{\text{XVEC}}$ signal indicates that the interrupt requested is being acknowledged.
3. The external logic supplies a vector to the $\overline{\text{CPU}}$ and asserts an $\overline{\text{RDY}}$ to end the bus cycle.

Daisychain Configuration

More than one VIC can be connected in a daisychain-enable configuration as shown in Figure 11. The three signals used are the acknowledge enable in ($\overline{\text{IAKEI}}$), interrupt acknowledge enable out high (IAKEOP), and interrupt acknowledge enable out low/error low ($\overline{\text{IAKEON}}$). When the $\overline{\text{IAKEI}}$ signal is asserted, the VIC can respond to the current interrupt acknowledge cycle. If no pending interrupts exist for the IPL line being acknowledged, the VIC asserts the $\overline{\text{IAKEON}}$ and IAKEOP lines to allow the next device in the chain to respond to the interrupt acknowledge cycle.

Interrupt Operation

The VIC receives interrupt requests from devices and posts interrupts to the CPU by asserting the appropriate IRQ lines. It also provides a vector address to the CPU during an interrupt acknowledge cycle if the XVE bit in the interrupt register is not set. If it is set, the VIC notifies the device that a vector address is required from the device.

Posting interrupts—When the VIC detects an assertion on a PIRQ input from a device, it sets a bit in the in Pending Summary register that corresponds to the PIRQ input. If the corresponding interrupt enable (IEN) bit in Interrupt Enable register is also set, an IRQ line is set to notify the CPU of the interrupt request. The IRQ line that is set is selected by one of the four Interrupt Map registers. Figure 11 shows the sequence for posting an interrupt request.

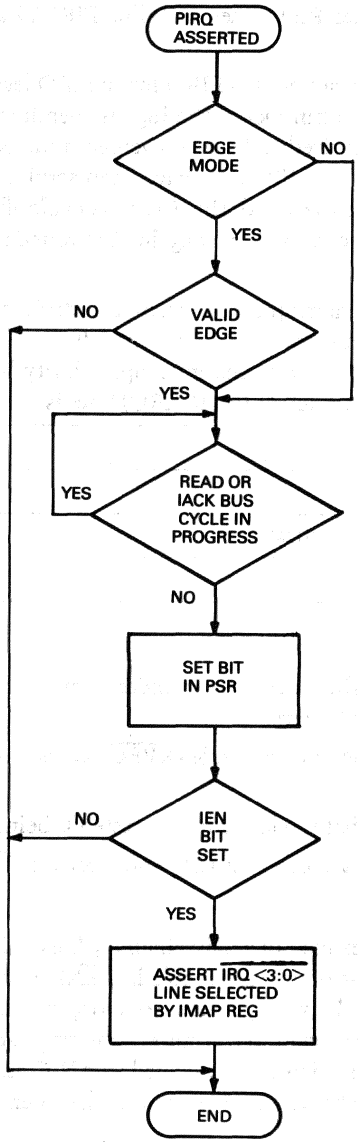


Figure 11 • MicroVAX 78516 Interrupt Request Posting Sequence

Interrupt acknowledge response—After the IRQ line is asserted, The CPU responds with an interrupt acknowledge cycle that transfers the interrupt priority level (IPL) of the interrupt being acknowledged on the DAL<04:00> lines. The VIC decodes this information to determine if it had requested the interrupt. If it had made the request, and the $\overline{\text{IAKEI}}$ input is asserted, the VIC blocks the propagation of the $\overline{\text{IAKEI}}$ signal to another VIC, and select the vector address associated with the PIRQ line that requested the interrupt. The VIC then transfers the vector to the CPU

through the DAL<15:00> lines. If an interrupt is not pending at the IPL being acknowledged, the VIC passes control to the next device in the daisychain by asserting the $\overline{\text{IAKEON}}$ signal and by placing the IAKEOP line in a high-impedance state. The interrupt acknowledge response sequence is shown in Figure 12.

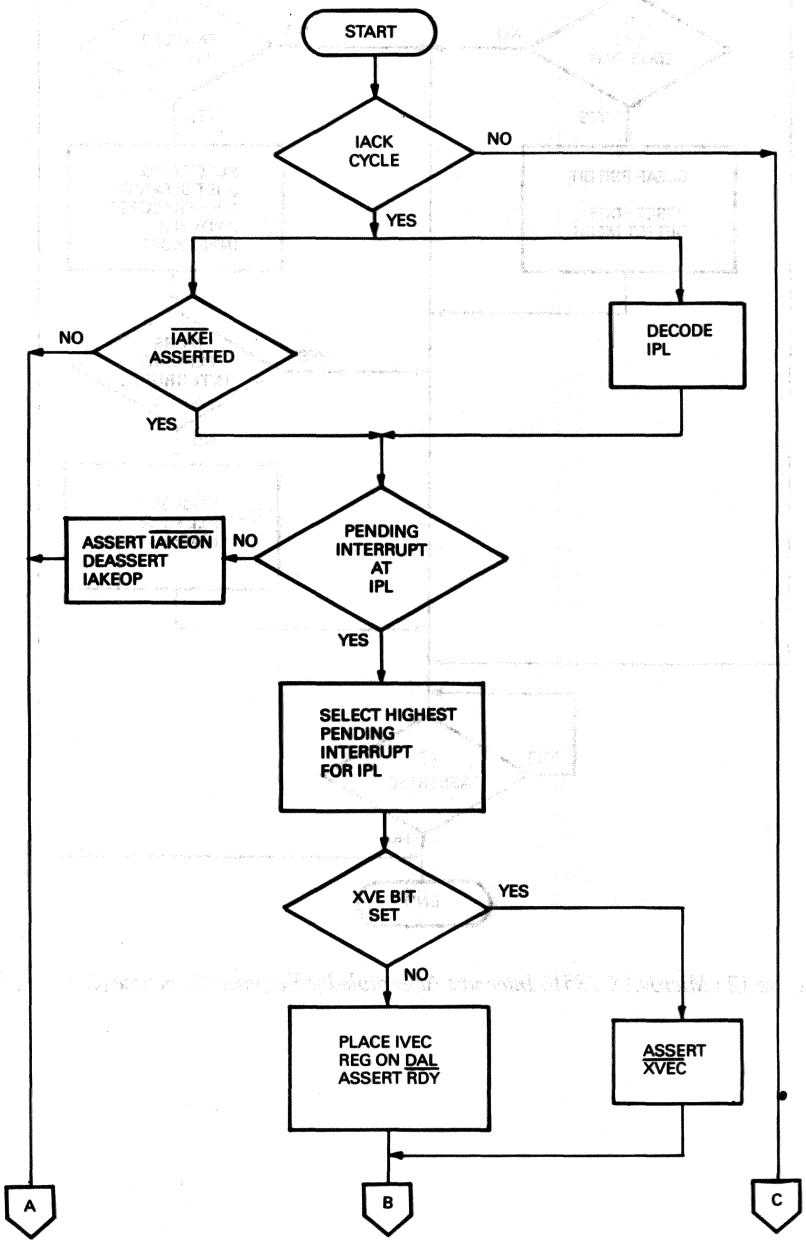


Figure 12 • MicroVAX 78516 Interrupt Acknowledge Response Sequence

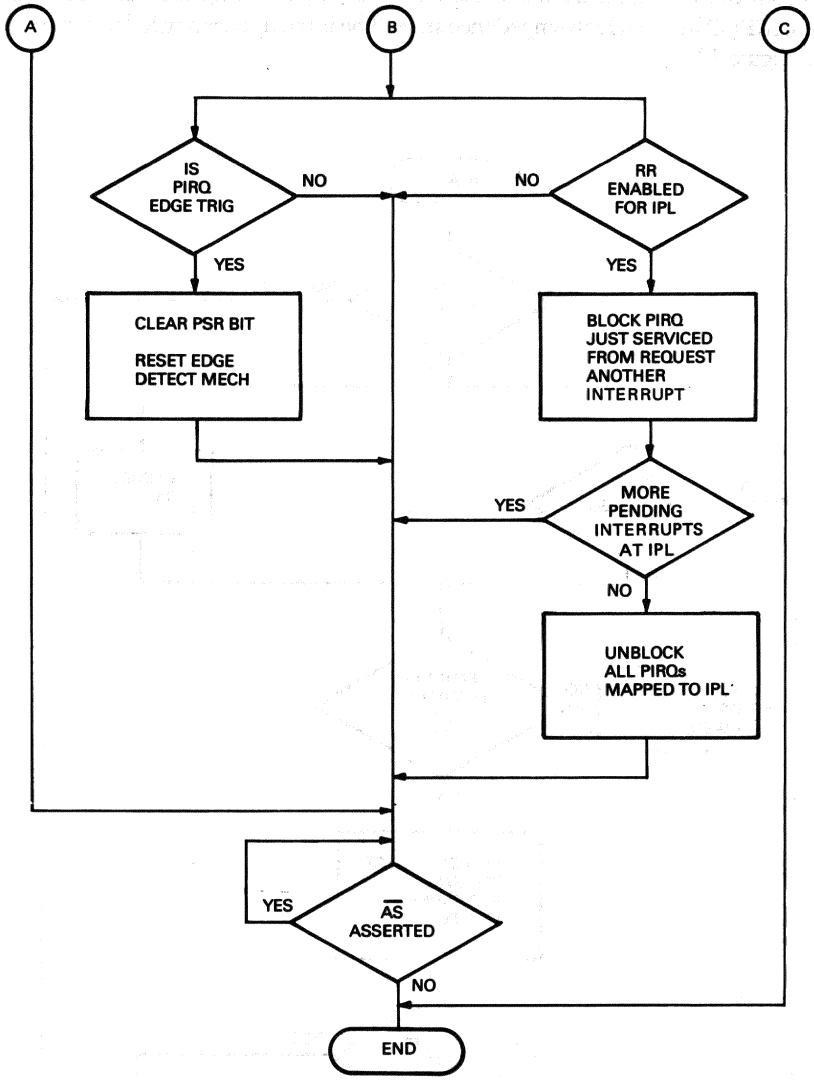


Figure 12 ▪ MicroVAX 78516 Interrupt Acknowledge Response Sequence (Continued)

• Interfacing Requirements

The VIC can be used with the MicroVAX 78031 CPU, CVAX 78034 CPU, or the rtVAX 78R32 CPU. It can be connected to either the CPU bus or to a bufferred I/O bus. A typical example of the VIC connected to a MicroVAX 78032 CPU is shown in Figure 13. The circuit includes separate address decode logic to assert the $\overline{\text{CSEL}}$ input that is used to select registers in the VIC.

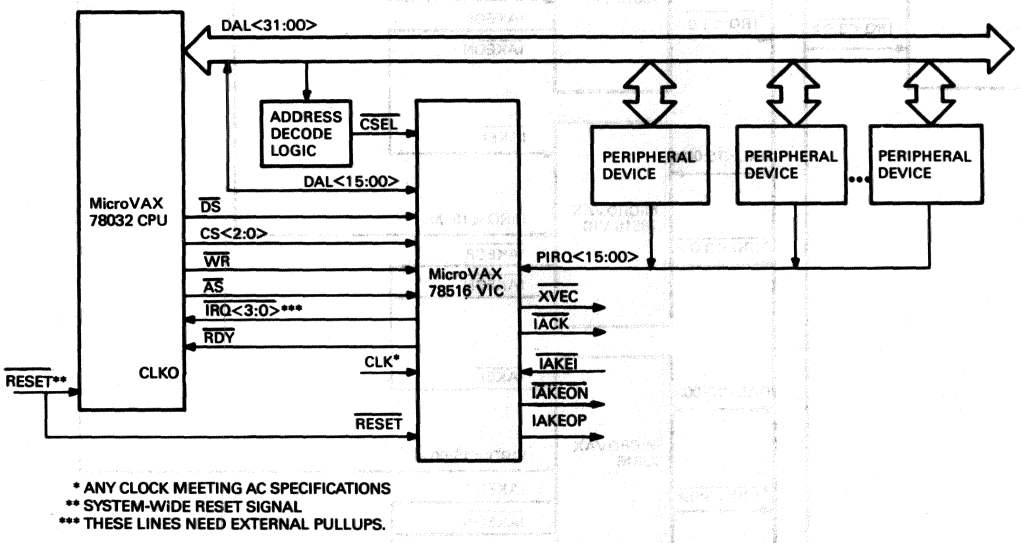


Figure 13 • MicroVAX 78516 Typical VIC and MicroVAX 78032 CPU Interface Configuration

Daisychain Wiring

Figure 14 is an example of a MicroVAX 78032 CPU and than one VIC and wired in a daisychain-enable configuration. The inital $\overline{\text{IAKEI}}$ input to the VIC is held asserted by a ground connection to allow the VIC to respond to to the current interupt acknowledge cycle. If a pending interrupt does not exist for the IPL being acknowledged, the VIC asserts the $\overline{\text{IAKEON}}$ and IAKEOP lines to allow the next device in the chain to respond to the interrupt acknowledge cycle.

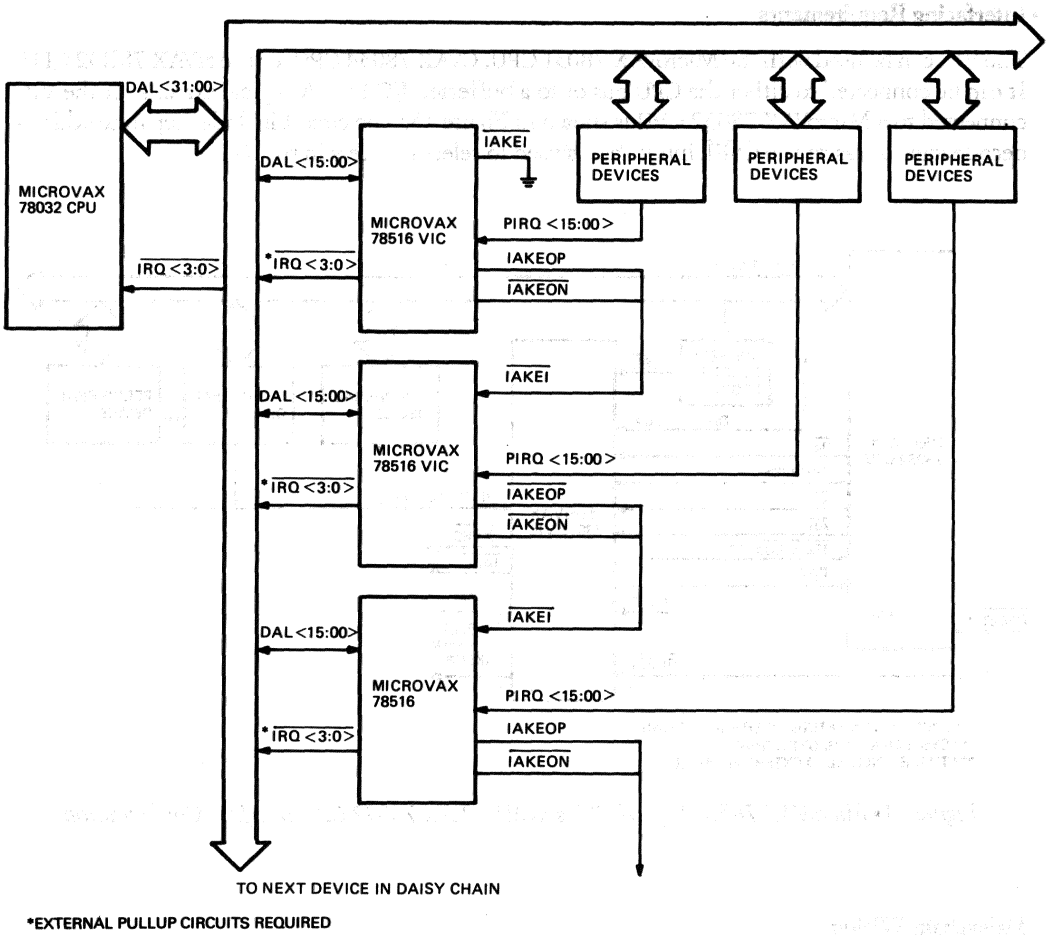


Figure 14 • MicroVAX 78516 IAK Daisychain Wiring Configuration

Bus Cycles

The VIC responds to read cycles, write cycles, and interrupt acknowledge cycles.

Register read cycle—A read cycle is performed by the CPU to read information from a VIC internal register. The VIC responds to the read cycle when the $\overline{CSEL}$ input is asserted and when the address on the DAL<15:00> lines corresponds to the address of a VIC internal register. The signal timing and parameters for register read cycle are shown in the Specification section. The register read cycle sequence follows.

1. The CPU transfers an address on the DAL<15:00> lines, indicates a read cycle on control lines CS<2:0>, and deasserts the $\overline{WR}$ input to the VIC.
2. The CPU asserts the $\overline{AS}$ signal to indicate that the information on the bus is valid.
3. The VIC latches the DAL<06:00>, CS<2:0>, and $\overline{WR}$ line information.
4. The external address decode logic, Figure 13, decodes the address on the DAL lines and asserts the $\overline{CSEL}$ input to the VIC.

5. The VIC transfers the content of the selected register to the bus and asserts the $\overline{RDY}$ output.
6. The CPU latches the data and deasserts the $\overline{DS}$ and $\overline{AS}$ outputs to indicate the end of the bus cycle. The VIC then deasserts the $\overline{RDY}$ output to the CPU to end the read cycle.

Register write cycle—A write cycle is performed by the CPU to write information into a VIC internal register. The VIC responds to the write cycle when the $\overline{CSEL}$ input is asserted by the CPU and the address on the DAL<06:00> lines corresponds to the address of a VIC internal register. The signal timing and parameters for register write cycle are in the Specification section. The register write cycle sequence follows.

1. The CPU transfers an address on the DAL<15:00> lines, indicates a write cycle on control lines CS<2:0>, and asserts the $\overline{WR}$ input to the VIC.
2. The CPU asserts the $\overline{AS}$ signal to indicate that the information on the bus is valid.
3. The VIC latches the DAL<06:00>, CS<2:0>, and $\overline{WR}$ line information.
4. The external address decode logic, Figure 13, decodes the address on the DAL lines and asserts the $\overline{CSEL}$ input to the VIC.
5. The VIC asserts the $\overline{RDY}$ signal, latches the data on the DAL<15:00> lines into the selected register, and deasserts the $\overline{RDY}$ output when the CPU deasserts the $\overline{DS}$ input.
6. The CPU deasserts the $\overline{AS}$ output to indicate the end of the bus cycle which also ends the register write cycle.

Interrupt acknowledge cycle—The CPU performs an interrupt acknowledge cycle in response to an interrupt request ($\overline{IRQ<3:0>}$) output from the VIC. The VIC responds to the interrupt acknowledge bus cycle when its $\overline{IAKEI}$ input is asserted and when its pending interrupt is the same interrupt level recognized by the CPU. The signal timing and parameters for the interrupt acknowledge cycle are in the Specification section. The sequence of the interrupt acknowledge cycle sequence follows.

1. The CPU transfers a hexadecimal value of the IPL on the DAL<15:00> lines, indicates an interrupt acknowledge cycle on control lines CS<2:0>, and deasserts the $\overline{WR}$ input to the VIC.
2. The CPU asserts the $\overline{AS}$ signal to indicate that the information on the bus is valid.
3. The VIC latches the DAL<06:00>, CS<2:0>, and $\overline{WR}$ line information.
4. The $\overline{IAKEI}$ input is asserted. The timing of this event depends on the location of the VIC in the daisychain.
5. The VIC transfers the vector address on the DAL<15:00> lines and asserts the $\overline{RDY}$ signal. If the device is required to supply the vector address, the VIC asserts the $\overline{XVEC}$ output to the device and the device asserts the $\overline{RDY}$ signal.
6. The CPU latches the vector address and deasserts the $\overline{DS}$ and $\overline{AS}$ outputs to indicate the end of the bus cycle. The VIC then deasserts the $\overline{RDY}$ output, if asserted, to end the cycle.

Power Supply Decoupling

Figure 15 shows the power supply connections to the VIC. The V_{DD} pins connect to 5 Vdc and the V_{SS} pins connect to a common ground. All the V_{DD} pins should be connected together and all the V_{SS} pins should be connected together. Decoupling is provided by connecting 1.0u f capacitors between the V_{DD} and V_{SS} pins as shown on the figure.

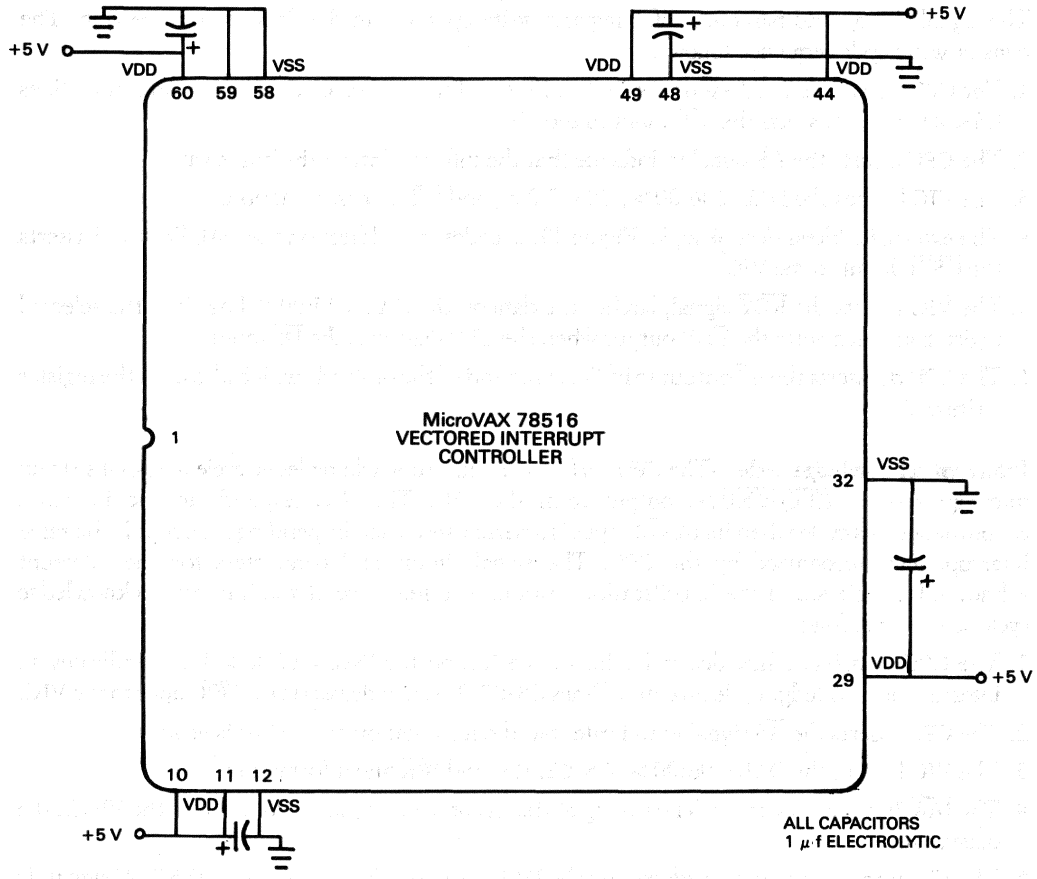


Figure 15 • MicroVAX 78516 Power Supply Decoupling

• Specifications

The mechanical, electrical, and environmental specifications for the VIC are contained in the following paragraphs. The test conditions for the values specified are listed as follows unless specified otherwise.

- Temperature: 70°C
- Power supply voltage (V_{DD}): 4.75 V
- Ground (V_{SS}): 0 V

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Storage temperature range: –55°C to 125°C
- Active temperature range: 0°C to 70°C
- Power supply voltage (V_{DD} to V_{SS}): 0 V to 6 V
- Input or output voltage applied: –0.3 V to (V_{DD} +0.3 V)

Recommended Operating Conditions

- Temperature: 0°C to 70°C
- Power supply voltage: 4.75 V to 5.25 V
- Power dissipation: 1.0 W (maximum)

dc Electrical Characteristics

The dc input and output parameters for the VIC are listed in Table 7.

Table 7 • MicroVAX 78516 dc Input and Output Parameters

Symbol	Parameter	Test Conditions	Requirements Min.	Requirements Max.	Units
V_{IH}	High-level input voltage		2.0	—	V
V_{IL}	Low-level input voltage		—	0.8	V
V_{OH}[1,2]	High-level output voltage	I_{OH} = –400 µA	2.4	—	V
V_{OL}[1]	Low-level output voltage	I_{OL} = 2.0 mA	—	0.4	V
I_{ILC}	Input leakage current	$0 < V_{in} < (V_{DD} - 0.6\ V)$	—	20	A
I_{OLC}	Output leakage current	$0 < V_{in} < (V_{DD} - 0.6\ V)$	—	100	µA
I_{CCAC}[3]	Active supply current		—	*	mA
V_{OLOD1}[4]	Open drain pulldown low-level output voltage	I_{OL} = 6.0 mA	—	0.4	V

Symbol	Parameter	Test Conditions	Requirements Min.	Max.	Units
V_{OLOD2}[5]	Open drain pulldown low-level output voltage	$I_{OL}=25$ mA	—	0.4	V
C_{in}	Input capacitance		—	*	pF
C_{out}	Output capacitance		—	*	pF

*To be determined.

[1]Only one output may be shorted to either supply rail at one time, and the duration of the short must be less than 2 seconds.

[2]This specification also applies to the open drain output on IAKEOP

[3]All outputs floating, all inputs connected to either supply rail. The CLK input is fully swinging between both supply rails at 20 MHz.

[4]Applies only to the $\overline{IRQ}<3:0>$ outputs.

[5]Applies only to the $\overline{RDY}$ and $\overline{IAKEON}$ outputs.

ac Electrical Characteristics

Figure 16 shows the input signal and clock signal waveforms and the parameters are listed in Table 8.

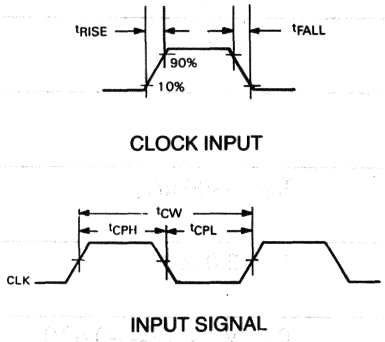


Figure 16 • MicroVAX 78516 Input and Clock Signal Timing

Table 8 • MicroVAX 78516 Input and Clock Signal Timing Parameters

Symbol	Definition	Requirements Min.	Max.
t_{CPG}	Input clock high	5.1 ns	500 µs
t_{CPL}	Input clock low	5.1 ns	500 µs
t_{CW}	Input clock period	50 ns	[1]
t_{RISE}	Input signal rise		15 ns[2]
t_{FALL}	Input signal fall		15 ns[2]

[1]To be determined
[2]Measured between 10% and 90% levels. Applies to all inputs except PIRQ<15:00>. Maximum t_{RISE} and t_{FALL} times for PIRQ<15:00> is 500 ns.

Figure 17 and 18 show the signal timing and symbols for a register read cycle and register write cycle, respectively, between the MicroVAX CPU and VIC. Figure 19 shows the signal timing and symbols for an interrupt acknowledge cycle when the VIC responds with a vector and when the external device supplies the vector. Figure 20 shows the timing and symbols for a daisychain configuration when the interrupt priority is not passed to the VIC and when it is passed to the VIC. Figure 21 shows the signal timing and symbols for the PIRQ input to IRQ output signal generation. It also includes the $\overline{\text{RESET}}$ input signal timing. Table 9 lists and defines the symbols and parameters used on the figures. The following notes apply to the table information.

- (T) = input clock period (t_{CW})
- All units are nanoseconds (ns) except where indicated.
- All times are specified with a 100-pF capacitive load on the outputs.
- All times are measured at the 50 percent levels of the waveforms except where indicated.

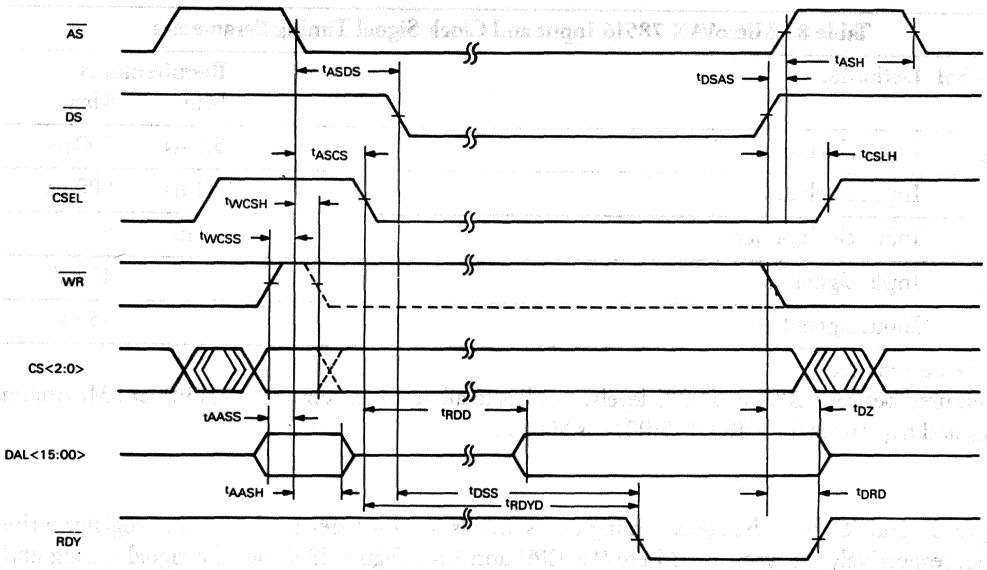


Figure 17 ▪ MicroVAX 78516 Register Read Cycle Timing

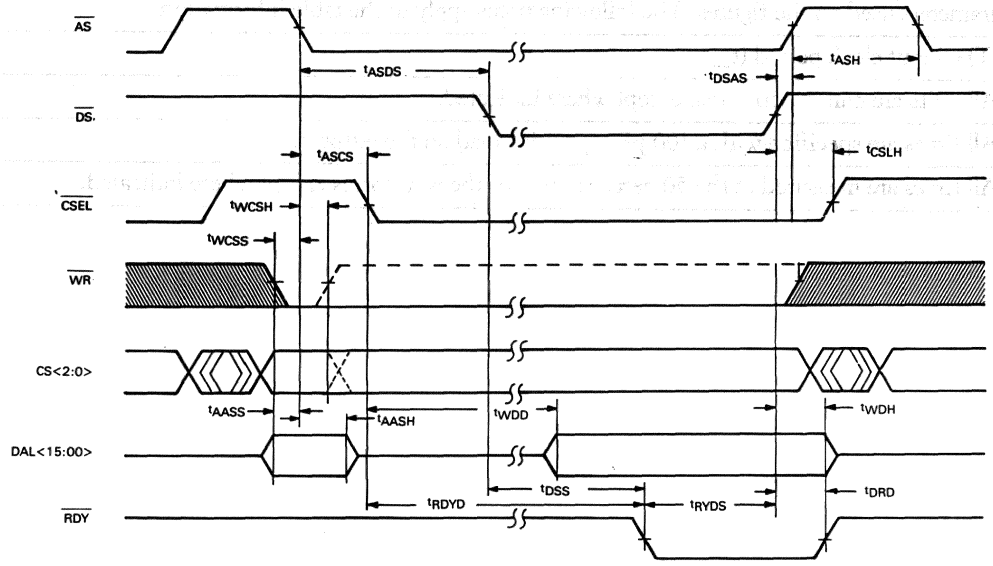
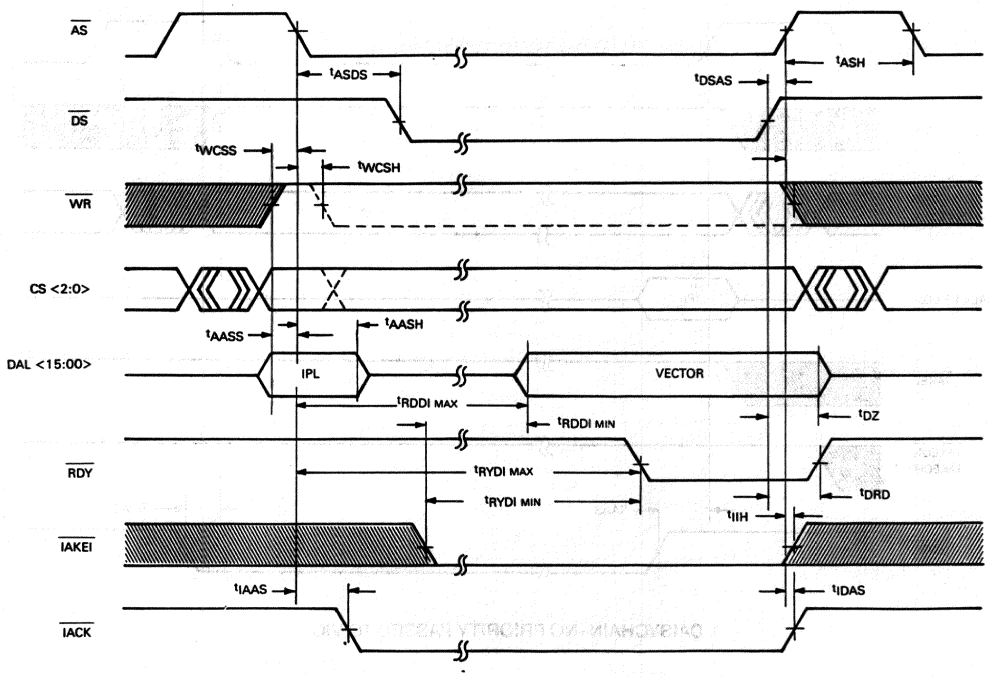
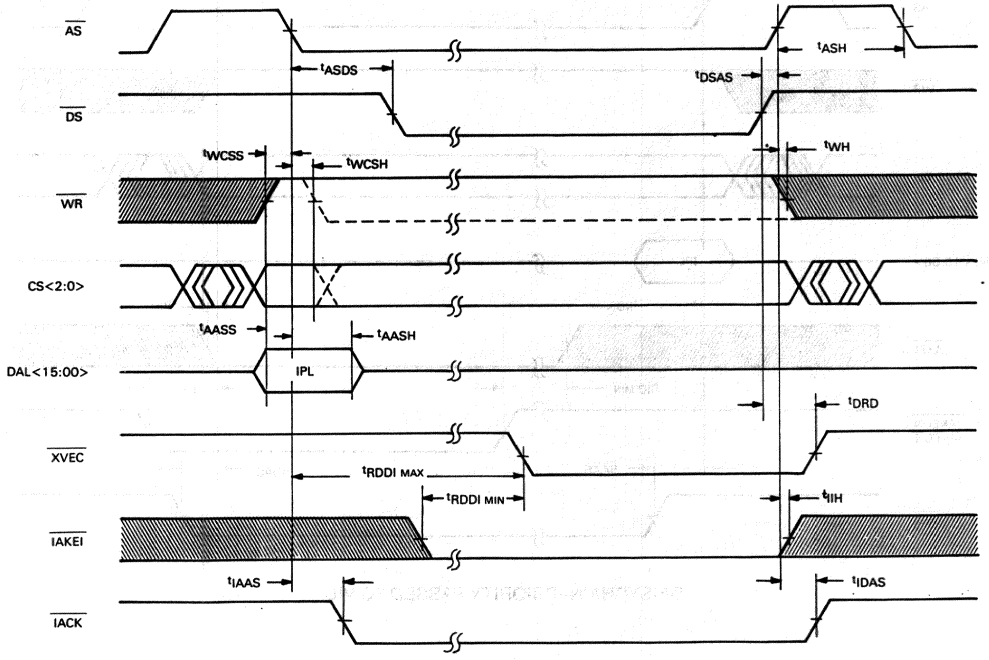


Figure 18 ▪ MicroVAX 78516 Register Write Cycle Timing

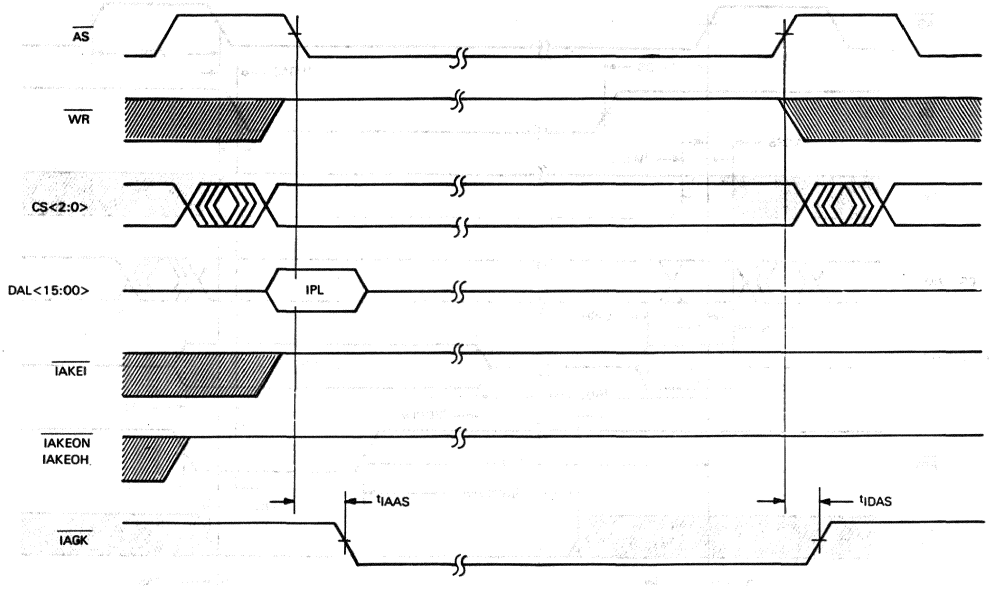


VIC SUPPLIES VECTOR

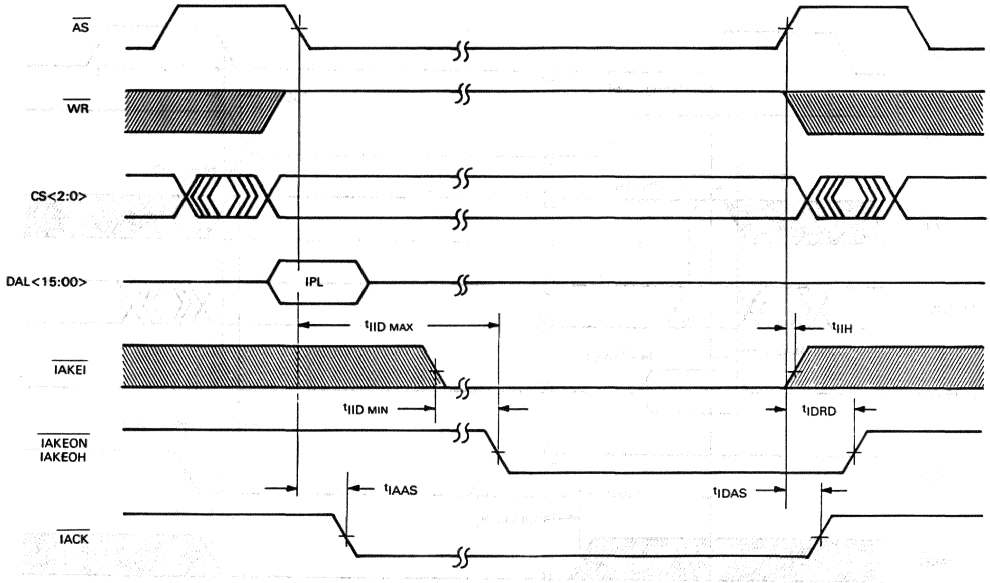


EXTERNAL DEVICE SUPPLIES VECTOR

Figure 19 ▪ MicroVAX 78516 Interrupt Acknowledge Cycle Timing

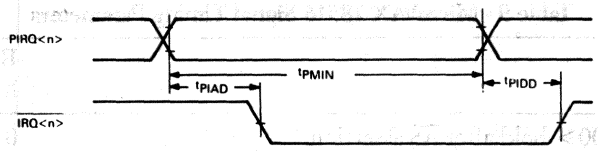


DAISYCHAIN–NO PRIORITY PASSED TO VIC

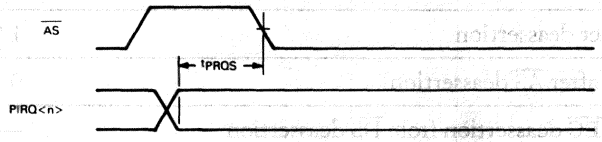


DAISYCHAIN–PRIORITY PASSED TO VIC

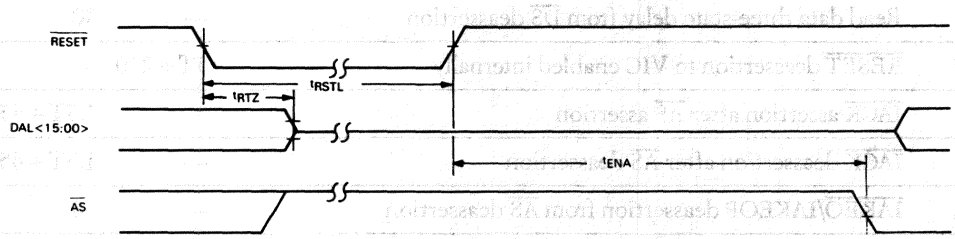
Figure 20 ▪ MicroVAX 78516 Daisychain Priority Signal Timing



PIRQ ASSERTED/DEASSERTED TO IRQ ASSERTED/DEASSERTED



PIRQ SETUP



RESET TIMING

Figure 21 ▪ MicroVAX 78516 PIRQ and Reset Signal Timing

Table 9 • MicroVAX 78516 Signal Timing Parameters

Symbol	Definition	Requirements Min.	Max.
t_{AASH}	DAL<06:00> hold after $\overline{AS}$ assertion	0	—
t_{AASS}	DAL<06:00> setup to $\overline{AS}$ assertion	15	—
t_{ASCS}	$\overline{CSEL}$ assertion after $\overline{AS}$ assertion	—	1 µs
t_{ASDS}	$\overline{DS}$ assertion after $\overline{AS}$ assertion	0	—
t_{ASH}	$\overline{AS}$ high after deassertion	1.5T	—
t_{CSLH}	$\overline{CSEL}$ hold after $\overline{AS}$ deassertion	0	—
t_{DRD}	$\overline{RDY}$ or $\overline{XVEC}$ deassertion from $\overline{DS}$ deassertion	—	1.5T+45
t_{DSAS}	$\overline{AS}$ deassertion after $\overline{DS}$ deassertion	0	—
t_{DSS}	$\overline{DS}$ setup before $\overline{RDY}$ assertion	30	—
t_{DZ}	Read data three-state delay from $\overline{DS}$ deassertion	—	30
t_{ENA}[1]	$\overline{RESET}$ deassertion to VIC enabled internally	5T+250	—
t_{IAAS}	$\overline{IACK}$ assertion after $\overline{AS}$ assertion	—	1.5T+45
t_{IDAS}	$\overline{IACK}$ deassertion after $\overline{AS}$ deassertion	—	1.5T+45
t_{IDRD}	$\overline{IAKEO}$/IAKEOP deassertion from $\overline{AS}$ deassertion	—	40
t_{IIDmin}	$\overline{IAKEON}$/IAKEOP delay from $\overline{IAKEI}$ assertion ($\overline{IAKEI}$ asserted 7.5T or more after $\overline{AS}$	25	—
t_{IIDmax}	$\overline{IAKEO}$/IAKEOP delay from $\overline{AS}$ assertion ($\overline{IAKEI}$ asserted less than 7.5T more after $\overline{AS}$	—	8.5T+25
t^{IIH}	$\overline{IAKEI}$ hold after $\overline{AS}$ deassertion	0	—
t_{PIAD}[2]	PIRQ assertion to IRQ assertion delay	0	100
t_{PIDD}	PIRQ deassertion to IRQ deassertion delay (applicable to level triggering only)	—	100
t_{Pmin}	PIRQ minimum assertion width (applicable to edge triggering only)	90	—
t_{PRQS}[3]	PIRQ setup (proper level/edge) before $\overline{AS}$ assertion	50	—
t_{RDD}	Read data delay from $\overline{CSEL}$ assertion	6.5T	7.5T+25
$t_{RDDImin}$	Read data or $\overline{XVEC}$ delay from $\overline{IAKEI}$ assertion ($\overline{IAKEI}$ asserts 7.5T or more after $\overline{AS}$)	6T	—
$t_{RDDImax}$	Read data or $\overline{XVEC}$ delay from $\overline{AS}$ assertion ($\overline{IAKEI}$ asserted less than 7.5T after $\overline{AS}$)	—	13.5T+25
t_{RDYD}	$\overline{RDY}$ delay from $\overline{CSEL}$ assertion	8.5T	9.5T+25

Symbol	Definition	Requirements Min.	Max.
t_{RSTL}[4]	Minimum $\overline{RESET}$ low time	200 µs	—
t_{RTZ}	$\overline{RESET}$ assertion to DALs three-state	—	100
$t_{RYDImin}$	$\overline{RDY}$ delay from $\overline{IAKEI}$ assertion ($\overline{IAKEI}$ asserted 7.5T or more after $\overline{AS}$)	8T	—
$t_{RYDImax}$	$\overline{RDY}$ delay from $\overline{AS}$ assertion ($\overline{IAKEI}$ asserted less than 7.5T after $\overline{AS}$)	—	15.5T+25
t_{RYDS}	$\overline{DS}$ deassertion from $\overline{RDY}$ assertion	0	—
t_{WCSH}	CS<2:0>, $\overline{WR}$ hold after $\overline{AS}$ assertion	0	—
t_{WCSS}	CS<2:0>, $\overline{WR}$ setup to $\overline{AS}$ assertion	15	—
t_{WDD}	Write data delay from $\overline{CSEL}$ assertion	—	3.5T–5
t_{WDH}	Write data hold after $\overline{DS}$ deassertion	20	—

[1]V_{DD} must be greater than or equal to 4.75 V during this period.

[2]Maximum time is 100 ns unless a PIRQ line is asserted during a register read or an IACK cycle. In these cases the IRQs will be asserted 100 ns after the end of the register read or IACK cycle.

[3]This ensures that the PIRQ signal will be recognized if the bus cycle started by the $\overline{AS}$ signal is a read or interrupt acknowledge cycle. Otherwise the VIC will not recognize the PIRQ signal until the bus cycle has ended.

[4]The VIC requires 5T+250 ns after $\overline{RESET}$ is deasserted to complete its internal reset. The $\overline{AS}$ signal should not be asserted until after this delay.

Mechanical Configuration

The MicroVAX 78516 is available as a 68 pin cerquad surface mount package or socket mount package. The physical dimensions of each package is contained in Appendix E.

Symbol	Definition	Requirements Min	Max
t_{wl}[1]	Minimum $\overline{PWOK}$ low time	200 ns	—
t_{rf}	$\overline{PWOK}$ rise/fall time (Table 6-10)	—	30
$t_{wh(a)}$	$\overline{PWOK}$ high to $\overline{PWR}$ asserted, $\overline{PWR}$ asserted $\overline{DST}$ or more (Table 7-2)	50 ns	—
$t_{s(a)}$	$\overline{MSE}$ delay from $\overline{PWR}$ asserted or $\overline{DST}$ asserted (Table 7-2)	—	1.5 T − 25
$t_{w(c)}$	$\overline{DS}$ deassertion time $\overline{PWR}$ asserted	0	
$t_{h(a)}$	$\overline{CS}$ + $\overline{DS}$ hold after $\overline{AS}$ (Table 7-2)	0	—
$t_{s(a)}$	$\overline{CS}$ + $\overline{DS}$ setup to $\overline{AS}$ assertion	15	—
$t_{w(a)}$	$\overline{AS}$ minimum delay from $\overline{PWR}$ assertion	—	3.5 T − 5
$t_{w(c)}$	Write data hold after $\overline{DS}$ deassertion	30	—

[1] t_{wl} must be greater than or equal to t_{rf} during the period.

Minimum delay of 300 ns allows the 8100 time to be used as a timing reference and an $\overline{AS}$ cycle to begin as soon as the $\overline{PWR}$ will be asserted. This allows the use of the regular memory $\overline{AS}$ cycle.

The assumption that the $\overline{PWR}$ signal will be asserted 1.5 times the bus cycle is met by the $\overline{PWR}$ signal has an internal delay of one half bus cycle. The $\overline{PWR}$ signal will be asserted 1.5 T + 25 ns after $\overline{PWR}$ is asserted to maintain the normal timing. The $\overline{AS}$ signal also is delayed by one half bus cycle.

Mechanical Description

The MicroVAX 78532 is an hybrid with a 64-pin dual-in-line package (DIP) and multiple analog or digital modules. The physical dimensions are shown in Figure 6-1 and Figure 6-2 in Appendix B.

Features

- Virtual memory DMA (direct memory access) controller, compatible with VAX and MicroVAX architectures
- Full 32-bit architecture and implementation
- Four independently programmable channels
- Performs byte alignment to accommodate information on arbitrary byte boundaries
- Operates with 8-, 16-, and 32-bit I/O devices
- Byte/word packing for increased throughput
- Peak data transfer rate of 10 Mbytes/sec
- Maximum DMA transfer length of 1 Gigabyte
- Maximum I/O bus address space of 16 Mbytes
- Provides simple interface to MicroVAX bus for intelligent I/O subsystems
- High-speed CMOS technology
- Single 5-Vdc power supply

Description

The MicroVAX 78532 MicroDMA controller is a high-performance, dual-ported, four-channel virtual memory DMA controller. Figure 1 is a block diagram of the MicroDMA controller.

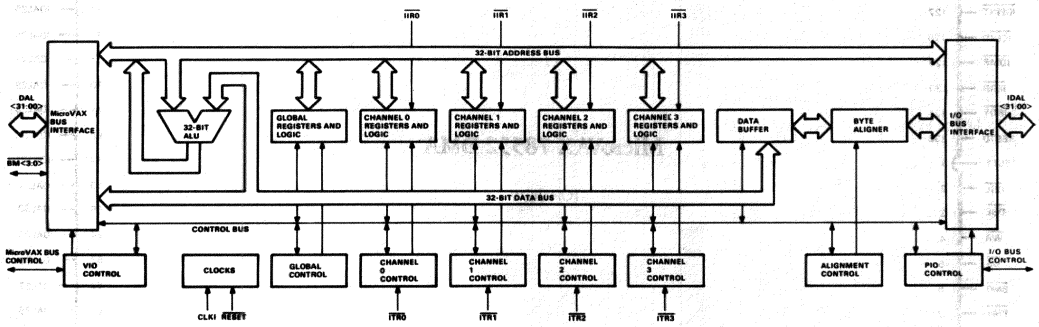


Figure 1 • MicroVAX 78532 MicroDMA Block Diagram

It provides two buses, one for the MicroVAX interface and one for the I/O devices. It interfaces the 32-bit MicroVAX bus with high-speed peripheral devices or intelligent I/O subsystems on the 8-, 16-, or 32-bit I/O bus. The MicroDMA is used for the following applications:

- For DMA transfers between memory (or devices) on the MicroVAX bus and memory (or devices) are the I/O bus
- For DMA transfers between memory and devices on the I/O bus
- As a window into MicroVAX memory for devices on the I/O bus
- As an access port to devices on the I/O bus for the MicroVAX.

The MicroDMA is a virtual memory DMA controller with full VAX compatible memory management capabilities. It processes address translation for DMA transfers so that this function is transparent to the user. Page table information is accessed from MicroVAX memory and used directly without alteration. The MicroDMA also performs data buffering and byte alignment for transfers between the MicroVAX bus and the I/O bus.

• Pin and Signal Definitions

The input and output signals and power and ground connections of the MicroDMA controller 132-pin package are shown in Figure 2. The signals are defined in the following paragraphs in two groups—signals that connect to the MicroVAX bus and signals that connect to the I/O bus. The power and ground connections are defined with the I/O bus signals.

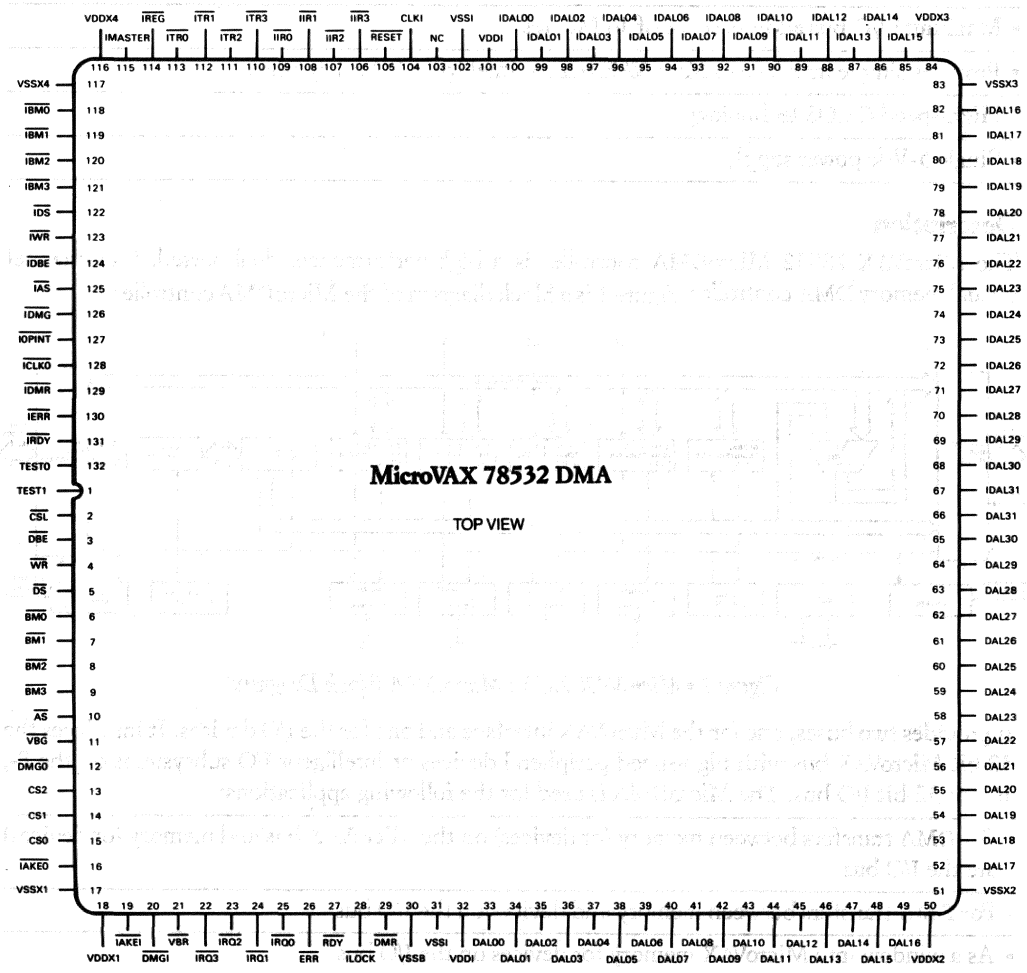


Figure 2 • MicroVAX 78532 Pin Assignments

MicroVAX Bus Interface Signals

The following lines connect to the MicroVAX bus and include data and address lines, bus control and status lines, interrupt control lines, and a clock input line. The bus signals are summarized in Table 1. A more detailed description of the signal functions is contained in the following paragraphs.

Table 1 • MicroVAX 78532 MicroVAX Bus Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
66-52 49-33	DAL<31:00>	input/output	Data and address lines <31:00>—Time multiplexed lines used to transfer data and address information between the MicroDMA and devices on the MicroVAX bus.
10	$\overline{AS}$	input/output	Address strobe—Asserted to indicate a valid address on the DAL<31:00> lines.
5	$\overline{DS}$	input/output	Data strobe—Asserted during a read cycle to indicate that the DAL<31:00> lines are available to receive data and deasserted to indicate that data has been received. Asserted during a write cycle to indicate that data is available on the DAL<31:00> lines and deasserted when the data is to be removed.
2	$\overline{CSL}$	input	Asserted to indicate a reference in the I/O bus range or to a MicroDMA register defined by the DAL<31:00> lines.
9-6	$\overline{BM<3:0>}$	input/output	Byte mask—Specifies the bytes of the DAL<31:00> lines that contain valid data.
4	$\overline{WR}$	input/output	Write—Specifies the direction of data transfer. Asserted to indicate that the source of data is the current MicroVAX bus master. Can be used to control the direction of the transceivers. Valid when the $\overline{AS}$ signal is asserted.
3	$\overline{DBE}$	input/output	Data buffer enable—When asserted, the DAL<31:00> transceivers and buffers are enabled.
27	$\overline{RDY}$	input/output	Ready—Used to synchronize data transfers between devices operating at different transfer rates on the MicroVAX bus. The current bus master must wait for the assertion of this signal before terminating the cycle and removing data from the bus.
26	$\overline{ERR}$	input/output	Error—Asserted to indicate an error condition in the current MicroVAX cycle.

Pin	Signal	Input/Output	Definition/Function
13-15	CS<2:0>	input/output	Control status—Indicates the type of cycle being performed on the MicroVAX bus.
22-25	$\overline{\text{IRQ<3:0>}}$	output	Interrupt request—Interrupt request lines for the devices on the MicroVAX bus that are maskable.
29	$\overline{\text{DMR}}$	output	DMA request—Asserted to indicate that a device on the MicroVAX bus is requesting control of the MicroVAX bus.
20	$\overline{\text{DMGI}}$	input	DMA grant input—Asserted to cause the MicroDMA controller to become bus master of the MicroVAX bus and perform a DMA transfer.
12	$\overline{\text{DMGO}}$	output	DMA grant output—Used only with systems having more than one MicroDMA controllers. Asserted when a DMA request is to be granted to another controller in the daisychain.
19	$\overline{\text{IAKEI}}$	input	Interrupt acknowledge enable in—Asserted to allow the MicroDMA to respond to a MicroVAX interrupt acknowledge.
16	$\overline{\text{IAKEO}}$	output	Interrupt acknowledge enable out—Asserted when an interrupt acknowledge is to be processed by another device in the interrupt daisychain.
104	CLKI	input	Clock in—A TTL clock input used for timing.
103	NC		No connection.

MicroVAX Bus Interface Lines

Data and Address (DAL<31:00>)—These are time-multiplexed bidirectional lines used to transfer data and address information between the MicroDMA controller and other MicroVAX bus devices such as MicroVAX CPU and MicroVAX memory. The strobe signals $\overline{\text{AS}}$ and $\overline{\text{DS}}$ determine whether data or address information is transferred.

Address Strobe ($\overline{\text{AS}}$)—The falling edge of the $\overline{\text{AS}}$ signal indicates that lines DAL<31:00> contain a valid address. On the falling edge of $\overline{\text{AS}}$, the MicroDMA controller latches the address and interprets it as a physical address. If the $\overline{\text{CSL}}$ line is also asserted, the CPU is performing a MicroDMA access operation. This access could be to a MicroDMA register or to a location in I/O bus memory space. Refer to the Access Operations section.

If the MicroDMA controller is bus master, the $\overline{\text{AS}}$ line indicates that lines DAL<29:02>) contain a valid MicroVAX physical memory address. The DAL<31:30> lines contain a 1 and a 0, respectively.

The falling edge of the $\overline{\text{AS}}$ signal also indicates that the information is valid on the $\overline{\text{BM<3:0>}}$, CS<2:0>, and $\overline{\text{WR}}$ lines.

Data Strobe ($\overline{DS}$)—This signal provides timing information for the data transfer portion of a read or write cycle. During a read cycle, the falling edge of $\overline{DS}$ signal indicates that the DAL<31:00> lines are available to receive data and the rising edge indicates that the data is about to be latched. During a write cycle, the falling edge of the $\overline{DS}$ signal indicates that data is present on the DAL<31:00> lines and the rising edge indicates that the data is about to be removed.

Byte Masks ($\overline{BM}$<3:00>)—These signals specify which bytes of the DAL<31:00> lines are valid during the current data transfer. During a write cycle, lines $\overline{BM}$<3:0> specify which bytes of the DAL<31:00> lines contain valid data. During a read cycle, lines $\overline{BM}$<3:0> specify which bytes of the DAL<31:00> lines must be supplied with valid data by an external device. The information on lines $\overline{BM}$<3:0> is valid on the falling edge of the $\overline{AS}$ signal. The byte mask assignments are shown in Table 2.

Table 2 • MicroVAX 78532 MicroVAX Bus Byte Mask Assignments

Byte Mask line	Data Valid
$\overline{BM3}$	DAL<31:24>
$\overline{BM2}$	DAL<23:16>
$\overline{BM1}$	DAL<15:08>
$\overline{BM0}$	DAL<07:00>

Write ($\overline{WR}$)—This signal specifies the direction of data transfer on lines DAL<31:00> for the current bus cycle. When asserted, the current bus master transfers data on the DAL<31:00> lines during the data transfer portion of the cycle. When $\overline{WR}$ is not asserted, an external device supplies the data during the data transfer portion of the cycle or lines DAL<31:00> do not contain data. The $\overline{WR}$ signal may be used by external logic to control the direction of DAL bus transceivers. The $\overline{WR}$ input is valid on the falling edge of the $\overline{AS}$ signal.

Data Buffer Enable ($\overline{DBE}$)—This signal may be used with the $\overline{WR}$ signal by external logic to control the DAL<31:00> buffers and transceivers. When the MicroDMA is bus master, it asserts the $\overline{DBE}$ signal to enable the buffers or transceivers and deasserts $\overline{DBE}$ to disable them.

Ready ($\overline{RDY}$)—This signal is asserted by external logic to indicate that it will complete the current bus cycle. When not asserted, it extends the current bus cycle for a slower memory or peripheral device. The $\overline{RDY}$ or $\overline{ERR}$ signal must be asserted to end the current bus cycle.

Bus Error ($\overline{ERR}$)—This signal is asserted by the external logic to indicate that an error associated with the current bus cycle, such as bus timeout or parity error, has occurred and to end the current bus cycle. The $\overline{ERR}$ or $\overline{RDY}$ signal must be asserted to end the current bus cycle. If the MicroDMA is bus master and detects the assertion of the $\overline{ERR}$ signal, it ends the bus cycle, interrupts the MicroVAX CPU (if enabled), and records the error. The MicroDMA asserts the $\overline{ERR}$ signal if an I/O bus error ($\overline{IERR}$) signal is detected during an access transfer.

Control Status (CS<2:0>)—The MicroDMA uses these lines together with the $\overline{WR}$ input to recognize and respond to the type of bus cycle currently in progress. Table 3 lists the bus cycle selections.

Table 3 • MicroVAX 78532 Bus Cycle Selection

Write $\overline{WR}$	CS line 2	1	0	Bus Cycle Type
1	0	0	0	reserved
1	0	0	1	reserved
1	0	1	0	reserved
1	0	1	1	interrupt acknowledge
1	1	0	0	read instruction
1	1	0	1	read lock*
1	1	1	0	read data, modify intent
1	1	1	1	read data, no modify intent*
0	0	0	0	reserved
0	0	0	1	reserved
0	0	1	0	reserved
0	0	1	1	reserved
0	1	0	0	reserved
0	1	0	1	write unlock*
0	1	1	0	reserved
0	1	1	1	write*

*Used by MicroDMA as bus master.

MicroVAX Bus Interrupt Control

Interrupt Request ($\overline{IRQ}$<3:0>)—These lines are used by the MicroDMA to interrupt the MicroVAX CPU. The vectors associated with these interrupts are provided to the MicroDMA by MicroVAX system software. The interrupt request assignments are listed in Table 4.

Table 4 • MicroVAX 78532 Interrupt Request Assignments

IRQ Line	Interrupt Level
$\overline{IRQ3}$	IPL 17
$\overline{IRQ2}$	IPL 16
$\overline{IRQ1}$	IPL 15
$\overline{IRQ0}$	IPL 14

IACK Enable Input ($\overline{\text{IAKEI}}$)—This signal is asserted to allow the MicroDMA to respond to a MicroVAX interrupt acknowledge. The interrupt acknowledge lines of the I/O devices on the MicroVAX bus are connected in daisychain. CS2 is connected to the $\overline{\text{IAKEI}}$ line of the I/O device of the highest priority. If this device does not have an interrupt pending at the correct priority level, it asserts its $\overline{\text{IAKEO}}$ signal, which is connected to the $\overline{\text{IAKEI}}$ line of the next device in the daisychain. A device cannot respond to an interrupt acknowledge until its $\overline{\text{IAKEI}}$ line is asserted.

IACK Enable Output ($\overline{\text{IAKEO}}$)—This line is asserted to enable lower priority devices in an interrupt daisychain to respond to the current interrupt acknowledge. The MicroDMA asserts the $\overline{\text{IAKEO}}$ signal for the the next I/O device in the daisychain if the $\overline{\text{IAKEI}}$ signal has been asserted and if the MicroDMA does not have an interrupt pending at the priority level being acknowledged.

MicroVAX Bus DMA Control

DMA Request ($\overline{\text{DMR}}$)—This signal is asserted by the MicroDMA controller to initiate a DMA transfer over the MicroVAX bus. On the cycle following the assertion of the $\overline{\text{DMR}}$ signal, the MicroVAX CPU disables the bus and asserts the $\overline{\text{DMG}}$ signal. This allows the MicroDMA to take control of the bus. After the MicroDMA has completed the DMA transfer, it deasserts the $\overline{\text{DMR}}$ signal to return control of the bus to the MicroVAX CPU. The $\overline{\text{DMR}}$ signal can be asserted by the MicroDMA only if the DMA enable bit of the global control register has been set.

DMA Grant Input ($\overline{\text{DMGI}}$)—When asserted, the MicroDMA controller takes control of the MicroVAX bus for a DMA transfer. This line is connected in a daisychain if more than one DMA device is in a system. The MicroVAX $\overline{\text{DMG}}$ output is tied to the $\overline{\text{DMGI}}$ line of the highest priority DMA device. If this MicroDMA controller does not have a DMA request pending, it asserts its $\overline{\text{DMGO}}$ line, which connects to the $\overline{\text{DMGI}}$ line of the next DMA device. A MicroDMA controller cannot perform a DMA transfer on the MicroVAX bus until its $\overline{\text{DMGI}}$ line is asserted.

DMA Grant Output ($\overline{\text{DMGO}}$)—This signal is used in systems with more than one DMA device and is asserted when a DMA request is to be granted to a lower priority DMA device in the daisychain.

Clock Signal

Clock in (CLKI)—A TTL clock signal used for timing bus operations on both the MicroVAX bus and I/O bus. The CLKI signal is the same as used by the MicroVAX CPU and MicroVAX FPU.

I/O Bus Interface Signals

The I/O bus interface lines connect to the I/O bus devices and consist of data and address lines and control lines. The bus signals are summarized in Table 5. A more detailed description of the signal functions is contained in the following paragraphs.

Table 5 • MicroVAX 78532 I/O Bus Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
67-82 85-101	IDAL<31:00>	input/output	Data and address lines <31:00>—Time multiplexed lines used to transfer data and address information between the MicroDMA and devices on the I/O bus.
110-113	$\overline{\text{ITR}}$<3:0>	input	I/O transfer request <3:0>—Asserted to indicate that a device on the I/O bus requires service. Lines ITR<3:0> corresponds to channel 3 through 0, respectively.
106-109	$\overline{\text{IIR}}$<3:0>	input	I/O interrupt request—Interrupt request lines for I/O bus devices. Lines IIR <3:0> corresponds to channel 3 through 0, respectively. Typically used to terminate a DMA transfer.
125	$\overline{\text{IAS}}$	input/output	I/O address strobe—Asserted to indicate that the IDAL<31:00> lines contain valid data.
122	$\overline{\text{IDS}}$	input/output	I/O data strobe—Asserted during a read cycle to indicate that the IDAL<31:00> lines are available to receive data and deasserted to indicate that the data has been received. Asserted during a write cycle to indicate that data is present on the IDAL<31:00> lines and deasserted to indicate that data is to be removed.
124	$\overline{\text{IDBE}}$	input/output	I/O data buffer enable—Asserted to enable the IDAL<31:00> transceivers.
121-118	$\overline{\text{IBM}}$<3:0>	input/output	I/O byte mask—Specifies the bytes on the IDAL<31:00> lines that contain valid data.
123	$\overline{\text{IWR}}$	input/output	I/O write—Specifies the direction of data transfer on the IDAL<31:00> lines. Asserted to indicate that the current bus master will be the source of the data. Can be used to control the direction of the IDAL<31:00> lines.
131	$\overline{\text{IRDY}}$	input/output	I/O ready—Used to synchronize data transfers between devices operating at different transfer rates on the I/O bus. The current bus master must wait for the assertion of this line before terminating the cycle and removing the data.
130	$\overline{\text{IERR}}$	input/output	I/O error—Asserted to indicate an I/O bus error condition. In the window mode, it also may indicate a MicroVAX bus error.
129	$\overline{\text{IDMR}}$	input/output	I/O DMA request—Asserted to indicate that a device on the I/O bus is requesting mastership of the I/O bus. The transfer could be I/O DMA, DMA, or I/O access.

Pin	Signal	Input/Output	Definition/Function
126	$\overline{IDMG}$	input/output	I/O DMA grant—Asserted to indicate that the I/O bus has been released by the current I/O bus master to allow a DMA transfer to occur.
21	$\overline{VBR}$	input	MicroVAX bus request—Causes the MicroDMA to request the use of the MicroVAX bus. Typically used when performing a window transfer.
11	$\overline{VBG}$	output	MicroVAX bus grant—Asserted to inform the I/O processor that the MicroVAX bus request has been granted.
114	$\overline{IREG}$	input	I/O bus register—Asserted by the I/O processor to access a MicroDMA register.
127	$\overline{IOPINT}$	output	I/O processor interrupt—Asserted when any DMA channel initiates an interrupt to an I/O processor.
28	$\overline{ILOCK}$	input	MicroVAX bus lock—Asserted by an I/O processor to cause a locked MicroVAX bus cycle during a window transfer.
115	$\overline{IMASTER}$	input	I/O master—Asserted to enable the MicroDMA to become the master of the I/O bus by default.
105	$\overline{RESET}$	input	Reset—Asserted to set the MicroDMA controller to a predefined state.
1,132	Test<1:0>	input	Test—Reserved for testing the MicroDMA controller during manufacture.
128	ICLKO	output	I/O clock output—A clock pulse output at one-fourth of the CLKI frequency.
32,101 18,50, 84,116	V_{DD}	input	Voltage—Power supply voltage.
31,102 17,51 83,117 30	V_{SS}	input	Ground—Ground reference.

I/O Data and Address

I/O Data and Address Bus (IDAL<31:00>)—These lines are time-multiplexed and bidirectional and are used to transfer data and address information between the MicroDMA controller and devices or controllers on the I/O bus. The lines can be programmed for 8-, 16-, or 32-bit data widths. The strobe signals $\overline{IAS}$ and $\overline{IDS}$ determine whether the bus carries data or address information.

I/O Address Strobe ($\overline{IAS}$)—If the MicroDMA is bus master, it uses the $\overline{IAS}$ signal to indicate that the IDAL<31:00> lines contain valid address information and that the information on $\overline{IBM}$<3:0> and $\overline{IWR}$ is also valid. The $\overline{IAS}$ line is used by external circuits to latch this address and to qualify the control signals.

If the MicroDMA is not bus master, the falling edge of the $\overline{IAS}$ signal causes the MicroDMA to latch the address and control information to determine whether a window transfer is indicated. If a window transfer is not specified, the MicroDMA does not participate in the cycle. If it is specified, the MicroDMA participates in the cycle by performing mapping (if required) and by asserting the $\overline{IRDY}$ line when the transfer is complete.

I/O Data Strobe ($\overline{IDS}$)—This signal provides timing information for the data transfer portion of an I/O bus read or write cycle. During a read cycle, the falling edge of the $\overline{IDS}$ signal indicates that the MicroDMA is ready to receive data and the rising edge indicates that the data has been latched by the MicroDMA and can be removed. During a write cycle, the falling edge of the $\overline{IDS}$ signal indicates that the MicroDMA has placed valid data on the IDAL lines and the rising edge indicates that the data is about to be removed.

I/O Byte Masks ($\overline{IBM}$<3:0>)—These signals indicate which bytes of the IDAL<31:00> lines are valid during the current data transfer. During a write cycle, the $\overline{IBM}$<3:0> lines specify the bytes that contain valid data for writing. During a read cycle, the $\overline{IBM}$<3:0> lines specify the DAL<31:00> lines that must be supplied with valid data by an external device. The information on the $\overline{IBM}$<3:0< lines is valid on the falling edge of the $\overline{IAS}$ signal.

The validity of the bytes also depends on the width of the current data transfer. If the current transfer is 1 byte wide, the information on the $\overline{IBM}$<3:0> lines is not significant because a byte is transferred only on the IDAL<07:00> lines. If the current transfer is 2 bytes wide, the $\overline{IBM}$<3:0> lines specify whether lines IDAL<15:08> and/or IDAL<07:00> contain valid information. If the current transfer is 4 bytes wide, the valid bytes are specified by $\overline{IBM}$<3:0>. Table 6 lists the I/O bus byte mask assignments.

Table 6 • MicroVAX 78532 I/O Bus Byte Mask Assignments

I/O Byte Mask Line	Valid Data
$\overline{IBM3}$	IDAL<31:24>
$\overline{IBM2}$	IDAL<23:16>
$\overline{IBM1}$	IDAL<15:08>
$\overline{IBM0}$	IDAL<07:00>

I/O Write ($\overline{IWR}$)—This signal specifies the direction of data transfer on the IDAL<31:00> lines for the current bus cycle. When the $\overline{IWR}$ signal is asserted, the current bus master drives the lines during the data transfer portion of the cycle. When the $\overline{IWR}$ signal is not asserted, an external device supplies the data during the data transfer portion of the cycle or the lines are idle. This signal may be used by external logic to control the direction of the IDAL<31:00> transceivers. The $\overline{IWR}$ information is valid on the falling edge of the $\overline{IAS}$ signal.

I/O Data Buffer Enable ($\overline{IDBE}$)—This signal may be used with the $\overline{IWR}$ signal to control external IDAL<31:00> transceivers and buffers. When asserted, it enables a transceiver or buffer and when not asserted, the outputs of the transceiver or buffer are disabled.

I/O Ready ($\overline{IRDY}$)—During I/O bus cycle data transfer, the bus master must wait for the assertion of this signal before terminating the current cycle and latching (or removing) data from the bus.

I/O Bus Error ($\overline{IERR}$)—This signal indicates a hardware bus error condition to the current bus master. If the MicroDMA detects the assertion of this signal during a DMA transfer, it ends the current bus cycle, and interrupts the MicroVAX CPU (if enabled). If the MicroDMA detects the $\overline{ERR}$ signal during a window transfer, it asserts $\overline{IERR}$ so that the bus error is apparent to the I/O processor.

I/O Bus Data Transfer Control

Transfer Request from I/O Device ($\overline{ITR<3:0>}$)—This signal is asserted by an I/O bus device that requires DMA service or by an address decoder that has detected a window reference. One line is assigned to each channel. Simultaneous requests are processed according to the channel number with $\overline{ITR0}$ at highest priority and $\overline{ITR3}$ at lowest. The MicroDMA responds to the assertion of a line by performing a DMA transfer if the channel is in DMA mode or a window transfer if the channel is in window mode. The MicroDMA does not acknowledge these signals, and transfer is performed automatically. The signals involved in the transfer must be decoded if a peripheral device requires an acknowledgment signal.

Interrupt Request from I/O Device ($\overline{IIR<3:0>}$)—These signals are asserted by a device on the I/O bus to interrupt the MicroDMA. One line is assigned to each channel. If a channel is enabled when the corresponding $\overline{IIR}$ line is asserted, the current transfer is immediately terminated and the channel is disabled except for the interrupt logic. Once a transfer has been terminated, the interrupt condition that caused the line to be asserted must be cleared before the channel can be enabled again.

I/O Bus DMA Control

I/O DMA Request ($\overline{IDMR}$)—This signal is asserted by a potential bus master to request I/O bus ownership. When the MicroDMA is the default bus master (IMASTER input is asserted), the $\overline{IDMR}$ is an input signal that is typically asserted by an I/O processor or DMA device on the I/O bus. When the MicroDMA is not the default bus master, it asserts the $\overline{IDMR}$ line to use the bus for a local memory transfer or a transfer to a peripheral register.

I/O DMA Grant ($\overline{IDMG}$)—This signal is asserted by the current bus master in response to the $\overline{IDMR}$ signal. It indicates that the bus master has released the I/O bus and that bus ownership may now be assumed by another device.

I/O Processor Control

MicroVAX Bus Request ($\overline{VBR}$)—This signal is asserted by an I/O processor to control the MicroVAX bus when performing a window transfer. It is used to avoid a conflict when the MicroVAX CPU accesses the I/O bus at the same time an I/O processor is to perform a window transfer. When asserted, the MicroDMA requests control of the MicroVAX bus, and asserts the $\overline{VBG}$ signal when control of the MicroVAX bus is granted. The I/O processor should deassert the $\overline{VBR}$ signal as soon as window transfer is complete in order to preserve the system throughput.

MicroVAX Bus Grant ($\overline{VBG}$)—This signal is asserted by the MicroDMA in response to the assertion of the $\overline{VBR}$ signal. It informs an I/O processor that control of the MicroVAX bus has been acquired and that the I/O processor may proceed with a window transfer.

I/O Bus Register Access ($\overline{\text{IREG}}$)—This signal is asserted by an I/O processor to access a MicroDMA internal register. It is asserted during the address portion of an I/O bus cycle to indicate that the information on the IDAL<08:00> lines should be interpreted as a register address. It should be deasserted at the end of the cycle.

I/O Processor Interrupt ($\overline{\text{IOPINT}}$)—This signal is asserted when any of the MicroDMA channels initiates an interrupt to an I/O processor. Because more than one channel may interrupt at the same time, the I/O processor must poll all the channels to determine the highest priority interrupt. The I/O processor software deasserts this signal by setting the ENABLE bit or clearing the DONE bit of the channel control register of the interrupting channel or by redirecting the interrupt to the MicroVAX CPU.

MicroVAX Bus Lock ($\overline{\text{ILOCK}}$)—This signal is asserted by an I/O processor to create locked MicroVAX bus cycles during a window reference. During window read and write operations, asserting this signal causes MicroVAX memory read and write operations to be performed with a code of 101 on lines CS<2:0> (read lock or write unlock). The MicroDMA continues to assert the $\overline{\text{DMR}}$ line while the $\overline{\text{ILOCK}}$ signal is asserted, but does not assert the read lock code while accessing the page table information required by the window access. It does not check the sequence of reads and writes.

Miscellaneous Signals

I/O Bus Master (IMASTER)—When asserted, the MicroDMA becomes bus master of the I/O bus by default and it responds to the assertion of the $\overline{\text{IDMR}}$ signal by asserting the $\overline{\text{IDMG}}$ line and by releasing the I/O bus.

Chip Select ($\overline{\text{CSL}}$)—This signal is used by the external logic to allow the MicroVAX CPU to access the I/O bus or a MicroDMA internal register. When asserted, the information on the DAL<23:00> lines is interpreted as an I/O bus physical address if lines DAL<23:09> are not all zeros or the information on lines DAL<08:00> are interpreted as an internal register address if the DAL<23:09> lines are all zeros.

Reset ($\overline{\text{RESET}}$)—When asserted, this signal sets the MicroDMA to a specified initial state.

Test (TEST <1:0>)—Reserved for manufacturing use. If Test0 is asserted, all outputs will be forced to a high-impedance state. Test <1:0> contains an internal pull-down circuit.

I/O Bus Clock Output (ICLKO)—A clock output at one-fourth of the CLKI frequency.

Power and Ground Connections

Voltage (V_{DD})—5-Vdc power supply.

Ground (V_{SS})—Ground reference.

MicroDMA Controller Operation

The MicroDMA controller is a multipurpose interface that can be used between the MicroVAX processor and a compatible 32-bit I/O bus for peripheral devices or controllers. The MicroDMA has four independently programmable channels through which DMA, window, and I/O bus data transfers can be performed. The four channels are assigned a fixed priority with channel 0 having the highest priority. Devices that transfer data at the highest rates or memory transfers should be assigned the lowest priority channels to allow the slower devices to access the bus. This section briefly describes the characteristics of these transfers and the other major functions of the MicroDMA.

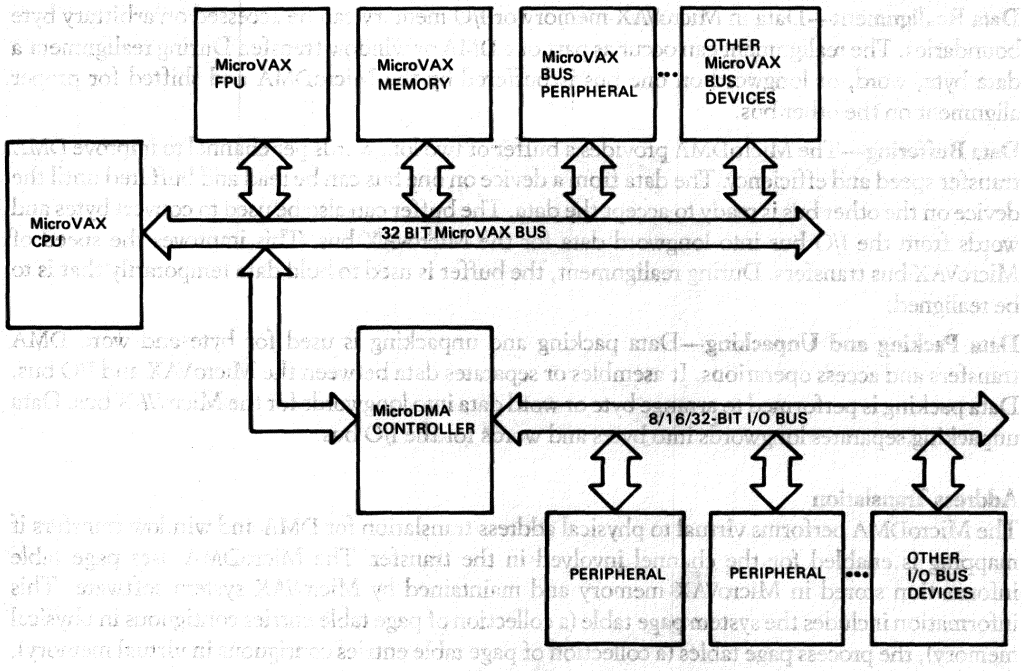


Figure 3 • MicroVAX 78532 MicroDMA System Interfacing

Types of DMA Transfers

The primary function of the MicroDMA is to perform block-mode DMA transfers. Large amounts of data can be transferred between a device or intelligent I/O subsystem on the I/O bus to a device or memory on MicroVAX bus without CPU intervention. DMA transfers may also involve address translation, data realignment, and data buffering.

Window Transfers—Intelligent devices on the I/O bus can access "windows" in the MicroVAX memory. Window transfers allow the devices to access buffers and to operate from work queues in memory. The location and size of a window are defined by the user application. Window transfers may involve address translation and data realignment.

I/O DMA Transfers—The MicroDMA controller can perform local I/O bus DMA transfers independently of other activities that may occur on the MicroVAX bus. The I/O DMA capability allows the I/O bus memory to be used as a large buffer for data rate smoothing. The I/O bus memory can be filled by a channel in I/O DMA mode while another channel in DMA mode transfers the data to MicroVAX memory. I/O DMA transfers do not involve address translation or data realignment.

Access Operations

The MicroDMA controller allows the MicroVAX CPU to access memory and devices on the I/O bus similarly to devices directly connected to the MicroVAX bus. These access operations are performed by the user application and define a region of MicroVAX physical address space as an I/O bus access range. When a range is referenced, an access to one or more equivalent locations on the I/O bus is performed. Access operations may occur in parallel with channel operations such as DMA, window and I/O DMA transfers. During access operations, data packing and unpacking is performed and address translation or realignment is not performed.

Data Realignment—Data in MicroVAX memory or I/O memory can be accessed on arbitrary byte boundaries. The realignment can occur as part of a DMA or window transfer. During realignment a data byte, word, or longword on one bus is buffered by the MicroDMA and shifted for proper alignment on the other bus.

Data Buffering—The MicroDMA provides a buffer of two longwords per channel to improve DMA transfer speed and efficiency. The data from a device on one bus can be read and buffered until the device on the other bus is ready to accept the data. The buffer can also be used to convert bytes and words from the I/O bus into longword data for the MicroVAX bus. This improves the speed of MicroVAX bus transfers. During realignment, the buffer is used to hold data temporarily that is to be realigned.

Data Packing and Unpacking—Data packing and unpacking is used for byte and word DMA transfers and access operations. It asembles or separates data between the MicroVAX and I/O bus. Data packing is performed to arrange byte or word data into longwords for the MicroVAX bus. Data unpacking separates longwords into bytes and words for the I/O bus.

Address Translation

The MicroDMA performs virtual to physical address translation for DMA and window transfers if mapping is enabled for the channel involved in the transfer. The MicroDMA uses page table information stored in MicroVAX memory and maintained by MicroVAX system software. This information includes the system page table (a collection of page table entries contiguous in physical memory), the process page tables (a collection of page table entries contiguous in virtual memory), and the global page table used to describe shared pages.

Bus Interfacing

The MicroDMA controller prevents different activities between the buses from interfering with each other. The controller appears as a DMA peripheral device on the MicroVAX bus. It requests and relinquishes the bus through the DMA request and grant logic. The bus interface uses signals with the same timing characteristics as those used for any other MicroVAX bus device such as MicroVAX memory.

The controller appears as an I/O processor or CPU on the I/O bus. The I/O bus supports the simultaneous use of 8-, 16-, and 32-bit devices and has an interfacing protocol similar to that of the MicroVAX bus.

The MicroDMA may operate as a master or a slave on either bus. Therefore, the bus control signal $\overline{AS}$ (address strobe), for example, can be used as an input or output. Table 7 summarizes the MicroDMA operations. It specifies operations for which the MicroDMA is bus master and shows the source and destination buses for each. MicroDMA internal registers can be accessed from either the MicroVAX bus or I/O bus.

Table 7 • MicroVAX 78532 Operation Summary

Operation	Bus Master	Source Bus	Destination Bus	Address Translation	Data Alignment	Data Packing
DMA Transfer	MicroDMA	MicroVAX/IO	IO/MicroVAX	Yes	Yes	Yes
Window Transfer	IOP*	IO/MicroVAX	MicroVAX/IO	Yes	Yes	No
I/O DMA Transfer	MicroDMA	IO	IO	No	No	No
Access Operation	MicroVAX	MicroVAX/IO	IO/MicroVAX	No	No	Yes

*IOP = Intelligent I/O device or I/O processor.

Registers

The MicroDMA controller contains 63 byte addressable, user-accessible registers. The MicroDMA controller contains one set of global registers that defines the overall state of the MicroDMA controller and four sets of channel registers that define the state of each channel. The registers may be accessed from either the MicroVAX bus or the I/O bus.

The registers occupy a 512-byte region in MicroVAX I/O address space. The base address of this region is defined by the user and must begin on a 16-Mbyte boundary. The user's application decodes lines DAL<31:00> and asserts the $\overline{CSL}$ line when an address in the register region is referenced. When $\overline{CSL}$ is asserted, the information on the DAL<08:00> lines is interpreted as a register address if the DAL<23:09> lines are all zeros. If DAL<23:09> are not all zeros, the information on DAL<23:00> is interpreted as an I/O bus address and an I/O access cycle will be performed. (Refer to Access Operations.) The addresses associated with the global registers are listed in Table 8. Table 9 lists the addresses assigned to the channel registers.

Table 8 • MicroVAX 78532 Global Registers Address Assignments

Address* (hexadecimal)	Mnemonic	Read/Write (R/W)	Description
000	DGCTL	RW	Global Control Register
004	DSBR	RW	System Base Register
008	DGBR	RW	Global Base Register
00C-03C			Reserved

*Register addresses must appear on the DAL<08:00> or IDAL<08:00> lines. References to reserved addresses will cause unpredictable results. Some registers have more than one function depending on the current operational mode of the MicroDMA controller.

Table 9 • MicroVAX 78532 Channel Registers Address Assignments

Address* Ch0	Ch1	Ch2	Ch3	Mnemonic†	Read/Write R/W	Register‡
040	080	0C0	100	DCCTLx	RW	DMA Channel Control
044	084	0C4	104			Reserved
048	088	0C8	108	DCINTx	RW	DMA Channel Interrupt Vector
04C	08C	0CC	10C	DCIOBAx	RW	DMA I/O Base Address
04C	08C	0CC	10C	DCIDSx	RW	DMA I/O Source Address
050	090	0D0	110	DCIBCx	RW	DMA Initial Byte Count
050	090	0D0	110	DCWMx	RW	Window Mask (window)
054	094	0D4	114	DCBOx	RW	Byte Offset (mapping on)
054	094	0D4	114	DCUPAx	RW	MicroVAX Physical Address (no map)
058	098	0D8	118	DCSPTEx	RW	SVAPTE Register (DMA, window)
058	098	0D8	118	DCIDDx	RW	I/O DMA Destination Address
060	0A0	0E0	120	DCCSVx	R	Current System Virtual Address of PTE
064	0A4	0E4	124	DCIOAx	R	Current I/O Bus Address
068	0A8	0E8	128	DCBCx	R	Current Byte Count
06C	0AC	0EC	12C	DCPTEx	R	Current Page Table Entry
070	0B0	0F0	130	DCPAx	R	Current Physical Address
074	0B4	0F4	134			Reserved
078	0B8	0F8	138			Reserved
07C	0BC	0FC	13C			Reserved

*Hexadecimal notation. Register addresses must appear on the DAL<08:00> or IDAL<08:00> lines. References to reserved addresses will cause unpredictable results.
†x = Register designations 0, 1, 2, or 3, depending on channel number.
‡Some registers have more than one function depending on the current operational mode of the MicroDMA controller.

To access a register from the I/O bus, the user's application decodes IDAL <23:00> and asserts the $\overline{\text{IREG}}$ signal to indicate a register access. The MicroDMA interprets the IDAL<08:00> information as a register address. The I/O bus write access to the registers is controlled by bit 10 in the DGCTL register, the value of which is usually determined by MicroVAX system software.

DMA Global Control Register—The DMA global control register (DGCTL) is used to control, configure, and determine the global status for the MicroDMA controller. The format of the register information is shown in Figure 4 and defined in Table 10.

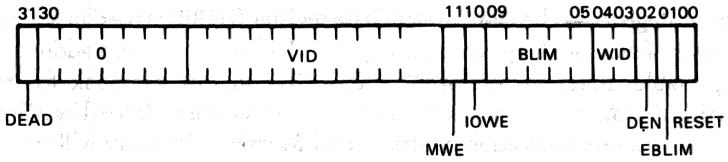


Figure 4 • MicroVAX 78532 DMA Global Control Register Format

Table 10 • MicroVAX 78532 Global Control Register Description

Bit	Description
31	DEAD (Deadlock)—A read-only bit that is set to indicate that a deadlock situation existed in a previous I/O bus access operation. Cleared during a reset operation or by writing a one to this bit.
30:24	Not used (read as zeros).
23:12	VID (Version identification)—A read-only field that contains the version number of the MicroDMA chip. For the initial version of this chip, the number is 00000001.
11	MWE (Maintenance write enable)—Setting this read/write bit enables writing to any register including the read-only registers. It is intended for diagnostic and manufacturing test usage only. The status/error bits cannot be set when this bit is set. Cleared during a reset operation.
10	IOWE (I/O bus register write enable)—A read/write bit set to enable an I/O device to write data to a channel register. When cleared, a write to external registers will be ignored. Cleared during a reset operation.
9:5	BLIM (Burst limit)—Specifies the maximum length (in bus cycles) of a DMA burst on the MicroVAX bus when EBLIM (bit 1) is set. Cleared during a reset operation.
4:3	WID (Width)—A read/write field that specifies the data width of I/O bus access operations as follows. These bits are cleared during a reset operation. **WID Bits 04 03 — Data Width** 0 0 — 1 byte 0 1 — 1 byte 1 0 — 2 bytes 1 1 — 4 bytes
2	DEN (DMA enable)—A read/write bit that must be set to allow the MicroDMA controller to perform DMA transfers on the MicroVAX bus. Clearing this bit prevents the MicroDMA from asserting $\overline{\text{DMR}}$. Cleared during a reset operation.
1	EBLIM (Enable burst limit)—A read/write bit. When set, BLIM (bits 9:5) define the maximum length of a DMA burst. When clear, the DMA burst length is unlimited. Cleared during a reset operation.
0	RESET—A read/write bit that is set to initiate a reset operation that forces the MicroDMA to a known initial state. Setting this bit has the same effect as asserting the $\overline{\text{RESET}}$ input.

DMA System Base Register—The DMA system base register (DSBR) is used in address translation and contains a copy of MicroVAX system base register which is the physical address of the base of the system page table. Refer to MicroVAX 78032 CPU in this databook for more detailed information on the MicroVAX system base register. The DSBR must be loaded by MicroVAX system software before any address translation occurs so that MicroVAX memory will not be corrupted. The format of the register information is shown in Figure 5 and defined in Table 11.

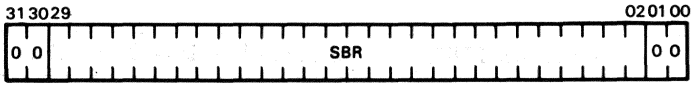


Figure 5 • MicroVAX 78532 DMA System Base Register Format

Table 11 • MicroVAX 78532 DMA System Base Register Description

Bits	Description
31:30	Not used (read as zeros).
29:02	SBR (System base register)—Contains the physical longword address of the system page table. The same as bits 29:02 of the MicroVAX system base register.
01:00	Not used (read as zeros).

DMA Global Base Register—The DMA global base register (DGBR) contains a copy of the information in the MicroVAX global base register. It is used by the MicroDMA during virtual-to-physical address translation to locate the global page table that describes the shared pages in system virtual memory. The register must be loaded by MicroVAX system software before any virtual DMA activity so that MicroVAX memory will not be corrupted. The format of the register information is shown in Figure 6 and defined in Table 12.

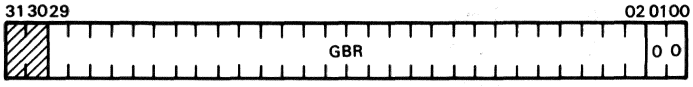


Figure 6 • MicroVAX 78532 DMA Global Base Register Format

Table 12 • MicroVAX 78532 DMA Global Base Register Description

Bits	Description
31:30	Not used (read/write).
29:02	GBR (Global base register)—Used to locate the global page table in the system virtual memory. The same as bits 29:02 on the MicroVAX global base register
01:00	Not used (read as zeros).

DMA Channel Control Registers (0-3)—The four channel control (DCCTL0 through DCCTL3) registers, one for each channel, are used to control, configure, and determine status for the four channels. Three translation error bits are used for system, process, and global translation errors.

Translation errors can occur when a bus error is detected while fetching a page table entry, by an invalid page table, and by a global page table entry that leads to another global page table entry.

The cause of errors is indicated only by the error bits that define the location in the translation process in which the error occurred. For example, if the error occurred during the global part of translation, then the global error bit will be set. The format of the register information is shown in Figure 7 and defined in Table 13.

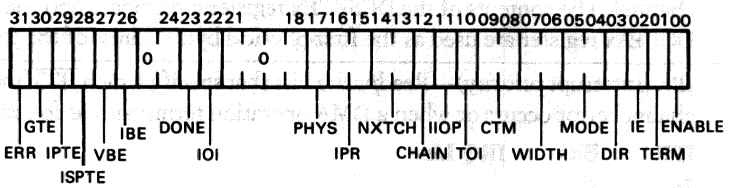


Figure 7 • MicroVAX 78532 DMA Channel Control Registers (0-3) Format

Table 13 • MicroVAX 78532 DMA Channel Control Registers (0-3) Description

Bits	Description
31	ERR (Error)—A read-only bit set to indicate that an error has occurred in a channel operation when any of bits 30:26 are set. Cleared during a reset operation, by writing a 1 to this location, or by setting the ENABLE (bit 00).
30	GTE (Global translation error)—A read-only bit set to indicate that an error occurred during an address translation involving global page tables. The ISPTE and IPTE bits indicate where in the process the error occurred Cleared during a reset operation by writing a 1 to this location, or when the ENABLE (bit 00) is set.
29	IPTE (Invalid process page table entry)—A read-only bit set to indicate that an invalid process page table entry has been fetched. Cleared during a reset operation, by writing a 1 to this location, or by setting the ENABLE (bit 00).
28	ISPTE (Invalid system page table entry bit)—A read-only bit set to indicate that an invalid system page table entry has been fetched. Cleared during a reset operation, by writing a 1 to this location, or by setting the ENABLE (bit 00).
27	VBE (MicroVAX bus error)—A read-only bit set to indicate that a bus error occurred on the MicroVAX bus during a DMA transfer on this channel. Cleared during a reset operation, by writing a 1 to this location, or by setting the ENABLE (bit 00).
26	IBE (I/O bus error)—A read-only bit set to indicate that a bus error occurred on the I/O bus during a DMA transfer on this channel. Cleared during a reset operation, by writing a 1 to this location, or by setting the ENABLE (bit 00).

Bits	Description
25:24	Not used (read as zeros).
23	DONE—A read-only bit set to indicate that the current channel operation has terminated. Cleared during a reset operation when written to a 1 or when ENABLE (bit 00) is set.
22	IOI (I/O interrupt)—A read/write bit set to indicate that the $\overline{IIR}$ line for this channel has been asserted. If the TOI (bit 10) is also set, the current transfer will terminate. Cleared during a reset operation or when ENABLE (bit 00) is set.
21:18	Not used (read as zeros).
17	PHYS (Physical)—A read/write bit set to disable the address translation for this channel. The contents of the DCSPTEx register are ignored and the contents of the DCUPAx register are used as the first physical byte address of the transfer.
16:15	IPR (Interrupt priority)—Read/write bits that specify which $\overline{IRQ}$ line is asserted if a channel error occurs or when a DMA operation terminates as follows:

IPR 16	Bits 15	IRQ Line
0	0	$\overline{IRQ0}$
0	1	$\overline{IRQ1}$
1	0	$\overline{IRQ2}$
1	1	$\overline{IRQ3}$

14:13	NXTCH (Next channel)—Read/write bits that indicate the number of the next channel to be enabled in a chaining operation when CHAIN (bit 12) is set.
12	CHAIN—A read/write bit set to start a transfer on the channel specified by NXTCH channel number when the current transfer terminates without error.
11	IIOP (Interrupt I/O processor)—A read/write bit when set and IE (bit 02) is set, it causes the channel interrupts to be directed to an I/O bus processor when the $\overline{IOPINT}$ signal is asserted. When cleared and IE (bit 02) is set, the channel interrupts are directed to the MicroVAX CPU as determined by the IPR line.
10	TOI (Termination on interrupt)—A read/write bit set to terminate DMA transfers on this channel upon the assertion of the $\overline{IIR}$signal for this channel.
09:08	CTM (Count mode)—Read/write bits that determine whether addresses on the MicroVAX bus and/or I/O bus will remain the same or be incremented for data transfers on this channel. MicroVAX bus addresses are incremented by 4. The I/O bus addresses are incremented according to the data width of the I/O device involved in the transfer.

CTM 09	Bits 08	MicroVAX Bus Address	I/O Bus Address
0	0	same	same
0	1	same	incremented
1	0	incremented	same
1	1	incremented	incremented

Bits	Description
07:06	WIDTH—Read/write bits that specify the data width of the I/O device associated with the channel as follows: **WIDTH Bits 07 06** — **Data Width** 0 0 — 1 byte 0 1 — 1 byte 1 0 — 2 bytes 1 1 — 4 bytes
05:04	MODE (Mode field)—Read/write bits that specify the operational mode of the channel. **MODE Bits 05 04** — **Operational Mode** 0 0 — illegal 0 1 — I/O DMA 1 0 — DMA 1 1 — window
03	DIR (Direction)—A read/write bit that specifies the direction of a data transfer in DMA mode. Set to specify a transfer from the I/O bus to the MicroVAX bus. Cleared to specify a transfer from the MicroVAX bus to the I/O bus.
02	IE (Interrupt enable)—A read/write bit set to enable the interrupts to devices on the MicroVAX bus and I/O bus. Must be set before or at the same time ENABLE (bit 00) is set. Cleared during a reset operation to immediately disable all channel interrupts.
01	TERM (Terminate)—A read/write bit, set to force the termination of the current channel operation, but allows buffered data to be written. Cleared during a reset operation or by setting ENABLE (bit 00).
00	ENABLE (Enable)—A read/write bit set to immediately clear bits 31:27 and 23:22 of this register and to configure the channel according to bits 17:03. Cleared during a reset operation to immediately abort the current channel operation. Any buffered data is lost.

DMA Channel Interrupt Registers (0-3)—The channel interrupt vector (DCINT0 through DCINT3) registers contain the vector value used by MicroVAX CPU to process interrupts related to the operation of the channel. The priority of the interrupt is specified by the IPR field of the register. The format of the register information is shown in Figure 8.

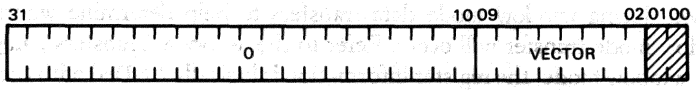


Figure 8 • MicroVAX 78532 DMA Channel Interrupt Registers (0-3) Format

DMA I/O Base Address Registers (0-3)—The DMA I/O base address (DCIOBA0 through DCIOBA3) registers contain the base address of the device or memory on the I/O bus that will participate in a DMA transfer on this channel. When the DMA transfer is started, the register is copied into the DCIOAx register which may be modified during the transfer. The DCIOBAx register information may be used in subsequent transfers or in chaining operations. The format of the register information is shown in Figure 9.

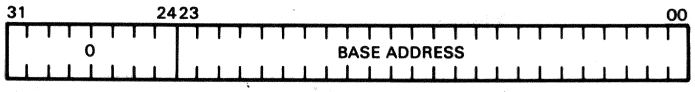


Figure 9 • MicroVAX 78532 DMA I/O Base Address Registers (0-3) Format

DMA I/O Source Address Registers (0-3)—The DMA I/O source address (DCIDS0 through DCIDS3) registers contain a 24-bit physical I/O bus address that specifies the source of an I/O DMA transfer. This address may be associated with a peripheral device or the start of a memory buffer in I/O bus memory. The format of the register information is shown in Figure 10.

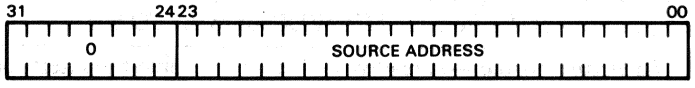


Figure 10 • MicroVAX 78532 DMA I/O Source Address Registers (0-3) Format

DMA Initial Byte Count Registers (0-3)—The initial byte count (DCIBC0 through DCIBC3) registers contain the initial byte count for a DMA or I/O DMA transfer. When the transfer is started, the register information is copied into the DCBCx register where it is decremented as the transfer proceeds. The maximum DMA transfer length is 1 Gbyte and the maximum I/O DMA transfer length is 16 Mbytes. The format of the register information is shown in Figure 11.

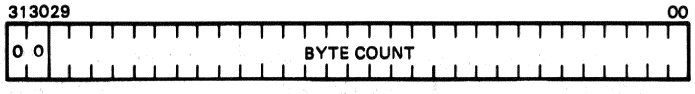


Figure 11 • MicroVAX 78532 DMA Initial Byte Count Registers (0-3) Format

DMA Window Mask Registers (0-3)—The DMA window mask (DCWM0 through DCWM3) registers are used during window mode data transfers to help determine where in MicroVAX memory a window mode transfer will occur. Refer to the Window Transfers paragraph for more information. In window mode, the register information is logically ANDed with the address on the I/O bus to specify an offset within the window. The format of the register information is shown in Figure 12.

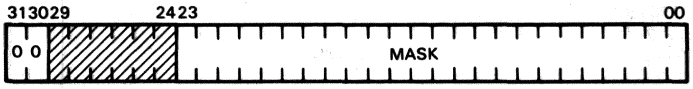


Figure 12 • MicroVAX 78532 DMA Window Mask Registers (0-3) Format

DMA Byte Offset Registers (0-3)—The DMA byte offset (DCBO0 through DCBO3) registers are used with the DMA system virtual PTE (DCSPTE0 through DCSPTE3) registers to determine a memory location where a mapped DMA buffer or MicroVAX window starts. The register information is used to find the physical address of the first (base) page of the transfer. Each register contains an offset (in bytes) that, when added to the base page address, specifies the physical address of the first byte of the buffer or window. Refer to the Address Translation paragraph for the use of these registers. The format of the register information is shown in Figure 13.

Figure 13 • MicroVAX 78532 DMA Byte Offset Registers (0-3) Format

DMA MicroVAX Physical Address Registers (0-3)—The DMA MicroVAX physical address (DCUPA0 through DCUPA3) registers contain the base physical address in MicroVAX memory for unmapped transfers in DMA and window modes. For unmapped DMA mode transfers, this register specifies the physical address in MicroVAX memory at which a DMA transfer will begin. For unmapped window mode transfers, an offset is added to the contents of these registers to determine the starting address of the transfer. The offset is obtained by ANDing the address on the I/O bus with the DCWMx register. The format of the register information is shown in Figure 14.

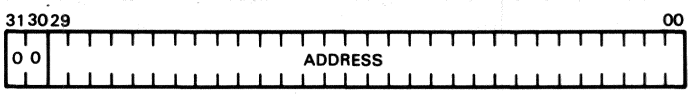


Figure 14 • MicroVAX 78532 DMA Physical Address Registers (0-3) Format

DMA System Virtual Address PTE Registers (0-3)—The system virtual address PTE (DCSPTE0 through DCSPTE3) registers contain the system virtual address of a page table entry. The page table entry points to the base address of the page at which a mapped DMA buffer or window begins. An offset (expressed as a number of bytes) contained in the DCBOx register is added to the base address to specify the address of the first byte in the buffer or window. The format of the register information is shown in Figure 15.

Figure 15 • MicroVAX 78532 DMA System Virtual Address PTE Registers (0-3) Format

DMA I/O Destination Address Registers (0-3)—The DMA I/O destination address (DCIDD0 through DCIDD3) registers contains a 24-bit physical I/O bus address that specifies the destination of an I/O DMA transfer. This address may be associated with a peripheral device or the start of a memory buffer in I/O bus memory. The format of the register information is shown in Figure 16.

Figure 16 • MicroVAX 78532 DMA I/O Destination Address Registers (0-3) Format

DMA Current System Virtual Address PTE Registers (0-3)—The current system virtual address PTE (DCCSV0 through DCCSV3) registers contain the system virtual address of the page table entry currently being accessed. If a translation error occurs (for example, when a page table entry is invalid), the system virtual address of the erroneous page table entry is in this register. These registers are read only. The format of the register information is shown in Figure 17.

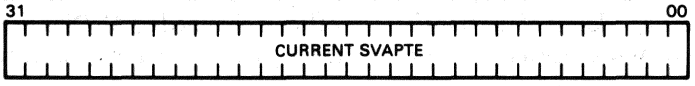


Figure 17 • MicroVAX 78532 DMA Current System Virtual Address PTE Registers (0-3) Format

DMA Current I/O Bus Address Registers (0-3)—If an I/O bus error occurs, the DMA current I/O bus address (DCIOA0 through DCIOA3) registers contain the I/O bus address associated with the error. These registers are read only. The format of the register information is shown in Figure 18.

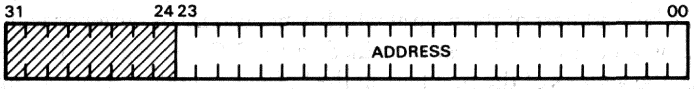


Figure 18 • MicroVAX 78532 DMA Current I/O Bus Address Registers (0-3) Format

DMA Current Byte Count Registers (0-3)—The current byte count (DCBC0 through DCBC3) registers contain the byte count for the transfer currently in progress. It specifies the number of bytes remaining in the transfer. These registers are read only. The format of the register information is shown in Figure 19.

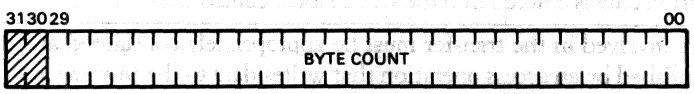


Figure 19 • MicroVAX 78532 DMA Current Byte Count Registers (0-3) Format

DMA Current Page Table Entry Registers (0-3)—The DMA current page table entry (DCPTE0 through DCPTE3) registers contain the page table entry currently being accessed. If a translation error occurs due to an invalid page table entry, the erroneous page table entry can be found in these registers. These registers are read only. The format of the register information is shown in Figure 20.

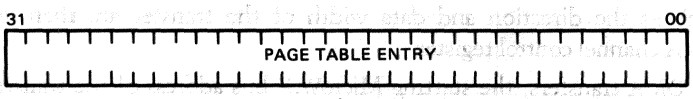


Figure 20 • MicroVAX 78532 DMA Current Page Table Entry Registers (0-3) Format

DMA Current Physical Address Registers (0-3)—If a MicroVAX bus error occurs, the DMA current physical address (DCPA0 through DCPA3) registers can usually be decremented by 4 to obtain the MicroVAX longword physical address associated with the error. An exception is when bits 8:0 = 0 during a mapped transfer (i.e., a page boundary is crossed) and the longword that caused the error is the last longword of the previous page. These registers are read only. The format of the register information is shown in Figure 21.

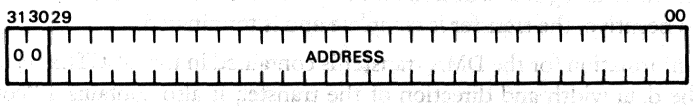


Figure 21 • MicroVAX 78532 DMA Current Physical Address Registers (0-3) Format

• Channel Operations

The type of operation performed by a channel is specified by MODE bits 05 and 04 of the appropriate DMA configuration register (DCCTL0 through DCCTL3). A channel operation is performed by the following:

- The configuration data is entered into the DMA global control register.
- The I/O device involved in the transfer must be appropriately configured and enabled. This is usually accomplished by an access operation that writes data to the I/O device.
- In the MicroDMA, the user registers that define the parameters of the channel operation are written (initialized) with appropriate data. If mapped transfers are to occur, the DMA global base register and DMA system base register must also be initialized.
- The data that configures the channel and initiates the channel operation is written into the DMA channel control register.

DMA Transfers

A DMA transfer requires the user to specify the starting MicroVAX bus address, the starting I/O bus address of the transfer, and the number of data bytes to be transferred. The configuration parameters such as the direction and data width of the transfer are then written into the appropriate DMA channel control register.

For unmapped DMA transfers, the starting MicroVAX bus address of the transfer is completely specified by the physical byte address of the beginning of the data buffer. This is contained in the DMA MicroVAX physical address register. For mapped DMA transfers, the system virtual address of the page table entry that points to the first buffer page and the byte offset from the start of that page to the first data byte must be entered. These addresses specify the beginning of the buffer in virtual memory when the DMA system base register and the DMA global base register are initialized. The system software computes these quantities from a virtual address and the process context, and loads them into the appropriate DCSPTEx register and the DCBOx register to define the virtual buffer. The starting I/O bus address of a transfer is contained in the DMA I/O base address register.

The initial byte count is contained in the DMA initial byte count register (DCIBCx). At the beginning of the transfer, the DCIBCx register information is loaded into the current byte count register (DCBCx) and the register is decremented as the transfer progresses. When DCBCx reaches zero or becomes negative, the transfer is complete and is terminated.

Configuration information for the DMA transfer is contained in the DCCTLx register. In addition to specifying the data width and direction of the transfer, it also contains a "count mode" that specifies which addresses will be incremented and the value of the increment as data is transferred from one bus to another. Addresses to or from the MicroVAX bus may be incremented by 0 or 4 because memory is always addressed in longwords on that bus. Addresses to or from the I/O bus may be incremented by 0, 1, 2, or 4 depending on the programmed width of the I/O bus. I/O bus addresses are not incremented for DMA transfers to or from a peripheral device through its data register. I/O bus addresses are incremented for I/O bus memory to MicroVAX memory transfers.

Table 14 lists the initial conditions of the registers involved with a DMA transfer. When these conditions have been established, a DMA transfer is initiated by the assertion of the $\overline{\text{ITR}}$ signal by the I/O device. A simplified flow diagram of the actions of an I/O device, the MicroDMA controller, the MicroVAX CPU, and MicroVAX memory during a typical DMA transfer is shown in Figure 22.

Table 14 lists the initial conditions of the registers involved with a DMA transfer. When these conditions have been established, a DMA transfer is initiated by the assertion of the $\overline{ITR}$ signal by the I/O device. A simplified flow diagram of the actions of an I/O device, the MicroDMA controller, the MicroVAX CPU, and MicroVAX memory during a typical DMA transfer is shown in Figure 22.

Table 14 • MicroVAX 78532 DMA Transfer Initial Conditions

Register	Bit	Content
DGCTL		OC(16) (DEN = 1, WID 03:02 (byte)
DCIOBA2		1001(16) (I/O base address)
DCIBC2		0A(16) (Initial byte count)
DCBO2		33 16) (Byte offset)
DCSPTE2		System virtual address of base page table entry
DCCTL02	03	1 (Transfer is to MicroVAX)
DCCTL2	05:04	10 (DMA mode)
DCCTL2	07:06	01 (Byte wide)
DCCTL2	09:08	10 (HOLD I/O address and INC MicroVAX address)

Data to be transferred is 01,02,03,04,05,06,07,08,09, and 0A.

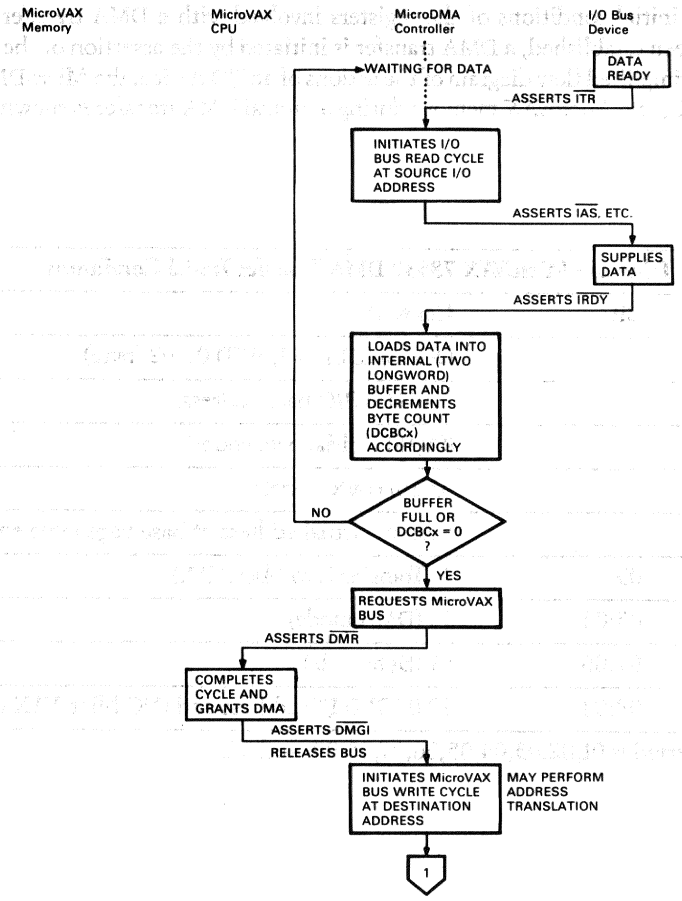


Figure 22 ▪ MicroVAX 78532 DMA Transfer Flow Diagram

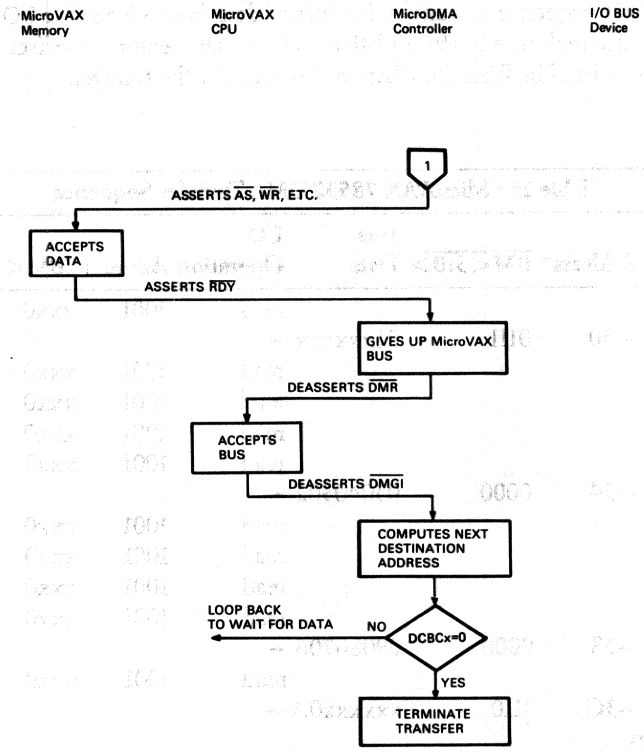


Figure 22 • MicroVAX 78532 DMA Transfer Flow Diagram (Continued)

An example of a DMA sequence to transfer 10 bytes of data from a bytewide I/O device at I/O bus address 1001 (hexadecimal) to a buffer in MicroVAX virtual memory at offset 33 (hexadecimal) from the base page is listed in Table 15. Channel 2 is used for the transfer.

Table 15 • MicroVAX 78532 DMA Transfer Sequence

Byte Count	MicroVAX Operation	Address*	$\overline{BM<3:0>}$	Bus Data	I/O Operation	Address	IBM<3:0>	Bus Data
10					read	1001	xxx0	xxxxxx01
	write	---30	0111	01xxxxxx	←			
9					read	1001	xxx0	xxxxxx02
8					read	1001	xxx0	xxxxxx03
7					read	1001	xxx0	xxxxxx04
6					read	1001	xxx0	xxxxxx05
	write	---34	0000	05040302	←			
5					read	1001	xxx0	xxxxxx06
4					read	1001	xxx0	xxxxxx07
3					read	1001	xxx0	xxxxxx08
2					read	1001	xxx0	xxxxxx09
1	write	---38	0000	09080706	←			
0					read	1001	xxx0	xxxxxx0A
	write	---3C	1110	xxxxxx0A	←			
Transfer terminates								

*The upper bits of the MicroVAX address depend on the page table information and are not shown. The lower two bits of the MicroVAX address are indeterminate and not necessarily zero.

Window Transfers

To perform a window transfer, a window (i.e., a set of I/O bus addresses that correspond to locations in MicroVAX memory) must first be defined by the user. The user application decodes I/O bus addresses (IDAL<23:00> such that the $\overline{ITRx}$ signal is asserted when an address in the window region is referenced.

When the I/O processor begins a window access, the MicroDMA uses the $\overline{IBM<3:0>}$ information, and the width of the I/O bus (byte, word, or longword) the effective byte count of the transfer and to perform substitution for the low one or two bits of the incoming address, shown in Figure 23 and Table 16.

For unmapped window transfers, the base of the window region is completely specified by the physical byte address of the window in MicroVAX memory. This must be loaded into the MicroDMA physical address register (DCUPAx), and is added to the window address derived from the I/O bus device requesting the window access.

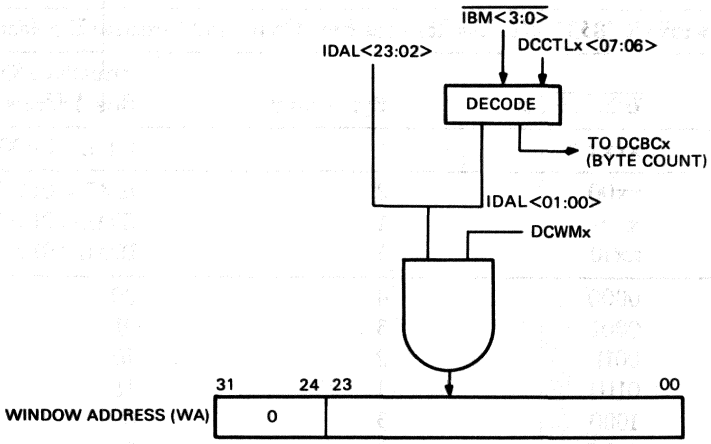


Figure 23 • MicroVAX 78532 Window Addressing Logic

This window address is derived by ANDing the effective 24-bit I/O address with the window mask register DCWMx). Using the window mask to mask off the "don't care" bits from the address, simplifies the external hardware required to decode the $\overline{\text{ITRx}}$ signal. The sum of this window address and the DCUPAx register forms the MicroVAX physical address of the window transfer. The actual transfer will require more than one access to MicroVAX memory is the access bytes cross of longword boundary.

For mapped window transfers, both the system virtual address of the page table entry and byte offset in the first window page must be specified similarly to a DMA transfer. The incoming I/O bus address and byte masks define a displacement from the base of the window. This is masked by the contents of DMA window mask register and used to compute a new system virtual address PTE and byte offset.

For both mapped and unmapped window transfers, the width of the channel is defined by DMA channel control register bits 7 and 6.

Table 16 • MicroVAX 78532 Window Transfer Byte Count and Effective Displacement

Channel Width	$\overline{\text{IBM}}$<3:0>	Byte Count	Effective I/O Bus Address <01:00>
Byte	xxxx	1	IDAL<01:00>
Word	xx00	2	IDAL<01>'0*
	xx01	1	IDAL<01>'1
	xx10	1	IDAL<01>'0
Longword	0000	4	00
	0001	3	01
	0011	2	10
	0111	1	11
	1000	3	00
	1001	2	01
	1011	1	10
	1100	2	00
	1101	1	01
	1110	1	00

*(') indicates concatenation.

If the MicroVAX CPU performs an I/O bus access when a window access is being performed, a deadlock may occur as the MicroVAX and I/O processor each wait for the other's bus.

The I/O processor can ensure a unique access to MicroVAX memory. Before beginning a window access, the I/O processor must assert $\overline{\text{VBR}}$ and wait for $\overline{\text{VBG}}$ to be asserted. Because the MicroVAX bus is acquired by the MicroDMA before the window access, the possibility of deadlock is eliminated. After the window access (or accesses), $\overline{\text{VBR}}$ should be deasserted to allow the MicroVAX processor to regain control of its bus.

The MicroDMA chip contains hardware that detects the presence of a deadlock if it occurs. It breaks the deadlock by causing a bus error on the MicroVAX bus by asserting the $\overline{\text{ERR}}$ signal and sets a bit in the DMA global control register to inform the MicroVAX CPU of the cause of the error.

Table 17 lists the initial conditions for an example of a window transfer. The sequence of events involved in this example is shown in Table 18. One byte is to be transferred from the I/O bus to the MicroVAX bus. The channel is configured in word mode. The user application decodes I/O bus addresses in the range xxx10000–xxx101FF (hexadecimal) as window references. The I/O processor writes data word 55xx to location 10022 (hexadecimal).

Table 17 • MicroVAX 78532 Window Transfer Initial Conditions

Register	Bit	Content
DGCTL		04 DEN = 1
DCWMx		1FF (Mask high bits of I/O bus address)
DCCTLx	05:04	11 (Window mode)
DCCTLx	17	1 (Physical addressing)
DCCTLx	07:06	10 (Word width)
DCUPAx		7341 (MicroVAX memory base physical address)

Table 18 • MicroVAX 78532 Window Transfer Sequence

I/O Bus Address	$\overline{IBM<3:0>}$	Data	Operation	MicroVAX Address	BM<3:0>	Data	Operation
10022	xx01	xxxx55xx	Write	7364	1110	xxxxxx55	Write

The MicroDMA adjusts the value of the lower bits of the I/O bus address, as shown in Table 16. A byte count of 1 is sent to the DMA current byte count register.

I/O DMA Transfer

An I/O DMA transfer is an unmapped, unbuffered data transfer between an I/O bus source and an I/O bus destination. Because buffering and data alignment are not performed, the source and destination addresses must be aligned on "natural" boundaries. If a channel is configured to perform word transfers, the addresses must be a multiple of 2. If a channel is configured to perform longword transfers, the addresses must be a multiple of 4. An I/O DMA operation has the following sequence:

1. The appropriate $\overline{ITR<3:0>}$ signal is asserted by an I/O device.
2. An I/O bus read cycle is performed at the address specified by the DMA I/O source address register (DCIDS).
3. If bit 8 of the DMA channel control register (DCCTL) is set, the DCIDS register is incremented by the width of the channel specified by bits 7:6 of the DCCTL register. If cleared, the DCIDS register is not changed.
4. An I/O bus write cycle is performed at the address specified by the DMA I/O destination address (DCIDD) register. The appropriate $\overline{ITR<3:0>}$ line should be deasserted by the end of this write cycle if the I/O device is not ready for another transfer.
5. If bit 9 of DCCTL register is set, the DMA destination address (DCIDD) register is incremented by the width of the channel. If cleared, the DCIDD register is not changed.
6. The byte count initially contained in DCIBC register and subsequently contained in DCBC register is decremented by the width of the channel. If the byte count is zero or a negative value, the transfer terminates. If it is not, the transfer continues at step 1.

• Access Operations

Figure 24 shows the relationship between a user-defined "access region" in MicroVAX physical address space and its counterpart in I/O bus address space. The user application decodes MicroVAX bus addresses such that $\overline{CSL}$ is asserted whenever a reference to the access region is made. The access region always starts on a 16-Mbyte boundary and the lower 512 bytes of the region are reserved for the MicroDMA internal registers.

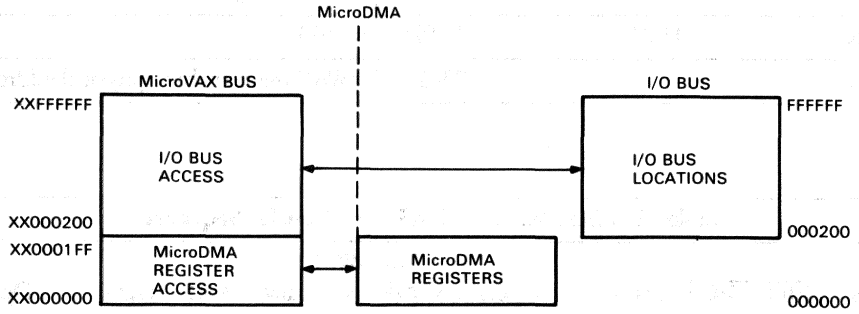


Figure 24 • MicroVAX 78532 Access Operation

When the $\overline{CSL}$ signal is asserted and the address on lines DAL<23:09> is equal to zero, a MicroDMA internal register is accessed by the address on lines DAL<08:00>. When the DAL<23:09> address is not zero, the I/O bus is accessed.

An access operation may require the transfer of a byte, word, or longword as determined by bits 4 and 3 of the DGCTL register and the information on lines $\overline{BM}$<3:0>. The following are more detailed examples of the access operation.

Example 1—Table 19 lists the sequence required to perform an access operation to write a word in I/O bus memory from the MicroVAX CPU. The number 1234 (hexadecimal) is to be written into location 2002 (hexadecimal) and the I/O bus width is 16 bits. The MicroDMA has an address of 21000000 (hexadecimal) in MicroVAX physical address space.

Table 19 • MicroVAX 78532 Access Operation Sequence 1

MicroVAX Address	$\overline{BM}$<3:0>	Data	Operation	I/O Address	$\overline{IBM}$<3:0>	Data	Operation
21002000	0011	1234xxxx	Write→	2002	xx00	1234	Write

Example 2—Table 20 lists the sequence required to perform access operation 2 from the MicroVAX CPU. The number 332211 (hexadecimal) is read from location 2001 (hexadecimal). The I/O bus width is 8 bits. The MicroDMA has an address of 21000000 (hexadecimal)in MicroVAX physical address space. The initial condition of the DGCTL register is bits 4:3 = 01 (I/O bus byte width).

Table 20 • MicroVAX 78532 Access Operation Sequence 2

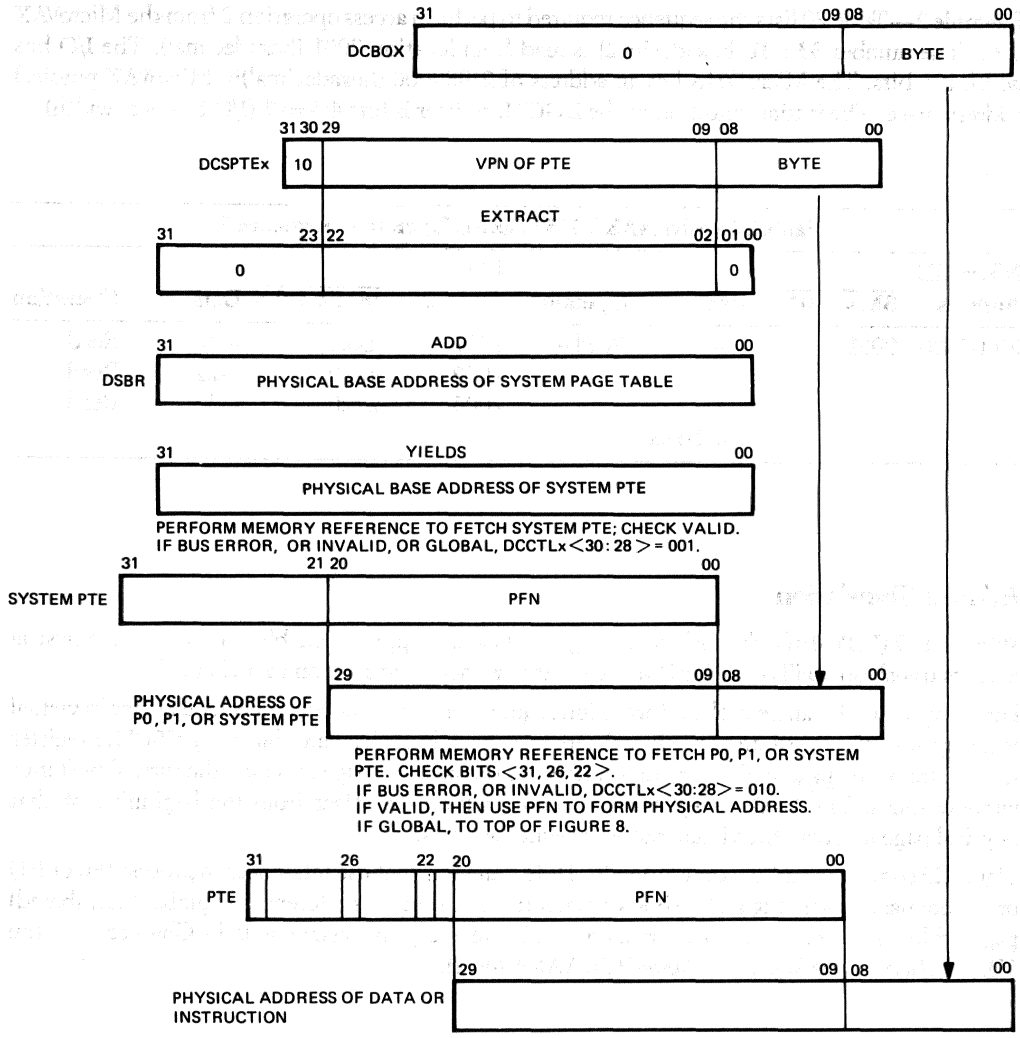
MicroVAX Address	$\overline{BM<3:0>}$	Data	Operation	I/O Address	$\overline{IBM<3:0>}$	Data	Operation
21002000	0001	-----	Read←	2001	xxx0	---11	Read
				2002	xxx0	---22	Read
				2003	xxx0	---33	Read
		332211xx					

• Address Translation

When bit 17 (PHYS) of a channel control register is clear, mapping is enabled and virtual to physical address translation will occur for DMA and window mode transfers on that channel.

For mapped DMA transfers, the information requirements to completely specify a buffer in virtual memory after the DSBR and DGBR registers have been initialized are that the DCSPTEx register must point to the page table entry that references the physical page in which the virtual buffer or window starts. The DCBOx register must contain the byte offset from the beginning of that physical page to where the virtual buffer or window starts.

Figure 25 shows the address translation for DMA transfers that use references to process (P0 or P1) or system page tables. Figure 26 shows a DMA transfer that uses references to global (i.e., shared) page tables. Further information related to VAX memory management is in Chapter 5 of the *VAX-11 Architecture Reference Manual* (EK-VAXAR-RM).



PTE<31, 26, 22>	PTE TYPE
1XX	VALID PFN
000	VALID PFN
001	GLOBAL PTE (SEE FIGURE 26)
01X	INVALID, I/O ABORT

Figure 25 ▪ MicroVAX 78532 DMA Address Translation for Process and PTE References

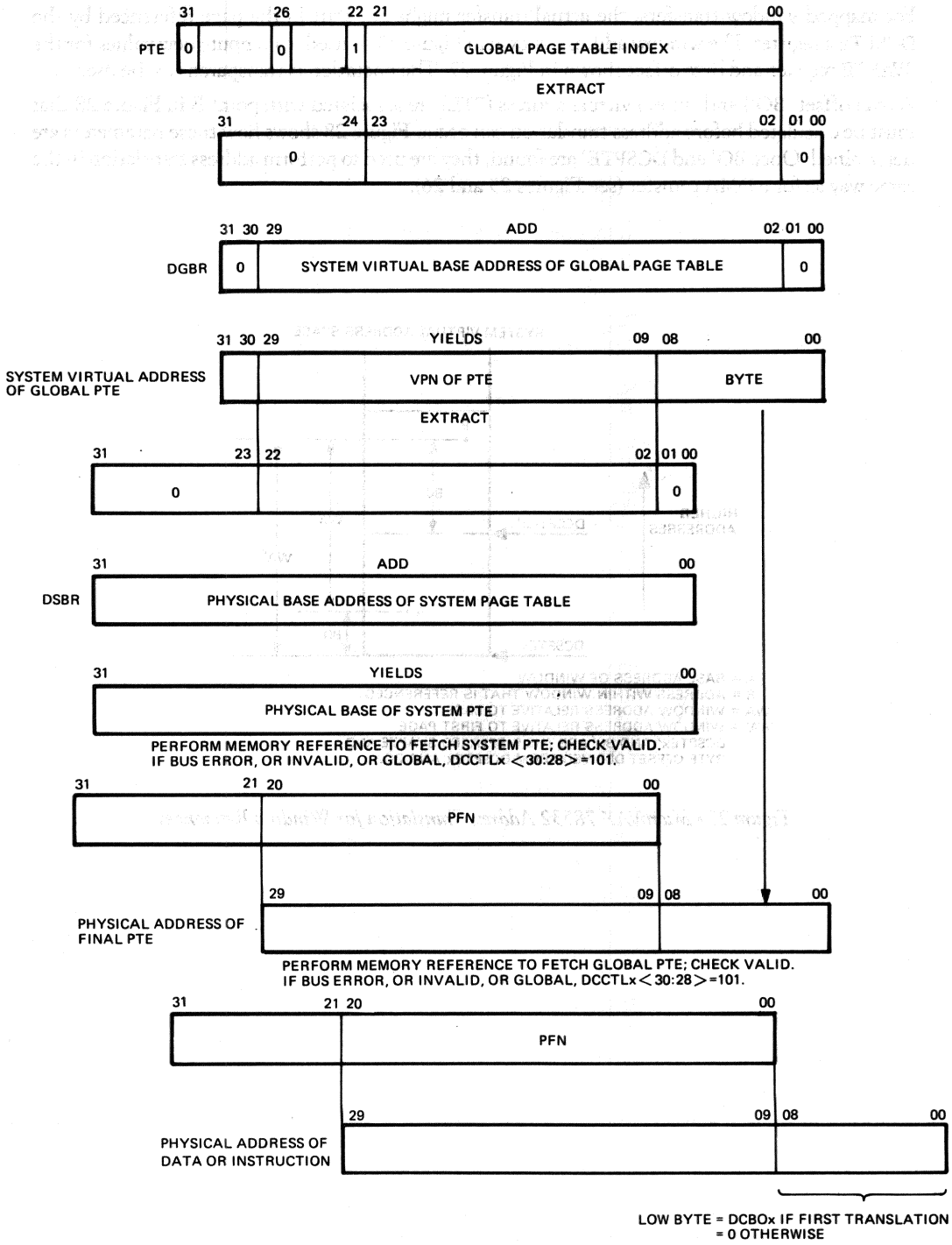
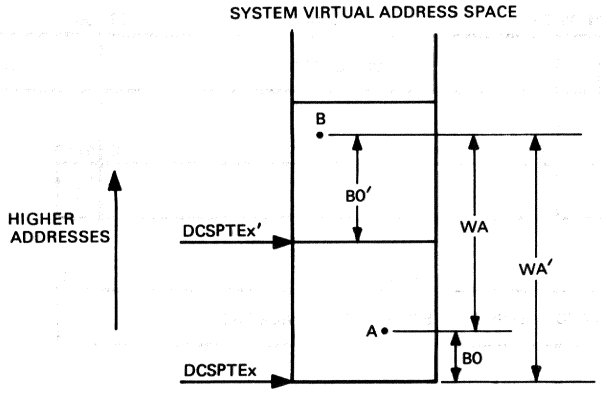


Figure 26 • MicroVAX 78532 DMA Address Translation for Global References

For mapped window transfers, the actual transfer might not start in the page referenced by the DCSPTEx register. The window address, shown in Figure 23, is used to compute new values for the SVAPTE register and byte offset shown in Figure 27. The operation is transparent to the user.

A byte offset (BO') and system virtual address (PTE) are associated with point B in Figure 28 that must be calculated before address translation can occur. Figure 28 shows how these parameters are determined. Once BO' and DCSPTE' are found, they are used to perform address translation in the same way as for a DMA transfer (see Figures 25 and 26).



A = BASE ADDRESS OF WINDOW
B = ADDRESS WITHIN WINDOW THAT IS REFERENCED
WA = WINDOW ADDRESS RELATIVE TO BASE
WA′ = WINDOW ADDRESS RELATIVE TO FIRST PAGE
DCSPTEx′ AND BO′ ARE THE DISPLACED SVAPTE AND BYTE OFFSET DERIVED FROM DCSPTEx AND BO.

Figure 27 • MicroVAX 78532 Address Translation for Window References

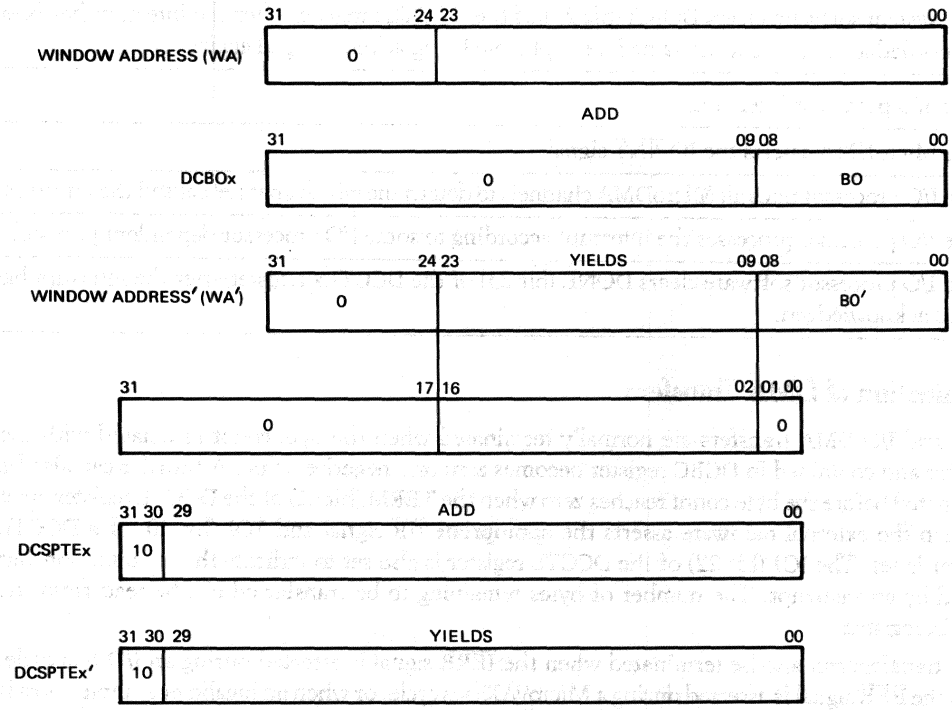


Figure 28 ▪ MicroVAX 78532 Window Reference Parameters

▪ MicroDMA Interrupts

The MicroDMA can interrupt the MicroVAX CPU or an I/O processor, depending on the contents of the interrupting channel's DCCTLx register.

The termination of a DMA transfer sets DONE (bit 3) of the DMA channel control (DCCTL) register resulting in an interrupt if IE (bit 02) is set. The interrupt will be processed by the MicroVAX CPU if IIOP (bit 11) is cleared or by an I/O processor if bit 11 is set.

For a MicroVAX CPU interrupt,

- The MicroDMA asserts an $\overline{IRQ<3:0>}$ line according to the level encoded by IPR (bits 16 and 15) of the DCCTLx register.
- The MicroVAX CPU responds by initiating an interrupt acknowledge bus cycle.The MicroDMA provides an interrupt vector from the the DCINTx register.
- If the system contains more than one MicroDMA controller, the MicroDMA closest to the CPU with respect to the interrupt daisychain that has posted an interrupt at the current level will participate in the interrupt acknowledge cycle.
- If more than one channel on the MicroDMA controller has posted an interrupt at the current level, the channel with the highest priority will return the interrupt vector to the CPU.

- The system software clears DONE (bit 23) of the DCCTLx register after the interrupt has been acknowledged to prevent the same interrupt from being acknowledged again.

For an I/O processor interrupt,

- The MicroDMA asserts the $\overline{\text{IOPINT}}$ signal.
- The I/O processor polls all MicroDMA channels to determine which channel caused the interrupt.
- The I/O processor processes the interrupt according to some I/O processor dependent protocol.
- The I/O processor software clears DONE (bit 23) of the DCCTLx register after the interrupt has been acknowledged.

Termination of DMA Transfers

DMA and I/O DMA transfers are normally terminated when the byte count associated with the transfer and contained in DCBC register becomes zero or a negative value. A transfer can also be terminated before the byte count reaches zero when the TERM (bit 01) of the DCCTL register is set or when the external hardware asserts the appropriate $\overline{\text{ITR}}$ signal and TOI (bit 10) of a DCCTL register is set. The IOI (bit 22) of the DCCTL register is also set to indicate that termination was caused by an interrupt. The number of bytes remaining to be transferred can be read from the DCBCx register.

DMA transfers can also be terminated when the $\overline{\text{IERR}}$ signal is asserted during an I/O bus cycle, when the $\overline{\text{ERR}}$ signal is asserted during a MicroVAX bus cycle, or when an invalid page table entry is referenced.A global page table entry reference that is either invalid or refers to another global page table entry will also terminate the transfer. If the transfer is terminated by an error, a corresponding bit in the DCCTL register is set to specify the error type.

The MicroDMA performs the following for all DMA transfer that are terminated.

- Sets DONE (bit 23) of the DCCTL register.
- Sets IOI (bit 22) of the DCCTL register if an I/O interrupt caused the termination.
- Clears ENABLE (bit 00) of the DCCTL register.
- Asserts the IOPINT signal if IE (bit 02) and IIOP (bit 11) of the DCCTL register are set.
- Initiates a MicroVAX interrupt at the level specified by IPR (bits 16 and 15) of the DCCTL register if IE (bit 02) is set and IIOP (bit 11) is cleared.

Chaining

The MicroDMA includes logic to automatically switch channels after the data transfer has terminated. This is defined as chaining and is normally used by I/O subsystems that continuously transfer data at high data rates. To reduce the time required to service an interrupt and reconfigure a channel following the termination of a DMA transfer, chaining is used to switch channels and the buffers associated with the channels to prevent data loss.

Chaining is enabled by setting CHAIN (bit 12) of the DCCTL register and the next channel in the chain is specified by NXTCH (bits 14 and 13). If the current DMA transfer terminates without error, the channel specified by the NXTCH bits begins operation. Interrupts that were enabled for the first channel will be serviced when the next channel in the chain is active.

The pins for the $\overline{\text{ITR}}<3:0>$ channels involved in a chain should be connected together to preserve the transfer requests.

The pins for the $\overline{ITR}$<3:0> channels involved in a chain should be connected together to preserve the transfer requests.

• MicroDMA Reset

When the RESET line is asserted, the RESET (bit 00) of the DGCTL register is set and the MicroDMA performs the following reset operation:

- DGCTL register—Clears the DEAD (bit 31), IOWE (bit 10), WID bit<04:03> and RESET (bit 00)
- All DCCTL registers—Clears the error condition (bits 31:27), DONE (bit 23), IOI (bit 22), IE (bit 02), TERM (bit 01), and ENABLE (bit 00)

• Bus Cycles

Operations such as DMA transfers, window transfers, and access operations require the execution of one or more bus cycles on the MicroVAX bus and/or the I/O bus. A bus cycle is a sequence of events that results in a transfer of information between a bus master and a bus slave. Bus cycles usually involve read, write, interrupt acknowledge, or DMA operations.

This section briefly describes the characteristics and protocols of the various MicroDMA bus cycles. Bus cycles on the I/O bus and the MicroVAX bus are similar. Detailed timing information is contained in the ac Characteristics section.

MicroVAX Bus Cycles

MicroVAX Bus Read Cycle—A MicroVAX bus master performs a MicroVAX bus read cycle when it requires information from another MicroVAX bus device. Address and control information is sent to the bus slave during the first part of a read cycle and the second part of the cycle is used to read the data. The sequence of events follows:

1. The bus master transfers the physical longword address of the location to be read on the DAL<29:02> lines.
2. The $\overline{WR}$ line is unasserted and the CS<2:0> lines are driven by the bus master to indicate the type of cycle being performed.
3. The bus master drives the $\overline{BM}$<3:0> lines.
4. The bus master asserts the $\overline{AS}$ signal to indicate that the address on the DAL lines is valid and can be latched. When the MicroDMA is being addressed for register or I/O bus access, the $\overline{CSL}$ signal must be asserted. The $\overline{AS}$ signal also qualifies the information on the CS<2:0>, $\overline{WR}$, and $\overline{BM}$<3:0> lines.
5. The bus master asserts the $\overline{DS}$ signal to indicate that the bus is available to receive the required information. The bus master also asserts the $\overline{DBE}$ signal at this time which can be used to control DAL line bus transceivers.
6. If the slave can supply valid data within minimum access time, it asserts the $\overline{RDY}$ signal at the first sample window after the assertion of $\overline{AS}$ and the master latches the data. If data is not available at this time, the master waits eight periods of the CLKI signal and samples the $\overline{RDY}$ signal again. This sequence continues every eight clock periods until the $\overline{RDY}$ signal is asserted. If a bus error occurs, the external logic or the bus slave will respond by asserting the $\overline{ERR}$ signal. The bus master must then process the error. The current bus cycle is completed when the $\overline{RDY}$ or $\overline{ERR}$ signals are asserted. The bus master latches the requested data and deasserts the $\overline{DS}$ line.
7. The bus master deasserts the $\overline{CSL}$ line if it is asserted, and asserts the $\overline{AS}$ and $\overline{DBE}$ lines to end the bus cycle.

MicroVAX Bus Write Cycle

A MicroVAX bus master performs a MicroVAX bus write cycle to transfer information to another MicroVAX bus device. During the first part of the bus cycle, address and control information is sent to the bus slave. During the second part, the data to be written is transferred. The sequence of events follows:

1. The bus master drives the physical longword address of the location to be read onto the DAL<29:02> lines.
2. The $\overline{WR}$ signal is asserted and the CS<2:0> lines are driven by the bus master as required.
3. The bus master drives the $\overline{BM}$<3:0> lines and asserts the $\overline{AS}$ signal to indicate that the address on the DAL lines is valid and can be latched. When the MicroDMA is being addressed for register or I/O bus access, the $\overline{CSL}$ signal must be asserted. The $\overline{AS}$ signal also qualifies the information on the CS<2:0>, $\overline{WR}$, and $\overline{BM}$<3:0> lines.
4. The bus master asserts the $\overline{DBE}$ signal, drives data onto the DAL lines, and asserts the $\overline{DS}$ line to indicate that the data is valid.
5. If the slave can accept valid data within the minimum write cycle time, it asserts the $\overline{RDY}$ signal at the first sample window after the assertion of the $\overline{AS}$ signal and latches the data when the $\overline{DS}$ line is deasserted. If the data cannot be accepted at this time, the master waits eight periods of the CLKI signal and samples the $\overline{RDY}$ signal again. This sequence continues every eight clock periods until the $\overline{RDY}$ line is asserted. If a bus error occurs, the external logic or the bus slave responds by asserting the $\overline{ERR}$ signal and the bus master must then process the error. The current bus cycle is completed when the $\overline{RDY}$ or $\overline{ERR}$ signal is asserted.
6. The bus master deasserts the $\overline{DS}$ signal to indicate that it will remove the data from the DAL lines and deasserts the $\overline{AS}$ and $\overline{DBE}$ lines to end the bus cycle.

MicroVAX Bus DMA Cycle

This cycle is used to force the bus master to release control of the DAL lines and related control signals to another MicroVAX bus device. The sequence of events follows:

1. A device requests the use of the MicroVAX bus from the bus master by asserting the $\overline{DMR}$ signal.
2. If the bus master is not performing a locked read cycle, it responds to the assertion of the $\overline{DMR}$ by releasing the DAL<31:00>, $\overline{AS}$, $\overline{DS}$, $\overline{DBE}$, $\overline{WR}$, and $\overline{BM}$<3:0>, lines.
3. The bus master asserts the $\overline{DMG}$ signal when it releases control of the bus and grants the use of the bus to the requesting device.
4. One or more read and/or write cycles occur on the bus between the requesting device (the new bus master) and its slave.
5. When the requesting device is finished with the bus, it deasserts the $\overline{DMR}$ line to return control of the bus to the original bus master.
6. The bus master deasserts the $\overline{DMG}$ signal and resumes operation on the bus.

MicroVAX Bus Interrupt Acknowledge Cycle

A MicroVAX bus master performs an interrupt acknowledge cycle to acknowledge an interrupt request from a slave through the $\overline{IRQ}$ lines and to read a vector. The timing for this cycle is the same as the MicroVAX bus read cycle shown in Figure 30. The sequence of events follows:

1. The bus master transfers the priority of the interrupt being acknowledged onto lines DAL<04:00>. The DAL<29:05> lines contain zeros and the DAL<31:30> lines contain the value of 10.
2. The CS<2:0> lines are driven by the bus master to indicate an interrupt acknowledge cycle.
3. The bus master asserts all the $\overline{BM}$<3:0> bits. The $\overline{WR}$ signal is unasserted.

4. The bus master asserts the $\overline{AS}$ signal to indicate that the interrupt priority on the DAL lines is valid and asserts the $\overline{DS}$ signal to indicate that the bus is available to receive incoming data. The bus master also asserts the $\overline{DBE}$ line, which can be used to control DAL line transceivers.
5. If no error occurs, the external logic or the bus slave transfers the interrupt vector on the DAL<09:02> lines, the normal processing/Q-bus processing flag on the DAL<00> line, and asserts the $\overline{RDY}$ signal. Refer to the *MicroVAX CPU User's Guide* for a description of the normal processing/Q-bus processing flag. The DAL<15:10,01> lines must be set to a valid high or low level in accordance with the setup times shown in Figure 30.
6. If an error occurs, the external logic or the bus slave asserts the $\overline{ERR}$ signal. The bus master cancels the cycle and ignores the data on the DAL lines.
7. The bus master latches the interrupt vector, deasserts the $\overline{DS}$ signal, and deasserts the $\overline{AS}$ and $\overline{DBE}$ signals to end the cycle.

I/O Bus Cycles

I/O Bus Read Cycle—An I/O bus master performs an I/O bus read cycle when it requires information from another I/O bus device. During the first part of a read cycle, address and control information is sent to the bus slave. During the second part of the cycle the data is read. The sequence of events follows:

1. The bus master drives the physical longword address of the location to be read onto the IDAL<23:00> lines.
2. The $\overline{IWR}$ signal is left unasserted. The bus master asserts the $\overline{IBM<3:0>}$ as required.
4. The bus master asserts the $\overline{IAS}$ signal to indicate that the address on the IDAL lines is valid and ready to be latched. The $\overline{IAS}$ signal also qualifies the information on the $\overline{IBM<3:0>}$ and $\overline{IWR}$ lines.
5. If the MicroDMA is not I/O bus master and is performing a window access, then the $\overline{ITR}$ signal for the requested window channel should be asserted when the $\overline{IAS}$ signal is asserted. If a MicroDMA register access is to be performed, the $\overline{IREG}$ signal should be asserted at this time.
6. The bus master asserts the $\overline{IDS}$ signal to indicate that the bus is available to receive the required information. At this time the bus master also asserts $\overline{IDBE}$ which can be used to control IDAL line transceivers.
7. If the slave can supply valid data within the minimum access time, it asserts the $\overline{IRDY}$ signal at the first sample window after the assertion of the $\overline{IAS}$ signal and the master latches the data. If it cannot supply valid data during this time, the master waits four periods of the CLKI signal and samples the $\overline{IRDY}$ signal again. This sequence continues every four clock periods until the $\overline{IRDY}$ line is asserted. If a bus error occurs, the external logic or the bus slave responds by asserting the $\overline{IERR}$ signal, and the bus master must then process the error. The current bus cycle is completed when the $\overline{IRDY}$ or $\overline{IERR}$ signals are asserted.
8. The bus master latches the requested data, deasserts the $\overline{IDS}$ signal, and deasserts the $\overline{IAS}$ and $\overline{IDBE}$ signals to end the bus cycle.

I/O Bus Write Cycle

An I/O bus master performs an I/O bus write cycle to transfer information to another I/O bus device. During the first part of the cycle, address and control information is sent to the bus slave. During the second part, the data is written. The sequence of events follows:

1. The bus master drives the physical longword address of the location to be read onto the IDAL<23:00> lines.
2. The $\overline{IWR}$ signal is asserted.

3. The bus master drives $\overline{IBM}<3:0>$ lines and asserts the $\overline{IAS}$ signal to indicate that the address on the IDAL lines is valid and should be latched. The $\overline{IAS}$ signal also qualifies $\overline{IBM}<3:0>$ and $\overline{IWR}$ line information.
4. If the MicroDMA is not bus master and is performing a window access, the $\overline{ITR}$ signal for the requested window channel should be asserted when the $\overline{IAS}$ signal is asserted. If a MicroDMA register access is to be performed, the $\overline{IREG}$ signal should be asserted at this time.
5. The bus master asserts the $\overline{IDBE}$ line which can be used to control the IDAL line transceivers, transfers data onto the IDAL lines, and asserts the $\overline{IDS}$ signal to indicate that the data is valid.
6. If the slave can accept valid data within the minimum write cycle time, it asserts the $\overline{IRDY}$ signal at the first sample window after the assertion of the $\overline{IAS}$ line and latches the data when the $\overline{DS}$ signal is deasserted. If the slave cannot accept the data during this time, the master waits four clock periods and samples the $\overline{IRDY}$ line again. This sequence continues every four periods of the CLKI signal until the $\overline{IRDY}$ signal is asserted. If a bus error occurs, the external logic or the bus slave responds by asserting the $\overline{IERR}$ signal and the bus master must then process the error. The current bus cycle is completed when the $\overline{IRDY}$ or $\overline{IERR}$ signal is asserted.
7. The bus master deasserts the $\overline{IDS}$ signal to indicate that it will remove the data from the IDAL lines and deasserts the $\overline{IAS}$ and $\overline{IDBE}$ signals to end the bus cycle.

I/O Bus DMA Cycle

This cycle is used to force the bus master to release control of the IDAL lines and control signals to another I/O bus device. The sequence of events follows:

1. A device requests the use of the I/O bus from the bus master by asserting the $\overline{IDMR}$ signal.
2. If the bus master is not performing a locked read cycle, it responds to the assertion of $\overline{IDMR}$ by releasing the IDAL<31:00>, $\overline{IAS}$, $\overline{IDS}$, $\overline{IDBE}$, $\overline{IWR}$, and $\overline{IBM}<3:0>$ lines.
3. The bus master asserts the $\overline{IDMG}$ line to release control of the bus and to grant the use of the bus to the requesting device.
4. One or more read or write cycles occur on the bus between the requesting device (the new bus master) and its slave.
5. When the requesting device has finished with the bus, it deasserts the $\overline{IDMR}$ signal to return control of the bus to the original bus master.
6. The bus master deasserts the $\overline{IDMG}$ signal and resumes operation on the bus.

• Specifications

The mechanical, electrical, and environmental characteristics and specifications for the MicroDMA are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Temperature: 70°C
- $V_{DD}=4.75$ V, $V_{SS}=0$ V

Mechanical Configuration

The physical dimensions of the 133-pin package are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device. The functional operation of the device at these or other conditions greater than indicated is not defined.

- Power supply voltage (V_{DD}): –0.5 V to 5.5 V
- Input or output voltage applied: V_{SS} –0.3 V to V_{DD} 0.3 V
- Storage temperature (T_S): –55°C to 125°C
- Relative humidity: 10% to 95% (noncondensing)

Recommended Operating Conditions

- Power supply voltage (V_{DD}): 5 V ±5%
- Supply current (I_{CC}) : 500 mA (maximum)
- Operating temperature (T_A): 0°C to 75°C

dc Electrical Characteristics

The dc electrical characteristics of the MicroVAX 78532 for the operating voltage and temperature ranges specified are listed in Table 21.

Table 21 • MicroVAX 78532 dc Input and Output Parameters

Symbol	Parameter	Test Condition	Requiements Min.	Max.	Units
V_{IH}	High-level input voltage		2.0	—	V
V_{IL}	Low-level input voltage		—	0.8	V
V_{OH}	High-level output voltage	I_{OH} = 400µA C_L = 100 pF	2.4	V_{DD}	V
V_{OL}	Low-level output voltage	I_{OL} = 2.0 mA C_L = 100 pF	—	0.4	V
V_{OLOD}	Low-level output open-drain voltage* $\overline{RDY}$, $\overline{ERR}$ $\overline{DMR}$, $\overline{IRA}$<3:0>)	I_{OL} = 12.5 mA, C_L = 100 pF	—	0.4	V

Symbol	Parameter	Test Condition	Requiements Min.	Max.	Units
I_{LI}	Input leakage current	$0 < V_{in} < V_{DDI}$	−10	10	A
I_{OL}	Output leakage current	$0.4 < V_{in} < V_{DDI}$	−10	10	A
I_{CC}	Active supply current	$I_{out} = 0$, $T_A = 0°C$	—	500	mA
C_{in}	Input capacitance		—	10	pF

*Minimum pullup resistor = 470 ±5%.

ac Electrical Characteristics

The electrical characteristics for the signals used to control the information transfers to and from the MicroDMA are defined in the following paragraphs. The following notes apply to both the MicroVAX bus timing diagrams and the I/O bus timing diagrams and their associated tables.

- The timing parameters are specified in terms of the clock (CLKI) period, where CLKI = t_{CIP} = P. P is nominally 25 ns.
- All times are in nanoseconds except where noted.
- ac characteristics are measured with a purely capacitive load of 100 pF. Times are valid for loads of up to 100 pF on all pins.
- ac high levels are measured at 2.0 V, and ac low levels at 0.8 V.
- S = the number of slipped microcycles during a bus cycle. A MicroVAX bus microcycle is nominally 8P or 200 ns and the I/O bus microcycle is normally 4P or 100 ns.
- N = the number of MicroVAX and I/O bus microcycles in a DMA transfer. N has a minimum value of 2.

The following notes apply to the MicroVAX bus timing diagrams and their associated tables.

- The sampling window is used to sample the $\overline{RDY}$ and $\overline{ERR}$ asynchronous signals. The $\overline{RDY}$ and $\overline{ERR}$ signals are qualified by the assertion of the $\overline{AS}$ signal. The effect of these signals on the current bus cycle is as follows:
  1. The bus cycle concludes at the end of the current microcycle if the $\overline{RDY}$ or $\overline{ERR}$ signal is asserted throughout the sampling window while the $\overline{AS}$ signal is asserted.
  2. If a transition of the $\overline{RDY}$ or $\overline{ERR}$ signals occurs during the sampling window while the $\overline{AS}$ line is asserted, the result is indeterminate.
  3. PS = Clock period (P) times slipped microcycles (S).

The following notes apply to I/O bus timing diagrams and their associated tables:

- The sampling window is used to sample the following asynchronous signals—$\overline{IRDY}$, $\overline{IERR}$, and $\overline{IDMR}$. Signals $\overline{IRDY}$ and $\overline{IERR}$ are qualified by the assertion of the $\overline{IAS}$ signal. The $\overline{IDMR}$ signal is qualified by the assertion of the $\overline{IAS}$ signal. The effect of these signals on the current bus cycle is as follows:
  1. The bus cycle concludes at the end of the current microcycle if the $\overline{IRDY}$ or $\overline{IERR}$ signal is asserted throughout the sampling window while the $\overline{IAS}$ signal is asserted.
  2. If a transition of the $\overline{IRDY}$ or $\overline{IERR}$ signals occurs during the sampling window while the $\overline{IAS}$ signal is asserted, the result is indeterminate.
  3. The $\overline{IDMR}$ signal is sampled at every I/O bus microcycle.

Clock Input Timing

Figure 29 shows the timing specifications for the clock input (CLKI) signal and Table 22 lists the timing parameters indicated in the diagram.

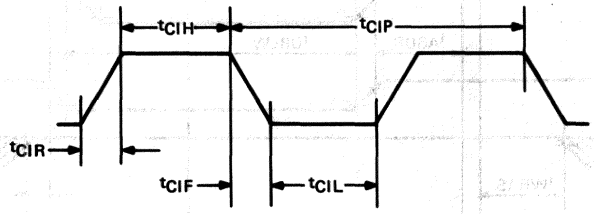


Figure 29 • MicroVAX 78532 CLKI Timing Waveform

Table 22 • MicroVAX 78532 Clock Input Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{CIF}	Clock input fall time	—	4.5
t_{CIH}	Clock input high	8.0	—
t_{CIL}	Clock input low	8.0	—
t_{CIP}	Clock period	25	50
t_{CIR}	Clock input rise time	—	4.5

MicroVAX Bus Read and Write Cycles

Figure 30 shows the MicroVAX bus master read cycle timing and Figure 31 shows the MicroVAX bus master write cycle timing. Table 23 defines the read and write cycle timing parameters.

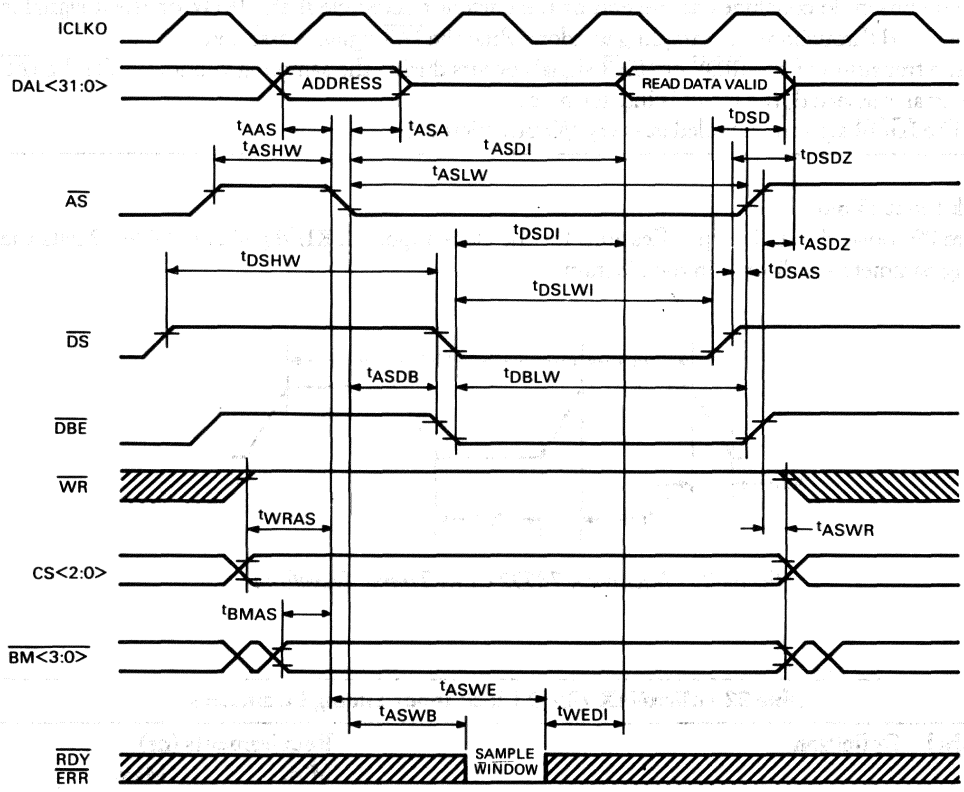


Figure 30 • MicroVAX 78532 MicroVAX Bus Master Read Cycle Timing

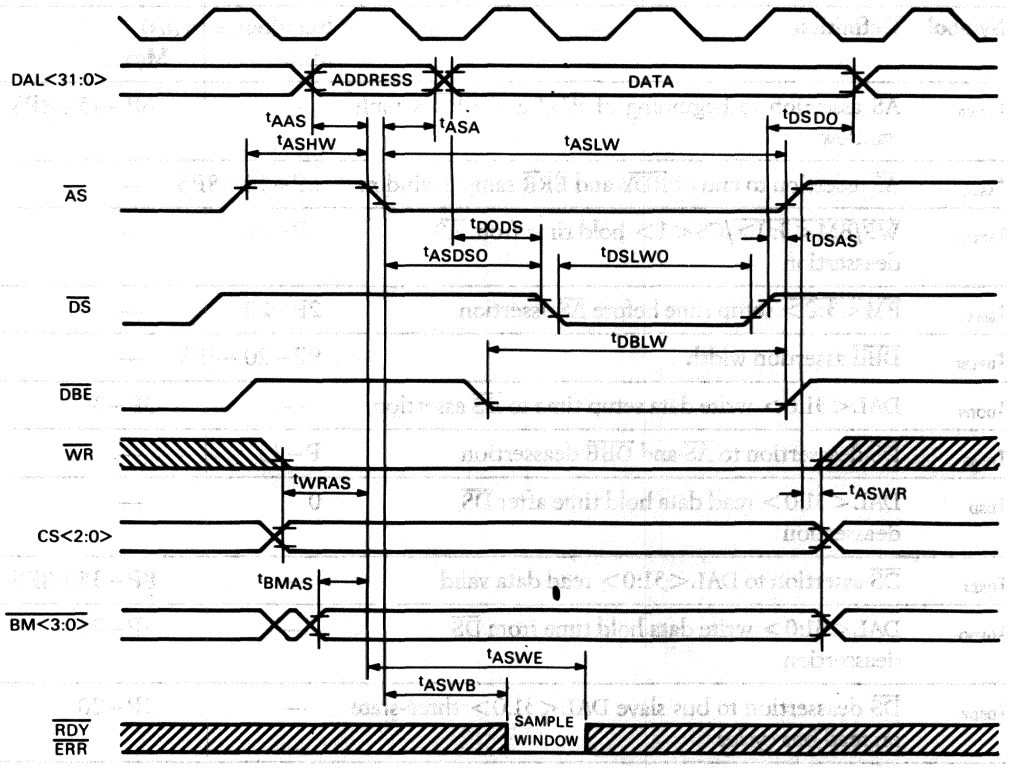


Figure 31 • MicroVAX 78532 MicroVAX Bus Master Write Cycle Timing

Table 23 • MicroVAX 78532 Bus Master Read and Write Cycle Timing Parameters

Symbol	Signal Definition	Requirements (ns) Min.	Max.
t_{AAS}	DAL<31:0> address setup time to $\overline{AS}$ assertion	2P−28	—
t_{ASA}	DAL<31:0> address hold time after $\overline{AS}$ assertion	2P−15	—
t_{ASDB}	$\overline{AS}$ assertion to $\overline{DBE}$ and $\overline{DS}$ (read) assertion	3P−15	3P+20
t_{ASDI}	$\overline{AS}$ assertion to read data valid*	—	11P−30+8PS
t_{ASDSO}	$\overline{AS}$ assertion to $\overline{DS}$ assertion (write)	5P−15	5P+20
t_{ASDZ}	$\overline{AS}$ and $\overline{DBE}$ deassertion to busslave DAL<31:0> three-state	—	2P−20
t_{ASHW}	$\overline{AS}$ deassertion width	4P−25	—
t_{ASLW}	$\overline{AS}$ assertion width	12P−15+8PS	—

Symbol	Definition	Requirements (ns) Min.	Max.
t_{ASWB}	$\overline{AS}$ assertion to beginning of $\overline{RDY}$ and $\overline{ERR}$ sample window	—	6P−45+8PS
t_{ASWE}	$\overline{AS}$ assertion to end of $\overline{RDY}$ and $\overline{ERR}$ sample window	6P+10+8PS	—
t_{ASWR}	$\overline{WR}$/$\overline{BM}$<3:0>/CS<1> hold time from $\overline{AS}$ deassertion	P−20	—
t_{BMAS}	$\overline{BM}$<3:0> setup time before $\overline{AS}$ assertion	2P−25	—
t_{DBLW}	$\overline{DBE}$ assertion width	9P−20+8PS	—
t_{DODS}	DAL<31:0> write data setup time to $\overline{DS}$ assertion	—	3P−30
t_{DSAS}	$\overline{DS}$ deassertion to $\overline{AS}$ and $\overline{DBE}$ deassertion	P−15	—
t_{DSD}	DAL<31:0> read data hold time after $\overline{DS}$ deassertion	0	—
t_{DSDI}	$\overline{DS}$ assertion to DAL<31:0> read data valid	—	8P−35+8PS
t_{DSDO}	DAL<31:0> write data hold time from $\overline{DS}$ deassertion	—	3P−20
t_{DSDZ}	$\overline{DS}$ deassertion to bus slave DAL<31:0> three-state on read bus cycles	—	3P−20
t_{DSHW}	$\overline{DS}$ deassertion width (read)	8P−50	—
t_{DSLWI}	$\overline{DS}$ assertion width (read)	8P−20+8PS	—
t_{DSLWO}	$\overline{DS}$ assertion width (write)	6P−20+8PS	—
t_{WEDI}	$\overline{RDY}$ internal sample window end to DAL<31:0> read data valid	—	5P−25
t_{WRAS}	$\overline{WR}$ and CS<1> setup time before $\overline{AS}$ assertion	3P−35	—

*Read data is valid if t_{ASDI} or t_{DSDI} conditions are satisfied.

Figure 32 shows the MicroVAX bus slave read cycle timing and Figure 33 shows the MicroVAX bus slave write cycle timing. Table 24 lists the timing parameters.

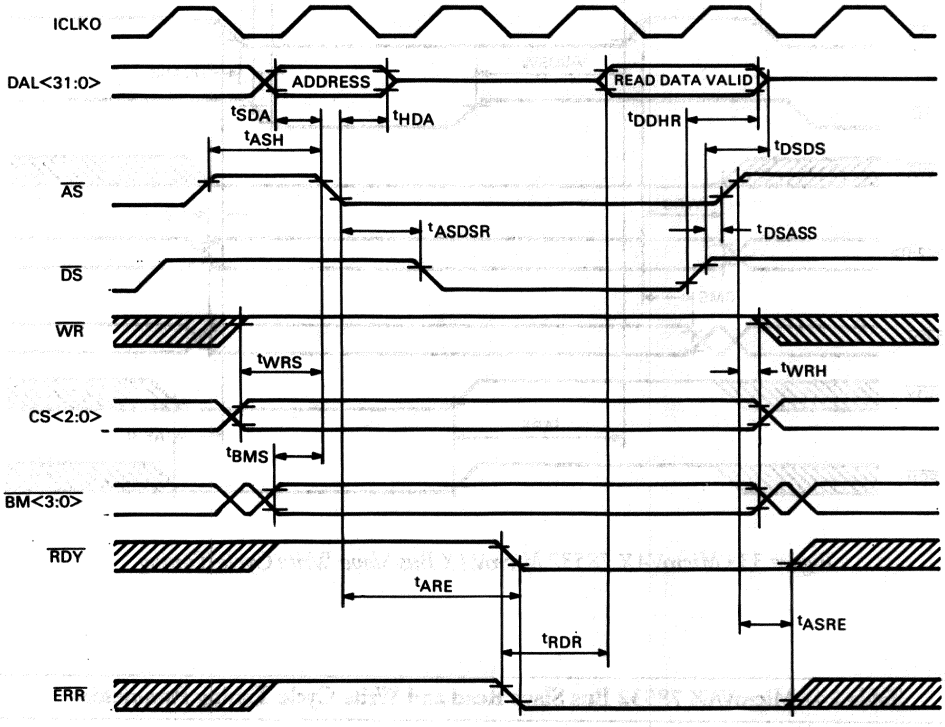


Figure 32 ▪ MicroVAX 78532 MicroVAX Bus Slave Read Cycle Timing

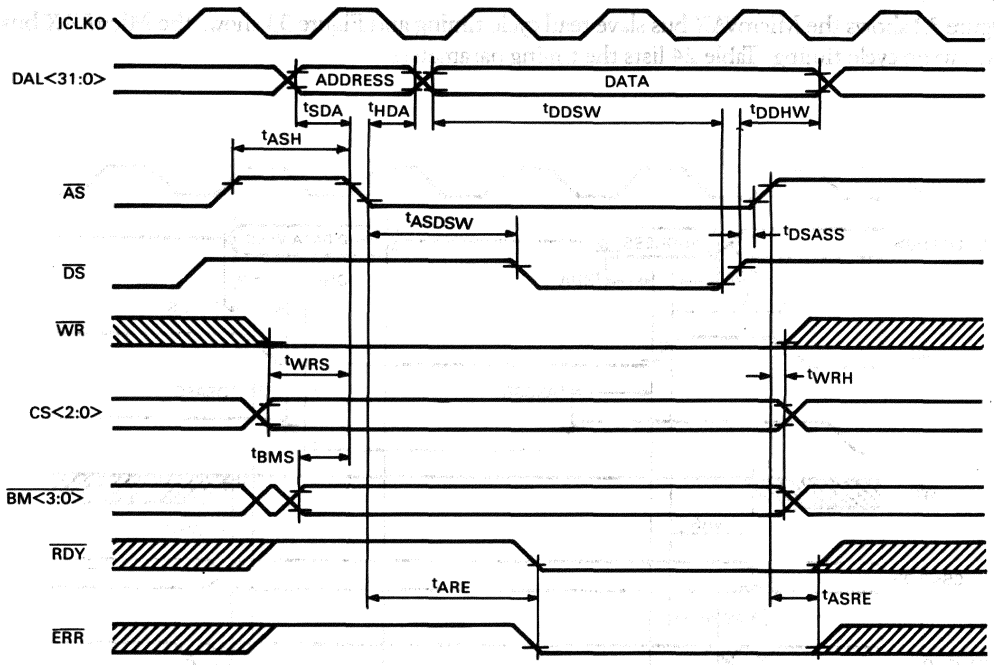


Figure 33 • MicroVAX 78532 MicroVAX Bus Slave Write Cycle Timing

Table 24 • MicroVAX 78532 Bus Slave Read and Write Cycle Timing Parameters

Symbol	Signal Definition	Requirements (ns) Min.	Max.
t_{ARE}	$\overline{AS}$ assertion to $\overline{RDY}/\overline{ERR}$ assertion for MicroDMA bus-slave cycles	*	*
t_{ASDSR}	Required $\overline{AS}$ assertion to $\overline{DS}$ assertion delay (read cycles)	3P−20	3P+25
t_{ASDSW}	Required $\overline{AS}$ assertion to $\overline{DS}$ assertion delay (write cycles)	5P−20	5P+25
t_{ASH}	$\overline{AS}$ deassertion width	2P+25	—
t_{ASRE}	$\overline{AS}$ deassertion to $\overline{RDY}/\overline{ERR}$ deassertion	—	100
t_{BMS}	$\overline{BM}$<31:00> setup time before $\overline{AS}$ assertion	2P−25	—
t_{DDHR}	DAL<31:00> data hold time after $\overline{DS}$ deassertion (slave reads)	0	—
t_{DDHW}	Required DAL<31:00> hold time after $\overline{DS}$ deassertion on MicroDMA bus-slave writes	35	—

Symbol	Signal Definition	Requirements (ns) Min.	Max.
t_{DDSW}	Required DAL<31:00> setup time before $\overline{DS}$ deassertion on MicroDMA bus-slave writes	20	—
t_{DSASS}	Required $\overline{DS}$ deassertion to $\overline{AS}$ deassertion delay	P−20	—
t_{DSDS}	$\overline{DS}$ deassertion to DAL<31:00> three-state	—	55
t_{HDA}	Required DAL<31:00> hold time after $\overline{AS}$ assertion	35	—
t_{RDR}	$\overline{RDY}$ assertion to DAL<31:00> data valid for MicroDMA bus-slave reads	—	35
t_{SDA}	Required DAL<31:00> setup time before $\overline{AS}$ assertion	15	—
t_{WRH}	$\overline{WR}$/$\overline{BM}$<31:00>/CS<2:0> hold time after $\overline{AS}$ deassertion	P−25	—
t_{WRS}	$\overline{WR}$/CS<2:0> setup time before $\overline{AS}$ assertion	3P−50	—

*t_{ARE} time depends on the system configutation. (memory speed, number of cycle slips, type of transfer, etc.)

Figure 34 shows the MicroVAX bus signal timing for the DMA cycle and Table 25 lists the MicroVAX bus DMA cycle timing parameters.

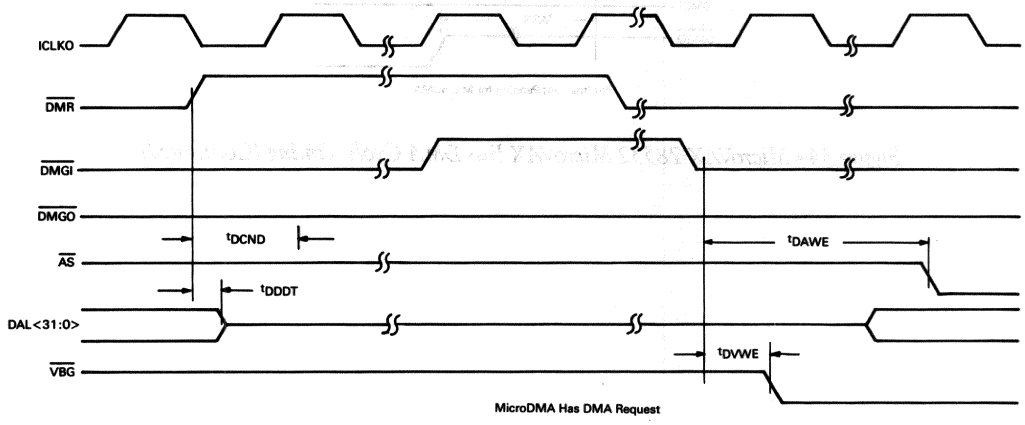


Figure 34 • MicroVAX 78532 MicroVAX Bus DMA Cycle Timing

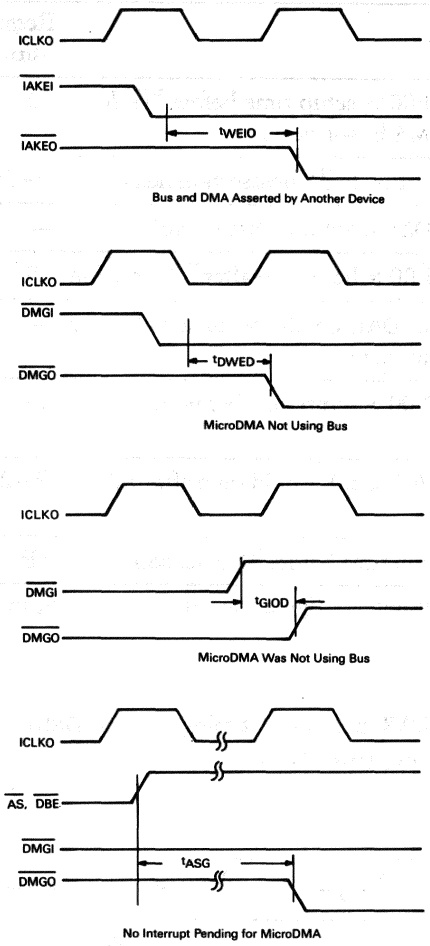


Figure 34 • MicroVAX 78532 MicroVAX Bus DMA Cycle Timing (Continued)

Table 25 • MicroVAX 78532 MicroVAX Bus DMA Cycle Timing Parameters

Symbol	Signal Definition	Requirements (ns) Min.	Max.
t_{ASG}	$\overline{AS}$ and $\overline{DBE}$ deassertion to $\overline{DMGO}$ assertion	21P−25	—
t_{DAWE}	End of $\overline{DMGI}$ sample window to $\overline{AS}$ assertion (DMA request pending for MicroDMA)	—	10P+20+4PK*
t_{DCND}	Deassert $\overline{DMR}$ to $\overline{AS}/\overline{DS}/\overline{WR}$/CS<1>/$\overline{DBE}$/ $\overline{BM}$<3:0> three-state	—	3P+25
t_{DDDT}	Deassert $\overline{DMR}$ to DAL<31:00> three-state	—	P+20
t_{DVWE}	End of $\overline{DMGI}$ sample window to $\overline{VBG}$ assertion	—	2P+33
t_{DWED}	End of $\overline{DMGI}$ sample window to $\overline{DMGO}$ assertion (no DMA request pending for MicroDMA)	—	2P+30
t_{GIOD}	Deassert $\overline{DMGI}$ to $\overline{DMGO}$ deassert	0	60
t_{WEIO}	$\overline{IAKEI}$ sample window end to $\overline{IAKEO}$ asserts	—	2P+30

*K = the number of microcycles (0, 1, 2, 3, 4) that the sequencer is busy.

MicroVAX Bus General Timing

Figure 35 shows the general signals for the MicroVAX bus timing and Table 26 lists the general timing signal parameters.

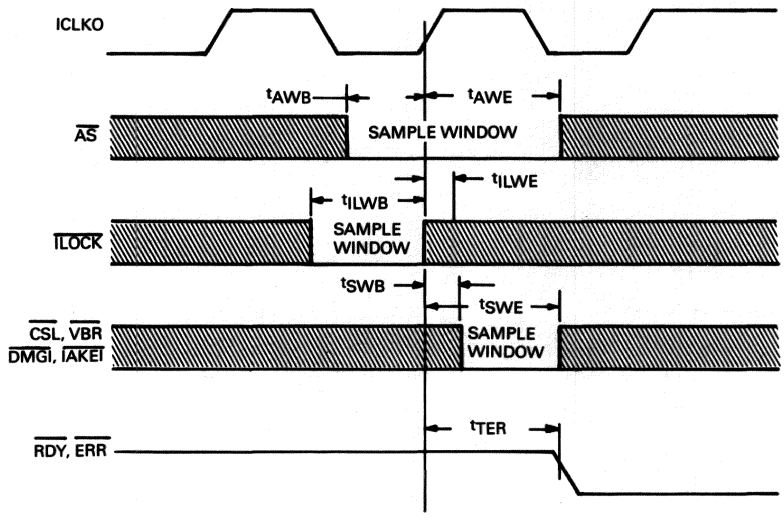


Figure 35 • MicroVAX 78532 MicroVAX Bus General Signal Timing

Table 26 • MicroVAX 78532 MicroVAX Bus General Timing Parameters

Symbol	Signal Definition	Requirements (ns) Min.	Max.
t_{AWB}	ICLKO to beginning of $\overline{AS}$ sample window	—	−35
t_{AWE}	ICLKO to end of $\overline{AS}$ sample window	3P+5	—
t_{ILWB}	ICLKO to beginning of $\overline{ILOCK}$ sample window	—	−50
t_{ILWE}	ICLKO to end of $\overline{ILOCK}$ sample window	5	—
t_{SWB}	ICLKO to beginning of $\overline{CSL}/\overline{VBR}/\overline{DMGI}/\overline{IAKEI}$ sample window	—	3P−50
t_{SWE}	ICLKO to end of $\overline{CSL}/\overline{VBR}/\overline{DMGI}/\overline{IAKEI}$ sample window	3P+5	—
t_{TER}	ICLKO to $\overline{ERR}/\overline{RDY}$ assertion	3P−5	3P+26

I/O Bus Master Read and Write Cycles

Figures 36 and 37 are timing diagrams for the I/O bus master read and write cycles, respectively. Table 27 lists the symbols and parameters for the timing signals.

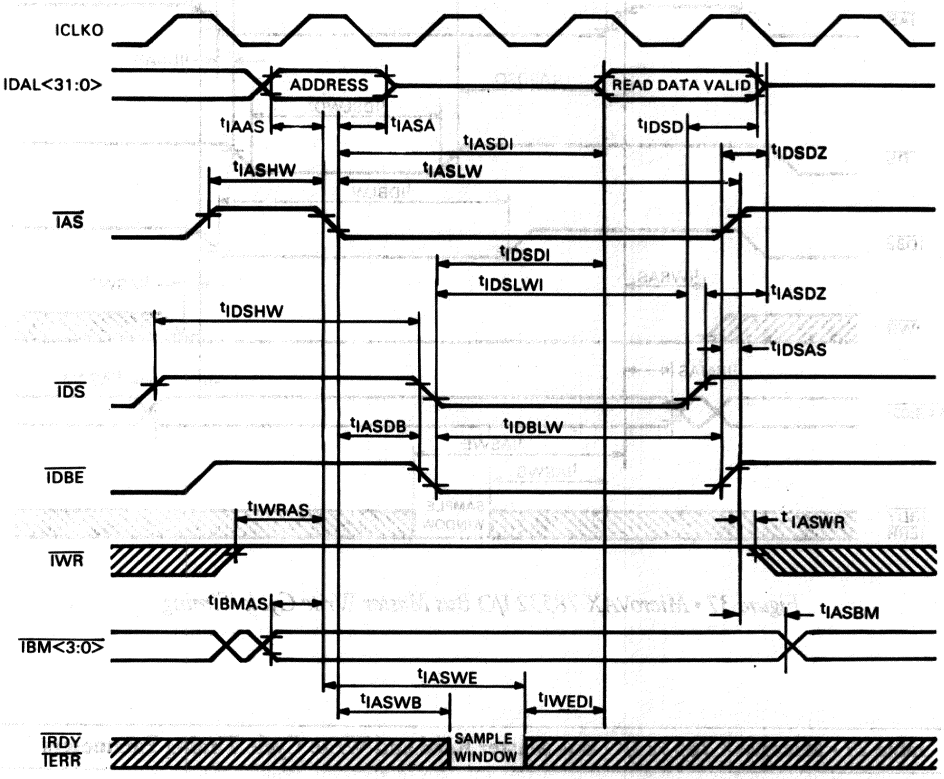


Figure 36 • MicroVAX 78532 I/O Bus Master Read Cycle Timing

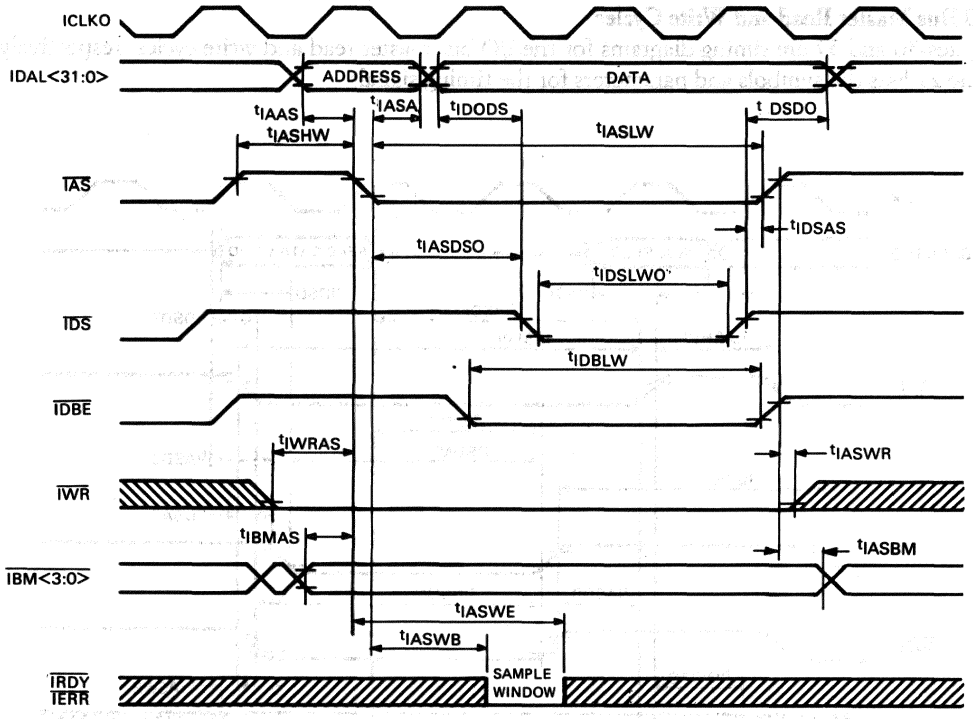


Figure 37 • MicroVAX 78532 I/O Bus Master Write Cycle Timing

Table 27 • MicroVAX 78532 I/O Bus Master Read and Write Cycle Timing Parameters

Symbol	Signal Definition	Requirements (ns) Min.	Max.
t_{IAAS}	IDAL<31:00> address setup time to $\overline{IAS}$ assertion	2P−28	—
t_{IASA}	IDAL<31:00> address hold time after $\overline{IAS}$ assertion	2P−15	—
t_{IASDB}	$\overline{IAS}$ assertion to $\overline{IDBE}$ and IDS (read) assertion	3P−15	3P+20
t_{IASDI}	$\overline{IAS}$ assertion to read data valid	—	11P−30+4PS
t_{IASDSO}	$\overline{IAS}$ assertion to $\overline{IDS}$ assertion (write)	5P−15	5P+20
t_{IASDZ}	$\overline{IAS}$ and $\overline{IDBE}$ deassertion to bus slave IDAL<31:00> three-state	—	2P−20
t_{IASHW}	$\overline{IAS}$ deassertion width	4P−25	—
t_{IASLW}	$\overline{IAS}$ assertion width	12P−15+4PS	—
t_{IASWB}	$\overline{IAS}$ assertion to beginning of $\overline{IERR}/\overline{IRDY}/\overline{IDMR}$ sample window	—	6P−45+4PS

Symbol	Signal Definition	Requirements (ns) Min.	Max.
t_{IASWE}	$\overline{IAS}$ assertion to end of $\overline{IRDY}/\overline{IERR}/\overline{IDMR}$ sample window	6P+10+4PS	—
t_{IASBM}	$\overline{IBM}$<3:0> hold time from $\overline{IAS}$ assertion	3P−20	—
t_{IASWR}	$\overline{IWR}$ hold time from $\overline{IAS}$ deassertion	P−20	—
t_{IBMAS}	$\overline{IBM}$<3:0> setup time before $\overline{IAS}$ assertion	P−25	—
t_{IDBLW}	$\overline{IDBE}$ assertion width	9P−20+4PS	—
t_{IDODS}	IDAL<31:00> write data setup time to $\overline{IDS}$ assertion	3P−30	—
t_{IDSAS}	$\overline{IDS}$ deassertion to $\overline{IAS}$ and $\overline{IDBE}$ deassertion	P−15	—
t_{IDSD}	IDAL<31:00> read data hold time after $\overline{IDS}$ deassertion	0	—
t_{IDSDI}	$\overline{IDS}$ assertion to IDAL<31:00> read data valid	—	8P−35+4PS
t_{IDSDO}	IDAL<31:00> write data hold time from $\overline{IDS}$ deassertion	3P−20	—
t_{IDSDZ}	$\overline{IDS}$ deassertion to bus slave IDAL<31:00> three-state on read bus cycles	—	3P−20
t_{IDSHW}	$\overline{IDS}$ deassertion width (read)	8P−50	—
t_{IDSLWI}	$\overline{IDS}$ assertion width (read)	8P−20+4PS	—
t_{IDSLWO}	$\overline{IDS}$ assertion width (write)	6P−20+4PS	—
t_{IWEDI}	$\overline{IRDY}$ internal sample window end to IDAL<31:00> read data valid	—	5P−25
t_{IWRAS}	$\overline{IWR}$ setup time before $\overline{IAS}$ assertion	3P−35	—

I/O Bus Slave Read and Write Cycles

Figures 38 and 39 are timing diagrams for the I/O bus slave read and write cycles, respectively. Table 28 lists the associated timing parameters.

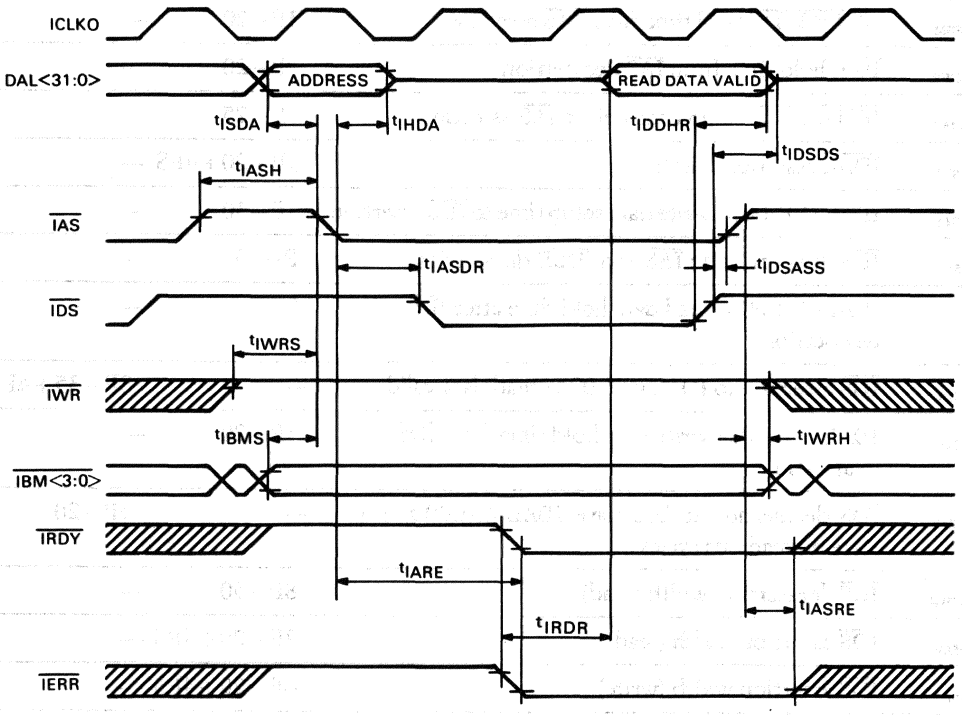


Figure 38 • MicroVAX 78532 I/O Bus Slave Read Cycle Timing

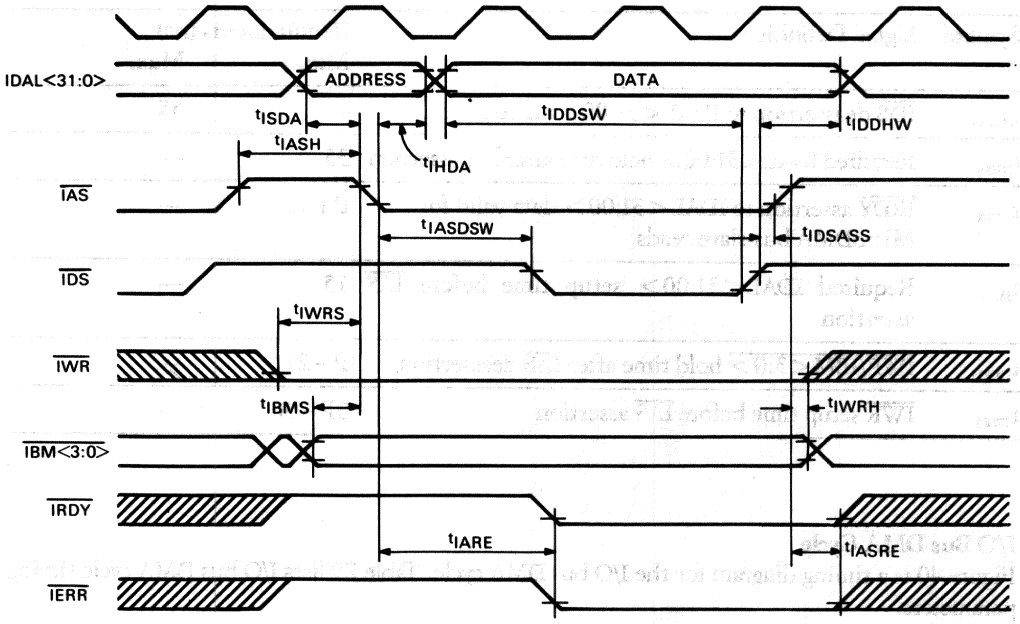


Figure 39 • MicroVAX 78532 I/O Bus Slave Write Cycle Timing

Table 28 • MicroVAX 78532 I/O Bus Slave Read and Write Cycle Timing Parameters

Symbol	Signal Definition	Requirements (ns) Min.	Max.
t_{IASDSR}	Required $\overline{IAS}$ assertion to $\overline{IDS}$ assertion delay (read cycles)	3P−20	3P+25
t_{IASDSW}	Required $\overline{IAS}$ assertion to $\overline{IDS}$ assertion delay (write cycles)	5P−20	5P+25
t_{IASH}	$\overline{IAS}$ deassertion width	2P+25	—
t_{IASRE}	$\overline{IAS}$ deassertion to $\overline{IRDY}$/$\overline{IERR}$ deassertion	—	100
t_{IBMS}	$\overline{IBM}$<3:0> setup time before $\overline{IAS}$ assertion	P−25	—
t_{IDDHR}	IDAL<31:00> data hold time after $\overline{IDS}$ deassertion (slave reads)	0	—
t_{IDDHW}	Required IDAL<31:00> hold time after $\overline{IDS}$ deassertion on MicroDMA bus-slave writes	35	—
t_{IDDSW}	Required IDAL<31:00> setup time before $\overline{IDS}$ deassertion on MicroDMA bus-slave writes	20	—
t_{IDSASS}	Required $\overline{IDS}$ deassertion to $\overline{IAS}$ deassertion delay	P−20	—

Symbol	Signal Definition	Requirements (ns) Min.	Max.
t_{IDSDS}	$\overline{IDS}$ deassertion to IDAL<31:00> three-state	—	55
t_{IHDA}	Required IDAL<31:00> hold time after $\overline{IAS}$ assertion	35	—
t_{IRDR}	$\overline{IRDY}$ assertion to IDAL<31:00> data valid for MicroDMA bus-slave reads	P+35	—
t_{ISDA}	Required IDAL<31:00> setup time before $\overline{IAS}$ assertion	15	—
t_{IWRH}	$\overline{IWR}$/$\overline{IBM}$<3:0> hold time after $\overline{IAS}$ deassertion	P−25	—
t_{IWRS}	$\overline{IWR}$ setup time before $\overline{IAS}$ assertion	3P−50	—

I/O Bus DMA Cycle

Figure 40 is a timing diagram for the I/O bus DMA cycle. Table 29 lists I/O bus DMA cycle timing parameters.

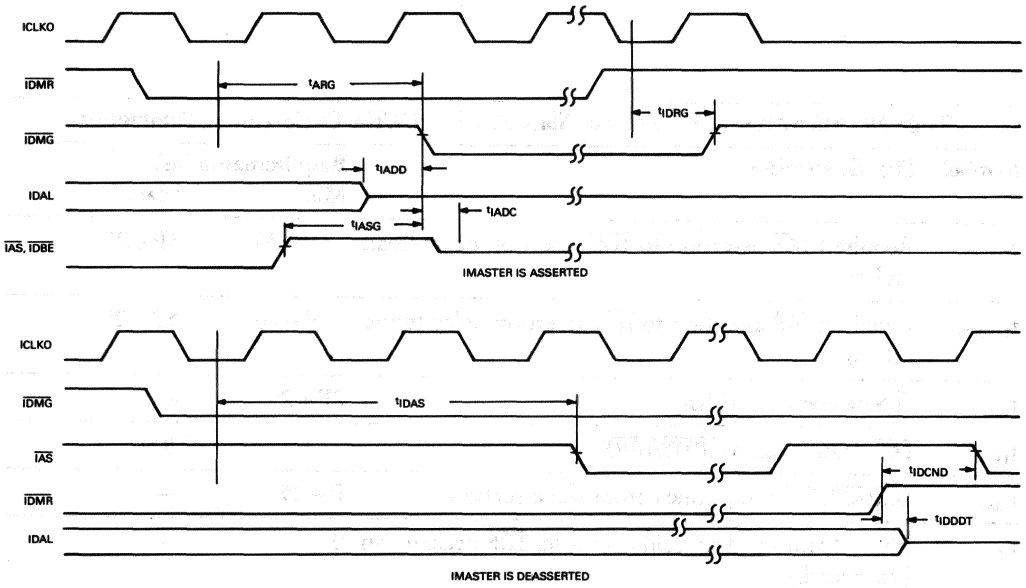


Figure 40 • MicroVAX 78532 I/O Bus DMA Cycle Timing

Table 29 • MicroVAX 78532 I/O Bus DMA Cycle Timing Parameters

Symbol	Signal Definition	Requirements (ns) Min.	Max.
t_{IADC}	Assert $\overline{IDMG}$ to $\overline{IAS}/\overline{IDS}/\overline{IWR}/\overline{IDBE}/\overline{IBM}<3:0>$ three-state	—	40
t_{IADD}	IDAL<31:00> three-state to assert $\overline{IDMG}$	—	2P+5
t_{IARG}	Asserted $\overline{IDMR}$ (internal) sample window end to $\overline{IDMG}$ assertion	6P	*
t_{IASG}	$\overline{IAS}$ and $\overline{IDBE}$ deassertion to $\overline{IDMG}$ assertion	4P−25	—
t_{IDAS}	Asserted $\overline{IDMG}$ (internal) sample window end to $\overline{IAS}$ assertion	—	10P+35+4PK†
t_{IDBM}	Asserted $\overline{IDMG}$ (internal) sample window end to $\overline{IBM}<3:0>$ assertion	—	9P+45+4PK†
t_{IDCND}	Deassert $\overline{IDMR}$ to $\overline{IAS}/\overline{IDS}/\overline{IWR}/\overline{IDBE}/\overline{IBM}<3:0>$ three-state	—	3P+35
t_{IDDDT}	Deassert $\overline{IDMR}$ to IDAL<31:00> three-state	—	P+30
t_{IDRG}	Deasserted $\overline{IDMR}$ (internal) sample window end to $\overline{IDMG}$ deassertion	—	2P+40

*Maximum value determine by latency specifications.
†K = The number of microcycles (0, 1, 2, 3, 4) that the sequencer is busy.

I/O Bus Transfer Request

Figure 41 shows the I/O bus transfer request signal timing and Table 30 list the timing parameters.

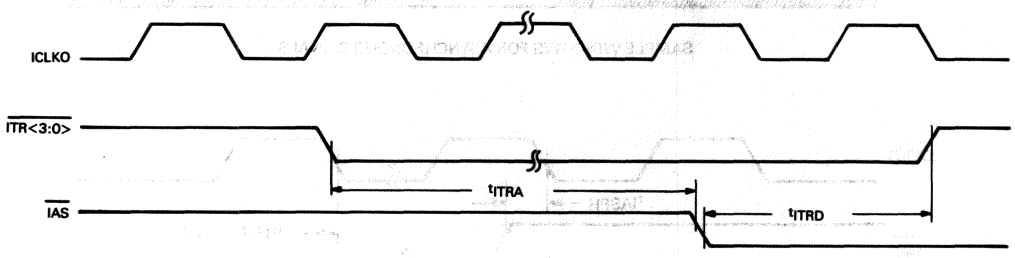


Figure 41 • MicroVAX 78532 I/O Bus Transfer Request Timing

Table 30 • MicroVAX 78532 I/O Bus Transfer Request Timing Parameters

Symbol	Signal Definition	Requirements (ns) Min.	Max.
t_{ITRA}	$\overline{ITR}$<3:0> assertion to $\overline{IAS}$ assertion for requesting channel	—	25P+30+4PK*
t_{ITRD}	$\overline{IAS}$ assertion to $\overline{ITR}$<3:0> deassertion to assure present requested bus cycle is last of ITR requested bus cycles (MicroDMA is bus master)	—	6P–35+4PS

*K = The number of microcycles (0, 1, 2, 3, 4) that the sequencer is busy.

I/O Bus General Signal Timing

Figure 42 shows the general timing for the I/O bus signals. Table 31 lists I/O bus general signal timing parameters which include sample windows times for the asynchronous signals.

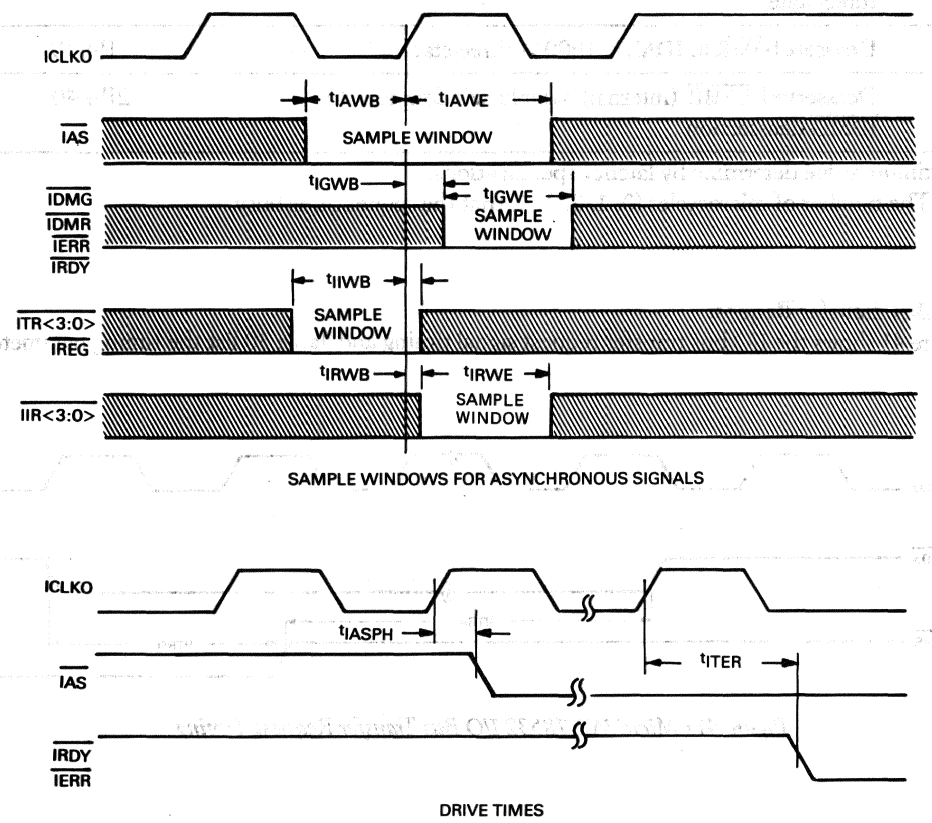


Figure 42 • MicroVAX 78532 I/O Bus General Signal Timing

Table 31 • MicroVAX 78532 I/O Bus General Timing Parameters

Symbol	Signal Definition	Requirements (ns) Min.	Max.
t_{IASPH}	ICLKO to $\overline{IAS}$ asserted	P−12	P+25
t_{IAWB}	ICLKO to beginning of $\overline{IAS}$ sample window	—	−35
t_{IAWE}	ICLKO to end of $\overline{IAS}$ sample window	3P+5	—
t_{IGWB}	ICLKO to $\overline{IDMG}$/$\overline{IDMR}$/$\overline{IERR}$/$\overline{IRDY}$ sample window beginning	—	3P−50
t_{IGWE}	ICLKO to $\overline{IDMG}$/$\overline{IDMR}$/$\overline{IERR}$/ $\overline{IRDY}$ sample window end	3P+5	—
t_{IIWB}	ICLKO to beginning of $\overline{ITR<3:0>}$/$\overline{IREG}$ sample window	—	−50
t_{IIWE}	ICLKO to end of $\overline{ITR<3:0>}$/$\overline{IREG}$ sample window	5	—
t_{IRWB}	ICLKO to beginning of $\overline{IIR<3:0>}$ sample window	—	P−50
t_{IRWE}	ICLKO to end of $\overline{IIR<3:0>}$ sample window	P+5	—
t_{ITER}	ICLKO to $\overline{IERR}$/$\overline{IRDY}$ assertion	3P−5	3P+26

• Interfacing Requirements

MicroDMA interface designs vary depending on the type of peripherals being interfaced. Figure 43 is a simplified example of a typical interface application. The MicroDMA is used as an interface between the MicroVAX CPU and an 8-bit peripheral chip similar to an Intel* device.
*Intel is a trademark of Intel Corporation.

An address latch and decoder enables the MicroDMA and other devices on the MicroVAX bus. If more than one device on the MicroVAX bus can respond to a DMA or an interrupt acknowledge in cycle, the devices are connected as a daisychain as shown. The peripheral chip has separate read and write controls. The assertion of the $\overline{IWR}$ and $\overline{IDS}$ lines indicates that a write operation is required. The assertion of the $\overline{IDS}$ signal without the $\overline{IWR}$ being asserted indicates that a read operation is required. The peripheral interface includes buffers for the IDAL data and addresses, and a decoder for asserting signals such as $\overline{DACK}$ (a DMA data transfer acknowledgment signal), $\overline{CS}$ (a peripheral chip select signal), and peripheral chip register addressing signals. The $\overline{IMASTER}$ signal is asserted (MicroDMA is the default master of the I/O bus) and the $\overline{IDMR}$ signal is deasserted so that another device cannot request control of the I/O bus.

Timing for the interface is from a common 40-MHz clock. The timing logic used depends on the type of peripheral chip(s) being interfaced and determines when an I/O bus cycle can be terminated and when $\overline{IRDY}$ line should be asserted. All the chips are reset by common reset circuit.

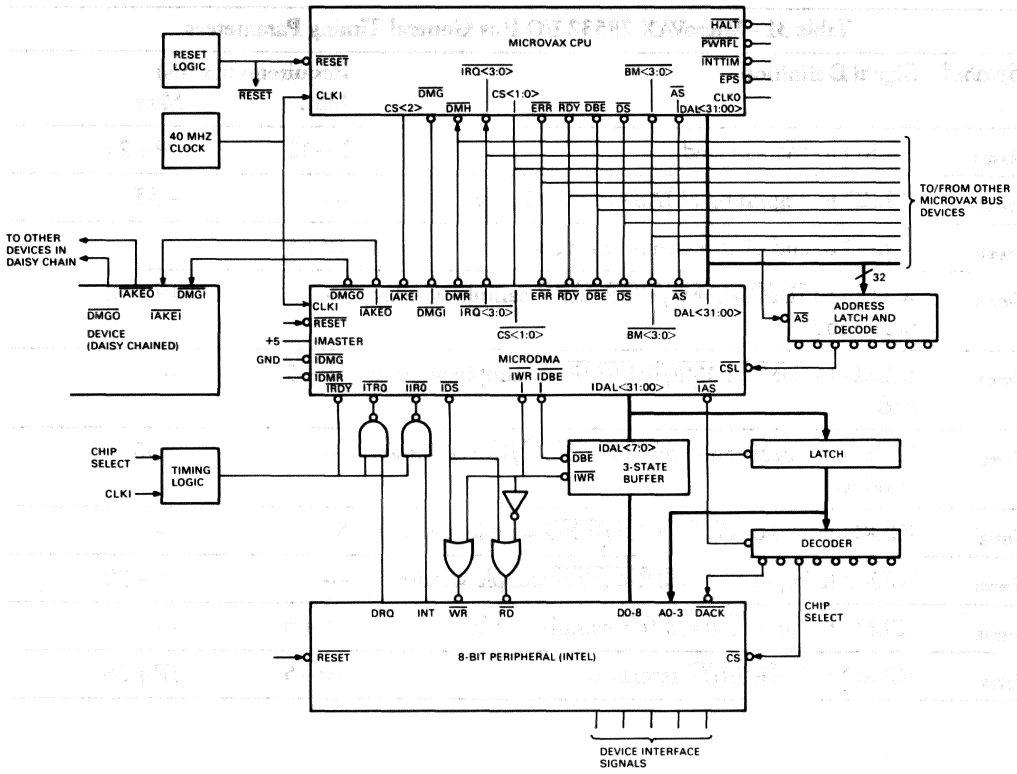


Figure 43 • MicroVAX 78532 Typical MicroDMA Interfacing

• Features

- Compatible with the MicroVAX 78032 CPU
- 32-bit memory data organization
- Controls operation of 4 Mbytes of 256K by 1-bit dynamic RAMs
- Generates multiplexed address, RAS and CAS signals for as many as four banks of memory
- Two access speeds for use with different-speed dynamic RAMs
- Four refresh modes
- Supports battery backup refresh
- Supports parity error reporting with address capture
- Bus timeout error detection and reporting
- Generates 100-Hz interval clock
- Double-metal CMOS technology
- Single 5-Vdc power supply
- Minimum parts count memory interface

• Description

The MicroVAX 78584 Dynamic Ram Controller (DYRC) provides a low cost interface between the MicroVAX 78032 CPU and 4 Mbytes of dynamic RAM (DRAM). The DYRC supports 256K by 1-bit DRAMs and supplies multiplexed address, timing strobes, and refresh/access arbitration control. Two operating speeds allow the designer to use different speed DRAMs. The choice of speed determines whether memory errors are reported during the same cycle or a following cycle. Error address capture logic is implemented in the DYRC to aid in the reporting of memory errors. The DYRC also provides battery backup refresh support, a 100-Hz interval timer, and bus timeout logic to report nonexistent addresses or no response to the address strobe. Figure 1 is a block diagram of the MicroVAX 78584 DYRC.

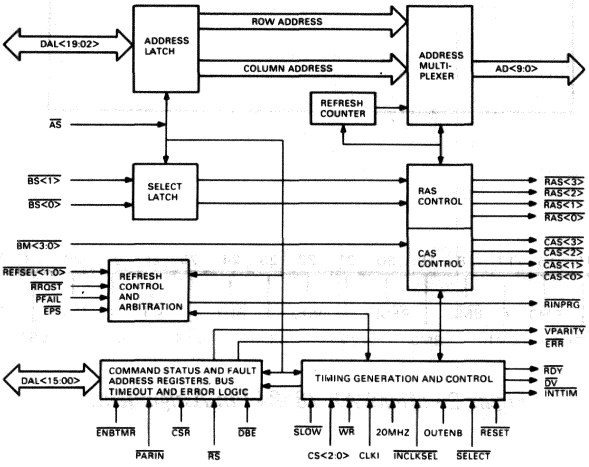


Figure 1 • MicroVAX 78584 DYRC Block Diagram

Using the DYRC results in a minimum part count 32-bit DRAM memory interface that requires a single 5-Vdc supply and is compatible with the MicroVAX 78032 CPU.

▪ Pin and Signal Description

This section provides a description of the input and output signals and power and ground connections used by the MicroVAX 78584 DYRC. The signal pin assignments are shown in Figure 2 and summarized in Table 1.

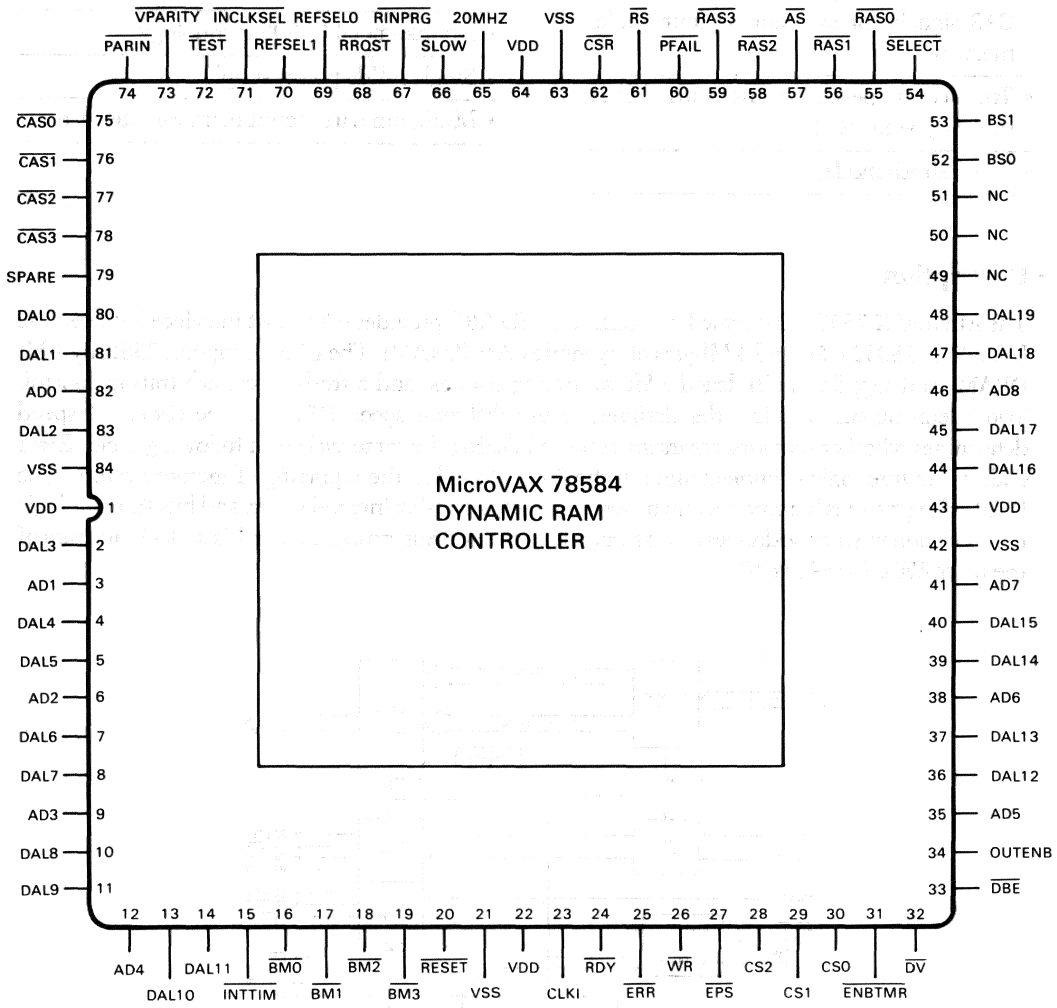


Figure 2 ▪ MicroVAX 78584 Pin Assignments

Table 1 • MicroVAX 78584 Pin and Signal Summary

Pin	Signal	Input/Output	Function/Definition
48,47,45,44, 40,39,37,36, 14,13,11,10, 8,7,5,4,2,83	DAL<19:00>	input/output	Data address lines <19:00>—During the address portion of a memory cycle, the address on DAL<19:02> is used to form the row and column addresses. The DAL<15:00> lines are used for the transfer of information to and from the command status and fault address registers.
66	$\overline{\text{SLOW}}$	input	Slow—Matches the operating speed of the DYRC with the speed of slower memory chips.
45	$\overline{\text{SELECT}}$	input	Select—Selects the chip for a memory access cycle.
61	$\overline{\text{RS}}$	input	Register select—Selects access to the two internal registers.
62	$\overline{\text{CSR}}$	input	Command status register select—Selects which of the two internal registers is to be accessed.
57	$\overline{\text{AS}}$	input	Address strobe—A strobe from the CPU that latches address and control information into the DYRC and starts a RAM access cycle if the $\overline{\text{SELECT}}$ signal is asserted or an internal register access if the $\overline{\text{RS}}$ signal is asserted.
28-30	CS<2:0>	input	Control status—Determines the type of bus cycle to be performed.
19-16	$\overline{\text{BM<3:0>}}$	input	Byte masks—Selects the byte(s) to be accessed.
53,52	BS<1:0>	input	Bank select—Selects the bank of memory to be accessed.
46,41,38,35 12,9,6,3,82	AD<8:0>	output	Address <8:0>—Provides the multiplexed memory address to the RAM array.
59,58,56,55	$\overline{\text{RAS<3:0>}}$	output	Row address strobe—Strobe signals used to latch the row address into the memory bank selected by BS<1:0>.
78-75	$\overline{\text{CAS<3:0>}}$	output	Column address strobe—Strobe signals used to latch the column address into the byte(s) of the memory array selected by BM<3:0>.
24	$\overline{\text{RDY}}$	output	Ready—Synchronizes the data transfers.
34	OUTENB	input	Output enable—Enables the DYRC outputs.

Pin	Signal	Input/Output	Function/Definition
26	$\overline{WR}$	input	Write—Indicates the direction of data transfer on the DAL<19:00> lines.
33	$\overline{DBE}$	input	Data buffer enable—Enables the three-state DAL<19:00> lines drivers during an internal register read.
70,69	REFSEL<1:0>	input	Refresh select—Select one of four refresh modes.
27	$\overline{EPS}$	input	External processor strobe—Provides a refresh request synchronization for processors that use external processor cycles.
68	$\overline{RRQST}$	input	Refresh request—Used by external logic to request a refresh cycle.
67	$\overline{RINPRG}$	output	Refresh in progress—Indicates that a refresh cycle is in progress.
31	$\overline{ENBTMR}$	input	Enable timeout timer—Enables the bus timeout function.
74	$\overline{PARIN}$	input	Parity in—Used by external logic to report a parity error to the DYRC.
25	$\overline{ERR}$	output	Error—Indicates a parity error or bus timeout condition to the CPU.
73	$\overline{VPARITY}$	output	Valid parity—For use by diagnostics to verify the operation of parity logic by forcing it to write a wrong parity.
32	$\overline{DV}$	output	Data valid—Indicates that the data being written to or read from memory is valid.
23	CLKI	input	Clock input—A clock input that provides timing for the DYRC and synchronization with the CPU.
65	20MHZ	input	20 MHz—An optional clock input for generating 100-Hz internal timing and bus timeout timing.
71	$\overline{INCLKSEL}$	input	Internal clock select—Selects the clock source to be used for generating internal timing and bus timeout timing.
15	$\overline{INTTIM}$	output	Internal timer—A 100-Hz clock that can be used to support operating system timing functions.

Pin	Signal	Input/Output	Function/Definition
60	$\overline{PFAIL}$	input	Powerfail—Continues memory refresh operation during a powerfail condition when refresh mode 0 or 1 is selected and memory has battery backup.
20	$\overline{RESET}$	input	Reset—Sets the DYRC to a known initial state.
1,22,43,64	V_{DD}	input	Voltage—Power supply voltage.
21,42,63,84	V_{SS}	input	Ground—Ground reference.
72	$\overline{TEST}$	input	Test—Reserved for manufacturing use.

Input and Output Signals

Data address lines (DAL<19:00>)—These lines are used to form the multiplexed address for the 256K dynamic memory chips and to transfer information between the CPU and the two internal registers. During a memory read or write cycle, lines DAL<19:02> are latched into the DYRC by the assertion of the $\overline{AS}$ input. The multiplexed row and column address is formed from this information. Lines DAL<19, 17, 15, 13, 11, 09, 07, 05, 03> are used to form the 9-bit row address and lines DAL<18, 16, 14, 12, 10, 08, 06, 04, 02> are used to form the 9-bit column address. The DAL<15:00> lines are used to transfer information between the two internal registers and the CPU. Access to the internal registers is controlled by the $\overline{RS}$ and $\overline{CSR}$ inputs.

Input Signals

Select ($\overline{SELECT}$)—This signal, when asserted by external address decode logic, enables a memory access cycle when the $\overline{AS}$ signal is asserted. The $\overline{CAS}$<3:0> lines and the $\overline{RDY}$ and $\overline{DV}$ signals are enabled when this input is asserted.

Register select ($\overline{RS}$)—This signal, when asserted by external address decode logic, enables a read or write access to the internal registers of the DYRC. The two internal registers can be positioned in the I/O page or mapped into memory space by the designer. The $\overline{RS}$ and $\overline{SELECT}$ inputs must be mutually exclusive.

Output enable (OUTENB)—This signal is asserted to enable the following outputs: AD<8:0>, $\overline{RAS}$<3:0>, $\overline{CAS}$<3:0>, DAL<15:00>, $\overline{RDY}$, $\overline{DV}$, $\overline{RINPRG}$, $\overline{ERR}$, $\overline{INTTIM}$, and $\overline{VPARITY}$. If the OUTENB input is not asserted, these outputs are high-impedance.

Address strobe ($\overline{AS}$)—This signal, when asserted, latches the DAL<19:02> and BS<1:0> line information into the DYRC. The assertion of the $\overline{AS}$ input starts a memory access cycle if the $\overline{SELECT}$ input is asserted or an internal register access cycle if the $\overline{RS}$ input is asserted. The $\overline{AS}$ input is also used to internally synchronize the refresh logic.

External processor strobe (EPS)—This signal and the $\overline{AS}$ and $\overline{SELECT}$ signals are used to internally synchronize the refresh logic.

Control status lines (CS<2:0>)—These signals are decoded by the DYRC with the $\overline{WR}$ input to monitor the type of bus cycle being performed. Table 2 lists the bus cycle assignments and indicates if a memory access is allowed for the cycle selected.

Table 2 • MicroVAX 78584 Bus Cycle Assignments

$\overline{WR}$	CS Line* 2	1	0	Bus Cycle Type	Memory Access
H	L	L	L	reserved	No
H	L	L	H	reserved	No
H	L	H	L	reserved	No
H	L	H	H	interrupt acknowledge	No
H	H	L	L	read (instruction)	Yes
H	H	L	H	read lock	Yes
H	H	H	L	read (data, modify intent)	Yes
H	H	H	H	read (data, no modify intent)	Yes
L	L	L	L	reserved	No
L	L	L	H	reserved	No
L	L	H	L	reserved	No
L	L	H	H	reserved	No
L	H	L	L	reserved	No
L	H	L	H	write unlock	Yes
L	H	H	L	reserved	No
L	H	H	H	write (data)	Yes

*H = high level, L = low level.

Byte masks ($\overline{BM}$<3:0>)—These signals are used to generate the information on the $\overline{CAS}$<3:0> outputs. During a memory read or write cycle, the byte mask $\overline{BM}$<3:0> lines that are asserted result in the corresponding $\overline{CAS}$<3:0> line being asserted.

Bank select (BS<1:0>)—These signals select one of the four banks of memory for access by selecting the $\overline{RAS}$ line to be asserted as described in Table 3.

Table 3 • MicroVAX 78584 Bank Select Decoding

BS Line* 1	0	RAS Line
L	L	$\overline{RAS}$<0>
L	H	$\overline{RAS}$<1>
H	L	$\overline{RAS}$<2>
H	H	$\overline{RAS}$<3>

*H = high level, L = low level.

Control status register ($\overline{CSR}$)—This signal is used with the $\overline{RS}$ input to select the internal register to be accessed. When the $\overline{CSR}$ input is asserted, the control status register is selected. When the $\overline{CSR}$ input is not asserted, the fault address register is selected.

Write ($\overline{WR}$)—This signal is used by the DYRC to detect a read or a write bus cycle.

Data buffer enable ($\overline{DBE}$)—This signal is used with the $\overline{RS}$ signal to enable the three-state DAL<15:00> drivers when one of the internal registers is being read.

Refresh select (REFSEL<1:0>)—These lines are used to select one of the four refresh modes. The DYRC selects the refresh mode when the $\overline{RESET}$ input is asserted. These pins should be connected to the proper voltage level. Table 4 lists the refresh mode selections.

Table 4 • MicroVAX 78584 Refresh Mode Selections

REFSEL Line 1	0	Mode
0	0	0
0	1	1
1	0	2
1	1	3

Refresh request ($\overline{RRQST}$)—This signal is asserted by external logic to request a memory refresh cycle when refresh mode 2 is selected. If the DYRC is operating in refresh mode 0, 1, or 3 and this signal is asserted, an extra refresh cycle will be performed. However, this does not have any affect on the refresh interval because the internal counters of the DYRC are not changed.

Slow ($\overline{SLOW}$)—This signal is used to match the operating speed of the DYRC with the operating speed of the DRAMs being used. When asserted, the $\overline{RAS}$ pulse width is increased and the $\overline{CAS}$<3:0> and $\overline{RDY}$ signals are delayed, allowing the DYRC to be used with slower DRAMs. This pin should be connected to V_{DD} or V_{SS}.

Enable timeout timer ($\overline{ENBTMR}$)—When asserted, this signal enables the bus timeout timer. When enabled and the $\overline{AS}$ input has been asserted for 25 µs, the DYRC will assert the $\overline{ERR}$ output to notify the CPU of an error. The CPU is then required to examine the CSR to determine the cause of the error. This pin should be connected to V_{DD} or V_{SS}.

Parity in ($\overline{PARIN}$)—This signal is asserted by external parity checking logic when a parity error has occurred. If this signal is not used, it should be connected to V_{DD} through an external pullup resistor.

Clock input (CLKI)—This is the input clock that provides the basic timing reference for the DYRC.

20 MHz (20MHZ)—This clock is used to generate the 100-Hz $\overline{INTTIM}$ output and the 25 µs bus timeout timer if selected by the $\overline{INCLKSEL}$ input.

Internal clock select ($\overline{INCLKSEL}$)—This signal selects the clock source to be used to generate the 100-Hz $\overline{INTTIM}$ output and 25-µs bus timeout timer. When the $\overline{INCLKSEL}$ signal is asserted, the CLKI divided by two is selected as the clock source. When the $\overline{INCLKSEL}$ signal is deasserted, the 20MHZ input is selected as the clock source. This pin should be connected to V_{DD} or V_{SS}.

Powerfail ($\overline{PFAIL}$)—This signal is asserted by external logic to notify the DYRC of a system power failure and that the DYRC is to continue refreshing memory from a backup power source. This signal is functional in refresh modes 0 or 1 only. When the DYRC is used without a backup power source, the $\overline{PFAIL}$ input must be pulled up by an external resistor.

Reset (RESET)—This line is asserted to set the DYRC to a known state.

Voltage (V_{DD})—Connects to the power supply voltage.

Ground (V_{SS})—Connects to the ground reference.

Test (TEST)—Reserved for manufacturing use. This pin must be connected to V_{DD}.

Output Signals

Address ($AD<8:0>$)—These are the multiplexed memory address outputs. When the selected $\overline{RAS<3:0>}$ output is asserted, these lines contain the row address for the DRAMs. When the selected $CAS<3:0>$ output or outputs are asserted, lines $\overline{AD<8:0>}$ contain the column address for the DRAMs. These outputs cannot drive the DRAMs directly. Therefore a memory driver is required between these lines and the memory array. Each line is capable of driving up to four memory driver inputs.

Row address strobe ($\overline{RAS<3:0>}$)—These signals are used to latch the row address into the selected bank of memory. The $\overline{RAS<3:0>}$ line or lines to be asserted are selected by the $\overline{BS<1:0>}$ inputs or by the refresh logic. The $\overline{RAZ<3:0>}$ outputs cannot directly drive the DRAMs. A memory driver is required between these lines and the memory array. Each line is capable of driving up to four memory driver inputs.

Column address strobe ($\overline{CAS<3:0>}$)—These signals are used to latch the column address into the selected byte(s) of the memory array. The lines to be asserted are selected by the $\overline{BM<3:0>}$ inputs. These outputs cannot drive the DRAMs directly. Therefore a memory driver is required between these lines and the memory array. Each line is capable of driving up to four memory driver inputs.

Ready ($\overline{RDY}$)—This signal is asserted to notify the controlling processor that the current memory access bus cycle or internal register data transfer bus cycle can be completed.

Data valid ($\overline{DV}$)—This signal is asserted by the DYRC to notify the external error detection and correction logic that the data being read from or written to memory is stable.

Error ($\overline{ERR}$)—This signal is asserted by the DYRC to notify the CPU that the parity checking logic has reported a parity error to the DYRC by asserting the $\overline{PARIN}$ signal or that a bus timeout condition has occurred. When there is more than one DYRC in a system, only one should report a bus timeout.

Refresh in progress ($\overline{RINPRG}$)—This signal is asserted to notify the external logic that a refresh cycle is in progress.

Interval timer ($\overline{INTTIM}$)—A 100-Hz timer for use by the operating system.

Valid parity (VPARITY)—This signal can be used by diagnostics to verify the operation of the external parity checking logic. This signal is controlled by the write wrong parity (WWP) bit in the command status register. When $\overline{VPARITY}$ is asserted, the parity logic should function normally. When deasserted, the parity should be inverted.

Functional Description

This section describes the basic operation and organization of the MicroVAX 78584 DYRC.

Memory Organization

The DYRC supports a 32-bit, byte oriented, memory data organization. Because of the byte orientation of the memory array, parity checking should be performed on each byte. This results in a 36-bit-wide memory array with eight data bits per byte plus one parity bit. The DYRC supports up to four banks of memory. Each bank contains 1 Mbyte when 256K dynamic RAMs are used for a maximum memory array of 4 Mbytes per DYRC.

Read/Write Operation

For a read/write operation, the DYRC provides the multiplexed memory address, row address strobe, and column address strobe to the memory array. It also generates the $\overline{RDY}$ signal required for bus cycle termination. The timing information for a read or write cycle is in the Specification section.

A memory read or write cycle is initiated by the assertion of the $\overline{SELECT}$ and $\overline{AS}$ signals. If a refresh request is not pending or in progress, the DYRC transfers the row address on the memory address bus and assert the $\overline{RAS<3:0>}$ output as selected by BS<1:0> lines. After the specified row address hold time, the DYRC transfers the column address onto the memory address bus and asserts the $\overline{CAS<3:0>}$ outputs as selected by the BM<3:0> lines. The DYRC asserts the $\overline{RDY}$ signal to notify the CPU that the current bus cycle can be completed. The assertion of the $\overline{RDY}$ line is determined by the type of memory cycle (slow or fast) being performed. The data on the DAL<31:0> lines is the data to be written into or read from the accessed DRAM chips. The $\overline{DV}$ signal is asserted to notify the external logic, such as parity or EDAC logic, that the data is valid. The $\overline{AS}$ signal is deasserted and the memory access is completed. The DYRC uses the "early write" mode of the DRAM for writing data into memory.

If a memory refresh is pending or in progress when the DYRC is selected and the $\overline{AS}$ signal is asserted, the memory access will be delayed until the refresh cycle is completed. The MicroVAX bus cycle is stretched by the delay of the assertion of the $\overline{RDY}$ signal from the DYRC.

Refresh Operation

The $\overline{AS}$, $\overline{SELECT}$, $\overline{EPS}$, and $\overline{PFAIL}$ inputs are used by the DYRC to arbitrate a refresh cycle. The DYRC performs the arbitration and control for the refresh cycles that may be started by the following:

- Detecting the deassertion of the $\overline{AS}$ signal when the $\overline{SELECT}$ signal is asserted. The DYRC knows when the next assertion of the $\overline{AS}$ signal will occur. This allows time for the DYRC to arbitrate between a refresh cycle and the next memory cycle.
- Detecting the deassertion of $\overline{EPS}$. The DYRC will perform the necessary refresh cycles during the execution of long floating-point instructions.
- Detecting the assertion of the $\overline{AS}$ signal when the $\overline{SELECT}$ or $\overline{EPS}$ are deasserted for an extended period of time (62.5 µs maximum). Four consecutive rows will be refreshed during the refresh cycle. This allows refresh cycles to be performed while the CPU is communicating with slow peripheral devices.

- At the assertion of the $\overline{PFAIL}$ signal. This condition inhibits memory accesses other than refresh cycles. When the $\overline{PFAIL}$ signal is asserted, the automatic refresh occurs only when refresh mode 0 or 1 is selected and there is a backup power source.

A refresh cycle consists of transferring the refresh address onto the memory address bus, asserting the $\overline{RAS<3:0>}$ line information, and incrementing the refresh row address counter by 1 until four rows have been refreshed. While a refresh cycle is in progress, the DYRC asserts the $\overline{RINPRG}$ signal. The refresh cycle is completed when the $\overline{RAS<3:0>}$ lines and the $\overline{RINPRG}$ signal are deasserted. Any memory accesses attempted during a refresh operation are deferred until the refresh is completed.

DMA devices that access memory controlled by the DYRC must consider the latency time that may occur as a result of a refresh cycle. The DMA device can use $\overline{RINPRG}$ to detect a refresh cycle in progress.

The four refresh modes that are selected by the REFSEL<1:0> lines are listed in Table 2. The refresh mode to be used is selected when the $\overline{RESET}$ line is asserted.

Mode 0—In this mode refresh operation is automatically controlled by the DYRC. The DYRC will refresh 256 consecutive locations in 4 ms when the clock input is 40 MHz. During the powerup sequence, the memory array is initialized with eight refresh cycles before any access is permitted. The $\overline{RINPRG}$ output will be asserted during refresh cycles.

Mode 1—In this mode, refresh operation is automatically controlled by the DYRC. The DYRC will refresh 512 consecutive locations in 4 ms when the clock input is 40 MHz or 256 consecutive locations in 4 ms when it is 20 MHz. During powerup, the memory array is initialized with eight refresh cycles before any access is permitted. The $\overline{RINPRG}$ output will be asserted during refresh cycles.

Mode 2—In this mode, refresh operation is controlled by external logic. The external logic requests a refresh cycle by asserting the $\overline{RRQST}$ input. The DYRC arbitrates the refresh cycle, asserts the $\overline{RINPRG}$ output, performs the four refresh cycles, and increments the refresh counter. The $\overline{RINPRG}$ output can be used to clear the $\overline{RRQST}$ signal. During the powerup sequence, the memory array is initialized with eight refresh cycles before any access is permitted. Automatic refresh is not performed after powerup.

Mode 3—In this mode the refresh operation is disabled and no refresh will occur during the powerup sequence.

Internal Registers

The DYRC contains two 16-bit registers. The control and status register (CSR) is a read/write register that is used to transfer control and status information between the processor and the DYRC. The fault address register (FAR) is a read-only register that is used to store the address of the page in memory being accessed at the time a parity error is reported. Access to the CSR and FAR is controlled by the $\overline{RS}$, $\overline{CSR}$, and $\overline{WR}$ inputs. The $\overline{RS}$ input selects the DYRC for a register access, the $\overline{CSR}$ input selects the register to be accessed, and the $\overline{WR}$ input determines whether a read or write transaction is to be performed. The addresses for the CSR and FAR registers must be on a longword boundary. Because these registers are 16 bits wide, they must be accessed using word instructions.

Control Status Register—The control status register (CSR) enables parity error support, reports parity and bus timeout status, and forces a wrong parity for diagnostic purposes. During the initial powerup sequence or at the assertion of the $\overline{RESET}$ input, this register is cleared. Figure 3 shows the CSR register format and Table 5 describes the function of each bit.

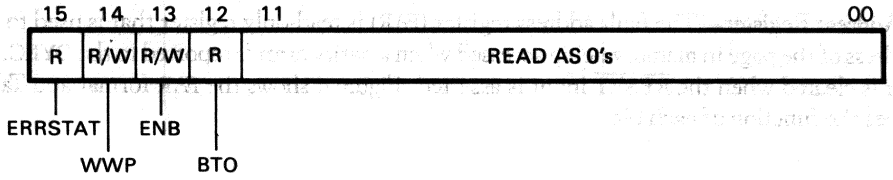


Figure 3 • MicroVAX 87584 Control Status Register Format

Table 5 • MicroVAX 87584 Control Status Register Description

Bit	Description
15	ERRSTAT (Error status)—This bit is used to report a parity error and is set when the ENB (bit 13) is set and the $\overline{\text{PARIN}}$ input is asserted. When set, this bit indicates that a parity error has been detected by the external parity logic. This bit is cleared when the CSR is read or the $\overline{\text{RESET}}$ input is asserted.
14	WWP (Write wrong parity)—This bit is set and cleared by software. When set, it causes the $\overline{\text{VPARITY}}$ output to be deasserted. When cleared, the $\overline{\text{VPARITY}}$ output is asserted. Can be used during diagnostics operations to verify the operation of the external parity logic by forcing wrong parity. Cleared when the $\overline{\text{RESET}}$ input is asserted.
13	ENB (Enable)—This bit is used to enable the reporting of parity errors: set to enable the parity error reporting function of the DYRC and cleared to disable the parity error reporting function including the ERRSTAT flag. Cleared when the $\overline{\text{RESET}}$ input is asserted. When reporting a parity error to the CPU, the DYRC disables the parity error reporting function by clearing this bit. This is done to keep multiple parity errors from corrupting the fault address in the FAR. After handling a parity, error software must reenable parity error reporting by setting this bit.
12	BTO (Bus timeout)—When set, this bit indicates that the bus has timed out. This bit is set enabled when the $\overline{\text{ENBTMR}}$ input is asserted. This bit is cleared when the CSR is read or the $\overline{\text{RESET}}$ input is asserted.
11:00	RAZ (Read as zeros)—Not used.

Fault Address Register—The fault address register (FAR) is read-only register that is used to store the address of the page in memory being accessed when a parity error is reported to the DYRC. This register is cleared when the $\overline{\text{RESET}}$ input is asserted. Figure 4 shows the FAR format and Table 6 describes the function of each bit.

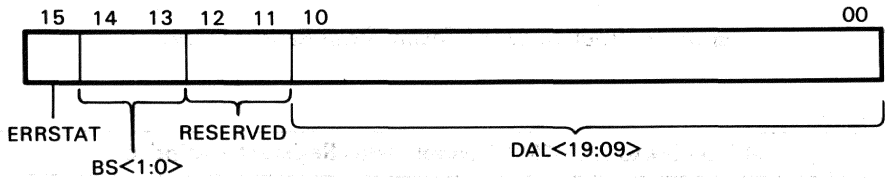


Figure 4 • MicroVAX 78584 Fault Address Register Format

Table 6 • MicroVAX 78584 Fault Address Register Description

Bit	Description
15	ERRSTAT (Parity error status)—This bit is set when external parity logic detects a parity error and asserts the $\overline{\text{PARIN}}$ input. When set, register contents cannot be changed. Cleared by a processor read transaction or when the $\overline{\text{RESET}}$ input is asserted.
14:13	BS<1:0> (Bank select)—These bits contain the value of the BS<1:0> information at the time the $\overline{\text{PARIN}}$ input was asserted. This value can be used to determine the bank of memory with the parity error. Cleared when the $\overline{\text{RESET}}$ input is asserted.
12:11	RESERVED (Reserved)—These bits are cleared at powerup and set after the first memory operation. They remain set until the $\overline{\text{RESET}}$ input is asserted.
10:00	DAL<19:09> (Data/Address<19:09>)—These bits contain the address of the page in memory being accessed at the time the $\overline{\text{PARIN}}$ pin was asserted. Cleared when the $\overline{\text{RESET}}$ input is asserted.

Error Reporting

The DYRC reports memory parity errors, bus timeout, and nonexistent memory address errors to the CPU. For a memory parity error, the DYRC provides the error reporting interface between external parity checking logic and the CPU. The bus timeout logic monitors the MicroVAX bus activity and reports a timeout error when the addressed device does not respond by asserting the $\overline{\text{RDY}}$ signal. When an error has been reported to the CPU by the DYRC, the error handling routine must read the CSR to determine the type of error being reported.

Parity Error Reporting—The DYRC provides the interface between external parity checking logic and the MicroVAX 78032 CPU for reporting a parity error. Parity error reporting is enabled by setting the ENB bit in the CSR. When a parity error has been detected by external parity checking logic, it asserts the $\overline{\text{PARIN}}$ input of the DYRC. This causes the page address to be captured in the FAR, the ERRSTAT bit in the CSR to be set, the parity error logic to be disabled by the clearing of the ENB bit in the CSR, and the CPU to be notified of an error by asserting the $\overline{\text{ERR}}$ output. The

DYRC will hold the $\overline{ERR}$ signal asserted until the next data stream access to memory. This ensures that the CPU will detect the error condition and respond. After responding to the parity error, the software must reenable the parity error logic by setting the ENB bit in the CSR. Refer to the *MicroVAX 78032 Central Processing Unit User's Guide* for information on error handling. The parity error reporting logic also aids parity and error detection and correction (EDAC) designs by providing a data valid strobe. The $\overline{DV}$ strobe can be used by external parity or EDAC logic as an indicator that the data on the bus is valid.

Bus timeout error—The bus timeout logic provides a means to monitor the MicroVAX bus activity and to notify the CPU of a nonexistent memory error or some other error that causes $\overline{AS}$ to be asserted for more than 25 µs. Proper operation of this logic requires a 40-MHz clock input with the $\overline{INCLKSEL}$ input asserted or a 20-MHz input with the $\overline{INCLKSEL}$ input deasserted. The bus timeout logic is enabled or disabled by connecting the $\overline{ENBTMR}$ input to V_{DD} or V_{SS}. When the bus timeout logic is enabled and the $\overline{AS}$ input has been asserted for more than 25 µs, the DYRC will set the BTO bit in the CSR and assert the $\overline{ERR}$ signal.

Interval Timer

The interval timer provides a 100-Hz output ($\overline{INTTIM}$) that can be used to support operating system timing functions. The clock source for this output is selected by the $\overline{INCLKSEL}$ input. When this input is asserted, the clock source ($\overline{CLKI}$) is divided by two and the output is the clock source for the timing circuit. When not asserted, the 20-MHz input is the clock source for the timing circuit. The $\overline{INTTIM}$ output will be 100 Hz when the input is 40 MHz or the 20 MHz input is 20 MHz.

Powerfail Standby

Powerfail standby is functional only when refresh mode 0 or 1 is selected. The powerfail logic provides automatic memory refresh for powerfail conditions when memory and the DYRC have a backup power source. Powerfail standby operation is enabled by the assertion of the $\overline{PFAIL}$ input. This input must be asserted by external logic before the system power supply becomes unstable. When asserted and refresh mode 0 or 1 is selected, any activity on the MicroVAX bus is ignored, and the DYRC continues to refresh memory until the $\overline{PFAIL}$ signal is deasserted or the backup power supply fails. The $\overline{PFAIL}$ signal should be deasserted a maximum of 10 µs before normal operation is resumed.

Reset/Powerup

The DYRC will reset its internal counters and timing sequencers when the $\overline{RESET}$ input is asserted for a minimum of 800 µs and the clock input is operating. When the $\overline{RESET}$ input is deasserted, the DYRC will initialize the DRAMs with eight refresh cycles. Memory access is delayed until the completion of the eight RAS-only refreshes. The DRAM data will be lost when $\overline{RESET}$ is asserted.

When power is first applied to the DYRC, the $\overline{RESET}$ input must be asserted for a minimum of 800 µs after the power supply voltages have stabilized.

Interfacing Requirements

A typical MicroVAX CPU and DYRC interface configuration is shown in Figure 5. The external logic required to interface a dynamic memory system is also shown. The actual logic may vary according the requirements of the system. The typical external components consist of external address decode logic, data bus transceivers with parity, memory address bus drivers, $\overline{RAS}$ and $\overline{CAS}$ drivers, and a write buffer.

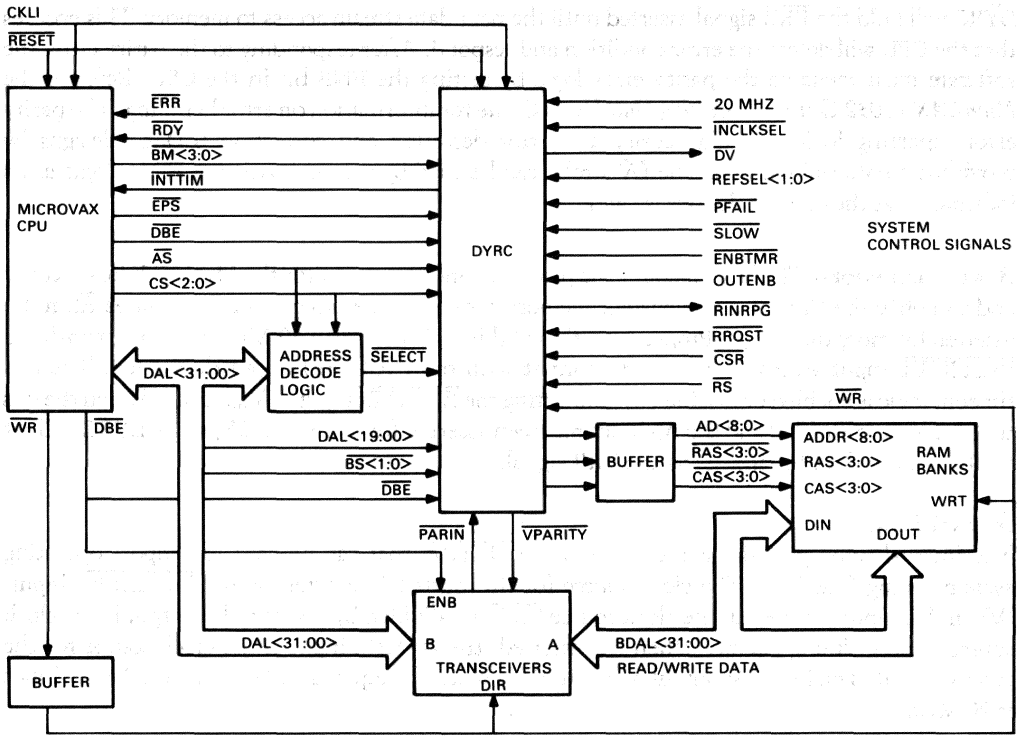


Figure 5 • MicroVAX 78584 Typical MicroVAX CPU and DYRC Interface Configuration

Dynamic RAM Requirements

The DYRC supports 256K dynamic RAMs that have the following characteristics.

- Multiplexed row and column addresses
- 9-bit memory address bus
- Data in and three-state data out to allow common input/output
- RAS only refresh
- 256 count or 512 count 4 ms refresh

The dynamic RAMs that meet the timing specifications in Table 7 will allow the MicroVAX 78032 CPU to access memory with no wait states or one cycle slip.

Table 7 • MicroVAX 78584 Dynamic RAM Specifications

Parameter	Access with no cycle slip (ns) Min.	Max.	Access with one cycle slip (ns) Min.	Max.
Access time from RAS	—	150	—	200
Access time from CAS	—	75	—	100
Row address hold time	20	—	25	—
Column address setup time	0	—	0	—
Column address hold time	45	—	55	—
RAS to CAS delay time	30	—	35	—
RAS precharge time	100	—	120	—
CAS precharge time	30	—	35	—
RAS pulse width	150	—	200	—

DMA Interface

The DYRC performs DMA read and write cycles similar to CPU read and write cycles. The DMA controller must control the DAL<19:0>, CS<2:0>, BS<1:0>, $\overline{WR}$, $\overline{BM}$<3:0>, $\overline{SELECT}$ and $\overline{AS}$ lines, and check the state of the $\overline{RDY}$ and $\overline{DV}$ signals to transfer data to or from memory. It must also check the state of the $\overline{ERR}$ line for error conditions.

When the DMA accesses a memory that is controlled by the DYRC, the DMA device must consider the latency time that may occur as a result of a refresh cycle. The DMA device can use the $\overline{RINPRG}$ signal to detect a refresh cycle in progress. The DYRC does not support refresh hold off. A DMA device must process a request without waiting or RAM data could be lost.

• Specifications

The mechanical, electrical, and environmental characteristics and specifications for the MicroVAX 78584 DYRC are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Power supply voltage (V_{DD}): 5 V ±5%
- Ground (V_{SS}): 0 V
- Temperature range (T_A): 70°C

Mechanical Configuration

The physical dimensions of the 78584 84-pin cerquad package are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Power supply voltage (V_{DD}): 5.0 V ±5%
- Input and output voltage applied: –0.5 V V_{DD} plus 0.5 V (5.5 V maximum)
- Storage temperature: –5°C to 125°C

Recommended Operating Conditions

- Power supply voltage (V_{DD}): 5 V ±5%
- Temperature (T_A) 0°C to 70°C

dc Electrical Characteristics

The dc electrical parameters of the MicroVAX 78584 DYRC for the operating voltage and temperature ranges specified are listed in Table 8.

Table 8 • MicroVAX 78584 dc Input and Output Parameters

Symbol	Parameter	Test Conditions	Requirements Min.	Requirements Max.	Units
V_{IH}	High-level input voltage		2.0	V_{DD}	V
V_{IL}	Low-level input voltage		0	0.8	V
V_{IHE}	High-level input voltage	(EPS only)	2.6	V_{DD}	V
V_{ILE}	Low-level input voltage	(EPS only)	0	0.2	V
V_{OH}	High-level output voltage	$I_{OH} = 28$ mA	2.4	V_{DD}	V
V_{OL}	Low-level output voltage	$I_{OL} = 11$ mA	0	0.4	V
I_{IL}	Input leakage current	V_{in}*	0	−10	μA
I_{OL}	Output leakage current	V_{in}*	0	−10	μA
I_{IH}	High-level input current			100	μA

Symbol	Parameter	Test Conditions	Requirements Min.	Max.	Units
I_{OH}	High-level output current			–40	mA
I_{DD}	Active supply current			300	mA
C_{in}	Input capacitance			5.0	pF
C_{out}	Output capacitance			5.0	pF

*To be determined.

ac Electrical Characteristics

The ac timing parameters for the MicroVAX 78584 are grouped according to their functions. Figure 6 shows the clock input waveform and symbols and the parameters are defined Table 9. Figure 7 shows the timing and symbols for the reset operation and the parameters are listed in Table 10. The memory read signal timing is shown in Figure 8 and the write signal timing in Figure 9. Table 11 lists the timing requirements for both memory read and write transactions. The refresh signal timing is shown in Figure 10 and the timing parameters are listed in Table 12. The register read and write timing is shown in Figures 11 and 12, respectively, and the parameters are listed in Table 13. Figure 13 shows the error reporting timing and Table 14 lists the timing requirements.

The following notes apply to Figures 7 through 13 and to the associated timing parameter tables.

- All times are in nanoseconds except where noted.
- The ac high levels are measured at 2.0 V and the low levels at 0.8 V.
- The ac characteristics are measured with a purely capacitive load at the output of 50 pF on $\overline{RAS}$<3:0>, $\overline{CAS}$<3:0>, $\overline{AD}$<8:0>, $\overline{RDY}$, and $\overline{DV}$.

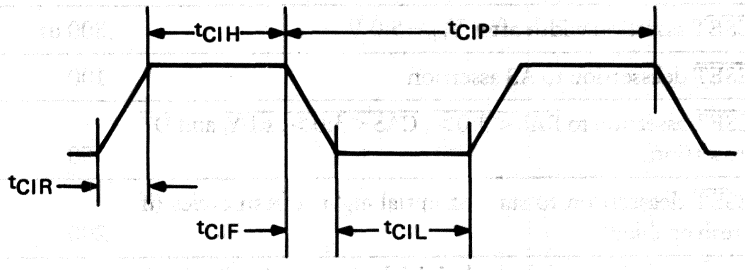


Figure 6 • MicroVAX 78584 Clock Input Timing

Table 9 • MicroVAX 78584 Clock Input Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{CIF}	Clock in fall time	—	4.5
t_{CIH}	Clock in high	8	—
t_{CIL}	Clock in low	8	—
t_{CIP}	Clock in period	25	50
t_{CIR}	Clock in rise time	—	4.5

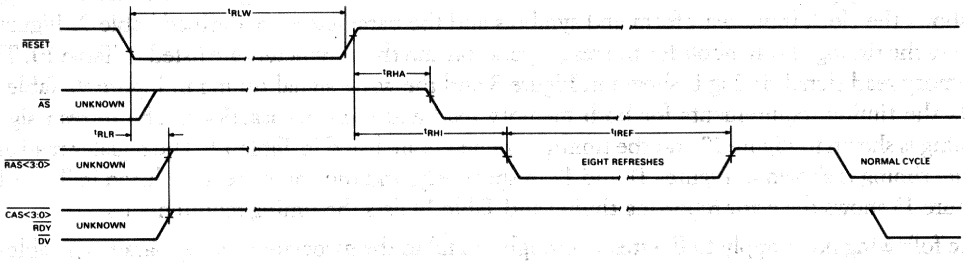


Figure 7 • MicroVAX 78584 Reset Input Signal Timing

Table 10 • MicroVAX 78584 Reset Input Timing Parameters

Symbol	Definition*	Requirements (ns) Min.	Max.
t_{RLW}	$\overline{\text{RESET}}$ assertion width after $V_{DD}=5.0$ V	800 µs	—
t_{RHA}	$\overline{\text{RESET}}$ deassertion to AS assertion	100	—
t_{RLR}	$\overline{\text{RESET}}$ assertion to $\overline{\text{RAS}}<3:0>$, $\overline{\text{CAS}}<3:0>$, $\overline{\text{RDY}}$, and $\overline{\text{DV}}$ deassertion	50	—
t_{RHI}	$\overline{\text{RESET}}$ deassertion to start of initial eight refresh cycles (if refresh enabled)	200	—
t_{IREF}	Time required to perform eight initial refresh cycles (if refresh enabled)	3 µs	—

*Delay from assertion of $\overline{\text{RESET}}$ to deassertion of AS by the MicroVAX 78032 CPU is typically 1.5 µs. CLKI input must be applied while $\overline{\text{RESET}}$ is being asserted.

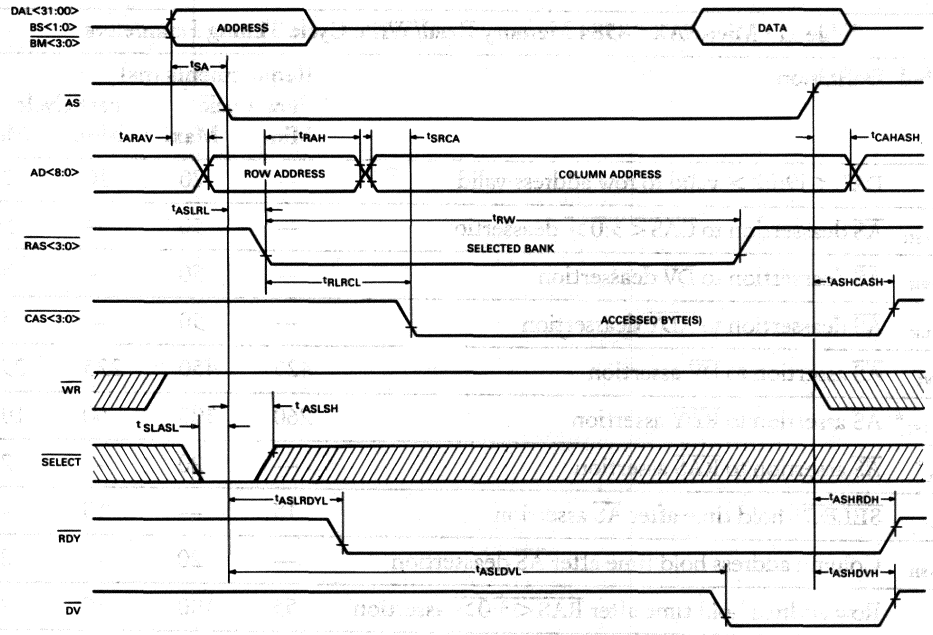


Figure 8 • MicroVAX 78584 Memory Read Cycle Timing

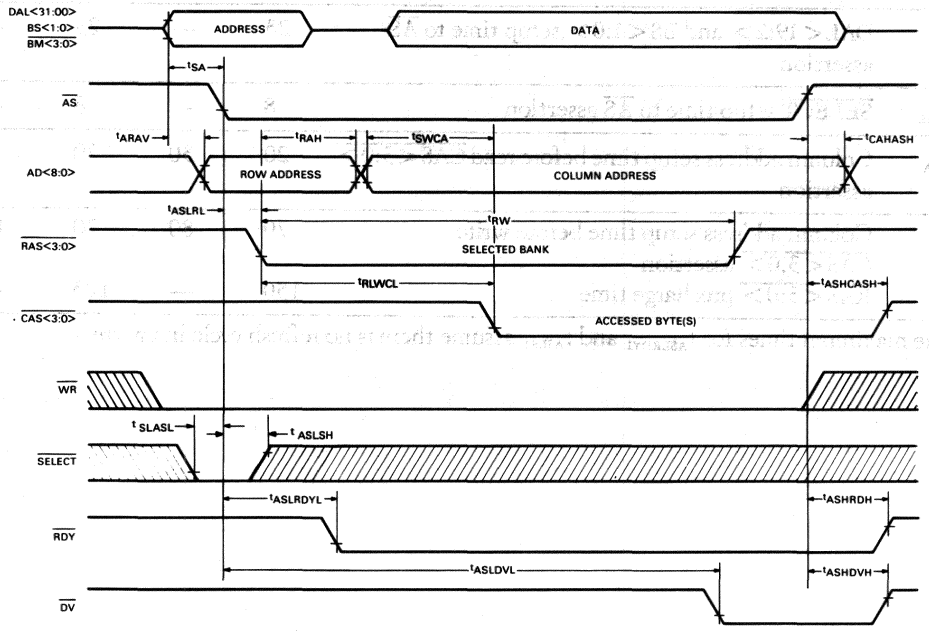


Figure 9 • MicroVAX 78584 Memory Write Cycle Timing

Table 11 • MicroVAX 78584 Memory Read/Write Cycle Timing Parameters

Symbol	Definition	Requirements (ns) Slow Cycle Min.	Slow Cycle Max.	Fast Cycle Min.	Fast Cycle Max.
t_{ARAV}	DAL<19:02> valid to row address valid	—	20	10	20
$t_{ASHCASH}$	$\overline{AS}$ deassertion to $\overline{CAS}$<3:0> deassertion	—	50	—	50
t_{ASHDVH}	$\overline{AS}$ deassertion to $\overline{DV}$ deassertion	—	50	—	50
t_{ASHRDH}	$\overline{AS}$ deassertion to $\overline{RDY}$ deassertion	—	50	—	50
t_{ASLDVL}	$\overline{AS}$ assertion to $\overline{DV}$ assertion	425	450	225	250
$t_{ASLRDYL}$*	$\overline{AS}$ assertion to $\overline{RDY}$ assertion	260	300	60	100
t_{ASLRL}*	$\overline{AS}$ assertion to $\overline{RAS}$ assertion	—	20	—	20
t_{ASLSH}	$\overline{SELECT}$ hold time after $\overline{AS}$ assertion	15	—	15	—
t_{CAHASH}	Column address hold time after $\overline{AS}$ deassertion	—	20	—	20
t_{RAH}	Row address hold time after $\overline{RAS}$<3:0> assertion	55	100	30	75
t_{RLRCL}	$\overline{RAS}$<3:0> assertion to read CAS assertion	105	150	55	110
t_{RW}	$\overline{RAS}$<3:0> pulse width	315	350	240	275
t_{RLWCL}	$\overline{RAS}$<3:0> assertion to write $\overline{CAS}$<3:0> assertion	115	140	115	140
t_{SA}	DAL<19:2> and BS<1:0> setup time to $\overline{AS}$ assertion	23	—	23	—
t_{SLASL}	$\overline{SELECT}$ setup time to $\overline{AS}$ assertion	8	—	8	—
t_{SRCA}	Column address setup time before read $\overline{CAS}$<3:0> assertion	20	30	20	30
t_{SWCA}	Column address setup time before write $\overline{CAS}$<3:0> assertion	70	80	70	80
	$\overline{RAS}$<3:0> precharge time	150	—	125	—

*The maximum times for $t_{ASLRDYL}$ and t_{ASLRL} assume there is no refresh cycle in progress.

Table 12 • MicroVAX 78584 Refresh Timing Parameters

Symbol	Definition	Requirements (ns) Slow Cycle Min.	Slow Cycle Max.	Fast Cycle Min.	Fast Cycle Max.
t_{ADS}	Refresh address setup time to $\overline{RAS<3:0>}$ assertion	100	—	25	100
t_{ADH}	Refresh address hold time after RAS<3:0> assertion	235	265	135	165
t_{REFPRE}	$\overline{RAS<3:0>}$ precharge	165	185	165	185
t_{REFW}	$\overline{RAS<3:0>}$ pulse width	215	235	165	185

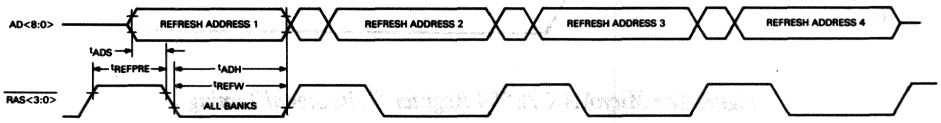


Figure 10 • MicroVAX 78584 Refresh Signal Timing

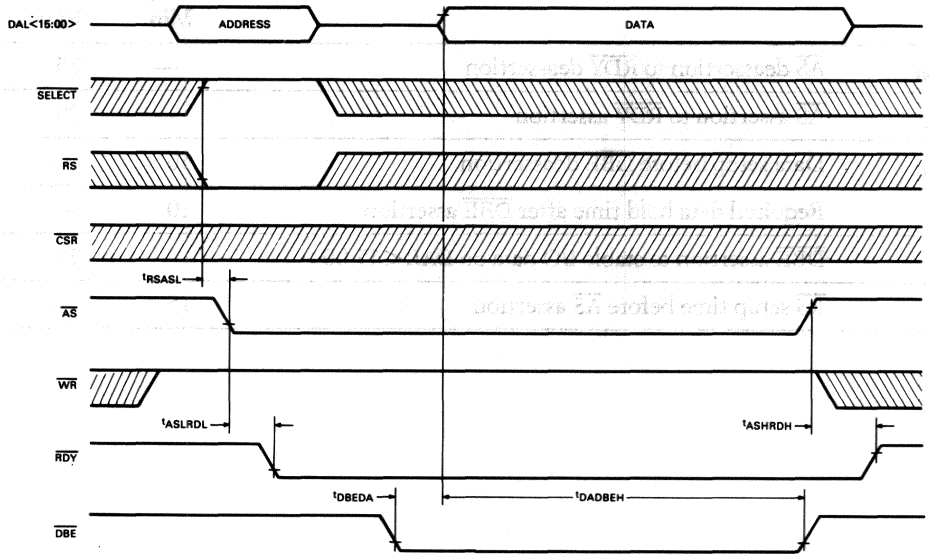


Figure 11 • MicroVAX 78584 Register Read Signal Timing

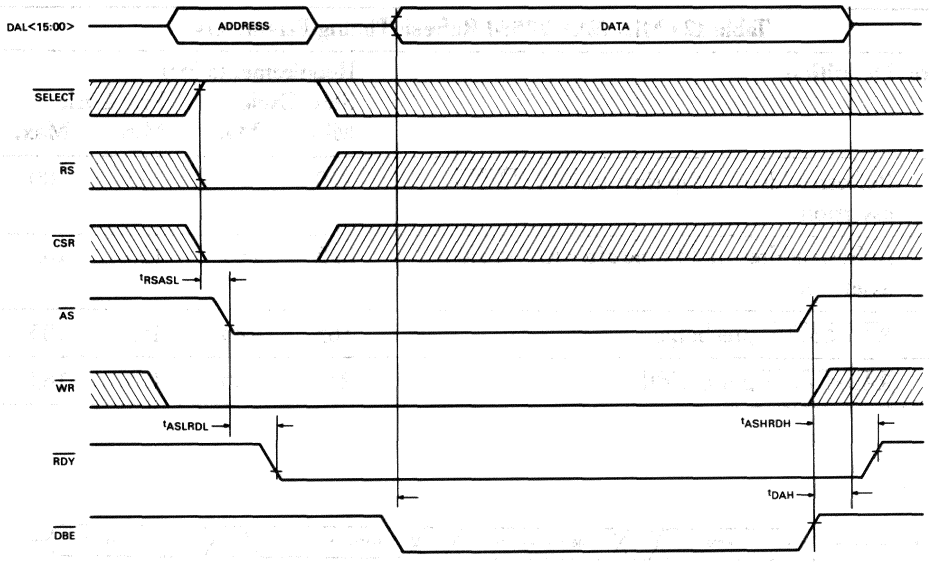


Figure 12 • MicroVAX 78584 Register Write Signal Timing

Table 13 • MicroVAX 78584 Register Read/Write Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{ASHRDH}	$\overline{AS}$ deassertion to $\overline{RDY}$ deassertion	—	35
t_{ASLRDL}	$\overline{AS}$ assertion to $\overline{RDY}$ assertion	—	25
t_{DADBEH}	Data setup before $\overline{DBE}$ deassertion	50	—
t_{DAH}	Required data hold time after $\overline{DBE}$ assertion	10	—
t_{DBEDA}	$\overline{DBE}$ assertion to stable I/O data on DAL<15:00>	—	25
t_{RSASL}	$\overline{RS}$ setup time before $\overline{AS}$ assertion	15	—

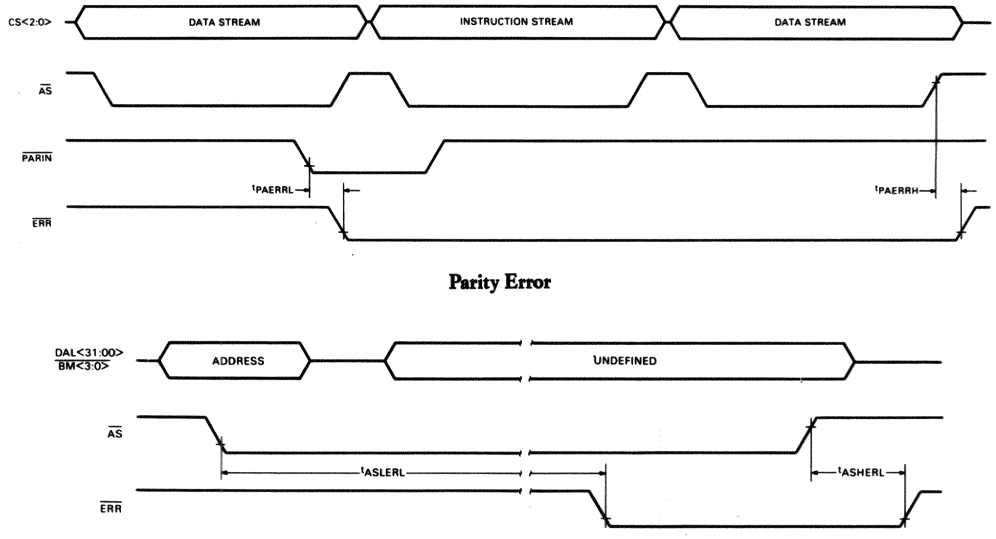


Figure 13 • MicroVAX 78584 Error Reporting Signal Timing

Table 14 • MicroVAX 78584 Error Reporting Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{PAERRL}	$\overline{PARIN}$ to $\overline{ERR}$ assertion	—	50
t_{PAERRH}	$\overline{ERR}$ deassertion after data stream cycle	—	50
t_{ASLERL}	$\overline{AS}$ assertion to $\overline{ERR}$ assertion	25 µs	—
t_{ASHERH}	$\overline{AS}$ deassertion to ERR deassertion	—	50

Features

- Low-cost MicroVAX development system contained a single module
- Full-speed incircuit emulation of MicroVAX 78032 microprocessor chip and MicroVAX 78132 floating-point unit chip
- PROM-resident monitor with powerful command set
- Hardware event detector for MicroVAX bus control signals and addresses
- 64 Kbytes of relocatable memory simulation RAM for target application
- Two RS-232-compatible serial lines for host and terminal connections, one with modem control
- Internal clock of 10-, 20-, or 40-MHz or external clock
- Bus timing controllable by ADVICE and target or by target alone (wait states only)
- Powerup diagnostic tests to verify its own operation
- Connectors for clock-in, trigger-in, trigger-out, power supply, and RS-232 ports
- Single 5-volt power supply

Description

The Applications Development MicroVAX Incircuit Emulator (ADVICE) is a low-cost, incircuit emulator used for the development of hardware and software products based on the MicroVAX 78032 32-bit microprocessor and the MicroVAX 78132 floating-point unit. Diagnostic or application programs for the user's target may be developed on a VAX/VMS host and downloaded to ADVICE. ADVICE contains all the necessary hardware and software required to quickly and easily debug the target.

The emulator is contained on a single module and includes a PROM-resident monitor, 32 Kbytes of monitor RAM, as many as 64 Kbytes of a memory simulation RAM for the user's target application, a switch-selectable clock, a MicroVAX microprocessor (CPU), a floating-point unit (FPU), and two serial lines—one for a console terminal and one for a host. An event detector is included to compare stored values to MicroVAX bus addresses or to the logical levels of various MicroVAX bus control signals.

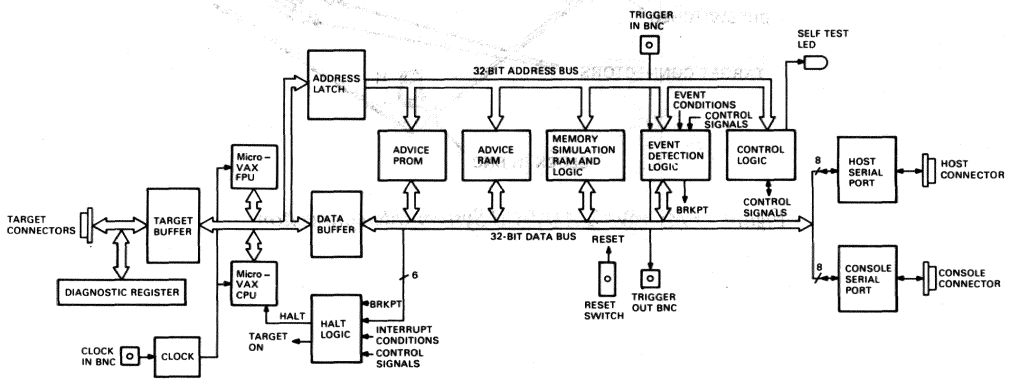


Figure 1 • ADVICE Block Diagram

System Overview

The ADVICE enables the hardware emulation of the MicroVAX CPU and FPU at full-speed (40-MHz) and provides the user with complete control over MicroVAX CPU operation. This includes the ability to start and stop the program operation and to single-step through a program. It contains hardware and software that can be used directly by the target application to simplify the development of the target hardware and software.

Figure 2 shows the system interconnections to the ADVICE module. The host computer is used to write, edit, and optionally debug the user programs that will eventually run on the target system. These programs are converted to hexadecimal format by the DECPROM software and downloaded to the ADVICE over a serial line. A local console terminal is used to enter commands and control the entire development process.

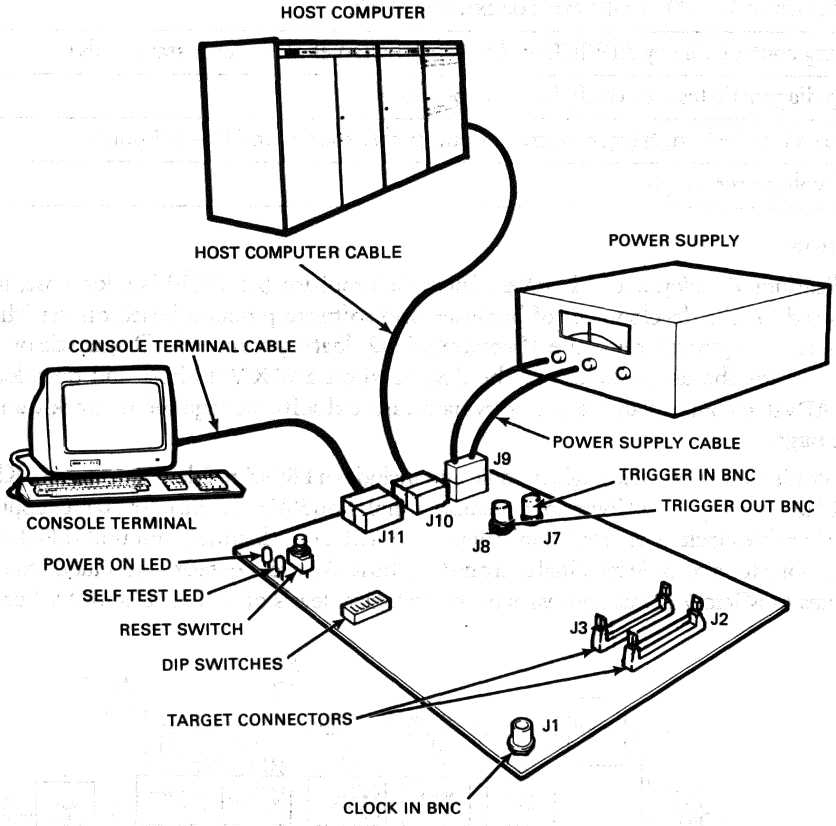


Figure 2 ▪ MicroVAX ADVICE System Interconnection

Module Connectors

Table 1 lists the connector types that are included on the ADVICE module and their function. Refer to Figure 2 for the location of the connectors on the module and to the *ADVICE User's Guide* for the connector applications and signals.

Table 1 • ADVICE Module Connectors

Connector	Type	Function
J1	BNC	Clock in
J2	40-pin	Target circuit
J3	40-pin	Target circuit
J7	BNC	Trigger in
J8	BNC	Trigger out
J9	4-pin	Power supply
J10	25-pin	Host computer port (serial-line)
J11	25-pin	Console port(serial-line)

Switches and Indicators

The ADVICE module contains eight switches in a dual-inline package (DIP) that are used to select various functions, a reset pushbutton switch, and two LED indicators. Table 2 lists the positions and selections of the switches on the DIP. The reset pushbutton is used to initialize the module and causes a series of diagnostic tests to be performed. The Power on light indicates that the dc power is applied and the Self-test indicator is used to verify the correct operation of the diagnostic programs.

Table 2 • ADVICE DIP Switch Functions

Switch 1	2	3	4	5	6	7	8	Selection
off	off							External Clock
off	on							10-MHz clock
on	off							20-MHz clock
on	on							40-MHz clock
		off						FPU disabled
		on						FPU enabled
			off					Comprehensive diagnostics
			on					Limited diagnostics

Switch 1	2	3	4	5	6	7	8	Selection
				off	off	off		110 baud rate
				on	off	off		300 baud rate
				off	on	off		600 baud rate
				on	on	off		1200 baud rate
				off	off	on		2400 baud rate
				on	off	on		4800 baud rate
				off	on	on		9600 baud rate
				on	on	on		19200 baud rate
							off	Modem control disabled
							on	Modem control enabled

PROM-resident Monitor

The ADVICE monitor is contained in 64 Kbytes of EPROM and allows the user to load, execute, and debug application programs. The monitor is controlled by the ADVICE commands and allows the user to perform the following functions:

- Upload/download programs to or from a host in Intel, Mostek, or Tektronix hexadecimal formats
- Assemble MACRO-32 instructions from a terminal or host
- Disassemble information stored in memory to display MACRO-32 instructions
- Start or stop program execution
- Single-step through a program
- Examine and deposit memory and the CPU registers
- Set and clear up to 16 software breakpoint and/or tracepoints
- Detect or ignore hardware events such as the assertion of various MicroVAX bus control signals
- Define or undefine symbols
- Set or show current operational modes such as radix, word width, and hexadecimal data format
- Maintain four command line buffers and allow repetitive execution of any buffer
- Define up to 22 cursor control keys
- Fill a memory range with any byte value
- Evaluate expressions in binary, decimal, octal, or hexadecimal
- Perform ADVICE hardware diagnostic tests
- Use command abbreviations
- Receive help in using ADVICE commands

• Modes of Operation

The ADVICE operates in transparent and nontransparent mode. The selection of the mode depends on application to be developed.

In transparent mode, it acts as a full-speed (40 MHz) incircuit MicroVAX processor emulator and appears to be a MicroVAX 78032 chip to the target application. It provides complete control over the CPU operation by allowing the user to start, stop, and single-step through program operation. The ADVICE does not interfere with the operation of the target application. This allows the debugging of user hardware and software that overlaps the ADVICE hardware and software. Debugging a boot PROM for the target application would typically be performed in transparent mode.

In nontransparent mode, the internal circuits of ADVICE are accessible to the user. This allows the use of ADVICE hardware and software as an extension of the target hardware and software. Nontransparent mode allows access to serial-line interface routines in the ADVICE monitor.

Physical Address Space

The physical address space available depends on the mode of operation. In transparent mode, the entire MicroVAX physical address space from 00000000 to 3FFFFFFF (hexadecimal) is available. In nontransparent mode, a 256 Kbyte block of memory (addresses 20040000 through 2007FFFF) is used by the ADVICE and is not available to the user. The memory map for the nontransparent mode is shown in Figure 3.

00000000	AVAILABLE TO USER
2003FFFF	
20040000	ADVICE PROM (64 KB)
2004FFFF	
20050000	ADVICE RAM (32 KB)
20057FFF	
20058000	RESERVED FOR FUTURE USE
2005FFFF	
20060000	RESERVED FOR ADVICE HARDWARE
20061FFF	
20062000	RESERVED FOR FUTURE USE
20063FFF	
20064000	RESERVED FOR ADVICE HARDWARE
20064004	RESERVED FOR ADVICE HARDWARE
20064010	STATUS REGISTER
20064020	CONSOLE UART
20064024	CONSOLE UART
20064028	CONSOLE UART
2006402C	CONSOLE UART
20064030	HOST UART
20064034	HOST UART
20064038	HOST UART
2006403C	HOST UART
20064040	RESERVED FOR ADVICE HARDWARE
20064070	
20064071	RESERVED FOR FUTURE USE
2007FFFF	
20080000	AVAILABLE TO USER
3FFFFFFF	

NOTE: ALL ADDRESSES ARE AVAILABLE TO USER IN TRANSPARENT MODE.

Figure 3 • ADVICE Nontransparent Mode Memory Map

Event Detector

The ADVICE contains an event detector that is primarily used during diagnostic operations to analyze the interaction between the advice and target application. Depending on the states of its inputs, the event detector provides a pulse output that can be used as a trigger to external circuits. The event detector monitors the following signals:

- The logical levels of MicroVAX bus control signals including $\overline{EPS}$, $\overline{AS}$, $\overline{WE}$, $\overline{DMR}$, $\overline{IRQ<3:0>}$, and CS<2:0>
- The output of an address comparator that indicates whether a MicroVAX bus address is greater than, less than, or equal to a stored value
- The logical level of a BNC connector input

The ADVICE can be programmed to ignore events or halt program execution when an event occurs. In either case, a trigger output from ADVICE indicates the occurrence of an event.

Memory Simulation

The ADVICE contains 32-Kbytes or 64-Kbytes of memory simulation RAM (MSR) that can be mapped anywhere in the target address space. The enabling of MSR ensures that users that valid memory is available during the application development.

Serial-line Ports

One EIA RS-232 serial-line port is used to connect the host processor to the ADVICE and includes modem control. The remaining EIA RS-232 serial-line port connects to the console terminal. These ports are used in nontransparent mode and an internal subroutine is included to access these ports.

Self-diagnostics

Several diagnostic tests are performed in the ADVICE during the powerup sequence or when the Reset pushbutton on the ADVICE module is pressed. These programs verify the integrity of the information in PROM and RAM (including MSR) and the operation of serial-line ports, event detection logic, and MicroVAX CPU and FPU. The user can select a limited diagnostic routine to preserve RAM contents or a comprehensive diagnostic routine that tests RAM without regard for its original contents.

User-supplied Equipment

The standard and optional equipment and software required for use with the ADVICE is listed as follows. The optional listing depends on the type of application to be developed.

- A VAX processor system with an available RS-232 serial-line port
- VAX/VMS operating system, Version 3.4 or higher
- DECPROM software, Version 1.0 or higher
- 5-Volt power supply
- Local VT100 compatible terminal
- Appropriate etch layout around target MicroVAX CPU surface mount pads to match ADVICE connector (Refer to the *ADVICE User's Guide)*
- RS232-compatible modem for remote host or terminal (optional)
- External clocks or triggering circuits including cables (optional)

Specifications

The physical and electrical specifications of the ADVICE are as follows:

Operating Environment

- Temperature: 15°C to 32°C at 200 linear feet/minute air flow
- Relative humidity: 20% to 80% (noncondensing)

Module Dimensions

- Height: 26.0 cm (10.25 inches)
- Length: 40.6 cm (16.0 inches)
- Width: 2.5 cm (1.0 inches)

Power Requirements

- Power supply: 5 V ±5 % at 6 A (maximum) 4.5 A (typical)

• Features

- Emulates the VAX processors
- Consists of the DC328 instruction/execution logic (I/E chip), the DC329 memory management logic (M chip), the DC330 floating-point accelerator logic (F chip), and five DC327 ROM/RAM memory chips
- Provides full VAX processor functionality on a single module

• Description

The V-11 chipset is designed for use on the Scorpio CPU module (KA820) that is a single module VAX processor. The V-11 chip set configuration for the Scorpio CPU is shown in the following diagram and consists of the I/E chip, M chip, F chip, and five ROM/RAMs.

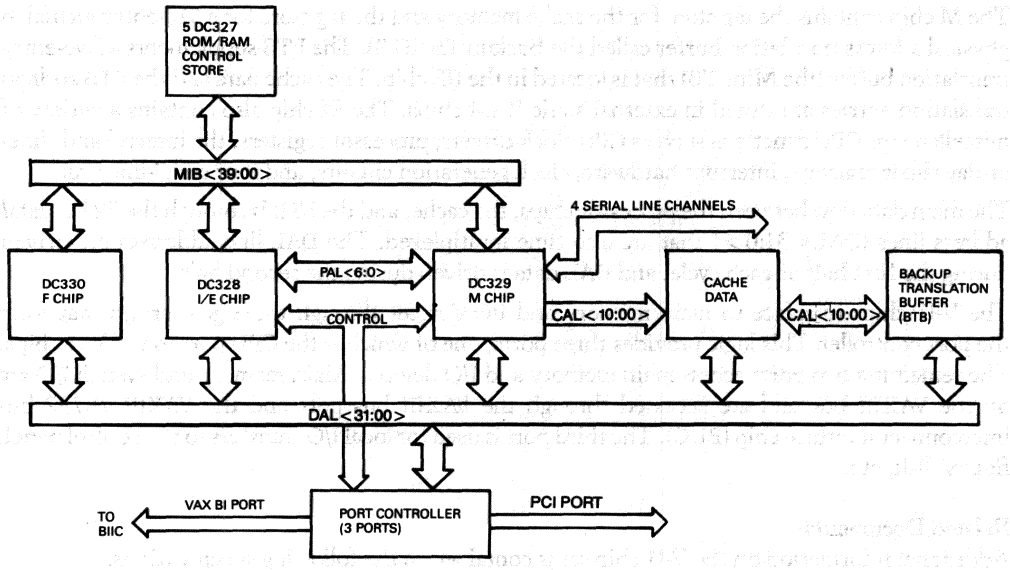


Figure 1 • V-11 Scorpio Microprocessor Block Diagram

The five ROM/RAMs are connected in parallel to provide a 16K by 40-bit control store memory. The CPU is positioned between the I/E chip and M chip that are controlled by microcode from the ROM/RAM chips. The optional F chip contains high-speed logic and provides internal microcode for accelerating the operation of floating-point and other mathematical instructions.

The I/E chip consists of the instruction buffer (I box), execution unit (E box), and memory interface logic which performs virtual-to-physical address translation through a five-entry address translation buffer (Mini-TB). It also contains the microsequencer and microinstruction prefetcher that interfaces to the microcontrol store. The microcontrol store is five ROM/RAM chips that form a combinational 16K by 40-bit ROM array and a 1K by 40-bit RAM array. The ROM section contains the V-11 microcode that controls the CPU chips, and the RAM section is used for implementing microcode patches to the ROM locations. These chips also provide 32 (total of 160) locations each of content addressable memory (CAM) that is used to store and compare the addresses of the patched ROM locations with the incoming microaddress.

The microcontrol store memory interfaces to the F chip, I/E chip, and M chip through the 40-bit, time multiplexed, microinstruction bus (MIB). The I/E chip drives the microaddress to the control store during the first half of microcycle, and each ROM/RAM chip drives 8 bits of the microinstruction onto the MIB during the second half of the cycle. The MIB lines that are not used for the microaddress are used for interchip communication during the address portion of the cycle.

The M chip contains the tag store for the cache memory and the tag store for a 512-entry virtual to physical address translation buffer called the Backup TB (BTB). The BTB supplements a five-entry translation buffer (the Mini-TB) that is located in the I/E chip. The cache data and the BTB address translation entries are stored in external static RAM chips. The M chip also contains a variety of miscellaneous CPU functions such as CPU clock drivers, processor registers, the interval and time-of-day timer registers, interrupt hardware, clock generation circuits, and four serial-line units.

The main data flow between the processor chips, the cache, and the BTB is through the 32 bit data/address lines (DAL<31:0>) that are also time multiplexed. The DAL line addresses are driven during the first half of each cycle, and DAL data is driven during the second half.

The V-11 chips interface to main memory and I/O devices through three gate arrays that form the port controller. This logic provides three ports, one of which is the CPU port to the V-11 chips. The remaining two ports access main memory and I/O devices. Main memory and system I/O are on the VAXBI bus and are accessed through the VAXBI bus port and the VAXBI 78732 bus interconnect interface chip (BIIC). The third port is used for local I/O transfers to the control panel, floppy disk, etc.

Related Documents

Additional information on the V-11 chip set is contained in the following specifications.

- *V-11 CPU Functional Specification*
- *DC327 ROM/RAM Chip Functional Specification*
- *DC328 I/E Chip Functional Specification*
- *DC329 M Chip Functional Specification*
- *DC330 F Chip Hardware Specification*
- *Scorpio Port Controller Specification*
- *KA820 CPU Module Specification*

Features

- Custom designed VLSI ROM/RAM chip for the V-11 processor.
- Contains 16K by 8-bit word ROM, 1K by 8-bit word RAM and 32 by 14-bit word CAM.

Description

The DC327 ROM/RAM, contained in a 44-pin package, is a VLSI chip designed for the V-11 processor. It contains 16K 8-bit words of masked programmed ROM (read-only memory), 1K by 8-bit words of RAM (random access memory) and 32 14-bit words of CAM (content addressable memory). The DC327 chip is designed to facilitate the replacing (patching) of incorrect ROM data with RAM data. Figure 1 is a block diagram of the DC327 chip.

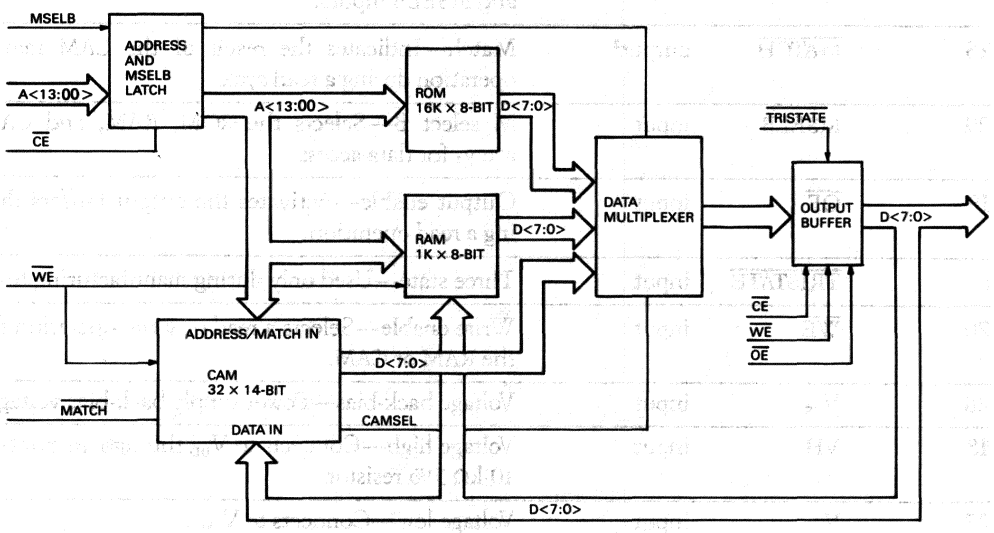


Figure 1 • DC327 ROM/RAM Block Diagram

• Pin and Signal Descriptions

The input and output signals and power and ground connections for the DC327 44-pin package are summarized in Table 1. The table also contains the physical pin locations that are shown on Figure 2. The following paragraphs provide more detailed descriptions of the inputs and outputs of the M chip.

Table 1 • DC327 Pin and Signal Summary

Pin	Signal	Input/output	Description/Function
17-14,10-7	D<7:00>	inputs/outputs[1]	Data—Transfers data from the ROM and transfers data to and from the RAM and CAM.
3,4,30-33, 36-43	A<13:00>	inputs	Address—The ROM, RAM, and CAM address inputs.
5	$\overline{CE}$	input	Chip enable—Clock signal to latch the A<13:00> and MSELB inputs.
26	$\overline{MATCH}$	output[2]	Match—Indicates the result of the CAM match operation during a read cycle.
29	MSELB	input	M select B—Selects the ROM, RAM, and CAM arrays for data access.
19	$\overline{OE}$	input	Output enable—Activates the output buffers during a read operation.
6	$\overline{TRISTATE}$	input	Three state—Used only during manufacturing test.
20	$\overline{WE}$	input	Write enable—Selects a read or write operation for the RAM or CAM.
28	V_{BB}	input	Voltage back-bias—Power supply back-bias voltage.
18	VH	input	Voltage high—Connects to V_{DD} through an external 10-kΩ 5% resistor.
27	V	input	Voltage low—Connects to V_{SS}.
21			Not used.
35,34,22, 23,13,2	V_{DD}	input	Voltage—Power supply voltage.
25,24,11, 12,1,44	V_{SS}	input	Ground—Common ground reference.

[1]Three-state output
[2]Open-drain

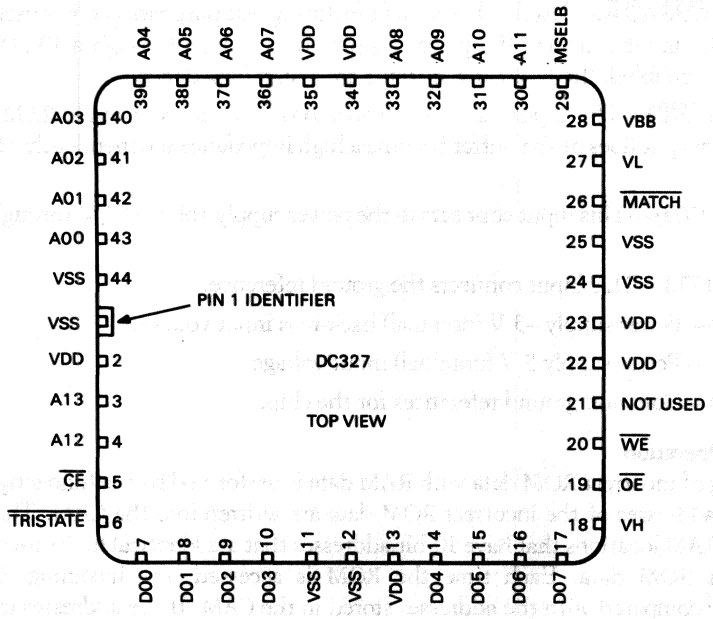


Figure 2 • DC327 Pin Assignments

Data (D<7:0>)—Bidirectional data lines used to transfer data from the ROM, and to and from the RAM and CAM.

Address (A<13:00>)—The address inputs to reference the ROM, RAM, and CAM. These inputs are latched during read and write cycles during the high-to-low transition of the $\overline{CE}$ input.

Chip Enable ($\overline{CE}$)—A clock signal that latches the A<13:00> and MSELB inputs, and initiates the internal access cycle with a high-to-low transition. When deasserted, the $\overline{CE}$ signal causes the data output buffers to become a high impedance.

Match ($\overline{MATCH}$)—An open-drain output that indicates the result of the CAM match operation during a read cycle. The CAM match operation and the $\overline{MATCH}$ output are active during RAM read cycles as well as ROM read cycles. The $\overline{MATCH}$ output is disabled only when the ROM/RAM chip is in test mode. The state of the $\overline{MATCH}$ signal is not defined during RAM/CAM write cycles.

M Select B (MSELB)—An address input that selects the ROM or CAM arrays and the RAM array for data access. This signal is latched during read or write cycles.

Output Enable (OE)—Activates the output buffers during a read operation.

Three State ($\overline{\text{TRISTATE}}$)—Used only for manufacturing test purposes and not used during normal operation. The input connects the power supply voltage (V_{DD}) through a 10-kΩ ±5% external resistor. When enabled, the D<7:0> outputs are a high impedance.

Write Enable ($\overline{\text{WE}}$)—This input selects the read and write operations to the RAM or CAM. When asserted, the output lines of the buffer become a high impedance independently of the state of $\overline{\text{OE}}$ input.

Voltage High (VH)—This input connects to the power supply voltage (V_{DD}) through a 10-kΩ ±5% resistor.

Voltage Low (VL)—This input connects the ground reference.

Voltage (V_{BB})—Power supply –3 V (nominal) back-bias input voltage.

Voltage (V_{DD})—Power supply 5 V (nominal) input voltage.

Ground (V_{SS})—Common ground references for the chip.

Functional Operation

The patching of incorrect ROM data with RAM data is performed by the following operation. The 14-bit ROM addresses of the incorrect ROM data are written into the CAM. The correct data is written into RAM locations that have 10-bit addresses that are identical to the low-order 10-bits of the incorrect ROM data. Each time the ROM is accessed, the incoming ROM address is automatically compared with the addresses stored in the CAM. If the addresses are the same, the $\overline{\text{MATCH}}$ output is asserted to inform the external logic that the ROM data currently being accessed is incorrect and that the correct data is in the RAM. The RAM is then accessed in the following cycle(s) with the same address. tThe ten low-order address bits must be identical to access the correct data.

The DC327 ROM/RAM chip contains an address/MSELB latch; a ROM, RAM, CAM, a data multiplexer, and an output buffer shown in Figure 1.

Address/MSELB Latch—This latch holds address inputs A<13:00> and the MSELB input. When the $\overline{\text{CE}}$ input is deasserted, the address and the MSELB inputs pass through the latch to the internal logic of the chip. A high-to-low transition of the $\overline{\text{CE}}$ input loads the state of the address and MSELB H information into the latch, and any transitions on those inputs that follow are ignored.

Read-only Memory (ROM)—The 16K by 8-bit word ROM is accessed by the address information on the A<13:00> lines from the output of the address latch. The ROM is accessed during a high-to-low transition of the $\overline{\text{CE}}$ input and the 8-bits of the ROM data are transferred to the data multiplexer.

Random Access Memory (RAM)—The 1K by 8-bit word RAM is accessed by the address information on the A<9:0> lines from the output of the address latch. A high-to-low transition of the $\overline{\text{CE}}$ input when the $\overline{\text{WE}}$ input is deasserted will access the RAM and transfer 8-bits of the RAM data to the data multiplexer. The RAM is written with the data on the D<7:0> lines when the latched MSELB signal is asserted, the $\overline{\text{CE}}$ signal makes a high-to-low transition, and the WE input is asserted with the proper timing.

Content Addressable Memory (CAM)—The CAM consists of 32 14-bit registers, 32 14-bit comparators, logic for detecting an address match, a CAM test address register, and logic for testing the CAM. When the CAM operates in the normal mode (not test mode), the latched address information on lines A<13:00> is compared with the addresses stored in the 32 14-bit CAM registers. If the latched address is the same as an address stored in the CAM registers, the $\overline{\text{MATCH}}$ output is asserted. The $\overline{\text{MATCH}}$ output is indeterminate during a RAM or CAM write cycle. In test mode, the $\overline{\text{MATCH}}$ output is disabled. The $\overline{\text{MATCH}}$ open drain output is internally disabled at the beginning of each cycle and must be pulled up to its inactive state by an external resistor.

CAM Write Operations

The mask programmable CAM write decoder enables up to eight ROM/RAM chips to be connected in parallel while maintaining a unique write access to each CAM register on each of the eight ROM/RAM chips. A ROM/RAM chip can access each of the 32 individual CAM registers. The CAM registers are written when the $\overline{WE}$ input is asserted, the MSELB input is zero, the $\overline{CE}$ input is asserted, and when the write decoder decodes one of the reserved addresses shown in Table 2.

Table 2 • DC327 CAM Write Decoder Addresses

Address A<13:00> (hexadecimal)*	Internal Structure Accessed CAM Register	Data
0000	0 <07:00>	D<7:0>
0001	0 <13:08>	D<5:0>
0002	1 <07:00>	D<7:0>
0003	1 <13:08>	D<5:0>
	(continued through)	
003E	31 <07:00>	D<7:0>
003F	31 <13:08>	D<5:0>

*Assume mask programmable bits A<8:6> of the CAM write decoder are zeros.

Data Multiplexer

The data multiplexer selects the data to be transferred to the output buffer and then to the D<7:0> lines. Eight bits of ROM data, RAM data, or CAM register match state are selected depending on the state of the internal CAMSEL signal and the latch MSELB signal listed in Table 3.

Table 3 • DC327 Data Output Selection*

CAMSEL	MSELB	D<7:0> output
0	0	ROM
0	1	RAM
1	X	CAM match state

*X = 1 or 0.

Output Buffer

The output buffer drives the output of the data multiplexer onto the D<7:0> lines. When not transferring data, the outputs of the buffer are a high impedance. Table 4 lists the state of the outputs depending on the input signal conditions.

Table 4 • DC327 Output Buffer States*

$\overline{TRISTATE}$	$\overline{CE}$	$\overline{WE}$	$\overline{OE}$	Output
1	0	1	0	Drive output
1	0	1	1	high-impedance
0	X	X	X	high-impedance
X	1	X	X	high-impedance
X	X	0	X	high-impedance

*X = 1 or 0.

Summary of Read and Write Operations

Table 5 lists the inputs required for read and write operations for each of the ROM/RAM chip sections.

Table 5 • DC327 Read and Write Operation Selection*

$\overline{CE}$	$\overline{TRISTATE}$	$\overline{WE}$	$\overline{OE}$	$\overline{MSELB}$	A<13:0>	Operation
0	1	1	0	0	3FFB-0000	ROM read
0	1	1	0	1	3FFB-0000[1]	RAM read
0	1	1	0	X	3FFF-3FFC	CAM match state read
0	1	0	X	0	003F-0000[2]	CAM write
0	1	0	X	0	0201-0200	CAM test address reg write
0	1	0	X	1	3FFB-0000[1]	RAM write

[1]The RAM is accessed by address bits A<09:00>. Bits A<13:10> are not used.
[2]Mask programmable bits A<08:06> of the CAM write decoder are zeros.
*X = 1 or 0.

• Specifications

The mechanical, electrical, and environmental specifications for the DC327 are contained in the following paragraphs. The test conditions for the parameters in these specifications, unless specified otherwise, are as follows:

- Ambient temperature (T_A): 0°C to 70°C
- Power supply voltage (V_{DD}): 5.0 V ±5% (maximum ripple 200 mV peak-to-peak)
- Power supply back-bias voltage (V_{BB}): −3.0 V ±15% (maximum ripple 200 mV peak-to-peak)

Mechanical Configuration

The mechanical dimensions for mounting the DC327 44-pin leadless package are shown in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to a device. Exposure to the absolute maximum rating conditions for extended periods may adversely affect the reliability of the device.

- Power supply voltages (V_{DD}): −1.5 V to 7.0 V
- Power supply substate voltage (V_{BB}): −6.0 V to 0 V
- Input/output pin voltage: −1.0 V to 10 V
- Storage temperature range: −55°C to 125°C
- Ambient temperature operating range (T_A): 0°C to 70°C

Recommended Operating Conditions

- Power supply voltage (V_{DD}): 5.0 V ±5%
- Power supply back-bias voltage (V_{BB}): − 3.0 V
- Ambient temperature (T_A): 25°C

ac and dc Electrical Characteristics

Refer to the *DC327 ROM/RAM Chip Functional Specification* for the dc input and output parameters and ac timing parameters.

Features

- Main processor element for the V-11 chipset
- Used with the M-chip and F-chip to emulate VAX instruction set and memory management
- Initiates memory references and contains address translation logic

Description

The DC328 Instruction/Execution (I/E) logic is contained in a 132-pin PGA package and is the main processing element of the V-11 processor chipset. It prefetches instructions, parses opcodes and specifiers, initiates all memory references, and contains the register file and arithmetic logic unit (ALU) and most of the address translation hardware. Together with the F chip and M chip, it emulates the VAX instruction set and memory architecture. The functional block diagram of the I/E chip is shown in Figure 1.

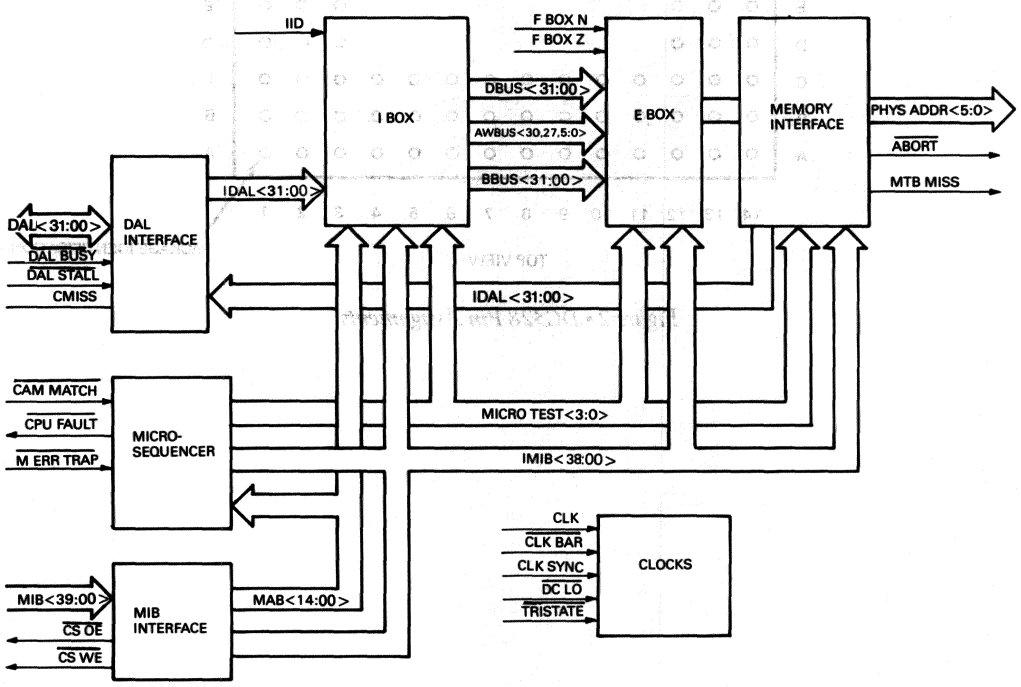


Figure 1 • DC328 I/E Chip Block Diagram

Pin and Signal Descriptions

The input and output signals and power and ground connections to the DC328 I/E chip are shown in Figure 2 and summarized in Table 1. The paragraphs that follow provide a more detailed description of the signal functions listed in Table 1.

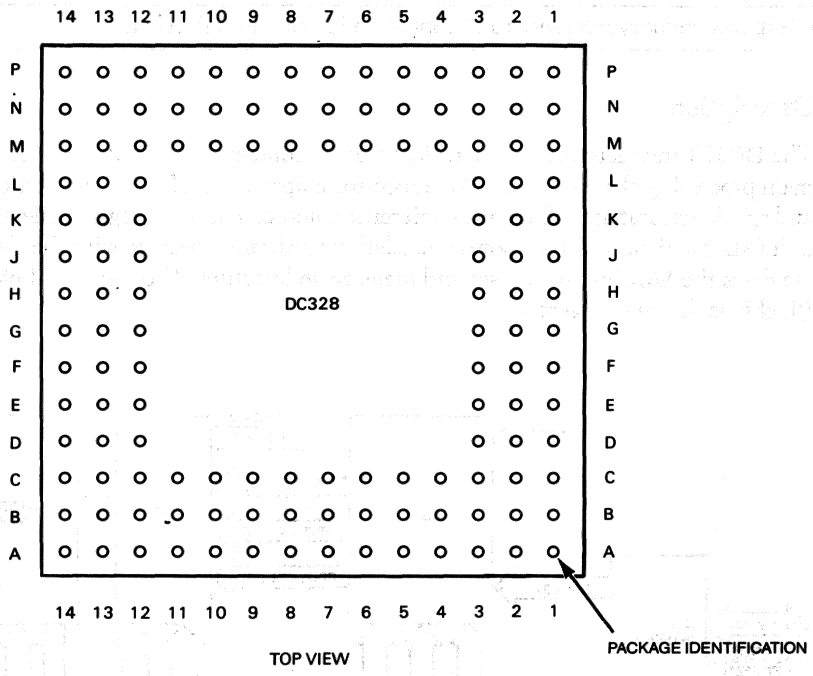


Figure 2 ▪ DC328 Pin Assignments

Table 1 • DC328 Pin and Signal Summary

Pin	Signal	Input/output	Description/Function
P4,P3,N3,P2, M3,N2,M2,P1, L3,N1,L2,M1, K3,L1,K2,K1, J2,H1,J1,G1, H2,F1,E2,E1, E3,D1,D2,C1, C2,B1-B3	DAL<31:00>	input/output	Data/address lines—Transfer data and address information to and from the I/E chip.
C02	$\overline{\text{DAL BUSY}}$	input	Data/address lines busy—Indicates that the port controller has control of the DAL lines.
A7	$\overline{\text{DAL STALL}}$	input	Data/address line stall—Indicates that the data source cannot respond to a data request during the same cycle.
B7	$\overline{\text{CMISS}}$	input	Cache miss—Indicates that the backup translation buffer miss has occurred.
K13	$\overline{\text{CAM MATCH}}$	input	Content addressable memory match—Indicates that an improper address has been detected by the ROM/RAM logic.
J12	$\overline{\text{CPU FAULT}}$	output	CPU fault—Indicates that a hardware error has been detected.
L14	$\overline{\text{M ERR TRAP}}$	input	Memory error trap—Indicates that the M chip has detected an error condition.
M14,A9,A10,B10, A11,C10,A12,B11, A13,B12,A14,B13, C13,B14,D13,C14, E13,D14,F12,E14, F13,F14,H14,P8, N9,P9,M9,P11, P10,P12,N12,M12, P13,M13,N13,L12, P14,L13,N14,K12	MIB<39:00>	input/output	Microinstruction bus—Transfers microaddress, microinstructions, and interchip status.
J13	$\overline{\text{CS OE}}$	output	Control store output enable—Controls the state of the control store output buffers.
K14	$\overline{\text{CS WE}}$	output	Control store write enable—Controls the writing of information into the RAM or CAM of the control store memory.
M14	MIB PAR	input/output	MIB parity—Indicates the odd parity of the information on the MIB<39:00> lines.

Pin	Signal	Input/output	Description/Function
P7	IID	output	Initial instruction decode—Indicates the start of a macroinstruction to an F chip.
P6	FBOX N	input	F chip sign N flag—Indicates that the result of an F chip computation is negative.
P5	FBOX Z	input	F chip sign Z flag—Indicates that the result of an F chip computation is zero.
A2,B4,A3,B5, A4,B6,A5	PAL<6:0>	output	Physical address lines—Transfer part of the DAL line address information to the M chip.
A0	$\overline{\text{ABORT}}$	output	Abort—Indicates that the current micro-instruction should be ignored.
A6	MTB MISS	output	Mini translation buffer miss—Indicates that a mini translation buffer lookup has failed.
G3	CLK	input	Clock—A MOS clock signal from the M chip.
F3	CLK BAR	input	Complemented MOS clock—A complementary CLK signal.
G14	CLK SYNC	input	Clock synchronization—Provides a phase 2 reference marker during a cycle.
N6	$\overline{\text{TRISTATE}}$	input	Tristate—Causes the output buffers to become a high impedance.
J14	$\overline{\text{DCLO}}$	input	dc low—Indicates that the dc power is not within the required specifications.
A1	V_{BB}	input	Voltage back-bias—Power supply voltage for the substrate of the chip. (-3.0 Vdc nominally)
B9,C3,C6,C9, C11,C12,D3,G2, H12,H13,J3,M5, M7,M11,N5,N7, N11	V_{DD}	input	Voltage—Power supply voltage.
B8,C4,C5,C8, D12,E12,F2,G12, G13,H3,M4,M6, M8,M10,N4,N8, N10	V_{SS}	input	Ground—Common ground reference.

Data Address Lines (DAL<31:00>—Bidirectional lines used to transmit address and data to and from the V-11 chipset, the port controller, and the cache and backup translation buffer (BTB) RAMs. The I/E chip transfers address information during the first half of the the microcycle. Data is transmitted and received by the DAL lines. The data is valid only during phase P7 when the data is received from the cache or BTB RAMs. When the data is received from other sources, it is valid during phases P7 and P8.

DAL Busy ($\overline{\text{DAL BUSY}}$)—This input is asserted when the port controller has control of the DAL<31:00> lines. When asserted, the DAL<31:00> outputs become a high impedance. The $\overline{\text{DAL BUSY}}$ signal is asserted during cache fill cycles after the port controller has accepted the command sent to it by the M chip. It is held by the port controller until the last longword of the cache fill has been received. The port controller also asserts the $\overline{\text{DAL BUSY}}$ signal during I/O invalidate cycles to hold the DAL lines while the controller transfers the invalid address to the M chip.

DAL Stall ($\overline{\text{DAL STALL}}$)—This input is asserted when a data source cannot respond to a data transfer request during the same cycle. The source that requested the data is stalled until the responder can respond. The $\overline{\text{DAL STALL}}$ signal must be deasserted by the end of phase P4 if it is to remain deasserted for the rest of the cycle.

Cache Miss ($\overline{\text{CMISS}}$)—This signal is asserted by the M chip during memory request write transactions, memory request read transactions that have cache misses, IB-fill transactions that have cache misses, load PTE operations, and Mini-TB miss cycles that result in a miss in the BTB. This input is used by the I/E chip only to indicate that a backup TB miss has occurred.

Content Addressable Memory Match ($\overline{\text{CAM MATCH}}$)—Used to detect patched ROM locations during a ROM access. This signal is asserted when the CAM in the DC327 control store detects an address with a patched microinstruction. The execution of the faulty microinstruction is suppressed by the $\overline{\text{ABORT}}$ signal.

CPU Fault ($\overline{\text{CPU FAULT}}$)—This line indicates that a hardware error has occurred and is used for driving an LED in the field. Refer to Microsequencer section for detailed information.

M Error Trap ($\overline{\text{M ERR TRAP}}$)—This line is asserted by the M chip to inform the I/E chip of an error. The assertion of this signal causes the I/E chip to abort the microtrap operation. It is asserted when a parity error occurs on the data read operation from one of the tag arrays during a memory request (MEM REQ), IB-fill, or Mini-TB miss operation. It is also asserted when the port controller asserts a port control error due to a data error that has occurred during a memory read or write operation.

Microinstruction Bus (MIB<38:00>)—This is the primary control bus. Internal status bits and microaddresses are transferred on this bus in phases P3 and P4 during the address half of the cycle. The DC327 control store drives the microinstructions on the MIB<38:00> lines in phases P7 and P8 during the last half of the cycle.

Control Store Output Enable ($\overline{\text{CSOE}}$)—Asserted to cause the output buffers of the control store to become a high impedance during phases P6 through P8.

Control Store Write Enable (CSWE)—Controls the writing of the 40-bit word to the RAM or CAM in the control store.

MIB Parity (MIB PAR)—This line indicates odd parity for the 39-bit control word during phases P7 and P8. It is not used during phases P3 and P4. When a parity error is detected, the I/E chip forces a microtrap.

Initial Instruction Decode (IID)—Indicates to the F chip that a new macroinstruction execution is beginning and that the opcode on lines MIB<22:15> is valid.

F Chip N Bit (FBOX N)—Indicates that the result of an F chip computation is negative during phases P7 and P8. When no F chip is present, a pullup resistor must be connected to this input. The F chip also indicates an error by asserting the N bit during phases P3 and P4.

F Chip Z Bit (FBOX Z)—Indicates that the result of an F chip computation is zero during phases P7 and P8.

Physical Address Lines (PAL<6:0>)—These lines provide the M chip with part of the address earlier than the address is available on the DAL<38:00> lines. When the MTB MISS signal is asserted, the value on these lines is a virtual address. If the MTB MISS signal is not asserted, the address on the DAL<38:00> lines is a physical address. The mapping of the virtual and physical address of the DAL lines to the PAL<6:0> lines is shown in Table 2.

Table 2 • DC328 Physical Address Line Virtual and Physical Address Correlation

Virtual Address	DAL Line<31>	<30:17>	<16:11>	<10:00>
	PAL Line	<0>		<6:1>
Physical Address	DAL Line	<28:13>	<12:06>	<05:00>
	PAL Line		<6:0>	

Abort ($\overline{\text{ABORT}}$)—This signal indicates that the current microinstruction should be ignored. This inhibits execution of the current microinstruction and forces the loading of the next microinstruction.

Mini-TB Miss (MTB MISS)—This output is asserted during memory requests and IB-fill operations when a Mini-TB lookup fails. It indicates that a Backup TB read cycle should be performed, a PTE should be passed from the BTB to the I/E chip, and the microinstruction or I box request should be initiated again.

Clock (CLK)—This signal is driven by the M chip and is used by the I/E chip, M chip, F chip, and port controller chips. The CLK input makes a transition from a 0 to 1 at the beginning of every odd phase. The CLK frequency is one half the frequency of the TTL CLK IN input to the M chip that is used to generate the CLK and $\overline{\text{CLK BAR}}$ signals. The nominal value for TTL CLK IN input is 40 MHz resuting in a 20-MHz CLK input.

Clock Bar (CLK BAR)—This signal is used by the I/E chip, M chip, F chip, and port controller chips. The CLK BAR signal makes a transition from a 0 to 1 at the beginning of every even phase. The frequency of this signal is one-half the frequency of the TTL CLK IN signal. The nominal value for TTL CLK IN is 40 MHz, thus producing a 20-MHz $\overline{\text{CLK BAR}}$ input.

Clock Synchronize (CLK SYNC)—This signal is from the M chip and is used to synchronize the chip set and port controller by supplying the reference marker for phase 2 in the cycle.

dc Low ($\overline{\text{DCLO}}$)—This signal is driven by the module to indicate that the dc power is not within specifications or is being restored.

Three State (TRISTATE)—This signal forces all of the output buffers to their high-impedance state. This feature is used only during parametric characterization, debugging operations, and for module test purposes.

Back-bias Voltage (V_{BB})—A –3.0 V from an external supply used for substrate bias.

Voltage (V_{DD})—Power supply 5.0 V (nominal) voltage.

Ground (GND<)—Ground reference.

I/E Chip Timing

The I/E chip microcycle is divided into eight time slots P1 through P8 shown in Figure 3. Two nonoverlapping clocks and a synchronization strobe are required to generate these phases. The eight time slots can be divided into five functional time slots—fetch, decode, drive the operands onto the input buses, ALU operation, and write the results. The I/E chip is pipelined such that the fetch and write cycles are overlapped. The overlap of microinstructions U1 and U2 is shown.

MICRO CYCLE	P1	P2	P3	P4	P5	P6	P7	P8
WRITE	DECODE U1		DRIVE U1 OPERANDS		U1 ALU OPERATION		WRITE U1 RESULTS	
FETCH	GET U2 MICROADDRESS		LOOKUP U2 IN CONTROL STORE				LOAD U2	

Figure 3 • DC328 Microcycle Timing

The I/E chip is interfaced with the system through the time multiplexed microinstruction bus (MIB<39:00>) and the data address lines (DAL<31:00>). The address is transferred during the first half of the cycle (P3 and P4) and the data is transmitted or received in the second half of the cycle (P7 and P8). Figure 4 shows the MIB and DAL timing.

MICRO CYCLE	P1	P2	P3	P4	P5	P6	P7	P8
MIB AND DAL BUS		DRIVE ADDRESS	ADDRESS VALID			DRIVE DATA	DATA VALID	

Figure 4 • DC328 MIB<39:00> and DAL<31:00> Timing

Functional Description

Figure 1 shows the functional elements, bus structure, and signal lines of the I/E chip. Refer to DC328 I/E Chip Functional Specification for detailed operation of this chip. A brief description of the I/E chip follows:

Instruction Prefetcher and Decoder (I Box)—The I box prefetches the instruction stream, generates microprogram fork addresses, and provides instruction data to the E box. The instruction programmable logic array (IPLA) is located in the I box.

Microsequencer—The microsequencer determines the address of the next microinstruction to be fetched and executed from the external control store ROM/RAM. Its facilities included an eight-entry stack, and two adders for fast next-address generation in conditional branches. It also has special hardware used to enhance the power of microdiagnostics and handles microtrap generation.

Execution Box (E Box)—The E box contains facilities to implement the VAX instruction set. The E box contains the general purpose registers (GPRs), temporary registers (TEMPs), and working registers. It also includes a shifter, a 32-bit arithmetic logic unit (ALU), a constant generator (KMUX), and the shift counter (SC) register.

Memory Interface—The memory interface converts program generated virtual addresses to physical addresses and governs all data requests of main memory or the cache. It also has facilities to improve the performance of common memory management exceptions.

DAL Interface—The DAL interface connects the I/E chip to the external data path. It performs byte rotation, buffering, and determines when the data on the DAL<31:00> lines is valid.

MIB Interface—The MIB interface connects the I/E chip to the external control store through the MIB. It receives and checks the parity of the microcode words that are read from the external control store, controls the sequencing of the ROM/RAM chips, and oversees the use of the MIB for passing interchip status signals. It is also used for writing the patch RAM section of the DC327 ROM/RAM chips.

Internal Bus Structure

Five of the internal I/E chip buses are 32-bit data buses and the remaining two are used for control. The following is a brief description of the major buses.

Internal Memory Information (IMIB<38:00>)—The IMIB<38:00> lines provide the main control and are used to transfer the microinstruction to all other logic functions in the chip. Because the I/E chip cannot stall the clocks, it must execute a microinstruction every cycle. The information on this bus is valid every cycle. The memory interface logic inhibits write operations that are associated with a stalled microinstruction.

Microtest<3:0>—The microtest bus is used to transfer the results of a conditional branch test to the microsequencer. The microsequencer uses this value and the branch command field to determine the next microaddress to be driven to the control store.

Internal Data/Address (IDAL<31:00>)—The IDAL bus connects the DAL<31:00> lines to the internal register file. It is driven by the memory interface during the address portion of the microcycle and during the data half of the microcycle if the data is being written by the I/E chip. It is driven by the DAL<31:00> lines if the I/E chip is executing a read transaction.

D-stream Bus (DBUS<31:00>—This bus connects the E box registers to the IDAL bus and is driven when the I/E chip reads D stream data or external registers. The microinstruction specifies the E box register to be loaded from this bus.

AW Bus (AWBUS<31:00>—This is the main data bus between the I box, E box, and memory interface and is used as one of the input buses to the ALU. It also transfers the result of the computation from the ALU. During a normal ALU operation, one of the operands (inputs to the ALU) is driven onto the AW bus by an E box register. After the ALU operation, the results are transferred onto this bus and are then written into an E box register. When data is being written from the E box to the DAL lines, the AW bus is used to transfer the data to the memory interface that then drives the DAL lines. It is also used to connect registers located in the memory interface and I box to the E box registers.

B Bus (B<31:00>)—This bus is located in the E box and is driven by E box registers to provide the remaining input to the ALU.

Specifications

The mechanical, electrical, and environmental specifications for the DC328 are contained in the following paragraphs. The test conditions for these specifications, unless specified otherwise, are as follows:

- Ambient temperature (T_A): 0°C to 70°C
- Supply voltage (V_{DD}): 5.0 V ±5% (maximum ripple 200 mV peak-to-peak)
- Back-bias voltage (V_{BB}): –3.0 V ±15% (maximum ripple 200 mV peak-to-peak)

Mechanical Configuration

The mechanical dimensions for mounting the DC328 132-pin PGA package are shown in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to a device. Exposure to the absolute maximum rating conditions for extended periods may adversely affect the reliability of the device.

- Supply voltage (V_{DD}): –0.5 V to 7.0 V
- Pin voltages: –1.0 V to 10 V
- Power dissipation (T_A=0°C): 7.5 W
- Power dissipation (T_A=70°C): 5.0 W
- Ambient temperature operating range (T_A): 0°C to 70°C
- Storage temperature range: –55°C to 125°C

Recommended Operating Conditions

- Supply voltage (V_{CC}): 5.0 V ±5%
- Ambient temperature operating range (T_A): 0°C to 70°C

ac and dc Electrical Characteristics

Refer to the DC328 I/E Chip Functional Specification for the dc input and output parameters and ac timing parameters.

• Features

- Contains memory management logic
- Provides tag store for cache memory and backup translation buffer
- Provides functions for interrupts, communications, and timing

• Description

The DC329 V-11 processor memory management (M chip) logic, contained in a 132-pin, pin grid array (PGA) package, includes most of the memory management hardware and the tag store for the cache memory and for a 512-entry virtual-to-physical address backup translation buffer (TB BTB0). The BTB supplements the mini-TB that is located in the I/E chip. The cache data and the BTB address translation entries are stored in external static RAM chips. The M chip also contains miscellaneous CPU functions such as CPU clock drivers, processor registers, the interval and time-of-day timer registers, interrupt hardware, clock-generation circuits, and four serial-line units. Figure 1 is a block diagram of the DC329 M chip.

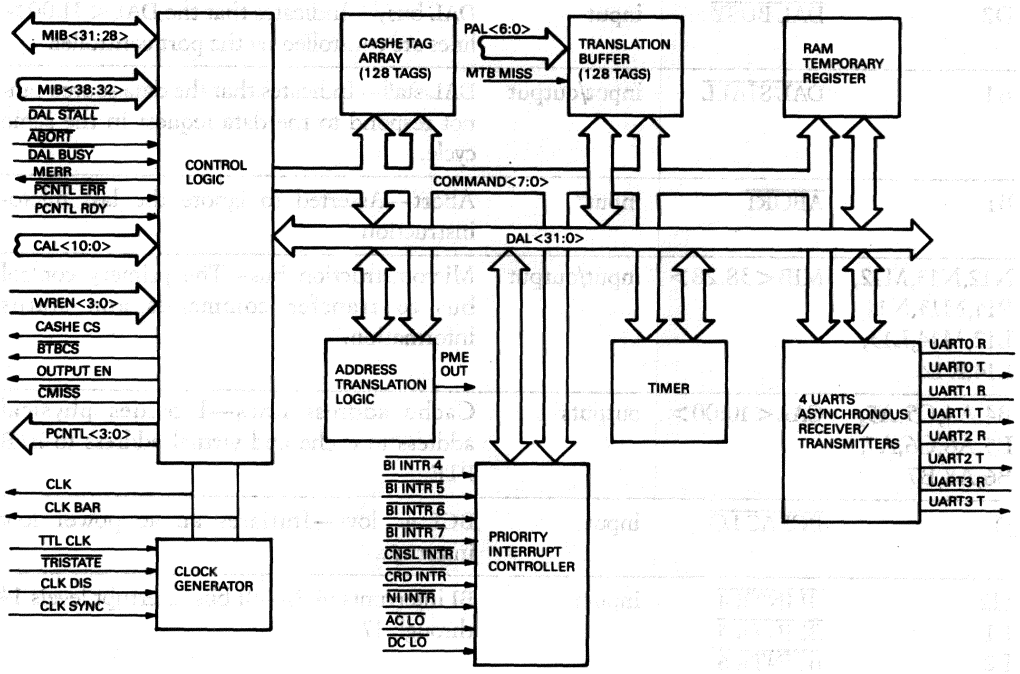


Figure 1 • DC329 Memory Management Logic Block Diagram

• Pin and Signal Descriptions

The input and output signals and power and ground connections for the 132-pin package are summarized in Table 1. The table also contains the physical pin locations that are shown in Figure 2. The following paragraphs provide a more detailed description of the inputs and outputs of the M chip.

Table 1 • DC329 Pin and Signal Summary

Pin	Signal	Input/output*	Description/Function
K13,K14,G13, J14,G12,H14, F13,G14,F12, F14,E13,E14, E12,D14,D13, C14,C13,B13, B14,B12,A14, B11,A13,B10, A12,C9,A11,B9, A10,C8,A9,B8	DAL<31:00>	input/output	Data/address lines—Transfers data and address information to and from the V-11 chipset.
D2	$\overline{\text{DAL BUSY}}$	input	DAL busy—Indicates that the DAL<31:00> lines are controlled by the port controller.
A1	$\overline{\text{DAL STALL}}$	input/output	DAL stall—Indicates that the data source cannot respond to the data request in the same cycle.
B1	$\overline{\text{ABORT}}$	input	Abort—Asserted to ignore the last microinstruction.
N12,N13,M12, P14,M13,N14, L12,M14,L13, L14,K12	MIB<38:28>	input/output	Microinstruction bus—The primary control bus to transfer commands and status information.
B4,A4,C5,A5, B5,A6,C6,A7, B6,A8,B7	CAL<10:00>	outputs	Cache address lines—Provides physical address to cache and virtual address to BTB PTE.
J1	$\overline{\text{BCI AC LO}}$	input	BCI ac low—Initiates an ac power low interrupt.
H2 L1 L2 L7	$\overline{\text{BI INTR 4}}$ $\overline{\text{BI INTR 5}}$ $\overline{\text{BI INTR 6}}$ $\overline{\text{BI INTR 7}}$	inputs	BI interrupts (4-7)—BI bus interrupt levels 14 through 17
A2	$\overline{\text{BTB CS}}$	output	BTB chip select—Asserted to transfer data to or from the BTB PTE RAMs.

Pin	Signal	Input/output*	Description/Function
B3	$\overline{\text{CACHE CS}}$	output	Cache chip select—Asserted to transfer data to or from the cache data RAMs.
M9	CLK	input/output	Clock—A MOS clock signal to the I/E chip, F chip, and port controllers.
M10	CLK BAR	input/output	Clock bar—A complement MOS CLK signal to the I/E chip, F chip, and port controllers.
N10	CLK SYNC	input/output	Clock synchronization—Provides Phase 2 reference for synchronization.
B2	$\overline{\text{CMISS}}$	input/output	Cache miss—Asserted during cache misses, load PTE operations, and cycles that miss in the MTB.
J2	$\overline{\text{CNSL INTR}}$	input	Console interrupt—Asserted to indicate an unmaskable interrupt.
K3	$\overline{\text{CRD INTR}}$	input	Corrected read data interrupt—Asserted to request an interrupt on IPL 16.
M5,N4,M4,N3	WR EN<3:0>	inputs	BTB/cache write enable—
P10	$\overline{\text{BCI DCLO}}$	input	BCI dc low—Indicates that the power supply voltage will be below the specified minimum value.
D1	MERR	output	M chip error—Informs the I/E chip of an error condition.
P9	MTB MISS	input	Mini-TB miss—Indicates whether a cache or BTB access is required when a mini-TB miss has occurred.
K2	$\overline{\text{NI INTR}}$	input	NI interrupt—Indicates that an unmaskable interrupt is posted.
A3	$\overline{\text{OUTPUT EN}}$	output	Output enable—Controls the output enable of the cache and BTB.
P3-P1,N1	PCMD<4:0>	output	Port controller command—Provides commands to the port controller.
C2	$\overline{\text{PCNTL ERR}}$	input	Port controller error—Indicates an error has occurred in the port controller during a data transfer.
C1	PCNTL RDY	input	Port controller ready—Indicates that a command can be sent to the port controller.
P7,N6,P6,M6,	PAL<6:0>	input	Physical address<6:0>—Provides a physical or virtual address to the M chip.

Pin	Signal	Input/output*	Description/Function
M2	PME OUT	output	Performance monitor enable—Indicates the state of the PME bit in the P1LR.
N9	TTL CLK IN	input	TTL clock—A TTL 20-MHz clock signal for UARTs and timers.
P8	$\overline{\text{CLK DIS}}$	input	Clock disable—Disables the clock outputs.
M1	$\overline{\text{TRISTATE}}$	input	Three state—Disables all outputs of the M chip except for the CLK, $\overline{\text{CLK BAR}}$, and CLK SYNC.
G3	UART0 RCV	input	UART0 receive—Serial-line input from UART0.
F2	UART0 XMIT	output	UART0 transmit—Serial-line output to UART0.
H1	UART1 RCV	input	UART1 receive—Serial-line input from UART1.
F1	UART1 XMIT	input	UART1 transmit—Serial-line output to UART1.
G2	UART2 RCV	input	UART2 receive—Serial-line input from UART2.
E2	UART2 XMIT	input	UART2 transmit—Serial-line output to UART2.
G1	UART3 RCV	input	UART3 receive—Serial-line input from UART3.
E2	UART3 XMIT	input	UART3 transmit—Serial-line output to UART3.
C3,C4,C10,C11, H3,H12,J12,M3, M7,M8,N8	V_{DD}	input	Voltage—Power supply voltage.
C7,C12,D3,D12, E3,H13,J12,J13, L3,M11,N7,N11	V_{SS}	input	Ground—Common ground reference.
N2	V_{BB}	input	Back-bias voltage—Power supply back-bias voltage.

*All pins have TTL compatible levels except CLK and $\overline{\text{CLK BAR}}$.

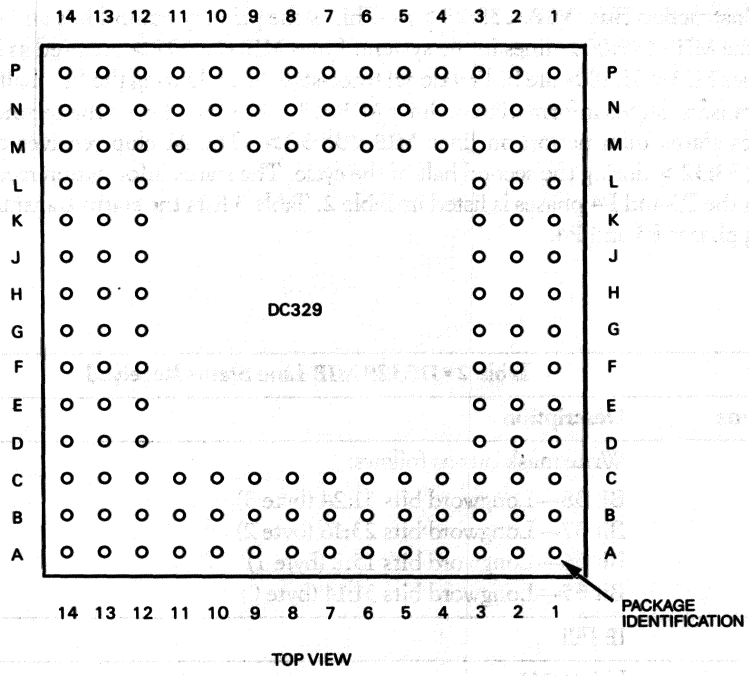


Figure 2 ▪ DC329 Pin Assignments

Data/address Lines (DAL<31:00>)—These lines are used for transmitting addresses and data to and from the V-11 chipset, the port controller, and the cache RAMS. The M chip receives address and data information and transmits data on this bus. The data is valid only during phase 7 when the data is from the cache data RAMs. When the data is from the M chip, it is valid for phases P7 and P8.

DAL Busy ($\overline{\text{DAL BUSY}}$)—This input is asserted when the port controller has control of the DAL<31:0> lines. When asserted, the DAL lines are a high impedance and the $\overline{\text{DAL STALL}}$ and $\overline{\text{CMISS}}$ are released by the M chip. It is asserted during cache fill cycles after the port controller has accepted the command sent to it by the M chip. It is held by the port controller until the last longword of the cache fill operation has been received. The port controller also asserts this input during I/O invalidate cycles to enable it to drive the invalidate address onto the DAL<31:0> lines.

DAL Stall ($\overline{\text{DAL STALL}}$)—This signal is asserted when a data source cannot respond to a data transfer request in the same cycle. The resource that requested the data is stalled until the source can respond. The M chip asserts this signal when the cache or backup translation buffer is busy with a previous command and a new command is received which requests the M chip. It is used during the first ready cycle of an MREQ read transaction that has a cache miss, and when there is a read miss or a cache write and the port controller is not ready. The M chip also asserts the $\overline{\text{DAL STALL}}$ signal when the I/E chip attempts to read the PTE adder and the M chip is calculating the result. It is also used to stall the I/E chip during the first cycle of the two cycle MXPR read operations and during the second cycle of the two cycle MXPR write operations if the microcode is attempting to initiate another operation that requires the M chip. The M chip does not assert this signal when the $\overline{\text{DAL BUSY}}$ input is asserted. The $\overline{\text{DAL STALL}}$ signal must be deasserted by the end of phase P4 if it is to remain deasserted for the rest of the cycle.

Microinstruction Bus (MIB<38:28>)—This is the primary control bus and the M chip receives 12 of the MIB<39:00> lines in the system. Lines MIB<38:32> are used as input to the M chip and lines MIB<31:28> are bidirectional three-state lines. During the first half of the cycle, the M chip transfers status information on lines MIB<31:28> and during the second half of the cycle it receives status information on lines MIB<38:32>. The M chip receives commands on lines MIB<38:32> during the second half of the cycle. The status information received by the M chip during the P3 and P4 phases is listed in Table 2. Table 3 lists the status transmitted by the M chip during phases P3 and P4.

Table 2 • DC329 MIB Line Status Received

MIB Line	Description
38-35	Write mask bits as follows: Bit 38—Longword bits 31:24 (byte 3) Bit 37—Longword bits 23:16 (byte 2) Bit 36—Longword bits 15:8 (byte 1) Bit 35—Longword bits 31:14 (byte 0)
34	IB Fill
33	Using DAL
32	Request second reference

The write mask (bits 38:35) indicate to the M chip which longword bytes are to be written into the cache and main memory or to the I/O device. The M chip uses these bits to enable the cache write enables if the cache hits on a MEM REQ write instruction. Each bit corresponds to the associated cache write enable output. These bits are asserted to 1 and are supplied to the port controller on the PCMD4:0 outputs. These bits must be set whenever the microcycle is not a MREQ write instruction and low when asserted on the MIB asserted implies that the byte is to be written.

The IB Fill MIB34 line is asserted low when the M chip is not completing a previous operation and initiates a read operation to read the requested VAX instruction located at the address present on the DAL lines during the address half of the DAL cycle.

The Using DAL MIB33 line is asserted high to indicate to the M chip that the current microinstruction uses the DAL lines. This bit is used in determining when the current microinstruction has been completed.

The Request 2nd reference MIB32 is asserted low to indicate to the M chip that the reference is unaligned and that two MREQ operations are required to transfer all of the data. This applies to unaligned read and unaligned write transactions. It is asserted for the first unaligned data reference and prevents the M-chip from updating the microinstruction that would cause it to reexecute the memory request (MREQ). The updated address is supplied by the I/E chip.

Table 3 • DC329 MIB Line Status Transmitted

MIB Line	Description
31	Length violation
30	System space
29	IID interrupt request
28	FPD interrupt request

The length violation MIB31 output reports the comparison of the values contained in a selected process or system length register with the MVA value. If the MVA changes in microcycle (*n*), the first microcycle that can branch on the new status is (*n* + 4).

The system space MIB30 output is equivalent to the value of MVA31.

The IID interrupt request MIB29 output is the interrupt request generated by the interrupt controller. It is asserted high when there is a pending maskable interrupt input that has a value above the current IPL, when there is an unmaskable interrupt pending, or when the ISTATUS HALT bit is set. Writing to the SISR, the PSL, or the ISTATUS can cause a change in the interrupt status on the MIB. This is in addition to the internal and external inputs that can cause interrupts. There is a delay between changing one of these registers and the result of that change on the IID input.

The FPD interrupt request MIB28 output is the interrupt request generated by the interrupt controller. It is asserted high when there is a pending maskable interrupt input that has a value above the current IPL or when an unmaskable interrupt is asserted. The timing is the same as for the IID interrupt request status signal.

The microinstruction formats that are recognized on the MIB during phases P7 and P8 are shown in Figure 3.

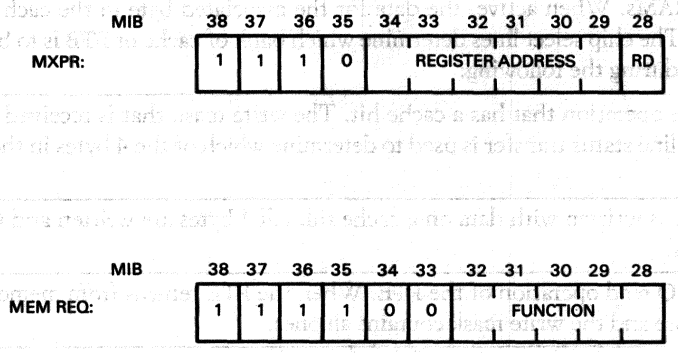


Figure 3 • DC329 MIB Microinstruction Format

Abort ($\overline{\text{ABORT}}$)—This input is asserted when the last instruction on the MIB in the previous P7 and P8 phase is to be ignored. Because the $\overline{\text{ABORT}}$ signal is generated by the I/E chip, the M chip may not be executing the same microinstruction as the I/E chip and therefore it applies to the instruction that the I/E chip is executing. When this occurs, the M chip ignores the abort. If the M chip asserts the MERR signal, the $\overline{\text{ABORT}}$ signal will be asserted by the I/E chip in a subsequent cycle, causing a microtrap to occur. This $\overline{\text{ABORT}}$ signal must be deasserted by the end of phase 3 if it is to remain deasserted for the rest of the cycle.

Micro Error (M ERR)—This signal is asserted by the M chip to inform the I/E chip of an error condition. It causes the I/E chip to abort and microtrap when there is a parity error on data read from one of the tag arrays in an MREQ, IB fill, or MTB miss operation and when the port controller asserts a port control error due to a data error on an MREQ read or write operation.

BCI ac Low ($\overline{\text{BCI ACLO}}$)—This is an interrupt input to the M-chip. When asserted, an interrupt is requested at IPL 16. This signal is sampled each cycle and is not latched by the M chip.

Port Controller Error ($\overline{\text{PCNTL ERR}}$)—This input informs the M chip that an error has occurred in the port controller as a result of a data transfer operation.

Port Controller Ready ($\overline{\text{PCNTL READY}}$)—This input informs the M chip that the port controller is ready to accept a command. When deasserted and the current instruction requires the port controller, the M chip will assert the $\overline{\text{DAL STALL}}$ signal until the port controller becomes ready.

Cache Address Lines (CAL<10:00>)—These outputs provide the physical address to the cache during cache read and write cycles and the virtual address to the BTB during PTE read and write cycles. The cache data array is organized into 2K longwords that require 11 address lines. The BTB contains 512 PTEs of one longword each. The CAL<10:00> outputs reflect the address sent to the M chip on the DAL<31:00> lines during the first cycle of cache read operation and MTB miss cycles. The source of the CAL<10:00> information is the MVA for the load PTE operation. During cache fill operations, the M chip modifies the value of the CAL<1:00> lines by incrementing them by modulo four (starting from the value of these bits supplied on the DAL lines) each time one of the four longwords are received. The CAL line value is derived from the address received on the DAL<31:00> lines or from the MVA.

Cache Chip Select ($\overline{\text{CACHE CS}}$)—This signal is asserted when data is to be transferred to or from the cache data RAMs during read and write operations. The direction of transfer is specified by the $\overline{\text{WREN<3:0>}}$ lines. The RAMs address is on the CAL<10:00> lines.

Write Enable ($\overline{\text{WREN<3:0>}}$)—These lines specify the direction of the data to the cache and BTB data and parity RAMs. When active, the data for the associated byte in the cache or BTB RAM arrays is written. The chip select lines determine which bank of cache or BTB is to be written. The lines are asserted during the following:

- An MREQ write operation that has a cache hit. The write mask that is received by the M chip during the MIB line status transfer is used to determine which of the 4 bytes in the longword are to be written.
- When the cache is written with data on a cache fill. All 4 bytes are written and the write mask contains all ones.
- During an MREQ read operation of the PTE. When the PTE returns from memory, it is loaded into the PTE store and the write mask contains all ones.
- During an MXPR write operation to the PTE. All 4 bytes are written and the write mask contains all ones.

BTB Chip Select ($\overline{\text{BTB CS}}$)—This signal is asserted when data is to be transferred to or from the BTB PTE RAMs during read and write operations. The direction of transfer is specified by the $\overline{\text{WREN}}$<3:0 lines. The RAM address is on the CAL<10:00> lines.

Output Enable ($\overline{\text{OUTPUT EN}}$)—This output is used to enable the cache and BTB data chips. When asserted, the selected bank of data RAMs will drive the DAL lines.

Cache Miss ($\overline{\text{CMISS}}$)—This signal is an output from the M chip when the $\overline{\text{DAL BUSY}}$ signal is not asserted and is an input to the M chip when the $\overline{\text{DAL BUSY}}$ signal is asserted. It is asserted on MREQ write and read operations that result in cache misses, IB fills transactions that result in cache misses, load PTE operations, and MTB miss cycles that result in a miss in the BTB. When asserted by the M chip and the MTB miss is not true, this signal indicates to the port controller that the command that was sent to it is to be executed. The $\overline{\text{CMISS}}$ signal is deasserted by the end of phase 4 if it is to remain deasserted for the rest of the cycle. During cache read misses and load PTE operations, the $\overline{\text{CMISS}}$ signal is asserted by the M chip for each cycle up to and including the first cycle in which the port controller is ready. The port controller then asserts the $\overline{\text{CMISS}}$ signal until the completion of the fill operation. The M chip also waits for the port controller to become ready during write cycles. The $\overline{\text{CMISS}}$ input is used to indicate that the M chip is to perform an I/O invalidate cycle. The port controller does this by deasserting this signal at the same time that the $\overline{\text{DAL STALL}}$ and $\overline{\text{DAL BUSY}}$ signals are asserted.

Port Command (PCMD<4:0>)—These outputs provide the command to the port controller during phases P3 and P4 and the write mask during phases P7 and P8. The command is executed only by the port controller if the $\overline{\text{CMISS}}$ is asserted later in that cycle. The write mask is used by only by the port controller during write operations.

Clock (CLK)—The CLK signal is a MOS level input. The M chip drives this output back to itself as an input and it is used on the module by the I/E chip, F chip, and port controller chips. A transition of this signal from a 0 to 1 occurs at the beginning of every odd phase. The frequency of this clock is one-half that of the TTL CLK IN signal. It can be used as an input for test purposes by asserting the $\overline{\text{CKL DIS}}$ signal and driving the CLK input from an external source.

Clock (REF CLK)—The REF CLK input is the 20-MHz clock reference for the time-of-day timer, interval counter registers, and UARTs.

Clock Bar (CLOCK BAR)—The CLK BAR signal is a MOS level input. The M chip drives this output back to itself as an input and it is used on the module by the I/E chip, F chip, and port controller chips. A transition of this signal from a 0 to 1 occurs at the beginning of every even phase. The frequency of this clock is one-half that of the TTL CLK IN signal. It can be used as an input for test purposes by asserting the $\overline{\text{TRISTATE}}$ signal and by driving CLK BAR input from an external source.

Clock Synchronize (CLK SYNC)—This signal is transferred to the V-11 chip set to synchronize the module by suppling the reference for phase 2 in the cycle. It can be used for test purposes by asserting the $\overline{\text{CLK DIS}}$ input and by driving this input from an external source.

Clock Disable ($\overline{\text{CLK DIS}}$)—When this input is asserted, all M chip outputs except for the CLK, CLK BAR, and CLK SYNC outputs are enabled. This input contains an internal pullup resistor for the normal operation state.

Three State ($\overline{\text{TRISTATE}}$)—When asserted, this input causes the output drivers to become a high impedance. This input contains an internal pullup resistor to hold it deasserted during normal operation.

Physical Address Lines (PAL<6:0>)—These inputs provide part of the address that was previously referenced on the DAL lines. Table 4 lists the address bit correlation for a virtual address (MTB MISS asserted) and for a physical address (MTB MISS not asserted).

Table 4 • DC329 PAL Line Virtual and Physical Address Correlation

Virtual Address	DAL Line	31	<16:11>
	PAL Line	0	<6:1>
Physical Address	DAL Line	<11:6>	
	PAL Line	<6:0>	

Mini-TB Miss (MTB MISS)—This input is asserted by the I/E chip when there is a mini-TB miss. The M chip uses this signal to indicate whether the BTB or the cache is to be accessed during an MREQ operation. It is a qualifier on the address received on the DAL and PAL<6:0> lines that indicates whether the address is virtual or physical. When asserted, the M chip transfers the virtual address onto the CAL<10:0> lines and reexecutes the last microinstruction.

BI Interrupt ($\overline{\text{BI INTR}}$<4:0>)—These are interrupt inputs to the M chip. When a line is asserted, an interrupt is requested on IPL 14 through IPL 17. These signals are sampled during each cycle and are not latched by the M chip.

Console Interrupt ($\overline{\text{CNSL INTR}}$)—When asserted, an unmaskable interrupt is posted. This signal is sampled each cycle and it is not latched by the M chip.

CRD Interrupt ($\overline{\text{CRD INTR}}$)—When asserted, an interrupt is requested at IPL 16. This signal is sampled each cycle and it is not latched by the M chip.

NI Interrupt ($\overline{\text{NI INTR}}$)—When asserted, an unmaskable interrupt is posted. This input is sampled each cycle and is not latched by the M chip.

ac Low ($\overline{\text{BCI ACLO}}$)—This input is asserted when the ac voltage is below the specified limit and initiates an interrupt request on IPL 16. This input is sampled each cycle.

dc Low ($\overline{\text{BCI DCLO}}$)—This input signal is asserted by the power supply when its output voltage is about to be lower than the specified minimum value. When asserted, all outputs of the M chip become a high impedance except for the CLK, CLK BAR, CLK SYNC, and $\overline{\text{OUTPUT EN}}$ outputs. The CLK and CLK BAR outputs are normal. The CLK SYNC signal and $\overline{\text{OUTPUT EN}}$ signals are deasserted. When the $\overline{\text{BCI DCLO}}$ signal is deasserted and the CLK and CLK BAR signals are asserted, the first assertion of the $\overline{\text{CLK SYNC}}$ signal is within four or six TTL CLK IN cycles. After this signal is deasserted, the M chip will remain in a reset state and the outputs will be a high impedance for the first three cycles of the CLK SYNC output. When the CLK SYNC signal has pulsed a minimum of four times, the M chip will respond to microinstructions on the MIB, and the high-impedance outputs will be enabled.

Voltage (V_{BB})—This input is used to supply the back-bias voltage to the module.

Performance Monitor Enable Output (PME OUT)—This output has the same value as the PME bit in the P1 length register (P1LR). Refer to the Register section for a description of the P1LR. If the state of PME bit changes, the PME OUT signal will be valid during phase P4 of the following two cycles. If the PME bit does not change, the PME OUT signal will be stable for phase P1 through P8.

TTL Clock Input (TTL CLK IN)—The TTL clock input is nominally 40 MHz and is used to generate all clock signals to the V-11 chip set. The M chip provides a 20-MHz MOS level clock pulse to the other V-11 chips by dividing the input by two as the reference.

UART Receive and Transmit (UART0 R–UART3 R and UART0 T–UART3 T)—These lines provide the serial line inputs from the four UARTS and serial-line outputs to the four UARTs. Each input is asynchronous to the reference clock.

Voltage (V_{DD})—Power supply voltage 5 V (nominal)

Ground (V_{SS})—Signal ground reference.

Functional Description

The DC329 contains the functions shown in Figure 1 and is described as follows.

- Address translation logic
- Backup translation buffer tags (BTB)
- Cache tag array
- Clock generator
- Control logic
- Priority interrupt/halt controller
- RAM temporary registers
- Timer logic
- Four UARTS

Address Translation Logic

The address translation logic (ATL) calculates a page-table entry address when the page-table entry necessary to translate a virtual address is not in the mini or backup translation buffers. The ATL contains the following registers:

- P0 base register (P0BR)
- P0 length register (P0LR)
- P1 base register (P1BR)
- P1 length register (P1LR)
- S0 base register (S0BR)
- S0 length register (S0LR)
- PTE translated address (PTE ADR)

Backup Translation Buffer

The backup translation buffer (BTB) is a direct mapped array with 128 entries, grouped into system and process spaces as determined by bit 31 of the virtual address. Each entry contains a tag that corresponds to a group of four page-table entries (PTEs) and four valid bits, one for each PTE. The 512 PTEs are not included in the chip and are typically industry-standard RAMs. This array is used on MREQ or IB FILL operations when the MTB MISS signal is asserted and on the MREQ read PTE operations. It is also used in some MXPR instructions and contains the following registers.

- Memory address register (MAR)
- Missed virtual address register (MVA)
- Invalidate address register (INVAR)
- Refresh address register (REFR)
- BTB tag, valid bits, and parity bits register (BTB ENTRY)
- BTB invalidate register (BTB INV)
- PTE array register (BTB PTE)
- BTB status register (BTB STAT)
- Error status register (ERR STAT)

Cache Tag Array

This section contains 128 direct mapped cache tags each of which is mapped to a 64-byte block of contiguous physical data. Each tag has four valid bits to indicate the validity of each octaword (16 bytes) in the allocated block. The registers in this section are

- Cache tag, valid bits, and parity bits (CACHE ENTRY)
- Cache status register (CACHE STAT)

Control Logic

The control logic is distributed throughout the chip. It receives the control inputs and generates the internal and external control and contains the following registers.

- Indirect MXPR register (INDIR)
- Indirect address value register (INDIR ADDR)
- No-Op diagnostic register (NO DRIVE)

Clock Generator

The clock generator divides the 40-MHz TTL CLK IN input by two and generates the clock sources (CLK, $\overline{\text{CLK BAR}}$, and CLK SYNC). It also receives the CLK, $\overline{\text{CLK BAR}}$, and CLK SYNC inputs so that it can generate the M chip internal phases.

Priority Interrupt Controller

The interrupt section of the M chip receives the internal and external interrupt requests and reports an active interrupt if there is an interrupt whose priority is higher than that of the processor's current priority. The software HALT request in the ISTATUS register is reported with the interrupt requests during the MIB status transaction. It also contains hardware logic to support REI instruction. The registers contained in this section are

- Interrupt priority level (IPL)
- Interrupt status register (ISTATUS)
- Asynchronous system trap level (ASTLVL)
- Processor status longword (PSL)
- Processor status longword temporary register (PSL TEMP)
- Software interrupt status register (SISR)

RAM Temporary Registers

The RAM temporary registers are MTEMP0 through MTEMP17.

Timer Logic

The timer logic consists of the interval counter (IVC) and time-of-day register (TODR). The IVC is used to provide a source of interrupts at a repeatable rate that can be programmed by software in 1 μs steps. The TODR supplies the real-time clock function, reflecting the relative time from its last initialization. The TODR also is used to provide a periodic source of unmaskable interrupts every 1.28 seconds. This logic contains the following registers:

- Interval counter control and status register (ICCS)
- Interval counter value register (ICR)
- Next interval counter value register (NICR)
- Time-of-day register (TODR)
- Time-of-day prescaler register (TODPRE)

Universal Asynchronous Receivers and Transmitters

Each of the four universal asynchronous receivers and transmitters (UART0 through UART3) contain a status register and a data register.

Specifications

The mechanical, electrical, and environmental specifications for the DC329 are contained in the following paragraphs. The test conditions for the parameters specifications, unless specified otherwise, are as follows:

- Ambient temperature (T_A): 0°C to 125°C
- Supply voltage (V_{DD}): 5.0 V ±5% (maximum ripple 200 mV peak-to-peak)
- Back-bias voltage (V_{BB}): −3.0 V ±15% (maximum ripple 200 mV peak-to-peak)

Mechanical Configuration

The mechanical dimensions for mounting the DC329 132-pin CERQUAD package are shown in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to a device. Exposure to the absolue maximum rating conditions for extended periods may adversely affect the reliability of the device.

- Power supply voltage (V_{DD}): –0.5 V to 7.0 V
- Back-bias voltage (V_{BB}): –7 V to 0 V
- All other pin voltages: –1.0 V to 10 V
- Power dissipation (T_A = 0°C): 5.25 W
- Power dissipation (T_A = 70°C): 3.5 W
- Ambient temperature operating range (T_A): 0°C to 70°C
- Storage temperature range: –55°C to 125°C

Recommended Operating Conditions

- Power supply voltage (V_{CC}): 5.0 V ±5%
- Ambient temperature operating range (T_A): 0°C to 70°C

ac and dc Electrical Characteristics

Refer to the *DC329 M Chip Functional Specification* for the dc input and output parameters and ac timing parameters.

• Features

- Contains a fraction processor and exponent and sign processor
- Accelerates a subset of the VAX instructions

• Description

The DC330 V-11 processor floating-point accelerator, contained in a 132-pin grid array (PGA) package, receives opcodes and normalized floating-point operands from the I/E chip, and executes the instructions faster than I/E chip microcode. It is optimized to accelerate a subset of the VAX instruction set. Figure 1 is a block diagram of the DC330 floating-point accelerator.

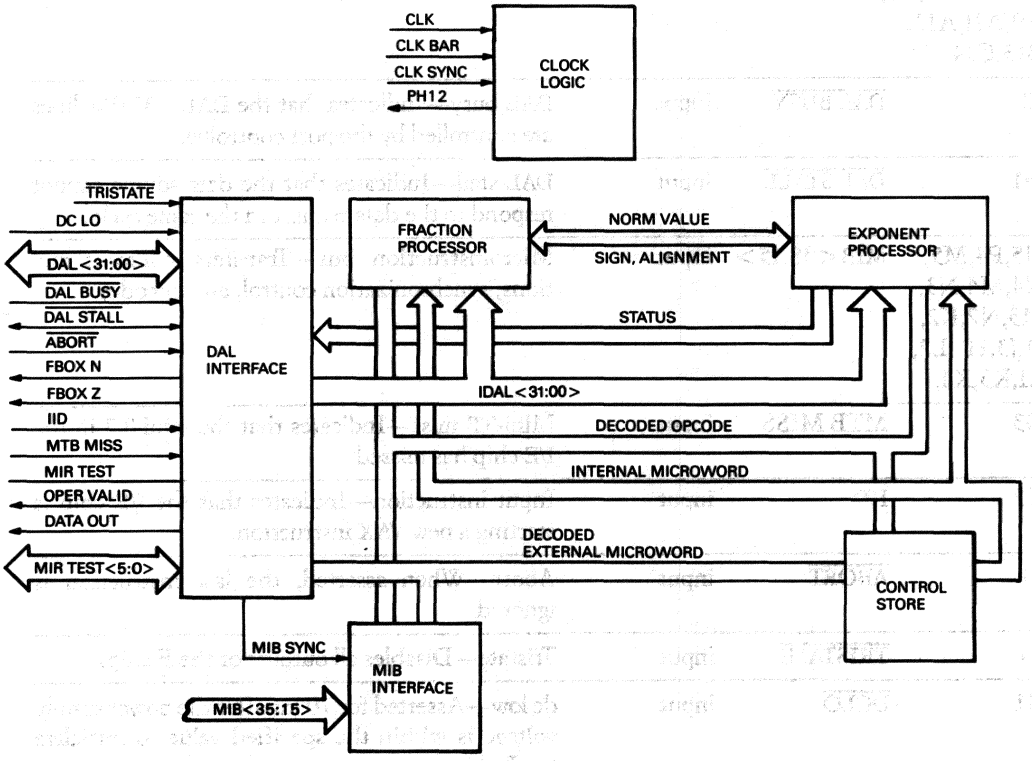


Figure 1 • DC330 Floating-point Accelerator Block Diagram

• Pin and Signal Descriptions

The input and output signals and power and ground connections of the DC330 are shown in Figure 2 and are summarized in Table 1. Refer to the paragraphs that follow for a more detailed description of the signal functions.

Table 1 • DC330 Pin and Signal Summary

Pin	Signal	Input/output	Description/Function
B3,A4,B6, B8,B9-B11, C12,C4,C5, A6,A8,A3,A5, B7,C8-C11, C13,B4,B5,A7, A9,A11,A13, B13,C14	DAL<31:00>	input/output	Data/address lines—Transfer data, status, and address information to and from the F chip.
J2	$\overline{\text{DAL BUSY}}$	input	DAL busy—Indicates that the DAL<31:0> lines are controlled by the port controller.
D1	$\overline{\text{DAL STALL}}$	input	DAL stall—Indicates that the data source cannot respond to the data request in the same cycle.
N5,P4,M5, N4,M4,N3, M3,N2,K2, J1,J3,H1,L2, L1,K3,K1,	MIB<38:15>	input	Microinstruction bus—Transfers microinstructions, synchronization control, and opcodes.
G3	MTB MISS	input	Mini-TB miss—Indicates that the mini-TB in the I/E chip has missed.
G2	IID	input	Input instruction—Indicates that the I/E chip is starting a new VAX instruction.
F1	$\overline{\text{ABORT}}$	input	Abort—When asserted, the last instruction is ignored.
E1	$\overline{\text{TRISTATE}}$	input	Tristate—Disables all outputs of the F chip.
G1	$\overline{\text{DCLO}}$	input	dc low—Asserted for 70 ms after the power supply voltage is within the specified value to initialize the F chip.
F2	F BOX N	output	F box N code—Indicates a negative value or an exception to the I/E chip at the completion of a floating-point calculation.
C3	F BOX Z	output	F box Z code—Indicates a zero value to the I/E chip at the completion of a floating-point calculation.

Pin	Signal	Input/output	Description/Function
M6	OPER VALID	input	Operand valid—Indicates the successful assembly of the operand.
P5	DATA OUT	output	Data out—Indicates that the F chip read result is pending.
E14	PH12	output	Phase 12—Internal phase 12 output.
H12,H13	CLK	input	Clock—The MOS clock signal from the I/E chip.
G12,G13	CLK BAR	input	Clock bar—A complement MOS CLK signal from the I/E chip.
J14	CLK SYNC	input	Clock synchronization—A synchronizing clock signal that indicates T_0 of the 200 ms microcycle.
K14,L13, L14,M13, L12,M12	MIR 9-4	outputs	Microcode test—Internal test bits MIR 9 through MIR 4.
C6,D12,D13, E2,E3,F12-F14,G14,M8, N8,P8	V_{DD}	input	Voltage—Power supply voltage.
C7,D2,D3, E12,E13,F3, H2,H3,J12, J13,K12,K13, M9,N9,P9	V_{SS}	input	Ground—Common ground reference.

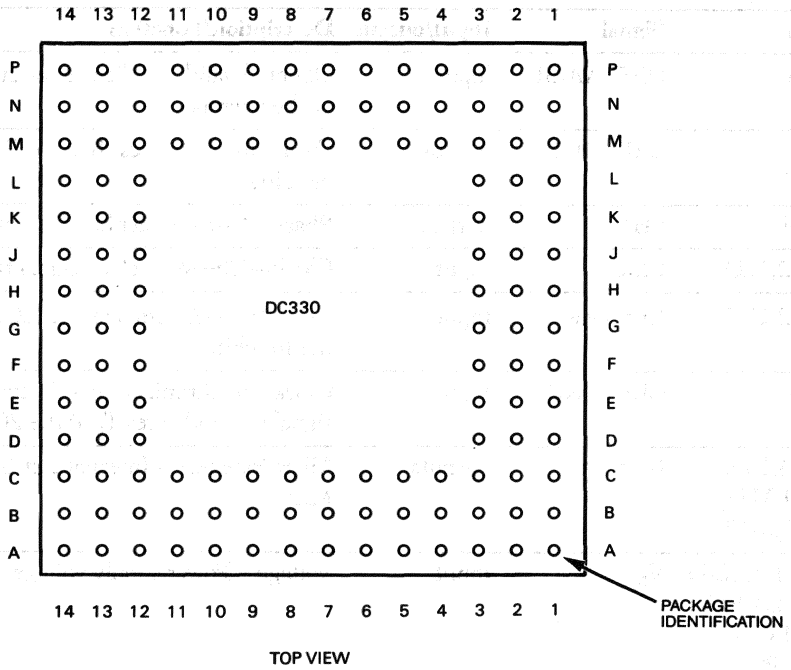


Figure 2 ▪ DC330 Pin Assignments

Data/address Lines (DAL<31:00>)—These are bidirectional, three-state lines that are used for receiving operands, transmitting results, and for reading and writing the control/status register of the F chip. Addresses are received from the DAL lines for detecting unaligned memory references and to properly assemble the data.

DAL Busy ($\overline{\text{DAL BUSY}}$)—This synchronization signal is driven by the port controller to indicate that it is driving the DAL<31:00> lines. Any microcoded data that was transferred during the cycle is ignored and must be sent again. The DAL<31:00> lines are a high impedance when this signal is asserted.

DAL Stall ($\overline{\text{DAL STALL}}$)—This synchronization signal is driven by the V-11 chips to stall the microinstructions so that operations that require more than one cycle can be completed. It is used when the I/E chip reads a result of a calculation that is not completed.

Abort ($\overline{\text{ABORT}}$)—When asserted, it indicates that the microword received during the last T150-T200 phase is not valid and should not be executed.

dc Low ($\overline{\text{DCLO}}$)—This signal is asserted for 70 ms after the 5-Vdc power is within the specified limits. It is used for initializing the control sequencers in the F chip.

F Box Negative (F BOX N)—This condition is transferred to the I/E chip to indicate a negative result upon completion of a floating-point calculation. It is also used to indicate exceptions.

F Box Zero (F BOX Z)—This condition is transferred to the I/E chip to indicate a zero result upon completion of a floating-point calculation.

Initial Instruction Decode (IID)—This input indicates that the I/E chip is starting a new VAX instruction and that a new opcode may be transferred to the F chip. If a calculation was in progress, it must be aborted and the new instruction started.

Microinstruction Bus (MIB < 38:15 >)—These lines contain a microinstruction from time T150 to T200 and provide synchronization control from time T50 to T100.

Mini Translation Buffer Miss (MTB MISS)—This input is used for DAL line synchronization. It indicates that the Mini-TB in the I/E chip has missed data, and that a page table entry is being transferred from the BTB on the DAL lines. The current microinstruction should be stalled.

Operation Valid (OPER VALID)—This is an input sequencer test signal that indicates the successful assembly of the operand. Control is granted to the main sequencer during the last OPER VALID signal that is a function of the data type.

Data Out (DATA OUT)—This is an output sequencer test signal indicating that the F chip result is ready and the read result is pending. An I/O sequence will take place as a result of this indication.

Microinstruction Test (MIR TEST < 5:0 >)—The microcode test outputs contain the values of the internal microinstruction register (MIR < 9:4 >). They are buffered and driven by open-drain devices and external pullup resistors of 1 k are required. The internal microcode changes every 100 ns at PHI1 and PHI5.

Clock (CLK)—This signal is the MOS clock input from the M chip and is used to generate the internal timing. The CLK input has a period of 50 ns and is high during odd phases.

Clock Bar (CLK BAR)—This signal is the complement of the MOS CLK signal from the M Chip and is used to generate the internal timing. The CLK BAR input is the nonoverlapping inverse of the CLK signal.

Clock Synchronize (CLK SYNC)—This signal indicates time T_0 of the 200 ns microcycle. It rises at time T175 and falls at T25.

Phase 12 (PHI2)—This is the internal PHI2 signal that is inverted and buffered by an open source. An external 1-k pulldown resistor is required.

Tristate ($\overline{\text{TRISTATE}}$)—Causes the DAL < 31:00 >, F BOX N, F BOX Z, and $\overline{\text{DAL STALL}}$ outputs to become a high impedance.

Voltage (V_{DD})—The power supply voltage to the 11 power input pins.

Ground (V_{SS})—The power supply return and signal ground to the 13 ground input pins.

Floating-point Instruction Set

The F chip is optimized to accelerate the following subset of the VAX instruction set. Refer to the VAX System Reference Manual (SRM) for a complete description of the operations. It does not execute the PDP-11 compatibility mode instruction set.

ADDF2, ADDD2, ADDG2, ADDF3, ADDD3, ADDG3, SUBF2, SUBD2, SUBG2, SUBF3, SUBD3, SUBG3, MULF2, MULD2, MULG2, MULF3, MULD3, MULG3, DIVF2, DIVD2, DIVG2, DIVF3, DIVD3, DIVG3, POLYF, POLYD, POLYG, CMPF, CMPD, CMPG, CVTLF, CVTLD, CVTLG, MULL2, MULL3, DIVL2, DIVL3, EMUL, and EDIV.

It executes in a nonoptimized manner the following subset of the VAX instuction set.

ADDF, ADDD, ADDG, SUBF, SUBD, SUBG, MULF, MULD, MULG, DIVF, DIVD,DIVG POLYF, POLYD, POLYG, CMPF, CMPD, CMPG, CVTLF, CVTLD, CVTLG, EDIV, and EMUL.

• Functional Operation

The F chip operates in the 200-ns cycle of the synchronous V-11 processor system. It receives microinstructions from the V-11 control store that can initiate sequences in the F chip controlled by the internal control store. The internal operations of the F chip cycle are performed at 100 ns. Figure 1 shows the main logic elements described in the following paragraphs and includes the following:

- A DAL and MIB interface
- A control store memory
- A fraction processor
- An exponent and sign processor
- Clock logic

DAL Interface—The DAL interface connects the F chip to the DAL<31:00> lines. Together with the MIB interface, its control logic determines when data on the DAL is valid. It provides the interfaces between the external DAL lines and the internal data paths, and it formats the data internally in a standard floating-point form. It also detects unaligned data and latches and rotates the data to the proper format. It detects short literals and unpacks them into the standard floating-point formats. The DAL Interface also contains the F chip control and status register (CSR) that is used by the I/E chip microcode to determine the exceptional conditions that may occur.

MIB Interface—This interface connects the F chip control logic to the microinstruction bus (MIB) and contains the currently executing external microinstruction, the current opcode and some of the DAL line synchronization bits. It contains logic that decodes the current external microinstruction and opcode, sending control information to the DAL interface and control store.

Control Store—The control store is a 160-word by 36-bit ROM that provides the internal microinstruction for the F chip. It also contains the simple microsequencer.

Fraction Processor—The fraction processor contains the 67-bit data path and control logic. It accelerates the execution of the VAX floating-point ADD, SUB, MUL, DIV, and POLY instructions on the Float, Double, and grand data types.It also accelerates the MULL and DIVL instructions. The fraction data path contains a register file of two temporary and eight constant registers, a 67-bit arithmetic logic unit (ALU), a 67-bit shifter, registers for storing the ALU information, shifter data and Q data, and a shift register able to shift up to 3 bits at a time. The operands are assembled in the I/O register that forms the interface between the fraction data path and the DAL interface. The fraction control logic receives encoded control signals from the control store (internal microinstruction) and MIB Interface (external microinstruction). It decodes these signals and drives the data-path control lines.

Exponent and Sign Processor—The exponent processor is a 13-bit data path with control logic. It allows calculations to be performed on exponents and signs of operands in parallel with operations in the fraction processor. The data path consists of a four location dual-ported RAM with zero detection on each output, an ALU, and latches for holding ALU data. A PLA is included to detect certain cases of exponent differences and a path for normalization data to get from the fraction data path to the exponent ALU. Literals can be transferred to the exponent data path from a dedicated ROM.

Clock Logic—The clock logic receives the V-11 high-level, 50 ns cycle time clock signals and the TTL-compatible 200 ns cycle time phase signal. It generates clocks that allow the DAL and MIB interfaces to synchronize to the 200 ns data transfers, and the internal data paths and control machines to cycle at 100 ns.

Specifications

The mechanical, electrical, and environmental specifications for the DC330 are contained in the *DC330 F Chip Hardware Specification*. The test conditions for the parameters in these specifications, unless specified otherwise, are as follows:

- Ambient temperature (T_A): 0°C to 70°C
- Supply voltage (V_{DD}): 5.0 V ±5% (maximum ripple 200 mV peak-to-peak)
- Back-bias voltage (V_{BB}): –3.0 V ±15% (maximum ripple 200 mV peak-to-peak)
- Ground (V_{SS}): 0 V

Mechanical Configuration

The mechanical dimensions for mounting the DC330 132-pin PGA package are shown in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to a device. Exposure to the absolute maximum rating conditions for extended periods may adversely affect the reliability of the device.

- Power supply voltage (V_{DD}): –5.0 V to 7.0 V
- Power supply voltage (V_{SS}): –3.0 V to –6.0 V
- Pin voltages: –1.0 V to 10 V
- Power dissipation (T_A = 100°C) 3.5 W
- Power dissipation (T_A = 0°C) 5.25 W
- Storage temperature range: –55°C to 125°C

Recommended Operating Conditions

- Supply voltage (V_{CC}): 5.0 V ±5%
- Ambient temperature operating range (T_A): 0°C to 70°C

ac and dc Electrical Characteristics

Refer to the *DC330 F Chip Hardware Specification* for the dc input and output parameters and ac timing parameters.

• Features

- Wide-range functionality in one package
  - High-performance processor
  - Memory management and protection
  - 46 floating-point instructions
  - Microdiagnostics
  - Clock generation
  - Console octal debugging technique (ODT)
  - Direct memory access arbitration
- Resident memory management
  - 4-Mbyte addressing
  - Direct addressing of tasks up to 128 Kbytes
  - Three levels of memory protection in multiuser and multitask environments
- Comprehensive orthogonal instruction set
  - 140 instructions including floating-point instructions
  - Compatibility with PDP-11 system software
- High-performance system-oriented architecture
  - Pipeline architecture
  - 32-bit internal data path
  - Supports optional cache memory
  - Supports multiprocessor operation
  - Supports coprocessor interface
- Powerful vectored multilevel interrupt and trap structure with four external interrupts

• Description

The DCJ11 is an advanced 16-bit microprocessor using very large-scale integrated (VLSI) CMOS technology. The DCJ11 is a complete implementation of the PDP-11 processor architecture and is compatible with PDP-11 software and operating systems. The DCJ11 consists of a data chip and control chip that are contained in a 60-pin dual-inline package (DIP). A block diagram of the DCJ11 microprocessor is shown in Figure 1.

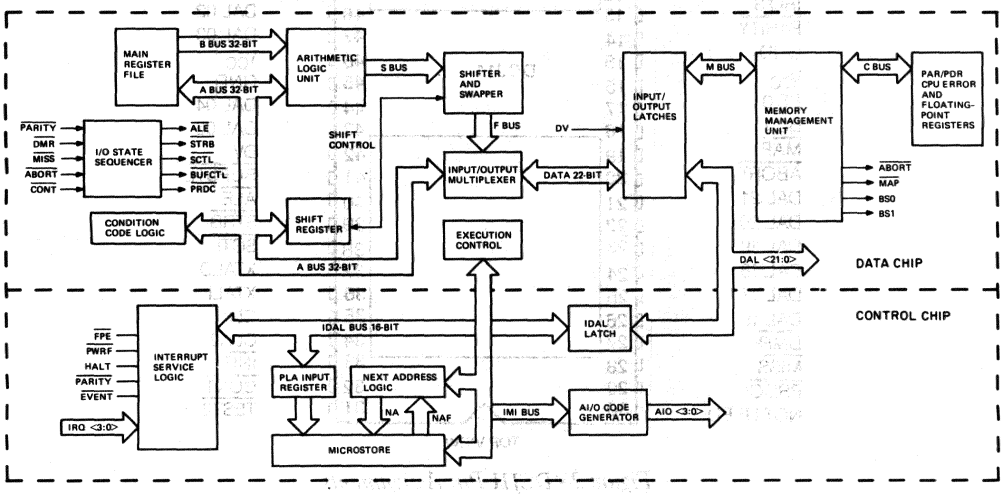


Figure 1 • DCJ11 Microprocessor Block Diagram

Many system design requirements and options are included as integral parts of the chip. A high-performance processor using a four-level prefetch pipeline, resident memory management, floating-point arithmetic, console octal debugging technique (ODT), microdiagnostics, and clock generation provide efficient system functionality in a single package. The orthogonal instruction set allows fast and efficient programming to minimize development time and cost. The DCJ11 combines leadership system functionality with complete system software, a highly integrated design, and low-power consumption to allow new classes of microprocessor applications. A block diagram of the DCJ11 is shown in Figure 1.

• Signal and Pin Descriptions

The input and output signals and the power and ground connections for the DCJ11 60-pin DIP are shown in Figure 2 and defined in Table 1. These signals are briefly described in the table and a more detailed description of the signal functions is contained in the following paragraphs. The system interface refers to the user's application of the DCJ11 and must be capable of providing or receiving these signals.

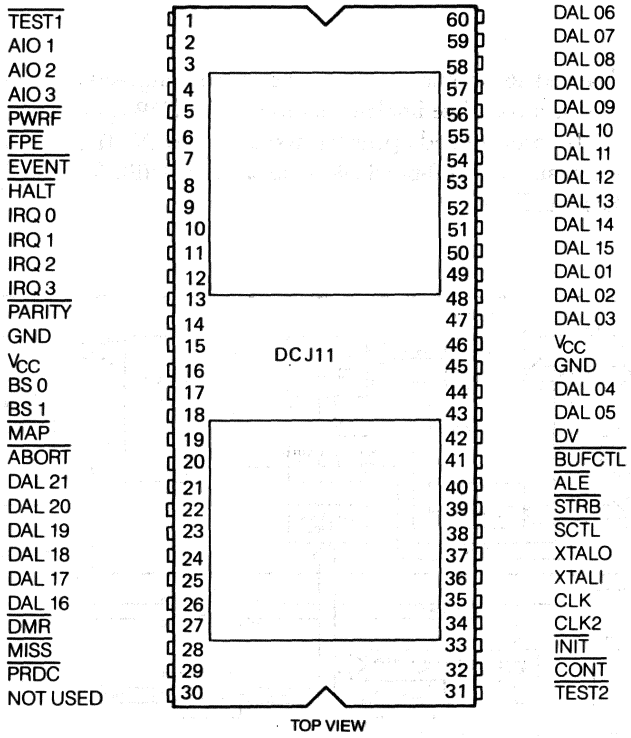


Figure 2 • DCJ11 Pin Assignments

Table 1 • DCJ11 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
21-26, 43-44, 47-60	DAL<21:00>	input/output[1]	Data/address lines—Time-multiplexed data and address bus.
17-18	BS<1:0>	output[1]	Bank select—These time-multiplexed signals define the type of physical address on the data/address bus, and indicate if either a cache memory bypass or a force miss occurs.
19	$\overline{\text{MAP}}$	output[1]	Map—This time-mulitiplexed signal indicates if the I/O map is enabled or if a DMA grant occurs.
2-5	AIO<3:0>	output[1]	Address input/output—These signals indicate the type of transaction currently being executed, i.e., read, write, or IACK.
40	$\overline{\text{ALE}}$	output[1]	Address latch enable—Latches addresses, AIO codes, map enable signals, and the BS control signals.
41	$\overline{\text{BUFCTL}}$	output[1]	Buffer control—Indicates the direction of data on the DAL bus. The line is active (low) when the DCJ11 is not driving to the DAL bus.
38	$\overline{\text{SCTL}}$	output[1]	Stretch control—Identifies the extended portion of stretched cycles. The edges can be used to strobe data.
39	$\overline{\text{STRB}}$	output[1]	Strobe—General purpose strobe signal.
29	$\overline{\text{PRDC}}$	output[1]	Predecode strobe—Indicates when the prefetch buffer is being decoded as the next macroinstruction.
20	$\overline{\text{ABORT}}$	input/output[1]	Abort—Indicates that an abort condition exists, i.e., a memory management or address error, bus timeout, nonexistent memory, or parity error.
28	$\overline{\text{MISS}}$	input[1]	Miss—Reports the hit or miss status of the current cache memory entry lookup.
42	DV	input[1]	Data valid—Set to latch data into the DCJ11.
32	$\overline{\text{CONT}}$	input[1]	Continue—Used to terminate all extended cycles.
27	$\overline{\text{DMR}}$	input[1]	Direct memory access request—Used to force a current cycle to be extended.
10-13	IRQ<3:0>	input[1]	Interrupt Request <3:0>—Four maskable interrupt request lines.
9	HALT	input[1]	Halt—A low-priority nonmaskable interrupt that forces the system into console mode.
8	$\overline{\text{EVENT}}$	input[1]	Event—A maskable interrupt that forces a trap through vector location 100.

Pin	Signal	Input/Output	Definition/Function
6	$\overline{PWRF}$	input[1]	Powerfail—A high-priority nonmaskable interrupt that forces a trap through vector location 24.
7	$\overline{FPE}$	input[1]	Floating-point enable—Reserved for a future FPA coprocessor implementation. A high-priority non-maskable interupt that forces a trap through vector location 244.
14	$\overline{PARITY}$	input[1]	Used to report parity errors.
35	CLK	output[2]	Clock—An output for intra-hybrid and diagnostic use only.
34	CLK2	output[2]	Clock 2—An output with identical frequency as CLK. Can be used as system clock.
33	$\overline{INIT}$	input[1]	Initialize—Initializes or resets the system by forcing it through the powerup procedure.
36	XTALI	input	Crystal input—External crystal input
37	XTALO	output	Crystal output—External crystal output
16,46	V_{cc}	input	Voltage—Power supply voltage
15,45	GND	input	Ground—Ground reference.
1	$\overline{TEST1}$[2]	input	Disables all outputs.
31	$\overline{TEST2}$[2]	input	Disables the clock outputs. Permits external logic to drive the DCJ11 clock circuitry through CLK output.

[1]TTL levels
[2]MOS levels

Figure 3 shows the input and output signals grouped according to signal function.

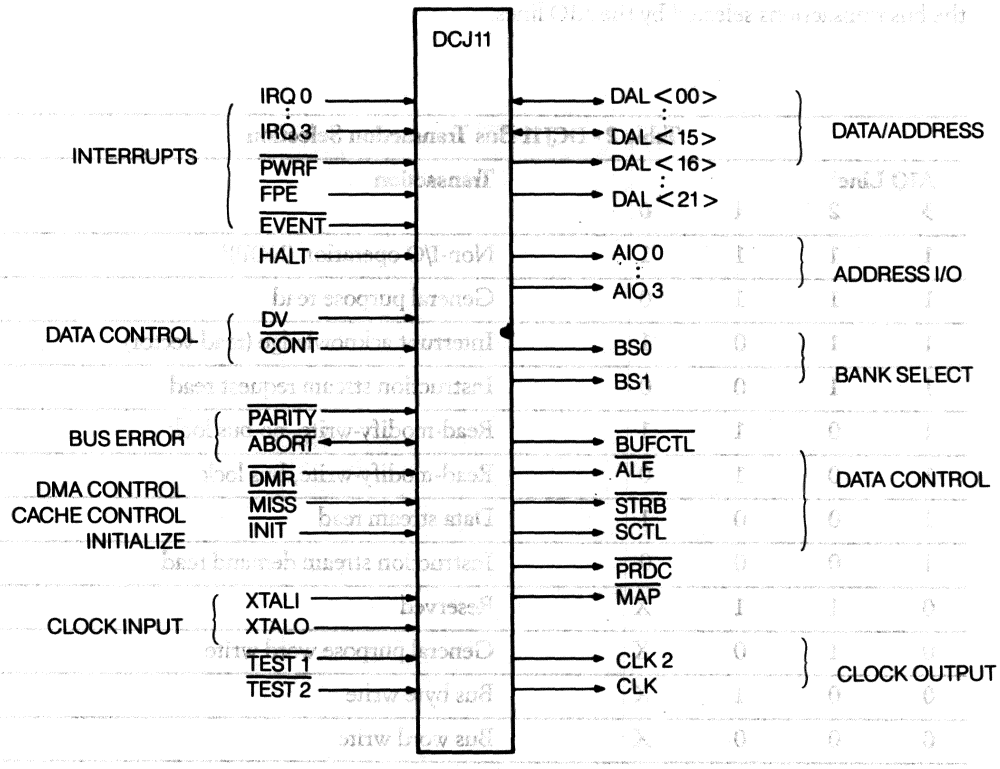


Figure 3 • DCJ11 Signal Functions

Data and Address Bus

Data and address bus (DAL<21:00>—The data and address bus consists of 22 time-multiplexed data and address lines. The basic bus consists of DAL<15:00> and is bidirectional. The extended bus consists of DAL<21:16> and is used for outputs only. During the first half of each transaction, the DCJ11 provides either a physical address, the acknowledged interrupt level or a general purpose code. The physical address can use all 22 bits of the bus. The acknowledged interrupt level uses DAL<03:00> and the general purpose code uses DAL<07:00>. During the second half of the transaction, the DCJ11 transmits or receives data on the basic bus (DAL<15:00>). The extended bus lines (DAL<21:16>) are driven with test information when the $\overline{\text{BUFCTL}}$ signal is asserted. The data being transmitted or received depends on the type of bus transaction being performed and is described under bus operations.

Address Input/Output Lines

Address input/output (AIO<3:0>)—The address input/output signals are latched at the start of any bus transaction and are coded to indicate the type of transaction being performed. Table 2 lists the bus transactions selected by the AIO lines.

Table 2 • DCJ11 Bus Transaction Selection

AIO Line[1] 3	2	1	0	Transaction
1	1	1	1	Non-I/O operation (NOP)[2]
1	1	1	0	General purpose read
1	1	0	1	Interrupt acknowledge (read vector)
1	1	0	0	Instruction stream request read
1	0	1	1	Read-modify-write, no bus lock
1	0	1	0	Read-modify-write, bus lock
1	0	0	1	Data stream read
1	0	0	0	Instruction stream demand read
0	1	1	X	Reserved
0	1	0	X	General purpose word write
0	0	1	X	Bus byte write
0	0	0	X	Bus word write

[1]X = logic 1 or 0

[2]A NOP transaction is an internal operation that does not require a bus transfer.

A bus transaction uses the DAL bus to access memory, I/O devices, or addressable registers. A general purpose transaction is used to access interface devices that are not directly addressable by the DAL bus. Interrupt acknowledge (IACK) transactions are in response to the DCJ11 granting an interrupt request.

Bank Select Lines

Bank select (BS<1:0>)—The bank select signals (BS1 and BS0) are time-multiplexed. During the first half of a bus transaction, they are used to define the type of address that is present on the DAL<21:00> bus. During the second half of the read and write transactions, these lines define the cache memory status. A cache memory bypass condition exists if the BS1 signal is asserted high, and a force cache miss exists if the BS0 signal is asserted high. Table 3 lists the address space selected by the bank select signals.

Table 3 • DCJ11 Bank Select Line Assignments

BS Line 1	BS Line 0	Address
0	0	Memory
0	1	System register
1	0	External I/O
1	1	Internal register

Table 4 lists the addresses assigned to the system registers and internal registers of the DCJ11 microprocessor. Physical addresses less than 17760000 are memory addresses. Addresses in the I/O page (17760000-17777777) that do not access a DCJ11 register are external I/O addresses. Addresses in the I/O page that access internal registers, except for CCR, are internal register addresses. Addresses in the range of 17777740 to 17777750 are classified as system registers.

Table 4 • DCJ11 Register Address Space Assignments

Address	Register	Register Classification
17 777 776	Processor status word (PSW)	Internal
17 777 772	Program interrupt request (PIRQ)	Internal
17 777 766	CPU error	Internal
17 777 752	Hit/Miss register	Internal
17 777 740—17 777 750	System register space	System
17 777 746	Cache control	System
17 777 707—17 777 700	CPU general registers	Internal
17 777 676—17 777 660	User data PAR, Reg. 0-7	Internal
17 777 656—17 777 640	User Instruction PAR, Reg. 0-7	Internal
17 777 636—17 777 620	User data PDR, Reg. 0-7	Internal
17 777 616—17 777 600	User instruction PDR, Reg. 0-7	Internal
17 777 576	MM Status Register 2 (MMR2)	Internal
17 777 574	MM Status Register 1 (MMR1)	Internal
17 777 572	MM Status Register 0 (MMR0)	Internal
17 777 516	MM Status Register 3 (MMR3)	Internal
17 772 376—17 772 360	Kernel data PAR, Reg. 0-7	Internal
17 772 356—17 772 340	Kernel instruction PAR, Reg. 0-7	Internal
17 772 336—17 772 320	Kernel data PDR, Reg. 0-7	Internal
17 772 316—17 772 300	Kernel instruction PDR, Reg. 0-7	Internal
17 772 276—17 772 260	Supervisor data PAR, Reg. 0-7	Internal
17 772 256—17 772 240	Supervisor instruction PAR, Reg. 0-7	Internal
17 772 236—17 772 220	Supervisor instruction PDR, Reg. 0-7	Internal
All other addresses in I/O Page 17 760 000-17 777 777		External I/O
0—17 757 777 physical memory space memory		

Data Control Lines

Buffer control (BUFCTL)—The BUFCTL line is asserted when the DAL<15:00> bus is not being driven by the DCJ11 when receiving data during read transactions and during the stretched portion of any nonwrite transaction. The BUFCTL signal is negated when the DCJ11 is writing an address or data to the DAL<15:00> bus.

Address latch enable (ALE)—The ALE line is asserted at the start of a transaction and is used to latch the address, I/O bank, I/O map and AIO code information. During the second half of the transaction, the signal is negated and can be used to latch the cache memory data.

Strobe (STRB)—The STRB is negated at the end of every transaction and asserted by the end of the second clock period of the next transaction. The STRB signal identifies the end of a transaction and can be used for external bus control.

Stretch control (SCTL)—The SCTL line is asserted for the stretched portion of a bus transaction. During write transactions, the leading or trailing edges of the SCTL signal may be used to latch data. During read transactions, the trailing edge of SCTL may be used to latch data.

Precode strobe (PRDC)—The PRDC line is asserted for the first two clock periods of any transaction and indicates to the system that the DCJ11 is decoding the next macroinstruction.

I/O map enable (MAP)—The MAP output is time multiplexed. If it is asserted during the first half of a bus transaction, the I/O map is enabled and the MMR3 bit 5 is set to 1. During the second half of the bus transaction, the MAP signal acknowledges that the DMR signal was recognized and will never be asserted during write transactions.

Data valid (DV)—The DV input is received from the system interface when the DCJ11 can latch valid data from the DAL bus. This input is sampled when the BUFCTL signal is asserted during nonwrite stretched transactions.

Continue (CONT)—The CONT input is received from the system interface to inform the DCJ11 that it is finished using the DAL bus and ends a stretched transaction.

Bus Error Lines

Parity error (PARITY)—The PARITY input is used by the system interface to report parity errors to the DCJ11. A parity interrupt is generated when PARITY is asserted without ABORT being asserted. A parity abort is generated when PARITY is asserted with ABORT being asserted. Assertion of the ABORT signal is used to differentiate between a parity interrupt and a parity abort. An abort immediately traps without completing the current instruction.

Memory cycle abort (ABORT)—If ABORT input is asserted during the first clock period, it indicates that the memory cycle should not be initiated. During a bus transaction, if an abort condition (register, timeout, parity errors, etc.) is detected by the system interface, the system interface asserts the ABORT signal with an open-collector device during the stretched part of the transaction.

Control Lines

DMA request (DMR)—The DMR line is driven by the system interface to gain control over the DAL<15:00> bus. The DMR line status is sampled by the DCJ11 at the rising edge of T0 and the request is acknowledged by the DCJ11 by asserting the MAP lines. The DMR signal is not acknowledged during write transactions. Asserting DMR ensures that the next transaction will be stretched. All write transactions are stretched.

Cache miss ($\overline{MISS}$)—The $\overline{MISS}$ input is received from the system interface to indicate the status of the current cache memory lookup entry. The DCJ11 samples the status of the $\overline{MISS}$ line at the rising edge of T3 during a read transaction. If the $\overline{MISS}$ signal is asserted to indicate the entry was not located in cache memory, the current read transaction will be stretched.

Initialize ($\overline{INIT}$)—The $\overline{INIT}$ input is driven by the system interface to initialize the DCJ11 by forcing it to go through a powerup routine.

Interrupt Lines

Interrupt request (IRQ<3:0>)—IRQ<3:0> are four maskable interrupt request lines that allow the system interface to interrupt DCJ11 operations. The four inputs represent four interrupt levels and are synchronized and latched by the DCJ11.

The interrupt is acknowledged only if the current PSW bits 07:05 are set to a lower level than requested by the system interface. Table 5 lists the interrupt level assignments.

Table 5 • DCJ11 Interrupt Level Assignments

Input	Level	PSW bits 07	06	05
IRQ 3	7	1	1	1
IRQ 2	6	1	1	0
IRQ 1	5	1	0	1
IRQ 0	4	1	0	0

Powerfail ($\overline{PWRF}$)—The $\overline{PWRF}$ input is a nonmaskable interrupt from the system interface. The DCJ11 traps to vector address 24 for the powerfail routine.

Floating-point exception ($\overline{FPE}$)—The $\overline{FPE}$ input is a nonmaskable interrupt from the system interface that causes the DCJ11 to trap to vector address 244 for the service routine.

Bus event ($\overline{EVENT}$)—The $\overline{EVENT}$ input is a bus interrupt from the system interface and the DCJ11 traps to vector address 100 for the service routine.

Halt ($\overline{HALT}$)—The $\overline{HALT}$ input is the lowest nonmaskable interrupt from the system interface and it forces the DCJ11 into the console mode.

Clock Lines

Crystal input (XTAL1 and XTAL0)—The XTAL1 and XTAL0 inputs provide connections for an external crystal as shown in Figure 4.

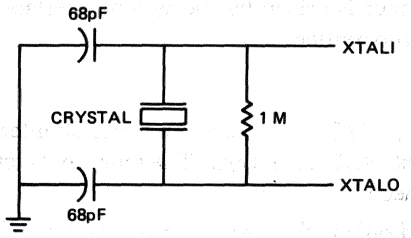


Figure 4 • DCJ11 Typical Crystal Connections

Clock (CLK)—The CLK signal is used for testing purposes and *should not* be used.

Clock 2 (CLK2)—The CLK2 signal is the same as the CLK and can be used by the system interface as the system clock. The frequency of this signal is the same as the crystal frequency.

Miscellaneous Lines

Test 1 ($\overline{\text{TEST1}}$)—The $\overline{\text{TEST1}}$ input is asserted by the system interface and disables all the DCJ11 output signals. The input is pulled up internally.

Test 2 ($\overline{\text{TEST2}}$)—The $\overline{\text{TEST2}}$ input is asserted by the system interface to disable the CLK and CLK2 outputs and to allow an external clock to drive the DCJ11. The input is pulled up internally.

Power Supply (V_{cc})—V_{cc} input provides the DCJ11 with 5-Vdc operating power.

Ground (GND)—The GND input connects the DCJ11 to the system interface ground reference.

• Architecture Summary

The DCJ11 implements the PDP-11 processor architecture to achieve processing power equivalent to high-performance minicomputers. The primary internal data path is 32 bits to optimize performance of floating-point arithmetic. The combination of a four-level instruction prefetch pipeline and an efficient internal architecture enables the DCJ11 to execute many instructions in four clock periods.

THe DCJ11 incorporates many system features not available with alternative 16-bit microprocessors. It includes a powerful and flexible set of control signals between the processor and the system interface allowing high-performance and high-reliability systems to be built. The processor provides support for cache memory, parity memory, multiprocessing, and coprocessing implementations. Reliability has been built into the processor. The CMOS technology provides reliability and includes error reporting registers and a firmware resident console monitor handler.

The key features of the DCJ11's register set memory management and interrupt structure are described in the following paragraphs. More comprehensive descriptions are provided by the PDP-11 System Handbooks for the PDP-11/44 or /70. Figure 5 is a programming model of the DCJ11 and shows the registers and accumulators available to the user for programming and maintenance purposes.

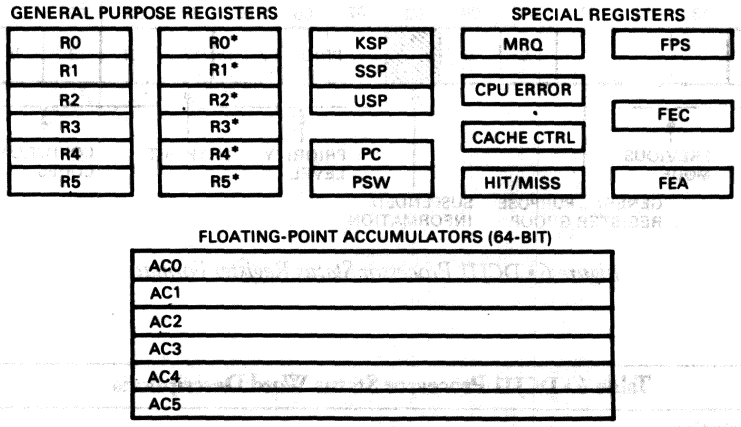


Figure 5 • DCJ11 Programming Model

General Purpose Registers

Two groups of 16-bit general purpose registers are used as accumulators, index reference, autoincrement, autodecrement, and stack pointers for temporary storage of data. Registers R0 through R5 are selected when bit 11 of the processor status longword (PSW) is set to zero. Registers R0 through R5 group is selected when bit 11 is set to one. These independent register banks can be used by software to minimize context switch delay.

Stack Pointers

The DCJ11 operates in three processor modes—kernel, supervisor, and user. Each processor mode has a stack pointer that is designated R6. The stack is used to store the processor status and current program counter values when interrupts, traps, or subroutine calls occur. The current mode is determined by bits 15:14 of the PSW. The current mode selects the stack pointer to be used for all instructions except for MOVPM instructions that use the previous mode, bits 13:12 of the PSW, to select the stack pointer. The DCJ11 can access only the selected stack pointer.

Program Counter

The program counter (PC) contains the 16-bit address of the next instruction to be executed. It is designated R7 and controls the sequencing of instructions. The PC is directly addressable by single- and double-operand instructions and is a general purpose register that is normally not used as an accumulator.

Processor Status Register

The processor status register (PSR) contains the processor status word (PSW) that contains information related to the current and previous processor mode. It includes the priority levels, condition codes, register sets, and trap information. The format of the PSR is shown in Figure 6 and the functions of the register bits are described in Table 6.

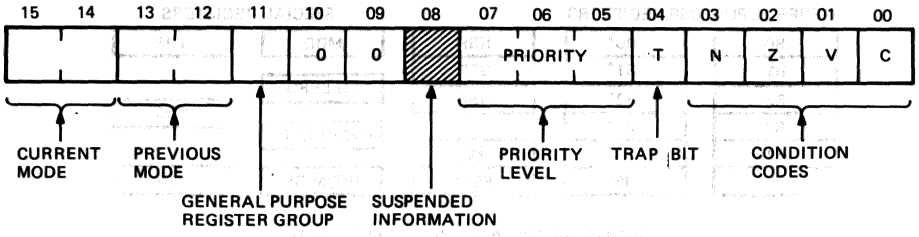


Figure 6 • DCJ11 Processor Status Register Format

Table 6 • DCJ11 Processor Status Word Descriptions

Bits*	Description
15,14	Current mode—Indicates the current operating mode as follows.

Bit 15	Bit 14	Mode
0	0	Kernel
0	1	Supervisor
1	0	Illegal
1	1	User

Bits*	Description
13:12	Previous mode—Indicates the previous operating mode and is coded the same as bits 15:14.
11	GP Register—Selects the group of general purpose registers being used. When the bit is set, the R0'-R5' group is selected and when cleared, the R0-R5 group is selected.
10:08	Not used.
07:05	Priority level—Indicates the current priority level of the microprocessor as follows:

Bit 07	Bit 06	Bit 05	Priority level
1	1	1	7
1	0	0	6
1	0	1	5
1	0	0	4
0	1	1	3
0	1	0	2
0	0	1	1
0	0	0	0

Bits*	Description
04	Trap—The trap bit is inactive when cleared. When set, the processor traps to location 14 at the end of the current instruction. It is used for debugging programs and setting breakpoints.

Bits*	Description
03:00	Condition codes—Theses bits contain information related to the result of the last CPU arithmetical or logical operation as follows: Bit 03 negative (N)—Set when the previous operation result was negative. Bit 02 Zero (Z)—Set when the previous operation resulted as zero. Bit 01 Overflow (V)—Set when the previous operation resulted in an arithmetic overflow. Bit 00 Carry (C)—Set when the previous operation caused a carry out of its most significant bit.

*All bits can be read or written except where indicated.

CPU Error Register

The CPU error register is a read-only register used by the DCJ11 to report CPU errors detected by the system software. Six separate error conditions cause the microprocessor to trap through location 4. The format of CPU error register is shown in Figure 7 and the functions of the bits are described in Table 7.

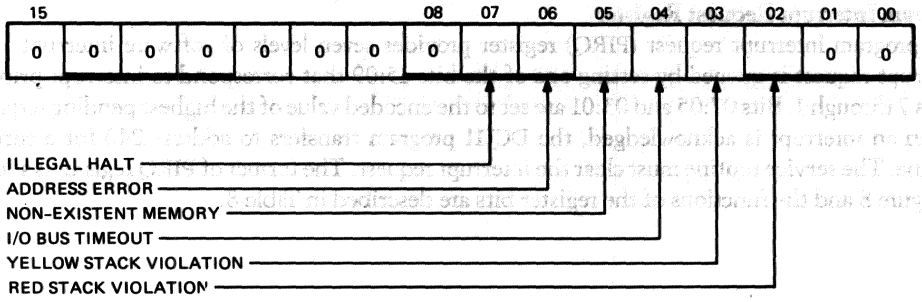


Figure 7 • DCJ11 CPU Error Register Format

Table 7 • DCJ11 CPU Error Register Descriptions

Bit*	Description
15:08	Not used
07	Illegal Halt—Set when the execution of a HALT instructin is attempted in user or supervisor mode.
06	Address Error—Set when a word access to an odd byte address or an instruction fetch from an internal register is attempted.
05	Nonexistent Memory—Set when a reference to main memory times out.
04	I/O Bus Timeout—Set when a reference to the I/O page times out.
03	Yellow Stack Violation—Set on a yellow zone stack overflow trap. (Kernel mode stack reference less than 400 octal.)
02	Red Stack Violation— Set on a red stack trap—a kernel stack push abort violation during an interrupt, abort, or trap sequence.
01:00	Not used

*All bits are read-only.

Program Interrupt Request Register

The program interrupt request (PIRQ) register provides seven levels of software interrupts. An interrupt request is queued by setting one of the bits 15:09 that correspond to interrupt priority levels 7 through 1. Bits 07:05 and 03:01 are set to the encoded value of the highest pending request. When an interrupt is acknowledged, the DCJ11 program transfers to address 240 for a service routine. The service routine must clear the interrupt request. The format of PIRQ register is shown in Figure 8 and the functions of the register bits are described in Table 8.

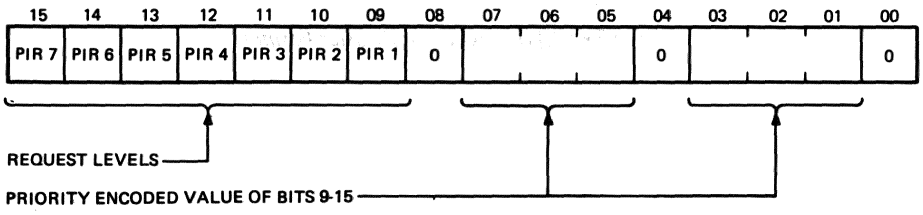


Figure 8 • DCJ11 Program Interrupt Request Register Format

Table 8 • DCJ11 Program Interrupt Request Register Descriptions

Bits*	Function
15:09	Request levels—Indicates the interrupt level requested as follows: bit 15 (Level 7)—Request an interrupt priority of level 7. bit 14 (Level 6)—Request an interrupt priority of level 6. bit 13 (Level 5)—Request an interrupt priority of level 5. bit 12 (Level 4)—Request an interrupt priority of level 4. bit 11 (Level 3)—Request an interrupt priority of level 3. bit 10 (Level 2)—Request an interrupt priority of level 2. bit 09 (Level 1)—Request an interrupt priority of level 1.
07:05	Encoded Value— Read-only bits 07:05 represent encoded value of highest priority level set in bits 15:09.
03:01	Encoded Value Read-only bits 03:01 represent the encoded value of the highest priority level set in bits 15:09. Same as bits 07:05.

*All bits can be read or written except where indicated.

Cache Control Register

The cache control register (CCR) controls the operation of the cache subsystem. Through the CCR, cache bypass and force miss signals can be controlled by software. The powerup microcode also sets the flush cache bit 8 to enable the orderly start of a cached machine. The format of the CCR is shown in Figure 9 and the functions of the bits are described in Table 9.

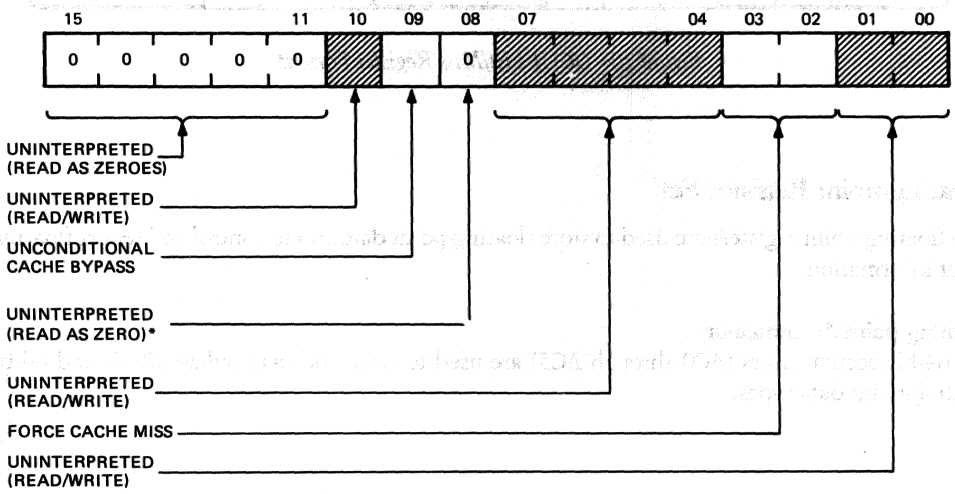


Figure 9 • DCJ11 Cache Control Register Format

Table 9 • DCJ11 Cache Control Register Descriptions

Bits*	Function
15:11	RAZ (read as zero and uninterpreted)
09	Bypass Cache—Asserts BS1 during the second half of read and write transactions.
08	Flush Cache—Causes the cache subsystem to flush cache. Always read as a zero.
07:04	Uninterpreted.
03:02	Force miss—When either of these bits are set, BS0 is asserted during the second half of read and write transactions.
01:00	Uninterpreted.

*All bits can be read or written except where indicated.

Hit/Miss Register

The hit/miss register (HMR) records the status of the miss input from the system interface. The HMR is a shift register that records a hit as a one and a miss as a zero for the most recent memory reads. A hit indicates that the data is located in the cache memory and a miss indicates that the data is located in the main memory. Bit 0 represents the most recent memory read and is shifted to the left on successive memory reads. The format of the HMR is shown in Figure 10.

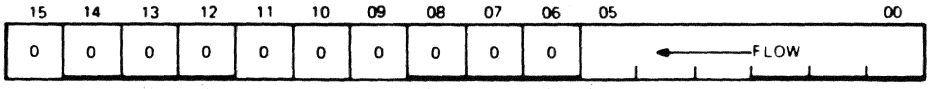


Figure 10 • DCJ11 Hit/Miss Register Format

• Floating-point Register Set

The floating-point registers are used to store floating-point data and to control and report floating-point information.

Floating-point Accumulator

Six 64-bit accumulators (AC0 through AC5) are used to store and manipulate 32-bit and 64-bit floating-point data types.

Floating-point Status Register

The floating-point status (FPS) register provides mode and interrupt control for the floating-point unit and conditions resulting from the execution of the previous instruction. The FPS register contains an error flag and four condition codes—carry, overflow, zero, and negative—that are equivalent to the CPU condition codes. The floating-point processor (FPP) recognizes the following floating-point exceptions.

- Detection of the presence of the undefined variable in memory
- Floating overflow
- Floating underflow
- Failure of floating to integer conversion
- Attempt to divide by zero
- Illegal floating opcode

For the first four exceptions, the bits in the FPS register are available to enable or disable interrupts. An interrupt caused by the last two exceptions can be disabled only by setting a bit that disables the interrupts of all seven of the exceptions. The error flag and condition codes are set by the FPP as part of the output of a floating-point instruction. The mode and interrupt control bits may be set by using the LDFS instruction. Figure 11 shows the format of the FPS register, and the functions of the register bits are described in Table 10.

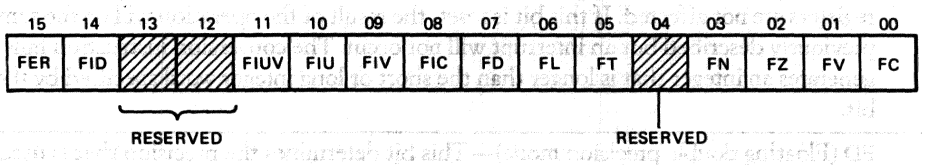


Figure 11 • DCJ11 Floating-point Status Register Format

Table 10 • DCJ11 Floating-point Status Register Description

Bit	Description
15	FER (Floating error)—This bit is set during a floating-point instruction when a division by zero occurs, an illegal opcode is specified, or any of the remaining errors are detected when the corresponding error interrupt is enabled. This bit is set by the Floating-point Processor (FPP) and cleared only an LDFPS instruction. The bit status is valid only if the most recent floating-point instruction produced a floating-point exception. This action is independent of the FID bit status.
14	FID (Floating interrupt disable)—When set, all floating-point interrupts are disabled. The FID bit is primarily a maintenance feature and should normally be clear to assure that the storage of –0 by an FPP is always accompanied by an interrupt. This bit is assumed to clear for all descriptions that follow that involving overflow, underflow, and occurrence of –0, and integer conversion errors.

Bit	Description
13:12	Not used—Reserved.
11	FIUV (Floating interrupt on undefined variable)—An interrupt occurs when this bit is set and a –0 is obtained from memory as an operand of an ADD, SUB, MUL, DIV, CMP, MOD, NEG, ABS, TST, or any LOAD instruction. The interrupt occurs before execution except for a NEG, ABS, and TST instructions when it occurs after execution. When FIUV is reset, –0 can be loaded and used in any FPP operation. The interrupt is not activated by the presence of –0 in any ac operand of an arithmetic instruction and trap on –0 does not occur in mode 0. The FPP will not store a result of –0 without a simultaneous interrupt.
10	FIU (Floating interrupt on underflow)—When set, a floating underflow will cause an interrupt. The fractional part of the result of the operation causing the interrupt will be correct. The biased exponent will be too large by 400 (octal) except for the special case of 0, which is correct. If the FIU bit is reset and if underflow occurs, no interrupt occurs and the result is set to exact 0.
09	FIV (Floating interrupt on overflow)—When set, a floating overflow will cause an interrupt. The fractional part of the result of the operation causing the overflow will be correct. The biased exponent will be smaller by a value of 400 (octal). No interrupt will occur if the FIV is reset and overflow occurs. The FPP returns to exact 0. Special cases of an overflow condition are defined in the detailed descriptions of the MOD and LDEXP instructions.
08	FIC (Floating interrupt on integer conversion)—When set and the conversion to an integer instruction fails, an interrupt will occur. The destination is set to 0, and all other registers are not affected. If this bit is reset, the result of the operation will be the same as previously described but an interrupt will not occur. The conversion instruction fails if it generates an integer that is longer than the short or long integer word specified by the FL bit.
07	FD (Floating double precision mode)—This bit determines the precision that is used for floating-point calculations. When set, double-precision is assumed; when reset, single-precision is used.
06	FL (Floating long integer mode)—This bit is used in the conversion between integer and floating-point format. When set, the integer format assumed is double-precision two's complement (i.e., 32 bits). When reset, the integer format is assumed to be single-precision two's complement (i.e., 16 bits).
05	FT (Floating chop mode)—When set, the result of any arithmetic operation is chopped (or truncated). When cleared, the result is rounded.
04	Not used—reserved.
03	FN (Floating negative)—This bit set when the result of the last floating-point operation was 0.
02	FZ (Floating zero)—This bit is set if the result of the last floating-point operation was 0.
01	FZ (Floating overflow)—This bit is set if the last floating-pointing operation resulted in an exponent.
00	FC (Floating carry)—This bit is set if the last operation resulted in a carry of the most significant bit. This can occur only in a floating or double-to-integer conversion.

Floating-point Exception Register

One interrupt vector is assigned to all floating-point exceptions. The six possible errors are coded in the 4-bit floating exception code (FEC) register as follows:

- 2 Floating-opcode error
- 4 Floating-divide by zero
- 6 Floating- or double-to-integer conversion error
- 8 Floating-overflow
- 10 Floating-underflow
- 12 Floating-undefined variable

The address of the instruction producing the exception is stored in the Floating Exception Address (FEA) register. The FEC and FEA registers are updated when one of the following occurs:

- Divide by zero
- Illegal opcode
- Any of the other four exceptions with the corresponding interrupt enabled

If one of the five exceptions occurs with the corresponding interrupt disabled, the FEC and FEA are not updated. Inhibition of interrupts by the Floating Interrupt Disable (FID) bit does not inhibit updating of the FEC and FEA if an exception occurs. The FEC and FEA are not updated if no exception occurs. Therefore, the store status (STST) instruction will return current information only if the most recent floating-point instruction produced an exception. No instructions are provided for storage into the FEC and FEA registers.

Memory Management

The DCJ11 implements the complete PDP-11 memory management and protection architecture with its extensions for extended direct addressing. This architecture provides a fully supported protection model for the design of reliable multiuser or multitasking systems. Address relocation and protection logic is integrated into the pipeline design so that performance penalty is not incurred by using the memory management unit (MMU).

The MMU provides three separate address spaces—kernel, supervisor, and user modes—with different privileges and independent sets of 16-bit mapping and protection registers. This structure protects the operating system from less privileged programs and minimizes system overhead from switching. The execution of some of the instructions is different depending on the current program mode.

Table 11 • DCJ11 Memory Management Mode Operation

Instruction	Kernel Mode	Supervisor/User Mode
HALT	Executes as specified	Traps through 4
WAIT, RESET, SPL	Executes as specified	Execute as a NOP instruction
RTI, RTT, MPTS	Alter PSR priority level bits 07:05 freely	Cannot alter PSR priority level bits 07:05
Stack references	Checked for overflow	Not checked for overflow

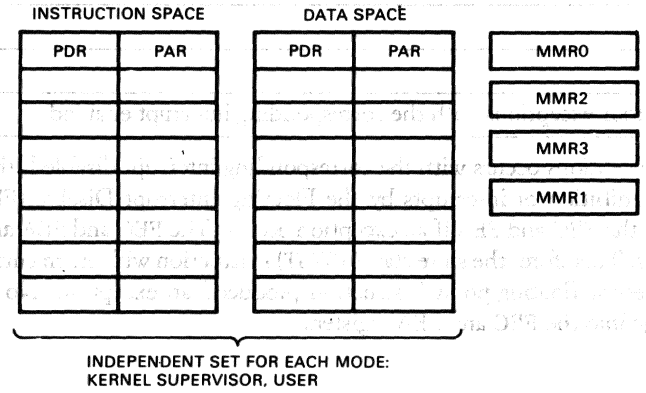


Figure 12 • DCJ11 Memory Management Registers

Table 12 • DCJ11 Page Descriptor Register Descriptions

Bits*	Description
15	Bypass cache—This bit implements a conditional cache bypass mechanism. If the PDR accessed during a relocation operation has this bit set, the time-multiplexed signal BS1 is asserted during the subsequent I/O cycle.
14:08	PLF (Page length field)—This field specifies the block number that defines the page boundary. The block number of the virtual address is compared against the page length field to detect length errors. An error occurs when expanding upward if the block number is greater than the page length field, and when expanding downward if the block number is less than the page length field.
07	Not used.
06	W (Page written)—This bit indicates whether the page has been written into since it was loaded in memory. When this bit is set, it indicates a modified page. This bit is automatically cleared when the PAR or PDR of that page is written into.

Bits*	Description
05:04	Not used.
03	ED (Expansion direction)—This bit specifies in which direction the page expands. If ED=0, the page expands upward from block number 0 to include blocks with higher addresses. If ED=1, the page expands downward from block number 127 to include blocks with lower addresses.
02:01	ACF (Access control field)—This field contains the access code for this particular page. The access code specifies the manner in which a page may be accessed and whether a given access should result in an abort of the current operation. The codes are

Access Code Bit 02	Bit 01	Page Access
0	0	Nonresident—abort all accesses
0	1	REAd only—abort on writes
1	0	Not used—abort all accesses
1	1	Read/write access.

00	Not used.

*All bits can be read or written except as indicated.

The MMU contains four memory management registers that are used to control and record the status of the memory management functions. The registers and address assignments are as follows:

- Memory management register 0 (address 17 777 572)
- Memory management register 1 (address 17 777 574)
- Memory management register 2 (address 17 777 576)
- Memory management register 3 (address 17 777 516)

Page Address and Page Descriptor Registers

Each operating mode is assigned 16 page-address registers (PAR) and 16 page-descriptor registers (PDR) to control the instruction and data space. These 96 registers are addressable to the external DAL bus. A PAR contains a 16-bit displacement that is added to bits 12:06 of the virtual PC or to the address received from the execution section to create part of the relocated physical address. A PDR contains information relative to page data such as expansion, length, and access control. The format of the information in the PDR is shown in Figure 13 and the function of the register bits is described in Table 12.

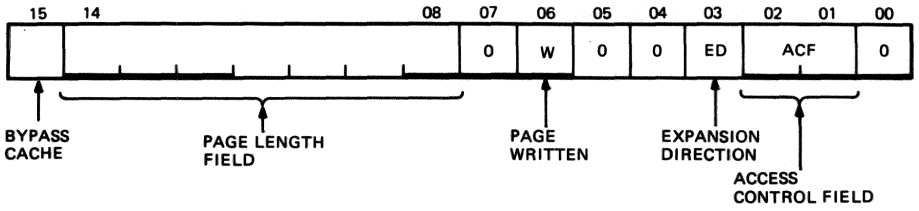


Figure 13 • DCJ11 Page Descriptor Register Format

The DCJ11 can optionally implement instruction and data space (I/D Space) relocation to expand the direct addressing range of a DCJ11 program or to facilitate efficient code sharing in a multiuser environment. I/D space relocation can be separately enabled for each of the processor modes. When I/D space relocation is enabled, the DCJ11 classifies memory references as instruction stream or data stream references and independently relocates them through the corresponding PAR and PDR. The memory references are classified by the DCJ11 as I space references. All other references are classified as D space references. If the I/D space relocation is disabled, all memory references are relocated via the instruction space for that mode that includes the following information:

- Instruction fetches
- Immediate operands (mode 27)
- Absolute addresses (mode 37)
- Index words
- First references in modes 17, 47, and 57

The classifications of memory references by addressing modes for the first, second, and third memory reference are listed in Table 13.

Table 13 • DCJ11 ID Space Relocation

Address Mode	Register	
0-6	7	
0	—	—
1	D	I
2	D	I
3	D/D	I/D
4	D	I
5	D/D	I/D
6	I/D	I/D
7	I/D/D	I/D/D

Memory Management Register 0

The memory management register 0 (MMR0) provides memory management register control and records status. The format of the information in the MMR0 is shown in Figure 14 and the function of the information is described in Table 14.

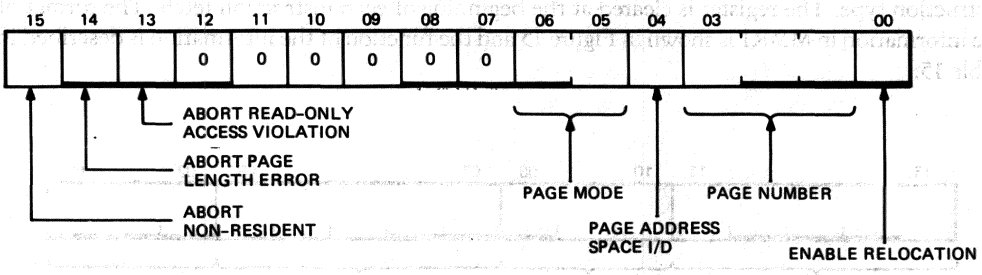


Figure 14 • DCJ11 Memory Management Register 0 Format

Table 14 • DCJ11 Memory Management Register 0 Description

Bits*	Description
15	Abort nonresident—Set by attempting to access a page with an access control field key equal to 0 or 2. It is also set by attempting to use memory relocation with an illegal processor mode (PSR 15:14=2).
14	Abort page length error—Set by attempting to access a location in a page with a block number (virtual address bits 12:06) that is outside the area authorized by the page length field of the page descriptor register for that page.
13	Abort read-only access violation—Set by attempting to write in a read-only page. Read-only pages have access keys of 1.
12:7	Not used.
06:05	Processor mode—A read-only bit that indicates the processor mode kernel/supervisor/user/illegal associated with the page causing the abort (kernel=00, supervisor=01, user=11, illegal=10). If the illegal mode is specified, an abort is generated and bit 15 is set.
04	Page space—A read-only bit that indicates the address space (I or D) associated with the page causing the abort (0=I space, 1=D space).
03:01	Page number—Read-only bits that contain the page number of the page causing the abort.
00	Enable relocation—When set, all addresses are relocated. When cleared, memory management is inoperative and addresses are not relocated.

*All bits can be read or written except as indicated.

Memory Management Register 1

Memory management register 1 (MMR1) records any autoincrement or autodecrement of a general purpose register, including explicit references through the PC. The amount that the register was incremented or decremented is stored in two's complement notation. This allows an effective recovery from an error resulting in an abort. The lower byte is used for all source operand instructions and the destination operand may be stored in either byte depending on the mode and instruction type. The register is cleared at the beginning of each instruction fetch. The format of the information in MMR1 is shown in Figure 15 and the function of the information is described in Table 15.

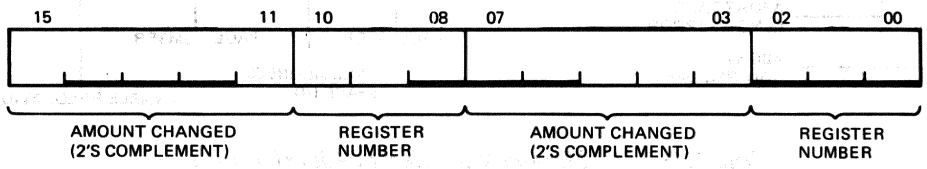


Figure 15 • DCJ11 Memory Management Register 1 Format

Table 15 • DCJ11 Memory Management Register 1 Description

Bits*	Description
15:11	Amount changed—Represents the amount of autoincrement or autodecrement in the two's complement notation for the register defined in bits 10:08.
10:08	Register number—Indentifies one of the eight general purpose registers.
07:03	Amount changed—Represents the amount of autoincrement or autodecrement in the two's complement notation for the register defined in bits 02:00.
02:00	Register number—Indentifies one of the eight general purpose registers.

*All bits are read-only.

Memory Management Register 2

Memory management register 2 (MMR2) is loaded with the program counter information of the current instruction and holds this information when an abort condition is recorded in MMR0.

Memory Management Register 3

Memory management register 3 (MMR3) enables the data space for the kernel, supervisor, and user operating modes. It also selects either 18-bit or 22-bit mapping and enables the I/O map and the request for the supervisor macroinstruction (CSM). The register format is shown in Figure 16 and the function of the information is described in Table 16.

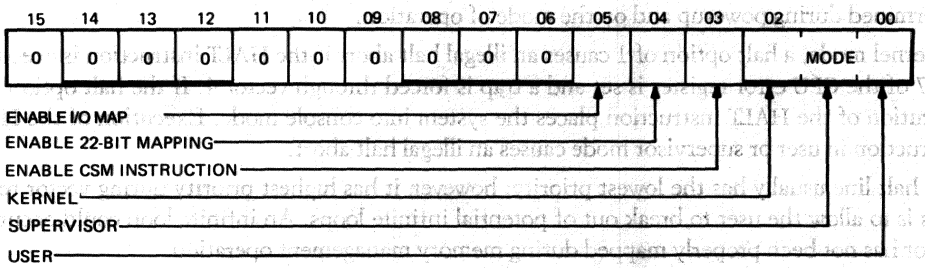


Figure 16 • DCJ11 Memory Management Register 3 Format

Table 16 • DCJ11 Memory Management Register 3 Description

Bits*	Description
15:06	Not used.
05	Enable I/O map—This bit is used to enable the external I/O mapping. The state of this bit is reflected in the $\overline{MAP}$ signal during the second half of the microcycle.
04	Enable 22-bit mapping—This bit enables the 22-bit memory addressing. The default is 18-bit addressing.
03	Enable CSM instruction—This bit enables the recognition of the call supervisor mode instruction.
02	Kernel—Enables the data space mapping for the supervisor operating mode.
01	Supervisor—Enables the data space mapping for the supervisor operating mode.
00	User—Enables the data space mapping for the user operating mode.

*All bits can be read or written except as indicated.

• Interrupts

The DCJ11 provides a set of trap, hardware, and software interrupt facilities. Four interrupt request lines allow the external hardware to interrupt the processor on four interrupt levels using an externally supplied vector. Eight levels of software interrupt requests are supported through use of the PIRQ register. Internally vectored traps are provided to flag error conditions. Table 17 identifies the DCJ11 asynchronous interrupts. The synchronous interrupts are listed in Table 18. The execution of a HALT instruction may cause different operations depending on the halt options determined during powerup and on the mode of operation.

In kernel mode, a halt option of 1 causes an illegal halt abort if the HALT instruction is executed. Bit 7 of the CPU error register is set and a trap is forced through vector 4. If the halt option is 0, execution of the HALT instruction places the system into console mode. Execution of the HALT instruction in user or supervisor mode causes an illegal halt abort.

The halt line usually has the lowest priority; however, it has highest priority during vector reads. This is to allow the user to break out of potential infinite loops. An infinite loop could occur if a vector has not been properly mapped during memory management operation.

Table 17 • DCJ11 Asynchronous Interrupts

Interrupt	Location	Vector Address	Priority Level*
Red stack trap (CPU error register bit 02)	Internal	4	NM
Address error (CPU error register bit 06)	Internal	4	NM
Memory management violation (MMR0 bits 13:15)	Internal	250	NM
Timeout/nonexistent memory (CPU error register bits 04,05)	Internal	4	NM
Parity error (PARITY, ABORT)	External	114	NM
Trace (T bit) Trap (PSW bit 04)	Internal	14	NM
Yellow stack trap (CPU error register bit 03)	Internal 4	NM	
Powerfail (PWRF)	External	24	NM
FP exception (FPE)	External	244	NM
PIR 7 (PIRQ bit 15)	Internal	240	7
IRQ 7	External	User-defined	7
PIR 6 (PIRQ bit 14)	Internal	240	7
EVENT	External	100	6
IRQ 6	External	User-defined	6
PIR 5 (PIRQ bit 13)	Internal	240	5

Interrupt	Location	Vector Address	Priority Level*
IRQ 5	External	User-defined	5
PIR 5 (PIRQ bit 12)	Internal	240	4
IRQ 4	External	User-defined	4
PIR 3 (PIRQ bit 11)	Internal	240	3
PIR 2 (PIRQ bit 10)	Internal	240	2
PIR 1 (PIRQ bit 09)	Internal	240	1
Halt line (HALT)	External	None—Places system in console mode.	

*NM = Nonmaskable

Table 18 • DCJ11 Synchronous Interrupts

Interrupt	Vector Address
Memory Management	250
FP instruction exception (FPS bits 11:08,15)	244
PIRQ	240
Memory Parity Error	114
TRAP (trap instruction)	34
EMT (emulator trap instruction)	30
IOT (I/O trap instruction)	20
BPT (breakpoint trap instruction)	14
Timeout and reserved instruction	4

• Bus Operation

The DCJ11 performs the bus transactions during the execution of the program instructions. Transaction requires a minimum of four clock periods and a maximum of eight clock periods. A transaction may be extended or stretched beyond its normal clock timing in increments of two clock periods. A read transaction can be extended by a minimum of four clock periods. This allows a transaction to be extended indefinitely. The four address input/output signals (AIO<3:0>) are latched at the DCJ11 output at the beginning of a transaction and indicate the type of transaction being performed. The AIO<3:0> codes for the type of transaction are identified in Table 2.

The bank select (BS1 and BS0) signals are used with bus and general purpose read or write transactions. These signals provide coded information to define the type of physical address that is on the DAL<21:00> lines. The bank select transactions are listed in Table 3.

Physical addresses that are less than 17760000 (octal) are memory references. Addresses in the I/O page (17760000-17777777) that do not access a DCJ11 register are external I/O references. Addresses in the I/O page access internal registers, except for CCR, are internal register references. System register references are addresses 17777740 through 17777750.

No Operation Transaction

During a no operation (NOP) transaction, the DCJ11 performs an internal operation and the bus is used for external data as shown in Figure 17. The assertion of $\overline{ALE}$ latches the AIO code that identifies the transaction as non-I/O. The transaction requires four periods to complete provided no DMA requests occur.

If $\overline{DMR}$ is asserted at the start of the cycle, the NOP transaction is stretched. A stretched transaction is shown in Figure 18. The DMA request stretches the transaction to a minimum of eight periods. The DMA request is received on $\overline{DMR}$ and is granted by $\overline{MAP}$ line. The $\overline{BUFCTL}$ and $\overline{SCTL}$ signals are asserted during the stretched portion. The transaction continues to stretch in two-period increments until the DCJ11 receives the $\overline{CONT}$ signal to end the transaction.

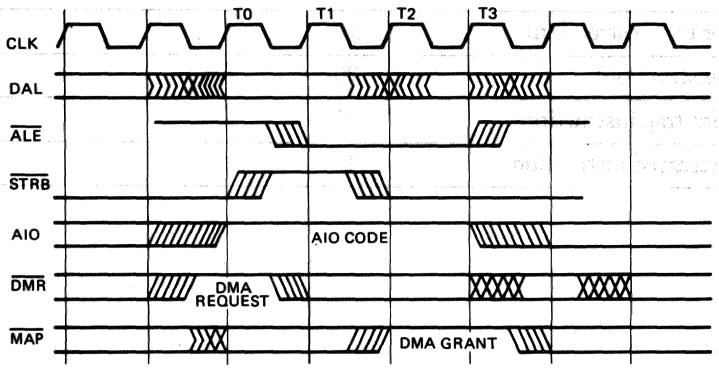


Figure 17 • DCJ11 Nonstretched Non-I/O Cycle Timing Sequence

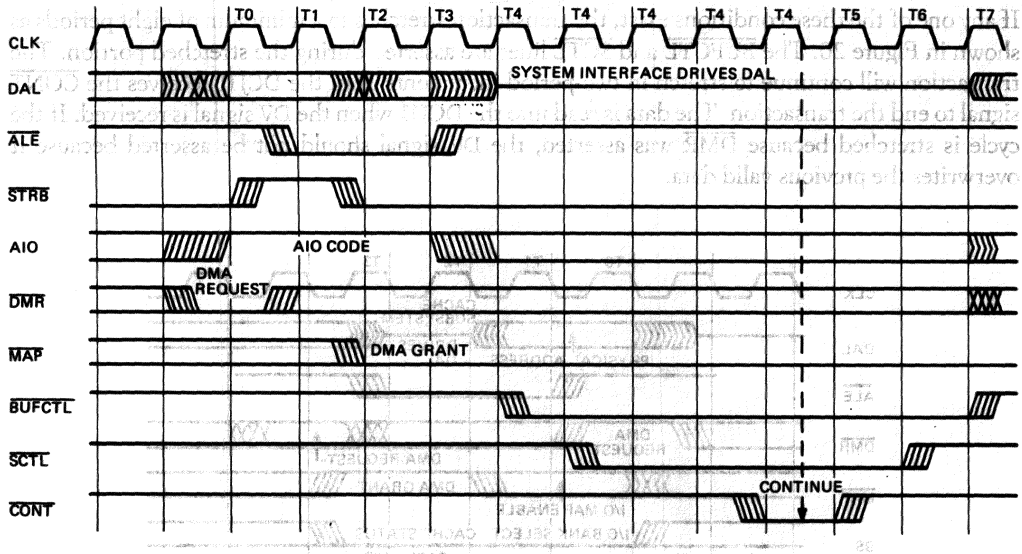


Figure 18 • DCJ11 Stretched Non-I/O Timing Sequence

Bus Read Transaction

A bus read transaction shown in Figure 19 uses the DAL bus to read information from memory, I/O, and other addressable registers. These transactions may be instruction stream read, data stream read, or the read portions of read-modify-write. The type of read transaction being performed is identified by the AIO code. The DCJ11 reads words and if a byte is required, the complete word is read and the excess byte is ignored.

The DCJ11 reports memory management or address errors on the $\overline{\text{ABORT}}$ output during the nonstretched portion of the transaction. If the $\overline{\text{ABORT}}$ signal is asserted, the information on DAL, BS<1:0>, and $\overline{\text{MAP}}$ lines should be ignored and the bus transaction should not be started.

The read transaction is initiated by the assertion of $\overline{\text{ALE}}$. This signal latches the AIO code, the physical address on DAL bus, the BS<1:0>, data, and the $\overline{\text{MAP}}$ (I/O map enable) signal. The DCJ11 latches the data on the rising edge of the T3 during a nonstretched transaction. A bus read is completed in four periods when all of the following conditions exist.

- BS<1:0> set to zeros (memory reference)
- No cache bypass
- No cache force miss
- No DMA grant
- No abort during a demand read
- No cache miss reported on MISS

If any one of the these conditions exist, the transaction stretches to a minimum of eight periods as shown in Figure 20. The $\overline{BUFCTL}$ and $\overline{SCTL}$ lines are asserted during the stretched portion. The transaction will continue to stretch in two-period increments until the DCJ11 receives the $\overline{CONT}$ signal to end the transaction. The data is read into the DCJ11 when the DV signal is received. If the cycle is stretched because $\overline{DMR}$ was asserted, the DV signal should not be asserted because it overwrites the previous valid data.

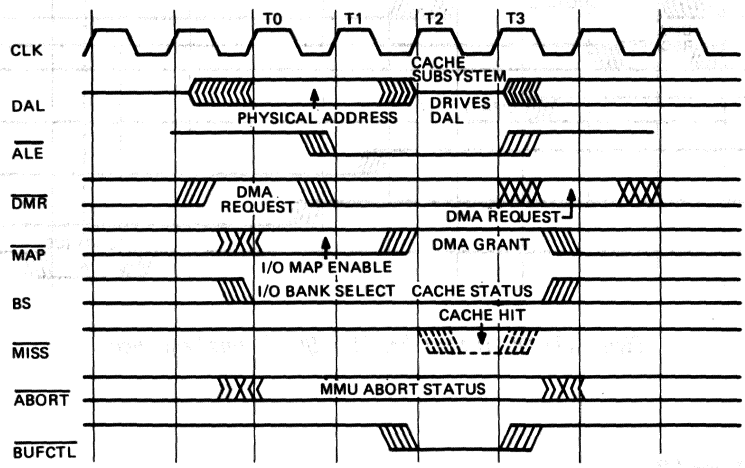


Figure 19 • DCJ11 Nonstretched Bus Read Cycle Timing Sequence

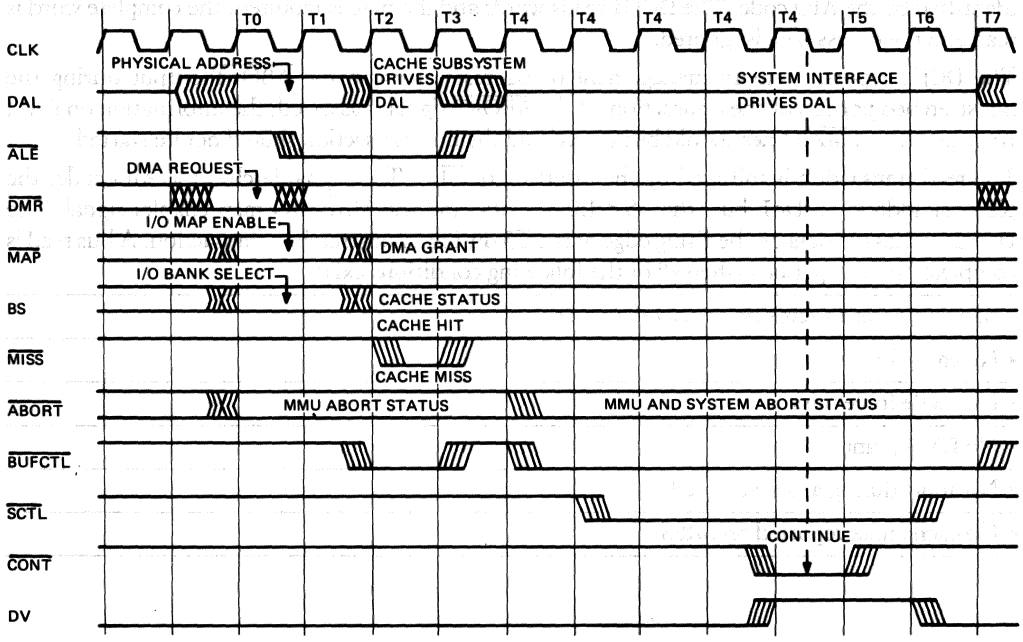


Figure 20 • DCJ11 Stretched Bus Read Cycle Timing

Bus Write Transaction

During a bus write transaction, shown in Figure 21, the DAL bus is used to write information into memory, I/O, and other addressable registers. The information can be a byte or a word as defined by the AIO code. The DCJ11 reports memory management or address errors on the $\overline{ABORT}$ during the first part of the cycle. If the $\overline{ABORT}$ signal is asserted, the information on DAL<21:00>, BS<1:0>, and $\overline{MAP}$ lines should be ignored.

The write transaction is initiated by the assertion of the $\overline{ALE}$ line. This signal latches the AIO code, the physical address on DAL<21:00>, BS<1:0>, and the $\overline{MAP}$ (I/O MAP enable) signals. A bus write requires a mimimum of eight periods and can be stretched in two-period increments until the DCJ11 receives the $\overline{CONT}$ signal to end the transaction. The SCTL signal is asserted during the stretched portion and the write data is valid on leading and trailing edges of the $\overline{SCTL}$ signal. Sixteen bits of the DAL bus are used for byte write with the correct data on the low byte if the address is even and the correct data on the high byte if the address is odd. The data on the remaining byte is not defined.

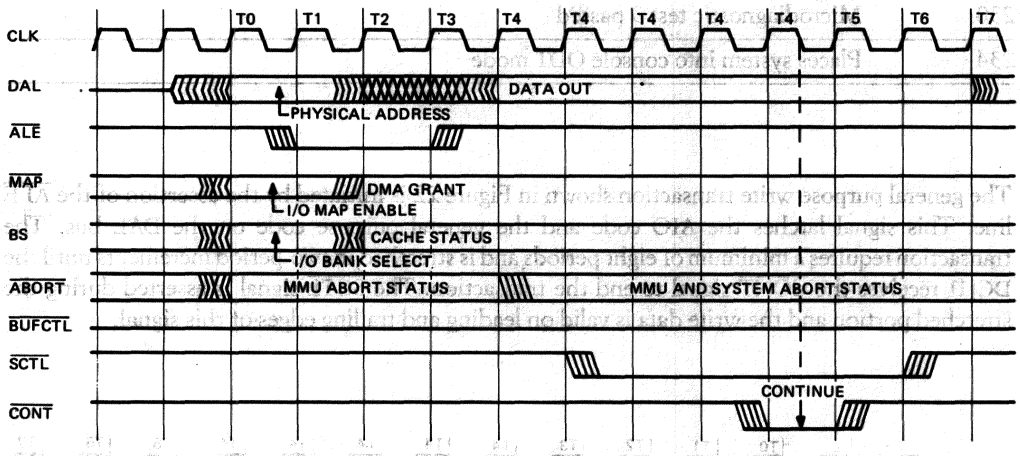


Figure 21 • DCJ11 Bus Write Cycle Timing Sequence

General Purpose Write Transactions

The general purpose write transaction is used to address non-PDP-11 interface hardware through the general purpose codes on DAL<07:00>. Either byte or word writes can be initiated as defined by the AIO code. The address on the DAL bus is 17600XXX. The XXX bits represent the general purpose write code. Table 19 lists the write code assignments for the general purpose read transaction.

Table 19 • DCJ11 General Purpose Write Code Assignments

Write Code	Function
003	Writes FPA 16-bit data
014	Asserts bus reset signal
034	Releases system from console ODT mode
040	Reserved for future use
100	Acknowledges EVENT
214	Negates bus reset signal
140	Acknowledges powerfail
220	Microdiagnostic test 1 passed
224	Microdiagnostic test 2 passed
230	Microdiagnostic test 3 passed
234	Places system into console ODT mode

The general purpose write transaction shown in Figure 22 is initiated by the assertion of the $\overline{\text{ALE}}$ line. This signal latches the AIO code and the general purpose code on the DAL bus. The transaction requires a minimum of eight periods and is stretched in two-period increments until the DCJ11 receives the $\overline{\text{CONT}}$ signal to end the transaction. The $\overline{\text{SCTL}}$ signal is asserted during the stretched portion and the write data is valid on leading and trailing edges of this signal.

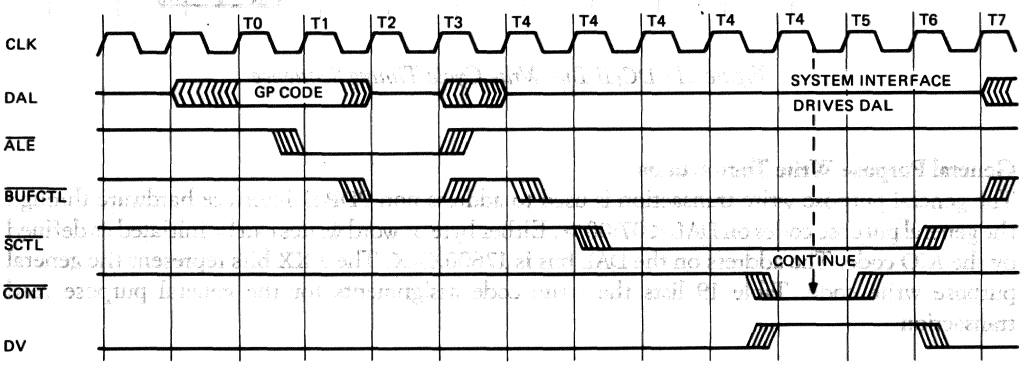


Figure 22 • DCJ11 General Purpose Write Cycle Timing Sequence

General Purpose Read Transaction

The general purpose write transaction is used to address non-PDP-11 interface hardware through the general purpose codes placed on the DAL<07:00> bus. Only words are read by the general purpose codes and if a byte is required, the excess byte is ignored. The address on the DAL is 17600XXX. The XXX bits represent the general purpose read code. Table 20 lists the read code assignments for the general purpose read transaction.

Table 20 • DCJ11 General Purpose Read Code Assignments

Read Code	Function
000	Reads the powerup mode, halt option, FPA option (FPA here), POKH signal, and boot address during initialization. Refer to Initialization section.
001	Reads FPA data.
002	Reads the powerup mode, halt option, FPA option, POK signal, and boot address, and clears FPA's FPS.
003	Acknowledges FPE and reads the floating exception code (FEC) register.

The general purpose read transaction, shown in Figure 23, is initiated by the assertion of the $\overline{\text{ALE}}$ line. This signal latches the AIO code and the general purpose code on the DAL bus. The transaction requires a minimum of eight periods and is stretched by two-period increments until the DCJ11 receives the $\overline{\text{CONT}}$ signal to end the transaction. The DCJ11 receives the DV signal and latches the data while $\overline{\text{SCTL}}$ is asserted.

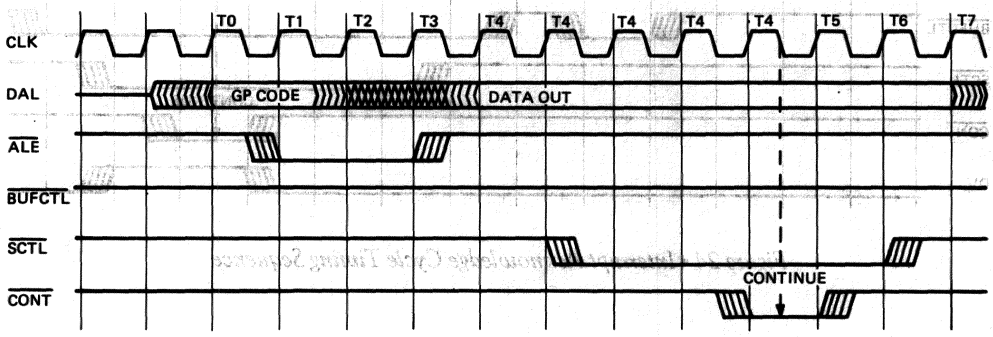


Figure 23 • DCJ11 General Purpose Read Cycle Timing Sequence

DMA Request and Grant Transaction

When the external system requests the use of the DAL bus or wants to stall the DCJ11, it asserts the $\overline{DMR}$ input. This disables the DCJ11 from the DAL bus and causes a stretched transaction. The DMR input is acknowledged after the I/O map information is on the $\overline{MAP}$ output. The $\overline{DMR}$ input is the DMA request and the $\overline{MAP}$ output is the DMA grant. These signals should be recognized during NOP or read transactions. The write transactions stretch beyond four periods and the DAL bus may contain write data. The DMA transfer stretches the transaction beyond eight periods by two period increments, until the DCJ11 receives the $\overline{CONT}$ signal to end the transaction.

Interrupt Acknowledge

The interrupt acknowledge transaction is used to acknowledge an interrupt request received through the IRQ<3:0> inputs. The vector address specified can be an internal predesignated address or an external address received on the DAL bus. The decoded interrupt level acknowledged is sent on the DAL<03:00> lines at the beginning of the transaction. The DAL<21:16> lines are set to one and DAL bits <15:04> are set to zero.

The interrupt acknowledge transaction shown in Figure 24, is initiated by the assertion of the $\overline{ALE}$ line that latches the AIO code and the acknowledged interrupt level. The transaction requires eight periods to read the vector address and can be stretched in two-period increments until the CONT input is asserted. The DV input is asserted to latch the interrupt vector address while the $\overline{SCTL}$ signal is asserted.

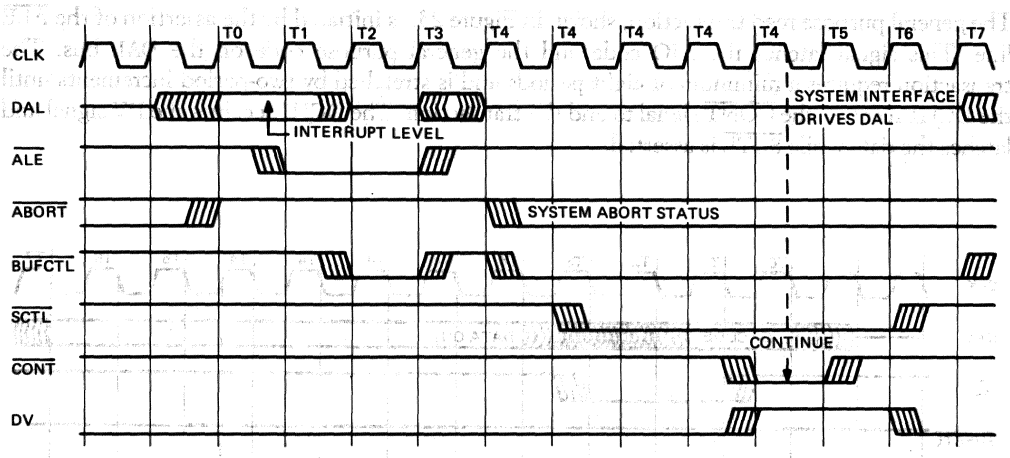


Figure 24 • Interrupt Acknowledge Cycle Timing Sequence

Initialization

The DCJ11 starts the initialization process when the system interface provides 5 volts (V_{CC}) and asserts $\overline{INIT}$ for a minimum of 25 clock periods. The $\overline{INIT}$ signal can be asserted by the system interface by using a power wakeup circuit. The initialization process can be implemented at any time the system interface asserts $\overline{INIT}$ as shown in Figure 25.

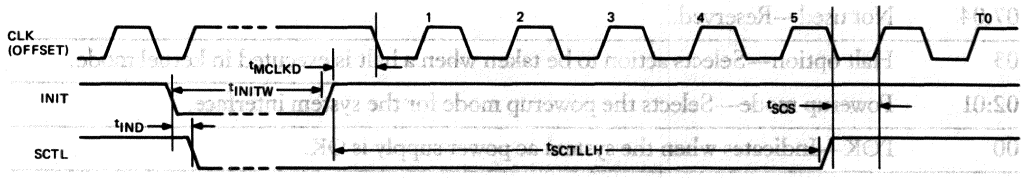


Figure 25 • DCJ11 Initialization Timing Sequence

The initialization process uses the powerup routine, the powerdown routine, and the routine that returns the DCJ11 from the console ODT mode. A 002 GP read transaction is performed during any routine, and the system interface provides the configuration data through the DAL<15:00>. The system interface provides this data by hardware, software, or firmware. Figure 26 shows the format of the information in the powerup configuration register and Table 21 lists the function of the information.

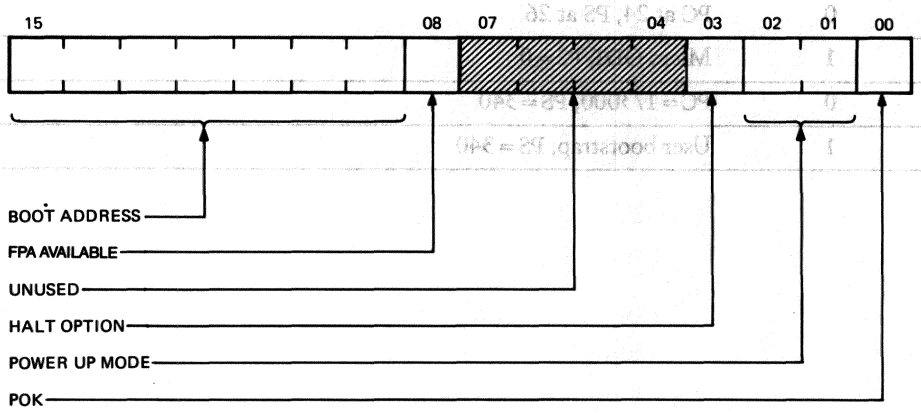


Figure 26 • DCJ11 Powerup Configuration Register Format

Table 21 • DCJ11 Powerup Configuration Register Description

Bits*	Function
15:09	Boot address—Provides the boot address as the PC for the users program when the powerup option #3 is used. The PSW is 340.
08	FPA available—Indicates the system interface is using a floating-point accelerator.
07:04	Not used—Reserved.
03	Halt option—Selects action to be taken when a halt is executed in kernel mode.
02:01	Powerup mode—Selects the powerup mode for the system interface.
00	POK—Indicates when the system ac power supply is OK.

*All bits are read-only

Powerup mode—Table 22 lists the four powerup options available to the user and selected by mode bits 02 and 01 of the configuration register.

Table 22 • DCJ11 Powerup Mode Selections

Powerup Bit 02	Powerup Bit 01	Mode Description
0	0	PC at 24, PS at 26
0	1	Micro ODT, PS = 0
1	0	PC = 173000, PS = 340
1	1	User bootstrap, PS = 340

- Powerup option 0—The processor reads physical memory locations 24 and 26 and loads the data into the PC and PSR, respectively. The processor services pending interrupts or starts program execution beginning at the memory location pointed to by the PC.
- Powerup option 1—The processor unconditionally enters Micro-ODT and the PSR is cleared. The pending service conditions are ignored.
- Powerup option 2—The processor sets the PC to 173000 and the PSR to 340. The processor then services pending interrupts or starts program execution beginning at the memory location defined by the PC. This option is used for the standard bootstrap procedure.
- Powerup option 3—The processor reads the value selected by the seven bootstrap address jumpers and loads the result into the PC<15:09>. PC<08:00> are cleared and the PSR is set to 340. The processor then services pending interrupts or starts the program execution beginning at the memory location pointed to by the PC.

Power OK—The power OK (POK) input is provided by the system interface to indicate that the ac supply is operating at the correct voltage. When bit 0 is set, the voltage is correct; when bit 0 is clear, the DCJ11 assumes the power is off.

Boot Address—The boot address reads the seven highest bits (15:09) of the starting address from the system interface. The remaining bits (08:00) are read as zeros. This allows the bootstrap address to reside on any 2,048-word boundary. The boot address is selected by powerup option 3.

FPA Available—The system interface sets this bit when a floating-point accelerator is in the system and is cleared when a floating-point accelerator is not included.

Halt—The halt option is used by the system interface to indicate the interpretation of the halt instruction in kernel mode. When set, this bit indicates that the processor will trap through vector address 4. When cleared, it indicates that the system will be placed into console ODT mode.

Initialization Sequence

The initialization sequences are shown in the following flow diagrams. The powerup routine is described in Figure 27, sheets 1 through 4. The powerdown routine is described in Figure 28. The return from the ODT routine is described in Figure 29, sheets 1 and 2. All these routines perform a variety of GP read and write transactions that are defined in the Bus Operation section.

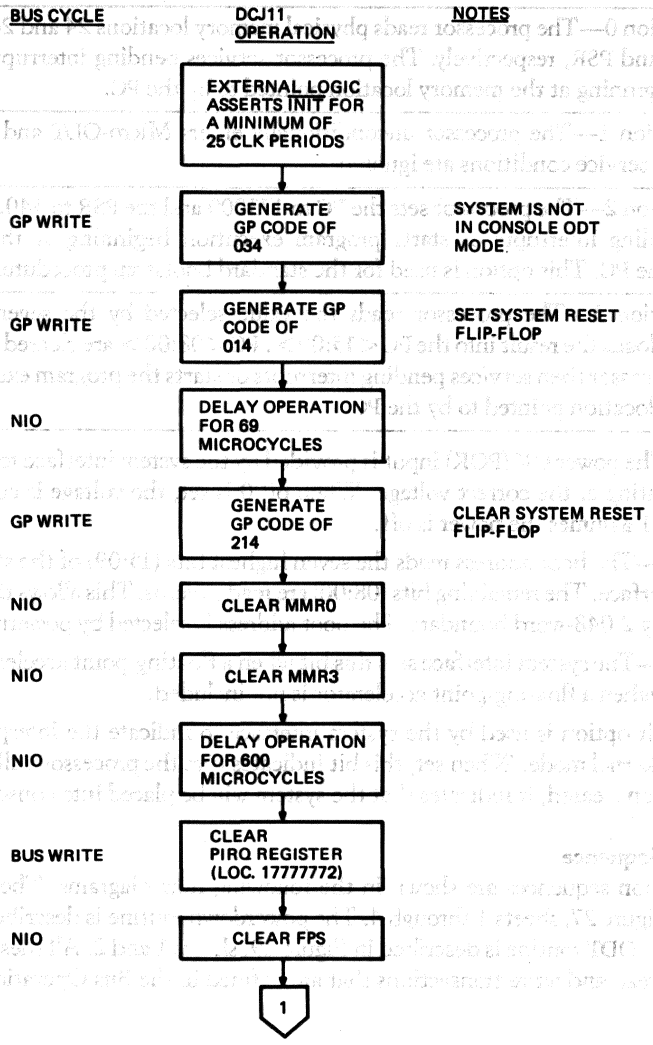


Figure 27 • DCJ11 Powerup Sequence Flow Diagram (sheet 1 of 4)

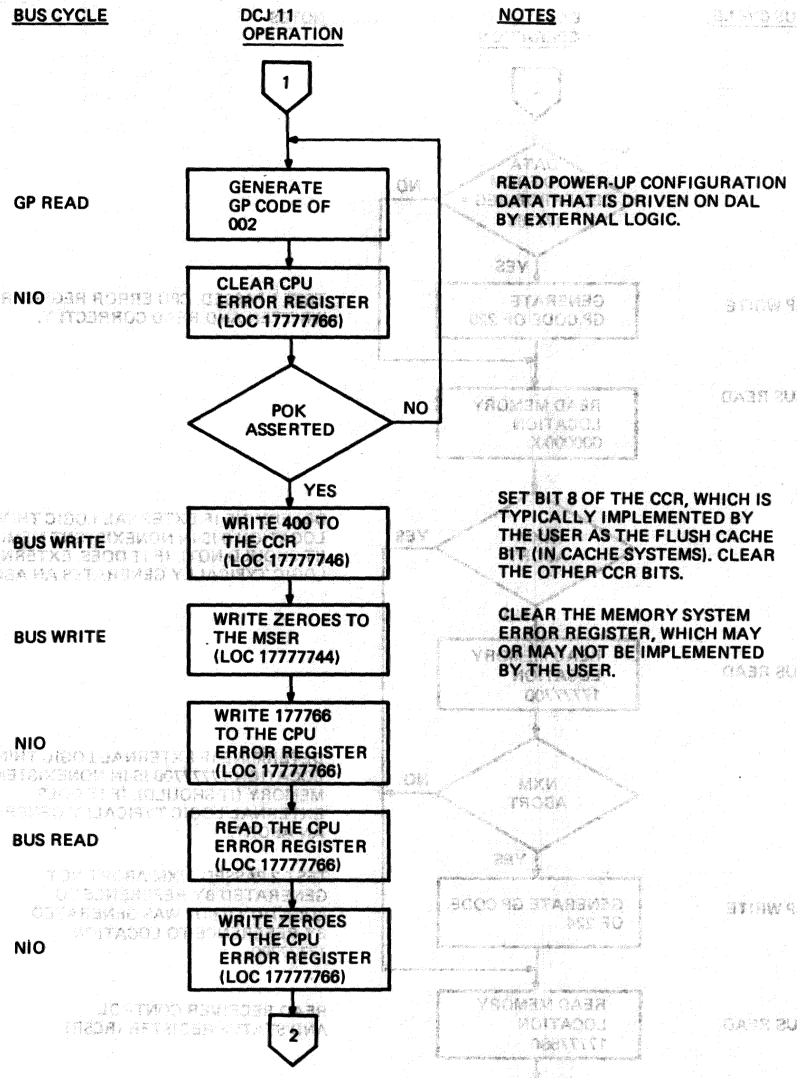


Figure 27 • DCJ11 Powerup Sequence Flow Diagram (sheet 2 of 4)

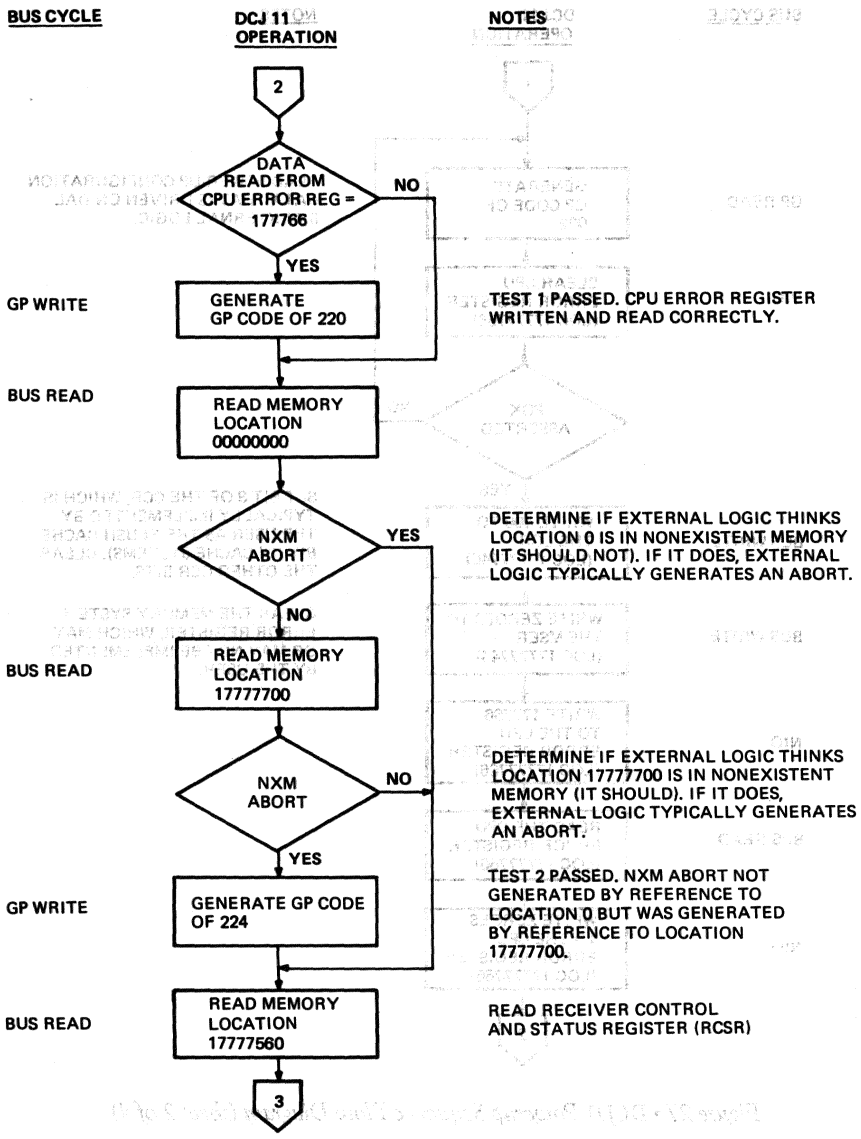


Figure 27 • DCJ11 Powerup Sequence Flow Diagram (sheet 3 of 4)

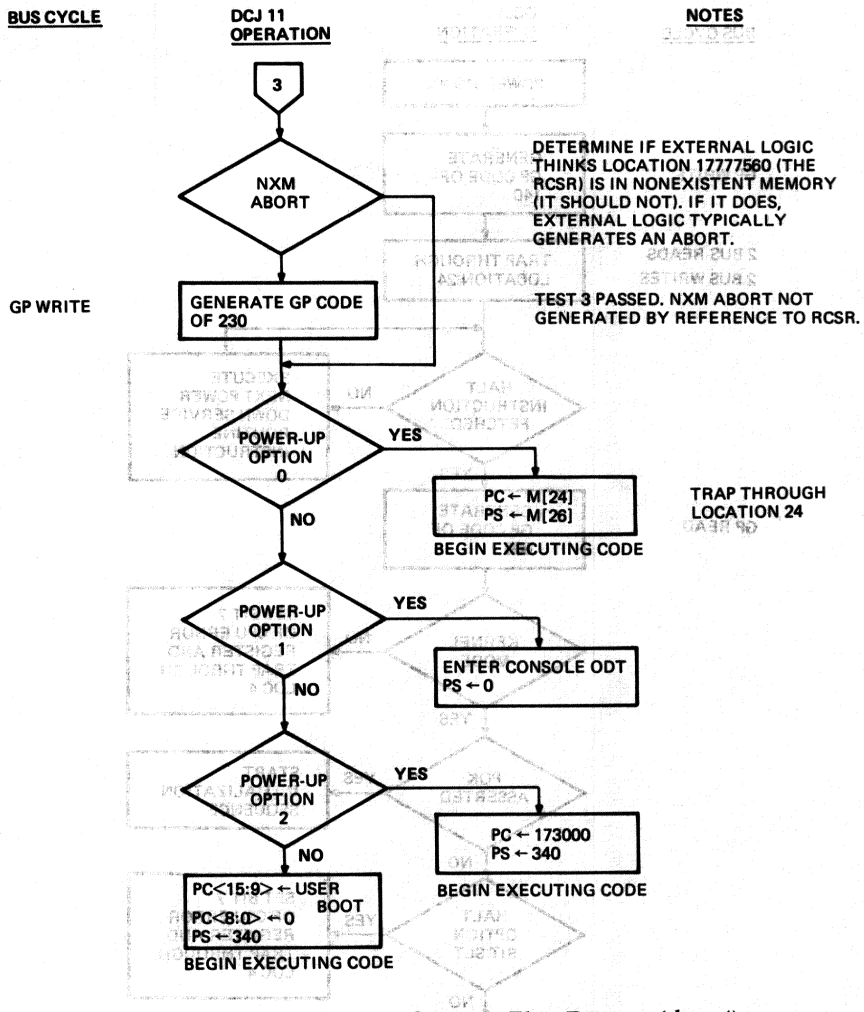


Figure 27 ▪ DCJ11 Powerup Sequence Flow Diagram (sheet 4)

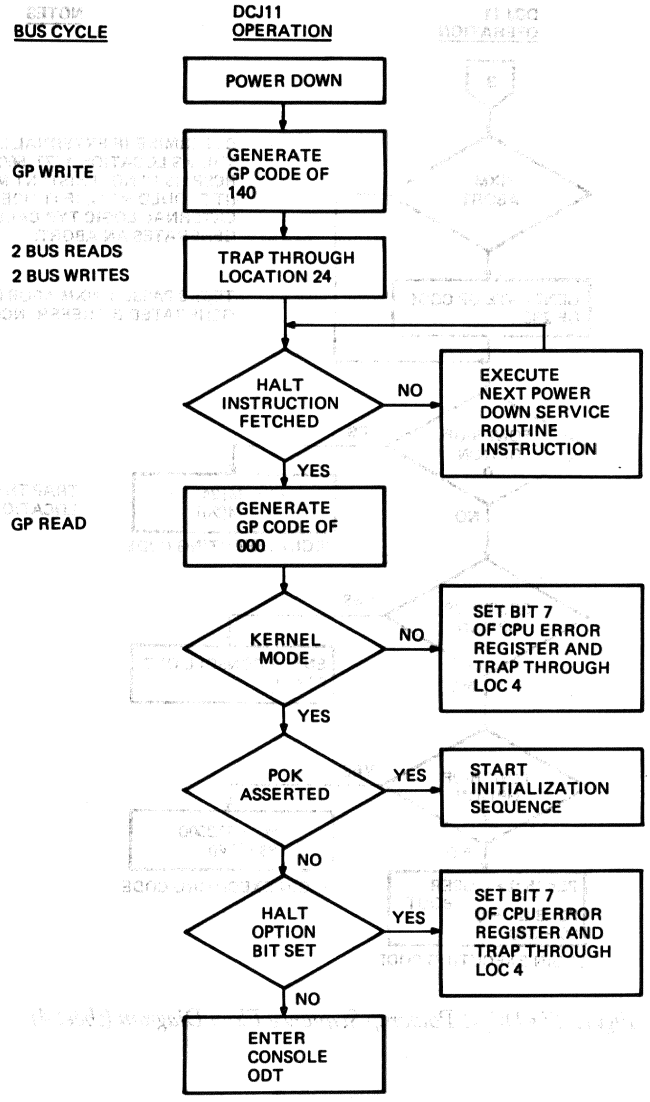


Figure 28 • DCJ11 Powerdown Sequence Flow Diagram

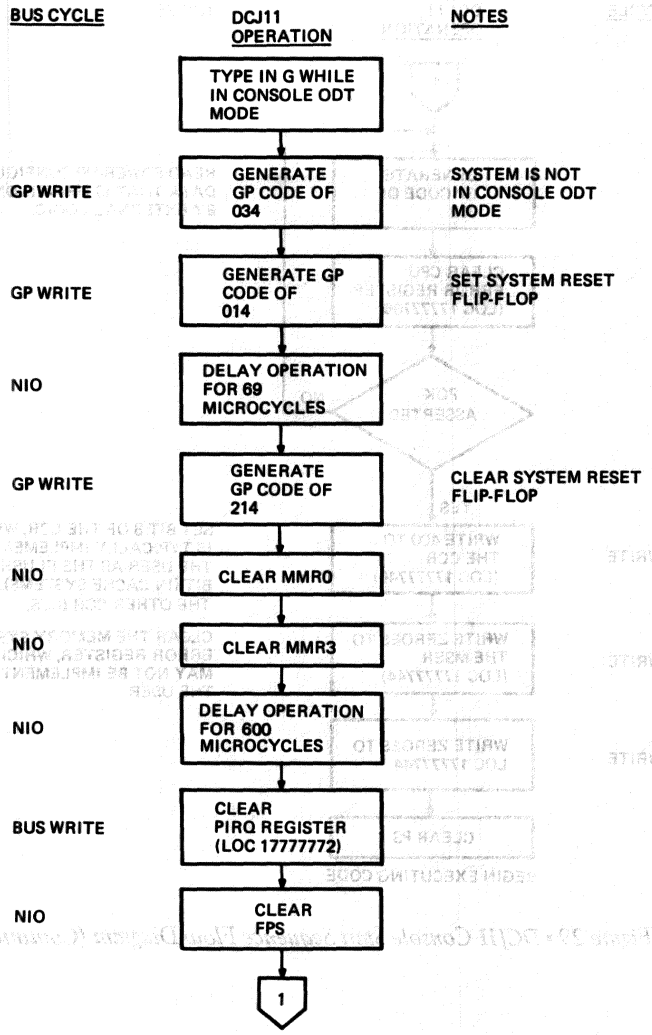


Figure 29 • DCJ11 Console Start Sequence Flow Diagram

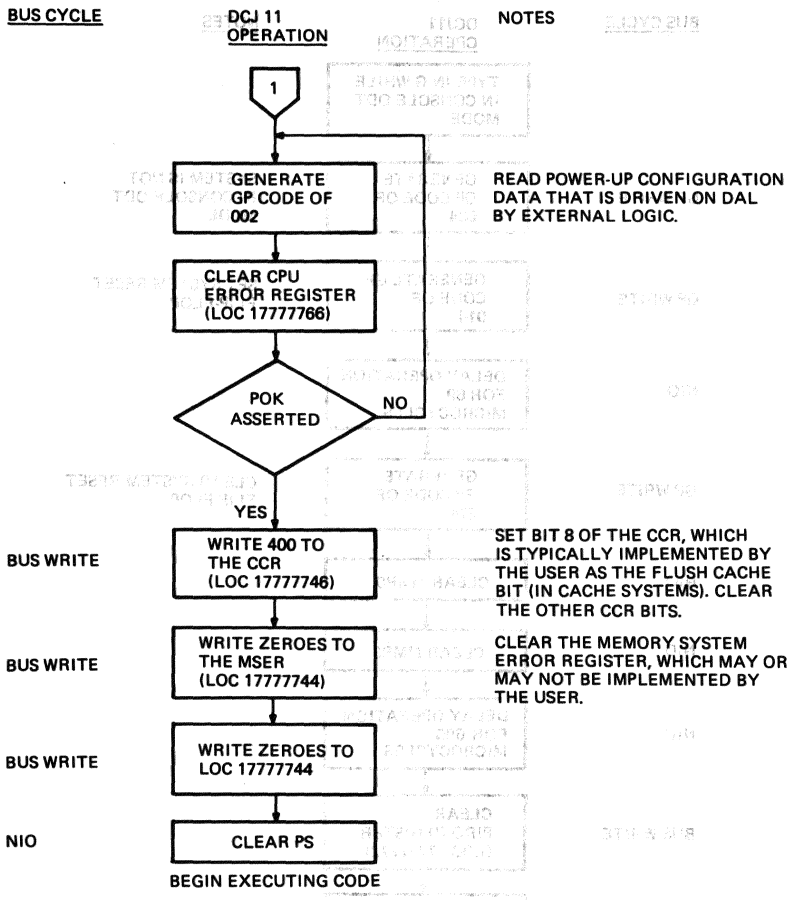


Figure 29 • DCJ11 Console Start Sequence Flow Diagram (Continued)

• Instruction Timing

The execution time for an instruction depends on the type of instruction executed, the mode of addressing used, and the type of memory being referenced. In general, the total execution time is the sum of the base instruction fetch and execute time plus the operand(s) address calculation/fetch time.

Tables 23 through 30 and the source and destination tables S1, D1 through D6, and F1 through F4 are used to determine the execution time of an instruction in microcycles. The execution times listed in the tables specify the number of microcycles required to fetch and execute the base instruction. The read/write (R/W) columns specify the number of read and write microcycles required. If the instruction involves a calculation or fetch operation of one or more operands, a source or destination table is referenced in the table. The source and destination tables specify the number of microcycles required to perform the calculation or fetch operation for the source or destination. It also lists the number of read and write microcycles required. During the remaining microcycles, no operations (NOP) are performed.

The table values are calculated assuming that a read from memory operation requires a minimum of four clock periods, a write to memory operation requires a minimum of eight clock periods, and a NOP instruction requires four clock periods when a DMA transaction is not initiated. The time for a wait state caused by a slow memory or resulting from a DMA transfer must be added to the total execution time. The first wait state of a nonstretched read cycle or NOP cycle requires four clock periods and can continue in increments of two clock periods. Additional wait states for stretched cycles occur in increments of two clock periods.

The execution time for floating-point instructions will vary depending on the type of data operation.

Instruction Timing Examples

The examples that follow show how the information from the tables is used to calculate the total time of an instruction.

Example 1: Determine the execution time of a MOV R0,@#2044 instruction.

1. The execution time for the MOV base instruction is 1 μc, or four CLK periods (Table 26). This consists of one read microcycle. The microcycle may be stretched depending upon the type of memory in the system. If the microcycle is stretched, it requires at least eight CLK periods and may be stretched thereafter in increments of two CLK periods.

2. To find the operand calculation and fetch time for the source operand R0, refer to Table S1. A mode 0, register 0 calculate and fetch requires 0 μc. Note that the operand is already available to the DCJ11 in the register file.

3. To find the operand calculation and fetch time for the destination operand (the contents of memory location 2044), refer to Table D3. A mode 3 register 7 calculate and fetch operation requires 3 μc. One is a read microcycle and one is a write microcycle. The remaining microcycle is an NOP microcycle. The type of memory in the system must be taken into account. If the read cycle is stretched, the stretched cycle requires a minimum of eight CLK periods and may be stretched thereafter in increments of two CLK periods. The duration of a write microcycle is a minimum of eight CLK periods and may be increased in increments of two CLK periods.

4. To determine the minimum time required, total up the microcycles. In this example, it is 1+0+3, or 4 μc (which is 16 CLK periods if no microcycle stretching occurs).

Example 2: The source and destination tables for floating-point instructions show a negative number in the microcycle column for certain mode 2 register 7 operations. This example shows a timing calculation for one of these.

Determine the execution time of a CLRD #2000 instruction.

1. The base instruction time for the CLRD instruction is 14 μc.

2. From Table F2, the calculation and fetch time for the operand (a mode 2 register 7 reference) is −1. 1 μc is subtracted from the base instruction time. For the memory write operation 1 μc must be added to this value. There are no memory read cycles to account for.

3. The total of the microcycles is 14−1+1=14 μc minimum. This value assumes the cycle is not stretched.

Table 23 • DCJ11 Single Operand Instruction Execution Times

Mnemonic	Instruction	Opcode	Execution Time (μc)	R/W	Source Table	Destination Table
General						
CLR(B)	Clear	0050DD	1	1/0	—	D3
COM(B)	Complement (one's)	0051DD	1	1/0	—	D4
INC(B)	Increment	0052DD	1	1/0	—	D4
DEC(B)	Decrement	0053DD	1	1/0	—	D4
NEG(B)	Negate (two's complement)	0054DD	1	1/0	—	D4
TST(B)	Test	0057DD	1	1/0	—	D4
Rotate and Shift						
ROR(B)	Rotate right	0060DD	1	1/0	—	D4
ROL(B)	Rotate left	0061DD	1	1/0	—	D4
ASR(B)	Arithmetic shift right	0062DD	1	1/0	—	D4
ASL(B)	Arithmetic shift left	0063DD	1	1/0	—	D4
SWAB	Swap bytes	0003DD	1	1/0	—	D4
Multiple-Precision						
ADC(B)	Add carry	0055DD	1	1/0	—	D4
sbC(B)	Subtract carry	0056DD	1	1/0	—	D4
SXT	Sign extend	0067DD	1	1/0	—	D3
Multiprocessing						
TSTSET	Test and set (low bit interlocked)	0072DD	5	1/1	—	D4
WRTLCK	Write interlocked	0073DD	4	1/1	—	D4

Table 24 • DCJ11 Double Operand Instruction Execution Times

Mnemonic	Instruction	Opcode	Execution Time (μc)	R/W	Source Table	Destination Table
General						
MOV(B)	Move	01SSDD	1	1/0	S1	D3
CMP(B)	Compare	02SSDD	1	1/0	S1	D2
ADD	Add	06SSDD	1	1/0	S1	D4
SUB	Subtract	16SSDD	1	1/0	S1	D4
Logical						
BIT(B)	Bit test (AND)	03SSDD	1	1/0	S1	D2
BIC(B)	Bit clear	04SSDD	1	1/0	S1	D4
BIS(B)	Bit set (OR)	05SSDD	1	1/0	S1	D4
Register						
MUL	Multiply	070RSS	22	1/0[5,11]	—	D1
DIV	Divide	071RSS	34	1/0[6,7,12]	—	D1
ASH	Shift automatically	072RSS	4	1/0	—	D1
ASHC	Arith shift combined	073RSS	5	1/0[13]	—	D1
XOR	Exclusive (OR)	074RDD	34	1/0	—	D4

Table 25 • DCJ11 Branch Instruction Execution Times

Mnemonic	Instruction	Opcode	Branch Not Taken Time (μc)	R/W	Branch Taken Time (μc)	R/W
Branches						
BR	Branch unconditional	000400	2	1/0	4	2/0
BNE	Br if not equal to 0	001000	2	1/0	4	2/0
BEQ	Br if equal to 0	001400	2	1/0	4	2/0
BPL	Br if plus	100000	2	1/0	4	2/0
BMI	Br if minus	100400	2	1/0	4	2/0
BVC	Br if overflow is clear	102000	2	1/0	4	2/0
BVS	Br if overflow is set	102400	2	1/0	4	2/0
BCC	Br if carry is clear	103000	2	1/0	4	2/0
BCS	Br if carry is set	103400	2	1/0	4	2/0
Signed Conditional Branches						
BGE	Br if greater or equal to 0	020000	2	1/0	4	2/0
BLT	Br if less than 0	002400	2	1/0	4	2/0
BGT	Br if greater than 0	003000	2	1/0	4	2/0
BLE	Br if less or equal to 0	003400	2	1/0	4	2/0
Unsigned Conditional Branches						
BHI	Br if higher	101000	2	1/0	4	2/0
BLOS	Br if lower or same	101400	2	1/0	4	2/0
BHIS	Br if higher or same	103000	2	1/0	4	2/0
BLO	Br if lower	103400	2	1/0	4	2/0
SOB	Subtract 1 and branch if not equal to 0	077RNN	3	1/0	5	2/0

Table 26 • DCJ11 Jump and Subroutine Instruction Execution Times

Mnemonic	Instruction	Opcode	Execution MC	R/W	Destination Table
JMP	Jump	0001DD	—	—	D5
JSR	Jump to subroutine	004RDD	—	—	D6[4]
RTS	Return from subroutine	00020R	5	3/0	14
MARK	Stack cleanup	0064NN	10	3/0	

Table 27 • DCJ11 Trap and Interrupt Instruction Execution Time

Mnemonic	Instruction	Opcode	Execution Time (μc)	R/W
EMT	Emulator trap	104000—104377	20	4/2
TRAP	Trap	104400—104777	20	4/2
BPT	Breakpoint trap	000003	20	4/2
IOT	Input/output trap	000004	20	4/2
RTI	Return from interrupt	000002	9	4/0
RTT	Return from interrupt	000006	9	4/0

Table 28 • DCJ11 Condition Code Operators Instruction Execution Times

Mnemonic	Instruction	Opcode	Execution Time (μc)	R/W
CLC	Clear C	000241	3	1/0
CLV	Clear V	000242	3	1/0
CLZ	Clear Z	000244	3	1/0
CLN	Clear N	000250	3	1/0
CCC	Clear all CC bits	000257	3	1/0
SEC	Set C	000261	3	1/0
SEV	Set V	000262	3	1/0
SEZ	Set Z	000264	3	1/0
SEN	Set N	000270	3	1/0
SCC	Set all CC bits	000277	3	1/0

Table 29 • DCJ11 Miscellaneous Instruction Execution Time

Mnemonic	Instruction	Opcode	Execution Time (μc)	R/W	Destination Table
HALT	Halt	000000	—		—
WAIT	Wait for interrupt	000001	—		—
RESET	Reset external bus	000005	—		—
NOP	(No operation)	000240	3	1/0	—
SPL	Set priority level to N	00023N	7	1/0	—
MFPI	Move from previous instr space	0065DD	5	1/1	D1
MTPI	Move to previous instr space	0066SS	3	2/0	D3
MFPD	Move from previous data space	1065DD	5	1/1	D1
MTPD	Move to previous data space	1066SS	3	2/0	D3
MTPS	Move byte to PSW PS—(svc)	1064SS	8	1/0	D1
MFPS	Move byte from PSW (dst)—PS 07:00	1067DD	1	1/0	D3
MFPT	Move from processor (R0 07:00)—proc code	000007	2	1/0	—
CSM	Call to supervisor mode	0070DD	28	3/3	D1

Table 30 • DCJ11 Floating-point Instruction Execution Time

Mnemonic	Instruction	Opcode	Timing (μc) Min.	Non Mode 0 Typ.	Max.	Table
ABSD	Make absolute	1706 fdst	23		24	F3
ABSF	Make absolute	1706 fdst	19		20	F3
ADDD	Add	172 (ac) fsrc	41	48	119	F1
ADDF	Add	172 (ac) fsrc	31	35	102	F1
CFCC	Copy floating condition codes	170000	5		5	—
CLRC	Clear	1704 fdst	14		14	F2
CLRF	Clear	1704 fdst	12		12	F2
CMPD	Compare	173(ac+4)	24		25	F1
CMPF	Compare	173(ac+4)	18		19	F1
DIVD	Divide	174(ac+4)	160		167	F1
DIVF	Divide	174(ac+4)	59		63	F1

Mnemonic	Instruction	Opcode	Timing (μc) Min.	Non Mode 0 Typ.	Max.	Table
LDCDF	Ld and C from D to F	177(ac+4)	24		26	F1
LDCFD	Ld and C from F to D	177(ac+4)	20		21	F1
LDCID	Ld and C Integer to D	177(ac) src	31		42	F4
LDCIF	Ld and C Integer to F	177(ac) src	26		36	F4
LDCLD	Ld and C Long Integer to D	177(ac) src	31		52	F4
LDCLF	Ld and C Long Integer to F	177(ac) src	26		44	F4
LDD	Load	172(ac+4)	16		17	F1
FDEXP	Load exponent	176(ac+4)	17		18	F4
LDF	Load	172(ac+4)	12		13	F1
LDFPS	Load FPP program status	1701 src	6		6	F4
MODD	Multiply and separate	171(ac+4)	202	217	268	F1
MODF	Integer and fraction	171(ac+4)	82	94	115	F1
MULD	Multiply	171(ac) fsrc	165		173	F1
MULF	Multiply	171(ac) fsrc	56		61	F1
NEGD	Negate	1707 fdst	22		23	F3
NEGE	Negate	1707 fdst	18		19	F3
SETD	Set floating double mode	170011	6		6	—
SETF	Set floating mode	170001	6		6	—
SETI	Set integer mode	170002	6		6	—
SETL	Set long integer mode	170012	6		6	—
STCDF	St and C from D to F	176(ac) fdst	17		20	F2
STCDI	St and C from D to integer	176(ac) fdst	26		38	F5
STCDL	St and C from D to long integer	176(ac) fdst	26		54	F5
STCFD	St and C from F to D	176(ac) fdst	19		20	F2
STCFI	St and C from F to integer	175(ac+4)	23		35	F5
STCFL	St and C from F to long integer	175(ac+4)	23		51	F5
STD	Store	174(ac) fdst	12		12	F2
STEXP	Store exponent	175(ac) dst	16		16	F5
STF	Store	174(ac) fdst	8		8	F2

Mnemonic	Instruction	Opcode	Timing (µc) Min.	Non Mode 0 Typ.	Max.	Table
STFPD	Store FPP program status	1702 dst	9		9	F5
STST	Store FPP status	1703 dst	7		7	F5
SUBD	Subtract	173(ac) fsrc	47	55	122	F1
SUBF	Subtract	173(ac) fsrc	37	41	104	F1
TSTD	Test	1705 fdst	11		12	F1
TSTF	Test	1705 fdst	9		10	F1

The following notes refer to the source and destination tables that follow.

1. Subtract 2 microcycles (µc) and one read cycle if both the source and destination modes autodecrement the PC, or if write-only or read-modify-write mode 07 or 17 is used.

2. Read-only and read-modify-write destination mode 47 references actually perform three READ operations. One of the READ operations is accounted for in the execute, fetch timing.

3. Read-only and read-modify-write destination mode 57 references actually perform four READ operations. One of the READ operations is accounted for in the execute, fetch timing.

4. Subtract 1 µc if the link register is the PC.

5. Add 1 µc if the source operand is negative.

6. Subtract 1 µc if the source mode is not zero.

7. Add 1 µc if the quotient is even. Add 2 µc if an overflow occurs. Add 5 µc and 1 read cycle if the PC is used as a destination register and source mode 47 or 57 is not used.

8. Add 1 µc per shift.

9. Add 1 µc if the source operand bits 15:06 is not zero.

10. Subtract 1 µc if one shift only.

11. Add 4 µc and 1 read cycle if the PC is used as a destination register and source mode 47 or 57 is not used.

12. The divide by zero execution time is 5 µc (see Note 6).

13. Timing for no shift. Add 1 µc if a left shift. (Notes 8, 9, 11 apply.) Add 2 µc for a right shift. (Notes 8, 10, 11 apply.)

14. Add 1 µc if a register other than R7 is used.

15. Mode 27 references access single word operands only. The execution time listed has been adjusted to accurately reflect the total execution time.

Table S1 • DCJ11 Source Address Time (Double-operand) Cycle Time

Source Mode	Source Register	Microcode Cycles	Read Memory Cycles
0	0-7	0	0
1	0-7	2	1
2	0-6	2	1
2	7	1	1
3	0-6	4	2
3	7	3	2
4	0-6	3	1
4	7	6	2[1]
5	0-6	5	2
5	7	8	3[1]
6	0-7	4	2
7	0-7	6	3

Table D1 • DCJ11 Destination Address (Read-only Single-operand) Cycle Time

Destination Mode	Destination Register	Microcode Cycles	Read Memory Cycles
0	0-7	0	0
1	0-7	2	1
2	0-6	2	1
2	7	1	1
3	0-6	4	2
3	7	3	2
4	0-6	3	1
4	7	7	2[2]
5	0-6	5	2
5	7	9	3[3]
6	0-7	4	2
7	0-7	6	3

Table D2 • DCJ11 Destination Address (Read-only Double-operand) Cycle Time

Destination Mode	Destination Register	Microcode Cycles	Read Memory Cycles
0	0-7	0	0
1	0-7	3	1
2	0-6	3	1
2	7	2	1
3	0-6	5	2
3	7	4	2
4	0-6	4	1
4	7	8	2[2]
5	0-6	6	2
5	7	10	3[3]
6	0-7	5	2
7	0-7	7	3

Table D3 • DCJ11 Destination Address (Write-only) Cycle Time

Destination Mode	Destination Register	Microcode Cycles	Memory Cycles Read	Memory Cycles Write
0	0-6	0	0	0
0	7	5	1	0
1	0-6	2	0	1
1	7	6	1	1
2	0-6	2	0	1
2	7	6	1	1
3	0-6	4	1	1
3	7	3	1	1
4	0-6	3	0	1
4	7	7	1	1
5	0-6	5	1	1
5	7	9	2	1
6	0-7	4	1	1
7	0-7	6	2	1

Table D4 • DCJ11 Destination Address (Read-modify-write) Cycle Time

Destination Mode	Destination Register	Microcode Cycles	Memory Read	Cycles Write
0	0-6	0	0	0
0	7	5	1	0
1	0-6	3	1	1
1	7	7	2	1
2	0-6	3	1	1
2	7	7	2	1
3	0-6	5	2	1
3	7	4	2	1
4	0-6	4	1	1
4	7	8	2	1[2]
5	0-6	6	2	1
5	7	10	3	1[3]
6	0-7	5	2	1
7	0-7	7	3	1

Table D5 • DCJ11 Destination Address (Jump) Cycle Time

Destination Mode	Destination Register	Microcode Cycles	Memory Read	Cycles Write
1	0-7	4	2	0
2	0-7	6	2	0
3	0-7	5	3	0
4	0-7	5	2	0
5	0-7	6	3	0
6	0-6	6	3	0
6	7	5	3	0
7	0-7	7	4	0

Table D6 • DCJ11 Destination Address (Jump-to-subroutine) Cycle Time

Destination Mode	Destination Register	Microcode Cycles	Memory Read	Cycles Write
1	0-7	9	2	1
2	0-7	10	2	1
3	0-6	10	3	1
3	7	9	3	1
4	0-7	10	2	1
5	0-7	11	3	1
6	0-6	10	3	1
6	7	9	3	1
7	0-7	12	4	1

Table F1 • DCJ11 Floating Source Modes 1 through 7 Cycle Time

Microcode Mode	Memory Register	Memory Cycles	Read	Write
Single Precision				
1	0-7	3	2	0
2	0-6	3	2	0
2	7	1	1	0
3	0-6	4	3	0
3	7	3	3	0
4	0-7	4	2	0
5	0-7	5	3	0
6	0-7	4	3	0
7	0-7	6	4	0
Double Precision				
1	0-7	5	4	0
2	0-6	5	4	0
2	7	0^{15}	1	0
3	0-6	6	5	0
3	7	5	5	0
4	0-7	6	4	0
5	0-7	7	5	0
6	0-7	6	5	0
7	0-7	8	6	0

Table F2 • DCJ11 Floating Destination Modes 1 through 7 Cycle Time

Microcode Mode	Memory Register	Memory Cycles	Read	Write
Single Precision				
1	0-7	3	0	2
2	0-6	3	0	2
2	7	1	0	1
3	0-6	4	1	2
3	7	3	1	2
4	0-7	4	0	2
5	0-7	5	1	2
6	0-7	4	1	2
7	0-7	6	2	2
Double Precision				
1	0-7	5	0	4
2	0-6	5	0	4
2	7	$(-1)^{15}$	0	1
3	0-6	6	1	4
3	7	5	1	4
4	0-7	6	0	4
5	0-7	7	1	4
6	0-7	6	1	4
7	0-7	8	2	4

Table F3 • DCJ11 Floating Read-modify-write Modes 1 through 7 Cycle Time

Microcode Mode	Memory Register	Memory Cycles	Read	Write
Single Precision				
1	0-7	5	2	2
2	0-6	5	2	2
2	7	1^{15}	1	1
3	0-6	6	3	2
3	7	5	3	2
4	0-7	6	2	2
5	0-7	7	3	2
6	0-7	6	3	2
7	0-7	8	4	2
Double Precision				
1	0-7	9	4	4
2	0-6	9	4	4
2	7	-2^{15}	1	1
3	0-6	10	5	4
3	7	9	5	4
4	0-7	10	4	4
5	0-7	11	5	4
6	0-7	10	5	4
7	0-7	12	6	4

Table F4 - DCJ11 Integer Source Modes 1 through 7 Cycle Time

Microcode Mode	Memory Register	Memory Cycles	Read	Write
Integer				
1	0-7	2	1	0
2	0-6	2	1	0
2	7	0^{15}	1	0
3	0-6	3	2	0
3	7	2	2	0
4	0-7	3	1	0
5	0-7	4	2	0
6	0-7	3	2	0
7	0-7	5	3	0
Long Integer				
1	0-7	4	2	0
2	0-6	4	2	0
2	7	0^{15}	1	0
3	0-6	5	3	0
3	7	4	3	0
4	0-7	5	2	0
5	0-7	6	3	0
6	0-7	5	3	0
7	0-7	7	4	0

Table F5 • DCJ11 Integer Destination Modes 1 through 7 Cycle Time

Microcode Mode	Memory Register	Memory Cycles	Read	Write
Integer				
1	0-7	4	0	2
2	0-6	4	0	2
2	7	2	0	1
3	0-6	5	1	2
3	7	4	1	2
4	0-7	5	0	2
5	0-7	6	1	2
6	0-7	5	1	2
7	0-7	7	2	2
Long Integer				
1	0-7	2	0	1
2	0-6	2	0	1
2	7	2	0	1
3	0-6	3	1	1
3	7	2	1	1
4	0-7	3	0	1
5	0-7	4	1	1
6	0-7	3	1	1
7	0-7	5	2	1

Instruction Set

Refer to Appendix B for a complete list of the DCJ11 microprocessor instruction set.

Specifications

The mechanical, electrical, and environmental characteristics and specifications for the DC319-AA are described in the following paragraphs. The test conditions used for the electrical values listed are as follows unless specified otherwise. Refer to Digital specification A-PS-2100002-GS for the general specifications for integrated circuits.

- Operating temperature (T_A): 70°C
- Power supply voltage (V_{CC}): 4.75 V
- (V_{SS}): 0 V

Mechanical Configuration

The physical dimensions of the DCJ11 60-pin ceramic DIP are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device. These ratings are for stress conditions only and do not imply that the device will function properly at these ratings or ratings above those indicated.

- Power supply voltage (V_{CC}): 7.0 V
- Input or output voltage applied (V_{SS}): –0.3 V, (V_{CC}): 0.3 V
- Active temperature: –55°C to 125°C
- Storage temperature: –65°C to 150°C

Recommended Operating Conditions

- Temperature range: 0°C to 70°C
- Voltage range: 5 V ±5%
- Relative humidity: 10% to 95% (noncondensing)

dc Electrical Characteristics

Table 31 contains the dc electrical parameters for the input and outputs of the DCJ11 for the specified operating voltage and temperature ranges. Refer to Appendix C for test circuit configurations referenced in the tables and used to perform the tests. Table 32 lists the applicable dc tests required for the input and outputs of the DCJ11.

Table 31 • DCJ11 dc Input and Output Parameters

Symbol	Parameter	Test Condition	Requirements Min.	Max.	Units	Test Circuit
V_{IH}	High-level MOS input		70% V_{CC}	—	V	C1,C2
V_{IL}	Low-level MOS input		—	30% V_{CC}	V	C1,C2
V_{IHT}	High-level TTL input		2.2	—	V	C1,C2
V_{ILT}	Low-level TTL input		0.6		V	C1,C2
I_I	Input-leakage current non-Test inputs	$0\ V \leq VI \leq V_{CC}$ $V_{CC}=5.25\ V$	–10	10	µA	C3,C4
I_{ILL}	Input current Test inputs	$V_{IN}=0\ V$ $V_{CC}=5.25\ V$	0.1	5.0	mA	C5
I_{OH}	Output current at high level	$V_{OUT}=V_{CC}-0.4\ V$	–2.0	—	mA	C1
I_{OL}	Output current at low level	$V_{OUT}=0.4\ V$	2.0	—	mA	C1,C2
I_{OHT}	Output current at high TTL level	$V_{OUT}=2.4\ V$	–2.0	—	mA	C2
I_{OSH}	High level sustainer current	$V_{OUT}=V_{CC}-1.0\ V$ $V_{CC}=5.25\ V$	–0.2	–0.6	mA	C6
I_{OSL}	Low level sustainer current	$V_{OUT}=1.0\ V$ $V_{CC}=5.25\ V$	0.2	0.8	mA	C6
I_{OZ}	Output leakage current[1]	$0\ V \leq VO \leq V_{CC}$ $V_{CC}=5.25\ V$	–10.0	10.0	µA	C8,C9
I_{CCSB}	Static power supply current[2]	$V_{CC}=5.25\ V$	—	30.0	mA	C7
C_{in}	Input only capacitance[3]		—	7.0	pF	

Symbol	Parameter	Test Condition	Requirements Min.	Max.	Units	Test Circuit
C_{IO}	Input/output capacitance[3]		—	15	pF	
C_{out}	Output capacitance[3]		—	15	pF	
C_{max}	DCJ11 capacitance plus external capacitance		—	100	pF	

[1]Applies only in the high-impedance condition.
[2]With $\overline{TEST1}$, $\overline{TEST2}$, and all outputs open circuit. All other inputs equal to V_{CC}.
[3]Sampled and guaranteed, but not tested. Does not apply to $\overline{TEST1}$ or $\overline{TEST2}$.

Table 32 • DCJ11 dc Signal Test Summary

Type	Name	Applicable dc Test
TTL input	IRQ<3:1>, HALT, $\overline{PWRF}$, $\overline{EVENT}$, $\overline{PARITY}$ DV, $\overline{MISS}$, $\overline{CONT}$, $\overline{DMR}$, $\overline{INIT}$ and $\overline{FPE}$	V_{IHT}, V_{ILT}, I_I
TTL output	$\overline{DAL}$<21:16>, AIO<3:0>, $\overline{ALE}$, $\overline{BUFCTL}$, $\overline{SCTL}$, $\overline{STRB}$, BS<1:0>, $\overline{MAP}$, and $\overline{PRDC}$	I_{OL}, I_{OHT}, I_{OZ}
MOS input	$\overline{TEST1}$ and $\overline{TEST2}$	V_{IH}, T_{IL}, I_{ILL}
MOS output	CLK and CLK2	I_{OH}, I_{OL}, I_{OZ}
TTL I/O	$\overline{ABORT}$*	V_{ILT}, I_{OL}, I_{OHT}, I_{OZ}, I_{OSH}
TTL I/O	DAL<15:00>	V_{IHT}, V_{ILT}, I_{OL}, I_{OHT}, I_{OZ}
Power	V_{CC}	I_{CCSB}

*$\overline{ABORT}$ must be driven with an open-collector driver because the DCJ11 has a pullup device that supplies I_{OSH}.

ac Electrical Characteristics

The timing references and signal parameters of the DCJ11 are shown in the following figures and tables. Figure 30 shows the input and output voltage waveform characteristics. The test conditions used to perform the ac measurements follow: Figure 33 shows the output load circuits referenced on the tables and used to perform the output measurements.

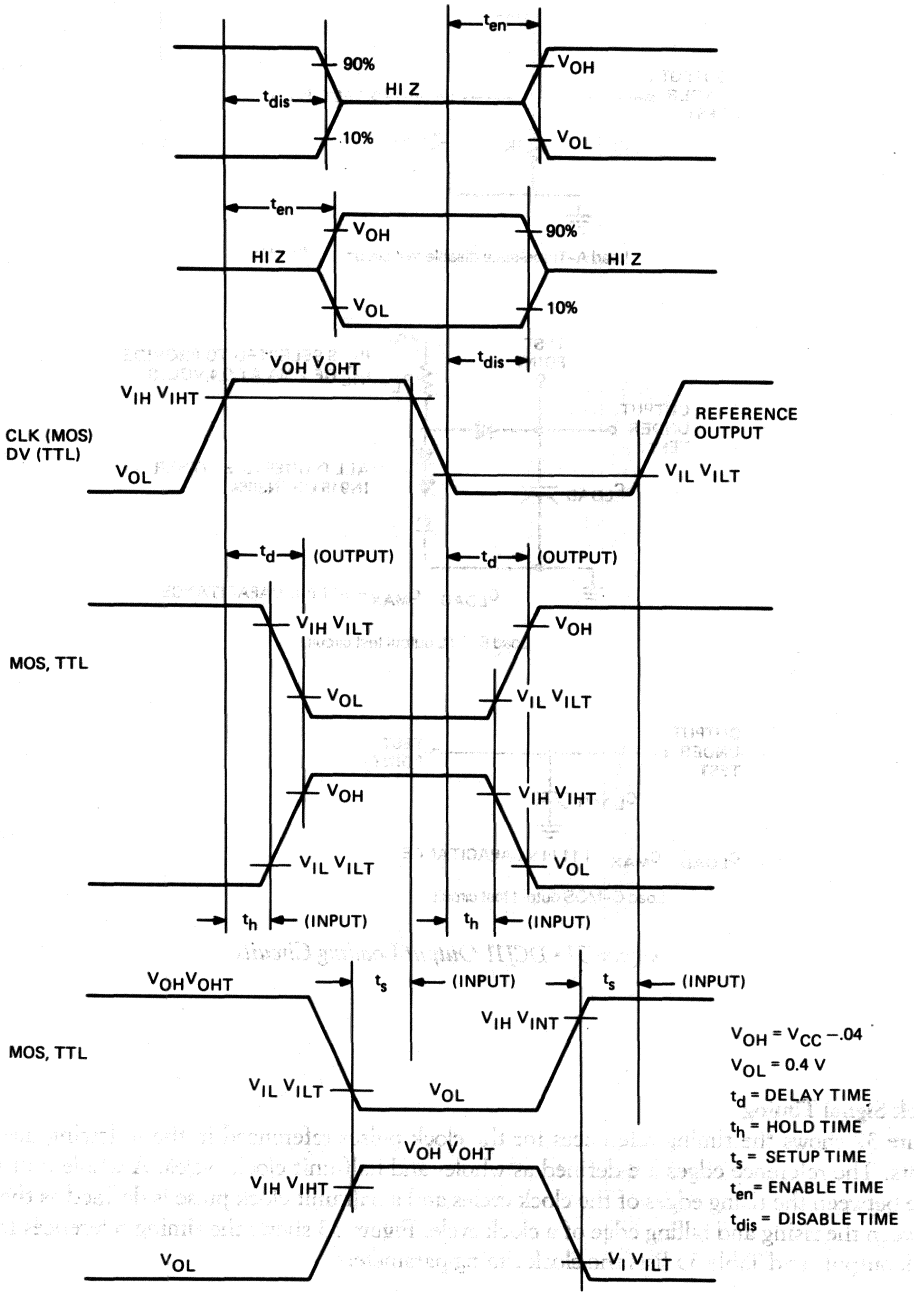


Figure 30 ▪ DCJ11 Input and Output Voltage Waveforms

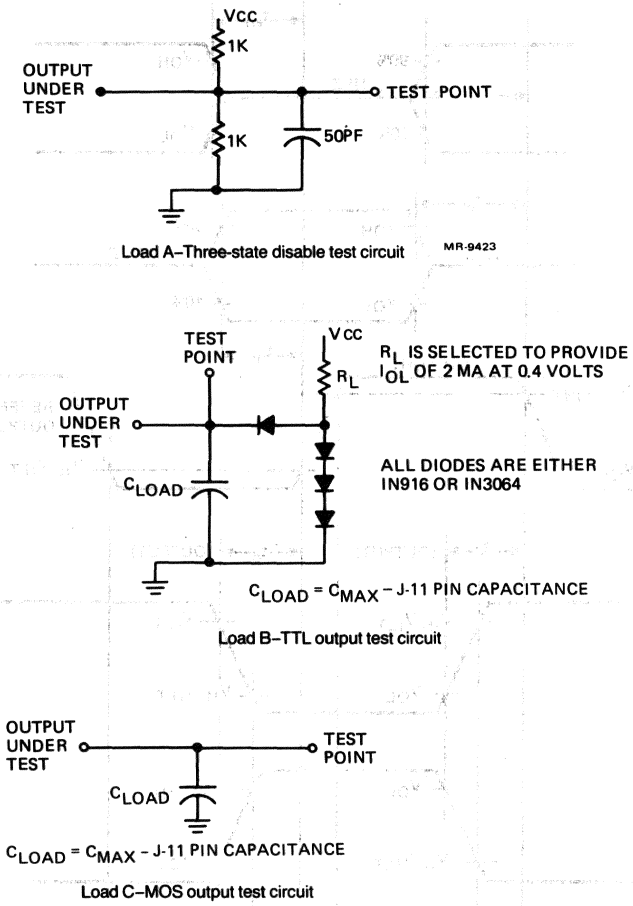


Figure 31 ▪ DCJ11 Output Loading Circuits

Clock Signal Timing

Figure 32 shows the timing references for the clock pulses referenced in the following measurements. The reference edges are defined as whole- and half-unit clock cycles. A whole unit is the time between the rising edges of the clock cycles and a half unit clock pulse is defined as the time between the rising and falling edge of a clock cycle. Figure 33 shows the timing references for the clock outputs and Table 33 lists the clock timing parameters.

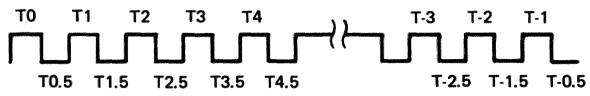


Figure 32 ▪ DCJ11 Clock Cycle Reference Edges

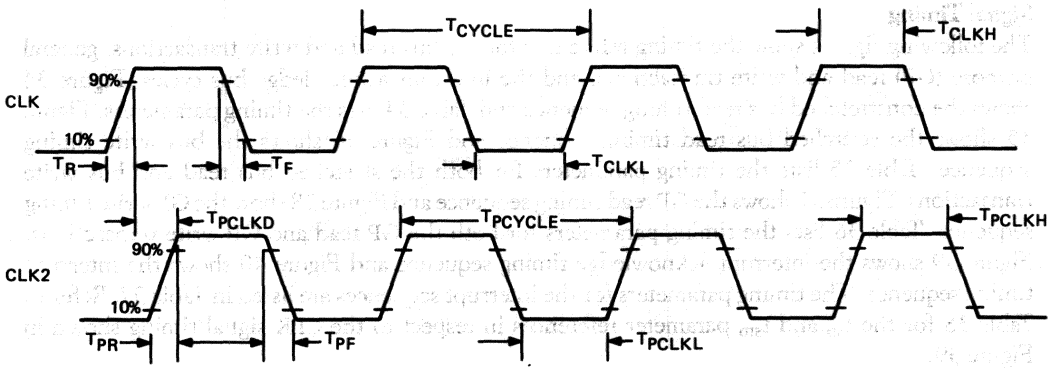


Figure 33 • DCJ11 Clock Output Timing Waveforms

Table 33 • DCJ11 Clock Output Timing Parameters

Symbol	Parameter	Requirements Min.	Max.	Units	Reference	Load Circuit[1]
t_{INITW}	$\overline{INIT}$ pulse width	10	—	clock periods	N/A	
t_{SHTLLH}	Initialization interval	225	—	ns	N/A	
t_{CYCLE}	CLK cycle time	67	—	ns	N/A	Load C
t_{CLKH}	CLK high width	28	—	ns	N/A	Load C
t_{CLKL}	CLK low width	28	—	ns	N/A	Load C
t_R	CLK rise time	—	7	ns	N/A	Load C
t_F	CLK fall time	—	7	ns	N/A	Load C
t_{PCYC}	CLK2 cycle time	67	—	ns	N/A	Load B
t_{PCLKD}[2]	CLK to CLK2 high time	—	—			
t_{PCLKH}	CLK2 high width	28	—	ns	N/A	Load B
t_{PCLKL}	CLK2 low width	28	—	ns	N/A	Load B
t_{PR}	CLK2 rise time	—	7	ns	N/A	Load B
t_{PF}	CLK2 fall time	—	7	ns	N/A	Load B

[1]Refer to Figure 31 for output load circuits used for the timing measurements.
[2]Dependent on output loading of CLK and CLK2.

Signal Timing

The following figures show the timing references for the bus read and write transactions, general purpose (GP) read and write transactions, and the interrupt acknowledge bus cycles. Figure 34 shows the nonstretched bus read timing sequence and Table 34 lists the timing parameters. Figure 35 shows the stretched bus read timing sequence and Figure 36 shows the bus write timing sequence. Table 35 lists the timing parameters for both the stretched bus read and bus write transactions. Figure 37 shows the GP read timing sequence and Figure 38 show the GP write timing sequence. Table 36 lists the timing parameters for both the GP read and GP write transactions. Figure 39 shows the interrupt acknowledge timing sequence and Figure 40 shows the interrupt timing sequence. The timing parameters for the interrupt sequences are listed in Table 37. Refer to Table 38 for the t_{SD} and t_{SID} parameter references in respect to the CLK signal timing shown in Figure 39.

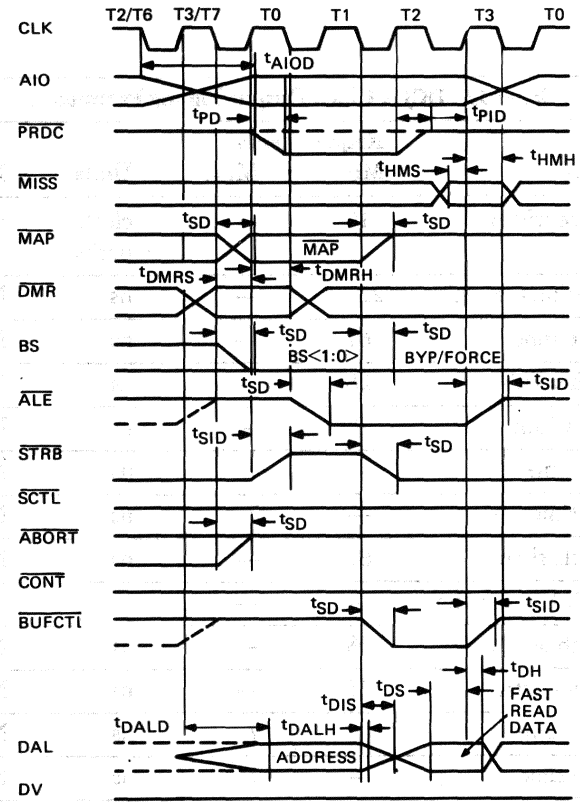


Figure 34 • DCJ11 Nonstretched Bus Read Timing Sequence

Table 34 • DCJ11 Nonstretched Bus Read Timing Parameters

Symbol	Parameter	Requirements Min.	Max.	Units	Reference	Load Circuit[1]
t_{AIOD}	AIO<3:0> delay	—	75	ns	T -1.5	Load B
t_{DALD}	DAL valid delay	—	65	ns	T -1, T1.5	Load B
t_{DALH}	DAL valid hold	5	—	ns	T 1.5, T3	Load B
t_{DIS}	DAL output disable	—	35	ns	T 1.5	Load A
t_{DMRS}	$\overline{DMR}$ setup[2]	30	—	ns	T0	
t_{DMRH}	$\overline{DMR}$ hold[2]	20	—	ns	T0	
t_{DS}	DAL<15:00> setup	35	—	ns	T3	
t_{DH}	DAL<15:00> hold	5	—	ns	T3	
t_{HMS}	MISS setup	30	—	ns	T3	
t_{HMH}	MISS hold	10	—	ns	T3	
t_{PD}	PRDC valid delay	—	50	ns	T0	Load B
t_{PID}	PRDC inactive delay	—	50	ns	T2	Load B
t_{SD}	Strobe active delay	0	35	ns	Table 38	Load B
t_{SID}	Strobe inactive delay	0	35	ns	Table 38	Load B

[1]Refer to Figure 31 for output load circuits used for the timing measurements.
[2]The setup and hold signal requirements ensure the recognition of the next sample point.

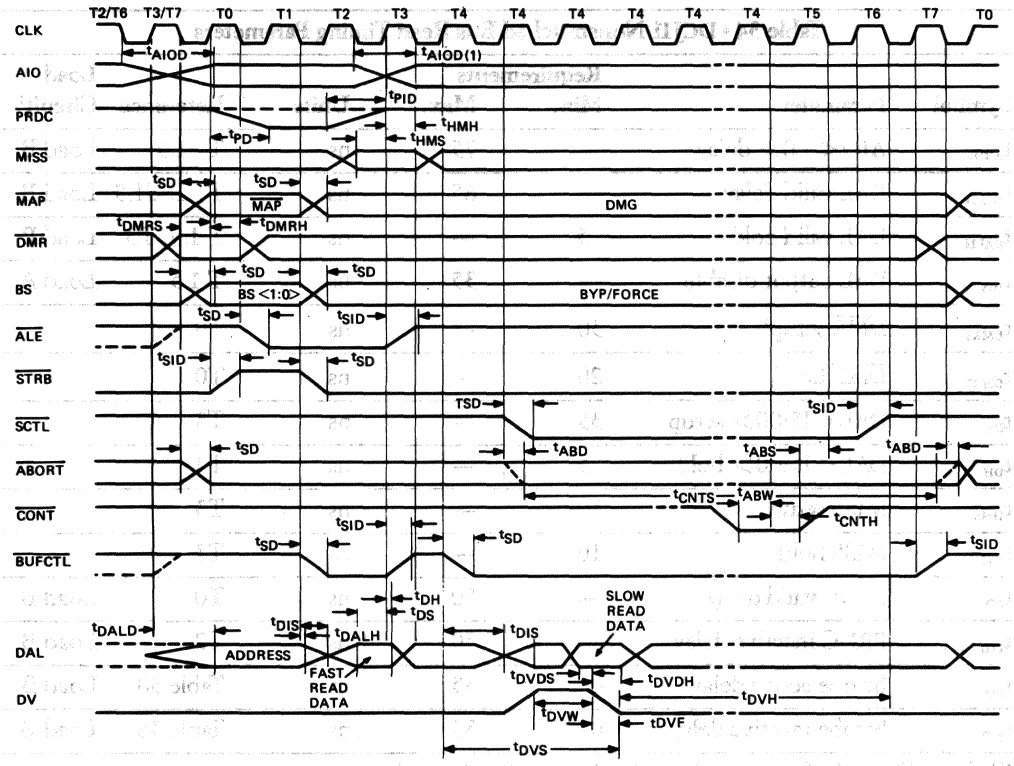


Figure 35 • DCJ11 Stretched Bus Read Timing Sequence

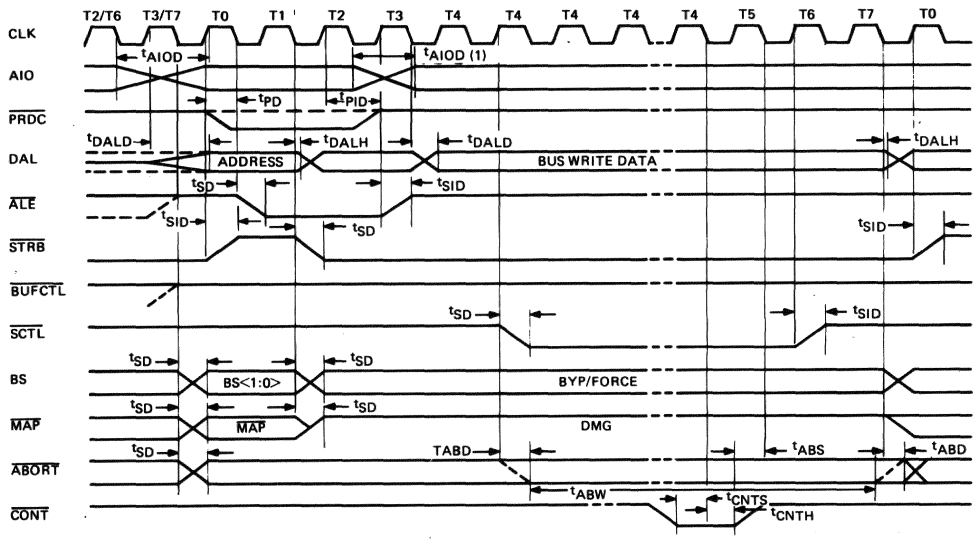


Figure 36 • DCJ11 Bus Write Timing Sequence

Table 35 • DCJ11 Stretched Bus Read and Write Timing Parameters

Symbol	Parameter	Requirements Min.	Max.	Units	Reference	Load Circuit[1]
t_{AIOD}	AIO<3:0> delay	—	75	ns	T–1.5	Load B
t_{CNTS}	$\overline{CONT}$ setup[2]	30	—	ns	T–3.5	
t_{CNTH}	$\overline{CONT}$ hold[2]	20	—	ns	T–3.5	
t_{DALD}	DAL valid delay	—	65	ns	T–1, T1.5	Load B
t_{DALH}	DAL valid hold	5	—	ns	T 1.5, T3	Load B
t_{DIS}	DAL output disable	—	35	ns	T 1.5, T4	Load A
t_{DMRS}	$\overline{DMR}$ setup[2]	30	—	ns	T0	
t_{DMRH}	$\overline{DMR}$ hold[2]	20	—	ns	T0	
t_{DVDH}	DAL<15:00> hold	35	—	ns	MDV-L	
t_{DVDS}	DAL<15:00> setup	35	—	ns	MDV-L	
t_{DVF}	DV fall time	—	15	ns	N/A	
t_{DVH}	DV deassertion	—	0	ns	T6.5	
t_{DVS}	DV deassertion	0		ns	T4	
t_{DVW}	DV pulse width	35	—	ns	N/A	
t_{PD}	$\overline{PRDC}$ valid delay	—	50	ns	T0	Load B
t_{PID}	$\overline{PRDC}$ inactive delay	—	50	ns	T2	Load B
t_{SD}	Strobe active delay	0	35	ns	Table 38	Load B
t_{SID}	Strobe inactive delay	0	35	ns	Table 38	Load B

[1]Refer to Figure 31 for output load circuits used for the timing measurements.
[2]The setup and hold signal requirements ensure the recognition of the next sample point.

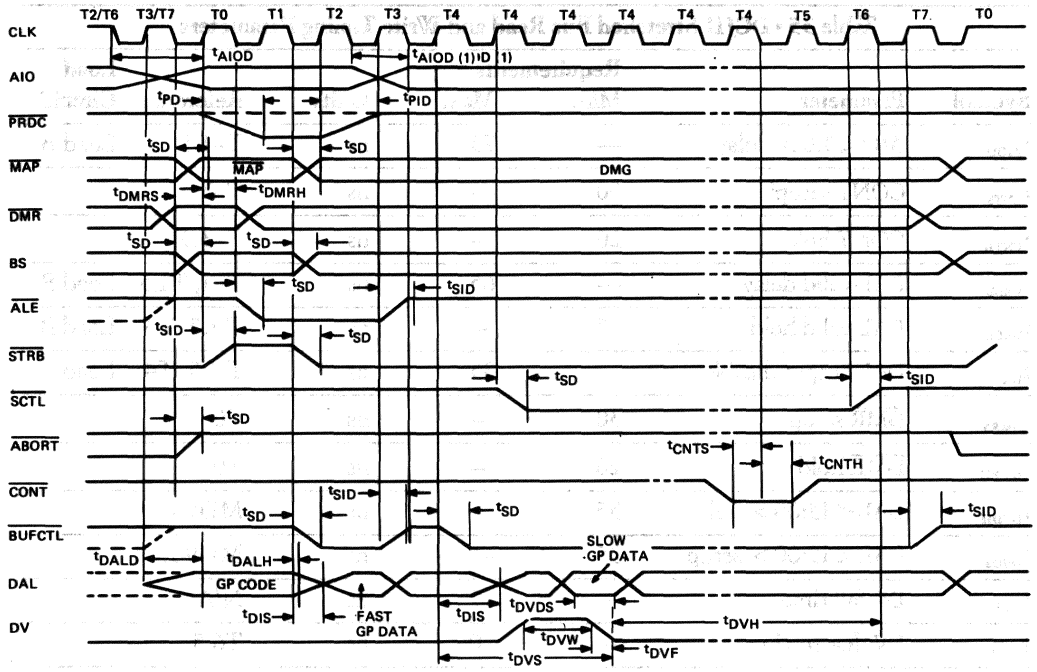


Figure 37 • DCJ11 GP Read Timing Sequence

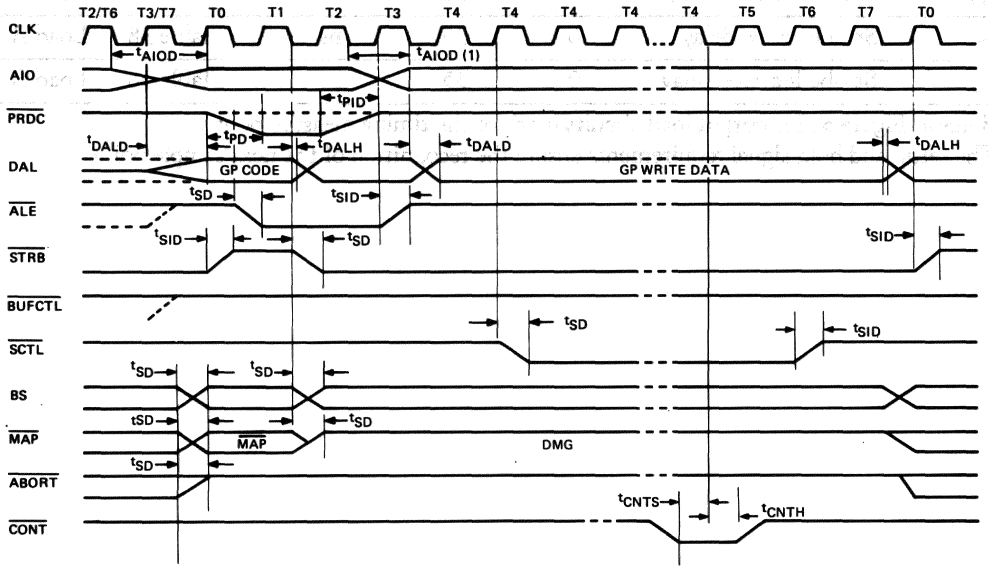


Figure 38 • DCJ11 GP Write Timing Sequence

Table 36 • CJ11 GP Read and Write Timing Parameters

Symbol	Parameter	Requirements Min.	Requirements Max.	Units	Reference	Load Circuit[1]
t_{ABD}	$\overline{\text{ABORT}}$ delay	0	—	ns		
t_{ABS}	$\overline{\text{ABORT}}$ drive	30	—	ns	T–2.5	
t_{ABW}	$\overline{\text{ABORT}}$ width	40+	t_{CLKH}	ns		
t_{AIOD}	AIO<3:0> delay	—	75	ns	T–1.5	Load B
t_{CNTS}	$\overline{\text{CONT}}$ setup[2]	30	—	ns	T–3.5	
t_{CNTH}	$\overline{\text{CONT}}$ hold	20	—	ns	T–3.5	
t_{DALD}	DAL valid delay	—	65	ns	T–1, T1.5	Load B
t_{DALH}	DAL valid hold	5	—	ns	T1.5, T3	Load B
t_{DH}	DAL<15:00> hold	5	—	ns	T3	
t_{DIS}	DAL output disable	—	35	ns	T1.5, T4	Load A
t_{DMRS}	$\overline{\text{DMR}}$ setup[2]	30	—	ns	T0	
t_{DMRH}	$\overline{\text{DMR}}$ hold[2]	20	—	ns	T0	
t_{DS}	DAL<15:00> setup	35	—	ns	T3	
t_{DVDH}	DAL<15:00> hold	35	—	ns	MDV-L	
t_{DVDS}	DAL<15:00> setup	35	—	ns	MDV-L	
t_{DVF}	DV fall time	—	15	ns	N/A	
t_{DVH}	DV deassertion	—	0	ns	T6.5	
t_{DVS}	DV deassertion	0		ns	T4	
t_{DVW}	DV pulse width	35	—	ns	N/A	
t_{HMS}	$\overline{\text{MISS}}$ setup	30	—	ns	T3	
t_{HMH}	$\overline{\text{MISS}}$ hold	10	—	ns	T3	
t_{PD}	$\overline{\text{PRDC}}$ valid delay	—	50	ns	T0	Load B
t_{PID}	$\overline{\text{PRDC}}$ inactive delay	—	50	ns	T2	Load B
t_{SD}	Strobe active delay	0	35	ns	Table 38	Load B
t_{SID}	Strobe inactive delay	0	35	ns	Table 38	Load B

[1]Refer to Figure 31 for output load circuits used for the timing measurements.
[2]The setup and hold signal requirements ensure the recognition of the next sample point.

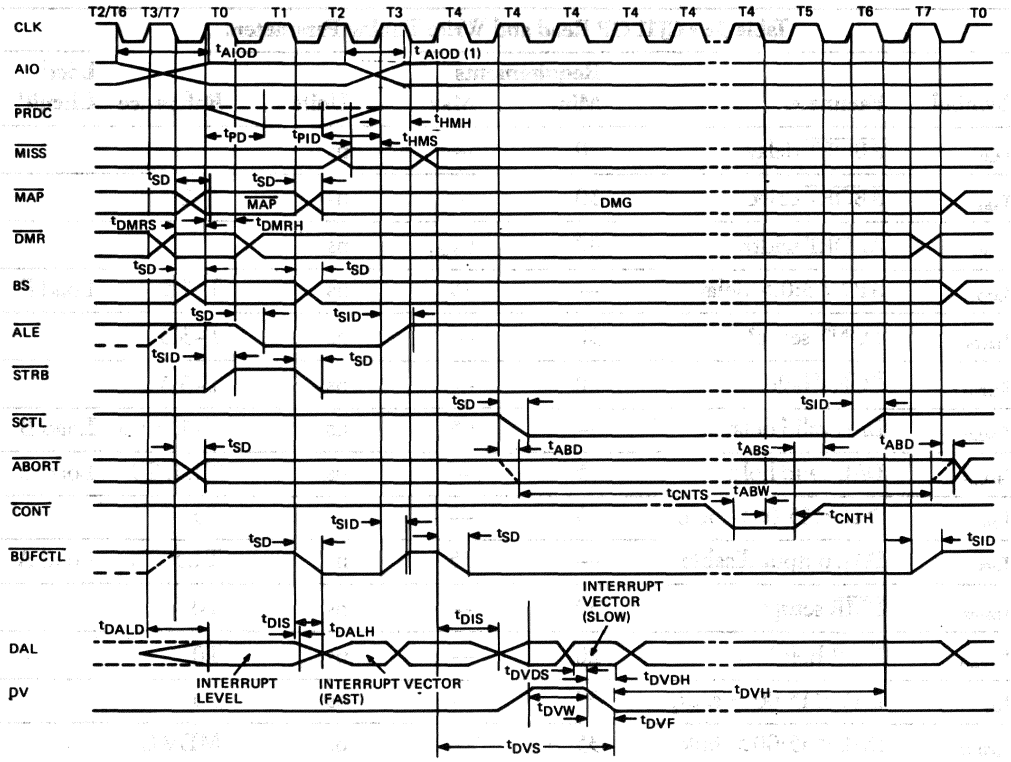


Figure 39 ▪ DCJ11 Interrupt Acknowledge Timing Sequence

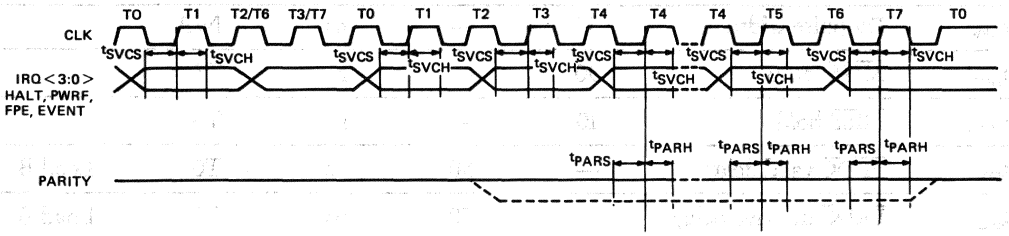


Figure 40 ▪ DCJ11 Interrupt Timing Sequence

Table 37 • DCJ11 Interrupt and Acknowledge Timing Parameters

Symbol	Parameter	Requirements Min.	Max.	Units	Reference	Load Circuit[1]
t_{ABD}	$\overline{\text{ABORT}}$ delay	0	—	ns		
t_{ABS}	$\overline{\text{ABORT}}$ drive	30	—	ns	T–2.5	
t_{ABW}	$\overline{\text{ABORT}}$ width	40 + t_{CLKH}	—	ns		
t_{AIOD}	AIO<3:0> delay	—	75	ns	T–1.5	Load B
t_{CNTS}	$\overline{\text{CONT}}$ setup[2]	30	—	ns	T–3.5	
t_{CNTH}	$\overline{\text{CONT}}$ hold	20	—	ns	T–3.5	
t_{DALD}	DAL valid delay	—	65	ns	T–1	Load B
t_{DALH}	DAL valid hold	5	—	ns	T1.5, T3	Load B
t_{DIS}	DAL output disable	—	35	ns	T1.5 T4	Load A
t_{DMRS}	$\overline{\text{DMR}}$ setup[2]	30	—	ns	T0	
t_{DMRH}	$\overline{\text{DMR}}$ hold[2]	20	—	ns	T0	
t_{DVDH}	DAL<15:00> hold	35	—	ns	MDV-L	
t_{DVDS}	DAL<15:00> setup	35	—	ns	MDV-L	
t_{DVF}	DV fall time	—	15	ns	N/A	
t_{DVH}	DV deassertion	—	0	ns	T6.5	
t_{DVS}	DV deassertion	0		ns	T4	
t_{DVW}	DV pulse width	35	—	ns	N/A	
t_{HMS}	$\overline{\text{MISS}}$ setup	30	—	ns	T3	
t_{HMH}	$\overline{\text{MISS}}$ hold	10	—	ns	T3	
t_{PARS}	$\overline{\text{PARITY}}$ setup	20	—	ns	Figure 39	
t_{PARH}	$\overline{\text{PARITY}}$ hold[2]	20	—	ns	Figure 39	
t_{PD}	$\overline{\text{PRDC}}$ valid delay	—	50	ns	T0	Load B
t_{PID}	$\overline{\text{PRDC}}$ inactive delay	—	50	ns	T2	Load B
t_{SD}	Strobe active delay	0	35	ns	Table 38	Load B
t_{SID}	Strobe inactive delay	0	35	ns	Table 38	Load B

Symbol	Parameter	Requirements Min.	Max.	Units	Reference	Load Circuit[1]
t_{SVCS}	IRQ<3:0>, $\overline{HALT}$, $\overline{PWRF}$, $\overline{FPE}$, $\overline{EVENT}$ setup[2]	20	—	ns	Figure 41	
t_{SVCH}	IRQ<3:0>, $\overline{HALT}$, $\overline{PWRF}$, $\overline{FPE}$, $\overline{EVENT}$ hold[2]	20	—	ns	Figure 41	

[1]Refer to Figure 31 for output load circuits used for the timing measurements.
[2]The setup and hold signal requirements ensure the recognition of the next sample point.

Table 38 • DCJ11 t_{SD} and t_{SID} Parameter References

Signal	t_{SD} Reference Edge	t_{SID} Reference Edge
$\overline{ALE}$	T0.5	T3
$\overline{STRB}$	T1.5	T0
$\overline{BUFCTL}$	T1.5, first T4	T3, T–1
$\overline{SCTL}$	Second T4 or T5	T–2
BS	T–0.5, T1	
$\overline{MAP}$	T1.5	
$\overline{ABORT}$	T–0.5	

Features

- Accelerates by five to eight times the DCJ11 floating-point instruction performance.
- Improves by three to five times the system performance in floating-point applications.
- Supports the complete FP11 floating-point instruction set.
- Supports single- and double-precision floating-point, as well as 16- and 32-bit integers.
- High-speed, double-metal ZMOS technology.
- Single 5-Vdc power supply.

Description

The FPJ11, shown in Figure 1, is a very large scale integration (VLSI) floating-point coprocessor for the DCJ11 microprocessor that implements the FP11 floating-point instruction set on a single 40-pin chip. The high performance of the FPJ11 significantly improves the performance of computation-intensive applications.

The FPJ11 interface provides the ability to overlap instruction execution in a DCJ11 system. This ability allows the effective execution time of floating-point instructions to be measured as the time required to execute the support microcode in the DCJ11 and any time waiting for a previous floating-point instruction to complete.

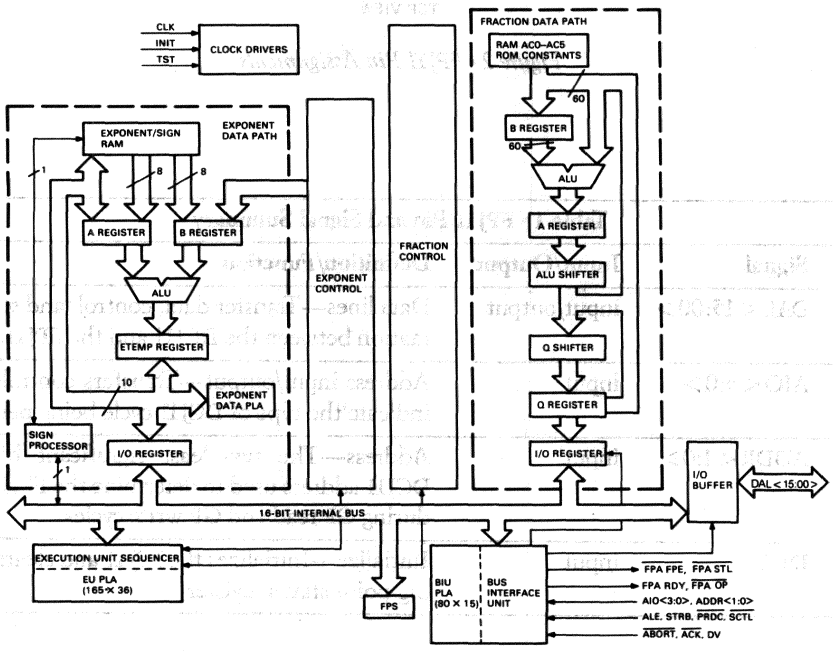


Figure 1 • FPJ11 FPA Block Diagram

• Pin and Signal Definitions

This section provides a description of the input and output signals and power and ground connections of the FPJ11 package. The pin assignments are identified in Figure 2 and the signals are summarized in Table 1.

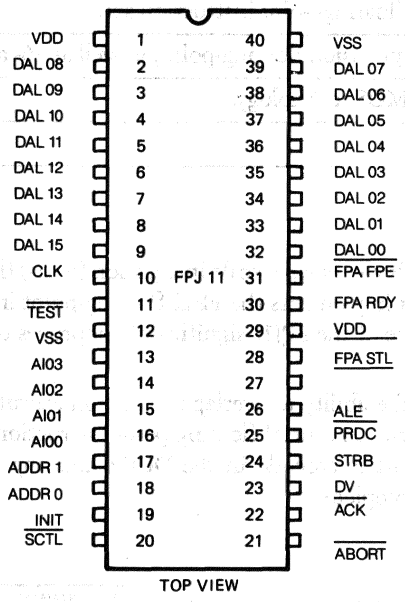


Figure 2 • FPJ11 Pin Assignments

Table 1 • FPJ11 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
2-9 32-39	DAL<15:00>	input/output	Data lines—Transfer data, control, and status information between the DCJ11 and the FPJ11.
13-16	AIO<3:0>	input	Address input/output—Transfers control signals to indicate the type of DCJ11 cycle being performed.
17,18	ADDR<1:0>	input	Address—The two least significant bits of the DCJ11 address used to determine the FPJ11 function during GP read and GP write cycles.
19	INIT	input	Initialize—Initializes the FPJ11 and clears the floating-point status register.

Pin	Signal	Input/Output	Definition/Function
20	$\overline{\text{SCTL}}$	input	Stretch control—Used to enable the sampling of the abort condition and clear the initialization condition.
21	$\overline{\text{ABORT}}$	input	Abort—Indicates a noncompletion of the current cycle to the FPJ11.
22	$\overline{\text{ACK}}$	input	Acknowledge—Enables the transfer of the FPJ11 output data onto DAL<15:00>.
23	DV	input	Data valid—An asynchronous strobe from the system interface used to latch input data during stretched reads and general purpose write cycles.
24	STRB	input	Strobe—A timing signal from the DCJ11 used to latch the $\overline{\text{PRDC}}$ input and to indicate the end of cycle.
25	$\overline{\text{PRDC}}$	input	Predecode—Indicates that an instruction is being decoded.
26	ALE	input	Address latch enable—A timing signal used to latch the AIO<3:0> and ADDR<1:0> inputs at low-to-high transition and to read cache data at high-to-low transition.
27	$\overline{\text{FPA OP}}$	output	Floating-point accelerator operation—Asserted to inform the system interface of write cycles that use FPJ11 data.
28	$\overline{\text{FPA STL}}$	output	Floating-point accelerator stall—Used to stall the operation of the DCJ11 through the DMR input.
30	FPA RDY	output	Floating-point accelerator ready—Indicates the FPJ11 output data is ready for transfer.
31	$\overline{\text{FPA FPE}}$	output	Floating-point accelerator floating-point exception—Asserted to inform the DCJ11 of a floating-point exception condition.
10	CLK	input	Clock—The timing signal for FPJ11 operation.
11	$\overline{\text{TEST}}$	input	Test—Used during manufacturing test only.
1,29	V_{DD}	input	Voltage—Power supply voltage.
12,40	V_{SS}	input	Ground—Ground reference.

Data Lines

Data lines (DAL<15:00>)—These lines are bidirectional I/O lines used for data communication with the DCJ11.

System Control

Clock (CLK)—Basic clock input to the FPJ11.

Address input/output (AIO<3:0>)—The AIO<3:0> lines indicate to the FPJ11 the type of I/O cycle as described in Table 2.

Table 2 • FPJ11 Address Input/Output Code Assignments

AIO line 3	2	1	0	Cycle type
1	1	1	1	Non-I/O microcycle (Non IO)
1	1	1	0	General purpose read (GP Read)
1	1	0	1	Not used
1	1	0	0	Instruction stream read request (I Read request)
1	0	1	X	Read-modify-Write (RMW)
1	0	0	1	Data stream read (D Read)
1	0	0	0	Instruction stream read demand (I Read demand)
0	1	1	X	Not used
0	1	0	X	General purpose word write (GP Write)
0	0	1	X	Not used
0	0	0	X	External word write (Write)

X=either 1 or 0

Address lines (ADDR<1:0>)—The ADDR<1:0> lines contain the two least significant bits of the DCJ11 address. They are used by the FPJ11 to decode the type of general purpose (GP) read and write cycles as described in Table 3.

Table 3 • FPJ11 GP Read and Write Address Code Assignments

ADDR Line 1	0	Cycle Function
GP Read cycle		
0	0	Read powerup options
0	1	Read data from the FPJ11
1	0	Read powerup options and clear floating-point status register
1	1	Read floating-point exception code and clear floating-point exception signal
GP Write Cycle		
1	1	Load data into the FPJ11

Address latch enable (ALE)—The low-to-high transition of this signal is used to latch the information from the AIO<3:0> and ADDR<1:0> lines. The high-to-low transition is used to the latch cache-hit input data to the FPJ11. The polarity of this signal is inverted from the DCJ11 output.

Strobe (STRB)—The high-to-low transition of this signal indicates the end of a DCJ11 cycle. During FPJ11 read cycles, the STRB signal indicates that data was loaded by either the ALE or DV signal. The low-to-high transition is used to sample the $\overline{\text{PRDC}}$ signal. The polarity of this signal is inverted from the DCJ11 output.

Predecode ($\overline{\text{PRDC}}$)—This signal indicates to the FPJ11 that the DCJ11 is initiating instruction decode.

Stretch control ($\overline{\text{SCTL}}$)—This signal enables the sampling of the $\overline{\text{ABORT}}$ line by the FPJ11 during a DCJ11 stretched I/O cycle. The low-to-high transition of $\overline{\text{SCTL}}$ is also used after the negation of the INIT signal to clear the initialization condition.

Abort ($\overline{\text{ABORT}}$)—This signal,if asserted with the $\overline{\text{SCTL}}$ signal, indicates to the FPJ11 that the current I/O cycle will not be completed.

Data valid (DV)—This signal is an asynchronous strobe from the system interface. The high-to-low transition of the DV signal is used to indicate that valid data is on the DAL<15:00> lines. The DV signal is used by the FPJ11 to latch input data during GP write or stretched FPJ11 read cycles.

Acknowledge ($\overline{\text{ACK}}$)—This signal is used to enable the operation of the FPJ11 output drivers. The low to high transition of ACK indicates that output data has been latched. The FPA RDY signal is then deasserted by the FPJ11.

Initialize (INIT)—This signal initializes the FPJ11 and clears the FPS register. The polarity of this signal is inverted from the DCJ11 input.

Output Signals

FPA operate ($\overline{\text{FPA OP}}$)—This signal is asserted by the FPJ11 to inform the system interface that data for the next write cycle will be provided by the FPJ11. The $\overline{\text{FPA OP}}$ signal is valid by the assertion of the ALE signal for DCJ11 bus write cycles. It is also asserted during GP read cycles that read the system powerup options to indicate the presence of an FPJ11 in the system.

FPA stall ($\overline{\text{FPA STL}}$)—This signal is asserted by the FPJ11 to stall the DCJ11. It should be OR gated into the DCJ11 DMR input. The system interface must assert the $\overline{\text{CONT}}$ signal to the DCJ11 after the negation of $\overline{\text{FPA STL}}$ to restart the DCJ11.

FPA floating-point exception ($\overline{\text{FPA FPE}}$)—This signal is asserted by the FPJ11 to indicate that the last completed floating-point instruction had caused an exception. The system interface must assert the $\overline{\text{CONT}}$ signal to the DCJ11 without performing the bus write cycle. The $\overline{\text{FPA FPE}}$ signal is cleared by a GP read operation of the floating-point exception code cycle. This cycle is described in the architecture section.

FPA ready (FPA RDY)—This signal is asserted by the FPA to indicate that output data is ready. The $\overline{\text{ACK}}$ signal must be asserted from the system interface during GP read cycles before the FPJ11 will assert the FPA RDY signal. The FPA RDY signal may be asserted prior to $\overline{\text{ACK}}$ for write cycles. The low-to-high transition of $\overline{\text{ACK}}$ negates the FPA RDY signal. It will not be reasserted until after completion of the current write or GP read cycle.

Miscellaneous

Test ($\overline{TEST}$)—This signal is reserved for use by the manufacturer. It is pulled up internally to the inactive state.

Power (V_{DD})—The 5-Vdc power supply.

Ground (V_{SS})—Ground reference.

• Architecture Summary

The FPJ11 architectural configuration, shown in Figure 3, contains six user-addressable 64-bit floating-point accumulators (AC0-AC05), a floating-point status (FPS) register and a floating-point exception code (FEC) register. The FP11 architecture also includes a floating-point exception address (FEA) register that is implemented in the DCJ11.

The FPJ11 operates on single-precision (F) and double-precision (D) floating-point, and 16- and 32-bit integer data. Single-precision format uses the 32 most significant bits of the floating-point accumulators and produces 8-decimal-digit precision. Double-precision format produces 17-decimal-digit precision.

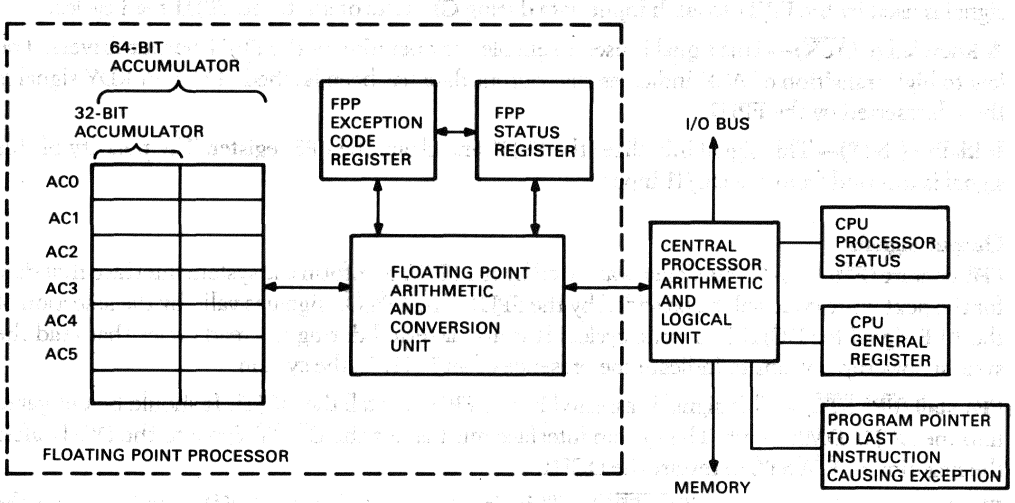


Figure 3 • FPJ11 Architectural Configuration

Operational Units

The FPJ11 consists of two main functional units. The execution unit (EU) consists of the fraction, exponent, and sign processors. The bus interface unit (BIU) controls all interface functions between the EU and DCJ11 system. Both units contain independent control sequencers that interact to allow possible performance improvement through parallel operation of the I/O operations in the BIU and instruction execution in the EU.

The BIU receives all instruction stream data and decodes instructions in parallel with the DCJ11. Support microcode in the DCJ11 initiates all I/O cycles required by the FPJ11. On completion of the support microcode, the DCJ11 proceeds to the next instruction. Subsequent integer instructions can proceed without FPJ11 intervention. For subsequent load class floating-point instructions, the

BIU can support the overlap of operand data loading while the EU completes execution of the previous instruction. For subsequent store class floating-point instruction, the DCJ11 proceeds to the bus write cycle and then waits for the FPJ11 to provide write data or to signal an exception by the $\overline{\text{FPA FPE}}$ input.

Floating-point Data Formats

A floating-point number may be defined as having the form $(+ \text{ or } -)(2^K) \times f$, where K is an integer and f is a fraction. For a nonzero number, K and f are determined by imposing the condition ½ ± < f 1. The fractional part (f) of the number is then normalized. For the number 0, f is assigned the value 0 and the value of K is indeterminate.

The FPJ11 floating-point data formats are derived from this representation for floating-point numbers. Two types of floating-point data are provided. In single precision, or floating mode, the data is 32 bits long. In double precision, or double mode, the data is 64 bits long. Sign magnitude notation is used.

Nonzero Floating-point Numbers

The fractional part (f) is assumed to be binary normalized, so that its most significant bit must be 1. This 1 is the hidden bit. It is not stored explicitly in the data word, but is restored by the FPJ11 before carrying out arithmetic operations. The floating and double modes reserve 23 and 55 bits, respectively, for f. These bits, with the hidden bit, imply effective fractions of 24 bits and 56 bits.

Eight bits are reserved for storage of the exponent K in excess-128 (200 octal) notation [i.e., K+200], giving a biased exponent. Thus, exponents from −128 to +127 are represented by 0 to 377 (octal), or 0 to 255 (decimal). A biased exponent of 0 (the true exponent of −200 octal), is reserved for floating-point 0. Therefore, exponents are restricted to the range of −127 to +127 inclusive (−177 to +177 octal) or, in excess-200 notation, 1 to 377 octal.

The remaining bit of the floating-point word is the sign bit. The number is negative if the sign bit is a 1.

Floating-point Zero

Because of the hidden bit, the fractional part is not sufficient to distinguish between 0 and nonzero numbers whose fractional part is exactly one-half. Therefore, the FPJ11 reserves a biased exponent of 0 for this purpose and any floating-point number with a biased exponent of 0 either traps or is treated as if it were an exact 0 in arithmetic operations. An exact or clean 0 is represented by a word whose bits are all zeros. A dirty 0 is a floating-point number with a biased exponent of 0 and a nonzero fractional part. An arithmetic operation for which the resulting true exponent exceeds 177 (octal) is regarded as producing a floating overflow; if the true exponent is less than −177 (octal), the operation is regarded as producing a floating underflow. A biased exponent of 0 can occur from arithmetic operations as a special case of overflow (true exponent = 200 octal).

Undefined Variable

The undefined variable is any bit pattern with a sign bit of 1 and a biased exponent of 0. The term "undefined variable" is used to indicate that these bit patterns are not assigned a corresponding floating-point arithmetic value. The undefined variable is also referred to as −0. The FPJ11 ensures that the undefined variable will not be stored as the result of any floating-point arithmetic instruction in a program that is run with the overflow and underflow interrupts disabled. This is achieved by storing an exact 0 on overflow and underflow if the corresponding interrupt is disabled.

Floating-point Data

The single- and double-precision floating-point data is stored in memory as shown in Figure 4.

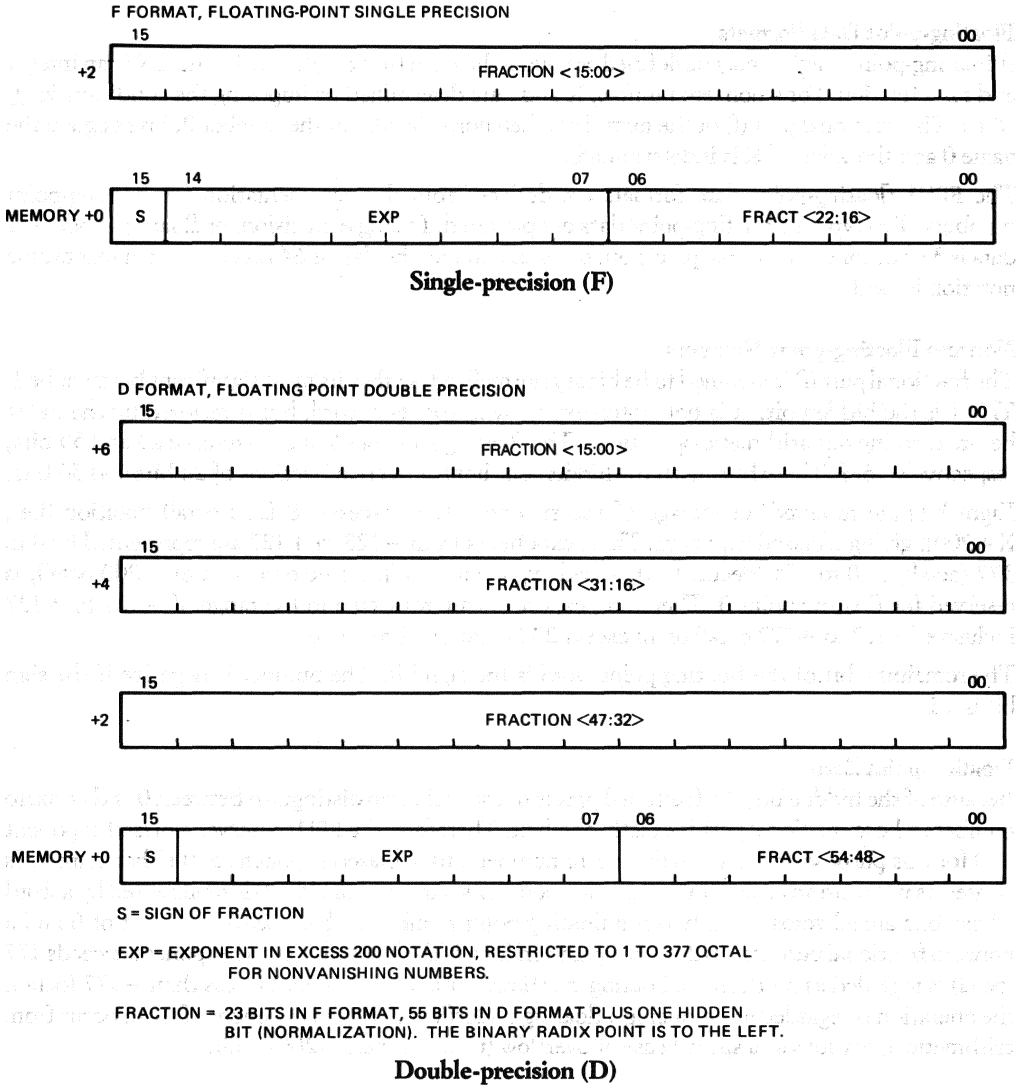


Figure 4 • FPJ11 Floating-point Data Formats

The FPJ11 provides for conversion of floating-point to integer format and integer format to floating-point format. The processor recognizes the 16-bit short integer (I), and the 32-bit long integer (L) shown in Figure 5. The numbers are in two's complement format.

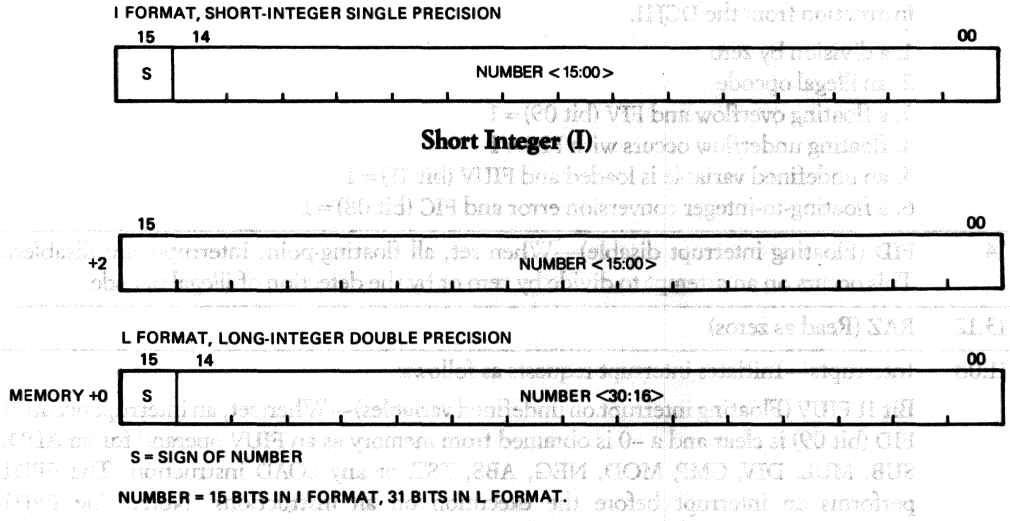


Figure 5 • FPJ11 Integer Data Format

Floating-point Status Register

The floating-point status register (FPS), shown in Figure 6, contains three mode control bits, five interrupt control bits, an error bit, and four condition codes. The FPS register bits are described in Table 4. This register is cleared during the powerup sequence or after a GP read cycle.

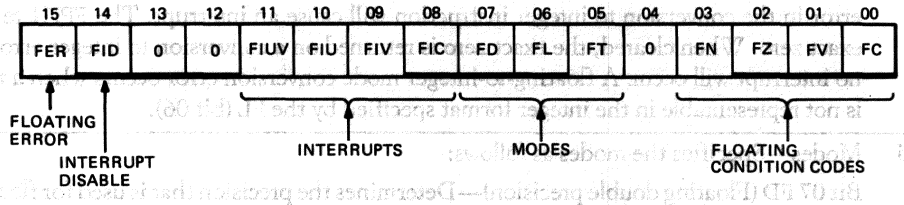


Figure 6 • FPJ11 Floating-point Status Register Format

Table 4 • FPJ11 Floating-point Status Register Description

Bit	Function
15	FER (Floating error)—This bit is set if one of the following conditions occurs. The setting of this bit is independent of the state of the FID (bit 14). Cleared only by the LDFPS instruction from the DCJ11. 1. a division by zero 2. an illegal opcode 3. a floating overflow and FIV (bit 09) = 1 4. floating underflow occurs with FIU = 1 5. an undefined variable is loaded and FIUV (bit 11) = 1 6. a floating-to-integer conversion error and FIC (bit 08) = 1
14	FID (Floating interrupt disable)—When set, all floating-point interrupts are disabled. This occurs on an attempt to divide by zero or by the detection of illegal opcode.
13,12	RAZ (Read as zeros)
11:08	Interrupts—Initiates interrupt requests as follows: Bit 11 FIUV (Floating interrupt on undefined variables)—When set, an interrupt occurs if FID (bit 09) is clear and a –0 is obtained from memory as an FIUV operand for an ADD, SUB, MUL, DIV, CMP, MOD, NEG, ABS, TST, or any LOAD instruction. The FPJ11 performs an interrupt before the execution on all instructions. Note: The FPJ11 instruction set is interrupted after the execution of NEG, ABS, and TST instructions. Bit 10 FIU (Floating interrupt on underflow)—When set and the FID (bit 14) is clear, a floating underflow will cause an interrupt. The fractional part of the result of the operation causing the interrupt will be correct. The biased exponent will be too large by a value of 400 (octal), except for the special case of 0 in which it is correct. If cleared and an underflow occurs, no interrupt occurs and the FPJ11 returns exact 0. Bit 09 FIV (Floating interrupt on overflow)—When set and FID (bit 14) is cleared, a floating overflow will cause an interrupt. The fractional part of the result of the operation causing the overflow will be correct. The biased exponent will be smaller by a value of 400 (octal). Bit 08 FIC (Floating interrupt on integer conversion)—When set and FID is cleared, an error in the conversion to integer instruction will cause an interrupt. The FPJ11 returns exact zero. When cleared, the exact zero is returned on a conversion to integer error but no interrupt will occur. A floating-to-integer mode conversion error occurs when a result is not representable in the integer format specified by the FL (bit 06).
07:05	Modes—Specifies the modes as follows: Bit 07 FD (Floating double precision)—Determines the precision that is used for floating-point calculations. When set, the double-precision mode is used. When cleared, the single-precision mode is used. Bit 06 FL (Floating long integer)—When set, the long integer format is used (32 bits). When cleared, the integer format is used (16 bits). Bit 05 FT (Floating chop)—When set, the result of an arithmetic operation is chopped (truncated). When cleared, the result is rounded.

Bit	Function
04	RAZ (Read as zero)
03:00	Floating condition codes—Specifies the results of an operation as follows:
	Bit 03 FN (Floating negative)—Set if the result of the last floating-point operation was negative.
	Bit 02 FZ (Floating zero)—Set if the result of the last floating-point operation was zero.
	Bit 01 FV (Floating overflow)—Set if the last floating-point operation resulted in an exponent overflow.
	Bit 00 FC (Floating carry)—Set if the last floating-point instruction was a conversion from double/floating to integer/long in which the result was not representable in the integer format specified by FL (bit 06).

Floating-point Exception Code and Address Registers

One interrupt vector is assigned in DCJ11 systems for all floating-point exceptions (location 244). A code for the six possible error conditions is contained in the 4-bit floating exception code (FEC) register shown in Figure 7. Table 5 lists the code and error conditions.

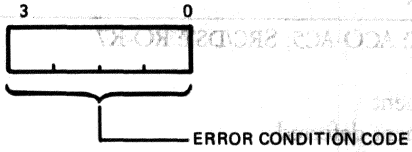


Figure 7 • FPJ11 Floating Exception Code Register Format

Table 5 • FPJ11 Floating Exception Code Register Description

Bit	Description	
3:0	An octal code that defines the error condition as folows.	
	Octal code	**Error condition**
	2	Floating opcode error
	4	Floating divide by zero
	6	Floating-to-integer or double-to-integer conversion error
	8	Floating overflow
	10	Floating underflow
	12	Floating undefined variable

A copy of the FEC register, which is updated during exception servicing, is located in the DCJ11. The address of the instruction producing the exception is stored in the floating exception address (FEA) register. The FEA register is located in the DCJ11 and updated by DCJ11 microcode. Therefore, the store status (STST) instruction does not require an output from the FPJ11 and is executed entirely by the DCJ11.

The FEC and FEA registers are updated under the same conditions that cause the FER bit to be set. No instruction is provided for writing into the FEC and FEA registers. The FPJ11 does not assert the $\overline{\text{FPA FPE}}$ signal until after the next instruction decode. A store status (STST) instruction may not return new values of the FEC and FEA registers for the immediately preceding floating-point instruction.

• Floating-point Instruction Addressing

Floating-point instructions use the same type of addressing as integer instructions. For addressing modes 1 to 7, a source or destination operand is specified by designating one of eight addressing modes and one of eight general registers. For addressing mode 0, floating-point operands are located in the specified floating-point accumulator (AC0-AC5). Integer operands are located in the general register file of the DCJ11 (R0-R7). Table 6 lists the addressing modes.

Table 6 • FPJ11 Addressing Modes

Mode	Description
0	FSRC/FDST: ACO-AC5; SRC/DST: RO-R7
1	Deferred
2	Autoincrement
3	Autoincrement-deferred
4	Autoincrement
5	Autoincrement-deferred
6	Indexed
7	Indexed-deferred

The autoincrement and autodecrement modes operate on increments and decrements of 4 for F format and 10 (octal) for D format. In mode 0, users can make use of all six floating-point accumulators (AC0-AC5) as their source or destination. Specifying floating-point accumulators AC6 or AC7 will result in an illegal opcode trap. In all other modes that involve transfer of data to or from memory or the general registers, users are restricted to the first four floating-point accumulators (AC0-AC3). When reading or writing a floating-point number to or from memory, the low memory word contains the most significant word of the floating-point number, and the high memory word contains the least significant word.

Accuracy

An instruction or operation is regarded as "exact" if the result is identical to an infinite precision calculation involving the same operands. All arithmetic instructions treat an operand, whose biased exponent is 0, as an exact 0 unless FIUV (bit 10) of the FPS is set and the operand is −0. A zero operand implies that the result of an arithmetic instruction is exact except when the operand is the divisor for a DIV instruction. In division, if the divisor is 0, the result is undefined and an interrupt occurs if FID (bit 14) is cleared.

For nonzero floating-point operands, the fractional part is binary normalized. It contains 24-bits for floating mode and 56-bits for double mode. For ADD, SUB, MUL, and DIV instructions, two guard bits are necessary and sufficient for the general case to assure the return of a chopped or rounded result identical to the corresponding infinite precision operation chopped or rounded to the specified word length. Thus, with two guard bits, a chopped result has an error bound of one least significant bit (LSB); a rounded result has an error bound of one-half LSB. These error bounds are realized by the FPJ11 for all instructions.

The occurrence of a floating-point overflow and underflow causes an error condition; the result of the calculation cannot be correctly stored because the exponent is larger than the eight bits reserved for it. When an underflow occurs, replacement of the correct answer by 0 can resolve the problem for many applications. This is performed by the FPJ11 if the underflow interrupt is disabled. The error incurred by this action is an absolute rather than a relative error; it is bounded (in absolute value) by 2^{-128} (decimal). For the case of overflow, if the overflow interrupt is disabled, the FPJ11 returns exact zero.

The FIV (bit 10) and FIU (bit 9) of the FPS register provide users with an opportunity to implement corrections of an overflow or underflow condition. The FPJ11 stores the correct fractional part and the low eight bits of the biased exponent after floating overflow or underflow if the corresponding interrupt is enabled. The cause of the interrupt can be identified by examination of the floating overflow FV (bit 1) of the FPS register or the floating exception register (FEC).

The biased exponent returned by the instruction is related to the correct exponent. On an overflow, it is smaller by 400 (octal) and on an underflow, if the biased exponent is 0, it is correct. If the biased exponent is not 0, it is larger by 400 (octal).

Users may rescale their variables using STEXP and LDEXP instructions to continue a calculation.

Floating-point Instructions

Floating-point instructions operate on either single- or double-precision numbers, depending on the status of the FD mode bit 7 in the FPS register. Similarly, the FL mode bit determines whether 32-bit or 16-bit integers are used in conversions between integer and floating-point representation. The floating source (FSRC) and floating destination (FDST) use floating-point addressing modes, while SRC and DST use CPU addressing modes. The FP instruction formats for single- and double-operand addressing are shown in Figure 8.

The six floating-point accumulators are used in numeric calculations and in interaccumulator data transfers. The first four accumulators (AC0-AC3) are also used for all data transfers between the FPJ11 and the general registers or memory.

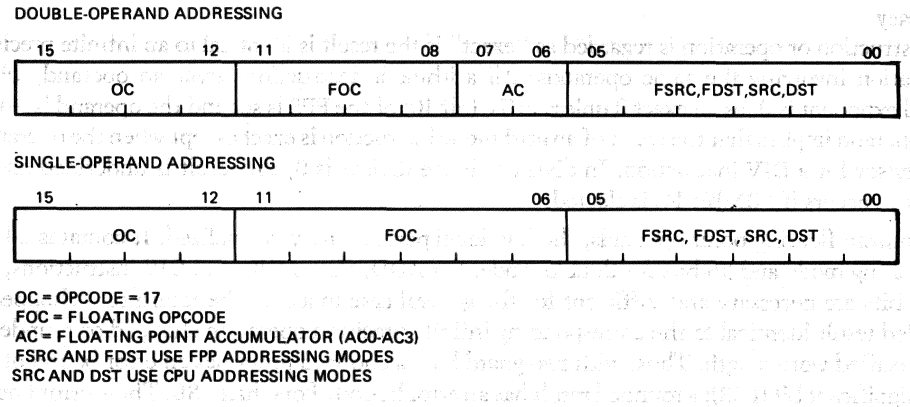


Figure 8 ▪ FPJ11 Single- and Double-operand Addressing Instruction Formats

The instruction set for the FPJ11 is listed in Table 7. A detailed description of each instruction as well as specific comments on accuracy can be found in the *DCJ11 Microprocessor User's Guide* or the *PDP-11 Architecture Handbook*. The condition code formats for Table 7 are: * = Conditionally set/cleared, – = Not affected, 0 = Cleared, 1 = Set.

Table 7 ▪ FPJ11 Instruction Set

Mnemonic	Opcode	Instruction	FN	FZ	FV	FC
ABSD, ABSF	1706 fdst	Make absolute dbl/fl	0	*	0	0
ADDD, ADDF	172(ac)fsrc	Add dbl/fl	*	*	*	0
CFCC	170000	Copy condition codes	—	—	—	—
CLRD, CLRF	1704 fdst	Clear dbl/fl	0 1	0	0	
CMPD, CMPF	173(ac+4)fsrc	Compare dbl/fl (to ac)	*	*	0	0
DIVD, DIVF	174(ac+4)fsrc	Divide dbl/fl	*	*	*	0
LDCDF, LDCFD	177(ac+4)fsrc	Load and convert dbl-fl/fl-dbl	*	*	*	0
LDCID, LDCIF	177(ac)src	Load and convert integer to dbl/fl	*	*	0	0
LDCLD, LDCLF	177(ac)src	Load and convert long integer to dbl/fl	*	*	0	0
LDD, LDF	172(ac+4)fsrc	Load dbl/fl	*	*	0	0
LDEXP	176(ac+4)src	Load exponent	*	*	*	0
LDFPS	1701 src	Load FPJ11 program status	—	—	—	—
MODD, MODF	171(ac+4)fsrc	Multiply and integerize dbl/fl	*	*	*	0
MULD, MULF	171(ac)fsrc	Multiply dbl/fl	*	*	*	0

Mnemonic	Opcode	Instruction	FN	FZ	FV	FC
NEGD, NEGF	1707 fdst	Negate dbl/fl	*	*	0	0
SETD	170011	Set floating double mode	—	—	—	—
SETF	170001	Set floating mode	—	—	—	—
SETI	170002	Set integer mode	—	—	—	—
SETL	170012	Set long integer mode	—	—	—	—
STCDF, STCFD	176(ac)fdst	Store and convert from dbl/fl to fl/dbl	*	*	*	0
STCDI, STCDL	175(ac+4)dst	Store and convert from dbl to int/long int	*	*	0	*
STCFI, STCFL	175(ac+4)dst	Store and convert from fl to int/long int	*	*	0	*
STD, STF	174(ac)fdst	Store dbl/fl	—	—	—	—
STEXP	175(ac)dst	Store exponent	*	*	0	0
STFPS	1702 dst	Store FPS	—	—	—	—
STST	1703 dst	Store FEA and FEC	—	—	—	—
SUBD, SUBF	173(ac)fsrc	Subtract dbl/fl	*	*	*	0
TSTD, TSTF	1705 fdst	Test dbl/fl	*	*	0	0

Performance

The combination of a fast internal cycle and optimized arithmetic algorithms provides excellent performance for the FPJ11. In addition, instruction overlap provides a substantial performance gain in floating-point intensive code. The bus interface unit (BIU) supports the overlap of operand data loading for the next floating-point instruction while the execution unit (EU) completes the processing of the current instruction. The effective execution time of a floating-point instruction in a DCJ11 system equals only the cycles required to execute the support microcode and any time waiting for the FPJ11 to complete a previous floating-point instruction. Therefore, only a portion of the FPJ11 instruction execution contributes to the overall program execution time.

As a coprocessor, floating-point instruction execution can occur simultaneously with integer code. This overlap can be used effectively to reduce the execution time of code that interleaves floating-point and nonfloating-point instructions.

Table 8 lists the cycle counts for the most frequent FPJ11 arithmetic operations. Cycle time can be determined as twice the input clock period.

Table 8 • FPJ11 Instruction Execution Times

Instruction	Minimum Cycles	Typical Cycles	Typical 15 MHz
ADDF/SUBF	7	9	1.2 μs
MULF	15	15	2.0 μs
DIVF	19	26	3.5 μs
ADDD/SUBD	7	9	1.2 μs
MULD	26	26	3.5 μs
DIVD	35	45	6.0 μs

• Interface

The FPJ11 supports a coprocessor interface with the DCJ11. All bus cycles are initiated by the DCJ11. The AIO<3:0> and ADDR<1:0> lines fully identify the type of bus cycle to the FPJ11.

The FPJ11 loads all instruction stream data into a prefetch buffer. The DCJ11 asserts the $\overline{\text{PRDC}}$ line when decoding instructions. $\overline{\text{PRDC}}$ is never asserted unless the prefetch buffer is valid. Floating-point instructions (opcode 15:12 = 17) begin execution in the FPJ11 in parallel with the DCJ11.

For instructions requiring data from memory, the DCJ11 executes the bus cycles necessary to fetch the operands. The DCJ11 then continues to the next instruction after checking for a FPE from a previous FP instruction. For register mode instruction, the DCJ11 continues to the next instruction immediately after the FPE check.

The FPJ11 outputs data only during store type instructions. Output data is supplied by the FPJ11 during write cycles and GP read cycles as required. The FPJ11 may cancel an output cycle by asserting the $\overline{\text{FPA FPE}}$ signal if the previous floating-point instruction caused a floating-point exception.

The FPJ11 asserts the $\overline{\text{FPA STL}}$ line when executing a floating-point instruction that does not output data if the execution unit is still busy with a previous instruction. The $\overline{\text{FPA STL}}$ signal is asserted before the DCJ11 test for the $\overline{\text{FPA FPE}}$ signal.

The timing and system interface requirements for read, write, GP read, GP write, and the FPA stalls are described in the following sections. Refer to timing diagrams Figures 11 through 15.

An STF/D, STFPS, CFCC, or STEXP to memory instruction should be executed as the first floating-point store instruction after the powerup sequence to initialize the FPJ11. This initialization is performed automatically by all Digital software operating systems except MicroPower/Pascal.

Instruction or Data Reads

The FPJ11 inputs read data on all instruction read cycles and data read cycles that are fetching FPA operands. Data is loaded at the high-to-low transition of the ALE signal for cache hits and again at the high-to-low transition of the DV signal for main memory reads. The FPJ11 uses the high-to-low transition of the $\overline{\text{STRB}}$ input to determine the end of the read cycle. It does not require cache hit/miss information. The system interface read sequence is not altered by the presence of the FPJ11 in the system. If the $\overline{\text{ABORT}}$ signal is asserted during a demand read cycle, the FPJ11 will abort the present instruction.

FPJ11 Write Transactions

The FPJ11 asserts the $\overline{\text{FPA OP}}$ line prior to the low-to-high transition of the ALE signal for all DCJ11 bus write cycles requiring data from the FPJ11. This informs the system interface that the write data is to be supplied by the FPJ11. The system interface can assert the $\overline{\text{ACK}}$ signal immediately upon recognizing a FPJ11 write cycle. The assertion of $\overline{\text{ACK}}$ enables the FPJ11 output drivers.

The FPA RDY signal is asserted to indicate that output data is ready to be driven to DAL<15:00>. The FPA RDY signal will not be asserted prior to the low-to-high transition of the $\overline{\text{STRB}}$ signal for a FPJ11 write. The system interface is required to wait for the FPA RDY signal before continuing the bus write cycle. There is no required precedence between FPA RDY and $\overline{\text{ACK}}$ signals for bus write cycles. During GP read cycles, however, the system interface must drive the $\overline{\text{ACK}}$ signal before the FPJ11 will assert the FDA RDY signal. The FPJ11 output data is valid within t_{ODV} time of the assertion of the FPA RDY signal or within t_{OE} time of the assertion of the $\overline{\text{ACK}}$ signal, whichever is longer. After recognizing the FPA RDY signal, the system interface latches the FPJ11 output data and then deasserts the $\overline{\text{ACK}}$ line. The FPJ11 does not assure the hold time after the negation of $\overline{\text{ACK}}$. Upon detection of the low-to-high transition of $\overline{\text{ACK}}$, the FPJ11 deasserts the FPA RDY line. The system interface completes the bus write cycle with the latched FPJ11 output data and asserts the $\overline{\text{CONT}}$ signal to the DCJ11. If more than one word of output data is required, the same sequence repeats for each 16-bit word of output data.

If the previous instruction caused a floating-point exception, the FPJ11 will assert the FPA FPE signal and not FPA RDY during the FPJ11 bus write cycles. If the $\overline{\text{FPA FPE}}$ signal is asserted, the system interface must assert $\overline{\text{CONT}}$ to the DCJ11 and not perform the write operation to memory.

GP Read Transactions

The general purpose (GP) read cycle is used by the DCJ11 to transfer data from a system interface register or the FPJ11. The ADDR<1:0> bits are necessary for the FPJ11 to distinguish the type of GP read cycle being performed. The four types of GP read cycles that are recognized by the FPJ11 are described in the following paragraphs.

GP read powerup options (ADDR<1:0>=0)—During the GP read cycle of the powerup options, the FPJ11 will assert the $\overline{\text{FPA OP}}$ signal, indicating the presence of the FPJ11 in the system configuration.

GP read floating-point condition codes or 16-bit data (ADDR<1:0>=1)—During the GP read cycle of the floating-point condition codes (FCC) or 16-bit data, the system interface asserts both the DV and $\overline{\text{ACK}}$ signals and waits for the FPJ11 to assert the FPA RDY signal. $\overline{\text{ACK}}$ must be asserted by the system interface before the FPJ11 will assert FPA RDY. Upon recognizing the FPA RDY signal, the system interface deasserts DV, strobing the FPJ11 output data into the DCJ11. Setup and hold requirements with respect to the FPA RDY signal and $\overline{\text{ACK}}$ signal for FPJ11 output data are identical for GP read and bus write cycles. After the data is latched, the system interface may deassert $\overline{\text{ACK}}$, causing FPJ11 output drivers to become a high impedance. The FPJ11 will then deassert FPA RDY. If $\overline{\text{FPA FPE}}$ is asserted instead of FPA RDY, indicating a floating-point exception, the FPJ11 will transfer a floating-point exception code. The system interface must still complete the bus cycle but the DCJ11 will not use the data. A subsequent GP read transaction to the FEC register will occur to read the register again and clear the exception condition.

GP read powerup options and clear FPS register (ADDR<1:0>=2)—During the GP read cycle of powerup options and to clear the FPS register, the FPJ11 will assert the $\overline{\text{FPA OP}}$ signal, indicating the presence of the FPJ11 in the system configuration. The FPJ11 will also clear the FPS register. The G command in the ODT command language causes this cycle.

GP read floating-point exception code (ADDR < 1:0 > = 3)—The GP read cycle of the floating-point exception code (FEC) operates in the same manner as the GP read of FCC or 16-bit data cycle. The FPJ11 deasserts the $\overline{\text{FPA FPE}}$ signal to acknowledge the servicing of the FPE at the low-to-high transition of STRB input.

GP write (ADDR < 1:0 > = 3)—The GP write cycle with ADDR < 1:0 > = 3 is used by the DCJ11 to write mode 0 integer source data to the FPJ11. Setup and hold requirements for the data with respect to DV is identical to that of a data read cycle.

DCJ11 Stall

The FPJ11 asserts the $\overline{\text{FPA STL}}$ signal when executing a floating-point instruction that does not transfer data if the EU is still busy with a previous instruction. This signal is asserted prior to the DCJ11 test for a floating-point exception. The $\overline{\text{FPA STL}}$ signal is also asserted to minimize DMA latency in a DCJ11 system.

The $\overline{\text{FPA STL}}$ signal should be OR gated into the DCJ11 DMA request ($\overline{\text{DMR}}$) input. The system interface must assert the $\overline{\text{CONT}}$ input to the DCJ11 after the negation of $\overline{\text{FPA STL}}$ to restart program execution in the DCJ11.

There are two cases when the FPJ11 will assert the $\overline{\text{FPA STL}}$ signal. The FPJ11 does not maintain a copy of the virtual address of the executing instruction. Therefore, the DCJ11 maintains the floating-point exception address register. The DCJ11 microinstruction sequence determines the extent to which floating-point instructions can be overlapped. The DCJ11 can be allowed to overlap execution of a subsequent instruction only up to a point of the floating-point exception check. The FPJ11 monitors DCJ11 microinstructions and always stalls the DCJ11 while allowing a maximum overlap.

During FPJ11 load class instructions, this overlap allows the DCJ11 to complete data fetch operations for a subsequent instruction before it is stalled by the FPJ11. The effect of load class overlap in floating-point intensive code is significant when much of the data is located in memory.

The FPJ11 also asserts the $\overline{\text{FPA STL}}$ signal to limit the worst case DMA latency of the DCJ11 system. The system interface cannot service DMA requests while waiting for FPJ11 data within a write cycle. The FPJ11 will assert this signal prior to the write cycle allowing the system interface to service DMA requests if FPJ11 output data will not be ready within worst case DMA latency time. This condition can occur only if the execution unit is executing a previous MOD or DIVD instruction when the store instruction is decoded by the DCJ11.

• System Configuration

A typical DCJ11 system configuration with the FPJ11 and cache memory is shown in Figure 9. In a single bus system configuration, the ADDR < 1:0 > lines would be connected to DAL < 01:00 > pins.

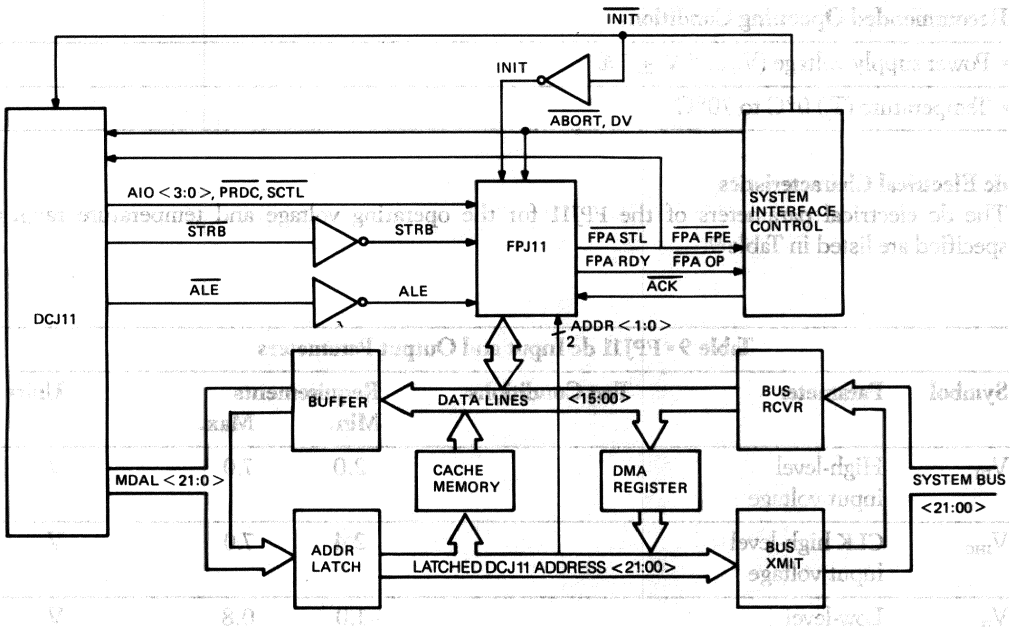


Figure 9 • FPJ11 Typical System Configuration

• Specifications

The mechanical, electrical, and environmental characteristics and specifications for the FPJ11 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Power supply voltage (V_{DD}): 5.0 V ±0.2 V
- Temperature range (T_A): 0°C to 70°C
- Ground (V_{SS}): 0 V

Mechanical Configuration

The physical dimensions of the FPJ11 40-pin package are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Power supply voltage (V_{DD}): 5 V ±5%
- Input voltage applied (V_{in}): −1.0 V to 7.0 V
- Power dissipation: 2.5 W
- Operating temperature (T_A): 0°C to 70°C
- Storage temperature: −65°C to 150°C

Recommended Operating Conditions

- Power supply voltage (V_{DD}): 5 V ±5%
- Temperature (T_J) 0°C to 70°C

dc Electrical Characteristics

The dc electrical parameters of the FPJ11 for the operating voltage and temperature ranges specified are listed in Table 9.

Table 9 • FPJ11 dc Input and Output Parameters

Symbol	Parameter	Test Conditions	Requirements Min.	Requirements Max.	Units
V_{IH}	High-level input voltage		2.0	7.0	V
V_{IHC}	CLK high-level input voltage		2.4	7.0	V
V_{IL}	Low-level input voltage		−1.0	0.8	V
V_{OH}	High-level output voltage	I_{OH} = −1.0 mA	2.5	—	V
V_{OL}	Low-level output voltage	I_{OL} = 4.0 mA	—	0.40	V
I_{IL}	Input-low leakage current	V_{in} = 0 V	—	±10	µA
I_{IH}	Input-high leakage current	V_{in} = 5.0 V	—	±10	µA
I_{OZL}	Low three-state leakage current	V_{in} = 0 V, $\overline{TEST}$ = 0 V	—	±100	µA
I_{OZH}	High three-state leakage current	V_{in} = 5.0 V, $\overline{TEST}$ = 0 V	—	±100	µA
I_{TST}	$\overline{TEST}$ short circuit current	$\overline{TEST}$ = 0 V	0.3	1.9	mA
I_{DD}	Power supply current	V_{DD} = 5.25 V	—	500	mA
C_{CLK}*	CLK capacitance	fc = 1 MHz. All unmeasured pins returned to GND	—	5.0	pF
C_{in}*	Input capacitance		—	5.0	pF
C_{out}*	Output capacitance		—	10	pF
C_{IO}*	I/O capacitance		—	15	pF

*T_A = 25°C, V_{DD} and V_{SS} = 0 V

ac Electrical Characteristics

The clock input waveform and timing symbols are shown in Figure 10. Table 10 contains the timing signal definitions and parameters for the clock input. The t_{CLKH} width is measured at 2.0 volts.

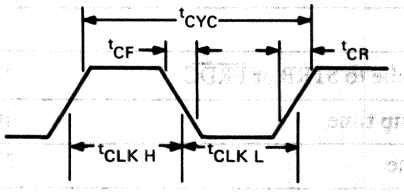


Figure 10 ▪ FPJ11 Clock Input Timing

Figures 11 through 15 show the signal timing and symbols for the following—A floating-point accelerator operation are shown in Figure 11. Figure 12 shows the timing sequence during FPA stall conditions. Figure 13 shows a read and cache hit transaction. Figure 14 shows a general purpose write and stretched read transaction. Figure 15 shows a write operation with the $\overline{RDY}$ signal asserted before the assertion of the stretch control ($\overline{SCTL}$) signal and a write and general purpose read operation with the system interface waiting for the RDY signal from the FPJ11.

The timing symbols and parameter definitions for the figures are listed in Table 10. The following specifications apply:

- All timing parameters values are in nanoseconds (ns).
- The ac characteristics are for a 100 pF capacitive load at the outputs.

Table 10 ▪ FPJ11 Signal Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{ACK}	Acknowledge response	—	60
t_{CDH}	Cache data hold	7	—
t_{CF}	Clock fall time	—	15
t_{CLKH}	Clock high width	T/2-3	—
t_{CLKL}	Clock low width	T/2-3	—
t_{CR}	Clock rise time	—	15
t_{CYC}	Clock cycle	T	—
t_{DLY}	Delay time $\overline{FPA\ STL}$, $\overline{FPA\ OP}$	—	2T
t_{DR}	Assert time $\overline{FPA\ FPE}$, FPA RDY	—	50
t_{DVF}	DV fall time	—	20

Symbol	Definition	Requirements (ns) Min.	Max.
t_{DVH}	DV data hold time	25	—
t_{DVPW}	DV pulse width	50	—
t_{DVR}	DV rise time	—	20
t_{IDPM}	Input data strobe to STRB or $\overline{PRDC}$	4T/5	—
t_{IDS}	Input data setup time	40	—
t_{IH}	Input hold time	25	—
t_{IS}	Input setup time	T/2	—
t_{LRDLY}	Last read delay to $\overline{FPA\ STL}$	—	3T/2
t_{ODH}	Output data hold time	0	—
t_{ODV}	Output data valid from FPA RDY	—	90
t_{OE}	Output data valid from $\overline{ACK}$	—	75

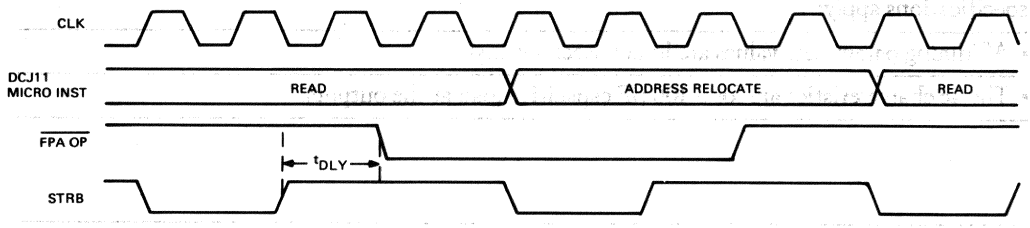
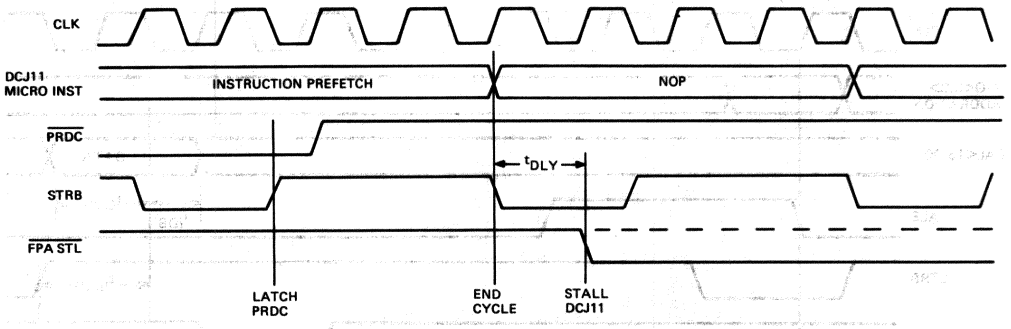
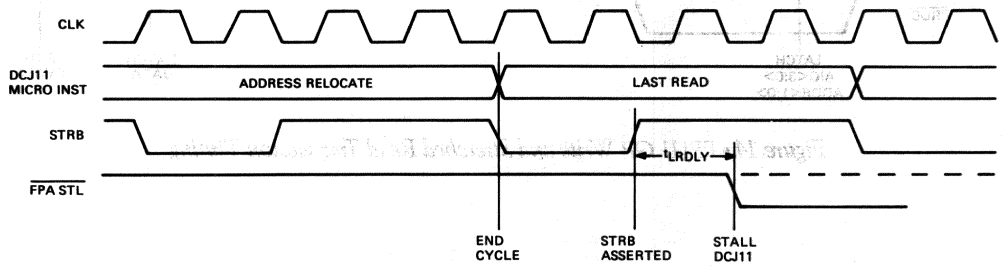


Figure 11 ▪ FPJ11 FPA Operation Timing



FPA Stall by Register Mode Instruction



FPA Stall after Overlap of Operand Fetch

Figure 12 • FPJ11 FPA Stall Condition Timing

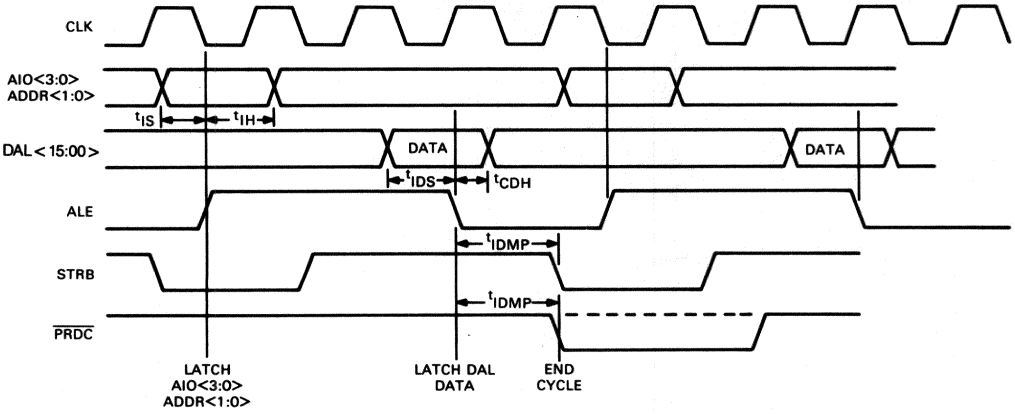


Figure 13 • FPJ11 Read and Cache Hit Transaction Timing

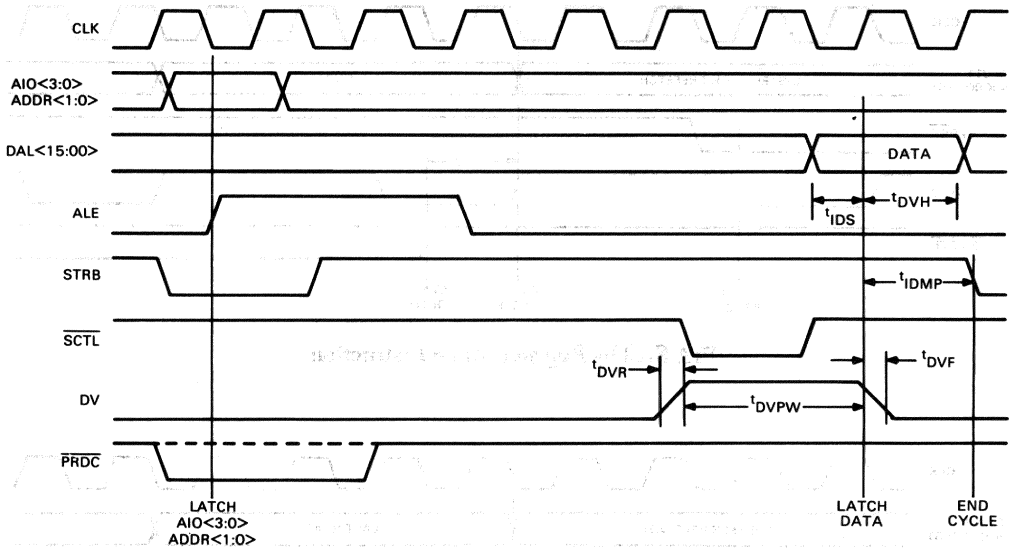
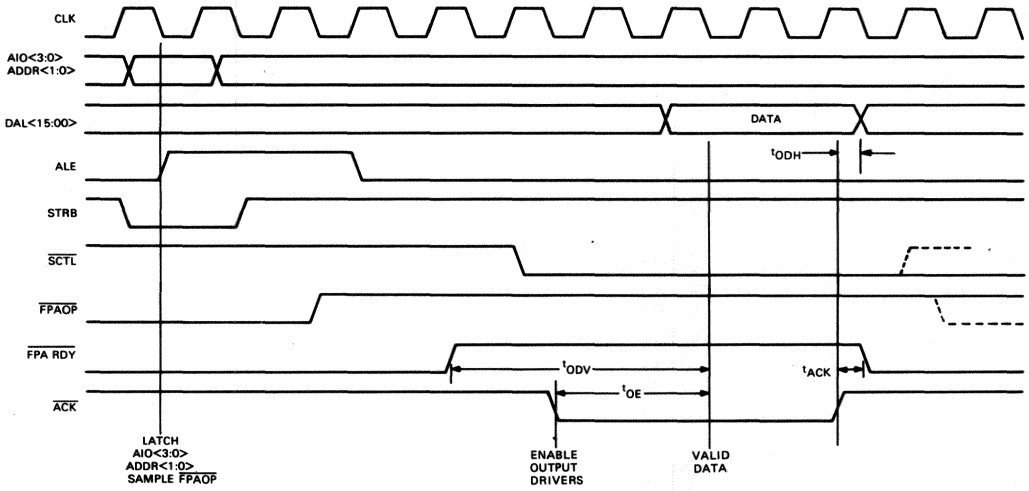
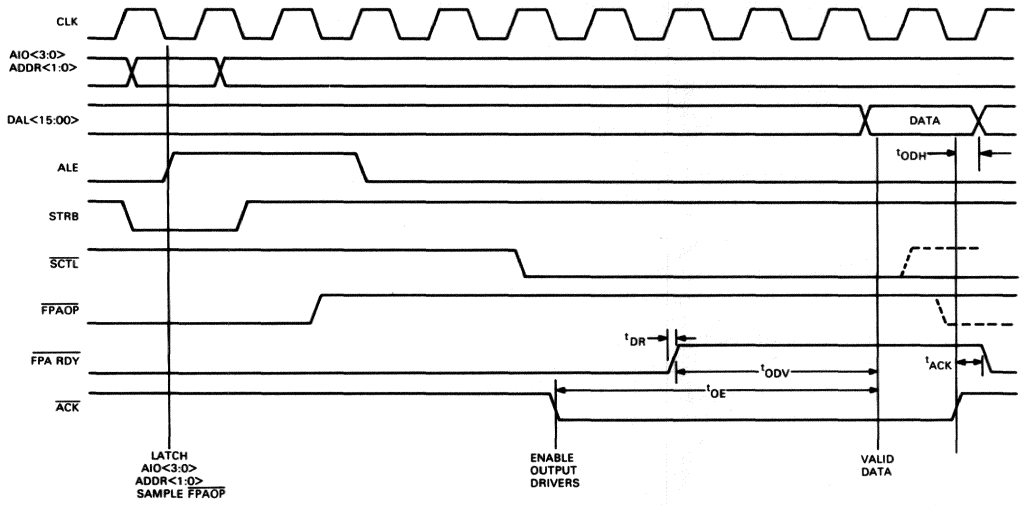


Figure 14 • FPJ11 GP Write and Stretched Read Transaction Timing

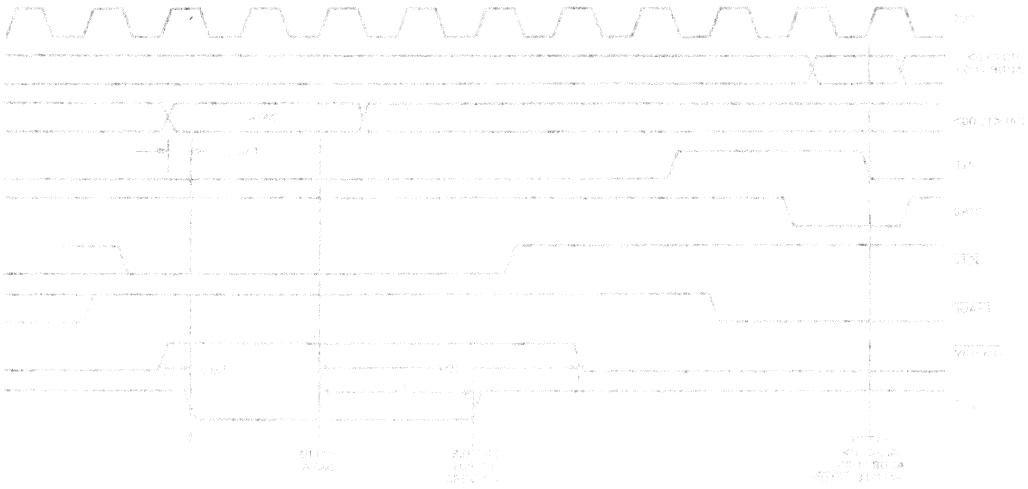


Writes with FPA RDY Asserted before SCTL



Writes and GP Read with System Interface Waiting for FPA RDY

Figure 15 • FPJ11 Write and GP Read Transaction Timing



Timing with FPA ITSY Internal Buffers Enabled

Timing and CP Load with Sensor Interface Timing for FPA ITSY

Figure 1-34 FPIU Internal CP Load Interface Timing

Features

- Basic PDP-11 instruction set (except for the MARK instruction)
- 16- or 8-bit data paths selected at initialization
- Interrupts on four priority levels with 15 internally generated vectors
- Option of having the interrupt device provide the vector address
- DMA arbitration
- Full dynamic memory support:
  - —Direct dynamic addressing
  - —$\overline{RAS}$ and $\overline{CAS}$ strobes
  - —Refresh counter
  - —Automatic refresh cycles
- Programmable mode register featuring
  - —8- or 16-bit external data bus
  - —Start and restart address selection
  - —Static or dynamic memory support
  - —Standard or long microcycles
  - —Bus synchronous or constant clock output
- Single 5-Vdc power supply

Description

The DCT11 microprocessor is a PDP-11 processor contained in a 40-pin, dual-inline package (DIP). It is available in two versions; one operates with a maximum clock frequency of 7.5 MHz (part no. 21-17311-00,01) and one operates with a maximum clock frequency of 10 MHz (part no. 21-17311-02). Full dynamic memory support is provided for 4K/16K or 64K chip memory. This includes timing strobes, dynamic address multiplexing, automatic refresh cycles, and a refresh counter. The interrupt is multilevel, using four priority levels. Vector addresses can be supplied by the DCT11 (15 internal vector addresses are available) or by the interrupting device. DMA arbitration is included in the DCT11. The maximum clock frequency is 10 MHz. All signals are TTL-compatible. Figure 1 is a block diagram of the DCT11 microprocessor.

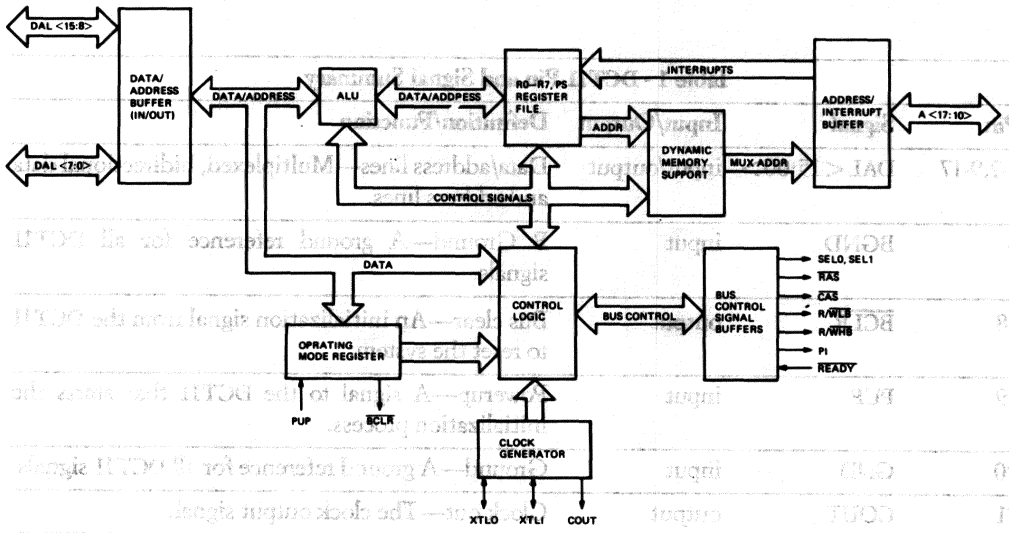


Figure 1 • DCT11 Microprocessor Block Diagram

• Pin and Signal Description

The DCT11 is a 40-pin microprocessor that functions with the input and output signals described in the following paragraphs. The signal pin assignments are identified in Figure 2 and summarized in Table 1.

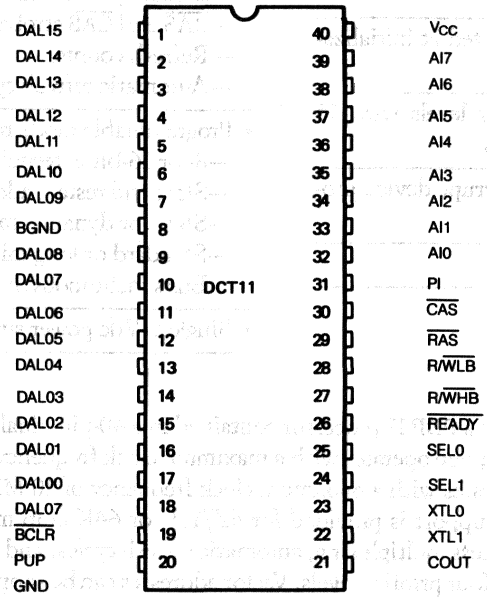


Figure 2 • DCT11 Pin Assignments

Table 1 • DCT11 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
1-7,9-17	DAL<15:00>	input/output	Data/address lines—Multiplexed, bidirectional data and address lines.
8	BGND	input	B Ground—A ground reference for all DCT11 signals.
18	$\overline{\text{BCLR}}$	output	Bus clear—An initialization signal from the DCT11 to reset the system.
19	PUP	input	Powerup—A signal to the DCT11 that starts the initialization process.
20	GND	input	Ground—A ground reference for all DCT11 signals.
21	COUT	output	Clock out—The clock output signal.

Pin	Signal	Input/Output	Definition/Function
22	XTL1		Crystal input 1—External crystal connection to the internal oscillator
23	XTL0		Crystal input 0—External crystal connection to the internal oscillator
24	SEL1	output	Select 1—Encoded with the SEL0 line to indicate the transaction being performed.
25	SEL0	output	Select 0—Encoded with the SEL1 line to indicate the transaction being performed.
26	$\overline{\text{READY}}$	input	Wait—Causes the DCT11 to become idle.
27	R/$\overline{\text{WHB}}$	input	Read/write high byte strobe—Provides read and write control to the system.
28	R/$\overline{\text{WLB}}$	output	Read/write low byte strobe—Provides read and write control to the system.
29	$\overline{\text{RAS}}$	output	Row address strobe—A system address strobe.
30	$\overline{\text{CAS}}$	output	Column address strobe—An address and chip select strobe.
31	PI	output	Priority in—The write, interrupt, and DMA request strobe.
32-39	AI<7:0> A<17:10>	input/output	Address/Interrupt lines—Multiplexed, bidirectional lines that transfer the dynamic memory address and receive all system interrupts and DMA requests.
40	V_{cc}	input	Voltage—Power supply voltage

Data Address and Interrupt Bus

Data and address bus (DAL<15:00>)—The data and address lines are time-multiplexed, bidirectional lines used to transfer address information and data. During a read or write transaction, the system address is transferred first followed by the data. The operation of the bus lines depends on the selection of the 8- or 16-bit mode.

Address and interrupt lines (AI<7:0>)—The address and interrupt lines are bidirectional, time-multiplexed lines used to address dynamic random access memory (RAM) and to receive interrupt and direct memory access requests. At the beginning of the bus cycle, the address on lines AI<7:0> is the same as the address on the DAL bus. The information on the AI lines depends on the selection of dynamic or static RAM mode of operation.

• Bus Control

Row and column address strobe ($\overline{RAZ}$ and $\overline{CAS}$)—When dynamic RAM support is selected, the row address strobe ($\overline{RAS}$) and column address strobe ($\overline{CAS}$) signals are used to latch the row and column address of lines AI<7:0> into dynamic memory. If static mode is selected, the $\overline{RAS}$ output is used as a system address strobe. The $\overline{CAS}$ output can be used by the external logic in either static or dynamic mode, as a chip select enable signal.

During read operations, the data on the data and address lines DAL<15:00> is read by the DCT11 when the $\overline{CAS}$ signal is negated. During write operations, the negation of $\overline{RAS}$ or $\overline{CAS}$ signals can be used to latch data into the system interface.

Read/wite high byte and low-byte (R/$\overline{WHB}$) and R/$\overline{WLB}$)—The read and write high byte and read and write low byte outputs specify the direction of the information transfer on the DAL<15:0> lines during the input or output portion of a read or write bus cycle. The function and name of the read and write lines depend on the selection of 8- or 16-bit data bus mode.

Select 1 and 0 (SEL1 and SEL0)—These lines are encoded by the DCT11 to indicate the type of bus transaction that is being performed.

Ready ($\overline{READY}$)—This line is asserted by external logic to extend the current bus cycle.

System Control

Bus clear ($\overline{BCLR}$)—The bus clear line is asserted by the processor during the powerup sequence and during the PDP-11 RESET instruction. The DAL<15:8> and DAL<1:0> lines receive the mode register information during the assertion of the $\overline{BCLR}$ signal and the selected bits are loaded into the DCT11 mode register.

Powerup (PUP)—The powerup signal resets the processor. When this signal is asserted, the DCT11 stops all operation and an initialization sequence is executed when the signal is negated.

Interrupt Control

Priority in (PI)—The priority in signal is used as the system priority enable strobe. When the PI line is asserted, the AI<7:0> lines receive interrupt inputs and the direct memory access request ($\overline{DMR}$). This signal may also be used as a data strobe for read or write operations.

Clock Signals

Clock out (COUT)—The clock-out line provides a clock signal that is controlled by the selection of the mode register. This output can be one-half of the DCT11 oscillator frequency or a pulse asserted once during each internal microcycle.

Crystal-inputs (XTAL0 and XTAL1)—The crystal-input lines are external crystal connections to the internal clock generator. A crystal or an external TTL-level clock generator may be used as an input. When an external oscillator is used, it connects to the XTL1 input and the XTL0 input connects to ground.

Power Supply Connections

Supply voltage (V_{cc})—Connects to the 5 Vdc power supply.

Ground and B Ground (GND and BGND)—These ground pins connect together and to the system ground to provide ground references for all lines of the DCT11.

• Architecture Summary

The DCT11 microprocessor contains eight 16-bit general purpose registers, a processor status register, and a mode register. These registers, except for the mode register, are accessible to the system programmer for developing application programs and to the user for checking the validity of programs and for performing maintenance functions.

General Purpose Registers

The eight 16-bit general purpose registers are shown in Figure 3. These registers operate as accumulators, index registers, autoincrement registers, autodecrement registers, or as stack pointers for temporary storage of data.

Registers R6 and R7 are dedicated. R6 operates as the stack pointer (SP) and stores the location (address) of the last entry in the hardware stack. Register R7 operates as the processor program counter (PC) and stores the address of the next instruction or operand to be used.

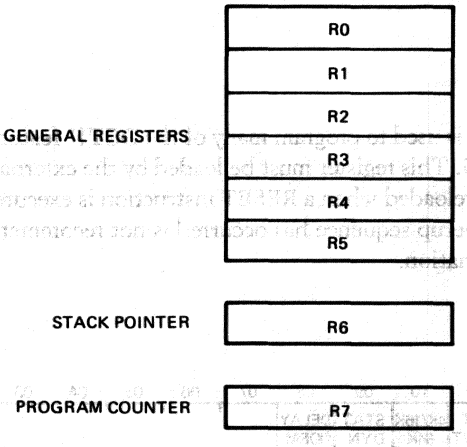


Figure 3 • DCT11 General Purpose Registers

Processor Status Register

The processor status register contains the processor status word (PSW) consisting of condition codes, trap bit, and current processor priority. The processor status register format is shown in Figure 4. Table 2 lists the functions of the register information.

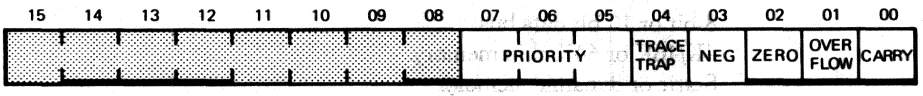


Figure 4 • DCT11 Processor Status Register Format

Table 2 • DCT11 Processor Status Register Description

Bit	Description
07-05	Priority level bits used by the software to determine which interrupt will be serviced.
04	The trace bit used in debugging programs. During a trap or interrupt operation, the trace bit can be set or cleared when returning from the interrupt by using a Return from Interrrupt (RTI) or Return from Trap (RTT) instruction.
03-00	The condition codes contain information about the result of the last CPU arithmetic or logical instructions. The bits are as follows: N = 1 The result was negative. Z = 1 The result was 0. V = 1 The operation resulted in an arithmetic overflow. C = 11 The operand resulted in a carry from the most significant bit or a 1 was shifted from the most significant bit or least significant bit.

Mode Register

The 16-bit mode register (MR) is used to program many of the DCT11 features. The mode register bit format is shown in Figure 5. This register must be loaded by the external hardware during the powerup sequence. It may be reloaded when a RESET instruction is executed; however, changing processor modes after the powerup sequence has occurred is not recommended. Table 3 lists the functions of the register information.

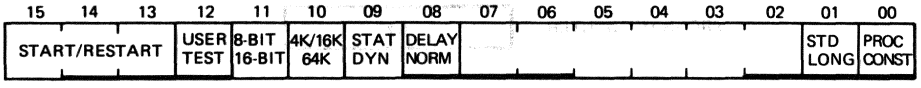


Figure 5 • DCT11 Mode Register Format

Table 3 • DCT11 Mode Register Description

Bit	Description
15-13	Start or restart address.
12	User or tester mode.
11	8-bit or 16-bit data bus.
10	4K/16K or 64K chip memory.
09	Static or dynamic memory.
08	Delay or normal read or write strobes.
07-02	Not used.
01	Standard or long microcycle.
00	Processor or constant mode clock.

Table 4 • DCT11 Mode Register Starting Address Assignments

Start/Restart (bits 15:13)	Start Address	Restart Address
7	172000	172004
6	173000	173004
5	000000	000004
4	010000	010004
3	020000	020004
2	040000	040004
1	100000	100004
0	140000	140004

• Bus Operation

The following paragraphs describe the operation of the DCT11 microprocessor bus during the execution of instructions. Each PDP-11 instruction consists of one or more transactions. A transaction is the activity on the bus required to perform the following operations.

- Read
- Write
- Refresh
- DMA (Direct memory access)
- IACK (Interrupt acknowledge)
- ASPI (Assert priority in)
- NOP (No operation)

Each transaction consists of one or two microcycles and a microcycle consists of three or four cycles of the basic oscillator. One internal microinstruction is executed for each microcycle. The number of microcycles required for a read and write transaction depends on the mode register selections. The standard microcycle mode uses three clock cycles for most transactions. The long microcycle mode uses four microcycles for all transactions. One microinstruction is executed during each microcycle. During a microinstruction, address information is transferred, data is transferred to and from the bus, and internal operations are performed. Four clock cycles are used during the REFRESH, IACK, DMA, and ASPI transactions.

Read and Write Transactions

During a read or write transaction, the microprocessor transfers address information, sends or receives data, and monitors interrupt and DMA requests. Read, write, and DMA operations are modified by the mode register during powerup. The use of control signals and the number of transactions required for each operation depends on the following selections.

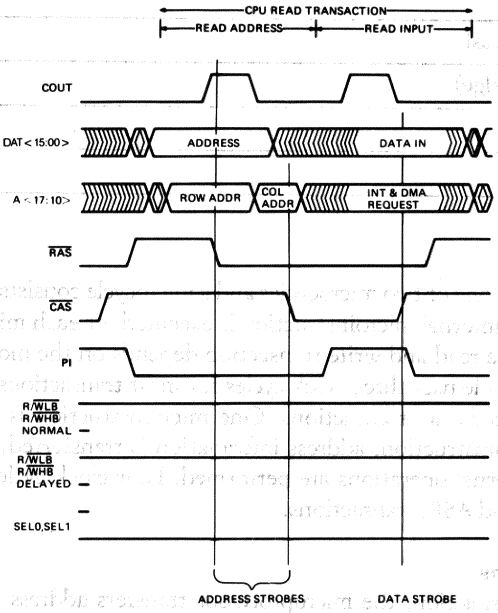
- Static or dynamic memory
- 4/16K or 64K chip memory
- 8- or 16-bit data bus operation

A write transaction is preceded by a read transaction except when writing to the stack during an interrupt or trap operation.

Address Selection

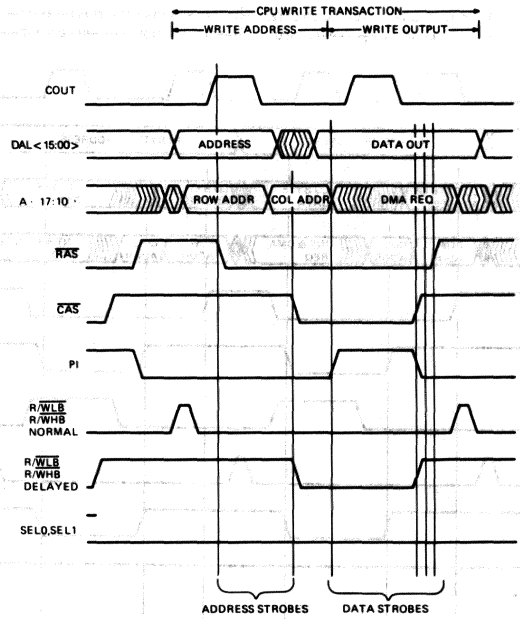
The address information is transferred on the DAL<15:00> and AI<7:0> lines depending on the type of operation.

Dynamic Operation—The timing sequence for the 16-bit read and write operation with dynamic memory is shown in Figure 6. The timing sequence for the 8-bit read or write operation with dynamic memory is shown in Figure 7. When dynamic RAM operation is selected, the address present on lines DAL<15:00> is time-multiplexed through the AI<7:0> outputs. The row address and then the column address is transferred at the beginning of a read or write transaction. The row address is valid before the assertion of the $\overline{RAS}$ signal and the column address is valid before the assertion of the $\overline{CAS}$ signal. The multiplexed address is the same address that is transferred by the DAL bus before the $\overline{RAS}$ signal occurs. Lines AI<7:0> are also used to transfer the internal refresh counter as a row address during the REFRESH transaction.



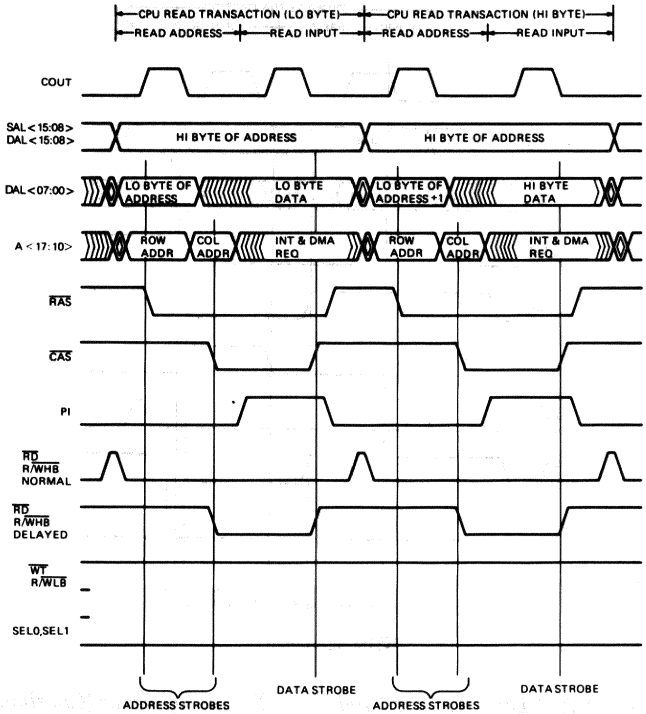
Read Sequence

Figure 6 • DCT11 16-bit Dynamic Read and Write Timing Sequence



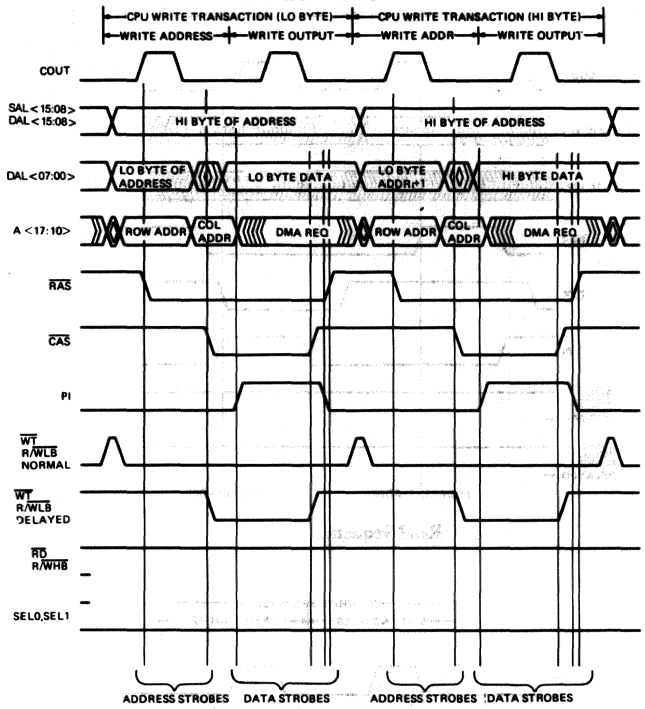
Write Sequence

Figure 6 ▪ DCT11 Dynamic Read and Write Timing Sequence (Continued)



Read Sequence

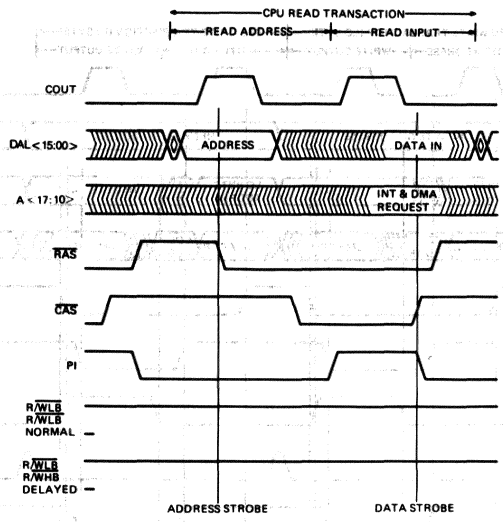
Figure 7 • DCT11 8-bit Dynanic Read and Write Timing Sequence



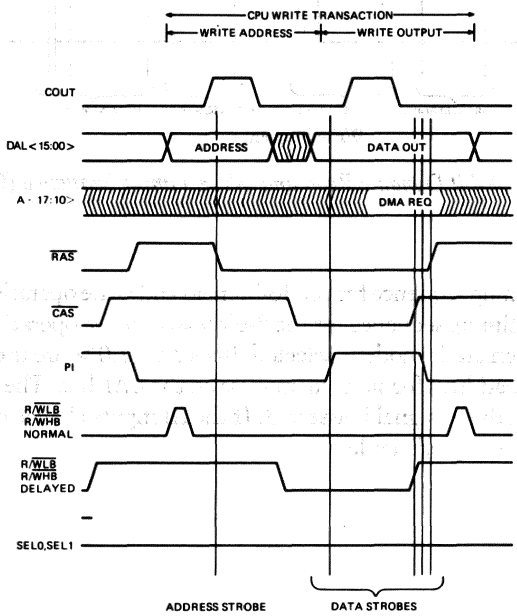
Write Sequence

Figure 7 • DCT11 8-bit Dynanic Read and Write Timing Sequence (Continued)

Static Operation—The timing sequence for the 16-bit read and write operation with static memory is shown in Figure 8. The timing sequence for the 8-bit read or write operation with static memory is shown in Figure 9. When static mode is selected, lines AI<7:0> are used to receive interrupt and DMA requests and the address for static memory is on the DAL bus. The information on the AI lines should be valid when the PI signal is asserted. If the AI inputs change during the assertion of the PI signal, the results are unpredictable.

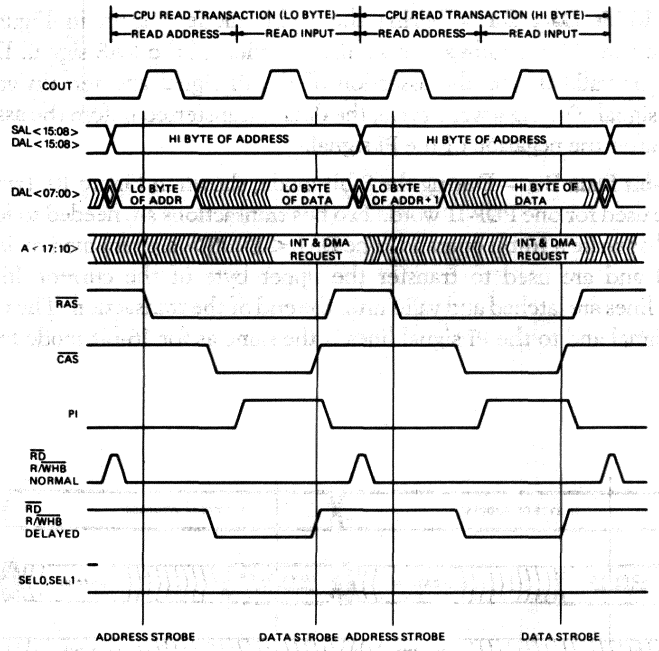


Read Sequence

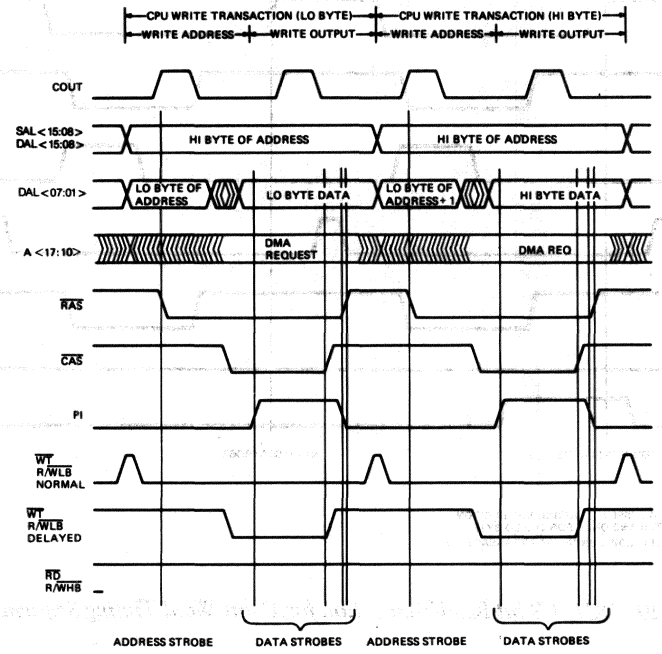


Write Sequence

Figure 8 • DCT11 16-bit Static Read and Write Timing Sequence



Read Sequence



Write Sequence

Figure 9 ▪ DCT11 8-bit Static Read and Write Timing Sequence

Address and Data, 16-bit Data Bus—During the 16-bit mode, shown in Figures 6 and 8, DAL<05:00> contain the 16-bit address before the assertion of the $\overline{RAS}$ signal. During a read cycle, the data must be valid during the assertion of the PI signal and remain valid until the negation of the $\overline{CAS}$ signal. During a write cycle, the data is transferred before the assertion of the PI signal and is valid after the negation of the PI signal.

Address and Data, 8-bit Data Bus—During the 8-bit mode, shown in Figure 10, two consecutive memory locations are used for one PDP-11 word. Two bus transactions are needed to fetch a PDP-11 instruction or to read or write a 16-bit operand. The DAL<15:08> are renamed to SAL<15:08> (static address lines) and are used to transfer the upper byte of the current 16-bit address. SAL<15:08> signal lines are latched and valid until the end of the transaction. The timing of data relative to the $\overline{CAS}$ signal and to the PI signal lines is the same as for 16-bit mode read and write transactions.

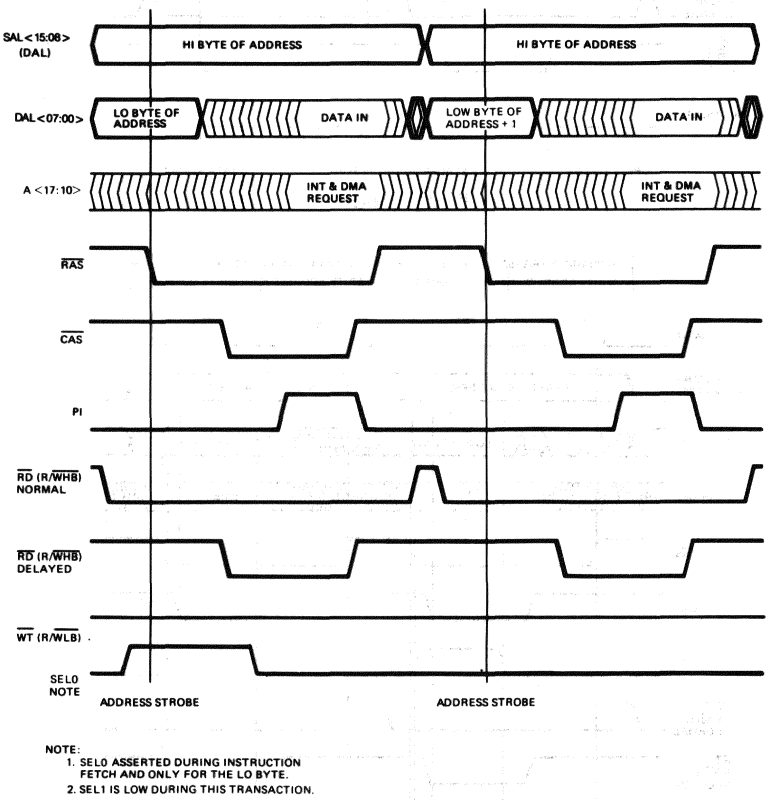


Figure 10 ▪ DCT11 8-bit Read Transaction for 16-bit Word Timing Sequence

For an 8-bit mode instruction fetch or for operations on 16-bit data, the SAL<15:08> lines contain the upper byte of the current 16-bit address and DAL<07:00> contain the lower byte of the address. As in 16-bit mode, the address is valid before the assertion of the $\overline{RAS}$ signal. The low byte of the word is then transferred in or out during the data part of the transaction. The DAL<07:00> outputs then the low byte of the word address + 1, and the high byte of the 16-bit word is transferred. When dynamic RAM support is selected, the address on DAL<15:00> is multiplexed through the AI<7:0> lines for each bus transaction.

When a PDP-11 BYTE instruction is executed, only one bus transaction is needed to transfer the source or destination operands. To fetch any instruction a word operation is always used.

Read and Write Control

The DCT11 specifies the current bus transaction by the signals on the R/$\overline{WHB}$ and R/$\overline{WLB}$ (read/write) lines which may be modified by the mode selections.

Selection of normal read/write mode causes the R/$\overline{WHB}$ and R/$\overline{WLB}$ signals to be asserted before the $\overline{RAS}$ signal and is valid throughout the transaction.

Selecting delayed read/write mode causes the R/$\overline{WHB}$ and R/$\overline{WLB}$ signals to be asserted with the same timing as the $\overline{CAS}$ signal. Selecting the 8- or 16-bit data bus operation changes the function of these lines.

Read/Write and 16-bit Data Bus—During 16-bit bus mode write transactions, Figures 6 and 8, the R/$\overline{WHB}$ and R/$\overline{WLB}$ signals indicate which byte of DAL<15:00> will contain valid write data. During a byte write transaction, all the DAL bus lines will contain information; however, the unused byte will be undefined. The valid write data is indicated by the read/write signals that is asserted low. When only one output is asserted, the data is a byte operand. When the R/$\overline{WHB}$ line asserted, the valid data is on DAL<15:08> and when the R/$\overline{WLB}$ line is asserted, the valid data is on DAL<07:00>. When neither read/write line is low, a read operation occurs and DAL<05:00> are in a high-impedance condition during this operations. Byte swapping is performed within DCT11 during read transactions.

Read/Write and 8-bit Data Bus—Selecting 8-bit data bus mode, shown in Figure 7 and 9, changes the functions of the signals on the read/write lines. The R/$\overline{WHB}$ line becomes a read signal when asserted low, and the R/$\overline{WLB}$ signal becomes a write signal when asserted low. The signal functions are as follows:

16-Bit Data Bus	8-Bit Data Bus
R/$\overline{WHB}$ (L = write operation)	R/$\overline{WHB}$ (L = read operation)
R/$\overline{WLB}$ (L = write operation)	R/$\overline{WLB}$ (L = write operation)
R/$\overline{WHB}$ (H = read operation)	
R/$\overline{WLB}$ (H = read operation)	

The functions of $\overline{RD}$ and $\overline{WT}$ signals are also affected by selecting normal or delayed read/write mode. Normal mode asserts the $\overline{RD}$ or $\overline{WT}$ signal before the leading edge of the $\overline{RAS}$ signal. These signals are valid for the complete transaction. In the delayed mode, the $\overline{RD}$ and $\overline{WT}$ signals have the same timing as the $\overline{CAS}$ signal.

Refresh—The dynamic memory of the system is automatically refreshed every 2 milliseconds by a 256-bit counter in the DCT11. A refresh transaction, shown in Figure 11, adds a microcycle to the transaction in progress

- After every other PDP-11 instruction fetch in 16-bit mode.
- After every PDP-11 instruction fetch in 8-bit mode.
- When an additional refresh occurs for addressing modes 5, 6, or 7.
- When a refresh microcycle occurs twice during a trap instruction.

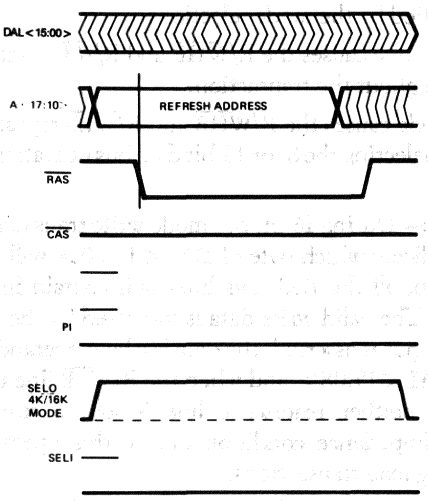


Figure 11 • DCT11 Refresh Transaction Timing Sequence

Extending Bus Transactions—Bus transactions can be extended for slow devices by asserting the $\overline{\text{READY}}$ signal during the transactions. Figure 12 shows the $\overline{\text{READY}}$ signal timing sequence. In addition to read and write transactions. the IACK and DMA transactions may also be extended.

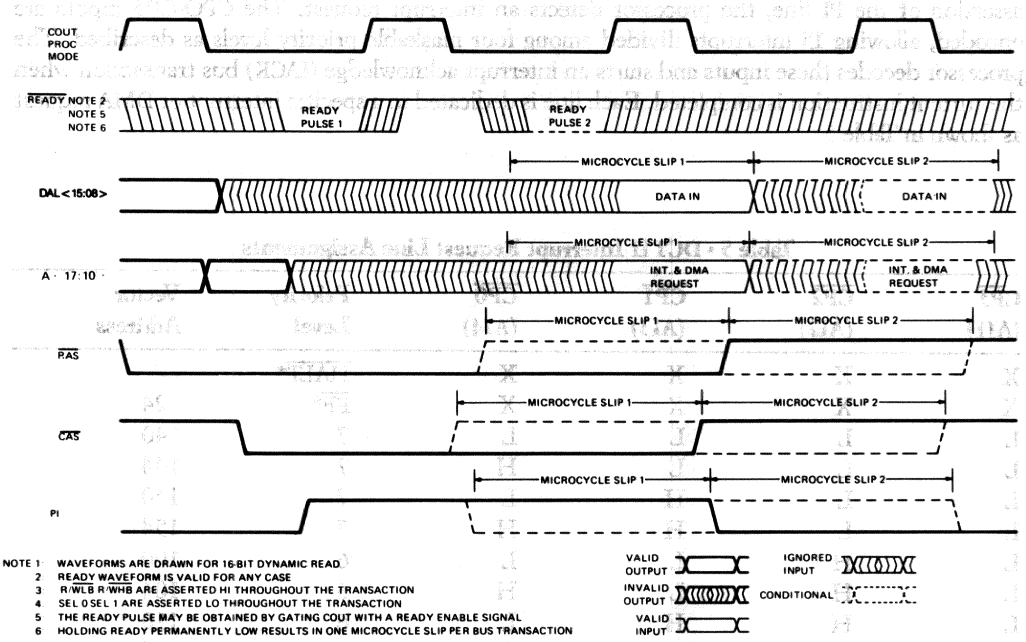


Figure 12 • DCT11 $\overline{\text{READY}}$ Timing Sequence

Each assertion of the $\overline{\text{READY}}$ line causes the processor to add one microcycle to the current bus operation. The length of the microcycle depends on the selection of standard microcycle mode or long microcycle mode operation. The $\overline{\text{READY}}$ input must be pulsed low for each "wait" microcycle to be added. When more than one added microcycle is needed for slow devices, the DCT11 requires a $\overline{\text{READY}}$ signal transition from low to high and then to low. If the $\overline{\text{READY}}$ line is connected to ground, one microcycle slip will occur for each bus transaction. The $\overline{\text{RAS}}$ signal must be asserted for the $\overline{\text{READY}}$ signal to have any affect.

Interrupt and DMA Requests

When the DCT11 asserts the PI line, interrupts and DMA requests can be received by the DCT11 through the AI<7:0> lines. The AI<7:0> lines are normally set to a high level by internal pullup circuits and must be driven low to cause an interrupt. If static mode is selected, these lines are used only as inputs and are kept high during the address part of the bus cycle. These lines provide $\overline{\text{PF}}$, $\overline{\text{HALT}}$, $\overline{\text{CP0}}$-$\overline{\text{CP3}}$ interrupts and the $\overline{\text{DMR}}$ request.

Direct Memory Request—When the processor detects a direct memory access request on line AI<0>, it releases control of the DMA bus to the device that asserts the $\overline{DMR}$ line. The DMA bus consists of the DAL<15:00>, AI<7:0> and read/write lines. The DCT11 maintains control of all other signals.

Interrupt Request—When one or more of the $\overline{CP0}$-$\overline{CP3}$ lines (AI<4:1>) are asserted during the assertion of the PI line, the processor detects an interrupt request. The $\overline{CP0}$-$\overline{CP3}$ inputs are encoded, allowing 15 interrupts divided among four maskable priority levels as described. The processor decodes these inputs and starts an interrupt acknowledge (IACK) bus transaction when the current instruction is completed. Each line is dedicated to a specific interrupt or DMA request as shown in Table 5.

Table 5 • DCT11 Interrupt Request Line Assignments

$\overline{CP3}$ (AI1)	$\overline{CP2}$ (AI2)	$\overline{CP1}$ (AI3)	$\overline{CP0}$ (AI4)	Priority Level	Vector Address
X	X	X	X	$\overline{HALT}$*	—
X	X	X	X	$\overline{PF}$*	24
L	L	L	L	7	140
L	L	L	H	7	144
L	L	H	L	7	150
L	L	H	H	7	154
L	H	L	L	6	100
L	H	L	H	6	104
L	H	H	L	6	110
L	H	H	H	6	114
H	L	L	H	5	124
H	L	H	L	5	130
H	L	H	H	5	134
H	H	L	L	4	60
H	H	L	H	4	64
H	H	H	H	No action	

*$\overline{HALT}$ and $\overline{PF}$ signals are nonmaskable interrupts. The $\overline{HALT}$ interrupt loads the PC with the restart address and the PSW with the value 340.

Vector Assignments—When the $\overline{VEC}$ line (AI5) is asserted during a priority interrupt and one or more of the $\overline{CP0}$-$\overline{CP3}$ lines are asserted, the processor will accept an external vector value during an IACK transaction. If the $\overline{VEC}$ line is not asserted at the same time that an interrupt is requested, the processor will provide a vector address from an internal table. The vector value will be one of the 15 assigned to each CP interrupt code. Four interrupt vectors are assigned to each priority level 7, 6, and 5 and three interrupt vectors are assigned to level 4 as indicated in Table 5.

Powerfail—A powerfail hardware interrupt that vectors through location 24 occurs if the $\overline{PF}$ signal is asserted on line AI6 when the PI signal is asserted. This interrupt is nonmaskable and is processed regardless of the current interrupt priority level.

Halt—The $\overline{HALT}$ interrupt request, assigned to line AI7, is a nonmaskable interrupt that has a higher priority than powerfail. After saving the current program counter and processor status values, the processor will set the priority to 340 and jump to the restart address. The restart address is selected during powerup when the mode register is loaded. The $\overline{HALT}$ input is pseudo-edge triggered and must be read as a negation before another assertion is accepted by the processor.

Direct Memory Access—The DCT11 provides a direct memory access (DMA) interface that can be connected to single-channel or multiple-channel DMA circuits. Requests for DMA are through the direct memory request $\overline{DMR}$ input on line AI0. A direct memory grant (DMG) will occur after internal arbitration. The processor completes the current bus transaction and relinquishes bus mastership. Figure 13 shows the timing sequence of the DMA transaction.

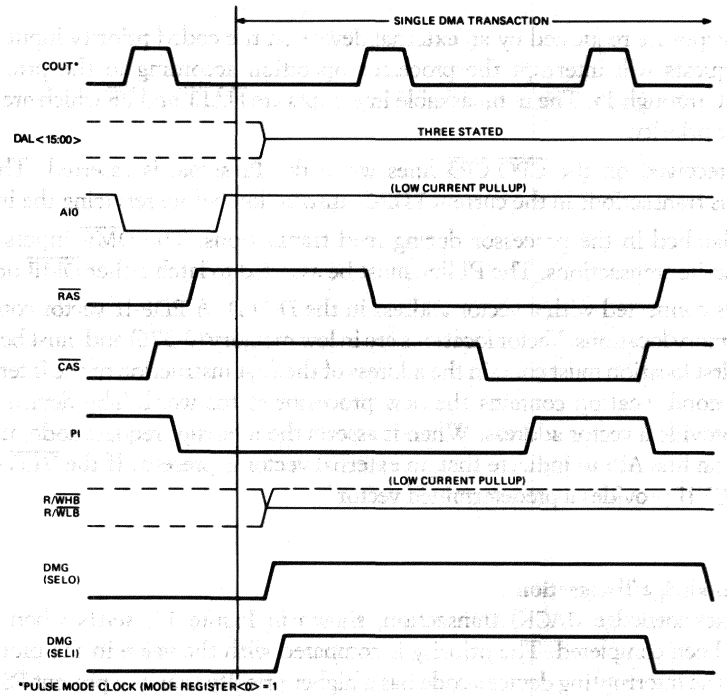


Figure 13 ▪ DCT11 DMA Timing Sequence

The new bus master gains control of the bus when the SEL0 and SEL1 signals are asserted. When a DMG (SEL0 and SEL1 are both high) is received, the bus master must take control of the DMA bus. The DMA bus consists of the R/$\overline{WHB}$ and R/$\overline{WLB}$ lines ($\overline{RD}$ and $\overline{WT}$ in 8-bit mode), the AI<7:0>, and the DAL<15:00> lines. The AI<7:0> and read/write lines are asserted through low-current pullup circuits and DAL<15:00> are in a high-impedance condition during the DMA transfer. The processor maintains control over $\overline{RAS}$, $\overline{CAS}$, PI, COUT, SEL0, and SEL1 lines to provides the new bus master with convenient timing signals for interfacing to dynamic memories.

The bus master must control the read and write functions, provide addresses, and be capable of driving or receiving data. In dynamic memory systems, an address must be multiplexed on lines AI<7:0> so that row and column addresses are provided at the correct times.

When a data input or output transfer occurs, the direction of the transfer is controlled by the state of the read and write lines. The DCT11 continues to issue grants by asserting the DMG signal until the $\overline{DMR}$ signal is not asserted. The DCT11 then continues its usual operation.

Because the DCT11 controls refresh operations based on the number of transactions needed to execute the PDP-11 instruction set, consecutive DMA operations are not recommended.

Interrupt Operation

The DCT11 uses a vectored, multilevel interrupt structure. Four priority levels are masked by the upper three bits of the processor status register. The two types of interrupts are maskable and nonmaskable.

Maskable interrupts are requested by an external device on the coded priority inputs lines $\overline{CPO}$-$\overline{CP3}$. These requests will interrupt the processor operation according to the priority level of interrupt codes 0 through 15. The nonmaskable interrupts are $\overline{HALT}$ and $\overline{PF}$ which are not masked by the processor priority.

Interrupts are received on the $\overline{CPO}$-$\overline{CP3}$ lines when the PI signal is asserted. The processor completes all bus transactions in the current PDP-11 instruction before servicing the interrupt.

Interrupts are latched in the processor during read transactions. The $\overline{DMR}$ inputs are latched during read or write transactions. The PI line must be asserted to latch either $\overline{DMR}$ or interrupts.

Each CP code is connected with a vector address in the DCT11. A PDP-11 vector consists of two consecutive memory locations. Vector locations are in low memory (0-376) and must be assigned by software. The first location must contain the address of the first instruction of the interrupt service routine. The second location contains the new processor status word. The device causing the interrupt may provide a vector address. When it asserts the interrupt request code, it also asserts the $\overline{VEC}$ signal on line AI5 to indicate that an external vector is present. If the $\overline{VEC}$ signal is not asserted, the DCT11 provides a predetermined vector.

Interrupt Acknowledge Transaction

The interrupt acknowledge (IACK) transaction, shown in Figure 14, starts when the current instruction has been completed. The priority is compared with the value in the processor status register PSR. If the interrupting device's code has a higher priority than the present PSR value, the interrupt request is serviced. The SEL0 and SEL1 lines indicate that the current bus transaction is an IACK. The $\overline{RAS}$ signal is the only timing strobe asserted during IACK. When it is asserted, DAL<15:08> (SAL<15:08> in 8-bit bus mode) transfer the priority interrupt code that is being acknowledged as shown in Figure 15. The DAL<07:02> lines contain the vector input if the $\overline{VEC}$ signal is asserted.

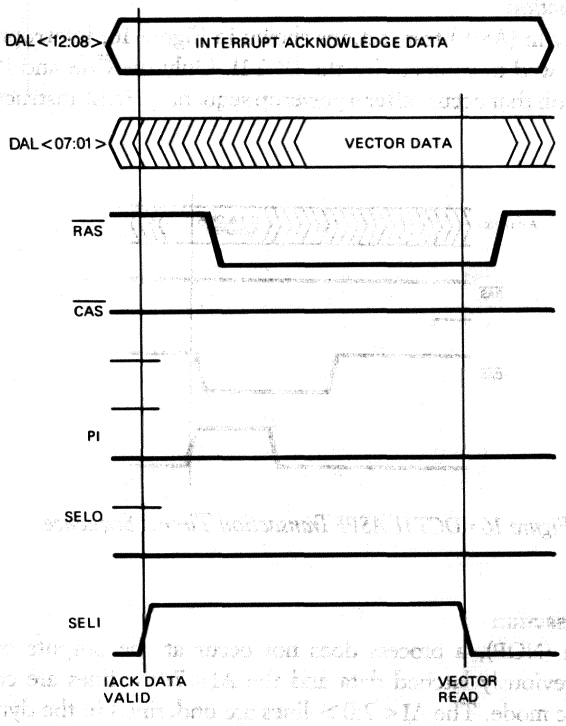


Figure 14 • DCT11 IACK Transaction Timing Sequence

The current contents of the PSR are then placed on the hardware stack. The program counter (PC) value at the time of the interrupt is then placed onto the stack. The PC is loaded with the address of the interrupt service routine from the vector location and the new PS is loaded into the PS register from the vector location + 2. When completed, the service routine ends with an RTI (Return from interrupt) instruction that causes the PC and PSW values to be recovered from the stack and the processor will continue executing the interrupted program.

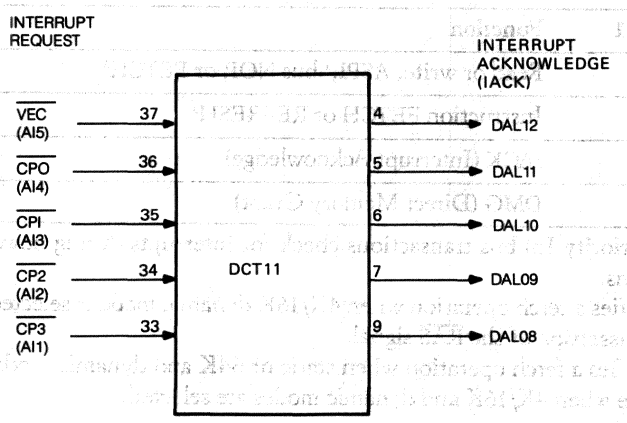


Figure 15 • DCT11 Interrupt Request/Acknowledge Lines

Assert Priority in Transaction

During an assert priority in (ASPI) transactions shown in Figure 16, interrupts or DMA requests that are pending are allowed to compete for the DCT11. Only the $\overline{CAS}$ and PI lines are asserted during an ASPI transaction that occurs after a powerup sequence, HALT instruction, halt interrupt, or WAIT instruction.

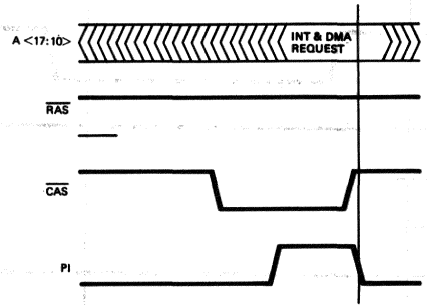


Figure 16 • DCT11 ASPI Transaction Timing Sequence

No Operation Bus Transaction

During a no operation (NOP), a process does not occur at the outputs of the DCT11. The DAL<15:00> have previously latched data and the AI<7:0> lines are connected to pullup circuits during the static mode. The AI<7:0> lines are undefined in the dynamic mode and all controls signals are unasserted.

Status Flags

The SEL0 and SEL1 lines are processor status indicators. The type of transaction in process can be detected by decoding these lines. The signal timing of the these lines depends on the type of transaction. Table 6 shows the select line assignments.

Table 6 • DCT11 Status Indicator Selection

SEL1	SEL1	Function
L	L	Read or write, ASPI,[1] bus NOP, or FETCH[2]
L	H	Instruction FETCH or REFRESH[3]
H	L	IACK (Interrupt Acknowledge)
H	H	DMG (Direct Memory Grant)

[1]ASPI (Assert Priority In) bus transactions check for interrupts during powerup and the PDP-11 WAIT intructions.

[2]This code specifies a fetch operation when 4K/16K dynamic mode is selected and AI0 is asserted low during the assertion of the $\overline{RAS}$ signal.

[3]This code specifies a fetch operation when static or 64K and dynamic modes are selected, and a refresh bus cycle when 4K/16K and dynamic modes are selected.

In addition to the select line outputs, the AI0 line indicates that an instruction fetch is in progress. This occurs before the assertion of the $\overline{RAS}$ signal when the 4K/16K chip dynamic RAM mode is selected. The AI0 line may also be asserted during a refresh transaction. The AI0 signal is controlled by the most significant bit of the internal refresh counter. The SEL0 and SEL1 signals can be used to verify that a refresh operation is in process. When 64K chip dynamic RAMs are used, AI0 is the processor address bit AI5.

Signal Line Summary

The following tables summarize the function of data and address bus control signal, and miscellaneous signal information of the DCT11.

Bus Operation Summary—Table 7 is a summary of the function of the information contained on the lines DAL<15:00> and AI<7:0> during bus operations.

Table 7 • DCT11 Data and Address Bus Summary

Pin	Line	Function
1-7,9	DAL<15:08>	16-bit mode
	SAL<15:08>	8-bit mode
10-17	DAL<07:00>	8- or 16-bit mode
	DAL<15:00>	Three-stated during DMA and asset priority (ASPI) transactions Contain previously data during a NOP or refresh transaction.
	DAL<01:00> DAL<15:08>	Read into the mode register on powerup or during the RESET instruction.
	DAL<07:02>	Input an external vector from the interrupting device during an interrupt acknowledge (IACK) transaction if the $\overline{VEC}$ signal was asserted during PI.
	DAL<12:08>	Used to output information present on AI<5:1> during the IACK transaction.
	AI<5:1>	Used to transfer interrupt request to the processor.
	AI0	Used to transfer a DMA request to the processor.
	AI<7:0>	Used to transfer the row and column addresses from the processor.
	AI5	Provides a control signal to indicate when an external vector is to be used.
	AI<7:0>	Receives inputs in static mode and contains previously latched data in dynamic mode during NOP and IACK transactions.

Address/Interrupt Lines—The interrupt and addressing information presented on the AI<7:0> lines is listed in Table 8.

Table 8 • DCT11 Address Line Functions

Pin	Line	All Modes @ PI	4K/16K Dynamic $\overline{RAS}$	4K/16K Dynamic $\overline{CAS}$	64K Dynamic $\overline{RAS}$	64K Dynamic $\overline{CAS}$
32	AI0	$\overline{DMR}$	FET/REF*	A14	A15	A14
33	AI1	$\overline{CP3}$	A1	A2	A1	A2
34	AI2	$\overline{CP2}$	A3	A4	A3	A4
35	AI3	$\overline{CP1}$	A5	A6	A5	A6
36	AI4	$\overline{CP0}$	A7	A8	A7	A8
37	AI5	$\overline{VEC}$	A9	A10	A9	A10
38	AI6	$\overline{PF}$	A11	A12	A11	A12
39	AI7	$\overline{HALT}$	A13	A14	A13	A14

*FET/REF AI0 indicates a fetch or refresh operation in 4K/16K dynamic mode. The encoded SEL0 and SEL1 lines determine which transaction is occurring.

Line AI0 specifies a fetch or refresh operation for a 4K/16K chip dynamic RAM mode. The SEL1 and SEL0 lines indicate which operation is in process as listed in Table 9.

Table 9 • DCT11 Dynamic RAM Select Line Functions

SEL1	SEL0	Transaction
L	L	FETCH
L	H	REFRESH

Control Signal Summary—A summary of the control signals is contained in Table 10.

Table 10 • DCT11 Control Signal Summary

Pin	Signal	Function
24	SEL1	Transaction select 1
25	SEL0	Transaction select 0
26	$\overline{\text{READY}}$	Idle state select
29	$\overline{\text{RAS}}$	Row address select
30	CAS	Column address select
31	PI	Priority Interrupt
27	R/$\overline{\text{WHB}}$	Read/Write high byte (16-bit) select
	$\overline{\text{RD}}$	Read (8-bit)
28	R/$\overline{\text{WLB}}$	Read/Write low byte (8-bit) select
	$\overline{\text{WT}}$	Write (8-bit)

Miscellaneous Signal Summary—A summary of the miscellaneous signals is listed in Table 11.

Table 11 • DCT11 Miscellaneous Signal Summary

Pin	Signal	Function
18	$\overline{\text{BCLR}}$	Bus clear
19	PUP	Powerup
21	COUT	Clock out
22	XTL1	Crystal input 1
23	XTL0	Crystal input 2

• Inititialization

The powerup (PUP) input has a Schmitt trigger that senses transitions from low to high and has a low-current internal pulldown device that is always enabled. When the PUP input is forced high, all DCT11 operation stops. Figure 17 shows the signal timing for the powerup operation. The assertion of the PUP line causes the $\overline{\text{BCLR}}$ signal to be asserted. The $\overline{\text{BCLR}}$ signal enables the mode register to be loaded with the configuration information. When the processor detects a high to low transition on the PUP line, the powerup sequence starts. All information in the internal registers is undefined. The information on DAL<15:00> and AI<7:0> lines is also undefined. The control and miscellaneous signals are not asserted. An external 1k 1% resistor must be connected from $\overline{\text{BCLR}}$ input to ground to assure the correct operation.

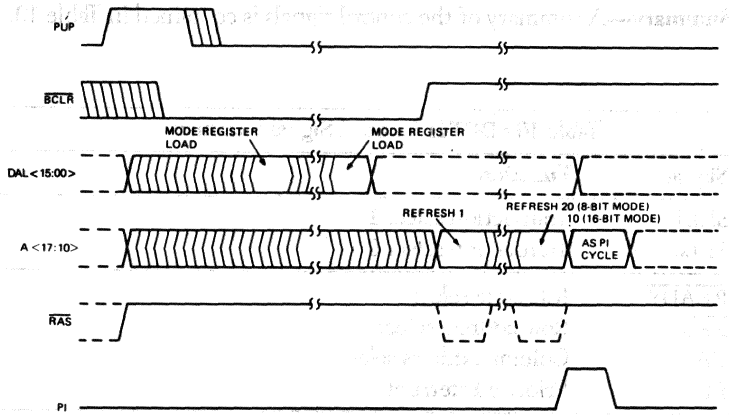


Figure 17 • DCT11 Powerup Timing Sequence

Loading the Mode Register

The DAL<15:00> provide information to change the mode register settings. Figure 18 shows the connections of the PUP line to allow the $\overline{BCLR}$ signal to load the DCT11 mode register information. During the assertion of $\overline{BCLR}$, DAL<02:00> and DAL<15:08> are connected to internal pullup circuits and are asserted unless driven low by the external signals. These lines contain the mode register information and DAL<07:03> are in a high-impedance condition at this time. The $\overline{BCLR}$ signal is asserted during the powerup sequence and when a RESET instruction is executed. The mode register will accept data during the assertion of the $\overline{BCLR}$ signal.

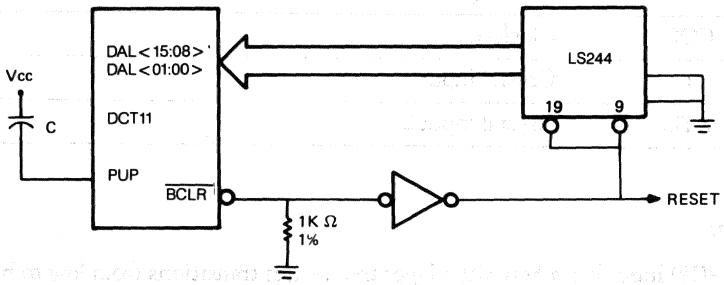


Figure 18 • DCT11 Mode Register Loading Configuration

Refresh and No Operation

The DCT11 mode information loded when the $\overline{BCLR}$ signal is negated. If dynamic RAM support has been selected, the refresh transactions will be executed to ensure the correct start for the dynamic RAMs. During 8-bit data bus mode, 20 REFRESH transactions will be executed and, during 16-bit mode, ten refresh transactions will be executed. During static mode, a NOP (No operation) bus transaction will be executed. After the refresh or NOP transactions, one ASPI transaction is performed to determine if interrupts are pending. The ASPI transaction asserts $\overline{CAS}$ and PI signals and the refresh transaction asserts only the $\overline{RAS}$ signal. No control signals are asserted during a NOP transaction while performing internal operations.

Starting Address

A read transaction to the starting address follows the refresh or NOP transaction. The first instruction fetch starts the DCT11 program operation. One of eight starting addresses can be selected by the three most significant bits of the mode register. Each address has an associated restart location. The restart address is the first location to be executed when a HALT instruction is executed or a halt interrupt is asserted.

Clock Output

The clock ouput (COUT) is a square wave controlled by bit 0 of the mode register as shown in Figure 19. Selecting constant clock mode produces a square-wave output at a one-half of the internal oscillator or XTL1 input frequency. When processor clock mode (PCM) is selected, the COUT output provides a pulse once during each microcycle. No close timing relationship exists between the edges of the COUT signal and the internal oscillator or XTL1 clock input.

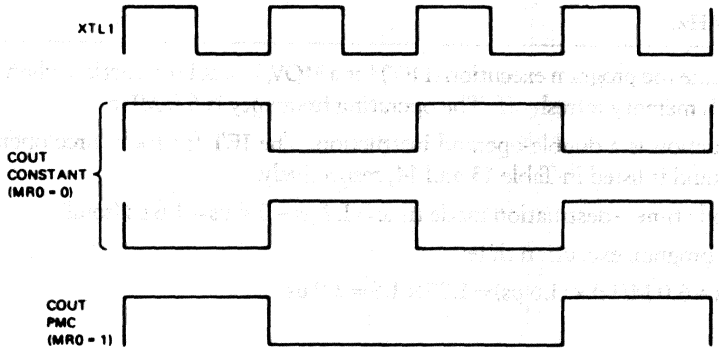


Figure 19 • DCT11 Clock Output Timing Sequence

• Performance Data

Tables 12 through 17 list the execution times for all DCT11 instructions and are used to determine the total program exection time. The tables list the minimum and maximum execution time of an instruction in 16-bit and 8-bit mode of operation and with the memory refresh function active (on) and not active (off). The possible system combinations are

- 16-bit mode with memory refresh on
- 16-bit mode with memory refresh off
- 8-bit mode with memory refresh on
- 8-bit mode with memory refresh off

With refresh on, the refresh cycle increases the execution time. The execution time of an instruction may also vary with memory refresh on. In 8-bit mode, refresh occurs every instruction cycle. In 16-bit mode, refresh occurs every other cycle. Addressing modes 5, 6, 7, I/O, and Trap also increase the time for execution.

The program execution time (PET) is calculated by totaling the average execution time of the instruction used in the program. The following information applies to Tables 12 through 17.

- All times are in microseconds.
  Add 0.4 μs for every $\overline{\text{READY}}$ pulse that occurs during a read or write transaction.
- The operating frequency is 7.5 MHz. To compute the program execution time (PET) for frequencies other than 7.5 MHz, the following equation is used.

  $PET = (7.5\ MHz \div F_{OP}) \times IET$

  where IET is instruction execution time in microseconds from the tables and F_{OP} is the operating frequency in MHz.

Example: Calculate the program execution (PET) for a MOV, R0, R1 instruction when in the static, 16-bit mode with memory refresh off. The operating frequency is 6.0 MHz.

The MOV instruction is a double-operand instruction. The IET for the source operand and the destination operand is listed in Table 13 and 14, respectively.

IET = source mode time + destination mode time = 1.2 μs + 0.4 μs = 1.6 μs (total).

To calculate the program execution time:

$PET = (7.5\ MHz \div 6.0\ MHz) \times (1.6\ \mu s) = 1.25 \times 1.6 = 2.0\ \mu s$

Table 12 • DCT11 XOR and Single-operand Instruction Execution Time

REFRESH Instructions*	Dest. Mode	16-Bit Mode ON Min.	16-Bit Mode ON Max.	16-Bit Mode OFF	8-Bit Mode ON Word Instr.	8-Bit Mode ON Byte Instr.	8-Bit Mode OFF Word Instr.	8-Bit Mode OFF Byte Instr.
CLR(B),COM(B),	0	1.60	1.73	1.6	2.53	2.53	2.4	2.4
INC(B),DEC(B),	1	2.80	2.93	2.8	5.33	3.73	5.2	3.6
NEG(B),ROR(B),	2	2.80	2.93	2.8	5.33	3.73	5.2	3.6
ROL(B),ASR(B),	3	3.60	3.73	3.6	6.93	5.33	6.8	5.2
ASL(B),SWAB,	4	3.20	3.33	3.2	5.73	4.13	5.6	4.0
ADC(B),SBC(B),	5	4.13	4.26	4.0	7.46	5.86	7.2	5.6
SXT,MFPS,	6	4.13	4.26	4.0	7.46	5.86	7.2	5.6
XOR	7	4.93	5.06	4.8	9.06	7.46	8.8	7.2
TST(B)	0	1.60	1.73	1.6	2.53	2.53	2.4	2.4
	1	2.40	2.53	2.4	4.13	3.33	4.0	3.2
	2	2.40	2.53	2.4	4.13	3.33	4.0	3.2
	3	3.20	3.33	3.2	5.73	4.93	5.6	4.8
	4	2.80	2.93	2.8	5.33	3.73	5.2	3.6
	5	3.73	3.86	3.6	6.26	5.46	6.0	5.2
	6	3.73	3.86	3.6	6.26	5.46	6.0	5.2
	7	4.53	4.66	4.4	7.86	7.06	7.6	6.8
MTPS	0	3.20	3.33	3.2	4.13	4.13	4.0	4.0
	1	4.00	4.13	4.0	4.93	4.93	4.8	4.8
	2	4.00	4.13	4.0	4.93	4.93	4.8	4.8
	3	4.80	4.93	4.8	6.53	6.53	6.4	6.4
	4	4.40	4.53	4.4	5.33	5.33	5.2	5.2
	5	5.33	5.46	5.2	7.06	7.06	6.8	6.8
	6	5.33	5.46	5.2	7.06	7.06	6.8	6.8
	7	6.13	6.26	6.0	8.66	8.66	8.4	8.4

*The XOR and single-operand instruction execution times include instructions fetch, instruction decode, operand fetch, instruction operation, and result output. In mode 0 and the TST(B) instruction, there is no output.

Table 13 • DCT11 Double-operand Instruction Source Mode Execution Time

REFRESH		16-Bit Mode ON	ON	ON	ON	OFF	8-Bit Mode ON	ON	OFF	OFF
		Dest.		Dest.						
	Src	Mode (0-4)		Mode (5-7)			Word	Byte	Word	Byte
Instructions	Mode*	Min.	Max.	Min.	Max.		Instr.	Instr.	Instr.	Instr.
MOV(B),CMP(B),	0	1.20	1.33	1.33	1.33	1.2	2.13	2.13	2.0	2.0
ADD,SUB,	1	2.00	2.13	2.13	2.13	2.0	3.73	2.93	3.6	2.8
BIT(B),BIC(B),	2	2.00	2.13	2.13	2.13	2.0	3.73	2.93	3.6	2.8
BIS(B)	3	2.80	2.93	2.93	2.93	2.8	5.33	4.53	5.2	4.4
	4	2.40	2.53	2.53	2.53	2.4	4.13	3.33	4.0	3.2
	5	3.33	3.33	3.33	3.46	3.2	5.86	5.06	5.6	4.8
	6	3.33	3.33	3.33	3.46	3.2	5.86	5.06	5.6	4.8
	7	4.13	4.13	4.13	4.26	4.0	7.46	6.66	7.2	6.4

*Source mode times include instruction fetch, instruction decode, and source operand fetch.

Table 14 • DCT11 Double-operand Instruction Destination Mode Execution Time

REFRESH		16-Bit Mode ON	ON	OFF	8-Bit Mode ON	ON	OFF	OFF
	Dest.				Word	Byte	Word	Byte
Instructions*	Mode	Min.	Max.		Instr.	Instr.	Instr.	Instr.
MOV(B),ADD,	0	0.4	0.4	0.4	0.40	0.40	0.4	0.4
SUB,BIC(B),	1	1.6	1.6	1.6	2.40	1.6	2.4	1.6
BIS(B)	2	1.6	1.6	1.6	2.40	1.6	2.4	1.6
	3	2.4	2.4	2.4	4.00	3.20	4.0	3.2
	4	2.0	2.0	2.0	2.80	2.00	2.8	2.0
	5	2.8	2.8	2.8	4.53	3.73	4.4	3.6
	6	2.8	2.8	2.8	4.53	3.73	4.4	3.6
	7	3.6	3.6	3.6	6.13	5.33	6.0	5.2
CMP(B),BIT(B)	0	0.4	0.4	0.4	0.40	0.40	0.4	0.4
	1	1.2	1.2	1.2	2.00	1.20	2.0	1.2
	2	1.2	1.2	1.2	2.00	1.20	2.0	1.2
	3	2.0	2.0	2.0	3.60	2.80	3.6	2.8
	4	1.6	1.6	1.6	2.40	1.60	2.4	1.6
	5	2.4	2.4	2.4	4.13	3.33	4.0	3.2
	6	2.4	2.4	2.4	4.13	3.33	4.0	3.2
	7	3.2	3.2	3.2	5.73	4.93	5.6	4.8

*Destination mode times include destination operand fetch, instruction operation, and result output. In destination mode 0 and the CMP(B) and BIT(B) instructions, there are no outputs.

Table 15 • DCT11 Branch, Trap, and Interrupt Instruction Execution Time

REFRESH Instructions*	Dest. Mode	16-Bit Mode ON Min.	16-Bit Mode ON Max.	16-Bit Mode OFF	8-Bit Mode ON Word Instr.	8-Bit Mode ON Byte Instr.	8-Bit Mode OFF Word Instr.	8-Bit Mode OFF Byte Instr.
BR,BNE,BEQ,BPL, BMI,BVC,BVS,BCC, BCS,BGE,BLT,BGT, BLE,BHI,BLOS, BHIS,BLO	NA	1.60	1.73	1.6	2.53	NA	2.4	NA
EMT,TRAP,BPT,IOT	NA	6.53	6.66	6.4	9.73	NA	9.6	NA
RTI	NA	3.20	3.33	3.2	4.93	NA	4.8	NA
RTT	NA	4.40	4.53	4.4	7.13	NA	7.0	NA

*1. Branch instruction execution times include instruction fetch, instruction decoding, doubling the offset, testing the conditions, and adding the offset to the PC if the conditions are met. The execution times are not affected whether or not a branch is taken.

2. Trap instruction execution times include instruction fetch, instruction decode, pushing the PSW and PC onto the stack, loading the PC with the contents of the vector location, and loading the PSW with the contents of the vector location plus two.

3. Return from Interrupt instruction execution times include instruction fetch, instruction decode, and removing the PC information and PSW from the stack.

Table 16 • DCT11 Jump and Subroutine Instruction Execution Time

REFRESH Instructions*	Dest. Mode	16-Bit Mode ON Min.	16-Bit Mode ON Max.	16-Bit Mode OFF	8-Bit Mode ON Word Instr.	8-Bit Mode ON Byte Instr.	8-Bit Mode OFF Word Instr.	8-Bit Mode OFF Byte Instr.
JMP	1	2.00	2.13	2.0	2.93	NA	2.8	NA
	2	2.40	2.53	2.4	3.33	NA	3.2	NA
	3	2.40	2.53	2.4	4.13	NA	4.0	NA
	4	2.40	2.53	2.4	3.33	NA	3.2	NA
	5	2.93	2.93	2.8	4.53	NA	4.4	NA
	6	2.93	2.93	2.8	4.53	NA	4.4	NA
	7	3.73	3.73	3.6	6.13	NA	6.0	NA
JSR	1	3.60	3.73	3.6	5.33	NA	5.2	NA
	2	4.00	4.13	4.0	5.73	NA	5.6	NA
	3	4.00	4.13	4.0	6.53	NA	6.4	NA
	4	4.00	4.13	4.0	5.73	NA	5.6	NA
	5	4.53	4.53	4.4	6.93	NA	6.8	NA
	6	4.53	4.53	4.4	6.93	NA	6.8	NA
	7	5.33	5.33	5.2	8.53	NA	8.4	NA

REFRESH Instructions*	Dest. Mode	16-Bit Mode ON Min.	16-Bit Mode ON Max.	16-Bit Mode OFF	8-Bit Mode ON Word Instr.	8-Bit Mode ON Byte Instr.	8-Bit Mode OFF Word Instr.	8-Bit Mode OFF Byte Instr.
RTS	NA	2.80	2.93	2.8	4.53	NA	4.4	NA
SOB	NA	2.40	2.53	2.4	3.33	NA	3.2	NA

*1. JMP/JSR destination mode 0 is an illegal instruction that traps to vector location 10.

2. JMP execution times include instruction fetch, instruction decode, operand fetch, and loading the PC.

3. JSR execution times include instruction decode, operand fetch, pushing the linkage register onto the stack, and loading the PC.

4. RTS execution times inlude instruction fetch, instruction decode, loading the PC, popping the stack, and loading the linkage register.

5. SOB execution times include instruction fetch, instruction decode, decrementing the count register, testing for zero, and branching, if necessary. The execution time is not affected whether or not a branch is taken.

Table 17 • DCT11 Miscellaneous and Condition Code Instruction Execution Time

REFRESH Instructions*	Dest. Mode	16-Bit Mode ON Min.	16-Bit Mode ON Max.	16-Bit Mode OFF	8-Bit Mode ON Word Instr.	8-Bit Mode ON Byte Instr.	8-Bit Mode OFF Word Instr.	8-Bit Mode OFF Byte Instr.
HALT	NA	5.73	5.86	5.6	8.4	NA	8.0	NA
WAIT	NA	1.60	1.73	1.6	2.43	NA	2.4	NA
RESET	NA	14.60	14.73	14.6	16.53	NA	16.4	NA
NOP	NA	2.40	2.53	2.4	3.33	NA	3.2	NA
CLC,CLV,CLZ,CLN, CCC,SEC,SEV,SEZ, SEN,SCC	NA	2.40	2.53	2.4	3.33	NA	3.2	NA
MFPT	NA	2.00	2.13	2.0	2.93	NA	2.8	NA

*1. The HALT instruction execution time includes instruction fetch, instruction decode, writing the PC information and PSW onto the stack, loading the PSW with 340, and loading the PC with the restart address.

2. The WAIT instruction execution time includes instruction fetch, instruction decode, pulsing PI to sample the interrupt lines, and doing a REFRESH cycle if refresh is on. If no interrupt lines were sensed by the DCT11 to be asserted during the PI pulse, the WAIT instruction will cycle in a 1.2μs loop pulsing PI. If refresh in on, the loop will be 1.33μs maximum. The looping will continue until an interrupt line is asserted and sensed by the DCT11.

3. The RESET execution time includes instruction fetch, instruction decode, the assertion of $\overline{BCLR}$, and the writing of DAL<15:00> into the mode register.
4. The NOP execution time includes instruction fetch, instruction decode, and idle time.
5. Condition code (CC) instruction execution time includes instruction fetch, instruction decode, and the setting or resetting of the appropriate status flags in the PSW.

Interrupt Latency

The interrupt latency is measured from the time the interrupt request (IRQ) is asserted on the AI inputs until the time that the DCT11 is ready to fetch the first instruction in the service routine. Figure 20 shows the interrupt latency parameters.

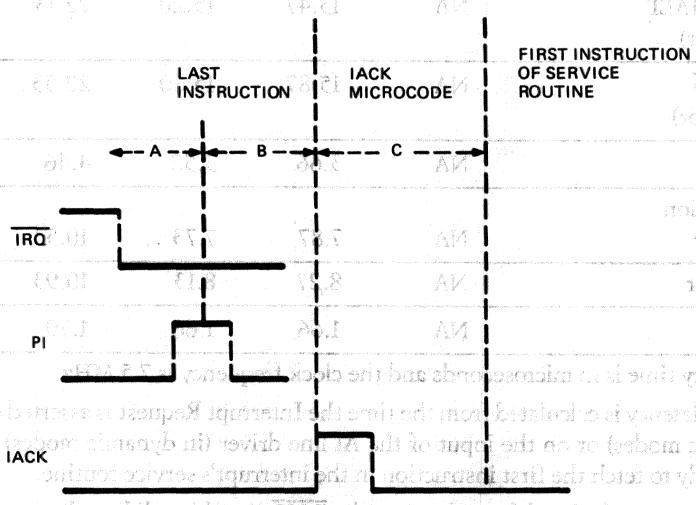


Figure 20 • DCT11 Interrupt Latency Measurements

The processor continues to execute instructions until the request is latched by the assertion of the PI signal which can occur during the instruction following the request. The current instruction is then completed and the IACK microcode is executed to acknowledge the interrupt request and the following operations are performed.

- Arbitrate priority.
- Issue the IACK signal.
- Generate vector or receive external vector.
- Store processor status and program counter values.
- Load new processor status and program counter values.

The interrupt request (IRQ) is assumed to occur during a PDP-11 Jump to Subroutine (JSR) instruction (mode 2 or 4) or during an emulator trap (EMT) sequence which provides the maximum delay for an IRQ. The maximum latencies times assume that the DCT11 is in the standard microcycle mode and the $\overline{\text{READY}}$ signal is not asserted during the transaction. Table 18 lists the maximum latency times for the 16- and 8-bit modes, during dynamic and static operation.

Table 18 • DCT11 8- and 16-bit Mode Latency Time (7.5 MHz)

	16-Bit Mode			8-Bit Mode	
Active Inputs*	**Dest. Mode**	**Dynamic**	**Static**	**Dynamic**	**Static**
$\overline{\text{CPO}}$-$\overline{\text{CP3}}$,$\overline{\text{PF}}$,$\overline{\text{HALT}}$ (Internal vector)	NA	15.47	15.20	22.13	21.60
$\overline{\text{VEC}}$,$\overline{\text{CPO}}$-$\overline{\text{CP3}}$ (External vector)	NA	15.87	15.60	22.53	22.00
$\overline{\text{DMR}}$	NA	3.66	3.52	4.46	4.32
WAIT Instruction Internal vector	NA	7.87	7.73	10.53	10.13
External vector	NA	8.27	8.13	10.93	10.53
$\overline{\text{DMR}}$	NA	1.66	1.66	1.79	1.66

*1. The latency time is in microseconds and the clock frequency is 7.5 MHz.

2. Interrupt latency is calculated from the time the Interrupt Request is asserted either on the AI lines (in static modes) or on the input of the AI line driver (in dynamic modes) to the time the DCT11 is ready to fetch the first instruction in the interrupt's service routine.

3. DMG latency is calculated from the time the $\overline{\text{DMR}}$ signal is valid on the input of the AI line driver to the time the DCT11 asserts the DMG signal.

4. The WAIT instruction latencies are the maximum encountered in the instruction's execution state and do not include the instruction fetch or the instruction decode.

5. The worst-case times refer to IRQ occurring during a JSR (mode 2 or 4) EMT sequence.

6. The worst-case times refer to $\overline{\text{DMR}}$ occurring during a MTPS (mode 0) instruction.

7. The timings assume that the DCT11 is not in long bus cycle and no ready slips occur.

DMA Latency

The maximum time between a DMA request and grant depends on the DCT11 mode. The maximum time occurs during a Move byte to Processor Status (*MTPS*) instruction in mode 0 and is measured from the time that the $\overline{\text{DMR}}$ signal is valid on line AI0 until the DCT11 asserts the DMG signal. The maximum latencies assume that the DCT11 is in the standard microcycle mode and the $\overline{\text{READY}}$ signal is not asserted during a transaction. The maximum DMR latencies are listed in Table 18.

• Instruction Set

Refer to Appendix B for a complete list of the DCT11 microprocessor instruction set.

• Specifications

The mechanical, electrical, and enviromental characteristics and specifications for the DCT11 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified differently in the tables. Refer to Digital specification A-PS-2100002-GS for general information on the integrated circuits.

- Power supply voltage (V_{CC}): 5.0 ±5%
- Temperature range (T_A): 0°C to 70°C
- Relative humidity: 10% to 90% (noncondensing)

Mechanical Configuration

The physical dimensions of the DCT11 40-pin DIP are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of this device.

- Pin voltage: –0.5 V to 7.0 V
- Power dissipation: 1.1 W (maximum) at 0°C
- Ambient operating temperature: 0°C to 70°C
- Storage temperature: –55°C to 125°C
- Relative humidity: 10% to 95% (noncondensing)

Recommend Operating Conditions

- Supply voltage (V_{CC}): 5.0 volts ±5%
- Supply current (I_{CC}): 190 mA (maximum)

dc Electrical Characteristics

The dc electrical characteristics of the DCT11 for the operating voltage and temperature ranges specified and with $V_{SS}=0$ V are listed in Table 19. Refer to Appendix C for test circuit configurations referenced in the tables.

Table 19 • DCT11 dc Input and Output Parameters

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units	Test Circuit
High-level input voltage on $\overline{\text{READY}}$, DAL<15:00>,AI<7:0>	V_{IH}		2.0	V_{CC}	V	C1,C2
High-level input voltage for PUP	V_{IH}		2.2	V_{CC}	V	C1,C2
Low-level input voltage on $\overline{\text{READY}}$, DAL<15:00>,AI<7:0>	V_{IL}		−0.5*	0.8	V	C1,C2
Low-level input voltage for PUP	V_{IL}		−0.5*	0.6	V	C1,C2
High-level output voltage for DAL<15:00>, COUT, PI, SEL1, SEL0	V_{OH}	$I_{OH}=-700\ \mu A$	2.4	—	V	C1
High-level output voltage for AI<7:0>	V_{OHA}	$I_{OH}=-700\ \mu A$	2.6	—	V	C1
High-level output voltage for $\overline{\text{BCLR}}$	V_{OHB}	$I_{OH}=-700\ \mu A$ terminated with 1kΩ resistor to V_{SS}	2.2	—	V	C1
High-level output voltage for $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, R/$\overline{\text{WLB}}$, R/$\overline{\text{WHB}}$	V_{OHC}	$I_{OH}=-700\ \mu A$	2.8	—	V	C1
Low-level output voltage for DAL<15:00>,AI<7:0>, COUT, PI, SEL1, SEL0, $\overline{\text{BCLR}}$, $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, R/$\overline{\text{WLB}}$, R/$\overline{\text{WHB}}$	V_{OL}	$I_{OL}=3.2$ mA	0	0.4	V	C2
Input capacitance for $\overline{\text{READY}}$, DAL<15:00>,AI<7:0>	C_{in}		—	10	pF	
Output capacitance for three-state load on DAL<15:00>,AI<7:0>, COUT, PI, SEL1, SEL0, $\overline{\text{BCLR}}$, $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, R/$\overline{\text{WHB}}$, R/$\overline{\text{WLB}}$	C_{out}		—	20	pF	

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units	Test Circuit
Low-input three-state leakage current on DAL<15:00>	I_{IL}	$V_{in}=0.4$ V	—	−50	µA	C5
High-input three-state leakage current on DAL<15:00>	I_{IL}	$V_{in}=V_{CC}$ max.	—	10	µA	C5
Minimum input current for internal pullups on AI<7:0>,$\overline{READY}$ DAL<02:00>,DAL<15:07>	I_{IH}	$V_{in}=2.4$ V	−0.1	−1.0	mA	C4
Maximum input current for internal pullups on AI<7:0>,$\overline{READY}$ DAL<02:00>,DAL<15:07>	I_{IH}	$V_{in}=0.4$ V	−0.1	—	mA	C4
Power supply current on V_{CC}	I_{CC}	static	—	190	mA	C7
High-input current on XTL1	I_{XLIH}	$2.4<V_{in}<V_{CC}$, XTL0 grounded	—	700	µA	
Low-input current on XTL1	I_{XLIL}	$-0.5<V_{in}<0.6$V, XTL0 grounded	—	−6.4	mA	

*−0.5 volts on input pins allows for ringing on unterminated lines.

ac Electrical Characteristics

The input and output signal timing for the DCT11 is shown in Figures 21 through 25. Most of the timing parameters listed in the tables apply to both the 7.5 MHz and 10 MHz versions of the DCT11. Parameters that apply only to the 10 MHz version are indicated in the tables. Table 20 lists the timing parameters for the clock-input waveform shown in Figure 21. Table 21 lists the parameters for the powerup and reset timing shown in Figure 22. Table 22 lists the parameters for the refresh and interrupt acknowledge timing shown in Figure 23. The DMA timing parameters shown in Figure 24 are listed in Table 23. The timing parameters for the read, write, and ready signal, shown in Figure 25, are listed in Table 24.

The following notes apply to Figures 22 through 25.

- Timings are measured at the following output voltages:

$\overline{RAS}$,$\overline{CAS}$,R/$\overline{WLB}$,R/$\overline{WHB}$	$V_{OL}=0.8$ V	$V_{OH}=2.4$ V
AI<7:0>	$V_{OL}=0.8$ V	$V_{OH}=2.2$ V
All others	$V_{OL}=0.8$ V	$V_{OH}=2.0$ V

- Output timings are measured using a purely capacitive load of 80 pF.

- Definitions and abbreviations:
  - —T = 1/XTAL frequency
  - —(l.e.) = leading edge, (t.e.) = trailing edge
  - —F_{OP} = XTL1, XTL0 internal oscillator operating frequency. If an external TTL clock is used, F_{OP} minimum is 0 Hz (i.e., dc). Internal data will not be lost if the input clock is stopped at either a high or low level. The leading (low-to-high) edge of the clock starts the internal timing to produce an output signal ($\overline{RAS}$, $\overline{CAS}$, PI, etc.). The trailing (high-to-low) edge causes the signal to be asserted at the pin of the DCT11. The delay from the edge of the input clock to the output signal is not defined.

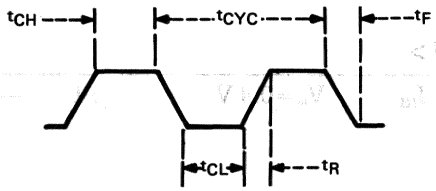


Figure 21 • DCT11 Clock Input Waveform

Table 20 • DCT11 Clock Input Timing Parameters

Symbol	Description	7.5 MHz Min.	7.5 MHz Max.	10 MHz Min.	10 MHz Max.	Units
(F_{OP})	Operating Frequency	0.1	7.5	0.1	10	MHz
(t_{CYC})	Cycle Time	133	—	100		ns
t_{CH} and t_{CL}	Pulse Width	20	—	20		ns
t_R and t_F	Rise and Fall	—	10	—	10	ns

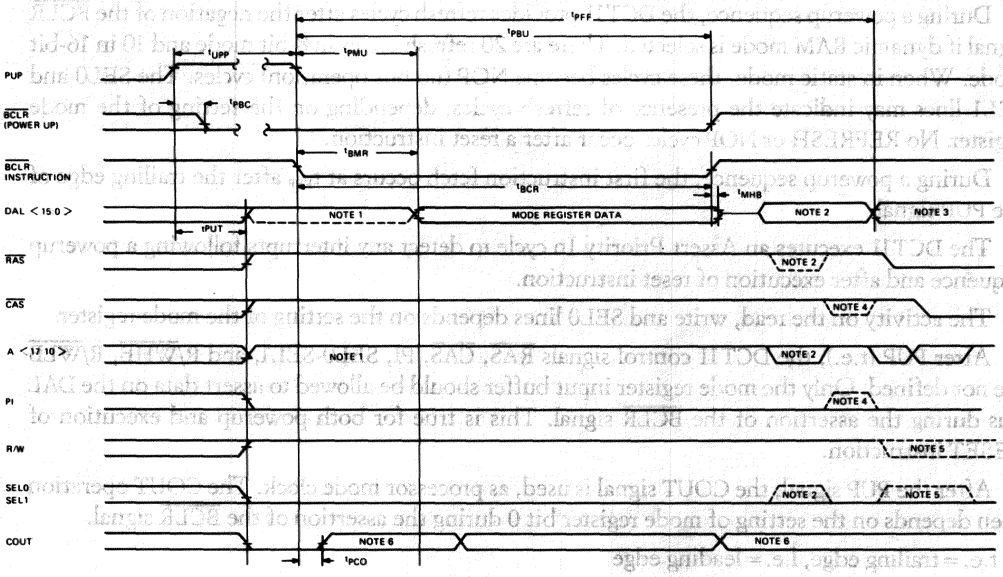


Figure 22 • DCT11 Powerup and Reset Signal Timing

Table 21 • DCT11 Powerup and Reset Timing Parameters

Symbol	Test Conditions*	Min.	Max.
t_{PFF}	PUP(t.e.)to first instruction fetch	295T	315T
t_{PBU}	PUP(t.e.)to $\overline{BCLR}$(t.e)	99T	100T
t_{UPP}	PUP with	100 µs	
t_{PMU}	PUP(t.e.)to DAL, mode data, valid input	—	74T
t_{PBC}	PUP(l.e.)to $\overline{BCLR}$(l.e.)	—	100 ns
t_{BMR}	$\overline{BCLR}$(l.e.) to DAL, mode data valid, PDP-11 RESET instruction	—	74T
t_{BCR}	$\overline{BCLR}$ width during a PDP-11 RESET instruction	84T	—
t_{MHB}	$\overline{BCLR}$(t.e.) to DAL, mode data hold	0 ns	—
t_{PUT}	PUP(l.e.) to DAL, input float	—	250 ns
t_{PCO}	PUP(t.e.) to COUT(l.e.)	—	T+60 ns

*1. Low-current pullup circuits are provided by the DCT11 on DAL<02:00>, DAL<15:08> and AI<7:0> lines while the $\overline{BCLR}$ line is a ,serted. This occurs both during powerup and during the execution of a RESET instruction. (DAL2 is reserved for use by Digital Equipment Corporation.)

2. During a powerup sequence, the DCT11 provides refresh cycles after the negation of the $\overline{BCLR}$ signal if dynamic RAM mode is selected. There are 20 refresh cycles in 8-bit mode and 10 in 16-bit mode. When in static mode, these cycles become NOP (no bus operation) cycles. The SEL0 and SEL1 lines may indicate the presence of refresh cycles, depending on the setting of the mode register. No REFRESH or NOP cycles occur after a reset instruction.

3. During a powerup sequence, the first instruction fetch occurs at t_{PFF} after the trailing edge of the PUP signal

4. The DCT11 executes an Assert Priority In cycle to detect any interrupts following a powerup sequence and after execution of reset instruction.

5. The activity on the read, write and SEL0 lines depends on the setting of the mode register.

6. After PUP (t.e.), the DCT11 control signals $\overline{RAS}$, $\overline{CAS}$, PI, SEL0-SEL1, and R/$\overline{WHB}$, R/$\overline{WLB}$ are not defined. Only the mode register input buffer should be allowed to assert data on the DAL bus during the assertion of the $\overline{BCLR}$ signal. This is true for both powerup and execution of RESET instruction.

7. After the PUP signal, the COUT signal is used, as processor mode clock. The COUT operation then depends on the setting of mode register bit 0 during the assertion of the $\overline{BCLR}$ signal.

8. t.e. = trailing edge, l.e. = leading edge

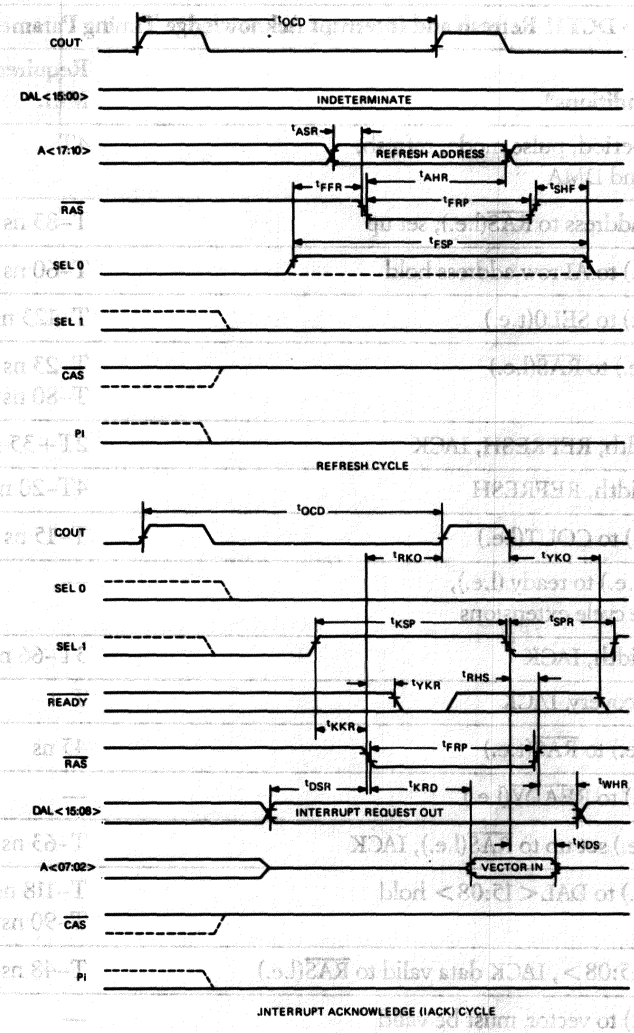


Figure 23 ▪ DCT11 Refresh and Interrupt Acknowledge Signal Timing

Table 22 • DCT11 Refresh and Interrupt Acknowledge Timing Parameters

Symbol	Test Conditions*	Requirements Min.	Max.
t_{OCD}	COUT period, pulse mode, refresh, IACK, and DMA	4T	—
t_{ASR}	AI row address to $\overline{RAS}$(l.e.), set up	T–83 ns	—
t_{AHR}	$\overline{RAS}$(t.e.) to AI row address hold	T–60 ns	—
t_{SHF}	$\overline{RAS}$(t.e.) to SEL0(t.e.)	T–123 ns	—
t_{FFR}	SEL0(l.e.) to $\overline{RAS}$(l.e.)	T–23 ns T–80 ns†	— —
t_{FRP}	$\overline{RAS}$ width, REFRESH, IACK	2T+35 ns	—
t_{FSP}	SEL0 width, REFRESH	4T–20 ns	—
t_{RKO}	$\overline{RAS}$(l.e.) to COUT(l.e.)	T–15 ns	—
t_{YKO}	COUT(t.e.) to ready (l.e.), multiple cycle extensions	—	2T–127 ns
t_{KSP}	SEL1 width, IACK	3T–66 ns	—
t_{SPR}	SEL1 recovery, IACK	T	—
t_{RHS}	SEL1(t.e.) to $\overline{RAS}$(t.e.)	45 ns	—
t_{YKR}	$\overline{RAS}$(l.e.) to $\overline{READY}$(l.e.)	—	T–103 ns
t_{KKR}	SEL1(l.e.) set up to $\overline{RAS}$(l.e.), IACK	T–63 ns	—
t_{WHR}	$\overline{RAS}$(t.e.) to DAL<15:08> hold	T–118 ns T–90 ns†	— —
t_{DSR}	DAL<15:08>, IACK data valid to $\overline{RAS}$(l.e.)	T–48 ns	
t_{KRD}	$\overline{RAS}$(l.e.) to vector, must be valid on DAL<07:02>	— —	2T–148 ns T–25 ns†
t_{KDS}	SEL1(t.e.) to DAL<07:02> vector hold	0 ns	—

*1. Assertion of the SEL0 line depends on mode selection of 4K/16K.

2. Add T(ns) if in long microcycle mode. If $\overline{READY}$ slips are initiated, add H T(ns) where $T = 1/F_{OP}$
H = number of $\overline{READY}$ pulses multiplied by 3 for standard microcycle mode, multiplied by 4 for long microcycle mode.

3. Violation of t_{YKR} will cause unpredictable results.

4 t.e. = trailing edge, l.e. = leading edge

†10 MHz version only

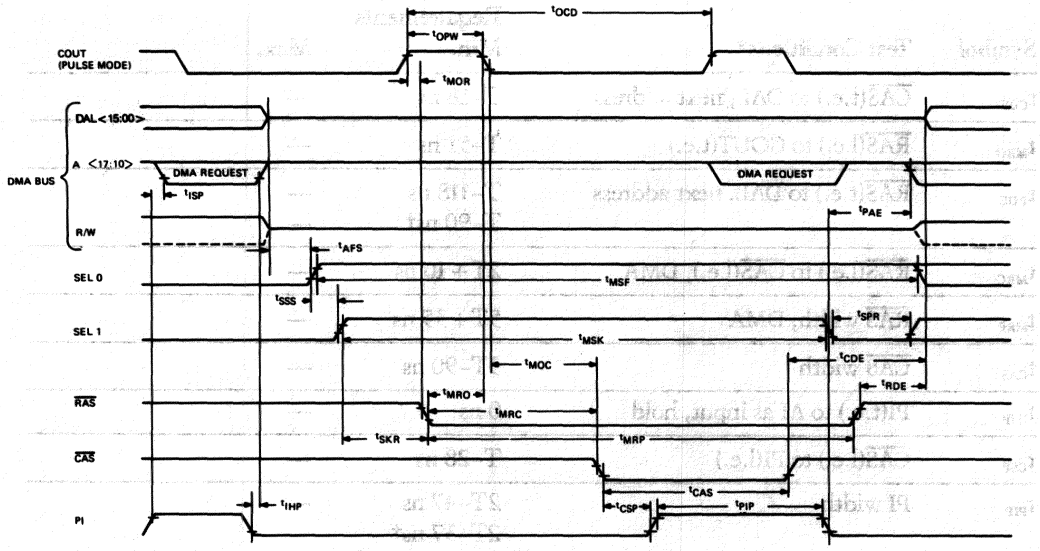


Figure 24 • DCT11 DMA Signal Timing

Table 23 • DCT11 DMA Timing Sequence Parameters

Symbol	Test Conditions*	Requirements Min.	Max.
t_{OCD}	COUT period, pulse mode, refresh, IACK, and DMA	4T	—
t_{OPW}	COUT width when high	T–33 ns	—
t_{MOR}	$\overline{RAS}$(l.e.) to COUT(l.e.)	— —	10 ns 0 ns†
t_{ISP}	PI(l.e.) to AI as input, data must be valid	—	2T–167 ns
t_{PAE}	PI(t.e.) to AI, next address	T–40 ns	—
t_{SKR}	SEL1(l.e.) $\overline{RAS}$(l.e.)	2T–63 ns	—
t_{AFS}	DMA bus(DAL,AI,R/W) disable to SEL0(l.e.)	0 ns	—
t_{MSF}	SEL0 width, DMA	8T–38 ns	—
t_{SSS}	SEL0(l.e.) to SEL1(l.e.)	0 ns	—
t_{SPR}	SEL1 recovery, IACK, DMA	T	—
t_{MSK}	SEL1 width, DMA	7T–68 ns	—
t_{MOC}	COUT(t.e.) to CAS(l.e.)	T + 10 ns	—

Symbol	Test Conditions*	Requirements Min.	Max.
t_{CDE}	$\overline{CAS}$(t.e.) to DAL, next address	T–18 ns	—
t_{MRO}	$\overline{RAS}$(l.e.) to COUT(t.e.)	T–51 ns	—
t_{RDE}	$\overline{RAS}$(t.e.) to DAL, next address	T–118 ns T–90 ns†	— —
t_{MRC}	$\overline{RAS}$(l.e.) to $\overline{CAS}$(l.e.), DMA	2T + 10 ns	—
t_{MRP}	$\overline{RAS}$ width, DMA	5T + 35 ns	—
t_{CAS}	$\overline{CAS}$ width	3T–90 ns	—
t_{IHP}	PI(t.e.) to AI as input, hold	0 ns	—
t_{CSP}	$\overline{CAS}$(l.e.) to PI(l.e.)	T–28 ns	—
t_{PIP}	PI width	2T–47 ns 2T–37 ns†	— —

*1. Add T(ns) if in long microcycle mode. If the $\overline{READY}$ signal slips are initiated, add H T(ns) where
$T = 1/F_{OP}$
H = number of $\overline{READY}$ pulses multiplied by 3 for standard microcycle mode and multiplied by 4 for long microcycle mode

2. Add 4T for each $\overline{READY}$ pulse.

3. Add 8T for each additional consecutive DMA cycle.

4. The DMA device asserts the DAL<15:00> line, AI<7:0> lines, and read and write signals. The DMA device is responsible for preventing three-state conflicts and for allowing for precharge times required by their external circuits.

5. t.e. = trailing edge, l.e. = leading edge

†10 MHz version only

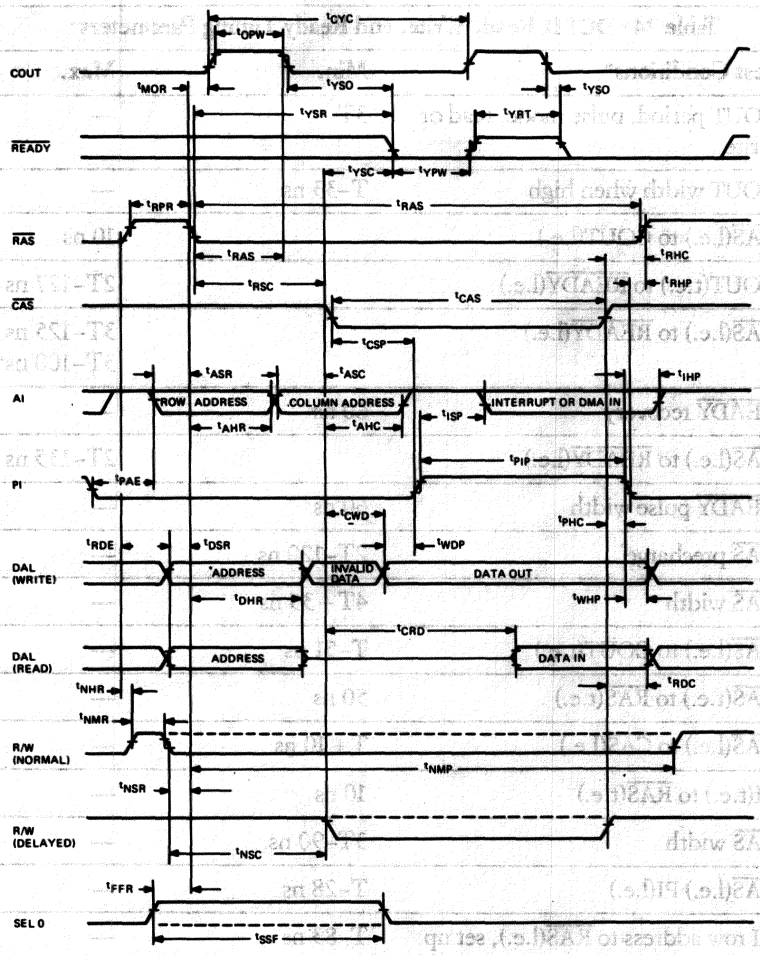


Figure 25 ▪ DCT11 Read, Write, and Ready Signal Timing

Table 24 • DCT11 Read, Write, and Ready Timing Parameters

Symbol	Test Conditions*	Min.	Max.
t_{OCP}	COUT period, pulse mode, read or write	3T	—
t_{OPW}	COUT width when high	T–33 ns	—
t_{MOR}	$\overline{RAS}$(l.e.) to COUT(l.e.)		10 ns
t_{YSO}	COUT(t.e.) to $\overline{READY}$(l.e.)		2T–127 ns
t_{YSR}	$\overline{RAS}$(l.e.) to $\overline{READY}$(l.e.)		3T–125 ns 3T–100 ns†
t_{YRT}	$\overline{READY}$ recovery	60 ns	—
t_{YSC}	$\overline{CAS}$(l.e.) to $\overline{READY}$(l.e.)		2T–135 ns
t_{YPW}	$\overline{READY}$ pulse width	60 ns	—
t_{RPR}	$\overline{RAS}$ precharge	2T–120 ns	—
t_{RAS}	$\overline{RAS}$ width	4T + 35 ns	—
t_{MRO}	$\overline{RAS}$(l.e.) to COUT(t.e.)	T–51 ns	—
t_{RHC}	$\overline{CAS}$(t.e.) to $\overline{RAS}$(t.e.)	50 ns	—
t_{RSC}	$\overline{RAS}$(l.e.) to $\overline{CAS}$(l.e.)	T + 10 ns	—
t_{RHP}	PI(t.e.) to $\overline{RAS}$(t.e.)	10 ns	—
t_{CAS}	$\overline{CAS}$ width	3T–90 ns	—
t_{CSP}	$\overline{CAS}$(l.e.) PI(l.e.)	T–28 ns	—
t_{ASR}	AI row address to $\overline{RAS}$(l.e.), set up	T–83 ns	—
t_{ASC}	AI column address to $\overline{CAS}$(l.e.), set up	12 ns	—
t_{IHP}	PI(t.e.) to AI as input, hold	0 ns	—
t_{AHR}	$\overline{RAS}$(l.e.) to AI row address hold	T–60 ns	—
t_{AHC}	$\overline{CAS}$(l.e.) to AI column address hold	T–53 ns	—
t_{ISP}	PI(l.e.) to AI as input, data must be valid	— —	2T–167 ns T–90 ns†
t_{PIP}	PI width	2T–47 ns 2T–37 ns †	
t_{PAE}	PI(t.e.) to AI, next address	T–40 ns	—
t_{CWD}	$\overline{CAS}$(l.e.) to DAL data valid, write cycle	80 ns	—

Symbol	Test Conditions*	Min.	Max.
t_{PHC}	$\overline{CAS}$(t.e.) to PI(t.e.)	10 ns	—
t_{RDE}	$\overline{RAS}$(t.e.) to DAL, next address valid	T–118 ns	—
t_{DSR}	DAL as address to $\overline{RAS}$(l.e.), set up	T–48 ns	—
t_{WDP}	DAL write data to PI(l.e.), set up	T–83 ns	—
t_{DHR}	$\overline{RAS}$(l.e.) to DAL, address hold	T–12 ns	—
t_{WHP}	PI(t.e.) to DAL, write data hold	T–88 ns	—
t_{CRD}	$\overline{CAS}$(l.e.) to DAL, read data must be valid	3T–180 ns T–10 ns†	—
t_{NHR}	$\overline{RAS}$(t.e.) to standard microcycle mode, R/W hold	T–108 ns T–80 ns†	—
t_{RDC}	$\overline{CAS}$(t.e.) to DAL, read data hold	0 ns	—
t_{NMR}	R/W recovery, standard microcycle mode	0 ns	—
t_{NMP}	R/W width, standard microcycle mode	6T–66 ns	—
t_{NSR}	R/W(l.e.) set up to $\overline{RAS}$(l.e.), standard mode	T–78 ns	—
t_{FFR}	SEL0(l.e.) to $\overline{RAS}$(l.e.), instruction FETCH	T–23 ns	—
t_{NSC}	R/W(l.e.) to $\overline{CAS}$(l.e.), standard microcycle mode	2T–37 ns	—
t_{SSF}	SEL0 width, instruction FETCH	3T–38 ns 3T–50 ns†	— —

*1. Time t_{OCP} is 3T if in standard microcycle mode and 4T if in long microcycle mode.

2. Add T(ns) if in long microcycle mode. If the $\overline{READY}$ signal slips are initiated, add H T, where $T = 1/F_{OP}$
H = number of $\overline{READY}$ pulses multiplied by 3 for standard microcycle mode, multiplied by 4 for long microcycle mode and DMA.

3. The $\overline{READY}$ signal is an edge-triggered input that is activated by asserting a low on its pin. When the $\overline{READY}$ signal is asserted before the leading edge of the $\overline{RAS}$ signal, it is internally activated by the leading edge of the $\overline{RAS}$ signal.

4. A violation of t_{YSR} will cause unpredictable results.

5. This timing parameter applies only to cases where multiple ready pulses are required, i.e., multiple cycle slips.

6. If in delayed mode, the read and write lines have the same timing as the $\overline{CAS}$ signal.

7. SEL0 is asserted to indicate an instruction fetch is in progress. The use of SEL0 as the fetch indicator is affected by DCT11 mode selection.

8. *t.e. = trailing edge, l.e. = leading edge

†10 MHz version only

Interfacing Techniques

This section describes typical circuit configurations that may be used to interface the DCT11. These examples should not be construed as specifications and recommendations.

Powerup Circuit

The circuit and voltage waveform for the powerup function is shown in Figure 26. The capacitor value must be greater than 0.04 μF and is determined by the following:

$C = 0.05\ t_R$(ms) where t_R is the rise time of the V_{CC} supply voltage

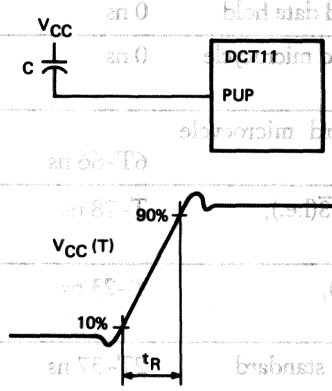


Figure 26 • DCT11 Powerup Circuit and Supply Voltage Waveform

Mode Register Loading

Figure 27 shows a typical circuit used to load the mode register. The $\overline{BCLR}$ line is asserted during the execution of a RESET instruction and is used to transfer the information from the LS244 IC to the mode register on lines DAL<01:00> and DAL<15:08>. The input lines require an input only when a low is required because the DCT11 provides pullup circuits on these lines when the $\overline{BCLR}$ signal is asserted.

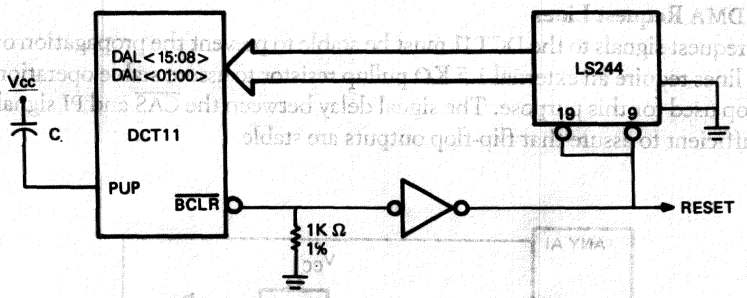
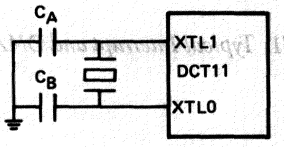


Figure 27 • DCT11 Typical Mode Register Load Circuit

Clock Circuits

The DCT11 clock circuit is controlled externally by a crystal or by a TTL oscillator as shown in Figure 28.



Crystal Clock

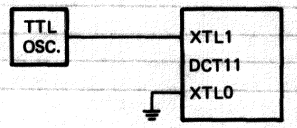


Figure 28 • DCT11 Typical Clock Input Circuit

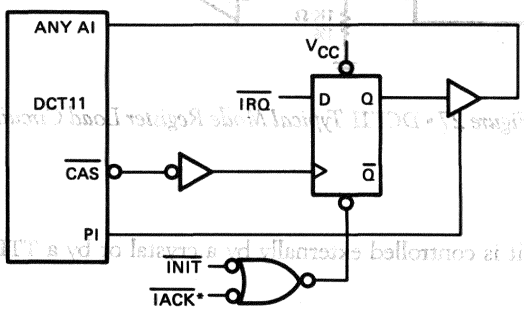
For a crystal-controlled clock, the capacitors are mica type with a nominal capacitive tolerance of ± 10% and the values are determined by the following:

$$C_A = 500\ pF \div fMHz,\ C_B = 200\ pF \div fMHz$$

The TTL input must conform to the clock input specifications shown in Figure 21. The TTL input must conform to the signal specifications listed in Table 20 and shown in Figure 28.

Interrupt and DMA Request Lines

The interrupt request signals to the DCT11 must be stable to prevent the propagation of undefined states. The AI lines require an external 1.5 KΩ pullup resistor to assure stable operation. Figure 29 shows a flip-flop used for this purpose. The signal delay between the $\overline{CAS}$ and PI signals (105 ns at 7.5 MHz) is sufficient to assure that flip-flop outputs are stable.

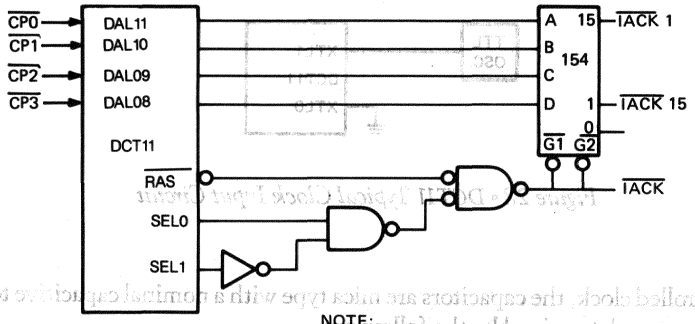


$*\overline{IACK} = \overline{(\overline{SEL\ 0} \bullet SEL\ 1)} + \overline{(RAS)}$

Figure 29 • DCT11 Typical Interrupt and DMA Request Circuit

IACK Information Decoding

Figure 30 shows a typical circuit used to decode the information from 15 interrupt devices.



NOTE:
$\overline{IACK}$ 1 CORRESPONDS TO ALL CPs ASSERTED AT IRQ TIME (I.E., INTERNAL VECTOR 140).
$\overline{IACK}$ 15 CORRESPONDS TO ONLY CP0 ASSERTED (I.E., INTERRUPT VECTOR 70).

Figure 30 • DCT11 Typical IACK Information Decoding Circuit

External Vectors

Figure 31 shows a typical circuit that enables the DCT11 to receive external vector information from a device on lines DAL<07:02> during an IACK transaction. The information is transferred when the $\overline{VEC}$ signal on line AI5 is asserted.

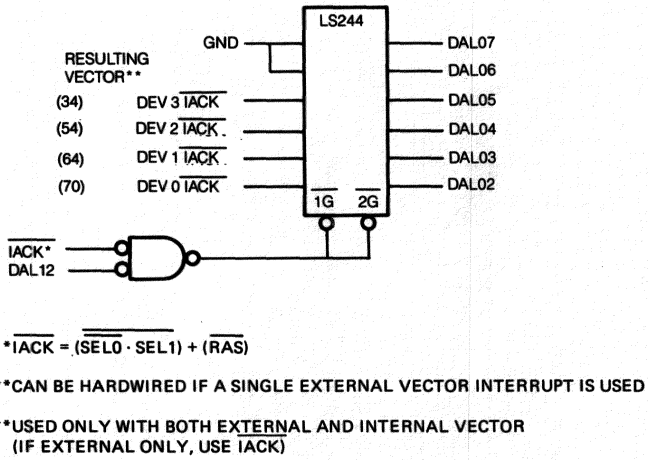


Figure 31 • DCT11 Typical IACK External Vector Circuit

DMA Request Interface

DMA requests to the DCT11 can be generated by the circuit shown in Figure 32. The flip-flop is set by the peripheral DMA request when the $\overline{CAS}$ signal is asserted provided that the word count overflow signal is not asserted. When the PI line is asserted, the request A10 is gated to the DCT11 on line AI0. With the $\overline{CAS}$ signal asserted, the assertion of the transaction complete ($\overline{TC}$) signal clears the flip-flop. The reset input during the powerup sequence will also clear the flip-flop.

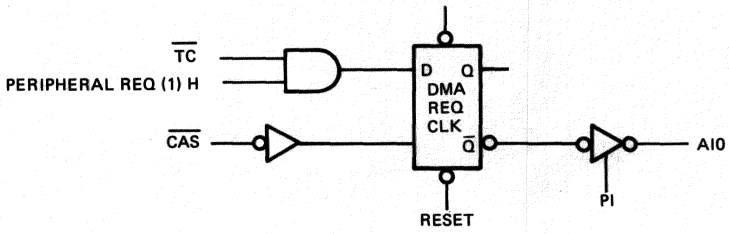


Figure 32 • DCT11 Typical DMA Interface Request Circuit

External Vectors

Figure 31 shows a typical circuit that enables the DCT11 to receive external vector information from a device on lines DAI <0/02> during an IACK transaction. The information is transferred when it is VTC signal on line AI5 is asserted.

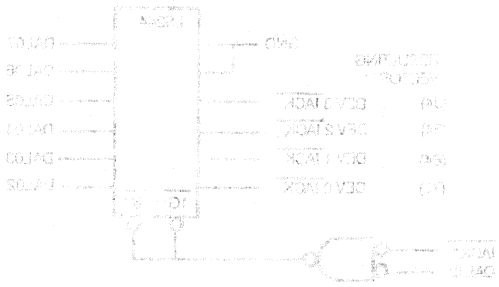


Figure 31 - DCT11 Typical IACK External Vector Circuit

DMA Request Interface

DMA requests to the DCT11 can be generated by the circuit shown in Figure 32. The flip-flop is set by the peripheral DMA request when the $\overline{DAI}$ signal is asserted indicating that the word count overflow signal is not asserted. When the P line is asserted, the interrupt AI is gated to the DCT11 on line AI0. With the $\overline{DIN}$ signal asserted, the assertion of the transaction completion ($\overline{TC}$) signal clears the flip-flop. The reset input during the power up sequence will also clear the flip-flop.

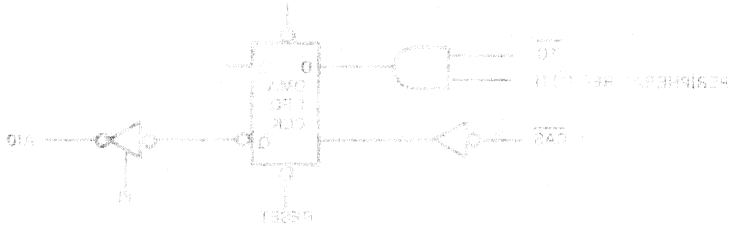


Figure 32 - DCT11 Typical DMA Interface Support Circuit

• Section 2—Video Processors and Controllers

Digital's new generation of video chips can be used for terminals and display systems and include new display features for larger screen sizes, varable pitch fonts, rectangular screen regions, and bit-map graphics generation.

78680 Video Processor—The *78680* video processor (VIPER) is a 68-pin HMOS cerquad device that is used with the 78690 video control to provide high-speed parallel processing of video data including display refresh, window scrolling, and screen updates. It transfers data into and out of the bit-map memory planes according to the instructions issued by the 78690 video controller.

78690 Video Control—The 78690 video control (ADDER) is an 84-pin ZMOS cerquad device used with the 78660 video processor to provide scan timing, system status generation, memory address generation for a video display refresh, and scroll and update operations.

DC503 Programmable Videodisplay Cursor Logic—The DC503 is a 44-pin cerquad device used to display a cursor font on the CRT. The shape of the cursor is programmed into the DC503 thereby enabling the selection of various character shapes or icons to be used as the cursor.

Section 2 — Video Processors and Controllers

Figure 2 now provides a local video chip can be used for terminals and display systems and include non-display features, as logo animation, variable-pitch fonts, resizing, hardware cursor, and bit-mapping.

7820 Video Processor — The 7820 is a proprietary VDEP is a 68-pin LMOS coprocessor device that is used with the 7850 video control to provide high-speed parallel processing of video data including display and pan, vertical scrolling and screen updates. It transfers data in and out of the bit-map memory at rates according to the instructions issued by the 7850 video controller.

7850 Video Controller — The 7850 video controller (ACDP2) is an 84-pin LMOS coprocessor device used with the 7820 video processor to provide timing, system data generation, memory address generation for video display, and serial and parallel operations.

DC 07 Programmable Video Display Chip Logic — The DC 708 is a 14-pin custom device used to drive a cursor for an LCD. The shape of the cursor is programmed into the DC708 thereby enabling the selection of various alternate shapes or fonts to be used as the cursor.

Features

- Two FIFO buffers for screen refresh and scrolling
- Four-bit wide video serial output
- Background fill on scroll
- Barrel shifter to align and scroll data
- Two mask registers to window write destination data
- 16 logic functions to combine source and destination data
- Two control store RAMs for source 1, source 2 and destination operations
- Data transfer in either X- or Z-mode
  —X-mode, one word = 16 pixels per plane
  —Z mode, one word = 16 planes per pixel

Description

The 78680 video processor (VIPER) is a data-path chip that is used with the 78690 video control (Adder) chip to implement a high-performance, bit-map graphics system using a color or monochrome display. Figure 1 is a block diagram of the video processor.

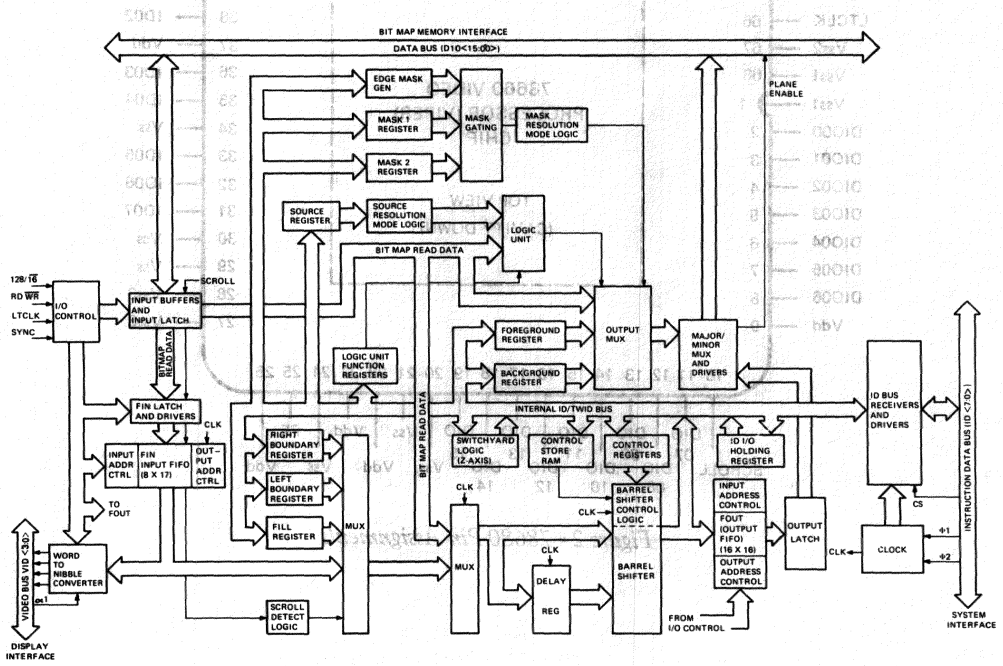


Figure 1 • 78680 Video Processor Block Diagram

It performs high-speed parallel processing of video data and transfers the data into and out of a bit-map memory plane. One video processor is required for each bit-map plane. The video processor receives commands from the video control and performs the data manipulations required for screen refresh, scrolling of windows, and screen updates. It provides the bandwidth necessary to display up to 850,000 pixels on a high-resolution CRT at an 80-Hz refresh rate. It contains a programmable bus-width structure and internal buffering for use with 64K or 256K RAMs.

• Pin and Signal Description

This section provides a brief description of the input and output signals and power and ground connections for the 78680 video processor 68-pin cerquad package. The pin assignments are identified in Figure 2 and the signals are summarized in Table 1.

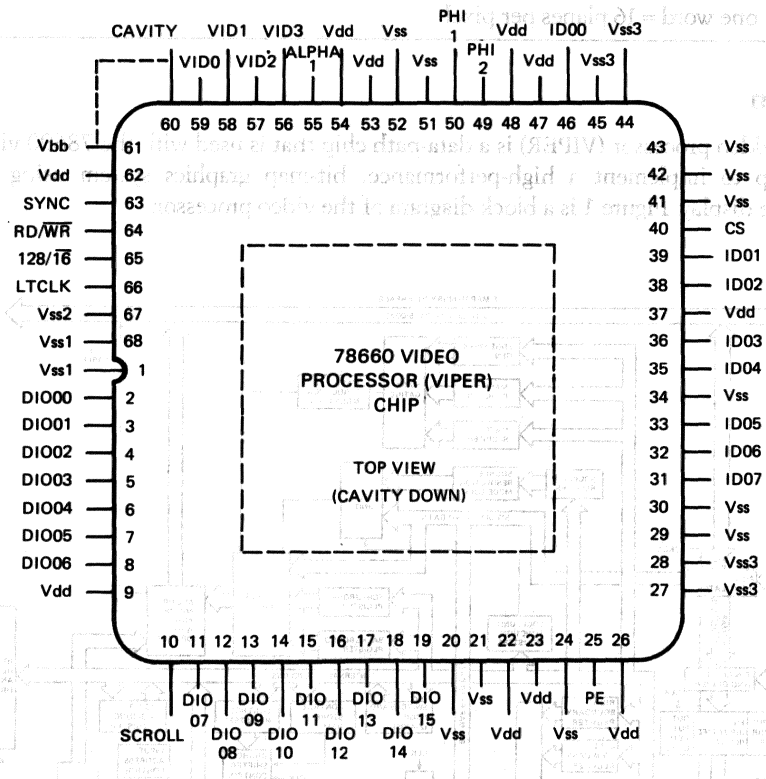


Figure 2 • 78680 Pin Assignments

Table 1 • 78680 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
2-8, 11-19	DIO<15:00>	input/output	Data bus—Bidirectional data lines.
10	SCROLL	input	Scroll—Specifies the word to be scrolled
25	PE	output	Plane enable—Enables a write to bit-map memory.
31-33, 35,36,38 39,46	ID<7:0>	input/output	Instruction/data bus—A bidirectional data bus.
40	CS	input	Chip select—Enables an instruction to be received.
49,50	PHI1,PHI2	inputs	Phase 1 and Phase 2—Clock signals for timing.
55	ALPHA1	input	Specifies a data transfer on VID<3:0> lines.
56-59	VID<3:0>	output	Video—Video output lines to refresh the display.
60*	CAVITY		Cavity—The 78680 chip substrate.
61*	V_{BB}	output	Voltage—Voltage generator output.
63	SYNC	input	Synchronize—Initializes internal logic and updates registers.
64	RD/$\overline{WR}$	input	Read/write—Specifies a bit-map read or write operation.
65	128/$\overline{16}$	input	Selects a 128-bit process for screen refresh or 16-bit process for updating bit-map memory.
66	LTCLK	input	Latch clock—Clock timing to latch data on DIO<15:00> lines.
9,22,23, 26,37,47 48,53,54, 62	V_{DD}	input	Voltage—Power supply 5 Vdc.
1,20,21,24 27-30,34 41-45,51,52 67,68	V_{SS}	input	Ground—Ground reference.

Bit Map Interface Signals

Bit-map Data Input/Output (DIO<15:00>)—These lines contain the bit-map data input and output data to each bit map memory. When the RD/$\overline{WR}$ signal is high, the DIO<15:00> output drivers are a high-impedance state.

Scroll (SCROLL)—This signal determines which of the data words received on the DIO<15:00> lines are to be scrolled.

Plane Enable (PE)—This signal determines if the bit-map memory may process a write operation.

128 or 16-bit Mode (128/$\overline{16}$)—When high, this signal enables the processing of 128 bits for screen refresh and scrolling operations. When low, 16 bit bit-map update operations are enabled.

Bit-map Memory Read/Bit-map Memory Write (RD/$\overline{WR}$)—When high, this signal allows the video processor to receive bit-map memory data and the DIO<15:00> are a high-impedance state. When low, the falling edge of the RD/$\overline{WR}$ signal latches the bit-map memory data on the DIO<15:00> lines and enables the DIO<15:00> output drivers.

Latch Clock (LTCLK)—When the RD/$\overline{WR}$ and 128/$\overline{16}$ lines are high, the trailing edge of the LTCLK signal latches in bit-map memory data on the DIO<15:00> lines. When this line is low and the 128/$\overline{16}$ line is high, the leading edge of LTCLK causes valid scrolled data to be shifted out on the DIO<15:00> lines.

Synchronize (SYNC)—This pulse initalizes all internal control flip-flops and latches and updates some master-slave registers.This signal does not perform the function of a reset input.

Video Bus Interface Signals

Video (VID<3:0>)—The rising edge of the ALPHA 1 signal shifts the screen refresh data out on these lines.

ALPHA 1—This signal causes the data to be shifted to the VID<3:0> lines.

Instruction/Data Bus Interface Signals

Phase 1 and Phase 2 Clocks (PHI1 and PHI2)—These are nonoverlapping clock inputs that determine the overall timing and control of the video processor.

Chip Select (CS)—When line ID<7> is high, this signal is asserted to allow the incoming instruction to be latched and executed. If line ID<7> is low, the incoming instruction is ignored when the CS signal is asserted. When receiving data on lines ID<7:0>, the polarity of CS and/or ID7 is irrelevant.

Instruction Data (ID<7:0>)—These input and output lines normally remains in a high-impedance state during input instructions and data. The output drivers are enabled by the internal execution of Z-axis or control store RAM instructions.

Power and Ground Connections

Power Supply (V_{DD})—Supplies 5 Vdc power to the 78680 video processor.

Ground (V_{SS})—Ground reference for all internal logic.

Ground (V_{SS1})—Ground reference for the output drivers of lines DIO<15:0>.

Ground (V_{SS2})—Ground reference for the LTCLK, RD/$\overline{WR}$, 128/$\overline{16}$ and SYNC input signals.

Ground (V_{SS3})—Ground reference for the ID<7:0>, VID<3:0>, and PE signals.

Cavity—Chip substrate that connects to V_{BB} pin 60.

• Architecture Summary

A typical bit-map processor system, shown in Figure 3, consists of high-speed timing logic, a local processor or remote processor that performs DMA operations, the 78690 video control, the 78680 video processor, bit-map memory planes, and the color map and shift register logic. The timing logic generates the system clock pulses. The local processor provides the commands to update the video control memory. The video control performs functions that are common to all memory planes such as raster computation operations, scan timing, system status generation, and memory address generation. Each video processor transfers information into and out of its bit-map memory plane. The output data is transferred through the shift register to a color map and digital-to-analog converter to provide the composite video to a monitor.

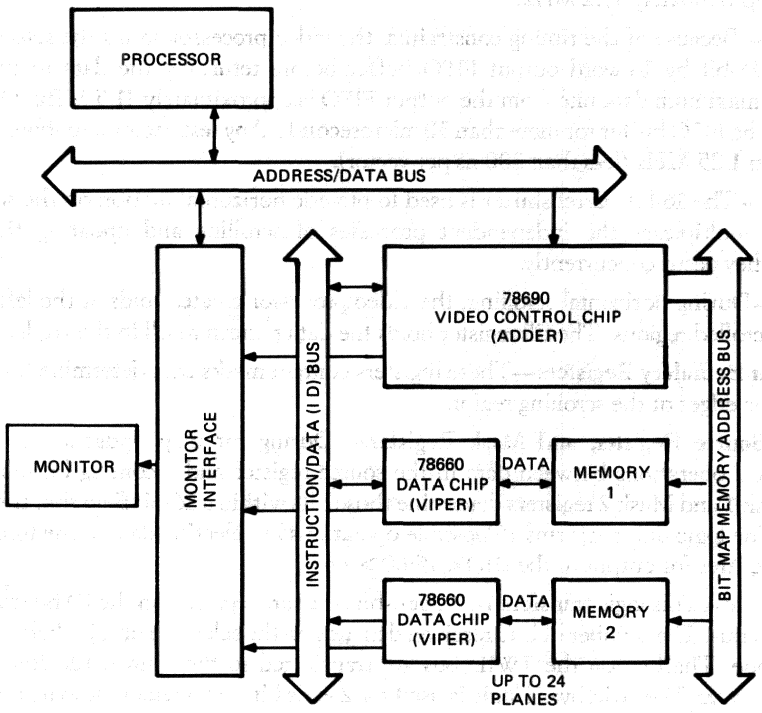


Figure 3 • 78680 Typical Bit-map Graphics System Configuration

The video processor communicates through the instruction/data (ID) bus, the bit-map memory address bus, and the display video bus.

The source, destination, barrel-shift constant, and edge mask data is transferred to the video processor registers through the ID bus which controls the operation of the video processor. The ID bus is used to load and read registers and to execute direct or indirect instructions. Data can also be exchanged between the registers of other video processors through this bus. During a source operation, the video processor receives the source information from the video control and locates the source in the bit-map memory. During the destination operation, the video processor receives a destination code and an edge mask to determine which bits of the data bus are to be written.

The display memory bus transfers the address required to automatically refresh the memory, the scrolling information, and the screen update information.

The display video bus connects the video processor to the display circuits including the shift registers, color map, and digital-to-analog converters.

Hardware Description

The following paragraphs provide a brief description of the major hardware functions of the video processor shown in Figure 1.

Input FIFO—The 8-bit by 17 word first-in/first-out (FIFO) buffer permits uninterrupted data flow between bit-map memory and the external display circuits for screen refreshing. The maximum data rate into the input FIFO buffer is approximately 11.5 MHz.

Word-to-nibble Converter—This converter reads the input FIFO buffer, divides the word into 4-bit nibbles, and transfers the nibbles to the video output bus. The maximum clocking rate of the converter is approximately 17.2 MHz.

Output FIFO—Because of the timing constraints, the video processor stores the scrolled data in the dynamic 16-bit by 16-word output FIFO buffer before returning the data to the bit-map memory. The maximum data rate from the output FIFO is approximately 11.5 MHz. Data should not remain in the FIFO buffer for more than 30 microseconds. Any test vector rate should therefore be greater than 1.25 MHz (less than 800 ns per vector).

Barrel Shifter—The 16-bit barrel shifter is used to provide horizontal motion on the screen. The barrel shifter multiplexes the independent processes of scrolling and updating the bit-map memories so they occur concurrently.

Fill Register—During horizontal scrolling, the video processor creates voids at the left and right edges of the scrolled regions. The fill register holds the data pattern to fill in the voids.

Left and Right Boundary Registers—These registers contain masks that determine the actual bit positions of the edges of the scrolling region.

Logic Unit, Source Register, and Mask Registers—During bit map updates, the logic unit performs logical operations between data in the source register and incoming bit-map memory data. The Mask 1 and Mask 2 registers determine those bits within a 16-bit field that the logic unit will modify. The logic unit performs 16 possible operations by selecting data in the foreground or background register for output to the DIO<15:00> lines.

Switchyard Logic—This logic transfers 1-, 2-, or 4-bits that are received on the ID bus to the 16-bit internal TWID bus. The number of bits received depends on the selection of full-, half-, or quarter-resolution mode. The bits on the TWID bus are transferred to the source, fill, foreground, or background register. The switchyard logic is used for Z-axis bit-map memory update operations.

ID I/O Holding Registers—These registers provide buffering between the 8-bit ID bus and the 16-bit internal TWID bus.

Edge Mask Generator—The edge mask generator converts an 8-bit code into a 16-bit mask, which is a window of variable width and position for raster operations. It may be placed anywhere within a 16-bit field.

Scroll Constant Register—This register stores barrel shifter control data. The scroll constant provides shift magnitude and direction. One register bit determines the up or down scroll direction and one bit enables the memory plane when writing scroll data back to the bit-map memory.

Plane Address Register—This register stores the device address. The video control compares the value in this register to the incoming address to determine if a Z-axis instruction should be executed.

Resolution Mode and Bus-width Mode Registers—These registers control the operating modes of the video processor.

Logic Unit Function Register—Four registers that control the logic unit and logic unit functions.

Control Store RAM (CSR)—Six registers that provide indirect instruction execution to control the origin and destination of data transferred on the internal TWID bus and the external ID bus.

Programming Functions

The following types of instructions are available to the applications programmer.

Register load—The register load instructions are used to load data in one of the 22 addressable registers in the video processor. Two registers are available for mode selection, four registers for direct address control, six registers for data storage, and ten registers for indirect control functions.

Z-dimension color codes—Color information is received from the address processor and is written into one of the four data registers—source, fill, foreground, and background. Up to 16 video processor chips can be addressed in parallel at the same time. During a Z-read cycle, all color data for a specific pixel can be accessed in parallel from the memory planes.

Active cycles—During an indirect data transfer between the bit-map memory planes and the address processor, one video processor transmits the information, and the remaining video processors can receive the data and store it in one of three data registers.

The video processor has a programmable indirect instruction set, which allows the address processor in a video subsystem to command direct memory access (DMA) operations between any memory components of the video subsystem. This enables data to be translated, rotated, and scaled. The video processor performs the following data transfer cycles.

Screen refresh—During this cycle, the data in the video memory is sampled and transferred to the video circuits.

Scroll—During this cycle, data in bit-map memory is read, barrel shifted, and written into the bit-map memory resulting in a scroll operation within a defined window of the display.

Bit-map update—This cycle requires an instruction and data. The type of instruction determines the source of data, indicates the destination of the data, and determines the disposition of the data at the new location.

Modes of Operation

The 78680 video processor operates in the following modes.

Bus width mode—This mode allows the user to select an effective bus width of 4, 8, or 16 bits to optimize the cost advantages from low-density, partial page displays to high-density RAMs.

Resolution mode—This mode enables the programming of color or intensity changes by 1, 2, or 4 bits. An increase in the intensity or color results in a decrease in the resolution on which those changes can be made. The resolution mode can be used effectively with terminals with low plane counts.

Axis mode—This mode provides a method of data transfer to a group of planes that overlap each other in the Z dimension, as opposed to the normal transfer between pixels in the same plane.

Register Descriptions

The video processor contains 22 user-loadable, static registers that combine data and control instructions. The width of the registers is 16 bits. However, many of the registers operate with less than 16 bits. The information written into the registers cannot be read. A register is accessed by the least significant 5 bits of ID<7:0> lines. Figure 4 shows the register load transfer sequence. The ID bus instruction is a one-byte code and most instructions are followed by a 16-bit data transfer or subinstruction as listed:

Parameter	Meaning
IN	Instruction
SI	Shift constant
LB	Data source–low byte
HB	Data source–high byte

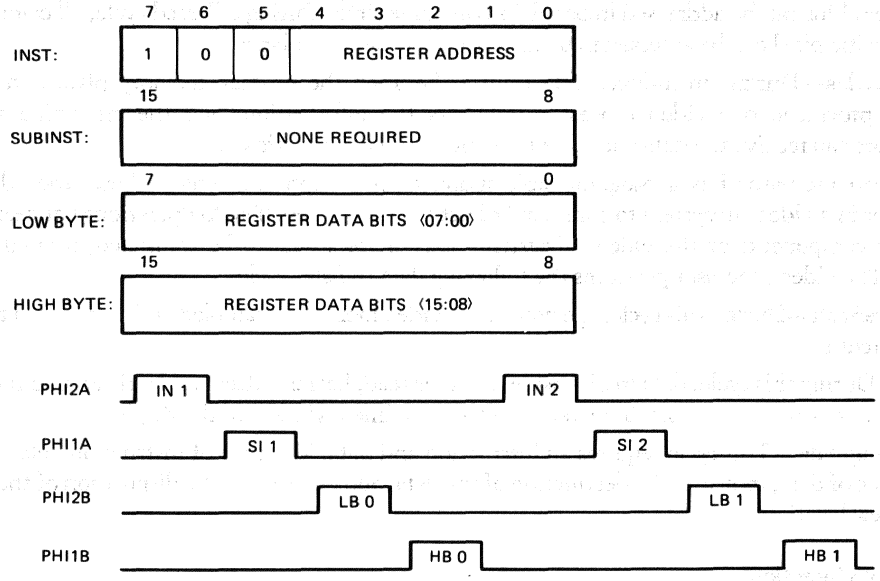


Figure 4 • 78680 Register Load Data Transfer Model

The video processor registers, type, and address assignments are listed in Table 2. Three of the registers are reserved.

Table 2 • 78680 Register Address Assignments

Address (hexadecimal)	Register Name	Width	Type
0	Resolution mode	1:0	Mode
1	Bus width	3:2	Mode
2	Scroll constant	6:0	Direct control
3	Plane address	5:0	Direct control
4	Logic function 0	7:0	Indirect control
5	Logic function 1	7:0	Indirect control
6	Logic function 2	7:0	Indirect control
7	Logic function 3	7:0	Indirect control
8	Data mask 1	15:0	Data
9	Data mask 2	15:0	Data
A	Source	15:0	Data
B	Fill	15:0	Data
C	Left scroll boundary	15:0	Direct control
D	Right scroll boundary	15:0	Direct control
E	Background	15:0	Data
F	Foreground	15:0	Data
10	Control store RAM (CSR0)	6:0	Indirect control
11	Control store RAM (CSR1)	6:0	Indirect control
12	Control store RAM (CSR2)	6:0	Indirect control
13	Reserved		
14	Control store RAM (CSR4)	6:0	Indirect control
15	Control store RAM (CSR5)	6:0	Indirect control
16	Control store RAM (CSR6)	6:0	Indirect control
17	Reserved		
18	Reserved		

The data and direct control registers A through F (hexadecimal) are 16-bit registers that are loaded by the ID bus load register commands. The mask and source registers are loaded by data transfers during the rasterop cycles. The source and fill registers can be loaded with a data constant to select a solid color during Z-axis register load commands. The least significant bit of the word determines the left pixel and the most significant bit determines the right pixel on the display.

Resolution mode—The resolution mode register controls the action of the mask bits (mask 1, mask 2, and edge mask), the source register, the barrel shifter constant, and the Z-axis command. This mode allows the 78680 video processor to respond as 1, 2, or 4 planes. Figure 5 shows the format of the register and Table 3 lists the functions of the register information. Because each video processor can be programmed for a different resolution, the single barrel-shifter constant should be truncated for low-resolution applications except when scrolling. Horizontal scrolling should be used for the lowest-resolution plane to prevent errors caused by truncating to result in the scrolling of an undesired region.

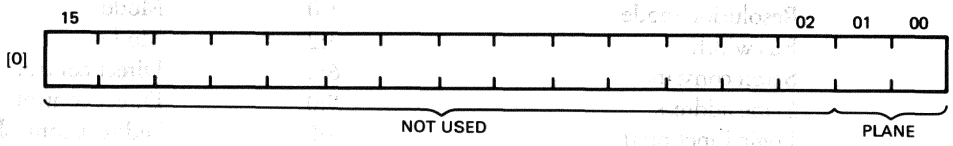


Figure 5 ▪ 78680 Resolution Mode Register Format

Table 3 ▪ 78680 Resolution Mode Register Description

Bit	Description			
15:02	Not used.			
01:00	Planes—Defines the planes of response as follows:			
	Bit 01	**Bit 00**	**Plane**	**Description**
	0	0	1	Full resolution
	0	1	2	Half resolution
	1	0	3	Undefined
	1	1	4	Quarter resolution

Plane 1—When plane 1 is specified, each bit in the mask registers and source register controls its respective mask multiplexer or logic unit bit independently of the remaining bits. The plane address can be programmed to any value and all barrel shifter constants are significant. During Z-axis operations, the video processor receives or transmits the bit that corresponds to its plane address.

Plane 2—When plane 2 is specified, the even and odd mask or source bits are OR gated, and the results are used to control the corresponding mask multiplexer or logic unit bits. The least significant bits of the barrel shifter constant is truncated so that the data will move only by a multiple of 2 bits. During Z-axis operations, the video processor receives or transmits 1 bit that corresponds to its plane address and the next most significant bit. The plane address bit must be programmed to an even value.

Plane 4—When plane 4 is specified, the 4 mask or source bits are OR gated and the results are used to control the corresponding mask multiplexer or logic unit bits. The two least significant bits of the barrel shifter constant are truncated so that the data will move only by a multiple of 4 bits. During Z-axis operations, the video processor receives or transmits the four bits that correspond to its plane address and the next most significant bit. The plane address bit must be programmed to a multiple of four.

Bus width—The bus width register selects the number of video bus bits depending on the number of pixels required for the display. The 78680 video processor and 78690 video control must be set to the same bus width. Figure 6 shows the register format and Table 4 lists the bit selections and the reduction in the video bus speed that results.

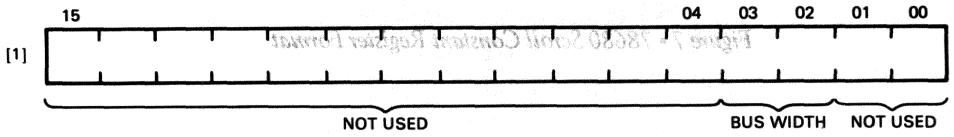


Figure 6 • 78680 Bus Width Register Format

Table 4 • 78680 Bus Width Register Descriptions

Bit	Description			
15:04	Not used.			
03:02	Bus width—Selects the number of video bus bits used for the display as follows:			
	Bit 03	**Bit 02**	**Data bus width**	**Video output**
	0	0	4-bits	4 bits every fourth alpha pulse
	0	1	8-bits	4 bits every other alpha pulse
	1	0	undefined	
	1	1	16-bits	4 bits every alpha pulse
01:00	Not used.			

Scroll constant—The scroll constant register is double buffered and is used to select the left or right horizontal scrolling and vertical scrolling operations. The data entered in the register becomes active on the following frame. Figure 7 shows the register format and Table 5 lists the functions of the register information.

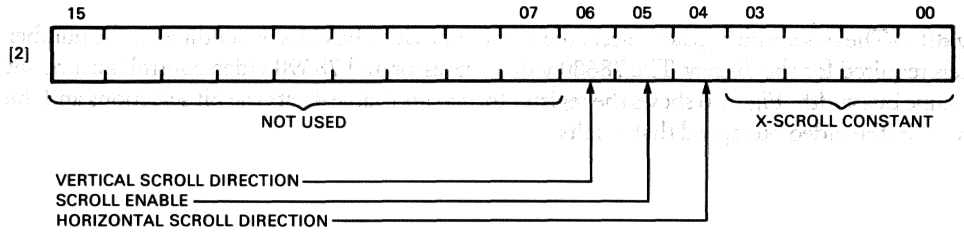


Figure 7 • 78680 Scroll Constant Register Format

Table 5 • 78680 Scroll Constant Register Description

Bit	Description
15:07	Not used.
06	Vertical scroll direction—Selects the direction for the vertical scroll as follows. Used to compensate for the asymmetries between the up and down scrolling directions. 0 = down, 1 = up, left, or right
05	Scroll enable—Set to enable the horizontal or vertical scrolling of the data accessed. Cleared to disable scrolling and the writing of memory planes.
04	Horizontal scroll direction—Control the direction of the X scroll constant specified by bits 3:0 as follows: Cleared when using the Y scroll constant. 0 = left, 1 = right
03:00	X-scroll constant—Selects the number of pixels per frame as follows: For left scrolls: 0 value = 0 pixel and 15 = 15 pixels For right scrolls: 0 value = 1 pixel and 15 = 16 pixels

Plane address—The plane address register determines the Z-axis operation. The plane addresses must be different for each 78680 video processor. Addresses cannot overlap. A permanent plane address is set if the scrolling process loads the fill register in Z-mode. If the fill register is loaded individually by the scrolling process, then different update regions can have different plane address arrangements. The register format is shown in Figure 8 and function of the register information is described in Table 6.

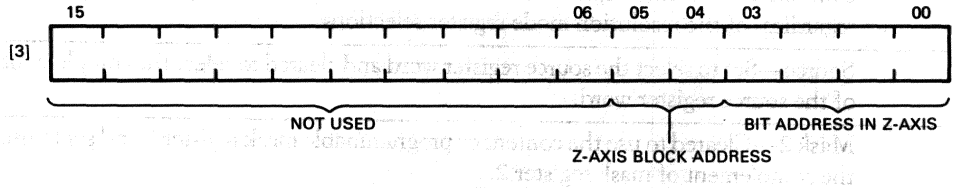


Figure 8 ▪ 78680 Plane Address Register Format

Table 6 ▪ 78680 Plane Address Register Description

Bit	Description
15:06	Not used.
05,04	Z-access block address 0 to 3—Defines the high-order bits of a 6-bit plane address when using more than 16 planes or subplanes within a system.
03:00	Bit address in Z-axis block 0 to 3—This address is set to the bit on which data will be exchanged on the ID bus during Z-mode transfers. For low resolution applications, this bit is the low-order bit and must be a multiple of 2 everything that is a multiple of 4 is a multiple of 2.

Logic unit function (0-3)—Each of the four logic unit function registers (0 through 3) include 8-bits of information. The registers combine the word read from destination (D) during a read-modify-write cycle with the contents of the source (S) register. The results are used to select the color of the foreground and background of the display. The format of the registers are shown in Figure 9 and described in Table 7.

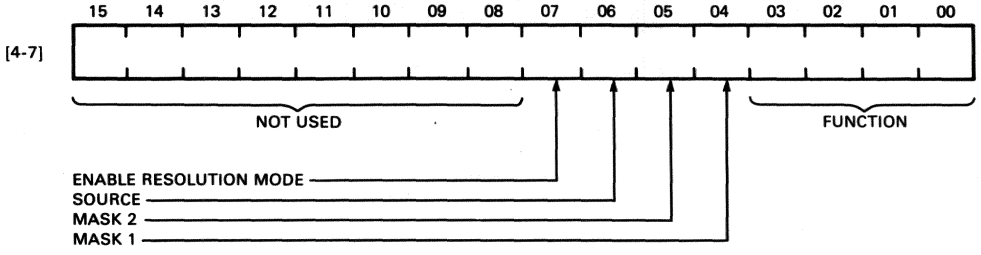


Figure 9 ▪ 78680 Logic Unit Function Registers (0-3) Format

Table 7 • 78680 Logic Unit Function (0-3) Register Description

Bit	Description
15:08	Not used.
07	Enable resolution mode—Cleared to enable the resolution mode for the source register. Set to disable the resolution mode. When disabled, the bits are not combined with the adjacent bits and are transferred through the logic unit regardless of the resolution mode register selections.
06	Source—Set to select the source register word and cleared to select the complement of the source register word.
05	Mask 2—Cleared to use the content of programmable mask register 2 and set to use the complement of mask register 2.
04	Mask 1—Cleared to use the content of programmable mask register 1 and set to use the complement of mask register 1.
03:00	Function—Specifies the contents of the register as follows:

Bits 03	02	01	00	Function
0	0	0	0	zeros
0	0	0	1	not (D or S)
0	0	1	0	(not D) and S
0	0	1	1	not D
0	1	0	0	D and (not S)
0	1	0	1	not S
0	1	1	0	D XOR S
0	1	1	1	not (D and S)
1	0	0	0	D and S
1	0	0	1	not (D XOR S)
1	0	1	0	S
1	0	1	1	(not D) or S
1	1	0	0	D
1	1	0	1	D or (not S)
1	1	1	0	D or S
1	1	1	1	ones

Mask 1 and mask 2—Programmable mask registers 1 and 2 are used to control the bits that a read-modify-write cycle will modify. Loading the data mask register 1 also loads data mask register 2 with the same information. The output of these registers and the edge register determine the bits that will be modified by the 78680 video processor. Figure 10 shows the format of the data mask registers.

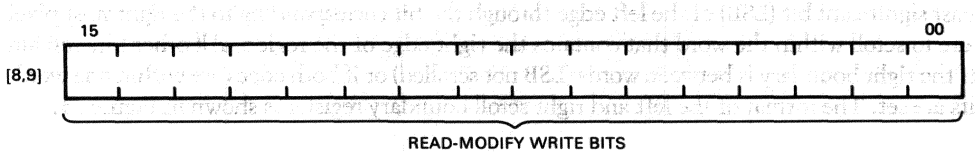


Figure 10 • 78680 Data Mask 1 and 2 Register Format

Source—The source register contains the source word for the logic unit that is combined with the destination information to select the foreground and background color for each pixel. The Z-mode address for this register is 00. The register format is shown in Figure 11.

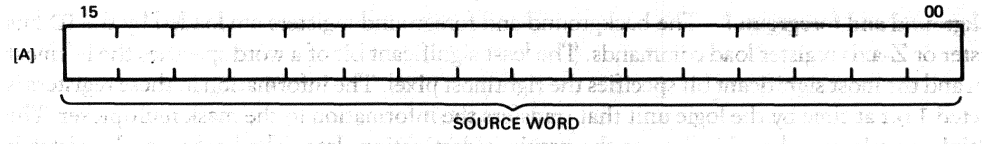


Figure 11 • 78680 Source Register Format

Fill—The fill register is double buffered and determines the fill area or blank space in memory that is created when scrolling. Normally, the Z-axis load command is used to select a solid color. If the register is loaded directly, all bits that correspond to the same subplane are set to the same value. The data from the register becomes active during the frame that follows.

Because the scroll region boundaries can be contained within one word, the leftmost and rightmost word can be programmed to any bit position. The scroll region boundaries of the 78690 video control, however, are limited to a multiple of four. When the video control is used, the boundaries must be selected with groups of 4 bits. Only the low 4 or 8 bits of this register are significant when using the 4- or 8-bit video bus widths, respectively. The register format is shown in Figure 12.

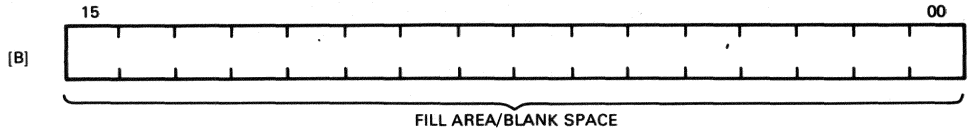


Figure 12 • 78680 Fill Register Format

Left and right scroll boundary—The left scroll boundary register defines the boundary for the left scroll. All bits corresponding to the pixels that contain the left edge of the region are cleared, and all other bits remain set. Normally, all bits from the pixel are cleared on the edge through the most significant bit of the word. If both of the edges are within one word, only the bits from the left edge through the right edge are cleared.

The right scroll boundary register defines the boundary for the right scroll. All bits are cleared from the least significant bit (LSB) of the left edge through the bit corresponding to the rightmost pixel that are to scroll within the word that contains the right edge of the region. All other bits remain set. If the right boundary is between words (LSB not scrolled) or if both edges are within one word, all bits are set. The format of the left and right scroll boundary register is shown in Figure 13.

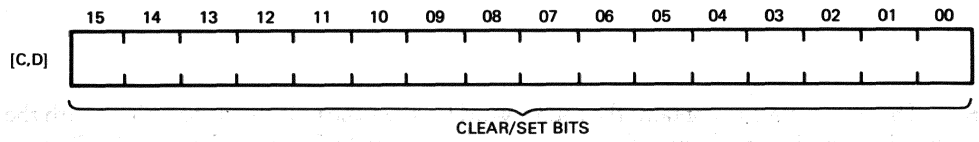


Figure 13 • 78680 Left and Right Scroll Boundary Register Format

Background and foreground—The background and foreground registers are loaded by the ID bus register or Z-axis register load commands. The least significant bit of a word specifies the leftmost pixel and the most significant bit specifies the rightmost pixel. The information in these registers is selected 1 bit at time by the logic unit that transfers the information to the mask multiplexer. The multiplexer selects either this data or the previous destination data. The background register is selected by a zero output and the foreground register by a one output from the logic unit. The register format is shown in Figure 14.

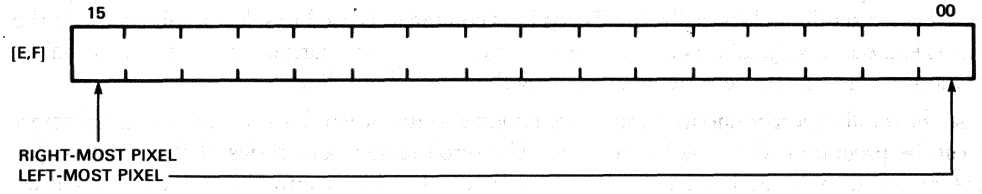


Figure 14 • 78680 Background and Foreground Register Format

Control Store RAM (CSR0-CSR6)—The control store RAM registers are CSR0 through CSR6. Register CSR3 is reserved. During raster operations, these registers control the transfer of data within the 78680 video processor and the data transfer to or from other ID bus devices. During each update memory cycle, the 78690 video control addresses the contents of these registers. When bank 1 is selected by the video control, CSR0 controls the first read source, CSR1 controls the second read source, and CSR3 controls the read-modify-write destination. When bank 2 is selected, CSR4 controls the first read source, CSR5 controls the second read source, and CSR6 controls the read-modify-write destination. Figure 15 shows the format of the control store RAM registers and Table 8 lists the function of the register information.

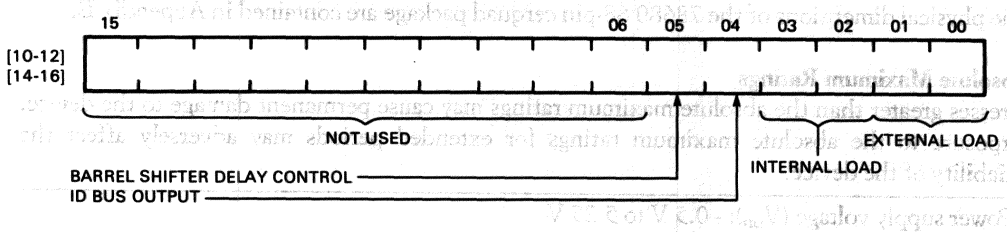


Figure 15 • 78680 Control Store RAM Registers (CSR0-CSR6) Format

Table 8 • 78680 Control Store RAM Description

Bit	Description
15:06	Not used
05	Barrel shifter delay control—Set during a fast mode raster operation to hold the previous word in a delay register so the remainder of each source word forms the next output word. This prevents excessive read operations to the source raster. This bit controls delay register one for CSR0 and CSR4 (for source 1) and delay register 2 for CSR1 and CSR5 (for source 2).
04	ID bus output—When a CSR is addressed during a memory read cycle, this bit is set to enable (one plane at a time is allowed) the transfer of data on the ID bus during a memory read cycle.
03:02	Internal Load—Select the register to receive the data through the barrel shifter from the local memory plane following a memory read operation as follows: **Bits** **03 02 Register** 0 0 none 0 1 source 1 0 mask 1 and mask 2 1 1 mask 2
01:00	External load—Selects the register to receive the input data from the ID bus following a memory read operation.

Specifications

The mechanical, electrical, and environmental characteristics and specifications for the 78680 video processor are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Power supply voltage (V_{DD}): 5.0 V
- Temperature range (T_A): 0°C to 70°C
- Ground (V_{SS1}),(V_{SS}),(V_{SS2}),(V_{SS3}): 0 V

Mechanical Configuration

The physical dimensions of the 78680 68-pin cerquad package are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Power supply voltage (V_{DD}): –0.5 V to 5.25 V
- Input voltage applied (V_{in}): –0.5 V to 6.0 V
- Output voltage applied (V_{out}): –0.5 V to 6.0 V
- Power dissipation (P_D): 1.9 W at 100°C, 2.8 W at 0°C
- Active temperature (T_A): 0°C to 70°C
- Storage temperature: –55°C to 125°C

Recommended Operating Conditions

- Power supply voltage (V_{DD}): 5 V ±5%
- Temperature (T_J) 0°C to 100°C
- Typical operating voltage: 5.0 V

dc Electrical Characteristics

The dc electrical parameters of the 78680 video processor for the operating voltage and temperature ranges specified are listed in Table 9.

Table 9 • 78680 dc Input and Output Parameters

Symbol	Parameter	Test Condition	Requirements Min.	Requirements Max.	Units
V_{IH}	High-level input voltage DIO<15:00>,CS,SCROL, ID<7:0>,SYNC, RD/WR,128/$\overline{16}$, LTCLK,ALPHA1, PHI1,PHI2		2.0	—	V
V_{IL}	Low-level input voltage DIO<15:00>,CS,SCROL, ID<7:0>,SYNC, RD/$\overline{WR}$,128/$\overline{16}$, LTCLK,ALPHA1, PHI1,PHI2		—	0.8	V
V_{OH}	High-level output voltage (all pins except ID<7:0>)	$V_{DD}=0$ $I_{OH}=-200\ \mu A$	2.4	—	V
	ID<7:0>		2.5	—	V
V_{OL}	Low-level output voltage DIO<15:00>,PE, VID<3:0>	$V_{DD}=0$ V $I_{OL}=4.0$ mA	—	0.4	V
	ID<7:0>	$I_{OL}=5.0$ mA			
I_{DD}	Active supply current	$V_{DD}=$ max $T_A=0°C$	—	110	mA
I_{IH}	Input leakage current (high)	$V_{DD}=$ max $V_{in}=V_{DD}$ (max.)	—	30	µA
I_{IL}	Input leakage current (low)	$V_{DD}=$ max. $V_{in}=0$ V	—	-30	µA
I_{OS}	Short-circuit output current[1] DIO<15:00>,PE, VID<3:0>	$V_{DD}=$ max.	-10	-90	mA
	ID<7:0>		-15	-140	mA
I_{OZL}	High-Z state output current	$V_{DD}=$ max. $V_{out}=0.4$ V	—	30	µA
I_{OZH}	High-Z state output current[2]	$V_{DD}=$ max. $V_{out}=2.4$ V	—	30	µA

Symbol	Parameter	Test Condition	Requirements Min.	Max.	Units
C_{in}	Input capacitance		—	4.7	pF
C_{out}	Output capacitance[3]		—	6.6	pF

[1]These are computer simulated measurements and cannot be tested. In the test simulation, only one output at a time may be short circuited to ground.
[2]The outputs include the driver current and input leakage current.
[3]Capacitance loading is for the output drivers and input receivers.

ac Electrical Characteristics

The ac timing parameters for the 78680 video processor are grouped according to major cycle timing, minor cycle timing ID bus and video bus timing, and synchronization cycle timing. The timing parameters in the tables are in microseconds unless listed otherwise.

Major Cycle Timing

Table 10 lists the symbols, definitions, and specifications for the major cycle timing shown in Figure 16. A major cycle begins on the rising edge of the PHI2 clock pulse and is equal to four PHI2 clock periods. A refresh read memory cycle has two major cycles—a read cycle and a screen refresh transfer cycle for editing or scrolling. Every second major cycle must be a read cycle. The video processor provides data to the display during the forward scan time of the screen refresh transfer cycle. During this cycle, a refresh read memory cycle transfers data from bit-map memory to the video bus.

Table 10 • 78680 Major Cycle Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{CYC1}	Period of LTCLK	54.5	—
t_{CYC2}	Period of PHI1 or PHI2	148	—
t_{CYC3}	Period of major cycles	*	
t_{D1}	Propagation delay of LTCLK rising to valid data on DIO<15:00>	—	32
t_{D3}	Propagation delay of 128/$\overline{16}$ switching to an updated value of PE.	—	37

Symbol	Definition	Requirements (ns) Min.	Max.
t_{HO1}	Hold time of RD/$\overline{WR}$ or 128/$\overline{16}$ switching to the falling edge of LTCLK. This parameter must be met relative to the last falling edge of LTCLK in a major cycle.	10	—
t_{HO2}	Hold time of DIO<15:00> or SCROLL to the falling edge of LTCLK	13	—
t_{HZ1}	Propagation delay of RD/$\overline{WR}$ going high to a high-impedance on DIO<15:00>	3.0	18
t_{NOV}	Nonoverlap time, PHI1 to PHI2	15.5	—
t_{PWH1}	Pulse width high of LTCLK	18	—
t_{PWH2}	Pulse width high of PHI1 or PHI2	55.5	—
t_{PWL1}	Pulse width low of LTCLK	36.5	—
t_{SU1}	Setup time of RD/$\overline{WR}$ switching to the falling edge of LTCLK. This parameter must be satisfied relative to the first falling edge of LTCLK in a major cycle.	13	—
t_{SU2}	Setup time of 128/$\overline{16}$ rising to the rising edge of LTCLK. This parameter must be satisfied relative to the first rising edge of LTCLK in a major cycle and applies only to transitions between minor and major write cycles (guaranteed by not exceeding t_{PD2}).	6.5	—
t_{SU3}	Setup time of DIO<15:00> or SCROLL to the falling edge of LTCLK	0	—
t_{SU10}	Setup time of RD/$\overline{WR}$ or 128/$\overline{16}$ switching to the first falling edge of PHI1 in any major cycle	26.5	—
t_{ZL1}	Propagation delay of RD/$\overline{WR}$ going low to valid data on DIO<15:00>	3.5	30

*Minimum t_{CYC3} = 4 PHI2 cycles or 16 ALPHA cycles

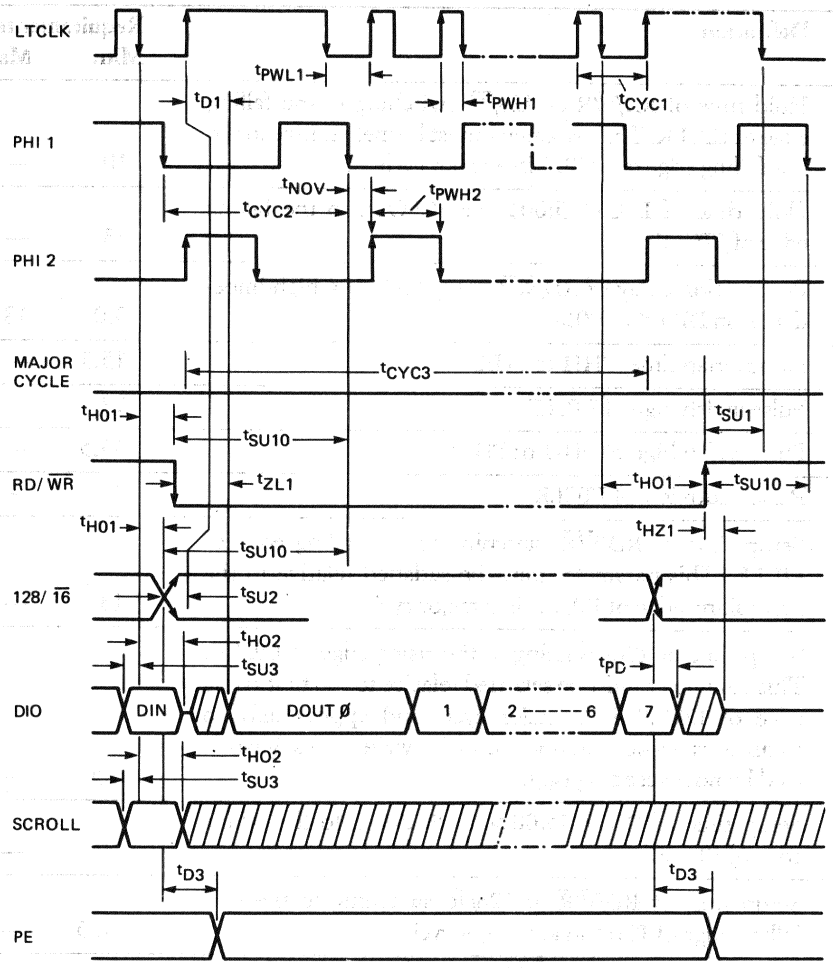


Figure 16 • 78680 Major Cycle Timing

Minor Cycle Signal Timing

Table 11 lists the specifications for the minor cycle events including a memory cycle shown in Figure 14. A memory cycle starts at the beginning of a major cycle and must end by the beginning of the next major cycle. Only one memory cycle must be in progress at a time.

Table 11 - 78680 Minor Cycle Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{D2}	Propagation delay of PHI2 rising to valid data on DIO<15:00>. This parameter applies only to Z-axis write operations.	—	63
t_{D5}	Propagation delay of ID data setup to following edge of the second PHI1 in a minor cycle to valid data on DIO<15:00>.	—	79.5
t_{HO3}	Hold time of an asserted CS to the falling edge of PHI2.	40	—
t_{HO4}	Hold time of DIO<15:00> to the second falling edge of PHI2 in a minor read cycle.	13	—
t_{HO5}	Hold time of DIO<15:00> to the falling edge of RD/$\overline{WR}$.	0	—
t_{PD}	Propagation delay of the rising edge of 128/$\overline{16}$ to DIO<15:00> rising or falling.	0	38.5
t_{SU4}	Setup time of an asserted CS to the falling edge of PHI2.	40	—
t_{SU5}	Setup time of DIO<15:00> to the falling edge of the second PHI2 in a minor cycle.	50	—
t_{SU6}	Setup time of DIO<15:00> to the falling edge of RD/$\overline{WR}$.	13	—
t_{ZL1}	Propagation delay of RD/$\overline{WR}$ going low to valid data on DIO<15:00>.	3.5	30

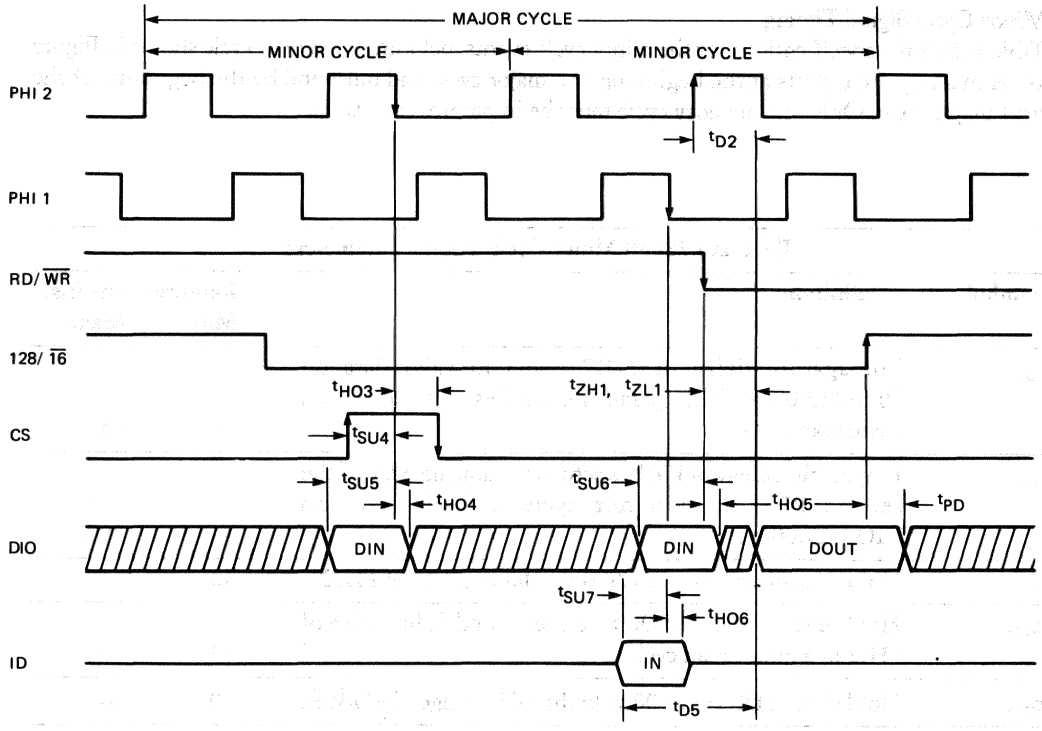


Figure 17 • 78680 Minor Cycle Signal Timing

ID Bus and Video Bus Signal Timing

Table 12 lists the specifications for the ID bus and video bus timing shown in Figure 18. A major cycle begins on the rising edge of the PHI2 clock pulse with 4 PHI2 clock periods. PHI1 and PHI2 must have the same period and must not overlap. The alpha clock has a timing period one-fourth of the PHI2 clock and has coincident rising edges.

Table 12 • 78680 ID Bus and Video Bus Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{32}	Time delay to valid data on VID <3:0> is 10 ALPHA cycles plus t_{D4} after start of present major cycle. (Not testable).	—	—
t_{ACC}	Propagation delay of PHI2 rising to valid data on VID<3:0> while ALPHA is high.	—	78
t_{CYC4}	Period of ALPHA.	36	—
t_{D4}	Propagation delay of alpha rising to valid data on VID<3:0>. The rate of updating the data depends on the the bus width mode: full page=once/ALPHA clock half page=once/two ALPHA clocks one-fourth page=once/four ALPHA clocks	3.0	29
t_{HO6}	Hold time of ID<7:0> to the falling edge of PHI1 or PHI2.	15.5	—
t_{HZ2}	Propagation delay of PHI1 or PHI2 going low to a High-Z condition on ID<7:0>.	3.0	40
t_{PWH3}	Pulse width high of ALPHA.	18	—
t_{PWL2}	Pulse width low of ALPHA.	18	—
t_{R2}	Input rise time (fall time) ID<7:0>	—	26.5
t_{SU7}	Setup time of ID<7:0> to the falling edge of PHI1 or PHI2.	13	—
t_{ZH2}	Propagation delay of PHI1 or PHI2 rising to valid data on ID<7:0>.	3.5	40

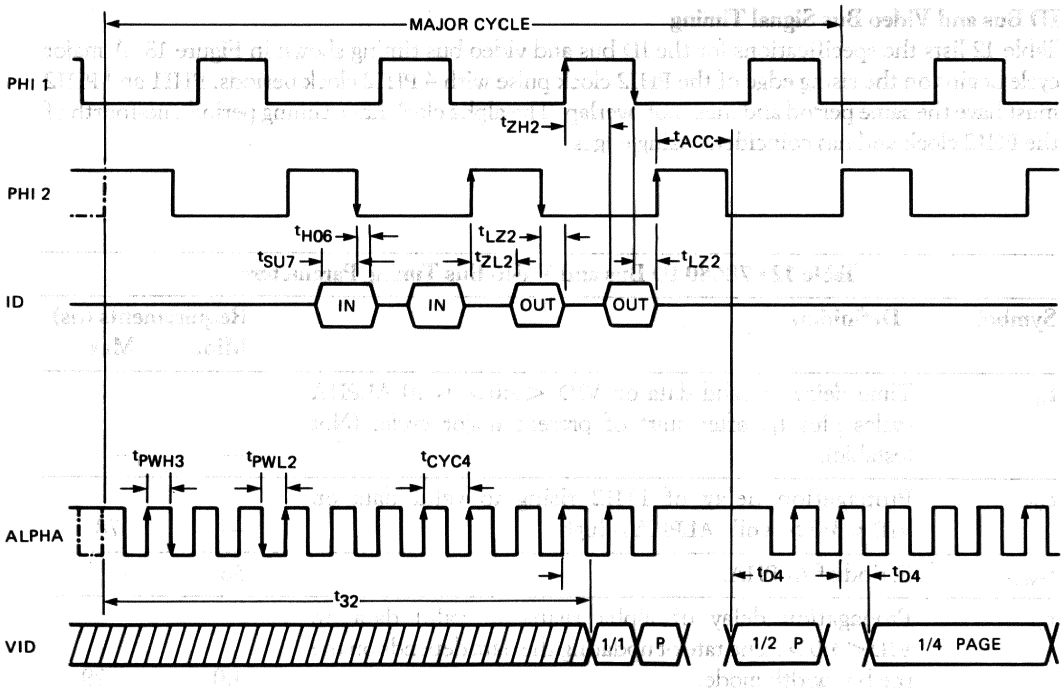


Figure 18 ▪ 78680 ID Bus and Video Bus Signal Timing

Synchronization Cycle Timing

Synchronization occurs once per frame during the vertical retrace period. When the SYNC signal is asserted, all internal clocks halt in a t=0+state. The video processor suspends all data processing and resets the counters that are used to transfer data out to the video bus and out of and into the ID and data bus. The LTCLK has eight time periods within a major cycle. These periods can occur within the cycle to accommodate different types of memories. Table 13 lists the synchronization timing parameters shown in Figure 19.

Table 13 • 78680 Synchronization Cycle Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{41}	Use the rising edge of PHI2 as the major cycle boundary except after a sync cycle. (Not testable).	—	—
t_{42}	The number of ALPHA cycles that must elapse before PHI2 goes low for the first time after a SYNC cycle is 1, 5, 9, or 13.	—	—
t_{CYC3}	Period of major cycles. (Not testable).	*	—
t_{PWH4}	Pulse width high of SYNC.	302	—
t_{SK}	Skew between PHI2 falling and ALPHA falling.	—	±5.0
t_{SU8}	Setup time of SYNC going low to the first falling edge of ALPHA.	302	—
t_{SU9}†	Setup time of LTCLK falling to the rising edge of PHI2.	26.5	—

†In order for the t_{32} specification (Table 12) to be valid, data must have been loaded into the internal FIFO buffers. Therefore, the first falling edge LTCLK and its associated data during a major read cycle must precede the third rising edge of PHI2.

*Minimum t_{CYC3}=4 PH12 cycles or 16 ALPHA cycles.

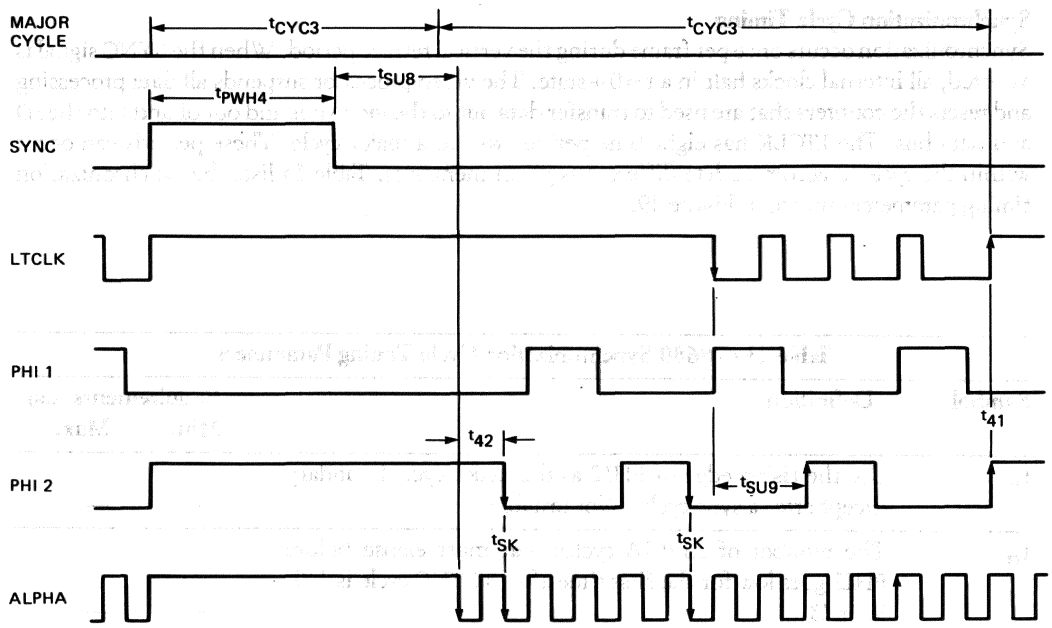


Figure 19 • 78680 Synchronization Cycle Timing

• Interfacing Techniques

The video processor communicates with the video subsystem using the three TTL-compatible interfaces shown in Figure 3—the system interface, the bit-map interface, and the display interface.

The system interface lines consist of the chip select (CS), the system clocks, and an 8-bit, bidirectional instruction/data bus. The system interface connects to the address processor memory planes and other memory planes.

The bit-map interface includes the 16-bit bidirectional data bus DIO<15:00> and the LTCLK, 128, RD, SCR, OL, PE, PHI1, and PHI2 signals that control data flow on the bus. The bit-map interface connects to the bit-map memory planes.

The display interface includes the video bus and the signals that control it. The video bus VID<3:0> connects to the display circuits. The control signals include the ALPHA clock pulses and the PHI1 and PHI2 system clocks pulses. This bus is used during screen refresh cycles.

The instruction/data (ID) bus is used to transfer information between the video processor and controllers. It is used to load and read registers and to execute direct or indirect instructions.

Interface Connections

The following circuit and connections are recommended for the proper operation of the 78680 video processor.

Power—The 78680 video processor operates with a 5 Vdc power supply. Ten of the chip pins connect to V_{DD} and one connects to V_{BB}. No special filtering is required. The 5 Vdc connects to the V_{DD} pins and the power supply ground connects to V_{SS}. Pin 61 (V_{BB}) is the voltage generator output and must be connected to the pin 60 (cavity) which is the substrate.

Powerup latch—The video processor includes a circuit to ensure that the ID bus is in a high-impedence state during the powerup sequence. A latch is set during the powerup sequence to force the ID bus to this state. The deassertion of the CS signal clears the latch.

ID bus lines—Connecting a 50-pF capacitor to ground on each ID bus line compensates for component variations in the interface design. Multiplane single-module systems may not require these capacitors. However, the etch capacitance to ground and to other signals should be evaluated. The noise margin should be limited to 0.2 V to 0.3 V. The total ID bus load capacitance should be less than 400 pF.

Features

- Programmable display resolution
- Programmable screen synchronous interrupt
- Displays up to 1024 by 864 pixels at 60 Hz noninterlaced
- Supports video memory address space up to 8K by 8K
- Up to 24 bit planes
- Hardware raster operations—two sources to a destination
- Raster operations at rates of 0.5 to 8 million pixels per second
- Smooth scrolling of rectangular windows
- Hardware clipping to the window boundaries
- Hardware rotation and noninteger scaling of source
- Hardware polygon fill with stipple pattern or solid color
- Addressing of pixels in X- or Z-mode

Description

The 78690 video control (Adder) is an 84-pin, ZMOS VLSI chip used with the 78680 video processor (Viper) chip when building a high-performance, bit-map graphics system with a color or monochrome display system. The video control performs functions common to all bit-map memory planes, such as scan timing, system status generation, and memory address generation for screen refresh and screen updates. The video control sends commands to one or more video processor, which perform the data manipulation for each memory plane. Figure 1 is a block diagram of the video control.

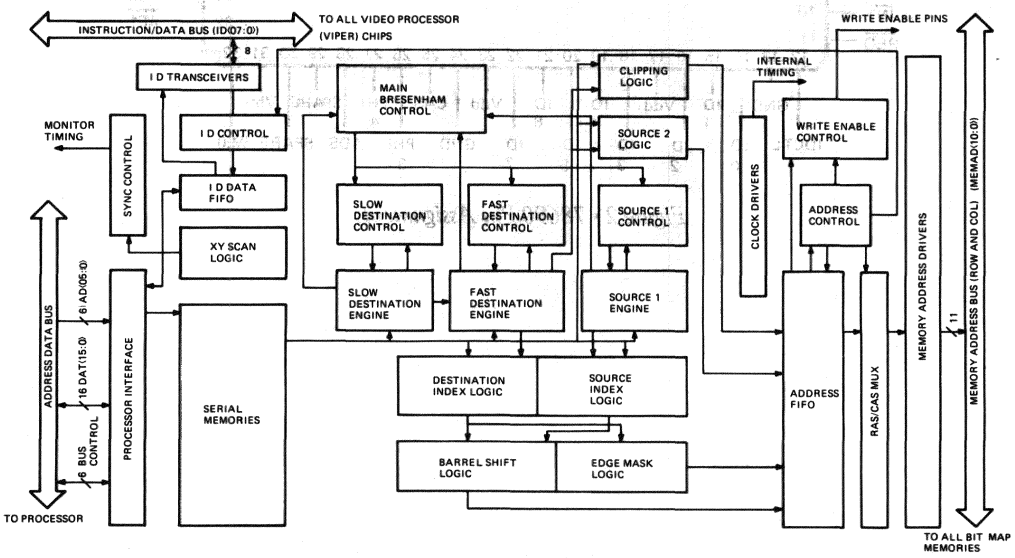


Figure 1 • 78690 Video Control Block Diagram

• Pin and Signal Description

This section provides a brief description of the input and output signals and power and ground connections for the 78690 Video Control 84-pin cerquad package. The pin assignments are identified in Figure 2 and the signals are summarized in Table 1.

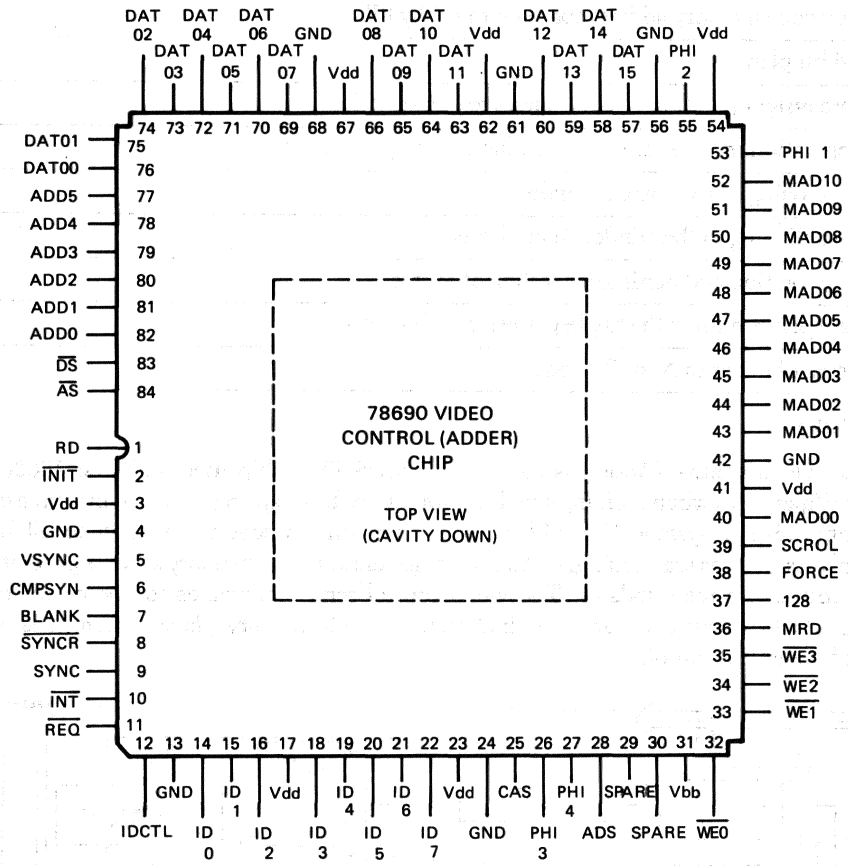


Figure 2 • 78690 Pin Assignments

Table 1 • 78690 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
1	RD	input	Read—Indicates a read or write processor bus cycle.
2	$\overline{\text{INIT}}$	input	Initialize—Initializes the video control and processor bus.
10	$\overline{\text{INT}}$	output	Interrupt—Initiates a processor interrupt request.
11	$\overline{\text{REQ}}$	output	Request—Indicates a DMA access request.
57-60 63-66 69-76	DAT<15:00>	input/output	Data <15:00>—Data bus lines
77-82	ADD<5:0>	input	Address <5:0>—Address bus lines
83	$\overline{\text{DS}}$	input	Data strobe—Enables data transfers on the internal or external data bus.
84	$\overline{\text{AS}}$	input	Address strobe—Initiates an interface transfer.
5	VSYNC	output	Vertical synchronize—A video vertical synchronization signal.
6	CMPSYN	output	Composite synchronize—A video composite horizontal synchronization signal.
7	BLANK	output	Blank—A video composite blank signal.
8	$\overline{\text{SYNCR}}$	output	Synchronize request—A system synchronization cycle request.
9	SYNC	input	Synchronize—Synchronizes the video subsystem.
12	IDCTL	output	ID control—Selects update or scroll select register.
14-16, 18-22	ID<7:0>	input/output	Instruction/Data <7:0>—Instruction/data bus lines.
25	CAS	input	Column address strobe—Bit-map memory address clock signal.
26	PHI3	input	Phase input 3—Phase 3 clock signal.
27	PHI4	input	Phase input 4—Phase 4 clock signal.
28	ADS	input	Address disable—Disables update activity.
29,30	Spare		No external connection recommended.
32-35	$\overline{\text{WE}}$<3:0>	output	Write enable—Enables the writing of the bit-map memory.
36	MRD	output	Map read—Specifies a read or write cycle of the bit-map memory.
37	128/$\overline{16}$	output	128/16—Indicates a major or minor bit-map memory cycle.

Pin	Signal	Input/Output	Definition/Function
38	FORCE	output	Force—Enables a down scrolling write cycle.
39	SCROL	output	Scroll—Selects refresh read memory cycles during scrolling.
40, 43-52	MAD<10:00>	outputs	Memory address <10:00>—Row and column address lines for bit-map memories.
53	PHI1	input	Phase input 1—Phase 1 clock signal.
55	PHI2	input	Phase input 2—Phase 2 clock signal.
31	V_{BB}		Not used.
3,62,67 17,23, 41,54	V_{DD}	input	Voltage—Power supply 5 Vdc.
4,61,68 13,24, 42,56	GND	input	Ground—Ground reference for signals and voltage.

Processor Interface Signals

Read/Write (RD)—This signal informs the video control that the processor access is a read or a write cycle. The processor access is initiated by the assertion of the $\overline{AS}$ signal.

Initialize ($\overline{INIT}$)—This signal causes the video control to be initialized to a known state. The processor interface becomes inactive forcing the ID bus drivers to a high-impedance state until the next SYNC signal is asserted.

Direct Memory Access Request ($\overline{REQ}$)—This signal provides a hardware flag for the bits that are enabled in the status register. The enabled bits are selected by the request register.

Interrupt ($\overline{INT}$)—This signal provides a hardware flag for the bits that are enabled in the status register. The enabled bits are selected by the interrupt register.

Data (DAT<15:00>)—These are bidirectional, parallel data lines, through which the video control interfaces to the processor. During a read operation, the bus master enables the $\overline{DS}$ input and the DAT<15:00> information drives the external data bus. During a write operation when the $\overline{DS}$ signal is asserted, the DAT<15:00> signals drive the internal data bus.

Address (ADD<5:0>)—These inputs select the video control register to be accessed.

Data Strobe ($\overline{DS}$)—During a register read cycle, this signal transfers data to the external data bus. During a register write cycle, it transfers data to the internal data bus.

Address Strobe ($\overline{AS}$)—This signal initiates a video control interfacetransfer. It performs a chip select function and latches the information into the addressed register.

Monitor Timing Signals

Vertical Synchronization (VSYNC)—This signal is a separate vertical SYNC signal for displays that do not use a composite SYNC signal.

Video Composite or Horizontal Synchronization (CMPSYN)—This signal can be programmed as either composite SYNC or horizontal SYNC signal.

Video Composite Blank Signal (BLANK)—The video control uses this signal to blank the monitor.

System Synchronization Request ($\overline{\text{SYNCR}}$)—The video control asserts this signal to request the external clock generation logic to generate a SYNC cycle. The clock logic asserts the SYNC signal and stops the clocks for a predetermined number of serial cycles.

System Synchronization (SYNC)—The external clock generation logic sends this signal to the video control and video processor to synchronize the video subsystem. This signal controls all of the timing generators on the chips.

ID Interface Signals

Chip Select Control (IDCTL)—The ID bus requires two chip select registers to allow independent selection of a video control for ID transactions. When the IDCTL signal is a low level, it selects the update chip selection register. When this signal is a high level, it selects the scroll chip select register.

Instruction/Data Bus (ID<7:0>)—This bus is a communications path between the video control and video processor.

Memory Interface Signals

Address Disable (ADS)—This signal is normally a low level to allow the video control to drive the memory interface. When this signal is high level, all update activity is stopped and all refresh and scroll activities continue. All update cycles on the bus are no operations (NOP) 16-bit read cycles.

Bitmap Memory Write Enable ($\overline{\text{WE}}$<3:0>)—These signals enable writing to groups of four bit map memories.

Memory Read/Write (MRD)—This signal indicates a read or a write cycle. A high level indicates a read cycle and a low level indicates a write cycle.

128/$\overline{16}$ Bit-map Memory Access Width—This signal indicates the type of cycle that the memory interface is about to initiate. A high level specifies the start of a 128-bit major cycle. A low level indicates the start of a 16-bit cycle consisting of two back-to-back minor cycles.

Force Signal for Down Scrolling Write Enables (FORCE)—This signal must occur during a down scroll to cause the write enable signals of the memory planes, with video processor chip plane enables, to write back the data during a scroll write-back cycle.

Scroll Enable (SCROL)—This signal indicates which refresh read memory cycles are to be written back for scrolling.

Bit-map Memory Address (MAD<10:00>—These lines provide the row and column addresses for the bit-map memories that are multiplexed from the video control. An address is available for each column address strobe (CAS) pulse.

Clock Signals

Phase Input 1 to Phase Input 4 (PHI1 to PHI4)—These clock inputs provide the timing control for all operations of the video control.

Column Address Strobe (CAS)—This input is a bit-map memory access strobe.

Power and Ground Connections

Power Supply (V_{DD})—Supplies 5-Vdc power to the 78690 video control.

Ground (GND)—Ground reference for all internal logic except for the output drivers.

• Architecture Summary

A typical bit-map processor system, shown in Figure 3, consists of a local processor or remote processor that performs DMA operations, the 78690 video control, the 78680 video processor, bit-map memory planes, the color map and shift register logic, and high-speed timing logic. The video control performs the functions that are common to memory planes which includes local processor interaction, raster operation computations, scan timing, system status generation, and memory address generation for screen refresh and updates.

The video control communicates with the microprocessor through the microprocessor bus and with the memory through a display memory address bus. It communicates with the video processors through the instruction/data bus.

The video control contains 64 registers that can be directly or indirectly loaded by the local processor to control the scan timing, refresh and scrolling operations. The registers are used to generate interrupts, report status, and to read and write data in the bit-map memory. They also communicate with the instruction/data bus.

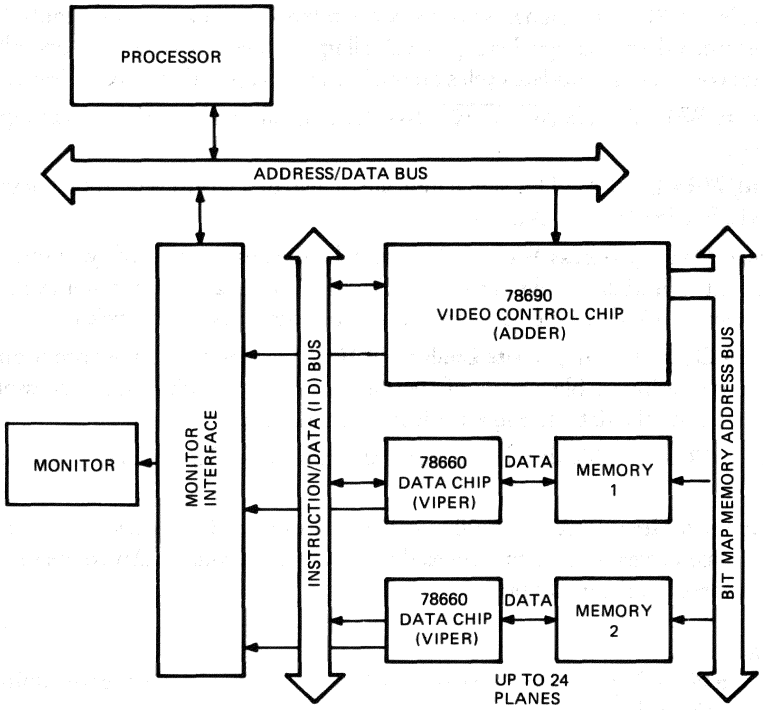


Figure 3 • 78690 Typical Bit-map Graphics System Configuration

The 78690 video control communicates with the processor through the data and address bus, with the bit map memory through the bit-map memory address bus, and with the 78680 video processors through the instruction/data (ID) bus.

The local processor sends the commands to the video control to update memory. The video control performs functions such as local processor interaction, raster computation operations, scan timing, system status generation, and memory address generation to refresh and update the display, that are common to all memory planes.

The display memory address bus transfers the addresses to update the display, to automatically refresh memory, and to scroll the information on the display.

The instruction/data bus loads the registers in the video processor with source, destination, barrel shift constant, and edge mask data. Data is also exchanged through this bus. During a screen update, the new information is transferred to the video processor. During a source operation, the video control sends a code specifying source 1 or source 2 and a barrel shift constant that shifts the source data to align with the destination. During a destination operation, the video control sends a destination code and an edge mask instructing the video processor that specifies which bits of the data bus are to be written.

Hardware Description

The following paragraphs provide a brief description of the major hardware block functions of the video control. Refer to Figure 1.

Processor Interface—This logic receives the parameters and commands from the local processor. The processor interface handles register accesses and controls timing and all necessary bus signals to allow the video control to act as a bus slave to the local processor or DMA device.

ID Control—The ID control selects the chip select registers that determine which video processors will update the bit-map memory and which ones will perform scrolling. The update process uses the ID data first-in/first-out (FIFO) buffer and the scroll process uses the IDS and ICS registers.

ID Data FIFO—During the bit-map memory update process, the video control uses the ID Data FIFO buffer to transfer data to and from the bit maps through the video processor. It is also used to load the video processor registers.

X and Y Scan Logic—The X-scan circuits generate the X components for the monitor display, the system synchronization signals, the refresh and scroll addresses, and the scroll enable signals. The Y-scan circuits generate the Y components for the monitor display, system synchronization signal, refresh and scroll addresses, and determines the active times for scrolling.

Sync Control—This circuit controls horizontal and vertical synchronization.

Serial Memories—The serial memory contains holds the commands and data for update operations.

Main Bresenham Control—This logic controls the serial logic that computes the address sequence for raster operations. It receives command and mode information from the serial memories and generates status bits and control flags for the main control and address collection subsections. A raster operation copies a contiguous section of bit-map memory (source) into a different location of bit-map memory (destination). The modes of raster operations supported are normal raster operations (rasterops), linear patterns, and polygon fill.

Source Generator—The source generator produces a sequence of addresses for the source data during raster operations. In normal mode, it traces a rectangular section of bit-map memory. In linear pattern mode, it loops on a small rectangle to create a repeating pattern until the destination completes its operation. In fill mode, it computes the address sequence for one edge of a polygon.

Destination Generator—The destination generator produces a sequence of addresses for the destination data during raster operations. In the normal and linear pattern modes, it traces a parallelogram by drawing a sequence of arbitrarily oriented parallel lines as the origins follow a trajectory defined by another arbitrarily oriented line. In fill mode, one part of the generator computes the address sequence of the second edge of the polygon and another part of the generator fills the polygon by drawing lines between the two polygon edges.

Source 2 Logic—This logic generates a second sequence of addresses for source data. The video processors may access data from two sources and combine it with the destination data. Source 2 data is generated by adding an offset value to the destination address. It can be used to generate a tile pattern that repeats the modulo of any power of 2 between 16 and 512.

Index Logic—The index logic permits an offset value to be added to the source and destination addresses to support the scrolling process. Scrolling is performed by physically moving data in the bit-map memory. When data is moved, the index may be changed so that the processor is not required to recompute its display list. The index logic allows data to be written into the bit-map memory during the scrolling process.

Clipping Logic—The clipping logic prevents the writing to the bit-map memory locations beyond a predefined rectangular window. Status is generated in the main control section to inform the host processor of the clipping action.

Barrel Shift Constant/Edge Mask Logic—This logic computes the data required by the video processors. The barrel shift constant indicates to the video processor the amount of the source data shift necessary to align it with the destination data. Each source operation generates a barrel shift constant. The edge mask logic defines the left and right edges of the destination. Each destination operation generates an edge mask.

Address FIFO Buffer—This buffer receives the serial logic signals for addresses and flags for destination update writes, source 1 update reads, and source 2 update reads. The FIFO multiplexes this data so it is available when needed.

Address Control—This logic controls the sequence of memory cycles on the memory interface including the refresh (read), scroll (write), and update (read-modify-write) cycles.

Write Enable Control—This logic controls the interface to the bit-map memories. The write enable logic generates the following:

- The timing signals to pass addresses from the RAS/CAS multiplexer to the memories
- The low 3 bits of the CAS addresses during major cycles
- The write enable signals for the memories during both scroll and update read-modify-write memory operations
- The SCROL, MRD and 128/$\overline{16}$ signal for memory
- The memory interface clocks and timing from the CAS clocks

RAS/CAS Multiplexer—This logic combines the X and Y address togenerate the row and column address for the bit-map memories. This multiplexer also assures the refreshing of the bit-map memories.

Programming Functions

When the local processor or DMA controller loads the command register, the video control executes one of the following commands:

Instruction/Data—The instruction/data commands are used to configure and control the video processor.

Raster—The raster commands are used to initiate raster operations as defined by the information previously loaded into registers.

Processor/Bit-map Transfer—These commands are used to to transfer data between the local processor or DMA controller and the video control.

Cancel—The cancel command terminates an update operation in progress and clears all raster operation status registers.

Modes of Operation

The 78690 video control operates in the following modes.

Normal Mode—This mode is used for raster operations when source 1 is a rectangle and the destination is a parallelogram, or source 2 is offset from the destination. Source 1 may be scaled up or down by a noninteger scale factor. When possible, the video control operates on the entire memory bus for operations. Otherwise, it operates on one pixel at a time. This action is transparent to the user.

Pattern Mode—This mode is used to generate dashed lines and thick patterned lines on the display. Scaling is possible in this mode because the source records its own size and is not dependent the destination. In this mode, the video control operates in slow mode.

Fill Mode—This mode is used to fill an area between two vectors. Both the slow source and destination create edge vectors. The fast destination draws either horizontal or vertical lines between the two edge vectors. However, the vectors may intersect each other. In addition, the area to a baseline may be filled so that one edge is a fixed Y value or a fixed X value. The video control uses fast mode for horizontal vectors and slow mode for vertical vectors and the modes can be used together with the fill mode.

Tile Mode—This mode is used to fill intersecting polygons so that the texture of each is continuous across the intersection and applies to source 2 only.

Register Descriptions

The 78690 video control contains 64 registers that can communicate with the processor data/address bus. Three of these registers are reserved for test purposes. The registers may be loaded directly or indirectly through the address counter (register 0) using an autoincrement mode. These registers contain the parameters for raster operations, to set the system timing, and to control the operation of the video control. The following rules apply:

- The registers are write-only unless otherwise specified.
- Local processor coordinates refer to images before adding index values.
- Device coordinates describe a physical position on the screen with 0, 0 being the upper-left corner with X increasing to the right and Y increasing downward.
- A DMA controller may load the registers so common combinations are adjacent thereby requiring a minimum of address register loads.

Table 2 • 78690 Register Address Assignments

Address*	Name	Width	Function
Control Registers			
0	Address counter (ADCT)	15:00	DMA device interface to registers
1	Request enable (REQ)	13:00	DMA status flag request enable
2	Interrupt enable (INT)	13:00	Interrupt enable for status flags
3	Status (STAT)	13:00	Video system status
4	Reserved for test		
5	Spare		
6	Reserved for test		
7	ID data (IDD)	15:00	ID bus data
8	Command (CMD)	14:00	Command register (same as command register location A)
9	Mode (MDE)	07:00	Sets various raster operation execution modes
A	Command (CMD)	14:00	Command register
Scroll Registers			
B	Reserved for test		
C	ID scroll data (IDS)	15:00	ID bus scroll data
D	ID scroll command (ICS)	15:00	ID command register for scroll process
E	Scroll X Minimum (PXMN)	13:00	Left boundary of scroll region
F	Scroll X Maximum (PXMX)	13:00	Right boundary of scroll region
10	Scroll Y Minimum (PYMN)	13:00	Top boundary of scroll region
11	Scroll Y Maximum (PYMX)	13:00	Bottom boundary of scroll region
12	Pause (PS)	10:00	Screen coordinate to set pause status
13	Y offset (PYOF)	13:00	Screen to memory coordinate offset
14	Y scroll constant (PYSC)	14:00	Vertical scroll distance in one frame
Update Control Registers			
15	Pending X index (PXI)	13:00	Pending X index
16	Pending Y index (PYI)	13:00	Pending Y index
17	New X index (NXI)	13:00	New X index
18	New Y index (NYI)	13:00	New Y index
19	Old X index (OXI)	13:00	Old X index
1A	Old Y index (OYI)	13:00	Old Y index
1B	Clip X minimum (CXMN)	13:00	Left clipping boundary
1C	Clip X maximum (CXMX)	13:00	Right clipping boundary
1D	Clip Y minimum (CYMN)	13:00	Top clipping boundary
1E	Clip Y maximum (CYMX)	13:00	Bottom clipping boundary
1F	Spare		

Address*	Name	Width	Function
Raster Control Registers			
20	Fast source 1 dX (FSDX)	13:00	Fast delta X for source 1
21	Slow source 1 dY (SSDY)	13:00	Slow delta Y for source 1
22	Source 1 X origin (SXO)	13:00	X coordinate of source 1
23	Source 1 Y origin (SYO)	13:00	Y coordinate of source 1
24	Destination X origin (DXO)	13:00	X coordinate of destination origin
25	Destination Y origin (DYO)	13:00	Y coordinate of destination origin
26	Fast destination dX (FDX)	13:00	X component of fast destination vector
27	Fast destination dY (FDY)	13:00	Y component of fast destination vector
28	Slow destination dX (SDX)	13:00	X component of slow destination vector
29	Slow destination dY (SDY)	13:00	Y component of slow destination vector
2A	Fast scale (FSC)	13:00	Fast vector scale factor
2B	Slow scale (SSC)	13:00	Slow vector scale factor
2C	Source 2 X origin (S2XO)	13:00	X coordinate of source 2
2D	Source 2 Y origin (S2YO)	13:00	Y coordinate of source 2
2E	Source 2 Height and Width (S2HW)	07:00	Size of source 2 tile
2F	Error 1 (ERR1)	13:00	Error adjust for slow destination
30	Error 2 (ERR2)	13:00	Error adjust for fast destination
Screen Format Control Registers			
31	Y scan count 0 (YCT0)	13:00	Vertical timing
32	Y scan count 1 (YCT1)	13:00	Vertical timing
33	Y scan count 2 (YCT2)	13:00	Vertical timing
34	Y scan count 3 (YCT3)	13:00	Vertical timing
35	X scan configuration (XCON)	08:00	Cycles, bus width, number of refresh rows
36	X limit (XL)	13:00	Width limit on refresh
37	Y limit (YL)	13:00	Height limit on refresh
38	X scan count 0 (XCT0)	15:00	X scan count 0
39	X scan count 1 (XCT1)	15:00	X scan count 1
3A	X scan count 2 (XCT2)	15:00	X scan count 2
3B	X scan count 3 (XCT3)	15:00	X scan count 3
3C	X scan count 4 (XCT4)	15:00	X scan count 4
3D	X scan count 5 (XCT5)	15:00	X scan count 5
3E	X scan count 6 (XCT6)	15:00	X scan count 6
3F	Sync phase (SYNP)	14:50	Sync phase adjustment

*Hexadecimal notation

Status and Control Registers

The video control logic contains status and control registers used to initiate requests and interrupts, report status, select modes of operation, and initiate command functions. These registers are described in the following paragraphs. The status and control register formats are shown in Figure 4.

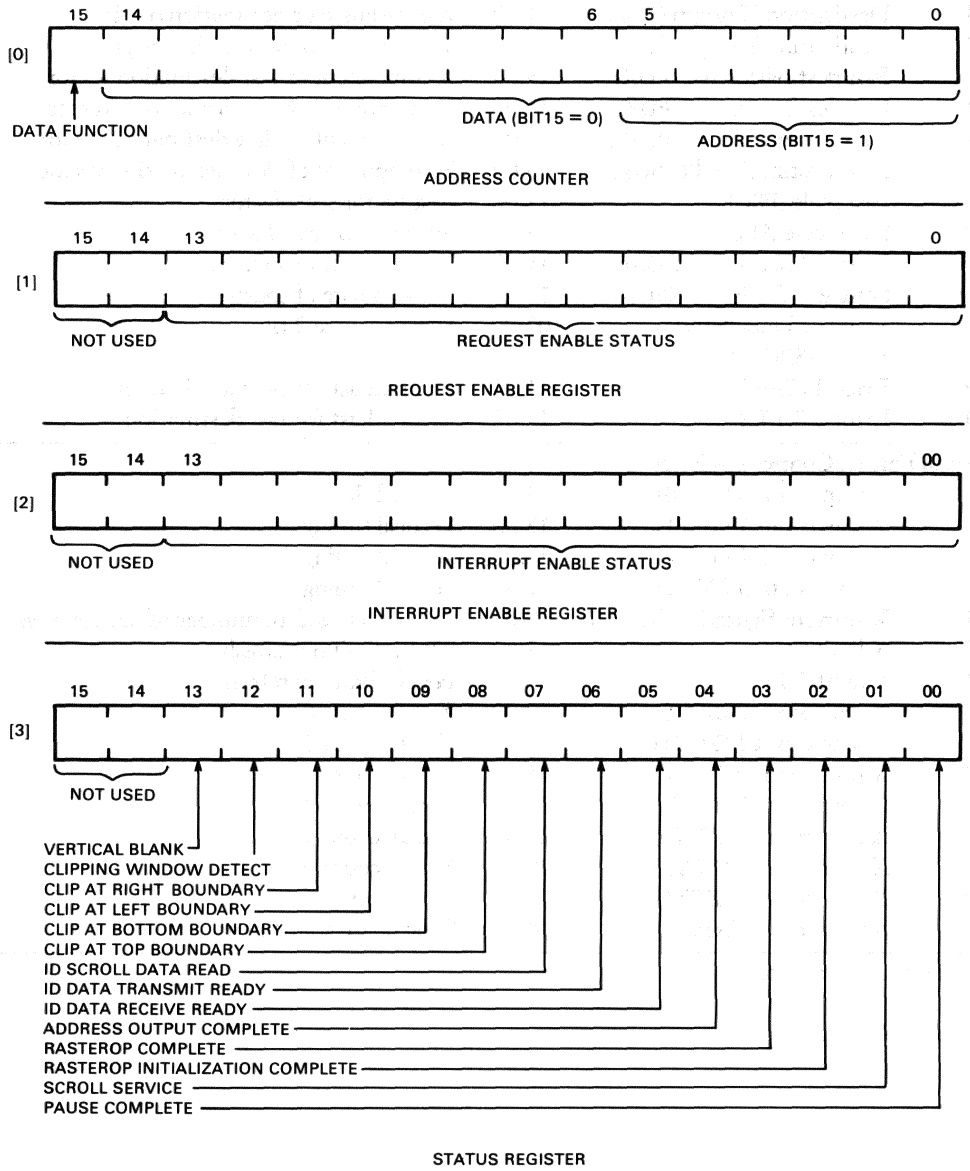


Figure 4 ▪ 78690 Status and Control Register Formats

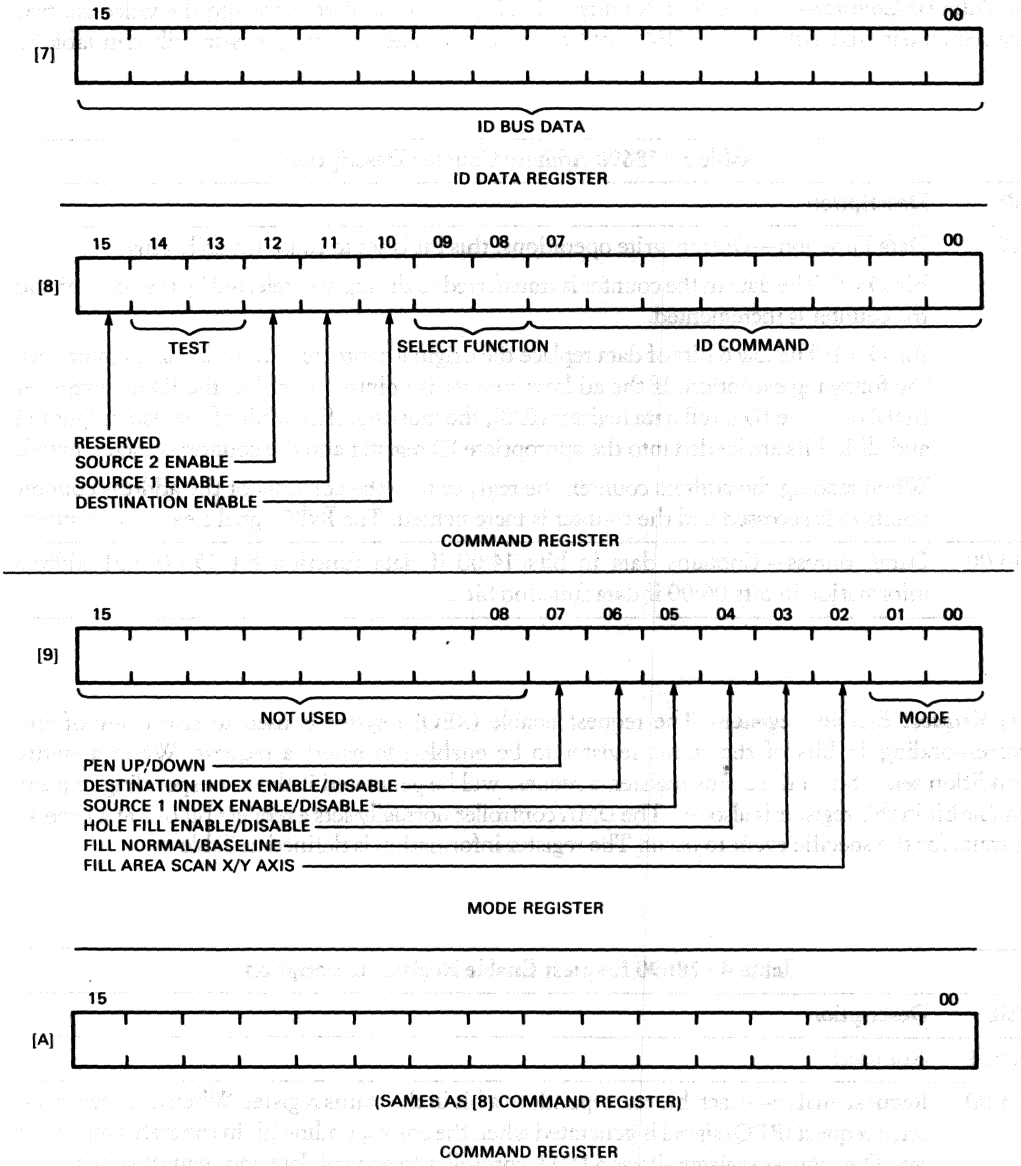


Figure 4 ▪ 78690 Status and Control Register Formats (Continued)

[0] Address Counter—The address counter (ADCT) provides indirect access to the video control registers and is used with standard DMA controllers. The register information is described in Table 3.

Table 3 • 78690 Address Counter Description

Bit	Description
15	Data Function—During write operations, this bit is set to cause the following: Bit 15 = 0: The data in the counter is transferred to the register selected by the counter and the counter is incremented. Bit 15 = 1: The low 6 bits of data replace the original contents of the address counter with the following exception. If the address counter is pointing to either the ID data register (IDD) or to the ID scroll data register (IDS), the most significant bit of the data is ignored and all 16 bits are loaded into the appropriate ID register and the counter is incremented. When reading the address counter, the register that the contents of the address counter points to is accessed and the counter is incremented. The $\overline{\text{INIT}}$ signal clears the counter.
14:00	Data/Address—Contains data in bits 14:00 if data function bit 15 = 0 and address information in bits 06:00 if data function bit 15 = 1.

[1] Request Enable Register—The request enable (REQ) register is used to select any of the corresponding 13 bits of the status register to be enabled to assert a request. When a status condition sets a bit in the status register, a request will be generated if the corresponding request enable bit in this register is also set. The DMA controller normally sets a request bit one at a time as it waits for the specific event to occur. The register information is defined in Table 4.

Table 4 • 78690 Request Enable Register Description

Bit	Description
15:14	Not used
13:00	Request enable—Each bit corresponds to a bit in the status register. When a request bit is set, a request (REQ) signal is generated when the corresponding bit in the status register is set. The request register allows a DMA controller to control data and request status.

[2] Interrupt Enable Register—The interrupt (INT) enable register is used by the local processor to select any of the corresponding 13 bits of the status register. When a status condition sets a bit in the status register, an interrupt will be generated if the corresponding interrupt enable bit in this register is also set. The register information is defined in Table 5.

Table 5 • 78690 Interrupt Enable Register Description

Bit	Description
15:14	Not used
13:00	Interrupt enable (INT)—Each bit corresponds to a bit in the status register. When a interrupt enable bit is set, an interrupt request (INT) signal is generated when the corresponding status condition causes the status bit to be set. The interrupt enable register provides the local processor with interrupt conditions.

[3] Status Register—The status (STAT) register is a read-only register that provides indications of the internal progress of the video control. The register information is described in Table 6.

Table 6 • 78690 Status Register Description

Bit	Description
15:14	Not used
13	Vertical blank—This bit is set at the start of the vertical blank interval and cleared by writing 0 to this bit position.
12	Clipping window detect—Set during a destination write cycle to indicate that part of a raster operation was inside the clipping rectangle. Cleared by writing a zero to this bit position.
11	Clipped right boundary—Set during a destination write cycle to indicate that part of a raster operation was clipped at the right boundary. Cleared by writing a zero to this position.
10	Clipped left boundary—Set during a destination write cycle to indicate that part of a raster operation was clipped at the left boundry. Cleared by writing a zero to this position.
09	Clipped bottom boundary—Set during a destination write cycle to indicate that part of a raster operation was clipped at the bottom boundary. Cleared by writing a zero to this position.
08	Clipped top boundary—Set during a destination write cycle to indicate that part of a raster operation was clipped at the top boundary. Cleared by writing a zero to this position.
07	ID scroll data—Set by the video control when a new data word can be loaded into the ID scroll data register. Cleared by the ID command to the ID scroll command register.

Table 6 • 78690 Status Register Description

06	ID data transmit ready—Set to indicate that data can be loaded into the ID data register during an ID command. Cleared when the new ID data command is loaded. This bit is also set at the completion of a Cancel command to indicate that the ID FIFO buffer is clear to load another command. The Cancel command initially clears this bit. When a raster operation or PBT command is not in progress, loading the ID data register or asserting the $\overline{\text{INIT}}$ signal will set this bit.
05	ID data receive ready—Set when the ID data FIFO buffer has a word to be read. This bit is cleared when the FIFO buffer is empty; when a raster operation, PBT, or cancel command is loaded into the command register; or when bus initialization occurs.
04	Address output complete—Set when all addresses calculated by pending raster operations or PBT commands have been used indicating that update parameters such as clipping boundaries, indexes, ID data, or other commands can be loaded. During a bit-map to processor (BTP) commands, this bit is set when the IDD FIFO buffer is empty. It is also set by the cancel command or $\overline{\text{INIT}}$ signal. Cleared by loading any raster operation or PBT command.
03	Raster operation complete—Set at the completion of raster operations or PBT address calculation when no further command is pending. It indicate that other raster operations can be loaded such as dX and dY pairs, source 2, or scale factor, that the mode register can be loaded, or that a new but not different raster operation can be initiated. It is also set by the Cancel command or $\overline{\text{INIT}}$ signal. Cleared by loading any raster operation or PBT command.
02	Raster operation initialization complete—Set at the completion of the initialization of a raster operation or a processor-to-bit-map transfer. It indicate that the source 1 origin, the destination origin, or a new but not different raster operation command (except PBT command) can be loaded. It is also set by a Cancel command or the $\overline{\text{INIT}}$ signal. Cleared by loading a raster operation or PBT command.
01	Scroll service (frame sync)—Set at the start of frame when new scroll parameters can be loaded. Cleared by writing a zero to this bit.
00	Pause complete—Set when the screen refresh process reaches the Y address of device coordinates in the pause register. Cleared by writing a zero to this bit.

[4] Test Register—Reserved for test purposes.

[5] Spare Register—Not used.

[6] Test Register—Reserved for test purposes.

[7] ID Data Register—The ID data (IDD) register is the ID data bus port from the six-word FIFO buffer. During PBI commands, data is transferred between the processor and bit-map memory through this register.

[8] Command Register—The command (CMD) register is used for all commands by the update process and can be accessed from either address [8] or [A] (hexadecimal). The register information is described in Table 7.

Table 7 • 78690 Command Register Description

Bit	Description
15:13	NOP (No operation)—Reserved and test functions as follows: Bit 15: Reserved and normally zero Bits 14:13: Test and normally zero
12	S2E (Source 2 enable)—Second source enable
11	S1E (Source 1 enable)—First source enable
10	DTE (Destination enable)—Enable the destination
09:80	(FUNC) Select Function—Selects the function of the command as follows: **Bit 09 / Bit 08 / Function** 0 0 Cancel all active and pending commands 0 1 ID command 1 0 Raster operation command 1 1 Processor bit-map (PBT) transfer command
07:00	ID command—Contains the opcode and address of the ID command.

[9] Mode Register—The mode (MDE) register sets the raster operation execution modes. The register information is described in Table 8.

Table 8 • 78690 Mode Register Description

Bit	Description
15:08	Not used
07	Pen up/dn—Selects the pen position as follows: Bit 7 = 0: pen up to disable writing Bit 7 = 1: pen down to enable writing
06	Destination indexing—Controls the indexing of the destination as follows: Bit 6 = 0: disable Bit 6 = 1: enable
05	Source 1 indexing—Controls the indexing of source 1 as follows: Bit 5 = 0: disable Bit 5 = 1: enable
04	Hole fill—Controls the hole fill operation as follows: Bit 4 = 0: disable (normal single pixel wide destination) Bit 4 = 1: enable (all other destinations)

Bit	Description
03	Fill area—Selects the fill area as follows: Bit 3 = 0: Normal two-edge fill Bit 3 = 1: Fill to a vertical or horizontal base line depending on the scan direction of the previous bit.
02	Fill area scan axis—Selects the fill area to be scanned as follows: Bit 2 = 0: Scanned parallel to the X axis Bit 2 = 1: Scanned parallel to the Y axis
01:00	Mode—Selects the operating mode as follows:

Bit 01	Bit 00	Mode
0	0	Normal: Source 1 is a scaled destination area.
0	1	Reserved
1	0	Linear pattern: Source 1 is from the dX and dY registers.
1	1	Fill: Destination slow generator computes the "A" edge vector. Source 1 generator computes the "B" edge vector and the video control fills the space between the vectors.

[A] Command Register—The functions of this command (CMD) register are the same as the [8] Command (CMD) register.

Scroll Registers

The scroll registers determine the scroll activity on the display. During active scrolling operations, the scroll registers must be loaded by the scroll process before the start of the vertical blanking. The video processor performs various functions during the vertical blanking depending on the type of scroll activity pending for the next frame. The scroll register formats are shown in Figure 5.

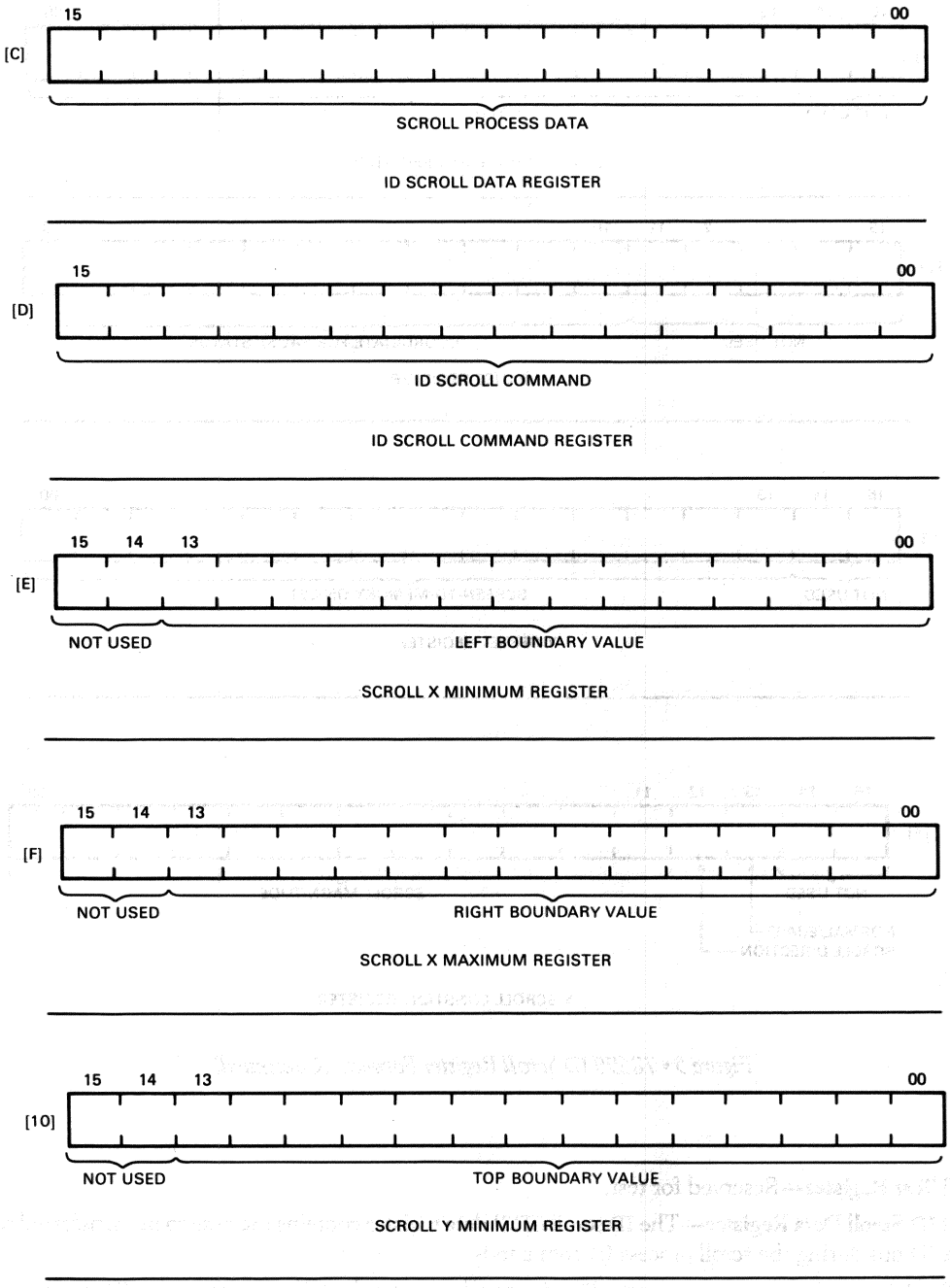


Figure 5 ▪ 78690 ID Scroll Register Formats

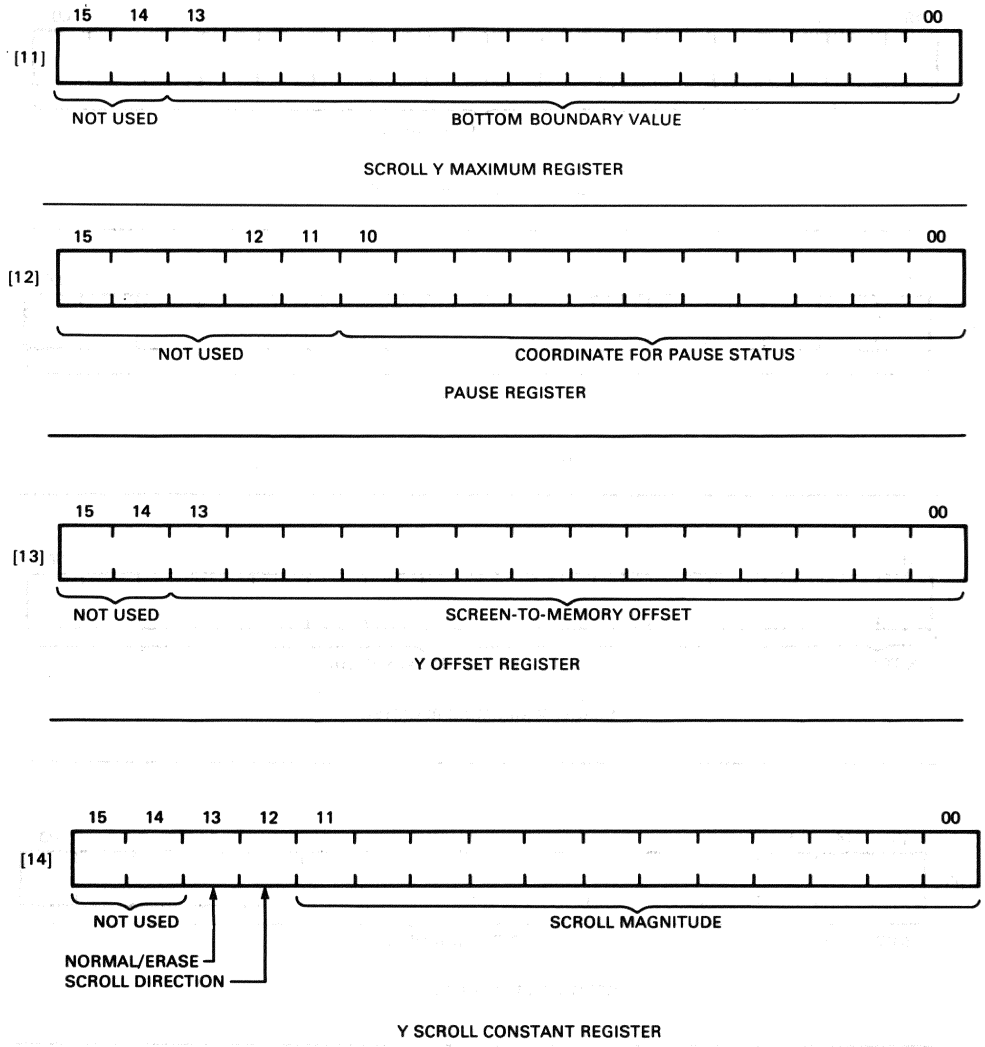


Figure 5 • 78690 ID Scroll Register Formats (Continued)

[B] Test Register—Reserved for test.

[C] ID Scroll Data Register—The ID scroll (IDS) data register contains the data to be transferred to the ID bus during the scroll process ID commands.

[D] ID Scroll Command Register—The ID scroll command (ICS) register is the ID command register for the scroll process. Commands are transferred through this register without interfering with update activities.

[E] Scroll X Minimum Register—The Scroll X minimum (PXMN) register information determines the left boundary of the scroll region and specifies the left-most pixel. The X boundary may only be specified to a multiple of four pixels. The scroll boundaries are not affected by index values and the register is double buffered so that the values loaded become active at the start of the following frame.

[F] Scroll X Maximum Register—The Scroll X maximum (PXMX) register information determines the right boundary of the scroll region. The scroll boundaries are not affected by index values and the register is double buffered so that the values loaded become active at the start of the following frame.

[10] Scroll Y Minimum Register—The Scroll Y minimum (PYMN) register information determines the the top boundary of the scroll region. The scroll boundaries are not affected by index values and the register is double buffered so that the values loaded become active at the start of the following frame.

[11] Scroll Y Maximum Register—The Scroll Y maximum (PYMX) register informatiom determines the bottom boundary of the scroll region. The scroll boundaries are not affected by index values and the register is double buffered so that the values loaded become active at the start of the following frame.

[12] Pause Register—The pause (PSE) register contains a value that specifies which scan, when displayed, will cause the pause complete (bit 0) of the status register to be set or a second pause event to be queued. This register is double buffered so that the new value loaded begins at the start of the following frame and continues through the frame.

[13] Y Offset Register—The Y Offset (PYOF) register contains the value when added to the device coordinates become the memory coordinate ranging from 0 to height of display portion of the bitmap memory. This value is the same as the value stored in the Y limit register minus 1. The register is double buffered so that the values loaded become active at the start of the following frame.

[14] Y Scroll Constant Register—The Y Scroll Constant (PYSC) register specifies the magnitude and direction of the vertical scroll in one frame time. The vertical distance specified by the value in this register and the horizontal distance specified by the value in the scroll constant register of the 78680 video processor will be scrolled. The register information is described in Table 9.

Table 9 • 78690 Y Scroll Constant Register Description

Bit	Description
15:14	Not used
13	Normal/Erase—Specifies the scroll condition as follows: Bit 13 = 0: Normal scrolling Bit 13 = 1: Erase mode
12	Scroll direction—Specifies the scrolling direction as follows: Bit 12 = 0: Up, left, or right scrolling Bit 12 = 1: Down scrolling.
11:00	Magnitude—Determines the unsigned vertical magnitude of the scroll.

Update Control Registers

The update control registers, shown in Figure 6, contain index values that are added to the raster operation addresses to adjust the addresses for scrolling and to specify location of the regions on the display. The pending values are automatically loaded into the new registers at the start of the next frame. If no scrolling takes place, the loading of the registers is not required. The pending register values are loaded first, followed by the new values, and then the old values.

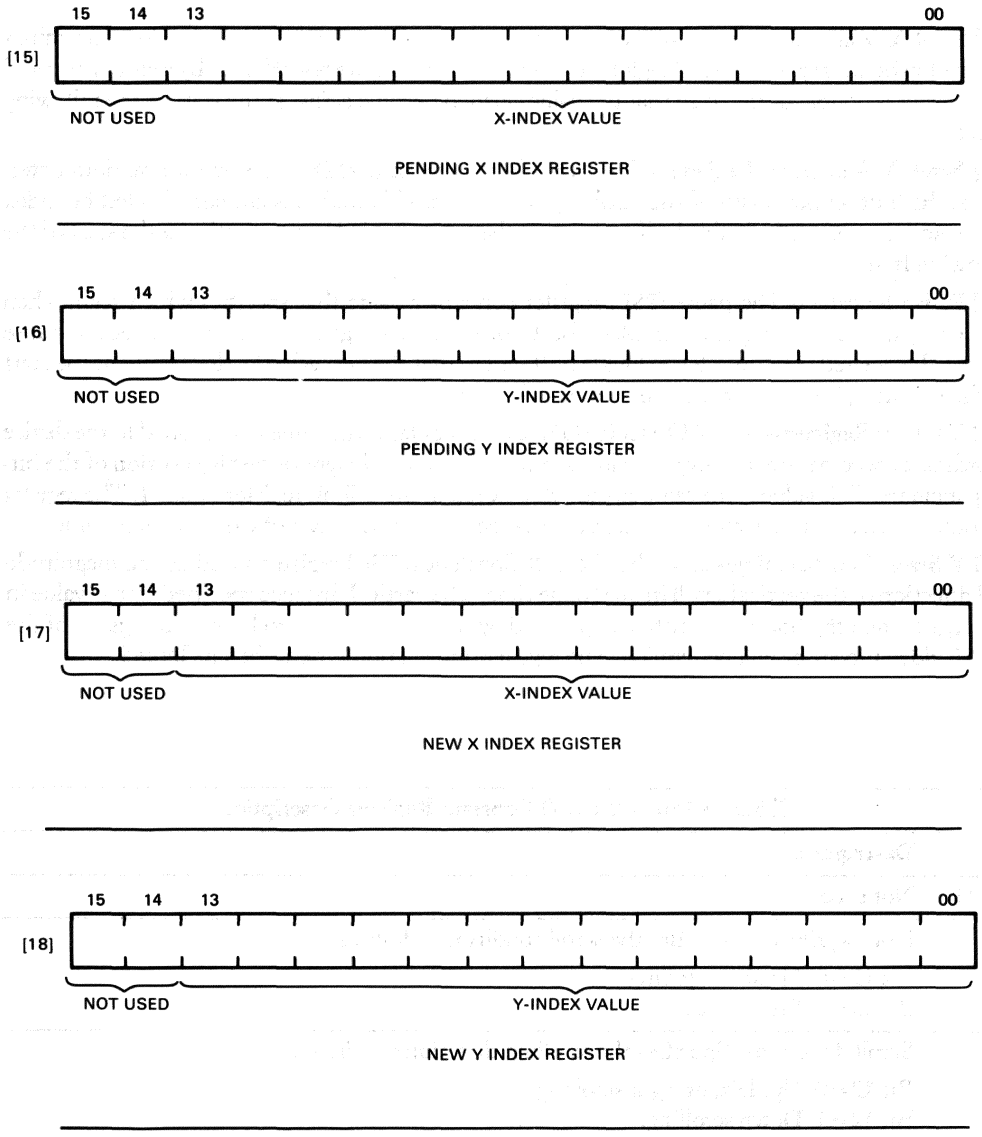


Figure 6 • 78690 Update Control Register Format

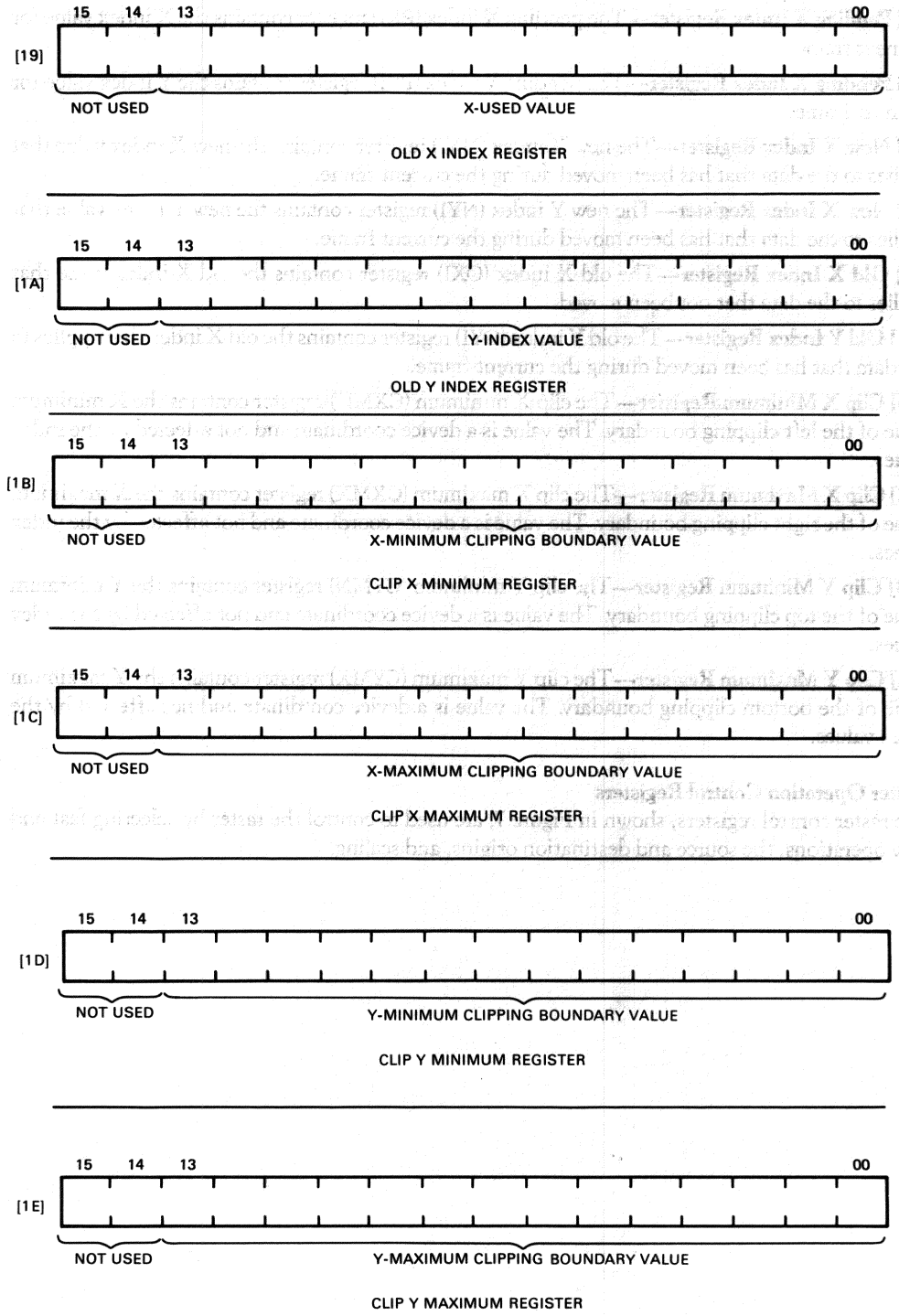


Figure 6 ▪ 78690 Update Control Register Format (Continued)

[15] Pending X Index Register—The pending X index (PXI) register contains the X index value for the next frame.

[16] Pending X Index Register—The pending Y index (PYI) register contains the Y index value for the next frame.

[17] New X Index Register—The new X index (NXI) register contains the new X index value that applies to the data that has been moved during the current frame.

[18] New X Index Register—The new Y index (NYI) register contains the new Y index value that applies to the data that has been moved during the current frame.

[19] Old X Index Register—The old X index (OXI) register contains the old X index value that applies to the data that not been moved.

[1A] Old Y Index Register—The old Y index (OYI) register contains the old X index that applies to the data that has been moved during the current frame.

[1B] Clip X Minimum Register—The clip X minimum (CXMN) register contains the X minimum value of the left clipping boundary. The value is a device coordinate and not affected by the index values.

[1C] Clip X Maximum Register—The clip X maximum (CXMX) register contains the X maximum value of the right clipping boundary. The value is a device coordinate and not affected by the index values.

[1D] Clip Y Minimum Register—The clip Y minimum (CYMN) register contains the Y minimum value of the top clipping boundary. The value is a device coordinate and not affected by the index values.

[1E] Clip Y Maximum Register—The clip Y maximum (CYMX) register contains the Y maximum value of the bottom clipping boundary. The value is a device coordinate and not affected by the index values.

Raster Operation Control Registers

The raster control registers, shown in Figure 7, are used to control the raster by selecting fast and slow operations, the source and destination origins, and scaling.

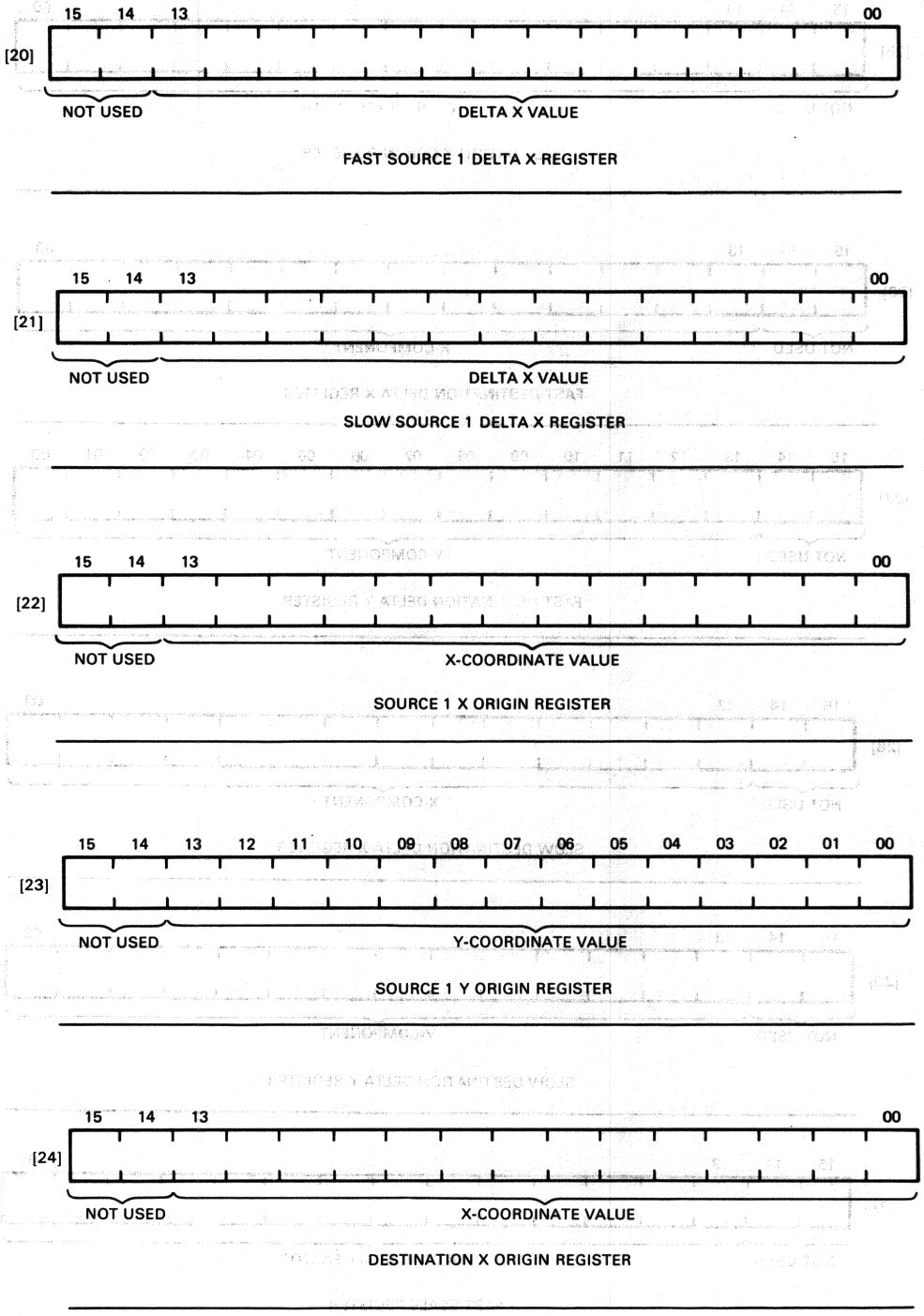


Figure 7 ▪ 78690 Raster Operation Control Register Format

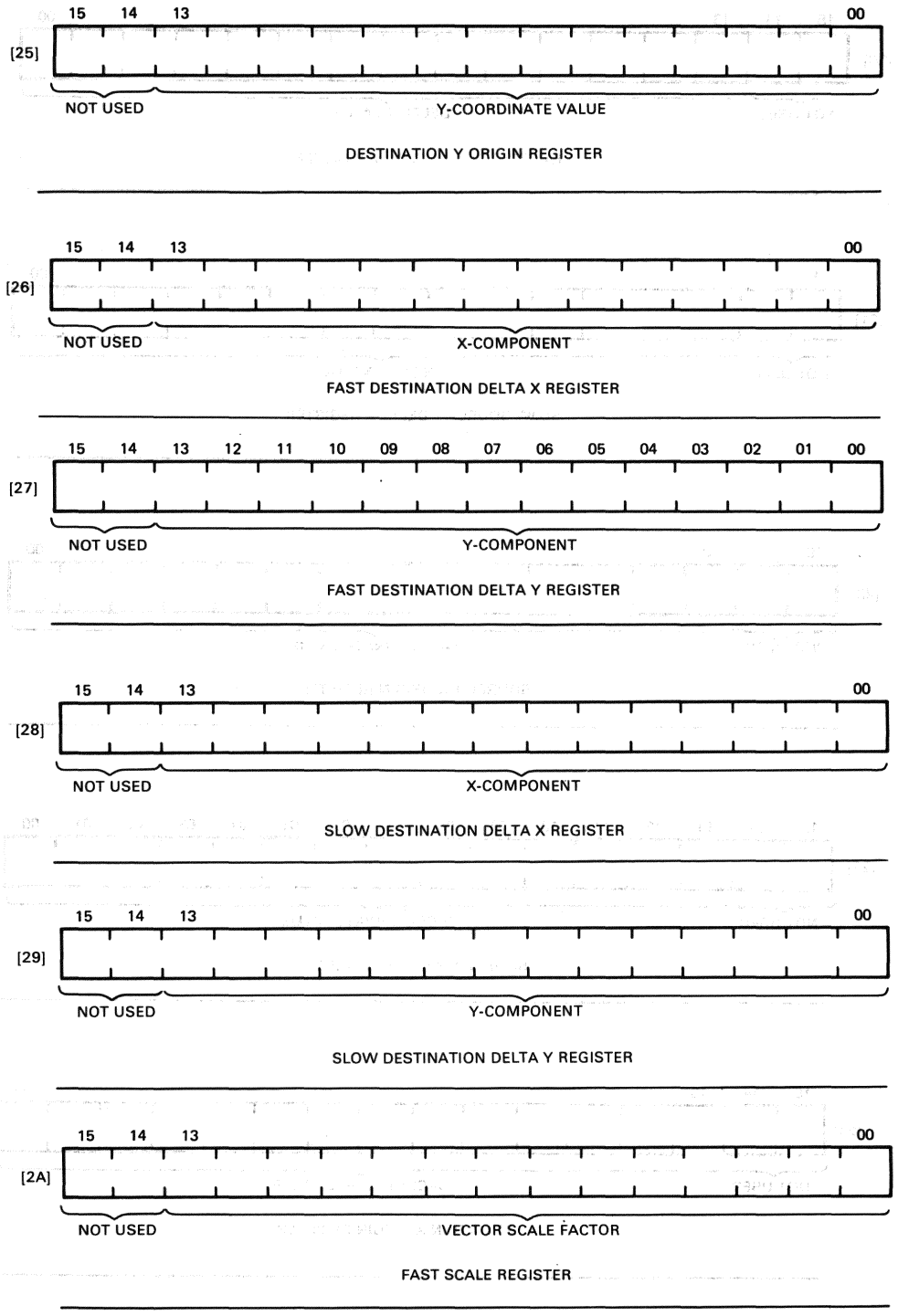


Figure 7 ▪ 78690 Raster Operation Control Register Format (Continued)

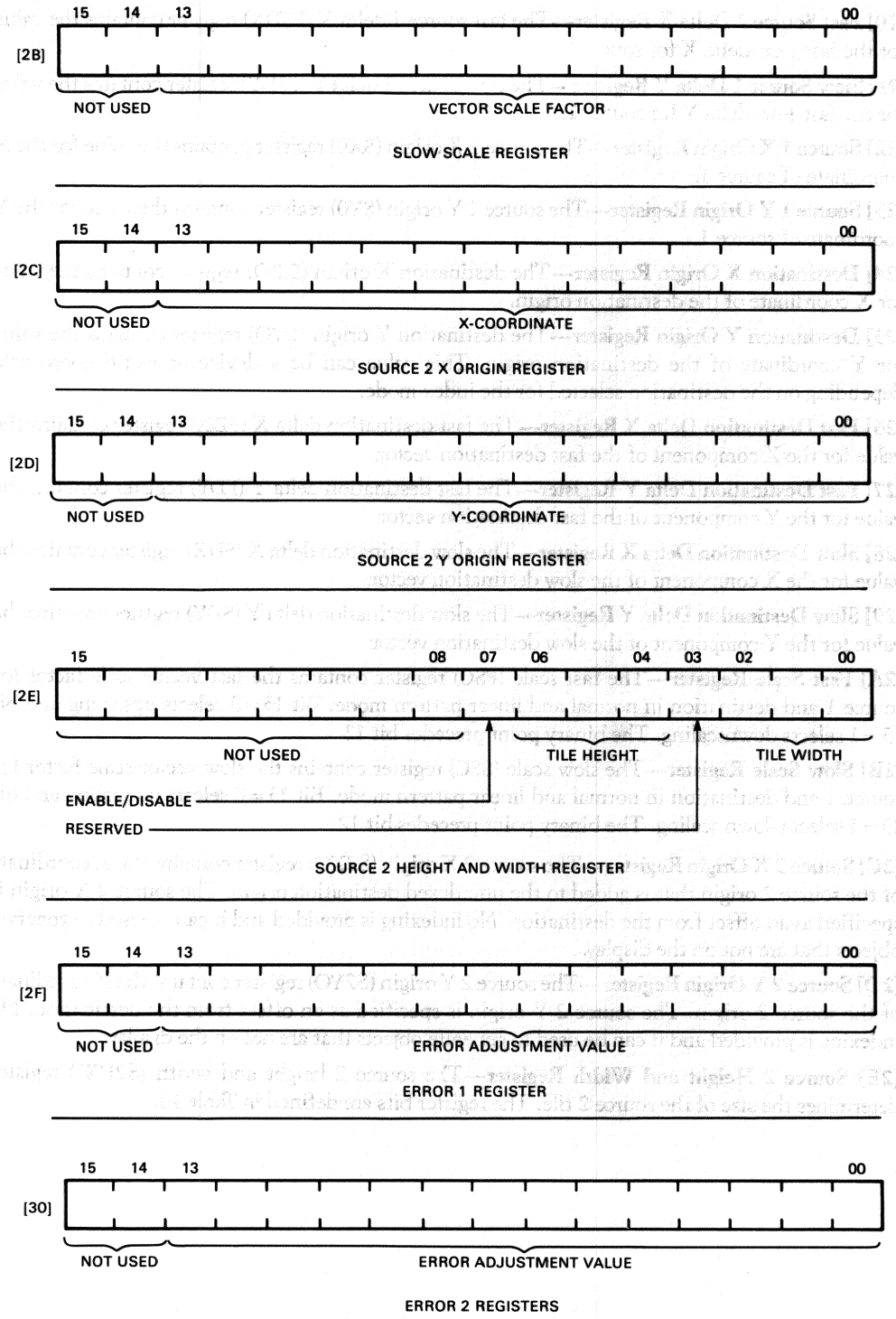


Figure 7 • 78690 Raster Operation Control Register Format (Continued)

[20] Fast Source 1 Delta X Register—The fast source 1 delta X (FSDX) register contains the value for the fast + or–delta X for source 1.

[21] Slow Source 1 Delta Y Register—The slow source 1 delta Y (SSDY) register contains the value for the fast + or–delta Y for source 1.

[22] Source 1 X Origin Register—The source 1 X origin (SX0) register contains the value for the X coordinate of source 1.

[23] Source 1 Y Origin Register—The source 1 Y origin (SY0) register contains the value for the Y coordinate of source 1.

[24] Destination X Origin Register—The destination X origin (DXO) register contains the value for X coordinate of the destination origin.

[25] Destination Y Origin Register—The destination Y origin (DYO) register contains the value for Y coordinate of the destination origin. This value can be a device or world coordinate depending on the destination selected for the index mode.

[26] Fast Destination Delta X Register—The fast destination delta X (FDX) register contains the value for the X component of the fast destination vector.

[27] Fast Destination Delta Y Register—The fast destination delta Y (FDY) register contains the value for the Y component of the fast destination vector.

[28] Slow Destination Delta X Register—The slow destination delta X (SDX) register contains the value for the X component of the slow destination vector.

[29] Slow Destination Delta Y Register—The slow destination delta Y (SDY) register contains the value for the Y component of the slow destination vector.

[2A] Fast Scale Register—The fast scale (FSC) register contains the fast vector scale factor for source 1 and destination in normal and linear pattern mode. Bit 13 = 0 selects upscaling and bit 13 = 1 selects downscaling. The binary point precedes bit 12.

[2B] Slow Scale Register—The slow scale (SSC) register contains the slow vector scale factor for source 1 and destination in normal and linear pattern mode. Bit 13 = 0 selects up scaling and bit 13 = 1 selects down scaling. The binary point precedes bit 12.

[2C] Source 2 X Origin Register—The source 2 X origin (S2XO) register contains the X coordinate of the source 2 origin that is added to the unindexed destination origin. The source 2 X origin is specified as an offset from the destination. No indexing is provided and it can be used to generate objects that are not on the display.

[2D] Source 2 Y Origin Register—The source 2 Y origin (S2YO) register contains the X coordinate of the source 2 origin. The source 2 Y origin is specified as an offset from the destination. No indexing is provided and it can be used to generate objects that are not on the display.

[2E] Source 2 Height and Width Register—The source 2 height and width (S2HW) register determines the size of the source 2 tile. The register bits are defined in Table 10.

Table 10 ▪ 78690 Source 2 Height and Width Register Description

Bit	Description
15:08	Not used.
07	Destination address bit function—Selects the destination address bits as follows: Bit 7 = 0: High bits of destination are truncated before adding to source 2 origin. Bit 7 = 1: All destination address bits are added to source 2 origin.
06:04	Tile height (H)—Selects the tile height. H = 0 to 7 which is $2^{(H+2)}$ from 4 to 512.
03	Reserved
02:00	Tile width (W)—Selects the tile width. W = 0 to 7 which is $2^{(W+2)}$ from 4 to 512. The tile width must not be set to less than the bus width.

[2F] Error 1 Register—The error 1 (ERR1) register contains the error adjustment vector added during raster initialization for the slow destination (B side) fill mode.

[2F] Error 2 Register—The error 2 (ERR2) register contains the error adjustment vector added during raster initialization for the fast destination in normal and linear pattern mode and for the source 1 (B side) in fill mode.

Screen Format Control Registers

The screen format control registers, shown in Figure 8, select the vertical and horizontal timing events, number of read cycles, bus width, and refresh rows.

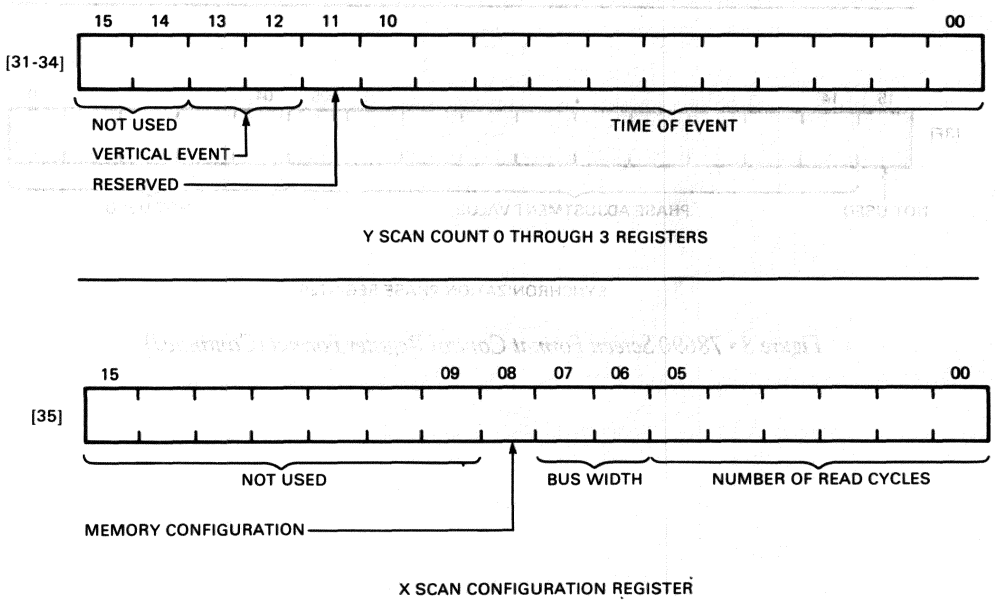


Figure 8 ▪ 78690 Screen Format Control Register Format

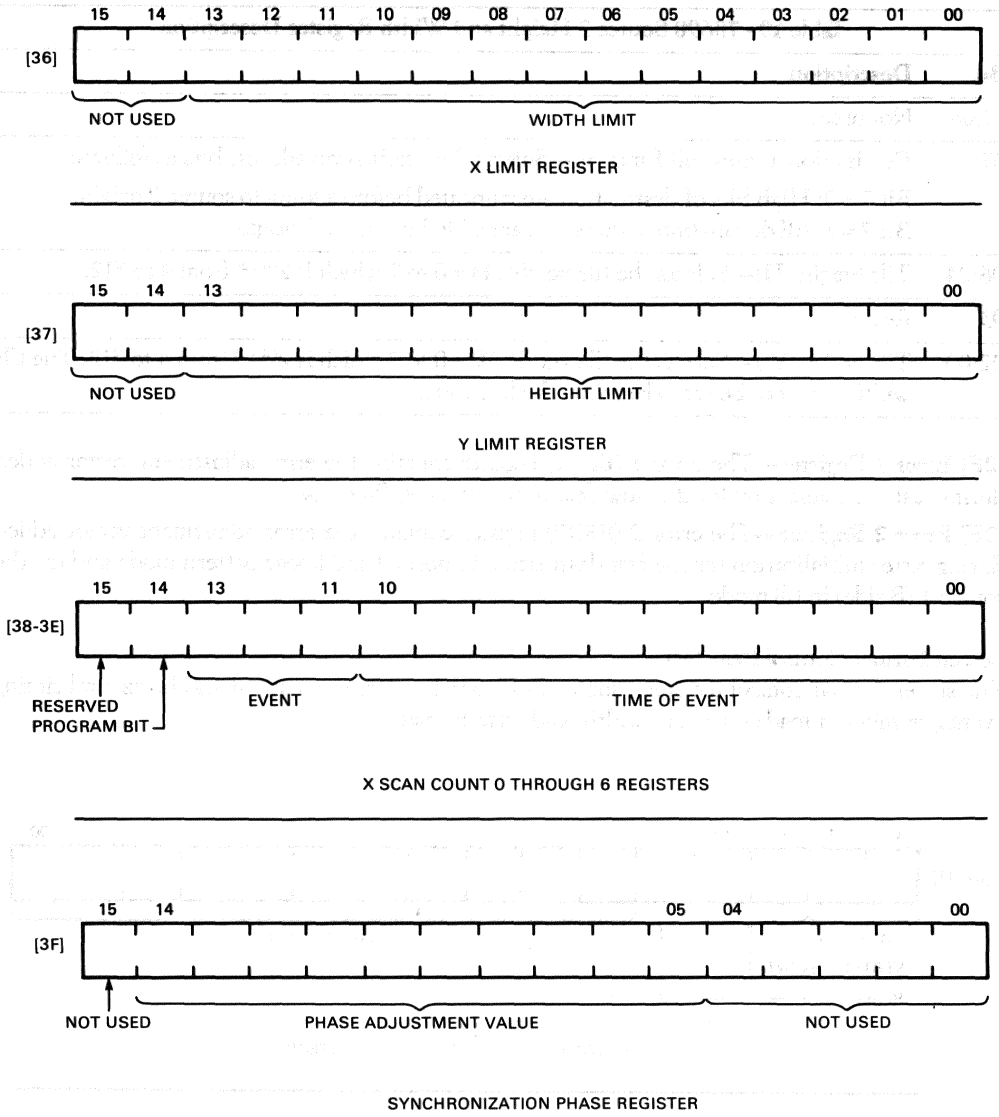


Figure 8 ▪ 78690 Screen Format Control Register Format (Continued)

[31-34] Y Scan Count Registers (0-3)—The Y scan count (YCT0 through YCT3) registers are used to program the vertical events. Each register determines the time for one vertical event, such as vertical blank time, in order of increasing time. If the horizontal period is an odd number of major cycles, the vertical period must be set for an even number of scans in a frame. Table 11 describes the function of the register information.

Table 11 • 78690 Y Scan Count Registers (0-3) Description

Bit	Description
15:14	Not used
13:12	Vertical event—Selects the vertical sync and blank events as follows: **Bit 13 / Event 12** 0 0 End vertical period. Set vertical blank low in the following scan. 0 1 Set vertical blank high. 1 0 Set vertical sync low. 1 1 Set vertical sync high. Example: YCT0 sets vertical blank high, YCT1 sets vertical sync high, YCT2 sets vertical sync low, and YCT3 sets vertical blank low and restarts the vertical counter. The video control generates system sync request in the scan prior to deasserting vertical blank. The X scan registers determine the exact timing.
11	Reserved (must be zero)
10:00	Time of event—Determines the time of the event from the deassertion of vertical blank (in scans), except for vertical period, which is set to the number of scans in a frame minus one.

[35] X Scan Configuration Register—The X scan configuration (XCON) register determines the bus widths, the number of read cycles, and the memory configuration. The register information is described in Table 12.

Table 12 • 78690 X Scan Configuration Register Description

Bit	Description
15:09	Not used
08	Memory configuration—Controls the number of row addresses refreshed on each scan.
07:06	Bus width—This mode is programmed in the video control and video processor before a bitmap memory access is performed as follows:

Bits 7	Bits 6	Bus width
0	0	4-bit
0	1	8-bit
1	0	undefined
1	1	16-bit

05:00	Number of read cycles—The number of major read cycles used for each scan. normally set to the smallest integer greater than or equal to the number of pixels to be displayed on each scan divided by 128, 64, or 32 in the 16-, 8-, or 4 bit bus width mode, respectively.

[36] X Limit Register—The X limit (XL) register selects the width of the memory that will be read during the refresh process. This value must be the number of read cycles in the X scan configuration register multiplied by 128, 64, or 32 in 16-, 8-, or 4-bit bus width modes, respectively.

[37] Y Limit Register—The Y limit (YL) register selects the height of memory that will be read during the refresh process. This value is set to the memory height plus the number of extra scans required during down scrolling.

[38-3E] X Scan Count Registers (0-6)—The X scan count (XCT0 through XCT6) registers are used to program most of the horizontal timing events such as horizontal blank time. Each register determines the time of one horizontal event and the events are stored in the order of increasing time. The information in the register is defined in Table 13.

Table 13 • 78690 X Scan Count Registers (0-6) Descriptions

Bit	Description
15	Reserved—Reserved for test. (normally zero)
14	Program bit—Set to one in the X scan count register following the X scan count register that contains the sync request event (bits 11:13). This must be cleared in the remaining X scan count registers.
13:11	Event—Selects the horizontal parameters as follows:

Bits 13	12	11	Event
0	0	0	Set horizontal blank low.
0	0	1	Set horizontal blank high.
0	1	0	Set horizontal sync low except during vertical sync.
0	1	1	Set horizontal sync high except during vertical sync.
1	0	0	Set horizontal sync low
1	0	1	Set horizontal synch high.
1	1	0	End horizontal period.
1	1	1	Set sync request event.

10:00	Time of event—Selects the time of event, in increments of 1/16 of a major cycle, from 3 1/4 major cycles that precede the start of the first memory cycle on a scan. The start of a memory cycle is at the rising edge of the PHI 2 input during the RAS precharge that begins the cycle.

[3F] Sync Phase Register—The sync phase (SYNP) register contains the sync phase adjustment value. This is the value that the video control loads into the horizontal sync counter at each system sync time (once per frame).

• Specifications

The mechanical, electrical, and environmental characteristics and specifications for the 78690 video control are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Power supply voltage (V_{DD}): 5.0 V ±5%
- Temperature range (T_A): 70°C

Mechanical Configuration

The physical dimensions of the 78690 84-pin cerquad package are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Power supply voltage (V_{DD}): –0.5 V to 6.0 V
- Input voltage applied (V_{in}): –0.5 V to 6.0 V
- Output voltage applied (V_{out}): –0.5 V to 6.0 V
- Power dissipation (P_D): 3.5 W at 0°C
- Active temperature (T_A): 0°C to 70°C
- Storage temperature: –55°C to 125°C

Recommended Operating Conditions

- Power supply voltage (V_{DD}): 5 V ±5%
- Temperature (T_A): 0°C to 70°C

dc Electrical Characteristics

The dc electrical parameters of the 78690 video control for the operating voltage and temperature ranges specified are listed in Table 14.

Table 14 • 78690 dc Input and Output Parameters

Symbol	Parameter	Test Condition	Requirements Min.	Max.	Units
V_{IH}	High-level input voltage		2.0	—	V
V_{IL}	Low-level input voltage		—	0.8	V
V_{OH}	High-level output voltage	I_{OH}=0.2 mA V_{DD}=0V	2.7	—	V
V_{OL}	Low-level output voltage DAT<15:00>	I_{OL}=–5 mA V_{DD}=0V	—	0.5	V
V_{OL}	Low-level output voltage all other outputs	I_{OL}=–5 mA	—	0.4	V
CLK_{IH}	Clock input high level		2.7	—	V
CLK_{IL}	Clock input low level		—	0.4	V
I_{DD}	Active supply current	V_{DD}=max	—	0.66	mA

Symbol	Parameter	Test Condition	Requirements Min.	Max.	Units
I_{IH}	Input high leakage current	V_{DD} = max $V_{in} = V_{DD}$(max)	—	20	µA
I_{IL}	Input low leakage current	V_{DD} = max V_{in} = 0 V	—	–20	µA
I_{ZH}	Hi-impedance input high leakage current	V_{DD} = max $V_{in} = V_{DD}$(max)	—	20	µA
I_{ZL}	Hi-impedance input low leakage current	V_{DD} = max V_{in} = 0 V	—	–20	µA
C_{in}	Input capacitance		—	10	pF
C_{IO}	Input/output capacitance		—	10	pF

ac Electrical Characteristics

The ac timing parameters for the 78690 video control are grouped according to clock input, processor interface, instruction/data bus, memory interface, and monitor timing and synchronization request. Table 15 lists the ac input specifications.

Table 15 • 78690 ac Test Limits and Specifications

Symbol	Definition	Requirements Min.	Max.	Units
C_{in}	Input capacitance	—	10	pF
C_{io}	Input/output capacitance	—	10	pF
t_{IR}	Input signal rise time	—	10	ns
t_{IF}	Input signal fall time	—	10	ns

The following conditions apply to the ac test conditions unless otherwise stated.

- The delay times extend from the 1.5 V level of the clock input to the V_{OH} or V_{OL} level of the measured signal.
- The rise times are measured from the 10% to 90% level of the signal transitions. Fall times are measured from the 90% to 10% of the signal transitions.
- The measurements are with a 50 pF capacitive load on the outputs. Exceptions to this are outputs ID<7:0> that have a 500 pF load, and DAT<15:00> that have a varying load up to 500 pF.

Clock Input Timing

The ac input parameters for the phase input clock signals PHI1 through PHI4 and the CAS signal are shown in Figure 9. The parameters are defined in Table 16.

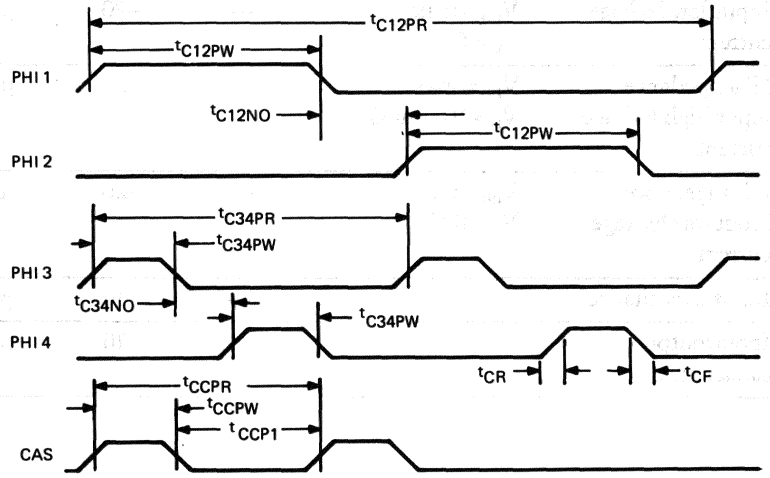


Figure 9 • 78690 Clock Input Timing

Table 16 • 78690 Clock Input Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{C12PR}	Period of PHI1 and PHI2	228	684
t_{C34PR}	Period of PHI3 and PHI4	114	342
t_{CCPR}	Period of CAS	85.5	257
t_{C12PW}	Pulse width of PHI1 and PHI2	85.5	—
t_{C34PW}	Pulse width of PHI3 and PHI4	28.5	—
t_{CCPW}	Pulse width of CAS	28.5	—
t_{CR}*	Rise time of PHI1, PHI2, PHI3, PHI4, CAS	—	5.0
t_{CF}*	Fall time of PHI1, PHI2, PHI3, PHI4, CAS	—	5.0
t_{C12NO}	Nonoverlap time of PHI1 and PHI2	23.5	—
t_{C34NO}	Nonoverlap time of PHI3, PHI4	23.5	—
t_{CCPL}	Low time between CAS pulses	57	—

*Rise time is measured from 0.4 V to 2.7 V; fall time, from 2.7 V to 0.4 V.

Processor Interface Timing

The processor interface timing is shown in Figure 10 and the parameters listed on the figure are defined in Table 17.

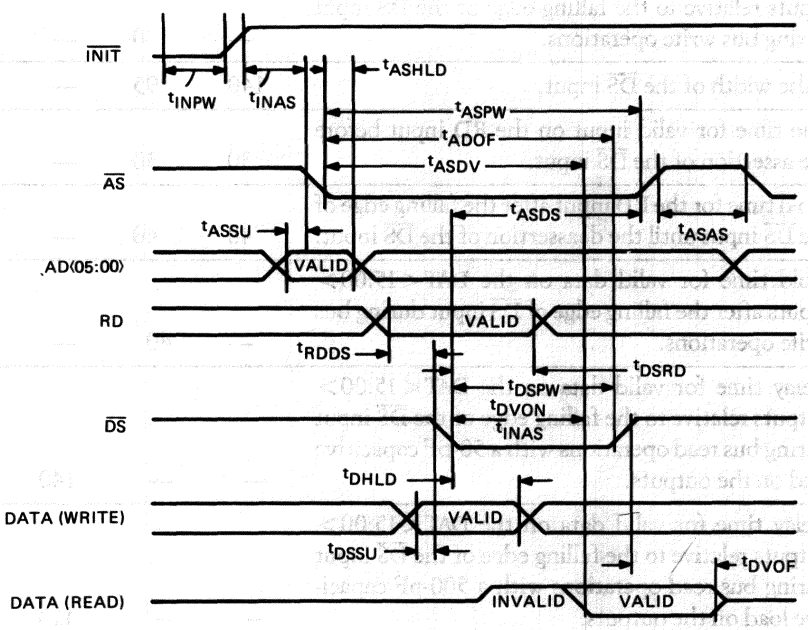


Figure 10 • 78690 Processor Interface Signal Timing

Table 17 • 78690 Processor Interface Timing Parameters

Symbol	Definition	Requirements (ns)* Minimum Read	Minimum Write	Maximum Read	Maximum Write
t_{ASAS}	The time that the $\overline{AS}$ input must not be asserted before being reasserted.	105	105	—	—
t_{ASSU}	Setup time for valid input data on the ADD<5:0> inputs relative to the falling edge of the $\overline{AS}$ input.	0	0	—	—
t_{ASPW}	Pulse width of the $\overline{AS}$ input if the $\overline{DS}$ input is not asserted.	180	180	—	—
t_{ASHLD}	Hold time for valid input data on the ADD<5:0> after the falling edge of the $\overline{AS}$ input.	40	40	—	—
t_{ASDS}	The time from assertion of the $\overline{DS}$ input to deassertion of the $\overline{AS}$ input.	140	75	—	—

Symbol	Definition	Requirements (ns)* Minimum Read	Minimum Write	Maximum Read	Maximum Write
t_{DSSU}	Setup time for valid data on the DAT<15:00> inputs relative to the falling edge of the $\overline{DS}$ input during bus write operations.	—	0	—	—
t_{DSPW}	Pulse width of the $\overline{DS}$ input.	140	95	—	—
t_{RDDS}	The time for valid input on the RD input before the assertion of the $\overline{DS}$ input.	30	30	—	—
t_{DSRD}	Hold time for the RD input after the falling edge of the $\overline{DS}$ input until the deassertion of the $\overline{DS}$ input.	40	40	—	—
t_{DHLD}	Hold time for valid data on the DAT<15:00> inputs after the falling edge of $\overline{DS}$ input during bus write operations.	—	40	—	—
t_{DVON}	Delay time for valid data on the DAT<15:00> outputs relative to the falling edge of the $\overline{DS}$ input during bus read operations with a 50-pF capacitive load on the outputs.	—	—	140	—
t_{DVONA}†	Delay time for valid data on the DAT<15:00> outputs relative to the falling edge of the $\overline{DS}$ input during bus read operations with a 500-pF capacitive load on the outputs.	—	—	410	—
t_{DVOF}	Hold time with previous data valid on the DAT<15:00> outputs after the rising edge of $\overline{DS}$ during bus read operations	0	—	50	—
t_{INAS}	The time from deassertion of the $\overline{INIT}$ input to the assertion of the $\overline{AS}$ input.	100	100	—	—
t_{INPW}‡	Pulse width of the $\overline{INIT}$ input.	12×t	12×t	—	—
t_{ADOF}	The time from assertion of the $\overline{AS}$ input to deassertion of the $\overline{DS}$ input.	180	135	—	—
t_{ASDV}	Delay time for valid data on the DAT<15:00> outputs relative to the falling edge of the $\overline{AS}$ during bus read operations.	—	—	180	—

*Timing is measured at the V_{OH} and V_{OL} levels

†t_{DVONA} = t_{DVON} +60 ns per 100-pF capacitance additional loading. (Intermediate values may be calculated.)

‡This parameter is 12 × t where t = PHI1 or PHI2 clock period.

Instruction/Data Bus Timing

Figure 11 shows the instruction/data bus signal timing and the timing parameters are listed in Table 18.

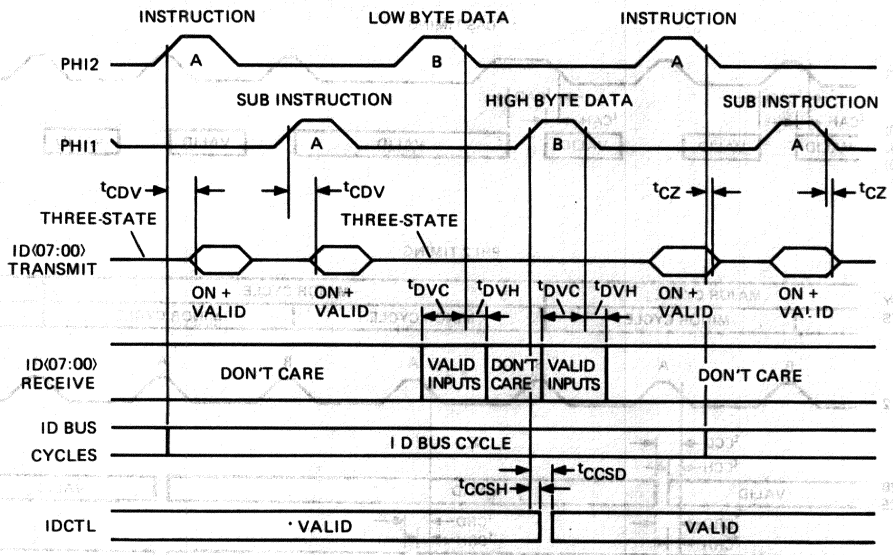


Figure 11 ▪ 78690 Instruction/Data Bus Signal Timing

Table 18 ▪ 78690 Instruction/Data Bus Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{CDV}	Maximum delay time for valid output data on the ID<7:0> lines relative to the rising edge of the PHI1 A and PHI2 A input.	—	60
t_{CZ}*	Delay time from the falling edge of the PHI1A and PHI2A input to the high-impedance level on the ID<7:0> lines.	5.0	25
t_{DVC}	Minimum setup time for valid input data on the ID<7:0> lines relative to the falling edge of the PHI1 B and PHI2 B input.	20	—
t_{DVH}	Minimum hold time for valid input data on the ID<7:0> lines after the falling edge of PHI1 B and PHI2 B input.	5.0	—
t_{CCSH}	With previous output data valid, t_{CCSH} is the minimum hold time on the IDCTL output relative to the rising edge of the PHI1 B signal. (PHI1 B is used but time is referenced to the PHI1 signal.)	0	—
t_{CCSD}	Maximum delay time for valid data on the IDCTL output relative to the PHI1 B signal going high. (PHI1 B is used but time is referenced to the PHI1 signal.)	—	75

*t_{CZ} is measured from the deasserted level of the 1.5-V clock input to the high-impedance level of the ID bus.

Memory Interface Timing

Figure 12 shows the memory interface signal timing and the timing parameters are listed in Table 19.

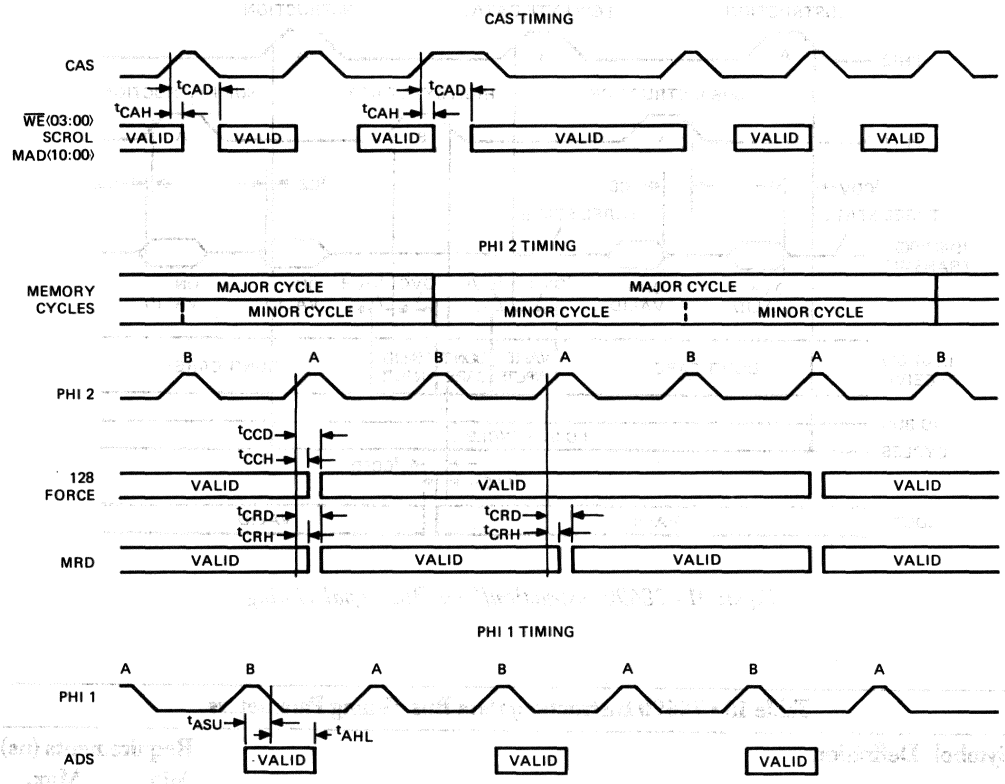


Figure 12 • 78690 Memory Interface Signal Timing

Table 19 • 78690 Memory Interface Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{CAH}	The delay time that the previous output of the MAD<10:00>, $\overline{WE}$<3:0>, and SCROL signals is valid after the rising edge of the CAS input.	8.0	—
t_{CAD}	The delay time for valid outputs of the MAD<10:00>, WE, and SCROL signals after the rising edge of the CAS input.	—	55
t_{CCH}	The delay time that the previous output of the 128/$\overline{16}$ and FORCE signals is valid after the rising edge of PHI2 A input.	5.0	—

Symbol	Definition	Requirements (ns) Min.	Max.
t_{CCD}	The delay time for valid output of the 128/16 and FORCE signals after the rising edge of the PHI2 A input.	—	55
t_{CRH}	The time that the previous output of the MRD signal is valid after the rising edge of the PHI2 A input.	5.0	—
t_{CRD}	The delay time for valid output of the MRD signal after the rising edge of the PHI2 A input.	—	55
t_{ASU}	The setup time for valid input of the ADS signal relative to the falling edge of the PHI2 B input.	50	—
t_{AHL}	Minimum hold time for valid input of the ADS signal relative to the falling edge of PHI1 B	—	27

Monitor Timing and Synchronization Request Timing

During the synchronization interval, the PHI1, PHI2, PHI3, and PHI4 clock inputs are set to a low level, and the CAS input is set to a high level. Figure 13 shows the normal and synchronization clock timing and the parameters are listed in Table 20. The SYNC input parameters are valid for the SYNC request from the video control or from another source. The clocks continue after the SYNC interval.

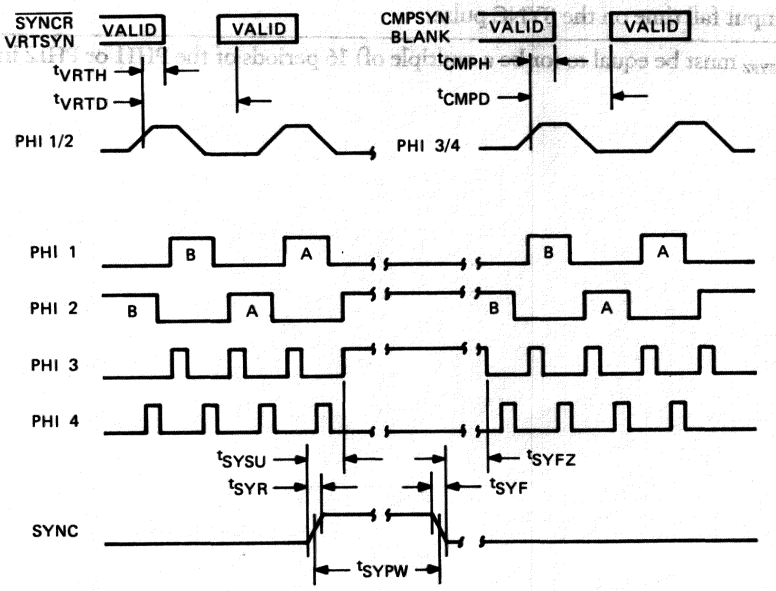


Figure 13 • 78690 Monitor and Synchronization Request Signal Timing

Table 20 • 78690 Monitor and Synchronization Request Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{VRTD}	The delay time for valid output of the VSYNC or $\overline{SYNCR}$ signal relative to the rising edge of the PHI1 or PHI2 input.	—	90
t_{VRTH}	The delay time that the previous output of the VSYNC or $\overline{SYNCR}$ signals is valid relative to the rising edge of the PHI1 or PHI2 inputs.	0	—
t_{CMPD}	The delay time for valid output of the CMPSYN or BLANK signals relative to the rising edge of the PHI3 or PHI4 input.	—	45
t_{CMPH}	The delay time that the previous output of the CMPSYN or BLANK signals is valid relative to the rising edge of PHI3 or PHI4 inputs.	5.0	—
t_{SYSU}	Minimum delay from the assertion of the SYNC signal to the PHI2 input going high. This delay starts the clock freeze period that is a restriction of the external clock generation control.	0	—
t_{SYPW}*	The pulse width of the SYNC signal.	800	24 µs
t_{SYFZ}*	The delay time from the deassertion of SYNC to the end of the clock freeze time. This delay is a restriction of the external clock generation control.	400	24 µs
t_{SYR}	Input rise time of the SYNC pulse.	—	50
t_{SYF}	Input fall time on the SYNC pulse.	—	50

*t_{SYPW} + t_{SYFZ} must be equal to (or be a multiple of) 16 periods of the PHI1 or PHI2 inputs.

Interrupt and Request Signal Timing

The signal timing for the interrupt and request are listed and defined in Table 21.

Table 21 • Interrupt and Request Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_A	The assertion propagation delay for the $\overline{INT}$ or $\overline{REQ}$ signal in response to a status bit being set when the corresponding bit of the Interrupt or Request Enable register has been set previously. Delay t_A is measured from the edge of the $\overline{DS}$ or system clock signal that sets the status bit.	0	140
t_D	The deassertion propagation delay for the $\overline{INT}$ or $\overline{REQ}$ signal in response to a status bit being cleared when the corresponding bit of the Interrupt or Request Enable register has been set previously. Delay t_D is measured from the edge of the $\overline{DS}$ signal that clears the status bit.	0	140
t_{AM}	The assertion propagation delay for the $\overline{INT}$ or $\overline{REQ}$ signal in response to setting a bit of the Interrupt or Request Enable register when the corresponding status bit has been set previously. Delay t_{AM} is measured from the edge of the $\overline{DS}$ signal that sets the Interrupt or Request Enable register bit.	0	180
t_{DM}	The deassertion propagation delay for the $\overline{INT}$ or $\overline{REQ}$ signal in response to clearing a bit of the Interrupt or Request Enable register when the corresponding status bit has been set previously. Delay t_{DM} is measured from the edge of the DS signal that initiates the resetting of the Interrupt or Request register bit.	0	180

• Interfacing Techniques

Up to twenty-four 78680 video processors may be used with each video control. The video control includes a system interface, a bit-map memory interface, and a display interface as shown in Figure 3. Refer to the *Dragon Video System Hardware Specification* for a complete description of a video system using the 78680 video processor and 78690 video control.

The system interface connects the video control to the local processor or DMA controller through the processor interface. The processor interface transfers 16 data bits, 6 address bits and 6 control bits and receives the parameters and commands from the local processor. The interface handles register accesses and controls timing and all necessary bus signals to allow the video control to act as a bus slave to the local processor or DMA device. The system interface also connects the video control to the instruction/data bus through the ID interface.

The ID interface transfers information on the 8-bit bidirectional instruction/data bus and the chip select control line. The instruction/data (ID) bus is used to transfer information within the video processor and to controls its operation. It is used to load and read registers and to execute direct or indirect instructions.

The bit-map memory interface includes the bit-map memory address bus and the signals that control it. It transfers address information on 11 lines, write enable information on 4 lines, and control information on 5 lines.

The display interface connects to the monochrome or color monitor interface for video and timing. The interface signals include the four synchronization and blanking signals.

The video control requires a 5-Vdc power supply.

• Features

- Programmable for videodisplays using a maximum of 4096 by 4096 pixel locations
- Compatible with Digital's DC322 video processor and DC323 videocontrol
- Interfaces with the MicroVAX 78032 microprocessor and Motorola 68000 series microprocessors
- Selectable cursor characters and single- or double-width hairline cursor
- Detects two independent programmable active regions

• Description

The DC503 programmable cursor chip (PCC) is contained in a 44-pin cerquad package and is used to provide a programmable cursor for use with videodisplay terminals. Figure 1 is a simplified block diagram of the DC503.

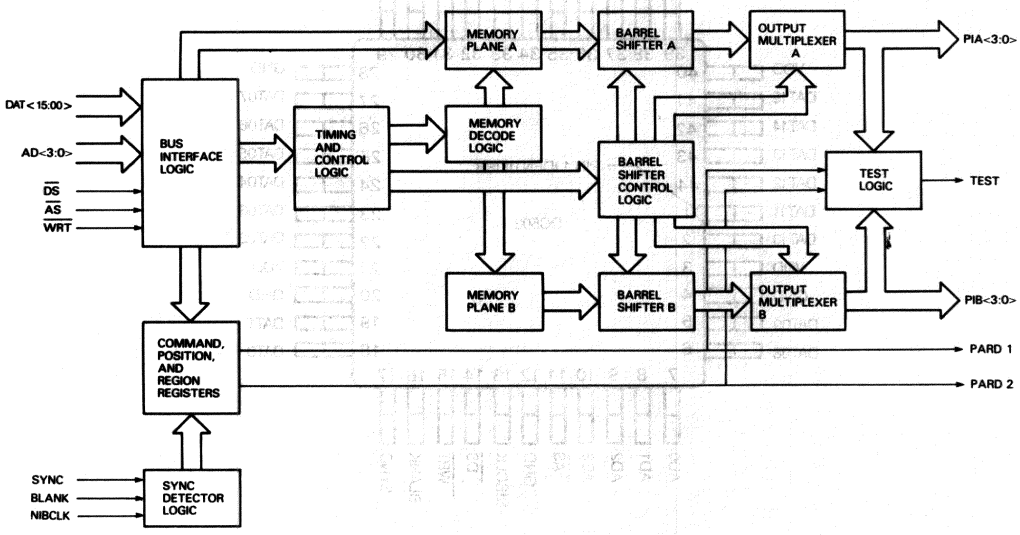


Figure 1 • DC503 Simplified Block Diagram

The DC503 enables cursor characters or icons to be programmed and positioned to the desired location of a videodisplay. The DC503 contains two 16 by 16 memory arrays to store the cursor sprite. This enables the cursor to select up to three colors and display normal video in its transparent region. In addition, the DC503 can be programmed to display either a single- or a double-width full screen hairline cursor. Two programmable boundary regions on the display can be detected and the cursor can be clipped in either of the two regions on the screen.

• Pin and Signal Description

The input and output signals and power and ground connections are shown in Figure 2 and summarized in Table 1.

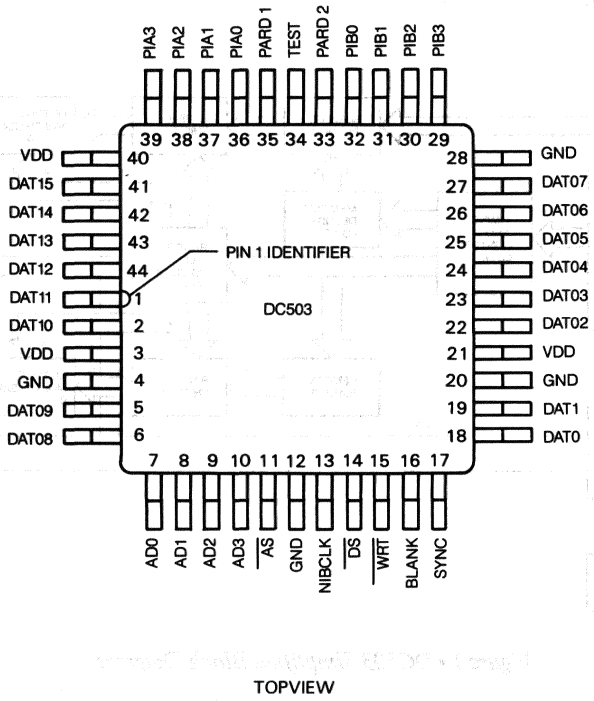


Figure 2 • DC503 Pin Assignments

Table 1 • DC503 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
41-44 1,2,5,6, 27-22 19,18	DAT<15:00>	inputs	Data lines <15:00>—Data inputs from the processor bus.
10-7	AD<3:0>	inputs	Address lines <3:0>—Address inputs from the processor bus.
11	$\overline{AS}$	input	Address strobe—Strobes the address inputs AD<3:0> into the bus interface.
14	$\overline{DS}$	input	Data strobe—Strobes the data inputs DAT<15:00> into the bus interface.
15	$\overline{WRT}$	input	Write enable—When asserted, a write operation is performed to the DC503.
17	SYNC	input	Synchronize—A horizontal or composite sync signal.
16	BLANK	input	Blank—A video composite blank signal.
13	NIBCLK	input	Nibble clock—The timing signal used to synchronize the output to the display.
39-36	PIA<3:0>	outputs	Plane information A <3:0>—Four-bits of the 16-bit data word for cursor plane A.
29-32	PIB<3:0>	outputs	Plane information B <3:0>—Four-bits of the 16-bit data word for cursor plane B.
35	PARD1	output	Programmable active region 1 detect—Indicates that active region 1 of the display has been detected.
33	PARD2	output	Programmable active region 2 detect—Indicates that active region 2 of the display has been detected.
34	TEST	output	Test—Used for test purposes only.
3,21, 40	V_{DD}	input	Voltage—Power supply voltage
4,12, 20,28	GND	input	Ground—Ground reference.

• Functional Description

The data (DAT<15:00>) and address (AD<3:0>) information from the bus is loaded into the bus interface by the data strobe ($\overline{DS}$) and address strobe ($\overline{AS}$) input respectively. The data determines the cursor font and specifies the coordinates for the cursor location. Address AD<3:0> selects the register that will receive the data. The write signal ($\overline{WRT}$) loads the information into the command, position, and region registers. The data is transferred to the memory plane A or B under control of the timing and control and memory decode logic. Information from each memory plane is transferred to barrel shifter A and B, under control of the barrel shifter logic, to the output multiplexers. Each multiplexer output provides 4-bits of information (PIA<3:0> and PIB<3:0>) every 37.8 ns for the 16-bit by 16-bit cursor font.

Timing and control is provided by the sync detector logic that receives the SYNC and BLANK signals from the controlling device and the NIBCLK clock pulses. Information from the sync detector is transferred to the registers that detect one of the two active regions on the display. The regions are indicated by the PARD1 and PARD2 outputs.

The test logic receives PARD1 and PARD2 signals and the output from each multiplexer. The TEST output is used to self-test the cursor memories and active region detectors.

Register Selections

Address bits AD<3:0> select a command, position, or active region register to be loaded. They also select the cursor memory by an indirect memory address to an internal address counter. Table 2 lists the address codes and register selections.

Table 2 • DC503 Address Register Select Functions

Address Line AD3	AD2	AD1	AD0	Register*
0	0	0	0	Load command register (CMDR)
0	0	0	1	Load X position register (XPOS)
0	0	1	0	Load Y position register (YPOS)
0	0	1	1	Load Xmin1 active region register
0	1	0	0	Load Xmax1 active region register
0	1	0	1	Load Ymin1 active region register
0	1	1	0	Load Ymax1 active region register
1	0	1	1	Load Xmin2 active region register
1	1	0	0	Load Xmax2 active region register
1	1	0	1	Load Ymin2 active region register
1	1	1	0	Load Ymax2 active region register
1	1	1	1	Cursor memory (indirect memory address via an internal address counter)

*All registers are write only.

Command Register Description

The command register (CMD) is a write-only register and is used for communication between the CPU and the programmable cursor chip. The command register format is shown in Figure 3 and the functions are listed in Table 3.

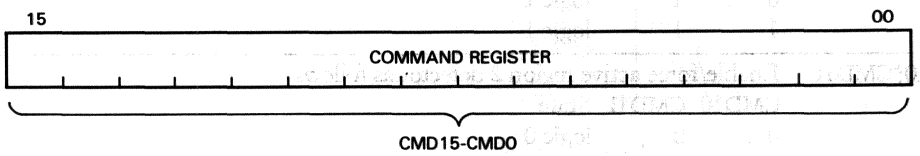


Figure 3 • DC503 Command Register Format

Table 3 • DC503 Command Register Descriptions

Bit	Description
CMD0*CMD1	Enable/force cursor plane A output as follows: CMD0 CMD1 State 0 0 logic 0 1 0 enable 0 1 logic 1 1 1 logic 1
CMD2*CMD3	Enable/force cursor plane B output as follows: CMD2 CMD3 State 0 0 logic 0 1 0 enable 0 1 logic 1 1 1 logic 1
CMD4*	Command 4—Hairline cursor enable 1 = enable, 0 = disable
CMD5*	Command 5—Clip cursor inside active region 1 = clip cursor, 0 = don't clip cursor
CMD6*	Command 6—Clip hairline cursor inside active region 1 or 2 1 = active region 1, 0 = active region 2
CMD7*	Command 7—Double width hairline cursor 1 = double width, 0 = single width

Bit	Description
CMD8,*CMD9	Enable/force active region 1 detector as follows: CMD8 CMD9 State 0 0 logic 0 1 0 enable 0 1 logic 1 1 1 logic 1
CMD10,*CMD11	Enable/force active region 2 detector as follows: CMD10 CMD11 State 0 0 logic 0 1 0 enable 0 1 logic 1 1 1 logic 1
CMD12	Command 12—Load cursor memories 1 = enable load, 0 = inhibit load
CMD13	Command 13 —Select Hi/Low active BLANK (vertical) 1 = active high blank, 0 = active low blank
CMD14	Command 14—Select Hi/Low active SYNC (scan) 1 = active high sync, 0 = active low sync
CMD15	Command 15—Reset signal TEST = 0 1 = Clear TEST to "0", 0 = Enable the test hardware

*These bits are double buffered during vertical blanks and any new value loaded is not acted upon until the next frame. CMD0 and/or CMD2 must be enabled for the hairline cursor to appear.

• Architectural Description

The DC503 operates with a videodisplay using 4K by 4K locations and having a maximum scan rate of 9.45 ns/pixel. It provides a single- or double-width hairline cursor in two independent programmable active regions. Clipping of the cursor in scan occurs in either active region 1 or active region 2. The cursor resolution is to a pixel position. The active region resolution is to a nibble (4-bit) position in the X direction and to a pixel position in the Y direction. Memory plane A and/or plane B output information can be selected and the planes can be forced to either 1 or 0. The blank areas and synchronization are controlled by the programmable active Hi/Low Blank and SYNC inputs. The origin of the display screen is the upper-left corner of the screen and the display is not interlaced.

Data from the DC503 is provided as two 4-bit nibbles every 37.8 ns assuming 9.45 ns/pixel. The CPU data path to load the information is 16 bits wide. The input clock period (NIBCLK) is aligned with a nibble output. Therefore, it is four times the pixel rate. The initial display of the cursor is not important while CRT is blanking. The SYNC input accepts either horizontal or composite monitor sync information. The BLANK input signal must be active for a minimum of two sync intervals and must include or coincide with SYNC signal. The SYNC input must be active for two NIBCLK clock periods. Registers must not be loaded between the assertion of vertical blank input until three horizontal sync pulses occur. During the powerup initialize time, the CPU must allow the DC503 to become self-initialized by two video frames and with the NIBCLK clock operating continuously.

The DC503 provides two similar logic paths for controlling the X and Y of the videodisplay axes. Each path consists of a set of registers, a random access memory (RAM), and multiplexing logic. Most of the registers are double buffered to allow a full frame time for loading. Both the X and Y paths are pipelined and the information in each paths must be complete before the entire content is valid.

Figure 4 shows the coordinates for the two programmable active regions and the cursor position. These 18 coordinates are programmed by the user and stored in registers. After the coordinates and cursor position have been loaded, they become the holding registers for a comparison with the internal X and Y counters that are used to determine the pixel reference.

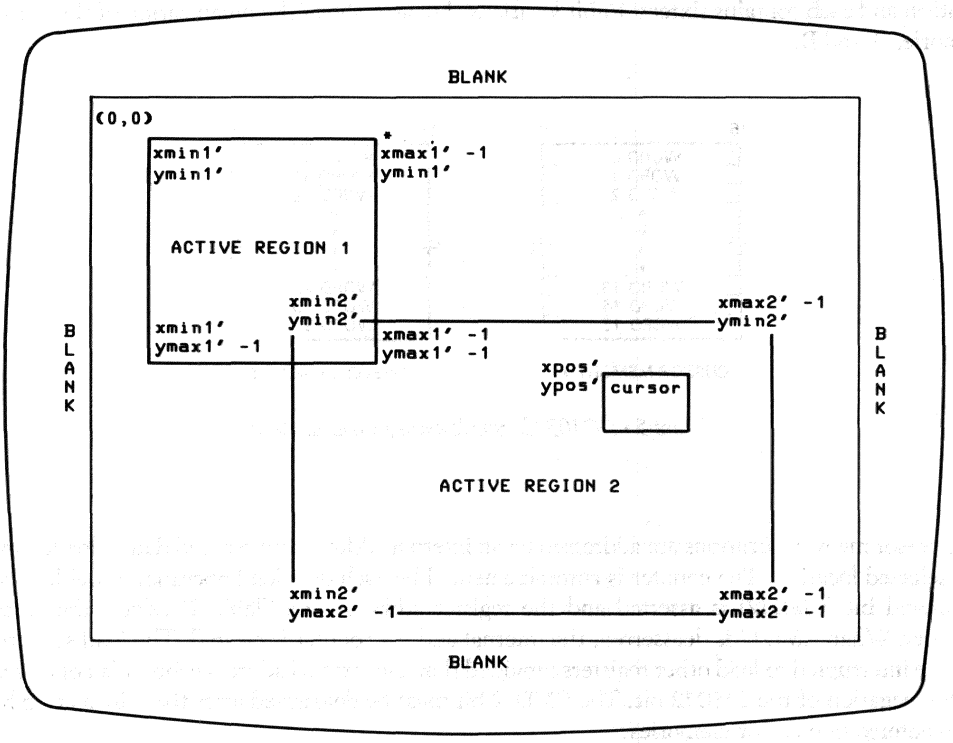


Figure 4 ▪ DC503 Programmable Active Region Display

*The maximum values of the active regions 1 and 2 define the limit that is reached by the region. As an example, the upper right location of region 1 includes Xmax-1 and Ymin (not Xmax and Ymin).

Y (pos max or min 1 and 2)' = Y (pos max or min 1 and 2) + the number of SYNC signals per BLANK signals after the first assertion of BLANK and SYNC signals.

X (pos max or min 1 and 2)' = X (pos max or min 1 and 2) + 6 NIBCLKH signals after the last assertion of the SYNC signal.

**Location where BLANK and SYNC are first asserted

Because the counters may not wrap around to a known position, the content of the internal X counter will be cleared at the end of each scan line and the content of the Y counter will be cleared at vertical blank time.

The X counter is incremented every NIBCLK clock pulse and is synchronized with the video stream data. The Y counter is incremented every scan line at the scan sync time.

The X and Y coordinates may change during a frame. However, the new coordinates will not become active until the present frame has been completed. This prevents part of the cursor from being displayed at one position on the screen and the remaining part displayed at the newly assigned position. The active regions may also be changed during a frame. However, the values are stored in registers until the next frame occurs.

The memory planes A and B contain the binary values of the cursor font. The memories are identical and each contains sixteen 16-bit locations. Figure 5 shows the organization of the cursor memories A and B.

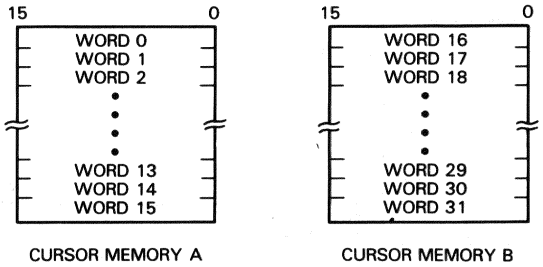


Figure 5 • DC503 Cursor Memory Organization

The cursor memory locations are addressed by an internal address counter, and data is loaded into the selected location. The counter is autoincremented by each data load operation provided that command bit (CMD12) is asserted and the register address code (Table 2) selects the cursor memory. When CMD12 is deasserted, the internal address counter is cleared. The load sequence may be interrupted to load other registers provided that the internal address counter is not cleared by the negation of the CMD12 bit. The CMD12 bit must be deasserted after the information has been entered into cursor memories.

The 16-bit words are read from the memory once for each scan line and shifted by the barrel shifter to enable the exact pixel alignment. The words are shifted by pixel amounts (0-3) and multiplexed into five consecutive 4-bit nibbles. The shift constant is determined by the two least significant bits of the Xpos address, since the offset can be up to three pixels. A nibble is generated for each NIBCLK clock pulse that results in an offset value and a multiplexing of five groups of four nibbles. Output PIA<3:0> is a nibble of cursor memory plane A, and PIB<0:3> is a nibble of cursor memory plane B.

The TEST output is used to selftest the cursor memories and active-region detectors. Writing a 0 to bit 15 of the Command register enables the test hardware. The assertion of any PIA<3:0>, PIB<3:0>, or PARD<2:1> output will assert the TEST output. The TEST output is disabled by writing a 1 to bit 15 of the Command register. During normal operation, the timing of the TEST output is unspecified unless bit 15 is cleared.

• Specifications

The mechanical, electrical, and environmental characteristics and specifications for the DC503 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Power supply voltage (V_{DD}): 5.0 V ±5%
- Temperature range (T_A): 0°C to 70°C

Mechanical Configuration

The physical dimensions of the DC503 44-pin cerquad package are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Power supply voltage (V_{DD}): –0.5 V to 5.5 V
- Pin voltage: –0.3 V to V_{DD}+0.3 V
- Power dissipation (T_J=0°C): 0.3 W
- Operating temperature (T_A): 0°C to 70°C
- Storage temperature: –55°C to 125°C

Recommended Operating Conditions

- Power supply voltage (V_{DD}): 5 V ±5%
- Temperature (T_A) 0°C to 70°C

dc Electrical Characteristics

The dc electrical parameters of the DC503 for the operating voltage and temperature ranges specified are listed in Table 4.

Table 4 • DC503 dc Input and Output Parameters

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units
High-level input voltage	V_{IH}		2.0	—	V
Low-level input voltage	V_{IL}		—	0.8	V
High-level clock input voltage	V_{IH}		2.7	6.0	V
Low-level clock input voltage	V_{IL}		0	0.4	V
Input high leakage current	I_{IH}	$V_{in} = V_{DD} = 5.25$ V	—	20	µA
Input low leakage current	I_{IL}	$V_{in} = 0$ V $V_{DD} = 5.25$ V	—	-20	µA
High-level output voltage	V_{OH}	$I_{OH} = 0.2$ mA	2.7	—	V
Low-level output voltage	V_{OL}	$I_{OL} = -5.0$ mA	—	0.4	V

ac Electrical Characteristics

The signal timing parameters for the NIBCLK clock input are shown in Figure 6 and defined in Table 5. The waveforms and propagation delays symbols for the input and output signals are shown in Figures 7 through 10. The parameters for the symbols on the figures are defined in Table 6. The specifications and conditions for the ac tests as as follows.

- Input capacitance: 10 pF
- Input signal rise and fall time: 10 ns
- All delay times extend from the 1.5V level of the clock input to the V_{OH} or V_{OL} levels of the measured signal.
- The rise times are measured from 10% to 90% and fall times are from 90% to 10% of signal transition.
- All timing parameters assume a 100 pF capacitive load on the output.

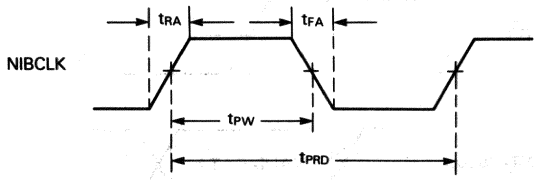


Figure 6 ▪ DC503 Clock Input Parameters

Table 5 ▪ DC503 Clock Signal Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{RA}	Rising-edge time	—	5.0
t_{FA}	Falling-edge time	—	5.0
t_{PRD}	Clock period	400	37.8
t_{PW}	Pulse width 50% duty cycle ±5ns	—	—

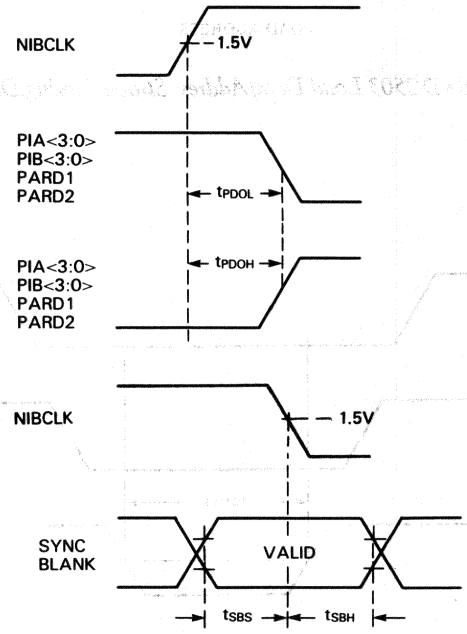
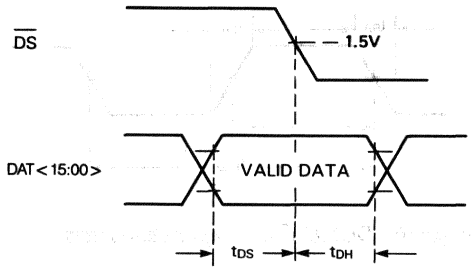
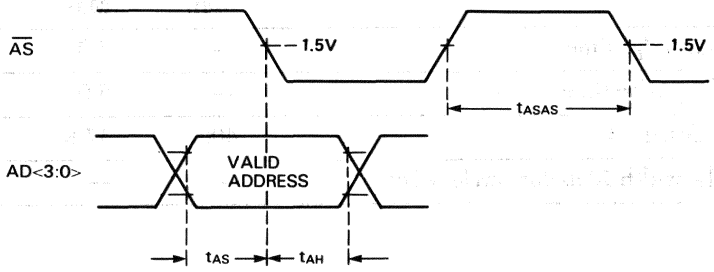


Figure 7 ▪ DC503 Clock to Input/Output Timing Delays



LOAD DATA



LOAD ADDRESS

Figure 8 ▪ DC503 Load Data/Address Strobe Timing Delays

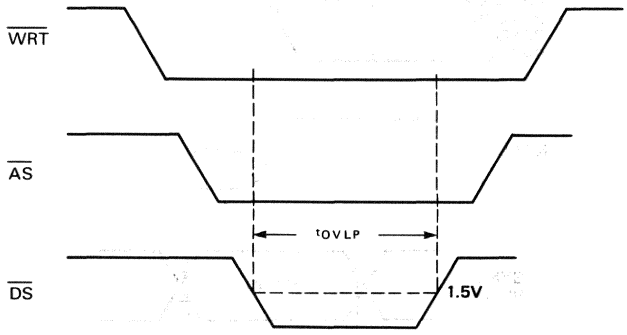


Figure 9 ▪ DC503 Write and Data/Address Strobe Timing Delay

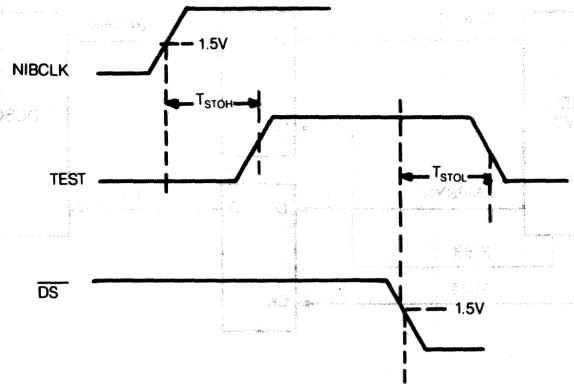


Figure 10 • DC503 TEST Output Timing Delay

Table 6 • DC503 ac Signal Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{PDOL}	Propagation delay output low	—	26
t_{PDOH}	Propagation delay output high	—	26
t_{SBS}*	SYNC/BLANK setup time	—	10
t_{SBH}*	SYNC/BLANK hold time	—	10
t_{DS}	Data setup time	—	0
t_{DH}	Data hold time	—	40
t_{AS}	Address setup time	—	0
t_{AH}	Address hold time	—	40
t_{ASAS}	Address strobe precharge time	—	105
t_{OVLP}	Overlap time	—	100
t_{STOH}	Propagation delay to TEST output high	—	40
t_{STOL}	Propagation delay to TEST output low	—	100

*The SYNC input signal must be active for a minimum of two NICBLK periods. The BLANK input signal must be active for a minimum of two SYNC signal intervals and must encompass or coincide with the SYNC signal.

• Interfacing Techniques

The DC503 programmable cursor chip can be interfaced to the DC323 adder and can operate with a full-page, half-page, and quarter-page display system. Figure 11 shows the full-page display interface configuration and associated signal timing. The DC323 generates the BLANK and CMPSYN signals to drive the BLANK and SYNC inputs of the DC503. The BLANK and CMPSYN signals can be latched into the flip-flops by the PHI3 or PHI4 phase clock of the DC323 and may rise on one PHI3/4 clock and fall on the other. The latched BLANK and SYNC signals are received by the DC503 on the falling edge of NIBCLK pulse. Latches are 374 or the equivalent.

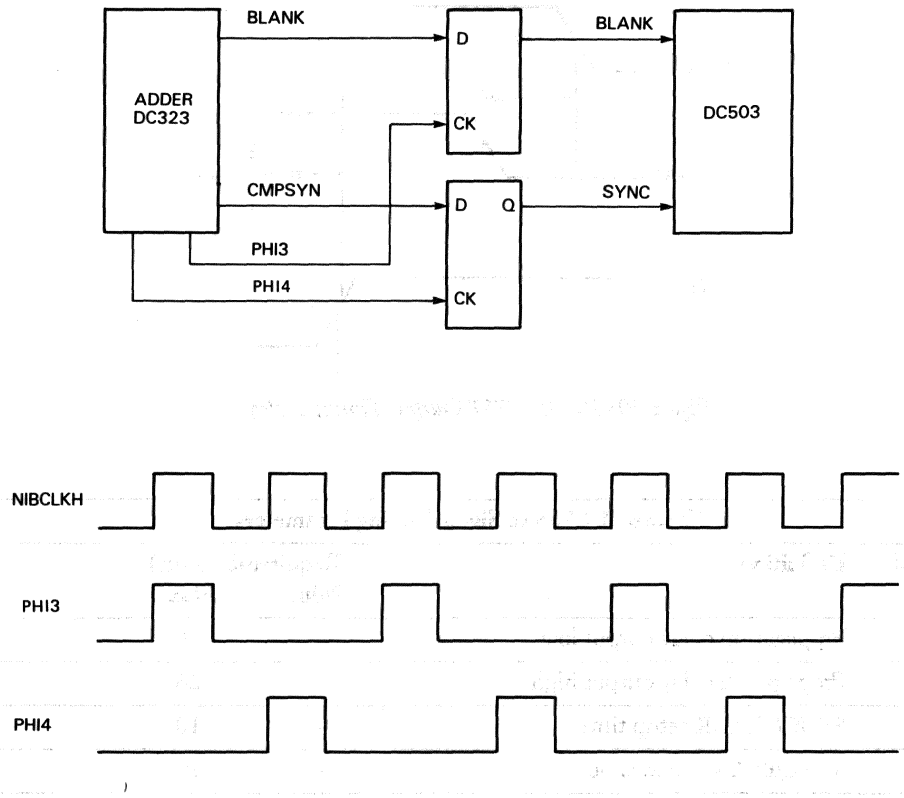


Figure 11 ▪ DC503 Full-page Display Interface and Timing

Figure 12 shows the half- and quarter-page interface configuration and signal timing. The BLANK and CMPSYN signals from the DC323 are used directly as inputs to the DC503.

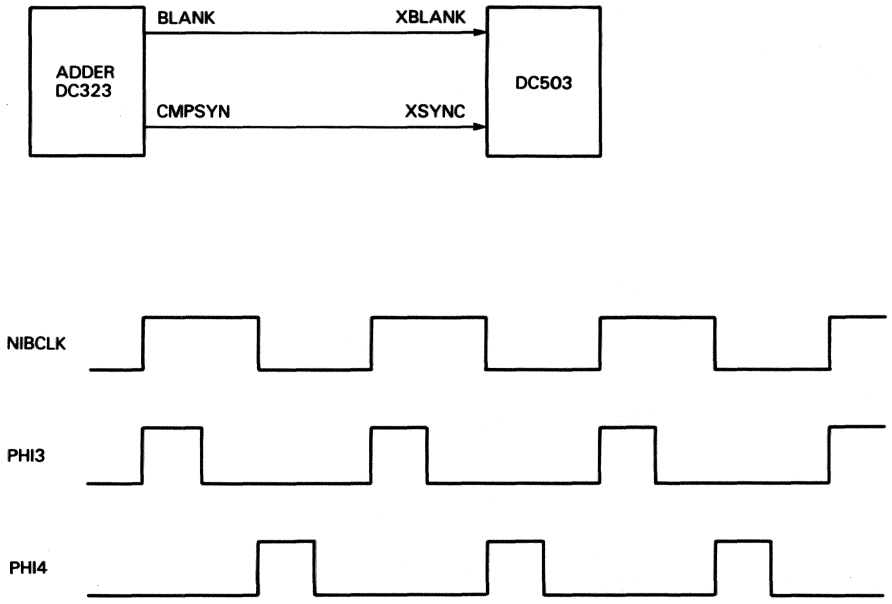


Figure 12 • DC503 Half- and Quarter-page Display Interface and Timing

• Section 3—Communication Devices

The asynchronous communication devices enable serial-line information transfers between local remote systems and terminals.

78808 Eight-channel Asynchronous Receiver/Transmitter—The octal ART is a 68-pin cerquad device that is programmable and allows the simultaneous transmission and reception of eight serial-line channels.

DC319 DL11 Compatible Asynchronous Receiver/Transmitter—The DLART is a 40-pin DIP device that allows data communication between Digital's microprocessors and console terminals or communication devices.

Features

- Eight independent full duplex serial data lines
- Programmable baud rates individually selectable for each line's transmitter/receiver (50 to 19,200 baud)
- Summary registers that allow a single read to detect a data set change or to determine the cause of an interrupt on any line
- Triple buffers for each receiver
- Device scanner mechanism that reports interrupt request due transmitter/receiver interrupts
- Independently programmable lines for interrupt-driven operation
- Modem status change detection for Data Set Ready (DSR) and Data Carrier Detect (DCD) signals
- Programmable interrupts for modem status changes
- Synchronizes critical read-only registers

Description

The 78808 Eight-channel Asynchronous Receiver/Transmitter (Octal ART) is a VLSI device for new generations of asynchronous serial communication designs and for microcomputer systems. This 68-pin device performs the basic operations necessary for simultaneous reception and transmission of asynchronous messages on eight independent lines. Figure 1 is a functional block diagram of the 78808 Octal ART.

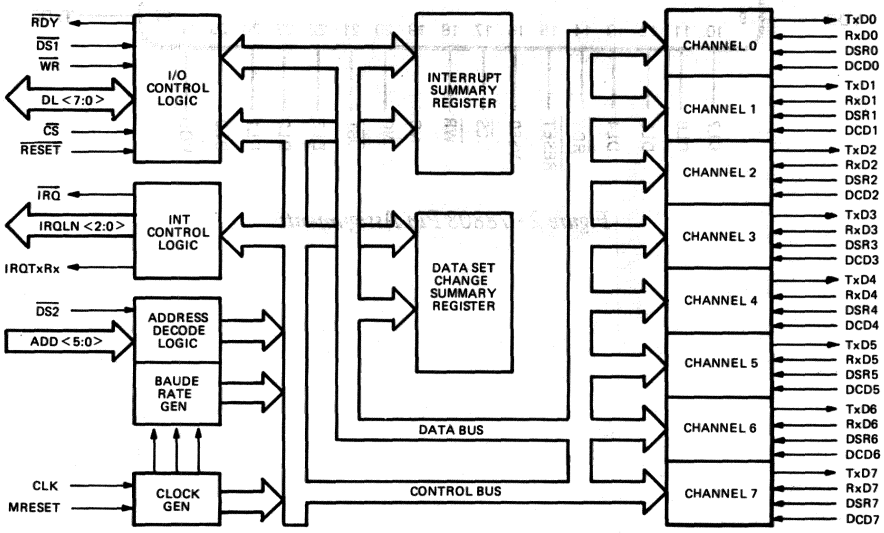


Figure 1 • 78808 Octal ART Functional Block Diagram

• Pin and Signal Definitions

The input and output pins and power and ground connections of the 78808 Octal ART are shown in Figure 2. Table 1 provides a summary of the signals defined in the following paragraphs.

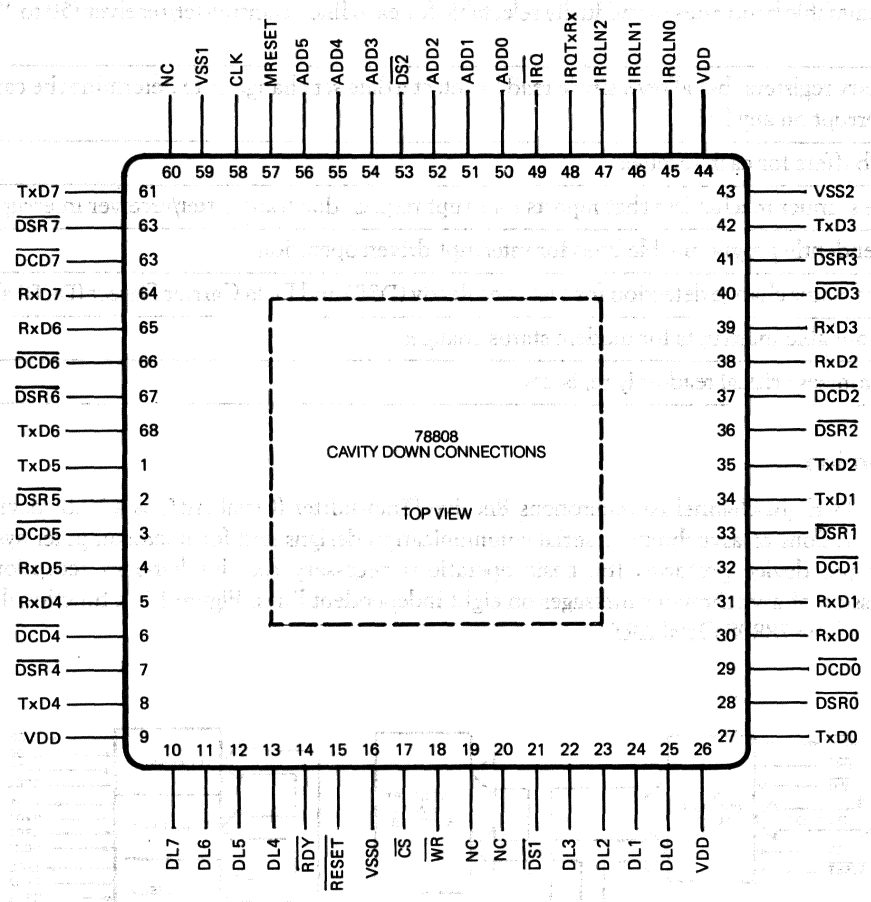


Figure 2 • 78808 Pin Assignments

Table 1 • 78808 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
10-13,22-25	DL<7:0>	input/output	Data lines <7:0>—Receives and transmits the parallel data.
50-52,54-56	ADD<5:0>	input	Address <5:0>—Selects the internal registers in the Octal ART.
17	$\overline{CS}$	input	Chip select—Activates the Octal ART to receive and transmit data over the DL<7:0> lines.
21,53	$\overline{DS1}$, $\overline{DS2}$	input	Data strobe 1 and 2—Receives timing information for data transfers. The $\overline{DS1}$ and $\overline{DS2}$ inputs must be connected together.
18	$\overline{WR}$	input	Write—Specifies direction of data transfer on the DL<7:0> lines.
14	$\overline{RDY}$	output	Ready—Indicates when the Octal ART is ready to participate in data transfer cycles.
15	$\overline{RESET}$	input	Reset—Initializes the internal logic.
57	MRESET	input	Manufacturing reset—For manufacturing use.
58	CLK	input	Clock—Clock input for timing.
62,67,2,7, 41,36,33,28	$\overline{DSR<7:0>}$	inputs	Data set ready—Monitor data set ready (DSR) signals from modems.
63,66,3,6, 40,37,32,29	$\overline{DCD<7:0>}$	inputs	Data set carrier detect—Monitor data set carrier detect (DCD) signals from modems.
49	$\overline{IRQ}$	output	Interrupt request—Requests a processor interrupt.
45-47	IRQLN<2:0>	output	Interrupt request line number—Indicates the line number of originating interrupt request.
48	IRQTxRx	output	Interrupt request transmit/receive—Indicates whether an interrupt request is for transmitting or receiving data.
61,68,1,8, 42,35,34,27	TxD<7:0>	outputs	Transmit data—Provides asynchronous bit-serial data output streams.
64,65,4,5, 39,38,31,30	RxD<7:0>	outputs	Receive data—Accepts asynchronous bit-serial data input streams.
44,26,9	V_{DD}	input	Voltage—Power supply voltage +5 Vdc.
16,59,43	V_{SS}	input	Ground—Ground reference

Data and Address

Data lines (DL<7:0>)—These lines are used for the parallel transmission and reception of data between the CPU and the Octal ART. The receivers are active when the data strobe ($\overline{DS1}$, $\overline{DS2}$) signal is asserted. The output drivers are active only when the chip select ($\overline{CS}$) signal is asserted, the data strobe ($\overline{DS1}$, $\overline{DS2}$) signal is asserted, and the write ($\overline{WR}$) signal is deasserted. The drivers will become inactive (high-impedance) within 50 nanoseconds when one or more of the following occurs: the chip select ($\overline{CS}$) signal is deasserted, the data strobe ($\overline{DS1}$, $\overline{DS2}$) signal is deasserted, or the write ($\overline{WR}$) signal is asserted.

Address (ADD<5:0>)—These lines select which Octal ART internal register is accessible through the data I/O lines (DL<7:0>) when the data strobe ($\overline{DS1}$, $\overline{DS2}$) and chip select ($\overline{CS}$) signals are asserted. Table 2 lists the addresses corresponding to each register. The receiver buffer and transmitter holding register for each line have the same address. When the ($\overline{WR}$) signal is deasserted, the address accesses the receiver buffer register and when asserted, it accesses the transmitter holding register.

Table 2 • 78808 Registers Address Selection

ADD Line*						Read/Write	Register
5	4	3	2	1	0		
0	0	0	0	0	0	Read	Line 0 Receiver Buffer
0	0	0	0	0	0	Write	Line 0 Transmitter Holding
0	0	0	0	0	1	Read	Line 0 Status
0	0	0	0	1	0	Read/Write	Line 0 Mode Registers 1, 2
0	0	0	0	1	1	Read/Write	Line 0 Command
0	0	1	0	0	0	Read	Line 1 Receiver Buffer
0	0	1	0	0	0	Write	Line 1 Transmitter Holding
0	0	1	0	0	1	Read	Line 1 Status
0	0	1	0	1	0	Read/Write	Line 1 Mode Register 1, 2
0	0	1	0	1	1	Read/Write	Line 1 Command
0	1	0	0	0	0	Read	Line 2 Receiver Buffer
0	1	0	0	0	0	Write	Line 2 Transmitter Holding
0	1	0	0	0	1	Read	Line 2 Status
0	1	0	0	1	0	Read/Write	Line 2 Mode Register 1, 2
0	1	0	0	1	1	Read/Write	Line 2 Command
0	1	1	0	0	0	Read	Line 3 Receiver Buffer
0	1	1	0	0	0	Write	Line 3 Transmitter Holding
0	1	1	0	0	1	Read	Line 3 Status
0	1	1	0	1	0	Read/Write	Line 3 Mode Register 1, 2
0	1	1	0	1	1	Read/Write	Line 3 Command
1	0	0	0	0	0	Read	Line 4 Receiver Buffer
1	0	0	0	0	0	Write	Line 4 Transmitter Holding
1	0	0	0	0	1	Read	Line 4 Status
1	0	0	0	1	0	Read/Write	Line 4 Mode Register 1, 2
1	0	0	0	1	1	Read/Write	Line 4 Command

ADD Line* 5	4	3	2	1	0	Read/Write	Register
1	0	1	0	0	0	Read	Line 5 Receiver Buffer
1	0	1	0	0	0	Write	Line 5 Transmitter Holding
1	0	1	0	0	1	Read	Line 5 Status
1	0	1	0	1	0	Read/Write	Line 5 Mode Register 1, 2
1	0	1	0	1	1	Read/Write	Line 5 Command
1	1	0	0	0	0	Read	Line 6 Receiver Buffer
1	1	0	0	0	0	Write	Line 6 Transmitter Holding
1	1	0	0	0	1	Read	Line 6 Status
1	1	0	0	1	0	Read	Line 6 Mode Register 1, 2
1	1	0	0	1	1	Read/Write	Line 6 Command
1	1	1	0	0	0	Read	Line 7 Receiver Buffer
1	1	1	0	0	0	Write	Line 7 Transmitter Holding
1	1	1	0	0	1	Read	Line 7 Status
1	1	1	0	1	0	Read/Write	Line 7 Mode Register 1, 2
1	1	1	0	1	1	Read/Write	Line 7 Command
X	X	X	1	0	0	Read	Interrupt Summary
X	X	X	1	0	1	Read	Data Set Change Summary

*X = Either 0 or 1.

Bus Transaction Control

Chip select ($\overline{CS}$)—This signal is asserted to permit data transfers through DL<7:0> to or from the internal registers. Data transfer is controlled by the data strobe ($\overline{DS1}$, $\overline{DS2}$) signal and write ($\overline{WR}$) signal.

Data strobe ($\overline{DS1}$, $\overline{DS2}$)—The data strobe inputs ($\overline{DS1}$ and $\overline{DS2}$) must be connected together. This input receives timing information for data transfers. During a write cycle, the CPU asserts the data strobe signal when valid output data is available and deasserts the data strobe signal before the data is removed. During a read cycle, the CPU asserts the data strobe signal and the Octal ART transfers the valid data. When the data strobe signal is deasserted, DL<7:0> become a high impedance.

Write ($\overline{WR}$)—The write ($\overline{WR}$) signal specifies the direction of data transfer on the DL<7:0> pins by controlling the direction of their transceivers. If the $\overline{WR}$ signal is asserted during a data transfer (the $\overline{CS}$, $\overline{DS1}$, and $\overline{DS2}$ signals asserted), the Octal ART is receiving data from DL<7:0>. If the $\overline{WR}$ signal is deasserted during a write data transfer, the Octal ART is driving data onto DL<7:0>.

Interrupt Request

Interrupt request $\overline{IRQ}$)—The $\overline{IRQ}$ pin is an open drain output. The integral interrupt scanner asserts the $\overline{IRQ}$ signal when it has detected an interrupt condition on one of the eight serial data lines.

Interrupt request transmit/receive (IRQTxRx)—This signal indicates when the interrupt scanner in the Octal ART stops and asserts $\overline{IRQ}$ because of a transmitter interrupt condition (the IRQTxRx signal is asserted) or because of a receiver interrupt condition (the IRQTxRx signal is deasserted). The signal is valid only while $\overline{IRQ}$ is asserted. The state of IRQTxRx signal also appears as bit 0 of the interrupt summary register.

Interrupt request line number (IRQLN<2:0>)—These lines indicate the line number at which the Octal ART interrupt scanner stopped and asserted the interrupt request ($\overline{IRQ}$) signal. The number on these lines is valid only while the $\overline{IRQ}$ signal is asserted. The IRQLN2 is the high-order bit and IRQLN0 is the low-order bit. The state of these signals also appears as bits in the interrupt summary register: IRQLN2 as bit 3, IRQLN1 as bit 2, and IRQLN0 as bit 1. Table 3 shows the line numbers corresponding to settings of IRQLN<2:0>.

Table 3 • 78808 Interrupt Request Line Asignments

IRQ Line 2	1	0	Line
0	0	0	0
0	0	0	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Serial Data

Transmit data (TxD<7:0>)—These outputs transmit the asynchronous bit-serial data streams. They remain at a high level when no data is being transmitted and a low level when the TxBRK bit in the associated line's command register is set.

Receive data (RxD<7:0>)—These lines accept asynchronous bit-serial data streams. The input signals must remain in the high state for at least one-half bit time before a high-to-low transition is recognized. (A high-to-low transition is required to signal the beginning of a "start" bit and initiate data reception.)

Modem Signals

Data set ready ($\overline{DSR}$<7:0>)—These eight input pins, one for each serial data line on the 78808, are typically connected via intervening level converters to the data set ready outputs of modems. A TTL low at a $\overline{DSR}$ pin causes the DSR bit (bit 7) in the corresponding line's status register to be asserted. A TTL high at a $\overline{DSR}$ pin causes the DSR bit in the corresponding line's status register to be deasserted. A change of this input from high-to-low, or low-to-high, causes the assertion of the data set change (DSCHNG) bit that corresponds to this line in the data set change summary register. Changes from one state to the other and back again that occur within one microsecond may not be detected.

Carrier detect (DCD<7:0>)–These eight input pins, one for each serial data line of the Octal ART, are typically connected through intervening level converters to the received line signal detect (also called carrier detect) outputs of modems. A TTL low at a $\overline{DCD}$ pin causes the DCD bit of the corresponding line's status register to be deasserted. A change of this input from high-to-low, or low-to-high, causes the assertion of the data set change (DSCHNG) bit corresponding to this line in the data set change summary register. Changes from one state to the other and back again that occur within one microsecond may not be detected.

General Control Signals

Ready ($\overline{RDY}$)—The $\overline{RDY}$ pin is an open drain output. Upon detecting a negative transition of chip select ($\overline{CS}$), the Octal ART asserts the $\overline{RDY}$ signal to indicate readiness to take part in data transfer cycles. The $\overline{RDY}$ signal deasserts after the trailing edge of $\overline{CS}$.

Reset ($\overline{RESET}$)—When the $\overline{RESET}$ input is asserted, the TxD<7:0> lines are asserted and all internal status bits listed in the "Architecture Summary" discussion are cleared.

Manufacturing reset (MRESET)—This signal is for manufacturing use only and the input should be connected to ground for normal operation.

Miscellaneous Signals

Clock in (CLK)—All baud rates and internal clocks are derived from this input. Normal operating frequency is 4.9152 MHz ±0.1 percent and duty cycle is 50 percent ±5 percent.

Power and Ground

Voltage (V_{DD})—Power supply 5 Vdc

Ground (V_{SS})–Ground reference

• Architecture Summary

The Octal ART functions as a serial-to-parallel, parallel-to-serial converter/controller. It can be programmed by a microprocessor to provide different characteristics for each of its eight serial data lines (stop bits, parity, character length, split baud rates, etc.).

Each serial line functions the same as a one-line UART-type device thereby reducing the number of chips and conserving space on communication devices that require multiple communications lines.

An integral interrupt scanner checks for device interrupt conditions on the eight lines. Its scanning algorithm gives priority to receivers over transmitters. The scanner can also check for interrupts resulting from changes in modem control signals $\overline{DSR}$ and $\overline{DCD}$.

Line-specific Registers

Each of the eight serial data lines in the Octal ART has a set of registers for buffering data into and out of the line and for external control of the line's characteristics. These registers are selected for access by setting the appropriate address on lines ADD<5:0>. Lines ADD<5:3> select one of the eight data lines. Lines ADD<2:0> select the specific register for that line. Refer to Table 2 for the register address assignments.

Receiver buffer register—Each line's receiver consists of a character assembly register and a two-entry FIFO that is the receiver buffer register. When the RxEN bit in a line's command register is set, received characters are moved automatically into the line's receiver buffer as soon as they have been deserialized from the associated communications line. When there are characters in this FIFO, the RxRDY bit is set in the status register for the line.

The assertion of the RxRDY signal for a line that already has the RxIE bit of its command register set causes the interrupt scanner logic to stop and generate an interrupt condition (the $\overline{IRQ}$ signal is asserted). When the receiver buffer is read, the interrupt condition is cleared (the $\overline{IRQ}$ signal is deasserted) and the interrupt scanner resumes operation.

If there is another entry in a line's FIFO, the RxRDY bit remains asserted. When the interrupt scanner reaches this line again, the assertion of RxRDY causes the scanner to halt and assert the $\overline{IRQ}$ again.

Asserting the $\overline{RESET}$ signal or clearing the RxEN bit initializes the receiver logic of Octal ART. The RxRDY flag is cleared and the receiver buffer register outputs become undefined. Any data in the FIFO at that time is lost.

Transmitter holding register—Each line has a writable transmitter holding register. When the TxEN bit in the line's command register is set, characters are moved automatically from the output of this register into the transmitter serialization logic whenever the serialization logic becomes idle.

When this register is empty, the TxRDY bit in the line's status register is set. If the transmitter interrupt enable (TxIE) bit in the line's command register is also set, the interrupt scanner logic halts and generates an interrupt condition. If a character is then loaded into the register, the interrupt is cleared and the scanner resumes operation.

Assertion of the $\overline{RESET}$ signal initializes the transmitter logic of the Octal ART. The TxRDY flag is cleared and the transmitter holding register's contents are lost. The transmitter enable (TxEN) bit in the line's command register is also cleared by $\overline{RESET}$. If at the end of the reset process, the TxEN is reasserted and TxRDY bit is reasserted. Software clearing of TxEN alone produces results different from the full $\overline{RESET}$ in that the transmitter holding register's contents are not lost; they are transmitted when TxEN is set again.

Status register—Each line has a read-only status register that provides information about the current state of the given line. This register indicates a line's readiness for transmission or reception of data and flags error conditions in its bit fields. Figure 3 shows the format of the status register. Table 4 lists the flag bits in each status register.

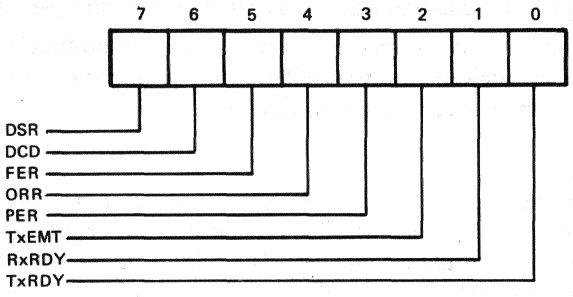


Figure 3 • 78808 Status Registers (Line 0-7) Format

Table 4 • 78808 Status Registers (Lines 0-7) Description

Bit	Description
7	DSR (Data set ready)—This bit is the inverted state of the $\overline{DSR}$ line.
6	DCD (Data set carrier detect)—This bit is the inverted state of the $\overline{DCD}$ line.
5	FER (Frame error)—Set when the received character currently displayed in the receiver buffer register was not framed by a stop bit. Only the first stop bit is checked to determine that a framing error exists. Subsequent reading of the receiver buffer register that indicates all zeros (including the parity bit, if any) can be interpreted as a Break condition. This bit is cleared by clearing RxEN (bit 2) of the command register, by asserting the $\overline{RESET}$ input, or by setting the reset error RERR (bit 4) of the command register.
4	ORR (Overrun error)—Set when the character in the receiver buffer register was not read before another character was received. Cleared by clearing RxEN (bit 2) of the command register, by asserting the the $\overline{RESET}$ input, or by setting reset error RERR (bit 4) of the command register.
3	PER (Parity error)—If parity is enabled and this bit is set, the received character in the receiver buffer register has an incorrect parity bit. This bit is cleared by clearing RxEN (bit 2) of the command register, by asserting the $\overline{RESET}$ input, by setting reset error RERR (bit 2) of the command register, or by reading the current character in the receiver buffer register.
2	TxEMT (Transmitter empty)—Set when the transmitter serialization logic for the associated line has completed transmission of a character, and no new character has been loaded into the transmitter holding register. Cleared by loading the transmitter holding register, by clearing TxEN (0) of the command register, or by asserting the $\overline{RESET}$ input.
1	RxRDY (Receiver buffer ready)—When set, a character has been loaded into the FIFO buffer from the deserialization logic. Cleared by reading the receiver buffer register, by clearing RxEN (bit 2) in the command register, or by asserting the $\overline{RESET}$ input.
0	TxRDY (Transmitter holding register ready)—When set, this bit indicates that the transmitter holding register is empty. Cleared when the program has loaded a character into the transmitter holding register, when the transmitter for this line is disabled by clearing TxEN (bit 0) in the command register, or by asserting the $\overline{RESET}$ input. This bit is initially set when the transmitter logic is enabled by the setting of TxEN (bit 0) and the transmitter holding register is empty. This bit is not set when the automatic echo or remote loopback modes are programmed. Data can be overwritten if a consecutive write is performed while TxRDY is cleared.

Mode registers 1 and 2—These read/write registers control the attributes (including parity, character length, and line speed) of the communications line.

Each of the eight communications lines has two of these registers, both accessed by the same address on ADD<5:0>. Successive access operations (either read or write, in any combination) alternate between the two registers at that address by use of an internal pointer. The first operation addresses mode register 1, the next address mode register 2, and another after that would recycle the pointer to mode register 1. The pointer is reset to point to mode register 1 by $\overline{\text{RESET}}$ or by a read of the command register for this line. These registers should not be accessed by bit-oriented instructions that do read/modify/write cycles such as the PDP-11 BIS, BIC, and BIT instructions.

Figure 4 shows the format of mode registers 1 and Table 5 describes the function of the register information.

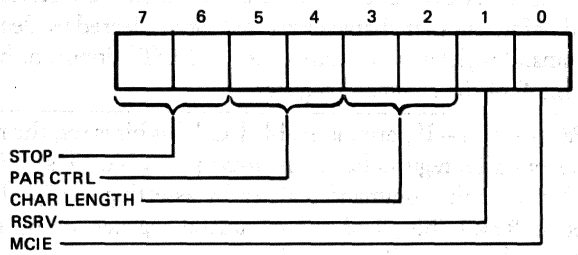


Figure 4 • 78808 Mode Registers 1 (Line 0-7) Format

Table 5 • 78808 Mode Registers 1 (Lines 0-7) Description

Bit	Description
7,6	STOP—These bits determine the number of stop bits that are appended to the transmitted characters as follows. These bits are cleared by asserting the $\overline{\text{RESET}}$ input.

Bits 7	Bits 6	Stop Bits
0	0	Invalid
0	1	1.0
1	0	1.5
1	1	2.0

Bit	Description
5,4	PAR CTRL (Parity control)—These bits determine parity as follows and are cleared by asserting the $\overline{\text{RESET}}$ input. X = either 1 or 0.

Bits 5	Bits 4	Parity Type
1	1	Even
0	1	Odd
X	0	Disabled

Bit	Description
3,2	CHAR LENGTH (Character length)—These bits determine the length (excluding start bit, parity, and stop bits) of the characters received and sent. Received characters of less than 8 bits are "right aligned" in the receiver buffer with unused high-order bits equal to zero. Parity bits are not shown in the receiver buffer. The character length bits are cleared by asserting the $\overline{\text{RESET}}$ input. The character length bits are defined as follows:

Bits 3	Bits 2	Bit Length
0	0	5
0	1	6
1	0	7
1	1	8

1	RSRV (Reserved and cleared by asserting the $\overline{\text{RESET}}$ input.)
0	MCIE (Modem control interrupt enable)—When set and RxIE (bit 5) of the command register is set, the modem control interrupts are enabled. Refer to the Interrupt Scanner and Interrupt Handling information. Cleared by asserting the $\overline{\text{RESET}}$ input.

Figure 5 shows the format of mode registers 2 and Table 6 indicates the baud rate selections of the register. Bits 7 through 4 of mode register 2 control the transmitter baud rate and bits 3 through 0 control the receiver baud rate. These registers are cleared by asserting $\overline{\text{RESET}}$ input.

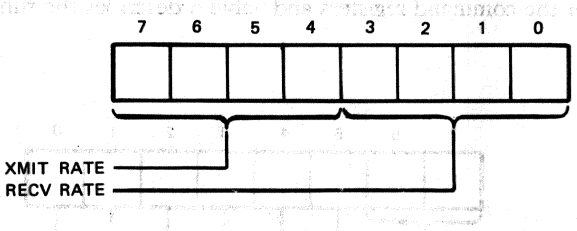


Figure 5 ▪ 78808 Mode Registers 2 (Line 0-7) Format

Table 6 • 78808 Mode Registers 2 (Lines 0-7) Description

Bit	Description										
7:0	XMIT RATE/RECV RATE (Transmitter/Recever Rate)—Selects the baud rate of the transmitter (bits 7:4) and receiver (bits 3:0) as follows:										
	Transmitter Bits				**Receiver Bits**				**Nominal**	**Actual**	**Error***
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Rate**	**Rate**	**(percent)**
	0	0	0	0	0	0	0	0	50	same	—
	0	0	0	1	0	0	0	1	75	same	—
	0	0	1	0	0	0	1	0	110	109.09	0.826
	0	0	1	1	0	0	1	1	134.5	133.33	0.867
	0	1	0	0	0	1	0	0	150	same	—
	0	1	0	1	0	1	0	1	300	same	—
	0	1	1	0	0	1	1	0	600	same	—
	0	1	1	1	0	1	1	1	1200	same	—
	1	0	0	0	1	0	0	0	1800	1745.45	3.03
	1	0	0	1	1	0	0	1	2000	2021.05	1.05
	1	0	1	0	1	0	1	0	2400	same	—
	1	0	1	1	1	0	1	1	3600	3490.91	3.03
	1	1	0	0	1	1	0	0	4800	same	—
	1	1	0	1	1	1	0	1	7200	6981.81	3.03
	1	1	1	0	1	1	1	0	9600	same	—
	1	1	1	1	1	1	1	1	19200	same	—

*The frequency of the clock input (CLK) is 4.9152 MHz. The clock input may vary by 0.1 percent. This variance results in an error that must be added the error listed.

Command register–These read/write registers control various functions on the selected line. Figure 6 shows the format of the command registers and Table 6 describes the function of the register information.

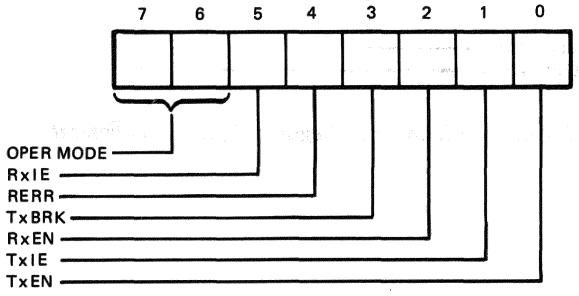


Figure 6 • 78808 Command Registers (Line 0-7) Format

Table 7 • 78808 Command Registers (Lines 0-7) Description

Bit	Description
7,6	OPER MODE (Operating mode)—These bits control the operating mode of the channel as follows. These bits are cleared by asserting the $\overline{\text{RESET}}$ input. **Bit 7 / Bit 6 — Operating Mode** 0 0 — Normal operation 0 1 — Automatic echo 1 0 — Local loopback 1 1 — Remote loopback
5	RxIE (Receiver interrupt enable)—When set, the RxRDY flag (bit 1) of the status register for this line will generate an interrupt.
4	RERR (Reset error)—When set, this bit clears the framing error, overrun error, and parity error of the status register associated with this line. This bit must be cleared before errors that occur will be recorded in the Status register. This bit is cleared by asserting the $\overline{\text{RESET}}$ input (not self-clearing).
3	TxBRK (Transmit break)—When set, this bit forces the appropriate TxD<7:0> line to the spacing state at the conclusion of the character presently being transmitted. When the program clears this bit, normal operation is restored, and any character pending in the transmitter holding register is moved into the serialization logic and transmitted. The minimum break length obtainable is twice the character length plus 1 bit time. The maximum break length depends on the amount of time between the program setting and clearing this bit, but is an integral number of bit times. This bit is cleared by asserting the $\overline{\text{RESET}}$ input.
2	RxEN (Receiver enable)—When set, this bit enables the receiver logic. When cleared, it stops the assembling of the received character, clears all receiver error bits and the RxRDY (bit 1) of the status register, clears any receiver interrupt conditions associated with this line, and initializes all receiver logic. This bit is cleared by asserting the $\overline{\text{RESET}}$ input.
1	TxIE (Transmit interrupt enable)—When set, the state of the associated TxRDY flag (bit 0) of the status register is made available to the interrupt scanner logic. When the interrupt scanner logic scans this line, it determines if the TxRDY flag is asserted and generates an interrupt by asserting the $\overline{\text{IRQ}}$ signal.
0	TxEN (Transmitter enable)—When set, this bit enables the transmitter logic. When cleared, it inhibits the serialization of the characters that follow but the serialization of the current character is completed. It also clears the TxRDY flag (bit 0) of the status register, clears any transmitter interrupt conditions associated with this line, and initializes all transmitter logic except that associated with the transmitter holding register. The character in the transmitter holding register is retained so that XON/XOFF situations can be properly processed. This bit is cleared by asserting the $\overline{\text{RESET}}$ input.

Bits 5 through 0 enable the line's receiver and transmitter, enable handling of interrupts, initiate the transmission of break characters, and reset error bits for the line. Refer to "Interrupt Scanner" and "Interrupt Handling" paragraphs for detailed interrupt information. Bits 7 and 6 control the operating mode of the line. The four modes that can be set are

- Normal operation—The serial data received is assembled in the receiver logic and transferred in parallel to the receiver buffer register. (The RxEN bit must be set.) Data to be transmitted is loaded in parallel into the transmitter holding register, then automatically transferred into the transmitter logic and serialized for transmission. (The TxEN bit must be set.)
- Automatic echo—The serial data received is assembled into parallel in the receiver logic (the RxEN bit must be set) and transferred to the receiver buffer register. Arriving serial data is also routed to the line's TxD<n> pin for serial output. TxEN is ignored and the transmitter logic is disabled. TxRDY flags and TxEMT indications are cleared. No transmitter interrupts are generated.
- Local loopback—The serial data from the RxD<n> input is ignored and the receiver serial input receives data from the transmitter serial output. That data is assembled into parallel form in the receiver logic (the RxEN bit must be set) and transferred to the receiver buffer register where it can be read by the program. Data to be transmitted to the receiver is loaded in parallel form into the transmitter holding register from which it is automatically moved into the transmitter logic and serialized for transmission. (The TxEN bit must be set.) The transmission goes only to the receiver serial input; the TxD<n> output is held high. As in normal operation, transmission and reception baud rates are controlled by the transmitter speed and receiver speed entries in mode register 2.
- Remote loopback—The serial data received on the RxD<n> line is returned to the TxD<n> line without further action. No data is received or transmitted. The RxRDY, TxRDY, and TxEMT flags are disabled. The TxEN and RxEN bits of the command register are held cleared, causing the transmitter and receiver logic to be disabled.

Summary Registers

The Octal ART contains two registers that summarize the current status of all eight serial data lines, making it possible to determine that a line's status has changed with a single read operation. These registers are selected for access by setting the appropriate address on pins ADD<2:0>. Because the registers are shared by eight serial lines, the line-selection bits (ADD<5:3>) are ignored when these registers are accessed. Refer to "Interrupt Scanner and Interrupt Handling" for detailed interrupt information.

Interrupt summary register—This read-only register indicates that a transmitter or receiver interrupt condition has occurred, and indicates the line number that generated the interrupt. Figure 7 shows the format of the interrupt summary register and Table 8 describes register information.

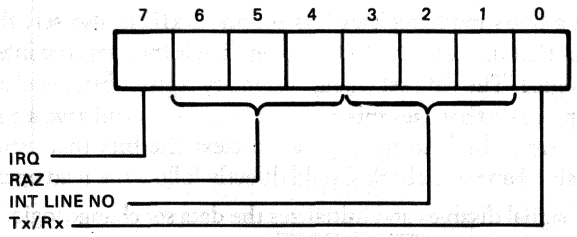


Figure 7 • 78808 Interrupt Summary Register Format

Table 8 • 78808 Interrupt Summary Register Description

Bit	Description
7	IRQ (Interrupt request)—When set, this bit indicates that the interrupt scanner has found an interrupting condition among the eight serial lines of the Octal ART. These conditions also result in the Octal ART asserting the $\overline{IRQ}$ signal.
6:4	RAZ (Read as zero)—Not used
3:1*	INT LINE NO (Interrupting line number)—These bits indicate the line number upon which an interrupting condition was found. These bits correspond to the IRQLN<2:0> signals—(bit 3=IRQLN<2>, bit 2=IRQLN<1>, and bit 1=IRQLN<0>. Refer to Table 3.
0*	Tx/$\overline{Rx}$ (Transmit/receive)—This bit indicates whether the interrupting condition was caused by a transmitter (Tx/$\overline{Rx}$ equals 1) or a receiver (Tx/$\overline{Rx}$ equals 0). This bit corresponds to the IRQTxRx signal of the Octal ART and is set when IRQTxRx is asserted.

*Bits 3-0 above represent the outputs of a free-running counter and are valid only when bit 7 is set.

Data set change summary register—When the $\overline{DSR}$ or $\overline{DCD}$ inputs that are associated with a line change state, the bit corresponding to that line in this read-only register is set. The current state of the $\overline{DSR}$ and $\overline{DCD}$ inputs can then be obtained from that line's status register. If the state of a line changes twice within one microsecond, The change in state may not be detected. Figure 8 shows the format of the data set change summary register.

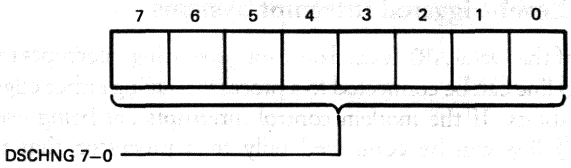


Figure 8 • 78808 Data Set Change Summary Register Format

When the MCIE bit in a line's mode register 1 is set and RxIE is also set, the modem control interrupts are enabled for that line. If DSCHNG for that line is then set, the interrupt scanner will halt and assert the $\overline{IRQ}$ signal. The data set change summary register bits are cleared by writing a 1 into the bit position. A program that uses this register should read and save a copy of its contents. The copy can then be written back to the register to clear the bits that were set. The system interrupts should be disabled and writeback should directly follow the read operation.

Assertion of the $\overline{RESET}$ signal disables and initializes the data set change logic. When the $\overline{RESET}$ signal is deasserted, future changes in $\overline{DSR}$ and $\overline{DCD}$ are reported as they occur.

• Interrupt Scanner and Interrupt Handling

The interrupt scanner is a four-bit counter that sequentially checks lines 0 through 7 for a receiver interrupt (counter positions 0-7) and then checks the lines in the same order for a transmitter interrupt (counter positions 8-15). If the scanner detects an interrupt condition, it stops and the $\overline{IRQ}$ signal is asserted. An interrupt must be serviced by software or no other interrupt request can be posted.

The scanner determines that a line has a receiver interrupt if the line's receiver buffer is ready and receiver interrupts are enabled for that line (RxRDY and RxIE = 1) or either of the line's modem status signals has changed state and both receiver and modem control interrupts are enabled for that line (DSCHNG and RxIE and MCIE = 1).

The scanner determines that a line has a transmitter interrupt if the line's transmitter holding the register is empty and transmitter interrupts are enabled for that line (TxRDY and TxIE = 1).

When the scanner detects an interrupt, it reports the line number on the IRQ<2:0> lines. The IRQTxRx signal is asserted for a transmitter interrupt and deasserted for a receiver interrupt. The appropriate bits are also updated in the interrupt summary register. The $\overline{IRQ}$ line is deasserted and the scanner is restarted for each of the following three types of interrupt conditions.

- Reading the receiver buffer or resetting the RxIE bit of the interrupting line for the first type of receiver interrupt previously described.
- Resetting the MCIE, RxIE, or DSCHNG bit of the interrupting line for the second type of receiver interrupt previously described.
- Loading the transmitter holding register or resetting the TxIE bit of the interrupting line for transmitter interrupts.

If the scanner was originally stopped by a receiver interrupt condition, the scanner resumes sequential operation from where it stopped, thus providing receivers with equal priority. If the scanner was stopped by a transmitter condition, the scanner restarts from position 0 (line 0's receiver), thus giving receivers priority over transmitters.

• Edge-triggered and Level-triggered Interrupt Systems

If the interrupt system of the Octal ART is used only for generating interrupts for thee RxRDY and/or TxRDY flags, the $\overline{IRQ}$ line can be connected to a processor having either edge-triggered or level-triggered interrupt capability. If the modem control interrupts are being used (MCIE in mode register 1 = 1), the $\overline{IRQ}$ line can be connected only to a processor that uses level-triggered interrupts.

Modem Handling

The TxEMT (transmitter empty) bit of the status register is typically used to indicate when a program can disable the transmission medium, as when deasserting the request-to-send line of a modem. A typical program will load the last character for transmission and then monitor the TxEMT bit of the status register.

The assertion of the TxEMT bit to indicate that transmission is complete may occur a substantial time after the loading of the last character. After the last character is loaded, one character is in the transmitter holding register and one character is in the serialization logic. Therefore, it will be two character times before the transmission process is completed. Waiting for the TxRDY signal to assert before monitoring the TxEMT status shortens this by one character time because the TxRDY status bit indicates that there are no characters in the transmitter holding register. The times involved are calculated by taking the reciprocal of the baud rate being used, multiplying by the number of bits per character (a start bit—5, 6, 7, or 8 data bits; plus parity bit if enabled; and 1, 1.5, or 2 stop bits), and multiplying by either two characters or one, depending on when TxEMT monitoring begins.

Specifications

The mechanical, electrical, and environmental characteristics and specifications for the Octal ART are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Temperature: 0°C to 70°C
- Power supply voltage (V_{DD}): 4.75 V to 5.25 V

Mechanical Configuration

The physical dimensions of the 68-pin package are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Power supply voltage (V_{DD}): 7.0 V
- Input or output voltage applied: –5 V to 7.0 V
- Storage temperature: –65°C to 125°C

Recommended Operating Conditions

- Power supply voltage (V_{DD}): 5 V ±5%
- Operating temperature (T_A): 0°C to 75°C

dc Electrical Characteristics

The dc electrical characteristics of the Octal ART for the operating voltage and temperature ranges specified are listed in Table 9.

Table 9 • 78808 dc Electrical Characteristics

Symbol	Parameter	Test Condition	Requirements Min.	Requirements Max.	Units
V_{IH}	High-level input voltage		2.0	—	V
V_{IL}	Low-level input voltage		—	0.8	V
V_{OH}	High-level output voltage	V_{DD} = Min. I_{OH} = −3.5 mA for DL<7:0> I_{OH} = −2.0 mA for all remaining output except $\overline{IRQ}$ and $\overline{RDY}$	2.4	—	V
V_{OL}	Low-level output voltage	V_{DD} = Min. I_{OL} = 5.5 mA for DL<7:0> I_{OL} = 3.5 mA for all remaining outputs	—	0.4	V
I_{IH}	Input current at maximum input voltage	V_{DD} = Max. V_I = V_{DD} (Max.)	—	10	µA
I_{IL}	Input current at minimum input voltage	V_{DD} = Max. V_I = 0.0 V	—	−10	µA
I_{OS}[1]	Short-circuit output current for DL<7:0>	V_{DD} = Max.	−50	−180	mA
	all remaining outputs except $\overline{IRQ}$ and $\overline{RDY}$		−30	−110	mA
I_{OZL}[2]	Three-state output current	V_{DD} = Max. V_O = 0.4 V	—	10	µA
I_{OZH}[2]	Three-state output current	V_{DD} = Max. V_O = 2.4 V	—	10	µA
I_{DD}	Supply current	V_{DD} = Max. T_A = 0°	—	240	mA
C_{in}	Input capacitance		—	4.0	pF
C_{IO}[3]	Input/output capacitance		—	5.0	pF

[1]No more than one output should be short circuited at a time and the duration of the short should not exceed 1 second.
[2]All three-state output drivers are wired in an I/O configuration. The parameters include the driver and input receiver leakage currents.
[3]The parameters include the capacitive loads of the output driver and the input receiver.

ac Electrical Characteristics

The device propagation delays specified in the ac characteristics figures and tables assume the loading conditions shown in Figure 9.

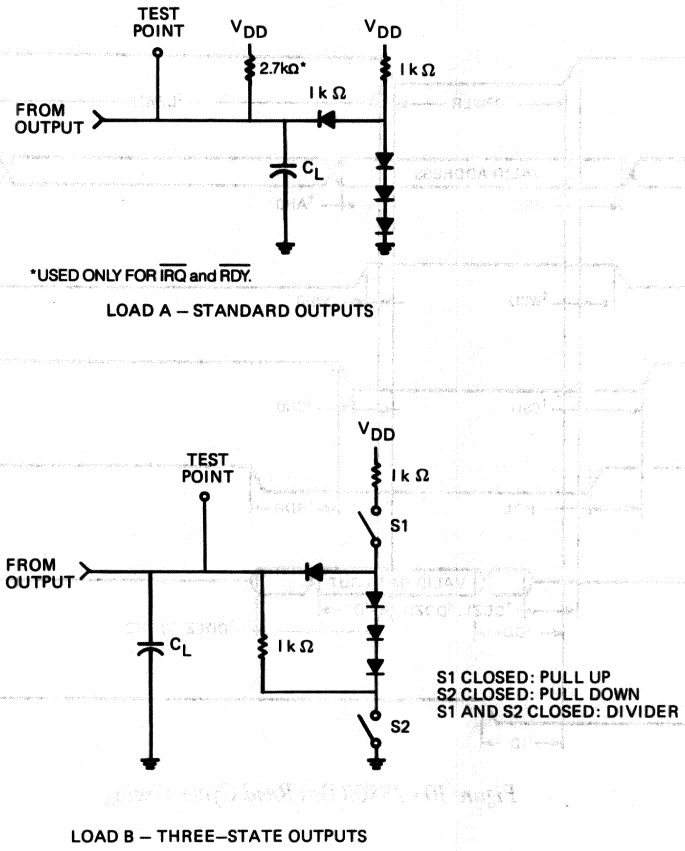


Figure 9 ▪ 78808 Output Load Circuits

Timing Parameters

Figure 10 shows the signal timing for a read cycle to transfer information from the Octal ART to the processor. Figure 11 shows the signal timing for a write cycle to transfer information from the processor to the Octal ART. Table 11 lists the timing parameters for the read and write cycles.

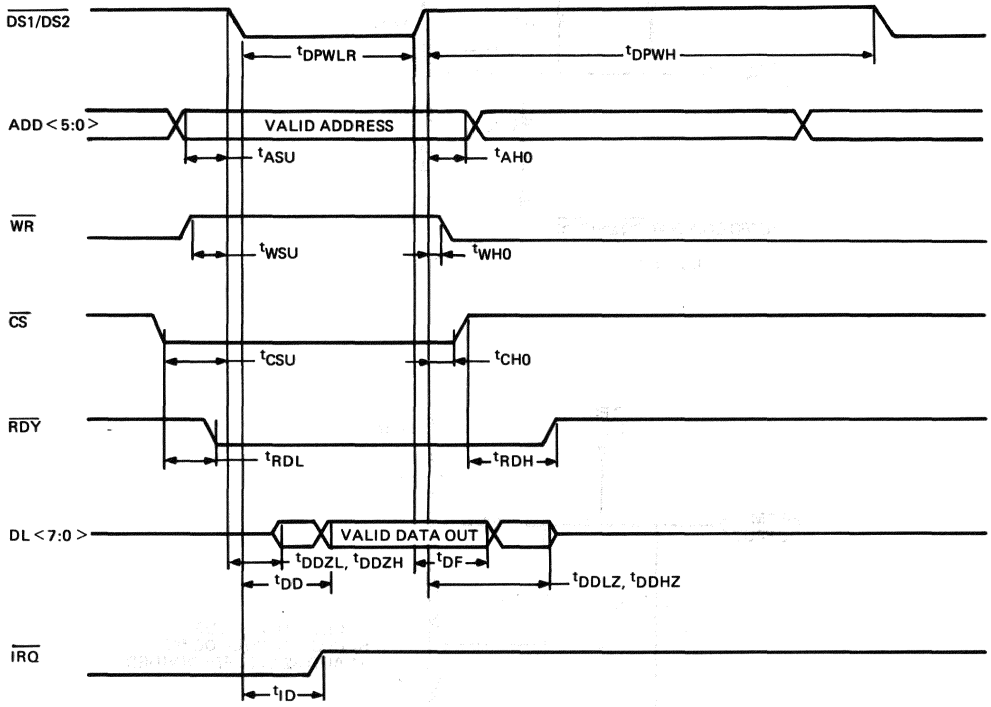


Figure 10 ▪ 78808 Bus Read Cycle Timing

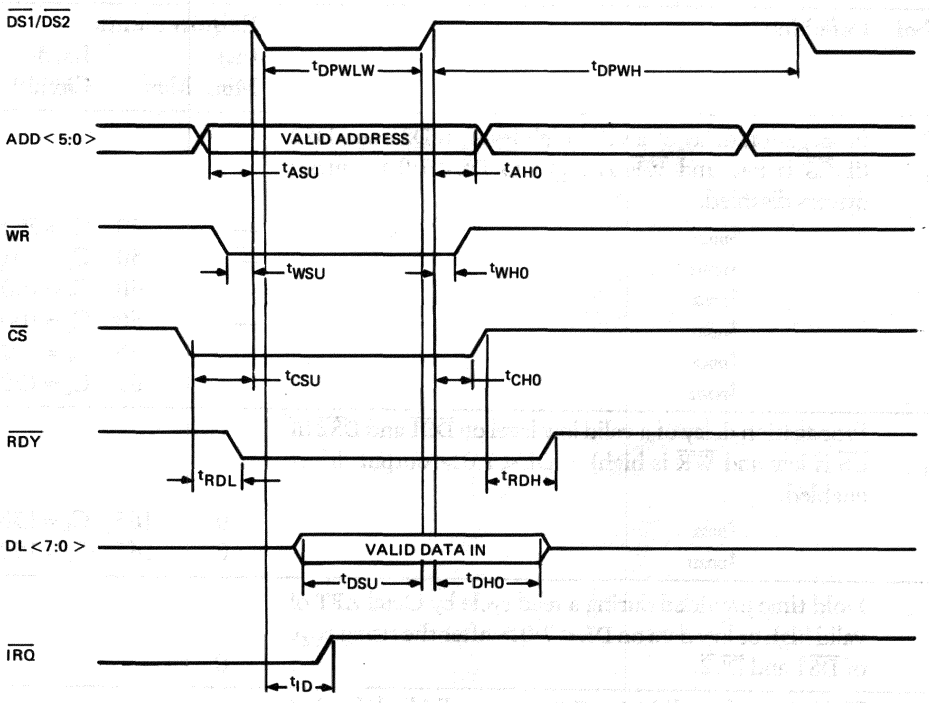


Figure 11 ▪ 78808 Bus Write Cycle Timing

Table 10 ▪ 78808 Bus Read and Write Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.	Load Circuit[1]
t_{AHO}	Hold time of a valid ADD<5:0> to a valid high level of $\overline{DS1}$ and $\overline{DS2}$.	10	—	
t_{ASU}	Setup time of a valid ADD<5:0> to the falling edge of $\overline{DS1}$ and $\overline{DS2}$.	30	—	
t_{CHO}	Hold time of a valid low level of $\overline{CS}$ to a valid high level of $\overline{DS1}$ and $\overline{DS2}$.	10	—	
t_{CSU}	Setup time of a valid low level of $\overline{CS}$ to the falling edge of $\overline{DS1}$ and $\overline{DS2}$.	30	—	
t_{DD}	Propagation delay of a valid low level on $\overline{DS1}$ and $\overline{DS2}$ (if $\overline{CS}$ is low and $\overline{WR}$ is high) to valid high or low data on DL<7:0>.	—	165	$C_L = 150$ pF

Symbol	Definition	Requirements (ns) Min.	Max.	Load Circuit[1]
t_{DDLZ}[2] t_{DDHZ}	Propagation delay of a valid high level on $\overline{DS1}$ and $\overline{DS2}$ (if $\overline{CS}$ is low and $\overline{WR}$ is high) to DL<7:0> output drivers disabled.			
	t_{DDLZ}	—	50	$C_L = 50$ pF
	t_{DDHZ}	—	50	$C_L = 50$ pF
	t_{DDLZ}	—	60	$C_L = 100$ pF
	t_{DDHZ}	—	60	$C_L = 100$ pF
	t_{DDLZ}	—	65	$C_L = 150$ pF
	t_{DDHZ}	—	65	$C_L = 150$ pF
t_{DDZL} t_{DDZH}	Propagation delay of a valid low level on $\overline{DS1}$ and $\overline{DS2}$ (if $\overline{CS}$ is low and $\overline{WR}$ is high) to DL<7:0> output driver enabled.			
	t_{DDZL}	0	165	$C_L = 150$ pF
	t_{DDZH}	0	165	$C_L = 150$ pF
t_{DF}	Hold time provided during a read cycle by Octal ART of valid high or low data on DL<7:0> after the rising edge of $\overline{DS1}$ and $\overline{DS2}$.	0	—	
t_{DHO}	Hold time of a valid DL<7:0> to a valid high level of $\overline{DS1}$ and $\overline{DS2}$.	30	—	
t_{DPWH}	Pulse width high of $\overline{DS1}$ and $\overline{DS2}$.	450	—	
t_{DPWLR}	Pulse width low of $\overline{DS1}$ and $\overline{DS2}$ when $\overline{WR}$ is high (read operation). Refer to timing parameter t_{DPWLW} also.	180	10,000	
t_{DPWLW}	Pulse width low of $\overline{DS1}$ and $\overline{DS2}$ when $\overline{WR}$ is low (write operation). Refer to timing parameter t_{DPWLR} also.	130	10,000	
t_{DSU}	Setup time of a valid DL<7:0> to the falling edge of $\overline{DS1}$ and $\overline{DS2}$.	130	—	
t_{ID}[3]	Propagation delay of a valid low level on $\overline{DS1}$ and $\overline{DS2}$ (if $\overline{CS}$ is low) to a high level on $\overline{IRQ}$.	—	635	$C_L = 50$ pF
t_{RDH}[4]	Propagation delay of a valid high level of $\overline{CS}$ to a valid high level on $\overline{RDY}$.	—	210	$C_L = 50$ pF
t_{RDL}	Propagation delay of a valid low level on $\overline{CS}$ to a valid low level on $\overline{RDY}$.	—	90	$C_L = 50$ pF
t_{WHO}	Hold time of a valid high or low level of $\overline{WR}$ to a valid high level of $\overline{DS1}$ and $\overline{DS2}$.	10	—	
t_{WSU}	Setup time of a valid high or low level of $\overline{WR}$ to the falling edge of $\overline{DS1}$ and $\overline{DS2}$.	30	—	

[1]Refer to Figure 9 for the load circuits used with these measurements.

[2]The t_{DDLZ} and t_{DDHZ} parameters are measured with $C_L = 150$ pF. The values of t_{DDLZ} and t_{DDHZ} for $C_L = 50$ pF and $C_L = 100$ pF have been derived for user convenience.
[3]Total rise time depends on internal delay plus the pullup delay introduced by the external resistor being used. The t_{ID} parameter can be calculated by the following: $t_{ID} = 500 + RC_L$ where R = value of the resistor that connects to capacitor C_L in load A, Figure 9.
[4]Total rise time depends on internal delay plus the pullup delay introduced by the external resistor being used. The t_{RDH} parameter can be calculated by the following: $t_{RDH} = 75 + RC_L$ where R = value of the resistor that connects to capacitor C_L in load A, Figure 9.

Figures 12 shows the signal timing for the clock input, interrupt timing, effect of the $\overline{RESET}$ input on data strobe, data set carrier detect (DCD) and data set ready (DSR) input timing, and the transmit data output timing. Table 11 lists the timing parameters for Figures 12.

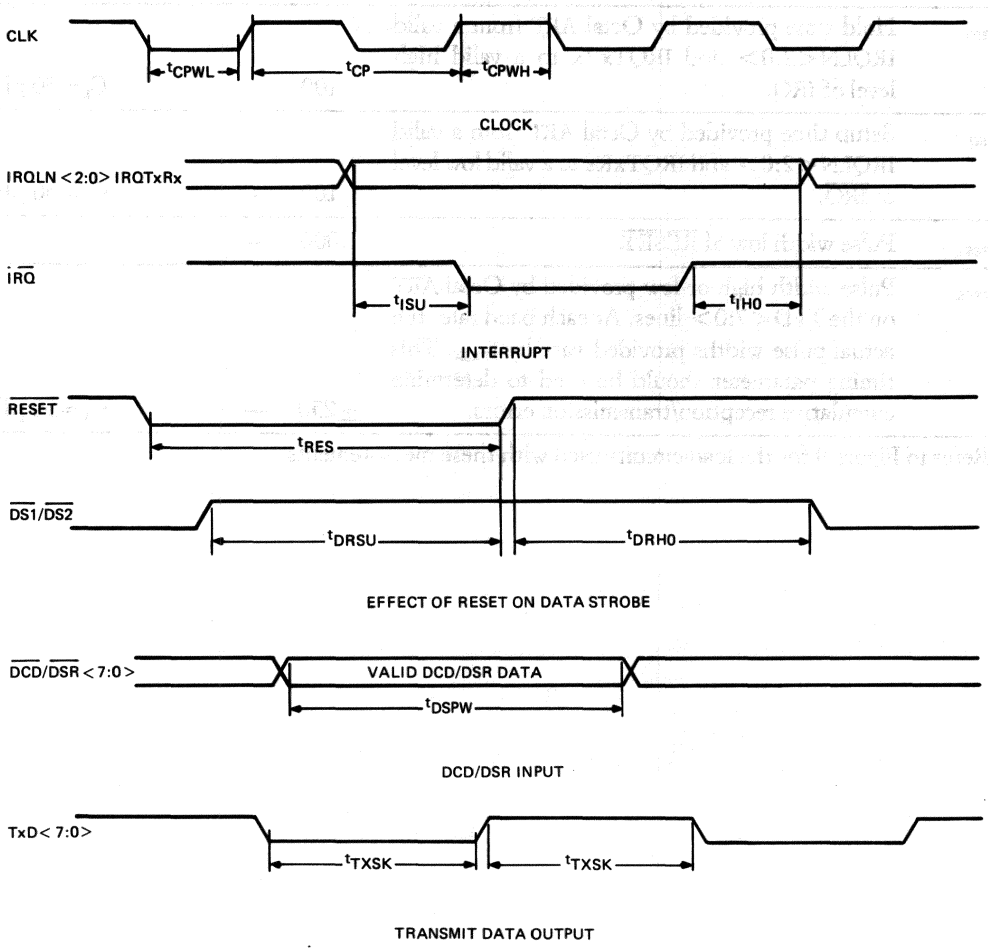


Figure 12 • 78808 Miscellaneous Signal Timing

Table 11 • 78808 Miscellaneous Write Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.	Load Circuit[1]
t_{CP}	Period of CLK.	203.45	(4.9152 MHz)	
t_{CPWH}	Pulse width high of CLK.	95	—	
t_{CPWL}	Pulse width low of CLK.	95	—	
t_{DRHO}	Hold time of a valid high level of $\overline{DS1}$ and $\overline{DS2}$ to a valid high level of $\overline{RESET}$.	1,000	—	
t_{DRSU}	Setup time of a valid high level of $\overline{DS1}$ and $\overline{DS2}$ to the rising edge of $\overline{RESET}$.	900	—	
t_{DSPW}	Pulse width high or low of DCD<7:0> and DSR<7:0>.	1,000	—	
t_{IHO}	Hold time provided by Octal ART from a valid IRQLN<2:0> and IRQTxRx to a valid high level of IRQ.	100	—	$C_L = 50$ pF
t_{ISU}	Setup time provided by Octal ART from a valid IRQLN<2:0> and IRQTxRx to a valid low level of $\overline{IRQ}$.	100	—	$C_L = 50$ pF
t_{RES}	Pulse width low of $\overline{RESET}$.	1,000	—	
t_{TXSK}	Pulse width high or low provided by Octal ART on the TxD<7:0> lines. At each baud rate, the actual pulse widths provided vary by t_{TXSK}. This timing parameter should be used to determine cumulative reception/transmission errors.	±250	—	$C_L = 50$ pF

*Refer to Figure 9 for the load circuits used with these measurements.

Figure 13 shows the input and output voltage waveforms for the propagation delay and setup and hold measurements. Figure 14 shows the waveforms for the three-state outputs measurements.

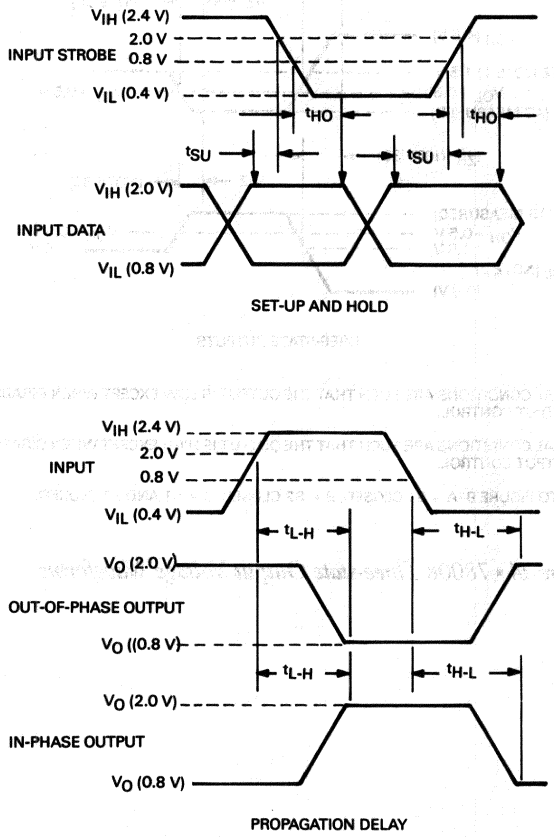
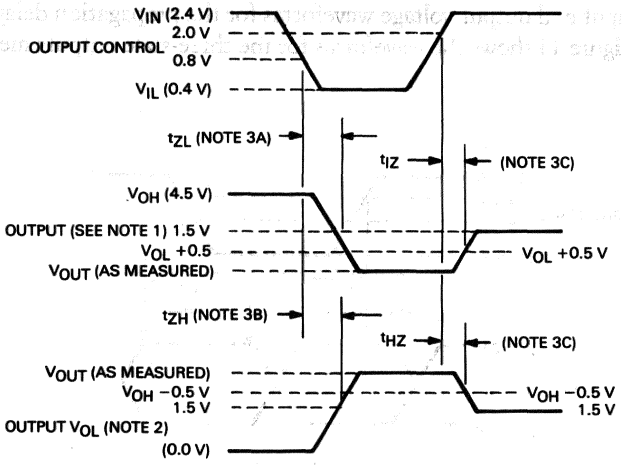


Figure 13 ▪ 78008 Propagation Delay and Setup and Hold Voltage Waveforms



THREE-STATE OUTPUTS

NOTES:

1. INTERNAL CONDITIONS ARE SUCH THAT THE OUTPUT IS LOW EXCEPT WHEN DISABLED BY THE OUTPUT CONTROL.
2. INTERNAL CONDITIONS ARE SUCH THAT THE OUTPUT IS HIGH EXCEPT WHEN DISABLED BY THE OUTPUT CONTROL.
3. REFER TO FIGURE 9. A = S1 CLOSED, B = S2 CLOSED, C = S1 AND S2 CLOSED.

Figure 14 • 78008 Three-state Output Voltage Waveforms

Features

- Hardware compatible with Digital's DL11 series of interfaces
- Asynchronous operation
- Overrun and framing error detection and brake detection
- Compatible with both 8- and 16-bit data paths
- Internal baud rate generation from 300 baud to 38.4k baud
- Four realtime clock interrupt outputs.
- One stop bit only
- Common baud rate for both transmitter and receiver
- Single 5-volt power supply
- Single TTL clock

Desciption

The DC319-AA is a Digital Link (DL11) compatible, asynchronous receiver/transmitter (DLART) designed for data communication between Digital's microprocessors and console terminals or communication devices. The DC319-AA, fabricated using N-channel MOS silicon technology, is contained in a 40-pin dual-inline (DIP) package that can be conveniently installed on a microprocessor module or interface module. Figure 1 is a block diagram of the DC319-AA DLART.

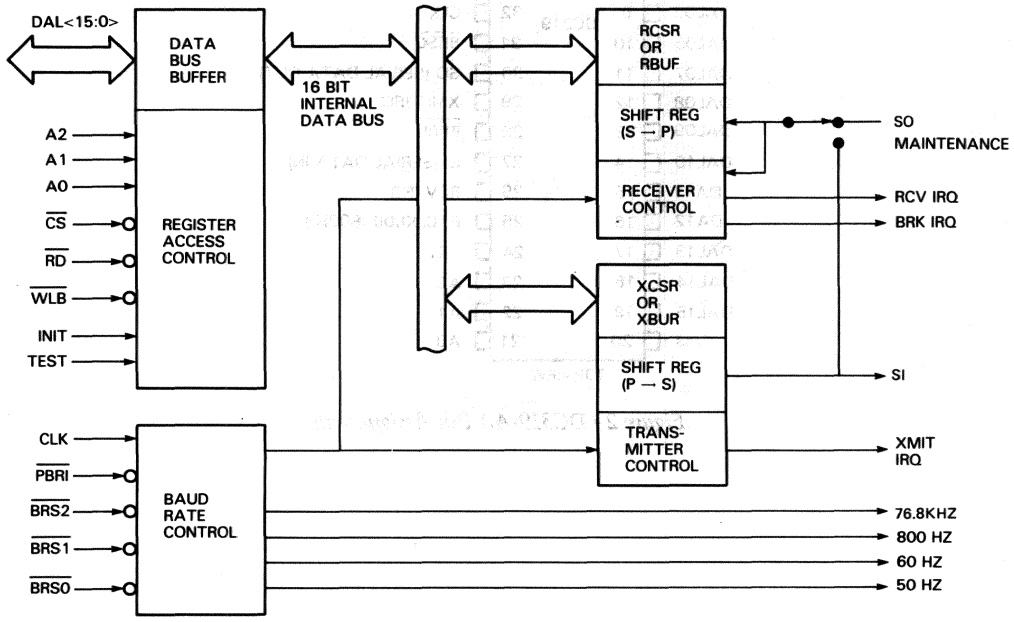


Figure 1 • DC319-AA DLART Block Diagram

The DLART is programmed by the CPU to operate in either 8-bit or 16-bit mode with asynchronous baud rates varying from 300 to 38.4k. The DLART accepts data characters from the CPU in parallel format and converts them into an asynchronous serial-data stream for transmission. It can simultaneously receive serial-data streams and convert them into parallel data characters for the CPU. The DLART notifies the CPU when it is ready to accept new characters for transmission or when it has received a character from the serial line. The DLART contains an internal baud-rate control to reduce support logic required to select baud rates. It also provides four realtime interrupt outputs. The CPU can read the complete status of the DLART at any time including the indication of data transmission errors and the status of control signals. Device address detection, vector generation, and interrupt arbitration must be provided externally. The DLART provides the DL-defined internal registers allowing it to operate with Digital software.

▪ Signal and Pin Descriptions

The input and output signals and the power and ground connections for the DC319-AA 40-pin DIP are shown Figure 2 and defined in Table 1.

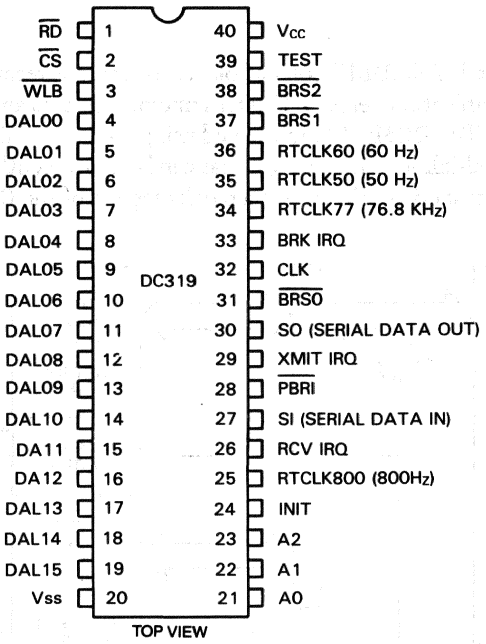


Figure 2 ▪ DC319-AA Pin Assignments

Table 1 • DC319-AA Pin and Signal Summary

Pin	Signal	Input/Output*	Definition/Function
1	$\overline{RD}$	input	Read—When asserted while the $\overline{CS}$ signal is asserted and the $\overline{WLB}$signal is unasserted, the content of the register selected by the A2, A1, and A0 lines is transferred to the DAL.
2	$\overline{CS}$	input	Chip select—When asserted while A0 is unasserted, the contents of the DAL<15:00> lines are transferred to the register selected the A2 and A1 inputs.
3	$\overline{WLB}$	input	Write low byte—When asserted and the A0 input is unasserted, the data on the low byte DAL<07:00> lines is written into the writable bits of the register selected by the A2 and A1 lines.
19-4	DAL<15:00>	input/output	Data address lines <15:00>—Multiplexed bidirectional data lines.
21	AO	input	Register byte select—When asserted, the high byte of the register selected by the A2 and A1 lines is multiplexed to the low byte DAL<07:00> lines.
23 22	A2 A1	inputs	Register address select—These inputs select the internal register that is accessible through the DAL lines when the $\overline{CS}$ line is asserted. **A2 A1 Register** 0 0 RCSR 0 1 RBUF 1 0 XCSR 1 1 XBUF
24	INIT	input	Initialize—This input is used to reset the RCV IE bit in the RCSR register, and the XMIT IE, MAINT, and XMIT BRK bits in the XCSR register.
25	RTCLK800	output	Realtime clock interrupt (800 Hz)—This output provides an 800-Hz, 50% duty cycle signal.
26	RCV IRQ	output	Receiver interrupt request—This interrupt output is asserted when both the RCV DONE and RCV IE bits in the RCSR are set.
27	SI	input	Serial input—This input accepts an asynchronous bit serial data stream. The input signal must remain in the high (marking) state for at least one-half bit time before a high-to-low (mark-to-space) transition is recognized. A mark-to-space transition is required to determine the beginning of a start bit and to initiate data reception.

Pin	Signal	Input/Output*	Definition/Function
28	$\overline{\text{PBRI}}$	input	Programmable baud rate inhibit—This input is optionally held low externally by a jumper to ground or held high internally. Holding this line low disables the software programmable baud rate selection (clears the PBR2-0 and PBRE bits) but makes the DLAR DL-software compatible.
29	XMIT IRQ	output	Transmitter interrupt request—This interrupt request output is asserted only when both the XMIT RDY and XMIT IE bits in the XCSR are set. This output can also be cleared externally by being forced low (clamped to ground) by an open-collector transistor for a minimum of 100 ns after being high for a minimum of 500 ns.
30	SO	output	Serial output—This output provides an asynchronous bit serial-data stream. This line remains high (marking) when no data is being transmitted. This line will remain low when the XMIT BRK bit in the XCSR is set.
32	CLK	input	Clock in—This input requires a 614.4-kHz, 0.1% square wave. All baud rates and clocks are derived from this input.
33	BRK IRQ	output	Break detected interrupt request—This output is asserted when the RCV BRK bit is set and is unasserted by the TEST input or when the RBUF register is read. This output can also be cleared externally by being forced low (clamped to ground) by an open-collector transistor for a minimum of 100 ns after being high for a minimum of 500 ns.
34	RTCLK77	output	Realtime clock interrupt (76.8 kHz)—This output provides a 76.8-kHz, 50% duty cycle signal. After being high for a minimum of 500 ns, this output can be cleared externally by being forced low (clamped to ground) with an open-collector transistor for a minimum of 100 ns.
35	RTCLK50	output	Realtime clock interrupt (50 Hz)—This output provides a 50-Hz, 50% duty cycle signal. After being high for a minimum of 500 ns, this output can be cleared externally by being forced low (clamped to ground) with an open-collector transistor for a minimum of 100 ns.

Pin	Signal	Input/Output*	Definition/Function
36	RTCLK60	output	Realtime clock interrupt (60 Hz)—This output provides a 60-Hz, 50% duty cycle signal. After being high for minimum of 500 ns, this output can be cleared externally by being forced low (clamped to ground) with an open-collector transistor for a minimum of 100 ns.
38,37, 31	BRS<2:0>	input	Baud rate select—These inputs select the receiver and transmitter baud rates when the PBRE bit is cleared as follows: The inputs are optionally asserted low by a jumper to ground or held high internally.

BRS Line 2	1	0	Baud rate
H	H	H	300
H	H	L	600
H	L	H	1,200
H	L	L	2,400
L	H	H	4,800
L	H	L	9,600
L	L	H	19,200
L	L	L	38,400

39	TEST	input	Test—This input is used during module assembly and test to disable all DLART outputs. It is also used in a system during powerup to reset all internal logic.
40	V_{CC}	input	Voltage—Power supply voltage.
20	V_{SS}	input	Ground reference

*Input and output signals are TTL levels.

Read and Write Control Functions

Table 2 lists the control signal levels and transitions required to select the read and write functions of the DC319-AA.

Table 2 • DC319-AA Read and Write Control Functions

Control Signals*				
A0	**$\overline{CS}$**	**$\overline{RD}$**	**$\overline{WLB}$**	**Function**
L	L	L	H	Read—Register bits 15:00 to DAL<15:00>
H	L	L	H	Read—Register bits 15:08 to DAL<07:00>
x	H	L	x	Read—no effect
x	x	L	L	Read—no effect
L	∧	x	L	Write—DAL<15:00> to register bits 15:00
L	L	x	∧	Write-DAL<07:00> to register bits 07:00
x	H	x	∧	Write—no effect
L	x	x	∧	Write—no effect

*x = either high or low
∧ = low-to-high transition

• Register Assignments

The DL11-defined internal registers are described in the following paragraphs and are available to the user to program and monitor the operation of the DC319-AA.

Receiver Control and Status Register

The receiver control and status register (RCSR) controls the operation of the receiver and indicates status. Figure 3 shows the format of register, and the register information is described in Table 3.

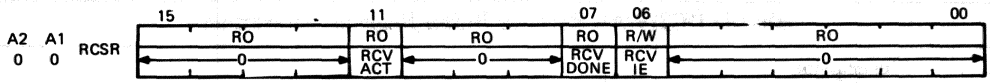


Figure 3 • DC319-AA Receiver Control and Status Register Format

Table 3 • DC319-AA Receiver Control and Status Register Description

Bit	Description
15-12	RAZ (Read as zero)
11	RCV ACT (Receiver active)—A read-only bit set when the receiver is active. This bit is set at the center of the start bit, which is the beginning of the input serial data and cleared one bit at a time before the leading edge of RCV DONE or TEST signal.
10-08	RAZ (Read as zero)
07	RCV DONE (Receiver done)—A read-only bit set when an entire byte has been received and transferred to the RBUF register. This bit is cleared by reading the RBUF register or by the TEST signal.
06	RCV IE (Receiver Interrupt Enable)—A read/write bit set under program control. The RCV IRQ line follows the RCV DONE bit and allows an interrupt request to be made when RCV DONE is set. This bit is cleared by the INIT signal and by the TEST signal.
05-00	RAZ (Read as zero)

Receiver Buffer Register

The receiver buffer (RBUF) register is a read-only register that stores the serial information received from the device and indicates error status. Figure 4 shows the format of the information in the RBUF register and Table 4 contains a description of the register information.

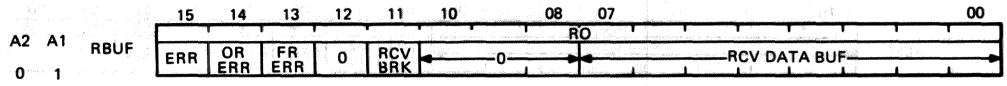


Figure 4 • DC319-AA Receiver Buffer Register Format

Table 4 • DC319-AA Receiver Buffer Register Description

Bit	Description
15	ERR (Error)—A read-only bit set when the overrun or the framing-error bit is set. It is cleared by removing the error-producing condition.
14	OR ERR (Overrun error)—A read-only bit set when a received byte is transferred to the RBUF register before the RCV DONE bit is cleared. An overrun error indicates that reading of the previously received byte was not completed prior to receiving a new byte. This bit is updated when byte is transferred to the RBUF register and is cleared by the TEST signal.
13	FR ERR (Framing error)—A read-only bit set when a received byte without a valid stop bit is transferred to the RBUF register. This bit is cleared by the TEST signal or when a received byte with a valid stop bit is transferred to the RBUF register.
12	RAZ—Read as zero.
11	RCV BRK (Received break)—A read-only bit set when the serial-in (SI) signal goes from a mark to a space and stays in the space condition for 11 bit times after serial reception starts. This bit is cleared when the SI signal returns to the mark condition or by the TEST signal.
10:08	RAZ—Read as zero.
07:00	RCV DATA BUFFER (Received data buffer)—Read-only bits that store the most recent byte received. When a new byte is transferred to the RCV DATA BUFFER, the RCV DONE bit in the RCSR is set. These bits are cleared by the TEST signal.

Transmitter Control and Status Register

The transmitter control and status register (XCSR) controls the operation of the transmitter in the DC319-AA. Figure 5 shows the format of the information in the register and Table 5 contains a description of the register information.

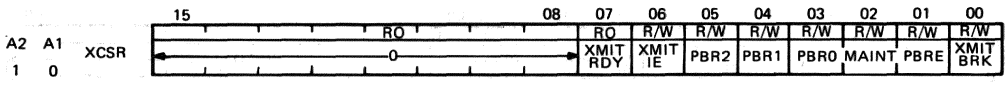


Figure 5 • DC319-AA Transmitter Control and Status Register Format

Table 5 • DC319-AA Transmitter Control and Status Register Description

Bit	Description
15:08	RAZ (Read as zero)
07	XMIT RDY (Transmitter ready)—A read-only bit is set when the XBUF is ready to accept a byte. This bit is cleared by writing to the XBUF and is set by TEST signal.
06	XMIT IE (Transmitter interrupt enable)—A read/write bit set under program control. The XMIT IRQ line follows the XMIT RDY bit and allows an interrupt request to be initiated when the XMIT RDY bit is set. This bit is cleared by the INIT and TEST signal.
05:03	PBR2-PBR0(Programmable baud rate select)—Indicates the transmitter baud rate selected as follows: These bits are cleared by the TEST or $\overline{\text{PBRI}}$ (programmable baud rate inhibit) signal. These bits are read-only as zero when the $\overline{\text{PBRI}}$ input is asserted.

Bit

05	04	03	Baud rate
0	0	0	300
0	0	1	600
0	1	0	1,200
0	1	1	2,400
1	0	0	4,800
1	0	1	9,600
1	1	0	19,200
1	1	1	38,400

02	MAINT (Maintenance)—A read/write bit used to facilitate a maintenance self-test. When this bit is set, the transmitter serial output is connected to the receiver serial input. The external serial input is disconnected. This bit is cleared by the INIT or TEST signal.
01	PBRE (Programmable baud rate enable)—This bit selects the internal and external baud rate. When set, the baud rate is determined by the $\overline{\text{PBR}}$<2:0> bit in the register. When clear, the baud rate is determined by the $\overline{\text{BRS}}$<2:0> inputs. This bit is cleared by the TEST or $\overline{\text{PBRI}}$ (programmable baud rate inhibit) signals. This bit is read-only as zero when the $\overline{\text{PBRI}}$ input is asserted. Otherwise it is read/write.
00	XMIT BRK (Transmit break)—A read/write bit set when the serial output (SO) line is forced to a space condition. This bit is cleared by the INIT and TEST signals.

Transmitter Data Buffer Register

The transmitter data buffer (XBUF) register stores the data in the DC319-AA for serial transfer to the device. Figure 6 shows the format of the information in the register and Table 6 contains a description of the register information.

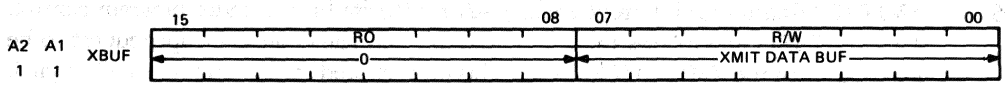


Figure 6 • DC319-AA Transmitter Data Buffer Register Format

Table 6 • DC319-AA Transmitter Buffer Register Description

Bit	Description
15-08	RAZ (Read as zero)
07-00	XMIT DATA BUFFER (Transmitter Data Buffer)—A read/write byte that stores a copy of the most recent byte written into it. When a byte is written into this register, the XMIT RDY bit in the XCSR register is cleared. This byte is copied into the transmitter serial-output register whenever the register is empty and the XMIT RDY bit is clear. The XMIT RDY bit is set when a byte is copied from the XBUF into the serial output register. This register is cleared by the TEST signal.

• Specifications

The mechanical, electrical, and environmental characteristics and specifications for the DC319-AA are described in the following paragraphs. The test conditions used for the electrical values listed are as follows unless specified otherwise. Refer to Digital specification A-PS-2100002-GS for the general specifications for integrated circuits.

- Operating temperature (T_A): 0°C to 70°C
- Power supply voltage (V_{CC}): 5 V ±5%

Mechanical Configuration

The physical dimensions of the DC319-AA 40-pin DIP are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device. These ratings are for stress conditions only and do not imply that the device will function properly at these ratings or ratings above those indicated.

- Power supply voltage (V_{CC}): –0.3 V to 7.0 V
- Ambient temperature under bias: 0°C to 70°C
- Voltage on any pin with respect to ground: –0.5 V to 7.0 V
- Storage temperature: –65°C to 150°C
- Relative humidity: 0 to 95% (noncondensing)
- Power dissipation: 1.0 W

Recommended Operating Conditions

- Power supply voltage (V_{CC}): 5 V ±5%
- Ambient temperature (T_A): 0°C to 70°C

dc Electrical Characteristics

The dc electrical parameters of the DC319-AA for the operating voltage and temperature ranges specified are listed in Table 7. Refer to Appendix C for the test circuit configurations referenced in the table. All input and output signals are TTL levels.

Table 7 • DC319-AA dc Input and Output Characteristics

Parameter	Symbol	Test Conditions	Requirements Min.	Requirements Max.	Units	Test Circuit
Low-level input voltage	V_{IL}		–0.5	0.8	V	C1,C2
High-level input voltage	V_{IH}		2.0	V_{CC}	V	C1,C2
Low-level output voltage	V_{OL}	I_{OL} = 2.2 mA	—	0.4	V	C2
High-level output voltage voltage	V_{OH}	I_{OL} = –400 µA	2.4	—	V	C1
Output float leakage current	I_{OFL}	V_{out} = V_{CC} to 0.4V	—	10	µA	

Parameter	Symbol	Test Conditions	Requirements Min.	Max.	Units	Test Circuit
Input leakage current	I_{IL}	$V_{in} = V_{CC}$ to 0.4 V	—	10	µA	C5
Power supply current	I_{CC}	All outputs = high	—	100	mA	C7
Standard V_{IH} output current	I_{OH}	$V_{out} = V_{CC}$ to 0.4 V	-40	—	µA	C1
DAL V_{IH} output current	I_{OH}	$V_{out} = V_{CC}$ to 0.4 V	-700	—	µA	C1
Standard V_{IL} output current	I_{OL}	$V_{out} = V_{CC}$ to 0.4 V	1.6	—	mA	C2
DAL V_{IL} output current	I_{OL}	$V_{out} = V_{CC}$ to 0.4 V	3.2	—	mA	C2

ac Electrical Characteristics

The switching characteristics of the output signals are listed in Table 8. Refer to the input and output waveforms in Appendix D for the symbols referenced in the table. The signal timing for a read and write data and control cycle is shown in Figure 7. Table 9 defines the timing parameters listed in Figure 7 for the operating temperature and voltage ratings specified.

Table 8 • DC319-AA Signal Switching Characteristics

Symbol	Description	Signal	Requirements (ns) Min.	Max.
t_R/t_F	Rise time/Fall time*	Interrupt request outputs (IRQ)	—	250
		Serial-data outputs (SI)	—	150
		Baud-rate clock (BRCLK)	—	150

*Each DAL line drives 200 pF load. All other outputs drive 1 TTL unit load and 50 pF load. Timing measurements are made at 2.0 V on a low-to-high transition and at 0.8 V at a high-to-low transition.

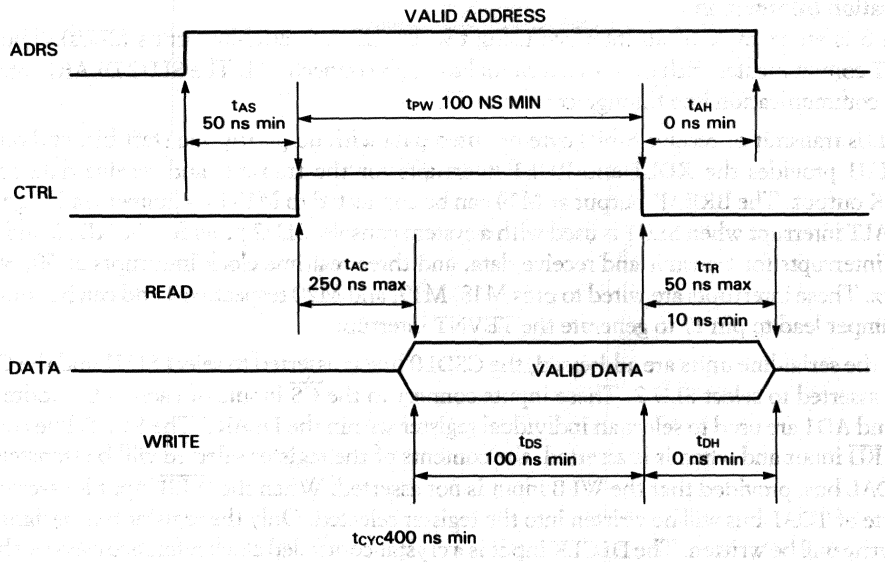


Figure 7 • DC319-AA Read and Write Data and Control Cycle Timing

Table 9 • DC319-AA Read and Write Cycle Timing Parameters

Time	Description[1]	Requirements (ns) Min.	Max.
t_{CYC}	Cycle time	400	—
t_{PW}	Controlling pulse width	100	—
t_{AS}	Address setup time	50	—
t_{AH}	Address hold time	0	—
t_{AC}	Access time	0	250
t_{TR}	Three-state time[2]	10	50
t_{DS}	Data setup time	100	—
t_{DH}	Data hold time	0	—

[1]Read control: $\overline{CS}$ and $\overline{RD}$ signal asserted and $\overline{WLB}$ signal unasserted.
Write control: $\overline{CS}$ and $\overline{WLB}$ signal asserted and A0 unasserted.
[2]t_{TR} (off) is measured with DAL drive = 100 μA.

Application Information

Figure 8 is an example of an interface using two DC319-AA serial-line units (SLUs). The SLU1 DLART communicates with a console terminal through connector J1. The SLU2 DLART interfaces with a communication line through connector J2.

The SLUs transmit or receive 8-bit, byte-oriented data with no parity, one start bit, and one stop bit. SLU1 provides the XDL1 and RDL1 interrupts for the transmit and receive data and the BREAK output. The BREAK output at M14 can be connected to M13 by a jumper lead to generate the HALT interrupt when SLU1 is used with a system console. SLU2 provides the XDL2 and RDL2 signal interrupts for transmit and receive data, and three realtime clock interrupts at 50, 60, and 800 Hz. These interrupts are wired to pins M18, M19, and M20 respectively and can be connected by a jumper lead to pin 17 to generate the TEVNT interrupt.

When the serial-line units are addressed, the CSDL0 line is asserted to select SLU1 and the CSDL1 line is asserted to select SLU 2. These inputs connect to the $\overline{CS}$ inputs of each SLU. Address bits AD2 and AD1 are used to select an individual register within the DLART. The READ line connects to the $\overline{RD}$ input and when it is asserted, the contents of the register selected will be transferred to the TDAL bus, provided that the $\overline{WLB}$ input is not asserted. When the $\overline{WLB}$ input is asserted, the low byte of TDAL bus will be written into the register selected. Only the register bits designated as read/write will be written. The DLCLK input is a crystal-controlled clock reference used by the SLU to generate baud rates and realtime clock outputs. If the BCLR input is asserted during a RESET instruction, the RCV IE bit of the RCSR register and the XMIT IE, MAINT, and XMIT BRK bits of the XCSR register are reset. When the DC LO input is asserted during powerup, all SLU outputs will be disabled and the internal logic and registers will be reset. The baud rate is set at 300 after the SLU is initialized by DC LO signal.

The RS232 and RS423 EIA standard signal levels for the interface connector are provided by E30 and E37 dual-line drivers and dual-line receivers. The slew rate for both channels is controlled by resistor R6. The factory configuration uses a 22-kΩ resistor to provide a 2-μs slew rate for operating at a 38.4k baud rate.

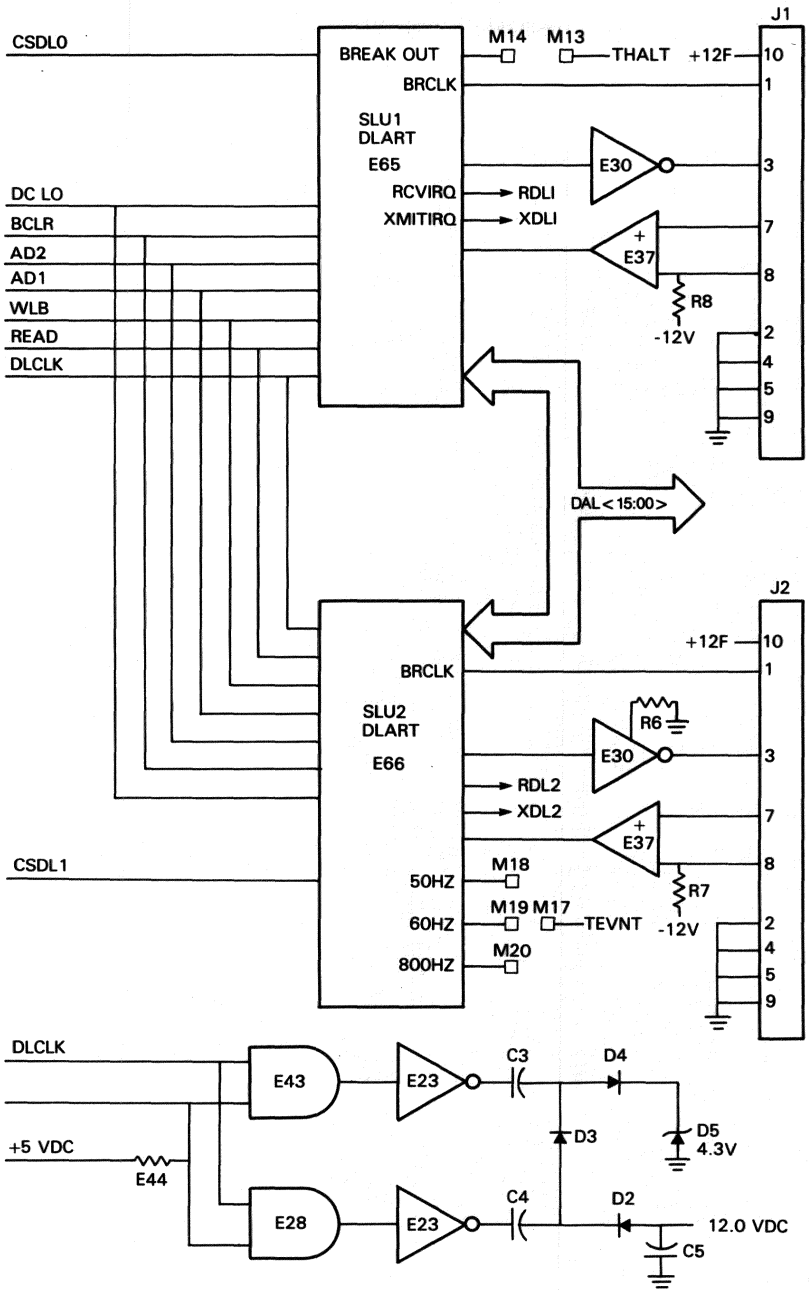


Figure 8 ▪ Dual DC319-AA Serial-line Unit Configuration

Section 4—Bus Support Devices

The bus support devices are available for Q-bus, UNIBUS, and VAXBI bus interface development.

VAXBI Bus Interfaces

The VAX bus interconnect (VAXBI) is low-cost, high-bandwidth, 32-bit synchronous bus used to connect VAX processors to memories, I/O controllers, I/O bus adapters, and other VAX processors. It provides a large addressing range and high data integrity and allows a single VAX processor to communicate with up to 16 nodes on the VAXBI bus.

VAXBI 78732 Bus Interconnect Interface—The BIIC is a 133-pin ZMOS chip that serves as the primary interface between the VAXBI bus and a master or slave port interface.

VAXBI 78743 BCI Adapter Interface—The BCAI is a 133-pin ZMOS chip that functions as a buffer file between user-designed processors, memories, and adapter modules and the VAXBI bus.

VAXBI 78733 BCI to MicroVAX II Bus Interface—The BCI3 is a 132-pin package used to connect the integrated circuit interconnect bus of the MicroVAX processor to the VAXBI bus through the VAXBI 78732 BIIC.

VAXBI 78701 Clock Driver and *VAXBI 78702 Clock Receiver*—The clock driver is a 14-pin bipolar DIP that serves as the clock source for the VAXBI bus system. The output of the clock driver is received by the 16-pin bipolar clock receiver to generate the timing signals used by the VAXBI bus system.

Q-bus Circuits and Kits

Digital provides a series of integrated circuits (ICs) and kits used in the development of device interfaces for the Q-bus. The Q-bus is the LSI-11 processor bus. The ICs are available separately or as part of the DCK11 series of chipkits. The chipkits contain a set of chips, a double-height wire-wrappable module (W9512), and connecting cables that can be used to design and implement most Q-bus interfaces. The chipkits minimize the number of chips necessary to develop custom program control or direct memory access (DMA) interfaces. The wire-wrappable modules provide additional space for mounting the special interface logic required. Refer to *Chipkit Users Manual* (document number EK-01387) for detailed information related to DCK11 series of chipkits and for application information. The specifications for the following Q-bus interface chips are contained in this reference guide.

DC003 Dual-interrupt Logic—The DC003 is an 18-pin DIP bipolar device that is used to perform an interrupt transaction in a computer system that uses a daisychain type of arbitration.

DC004 Register Selector Logic—The DC004 is a 20-pin DIP bipolar device that operates as a register selector to control the transfer of data to and from up to four word registers.

DC005 4-Bit Transceiver—The DC005 is a 20-pin DIP bipolar device that is used in interfaces as a bidirectional buffer between the Q-bus and device data bus.

DC006 Word Count/Bus Address Logic—The DC006 is a 20-pin DIP bipolar device that is used to control direct memory access (DMA) data transfers.

DC010 Direct Memory Access Logic—The DC010 is a 20-pin DIP bipolar device that provides the logic to perform the protocol operations required to request and gain control of the Q-bus.

Chipkit Description—The following LSI-11 chipkits are available and contain the components listed.

- DCK11-AA Program Control Bus Interface Chipkit
  1 DC003 Dual-interrupt Logic
  1 DC004 Register Selector Logic
  4 DC005 4-bit Transceiver Logic

- DCK11-AB Designer's Program Control Bus Interface Chipkit
  1 DC003 Dual-interrupt Logic
  1 DC004 Register Selector Logic
  4 DC005 4-bit Transceiver Logic
  1 W9512 Double-height, wire-wrappable module
  1 BC07-D 10-foot, 40-conductor, plug-in cable

- DCK11-AC DMA Bus Interface Chipkit
  1 DC003 Dual-interrupt Logic
  1 DC004 Register Selector Logic
  4 DC005 4-bit Transceiver Logic
  2 DC006 Word count/Bus Address Logic
  1 DC010 Direct Memory Access Logic

- DCK11-AD Designer's DMA Bus Interface Chipkit
  1 DC003 Dual-interrupt Logic
  1 DC004 Register Selector Logic
  4 DC005 4-bit Transceiver Logic
  2 DC006 Word Count/Bus Address Logic
  1 DC010 Direct Memory Access Logic
  1 W9512 Double-height, wire-wrappable module
  1 BC07-D 10-foot, 40-conductor, plug-in cable

UNIBUS Devices

The UNIBUS is an asynchronous bus used with the PDP-11 and VAX processors. The UNIBUS devices facilitate the development of the bus interfaces.

DC013 UNIBUS Request Logic—The DC013 is a 16-pin DIP device that contains the logic required to perform bus requests and to gain control of the UNIBUS.

DC021 Octal Bus Transceiver—The DC021 is a 20-pin DIP device that contains eight bus transceivers used to transfer information between the UNIBUS and a user-developed interface.

• Features

- Enables low interface cost
- High-level integration reduces module area required
- Data access time in less than 800 nanoseconds
- Peak bandwidth greater than 13 Megabytes per second
- Extensive error detection
- Supports memories, processors, I/O devices, and caches
- Complete VAXBI arbitration, address decoding and matching logic to reduce hardware and software protocol

• Description

The VAXBI 78732 bus interconnect interface chip (BIIC) is contained on a 133-pin ZMOS integrated circuit and serves as the primary interface between the VAXBI ™ bus and the user-developed interface of a node. The BIIC is a general purpose interface for processors, memories, and adapters that operate with the VAXBI bus. It performs bus transactions, address decoding and matching, and controls the user's interface signals. Figure 1 is a functional block diagram of the VAXBI 78732 interface.

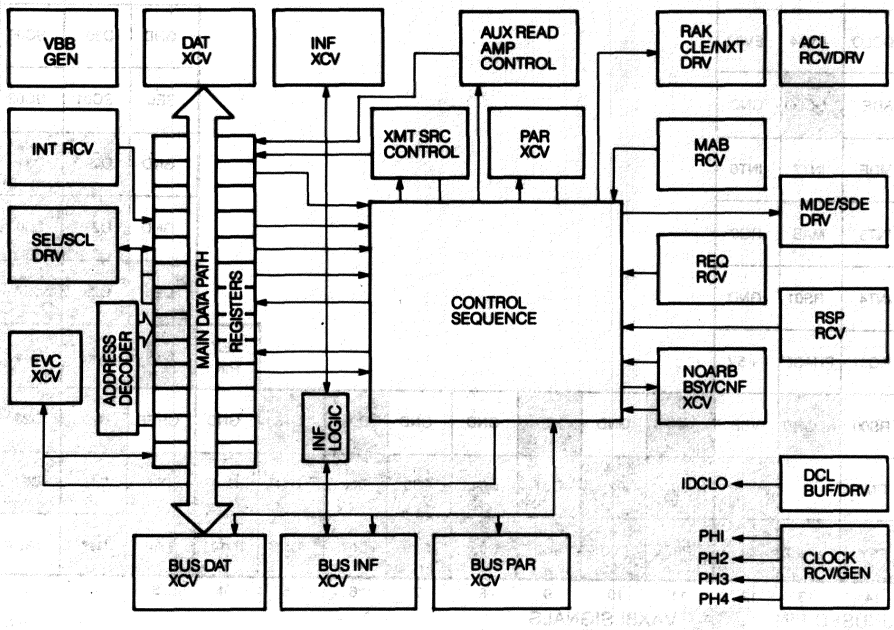


Figure 1 • VAXBI 78732 Bus Interconnect Interface Block Diagram

The VAXBI bus is a 32-bit, general purpose synchronous bus that can effectively be used with single-processor or multiprocessor systems that are based on the VAX processors or other 32-bit processors or compatible devices. The VAXBI bus can have a maximum length of 1.5 meters and can connect with as many as 16 intelligent nodes contained on a maximum of 36 modules. The aggregate throughput rate of the nodes is 13.3 Mbytes per second.

• Pin and Signal Description

The VAXBI 78732 is a 133-pin interface that functions with the input and output signals described in the following paragraphs. Figure 2 shows the connection pins and signals of the VAXBI Bus interface chip. The BIIC interface contains two groups of signals. The BI signals connect to the VAXBI bus and are shown in shaded blocks. The remaining blocks are BCI signal pins that connect to the user's interface and to power and miscellaneous signal lines.

	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
P	ACLO	I02	D00	D02	D03	D06	D07	D10	D11	D14	D16	D19	D20	VBB	P
N	P0	ACLO	I01	I03	D01	D05	D08	D09	D12	D15	D18	D22	*	D25	N
M	NXT	CLE	I00	+5V	GND	D04	GND	GND	D13	D17	D21	D23	D24	D26	M
L	EV01	RAK	GND									+5V	D27	D29	L
K	EV03	EV00	GND									GND	D28	D31	K
J	DCLO	EV04	EV02									GND	D30	SC00	J
H	SDE	DCLO	GND									SEL	SC01	SC02	H
G	MDE	INT7	INT6									GND	D29	D31	G
F	INT5	MAB	RQ0									GND	D28	D30	F
E	INT4	RS01	GND									D24	D26	D27	E
D	RQ01	PHASE	+5V								KEY PIN	D19	D22	D25	D
C	RS00	NOARB	VBB	CNF0	GND	I03	GND	GND	D09	D16	GND	GREF	D20	D23	C
B	TIME	*	CNF1	I00	I02	D01	D04	D05	D08	D11	D13	D15	D18	D21	B
A	BSY	CNF2	I01	P0	D00	D02	D03	D06	D07	D10	D12	D14	D17	VREF	A
	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

*UNUSED PIN VAXBI SIGNALS

MLO-088-85

Figure 2 • VAXBI 78732 Pin Assignments

VAXBI Bus Line Functions

Table 1 lists the signal and functions of the VAXBI bus lines, power connections and ground lines that connect to the BIIC. Each signal line includes a pullup resistor circuit and all BI driver lines are open-collector outputs. The BI bus signals are grouped by the functions shown in Figure 3. All signal lines are synchronous except for the asynchronous control signals.

Table 1 • VAXBI 78732 BI Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
A02-A10 B01-B09 C01,C02 C05,C06 E01-E03 F01,F02 G01,G02	$\overline{\text{BI D<31:00>}}$	input/output[1]	BI Data—Used to transfer data and address information and to perform arbitration.
B10,B11 C09	$\overline{\text{BI I<3:0>}}$	input/output[1]	BI Identification—Used to transfer commands, encoded master identification, read status codes, and write masks.
A11	$\overline{\text{BI P0}}$	input/output[1]	BI Parity odd—Used to transfer odd parity for the $\overline{\text{BI D<31:00>}}$ and $\overline{\text{BI I<3:0>}}$ lines. The line is asserted when an even number of bits on the lines is asserted.
C13	$\overline{\text{BI NO ARB}}$	input/output[1]	BI No arbitration—Asserted to inhibit arbitration on the $\overline{\text{BI D<31:00>}}$ lines. It is also asserted during the BIIC self-test to prevent other nodes from starting transactions until all nodes are ready to participate.
A14	$\overline{\text{BI BSY}}$	input/output[1]	BI Busy—Asserted to indicate that a transaction is in progress.
A13,B12 C11	$\overline{\text{BI CNF<2:0>}}$	input/output[1]	BI Confirmation—Used to transfer responses during command and data cycles.
P14	$\overline{\text{BI AC LO}}$	input[2]	BI ac low—Used with the $\overline{\text{BI DC LO}}$ line to perform power sequences.
H13	$\overline{\text{BI DC LO}}$	input[2]	BI dc low—Used with the $\overline{\text{BI DC LO}}$ line to perform power sequences.

[1]Open drain [2]Open collector [3]Differential ECL

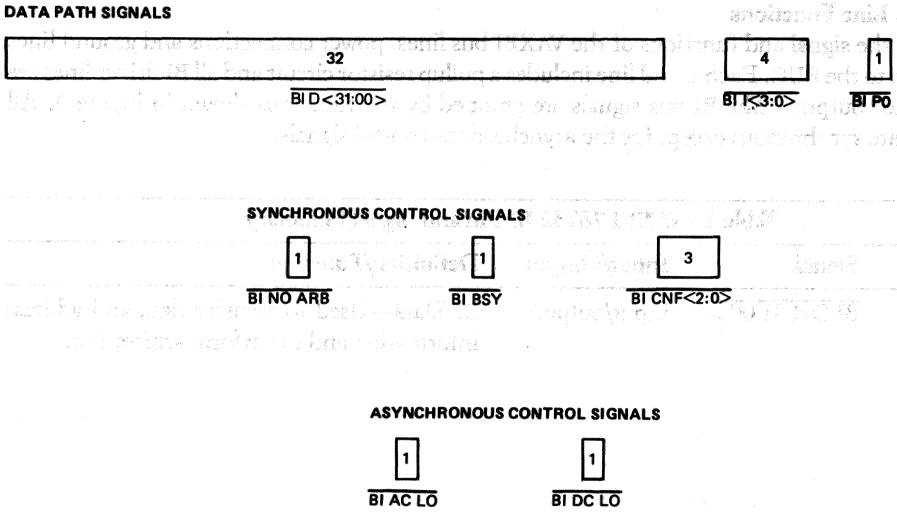


Figure 3 ▪ VAXBI 78732 BI Line Functions

BI Data-path Signals

The BI data-path signals are grouped into the following categories. All data, arbitration, commands, and address information is transferred through these lines.

BI Data and Address ($\overline{\text{BI D<31:00>}}$)—These lines transfer data and address information and are used during the arbitration sequence.

BI Identification ($\overline{\text{BI I<3:0>}}$)—These lines transfer commands, encoded master identification, read status codes, and write masks. Commands can be directed to one or more nodes depending on the type of command. The command codes and types are listed in Table 2.

Table 2 ▪ VAXBI 78732 BI Command Code Assignments

BI I Line 3	2	1	0	Type*	Command/Description
H	H	H	H	—	Reserved
H	H	H	L	SR	READ
H	H	L	L	SR	IRCI/Interlock read with cache intent
H	H	L	H	SR	RCI/Read with cache intent
H	L	H	H	SR	WRITE
H	L	H	L	SR	WCI/Write with cache intent
H	L	L	H	SR	UWMCI/Unlock write mask with cache intent
H	L	L	L	SR	WMCI/Write mask with cache intent

BI I Line 3	2	1	0	Type*	Command/Description
L	H	H	H	MR	INTR/Interrupt
L	H	H	L	SR	IDENT/Identify
L	H	L	H	—	Reserved
L	H	L	L	—	Reserved
L	L	H	H	MR	Stop
L	L	H	L	MR	INVAL/Invalidate
L	L	L	H	MR	BDCST/Broadcast(reserved)
L	L	L	L	MR	IPINTR/Interprocessor interrupt

*SR is a single responder, MR is more than one responder

BI Parity ($\overline{\text{BI P0}}$)—Indicates the parity of the $\overline{\text{BI D<31:00>}}$ and $\overline{\text{BI I<3:0>}}$ lines. (Refer to Parity Checking and Generation.)

BI Synchronous Control Signals

The BI synchronous control lines provide control functions and responses to data and command cycles.

BI No Arbitratration ($\overline{\text{BI NO ARB}}$)—This signal controls the access to the VAXBI data lines for arbitration. The nodes monitor this signal so that the data and command/address information does not contend with the arbitration information. If the $\overline{\text{BI NO ARB}}$ signal is asserted during a VAXBI cycle, the nodes are prevented from arbitrating during the next VAXBI cycle. This signal is asserted by

- Nodes arbitrating for the bus during the arbitration cycle.
- The pending bus master from the cycle after it wins the arbitration until it becomes bus master.
- The bus master during the following:
  —Embedded ARB cycle of longword transaction
  —Embedded ARB cycle and the following cycle of a quadword transaction
  —Embedded ARB cycle through the cycle after the second ACK data cycle of an octaword transaction
- The slave for all data cycles except for the last cycle.
- All potential slaves for the third cycle (decoded master ID) and for the IDENT arbitration of the IDENT command.
- Nodes performing loopback transactions.
- The bus master during its command/address cycle to prevent bus arbitration from occurring. This allows the bus master to start a bus transaction following the current bus transaction. (This mode is reserved for use by Digital.)
- Nodes performing self-test operations until the VAXBI registers can be accessed.

BI Busy ($\overline{\text{BI BSY}}$)—This signal provides the orderly transition of bus mastership from one node to another node. The nodes monitor this signal to determine the action to be performed during the following cycle. The deasserted state of the $\overline{\text{BI BSY}}$ signal indicates that the current transaction has been completed. The node that won the last arbitration may become bus master in the cycle following the cycle in which the deasserted state of the signal was detected. The signal is asserted by

- The bus master during the following cycles of its transaction
  - —Command/address and embedded ARB cycle of a longword transaction
  - —Command/address, embedded ARB, and the following cycle of a quadword transaction
  - —Command/address, embedded ARB through the cycle following the second ACK data cycle of an octaword transaction
- A node to delay the start of the next bus transaction until it can respond to another bus transaction. The maximum stall time by a node should not exceed 16 consecutive cycles and is limited by a timeout circuit to 127 consecutive cycles.
- The slaves for all except the last data cycle.
- Nodes performing loopback transactions.

BI Confirmation ($\overline{\text{BI CNF}}$<2:0>)—These lines provide handshake functions between master and slave nodes to reflect detected errors and to indicate the current state of the slave. Table 3 lists the response codes and information. During a transaction, the node must first respond to the command. For read-type, write-type, and IDENT commands, the slave must respond during each data cycle following the command confirmation cycle.

Table 3 • VAXBI 78732 BI Confirmation Line Code Assignments

BI CNF Line 2	1	0	Description
H	H	H	no acknowledge
H	H	L	illegal
H	L	H	illegal
H	L	L	acknowledge
L	H	H	illegal
L	H	L	stall
L	L	H	retry
L	L	L	illegal

Power Control Signals

The $\overline{\text{BI AC LO}}$ and $\overline{\text{BI DC LO}}$ signals are used to control the powerup and powerdown sequences of the nodes on the bus. These signals enable the VAXBI to store and retrieve the parameters required in the event of a power failure or interruption. The $\overline{\text{BI AC LO}}$ is asserted when the line voltage is below the minimum level and the $\overline{\text{BI DC LO}}$ signal is asserted to indicate an impending loss of the dc power and is also used for initialization during the powerup sequence.

▪ BCI Line Functions

The BCI bus consists of 64 lines used to transfer data, address, status, and control information between the BIIC and the user's interface. Table 4 lists the pins and functions of the BCI lines. The BCI lines are grouped by the functions shown in Figure 4.

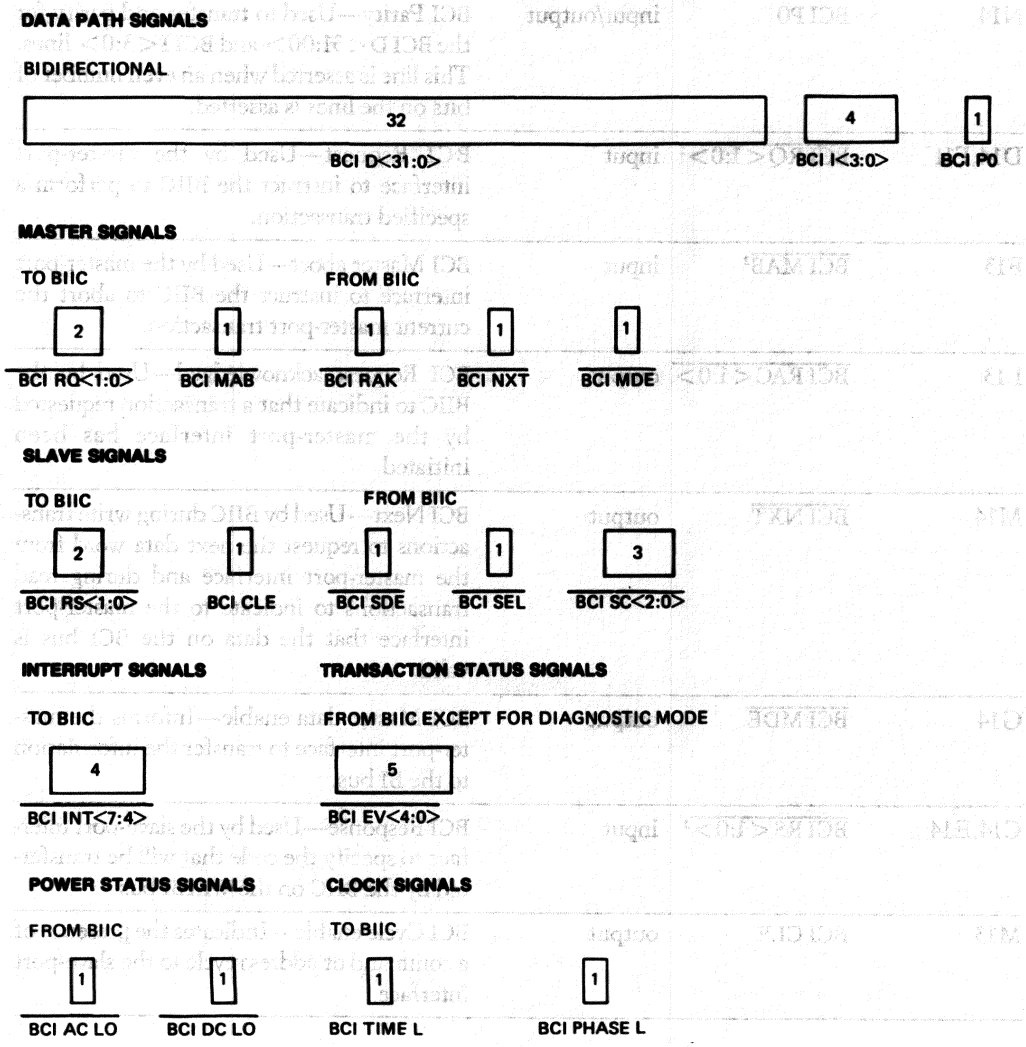


Figure 4 ▪ VAXBI 78732 BCI Line Functions

Table 4 • VAXBI 78732 BCI Pin and Signal Summary

Pin	Signal	Input/Output[1]	Definition/Function
J02 K01,K02 L01,L02 M01-M06 N01,N03-N10 P03-P12	BCI D<31:00>	input/output	BCI Data—Used to transfer data and address information.
M12 N11,N12 P13	BCI I<3:0>	input/output	BCI Information—Used to transfer commands, read status codes, and write masks.
N14	BCI P0	input/output	BCI Parity—Used to transfer odd parity for the BCI D<31:00> and BCI I<3:0> lines. This line is asserted when an even number of bits on the lines is asserted.
D14,F11	$\overline{\text{BCI RQ<1:0>}}$[2]	input	BCI Request—Used by the master-port interface to instruct the BIIC to perform a specified transaction.
F13	$\overline{\text{BCI MAB}}$[2]	input	BCI Master abort—Used by the master-port interface to instruct the BIIC to abort the current master-port transaction.
L13	$\overline{\text{BCI RAC<1:0>}}$	output	BCI Request acknowledged—Used by the BIIC to indicate that a transaction requested by the master-port interface has been initiated.
M14	$\overline{\text{BCI NXT}}$	output	BCI Next—Used by BIIC during write transactions to request the next data word from the master-port interface and during read transactions to indicate to the master-port interface that the data on the BCI bus is valid.
G14	$\overline{\text{BCI MDE}}$	output	BCI Master data enable—Informs the master-port interface to transfer the information to the BI bus.
C14,E14	$\overline{\text{BCI RS<1:0>}}$[2]	input	BCI Response—Used by the slave-port interface to specify the code that will be transferred by the BIIC on the VAXBI bus.
M13	BCI CLE	output	BCI Cycle enable—Indicates the presence of a command or address cycle to the slave-port interface.

Pin	Signal	Input/Output[1]	Definition/Function
H14	$\overline{\text{BCI SDE}}$	output	BCI Slave data enable—Indicates to the slave-port interface that the information to be transferred on the VAXBI bus should be transferred to the BCI data lines.
H03	$\overline{\text{BCI SEL}}$	output	BCI Select—When asserted, the $\overline{\text{BCI SC<2:0>}}$ lines select a slave-port interface for a VAXBI transaction.
H01,H02, J01	$\overline{\text{BCI SC<2:0>}}$	output	BCI Slave control—Selects a slave-port interface when the $\overline{\text{BCI SEL}}$ line is asserted.
E14,F14 G12,G13	$\overline{\text{BCI INT<7:4>}}$[2]	input	BCI Interrupt—Interrupt request lines from the user's interface. Also used with the $\overline{\text{BCI RS<2:0>}}$ lines to select functions during the diagnostic mode.
J12,J13 K13,K14 L14	$\overline{\text{BCI EV<4:0>}}$	output	BCI Event—Indicates the occurrence of significant events within the BIIC or on the VAXBI bus. Not used during the diagnostic mode.
N13	$\overline{\text{BCI AC LO}}$	output	BCI ac low—Indicates the status of the ac power to the user's interface.
J14	$\overline{\text{BCI DC LO}}$	output	BCI dc low—Indicates the status of the dc power to the user's interface.
B14	BCI TIME L	input	BCI Time—A 20-MHz TTL signal from the VAXBI clock receiver in the user's interface. Used with the BCI PHASE L line by the BIIC and user's interface to generate the required timing signals.
D13	BCI PHASE L	input	BCI Phase—A 5-MHz TTL signal from the VAXBI clock receiver in the user's interface. Used with the BCI TIME L line by the BIIC and user's interface to generate the required timing signals.

[1]All user interface signals are TTL levels
[2]These signals must be connected to a high level when not used in the node design.

BCI Data Path Signals

The BCI data path consists of bidirectional three-state data, information, and parity lines. The direction of transfer is controlled by the $\overline{\text{BCI MDE}}$ and $\overline{\text{BCI SDE}}$ signals.

BCI Data and Address (BCI D<31:00>)—These lines provide a 32-bit path between the BIIC and user's interface to transfer data and address information. During command/address cycles, the BCI D<31:30> lines contain the data length code and lines BCI D<29:00> provide the address. Table 5 lists the data length code assignments.

Table 5 • VAXBI 78732 BCI Data Length Code Assignments

BCI D Lines 31	30	Data Length
L	L	reserved
L	H	longword
H	L	quadword
H	H	octaword

BCI Information (BCI I<3:0>)—These lines are used to transfer commands, encoded master ID, read status, and write masks. Table 6 lists the BCI command code assignments, Table 7 lists the read-status code assignments, and Table 8 lists the write-mask code assignments.

Table 6 • VAXBI 78732 BCI Command Code Assignments

BCI I Lines 3	2	1	0	Type[1]	Command	Description
L	L	L	L	—	—	reserved
L	L	L	H	SR	READ	read
L	L	H	L	SR	IRCI	interlocked read with cache intent
L	L	H	H	SR	RCI	read with cache intent
L	H	L	L	SR	WRITE	write
L	H	L	H	SR	WCI	write with cache intent
L	H	H	L	SR	UWMCI	unlock write mask with cache intent
L	H	H	H	SR	WMCI	write mask with cache intent
H	L	L	L	MR	INTR	interrupt
H	L	L	H	SR	IDENT	identify
H	L	H	L	—	—	reserved
H	L	H	H	—	—	reserved
H	H	L	L	MR	STOP	stop
H	H	L	H	MR	INVAL	invalidate
H	H	H	H	MR	BDCST	broadcast[2]
H	H	H	H	MR	IPINTR	interprocessor interrupt

[1]SR = single responder, MR = multiresponder.
[2]Refer to Broadcast Transactions (Appendix A) of the *VAXBI System Reference Manual.*

Table 7 • VAXBI 78732 BCI Read Status Code Assignments

BCI I Lines 3	2	1	0	Description
L	*	L	L	reserved
L	*	L	H	read data
L	*	H	L	corrected read data
L	*	H	H	read data substitute
H	*	L	L	reserved
H	*	L	H	read data/do not cache
H	*	H	L	corrected read data/do not cache
H	*	H	H	read data substitute/do not cache

*Line <2> is reserved. The slave must deassert (L) this line for all status types and the masters must ignore the state of this line.

Table 8 • VAX 78732 BCI Write Mask Code Assignments

BCI I Line asserted	BCI D Line Write Byte
3	<21:24>
2	<23:16>
1	<15:08>
0	<07:00>

BCI Parity (BCI P0)—This line contains the parity indicator for the BCI I<3:0> and BCI D<31:00> line signals. The parity sense is odd. Therefore the BCI P0 line should be asserted during user's interface-generated parity mode when an even number of BCI D<31:00> and BCI I<3:0> lines are asserted.

The state of the BCI P0 line is controlled by the user's interface during system powerup. During powerup when the $\overline{\text{BCI DC LO}}$ signal is asserted, the BIIC loads the bus error register with the parity mode. If the user's interface asserts the BCI P0 line or allows it to becomes a high level by default when the $\overline{\text{BCI DC LO}}$ line is asserted, the BIIC will be configured to generate the parity when the BCI data is to be transmitted on the VAXBI bus. This is defined as the BIIC-generated parity mode and a longer setup time is required for the BCI D<31:00> and BCI I<3:0> lines to allow the BIIC to generate the parity. If the user's interface does not control the BCI P0 line during powerup, the BIIC will generate parity by default. In BIIC-generated parity mode, the $\overline{\text{BCI P0}}$ line does not have to connect to the BIIC.

If the user's interface deasserted the BCI P0 line when the $\overline{\text{BCI DC LO}}$ line is asserted, the BIIC will be configured for user's interface-generated parity. The user's interface must provide the proper parity on the BCI P0 line when the BIIC solicits data. ($\overline{\text{BCI MDE}}$ or $\overline{\text{BCI SDE}}$ is asserted.) In this mode, a shorter setup time is permitted on the BCI D<13:00> and BCI I<3:0> lines for the data to be transmitted on the VAXBI bus. If the user's interface generates incorrect parity, the parity will be transmitted on the bus, and a bus error will result.

When data received from the VAXBI bus is transferred through the BIIC to the BCI lines, the BCI P0 line reflects the received state of the BI P0 line. During loopback transaction cycles, the parity generated by the user's interface or by the BIIC is transferred to the BCI.

BCI Master Signals

The BCI master signals are used to request, execute, and terminate transactions.

BCI Request ($\overline{\text{BCI RQ}}$<1:0>)—These lines are used by the master-port interface to request that the BIIC perform a transaction. The lines are also used to select the BIIC diagnostic mode. Table 9 lists the BCI request codes assignments.

Table 9 • VAXBI 78732 BCI Request Code Assignments

BCI RQ Line 1	BCI RQ Line 0	Description
H	H	no request
H	L	VAXBI transaction request
L	H	loopback request
L	L	diagnostic mode

BCI Master Abort ($\overline{\text{BCI MAB}}$)—This signal is asserted by the master-port interface to indicate to the BIIC that the present master-port transaction from this node is to be aborted. It is also used to clear the retry state of the BIIC entered after a retry confirmation is received. The assertion of the $\overline{\text{BCI MAB}}$ signal does not affect BIIC-generated transactions.

The user's interface usually cannot abort a requested transaction without generating bus errors that will be detected by other nodes. The $\overline{\text{BCI MAB}}$ line therefore should be used to abort a transaction request only following an error condition. A pipeline-request master should abort a transaction request only for the transaction subsequent to one that fails.

The user's interface should deassert the request lines in the same cycle that $\overline{\text{BCI MAB}}$ is asserted. The actions performed by the BIIC depend on when the $\overline{\text{BCI MAB}}$ signal is asserted as follows:

- If the $\overline{\text{BCI MAB}}$ signal is asserted before the BIIC arbitration cycle, no transaction is initiated.
- If the BIIC has won the arbitration cycle and has not transmitted the command/address, it will deassert the $\overline{\text{BI NO ARB}}$ line in the next cycle.
- If the $\overline{\text{BCI MAB}}$ signal is asserted after a retry confirmation or timeout event code is received (RCR or RTO), the BIIC aborts its retry state so that it can accept a new request. The user's interface can abort the retry state only by asserting the $\overline{\text{BCI MAB}}$ signal. Following the assertion of $\overline{\text{BCI MAB}}$, the user's interface should not assert a new request for a minimum of three cycles.
- In a node that allows pipeline requests, the assertion of the $\overline{\text{BCI MAB}}$ signal may occur during a transaction from this node. If the $\overline{\text{BCI MAB}}$ signal is asserted during a transaction, the BIIC aborts the transaction. The user's interface may not receive an event code for a transaction that is aborted. A minimum of three cycles following the assertion of $\overline{\text{BCI MAB}}$ must be allowed by the user's interface before a new request is generated. The use of this mode should be avoided.

BCI Request Acknowledge ($\overline{\text{BCI RAK}}$)—This line is used by the BIIC to indicate that the transaction requested by the master-port interface has been initiated. The $\overline{\text{BCI RAK}}$ line is asserted only during master port transactions and is not asserted during BIIC-generated interrupt and interprocessor interrupt transactions. The line is asserted during the first cycle of a VAXBI or loopback transaction and remains asserted for the duration of the transaction. It is deasserted during the cycle when the event code is transferred from the master. During write-type and broadcast transactions, the $\overline{\text{BCI RAK}}$ line remains asserted until the final acknowledge cycle of the slave is on the VAXBI bus. This occurs three cycles after the last data cycle during write-type and broadcast transactions. During read-type and identification transactions, the line remains asserted until two cycles after the last data cycle to allow time for an internal parity check. The $\overline{\text{BCI RAK}}$ line is deasserted during the sixth cycle for a stop, invalidate, and master-port interprocessor transactions.

If the user's interface reasserts the $\overline{\text{BCI RQ}}$<1:0> line before the deassertion of the $\overline{\text{BCI RAK}}$ as during a pipeline request, the normal timing of the $\overline{\text{BCI RAK}}$ signal is altered and the $\overline{\text{BCI RAK}}$ signal is deasserted in the cycle after the $\overline{\text{BCI RQ}}$<1:0> was reasserted. It will then be reasserted at the start of the newly requested transaction of master-port interface.

BCI Next ($\overline{\text{BCI NXT}}$)—This line is asserted by the BIIC to request the next data word from the master-port interface during VAXBI write-type commands and to indicate to the master-port interface that the data on the BCI is valid during a VAXBI read-type commands. During read-type data cycles in which the master is receiving data, the $\overline{\text{BCI NXT}}$ line is not asserted when a retry, stall, or no acknowledge confirmation code is received with the data, or when incorrect parity of the received data occurs.

During write-type transactions, the asserting edge of the $\overline{\text{BCI NXT}}$ signal is used to request the next write-data longword. The new data to be written must be properly set up on the BCI D<31:00> lines before the beginning of the next cycle. Because the BIIC contains buffer storage, the $\overline{\text{BCI NXT}}$ line is asserted only once for each data word, even if the transaction is retried or stalled for one or more times.

During read-type transactions, the asserting edge of the $\overline{\text{BCI NXT}}$ signal indicates that the data on the BCI D<31:00> lines is valid. This may be used to clock the read data into the user's interface.

BCI Master Data Enable ($\overline{\text{BCI MDE}}$)—This signal is asserted by the BIIC during each cycle to indicate to the master-port interface that the command/address information or data should be transferred to the BCI D<31:00>, BCI I<3:0>, and BCI P0 lines.

Because of the buffer storage in the BIIC, the $\overline{\text{BCI MDE}}$ signal is asserted only once to acquire the command/address information and once to transfer the first data longword. This capability simplifies the design of nodes that perform master-port transactions of longwords only. For transactions greater than one longword, the BIIC may assert the $\overline{\text{BCI MDE}}$ many times for the same data longword. The $\overline{\text{BCI NXT}}$ line however is asserted only once for each data longword.

BCI Slave Signals

The BCI slave signals respond to transactions directed to a node.

BCI Response ($\overline{\text{BCI RS}}$<1:0>)—These lines are used by the slave-port interface to select the code on the $\overline{\text{BI CNF}}$<2:0> lines in response to command and data cycles. They are also used for diagnostic mode function selection. The slave-port interface must respond with the appropriate codes whenever a transaction that involves the slave-port interface occurs. Transactions to BIIC registers do not involve the slave-port interface. A response is required for each cycle in which the slave node transfers information to the $\overline{\text{BI CNF}}$<2:0> lines. Table 10 lists the slave response codes.

Table 10 • VAXBI 78732 BCI Slave Response Code Assignments

BCI RS Lines 1	0	Response	BI CNF Code Result
H	H	no acknowledge (NO ACK)	no acknowledge (NO ACK)
H	L	acknowledge (ACK)	ACK or NO ACK
L	H	stall	stall, ACK, or NO ACK
L	L	retry	retry or NO ACK

Correlation does not always exist between the response code and the corresponding confirmation code transferred on the VAXBI bus. When a code is not involved in a transaction, the BIIC responds with the NO ACK confirmation code regardless of the code on the $\overline{\text{BCI RS}}$<1:0> lines.

Some confirmation codes are illegal during certain types of cycles. During multiresponder transactions, a retry is an illegal response. If the slave-port interface sends an illegal code during a cycle, the BIIC will reply with a NO ACK code on the $\overline{\text{BI CNF}}$<2:0> lines except when a stall response occurs to a multiresponder command.

BCI Command Latch Enable (BCI CLE)—This line is monitored by the user's interface to detect the presence of a command/address cycle. The deasserting edge of the signal can be used to latch the received command/address information from the BCI D<31:00> and BCI I<3:0> lines. The BIIC asserts BCI CLE signal during the cycle after the BIIC recognizes that the $\overline{\text{BI NO ARB}}$ signal is asserted and the $\overline{\text{BI BSY}}$ signal is deasserted. This no arbitration/busy state corresponds to an arbitration cycle or the last data cycle of a transaction that has a pending master. The BIIC deasserts the BCI CLE signal during the cycle after the command/address cycle when the command/address information is on the BCI D<31:00> and BCI I<3:0> lines. The command/address cycle is detected at the transition of the $\overline{\text{BI BSY}}$ signal from the deasserted to the asserted state, when the $\overline{\text{BI NO ARB}}$ signal was asserted during the previous cycle.

The BCI CLE signal may be asserted for more than one cycle following a powerup sequence when different BIIC nodes complete the self-test operation at different times. This can occur during the burst mode and following a pending abort by a master. During loopback transactions, the BIIC sequences the state of the BCI CLE line without regard to the state of the $\overline{\text{BI BSY}}$ and $\overline{\text{BI NO ARB}}$ signals, however; the timing relative to BCI signals is the same as for a transaction on the VAXBI bus.

The state of the BCI CLE signal does not depend on the receipt of valid parity or whether the node is selected as a slave. A silo or cache-resident node that monitors the bus and examines all VAXBI transactions can use the deassertion of the BCI CLE signal to latch the information during any command/address cycle on the VAXBI bus.

The assertion of the BCI CLE line should be used to force the slave-port interface to a state from which it can respond to an SC code during the following cycle. In addition, the slave-port interface should ensure that the stall response code is removed during the cycle when the BCI CLE signal is asserted.

BCI Slave Data Enable ($\overline{\text{BCI SDE}}$)—This line is used by the BIIC to indicate to the slave-port interface that the data to be transmitted on the VAXBI bus should be transferred to the BCI D<31:00>, BCI I<3:0>, and BCI P0 lines.

BCI Select ($\overline{\text{BCI SEL}}$)—This line is asserted by the BIIC to inform the slave-port interface that it has been selected by a VAXBI transaction. The $\overline{\text{BCI SEL}}$ line is asserted during the embedded arbitration cycle of a transaction if the slave was selected by the command/address information from the previous cycle. The assertion of the $\overline{\text{BCI SEL}}$ signal is dependent on the receipt of valid parity during the command/address cycle.

The assertion of the $\overline{\text{BCI SEL}}$ signal is accompanied by the select code on the $\overline{\text{BCI SC<2:0>}}$ lines. The BCI control and status register (BCICSR) allows the user's interface to create a customized subset of VAXBI transactions that will select the slave-port interface. Nodes that are not required to respond to multicast space read-type and write-type commands can clear the MSEN bit in the BCICSR and the external decoding of the multicast space (SC) code is not required. (Refer the BIIC register descriptions.)

The $\overline{\text{BCI SEL}}$ line is asserted when the following commands have been received:

- A read- or write-type command whose address falls within the range defined by the starting and ending address registers of the BIIC.
- A read- or write-type command whose address falls within multicast space and the MSEN bit is set in the BCICSR.
- A read- or write-type command that matches the user's interface CSR space of the node and the UCSREN bit is set in the BCICSR.
- A read- or write-type command that matches the BIIC CSR space of the node and the BICSREN bit is set in the BCICSR.
- An IDENT command and the IDENTEN bit is set in the BCICSR.
- A BDCST command directed at the node and the BDCSTEN bit is set in the BCICSR.
- A STOP command directed at the node and the STOPEN bit is set in the BCICSR.
- A RESERVED command and the RESEN bit is set in the BCICSR.
- An IPINTR command directed at the node that matches the IPINTR mask register when the IPINTREN bit is set in the BCICSR.
- An INTR command directed at the node when the INTREN bit is set in the BCICSR.
- An INVAL command or a write-type command that is not directed to the range of addresses defined by the starting and ending address registers when the INVALEN or the WINVALEN bit is set in the BCICSR.

BCI Select Code ($\overline{\text{BCI SC<2:0>}}$)—These lines transfer detailed selection information from the BIIC to the slave-port interface. The presence of select code is indicated by the assertion of the $\overline{\text{BCI SEL}}$ line. The assertion of the select code depends on the receipt of valid parity for the command/address information. If the user's interface fully decodes the $\overline{\text{BCI SC<2:0>}}$ lines, the use of $\overline{\text{BCI SEL}}$ line is not required. The BCI select codes are listed in Table 11.

Table 11 • VAXBI 78732 Select Code Assignments

BCI SC Line 2	1	0	Description
H	H	H	A default state to indicate that a selection has not occurred.
H	H	L	A read- or write-type command directed to the control/status register of the user's interface has been received and the UCSREN bit in the BCI control and status register (BCICSR) is set. A read- or write-type command directed to the control/status register of the BIIC has been received and the BICSREN bit in the BCICSR is set.
H	L	H	A read- or write-type command directed to the range of address space defined by the starting and ending address registers of the BIIC has been received and the MSEN bit in the BCICSR is set.
H	L	L	A read- or write-type command directed to the range of addresses defined as multicast space has been received and the MSEN bit in the BCICSR is set.
L	H	H	An identification transaction has been received and the IDENTEN bit in the BCICSR is set.
L	H	L	An interrupt transaction with a destination that matches the node has been received and the INTREN bit in the BCISCR is set or an interrupt transaction with a destination that matches the node and a source that matches the IPINTR mask register has been received and the IPINTREN bit in the BCISCR is set.
L	L	H	An invalid or write-type command that is not directed to the range of addresses defined by the starting and ending address registers of the BIIC has been received, the BCI D29 bit is not asserted indicating that the address is not within I/O space and the INVALEN bit or the WINVALEN bit in the BCISCR is set
L	L	L	A BDCST, STOP, or RESERVED command with a destination that matches the node (except for the RESERVED command) has been received and the BDCSTEN, STOPEN, or RESEN enable bit in the BCICSR is set.

BCI Interrupt Request Signals

The interrupt request lines are used to interrupt current processor operation and cause the processor to branch to a routine to service the interrupt.

BCI Interrupt Request ($\overline{\text{BCI INT}}$<7:4>)—These lines are used by the user's interface to request that interrupts be performed by the BIIC. The $\overline{\text{BCI INT}}$<7:4> signals must be synchronously asserted. Each INT line signal causes an interrupt at the level corresponding to its bit position. The $\overline{\text{BCI INT}}$<7> initiates level 7 interrupts, $\overline{\text{BCI INT}}$<6> initiates level 6 interrupts, etc. Interrupts can also be generated by writing to the interrupt control register of the user's interface to set one or more of the force bits

The BCI INT lines are "pseudo-edge" triggered. The BIIC samples the state of the lines during each T150/0. The BIIC determines when a transition has occurred on these lines by comparing the received state of the line from the previous cycle with the received state of the current cycle. The transition from the deasserted state to the asserted state specifies an interrupt request.

The $\overline{\text{BCI INT}}$<7:4> lines are also used with the $\overline{\text{BCI RS}}$<1:0> lines for diagnostic mode function selection.

BCI Transaction Status Signals

The transaction status lines provide status information related to the VAXBI interface, the VAXBI bus, or diagnostic program.

BCI Event Code ($\overline{\text{BCI EV}}$<4:0>)—These lines indicate the occurrence of significant events within the BIIC or on the VAXBI except during the diagnostic mode. The three classes of EV codes are as follows. Table 12 lists the event codes.

- Summary EV codes that provide status at the end of a transaction
- Status EV codes that provide status during a transaction
- Special EV codes that provide self-test status and bus timeout information

Table 12 • VAXBI 78732 BCI Event Code Assignments

BCI EV Line 4	3	2	1	0	Mnemonic	Description
H	H	H	H	H	NEV	The default deasserted state
H	H	H	H	L	MCP	Master-port transaction complete
H	H	H	L	H	AKRSD	Acknowledge received from slave read data
H	H	H	L	L	BTO	Bus timeout
H	H	L	H	H	STP	Self-test passed
H	H	L	H	L	RCR	Retry confirmation received for master-port command
H	H	L	L	H	IRW	Internal register written
H	H	L	L	L	ARCR	Advanced retry confirmation received

BCI EV Line 4	3	2	1	0	Mnemonic	Description
H	L	H	H	H	NICI	No acknowledge or illegal confirmation received for interrupt command
H	L	H	H	L	NICIPS	No acknowledge of illegal confirmation received for Force-bit interprocessor/stop command
H	L	H	L	H	AKRE	Acknowledge confirmation received for error vector
H	L	H	L	L	IAL	Identification arbitration lost
H	L	L	H	H	EV4	External vector selected—level 4
H	L	L	H	L	EV5	External vector selected—level 5
H	L	L	L	H	EV6	External vector selected—level 6
H	L	L	L	L	EV7	External vector selected—level 7
L	H	H	H	H	STO	Stall timeout on slave transaction
L	H	H	H	L	BPS	Bad parity received during slave transaction
L	H	H	L	H	ICRSD	Illegal confirmation received for slave data
L	H	H	L	L	BBE	Bus busy error
L	H	L	H	H	AKRNE4	Acknowledge confirmation received for nonerror vector—level 4
L	H	L	H	L	AKRNE5	Acknowledge confirmation received for nonerror vector—level 5
L	H	L	L	H	AKRNE6	Acknowledge confirmation received for nonerror vector—level 6
L	H	L	L	L	AKRNE7	Acknowledge confirmation received for nonerror vector—level 7
L	L	H	H	H	RDSR	Read data substitute or reserved status code received
L	L	H	H	L	ICRMC	Illegal confirmation received for master-port command
L	L	H	L	H	NCRMC	No acknowledge confirmation received for master-port command
L	L	H	L	L	BPR	Bad parity received
L	L	L	H	H	ICRMD	Illegal confirmation received by master-port data cycle
L	L	L	H	L	RTO	Retry timeout
L	L	L	L	H	BPM	Bad parity received during master-port transaction
L	L	L	L	L	MTCE	Master transmit error check

BCI Power Status Signals

The power status lines provide information to the user's interface related to the condition of the ac , and dc power.

BCI ac Power ($\overline{\text{BCI AC LO}}$)—This signal is a buffered and synchronized $\overline{\text{BI AC LO}}$ signal to the user's interface to allow the interface to monitor the power status. The BIIC receives the state of $\overline{\text{BI AC LO}}$ signal and allows a two-cycle delay to verify that the received state is stable. It then transmits the state of the signal to the $\overline{\text{BCI AC LO}}$ line. The BIIC will not change the state of the $\overline{\text{BCI AC LO}}$ signal unless the state of the line is different from the preceding state for two cycles.

BCI dc Power ($\overline{\text{BCI DC LO}}$)—This signal indicates the status of the dc power to the user's interface. The BIIC asynchronously asserts the $\overline{\text{BCI DC LO}}$ line following the assertion of the $\overline{\text{BI DC LO}}$ signal. The maximum assertion delay is specified by the ac timing specifications. When the $\overline{\text{BI DC LO}}$ is asserted for more than 50 nanoseconds, the $\overline{\text{BCI DC LO}}$ line is asserted. The BIIC deasserts the $\overline{\text{BCI DC LO}}$ synchronously with the T_{dde} cycles following a valid deassertion of $\overline{\text{BI DC LO}}$. A valid deassertion requires that the $\overline{\text{BI DC LO}}$ line remain stable and deasserted for two consecutive cycles. While the $\overline{\text{BI DC LO}}$ is asserted, the BIIC disables all the VAXBI drivers including the BCI D<31:00>, BCI I<3:0>, and BCI P0 drivers. This facilitates testing of the VAXBI module by allowing the modules to be tested independently of the BIIC.

When the node reset bit of the VAXBI control and status register is set, the BIIC asserts the $\overline{\text{BCI DC LO}}$ line regardless of the state of the $\overline{\text{BI DC LO}}$ line. When the $\overline{\text{BCI DC LO}}$ is deasserted, the BIIC initiates the self-test. The BIIC maintains a valid low-level output voltage on the $\overline{\text{BCI DC LO}}$ when the $\overline{\text{BI DC LO}}$ is asserted although the V_{CC} supply voltage to the BIIC is in transition.

The BIIC loads the node identification, the device type and revision, and parity mode identification during the cycle before the $\overline{\text{BCI DC LO}}$ signal is deasserted. Normally, the user's interface transfers the this information to the BCI I<3:0>, BCI D<31:00>, and BCI P0 lines while the $\overline{\text{BCI DC LO}}$ is asserted. The BCI D<31:00> and BCI P0 lines contain internal pullups circuits that are enabled during assertion of the $\overline{\text{BCI DC LO}}$ lines and cause the lines to become high by default. The default condition may be useful in designing a node. The output current characteristics of the pullup devices should be verified.

BCI Clock Signals

The clock signals from the user's interface provide the basic timing information for the VAXBI interface and user's interface.

BCI Time (BCI TIME L)—This is a 20-MHz TTL timing signal supplied by the VAXBI clock receiver in the user's interface. It is used with the BCI PHASE L signal by the BIIC and user's interface to generate all the required timing signals.

BCI Phase (BCI PHASE L)—This is a 5-MHz TTL timing signal supplied by the VAXBI clock receiver in the user's interface. It is used with the BCI TIME L signal by the BIIC and the user's interface to generate all required timing signals.

• VAXBI Interface Registers

The VAXBI interface contains a complete set of VAXBI registers and four general purpose registers that are available to user-defined node logic. The BIIC registers are located in the first 256 bytes of the BIIC nodespace, which is designated as the BIIC CSR space. Most locations in the CSR node space are unused. When a read-type command accesses the unused locations, the BIIC responds by reading zeros. When a write-type command accesses the unused locations, the BIIC accepts the command but ignores the data. Figure 5 shows the BIIC register format and the address assignments.

	31 24 23 16 15 08 07 00	
bb+00	DEVICE REGISTER	
bb+04	RESERVED	VAXBI CONTROL / STATUS REGISTER
bb+08	BUS ERROR REGISTER	
bb+0C	ERROR INTERRUPT CONTROL	ERROR VECTOR
bb+10	RESERVED	INTERRUPT DESTINATION
bb+14	IP INTERRUPT MASK REGISTER	RESERVED
bb+18	RESERVED	FORCE IPINTR/STOP DESTINATION
bb+1C	IP INTERRUPT SOURCE REGISTER	RESERVED
bb+20	STARTING ADDRESS	UNUSED
bb+24	ENDING ADDRESS	UNUSED
bb+28	BCI CONTROL/STATUS REGISTER	
bb+2C	WRITE STATUS REGISTER	
bb+30		FORCE IPINTR/STOP COMMAND
bb+34	UNUSED	
bb+38	UNUSED	
bb+3C	UNUSED	
bb+40	USER INTERFACE INTR CONTROL	VECTOR
bb+44	UNUSED	
bb+EC		
bb+F0	GENERAL PURPOSE REGISTER 0	
bb+F4	GENERAL PURPOSE REGISTER 1	
bb+F8	GENERAL PURPOSE REGISTER 2	
bb+FC	GENERAL PURPOSE REGISTER 3	

Figure 5 • VAXBI 78732 BIIC Registers and Address Assignments

The BIIC registers can be accessed by a VAXBI transaction from a node to its associated BIIC or to a BIIC associated with another node or by a loopback transaction from the master-port interface of a node. When the registers are accessed by a loopback command from a master-port interface or by a VAXBI transaction, the high-order bits BI D<29:13> of the address that select the node address space are ignored by the BIIC. The low-order bits (D<12:01>) determine whether an internal register is selected. A register is selected when bits D<12:08> are equal to zero and the remaining bits D<07:00> specify the register to be accessed.

The BIIC supports write-mask transactions but does not support the lock functionality for the internal registers. The interlock-read with cache intent (IRCI) transaction is performed as a normal read transaction and the unlock-write-mask with cache intent (UWMCI) transaction is performed as a write-mask transaction.

BIIC Registers Description

The register description tables contain a code in the "Type" column that defines the type of bits in the register. Table 13 lists the codes and definitions.

Table 13 • VAXBI 78732 Register Type Codes

Type	Definition*
DCLOC	Cleared following a successful self-test when the $\overline{\text{BCI DC LO}}$ signal from the BIIC is deasserted.
DCLOL	Loaded on the last cycle in which $\overline{\text{BCI DC LO}}$ is asserted. Set if the BCI signal lines are not driven during this cycle.
DCLOS	Set following a successful self-test.
DMW	Can be written during the BIIC diagnostic mode which is reserved for use by Digital.
RO	Read-only.
R/W	Normal read/write
SC	Special case. Defined in the *VAXBI* System Reference Manual.
STOPC	Cleared by a STOP command to the node.
STOPS	Set by a Stop command to the node.
W1C	Write-1-to-clear. Cannot be set by the user's interface.
NA	Not applicable.
DS	Disables selection.

*Set refers to high level and clear refers to a low level.

Device Register—The device (DTYPE) register contains information to identify the node. It consists of a device revision and device type field that are loaded from the BCI D<31:00> lines during the last cycle in which the $\overline{\text{BCI DC LO}}$ line is asserted. The register will be set to all one bits if the information the BCI D<31:00> lines is not present when the $\overline{\text{BCI DC LO}}$ line is asserted. Figure 6 shows the format of the register information and Table 14 describes the bits.

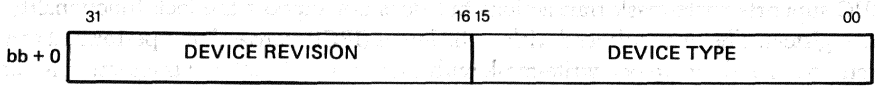


Figure 6 ▪ VAXBI 78732 Device Register Format

Table 14 ▪ VAXBI 78732 Device Register Description

Bit	Type	Description
31:16	R/W, DMW, DCLOL	Identifies the revision level of the device
15:00	R/W, DMW, DCLOL	Identifies the type of node

Control and Status Register—The VAXBI control and status register (VAXBICSR) contains interface identification, error and control information. The register information is shown in Figure 7 and described in Table 15.

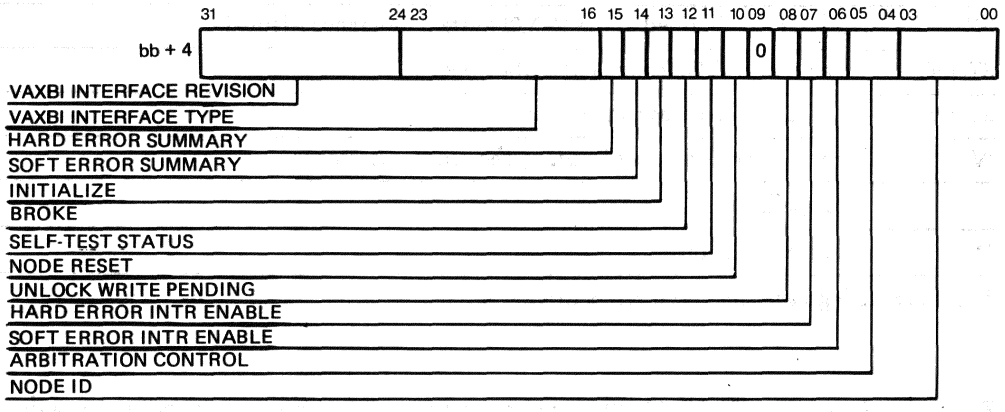


Figure 7 ▪ VAXBI 78732 Control and Status Register Format

Table 15 - VAXBI 78732 VAXBI Control and Status Register Description

Bit	Type	Description
31:24	RO	IREV (Interface revision)—Indicates the revision level of the BIIC. The content of this field is incremented for each major revision.
23:16	RO	ITYPE (Interface type)—Contains the following code (0000 0001).
15	RO	HES (Hard error summary)—Set to indicate that one or more hard errors bits in the bus error register are set.
14	RO	SES (Soft error summary)—Set to indicate that one or more soft error bits are set in the bus error register.
13	W1C, DCLOS STOPS	INIT(Initialize)—Implementation dependent.
12	W1C, DCLOS	BROKE (Self-test fail)—Set to indicate that the BIIC did not pass the self-test routine.
11	R/W, DCLOS	STS (Self-test status)—Set to indicate that the BIIC has passed the self-test.
10	SC	NRST (Node reset)—Set to initiate a complete node self-test. Reading this bit returns a zero. The STS bit is automatically reset by the BIIC when this bit is set to allow the recording of the new test results.
09	RO	Reserved and cleared to zero.
08	W1C, DCLOC, SC	UWP (Unlock write pending)—Set to indicate that an interlock read with cache intent (IRCI) transaction has been completed by the master-port interface at this node and this node has not issued a subsequent unlock write mask with cache intent (UWMCI) command. Cleared by a master-port UWMCI transaction that has been successfully completed. If a UWMCI transaction is attempted by the master-port interface when this bit is not set, the ISE bit in the bus error register will be set.
07	R/W, DCLOC, STOPC	HEIE (Hard error interrupt enable)—Set to enable the generation of error interrupt when HES (bit 15) is set.
06	R/W, DCLOC, STOPC	SEIE (Soft error interrupt enable)—Set to enable an error interrupt to be generated when SES (bit 14) is set.
05:04	R/W, DCLOC	ARB (Arbitration control)—Indicates the mode of arbitration to be used as follows: **Bit 5 / Bit 4 — Description** 0 0 — Dual round robin arbitration 0 1 — Fixed high priority (reserved) 1 0 — Fixed low priority (reserved) 1 1 — Disable arbitration (reserved)
03:00	RO, DMW, DCLOC	NODE ID (node identification)—An identification number assigned to the node. This information is loaded from the BCI I<3:0> lines during the last cycle in which the $\overline{\text{BCI DC LO}}$ signal is asserted.

Bus Error Register—The bus error register (BER) provides hard error and soft error indications, and parity generation control. All bits in the error register can be set during the VAXBI and loopback transactions except where noted. The register information is shown in Figure 8 and described in Table 16.

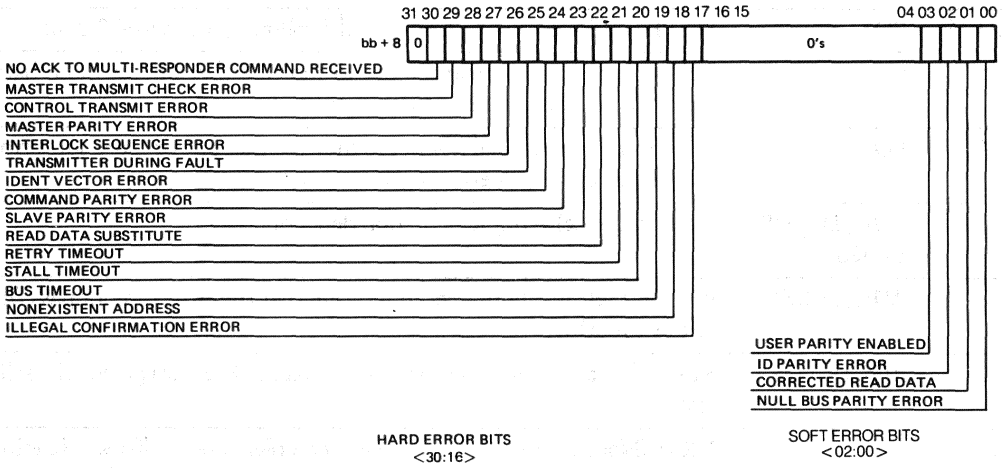


Figure 8 ▪ VAXBI 78732 Bus Error Register Format

Table 16 ▪ VAXBI 78732 Bus Error Register Description

Bit	Type	Description
31	RO	Reserved and cleared to zero.
30	W1C, DCLOC	NO ACK (No acknowledge)—Set when a master-port receives a no acknowledge command in response to an INVAL, INTER, IPINTER, BDCST, or RESERVED command.
29	W1C, DCLOC	MTCE (Master transmit check error)—Set when the data transmitted is different from the data received. During the transaction cycles in which the master is the only source of data on the BI data path, the BIIC verifies that the data to be transferred from the master is the same as the data that the master receives. If the data is different, this bit is set. This check is not performed when the encoded ID is transferred from the master during embedded ARB cycles.
28	W1C, DCLOC	CTE (Control transmit error)—Set to indicate that a node has detected a deasserted state of the $\overline{\text{BI NO ARB}}$ $\overline{\text{BI BST}}$ or $\overline{\text{BI CNF}}$<2:0> line during a cycle when the node is attempting to assert the signal. No check is performed during burst-mode transactions.
27	W1C, DCLOC	MPE (Master parity error)—Set if the master detects a parity error on the bus during a read-type or vector acknowledge data cycle.

Bit	Type	Description
26	W1C, DCLOC	ISE (Interlock sequence error)—Set when the node successfully completes a unlock write-mask with cache intent (UWMCI) command and no corresponding interlock read with cache intent (IRCI) was previously issued. The unlock write pending bit in the VAXBI control and status register is not set.
25	W1C, DCLOC	TDF (Transmitter during fault)—Set when the master or slave detects a parity error during the following cycles in which the master or slave was responsible for transferring the proper parity on the VAXBI bus. This bit is not set by parity errors that occur during loopback transactions. —Command/address cycles set by the master —Read-type acknowledge data cycles set by the slave —Write-type data cycles set by the master —Broadcast data cycles set by the master —Vector acknowledge data cycles set by the slave —Embedded arbitration cycles with the encoded master identification set by the master
24	W1C, DCLOC	IVE (Identification vector error)—Set when an acknowledge response is not received from the master to indicate that the interrupt vector was not received correctly.
23	W1C, DCLOC	CPE (Command parity error)—Set if a parity error is detected in a command/address cycle of a VAXBI or loopback transaction.
22	W1C, DCLOC	SPE (Slave parity error)—Set when a parity error is detected by a slave during a write-type acknowledge, write-type stall, or broadcast acknowledge data cycle.
21	W1C, DCLOC	RDS (Read data substitute)—Set when read data substitute of reserved status code is received during a read-type or vector status identification data cycle. Valid parity must be received during both transactions.
20	W1C, DCLOC	RTO (Retry timeout)—Set when the master receives 4096 consecutive retry responses for the same master port transaction.
19	W1C, DCLOC	STO (Stall timeout)—Set when the slave port asserts the stall code information on the $\overline{\text{BCI RS}}$<1:0> lines for 128 consecutive cycles.
18	W1C, DCLOC	BTO (Bus timeout)—Set when the node fails to initiate one of several transaction that are pending before 4096 cycles have elapsed.
17	W1C, DCLOC	NEX (nonexistent address)—Set when the node receives a no acknowledge response for a read-type or write-type command after a successful parity check of the node and master parity check has occurred.
16	RO	ICE (Illegal confirmation error)—Set by the master or slave node when a reserved or illegal confirmation code is received by the BIIC.
15:04	RO	Reserved and cleared to zero

Bit	Type	Description
3	RO, DCLOC	UPEN (User parity enable)—Indicates the parity mode of the BIIC. Set to indicate that the user's interface will generate parity. Cleared to indicate that the BIIC will generate parity. The user's interface provides parity on the BCI P0 line when data is requested from the user's interface. The levels are reversed from those on the $\overline{\text{BCI P0}}$ line.
2	W1C, DCLOC	IPE (Identification parity error)—Set if a parity error is detected on the BI I<3:0> lines when the encoded ID of the master is asserted during embedded arbitration cycles. This bit is not set during loopback transactions.
1	W1C, DCLOC	CRD (Corrected read data)—Set when the master receives a corrected read data status code after valid parity has been received. Valid parity must be received during the data cycle that contains the corrected read data code. This bit is set although the transaction may be aborted after the status code has been received.
0	W1C, DCLOC	NPE (Null bus parity error)—Set when odd parity is detected on the bus during the second cycle of a two-cycle sequence during which the $\overline{\text{BI ARB}}$ and $\overline{\text{BI BSY}}$ signals were not asserted.

Error Interrupt Control Register—The error interrupt control register (EINTRCSR) controls the operation of the interrupts initiated when the BIIC detects a bus error or when the force bit in this register is set. An error interrupt request can be initiated provided the appropriate interrupt enable bit in the BIIC control and status register was previously set. The register information is shown in Figure 9 and described in Table 17.

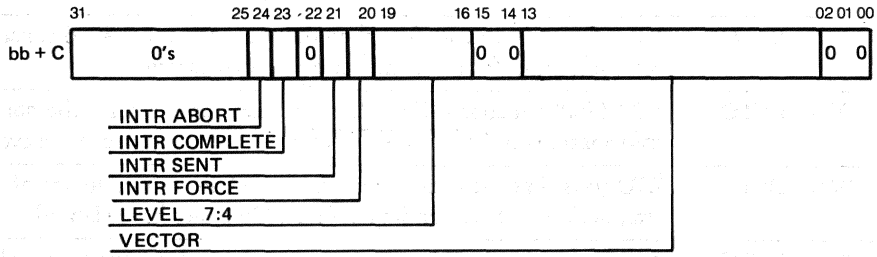


Figure 9 ▪ VAXBI 78732 Error Interrupt Control Register Format

Table 17 • VAXBI 78732 Error Interrupt Control Register Description

Bit	Type	Description
31:25	RO	Reserved and cleared to zero
24	W1C, DCLOC, SC	INTRAB (Interrupt abort)—Set by the BIIC when an interrupt command initiated by this register is aborted. A command is aborted when no acknowledge or an illegal confirmation code is received. When set, the BIIC is not inhibited from initiating or responding to subsequent interrupt or identification transactions. This bit is cleared by the user's interface.
23	W1C, DCLOC, SC	INTRC (Interrupt complete)—Set when the vector for an error interrupt has been successfully transferred or when an interrupt command transferred under the control of this register is aborted. This bit is cleared when the error interrupt request has been removed. When set, no interrupts can be generated by this register. When this bit is set, this register will not respond to identification transactions.
22	RO	Reserved and cleared to zero.
21	W1C, DCLOC, STOPC, SC	INTRS (Interrupt sent)—Set when an interrupt command has been sent. Removal of the error interrupt request will clear this bit during an identification transaction following the detection of a level match and a master identification match. When cleared, the interrupt can be sent again if the node loses the identification arbitration or if the node wins the arbitration but the vector transmission fails.
20	R/W, DCLOC STOPC	INTRF (Interrupt force)—When set, an error interrupt is posted the same as in the bus error register except that the request is not qualified by HEIE bit 7 and SEIE bit 6 in the control and status register.
19:16	R/W, DCLOC	LEVEL 7:4 (Interrupt command level)—Indicates the level(s) at which the interrupt commands under control initiated by this register are transferred.
15:14	RO	Reserved and cleared to zero
13:2	R/W, DCLOC	VECTOR (Error interrupt vector)—Contains the vector used during the error interrupt sequence.
01:00	RO	Reserved and cleared to zero

Interrupt Destination Register—The interrupt destination (INTRDES) register identifies the nodes that have been selected by the interrupt commands. The destination is transferred during the interrupt command and is monitored by all the nodes to determine which node is to respond. The register information is shown in Figure 10 and described in Table 18.

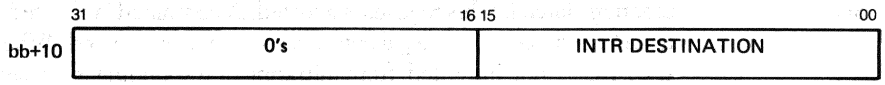


Figure 10 • VAXBI 78732 Interrupt Destination Register Format

Table 18 • VAXBI 78732 Interrupt Destination Register Description

Bit	Type	Description
31:16	RO	Reserved and cleared to zero.
15:00	R/W, DCLOC	INTR DEST (Interrupt destination)—During an identification command, a node compares the decoded identification from the master with the coded information in this field. When the information code is the same, this node responds to the identification sequence provided that the unserviced interrupt request is the same level as the level transferred during the identification command.

Interprocessor Interrupt Mask Register—The interprocessor (IP) interrupt mask (IPINTRMSK) register identifies the nodes that are allowed to send interprocessor interrupts to the BIIC. The register information is shown in Figure 11 and described in Table 19.

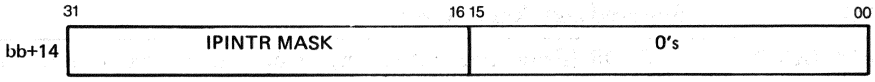


Figure 11 • VAXBI 78732 IP Interrupt Mask Register Format

Table 19 • VAXBI 78732 IP Interrupt Mask Register Description

Bit	Type	Description
31:16	R/W, DCLOC	IPINTR MASK (Interprocessor Interrupt mask)—When a bit in this field is set, the interprocessor interrupts directed to the BIIC from the corresponding node will allow selection provided that the IPINTREN (bit 5) of the BCI control and status register is set.
15:00	RO	Reserved and cleared to zero.

Force-bit IPINTR/STOP Destination Register—The force-bit interprocessor interrupt/stop destination (FIPSDES) register identifies the nodes that will receive the force-bit interprocessor interrupts or stop commands issued by the BIIC. The register information is shown in Figure 12 and described in Table 20.

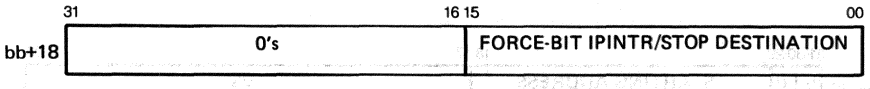


Figure 12 ▪ VAXBI 78732 Force-bit IPINTR/STOP Destination Register Format

Table 20 ▪ VAXBI 78732 Force-bit IPINTR/STOP Destination Register Description

Bit	Type	Description
31:16	RO	Reserved and cleared to zero.
15:00	R/W, DCLOC	FORCE-BIT IPINTR/STOP DEST (Force-bit interprocessor interrupt/stop destination)—Command/address information provides by the user's interface for the master-port interprocessor interrupt transactions.

Interprocessor Interrupt Source Register—The interprocessor interrupt source register (IPINTRS) stores the decoded identification of a node that sends an interprocessor interrupt command to the BIIC. The register information is shown in Figure 13 and described in Table 21.

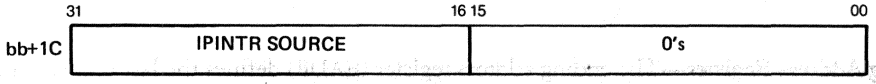


Figure 13 ▪ VAXBI 78732 IP Interrupt Source Register Format

Table 21 ▪ VAXBI 78732 IP Interrupt Source Register Description

Bit	Type	Description
31:16	W1C, DCLOC, SC	IPINTR SOURCE (Interprocessor interrupt source)—Each bit in this field corresponds to one node and is set when the destination of an interprocessor command to the BIIC is the same as the identification in this field.
15:00	RO	Reserved and cleared to zero.

Starting Address Register—The starting address register (SADR) defines the starting address a 256-Kbyte block of storage in a memory space or I/O space excluding node space or multicast space. The end of the block is defined by the ending address register. If the value of this address is greater than the ending address value, the address will not be recognized. The register information is shown in Figure 14 and described in Table 22.

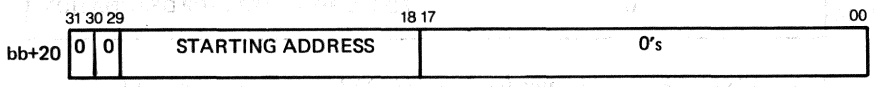


Figure 14 ▪ VAXBI 78732 Starting Address Register Format

Table 22 ▪ VAXBI 78732 Starting Address Register Description

Bit	Type	Description
31:30	RO	Reserved and cleared to zero.
29:18	R/W, DCLOC	START ADDR (Starting address)—Identifies the address of the first location of a 256-Kbyte block of storage to be recognized by the BIIC for selection of the slave port.
17:00	RO	Reserved and cleared to zero.

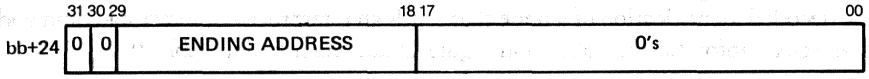


Figure 15 ▪ VAXBI 78732 Ending Address Register Format

Ending Address Register—The ending address register (EADR) defines the last address of a 256-Kbyte block of storage in a memory space or I/O space excluding node space or multicast space. The starting address of the block is defined by the starting address register. If the starting address value of this address is greater than the ending address value, the address will not be recognized. The register information is shown in Figure 15 and described in Table 23.

Table 23 • VAXBI 78732 Ending Address Register Description

Bit	Type	Description
31:30	RO	Reserved and cleared to zero.
29:18	R/W, DCLOC	END ADDR (Ending address)—Specifies an address value that is greater by one than the highest address recognized by the BIIC for selection by the slave port. The address must be the first location of the 256-Kbyte block of addresses. If the starting address register contains the value 1C44 0000 (hexadecimal), and the ending address register contains the value 1D68 0000 (hexadecimal), then the BIIC will recognize addresses 1C44 000 through 1D67 FFFF for selection of the slave port.
17:00	RO	Reserved and cleared to zero.

BCI Control and Status Register—The BCI control and status register (BCICSR) contains command enable bits to control the operation of the slave-port interface. The register information is shown in Figure 16 and described in Table 24. Figure 16 shows the register format and Table 14 lists the function of the information.

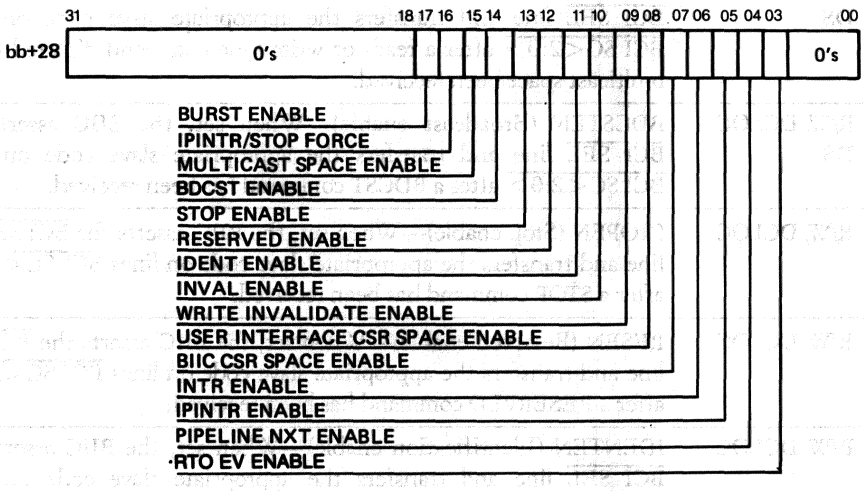


Figure 16 • VAXBI BCI Control and Status Register Format

Table 24 • VAXBI BCI Control and Status Register Description

Bit	Type	Description
31:18	RO	Reserved and cleared to zero.
17	R/W, DCLOC, NA	BURSTEN (Burst enable)—When set, the BIIC asserts the $\overline{\text{BI NO ARB}}$ signal after the next successful arbitration until this signal is reset or until the $\overline{\text{BCI MAB}}$ signal is asserted. The assertion of the $\overline{\text{BI MAB}}$ signal clears the burst-mode state of the BIIC but does not clear this bit. The $\overline{\text{BI NO ARB}}$ signal will remain asserted through the next successful arbitration unless it is cleared by a subsequent transaction.
16	R/W, DCLOC, SC, NA	IPINTR/STOP FORCE (Interprocessor interrupt/stop force)—When set, the BIIC arbitrates for the bus and transfers an IPINTR or STOP command. The command that is sent is stored in the force-bit interprocessor command register and the destination field is contained in the force-bit interprocessor/stop destination register. This bit is cleared by the BIIC after the interprocessor interrupt transaction. If the transmission fails, a no acknowledge or illegal confirmation for force-bit for INTR/STOP command (NICIPS) event code is sent and the no acknowledge to multiresponder command received (NMR) bit in the bus error register is set.
15	R/W, DCLOC, DS	MSEN (Multicast space enable)—When set, the BIIC asserts the $\overline{\text{BCI SEL}}$ line and transfers the appropriate slave code on lines $\overline{\text{BCI SC<2:0>}}$ after a read- or write-type command directed to the multicast space been received.
14	R/W, DCLOC, DS	BDCSTEN (Broadcast enable)—When set, the BIIC asserts the $\overline{\text{BCI SEL}}$ line and transfers the appropriate slave code on lines $\overline{\text{BCI SC<2:0>}}$ after a BDCST command has been received.
13	R/W, DCLOC	STOPEN (Stop enable)—When set, the BIIC asserts the $\overline{\text{BCI SEL DS}}$ line and transfers the appropriate slave code on lines $\overline{\text{BCI SC<2:0>}}$ after a STOP command has been received.
12	R/W, DCLOC	RESEN (Reserved enable)—When set, the BIIC asserts the $\overline{\text{BCI SEL}}$ line and transfers the appropriate slave code on lines $\overline{\text{BCI SC<2:0>}}$ after a RESERVED command has been received.
11	R/W, DCLOC	IDENTEN (Identification enable)—When set, the BIIC asserts the $\overline{\text{BCI SEL}}$ line and transfers the appropriate slave code on lines $\overline{\text{BCI SC<2:0>}}$ after an identification command has been received. The BIIC will participate in an identification transaction to this node when this bit is cleared.
10	R/W, DCLOC, DS	INVALEN (Invalidate enable)—When set, the BIIC asserts the $\overline{\text{BCI SEL}}$ line and transfers the appropriate slave code on lines $\overline{\text{BCI SC<2:0>}}$ after an INVAL command has been received.

Bit	Type	Description
09	R/W, DCLOC, SC	WINVALEN (Write invalid enable)—When set, the BIIC asserts the $\overline{\text{BCI SEL}}$ line and transfers the appropriate slave code on lines $\overline{\text{BCI SC<2:0>}}$. This occurs after the BIIC receives a write-type command with an address not within the range of the starting and ending address register values and not within the I/O space.
08	R/W, DCLOC, SC	UCSREN (User's interface CSR space enable)—When set, the BIIC asserts the $\overline{\text{BCI SEL}}$ line and transfers the appropriate slave code on lines $\overline{\text{BCI SC<2:0>}}$ after receiving a read- or write-type command directed to the control and status register in the user's interface.
07	R/W, DCLOC, SC	BICSREN (BIIC control/states register space enable)—When set, the BIIC asserts the $\overline{\text{BCI SEL}}$ line and transfers the appropriate slave code on lines $\overline{\text{BCI SC<2:0>}}$ after receiving a read- or write-type command directed to the control and status register in the BIIC. The BIIC always participates in transactions that access this space.
06	R/W, DCLOC, DS	INTREN (Interrupt enable)—When set, the BIIC asserts the $\overline{\text{BCI SEL}}$ line and transfers the appropriate slave code on lines $\overline{\text{BCI SC<2:0>}}$ after receiving a read- or write-type command directed to the BIIC.
05	R/W, DCLOC, SC	IPINTREN (Interprocessor interrupt enable)—When set, the BIIC asserts the $\overline{\text{BCI SEL}}$ line and transfers the appropriate slave code on lines $\overline{\text{BCI SC<2:0>}}$ after receiving an IPINTR command from a node that is included in the interprocessor interrupt mask register. The BIIC receives the IPINTR commands regardless of the state of this bit.
04	R/W, DCLOC, NA	PNXTEN (pipeline next enable)—When set, the BIIC provides an additional next data word cycle after the last longword is transferred during write-type and broadcast transactions. This cycle can be used to transfer pointers to first-in/first-out (FIFO) buffers in the master port.
03	R/W, DCLOC, NA	RTOEVEN (Retry timeout event enable)—When set, the BIIC transfers a retry timeout (RTO) instead of retry confirmation received for master-port command (RCR) event code on lines $\overline{\text{BCI EV<4:0>}}$ after a retry timeout has occurred. If this bit is cleared, the RTO bit in the bus error register will be set and an error interrupt will be generated if enabled.
02:00	RO	Reserved and cleared to zero.

Write Status Register—The write status (WSTAT) register indicates which of the general purpose registers have received information during a write VAXBI transaction. The register information is shown in Figure 17 and described in Table 25.

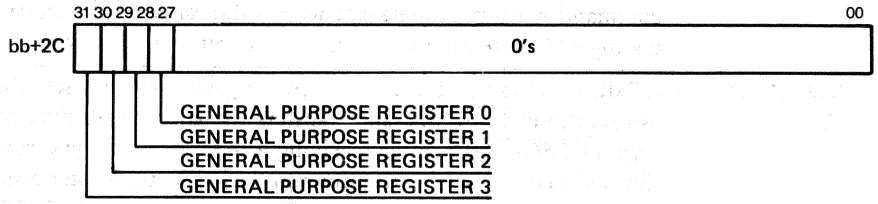


Figure 17 • VAXBI 78732 Write Status Register Format

Table 25 • VAXBI 78732 Write Status Register Description

Bit	Type	Description
31	W1C, DCLOC	GPR3 (General purpose register 3)—Set by a write VAXBI transaction to general purpose register 3 if valid parity is received. This bit is not set by loopback transactions.
30	W1C, DCLOC	GPR2 (General purpose register 2)—Set by a write VAXBI transaction to general purpose register 2 if valid parity is received. This bit is not set by loopback transactions.
29	W1C, DCLOC	GPR1 (General purpose register 1)—Set by a write VAXBI transaction to general purpose register 1 if valid parity is received. This bit is not set by loopback transactions.
28	W1C, DCLOC	GPR0 (General purpose register 0)—Set by a write VAXBI transaction to general purpose register 0 if valid parity is received. This bit is not set by loopback transactions.
27:00	RO	Reserved and cleared to zero.

Force-bit Interprocessor Interrupt/Stop Command Register—The force-bit IPINTR/STOP command (FIPSCMD) register allows an interprocessor interrupt or stop transaction to be initiated when one of the interrupt force-bits in the user interface interrupt control register is set. The register information is shown in Figure 18 and described in Table 26.

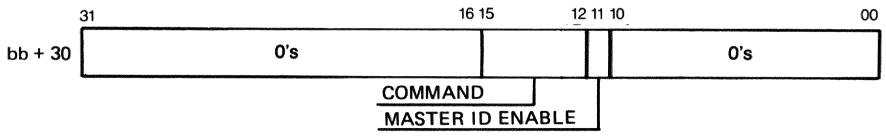


Figure 18 • VAXBI 78732 Force-bit IPINT/STOP Command Register Format

Table 26 • VAXBI 78732 Force-bit IPINT/STOP Command Register Description

Bit	Type	Description
31:16	RO	Reserved and cleared to zero.
15:12	R/W, DCLOC	CMD (Command)—Indicates the 4-bit command code used to initiate an interprocessor interrupt or stop transaction when one of the interrupt force-bits in the user interface interrupt control register is set. The command codes are **Bit 15 14 13 12 — Command** 1 1 1 1 — IPINTR (Interprocessor interrupt) 1 1 0 0 — STOP
11	R/W, DCLOC	MIDEN (Master identification enable)—Set to enable the master identification to be transferred on the $\overline{\text{BI D<31:00>}}$ lines when the command field (bits 15:12) contains an interprocessor command code during the command/address cycle. When the command field contains a STOP command, this bit should be cleared.
10:00	RO	Reserved and cleared to zero.

User Interface Interrupt Control Register—The user interface interrupt control register (UNITRCSR) controls the operation of the interrupts initiated by the user's interface. The register information is shown in Figure 19 and described in Table 27. The interrupt request referred to in Table 27 are interrupts initiated by the $\overline{\text{BCI INT<7:0>}}$ lines or by setting the force-bits in this register.

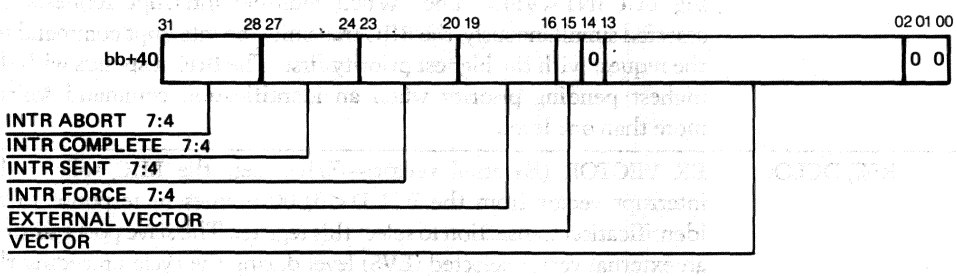


Figure 19 • VAXBI 78732 User Interface Interrupt Control Register Format

Table 27 • VAXBI 78732 User Interface Interrupt Control Register Description

Bit	Type	Description
31:28	W1C, DCLOC, SC	INTRAB (Interrupt abort 7:4)—These bits correspond to interrupt levels 7 through 4. A bit is set when an interrupt command sent under control of this register is aborted causing a no acknowledge or illegal confirmation code to be received by the BIIC. Cleared by the user's interface. The state of this bit does not prevent the BIIC from responding to other interrupt or identification transactions.
27:24	W1C, DCLOC, SC	INTRC (Interrupt complete 7:4)—These bits correspond to interrupt levels 7 through 4. A bit is set when the vector for an interrupt has been successfully transmitted or when an interrupt command sent under control of this register is aborted. When a bit is set, no additional interrupt requests at the level specified will be generated. No response to identification transactions will occur when this bit is set at the IDENT level. A bit is cleared when the corresponding interrupt request is removed.
23:20	W1C, DCLOC, STOPC, SC	SENT (Interrupt sent)—These bits correspond to the interrupt levels 7 through 4. A bit is set when an interrupt command for the corresponding level has been successfully transferred. This bit is cleared during an identification command that follows the detection of a level and a match of the master identification. This bit is cleared when the interrupt request is deasserted. When cleared, an interrupt request can be sent again if the BIIC has lost the identification arbitration or if the BIIC has won the arbitration but the vector transmission failed.
19:16	R/W, DCLOC, STOPC	FORCE (Interrupt force 7:4)—These bits correspond to interrupt levels 7 through 4. When set, the BIIC generates interrupts at the specified level. Setting a bit is equivalent to asserting the corresponding $\overline{\text{BCI INT}}$<7:4> line. When multiple interrupt requests are asserted simultaneously, the BIIC transmits the interrupt command for the request with the highest priority first. The BIIC responds with the highest pending priority when an identification command solicits more than one level.
15	R/W, DCLOC	EX VECTOR (External vector)—When set, the BIIC solicits the interrupt vector from the BCI D<31:00> lines in response to an identification transaction to select this register. The slave port transfers an external vector selected (EVS) level during the cycle preceding the vector transfer on the BCI D<31:00> lines. A slave-port interface that is using the BIIC must stall the vector by transferring a stall code on the $\overline{\text{BCI RS}}$<2:0> lines for at least one cycle during the identification arbitration cycle before transmitting an acknowledge (with vector) or a retry response.

Bit	Type	Description
14	RO	Reserved and cleared to zero.
13:02	R/W, DCLOC	VECTOR (Interrupt sequence vector)—Contains the vector used during the user's interface interrupt sequences except when the external vector bit 15 is set. The vector is transferred when the BIIC wins the identification arbitration that is the same as the conditions specified by the user interface interrupt control register.
01:00	RO	Reserved and cleared to zero.

General Purpose Registers

The BIIC contains four general purpose registers (GPR0 through GPR3) that are available to the user and are implementation specific. The type of bits in these registers are R/W and DCLOC. When information is written to a GPR, the write status register identifies the GPR that received the information. Figure 20 shows the register format.

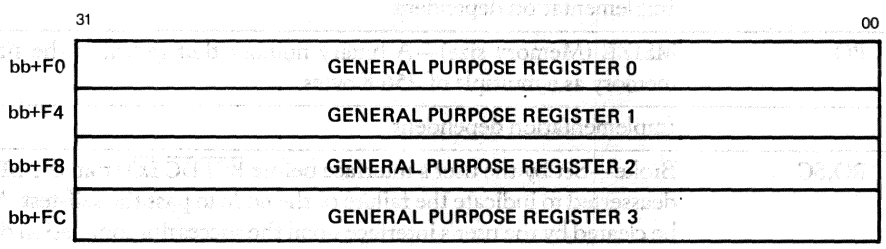


Figure 20 • VAXBI 78732 General Purpose Register Format

Slave-only Status Register

The slave-only status register (SOSR) is located out of the BBIC CSR space and is used by slave nodes that have a device-type code and contain zeros in bits 14:08. The register format is shown in Figure 20A and the information is defined in Table 27A. The Broke bit and Memory Size (MSIZE) field must contain valid information. This is a read-only register and must not be written into.

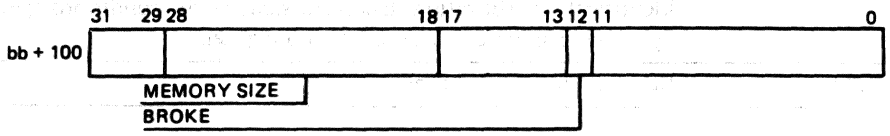


Figure 20A • VAXBI 78732 Slave-only Status Register Format

Table 27A • VAXBI 78732 Slave-only Status Register Description

Bit	Type	Description
31:29		Implementation dependent
28:18	RO	MSIZE (Memory size)—A binary number that indicates the size of memory as a multiple of 256 Kbytes.
17:13		Implementation dependent
12	RO,SC	Broke—Set by the user's interface before $\overline{\text{BCI DC LO}}$ from the BIIC is deasserted to indicate the failure of the node to pass the self-test. Must be cleared by the user's interface upon the successful completion of the self-test.
11:00		Implementation dependent

Receive Console Data Register

The receive console data (RXCD) register, implemented by VAXBI nodes that have a console terminal on the VAXBI, is used to receive data from other consoles. Nodes without a console must issue a no acknowledge response to read commands to this register or return a longword with the Busy 1 (bit 15) set before the Broke bit of the slave-only register is cleared. The RXCD register responds to longword VAXBI bus transactions directed to this register. If the RXCD register is in a primary console node, a lock bit must be implemented in the interface. When the optional upper word of a unlock write with cache intent (UWMCI) command to this register is not used, the mask bits can be ignored (assumed to be all set).

The register format is shown in Figure 20B and the information is described in Table 27B.

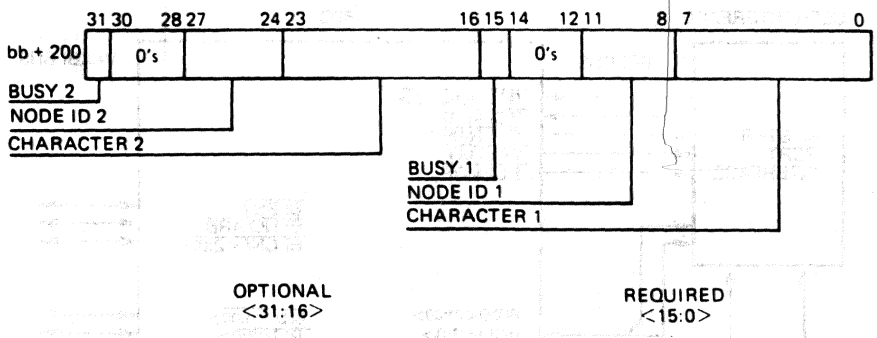


Figure 20B • VAXBI 78732 Receive Console Data Register Format

Table 27B • VAXBI 78732 Receive Console Data Register Description

Bit	Type	Description
31	R/W	Busy 2—Set to indicate that the character 2 (CHAR 2) field (bits 23:15) has not been read by the remote node. This bit must be cleared before CHAR 2 is available for another character.
30:28	RO	Reserved and cleared to zeros.
27:24	R/W	NODE ID 2 (Node identification 2)—Contains the identification of the local node that sent information to the Character 2 field.
23:16	R/W	CHAR 2 (Character 2)—Contains the console command character or console message sent from the local to the remote node.
15	R/W	Busy 1—Set to indicate that the character 1 (CHAR 1) field (bits 07:00) has not been read by the remote node. This bit must be cleared before CHAR 1 is available for another character.
14:12	RO	Reserved and cleared to zeros.
11:08	R/W	NODE ID 1 (Node identification 2)—Contains the identification of the local node that sent information to the Character 1 field.
07:00	R/W	CHAR 1 (Character 1)—Contains the console command character or console message sent from the local to the remote node.

• VAXBI Interface Node

Figure 21 is a diagram of a VAXBI node that shows the connecting lines and signals between the BIIC and master-port and slave-port user's interfaces, and between the BIIC and VAXBI bus. The BIIC decodes and matches the addresses from the VAXBI bus and performs all arbitration functions between the VAXBI bus and the BIIC. It includes the primary receiver and protocol logic required to interface to the VAXBI bus and all VAXBI bus transceivers associated with the data transfers and most of the bus receivers. The master-port and slave-port interfaces communicate with the BIIC through the synchronous interface BCI bus. The BCI bus consists of 64 lines that transfer data, address, and control information to and from the BIIC. The user's interface can request transfers through the BCI bus under control of the BIIC.

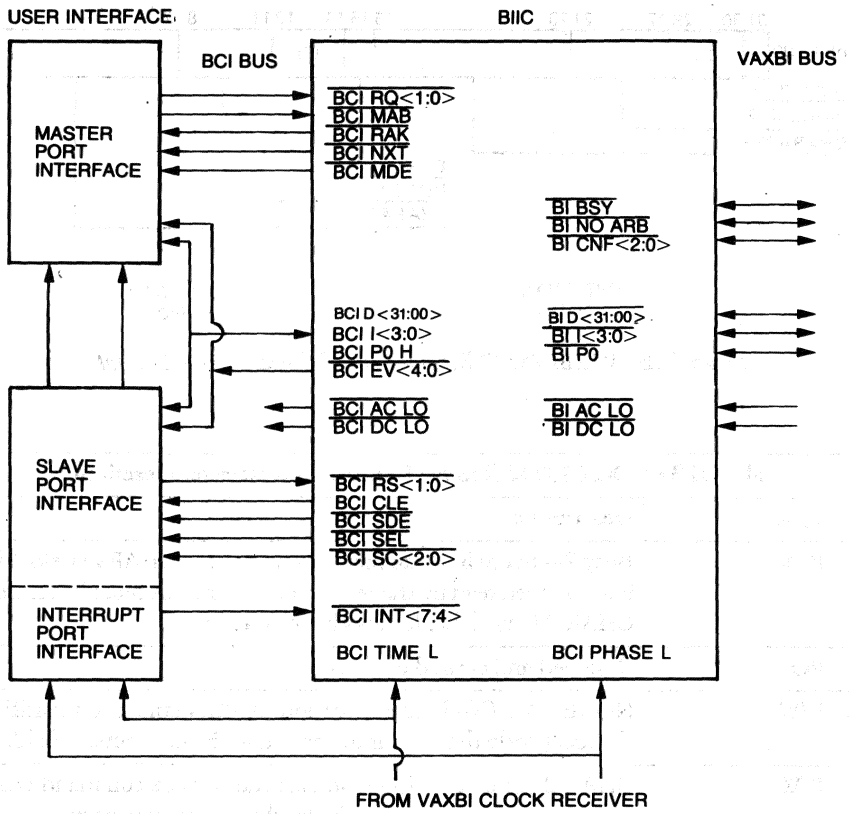


Figure 21 • VAXBI 78732 Interface Node Connections

The BIIC can detect and transfers commands from the VAXBI bus to the master-port or slave-port interface. In a multiprocessor environment, the addresses transmitted on the VAXBI bus are available to the user's interface for monitoring cache invalidate transactions. The BIIC supports the transfer of information between the master and slave within a single node.

Data and address information is transferred between the user's interface and the BIIC through the time multiplexed three-state BCI D<31:00> lines. Commands are transferred through the BCI I<3:0> lines. The direction control is provided by the BIIC. The master-port interface initiates command sequences by transferring a code on the request $\overline{\text{BCI RQ<1:0>}}$ lines and the BIIC responds by asserting the appropriate enable signal that transfers the command information to BCI bus. Command and data confirmation is transferred from the user's interface to the BIIC through the response lines. Transaction status is transferred to the user's interface through the event $\overline{\text{BCI EV<4:0>}}$ lines. Several registers in the BIIC control the operation of interrupt requests. Interrupts can be generated by asserting one of the $\overline{\text{BCI INT<7:4>}}$ lines or by writing information into one of the interrupt control registers. The BIIC provides vector information from an internal register in response to an identification command or it can solicit vector information from a user's interface.

Master-port Transactions

The master-port BCI signals are used to generate internode and intranode transactions. Transactions directed to other nodes are internode transactions and are limited to longwords. Intranode transactions can be VAXBI transactions that are issued by the master port through the VAXBI bus or loopback transactions that are not transferred through the VAXBI bus. These transactions can transfer longwords, octawords, or quadwords.

The master-port interface requests a transaction by transferring a code on the $\overline{\text{BCI RQ}<1:0>}$ lines. The BIIC asserts the BCI MDE line to inform the user's interface that the command/address information is required on lines BCI I<3:0> and BCI D<31:00>. In subsequent cycles the BIIC asserts the $\overline{\text{BCI NXT}}$ and $\overline{\text{BCI MDE}}$ lines to request the data from the user's interface.

Slave-port Transactions

The slave-port interface responds to read and write requests to memory locations in this node but not to transactions that access the BIIC registers. It also receives commands directed to more than one responder such as interrupt commands. The slave port will respond to all transactions directed to it including more than one transaction received sequentially. It can operate at the sustained peak bandwidth of the VAXBI bus if the user's interface is capable of this transfer rate.

Nodes that generate interrupts can use the interrupt port of the slave to request the transfer of an interrupt and to respond with an external vector to an identification command.

• VAXBI Address Space

The VAXBI address space is grouped into memory address space locations and input/output (I/O) address space locations. During the first cycle of a read-type, write-type, and invalid transaction, a 30-bit physical address A29:00 is transferred on the $\overline{\text{BI D}<29:00>}$ lines. Lines $\overline{\text{BI D}<31:30>}$ specify the length of the transfer in longwords. When bit A29 is zero, the 512-Mbyte memory space from 0000 0000 to 1FFF FFFF (hexadecimal) is accessed. When address bit A29 is a one, the 512-Mbyte I/O address space from 2000 0000 to 3FFF FFFF (hexadecimal) is selected. The address space allocations are shown in Figure 22.

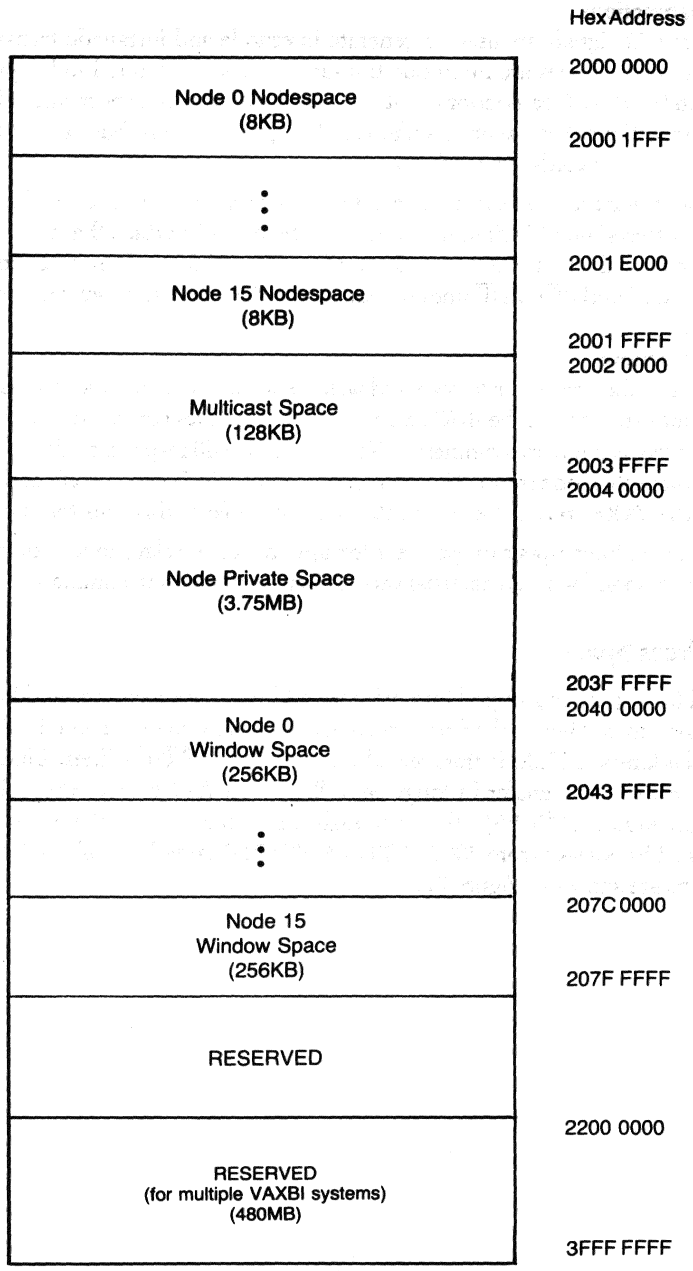


Figure 22 ▪ VAXBI 78732 Input/Output Address Space Allocations.

The VAXBI architecture defines the use of the I/O address space that contains node space, multicast space, window space, and reserved space. Up to 16 VAXBI buses can be accessed. Address bits A28:25 define the mapping mechanism used to access these buses. If the I/O space is selected and an address bits A28:25 is ones, a reserved location of 2200 0000 through 3FFF FFFFF (hexadecimal) is selected.

The I/O spaces are selected by the address configurations shown in Figure 23. Two blocks of I/O space are partitioned according to the identification of the node. The node space is positioned at the low end of the I/O address space and consists of 16 address blocks each containing 8 Kbytes. One nodespace is assigned to each node that can be implemented on the VAXBI bus. The first 256 bytes of each space consists of BIIC control and status register space and the remaining space is assigned to the user's interface control and status information. The window space starts at address 2040 0000 (hexadecimal) and contains 16 blocks of 256 Kbytes each and can be used by adapters to map VAXBI transactions onto the selected bus.

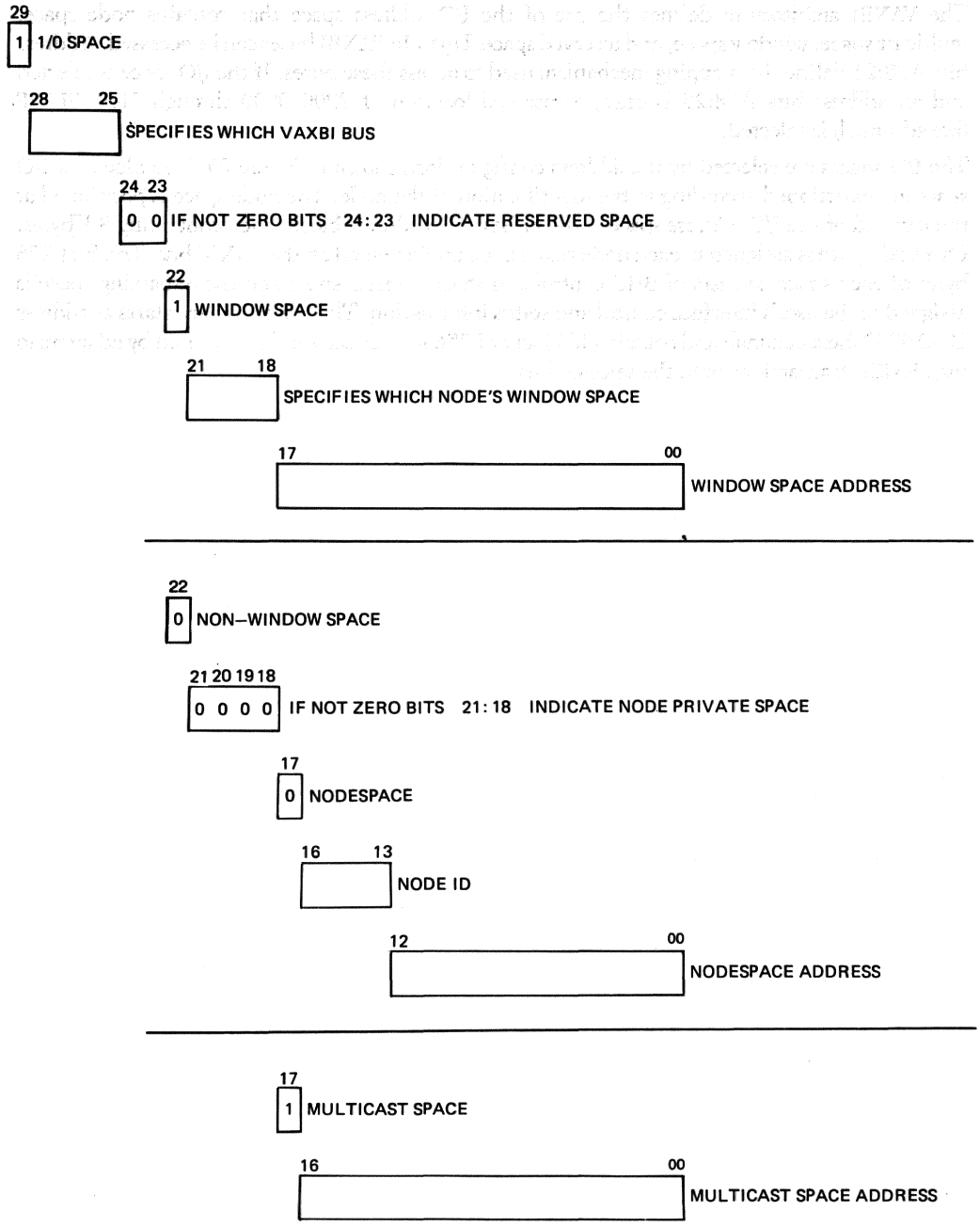


Figure 23 • VAXBI 78732 I/O Space Addressing

Node Space Assignments

Each node that interfaces to the VAXBI bus is assigned an identification number (node ID) of from 0 to 15. The ID code determines the bus and interrupt priority level assignments and the locations of the node's registers. The ID is selected by jumper lead connections on the VAXBI backplane.

Figure 24 shows the node space allocations. The assignment of the 8-Kbyte node space depends on the type of node as defined by this identification. The starting address for a node is 2000 0000 (hexadecimal) plus 8K times the node ID. The base address is referred to as "bb."

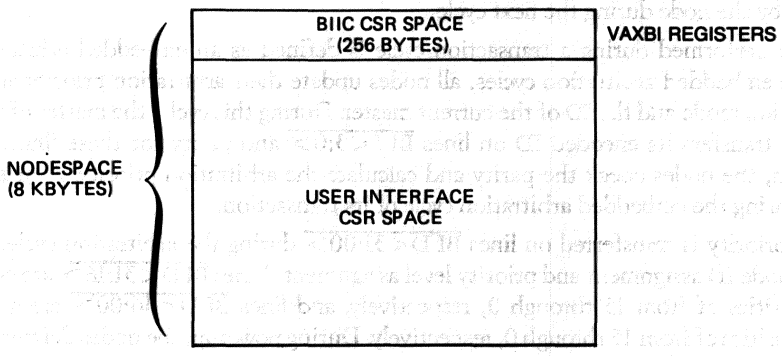


Figure 24 ▪ VAXBI 78732 Node Space Allocation

The first 256 bytes of a node space is reserved for the VAXBI registers and the remaining space is assigned to user's interface control and status registers (CSRs). The CSR space (bb+100) contains slave-only status and is used by memory nodes that do not implement the Broke bit of the VAXBI control and status register. Location bb+200 is reserved for the receive console data register (RXCD) and is implemented by nodes capable of performing transactions with the console terminal. Nodes that do not have the console capability respond to read commands with a no acknowledge or with a longword in which the RXCD busy (bit 15) is set.

Because the BIIC has one starting and one ending address register, a node cannot respond with a window space and region of memory space. Responses to multicast space and user CSR space can be disabled by the BCI control and status register.

The BIIC can be configured to respond to any combination of the following:

- The node space of the node
- The space defined by the starting address and ending address of the node
- The multicast space

• VAXBI Protocol and Cycle Types

The VAXBI nodes contain arbitration logic. Each node provides two arbitration levels. To become bus master, a node asserts one of the 32 data lines during an the arbitration transaction and monitors the remaining data lines to determine if a lower number line has been asserted. If no line is asserted, the node becomes bus master and transfers command/address information immediately or when the current bus transaction if has been completed.

The $\overline{\text{BI NO ARB}}$ line controls access to the bus data path for arbitration. Arbitration can occur during a cycle or after a cycle following the deassertion of this line. Arbitration cycles can occur during a transaction or after a transaction. The command/address information from the bus master is decoded by the node during the next cycle.

Arbitration performed during a transaction cycle is defined as an embedded arbitration cycle. During the embedded arbitration cycles, all nodes update their arbitration priority according to the arbitration mode and the ID of the current master. During this cycle, the master of the current transaction transfers its encoded ID on lines $\overline{\text{BI I<3:0>}}$ and parity for these lines. From this information, the nodes check the parity and calculate the arbitration priority. A master cannot arbitrate during the embedded arbitration cycle of its transaction.

The node priority is transferred on lines $\overline{\text{BI D<31:00>}}$ during the arbitration cycle. Figure 25 shows the node ID assignment and priority level assignment. Lines $\overline{\text{BI D<31:16>}}$ are assigned the lowest priorities of from 15 through 0, respectively, and lines $\overline{\text{BI D<16:00>}}$ are assigned the highest priorities of from 15 through 0, respectively. During powerup, the nodes default to the low-priority word.

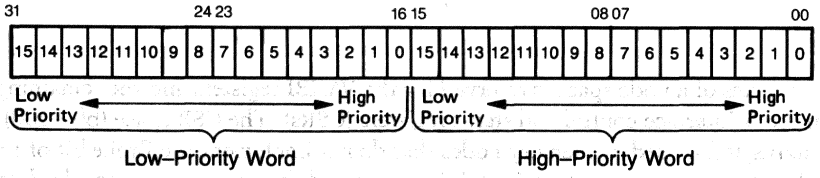


Figure 25 • VAXBI 78732 Node Identification and Priority Assignments

Arbitration Modes

The arbitration modes, defined by the VAXBI protocol, are dual round robin, fixed-high priority, and fixed-low priority. The dual round robin mode is the only user authorized mode. The remaining modes are reserved for use by Digital. The modes are selected by arbitration control (bits 5 and 4) of the VAXBI control and status register and can be changed by a node during system operation. Any combination of arbitration modes can exist on the VAXBI bus, however; the fixed-high and fixed-low modes are reserved for use by Digital. All nodes default to the dual round-robin mode during the powerup sequence.

Dual Round Robin Mode—During this mode, the node arbitrates on the low-priority word when the ID of this node is less than or equal to the ID of the previous bus master. When the ID of the node is greater than the ID of the previous bus master, it arbitrates on the higher priority word. If this mode is selected by all nodes on the VAXBI bus, all nodes will have equal access to the bus after a period of time. In multiprocessor configurations, this mode prevents excessive bus latency time that may occur when a node is denied bus access because several processors are performing instruction loops.

Fixed Low-priority Mode—This reserved mode can be assigned to nodes that require access to the bus infrequently.

Fixed High-priority Mode—This reserved mode can be assigned to nodes where critical access time to the bus is required.

Transaction Cycles

The VAXBI transactions are performed using the command/address, embedded arbitration, and data cycles shown in Figure 26. The basis operation is controlled by the $\overline{\text{BI NO ARB}}$ and $\overline{\text{BI BSY}}$ lines.

COMMAND/ ADDRESS (C/A) CYCLE	IMBEDDED ARBITRATION (IA) CYCLE	DATA CYCLE	DATA CYCLE	

Figure 26 • VAXBI 78732 Transaction Cycle Format

Command/Address Cycle—This is the first cycle of all VAXBI transactions and is identified by a node when the $\overline{\text{BI BSY}}$ signal is asserted following a cycle where the $\overline{\text{BI NO ARB}}$ signal was asserted. During this cycle, the master transmits a 4-bit command code in lines $\overline{\text{BI I<3:0>}}$ and the information to select the appropriate slave on lines $\overline{\text{BI D<31:00>}}$. Each transaction type is restricted to one of the formats listed in Table 28.

Table 28 • VAXBI 78732 Command/Address Transaction Type

Transaction	VAXBI Bus Lines BI D<31:16>	BI D<15:00>
Read-type	Length code and 30-bit address	
Write-type	Length code and 30-bit address	
Invalid	Length code and 30-bit address	
Interprocessor interrupt	Decoded master ID	Destination mask
Interrupt	Level	Destination mask
Stop	Reserved	Destination mask
Broadcast	Reserved	Destination mask
Identification	Level	Reserved

During read-type, write-type, and invalid transactions, the selection information consists of a length code and 30-bit address. The selection information can also be a 16-bit destination mask in which each bit corresponds to a node ID. This enables from 1 to 16 nodes to be involved in the same transaction. The destination mask is used for all multiresponder transactions except for the invalid transaction. The interprocessor interrupt transaction uses the decoded mask and destination mask to select a slave. The level field selects the slave during a identification transaction.

Embedded Arbitration Cycle—During the second cycle of a transaction, the encoded ID of the master is transferred on lines $\overline{\text{BI I<3:0>}}$ and the VAXBI data path is available for arbitration of the other nodes unless the burst mode is selected.

Data Cycles—One or more data cycles can follow the imbedded arbitration cycle. During these cycles, data is transferred between the master and slave node through the VAXBI data path. The number of data cycles required normally depends on the length of the transfers and the number of stall responses issued by the slave. During identification transactions, the number of data cycles depends only on the number of stall responses. Multiresponder transactions, except for broadcast, use one reserved transaction. During broadcast transactions, data cycles cannot be stalled. Therefore, the number of data cycles depends on the length of the transfer. Table 29 lists VAXBI bus information transferred during the data cycle for each transaction type.

Table 29 • VAXBI 78732 Data Cycle Information Transfer

Transaction	VAXBI Bus Lines BI D<31:00>	BI I<3:0>
Read-type	read data	read status
Write	write data	reserved
Write with cache intent	write data	reserved
Write mask with cache intent	write data	write mask
Unlock write with cache intent	write data	write mask
Invalid	reserved	reserved
Interprocessor interrupt	reserved	reserved
Invalid	reserved	reserved
Stop	reserved	reserved
Broadcast	reserved	reserved
Identification	interrupt vector	vector status

• VAXBI Bus Transactions

This section describes the types of transactions that are supported by the VAXBI bus and defines their use.

Single and Multiresponder Transactions

Single-responder transactions are directed to one node and multiresponder transactions are directed to more than one responder. Table 6 lists the commands that can be used with these transactions.

During single-responder transactions, data is transferred between a master and a slave node. The master requests that a node be a slave with a 30-bit address. The node receiving the address uses this and other information transmitted during the command/address cycle to determine its slave status.

Multiresponder transactions are initiated by the INTR, IPINTR, INVAL, STOP, and BDCST commands from a master. During these transactions, the master sends a destination mask instead of an address. Interrupts are generated by a master when it issues a INTR command message to one or more slaves capable of accepting interrupt requests or when it issues an IPINTR command to other processors. The INVAL command is used to notify a node that the cache data in its memory is invalid. The STOP command is used to diagnose errors. The BDCST command is reserved by Digital and is used to send information through the entire system.

Interlock Transactions

The IRCI and UWMCI interlock commands and the IPINTR interprocessor commands support interprocessor communications. These commands allow processors to communicate by exchanging messages that are deposited in a shared memory. The shared memory access is synchronized because the memory access from one processer may be interspersed with memory accesses from another processor. Software-level synchronization is achieved through the use of the VAX interlock and queue instructions, and implemented by the IRCI and UWMCI transactions.

During IPINTR transactions, one processor interrupts the operation of another processor. Both shared memory and interprocessor interrupts can also be used. One processor can deposit a message in a specific location of shared memory and then issue a IPINTR command to notify the other processor.

Indivisible operations are performed on processor nodes and adapters by the IRCI and UWMCI interlock transactions. The interlock feature of these transactions must be implemented for memory and I/O space addresses. IRCI transactions that lock a block of addresses must be followed as soon as possible by a UWMCI transaction to unlock the block. If a VAXBI node issues an IRCI command to a locked location, the node will receive a retry response. A retry timeout will occur from repeated IRCI commands to a locked location. Refer to the *VAXBI System Reference Manual* for other interlock considerations.

Data Transfer Transactions

During the command/address cycle of read-type and write-type transactions, lines $\overline{\text{BI D<31:30>}}$ specify the number of bytes to be transferred and lines $\overline{\text{BI D<29:00>}}$ specify the address. Table 30 lists the data length selections.

Table 30 • VAXBI 78732 Data Length Codes

BI D Lines 31	30	Data Length	Bytes
H	H	Reserved	
H	L	LW (longword)	4
L	H	QW (quadword)	8
L	L	OW (octaword)	16

The low address of the block of data transferred is a multiple of the size of the block of data in bytes. During read-type transactions, the address supplied during the command/address cycle may not contain the low address of the block of data transferred.

Address Interpretation—The interpretation of the address during read-type and write-type transactions depends on the transaction type, address space, data-length field, and low-order address bits. Figure 27 shows the longword and byte references in an octaword block.

	31			00
D1 =	B3	B2	B1	B0
D2 =	B7	B6	B5	B4
D3 =	B11	B10	B9	B8
D4 =	B15	B14	B13	B12

Figure 27 • VAXBI 78732 Longword and Byte References in an Octaword Block

Address B0 specifies the data length as follows: A<29:02>'00=longword, A<29:03>'000=quadword, A<29:04>'000=octaword.

Read-type Transactions—Table 31 lists the interpretation values of the addresses during read-type transactions. During this transaction, the slave first transfers the addressed longword of data and the remaining longwords depend on the implementation of the node. The address normally is data-length aligned and the remaining longwords are transferred in ascending address order. If the initially addressed longword is not data-length aligned, the remaining longwords are transmitted in ascending order until the beginning of the data-length aligned block is reached. A wrap will then occur and the next longword transferred will be located at the base address of the block. Longwords are then transferred in ascending order until the entire block has been transferred.

Table 31 • VAXBI 78732 Read-type Transaction Address Interpretation

Data length[1]	Address space[2]	Address transmitted	Address received	Return data order
OW	NWS	A29:04'00XX	A29:04'00—	D1, D2, D3, and D4
OW	NWS	A29:04'01XX	A29:04'01—	D2, D3, D4, and D1
OW	NWS	A29:04'10XX	A29:04'10—	D3, D4, D1, and D2
OW	WS	A29:04'11XX	A29:04'11—	D4, D1, D2, and D3
QW	NWS	A29:03'0XX	A29:03'0—	D1 and D2
QW	NWS	A29:03'1XX	A29:03'1—	D2 and D1
QW	WS	Not used[3]		
LW	NWS	A29:02'XX	A29:02'—	D1 (B3, B2, B1, and B0)
LW	WS/L	A29:02'XX	A29:02'—	D1 (B3, B2, B4, and B5)
LW	WS/W	A29:02'0X	A29:02'0—	D1 (XX, XX, B1, and B0)
LW	WS/W	A29:02'1X	A29:02'1—	D1 (B3, B2, XX, and XX)
LW	WS/B	A29:02'00	A29:02'00	D1 (XX, XX, XX, and B0)
LW	WS/B	A29:02'01	A29:02'01	D1 (XX, XX, B1, and XX)
LW	WS/B	A29:02'10	A29:02'10	D1 (XX, B2, XX, and XX)
LW	WS/B	A29:02'11	A29:02'11	D1 (B3, XX, XX, and XX)

[1]Refer to Table 30 for data length abbreviations.
[2]NWS = nonwindow space, WS = window space, B = byte accessible, W = word accessible, L = longword accessible, X = any data, — = ignored by slave, ' = concatenation
[3]Slave must respond with a no acknowledge.

Write-type Transactions—Table 32 lists the interpretation values of the addresses during write-type transactions. During VAXBI write transactions, data is transferred in ascending order.

Table 32 • VAXBI 78732 Write-type Transaction Address Interpretation

Data length[1]	Address space[2]	Address transmitted	Address received	Return data order
OW	NWS	A29:04'00XX	A29:04'00—	D1, D2, D3, and D4
OW	WS	not used[3]		
QW	NWS	A29:03'0XX	A29:03'0—	D1 and D2
QW	WS	not used[3]		
LW	NWS	A29:02'XX	A29:02'—	D1
LW	WS/L	A29:02'XX	A29:02'—	D1 (B3, B2, B1, and B0)
LW	WS/W	A29:02'0X	A29:02'0—	D1 (—, —, B1, and B0)
LW	WS/W	A29:02'1X	A29:02'1—	D1 (B3, B2, —, and —)
LW	WS/B	A29:02'00	A29:02'00	D1 (—, —, —, and B0)
LW	WS/B	A29:02'01	A29:02'01	D1 (—, —, B1, and —)
LW	WS/B	A29:02'10	A29:02'10	D1 (—, B2, —, and —)
LW	WS/B	A29:02'11	A29:02'11	D1 (B3, —, —, and —)

[1]Refer to Table 30 for data length abbreviations.
[2]NWS = nonwindow space, WS = window space, B = byte accessible, W = word accessible, L = longword accessible, X = any data, — = ignored by slave, ' = concatenation
[3]Slave must respond with a no acknowledge.

Memory Cache Data—VAXBI nodes that contain data caches must monitor the VAXBI write-type transactions. If a cache location is accessed, the data must be marked as invalid. If the node cannot mark the data as invalid before the monitoring transaction is complete, the monitoring transaction must be extended. Refer to the *VAXBI Systems Reference Manual* for detailed information on data caches.

Write Mask—During WMCI and UWMCI data cycles, the write mask is transferred on the $\overline{\text{BI I}<3:0>}$ lines. When a mask bit is set, the corresponding byte is modified by the information on the data lines. These lines are not defined for write-type data cycles that do not use the mask. Table 33 shows the byte assignment for the write mask codes.

Table 33 • VAXBI 78732 Write-mask Code Assignments

Asserted BI I line	Byte BI D line
3	31:24
2	23:16
1	15:08
0	7:00

Read Status—The $\overline{\text{BI I<3:0>}}$ lines transfer a read-status code from the slave to the master during the acknowledge data cycle of a read-type or identification transaction. The code defines the type of data returned to the master. Table 34 shows the read status code assignments.

Table 34 • VAXBI 78732 Read Status Code Assignments

BI I line 3	2	1	0	Status
H	*	H	H	Reserved
H	*	H	L	Read data
H	*	L	H	Corrected read data
H	*	L	L	Read data substitute
L	*	H	H	Reserved
L	*	H	L	Read data/do not cache
L	*	L	H	Corrected read data/do not cache
L	*	L	L	Read data substitute/do not cache

*Reserved

Write-type Transactions

The VAXBI bus supports four write-type transactions that are used to transfer data from a master node to a slave node. The following paragraphs describe the sequence used during the transactions. The abbreviations referenced on the write transaction timing diagrams are:

M = master node, S = slave node, Ss = more than one slave, AAN = all arbitrating nodes, AN = all nodes, APS = all potential slaves for identify transactions prior to identification arbitration selection. A (>) before a response indicates the CNF code transferred during the transaction.

WRITE Command—The Write transaction transfers data from a master to a slave when the master does not store the data in cache memory. Figure 28 shows the transaction timing of a WRITE and Write with Cache Intent (WCI) command for an octaword.

CYCLE	C/A	IA	D1	D2	D3	D4		
31	DATA LENGTH							
30								
29								
28								
27								
26								
25		DECODED ID LOW PRIORITY						
24								
23								
22								
21								
20								
19								
18								
17								
16	30 BIT ADDR							
BI D<31:00> 15			DATA 1	DATA 2	DATA 3	DATA 4		
14								
13								
12								
11								
10								
9		DECODED ID HIGH PRIORITY						
8								
7								
6								
5								
4								
3								
2								
1								
0								
SOURCE	M	AAN	M	M	M	M		
BI I<3:0>	COMMAND	MASTER ID	WRITE MASK	WRITE MASK	WRITE MASK	WRITE MASK		
SOURCE	M	M	M	M	M	M		
BI P0 GEN	M	M	M	M	M	M		
CHK	AN	AN	S	S	S	S		
BI CNF<2:0>			>ACK NO ACK STALL RETRY	>ACK NO ACK STALL	>ACK NO ACK STALL	>ACK NO ACK STALL	>ACK NO ACK	>ACK NO ACK
SOURCE			S	S	S	S	S	S
BI BSY	M	M	M,S	M,S	M,S			
BI NO ARB		M,AAN	M,S	M,S	M,S			

Figure 28 ▪ VAXBI 78732 WRITE and WCI (octaword) Transaction Timing

During the command/address cycle, the master specifies the length of the data on lines $\overline{BI\ D<31:30>}$, the address on lines $\overline{BI\ D<29:00>}$, and the command on lines $\overline{BI\ I<3:0>}$. Parity is generated by the master and checked by the nodes. The $\overline{BI\ BSY}$ line is asserted until the last acknowledge data cycle. During the command/address cycle, the $\overline{BI\ NO\ ARB}$ line is deasserted and then asserted with the $\overline{BI\ BSY}$ line until the end of the cycle. During the embedded arbitration cycle, all nodes except for the present master can arbitrate for the bus control.

The slave transfers a command confirmation code to the master during cycle D1. This code indicates the slave status and errors conditions. Subsequent confirmation codes provide information related to the data transfers.

The master sends data to be written during D1 and succeeding data cycles. Slaves that are unable to receive the data at the specified time can issue a stall response for a maximum of 127 cycles to delay the data transfer until it is ready. During the data cycles of WRITE command, the information on the $\overline{BI\ I<3:0>}$ lines is undefined. During the data cycles, the master generates the parity that is checked by the slave.

Write with Cache Intent—During the Write with Cache Intent (WCI) transaction shown in Figure 28, the data transferred may be written into cache memory. This can occur only if the data previously written into the same location in the cache is valid. The slave node must issue an INVAL command for subsequent write transactions to the same locations that are not performed with a WCI VAXBI transaction. The WCI transaction is always performed if the node is unable to determine if the data transferred will be written into cache. The response to this command by the slave is the same as a WRITE command. During the data cycles of WCI command, the information on the $\overline{BI\ I<3:0>}$ lines is undefined.

Write Mask with Cache Intent—The Write Mask with Cache Intent (WMCI) command is similar to the WCI command except that the bytes of the address locations to be modified are selected by the master. The write mask is transferred on the $\overline{BI\ I<3:0>}$ lines during each data cycle. The master generates parity for the entire VAXBI data path regardless of the bytes to be modified. The WMCI transaction timing is shown in Figure 29.

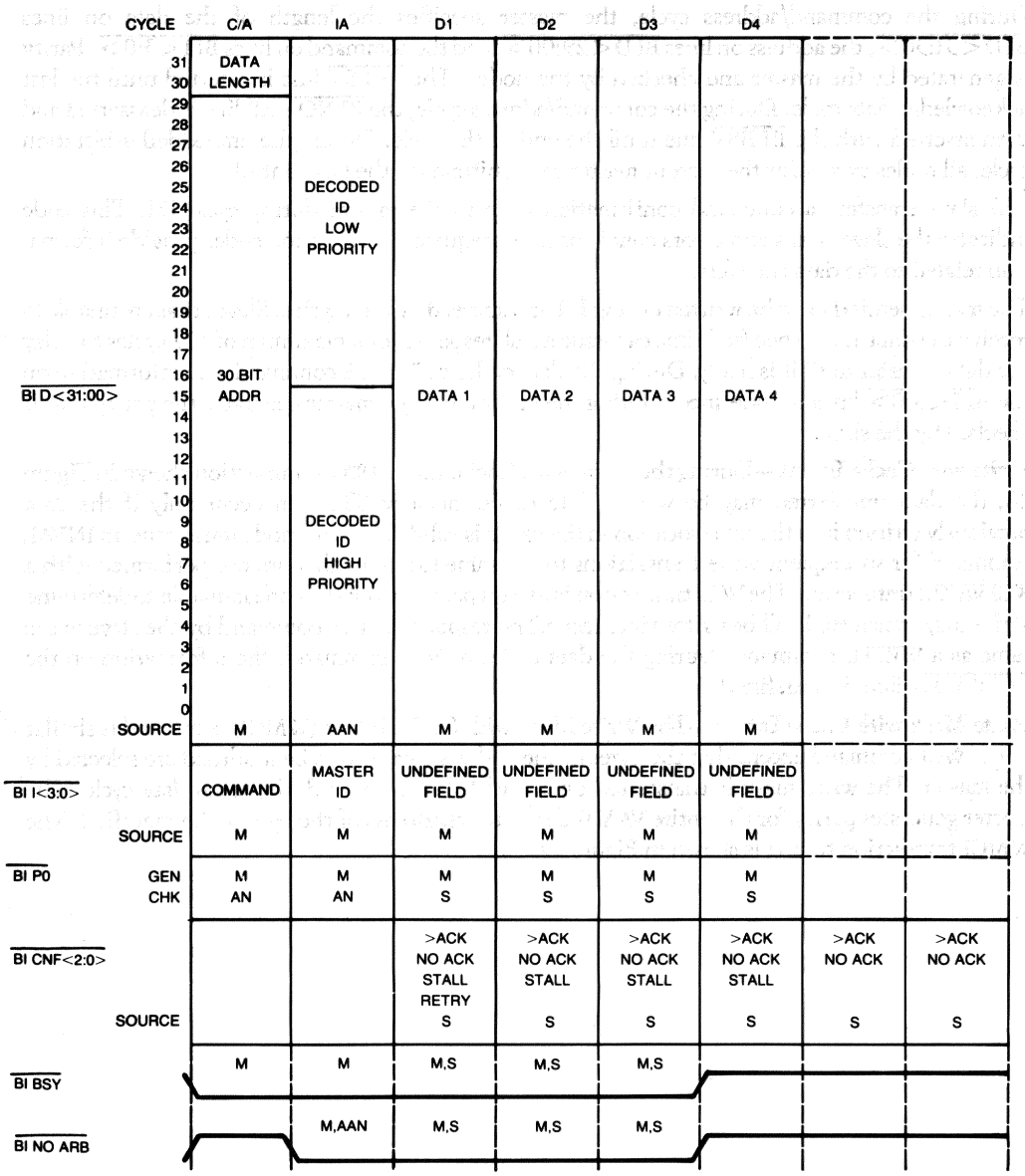


Figure 29 ▪ VAXBI 78732 WMCI and UWMCI (octaword) Transaction Timing

Unlock Write Mask with Cache Intent—The Unlock Write Mask with Cache Intent (UWMCI) command is used to complete a read-modify-write operation that began with an interlock read with cache intent (IRCI) command. It is used to unlock a shared memory structure. The slave should not clear the lock bit if a parity error occurs duing this transaction. A node must issue this transaction as soon as possible after issuing an IRCI command. A write mask is transferred on lines $\overline{BII<3:0>}$ during each data cycle. The UWMCI transaction timing is shown in Figure 29.

Read-type Transactions

The VAXBI bus supports three read-type transactions that are used to transfer data to a master node from a slave node. The following paragraphs describe the sequence used during the transactions. The abbreviations referenced on the write transaction timing diagrams are

M = master node, S = slave node, Ss = more than one slave, AAN = all arbitrating nodes, AN = all nodes, APS = all potential slaves for identify transactions prior to identification arbitration selection. A (>) before a response indicates the CNF code transferred during the transaction.

READ Command—The read transactions transfer data from slave to master when the data will not be stored in cache memory. Figure 30 shows the transaction timing of a READ command that transfers an octaword.

	CYCLE	C/A	IA	D1	D2	D3	D4		
	31	DATA LENGTH							
	30								
	29								
	28								
	27								
	26								
	25		DECODED ID LOW PRIORITY						
	24								
	23								
	22								
	21								
	20								
	19								
	18								
	17								
	16	30 BIT ADDR							
BI D<31:00>	15			DATA 1	DATA 2	DATA 3	DATA 4		
	14								
	13								
	12								
	11								
	10								
	9		DECODED ID HIGH PRIORITY						
	8								
	7								
	6								
	5								
	4								
	3								
	2								
	1								
	0								
	SOURCE	M	AAN	S	S	S	S		
BI I<3:0>		COMMAND	MASTER ID	STATUS	STATUS	STATUS	STATUS		
	SOURCE	M	M	S	S	S	S		
BI P0	GEN	M	M	S	S	S	S		
	CHK	AN	AN	M	M	M	M		
BI CNF<2:0>				>ACK NO ACK STALL RETRY	>ACK NO ACK STALL	>ACK NO ACK STALL	>ACK NO ACK STALL	>ACK NO ACK	>ACK NO ACK
	SOURCE			S	S	S	S	M	M
BI BSY		M	M	M,S	M,S	M,S			
BI NO ARB			M,AAN	M,S	M,S	M,S			

Figure 30 ▪ VAXBI 78732 READ, RCI, and IRCI (octaword) Transaction Timing

During the command/address cycle, the master specifies the length of the data word on lines BI D<31:30>, the address on lines BI D<29:00>, and the command on lines BI I<3:0>. Parity is generated by the master and checked by the nodes. The BI BSY line is asserted until the last acknowledge data cycle. During the command/address cycle the BI NO ARB line is deasserted and then asserted with the BI BSY line until the end of the cycle. During the embedded arbitration cycle, all nodes except for the present master can arbitrate for the bus control for the next transaction.

The slave transfers a command confirmation code to the master during cycle D1. This code indicates the status of the slave and any errors conditions that may exist. Subsequent confirmation codes provide information related to the data transfers. A confirmation code providing information related to the last two data cycles is transferred from the master to the slave.

The slave sends data to be written during D1 and succeeding data cycles. Slaves that are unable to send data at the specified time can issue a stall response for a maximum of 127 cycles to delay the data transfer until it is ready. During all data cycles of *Read* command, the parity generated by the slave is checked by the master and the read status from the slave on lines BI I<3:0> provides the master with status information.

Read with Cache Intent—The Read with Cache Intent (RCI) transaction , shown in Figure 30, is used to read data that is intended to be stored in cache. If a "do not cache" read status is transferred on lines BI I<3:0> from the slave, the master must not store the data in cache memory. The response from the slave for this command is the same as for the *Read* command. This command is used for cached multiprocessor systems to inform the slave that the data read will be stored in the cache memory of the master.

Interlock Read with Cache Intent—The Interlock Read Data with Cache Intent (IRCI) transaction supports read-modify-write operations and is used with the UWMCI command. The transaction timing shown in Figure 30 is the same as for the read transaction. When the memory space of a node has been successfully accessed by this command, the node must set a lock bit that will cause susequent IRCI transactions directed to the same locked address to be repeated. The lock bit must remain set until a UWMCI transaction directed to the same locked address range is successful. The minimum size of an address range controlled by a single lock bit in the window range is a byte, and beyond the window range, the minimum size of the address range is an octaword.

If the slave transfers a read data substitute status code, the IRCI transaction is unsuccessful and the lock bit should not be set and the master should not initiate the UWMCI transaction. If an IRCI or UWMCI command is received before the errors are detected in the previous IRCI command, the slave should issue a stall or retry confirmation until the state of the lock is determined.

A multiport memory with a lock bit set by any port will issue a retry response to an IRCI command. An IRCI command from the VAXBI bus to a UNIBUS adapter is interpreted as a data-in-pause (DATAIP) transaction to the UNIBUS and a DATAIP transaction from the UNIBUS to the VAXBI bus must be translated as an IRCI command to the VAXBI bus.

Invalidate Transaction

The invalidate (INVAL) command is used by processors and intelligent nodes during write operations to local memory to inform the nodes that their cached data may be invalid. Figure 31 shows the transaction timing of a INVAL command.

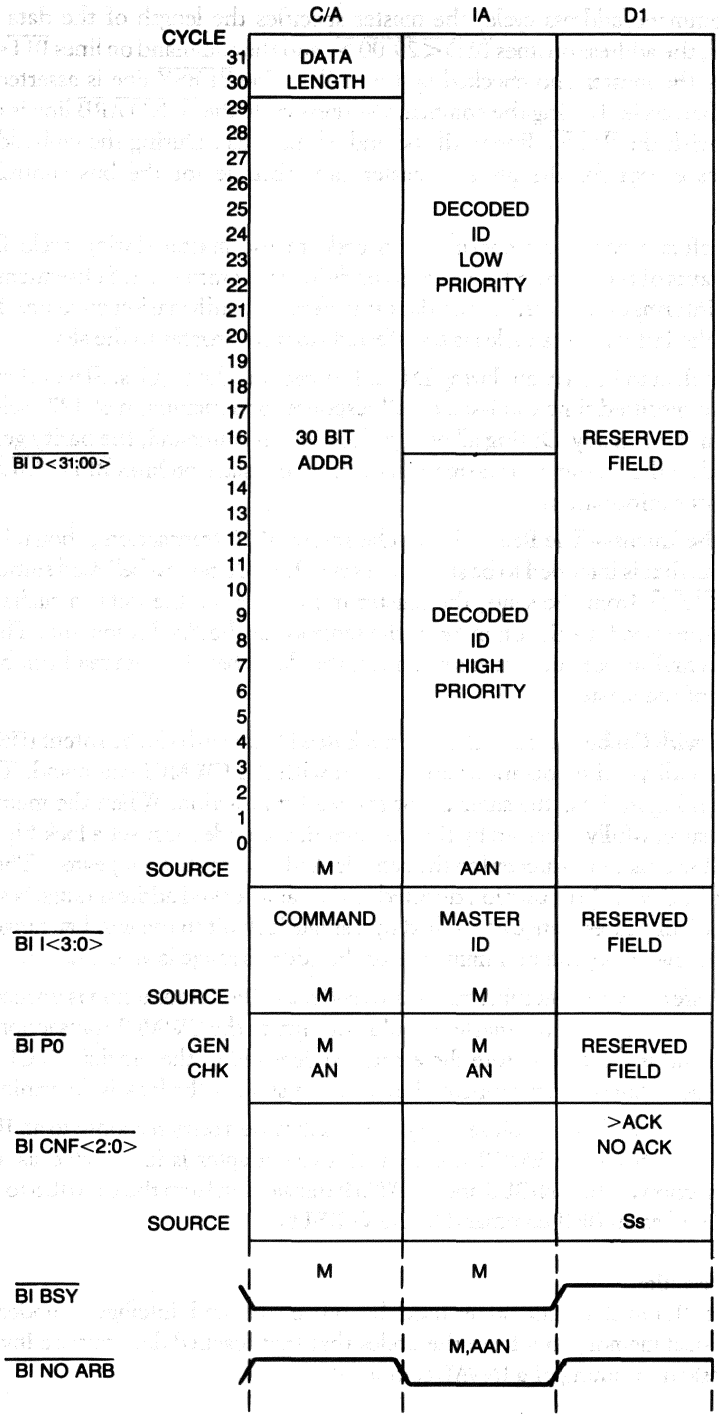


Figure 31 ▪ VAXBI 78732 INVAL Transaction Timing

During the command/address cycle, the master specifies the length of the data word on lines $\overline{\text{BI D<31:30>}}$, the address on lines $\overline{\text{BI D<29:00>}}$, and the command on lines $\overline{\text{BI I<3:0>}}$. The data length code specifies the number of consecutive addresses to be invalidated. The low-order address bits are reserved. Parity is generated by the master and checked by the nodes. Table 35 shows the address interpretation during this transaction.

Table 35 • VAXBI 78732 Invalid Transaction Address Interpretation

Data length	Address transmitted	Address received
Octaword	A28:04'0000	A29:04'----
Quadword	A28:03'000	A29:03'---
Longword	A28:02'00	A29:02'--

The nodes, except for the present master, can arbitrate for the bus control for the next transaction during the IA cycle. The acknowledge and no acknowledge are the only valid responses from the slaves to this command.

A node can delay the start of the next transaction to allow time to invalidate its cache by asserting the $\overline{\text{BI BSY}}$ signal through cycle D1 and until the data is invalidated.

• Interrupt Transactions

The VAXBI bus supports device interrupts consisting of INTR and INVAL transactions and interprocessor interrupt IPINTR transactions. During device interrupts, the interrupting device supplies an interrupt vector in response to the identify transaction that is unique to the device. During IPINTR transactions, the interrupt vector and level are the same for all interrupts and are stored in the receiving node.

Device Interrupts

Nodes that are capable of generating interrupt requests contain a vector that is used by the VAX processors to select one or more 512-byte locations in memory. These locations contain address pointers used to select interrupt service routines.

During the command/address cycles of the interrupt transaction, each $\overline{\text{BI D<19:16>}}$ line corresponds to an interrupt level. Line 19 is assigned the highest-priority interrupt (level 7) and line 16 the lowest priority interrupt (level 4). These levels correspond to the VAX processor interrupt priority levels (IPL17 through IPL14).

An interrupt node issues an identify transaction when it is ready to service an interrupt request. The interrupt level field from the node must contain only one level of the interrupt it is ready to service. This level must be the highest priority for which the bus master has received an interrupt request and has not responded with a identify transaction.

Nodes that have an interrupt pending at the Ident level respond by arbitrating for the bus during the identify arbitration cycle of the identify transaction. The winner of the arbitration transfers its interrupt vector during the next cycle. The VAXBI interrupt vector is different from the VAX system interrupt vector described in the *VAX-11 Architecture Reference Manual.*

Interrupt vector values of zero and vector values that are multiples of 200 (hexadecimal) are null interrupts that indicate no action is required to service the interrupt. If more than one bit is set in the destination field of a node, two or more processors will attempt to service this interrupt and each processor will issue an identify transaction. If only one node issues the interrupt, the first processor to issue the identify transaction will service the node. The remaining processors will issue identify transactions, however; an interrupt vector will not be returned because no contenders exist during the interrupt arbitration cycle. The null interrupt indicates to the processor that no nodes are waiting to be serviced.

Interrupt Command—The interrupt (INTR) command is used to initiate an interrupt request to one or more nodes on the bus. Figure 32 shows the transaction timing of a INTR command.

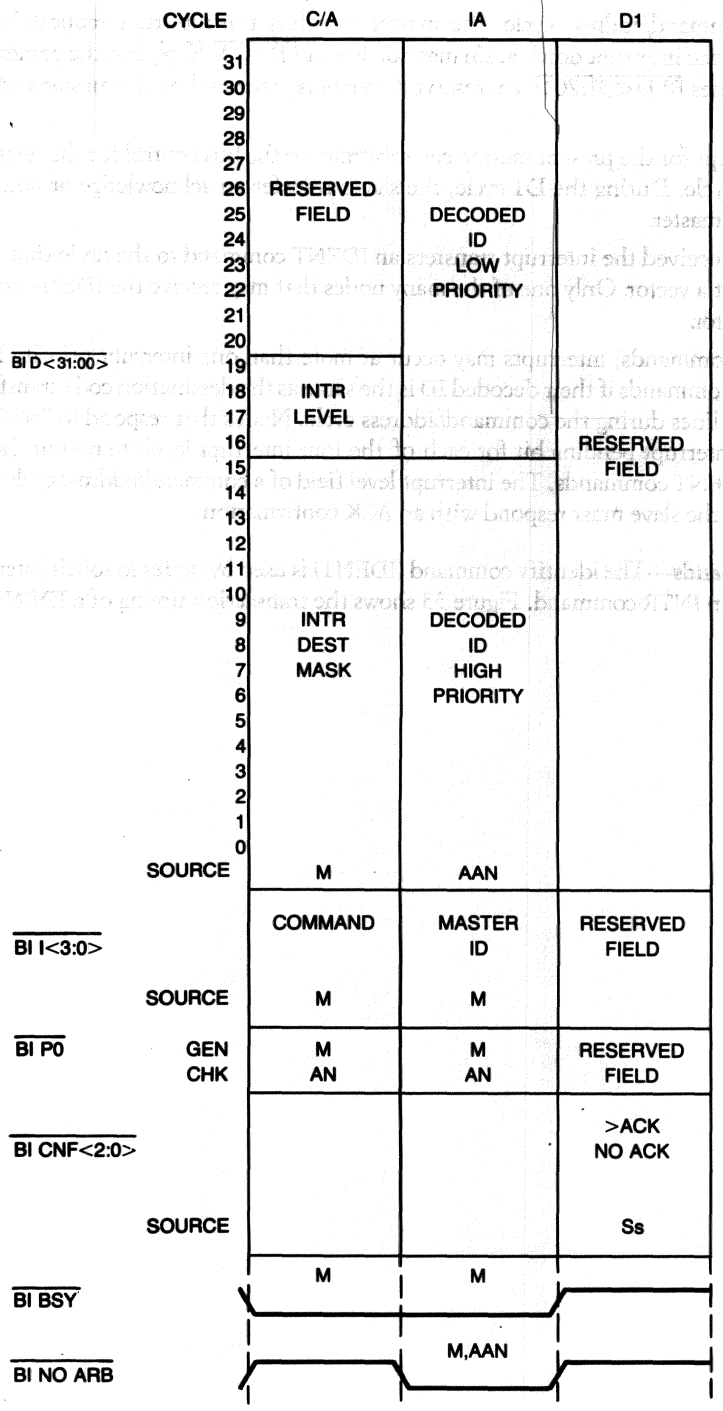


Figure 32 ▪ VAXBI 78732 INTR Transaction Timing

During the command/address cycle, the master transfers the interrupt request level on lines $\overline{\text{BI D<19:16>}}$, the interrupt destination mask on lines $\overline{\text{BI D<15:00>}}$, and the command on lines $\overline{\text{BI I<3:0>}}$. Lines $\overline{\text{BI D<31:20>}}$ are reserved. Parity is generated by the master and checked by the nodes.

The nodes, except for the present master, can arbitrate for the bus control for the next transaction during the IA cycle. During the D1 cycle, the slaves transfer an acknowledge or no acknowledge response to the master.

The node that received the interrupt transfers an IDENT command to the node that initiated the request to solicit a vector. Only one of the many nodes that may receive the IDENT command will transfer the vector.

During INTR commands, interrupts may occur at more than one interrupt priority level. Nodes respond to the commands if their decoded ID is the same as the destination code transferred on the $\overline{\text{BI D<15:00>}}$ lines during the command/address cycle. Nodes that respond to INTR commands must store an interrupt pending bit for each of the four interrupt levels to permit them to solicit vectors with IDENT commands. The interrupt level field of a command/address cycle may contain zeros, however, the slave must respond with an ACK confirmation.

Identify Commands—The identify command (IDENT) is used by nodes to solicit interrupt vectors in response to an INTR command. Figure 33 shows the transaction timing of a IDENT command.

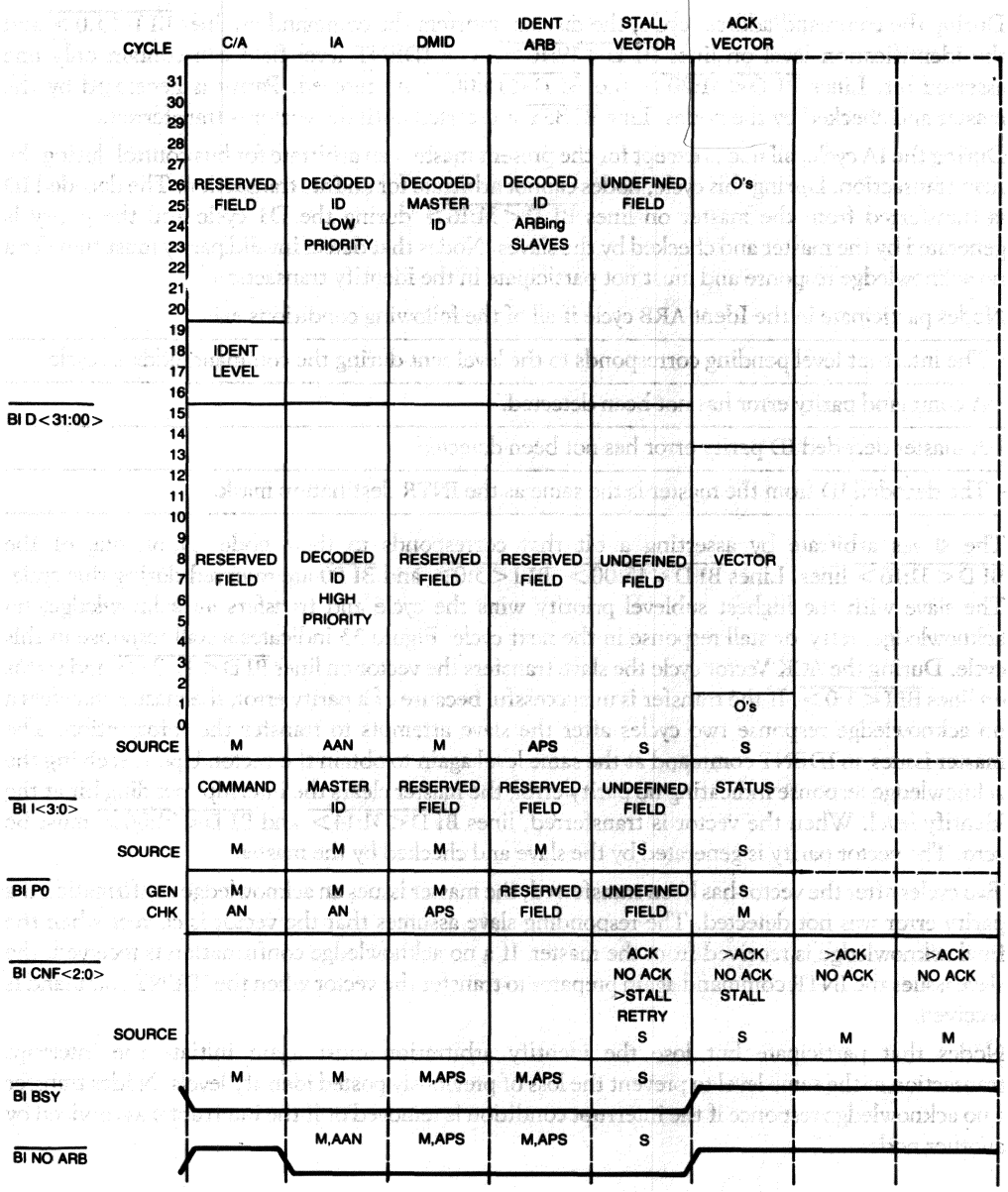


Figure 33 ▪ VAXBI 78732 IDENT Transaction Timing

During the command/address cycle, the master transfers the command on lines BI I<3:0> and the identification level on lines BI D<19:16>. The IDENT level field can contain only one asserted bit. Lines BI D<31:20> and BI D<15:00> are reseved. Parity is generated by the master and checked by the nodes. Line BI BSY is asserted until the vector is transferred.

During the IA cycle, all nodes except for the present master can arbitrate for bus control during the next transaction. During this cycle, nodes cannot arbitrate for an Intr transaction. The decoded ID is transferred from the master on lines BI D<31:16> during the D1 cycle and the parity is generated by the master and checked by the slaves. Nodes that detect invalid parity must transfer a no acknowledge response and must not participate in the identify transaction.

Nodes participate in the Ident ARB cycle if all of the following conditions exist:

- The interrupt level pending corresponds to the level sent during the command/address cycle.
- A command parity error has not been detected.
- A master decoded ID parity error has not been detected.
- The decoded ID from the master is the same as the INTR destination mask.

The slaves arbitrate by asserting a bit that corresponds to their node ID on one of the BI D<31:16> lines. Lines BI D<15:00>, BI I<3:0> and BI P0 are reserved during this cycle. The slave with the highest sublevel priority wins the cycle and transfers an acknowledge, no acknowledge, retry, or stall response in the next cycle. Figure 33 indicates a stall response in this cycle. During the ACK Vector cycle the slave transfers the vector on lines BI D<13:02> and status on lines BI I<3:0>. If the transfer is unsuccessful because of a parity error, the master transfers a no acknowledge response two cycles after the slave attempts to transfer the information. The master issues an IDENT command at the same level again to obtain the vector. Upon receiving the acknowledge response indicating no parity error, the master clears the interrupt pending bit at the identify level. When the vector is transferred, lines BI D<31:14> and BI D<01:00> must be zero. The vector parity is generated by the slave and checked by the master.

Two cycles after the vector has been transferred, the master issues an acknowledge confirmation if a parity error was not detected. The responding slave assumes that the vector is correct when the final acknowledge is received from the master. If a no acknowledge confirmation is received, the slave issues the INTR command again prepares to transfer the vector when the IDENT command is received.

Nodes that participate but lose the identify arbitration must again initiate the interrupt transaction at the same level to prevent the loss of previously posted identify levels. Nodes transfer a no acknowledge responce if the interrupt condition is removed or if the interrupt was serviced by another node.

VAXBI Interrupt Vectors

Two types of interrupt vectors are issued by VAXBI adapters and each adapter is allocated four interrupt vectors. The vector formats are shown in Figure 34.

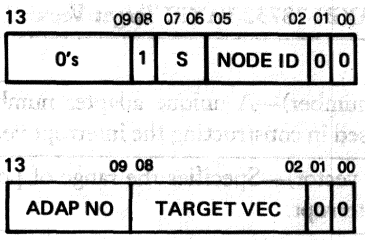


Figure 34 • VAXBI 78732 VAXBI Interrupt Vector Formats

The node ID vector is located between address locations 100 and 1FFF (hexadecimal). The node ID vector field assignments are described in Table 36.

Table 36 • VAXBI 78732 VAXBI Node ID Vector Descriptions

Bits	Description
13:09	MBZ (must be zero)
08	Set to one
07:06	S (Interrupt vector number)—One of four interrupt vector values assigned to a node.
05:02	NODE ID (Node identification)—A interrupting node value of from 0 through 15.
01:00	MBZ (must be zero)

The target vector specifies one of up to 128 interrupt service routines. Each adapter that issues this vector has a unique adapter number and the range of numbers determines the range of possible vectors. The target vector field assignments are described in Table 37.

Table 37 • VAXBI 78732 VAXBI Target Vector Descriptions

Bits	Description
13:09	ADAP NO (Adapter number)—A unique adapter number assigned by the operating system software and used in constructing the interrupt vector.
08:02	TARGET VEC (Target vector)—Specifies the range of possible vectors in a system. All zeros indicate a null interrupt.
01:00	MBZ (must be zero)

Interprocessor Interrupt—The interprocessor interrupt (IPINTR) is used by processors to interrupt the operation of other processors. This command is similar to the INTR command except that the level and vector are not transferred during the transaction but are stored in the node that receives the command. Figure 35 shows the transaction timing of a IPINTR command.

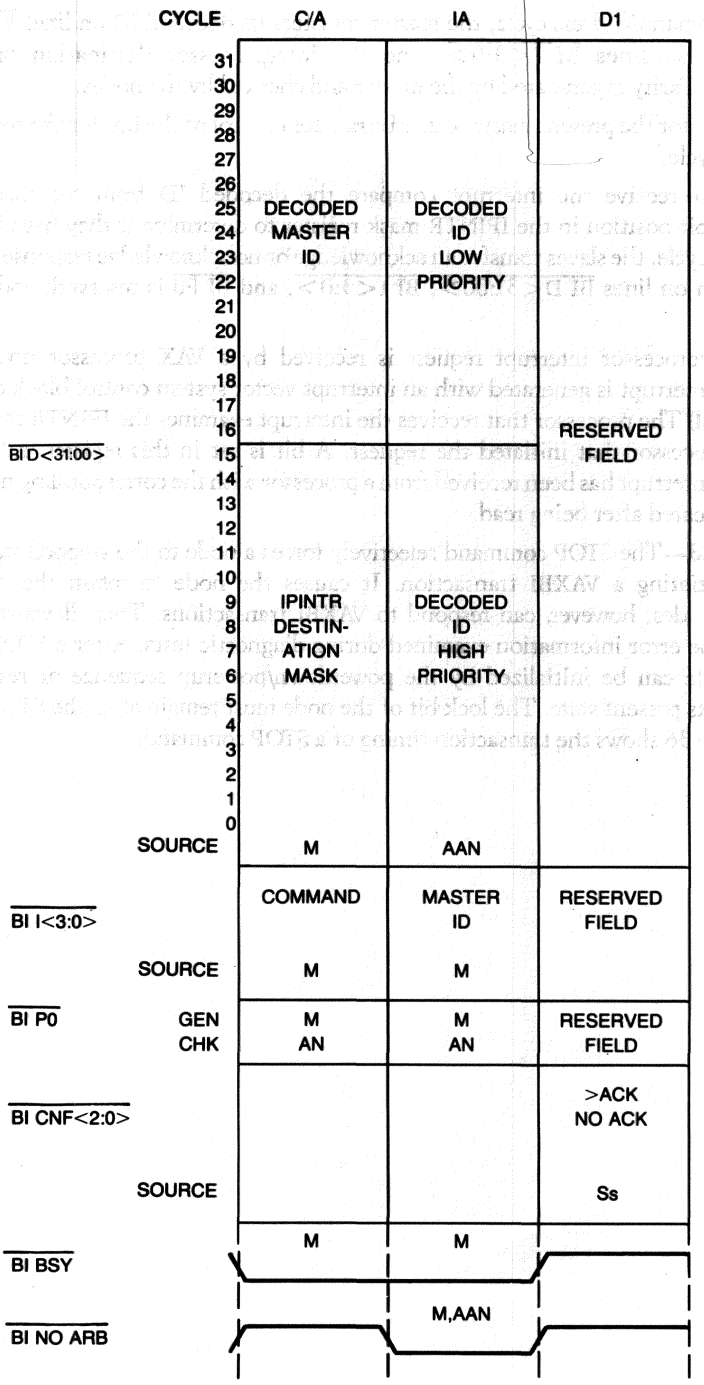


Figure 35 ▪ VAXBI 78732 IPINTR Transaction Timing

During the command/address cycle, the master transfers its decoded ID on lines $\overline{\text{BI D<31:16>}}$, the command on lines $\overline{\text{BI I<3:0>}}$, and the interprocessor destination mask on lines $\overline{\text{BI D<15:00>}}$. Parity is generated by the master and checked by the nodes.

All nodes except for the present master can arbitrate for control of the bus for the next transaction during the IA cycle.

The nodes that receive the interrupt compare the decoded ID from the master with the corresponding bit position in the IPINTR mask register to determine if they have been selected. During the D1 cycle, the slaves transfer an acknowledge or no acknowledge response to the master; the information on lines $\overline{\text{BI D<31:00>}}$, $\overline{\text{BI I<3:0>}}$, and $\overline{\text{BI P0}}$ is reserved; and parity is not generated.

When an interprocessor interrupt request is received by a VAX processor node, a level 14 (hexadecimal) interrupt is generated with an interrupt vector system control block offset value of 80 (hexadecimal) The processor that receives the interrupt examines the IPINTR souce register to identify the processor that initiated the request. A bit is set in this register indicates that an interprocessor interrupt has been received from a processor with the corresponding node ID. These bit should be cleared after being read.

STOP Command—The STOP command selectively forces a node to the stopped state preventing them from initiating a VAXBI transaction. It causes the node to retain the available error information. Nodes, however, can respond to VAXBI transactions. This allows the node to be accessed and the error information examined during diagnostic tests. After a STOP command is received, a node can be initialized by the powerdown/powerup sequence or restarted by the software from its present state. The lock bit of the node must remain after the STOP command is received. Figure 36 shows the transaction timing of a STOP command.

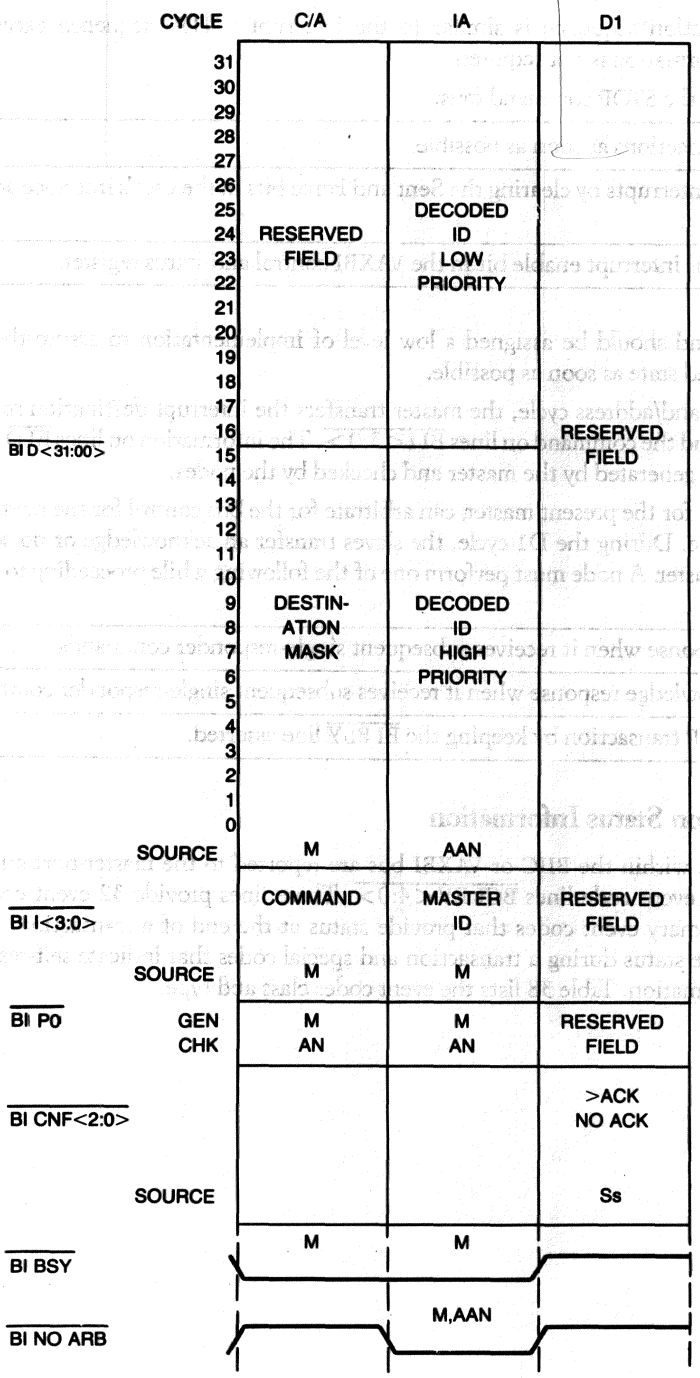


Figure 36 ▪ VAXBI 78732 STOP Transaction Timing

The STOP transaction sequence is similar to the interrupt (INTR) sequence except that the interrupt level information is not required.

Nodes selected by the STOP command must

- Stop issuing transactions as soon as possible.
- Remove posted interrupts by clearing the Sent and Force bits in the user's interface and interrupt control registers.
- Set the hard error interrupt enable bit in the VAXBI control and status register.

The Stop command should be assigned a low level of implementation to assure that the node reaches the stopped state as soon as possible.

During the command/address cycle, the master transfers the interrupt destination mask on lines $\overline{BI\,D<15:00>}$, and the command on lines $\overline{BI\,I<3:0>}$. The information on lines $\overline{BI\,D<31:16>}$ is reserved. Parity is generated by the master and checked by the nodes.

The nodes, except for the present master, can arbitrate for the bus control for the next transaction during the IA cycle. During the D1 cycle, the slaves transfer an acknowledge or no acknowledge response to the master. A node must perform one of the following while proceeding to the stopped state.

- Issue a retry response when it receives subsequent single-responder commands.
- Issue a no acknowledge response when it receives subsequent single-responder commands.
- Extend the STOP transaction by keeping the $\overline{BI\,BSY}$ line asserted.

BIIC Transaction Status Information

Significant events within the BIIC or VAXBI bus are reported to the master-port and slave-port interface through event code lines $\overline{BCI\,EV<4:0>}$. These lines provide 32 event codes that are grouped into summary event codes that provide status at the end of a transaction, status event codes that provide status during a transaction and special codes that indicate self-test status and bus timeout information. Table 38 lists the event codes class and type.

Table 38 • VAXBI 78732 BCI Event Code Assignments

Mnemonic	Type*	Definition
Summary		
NEV	—	No event
MCP	M:info	Master-port transaction complete
AKRSD	S:info	Acknowledge received for slave read data
RCR	M:info	Retry confirmation received for master-port command
IRW	S:info	Internal register written
NICI	M:error/info	No acknowledge or illegal confirmation received for interrupt command
NICIPS	I:error/info	No acknowledge of illegal confirmation received for Force-bit interprocessor/stop command
AKRE	I:info	Acknowledge confirmation received for error vector
STO	S:error	Stall timeout on slave transaction
BPS	S:error	Bad parity received during slave transaction
ICRSD	S:error	Illegal confirmation received for slave data
BBE	S:error	Bus busy error
AKRNE4	I:info	Acknowledge confirmation received for nonerror vector—level 4
AKRNE5	I:info	Acknowledge confirmation received for nonerror vector—level 5
AKRNE6	I:info	Acknowledge confirmation received for nonerror vector—level 6
AKRNE7	I:info	Acknowledge confirmation received for nonerror vector—level 7
RDSR	M:error	Read data substitute or reserved status code received
ICRMC	M:error	Illegal confirmation received for master-port command
NCRMC	M:error/info	No acknowledge confirmation received for master-port command
ICRMD	M:error	Illegal confirmation received by master-port data cycle
RTO	M:error	Retry timeout
BPM	M:error	Bad parity received during master-port transaction
MTCE	M:error	Master transmit error check
Status		
ARCR	M:info	Advanced retry confirmation received
IAL	I:info	Error identification arbitration lost
EVS4	I:info	External vector selected—level 4
EVS5	I:info	External vector selected—level 5
EVS6	I:info	External vector selected—level 6
EVS7	I:info	External vector selected—level 7
BPR	M/S:error	Bad parity received
Special Case		
BTO	M:error	Bus timeout
STP	info	Self-test passed

*M = master, S = slave, I = interrupt, info = information

Summary Event Code Operation—The summary EVent (EV) code indicates the successful completion of a transaction or an error condition resulting from a transaction with a node. The master-port interface receives one master summary EV code for each transaction unless the transaction is aborted. The slave-port interface receives this code for transactions in which an error is detected and for successful read-type and identification transactions, and when information is written into the VAXBI registers from the VAXBI bus.

The summary EV code lines are shared by the master and slave ports. The information is time-multiplexed with a transaction cycle dedicated to the master codes and a transaction cycle dedicated to the slave codes to assure that there is no contention between ports for the information.

Figure 37 shows the summary event code timing for a successful write-type and broadcast transaction of a longword. Figure 38 shows the event code timing for a successful read-type transaction of a longword. When a bus error causes the transaction to abort, the event code may occur before the times indicated unless the transaction is intranode.

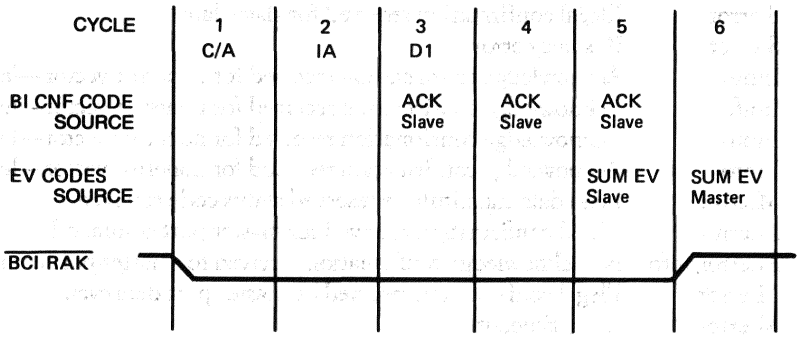


Figure 37 • VAXBI 78732 Summary Event—WRITE and BDCST Longword Transaction Timing

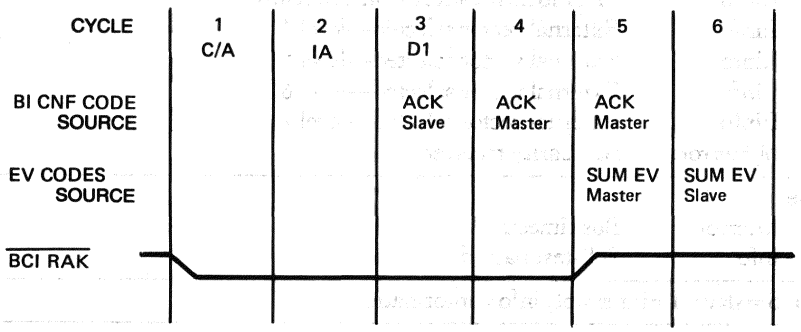


Figure 38 • VAXBI 78732 Summary Event—READ Transaction Timing

Figure 39 shows the summary event code timing for a successful STOP, INTR, IPINTR, and INVAL transaction from a master-port. No event code is transferred by the slave port for these transactions. During the STOP, IPINTR, and INVAL transactions, the event code is transferred during cycle 6. A no acknowledge or illegal confirmation received EV code is transferred during cycle 6 by unsuccessful IPINTR and INTR transactions generated by the BIIC.

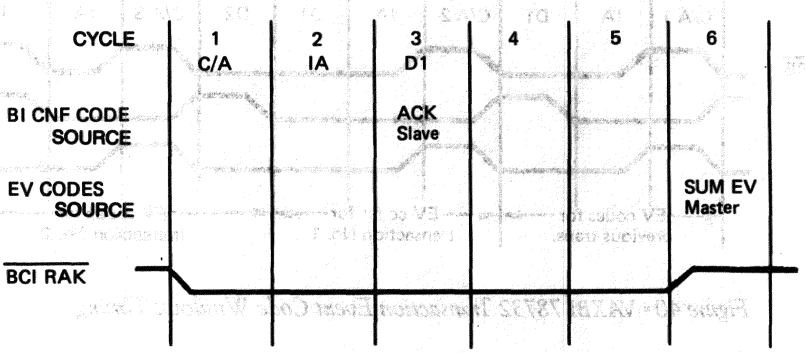


Figure 39 • VAXBI 78732 Summary Event—STOP, INTR, IPINTR, and INVAL Transaction Timing

Status Event Code Operation—Status event codes provide status information during a transaction. The bad parity received event codes (BPM and BPS) are transferred the cycle after a data cycle parity error has been detected by a slave or master node. This allows a write-type transaction to be terminated soon after the error has been detected. These event codes are also transferred at the end of the designated cycle.

The INTR command results in two types of slave status codes—The external vector selected (EVS4 through EVS7) codes and the identification lost (IAL) code. The EVS codes are transferred during cycle 4 and the IAL code during cycle 5.

Special Event Code Operation—The bus timeout (BTO) and self-test passed (STP) codes are special event codes that are not related to a transaction. The BTO code can be transferred during any transaction cycle except the data cycles of a master transaction from this node. Other event codes are transferred before the BTO code. This code is then transferred continuously until the request(s) are removed or the transaction begins. The STP code is transferred after the BIIC has completed the self-test operation.

Event Code Windows—The interval of time that the summary or status event codes can be transferred is from cycle 4 of the transaction until cycle 3 of the following transaction. The node is required to relate the transaction and associated event codes. Figure 40 shows the transaction event code windows.

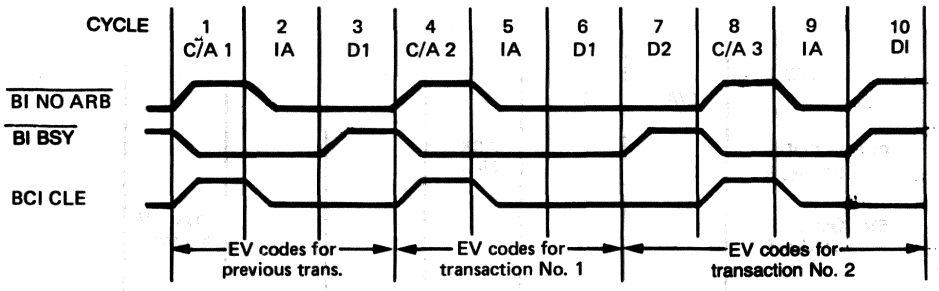


Figure 40 • VAXBI 78732 Transaction Event Code Windows Timing

Event Codes and Bus Error Register

Most event codes that are transferred are dependent on the status of the bus error register (BER) bits. Figure 41 shows the relationship of the codes to the BER. The event codes are defined in Table 39.

Event Codes	Bus Error Register Bits														
	NMR	MTCE	CTE	MPE	ISE	TDF	IVE	CPE	SPE	RDS	RTO	STO	BTO	NEX	ICE
BTO													↵↑		
NICI	↑														↑I
NICIPS	↑														↑I
STO												↵↑			
BPS									↑						
ICRSD							↵								↑
BBE															
RDSR										↵↑					
ICRMC															↑
NCRMC	↑													↵	
BPR				↵					↵						
ICRMD															↑
RTO											↑				
BPM				↑											
MTCE		↑I													

KEY

↵↑ Direct correlation between the output of the EV code and the setting of the Bus Error Register bit.

↵↑I Same as ↵↑ for master port interface transactions. BIIC-generated transactions will never cause the output of the MTCE EV code if the error condition is detected. The MTCE bit, however, will be set.

↑ This EV code represents one error condition, but not the only condition, that will result in the setting of this BER bit.

↑I Same as ↑ for illegal CNF responses; however, this error bit will not be set if the response was NO ACK.

↵ This BER bit represents one error condition, but not the only condition, that will result in the output of this EV code.

Figure 41 ▪ VAXBI 78732 Event Code and Bus Error Register Correlation

Table 39 • VAXBI Event Code Descriptions

Event	Description
NEV	No event reported (default deasserted state)
MCP	Indicates that the last master-port transaction on the VAXBI bus has been completed successfully. During nonpipeline requests, the MPC code is transferred during the cycle in which the $\overline{\text{BIC RAK}}$ signal is asserted. If a master performs pipeline requests, this code may appear during the C/A cycle or embedded ARB cycle. The master must associate this code with the related transaction. If a stop transaction selects the slave-port interface of the master, no summary event code will be sent.
AKRSD	Sent from a slave to indicate that the last read-type transaction was successfully completed.
BTO	Indicates that a node was unable to start a transaction within 4096 cycles after a request from the BIIC or a master-port interface had been posted. Sets the BTO bit in the bus error register. After the bus timeout, the BTO event code is transferred until all requests are completed or a transaction occurs.
STP	Indicates that the BIIC self-test has been successfully completed.
RCR	Indicates that the confirmation received from the slave during the C/A cycle from a slave or master has occurred previously (retry).
IRW	Indicates the completion of a VAXBI write-type transaction that was directed to the BIIC control and status register space of this node.
ARCR	Used by the master-port interface to support pipeline requests. This code is transferred by the BIIC during the the cycle that follows the receipt of the retry confirmation. Because this event code is transferred one or two cycles before the retry confirmation received event code, it is useful in support the master-port design of pipeline requests.
NICI	Transferred if the confirmation received for an Intr transaction is a no acknowledge code, reserved code, or an illegal response code.
NICIRS	Transferred for IPINTR or STOP commands that are initiated by setting the IPINTR/STOP bit in the BCI control and status register.
AKRE	Transferred after the slave receives an acknowledge confirmation from the identifying master for the transmitted error vector information from the slave.
IAL	Transferred by the BIIC two cycles after the identification arbitration was lost by the slave.
EVS4-EVS7	Used to solicit an external vector from the user's interface when the BIIC participates in the identification arbitration, when the EV bit is set in the interrupt control register of the user's interface, and when no error interrupt is pending at this node at a level selected by the IDENT command.
STO	Transferred if the user's interface stalls a data cycle for more than 127 consecutive cycles. A node that is not a slave will receive this code if it extends a VAXBI transaction for more than 127 cycles.

Event	Description
BPS	Transferred if a slave detects a parity error during a write-type acknowledge or stall cycle or during a broadcast acknowledge data cycle. This condition sets the SPE bit in the bus error register.
BPR	Transferred by a master or slave during the cycle after the BIIC detects a parity error during the following data cycle types: —Read-type acknowledge (for the master) —Vector acknowledge (for the master) —Write-type acknowledge or stall (for the slave) —Broadcast acknowledge (for the slave)
ICRMD	Transferred by the BIIC during read-type, write-type, and broadcast transactions if the slave returns an illegal confirmation code after the command confirmation has been received.
RTO	Transferred by the master for each retry response after 4095 retries from the slave when the RTOEVEN bit is set in the BCI control and status register.
BPM	Transferred when the master detects a parity error on the VAXBI bus during a read-type or vector acknowledge data cycle.
MTCE	Transferred for master-port transactions when the data received on the $\overline{\text{BI D<31;0>}}$, $\overline{\text{BI I<3:0>}}$ and $\overline{\text{BI P0}}$ lines is not the same as the data transmitted from this node. This occurs during a cycle in which the master should be the only node to transfer information on these lines except when encode ID of the master is transferred during an embedded arbitration cycle. The BIIC also sets the MTCE bit in the bus error register.

• BIIC Operation

The following describes the operation of the BIIC during powerup sequence and during VAXBI transactions.

Powerup Sequence

During the powerup sequence, the BIIC disables the VAXBI driver circuits, loads the configuration information from the user's interface into the BIIC registers, and performs a self-test operation. The BIIC asynchronously asserts the $\overline{\text{BCI DC LO}}$ signal when $\overline{\text{BI DC LO}}$ is asserted. Information from the BCI D<31:00> lines is loaded into the device register, the node ID information from the BCI I<3:0> lines is loaded into the VAXBI control and status register, and the state of the BCI P0 line is loaded into the user parity enable (UPEN) bit of the bus error register. The state of the BCI P0 line determines if the BIIC or user's interface will generate parity for the data transferred from the BIIC.

Self-test Operation

The BCI DC LO line must be asserted for a minimum of 72 cycles (14.4 microseconds) to allow the BIIC to initialize the registers associated with self-test. When the BI DC LO line is deasserted, the BIIC deasserts the BCI DC LO line and starts the self-test. The BI NO ARB line is asserted during the self-test operation to prevent VAXBI bus activity. After the self-test has been successfully completed, the BIIC transfers the EV code (STP) for one cycle during the cycle after the self-test completes. BI NO ARB is not asserted during the node reset sequence. The user's interface can monitor this code or can read the VAXBI control and status register to determine the results of the self-test. Table 40 lists the status of of the BIIC and user's interface signals during the self-test operation.

Table 40 • VAXBI 78732 Self-test Signal Status

BIIC asserted lines	BIIC deasserted lines	User asserted lines*
BCI D<31:00>	BCI MDE	BCI RQ<1:0>
BCI I<3:0>	BCI SDE	BCI INT<7:4>
BCI P0	BCI NXT	BCI RS<1:0>
BI NO ARB**	BCI CLE	BCI MAB
	BCI SEL	
	BCI SC<2:0>	
	BCI EV<4:0>	
	BI D<31:00>	
	BI P0	
	BI BSY	

*These lines are optionally asserted. During the self-test, the diagnostic mode code should not appear on lines BCI RQ<1:0> to prevent the termination of the self-test before completion.

**The BIIC asserts the BI NO ARB signal during powerup self-test but not during NODE reset self-test.

Retry State

The BIIC enters the retry state in response to a legal retry response code from a slave during read-type, write-type, and identification transactions. The BIIC transfers the RCR event code and stores the command/address information of the transaction and the first data longword in its buffers during write-type and broadcast transactions. When the user's interface deasserts and then reasserts the request, the BIIC reinitiates the transaction. This provides the node with a variable delay before the transaction is initiated again. Nodes can initiate a retry transaction by disabling the VAXBI transaction request with the RCR event code.

After the BIIC receives 4096 consecutive retry confirmation responses, it issues the RTO event code. The user's interface can then continue to retry the transaction and the BIIC will continue to transfer the RTO code each time it receives a RCR response for the recent transaction. The user's interface must assert the BCI MAB line to terminate the transaction.

Request Codes

The BIIC monitors the state of the request lines to determine if a transition has occurred. It compares the received state of the line from the previous cycle with the received state of the current cycle. The transition of the request lines from the deasserted state (no request) to the asserted state (a request other than diagnostic mode) is interpreted as a request. The request code on the $\overline{\text{BCI RQ}}$<1:0> lines can be removed during any cycle after the assertion of the $\overline{\text{BCI RAK}}$ signal by the BIIC. A new request will not be recognized by the BIIC until the request lines have been deasserted for at least one cycle. Therefore a new request cannot be presented until the second cycle after a request has been removed.

When the BIIC receives a VAXBI request, it initiates the bus arbitration for the new transaction as soon as posible. If the bus master simultaneously initiates a new request, the BIIC will arbitrate for the new request in the next available cycle after the last cycle of the present transaction.

A pipeline request is a request posted prior to the deassertion of the $\overline{\text{BCI RAK}}$ signal for the present master-port transaction. Figure 42 shows the signal timing for the pipeline requests that allow the master-port interface to transfer data at a throughput rate of 11.4 Mbytes per second.

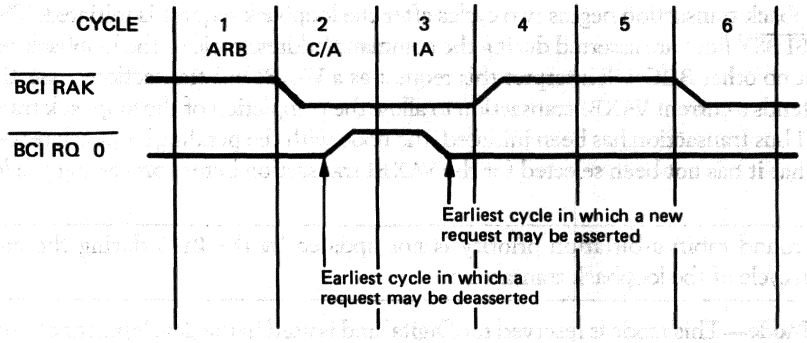


Figure 42 • VAXBI 78732 Request Signal Timing

VAXBI Transaction Request—The master-port interface uses the VAXBI transaction request code to request transactions on the VAXBI bus. The transactions can be directed to other nodes on the bus or to the slave node on the same VAXBI interface. Only longwords can be transferred to the slave port.

All VAXBI commands can be initiated by the user's interface except for the interrupt command. Interrupts are initiated by asserting a BCI INT<7:4> line or by setting a force-bit in the user's interface interrupt control register in the BIIC. The interprocessor interrupts can also be initiated by setting the IPINTR bit in the BCI control and status register.

Loopback Requests—The master-port interface uses the loopback request code to initiate longword intranode read-type and write-type transactions to node spaces that do not require the use of the the VAXBI data lines. Loopback transactions permit fast access to the BIIC and slave-port registers regardless of the activity on the VAXBI bus and during some bus failures. A node can access its node space registers without reference to the node ID. During loopback transactions, the BIIC disables the VAXBI drivers except for the $\overline{\text{BI NO ARB}}$ and $\overline{\text{BI BSY}}$ lines, and the transaction data is looped back to the VAXBI bus receive logic.

Loopback transactions and VAXBI bus transactions can occur concurrently. A loopback transaction can occur at one node while two other nodes are performing a VAXBI transaction. To assure proper bus operation, however, the BIIC will not initiate a loopback transaction when another node is initiating a VAXBI transaction. Because loopback transactions extend the bus cycle, the system access latency may be degraded.

Loopback transactions are received by the BIIC similarly to VAXBI transactions except for the following:

- High-order address bits on lines BI D<29:13> are ignored by the address selection logic of the BIIC except for parity qualification. The BIIC completes the transfer the same as if these lines contained the value 10 0000 0000 000n nnn (hexadecimal): where n nnn is the ID of this node. The address transferred to the slave port interface will be the same as the address received by the BIIC from the master-port interface. Read-type and write-type loopback transactions have limited addressing capability because the node ID is not required. The user's interface can access only the node space within the node that connects to the user's interface. Addresses defined by the starting and ending address registers of the BIIC can be accessed only by VAXBI transaction requests.
- The BIIC does not arbitrate for the bus and VAXBI transactions are not generated. If the bus is idle, a loopback transaction begins two cycles after the loopback request is initiated. The $\overline{\text{BI NO ARB}}$ and $\overline{\text{BI BSY}}$ lines are asserted during the command/address cycle of the loopback request to assure that no other BIIC will interpret this request as a VAXBI bus transaction. Asserting these signals extends a current VAXBI transaction to allow the completion of the loopback transaction. If a VAXBI bus transaction has been initiated, the node with the pending loopback request waits to verify that it has not been selected for the VAXBI transaction before processing the loopback request.
- The dual round robin arbitration priority is not updated by the BIIC during the embedded arbitration cycle of the loopback transaction.

Diagnostic Mode—This mode is reserved for Digital and is used in the development of bus testers, bus monitors, and other diagnostic equipment. It facilitates testing of the BIIC and provides access to the VAXBI bus. Refer to the *VAXBI Systems Reference Manual* for detailed information on the diagnostic mode.

The BIIC supports the BCI-to-BI and BI-to-BCI transparent mode operations where the BCI signals are reassigned to correspond with the VAXBI bus signals except for the $\overline{\text{BI AC LO}}$ and $\overline{\text{BI DC LO}}$ signals. The assignments and state of the signals are shown in Table 41.

Table 41 • Diagnostic Mode Bus Signal Assignments

BCI Line	State	BI Line
D<31:00>	inverted	D<31:00>
I<3:0>	inverted	I<3:0>
PO	inverted	PO
EV0	not inverted	CNF0
EV1	not inverted	CNF1
EV2	not inverted	CNF2
EV3	not inverted	NO ARB
EV4	not inverted	BSY

In the BCI-to-BI transparent mode, the user's interface transfers data on the BCI D<13:00>, BCI I<3:0>, and BCI P0 lines with the normal setup time and the data is synchronously transferred to the VAXBI bus by the BIIC.

In the BI-to-BCI transparent mode, the data received from the VAXBI bus is latched during each cycle and transferred to the BCI lines by the BIIC during each cycle. Data is transferred to the BCI in the same timing sequence as a nontransparent mode.

The diagnostic mode supports a command that allows the loading of the device type, node ID, and parity mode information at the end of the assertion of the $\overline{\text{BCI DC LO}}$ signal.

During the diagnostic mode, the BIIC examines the BCI interrupt and response lines $\overline{\text{BCI RQ<1:0>}}$ to determine the diagnostic mode operation. The code on these lines must not be transferred until the BIIC completes the self-test. The diagnostic mode control signals on lines $\overline{\text{BCI RQ<1:0>}}$, $\overline{\text{BCI RS<1:0>}}$, and $\overline{\text{BCI INT<7:4>}}$ may be transferred during the same cycle in which the BIIC is set to transparent mode. The operating mode can be changed by the $\overline{\text{BCI RS<1:0>}}$ and $\overline{\text{BCI INT<1:0>}}$ lined without deasserting the $\overline{\text{BCI RQ<1:0>}}$ lines. The new operation begins within three cycles after the mode change. Table 42 lists the response codes for the diagnostic mode.

Table 42 • VAXBI 78732 Diagnostic Mode Operating Codes

BCI RS Line		BCI INT Line				Operation
1	0	7	6	5	4	
H	H	H	H	H	H	No operation
H	H	L	L	L	L	BCI-to-BI transparent mode
H	L	H	L	H	L	Load configuration data
L	L	L	L	H	H	BI-to-BCI transparent mode

Read-type Transactions

During master-initiated read-type transaction, the master-port interface requests a VAXBI transaction through the $\overline{\text{BCI RQ}}$<1:0> lines. The BIIC responds to this request by asserting the $\overline{\text{BCI MDE}}$ signal to indicate to the master that a VAXBI command, address, and parity information (for user-generated parity) should be transferred on the BCI lines. When the BIIC wins control of the VAXBI bus, it transfers this information to the bus and asserts $\overline{\text{BCI RAK}}$ line. The asserting edge of the $\overline{\text{BCI NXT}}$ signal indicates when valid read data is on the BCI lines. If a STALL command is received from a slave, the assertion of the $\overline{\text{BCI NXT}}$ signal is delayed until an acknowledge is received. Following the last data cycle, the BIIC issues two acknowledge confirmations on the VAXBI bus if the transfers were successful. The final acknowledge can be inhibited by the user's interface if it asserts the $\overline{\text{BCI MAB}}$ signal. The MPC event code is transferred during the cycle that the final acknowledge is on the VAXBI bus. The $\overline{\text{BCI RAK}}$ line is also asserted at this time unless the master port issues a pipeline request.

All BIIC nodes deassert the $\overline{\text{BCI CLE}}$ line during the embedded arbitration cycle to allow the data from the BCI to be loaded into the BIIC. Each BIIC determines if the transmitted address is within the range of addresses that are allocated to its node. The BIIC in the selected node asserts the $\overline{\text{BCI SEL}}$ line and issues the appropriate select code on the $\overline{\text{BCI SC}}$<2:0> lines. The command response of the slave must be transferred on the $\overline{\text{BCI RS}}$<1:0> lines before the end of the embedded ARB cycle. An acknowledge response is a positive command confirmation and indicates to the master that the data cycle in process contains valid read data. During a read-type transaction, a stall response indicates that the data is not valid, a retry response indicates that the command cannot presently be completed, and a no acknowledge response indicates that the node was not selected by the transmitted address. The slave provides read data, data status, and responses continually until all data is transferred from slave to master. At the end of successful transactions, the BIIC of the master node transfers two acknowledge responses on the VAXBI bus. The BIIC of the slave node responds with a AKRSD event code during the cycle following the final acknowledge response of the BIIC.

The BIIC controls the read transactions with its internal registers. The user's interface, however, can monitor the read transactions of its node space if the BCISREN bit in the BCI control and status register was set. When a successful read-type transaction to an internal register of the BIIC has been performed, the BIIC issues an AKRSD event code.

Write-type Transactions

The master-port interface requests a VAXBI transaction on lines $\overline{\text{BCI RQ}}$<1:0>. The BIIC responds to this request by asserting the $\overline{\text{BCI MDE}}$ line to indicate to the master that a VAXBI command, address, and parity information (for user-generated parity) should be transferred on the BCI lines. After the BIIC wins the VAXBI bus, it transfers this information to the bus and asserts the $\overline{\text{BCI RAK}}$ line. During the embedded arbitration cycle, the assertion edge of the $\overline{\text{BCI NXT}}$ signal indicates that the first data word should be ready for transfer to the bus. During the same cycle the BIIC asserts the $\overline{\text{BCI MDE}}$ line that transfers the first data longword from the user's buffer to the BCI bus. The $\overline{\text{BCI NXT}}$ and $\overline{\text{BCI MDE}}$ signals transfer each data word during a cycle until the transfer is completed. An additional $\overline{\text{BCI NXT}}$ cycle occurs at the end of this transaction if the pipeline NXT enable bit in the BCI control and status register is set. After the last write-data cycle, the slave transfers two acknowledge responses. The BIIC then issues an MCP event code to the user's interface and in the same cycle the $\overline{\text{BCI RAK}}$ line is deasserted unless a pipeline request was issued from the master-port interface.

All BIIC nodes deassert the $\overline{\text{BCI CLE}}$ line during the embedded arbitration cycle to allow the data from the BCI to be loaded into the BIIC. Each BIIC determines if the transmitted address is within

the range of addresses that are allocated to its node. The BIIC in the selected node asserts the $\overline{\text{BCI SEL}}$ line and issues the appropriate select code on the $\overline{\text{BCI SC}}<2:0>$ lines. The command response of the slave must be transferred on the $\overline{\text{BCI RS}}<1:0>$ lines before the end of the embedded ARB cycle. An acknowledge response is a positive command confirmation and indicates to the master that the data cycle in process contains valid read data. A stall response indicates that the data from the first data cycle should be repeated in the second data cycle. A retry response indicates that the command cannot presently be completed. A no acknowledge response indicates that the node was not selected by the address transferred.

The slave-port interface provides an acknowledge or stall response for each data cycle until all the data is transferred from master to slave. At the end of successful transactions, the BIIC of the slave node transfers two acknowledge responses on the bus and the BIIC of the master node responds with a MCP event code during the cycle following the final acknowledge response of the slave.

The BIIC controls the write transactions with its internal registers. The user's interface however can monitor the write transactions if the BCISREN bit in the BCI control and status register is set. A write transaction to an internal register or to a node is similar except that the information on the $\overline{\text{BCI RS}}<1:0>$ lines has no effect on the confirmation responses of the BIIC. When a successful write-type transaction to an internal register of the BIIC has been performed, the BIIC issues an IRW event code.

Interrupt and Identification Transactions

The BIIC can generate a user's interface interrupt or an error interrupt by transmitting an INTR command on the VAXBI bus. Error interrupt requests have the highest priority for transaction transmission and in response to IDENT commands.

The user's interrupt is initiated by the interface when it asserts one of the $\overline{\text{BCI INT}}<7:4>$ lines or when it performs a write transaction to set the appropriate interrupt force-bit in the user's interface interrupt control register.

The error interrupt is automatically generated by the BIIC when a bus error is detected and the hard error interrupt enable (HER) bit of the VAXBI control and status register is set. The user's interface can also cause an error interrupt by setting the interrupt force (INTR force) bit in the error interrupt control register.

Following an interrupt request, the BIIC arbitrates for the VAXBI bus and, after winning the bus, it initiates the interrupt transaction. The user's interrupt and error interrupt use the INTR destination register to select a node. During the command/address cycle of the interrupt transaction, the appropriate INTR Sent bit is set in the user's interface interrupt control register or the error interrupt control register.

If more than one interrupt level is pending for the user's interface, the BIIC will transfer an interrupt request with all interrupt levels indicated when the VAXBI bus is available. Because only the INTR Sent bit of the highest pending level is set, the BIIC will arbitrate for the VAXBI bus again to send the remaining levels of the pending interrupts. The BIIC transmits the INTR command without interrupt levels if the interrupt condition is removed and the user's interface deasserts $\overline{\text{BCI INT}}<7:4>$ one or two cycles before the arbitration cycle has occurred.

Slave nodes capable of receiving interrupts should set the appropriate interrupt pending bit or its equivalent to record the interrupt level received. This information is transferred in the level field on the BCI D<19:16> lines during the identification command/address cycle. When the interrupt command is not successfully received, the BIIC sets the INTRAB and INTRC bits in the user interrupt control and status register at the levels received during the interrupt command. The INTRC bit prevents additional interrupts at that level from being transferred. Therefore this bit

must be cleared and reset by a node to again initiate an interrupt request. The INTRC and INTRAB bits can be cleared by performing a write transaction to the interrupt control register.

The node that receives the interrupt request issues an IDENT command to obtain the vector information from the node that initiates the interrupt. The IDENT command is initiated by the master-port interface to request a VAXBI transaction. The identification level information consisting of only one level bit is provided by the user's interface for transfer during the IDENT command cycle.

All slaves with interrupt requests pending at the same identification level will participate in IDENT commands by verifing that the decoded master ID transferred during the third cycle is the same as the bit set in their interrupt destination register. When both condition exist, the node participates in the identification arbitration. User's interface data from the slave port is not required unless the external vector bit of the user interface interrupt control register is set. The BIIC can transfer an external vector selected event code (EVS7 through EVS4) during the identification arbitration cycle if required. The event arbitration lost (IAL) event code causes the slave-port interface to return to the idle state. The use of this code is shown in Figure 43. Nodes that use external vectors must stall the vector transfer for a minimum of one cycle.

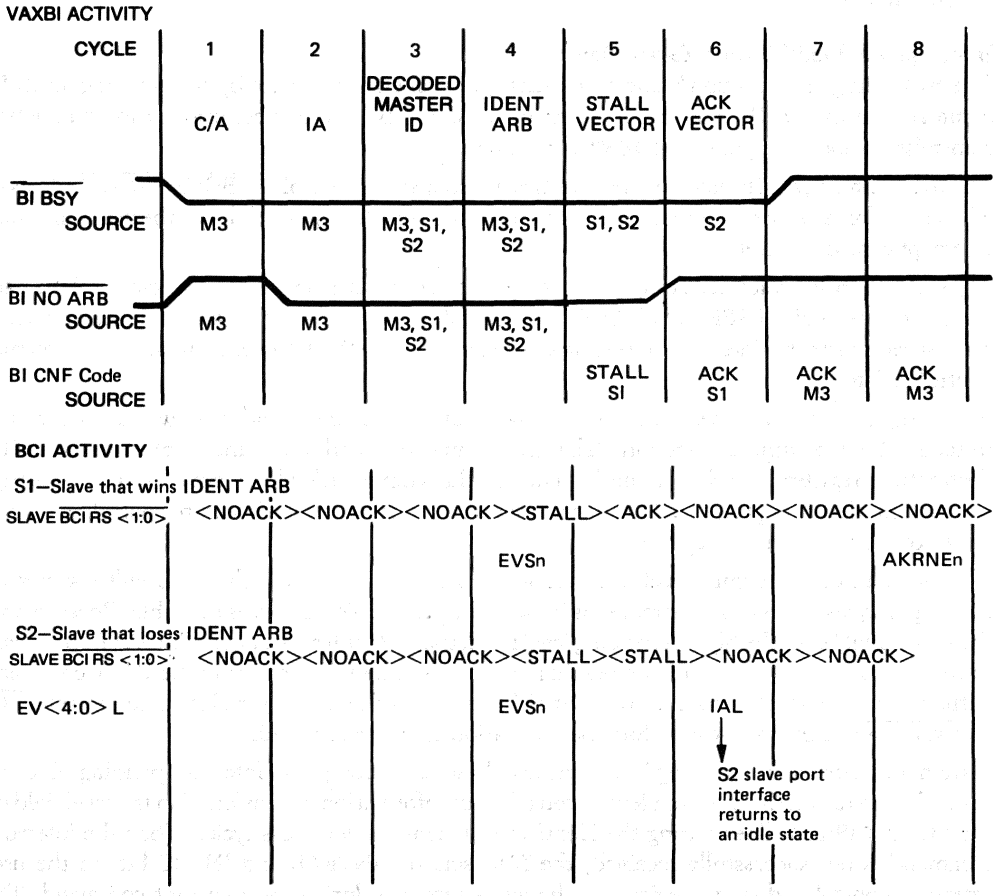
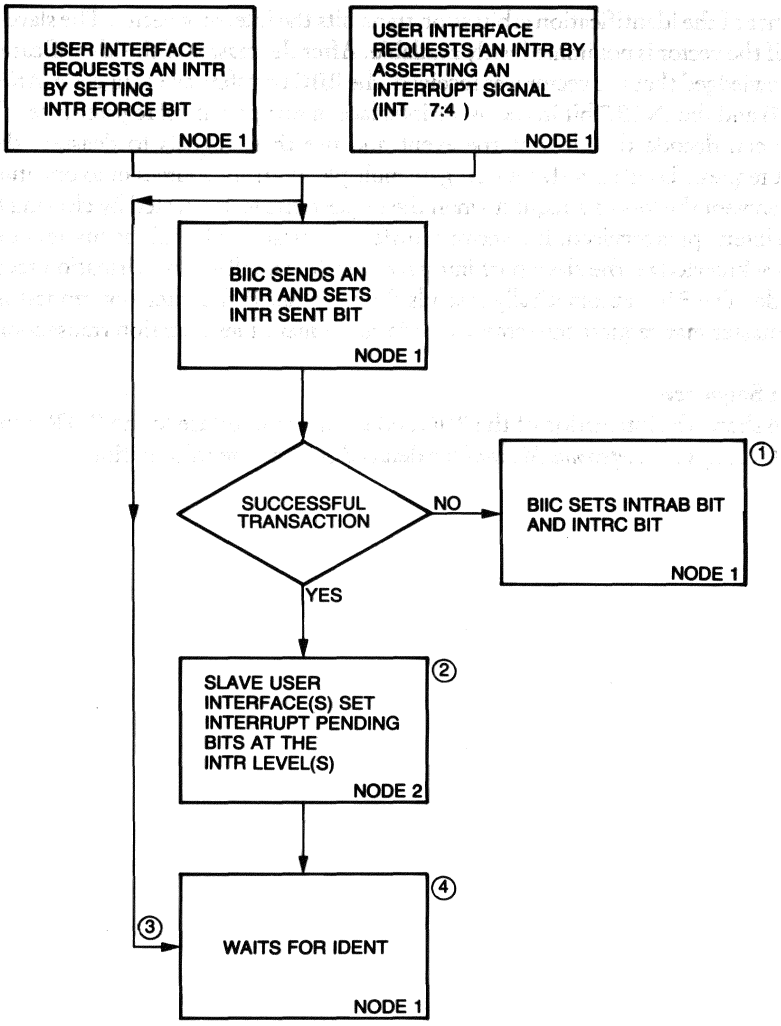


Figure 43 • VAXBI 78732 Identification Transaction Event Code Timing

The winner of the identification arbitration transmits the interrupt vector. The slave-port interface can stall if the vector is not immediately available. After the master of the identification transaction has acknowledged that the vector was received, the BIIC transfers an event code (AKRNE7 through AKRNE4) and the INTRC bit in the user's interface interrupt control register is set. The slave-port interface can decode the class of the event and use these signals to deassert the appropriate interrupt request. User's interfaces that gate multiple interrupt sources onto one interrupt request line can prevent the loss of a request when the request line is deasserted by clearing the INTRC bit after the interrupt is serviced. If a vector transfer to a master fails and the master does not transfer the final acknowledge, the slave-port interface transfers an illegal confirmation received (ICRSD) event code. The BIIC automatically resends the interrupt if the interrupt request is still pending and the master may request the same vector in subsequent identification transactions.

Interrupt Sequence

Figure 44 shows the interaction of the BIIC and the user's interface to the INTR transaction. Refer to the *VAXBI System Reference Manual* for detailed transaction information.



NODE 1 Node sending the INTR

NODE 2 Node responding to the INTR

Figure 44 ▪ VAXBI 78732 BIIC and User's Interface INTR Sequence Flow Diagram

Identification Sequence

Figure 45 shows the interaction of the BIIC and the user's interface to the IDENT transaction. Refer to the *VAXBI System Reference Manual* for detailed transaction information.

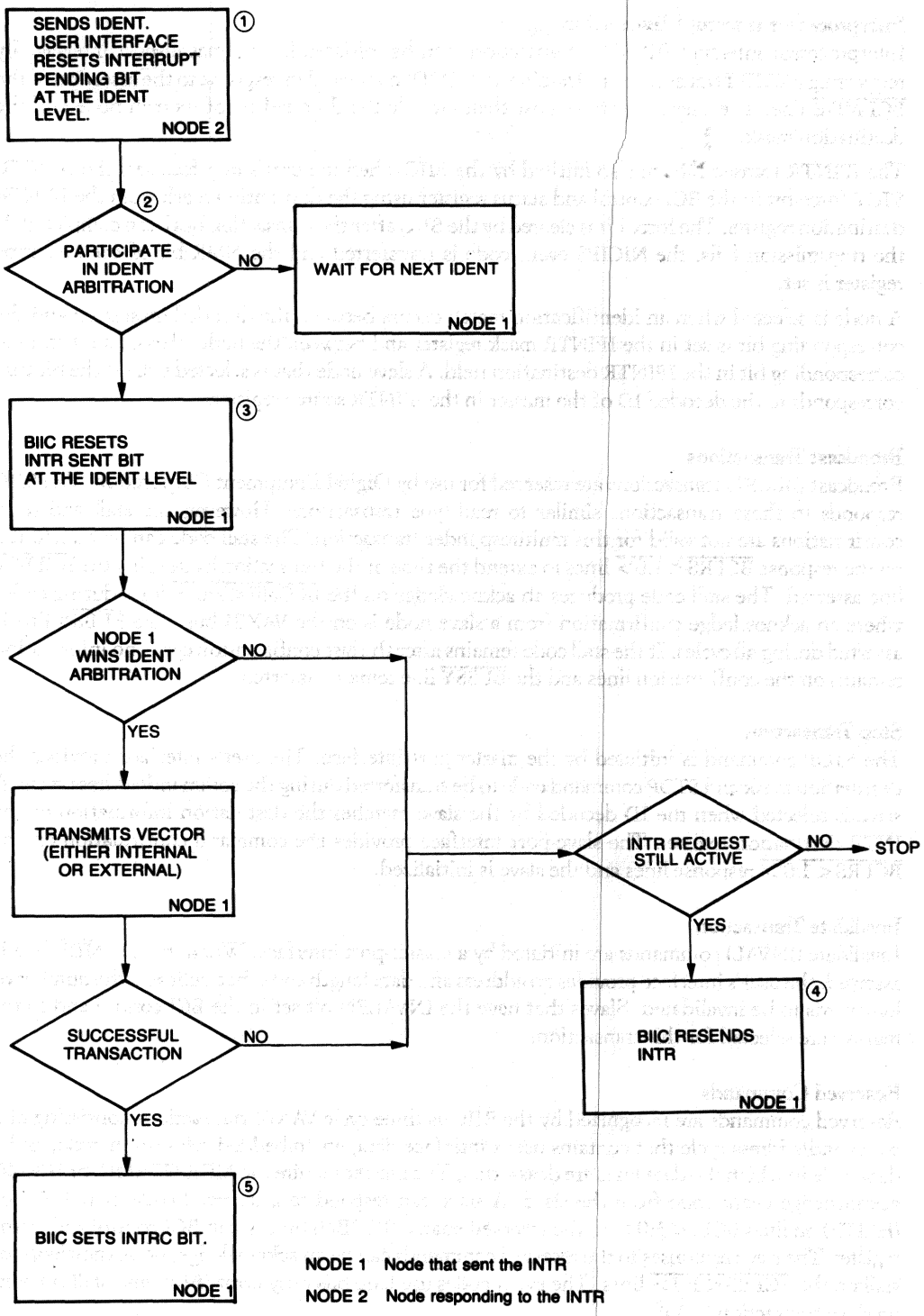


Figure 45 ▪ VAXBI 78732 BIIC and User's Interface IDENT Sequence Flow Diagram

Interprocessor Interrupt Transaction

Interprocessor interrupt (IPINTR) transactions can be initiated by the master-port interface by requesting a VAXBI transaction and issuing an IPINTR command in response to the assertion of the $\overline{\text{BCI MDE}}$ line. The user's interface must then provide the decoded ID of its own node and the destination mask.

The IPINTR transaction are also initited by the BIIC when the user's interface sets the IPINTR/STOP force-bit in the BCI control and status register using the destination mask from the IPINTR destination register. The force bit is cleared by the BIIC after the transaction has been completed. If the transmission fails, the NICIPS event code is transferred and the NMR bit of the bus error register is set.

A node is selected when an identification match occurs between the decoded master ID and the corresponding bit is set in the IPINTR mask register and between the nodes decoded ID and the corresponding bit in the IPINTR destination field. A slave node that is selected will set the bit that corresponds to the decoded ID of the master in the IPINTR source register.

Broadcast Transactions

Broadcast (BRCST) transactions are reserved for use by Digital Equipment Corporation. The BIIC responds to these transactions similar to read-type transactions. However, the stall and retry confirmations are not valid for this multiresponder transaction. The stall code can be transferred on the response $\overline{\text{BCI RS}}<1:0>$ lines to extend the time of the transaction by keeping the $\overline{\text{BCI BSY}}$ line asserted. The stall code produces an acknowledge on the $\overline{\text{BI CNF}}<2:0>$ lines during cycles where an acknowledge confirmation from a slave node is on the VAXBI bus. The $\overline{\text{BI BSY}}$ line is asserted during all cycles. If the stall code remains after the last confirmation cycle, no information remains on the confirmation lines and the $\overline{\text{BI BSY}}$ line remains asserted.

Stop Transaction

The STOP command is initiated by the master-port interface. The user's interface provides the destination mask and STOP command code to be transferred during the command/address cycle. A slave is selected when the ID decoded by the slave matches the destination information of the INTR destination register. The slave-port interface provides the command confirmation on the $\overline{\text{BCI RS}}<1:0>$ response lines and the slave is initialized.

Invalidate Transaction

Invalidate (INVAL) commands are initiated by a master-port interface. When the $\overline{\text{BCI MDE}}$ line is asserted, the user's interface provides an address and data length code that indicates the number of longwords to be invalidated. Slaves that have the INVALEN bit set in the BCI control and status register are selected for this transaction.

Reserved Commands

Reserved commands are recognized by the BIIC as three cycle VAXBI transactions consisting of a command/address cycle that contains user's interface data, an embedded arbitration cycle, and a data cycle in which the data lines are deasserted. The master requires an MCP, NCRMC, or ICRMC acknowledge event code from the slave. A slave can respond to a reserved code of (HLHL) or (HLHH) on lines BCI I<3:0> if the reserved enable (RESEN) bit is set in BCI control and status register. The slave responses to the reserved commands can be an acknowledge, no acknowledge or stall on the $\overline{\text{BCI RS}}<1:0>$ lines. The event codes used are bus busy error (BBE) and stall timeout on slave transaction (STO).

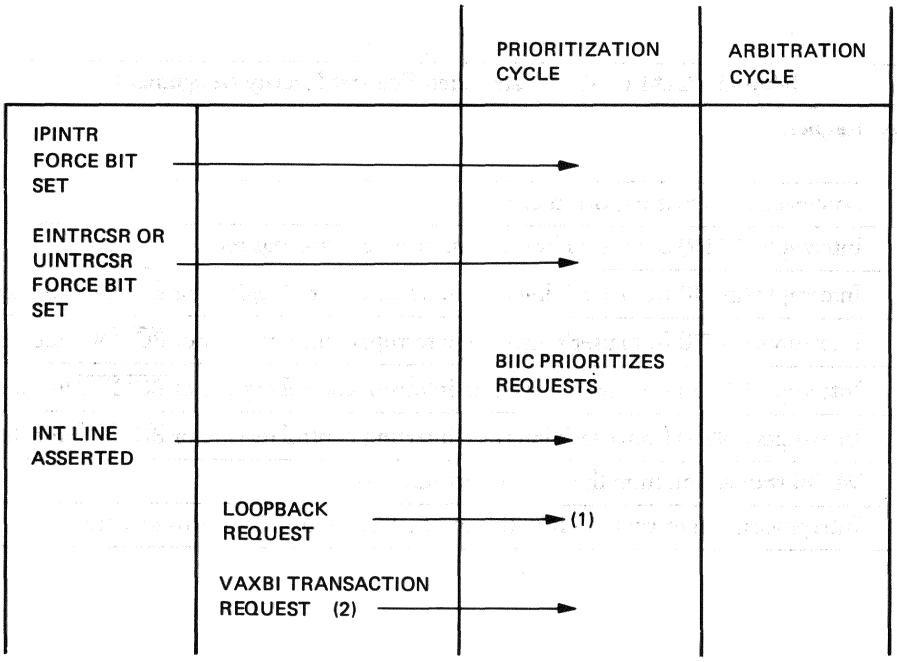
Transaction Priority

The BIIC recognizes more than one pending request. A VAXBI transaction request, an error interrupt request, an interprocessor request, and a user's interface interrupt request may be pending simultaneously. The priority for processing multiple requests is listed in Table 43.

Table 43 • VAXBI 78732 Transaction Request Priority Assignments

Priority Level	Request
1	Loopback from master-port interface
2	Interrupts (INTR) controlled by error interrupt control register
3	Interrupts (INTR) from user's interface interrupt control register or $\overline{\text{BCI INT}}$ line (level 7)
4	Interrupts (INTR) from user's interface interrupt control register or $\overline{\text{BCI INT}}$ line (level 6)
5	Interrupts (INTR) from user's interface interrupt control register or $\overline{\text{BCI INT}}$ line (level 5)
6	Interrupts (INTR) from user's interface interrupt control register or $\overline{\text{BCI INT}}$ line (level 4)
7	VAXBI transaction from the master-port interface
8	Interprocessor interrupts (IPINTR) from the BCI control and status register.

Figure 46 shows the types of requests and the timing relationship between the posting of the request and the priority assignment by the BIIC. The BIIC examines the request and establishes its priority during the prioritization cycle.



NOTES:

1. Loopback transactions have a dummy ARB cycle at this point.
2. If the VAXBI transaction request is posted while other types of requests are present, the BIIC prioritizes the VAXBI transaction request along with the other requests during the prioritization cycle. However, if no other types of requests are present, the BIIC attempts to arbitrate in the next cycle.

Figure 46 • VAXBI 78732 BIIC Transaction Priority Sequence

When no requests are pending, the BIIC arbitrates the VAXBI transaction request in the cycle that follows the request to minimize the bus latency time. If a VAXBI transaction request is posted while other requests are present, the VAXBI transaction is assigned priority together with the other requests during the prioritization cycle.

• Transaction Timing Sequences

Figures 47 through 74 are functional diagrams that show the timing sequence of the signals used to perform VAXBI interface transactions. The signals that communicate with the VAXBI bus are prefixed with a BI designation and signals that communicate with the user's interface are prefixed with a BCI designation. Table 44 lists the figure numbers and captions that includes notes that define the transaction operation.

Table 44 • VAXBI 78732 Functional Timing Diagram Descriptions

Figure	Title	Condition
48	Loopback longword read-type	To BIIC CSR space with an idle bus and with BICSREN set.
49	Loopback longword read-type with Stall	Occurs within node on top of a longword read with a stalled VAXBI transaction between master-port interface and node 1 and the slave-port interface and node 2.
52	Quadword read-type with pipeline request	Master quadword read to same slave node.
59	Force-bit requested interrupt	An INTR transaction initiated by a force bit in the UINTRCSR.
62	Force-bit requested interprocessor interrupt	An IPINTR transaction initiated by a force bit in the UINTRCSR.
66	Stop	Indicates the results of one allowable response to a STOP command. Although BIIC 2 wins the imbedded ARB in cycle 5, it asserts $\overline{\text{BI NO ARB}}$ in cycle 6 because of the STOP command. If user 2 holds $\overline{\text{BCI REQ}}$ asserted, BIIC 2 would arbitrate in cycle 7 and continue the transaction.
67	Stop with extension	Slave-port interface initiates a stall response to assert $\overline{\text{BI BSY}}$ signal while node completes Stop operation.
69	Burst-mode write operations with pipeline request	Begins at first arbitration cycle won by this node after the BURSTEN bit of BCICSR was set by previous transaction. A quadword write is followed by a longword write to the BCICSR to clear BURSTEN bit.
70	Burst mode writes with pipeline request and PNXTEN bit set	BCI CLE is asserted in a cycle following a cycle with $\overline{\text{BI NO ARB}}$ asserted and $\overline{\text{BI BSY}}$ deasserted and until T_0 of the command /address cycle.
71	Special case 1	1. Master 1 wins arbitration and initiates a quadword WMCI transaction to slave. 2. Master 2 requests the bus, arbitrated in the embedded arbitration cycle, and becomes the pending master. 3. BIIC 2 receives command/address data from master 2 during the embedded arbitration cycle to avoid BCI bus contention. 4. Master 2 performs a longword WMCI intranode transfer to an internal register in slave 2 (intranode transfer).
72	Special case 2	Master 1 initiates a longword read-type transaction to its slave port interface (slave 2)

Figure	Title	Condition
73	Special case 3	1. Master 1 and master 2 arbitrate in cycle 3. 2. Master 1 wins the arbitration and initiates a quadword WMCI transaction to slave 2. 3. Master 2 arbitrates again in the embedded arbitration cycle of master 1 and becomes the pending master. 4. BIIC 2 receives the command/address data from master 2 during the embedded arbitration cycle to avoid BCI bus contention. 5. Master 2 performs a longword WMCI intranode transfer to an internal register in slave 2.
74	Special case 4	1. Master 2 wins the arbitration and initiates a quadword read-type transaction to slave 1. 2. Master 1 requests the bus, arbitrates during its embedded arbitration cycle becomes the pending master. 3. BIIC 1 cannot receive the command/address data during the embedded arbitration cycle and waits until $\overline{\text{BCI SDE}}$ is deasserted in cycle 7 before asserting BCI MDE. 4. Master 1 performs a quadword read-type transaction to slave 3.

Table 45 lists the abbreviations used on the functional timing diagrams except for the event code abbreviations. The numbers that follow designations M, S, USER, and BIIC are node identifications (ID).

Table 45 • VAXBI 78732 Timing Diagram Abbreviations

Abbreviation	Description
AAN	all arbitrating nodes
AAS	all arbitrating slaves
ACK	acknowledge confirmation code
ADR	address including data length field
ARP	ABR pattern (decoded ID from all arbitrating nodes)
CMD	command
DA1	data word 1
DMI	decoded master ID
DSI	decoded slave ID on lines D<31:16> from all arbitrating nodes
DSM	destination mask including length field for BDCST command
IDD	master ID and destination code
ILV	identify level(s) on lines D<19:16>
INTR REQ	interrupt request
LB REQ	loopback request
LCD	level and destination code
M	master node
MID	master ID
MK1	write mask 1
NAK	no acknowledge confirmation code
REC	retry confirmation received for master-port command
RES	reserved
RET	retry confirmation code
S	slave node
STA	stall confirmation code
STS	read status code
USER	user's interface
UDF	undefined data
VAXBI REQ	VAXBI request
VEC	identify vector
VST	identify vector status
WS	winning slave
item	the potential occurrence of an item that is not shown on the diagram

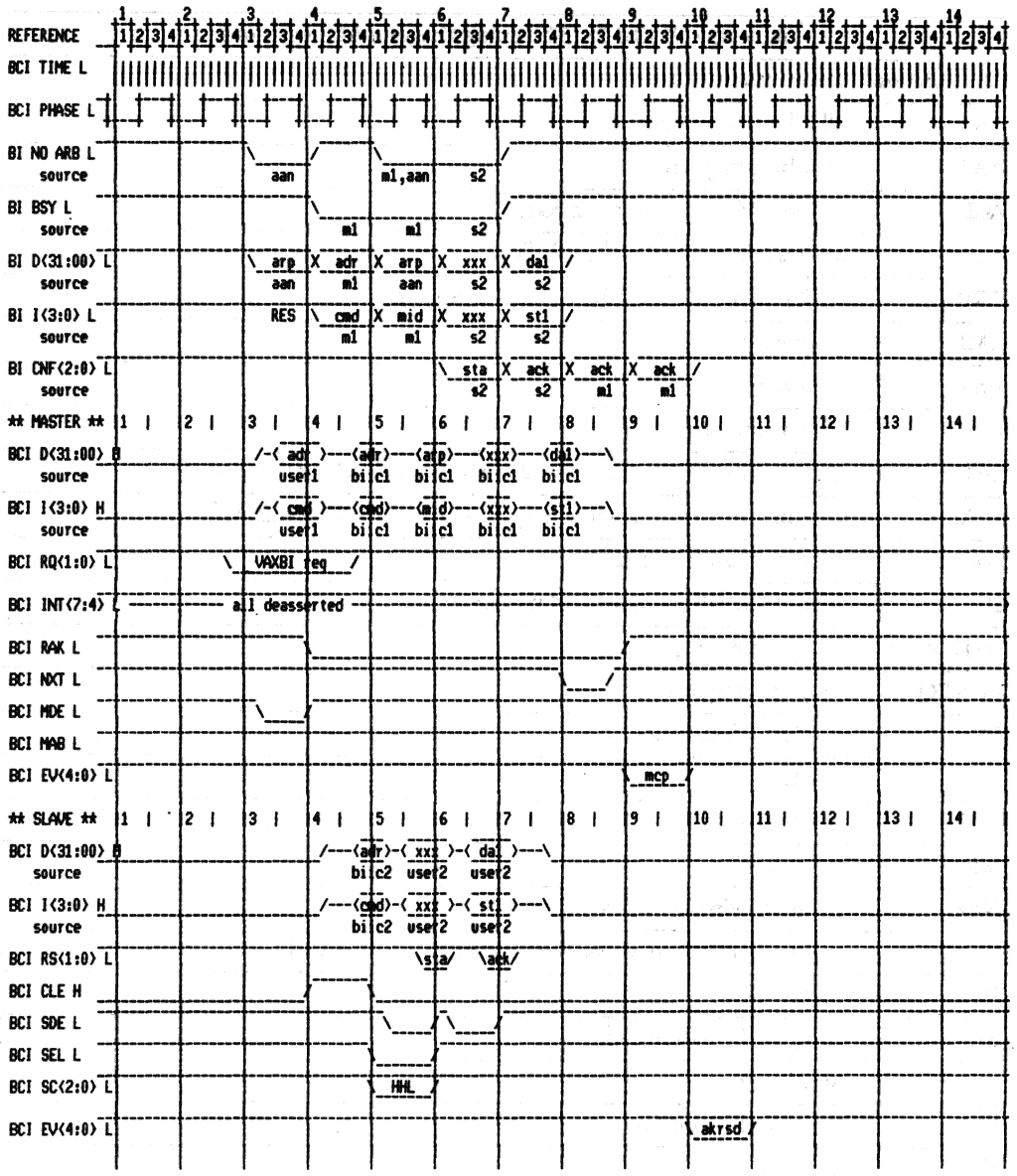


Figure 47 - VAXBI 78732 Longword Read-type (Stall) Transaction Timing Sequence

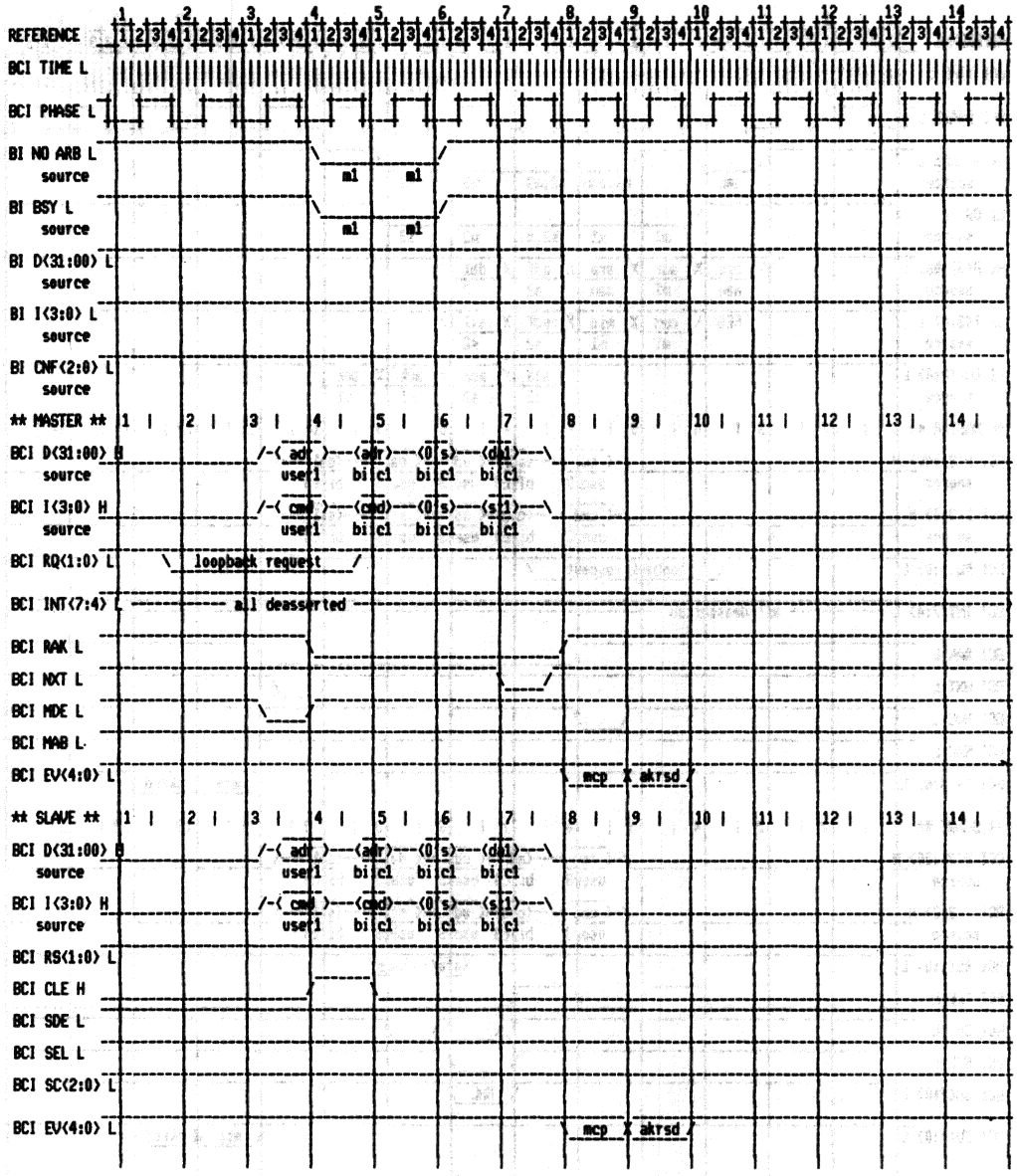


Figure 48 ▪ VAXBI 78732 Loopback Longword Read-type Transaction Timing Sequence

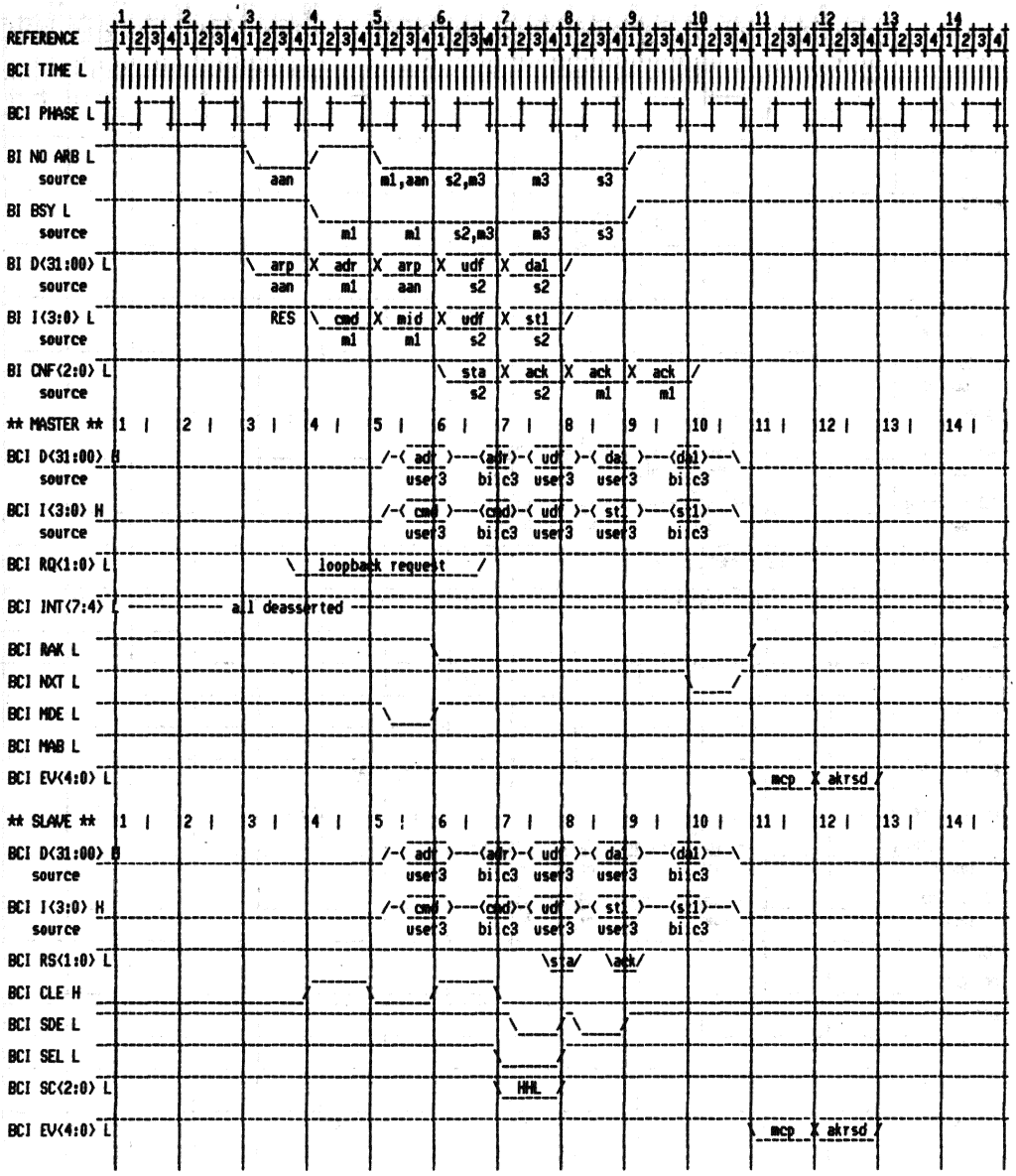


Figure 49 - Loopback Longword Read-type (Stall) Transaction Timing Sequence

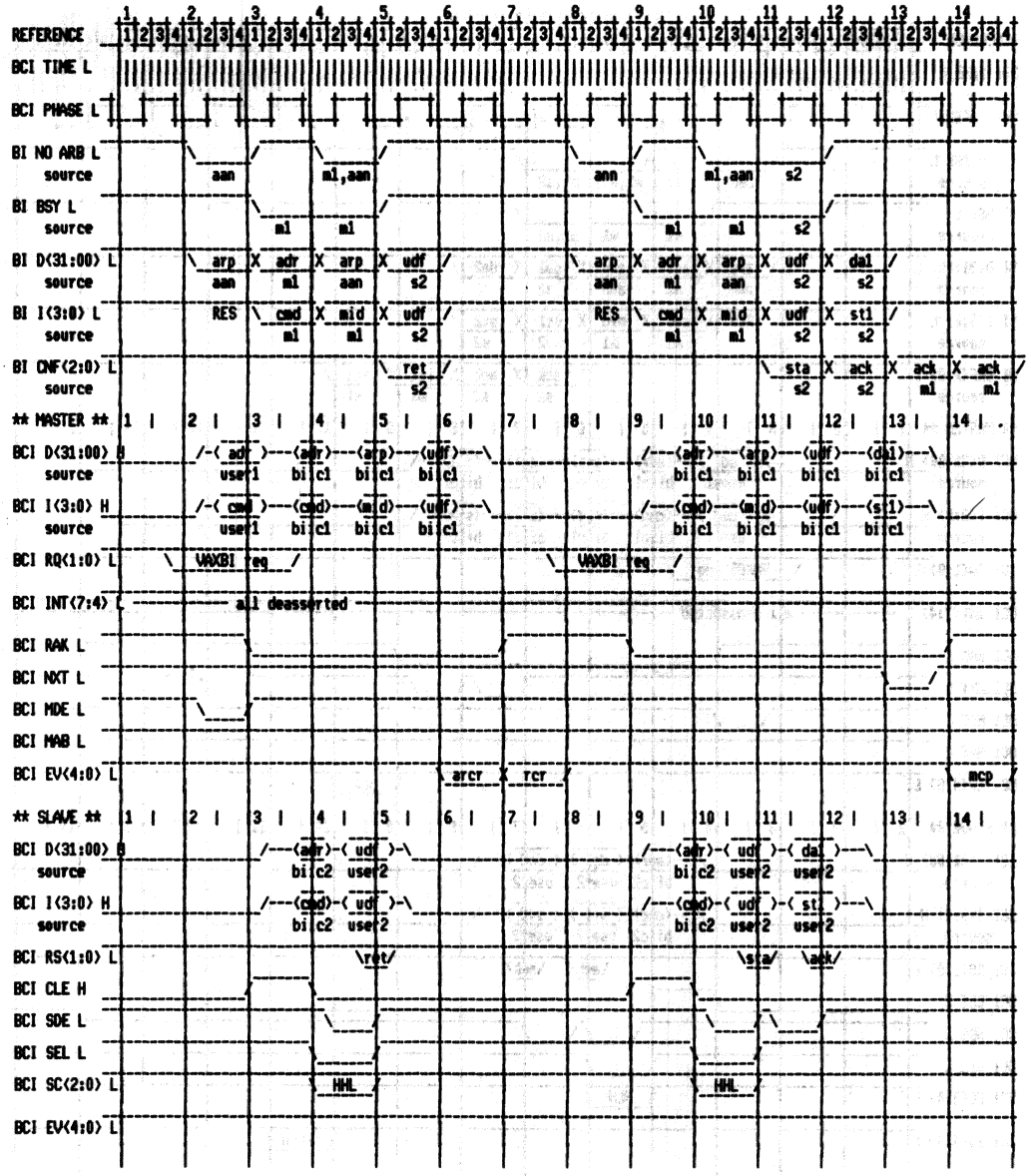


Figure 50 • VAXBI 78732 Retry Longword Read-type (Stall) Transaction Timing Sequence

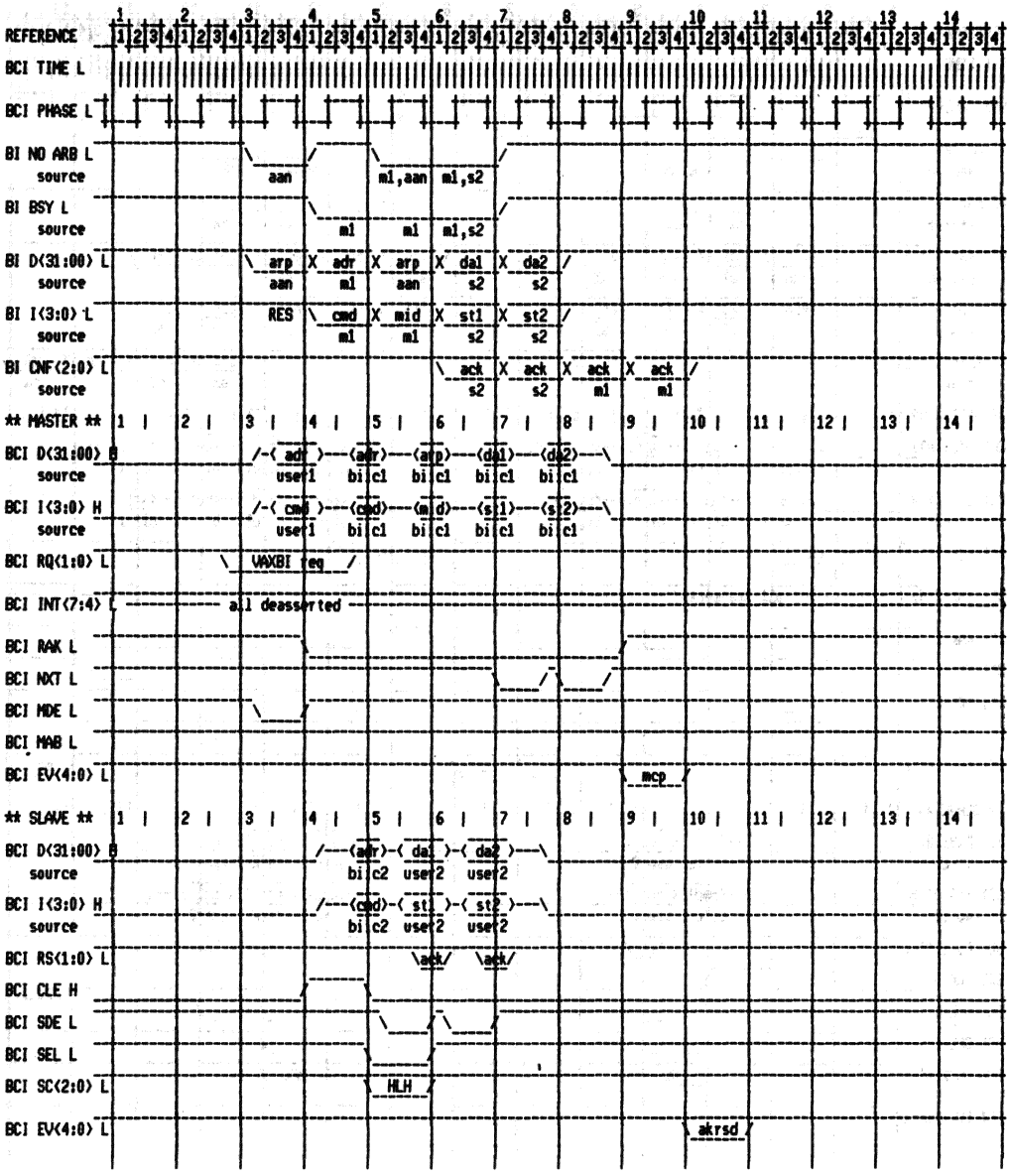


Figure 51 • VAXBI 78732 Quadword Read-type Transaction Timing Sequence

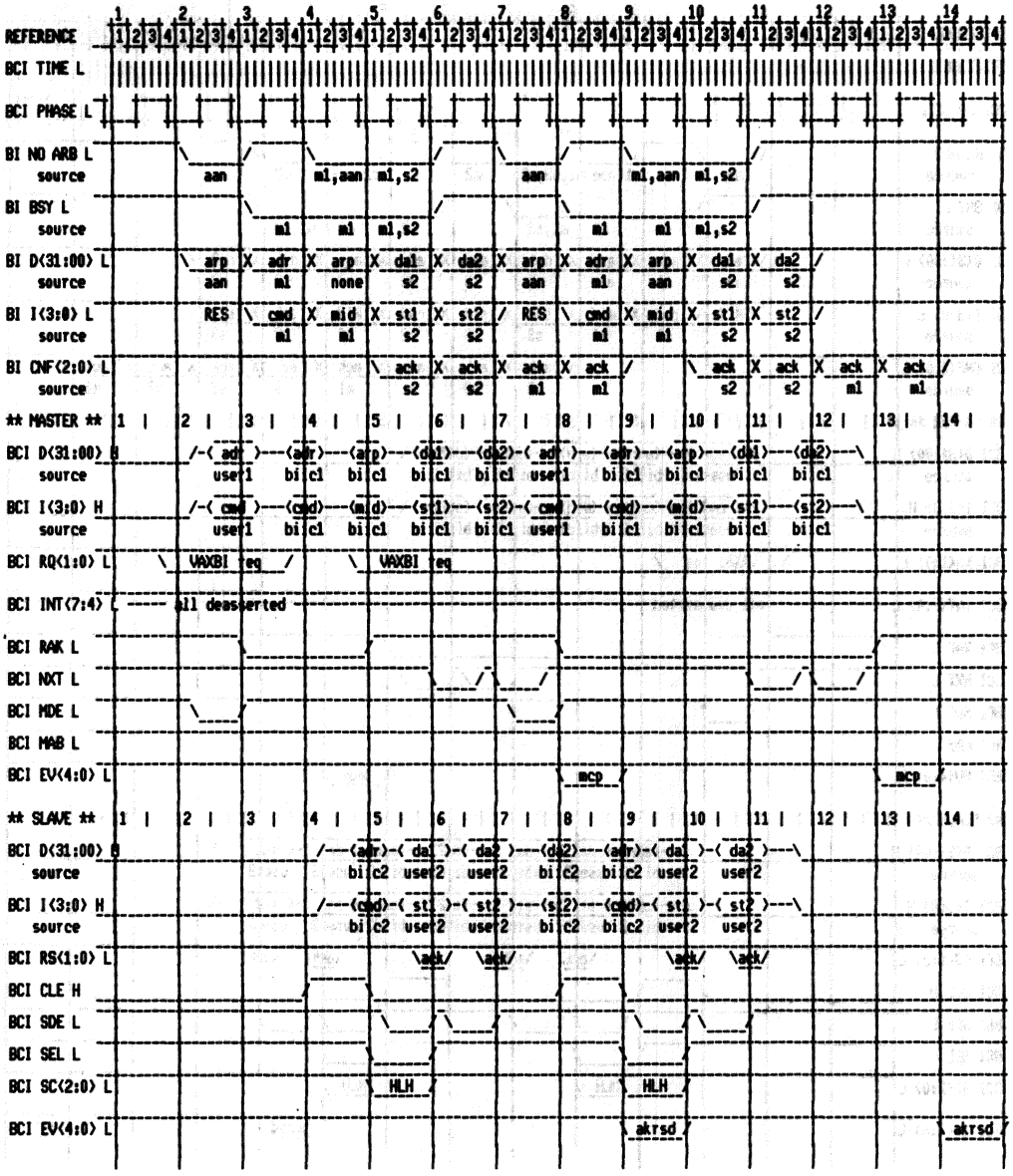


Figure 52 ▪ VAXBI 78732 Quadword Read-type (Pipeline request) Transaction Timing Sequence

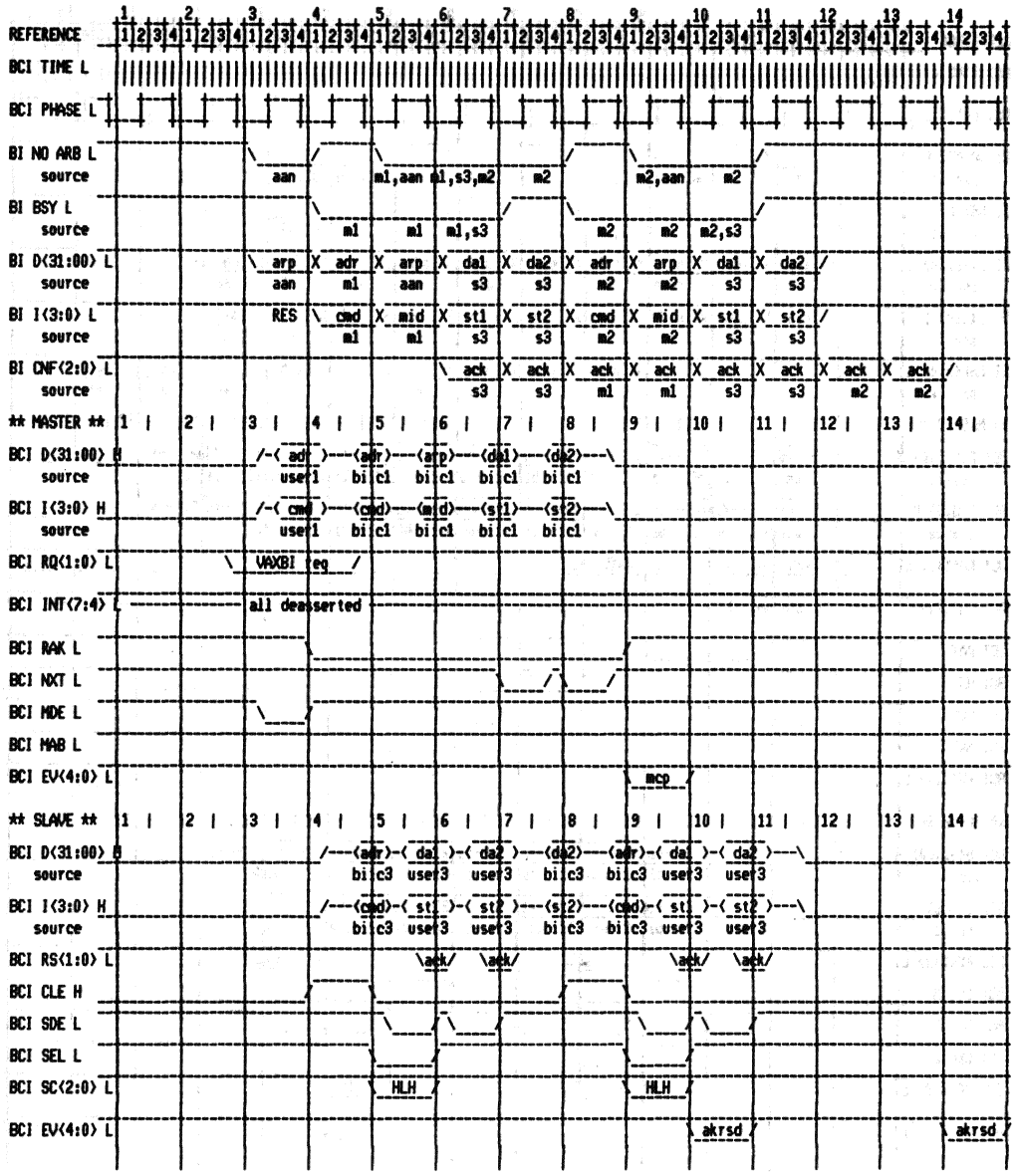


Figure 53 - VAXBI 78732 Quadword Read-type (Pending master) Transaction Timing Sequence

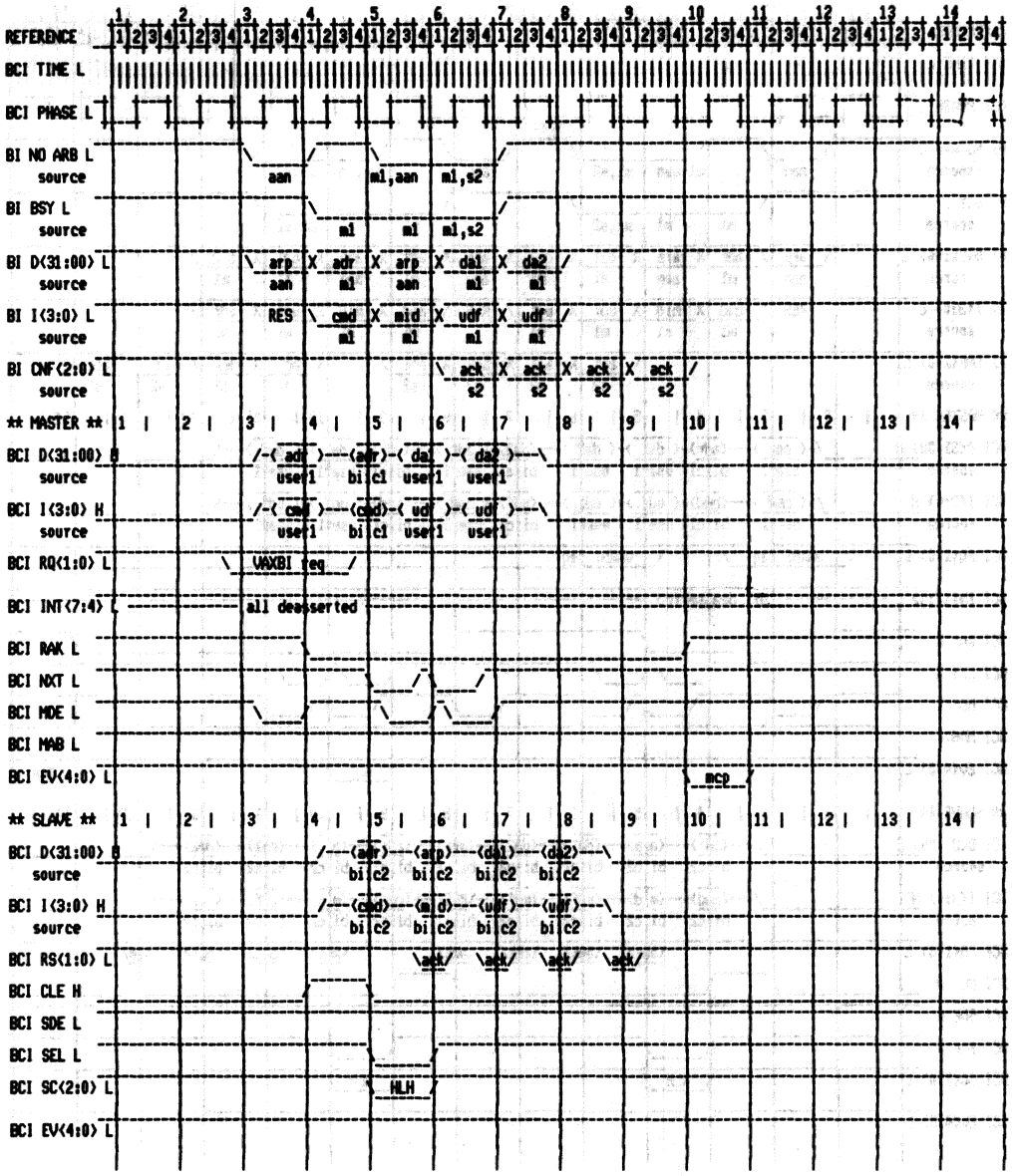


Figure 54 • VAXBI 78732 Quadword Write Transaction Timing Sequence

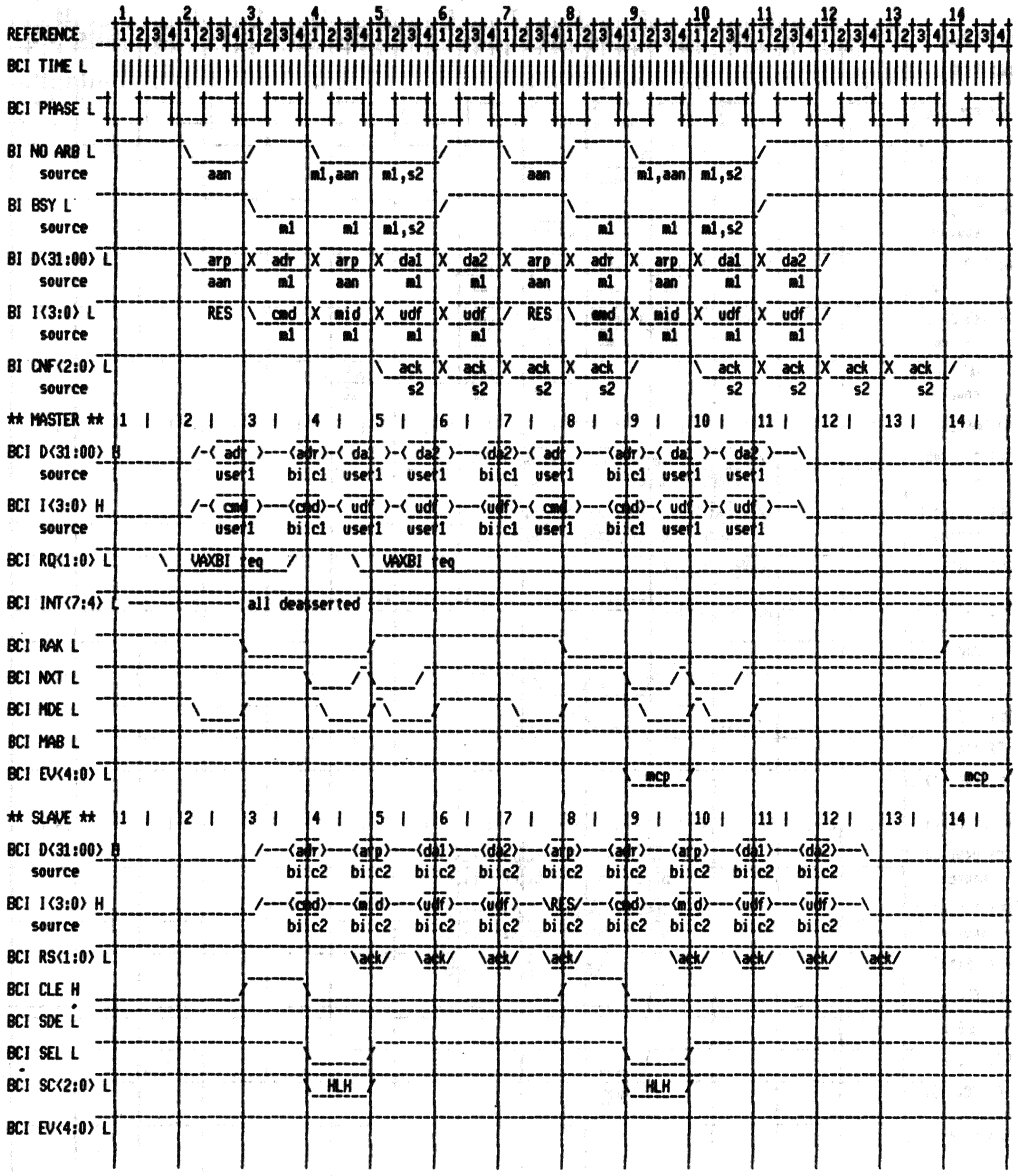


Figure 55 • VAXBI 78732 Quadword Write (Pipeline request) Transaction Timing Sequence

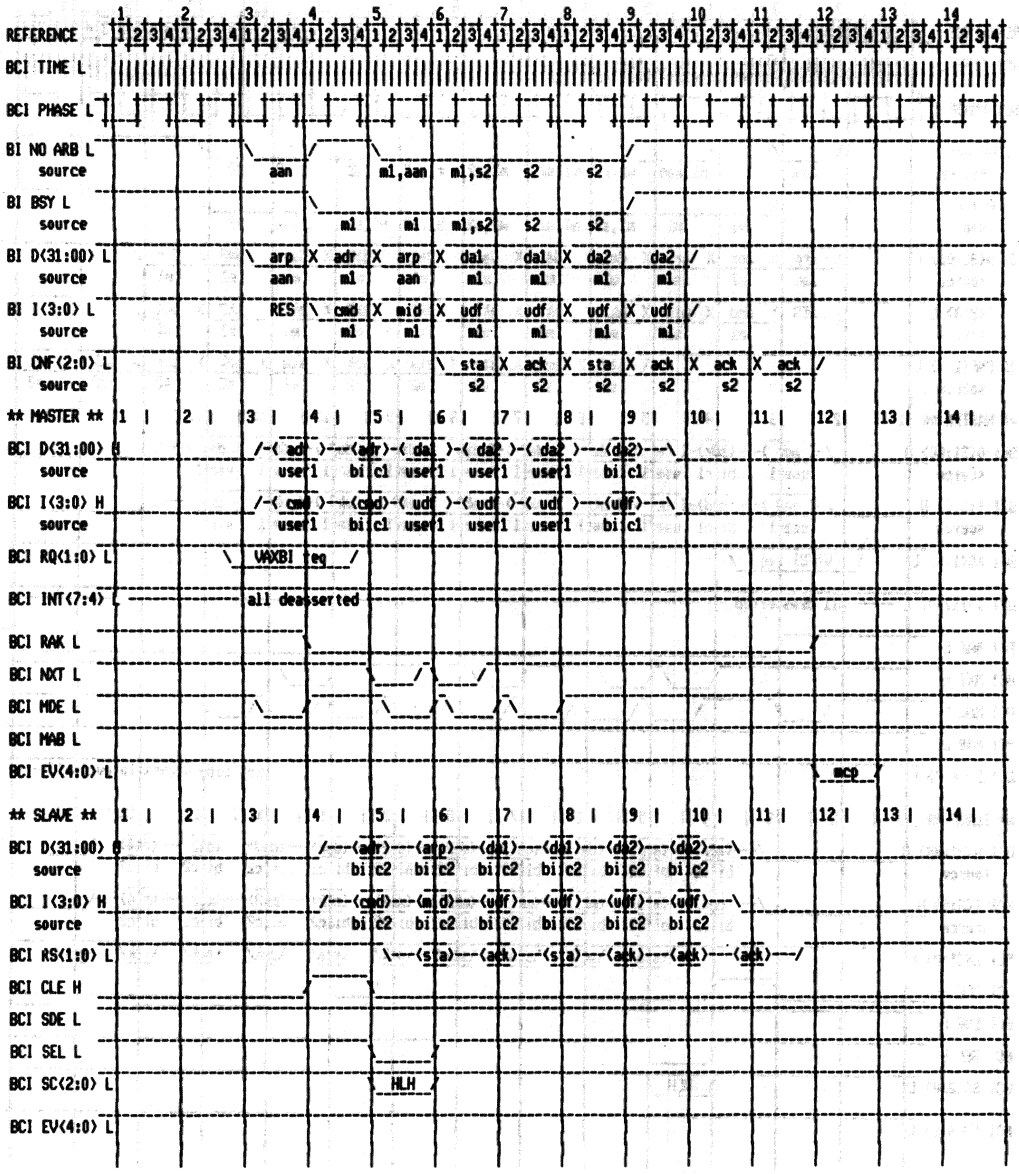


Figure 56 • VAXBI 78732 Quadword Write (Stall for each longword of data) Transaction Timing Sequence

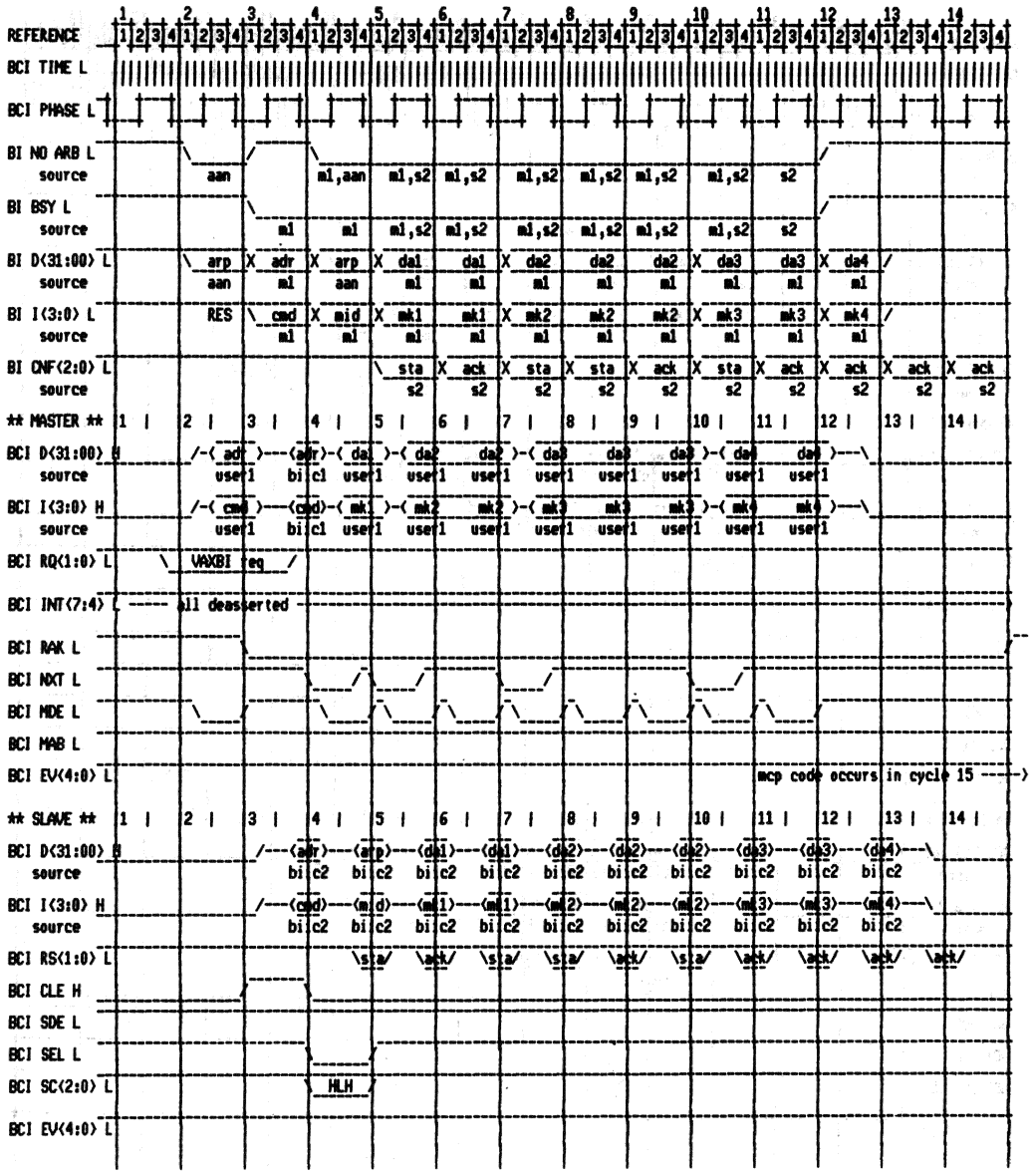


Figure 57 • VAXBI 78732 Octaword WMCI (Variable stall for each longword of data) Transaction Timing Sequence

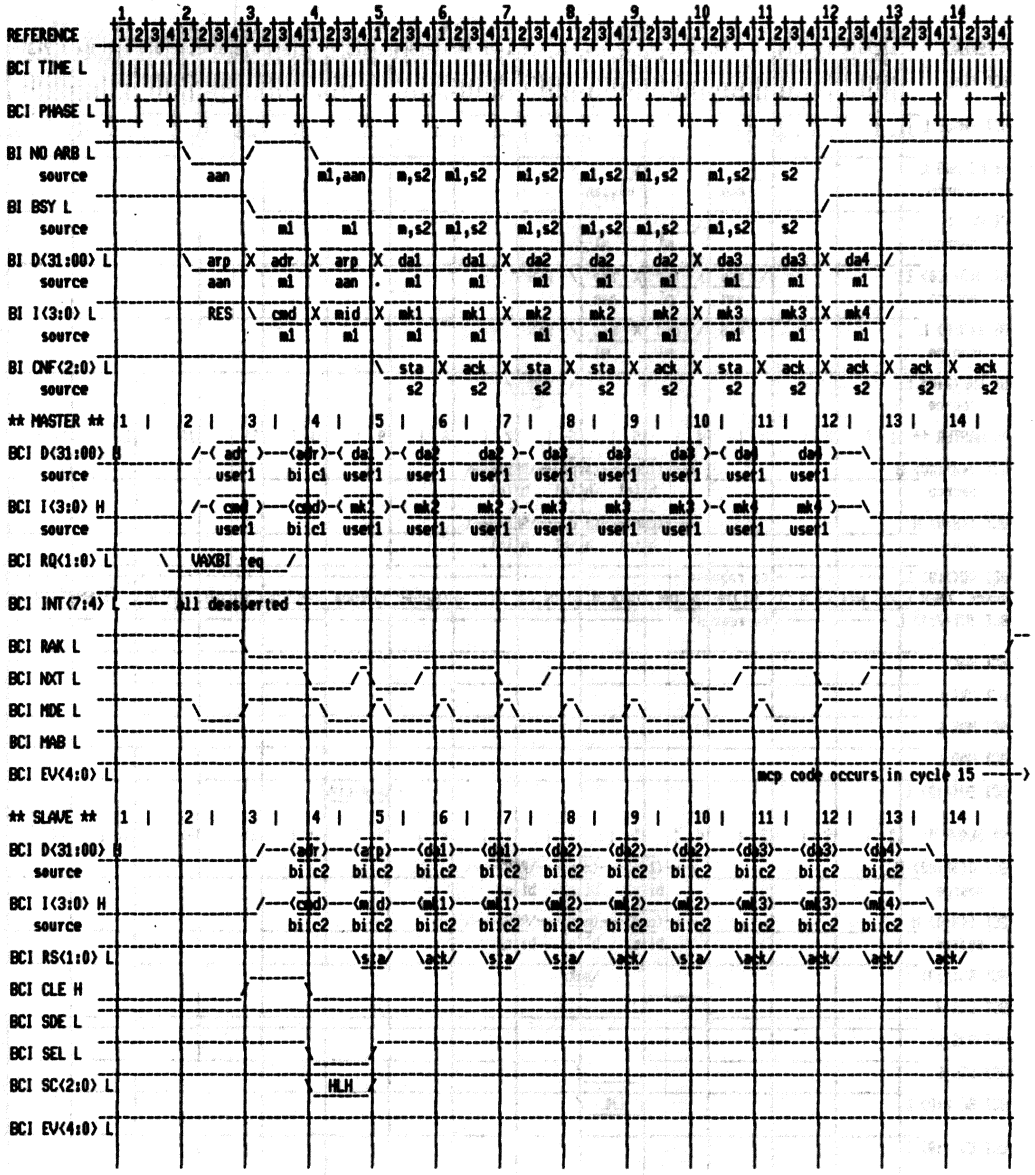


Figure 58 • VAXBI 78732 Octaword WMCI (Variable stall for each longword of data and Pipeline NXT Enable bit set) Transaction Timing Sequence

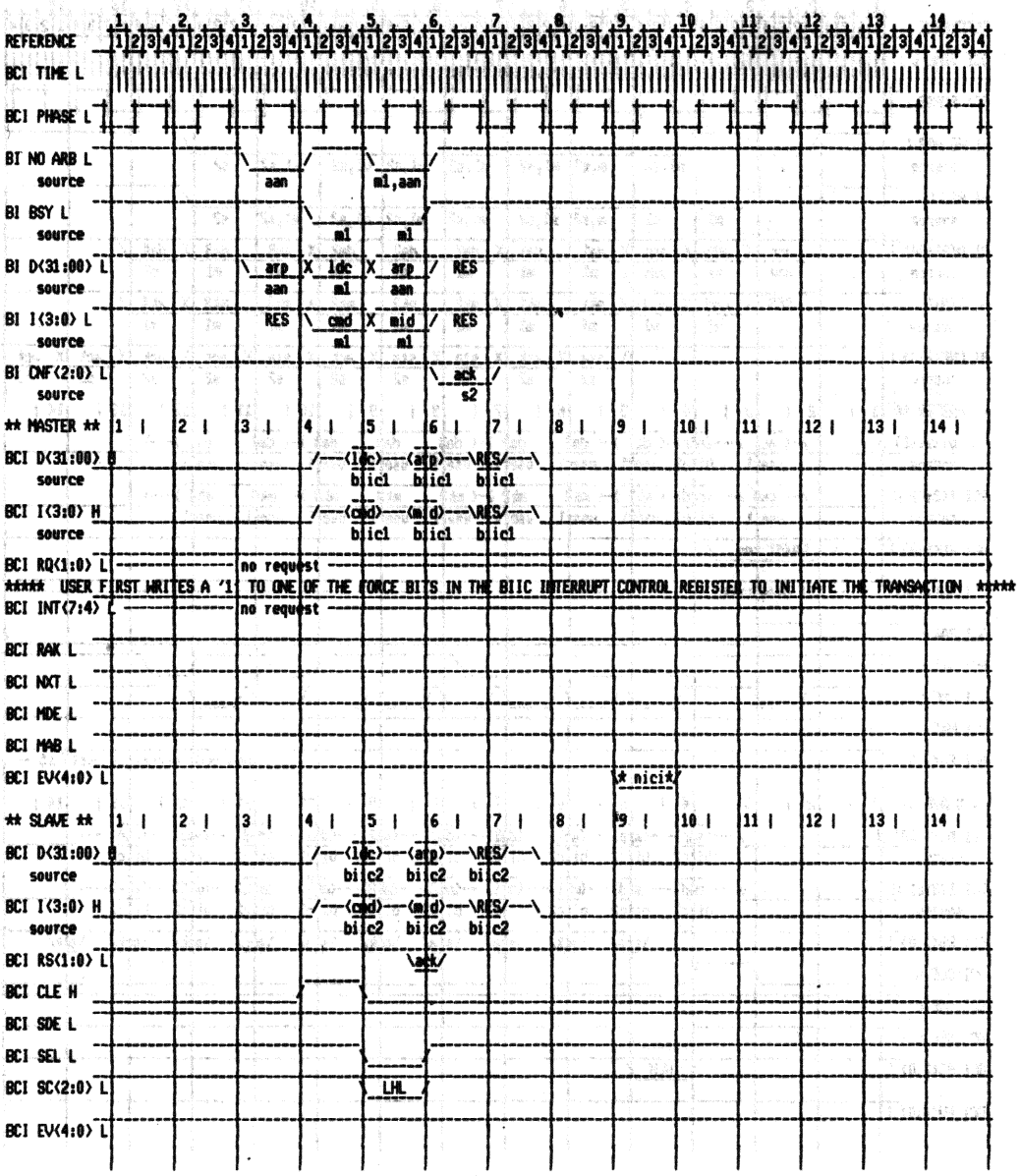


Figure 59 • VAXBI 78732 Force-bit Requested Interrupt Transaction Timing Sequence

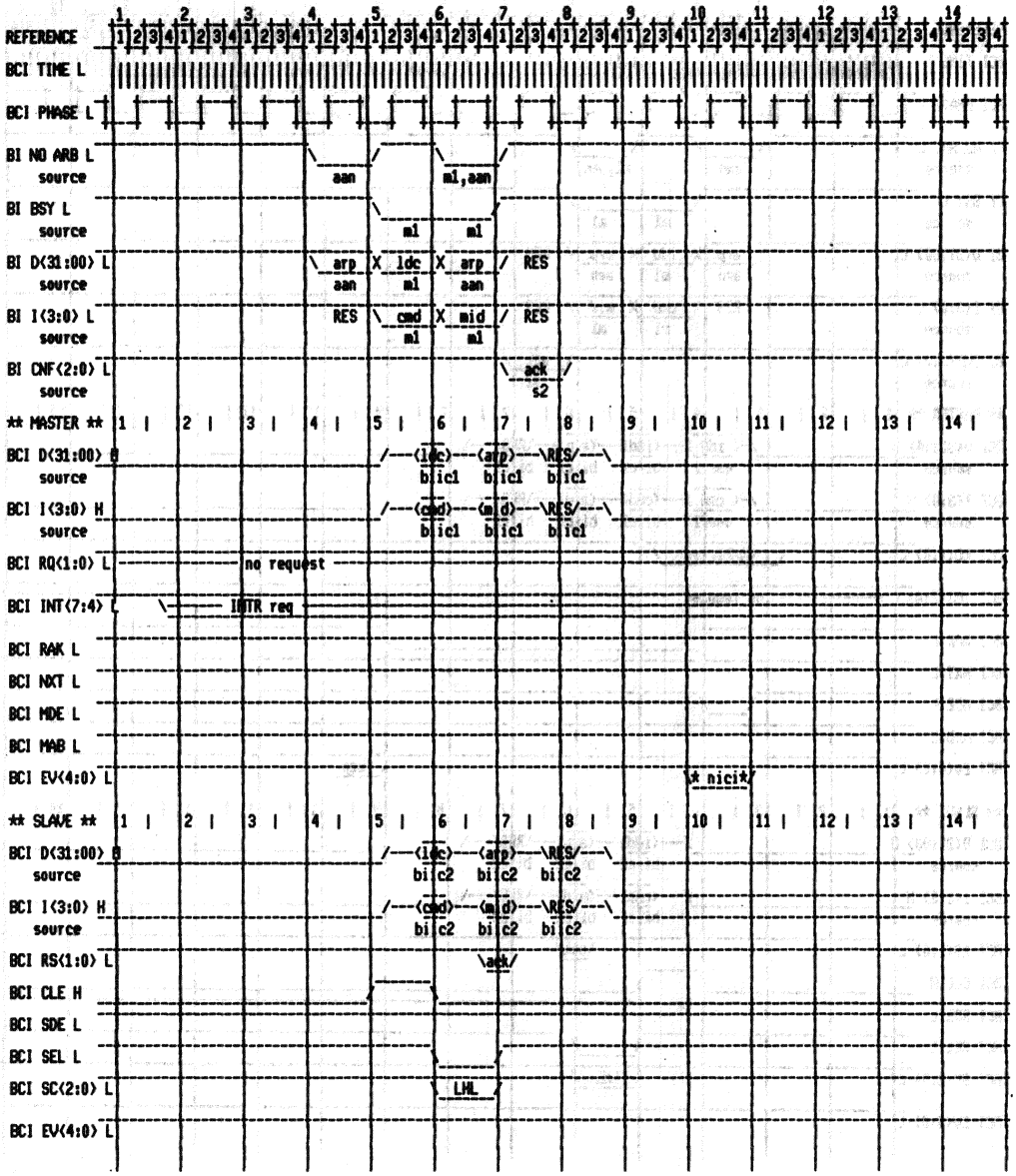


Figure 60 • VAXBI 78732 INT<7:4> Requested Interrupt Transaction Timing Sequence

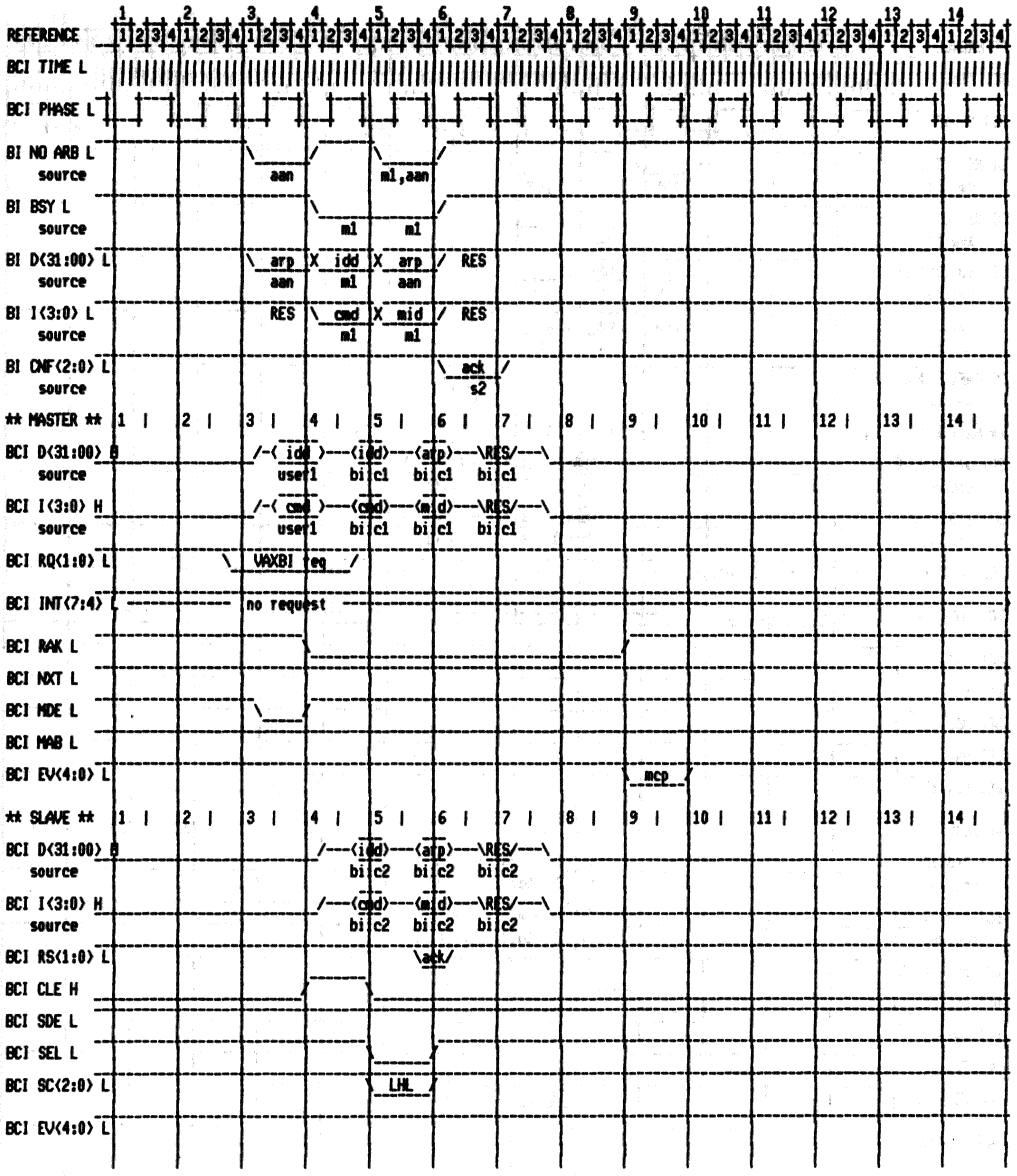


Figure 61 ▪ VAXBI 78732 Master-port Interprocessor Interrupt Transaction Timing Sequence

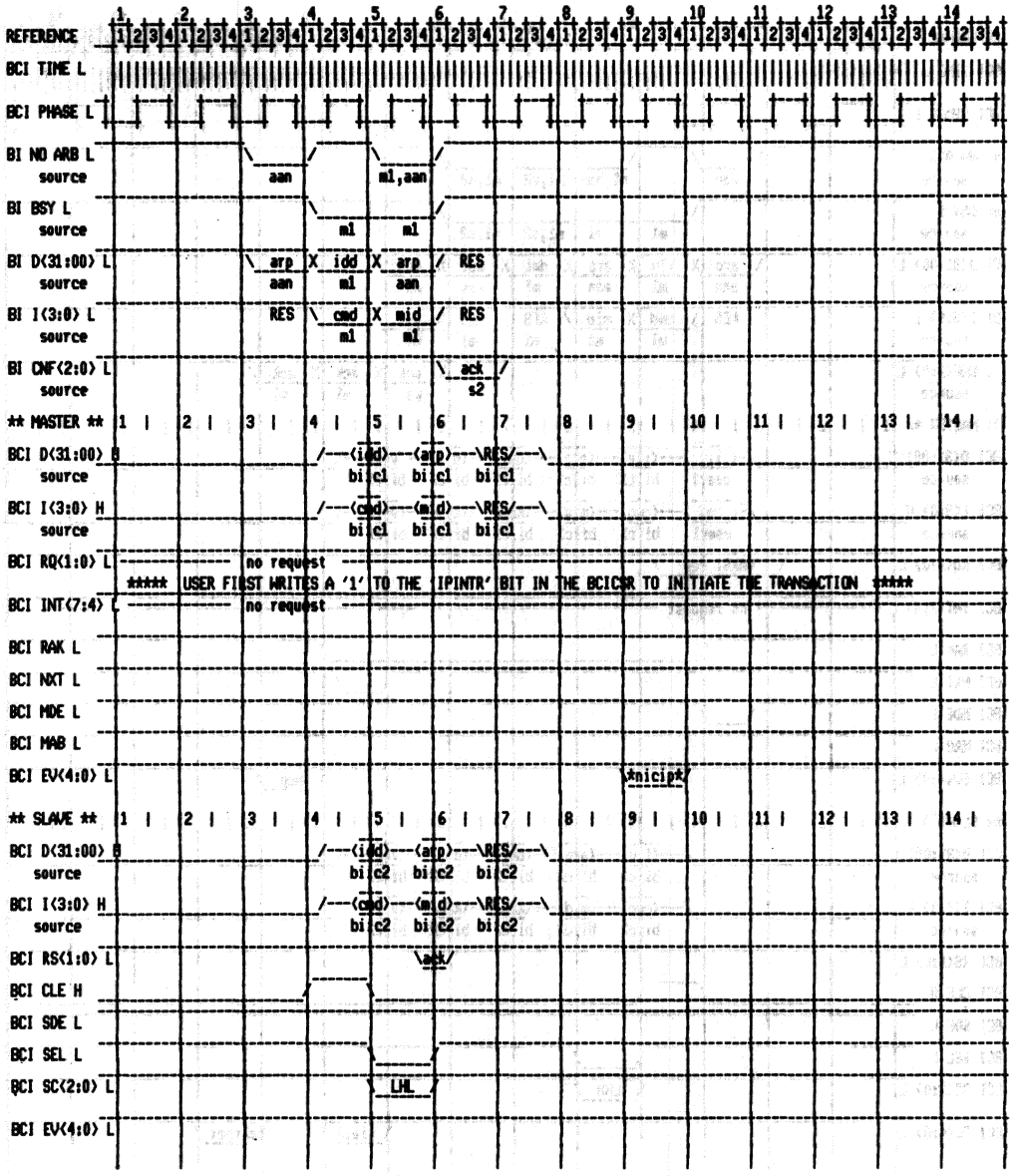


Figure 62 - VAXBI 78732 Force-bit Requested Interprocessor Interrupt Transaction Timing Sequence

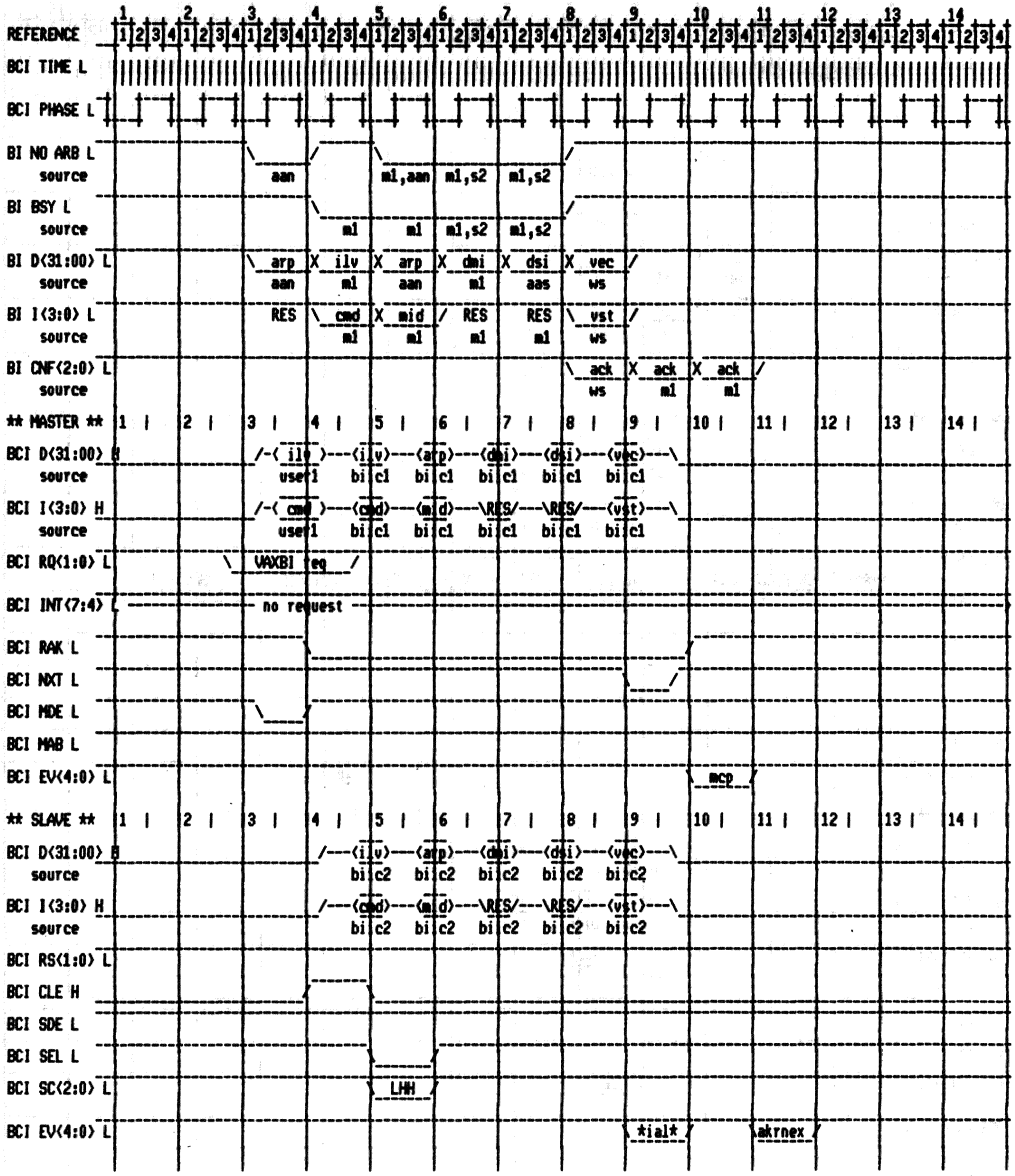


Figure 63 - VAXBI 78732 IDENT (Internal vector) Transaction Timing Sequence

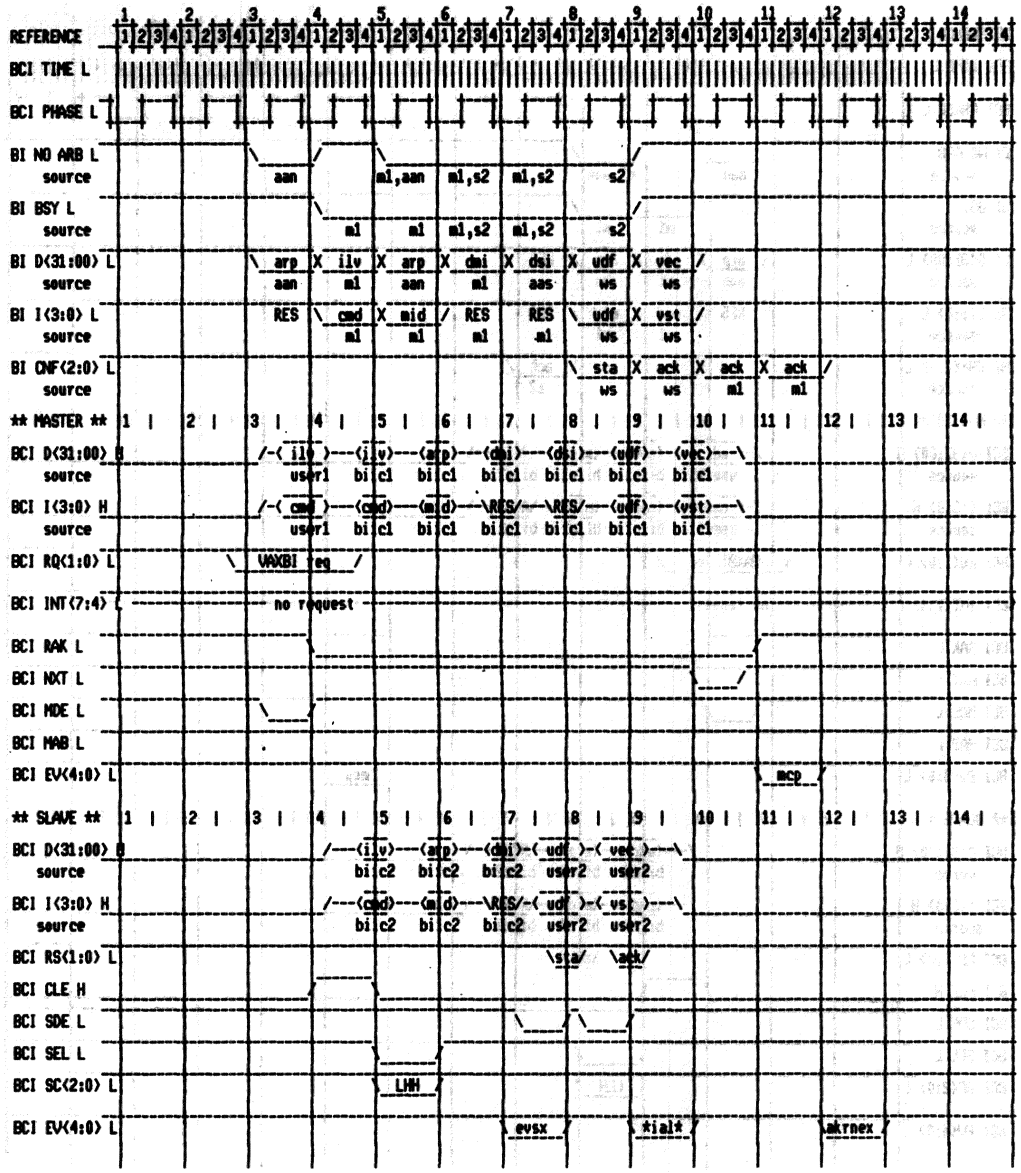


Figure 64 • VAXBI 78732 IDENT (External vector) Transaction Timing Sequence

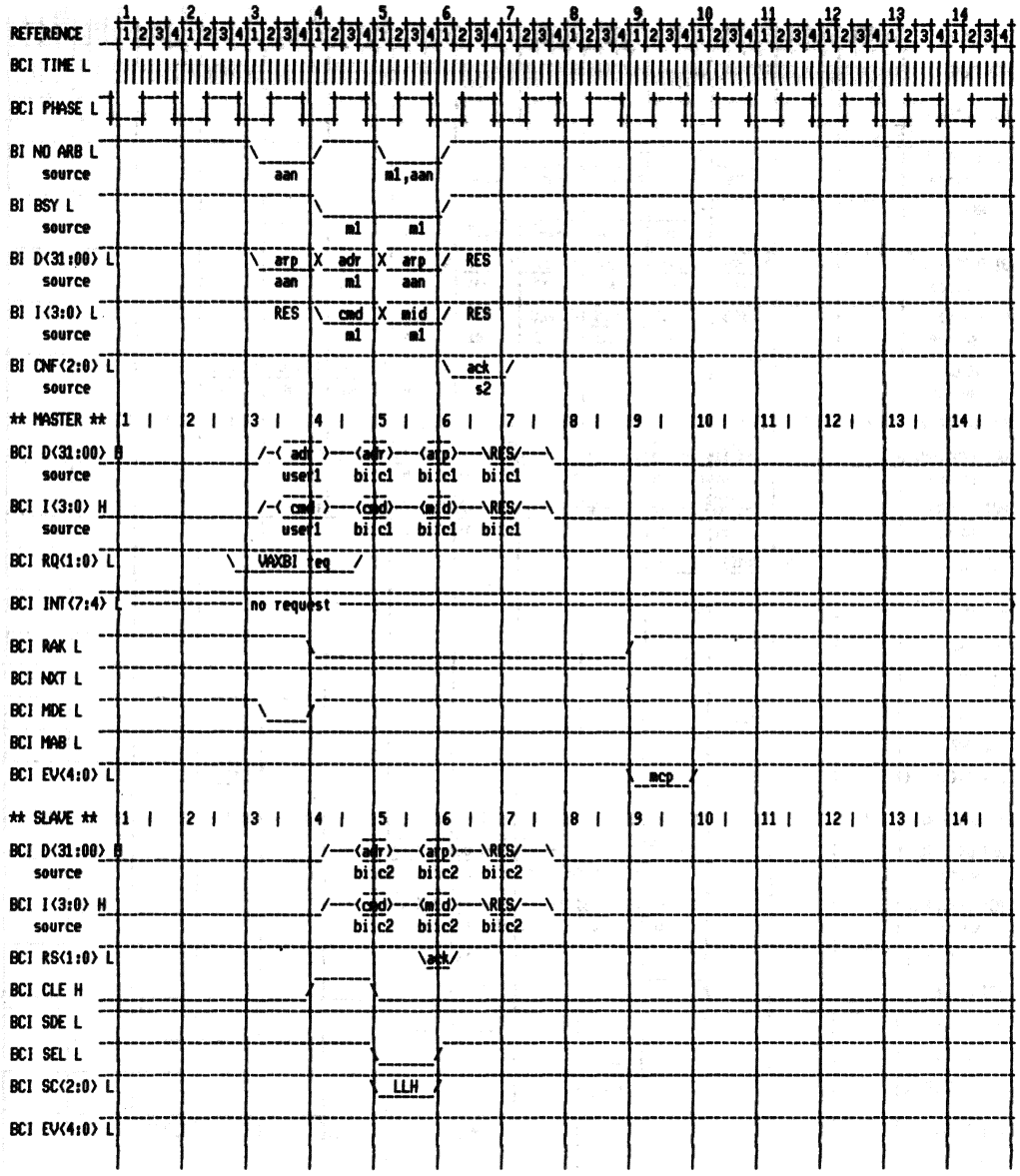


Figure 65 - VAXBI 78732 INVAL Transaction Timing Sequence

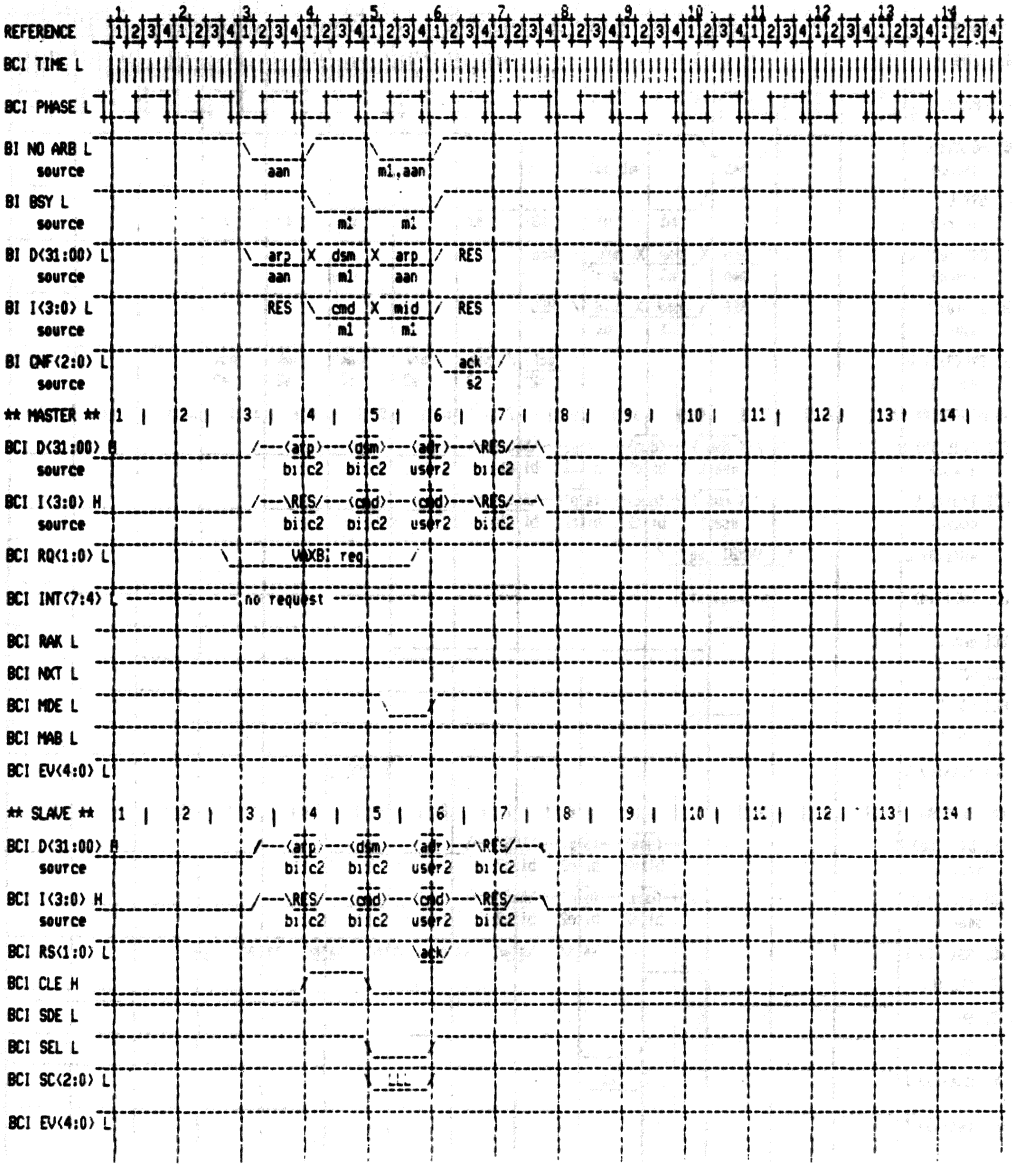


Figure 66 - VAXBI 78732 STOP Transaction Timing Sequence

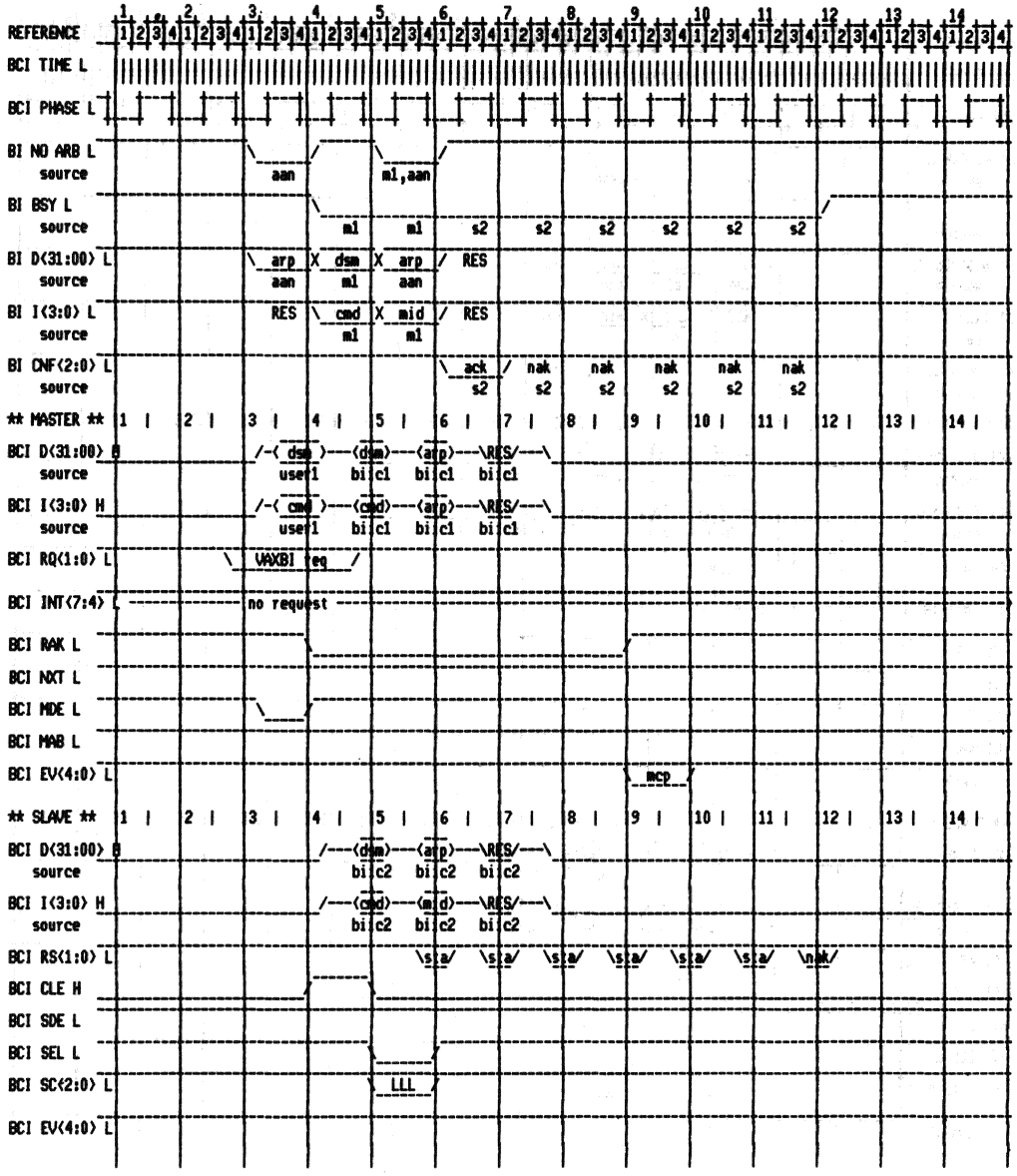


Figure 67 ▪ VAXBI 78732 STOP Transaction (Extension) Timing Sequence

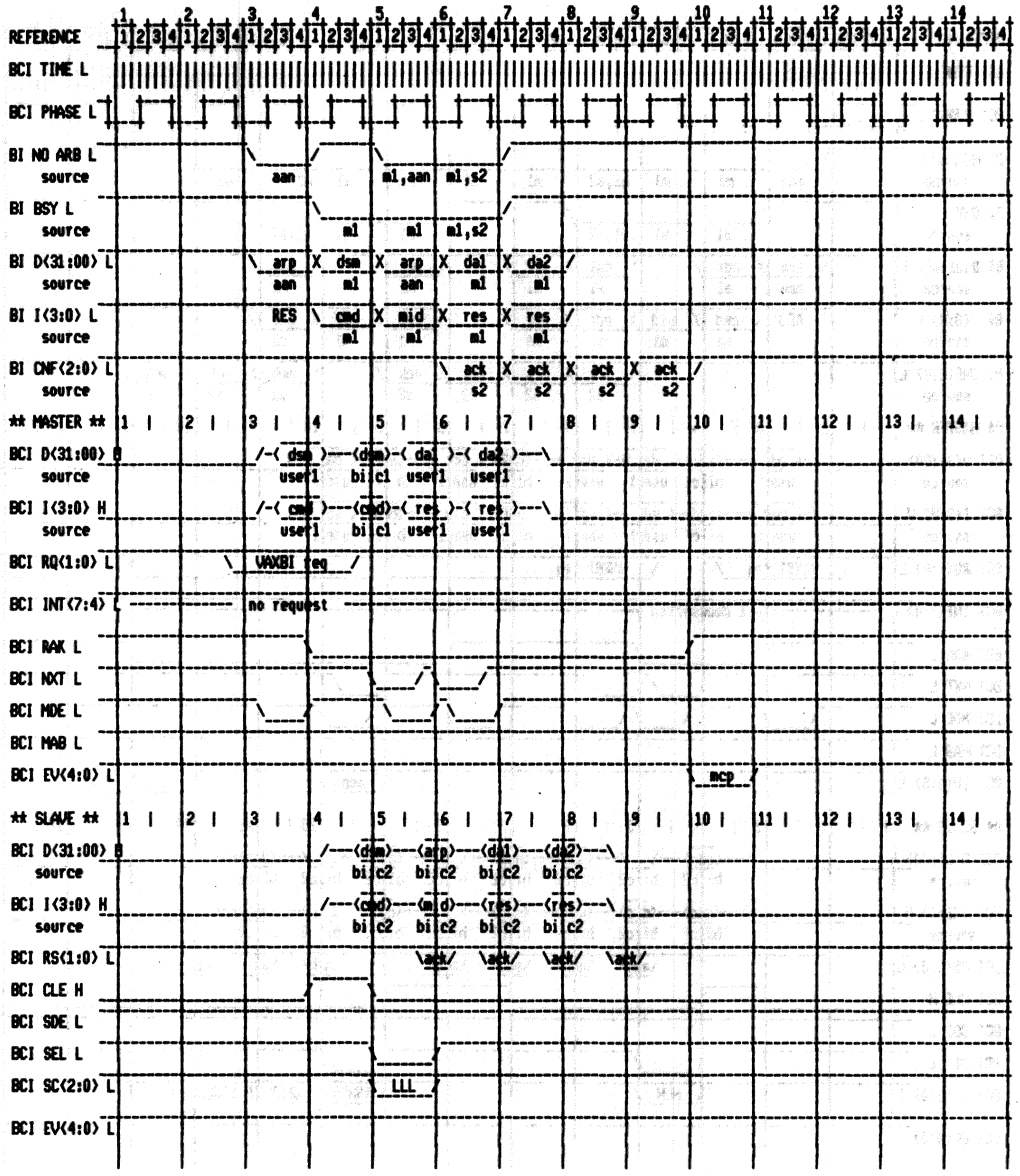


Figure 68 • VAXBI 78732 Quadword BDCST Transaction Timing Sequence

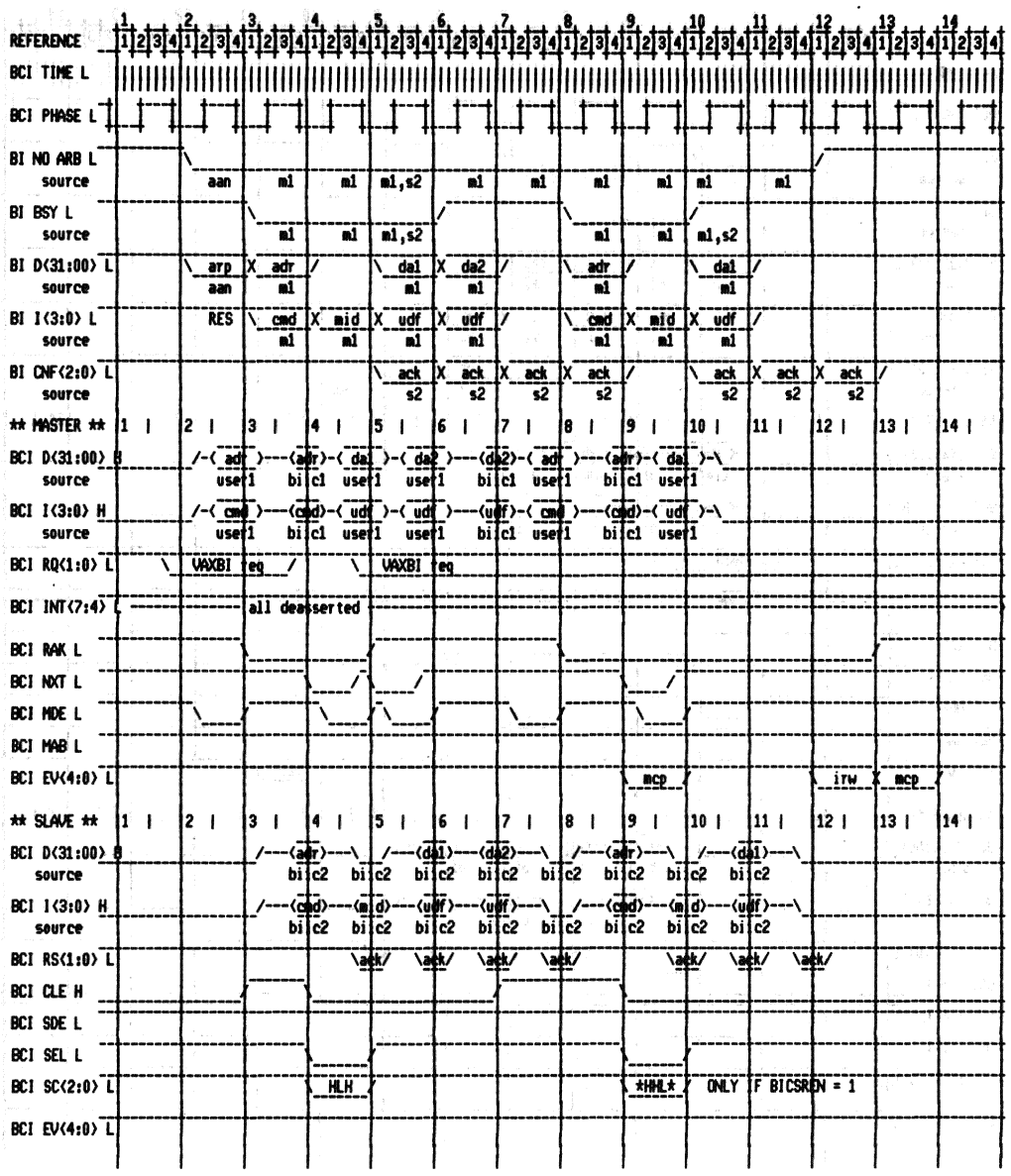


Figure 69 - VAXBI 78732 Burst-mode Write (Pipeline request) Transaction Timing Sequence

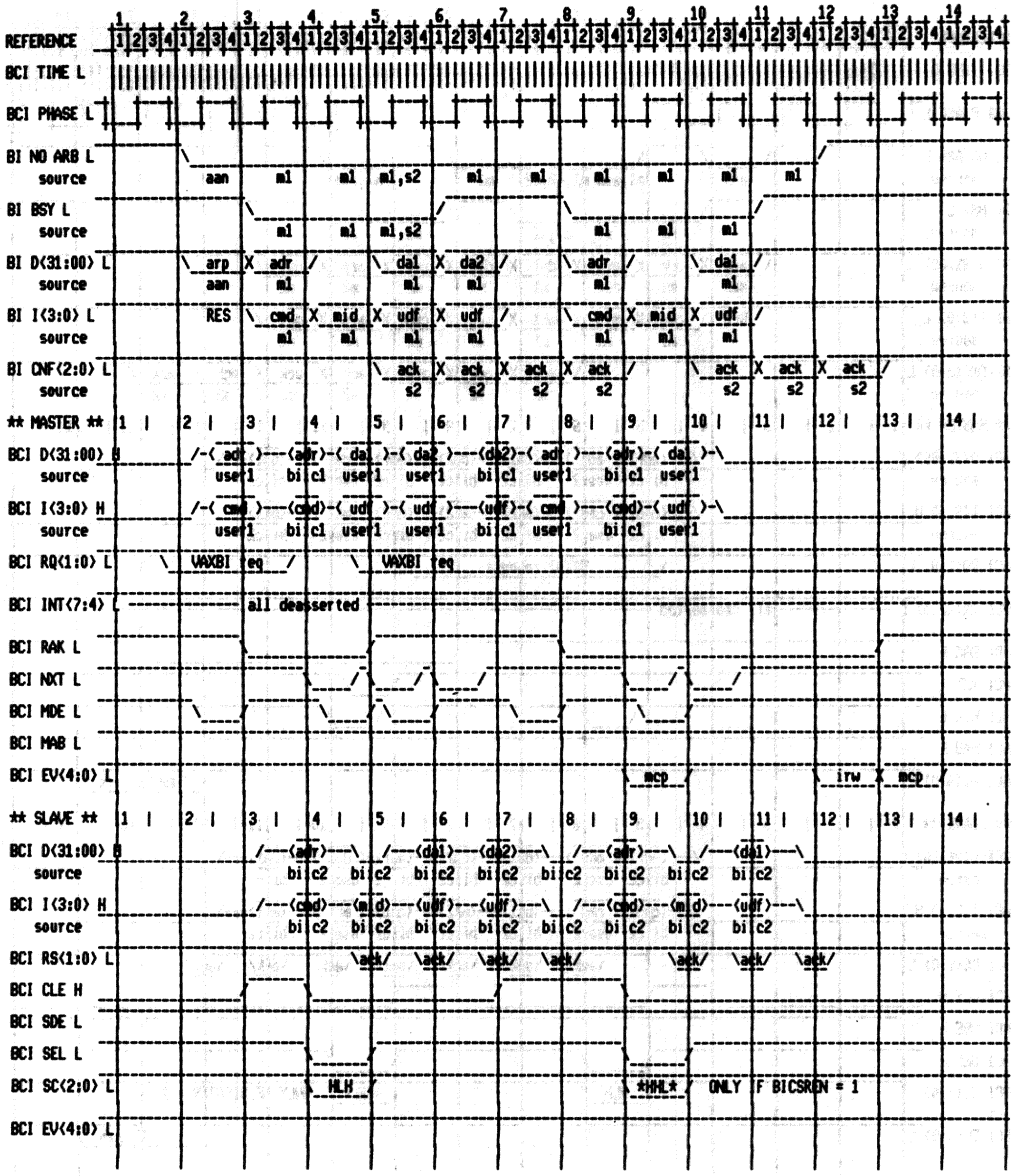


Figure 70 • VAXBI 78732 Burst-mode Write (Pipeline request and Pipeline NXT Enable bit set) Transaction Timing Sequence

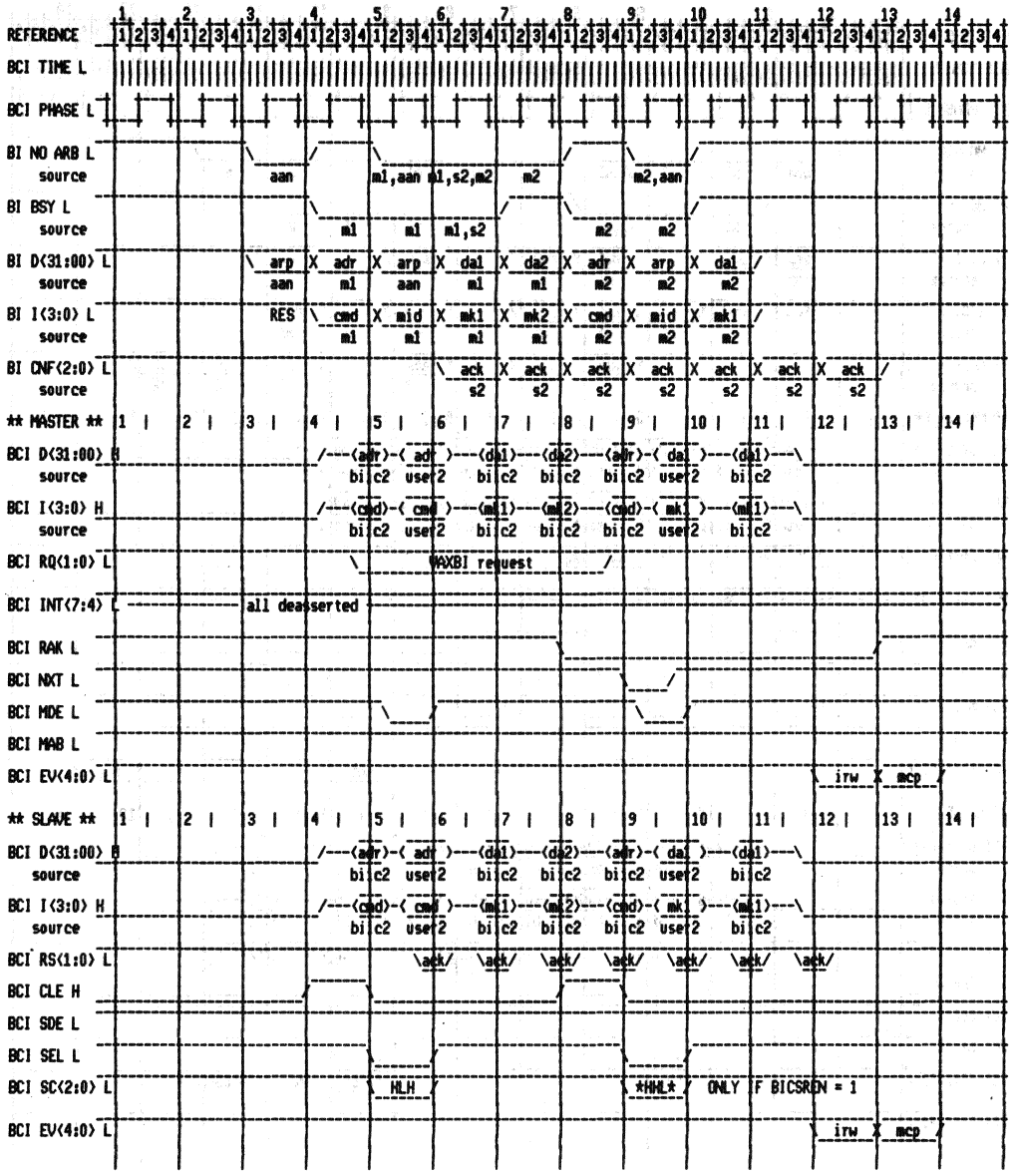


Figure 71 • VAXBI 78732 Special Case 1 Transaction Timing Sequence

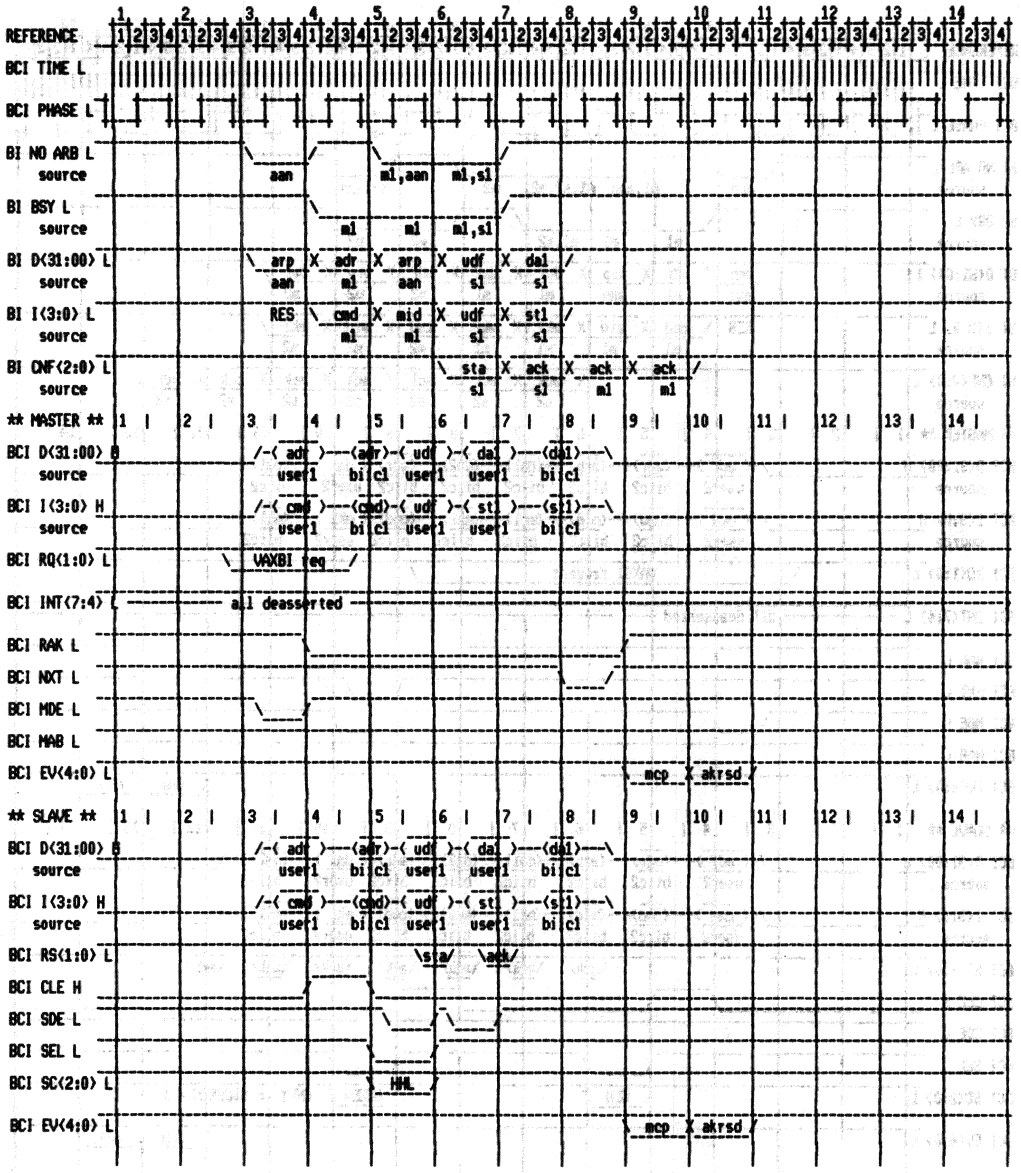


Figure 72 • VAXBI 78732 Special Case 2 Transaction Timing Sequence

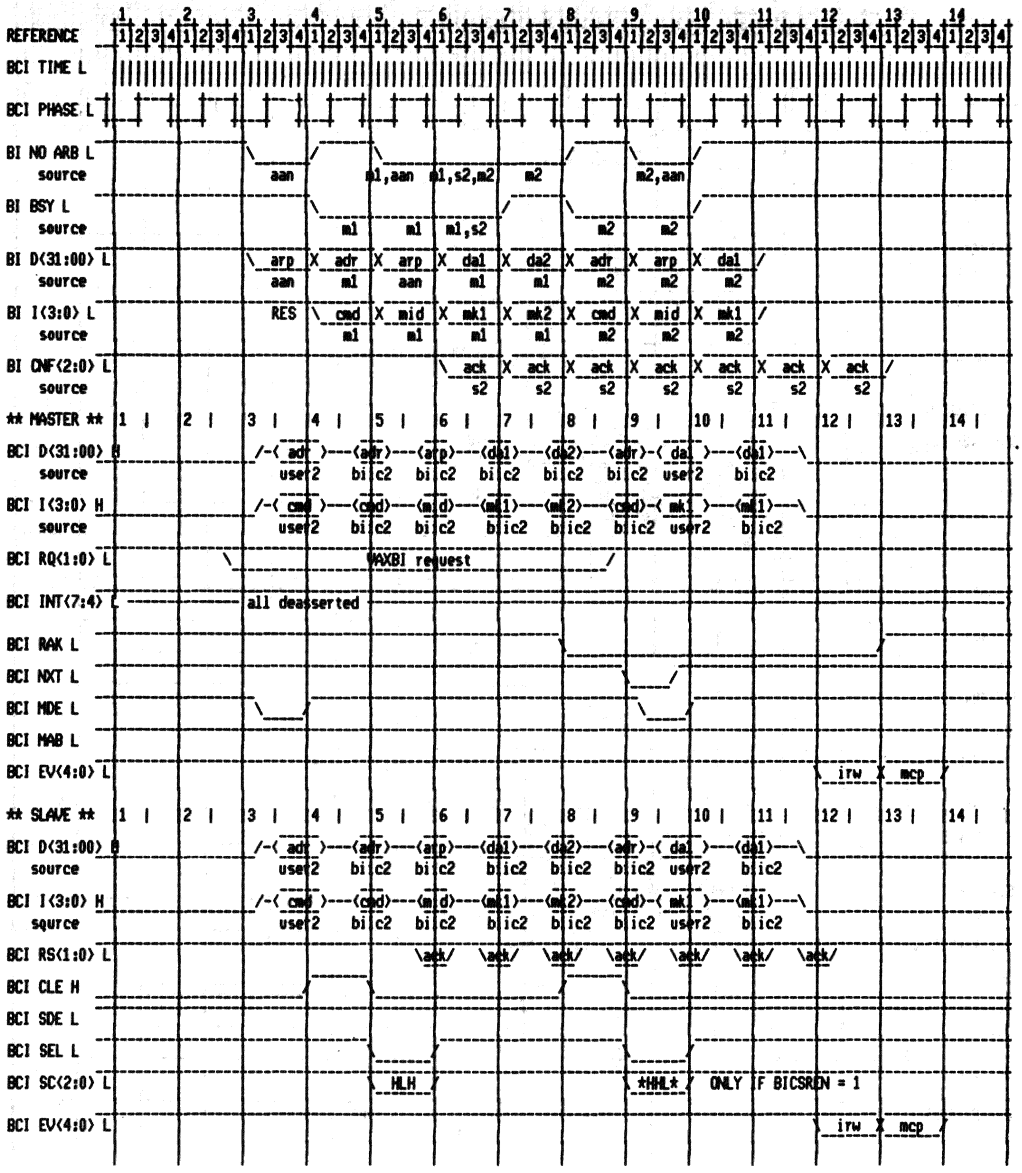


Figure 73 • VAXBI 78732 Special Case 3 Transaction Timing Sequence

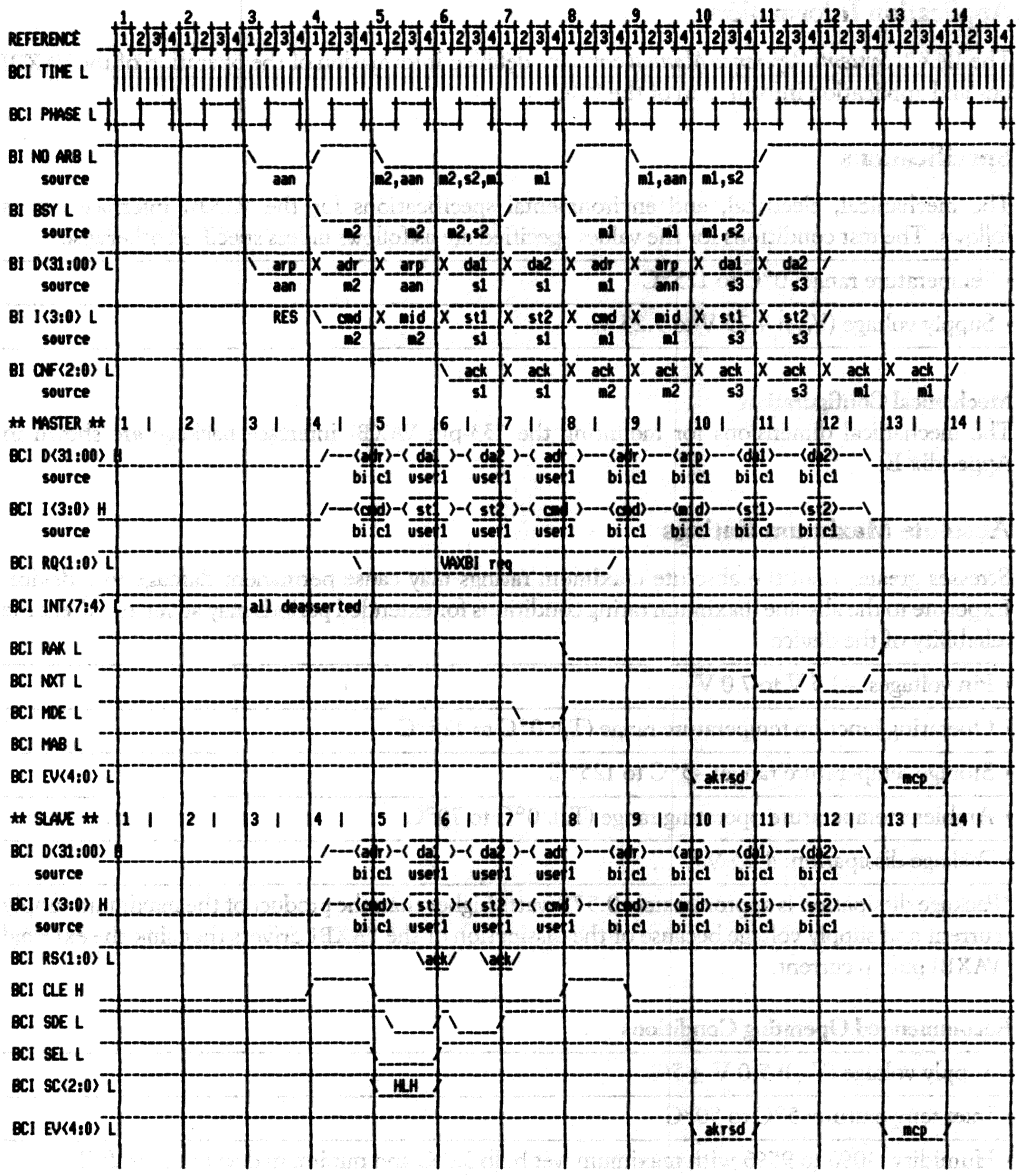


Figure 74 • VAXBI 78732 Special Case 4 Transaction Timing Sequence

Application Information

The *VAXBI Systems Reference Manual* contains detailed information of the operation of the VAXBI bus and application information of the BIIC.

Specifications

The mechanical, electrical, and environmental specifications for the VAXBI interface are as follows. The test conditions for the values specified are as follows unless specified otherwise.

- Temperature range: 0°C to 125°C
- Supply voltage (V_{CC}): 4.75 V to 5.25 V

Mechanical Configuration

The mechanical dimensions for mounting the 133-pin VAXBI interface package are shown in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to a device. Exposure to the absolue maximum rating conditions for extended periods may adversely affect the reliability of the device.

- Pin voltages: −1.5 V to 7.0 V
- Operating junction temperature range (T_J): 0°C to 125°C
- Storage temperature range: −5°C to 125°C
- Ambient temperature operating range (T_A): 0°C to 70°C
- Package dissipation: 4.25 W*

*Package dissipation is approximately 0.575 watts higher than the product of the maximum supply current and supply voltage because of the dissipation of the VAXBI drivers that sink the external VAXBI pullup current.

Recommended Operating Conditions

- Supply voltage (V_{CC}): 5.0 V ±5%
- Inlet temperature: 5°C to 50°C
- Humidity: 10% to 95% with maximum wet bulb 32°C and minimum dew point of 2°C
- Altitude: 0 to 2.4 kilometers
- Airflow: 200 linear feet per minute

dc Electrical Characteristics

Table 46 contains the dc electrical parameters for the input and outputs pins of the VAXBI interface chip. Unless otherwise specified, all specifications are at $T_J = 0$ to 125 degrees C and $V_{CC} = 4.75$ to 5.25 V.

Table 46 • VAXBI 78732 dc Input and Output Parameters

Symbol	Parameter	Min.	Max.	Unit	Test Condition
BCI I_I	Input current[1]	0.33	0.10	mA	$0.2 < V_{in} < 5.25$ V $0 < V_{CC} < 5.25$ V
BCI I_{ID}	Input current while $\overline{\text{BCI DC LO}}$ is asserted[2]	— —	−0.25 −1.0	mA mA	$V_{in} = 2.4$ V $V_{in} = 0.5$ V
BCI I_{OH}	Output high current (except $\overline{\text{BCI DC LO}}$)	−400	—	µA	$V_{out} = \text{BCI } V_{OH}$
BCI I_{OHD}	Output high current (only for $\overline{\text{BCI DC LO}}$)	−5.4	—	mA	$V_{out} = \text{BCI } V_{OH}$
BCI I_{OL}	Output low current	4.0	—	mA	$V_{out} = \text{BCI } V_{OL}$
BCI I_{OLD}	Output low current (only for $\overline{\text{BCI DC LO}}$ when $V_{CC} < V_{CC}$ min.)	100	—	µA	$V_{out} = \text{BCI } V_{OL}$ $0 < V_{CC} < 4.75$ V
BCI V_{IL}	Input low voltage	−1.0	0.8	V	
BCI V_{IH}	Input high voltage (except $\overline{\text{BCI TIME}}$) $\overline{\text{BCI TIME}}$	2.0 2.4	— —	V V	
BCI V_{OH}	Output high voltage	2.7	—	V	$I_{out} = \text{BCI } I_{OH}$
BCI V_{OL}	Output low voltage	—	0.5	V	$I_{out} = \text{BCI } I_{OL}$
BCI I_{OS}	Short circuit output current	—	−150	mA	Not more than one at a time or > 1 second
BCI C_{IO}	Pin capacitance	—	10	pF	$0 < V_{IO} < V_{CC}$
BI I_I	Input current[1]	−270	30	µA	$0 < V_{IO} < \text{BI } V_{OH}$
BI I_{OL}	Output low current	21	—	mA	$V_{out} = \text{BI } V_{OL}$
BI V_{OL}	Output low voltage	—	0.6	V	$I_{out} = \text{BI } I_{OL}$
BI V_{OH}	Bus voltage high	2.3	3.5	V	These are bus terminator specifications
BI V_{IH}	Input high voltage	1.95	—	V	
BI V_{HHY}	Input high voltage (hysteresis voltage)[3]	1.45	—	V	
BI V_{IL}	Input low voltage	−1.0	1.1	V	
BI V_{LHY}	Input low voltage (hysteresis voltage)[3]	—	1.4	V	

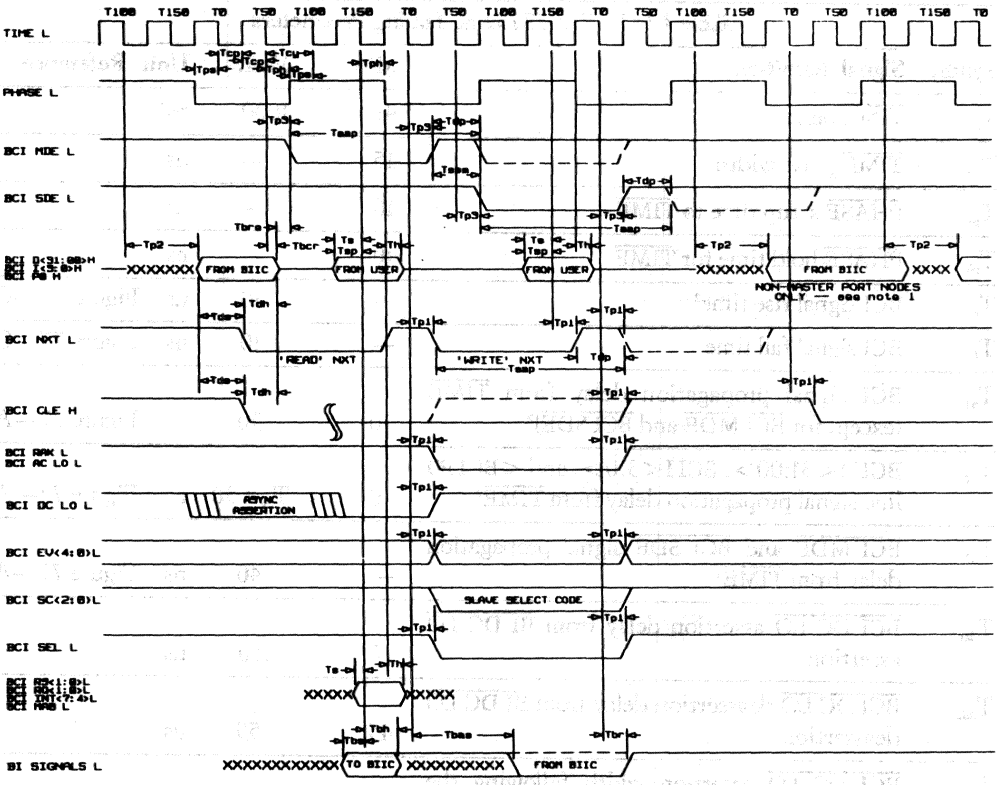
Symbol	Parameter	Min.	Max.	Unit	Test Condition
BI C_{IO}	Pin capacitance[4]	—	6.0	pF	$V_{IO}=2.5$ V
I_{CC}	Power supply current	—	700	mA	$I_{out}=0, T_J=27°C$[5]
		—	530	mA	$I_{out}=0, T_J=85°C$

[1]These I_I specifications apply only when data is not being driven by the output under test.
[2]While $\overline{\text{BCI DC LO}}$ is asserted, the BCI D and P lines are internally pulled up. While pulled up, these lines can source a minimum of 250 μA at 2.4 V. The user interface logic must sink a minimum of 1.0 mA at 0.5 V to drive any of these lines low when $\overline{\text{BCI DC LO}}$ is asserted.
[3]The hysteresis voltage is defined as follows: BI V_{IH}—The BIIC will not detect a change in input state if the input voltage drops to BI V_{HHY} following the application of BI B_{IH}. BI V_{IL}—The BIIC will not detect a change in input state if the input voltage rises to BI V_{LHY} following the application of BI V_{IL}.
[4]The device under test must be powered up during this test and $\overline{\text{BI DC LO}}$ should be asserted, except when measuring C_{IO} for $\overline{\text{BI DC LO}}$.
[5]T_J is junction temperature with ambient temperature (T_A) = 0°C.

ac Timing Specifications

Unless otherwise specified, test conditions are at $T_A=0°C$ to 70°C, $V_{CC}=4.75$ V to 5.25 V, and the BCI signal capacitance load = 50 pF.

Figure 75 shows the ac timing for the BIIC and Table 47 lists the timing parameters for the symbols on the figure. Figure 76 shows the waveforms and symbols used for the measurements of the VAXBI bus. Figure 77 shows the waveforms and symbols used for the measurements of the BCI bus. Figure 78 shows the load circuit for the VAXBI bus driver measurements. Figure 79 shows the load circuit for the BCI bus driver measurements.



Note 1: This extended BCI data window is only usable in designs in which BCI MDE L never asserts (i.e., slave port only designs such as memory), and even in these designs it only applies to data cycles (i.e., command/address cycles still have the narrow window).

Figure 75 • VAXBI 78732 ac Timing Sequence

Table 47 • VAXBI 78732 ac Timing Parameters

Symbol	Signal definition	Min.	Max.	Unit	Reference
T_{cy}	$\overline{TIME}$ period[1]	49	1000	ns	
T_{cp}	$\overline{TIME}$ pulse width	15	—	ns	
T_{ps}	$\overline{PHASE}$ setup time to $\overline{TIME}$	10	—	ns	
T_{ph}	$\overline{PHASE}$ hold time for $\overline{TIME}$	10	—	ns	
T_r	BCI signal rise time[2]	—	10	ns	Figure 77—C
T_f	BCI signal fall time[2]	—	10	ns	Figure 77—C
T_{p1}	BCI signal propagation delay from $\overline{TIME}$ (except for $\overline{BCI\ MDE}$ and $\overline{BCI\ SDE}$)[2]	0	30	ns	Figure 77—A
T_{p2}	BCI D<31:00>, BCI I<3:0>, and <BCI P0 line signal propagation delay from $\overline{TIME}$[2]	0	$T_{cy}+30$	ns	Figure 77—A
T_{p3}	$\overline{BCI\ MDE}$ and $\overline{BCI\ SDE}$ signal propagation delay from $\overline{TIME}$[2]	—	40	ns	Figure 77—A
T_{das}	$\overline{BCI\ DC\ LO}$ assertion delay from $\overline{BI\ DC\ LO}$ assertion	0	150	ns	
T_{dde}	$\overline{BCI\ DC\ LO}$ deassertion delay from $\overline{BI\ DC\ LO}$ deassertion	45	55	us	
T_{nrw}	$\overline{BCI\ DC\ LO}$ assertion width following the setting of the NRST bit	45	55	us	
T_{sp}	BCI D<31:00> and BCI I<3:0> line setup time with BIIC configured for BIIC-generated parity	20	—	ns	Figure 77—B
T_s	BCI signal setup time with BIIC configured for user interface to supply parity	0	—	ns	Figure 77—B
T_h	BCI signal hold time from $\overline{TIME}$	15	—	ns	Figure 77—B
T_{ber}	BCI D<31:00> and BCI I<3:0> release time from $\overline{TIME}$	0	40	ns	
T_{bre}	BCI D<31:00> and BCI I<3:0> release time to $\overline{BCI\ MDE}$ or $\overline{BCI\ SDE}$ assertion	0	—	ns	
T_{ssm}	$\overline{BCI\ SDE}$ ($\overline{BCI\ MDE}$) deassertion setup time to $\overline{BCI\ MDE}$ ($\overline{BCI\ SDE}$) assertion[3]	$T_{cy}-15$	—	ns	
T_{dp}	$\overline{BCI\ NXT}$, $\overline{BCI\ MDE}$, and $\overline{BCI\ SDE}$ deassertion pulse width	$T_{cy}-10$	—	ns	
T_{dh}	BCI D<31:00> and BCI I<3:0> hold time from BCI CLE and $\overline{BCI\ NXT}$	$T_{cy}-25$	—	ns	

Symbol	Signal definition	Min.	Max.	Unit	Reference
T_{ds}	BCI D<31:00> and BCI I<3:0> setup time to BCI CLE and $\overline{\text{BCI NXT}}$	T_{cy}—25	—	ns	
T_{asp}	Minimum assertion to assertion period for $\overline{\text{BCI NXT}}$, $\overline{\text{BCI MDE}}$, and $\overline{\text{BCI SDE}}$[4]	190	—	ns	
T_{bs}[5]	BI signal setup time to $\overline{\text{TIME}}$	20	—	ns	Figure 76—B
T_{bh}	BI signal hold time from $\overline{\text{TIME}}$	20	—	ns	Figure 76—B
T_{bas}	BI signal assertion delay from $\overline{\text{TIME}}$	0	85	ns	Figure 76—A
T_{br}	BI signal release time from $\overline{\text{TIME}}$	0	85	ns	Figure 76—A
T_{bif}	BI signal fall time (C_l=410 pF)	20	—	ns	Figure 76—C

[1]The minimum specification allows for proper BIIC operation in environments with up to ±1.0 ns of clock period noise jitter.

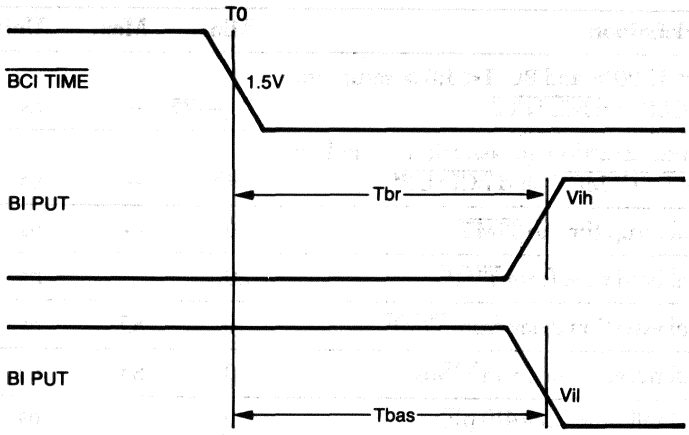
[2]Parameters T_r, T_f, T_{p1}, T_{p2}, and T_{p3} are for 50-pF loads. The following equations describe the degradation in rise and fall time and propagation delays for a BCI line with between 50 and 100 pF load.

T_r (50 pF < C_l < 100 pF) = T_r (50 pF) + C_l/17.8—2.8.
T_f (50 pF < C_l < 100 pF) = T_f (50 pF) + C_l/17.8—2.8.
T_{p1} (50 pF < C_l < 100 pF) = T_{p1} (50 pF) + C_l/5.5—9.1.
T_{p2} (50 pF < C_l < 100 pF) = T_{p2} (50 pF) + C_l/5.5—9.1.
T_{p3} (50 pF < C_l < 100 pF) = T_{p3} (50 pF) + C_l/5.5—9.1.

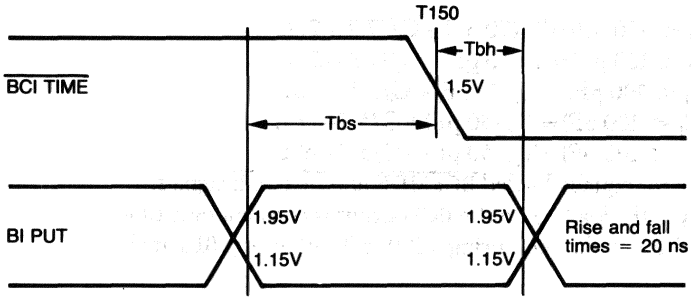
[3]T_{ssm} applies only for equally loaded $\overline{\text{BCI MDE}}$ and $\overline{\text{BCI SDE}}$ signals.

[4]T_{asp} applies only if the loading on the BCI output is constant over time.

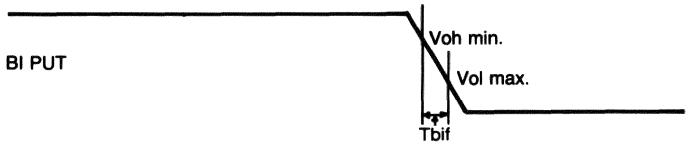
[5]Only one transition is allowed during T100 to T130 on any BI signal.



A. VAXBI Driver Output Waveforms

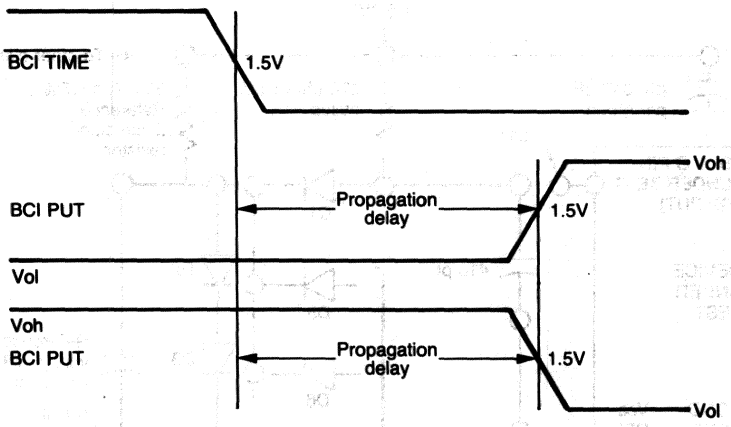


B. VAXBI Receiver Setup and Hold Time Waveforms

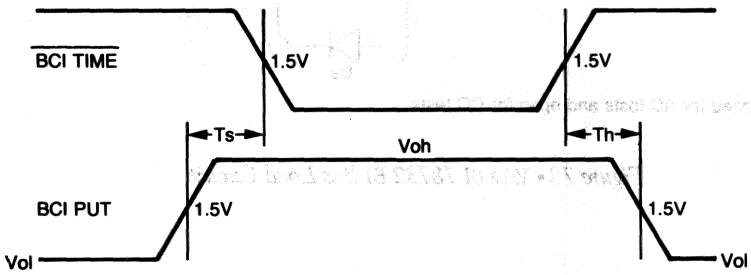


C. VAXBI Driver Minimum Fall Time Waveform

Figure 76 ▪ VAXBI 78732 BI Bus Voltage Waveforms



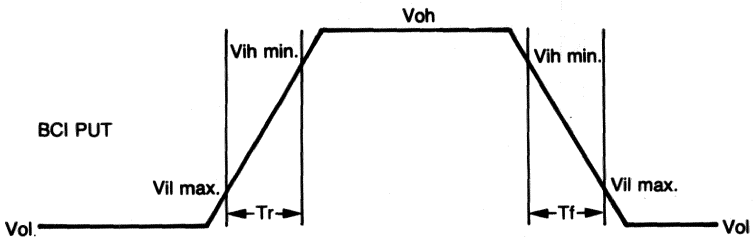
A. BCI Transceiver Propagation Delay Waveforms



NOTE:

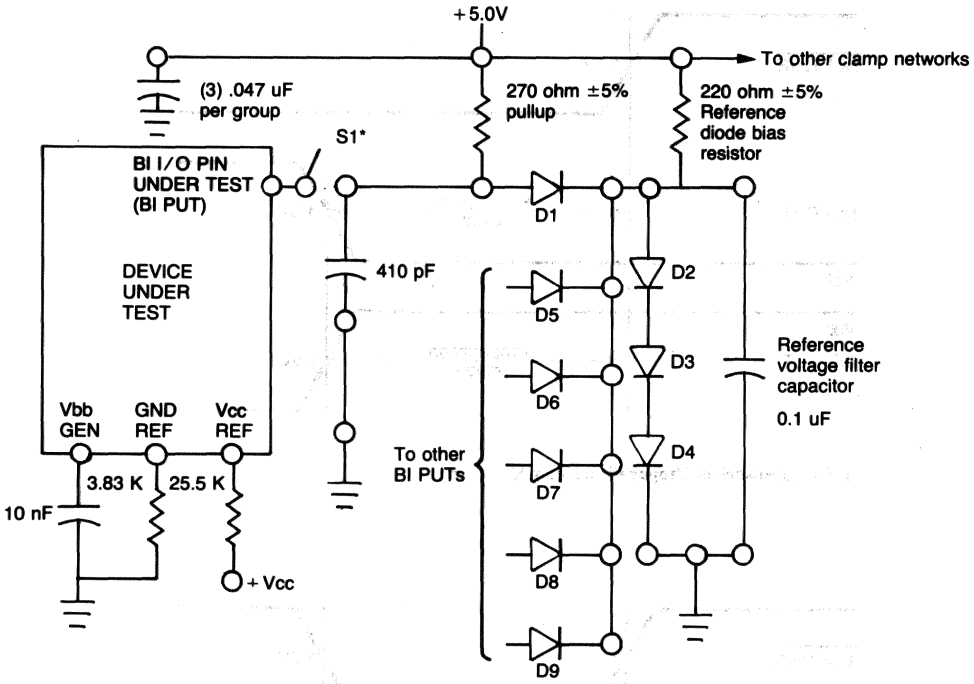
BCI input signals should have rise and fall times of < 5 ns (between 0.8 and 2.0 volt points).

B. BCI Receiver Setup and Hold Time Waveforms



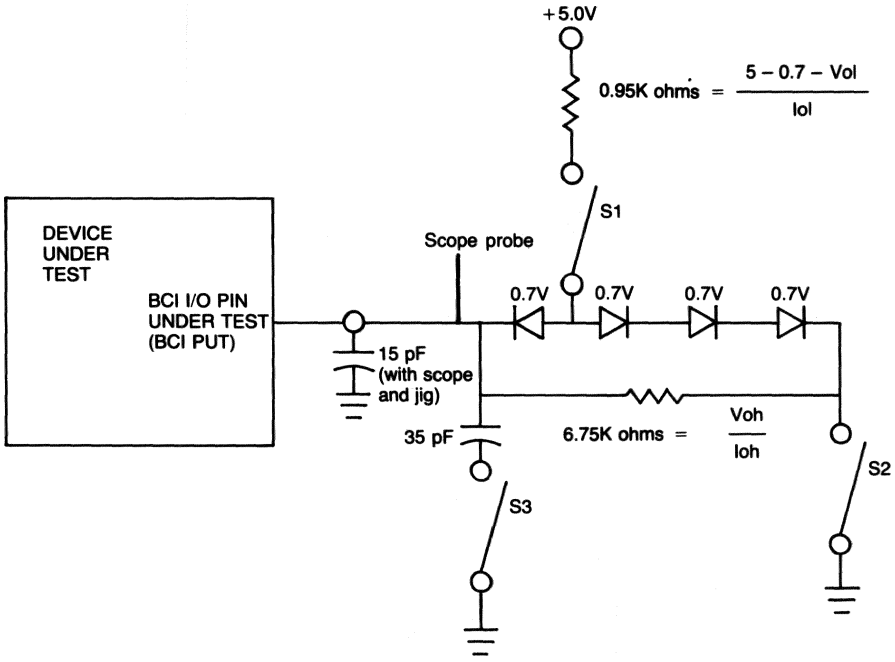
C. BCI Driver Rise and Fall Time Waveforms

Figure 77 ▪ VAXBI 78732 BCI Bus Voltage Waveforms



*Switch S1 is closed for AC tests and open for DC tests.

Figure 78 • VAXBI 78732 BI Bus Load Circuit



BCI Signal Change	S1	S2	S3
0 to 1	OPEN	CLOSED	CLOSED
1 to 0	CLOSED	CLOSED	CLOSED
Z to 0	CLOSED	OPEN	CLOSED
Z to 1	OPEN	CLOSED	CLOSED
0 to Z or 1 to Z	CLOSED	CLOSED	OPEN

NOTE:

The SENTRY test fixture does not match this test configuration.

Figure 79 ▪ VAXBI 78732 BCI Bus Load Circuit

Features

- Supports VAXBI bus system features of low interface cost, less than 800-nanosecond data access time, and high data integrity
- Supports the full range of data lines needed for an efficient VAXBI interface
- DMA master port allows data transfer at full VAXBI speeds
- Autoincrementing master port address registers
- Page overflow detection features
- Single 5-volt supply

Description

The VAXBI 78743 BCI Adapter Interface (BCAI) is a ZMOS custom integrated circuit that functions as a buffer file between user-designed processors, memories, and adapter modules and Digital's high-performance VAXBI (VAX bus interconnect) bus. A block diagram of the BCAI is shown in Figure 1.

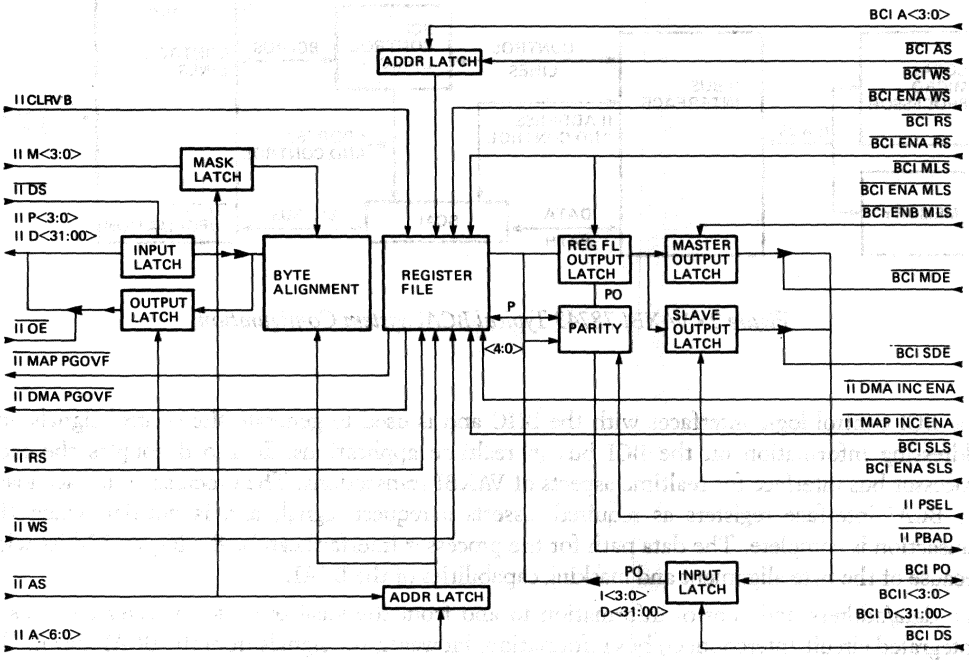


Figure 1 ▪ VAXBI 78743 BCAI Block Diagram

The VAXBI bus is a 32-bit general purpose synchronous bus that can be effectively used in single or multiprocessing systems based on VAX computers, other 32-bit processors, and compatible devices. The VAXBI bus has a maximum length of 1.5 meters and can contain up to 16 intelligent nodes on as many as 36 modules with an aggregate throughput of 13.3 Mbytes per second. The BCAI supports the full range of data lines needed to fulfill an efficient VAXBI interface.

The BCAI contains a dual-port register file and byte shifter that functions as a buffer file between the VAXBI bus interconnect interface chip (BIIC) and a local adapter bus on a VAXBI module. Individual registers within the register file have been customized to support the following functions:

- A VAXBI master port used for high bandwidth data transfers (DMA master port)
- A second master port used for lower bandwidth local processor access to the VAXBI (mapped master port)
- A slave port used to transfer VAXBI transactions onto the local bus (slave port)

A typical VAXBI 78743 BCAI interface configuration is shown in Figure 2. The BCAI performs data-path operations and requires external control logic to fully implement a BCI adapter interface. The data path and the registers of the DMA master port are designed to allow data transfer at full VAXBI speeds.

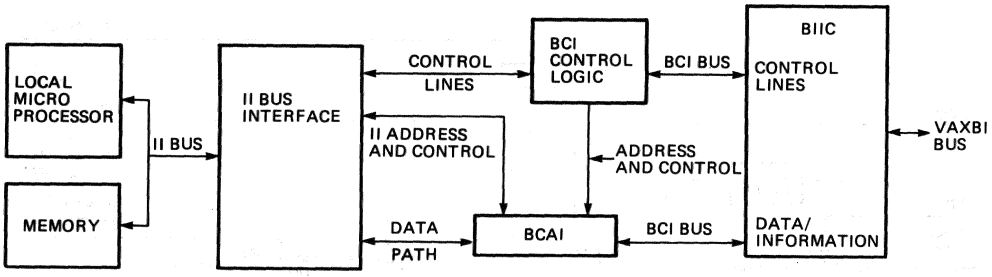


Figure 2 • VAXBI 78743 Typical BCAI System Configuration

The BCI control logic interfaces with the BIIC and is used to generate the control signals and addressing information on the BCI bus in realtime applications. It also decouples the local processor bus interface for realtime aspects of VAXBI transactions. The processor interface loads the BCAI interface registers as required, asserts a request signal, and is notified when the transaction is complete. The data path for the processor interface can be 8-, 16-, or 32-bits wide because of the byte alignment and masking capabilities of the BCAI.

The data/address and control information to and from the local processor is referred to as II (Integrated circuit Interconnect) bus information. However, the signals from the BCAI may not be compatible with every specific II implementation. The II bus address and control functions can be implemented by a ROM-based microsequencer or by other logic interfaced to the microprocessor bus.

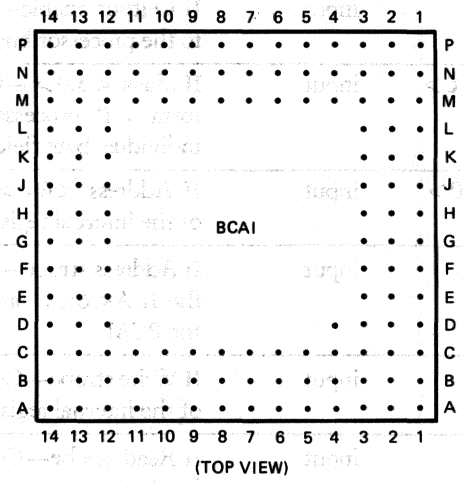
The BCAI contains a dual-port register file consisting of 15 registers, each of which is 41-bits wide. The file is accessed from the II bus interface as 21 32-bit registers and from the BCI bus as 14 36-bit registers. The BCI bus register information is transferred on 32 data lines and 4 information lines.

Associated with the II bus is a byte swapper and special address decode logic that permits the user to access one to four contiguous bytes starting from any byte boundary. The registers must be accessed along longword boundaries from the BCI bus. The register file and byte swapper logic includes several address and data latches. The BCI interface port has two output latches, one for the master port and one for the slave port. A parity section is also available and checks parity in the register file.

Figure 1 shows the functional blocks and the I/O lines of the BCAI. Four of the BCI bus control lines ($\overline{\text{BCI RS}}$, $\overline{\text{BCI WS}}$, $\overline{\text{BCI MLS}}$, and $\overline{\text{BCI SLS}}$) have enable lines that gate the information from the primary line into the BCAI. This simplifies external controller design by allowing external clock signals to generate the precise timing required.

• Pin and Signal Definitions

The VAXBI 78743 BCAI, contained in a 133-pin package, functions with the input and output signals and power and ground connections shown in Figure 3. The inputs and outputs are described in the following paragraphs. The signals are grouped by II bus interface signals and BCI bus interface signals.



REFER TO TABLE 1 FOR II BUS INTERFACE SIGNALS, POWER, AND GROUND PIN ASSIGNMENTS.
REFER TO TABLE 4 FOR BCI BUS INTERFACE SIGNAL PIN ASSIGNMENTS.

Figure 3 • VAXBI 78743 Pin Assignments

II Bus Interface Signals

Table 1 is a summary of the II bus signals that connect the BCAI to the microprocessor bus interface. The signal functions are described in the paragraphs that follow. Table 1 also includes the power and ground connections to the chip.

Table 1 • VAXBI 78743 II Bus Interface Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
M2,M4,M5,M6 N7,N9,N10,M10 M3,P2,P4,N6, N8,M9,P12,M12, N2,P3,N5,P6, P8,P10,N11,N13, N3,N4,P5,P7, P9,P11,P13,P14	II D<31:00>	input/output	II Data <31:00>—Transfers data to and from the processor bus interface.
N1,M1,L2,K2	II P<3:0>	input/output	II Parity<3:0>—Parity for each of the four bytes on lines II D<31:00>.
L13	$\overline{\text{II DS}}$	input	II Data strobe—Loads the II D<31:00> line information into the BCIA.
M14	$\overline{\text{II OE}}$	input	II Output enable—Controls the data output to the processor bus interface.
L1,J2,K1,H3	II M<3:0>	input	II Mask <3:0>—Controls the ability to perform a II processor interface operation to individual byte fields.
H1,G1,F1,G2, F2,J1,H2	II A<6:0>	input	II Address <6:0>—Controls the selection of the internal registers in the register file.
E1	$\overline{\text{II AS}}$	input	II Address strobe—Loads the information on the II A<6:0> and II M<3:0> lines into the BCAI.
K13	$\overline{\text{II WS}}$	input	II Write strobe—Controls the write operation of the internal register file.
L14	$\overline{\text{II RS}}$	input	II Read strobe—Controls the read operation for the II interface port of the register file.
J12	II CLRVB	input	II Clear byte valid—Clears all master port byte valid bits.
B1	II PSEL	input	II Parity select—Selects the user supplied or internally generated parity indication on the BCI PO line during data output transfers to the BIIC.

Pin	Signal	Input/Output	Definition/Function
D12	$\overline{\text{II PBAD}}$	input	II Parity bad—Indicates that the parity is not valid during data transfers to and from the BIIC.
J14	$\overline{\text{II DMA PGOVF}}$	output	II DMA page overflow—Indicates that the DMA address register is full.
J13	$\overline{\text{II MAP PGOFV}}$	output	II Map page overflow—Indicates that the MAP address register is full.
N14	$\overline{\text{II DMA INC ENA}}$	input	II DMA increment enable—Allows the masterport DMA address register to be incremented.
M13	$\overline{\text{II MAP INC ENA}}$	input	II Map increment enable—Allows the masterport map address register to be incremented.
L3,C10,A14,M11	V_{CC}	input	Voltage—Power supply voltage.
P1,C12	V_{BB}	output	Voltage—Back-bias voltage.
C4,C7,C8,F3 G3,J3,K3,K12, K14,L12,M7,M8, N12	GND	input	Ground—Ground reference.

II Data (II D<31:00>)—Bidirectional data lines that connect to a transparent input latch. The latch is controlled by the $\overline{\text{II DS}}$ signal. The three-state drivers are enabled by the $\overline{\text{II OE}}$ input.

II Parity (II P<3:0>)—Parity bits associated with each of the four bytes on the II D<31:00> lines. Valid byte parity must be generated by the user and loaded into the BCAI on lines II P<3:0> when transferring data, addresses, or command/mask/status information into the BCAI. When loading 4-bit command/mask/status information, the parity generated must be for the complete byte, including the zeros in the unimplemented portion of the byte. The BCAI also generates parity for data loaded into the BCAI from the BIIC and compares the parity it generates to the BCI P0 bit. The $\overline{\text{II PBAD}}$ line is set if an error is detected regardless of the direction of data flow. The II parity bits are latched with the II D<31:00> information by $\overline{\text{II DS}}$ signal and enabled by $\overline{\text{II OE}}$ signal.

II Data Strobe ($\overline{\text{II DS}}$)—This signal controls the transparent latches for the II D<31:00> input data. The input latch is transparent when the $\overline{\text{II DS}}$ input is asserted and the information is latched when the signal is deasserted.

II Output Enable ($\overline{\text{II OE}}$)—Controls the output drivers for the II D>31:00> and II P<3:0> lines. When $\overline{\text{II OE}}$ is asserted, the contents of the II D<31:00> output latch are transferred to the II D<31:00> data bus. When it is deasserted, the II D<31:00> lines become a high-impedance state. Line $\overline{\text{II OE}}$ has a 50-μA pullup circuit so that if the pin is not connected, it will remain deasserted.

II Mask (II M<3:0>)—Controls the ability to perform an II operation to individual byte fields. When the selected II M<3:0> lines are deasserted, an II bus write operation for the corresponding byte field is suppressed and an II read operation returns all zeros including the parity bit. The mask information is latched by the assertion of the $\overline{\text{II AS}}$ line. Table 2 lists the II bus interface mask bit assignments.

Table 2 • VAXBI 78743 II Bus Mask Bit Assignments

II Mask lines* 3	2	1	0	Valid data	Valid parity
1	1	1	1	II D<31:00>	II P<3:0>
0	0	0	1	II D<07:00>	II P0
0	0	1	0	II D<15:00>	II P1
0	1	0	0	II D<23:16>	II P2
1	0	0	0	II D<31:24>	II P3

*All other input combinations that specify the validity of the bytes on the II D<31:00> lines are allowed.

II Address (II A<6:0>)—Controls the selection of the internal registers in the register file. Refer to Figure 4 for the hexadecimal address values assigned to the registers. The II A<6:2> signals pass through a transparent latch controlled by $\overline{\text{II AS}}$ input and may be used in a latched or unlatched mode. Lines II A<6:2> are used to select the primary longword register being accessed and lines II A<1:0> control the byte offset multiplexers attached to the internal registers as listed in Table 3.

Table 3 • VAXBI 78743 Byte Offset

II A line* 1	0	Byte offset
L	L	none
L	H	1
H	L	2
H	H	3

*H = high level, L = low level.

For registers within the dual octaword buffer, the bytes that extend beyond the primary longword register are contained in the next adjacent register (A<6:2> + 1) except when II A <6:2> = 00111. When the exception exists, the primary longword register is at the bottom of the buffer and the offset is transferred to the longword register addressed by 00000. For registers not in the dual octaword buffer, the bytes that are offset beyond the primary longword register are not written during write operations and are returned as all zeros on read operations.

II Address Strobe (II AS)—Controls the transparent latch for the II A<6:0> data and mask bits II M<3:0>. The input latch is transparent when $\overline{\text{II AS}}$ is asserted and latched when deasserted.

II Write Strobe (II WS)—Controls the writing of the internal register file. The input data from the transparent latches on lines II D<31:00> is loaded into the selected register during assertion of the $\overline{\text{II WR}}$ strobe. The deassertion of the selected II M<3:0> lines will inhibit the write operation for the corresponding byte field. When accessing the DMA octaword data buffers, the byte valid bit for the addressed location is set when its byte is written.

II Read Strobe (II RS)—Controls the read operations for the II bus port of the register file. The read operation is initiated when the $\overline{\text{II RS}}$ line is asserted and the resulting output data is held in the II D<31:00> output latch when the $\overline{\text{II RS}}$ signal is deasserted. Deassertion of the selected II M<3:0> lines will inhibit the read operation for the corresponding byte field.

II Clear Valid Byte (II CLRVB)—This signal is used to clear all the master port Byte Valid bits associated with the dual octaword buffer.

II Parity Select (II PSEL)—Selects which source of parity (user supplied or internally generated) is passed to BCI PO output when data is transferred to the BIIC. A low level selects user parity and makes the errors within the processor bus interface or the BCAI visible to the VAXBI bus. A high level selects the internal parity that always provides correct parity for the data being passed to the BCI bus. This signal is latched by the $\overline{\text{BCI AS}}$ line.

II Parity Bad (II PBAD)—When set, it indicates one of two conditions:

- When transmitting data to the BIIC, the result of the internal parity generation from the BCI D<31:00> and BCI I<3:0> lines for this cycle do not agree with the parity bits associated with the 5 bytes being transmitted.
- When receiving data from the BIIC, the BCI PO line from the BIIC does not agree with the internal parity generated from the 5 bytes being received on the BCI D<31:00> and BCI I<3:0> lines.

II DMA Page Overflow ($\overline{\text{II DMA PGOVF}}$)—Asserted to indicate that the DMA address register has reached the boundary of a 512-byte page.

II MAP Page Overflow ($\overline{\text{II MAP PGOFV}}$)—Asserted to indicate that the MAP address register has reached the boundary of a 512-byte page.

II DMA Increment Enable ($\overline{\text{II DMA INC ENA}}$)—Enables the low-order 9 bits of the master port DMA address register to be incremented by a value of 4, 8, or 16 as specified by the length field (bits 31:30 of the DMA address register) whenever the master port DMA address register is accessed by a BCI read operation.

II Map Increment Enable ($\overline{\text{II MAP INC ENA}}$)—Enables the low-order 9 bits of the master port map address register to be incremented by a value of 4, 8, or 16 as specified by the length field (bits 31:30) of the map address register when the master port map address register is accessed by a BCI read operation.

BCI Bus Signals

Table 4 is a summary of the BCI bus signals that connect the BCAI to the BIIC interface. The signal functions are described in the paragraphs that follow.

Table 4 • VAXBI 78743 BCI Bus Interface Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
C2,D3,A1,C3, B2,A2,C5,B3, A3,B4,A4,B5, A5,C6,B6,A6, A7,B7,B8,A8, A9,B9,A10,A11, C9,B10,A12,B11, A13,B12,C11,B13	BCI D<31:00>	input/output	BCI Data <31:00>—Data lines that transfer data between the BCIA and the BIIC interface.
E14,F13,D14, F12	BCI I<3:0>	input/output	BCI Information <3:0>—Information lines used to transfer command, mask, and status information between the BCAI and BIIC interface.

Pin	Signal	Input/Output	Definition/Function
E13	BCI P0	input/output	BCI Parity—A parity indicator from the BIIC when data is received and to the BIIC when data is transferred to the the BIIC.
F14	$\overline{\text{BCI DS}}$	input	BCI Data strobe—Loads the information on the BCI D<31:00> and BCI I<3:0> lines into the BCAI.
D1,E3,C1,D2	BCI A<3:0>	input	BCI Address—Controls the selection of the internal registers in the register file.
E2	$\overline{\text{BCI AS}}$	input	BCI Address strobe—Controls the loading of the A<3:0> input information from the BIIC.
H12	$\overline{\text{BCI WS}}$	input	BCI Write strobe—Controls the writing or from the BIIC to the register file.
H13	$\overline{\text{BCI ENA WS}}$	input	BCI Enable write strobe—Enables the $\overline{\text{BCI WS}}$ input to the BCAI.
H14	$\overline{\text{BCI RS}}$	input	BCI Read strobe—Controls the read operation of the register file from the BIIC.
G14	$\overline{\text{BCI ENA RS}}$	input	BCI Enable read strobe—Enables the operation of the $\overline{\text{BCI RS}}$ signal from the BIIC.
C13	$\overline{\text{BCI ENA MLS}}$	input	BCI Enable A master latch strobe—Enables the operation of the $\overline{\text{BCI RS}}$ signal input from the BIIC.
B14	$\overline{\text{BCI ENB MLS}}$	input	BCI Enable B master latch strobe—Enables the operation of the $\overline{\text{BCI RS}}$ signal input from the BIIC.
E12	$\overline{\text{BCI MLS}}$	input	BCI Master latch strobe—Controls the operation of the transparent output register.
C14	$\overline{\text{BCI SLS}}$	input	BCI Slave latch strobe—Controls the transfer of information on the BCI D<31:00> and BCI I<3:0> lines to the BIIC.
D13	$\overline{\text{BCI ENA SLS}}$	input	BCI Enable slave latch strobe—Enables the operation of the $\overline{\text{BCI SLS}}$ input from the BIIC.
G12	$\overline{\text{BCI MDE}}$	input	BCI Master data enable—Controls the slave output data from the BCAI to the BIIC.
G13	$\overline{\text{BCI SDE}}$	input	BCI Slave data enable—Controls the transfer of the slave data from the BCAI to the BIIC.

BCI Data (BCI D<31:00>)—Bidirectional data lines with a transparent input latch and two output latches. The input latch is controlled by the $\overline{\text{BCI DS}}$ line. The transmitters for the output latches are controlled by $\overline{\text{BCI MDE}}$ for the master data output latch and by the $\overline{\text{BCI SDE}}$ input for the slave data output latch.

BCI Information (BCI I<3:0>)—These lines are used to transfer SCMD, DCMD, and MCMD commands, mask, and status information to and from the BIIC. The lines are latched and enabled by the same signals as the BCI D<31:00> lines.

BCI Parity (BCI PO)—This bidirectional line receives the parity information when receiving data from the BIIC and it is compared to the parity from the internal parity generator. Internal parity is generated when the BCI D<31:00> and BCI I<3:0> lines are latched. When transmitting data, this line supplies user parity or internal parity, as selected by line II PSEL. If internal parity does not agree with externally supplied parity in either direction, the $\overline{\text{II PBAD}}$ line is asserted. The $\overline{\text{BCI PO}}$ information is latched and enabled the same as the BCI D<31:00> lines.

BCI Data Strobe (BCI DS)—Controls the transparent input latch for BCI D<31:00>, BCI I<3:0>, and BCI PO input data. The input latch is transparent when $\overline{\text{BCI DS}}$ is asserted and the information is latched when it is deasserted.

BCI Address (BCI A<3:0>)—Controls the selection of the internal register file registers. The BCI A<3:0> lines transfer through a transparent latch controlled by the $\overline{\text{BCI AS}}$ signal and may be used in a latched or unlatched mode.

BCI Address Strobe ($\overline{\text{BCI AS}}$)—Controls the transparent latch for BCI A<3:0> input data and for II PSEL input. The latch is transparent when $\overline{\text{BCI AS}}$ is asserted and the information is latched when it is deasserted.

BCI Write Strobe ($\overline{\text{BCI WS}}$)—Controls BCI bus write operations to the internal register file. The input data BCI D<31:00> from the input latch is loaded into the selected BCAI register during assertion of this signal.

BCI Enable Write Strobe ($\overline{\text{BCI ENA WS}}$)—Gates the $\overline{\text{BCI WS}}$ level into the BCAI.

BCI Read Strobe ($\overline{\text{BCI RS}}$)—Controls BCI bus read operations of the internal register file. The operation is initiated when $\overline{\text{BCI RS}}$ is asserted and the resulting data is held in an internal register upon the deassertion of $\overline{\text{BCI RS}}$. The byte valid bits for the addressed register are reset when a byte within the DMA data buffer is read. The $\overline{\text{BCI ENA RS}}$ signal must be asserted or this line will be disabled.

BCI Enable Read Strobe ($\overline{\text{BCI ENA RS}}$)—Gates the $\overline{\text{BCI RS}}$ signal into the BCAI.

BCI Master Latch Strobe ($\overline{\text{BCI MLS}}$)—Controls the transparent output register for BCI D<31:00> and BCI I<3:0> master output data. The latch is transparent when $\overline{\text{BCI MSL}}$ is asserted and the information is latched when it is deasserted. Either the $\overline{\text{BCI ENA MSL}}$ or $\overline{\text{BCI ENB MLS}}$ signal must be asserted or this line will be disabled.

BCI Enable A Master Latch Strobe (BCI ENA MLS)—Gates the $\overline{\text{BCI MLS}}$ signal into the BCAI.

BCI Enable B Master Latch ($\overline{\text{BCI ENB MLS}}$)—Same function as the $\overline{\text{BCI MLS}}$ signal.

BCI Slave Latch Strobe ($\overline{\text{BCI SLS}}$)—Controls the transparent output latch for BCI D<31:00> and BCI I<3:0> slave output data. The latch is transparent when the $\overline{\text{BCI SLS}}$ is asserted and the information is latched when it is deasserted. The $\overline{\text{BCI ENA SLS}}$ signal must be asserted or this line is disabled.

BCI Enable Slave Latch Strobe ($\overline{\text{BCI ENA SLS}}$)—Gates the $\overline{\text{BCI SLS}}$ signal into the BCAI.

BCI Master Data Enable ($\overline{\text{BCI MDE}}$)—Controls the transfer of the data in the master output latch to the BCI D<31:00> and BCI I<3:0> lines. This signal has an internal 50 μA pullup device so that the BCI D<31:00> and BCI I<3:0> lines remain a high impedance when the $\overline{\text{BCI MDE}}$ input is not connected.

BCI Slave Data Enable ($\overline{\text{BCI SDE}}$)—Controls the transfer of the data in the slave output latch to the BCI D<31:00> and BCI I<3:0> lines. This signal has an internal 50 μA pullup device so that the BCI D<31:00> and BCI I<3:0> lines remain a high impedance when the $\overline{\text{BCI SDE}}$ input is not connected.

General Register Addressing

Figures 5 shows the memory map configuration and information of the BCAI registers when accessed by the BCI bus interface. Figure 6 show register memory map configuration and information of the BCAI registers accessed from the BCI bus interface. The hexadecimal address assignments and read/write capablities of each register are listed in the figures.

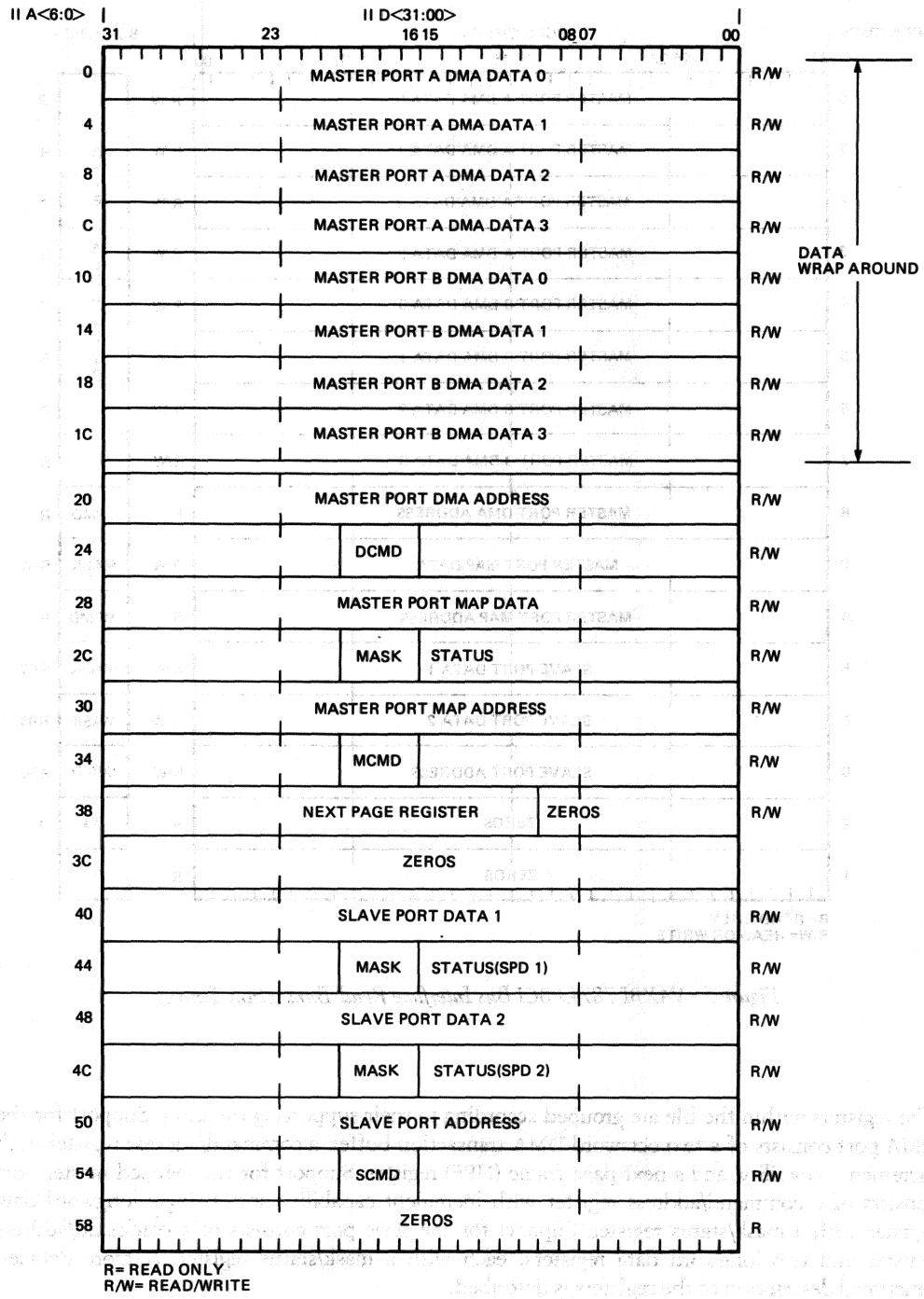


Figure 4 • VAXBI 78743 II Bus Interface Register Memory Map

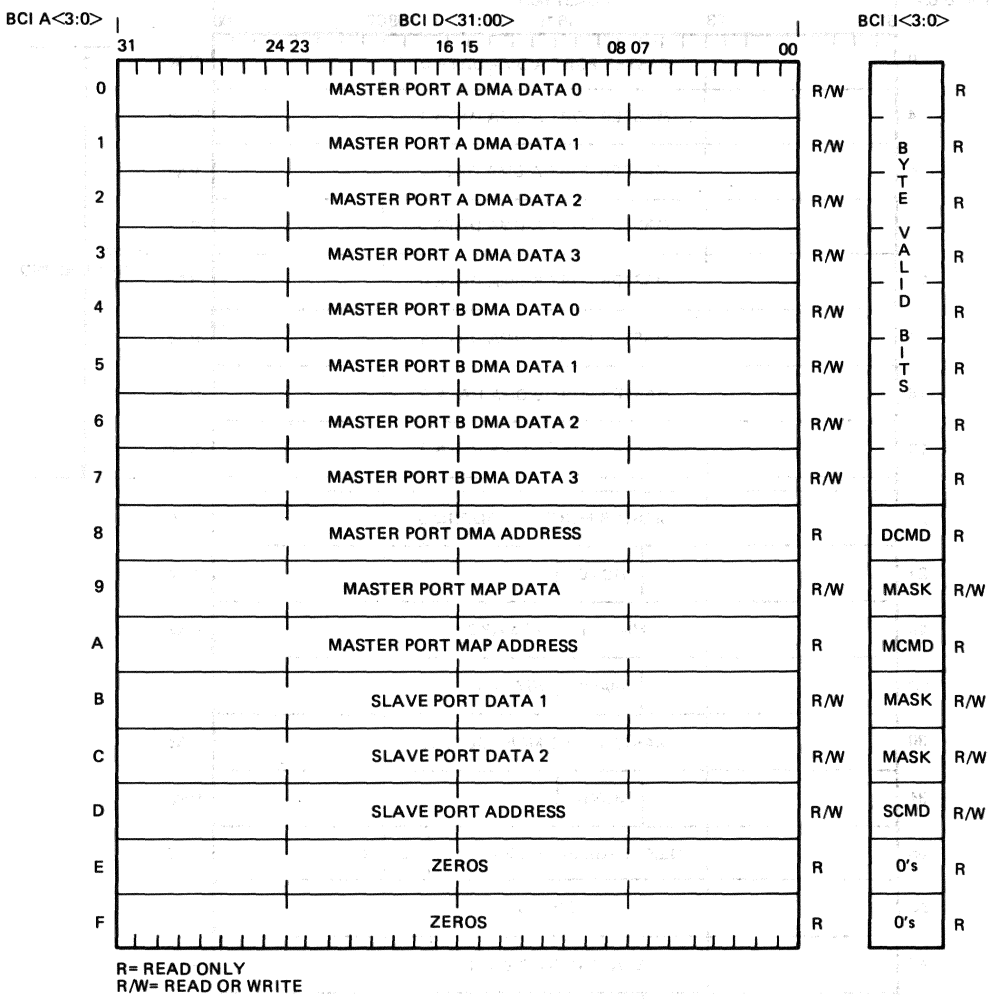


Figure 5 ▪ VAXBI 78743 BCI Bus Interface Read Transaction Timing

The registers within the file are grouped according to their supporting function. Support for the DMA port consists of a two octaword DMA transaction buffer, a command/address register with increment capability, and a next page frame (NPF) register. Support for the mapped master port consists of a command/address register with increment capability and a single longword data register with a mask/status register. Support for the slave port consists of a command/address register and two longword data registers, each with a mask/status register. A more detailed functional description of the registers is described.

Master Port Registers

The master port of the BCAI is used for II bus-initiated transfers to the VAXBI bus. The BCAI provides registers for two master ports—a high-speed DMA master port that is optimized for block transfers and a map master port. This allows a local processor to perform longword accesses to the VAXBI bus in the middle of a block DMA transaction without requiring the state of the previous transaction to be saved. The main difference between the two master ports is the size of their data buffers. The DMA master port has two octaword buffers and the map master port has a single longword data register. Both master ports have a command/address register with autoincrement capability and page-cross detection. The DMA master port includes a next page frame register that holds the map for the next page.

Master Port A and B DMA Data Registers—The DMA data buffer consists of eight contiguously addressed longwords of read/write memory organized as two octaword buffers, A and B. These buffers allow one transaction buffer to be accessed by an II bus transaction while the other is accessed by a master VAXBI bus transaction. To support all possible address alignments, the BCAI byte multiplexer directs an II bus transfer to any four sequential bytes in the transaction buffers. An overflow that occurs when reading or writing from either octaword is automatically directed to the first bytes of the other octaword. Therefore, the fourth longword of an unnaturally aligned octaword transaction can extend into the first three bytes of the other octaword.

Each DMA data buffer byte location has a corresponding valid bit. A logical 1 indicates that the byte is valid. All valid bits are cleared when the II CLRVB signal is asserted or when an II bus read operation is performed with an address of 7F (hexadecimal). When a write operation from the II bus to a byte location within the DMA data buffer is performed, the corresponding valid bit for that location is set. When data is read by the BCI bus, the valid bits are supplied to the BCI I<3:0> lines and can be used to support the VAXBI bus masked write commands. Typically a user would clear the valid bits, load a portion of an octaword into the DMA data buffer, and then initiate a VAXBI write mask transfer.

Master Port Map Data Register—This register stores the data to be read and written for map master port transactions. The mask/status register associated with the data register is separately accessed from the II bus by bits 19:16 of the address immediately following the data register address. When the data register is accessed by the BCI bus, the mask/status register transfers the information on the BCI I<3:0> lines.

Master Port Command/Address Registers—The master port DMA address register and the master port map address register are used to provide the VAXBI address for the command/address cycle of a VAXBI master port transaction. If the $\overline{\text{II DMA INC ENA}}$ or $\overline{\text{II MAP INC ENA}}$ lines are asserted when the corresponding address register is read from the BCI bus, the low-order 9 bits of that address are incremented by the size of the transaction as indicated by the length field (bits 31:30) of the address register. This increment feature enhances block DMA transfers because the user is not required to reload the address for every VAXBI bus transaction during sequential transfers. Parity for the two bytes affected by the increment is automatically recalculated.

The access time of the address registers is same as the other registers. However, a minimum of 500 ns must be allowed between two BCI read operations to the address register with the increment feature enabled. If this time is not allowed, the present address will be read correctly, but the next read operation will result in an incorrect address. The $\overline{\text{INC ENA}}$ signal that is essentially the carry input to the lowest bit of the incrementer must also be asserted 500 ns before a BCI read operation of the associated address register.

The master port DMA command and master port map command registers are used to provide the VAXBI command for the command/address cycle of a VAXBI master port transaction. From the II bus, the command register is located in the longword immediately above the address register and the commmand information is contained in bits 19:16 of this register. For the BI bus, the command and address registers are at the same location. When the command/address register is read from the BCI bus, the command data is transferred on the BCI I<3:0> lines and the address data is transferred on the BCI D<31:00> lines.

Next Page Frame Register—The DMA master port has a Next Page Frame (NPF) register that can be preloaded with the upper address bits for the page following the current page. When the DMA address register reaches a page boundary and the $\overline{\text{II DMPGOVF}}$ signal is asserted, bits 31:09 of this register are automatically transferred to the DMA address register bits 31:09. This feature can increase throughput during block DMA transactions. Instead of halting while waiting for a new map, VAXBI transactions can continue for as long as a page while the adapter controller fetches and loads the next map. If an update of the NPF register occurs concurrent with the second page crossing, the value loaded into the DMA address register is unpredictable.

The low-order 9 bits of the DMA address are not loaded from the NPF register, so the $\overline{\text{II DMA PGOVF}}$ signal remains asserted until the next address increment or until an II bus write transaction occurs to the DMA address register or the NPF register. When loading the NPF register from the II bus, the low-order 9 bits are not stored. Therefore, parity for the second byte (II D<15:08>) must be calculated as if the II D08 bit was zero. The NPF register cannot be directly accessed by the BCI interface.

Slave Port Registers

The BCAI provides a commmand/address register and two longword data registers with associated mask/status registers for the slave port.

Slave Port Data Registers 1 and 2—Slave port data register 1 and slave port data 2 registers are read/write registers used for slave read transactions and for slave write transactions. These registers allow a read response code to be written into the the slave read data status register from a II bus device having to rewrite it following a slave write transaction.

Slave Port Command/Address Register—The slave port command/address registers are used to store the command and address data for the command/address cycle of a VAXBI bus master port transaction. From the II bus, the registers are addressed separately with the command register located in the longword immediately above the address register. The command information is located in bits 19:16 of this register. From the VAXBI bus, the command and address registers reside in the same location. When the command/address register is written from the BCI interface, the command data is loaded from the BCI I<3:0> lines and the address data is loaded from the BCI D<31:00> H lines.

Page Boundary Detection

VAX processor memory management functions require that memory addresses are mapped from virtual to physical space on a per page basis. To facilitate the mapping process, the BCAI provides the $\overline{\text{II DMA PGOVF}}$ signal for the DMA master port and the $\overline{\text{II AP PGOVF}}$ signal for the map master port. These signals indicate to the processor that an address register has incremented beyond a page boundary. A flag bit in each address register indicates that a carry signal was produced when an address increment occurred. During a BCI read operation, the $\overline{\text{II DMA PGOVF}}$ and $\overline{\text{II MAP PGOVF}}$ signals are updated with the contents of the respective carry signals when the $\overline{\text{BCI RS}}$ signal is asserted. Therefore, when a BCI read operation of an address register causes the address to increment from all ones to all zeros, the overflow bit for that register will be asserted. The

subsequent read and increment operation will deassert the PGOVFL bit. The $\overline{\text{II MAP PGOVF}}$ bit is also deasserted by II bus write operation to the map address register. The $\overline{\text{II DMA PGOVF}}$ bit is deasserted by an II bus write operation to either the DMA address register or to the next page frame register. If a BCI interface read operation results in a page crossing at the same time an II bus write operation occurs to one of the registers that affect that $\overline{\text{II PGOVF}}$ bit, the state of the $\overline{\text{II PGOVF}}$ bit is unpredictable.

• Parity Data Path

All parity generated or checked by the BCAI is odd parity. A parity bit is associated with each byte of data in the BCAI register file, including bytes that contain command, mask, or status bits. For the 4-bit command, mask, or status information, the parity is calculated and supplied based on the complete byte that includes the four zeros above the register. Byte parity is supplied by the user on the II P<3:0> during II bus read operations. Nonexistent bytes, including upper bytes of shifted read to nonwraparound registers and registers that are defined as zero are read as all zeros. The parity bits are read as ones. Therefore nonexistent bytes have good parity.

During master-write or slave-read transactions that result in data transfers from the BCAI to the BIIC, the BCAI compares the byte parity bits from the user after the byte parity and data have been transferred through the BCAI, with the results of its own internal parity generator that operates on the BCI D<31:00> and BCI I<3:0> lines of the register file. If the parity bits do not agree, the $\overline{\text{II PBAD}}$ line is asserted. The II PSEL line selects which of the two sources of parity is passed to the BCI PO line. If the II PSEL line is a low level, the user supplied parity is given to the BIIC interface. Therefore if an error has occurred in the BCAI or in the II bus interface, the bad parity is passed to the VAXBI bus where it will stop the memory write operation and the transaction. If user parity is not required, the II PSEL line can be connected to a high level and the parity will match the data transferred to the BIIC regardless of the previous parity conditions.

During master read and slave write transactions that transfer data from the BIIC to the BCAI, the BCAI generates parity based on the incoming BCI D<31:00> and BCI I<3:0> line information and compares the result with the BCI PO line from the BIIC. If parity is different, the $\overline{\text{II PBAD}}$ line is asserted. An internal parity generator also determines the parity for each byte of data that is then transferred through the BCAI to the II bus interface. The byte parity is read on the II P<3:0> lines when the register file is read.

The state of the II PSEL line is latched by the assertion of $\overline{\text{BCI AS}}$ signal. The $\overline{\text{II PBAD}}$ signal is automatically deasserted when the $\overline{\text{BCI RS}}$ or $\overline{\text{BCI WS}}$ signal is asserted and is updated when the $\overline{\text{BCI RS}}$ or $\overline{\text{BCI WS}}$ signal is deasserted.

• Initialization

All the valid bits of the BCAI are cleared by the assertion of the II CLRVB signal or by an II bus read operation with lines II A<6:0> = 1111111. The II bus read operation enables the microprocessor to perform the initialization without the use of external logic. Other states within the BCAI are undefined during the powerup sequence.

If a write mask is the first DMA master-port transaction to follow the powerup sequence, the valid bits may have been cleared by the II CLRVB signal but the data bytes and parity bits that are not loaded will be undefined. The mask bits for those bytes will be zero and the BCAI and the VAXBI interface will still be checking their parity. To prevent erroneous parity, the correct parity should be entered by the user, the II CRLVB line asserted, and the transaction started at a known state.

Specifications

The mechanical, electrical, and environmental specifications for the VAXBI interface are as follows. The test conditions for the values specified are as follows unless specified otherwise.

- Junction temperature (T_J): 0°C to 125°C
- Supply voltage (V_{CC}): 4.75 V to 5.25 V

Mechanical Configuration

The mechanical dimensions for mounting the 133-pin VAXBI interface package are shown in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to a device. Exposure to the absolute maximum rating conditions for extended periods may adversely affect the reliability of the device.

- Pin voltages: −1.5 V to 7.0 V
- Operating junction temperature range (T_J): 0°C to 125°C
- Storage temperature range: −55°C to 125°C
- Package dissipation: 3.0 watts

Recommended Operating Conditions

- Supply voltage (V_{CC}): 4.75 V to 5.25 V
- Ambient temperature operating range (T_A): 0°C to 70°C

dc Electrical Characteristics

Table 5 contains the dc electrical parameters for the input and outputs of the BCAI interface chip.

Table 5 • VAXBI 78743 dc Input and Output Parameters

Symbol	Parameter	Test Conditions	Requirements Min.	Max.	Units
V_{IH}	High-level input voltage		2.0	V_{CC}	V
V_{IL}	Low-level input voltage		−1.0	0.8	V
V_{OH}	High-level output voltage	I_{out} = II I_{OH}	2.7	—	V
V_{OL}	Low-level output voltage	I_{out} = II I_{OL}	—	0.5	V
I_{OH}	High-level output current	V_{out} = II V_{OH}	−400	—	A
I_{OL}	Low-level output current	V_{out} = II V_{OL}	4.0	—	mA
I_I	Input current[1]		—	±20	µA
II_{LA}	Input current open latch[1]		−230	25	µA
I_{OS}	Output current short circuit[2]		—	−150	mA
I_{OE}	Enable line current	$\overline{\text{BCI MDE}}$, $\overline{\text{BCI SDE}}$, $\overline{\text{II OE}}$ inputs	50	200	µA
I_{DD}	Power supply current		—	500	mA
V_{BB}	Substrate bias voltage	Generated internally	−3.6	−2.4	V
C_{IO}	Input/output capacitance	$0 < V_{IO} < V_{CC}$	—	10	pF

[1]Applies to the following three-state bidirectional signals: BCI D<31:00>, BCI I<3:0>, BCI P0, II D<31:00>, and II P<3:0>.
II_{LA} applies when the following inputs are open and I_I when closed: BCI A<3:0>, II M<3:0>, II A<6:0>, and II PSEL.
[2]Not more than one output must be short circuited at a time and the duration of the short must not exceed 1 second.

ac Electrical Characteristics

Figure 6 shows the signal timing for a read transaction from the BCI bus interface and Table 6 lists the timing parameters. Figure 7 shows the signal timing for a write transaction from the BCI bus interface and Table 7 lists the timing parameters. Figure 8 shows the signal timing for an address increment and Table 8 lists the timing parameters. The signal timing for a II bus interface read transaction is shown in Figure 9 and the timing parameters are listed in Table 9. Table 10 lists the timing parameters for a II bus interface write transaction shown in Figure 10.

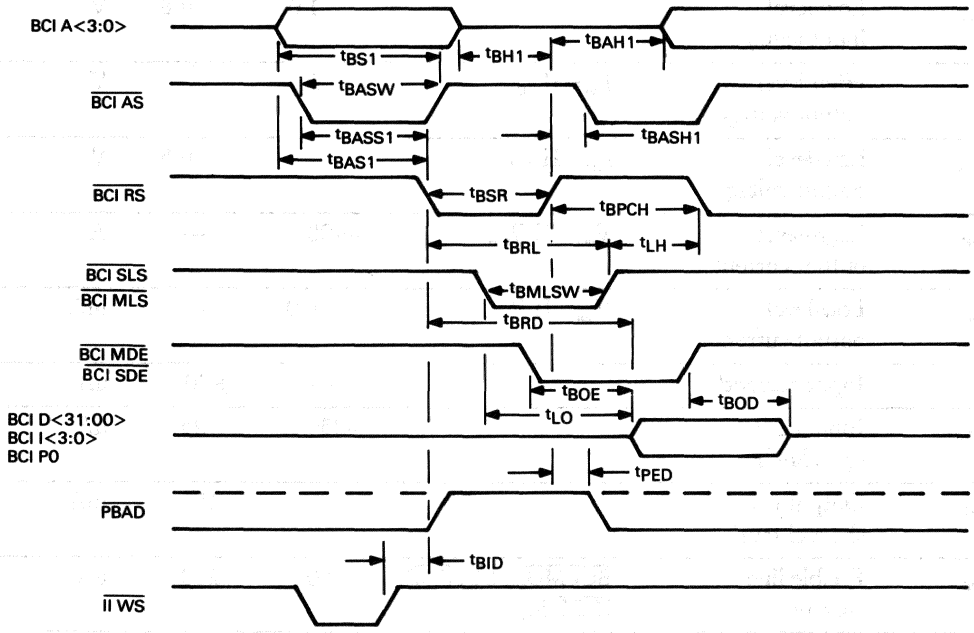


Figure 6 • VAXBI 78743 BCI Bus Interface Read Transaction Timing

Table 6 • VAXBI 78743 BCI Bus Interface Read Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{BS1}	BCI A<3:0> to $\overline{\text{BCI AS}}$ setup time	15	—
t_{BH1}	BCI A<3:0> to $\overline{\text{BCI AS}}$ hold time	10	—
t_{BASW}	BCI address latch strobe width	15	—
t_{BASS1}	$\overline{\text{BCI AS}}$ to $\overline{\text{BCI RS}}$ setup time	45	—
t_{BAS1}	BCI A<3:0> to $\overline{\text{BCI RS}}$ setup time	45	—

Symbol	Definition	Requirements (ns) Min.	Max.
t_{BASH1}	$\overline{\text{BCI AS}}$ from $\overline{\text{BCI RS}}$ hold time	15	—
t_{BAH1}	BCI A<3:0> from $\overline{\text{BCI RS}}$ hold time	15	—
t_{BSR}	Read strobe width	90	—
t_{BPCH}	Preset width ($\overline{\text{BCI RS}}$ and $\overline{\text{BCI WS}}$ unasserted)	40	—
t_{BRL}	Output latch close time after read access	—	100
t_{BRD}	Read access time	—	110
t_{BMLSW}	$\overline{\text{BCI MLS}}$, $\overline{\text{BCI SLS}}$ strobe pulse width	25	—
t_{LH}	$\overline{\text{BCI MLS}}$, $\overline{\text{BCI SLS}}$ strobe hold time	15	—
t_{BOE}	BCI output enable time	—	40
t_{BOD}	BCI output disable time	—	40
t_{PED}	Parity error output delay	—	60
t_{LO}	Latch to output delay	—	50
t_{BID}	$\overline{\text{II WS}}$ deasserts to BCI read (same register)	0	—

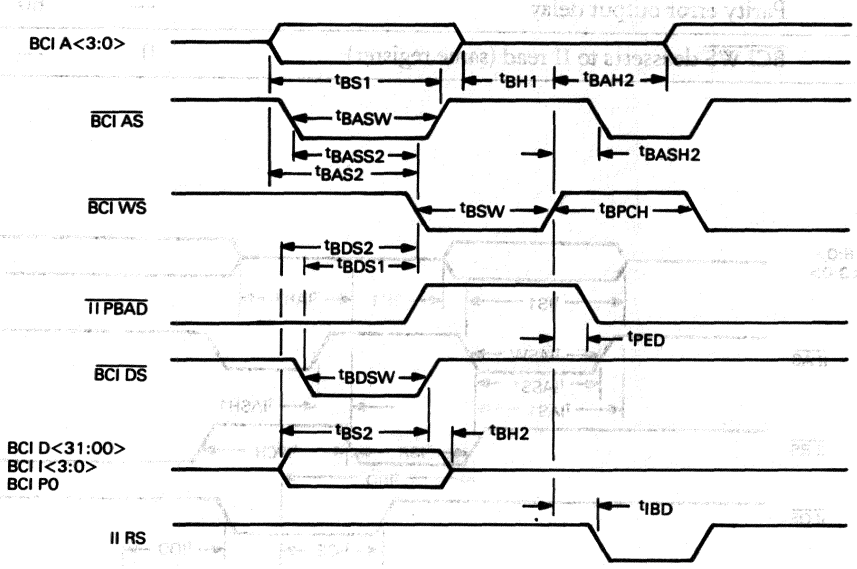


Figure 7 ▪ VAXBI 78743 BCI Bus Interface Write Transaction Timing

Table 7 ▪ VAXBI 78743 BCI Bus Interface Write Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{BS1}	BCI A<3:0> to $\overline{\text{BCI AS}}$ setup time	15	—
t_{BH1}	BCI A<3:0> to $\overline{\text{BCI AS}}$ hold time	10	—
t_{BASW}	BCI address latch strobe width	15	—
t_{BASS2}	$\overline{\text{BCI AS}}$ to $\overline{\text{BCI WS}}$	45	—
t_{BAS2}	BCI A<3:0> to $\overline{\text{BCI WS}}$ setup time	45	—
t_{BAH2}	BCI A<3:0> from $\overline{\text{BCI WS}}$ hold time	15	—
t_{BASH2}	$\overline{\text{BCI AS}}$ from $\overline{\text{BCI RS}}$ hold time	15	—
t_{BDS1}	$\overline{\text{BCI DS}}$ to $\overline{\text{BCI WS}}$ setup time	0	—
t_{BDS2}	BCI D<31:00> to $\overline{\text{BCI WS}}$ setup time	0	—
t_{BSW}	Write strobe width	90	—
t_{BPCH}	Preset width ($\overline{\text{BCI RS}}$ and $\overline{\text{BCI WS}}$ unasserted)	40	—
t_{BDSW}	BCI data strobe pulse width	15	—
t_{BS2}	BCI data strobe setup time	15	—
t_{BH2}	BCI data strobe hold time	10	—
t_{PED}	Parity error output delay	—	60
t_{IBD}	$\overline{\text{BCI WS}}$ deasserts to II read (same register)	0	—

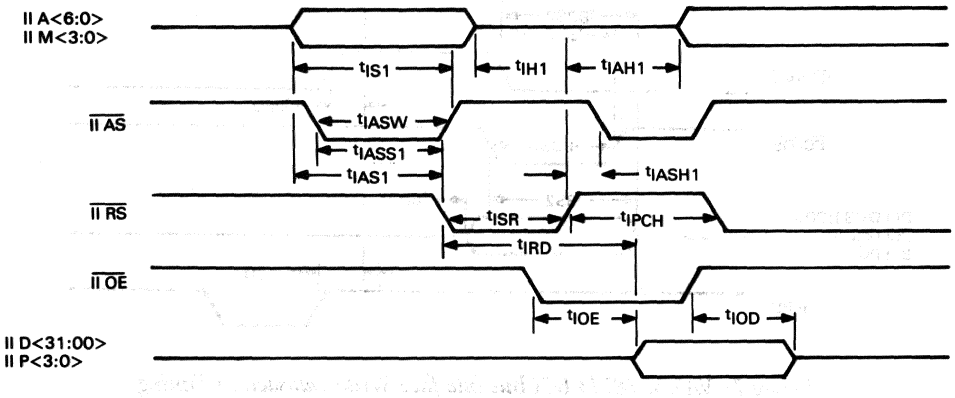


Figure 8 ▪ VAXBI 78743 II Bus Interface Read Transaction Timing

Table 8 • VAXBI 78743 II Bus Interface Read Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{IS1}	II A<3:0> to $\overline{\text{II AS}}$ setup time	15	—
t_{IH1}	II A<3:0> to $\overline{\text{II AS}}$ hold time	10	—
t_{IASW}	II address latch strobe time	15	—
t_{IASS1}	$\overline{\text{II AS}}$ to $\overline{\text{II RS}}$ setup time	60	—
t_{IAS1}	II A<3:0> to $\overline{\text{II RS}}$ setup time	60	—
t_{IAH1}	II A<3:0> to $\overline{\text{II RS}}$ hold time	15	—
t_{IASH1}	$\overline{\text{II AS}}$ from $\overline{\text{II RS}}$ hold time	15	—
t_{ISR}	Read strobe width	45	—
t_{IPCH}	Preset width ($\overline{\text{II RS}}$ and $\overline{\text{II WS}}$ unasserted)	40	—
t_{IRD}	Read access time	—	90
t_{IOE}	II output enable time	—	40
t_{IOD}	II output disable time	—	40

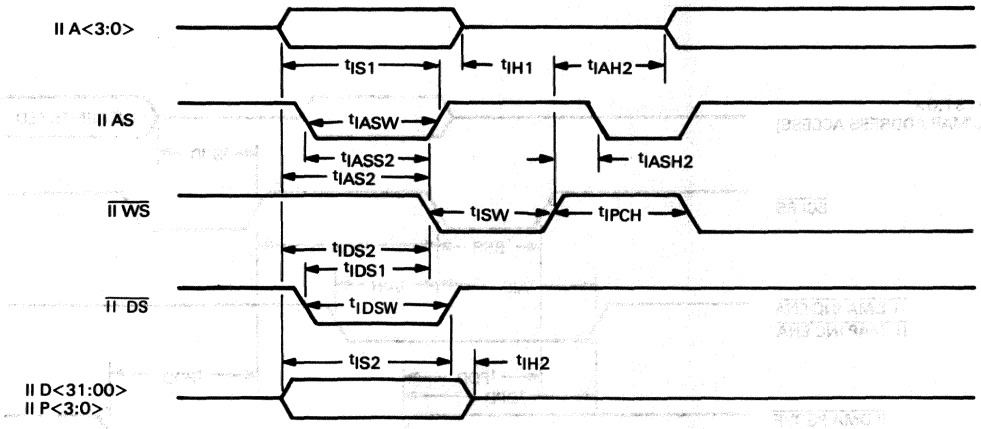


Figure 9 • VAXBI 78743 II Bus Interface Write Transaction

Table 9 • VAXBI 78743 II Bus Interface Write Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{IS1}	II A<3:0> to $\overline{\text{II AS}}$ setup time	15	—
t_{IH1}	II A<3:0> to $\overline{\text{II AS}}$ hold time	10	—
t_{IASW}	II address latch strobe time	15	—
t_{IASS2}	$\overline{\text{II AS}}$ to $\overline{\text{II WS}}$ setup time	60	—
t_{IAS2}	II A<3:0> to $\overline{\text{II WS}}$ setup time	60	—
t_{IAH2}	II A<3:0> to $\overline{\text{II WS}}$ hold time	15	—
t_{IASH2}	$\overline{\text{II AS}}$ from $\overline{\text{II RS}}$ hold time	15	—
t_{IDS1}	$\overline{\text{II DS}}$ to $\overline{\text{II WS}}$ setup time	0	—
t_{IDS2}	II D<31:00> to $\overline{\text{II WS}}$	0	—
t_{ISW}	Write strobe width	45	—
t_{IPCH}	Preset width ($\overline{\text{II RS}}$ and $\overline{\text{II WS}}$ unasserted)	40	—
t_{IDSW}	II data strobe width	15	—
t_{IS2}	II D<31:00> and II P<3:0> to $\overline{\text{II DS}}$ setup time	15	—
t_{IH2}	II D<31:00> and II P<3:0> from $\overline{\text{II DS}}$ hold time	10	—

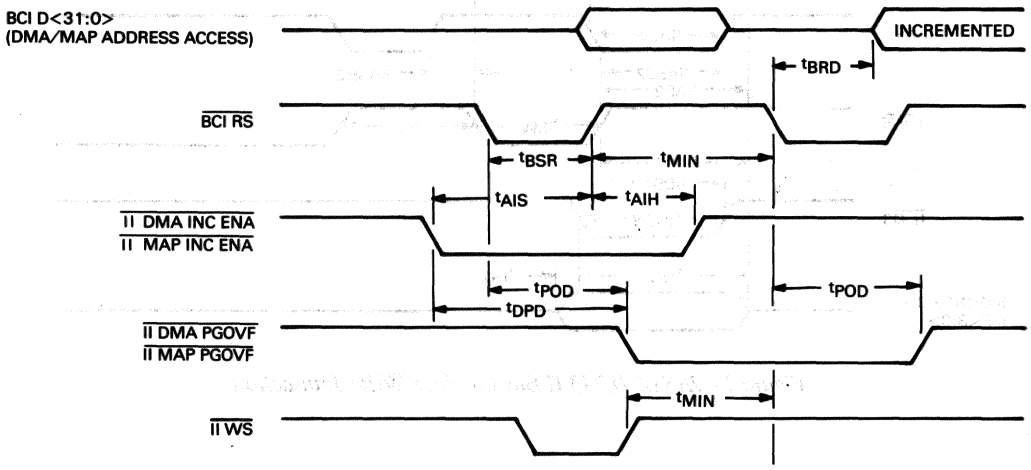


Figure 10 • VAXBI 78743 BCI DMA and MAP Address Increment Timing

Table 10 • VAXBI 78743 DMA and MAP Address Increment Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{BSR}	Read strobe width	90	—
t_{BRD}	Read access time	—	110
t_{POD}	Page overflow delay from $\overline{BCI\ RS}$ asserted	0	70
t_{AIS}	$\overline{II\ DMA\ INC\ ENA}$ or $\overline{II\ MAP\ INC\ ENA}$ setup to to BCI $\overline{RS}$ deassertion.	500	—
t_{AIH}	$\overline{II\ DMA\ INC\ ENA}$ or $\overline{II\ MAP\ INC\ ENA}$ hold time from to $\overline{BCI\ RS}$ deassertion.	—	10
t_{MIN1}	Minimum time between BCI reads with increment	500	—
t_{MIN2}	Minimum time between II writes and BCI reads with increment	500	—
t_{DPD}	Page overflow delay from $\overline{II\ DMA\ INC\ ENA}$ and $\overline{II\ MAP\ INC\ ENA}$	520	—

• Features

- Operates as a memory-mapped peripheral to the MicroVAX CPU chip.
- Responds to VAXBI bus transactions issued by other VAXBI nodes that access local memory or seek to interrupt the BCI3's MicroVAX CPU.
- Automatically translates any Integrated Circuit Interconnect (II) bus transaction into the equivalent VAXBI transaction whenever the II bus transactions references a nonlocal resource.
- Fast transfer of data blocks between local memory and VAXBI memory at the request of the MicroVAX CPU.
- Generates arbitrary VAXBI transactions when requested by the MicroVAX CPU.

• Description

The VAXBI 78733 bus interface adapter (BCI3), contained in a 132-pin ceramic pin grid array (PGA) package, is used to connect the Integrated Circuit Interconnect (II) bus of the MicroVAX 78032 processor to the VAXBI bus through the VAXBI 78732 Bus Interface Interconnect (BIIC) chip. The BCI3 is responsible for the low-level interface functions of the II bus and the high-level protocol translation necessary for interbus communication. It assists in managing the exchange of data and provides transparent translations of read and write transactions, error conditions, and interrupt requests. A block diagram of the BCI3 is shown in Figure 1.

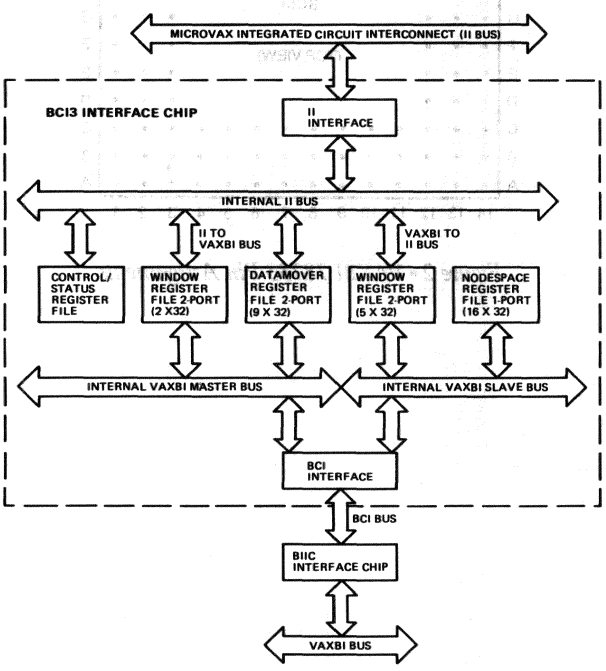


Figure 1 • VAXBI 78733 BCI3 Block Diagram

The BCI3 translates protocol, routes transactions between buses, and detects and resolves bus transactions that result in a deadlock condition. Read, write, and interrupt request transactions from the processor are automatically translated into VAXBI transactions. This includes the transfer of data and the exchanges of error information when necessary. Devices on the VAXBI bus can access the II bus resources and interrupt the operation of the processor.

In addition to supporting processor nodes, the BCI3 can also be used in intelligent MicroVAX-based peripheral devices. It contains a 5 Mbyte-per-second datamover for high bandwidth block data transfers used by I/O devices.

▪ Pin and Signal Descriptions

The VAXBI 78733 is a 132-pin interface that functions with the input and output signals and voltage and ground connections described in the following paragraphs. The pin and signal descriptions are grouped by II bus signals and by BCI bus signals. The pin assignments are shown in Figure 2.

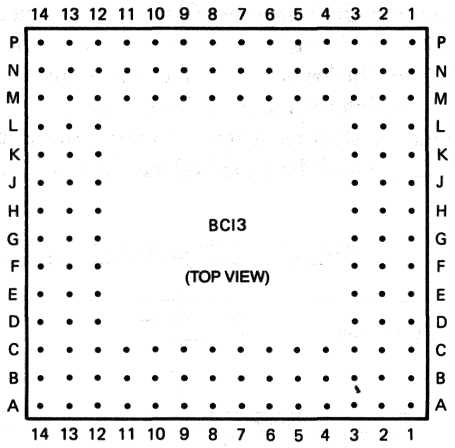


Figure 2 ▪ VAXBI 78733 Pin Assignments

II Bus Signals

Table 1 is a summary of the input and output signals that connect to the II bus. It also includes the power and ground connections. The signal functions are described in the following paragraphs.

Table 1 • VAXBI 78733 II Bus Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
N3,P3,M3,P2 N2,P1,M2,N1 K3,M1,L2,L1 K2,K1,J2,J1 H1,G2,G1,G3 F1,F2,E1,E2 D1,D2,C1,E3 B1,C2,A1,C3	II DAL<31:00>	input/output	II Data address lines <31:00>—Supplies data and address information.
N4,N6,N5,M6	$\overline{\text{II BM<3:0>}}$	input/output	II Byte mask <3:0> lines—Specifies the bytes to be read or written.
N7,P5,M7	IICS<2:0>	input	II Cycle status <2:0> lines—Used with the $\overline{\text{II WR}}$ line to specify the operation of the current cycle.
M8	$\overline{\text{II ERR}}$	input/output	II Error—Bus error indicator
M9,N12,P14, M12	$\overline{\text{II IRQ<3:0>}}$	output	II Interrupt request <3:0> lines—Interrupt request IPL17 to IPL14.
N8	$\overline{\text{II RLS}}$	input	II Retry latch strobe—Controls the loading of the retry address register.
N9	$\overline{\text{II DMR}}$	output	II DMA request—Requests a DMA transfer.
N11	$\overline{\text{II RESET}}$	input	II Reset—Starts the initialization process.
P4	$\overline{\text{II WR}}$	input	II Write—Provides read and write control of the bus.
P6	$\overline{\text{II AS}}$	input/output	II Address strobe—Indicates when valid address, cycle status, and write and mask information is available.
P7	$\overline{\text{II DS}}$	input/output	II Data strobe—Indicates when data can be transferred on the bus.
P8	$\overline{\text{II RLOE}}$	input	II Retry latch output enable—Enables the transfer of the retry address latch information.
P9	$\overline{\text{II DBE}}$	input/output	II Data buffer enable—Controls the DAL bus transceivers.
P10	$\overline{\text{II RDY}}$	input/output	II Ready—Indicates the status of a data transfer.

Pin	Signal	Input/Output	Definition/Function
P11	$\overline{\text{II RETRY}}$	input/output	II Retry—Requests a change in bus mastership and indicates a locked location.
P12	$\overline{\text{II DMG}}$	input	II DMA grant—Indicates that the BCI3 has been granted the bus for a DMA transfer.
P13	$\overline{\text{II NLMR}}$	input	II Nonlocal memory reference—Indicates that the memory space is not local.
L13	$\overline{\text{TOD}}$	input	Test output disable—Causes the BCI3 outputs to become a high impedance. Used for test purposes.
D3,D12,L3,L12,	V_{DD}	input	Voltage—Power supply voltage
C5,E12,E13,M4 M5,M10,M11,	V_{SS}	input	Ground—Common ground connection.

II Data and Address Lines (II DAL<31:00>)—These lines transfer data and address information between the BCI3 and II bus.

II Address Strobe ($\overline{\text{II AS}}$)—The II bus master asserts this signal to indicate that a valid address is available on the II DAL<31:00> lines and that valid status information is available on the II CS<2:0>, $\overline{\text{II WR}}$, and $\overline{\text{II BM<3:0>}}$ lines. The subsequent deassertion of this signal indicates the end of the bus cycle. This signal is synchronized internally by the BCI3.

II Data strobe ($\overline{\text{II DS}}$)—The II bus master asserts this signal during a read and interrupt acknowledge transaction to allow the slave to transfer data on the II DAL<31:00> lines. When the line is deasserted, it indicates that the selected slave may release the DAL<31:00> lines. During write cycles, this signal is asserted to indicate that valid data is available on the DAL<31:00> lines and deasserted to indicate that the master will soon release the DAL<31:00> lines. This signal is sychronized internally by the BCI3.

II Data Buffer Enable ($\overline{\text{II DBE}}$)—This signal is used by the bus master to enable the DAL<31:00> transceivers.

II Byte Mask ($\overline{\text{II BM<3:0>}}$)—This signal is used by the bus master during a read or write transaction to indicate which bytes within the data longword are to be read or written.

II Ready ($\overline{\text{II RDY}}$)—This signal is asserted during a read transaction to indicate to the bus master that data is ready for transfer. During a write transaction, it is asserted to indicate to the bus master that data has been latched. The signal is sychronized internally by the BCI3.

II Bus Error Indicator ($\overline{\text{II ERR}}$)—Assertion of this signal by the BCI3 during a MicroVAX-initiated II bus transaction (other than an instruction prefetch or interrupt acknowledge) causes the processor to perform a machine-check exception. Other devices on the VAXBI may assert the $\overline{\text{II ERR}}$ signal when the BCI3 is II bus master to indicate that the current II bus transaction has failed.

II Reset (II RESET)—If another device on the II bus asserts this signal for at least 2 microseconds, the BCI3 will reinitialize itself. The signal is synchronized internally by the BCI3.

II Write (II WR)—This signal specifies the direction of a data transfer on the II bus.

II Cycle Status (II CS<2:0>)—Together with the WR signal, these lines indicate the current bus cycle type requested by the II bus master. Table 2 lists the bus cycle selections.

Table 2 • VAXBI 78733 Bus Cycle Status Code Assignments

WR*	CS line* 2	1	0	Bus cycle type
H	L	L	L	not used
H	L	L	H	not used
H	L	H	L	not used
H	L	H	H	interrupt acknowledge
H	H	L	L	Read(I stream)
H	H	L	L	Read lock
H	H	L	L	Read(D stream, modify intent)
H	H	L	L	Read(D stream, no modify intent)
L	L	L	L	not used
L	L	L	H	not used
L	L	H	L	not used
L	L	H	H	not used
L	H	L	L	not used
L	H	L	H	write unlock
L	H	H	L	not used
L	H	H	H	write (D stream)

*H = high level, L = low level

II Interrupt Requests (II IRQ<3:0>)—These lines are used by the BCI3 to request bus mastership from the MicroVAX. Each line corresponds to an interrupt priority line IPL as listed in Table 3.

Table 3 • VAXBI 78733 Interrupt Request Line Assignments

Request line	Request level (hexadecimal)
IRQ3	IPL 17
IRQ2	IPL 16
IRQ1	IPL 15
IRQ0	IPL 14

II DMA Request ($\overline{\text{II DMR}}$)—This line is asserted by the BCI3 to request bus mastership from the MicroVAX CPU.

II DMA Grant ($\overline{\text{II DMG}}$)—This line is asserted to indicate to the BCI3 that it has been granted mastership of the II bus. After the $\overline{\text{II DMR}}$ line is negated, this line must be negated before the $\overline{\text{II DMR}}$ signal can be reasserted for the next II bus master transaction sequence.

II Nonlocal Memory Reference ($\overline{\text{II NLMR}}$)—The memory controller asserts this signal to indicate that the address of a memory-space II read or write cycle does not reside in local II bus memory.

II Retry ($\overline{\text{II RETRY}}$)—This signal is asserted by the BCI3 to request that the MicroVAX relinquish bus mastership and reissue the current II bus transaction at a later time. The memory controller asserts the $\overline{\text{II RETRY}}$ signal in response to an II bus read lock to indicate that a location is locked. The BCI3 assumes that this signal will not be asserted when it is II bus master except in response to the II bus read-lock transaction.

II Retry Latch Strobe ($\overline{\text{II RLS}}$)—This signal is asserted to latch the address information on the II DAL<31:00> into the retry address register. When negated, the retry address register is transparent.

II Retry Latch Output Enable ($\overline{\text{II RLOE}}$)—This signal is used to assert the contents of the retry address register on the II DAL<31:00> lines.

BCI Bus Signals

Table 4 contains a summary of the input and output signals that connect to the BCI bus. The signal functions are described in the following paragraphs.

Table 4 • VAXBI 78733 BCI Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
B2,A2,C4,A3, B3,A4,B4,A5, B5,A6,B6,A7, C7,A8,B7,A9, A10,B9,A11,C9, A12,B10,A13,B11, A14,B12,B13,C11, B14,C12,C14,C13	BCI D<31:00>	input/output	BCI data <31:00>—Used to transfer data between the BCI3 and the BCII.
B8,C8	BCI POWER	input	BCI power—Monitors the dc power of the BIIC.
C6,C10	BCI GND	input	BCI ground—Common ground reference for the BIIC signals.
E14,F13, D14,D13	BCI I<3:0>	input/output	BCI Information <3:0>—Transfers command values read status and write masks.
F12	$\overline{\text{BCI MDE}}$	input	BCI Master data enable—Indicates when data on lines BCI D<31:00> and BCI I<3:0> should be transferred to the BIIC.

Pin	Signal	Input/Output	Definition/Function
F14	BCI CLE	input	BCI Command latch enable—Indicates when BI command/address information is available during slave-port operation.
G12	$\overline{\text{BCI SDE}}$	input	BCI Slave data enable—Indicates when the BCI3 should transfer data on the BCI D<31:00> and BCI<3:0> lines.
G13,G14	$\overline{\text{BCI RS<1:0>}}$	output	BCI Slave response <1:0> lines—Used by the BCI3 to acknowledge the slave-port operation.
H12,K14,J13, J12,L14	$\overline{\text{BCI EV<4:0>}}$	input	BCI Event <4:0> lines—Contains the event status information from the BIIC.
P14,K12	$\overline{\text{BCI RQ<1:0>}}$	output	BCI Request—Initiates BIIC master-port operations.
J14,H13,H14	$\overline{\text{BCI SC<2:0>}}$	input	BCI Slave code <2:0> lines—Used to transfer status information during BIIC slave-port operation.
M13	$\overline{\text{BCI TIME}}$	input	BCI Time—A 20-MHz TTL clock reference signal from the BIIC.
M14	$\overline{\text{BCI RAK}}$	input	BCI Request acknowledge—Provides transaction timing.
N13	$\overline{\text{BCI PHASE}}$	input	BCI Phase—A 5-MHz clock reference signal from the BIIC.
N14	$\overline{\text{BCI MAB}}$	output	BCI Master abort—Used to halt current BIIC master-port operation.

BCI Data (BCI D<31:00>)—These lines are used to exchange VAXBI data and address field values with the BIIC.

BCI Information (BCI I<3:0>)—These lines are used to exchange VAXBI command field values, write masks, and read status with the BIIC.

BCI Master Request ($\overline{\text{BCI RQ<1:0>}}$—These lines are used to initiate BIIC master-port operations.

BCI Request Acknowledge ($\overline{\text{BCI RAK}}$)—This signal provides timing information for BIIC master-port operations.

BCI Next ($\overline{\text{BCI NXT}}$)—This signal provides timing information for BIIC master-port operations.

BCI Master Abort ($\overline{\text{BCI MAB}}$)—This signal aborts current BIIC master-port operations.

BCI Master Data Enable ($\overline{\text{BCI MDE}}$)—This signal indicates when the BCI3 should transfer data on the BCI D<31:00> and BCI I<3:0> lines during a BIIC master-port operation.

BCI Slave Response ($\overline{\text{BCI RS}}$<1:0>)—These signals are used by the BCI3 to acknowledge BIIC slave port action when it is acting as the selected VAXBI slave.

BCI Command Latch Enable (BCI CLE)—This signal indicates when VAXBI command/address information is available on the BCI D<31:00> and BCI I<3:0> lines during BIIC slave-port operation.

BCI Slave Select Code ($\overline{\text{BCI SC}}$<2:0>)—These signal provide status information during BIIC slave-port operations.

BCI Slave Data Enable ($\overline{\text{BCI SDE}}$)—This signal indicates when the BCI3 should transfer data onto the BCI D<31:00> and BCI I<3:0> lines during BIIC slave-port operations.

BCI Event Code Information ($\overline{\text{BCI EV}}$<4:0>)—These signals provide status information during BIIC master-port and slave-port operations.

BCI Time ($\overline{\text{BCI TIME}}$)—This signal is the local BI node's version of the 20-MHz VAXBI clock reference signal.

BCI Phase ($\overline{\text{BCI PHASE}}$)—This signal is the local VAXBI node's version of the 5-MHz VAXBI clock reference signal.

Test Output Disable (TOD)—This signal forces all BCI3 outputs into a high-impedance state. It is intended to support board-level manufacturing tests and includes an internal pullup circuit.

Power (V_{DD}) and Ground (V_{SS})—V_{DD} connects to the power supply positive voltage and V_{SS} connects to the power supply and signal ground reference.

• Architectural Summary

Figure 3 shows the internal organization of the BCI3 interface that includes four major subsystems—an II bus controller that initiates and responds to microprocessor bus transactions, a VAXBI master controller that initiates system bus transactions, a VAXBI slave controller that responds to the bus transactions and a datamover controller that coordinates the high-level DMA operations. Each is individually controlled and the three bus management subsystems each contain a 36-bit (32-bit data and 4-bit tag) bidirectional data bus. These four subsystems operate independently and can simultaneously service multiple transactions.

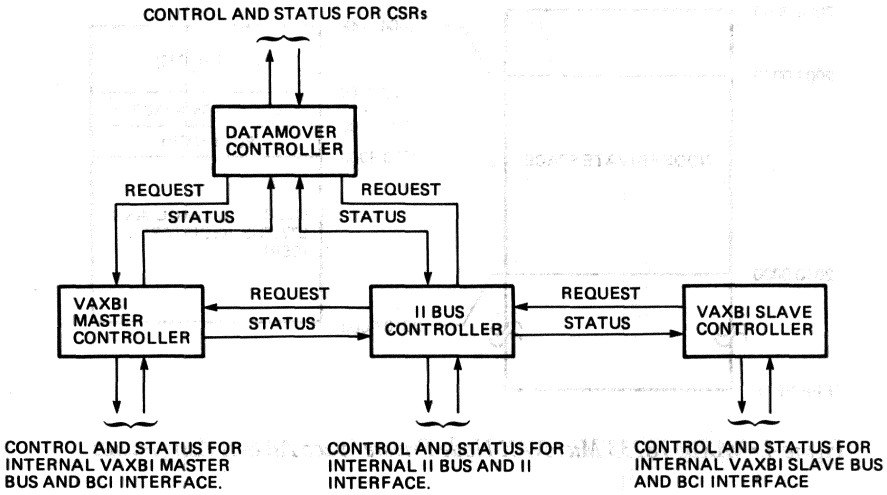


Figure 3 ▪ VAXBI 78733 Internal Subsystem Organization

In addition to performing requests from the MicroVAX or controller, the BCI3 automatically translates the transactions directly into VAXBI transactions and responds to all interrupt acknowledge transactions. It interrupts the MicroVAX if an interrupt request is received from another VAXBI node. In response to DMA requests from the VAXBI bus, the BCI3 accesses the II bus using the standard DMA protocol.

Memory References

When the MicroVAX processor references a memory space location, the memory controller or equivalent logic determines when the address space is within the range allocated to the BCI3 interface. The controller asserts the nonlocal memory ($\overline{\text{II NLMR}}$) signal to indicate the reference is not a memory location. The BCI3 translates the reference into the equivalent VAXBI transaction, transfers it to the VAXBI bus, and reports the results to the MicroVAX using the II bus read and write protocol.

The response of the BCI3 to the MicroVAX I/O space references depends on the address range specified. If the address of the MicroVAX is within the node private space, the response of the BCI3 depends the location accessed. The address assignments of the MicroVAX and the allocated space of the BCI3 interface are shown in Figure 4.

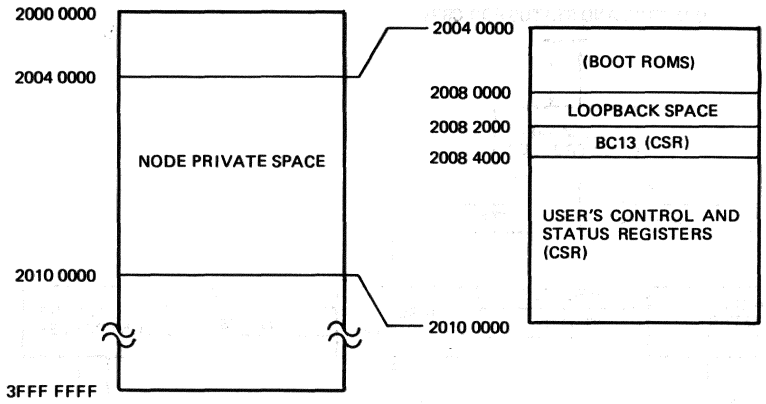


Figure 4 ▪ VAXBI 78733 MicroVAX Node Private Space Address Assignments

The lowest 256 Kbytes of II bus address space is assigned to the ROMs that contain the bootstrap program. The next address range consists of an 8-Kbyte loopback space and the BCI3 will respond to addresses in this range by using the loopback feature of the BIIC to permit access to the BIIC and BCI3 nodespace registers without performing VAXBI transactions.

The BCI3 control and status registers (CSRs) are located in the next higher 8-Kbyte range. These registers are accessible through the II bus and are accessed by the BCI3 without involving the BIIC or VAXBI bus. The user CSR space is ignored by the BCI3 and may be used by local II bus devices. If the address of a II bus transaction is not within the private nodespace, the BCI3 translates the transaction into a VAXBI bus transaction and reports the results to the processor using standard read or write protocol.

Interrupt Vector Requests—An II bus interrupt request can be initiated as a result of a transaction from the VAXBI, by the BCI3 interface, or by a device on the local VAXBI bus node other than the BCI3. When the MicroVAX acknowledges an interrupt request, the BCI3 serves as the local II bus interrupt arbitrator by determining which of the outstanding interrupt requests will be serviced by this transaction. If a VAXBI interrupt is being serviced, the BCI3 will solicit an interrupt vector from the interrupting VAXBI node using the VAXBI IDENT transaction. If the interrupt being serviced is a result of a VAXBI IPINTR (IPL 14 only), the BCI3 returns a fixed vector of 80 (hexadecimal). If the interrupt is BCI3 initiated (IPL 16 only), the BCI3 will return a fixed vector of 58 (hexadecimal). If none of these conditions are pending, the BCI3 interface returns a vector on behalf of a local device—the value of which depends on the IPL of the interrupt vector as follows: F0 (IPL 14), F4 (IPL 15), F8 (IPL 16), or FC (IPL 17).

▪ Register Description

The BCI3 interface contains two groups of registers. The nodespace register file is accessible from the VAXBI bus and consists of 12 general purpose registers and a two-ported toggle registers. The BCI3 private registers are accessible from the II bus and consist of 28 registers that are used to control the transfer of information, to report status, and to generate interrupt requests.

Nodespace Registers

The BCI3 interface contains 14 nodespace registers used to facilitate communication between different BI nodes. The nodespace registers and address assignments are shown in Figure 5.

	31 — 00
BB+200	GENERAL PURPOSE REGISTER 0 (BI RXCD ONLY)
BB+204	GENERAL PURPOSE REGISTER 1
BB+208	GENERAL PURPOSE REGISTER 2
BB+20C	GENERAL PURPOSE REGISTER 3
BB+210	GENERAL PURPOSE REGISTER 4
BB+214	GENERAL PURPOSE REGISTER 5
BB+218	GENERAL PURPOSE REGISTER 6
BB+21C	GENERAL PURPOSE REGISTER 7
BB+220	GENERAL PURPOSE REGISTER 8
BB+224	GENERAL PURPOSE REGISTER 9
BB+228	GENERAL PURPOSE REGISTER 10
BB+22C	GENERAL PURPOSE REGISTER 11
BB+230	TOGGLE REGISTER A SET PORT
BB+234	TOGGLE REGISTER A CLEAR PORT
BB+238	TOGGLE REGISTER B SET PORT
BB+23C	TOGGLE REGISTER B CLEAR PORT

Figure 5 • VAXBI 78733 Nodespace Register File

General purpose registers 1 through 11 and the two toggle registers are accessed by 16 VAXBI longword addresses; bb+200 through bb+23C (hexadecimal). The nodespace registers can be locked by a read lock command issued to any one register. This command locks the entire set until a write unlock command is issued.

Each of the toggle registers can be accessed from two addresses. Writing a 1 into a bit of the lower address register will set the corresponding bit in the register. Writing a 1 into the higher address register will clear the corresponding bit in the register.

Private Registers

The private registers are accessible only from the II bus. They may be referenced using longword length transactions but do not support the masking of write operations. The register file and assigned addresses are shown in Figure 6. Some BCI3 internal address registers in this address space can be used for diagnostic purposes. However, they must not be used during normal BCI3 operation.

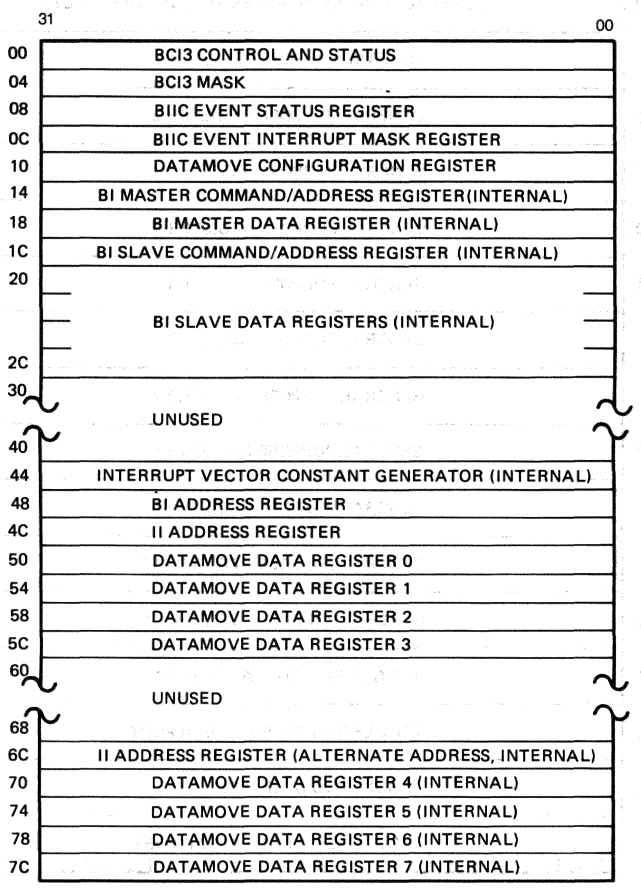


Figure 6 ▪ VAXBI 78733 Private Register Assignments

BCI3 Control and Status Register—The BCI3 control and status register (BCI3 CSR) contains 16 nodespace write sense bits, one for each of the nodespace register addresses shown in Figure 4. These bits are set when the BCI3 is accessed by another VAXBI bus node. When information is written into a nodespace register, an interrupt request (IPL 16) is generated and the corresponding bit in this register is set. The interrupt can be masked by the information in the control and status mask register. The register also contains control and status bits used by the datamover. The register format is shown in Figure 7 and the information is described in Table 5.

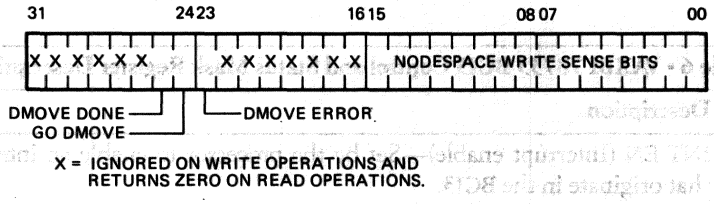


Figure 7 • VAXBI 78733 BCI3 Control and Status Register Format

Table 5 • VAXBI 78733 BCI3 Control and Status Register Description

Bit	Description
31:26	Not used.
25	DMOVE DONE (Datamove done)—When set, it indicates that the datamove operation completed successfully.
24	GO DMOVE (Go datamove)—Set to initiate a datamove operation.
23	DMOVE ERR (Datamove error)—When set, it indicates that the datamove operation was aborted due to an error. Cleared by writing a 1 to this bit.
22:16	Not used.
15:00	Nodespace write sense bits—Set when the corresponding nodespace registers have received data from other BI nodes. Cleared by writing a 1 to this bit.

BCI3 Control and Status Mask Register—The BCI3 control and status mask (BCI3 CSMR) register contains 16 nodespace write sense mask bits that are used to mask the sense bits in the BCI3 control and status register to prevent the generation of an interrupt. The register also contains bits to enable datamover interrupts and global interrupt enable bits. Figure 8 shows the register format and register information is described in Table 6.

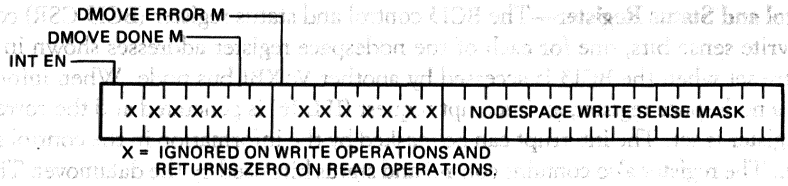


Figure 8 ▪ VAXBI 78733 BCI3 Control and Status Mask Register Format

Table 6 ▪ VAXBI 78733 BCI3 Control and Status Mask Register Description

Bit	Description
31	INT EN (Interrupt enable)—Set by the processor to enable an interrupt request that originate in the BCI3.
30:26	Not used.
25	DMOVE DONE M (Data move done mask)—Set by the processor to allow an interrupt when the corresponding bit in the control and status register is set.
24	Not used.
23	DMOVE ERROR M (Data move error mask)—Set by the processor to allow an interrupt when the corresponding bit in the control and status register is set.
22:16	Not used.
15:00	NODESPACE WRITE SENSE MASKS—Set to enable interrupts when the corresponding bits of the control and status register are set.

BIIC Event Status Register—The BIIC event status register (BIIC ESR) indicates the history of events specified on the VAXBI event lines $\overline{\text{BCI EV}<4:0>}$, except for the Interlock error indicator (bit 0). Bit 0 is set if the read lock fails after the BCI3 has read data from the II bus resulting in a II bus memory lock condition. The bits in this register are set by the BCI3 and cleared by the processor by writing a 1 into the required location. The BCI3 will interrupt the processor when a bit is set provided that the corresponding mask bit in the event mask register is set. The register format is shown in Figure 9 and the bits are defined in Table 7.

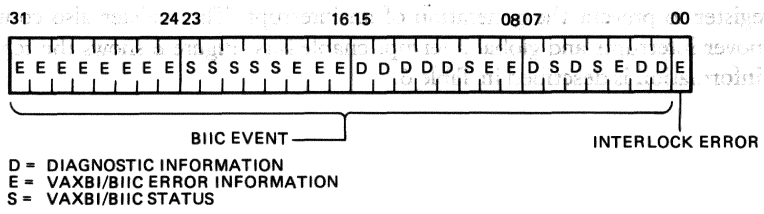


Figure 9 ▪ VAXBI 78733 BIIC Event Status Register Format

Table 7 • VAXBI 78733 BIIC Event Status Register Description

Bit	Description
31:01	Event Code bits—Specifies the BIIC events as follows:
31	MTCE (Master transmit error check)
30	BPM (Bad parity received during master-port transaction)
29	RTO (Retry timeout)
28	ICRMC (Illegal confirmation received for master-port command)
27	BPR (Bad parity received)
26	NCRMC (No acknowledge confirmation received for master-port command)
25	ICRMD (Illegal confirmation received by master-port data cycle)
24	RDSR (Read data substitute or reserved status code received)
23	AKRNE7 (Acknowledge confirmation received for nonerror vector—level 7)
22	AKRNE6 (Acknowledge confirmation received for nonerror vector—level 6)
21	AKRNE5 (Acknowledge confirmation received for nonerror vector—level 5)
20	AKRNE4 (Acknowledge confirmation received for nonerror vector—level 4)
19	BBE (Bus busy error)
18	ICRSD (Illegal confirmation received for slave data)
17	BPS (Bad parity received during slave transaction)
16	STO (Stall timeout on slave transaction)
15	EV7 (External vector selected—level 7)
14	EV6 (External vector selected—level 6)
13	EV5 (External vector selected—level 5)
12	EV4 (External vector selected—level 4)
11	IAL (Identification arbitration lost)
10	AKRE (Acknowledge confirmation received for error vector)
09	NICIPS (No acknowledge of illegal confirmation received for force-bit interprocessor/ stop command)
08	NICI (No acknowledge or illegal confirmation received for interrupt command)
07	ARCR (Advanced retry confirmation received)
06	IRW (Internal register written)
05	RCR (Retry confirmation received for master-port command)
04	STP (Self-test passed)

Bit	Description
03	BTO (Bus timeout)
02	AKRSD (Acknowledge received from slave read data)
01	MCP (Master-port transaction complete)
00	INT ERR (interlock error)—Set to interrupt the MicroVAX during an error condition provided that the corresponding mask bit is set in the BIIC event mask register.

BIIC Event Interrupt Mask Register—The BIIC event interrupt mask register (BIIC EIMR) is used to mask the event indications in the event status register to prevent the generation of an interrupt. The bit mask assignments are shown in Figure 10 and the event codes are defined in Table 7.

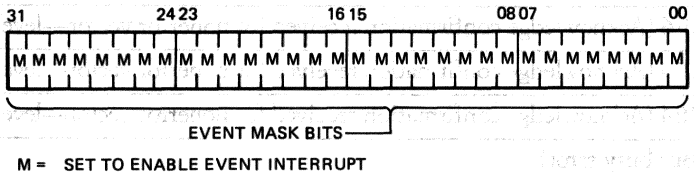


Figure 10 • VAXBI 78733 BIIC Event Interrupt Mask Register Format

Datamover Configuration Register—The datamover configuration register (DMCR) is used during datamove operations and to specify diagnostic mode operations. The register format is shown in Figure 11 and described in Table 8.

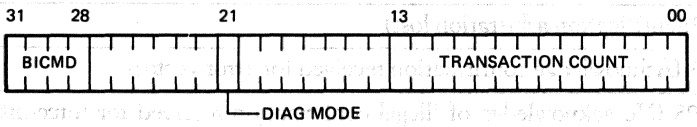


Figure 11 • VAXBI 78733 Datamover Configuration Register Format

Table 8 • VAXBI 78733 Datamove Configuration Register Description

Bit	Description
31:28	BI CMD (BI command)—Specifies a read or write BI command.
27:20	Not used.
21	DIAG MODE (Diagnostic mode)—Set to initiate diagnostic mode operation.
20:14	Not used.
13:00	Transaction count—Contains the two's complement of the number of octawords to be transferred (from 1 to 16K octawords) during the datamove.

BI Bus Address Register—The BI address register (BI AR) contains the starting VAXBI address and VAXBI transaction length code to be used by the datamover. The contents of this register will be altered during datamover operations. The register format is shown in Figure 12.

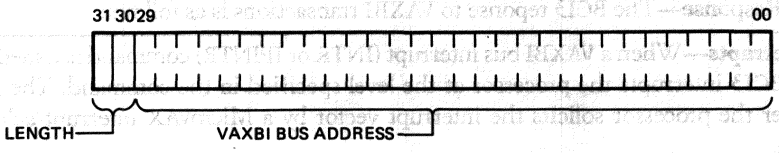


Figure 12 • VAXBI 78733 VAXBI Bus Address Register Format

II Bus Address Register—The II bus address register (II AR) contains the starting II bus address for the datamover. The contents of this register will be altered during the operation of the datamover. The register format is shown in Figure 13.

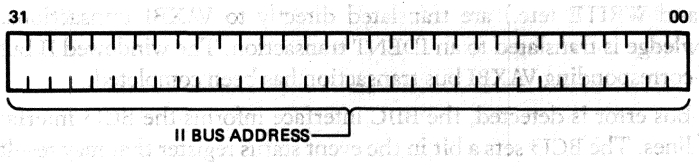


Figure 13 • VAXBI 78733 II Bus Address Register Format

Datamove Data Registers (0-3)—The datamove data registers (DDR0 through DDR3) are used as buffers to transfer octawords between the II bus and VAXBI bus during a datamove operation. Each register contains 32 bits.

VAXBI Transactions

All transactions between the BCI3 and VAXBI bus are facilitated through the BIIC interface that communicates directly with the BCI3. The BCI3 can be a master or a slave on the VAXBI bus. As a master, it performs the transactions requested by the processor. As a slave, it responds to the VAXBI transactions that are directed to it and translates the VAXBI bus transaction onto the II bus. When the transaction has been completed, the BCI3 reports the transaction status to the VAXBI master.

The BCI3 responds to VAXBI bus addresses in the 8-Kbyte region of I/O space (nodespace) and to a region specified by the starting and ending address register (configurable by the MicroVAX software). This software defined region typically resides in memory space but during certain conditions it may reside in the VAXBI I/O space.

The nodespace region starts at address 20000000+2000 (hexadecimal) node _ID defined as the nodespace base address (bb). Each of the 16 nodes than can be assigned to the VAXBI bus is identified by a node ID. The nodespace contains some registers that are defined by the VAXBI and some that are defined by the BIIC and BCI3 interfaces.

The remaining region is defined by the starting and ending address registers of the BIIC interface at locations bb + 20 and bb + 24, respectively. The BCI3 responds to addresses numerically greater than or equal to the starting address but less than the ending address. The low-order 18 bits of these registers must contain zeros so that the addresses are located in multiples of 256 Kbytes. The MicroVAX software should load these addresses during VAXBI node initialization.

Transaction Response—The BCI3 reponse to VAXBI transactions is as follows:

- **VAXBI Interrupts**—When a VAXBI bus interrupt (INTR or IPINTR) command is issued to a local node, the BCI3 interrupts the processor at the level specified in the command. The request is cleared after the processor solicits the interrupt vector by a MicroVAX interrupt acknowledge transaction.
- **VAXBI Identification**—The BIIC interface responds directly to IDENT transactions and the BCI3 is not involved in this transfer.
- **VAXBI Invalidate**—The BCI3 ignores the VAXBI INVAL transactions.
- **VAXBI Stop**—The BCI3 acknowledges STOP transactions but performs no operations. External user logic may decode this transaction for node specific responses.

II Bus to VAXBI Bus Translations

All II bus transactions directed to the VAXBI are transparent to the processor software. The READ, READ LOCK, and WRITE (etc.) are translated directly to VAXBI transactions. A MicroVAX interrupt acknowledge is translated to an IDENT transaction. The windowed II bus transaction is stalled until the corresponding VAXBI bus transaction has been completed.

When a VAXBI bus error is detected, the BIIC interface informs the BCI3 interface through the $\overline{\text{BCI EV}}$<4:0> lines. The BCI3 sets a bit in the event status register that may result in a processor interrupt. Critical errors such as BI master transaction errors can terminate the corresponding BCI3 operation. The BCI3 notifies the processor of errors detected during windowing operations using the $\overline{\text{II ERR}}$ signal. Errors detected during datamove operations cause the BCI3 to abort the operation and to set the datamove error bit in the datamove register.

II Bus Read Translations—The II bus to VAXBI bus read translations are

II Bus command	VAXBI Bus command
Read (I stream)	READ
Read Lock	READ LOCK
Read (D stream, modify intent)	READ
Read (D stream, no modify intent)	READ

The length of the VAXBI-generated transaction is a longword with D<31:30> equal to 01 in the command/address cycle. Bits D<29:00> are transferred to the VAXBI bus. VAXBI nodes usually ignore bits D<01:00>.

The BCI3 generates a II bus error signals when

- The VAXBI transaction is not acknowledged or exceeds the time specified for completion.
- Invalid parity is detected or an illegal confirmation code is received.
- A read data substitute or reserved read status code is received from the VAXBI bus.

II Bus Write Translations—The II bus to VAXBI bus write translations are

II Bus command	VAXBI Bus command
Write (D stream)	WRITE MASK
Write unlock	WRITE UNLOCK

The write (D stream) transaction is translated into a VAXBI WRITE MASK and a II write unlock is always translated into a VAXBI WRITE UNLOCK transaction. The length of the generated VAXBI transaction is always a longword (D<31:30> =01) in the VAXBI command address cycle. Bits <29:00> of the II bus address are transferred to the VAXBI address.

The $\overline{\text{II ERR}}$ signal is asserted by the BCI3 by the following:

- The VAXBI transaction times out.
- The VAXBI transaction is not acknowledged or receives an illegal confirmation.
- Invalid parity or VAXBI bus errors are detected on the VAXBI bus.

II Bus Interrupt Acknowledge Translations—When the II bus issues an interrupt acknowledge, the BCI3 issues a VAXBI IDENT command if a VAXBI interrupt (INTR) is pending at the priority level specified by the acknowledge. The vector returned by the IDENT command completes the II bus interrupt acknowledge transaction. The VAXBI interrupt pending state is cleared at the completion of the transaction or when an error is detected. The BCI3 generates a II bus error signal when

- The IDENT command is not acknowledged or exceeds the time specified for completion.
- The BCI3 receives an illegal VAXBI confirmation code.
- Parity is not valid or an illegal confirmation code is received.
- A read data sustitute or reserved vector status code is received from the VAXBI bus.

VAXBI Bus to II Bus Transactions

The VAXBI bus read and write commands that have an address between the starting and ending address register values of the BIIC interface are translated to II bus read or write commands. VAXBI bus quadword-length transactions are translated into two longword length transactions. VAXBI bus octaword transactions are translated into four longword-length transactions. Table 10 lists the translation between the VAXBI bus command and II bus transactions.

Table 9 • VAXBI 78733 VAXBI Command to II Bus Command Transactions

VAXBI Bus commands	II Bus commands
READ	Read (D stream, no modify)
INTERLOCK READ*	Read lock
READ WITH CACHE INTENT	Read (D stream, no modify)
WRITE	Write (D stream)
WRITE WITH CACHE INTENT	Write (D stream)
UNLOCK WRITE MASK	Write unlock
WRITE MASK	Write (D-stream)

*Commands of quadword length are translated into a II bus read lock transaction followed by a Read (D stream, no modify) transaction. Commands of octaword length are translated into a II bus read lock transaction followed by three Read (D stream, no modify) transactions.

Read Translations—During read-type transactions from the VAXBI bus, the BCI3 will normally stall the VAXBI node until the translated transaction has completed. When the II bus memory controller transfers the requested data signified by the assertion of the $\overline{\text{II RDY}}$ signal, the BCI3 returns the data through the VAXBI bus with a "read data don't cache" status. If the memory controller responds with an error ($\overline{\text{II ERR}}$) or retry ($\overline{\text{II RETRY}}$) signal, the BCI3 returns the appropriate no acknowledge or retry confirmation on the VAXBI bus and terminates the transaction.

Write Translations—The BCI3 acknowledges a VAXBI write-type transaction when the data has been successfully received but before the completion of the actual write operation. Write transactions that fail because of nonexistent memory and other conditions are not reported to the VAXBI. This disconnecting of write transactions allows the VAXBI bus to participate in other transactions while the BCI3 completes the write (detached) transaction to the II bus. If the BCI3 receives another VAXBI read- or write-type transaction before the last write operation has been completed, it will retry all subsequent VAXBI transactions until the detached write transaction has completed.

Error Handling

The following events occur during an error condition.

- When an error occurs on the II bus side during a VAXBI bus to II bus window read operation, the BCI3 indicates an error by issuing a no acknowledge reply during the command/address cycle of the VAXBI read transaction. All II bus errors are specified by the assertion of the $\overline{\text{II ERR}}$ signal.
- When an error occurs on the II bus side during a VAXBI bus to II bus window write operation, the BCI3 inhibits additional II bus write transactions that are normally required for a multilongword write transfer. No error will be reported to the VAXBI bus.
- The $\overline{\text{II ERR}}$ signal is used to inform the MicroVAX that a II bus to VAXBI bus windowed transaction has been completed unsuccessfully. It is not asserted to indicate an error resulting from an incoming VAXBI bus transactions.
- If an error indication is received during an incoming VAXBI to II bus write transaction, no writing will occur on the II bus.
- If the VAXBI read lock fails after the BCI3 has read the data from the II bus, the BCI3 will not unlock the II bus memory.
- During other VAXBI to II bus conditions such as nonlocking read transactions, a bit in the event status register will be set to interrupt the processor if the corresponding interrupt mask bit has not been set.

Special Operations

The BIC3 interface has the capabilities to perform special operations to improve the efficiency of data transactions and for diagnostic test purposes. The datamove command allows the BCI3 to transfer a sequence of octawords between the II bus and VAXBI bus.

Datamove Operations

During datamove operations, the MicroVAX causes the BCI3 to transfere a sequence of octawords between the II bus and VAXBI bus. The BCI3 loads an internal buffer with octawords from one bus and transfers them to the other bus. When completed, the BCI3 interrupts the processor. Because the datamove operation uses the VAXBI octaword transfer, it is more efficient than the MicroVAX MOVx command that transfers sequences of individual longwords. In addition, it allows the processor to perform other operations when the datamove transaction is in process.

If a RETRY request from the VAXBI bus is received during the BI transaction, the BCI3 will repeat the transaction until it is successfully completed or until a timeout condition occurs. If the transaction is not acknowledged or times out, the BCI3 will set the DMOVE ERROR bit in the control and status register. The cause of the failure must be determined from the BCI3 and BIIC interface status registers.

The datamove command uses the datamove configuration register, control and status register, control and status mask register, II bus address register, and the BI bus address register in the BCI3. The data to be transferred must be aligned by octawords and the source and destination blocks may not cross a 256-Kbyte address boundary.

Data is transferred from the II bus to the VAXBI bus by the following operations. The data to be transferred must be octaword aligned on the II bus and VAXBI bus.

- The starting address of the of II bus data (source) is loaded into the BCI3 II bus address register by the microprocessor.
- The starting address of the VAXBI address (destination) is loaded into the VAXBI bus address register in the BCI3 by the microprocessor.
- The BI WRITE command (code 0100) is written into the BICMD field and the two's complement of the the number of octawords to be moved is written into the transfer count field of the datamove configuration register by the processor.
- The GO DMOV bit of the BCI3 control and status register is set by the microprocessor to initiate the data move operation.
- The BCI3 sets the DMOV DONE bit in the control and status register at the end of the transaction or sets the DMOV ERROR bit during the transaction if an error detected during the transaction. These bits can generate a microprocessor interrupt If enabled by the corresponding mask bits in the controland status mask register.

The sequence required to transfer data from the VAXBI bus to the II bus is similar to the previous transfer except that a BI READ command code (0001) is written into the BICMD field of the data configuration register by the microprocessor.

Application Information

Refer to the *BCI3 Engineering Specification (DC344)* for more detailed information of the operation of the VAXBI 78733 interface, the *VAXBI System Reference Manual* (document number EK-VBISY-RM) for information on the VAXBI bus operation, and the *VAXBI 78732 Bus Interconnect Interface Chip* (BIIC) information contained in this databook.

Figure 14 shows a typical system implementation consisting of three microprocessors. Each microprocessor communicates with memory and devices through its II bus and to the VAXBI bus through the BCI3 and BIIC interface.

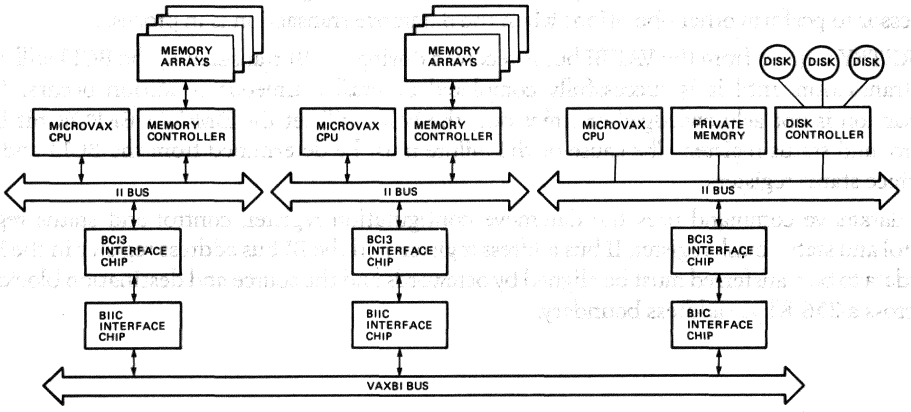


Figure 14 • VAXBI 78733 Typical System Configuration

The II bus is a 10 Mbyte-per-second private bus that communicates with the 13-Mbyte-per-second VAXBI system bus through the BCI3 and BIIC interface structure. Both buses implement a common 1-Gbyte address space. All resources with the exception of certain nodespace locations can be accessed through the uniform addressing arrangement.

Specifications

The mechanical, electrical, and environmental specifications for the VAXBI interface are as follows. The test conditions for the values specified are as follows unless specified otherwise.

- Temperature: 70°C
- Supply voltage (V_{cc}): 4.75 V
- Ground (V_{ss}): 0 V

Mechanical Configuration

The mechanical dimensions for mounting the 132-pin PGA interface package are shown in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to a device. Exposure to the absolute maximum rating conditions for extended periods may adversely affect the reliability of the device.

- Input and output voltages applied: –1.0 V to 10 V
- Active temperature range: 0°C to 125°C
- Power supply voltage: 0.5 V to 7.0 V

Recommended Operating Conditions

- Power supply voltage range (V_{cc}): 4.75 V to 5.25 V
- Temperature range: 0°C to 70°C

dc Electrical Characteristics

Table 10 contains the dc electrical parameters for the input and outputs of the BCI3 interface chip.

Table 10 • VAXBI 78733 dc Input and Output Parameters

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units
High-level input voltage	V_{IH}		2.0	—	V
Low-level input voltage	V_{IL}		-0.5	0.8	V
High-level output voltage	V_{OH}	$I_{OH} = -400$ A	2.4	—	V
Low-level output voltage	V_{OL}	$I_{OL} = 2.4$ mA $I_{OL} = 4.0$ mA*	—	0.4	V
Input leakage current	I_{IL}	$0 V < V_{in} < 5.25 V$	-20	20	A
Output high-impedance leakage current	I_{ZO}	$0 V < V_{in} < 5.25 V$	-20	20	A
Supply current	I_{CC}		—	500	mA
Input capacitance	C_{in}	BCI $\overline{MDE}$, BCI $\overline{SDE}$ all other signals	— —	50 10	pF pF
Output capacitance	C_{out}		—	10	pF

*Open-drain outputs

▪ ac Timing Specifications

The timing for the ac parameters that follow are measured with a 100-pF capacitive load on each of the outputs.

Figures 15 and 16 show the ac timing for the signals used during II bus transactions when the BCI3 is operating as a slave. Figure 15 is a read and interrupt acknowledge transaction diagram. Figure 16 is a write transaction diagram. Table 11 lists the timing parameters for the symbols shown in the figures.

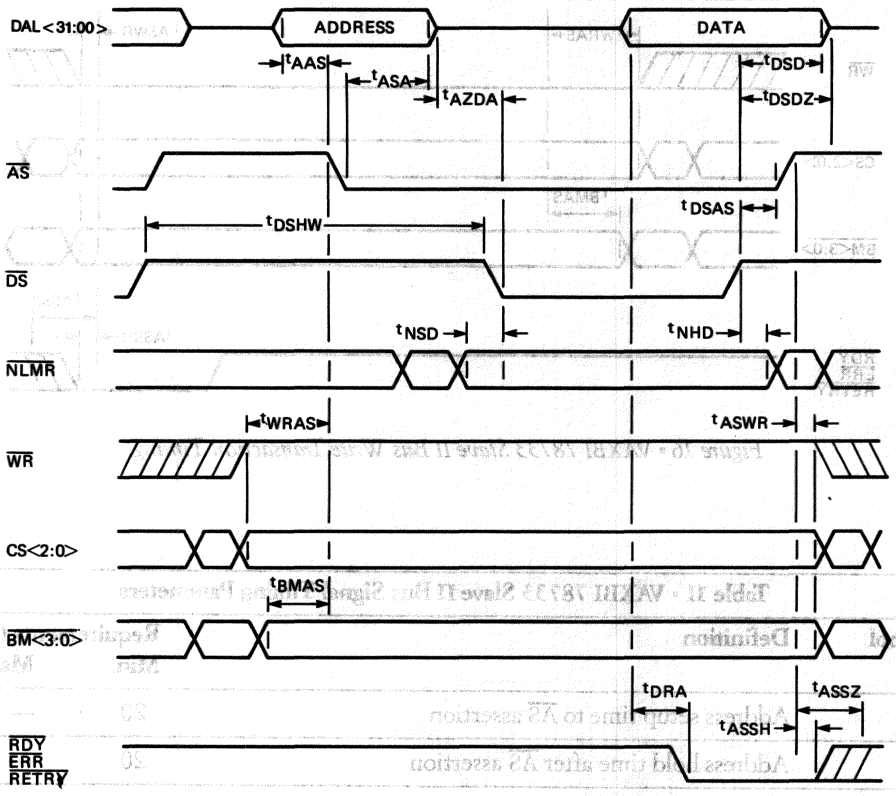


Figure 15 ▪ VAXBI 78733 Slave II Bus Read and Interrupt Acknowledge Transaction Timing

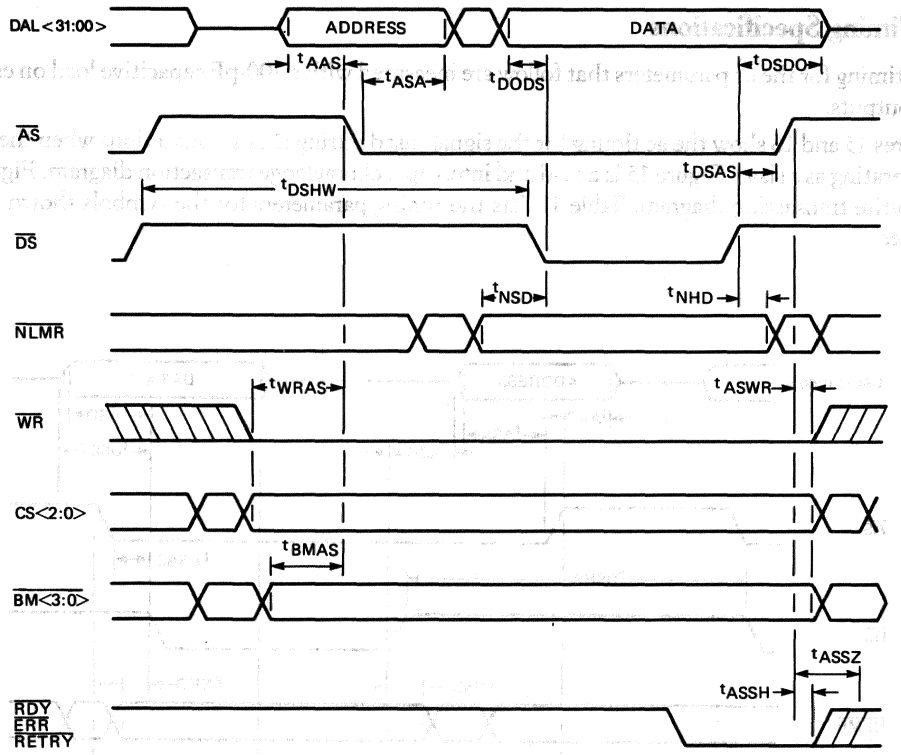


Figure 16 ▪ VAXBI 78733 Slave II Bus Write Transaction Timing

Table 11 ▪ VAXBI 78733 Slave II Bus Signal Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{AAS}	Address setup time to $\overline{AS}$ assertion	20	—
t_{ASA}	Address hold time after $\overline{AS}$ assertion	20	—
t_{ASSH}	Status hold from $\overline{AS}$ negation	0	—
t_{ASSZ}*	Status high impedance from $\overline{AS}$ negation	—	40
t_{ASWR}	$\overline{WR}$, $\overline{BM<3:0>}$, CS<2:0> hold time from $\overline{AS}$ negation	0	—
t_{AZDA}	Address high impedance to $\overline{DS}$ (read-only)	0	—
t_{BMAS}	$\overline{BM<3:0>}$ setup time before $\overline{AS}$ assertion	20	—
t_{DODS}	Write data setup time before $\overline{DS}$ assertion	0	—

Symbol	Definition	Requirements (ns) Min.	Max.
t_{DRA}	Read data setup to $\overline{RDY}$ assertion	10	—
t_{DSAS}	$\overline{DS}$ negation to $\overline{AS}$ negation	0	—
T_{DSD}	Read data hold time after $\overline{DS}$ negation	0	—
t_{DSDO}	Write data hold time from $\overline{DS}$ negation	0	—
t_{DSDZ}	$\overline{DS}$ negation to read data three-state	—	55
t_{DSHW}	$\overline{DS}$ deassertion width	150	—
t_{NHD}	$\overline{NLMR}$ hold from $\overline{DS}$ negation	0	—
t_{NSD}	$\overline{NLMR}$ setup to $\overline{DS}$ assertion	0	—
t_{WRAS}	$\overline{WR}$, CS<2:0> setup time before $\overline{AS}$ assertion	20	—

*Designed for but not tested.

Figures 17 and 18 show the ac timing for the signals used during II bus transactions when the BCI3 is operating as a master. Figure 17 is a read transaction diagram. Figure 18 is a write transaction diagram. Table 12 lists the timing parameters for the symbols shown in the figures.

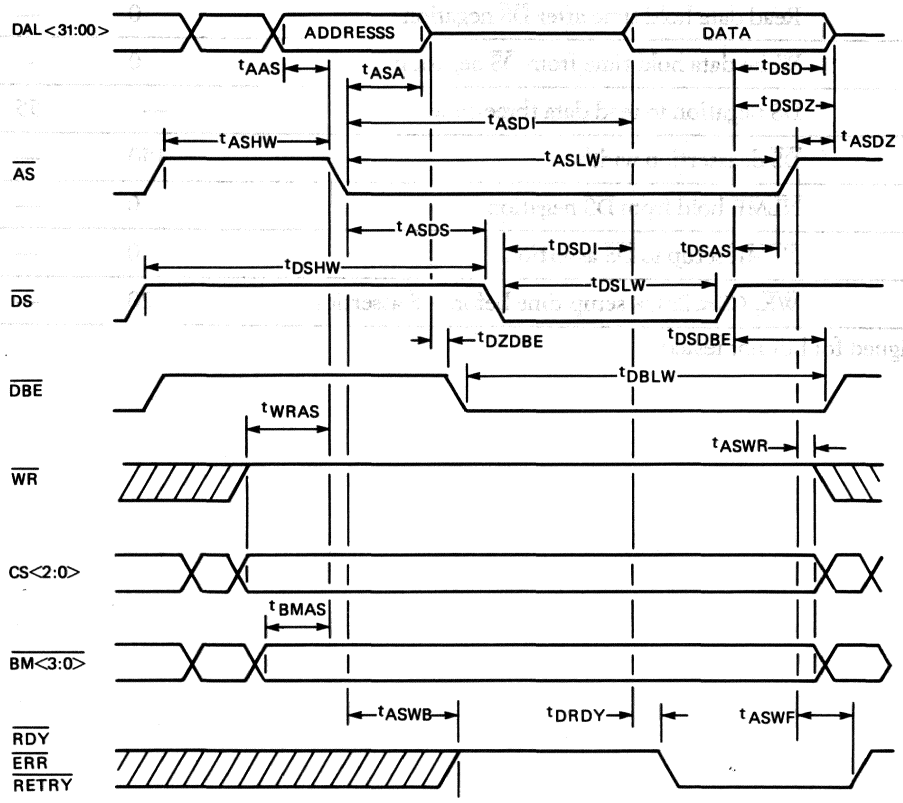


Figure 17 • VAXBI 78733 Master II Bus Read Transaction Timing

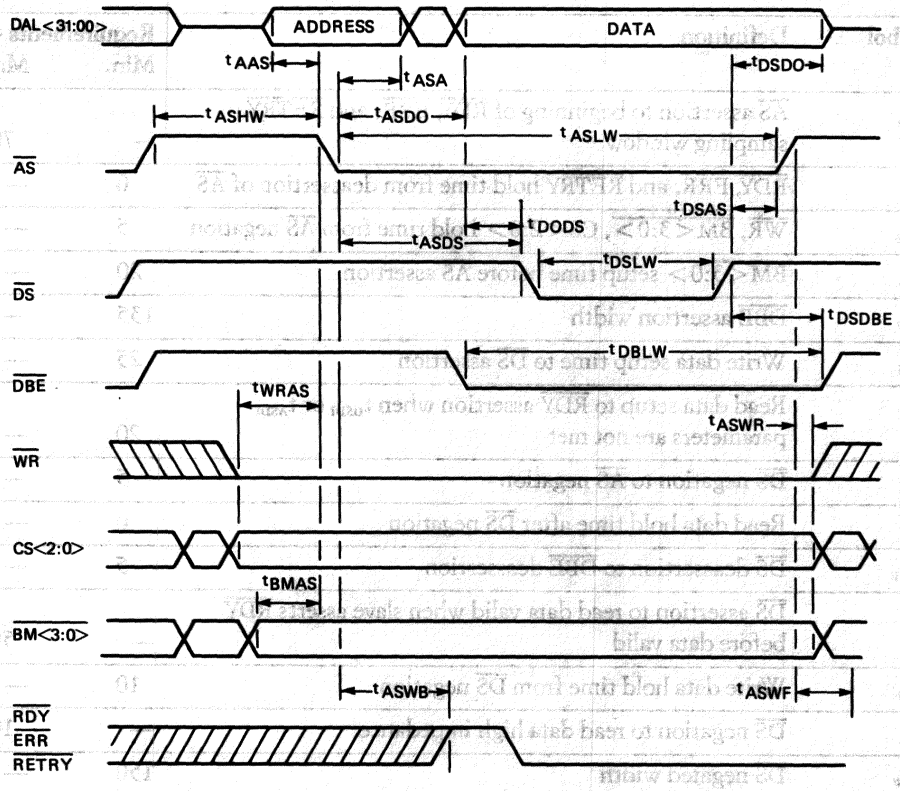


Figure 18 • VAXBI 78733 Master II Bus Write Transaction Timing

Table 12 • VAXBI 78733 Master II Bus Signal Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{AAS}	Address setup time to $\overline{AS}$ assertion	20	—
t_{ASA}	Address hold time after $\overline{AS}$ assertion	20	—
t_{ASDI}	$\overline{AS}$ assertion to read data valid (only if slave asserts data $\overline{RDY}$ before data valid)	—	220
t_{ASDO}	$\overline{AS}$ assertion to data valid	—	105
t_{ASDS}	$\overline{AS}$ assertion to $\overline{DS}$ assertion	120	—
t_{ASDZ}	$\overline{AS}$ negation to read data high impedance	—	5.0
t_{ASHW}	$\overline{AS}$ negated width	75	—
t_{ASLW}	$\overline{AS}$ asserted width	245	—

Symbol	Definition	Requirements (ns) Min.	Max.
t_{ASWB}	$\overline{AS}$ assertion to beginning of $\overline{RDY}$, $\overline{ERR}$, and $\overline{RETRY}$ sampling window	—	70
t_{ASHF}	$\overline{RDY}$, $\overline{ERR}$, and $\overline{RETRY}$ hold time from deassertion of $\overline{AS}$	0	—
t_{ASWR}	$\overline{WR}$, $\overline{BM}$<3:0>, CS<2:0> hold time from $\overline{AS}$ negation	5	—
t_{BMAS}	$\overline{BM}$<3:0> setup time before $\overline{AS}$ assertion	20	—
t_{DBLW}	$\overline{DBE}$ assertion width	135	—
t_{DODS}	Write data setup time to $\overline{DS}$ assertion	25	—
t_{DRDY}	Read data setup to $\overline{RDY}$ assertion when t_{DSDI} or t_{ASDI} parameters are not met	20	—
t_{DSAS}	$\overline{DS}$ negation to $\overline{AS}$ negation	5	—
t_{DSD}	Read data hold time after $\overline{DS}$ negation	0	—
t_{DSDBE}	$\overline{DS}$ deassertion to $\overline{DBE}$ deassertion	5	—
t_{DSDI}	$\overline{DS}$ assertion to read data valid when slave asserts $\overline{RDY}$ before data valid	—	50
t_{DSDO}	Write data hold time from $\overline{DS}$ negation	10	—
t_{DSDZ}	$\overline{DS}$ negation to read data high impedance	—	10
t_{DSHW}	$\overline{DS}$ negated width	150	—
t_{DSLW}	$\overline{DS}$ assertion width	80	—
t_{DZDBE}	DAL<31:00> three-state to $\overline{DBE}$ assertion	0	—
t_{WRAS}	$\overline{WR}$, CS<2:0> setup time before $\overline{AS}$ assertion	20	—

Figure 19 shows the DMA request and grant signal timing when the BCI3 requests the use of the II bus. Table 13 lists the timing parameters.

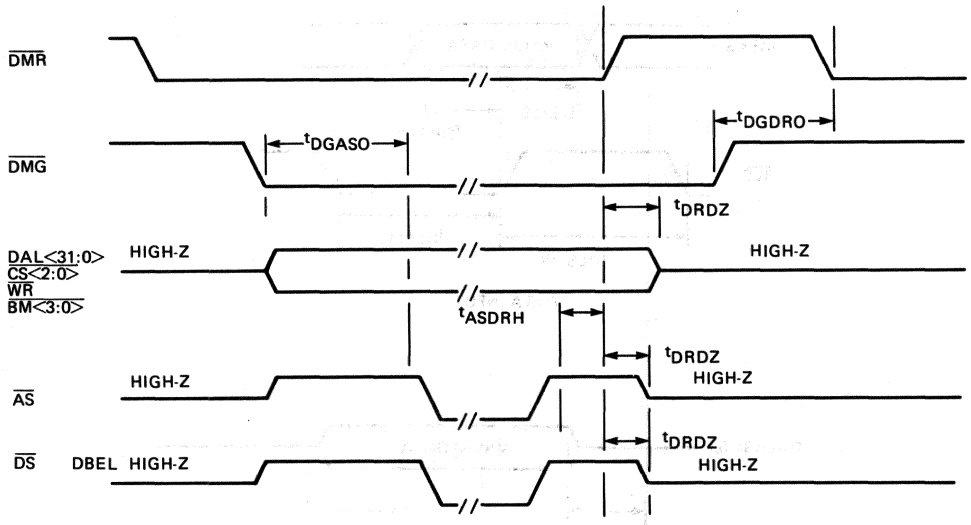


Figure 19 • VAXBI 78733 DMA Request and Grant Signal Timing

Table 13 • VAXBI 78733 DMA Transaction Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{DRDZ}	$\overline{DMR}$ deassertion to output high impedance	50	—
t_{ASDRH}	$\overline{DMR}$ hold time from $\overline{AS}$ assertion	0	—
t_{DGASO}	$\overline{DMG}$ assertion to $\overline{AS}$ assertion	210	—
t_{DGDRO}	$\overline{DMG}$ deassertion to $\overline{DMR}$ assertion	0	—

Figure 20 shows the signal timing for data transfers to and from the retry address latch (register). Table 14 lists the timing parameters.

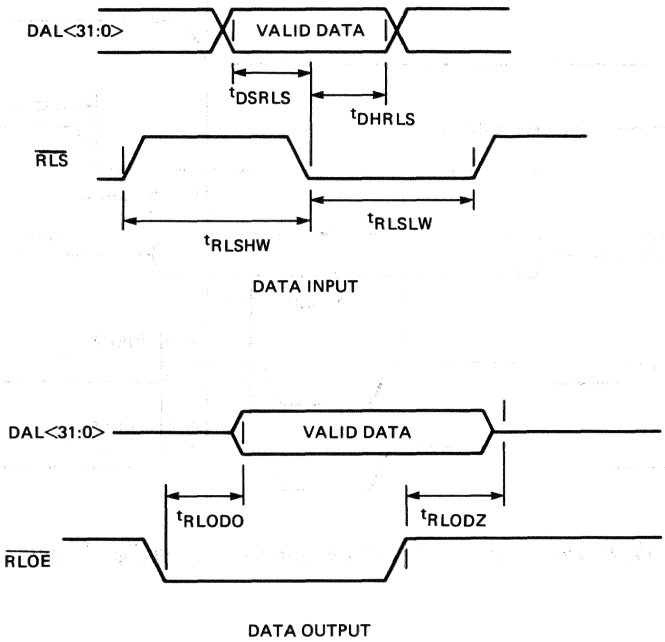


Figure 20 ▪ VAXBI 78732 Retry Address Latch Input/Output Signal Timing

Table 14 ▪ VAXBI 78733 Retry Address Latch Input/Output Timing Parameters

Symbol	Definition	Requirements (ns) Min.	Max.
t_{DSRLS}	Data setup to $\overline{RLS}$ assertion	20	
t_{DHRLS}	Data hold after $\overline{RLS}$ assertion	20	
t_{RLSHW}	$\overline{RLS}$ deassertion width	100	
t_{RLSLW}	$\overline{RLS}$ assertion width	100	
t_{RLODO}	$\overline{RLOE}$ assertion to valid		60
t_{RLODZ}	$\overline{RLOE}$ assertion to data high impedance		50

Features

- Custom 14-pin DIP bipolar integrated circuit
- Drives 16 clock receivers distributed over a maximum 1.524 meters of etch
- Single 5-V power supply
- Differential ECL drive outputs to both BI TIME +/− and BI PHASE +/−
- Uses an externally applied 40-MHz crystal oscillator input
- Provides TTL outputs and differential ECL outputs

Description

The VAXBI 78701 clock driver is a custom-built 14-pin DIP bipolar integrated circuit (IC) that serves as the clock source in VAXBI systems. The clock driver is designed to drive 16 clock receivers distributed over a maximum 5 feet of etch. The device requires only a 5 Vdc operating voltage. Figure 1 is a functional diagram of the clock driver.

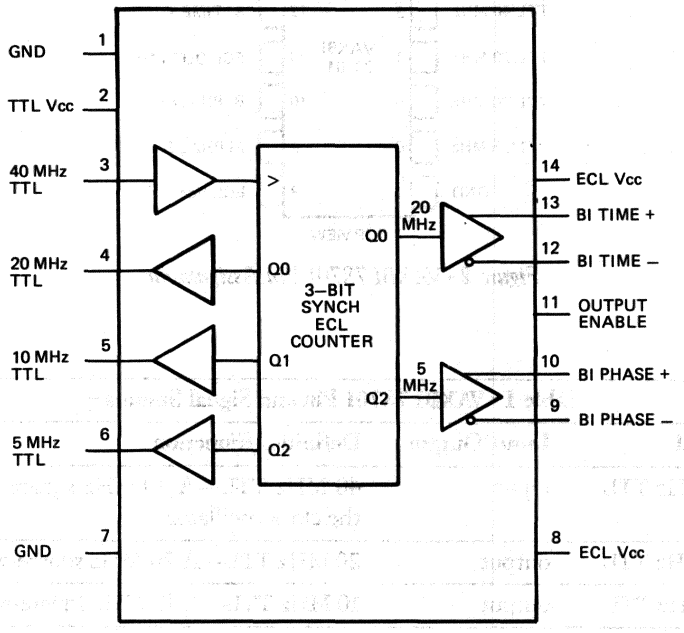


Figure 1 • VAXBI 78701 Clock Driver Logic

The circuit provides differential ECL drive outputs and TTL outputs from an externally applied 40-MHz crystal oscillator. The ECL outputs drive both VAXBI bus BI TIME +/− and BI PHASE +/− inputs. The TTL outputs are not recommended for use in a VAXBI system.

The VAXBI 78702 clock receiver must be used with each VAXBI clock driver source to provide the proper VAXBI clock receive function if the driving source of the VAXBI clock is also a VAXBI node. The clock driver must be at the electrical beginning of the bused clock etch lines.

An output enable included with the device must be used if more than one node can be installed into the drive slot of the VAXBI bus. Only one source of clocks in VAXBI systems is allowed and the first slot should be wired to enable this device.

Pin and Signal Descriptions

The input and output signals and the power and ground connections of the clock driver are shown in Figure 2. Table 1 provides a summary of the input and output signals.

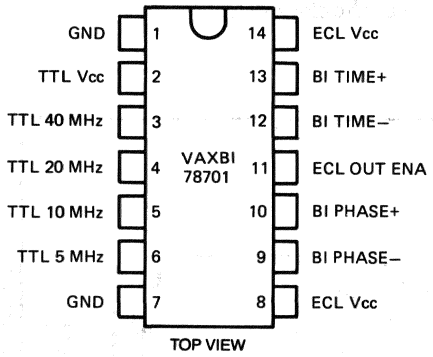


Figure 2 ▪ VAXBI 78701 Pin Assignments

Table 1 ▪ VAXBI 78701 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
3	40 MHz TTL	input	40 MHz TTL—A 40-MHz square-wave input from the clock oscillator.
4	20 MHz TTL	output	20 MHz TTL—A 20-MHz square-wave output.
5	10 MHz TTL	output	10 MHz TTL—A 10-MHz square-wave output.
6	5 MHz TTL	output	5 MHz TTL—A 5-MHz square-wave output to logic functions.
11	ECL ENA	input	ECL enable—Enables the ECL outputs when at a high level. Can be connected to ECL V_{CC}.

Pin	Signal	Input/Output	Definition/Function
9 10	BI PHASE – BI PHASE +	outputs	BI PHASE +/– (5 MHz)—Differential ECL outputs for the phase signal inputs to the VAXBI 78702 clock receiver and VAXBI bus.
12 13	BI TIME – BI TIME +	output	BI TIME +/– (20 MHz)—Differential ECL outputs for the time signal inputs to the VAXBI 78702 clock receiver and VAXBI bus.
2	TTL V_{CC}	input	TTL voltage—Power supply voltage to TTL logic.
8,14	ECL V_{CC}	input	ECL voltage—Power supply voltage to ECL logic.
1,7	GND*	input	Ground—Common ground reference.

*The two GND pins are not connected internally.

• Application

A typical VAXBI clock driver and clock receiver system configuration is shown in Figure 3. The clock driver receives the 40 MHz TTL input from an external clock oscillator. An internal 3-bit synchronous ECL counter provides the differential 5-MHz BI PHASE +/– ECL outputs and differential 20-MHz BI TIME +/– outputs for the clock receiver. It also supplies a 20-MHz, 10-MHz and 5-MHz TTL square-wave output. Refer to the VAXBI 78702 clock receiver for a description of the VAXBI 78702 clock receiver functions.

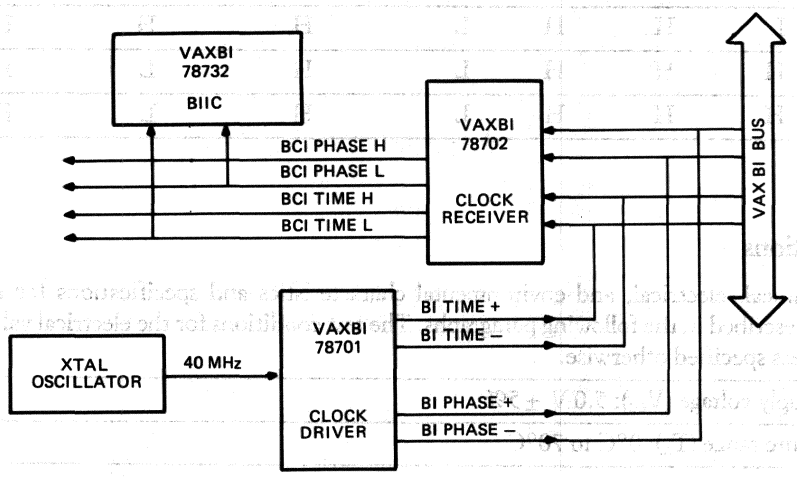


Figure 3 • VAXBI 78701 Clock Driver with VAXBI 78702 Clock Receiver System Configuration

Table 2 lists the relationship of the input and output levels of the VAXBI 78701 clock driver. The states of the inputs and outputs at startup time are not defined.

Table 2 • VAXBI 78701 Input and Output Signal Levels

TTL Input	TTL Outputs			ECL Outputs			
40 MHz	20 MHz	10 MHz	5 MHz	BI PHASE +	BI PHASE –	BI TIME +	BI TIME –
L	L	L	L	H	L	H	L
H	L	L	L	H	L	H	L
L	H	L	L	H	L	L	H
H	H	L	L	H	L	L	H
L	L	H	L	H	L	H	L
H	L	H	L	H	L	H	L
L	H	H	L	H	L	L	H
H	H	H	L	H	L	L	H
L	L	L	H	L	H	H	L
H	L	L	H	L	H	H	L
L	H	L	H	L	H	L	H
H	H	L	H	L	H	L	H
L	L	H	H	L	H	H	L
H	L	H	H	L	H	H	L
L	H	H	H	L	H	L	H
H	H	H	H	L	H	L	H

• Specifications

The mechanical, electrical, and environmental characteristics and specifications for the clock driver are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Power supply voltage (V_{CC}): 5.0 V ±5%
- Temperature range (T_A): 0°C to 70°C

Mechanical Configuration
The physical dimensions of the 78701 14-pin CERDIP package are contained in Appendix E.

Absolute Maximum Ratings
Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Power supply voltage (V_{CC}): 7.0 V
- Input voltage applied: −1.5 V to 7.0 V (40 MHz TTL)
- Output voltage applied: 3.0 V to 7.0 V (Differential ECL)
- TTL output applied current: 40 mA (low state)
- Active temperature (T_A): 0°C to 70°C
- Storage temperature: −55°C to 125°C

Recommended Operating Conditions

- Power supply voltage (V_{CC}): 5 V ±5%
- Ambient temperature (T_A): 0°C to 70°C
- Input frequency: 0 to 50 MHz
- Power dissipation: approximataly 500 mW (no load)

dc Electrical Characteristics
The dc electrical parameters of the clock driver for the operating voltage and temperature ranges specified and when at thermal equilibrium are listed in Table 3 and 4. All dc specifications, including capacitance values, pertain to packaged parts. Table 4 lists the TTL input and output parameters and Table 5 lists the ECL input and output parameters. Figure 4 shows the dc test circuits use for the measurements. The ECL output load resistor values used to test this device are not the same as the VAXBI clock termination values.

Table 3 • VAXBI 78701 dc TTL Input and Output Parameters

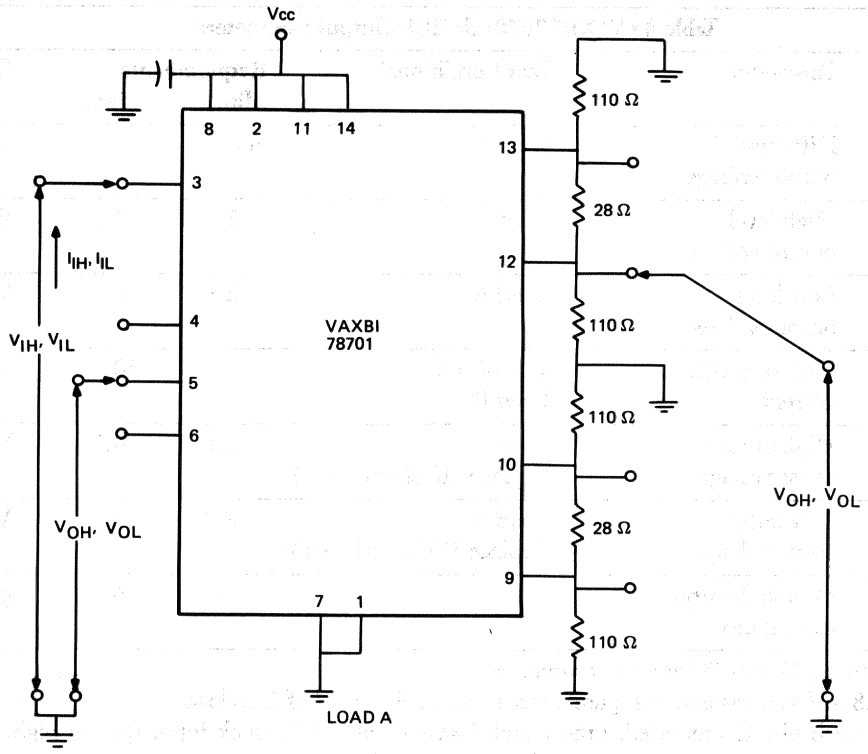
Symbol	Parameter	Test Conditions	Requirements Min.	Max.	Unit
V_{IH}	High-level input voltage	V_{CC}=5.25 V Pin 3	2	—	V
V_{IL}	Low-level input voltage	V_{CC}=5.25 V Pin 3	—	0.8	V
I_{IH}	High-level input current	V_{IH}=2.7 V V_{CC}=5.25 V Pin 3	—	20	µA
I_{IL}	Low-level input current	V_{IL}=0.5 V V_{CC}=5.25 V Pin 3	—	-2	mA
V_{IC}	Input clamp voltage	I_{in}=-18 mA V_{CC}=5.25 V Pin 3	—	-1.2	V
I_{INBV}	High-level input current breadown	V_{IH}=7 V V_{CC}=5.25 V Pin 3	—	100	µA
C_{in}	Input capacitance	V_{IH}=4 V Pin 3	—	12	pF
V_{OH}[1]	High-level output voltage	I_O=I_{OH} max. V_{CC}=4.75 V Pins 4-6	2.4	—	V
V_{OL}[1]	Low-level output voltage	I_O=20 mA V_{CC}=4.75 V Pins 4-6	—	0.5	V
I_{OH}[1]	High-level output current	V_O=V_{OH} min. Pins 4-6 V_{CC}=4.75 V	—	-1000	µA
I_{OL}[1]	Low-level output current	V_O=0.5 V V_{CC}=4.75 V Pins 4-6	20	—	mA
I_{OS}[2]	Short circuit output current	V_{CC}=5.25 V Pins 4-6	-25	-150	mA
I_{CC}	Power supply current	V_{CC}=5.25 V No load Pin 2	—	100	mA

[1]Testing either (V_{OH} and V_{OL}) or (I_{OH} and I_{OL}) satisfies these requirements.
[2]TTL outputs shorted to GND, ECL outputs shorted to V_{CC}, or either outputs open a maximum duration of 5 minutes. I_{OS} is current into the output pin when the input conditions cause the outputs be a logic one before the application of the short.

Table 4 • VAXBI 78701 dc ECL Output Parameters

Symbol	Parameter	Test Conditions[1]	Requirements Min.	Requirements Max.	Unit
V_{OD}[1,2]	Differential output voltage	Load A	700	—	mV
V_{OH}	High-level output voltage	Load A	3.5	4.3	V
V_{OL}	Low-level output voltage	Load A	2.8	3.5	V
I_{OZ}[3]	Output disable current	V_{OZ}=4.5 V Load B	—	50	µA
V_{OHS}	High output stress voltage	Load A Replace 110 Ω with 60 Ω	3.5	4.3	V
V_{OLS}	Low output stress voltage	Load A Replace 110 Ω with 60 Ω	2.8	3.5	V
C_{OZ}[4]	Disabled output capacitance		—	6	pF

[1]Pins 9 and 10, 12 and 13 for all measurements
[2]V_{OD} is 318 mV as measured in Figure 4 across points A and A′ of ECL load.
[3]Perform with pin 11 grounded, pins 1 and 7=GND, and TTL clock input (pin 3) high. Also performed with pins 2, 8, and 14 grounded and at 5.0 V.
[4]Difference in output capacitance (C_{OZ}+ minus C_{OZ}−)=2 pF



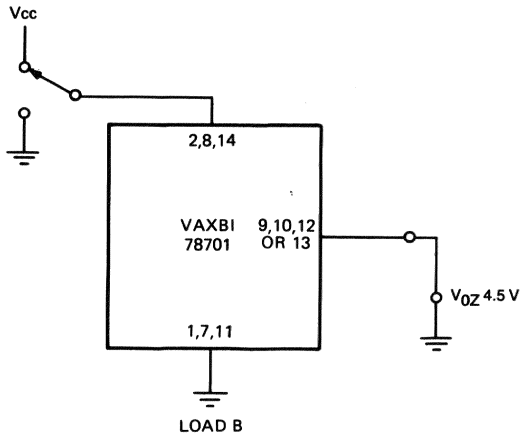
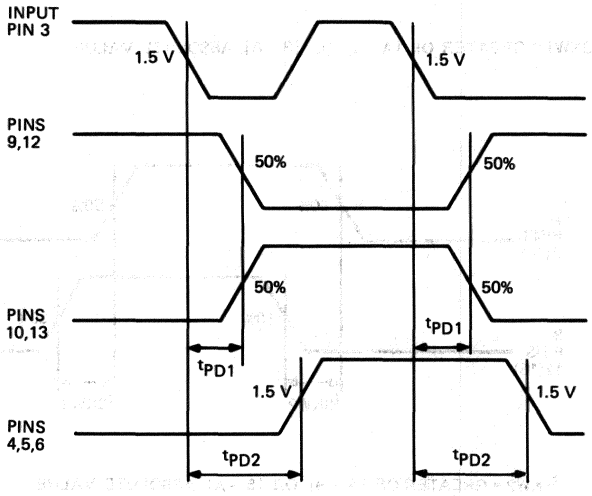


Figure 4 ▪ dc Test Circuits

ac Electrical Characteristics

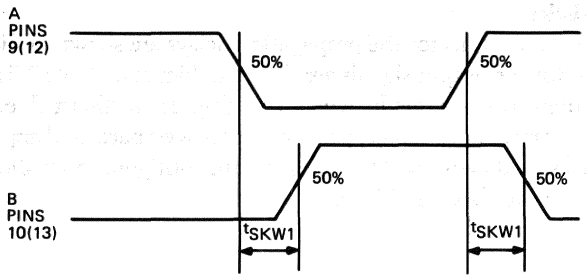
The timing symbols and waveforms for the propagation delays are shown in Figure 5. The timing symbols and waveforms for the output signals are shown in Figure 6. Table 7 lists the symbols and parameters. All measurements are made when the chip is at thermal equilibrium. All ac specifications, including capacitance values, pertain to packaged parts and apply when all outputs are loaded and switched simultaneously. The ac test circuit configuration is shown in Figure 7 and TTL and ECL load circuits are shown in Figure 8.



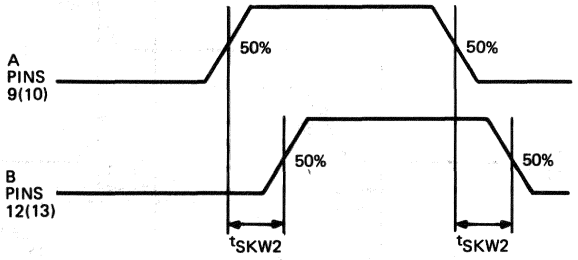
NOTES:

1. LOAD OUTPUTS PER FIGURE 8
2. INPUT RISE AND FALL TIMES EQUAL 1.0 ns

Figure 5 ▪ VAXBI 78701 Propagation Delay Timing



t_{SKW1} = GREATER OF [A - B] OR [B - A] ABSOLUTE VALUE



t_{SKW2} = GREATER OF [A - B] OR [B - A] ABSOLUTE VALUE

Figure 6 ▪ VAXBI 78701 Output Signal Timing

Table 5 ▪ VAXBI 78701 ac Timing Parameters

Symbol	Parameter	Requirements (ns) Min.	Max.
t_{PD2}	Propagation delay	—	10
t_R	TTL output rise time (V_{OL} max. to V_{OH} min.)	—	8.0
t_F	TTL output fall time (V_{OH} min. to V_{OL} max.)	—	8.0
t_{SKW1}	Single differential output skew	—	1.0
t_{SKW2}	Differential output to output skew	—	1.0
t_{PD1}	Propagation delay	—	12
t_R	ECL output rise time (30% to 70%)	0.5	2.5
t_F	ECL ouput fall time (30% to 70%)	0.5	2.5

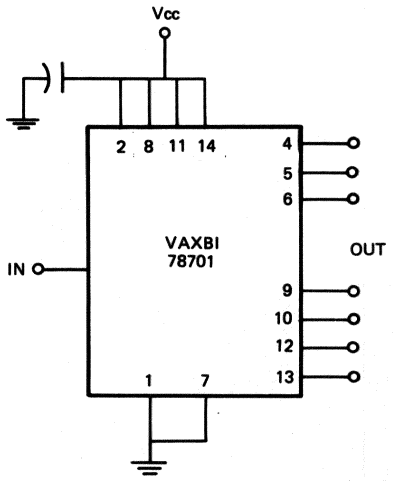


Figure 7 ▪ VAXBI 78701 ac Test Circuit

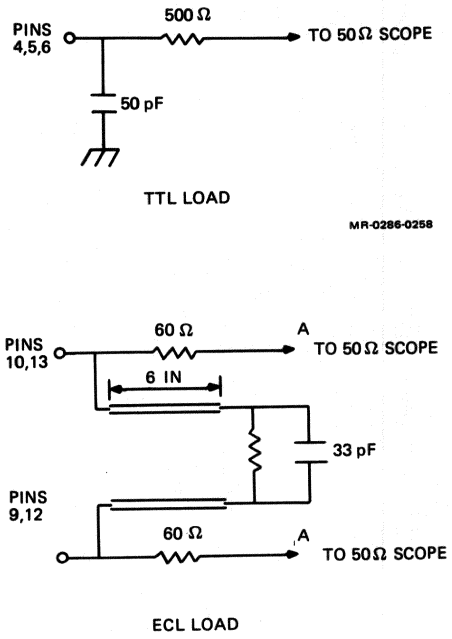


Figure 8 ▪ VAXBI 78701 ac Load Circuits

• Features

- Custom bipolar 16-pin integrated circuit used by all VAXBI nodes
- Receives the differential ECL BI Time +/− and BI PHASE +/− signals
- Single 5-V power supply
- Approved for use in the VAXBI corner application

• Description

The VAXBI 78702 clock receiver is a custom-built bipolar 16-pin integrated circuit (IC) that must be used by all VAXBI nodes to receive the differential ECL BI TIME +/− and BI PHASE +/− signals. The device requires only a 5-V operating voltage.

The clock receiver provides both true and complement TTL levels of both received differential VAXBI signals to the adapter. It is approved for use only in the VAXBI corner application. Figure 1 is a functional diagram of the clock driver.

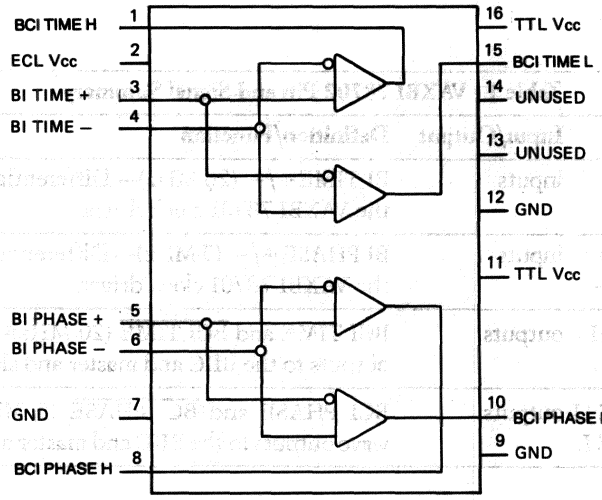


Figure 1 • VAXBI 78702 Clock Receiver Logic

• Pin and Signal Descriptions

The input and output signals and the power and ground connections of the clock receiver are shown in Figure 2. Table 1 provides a summary of the input and output signals.

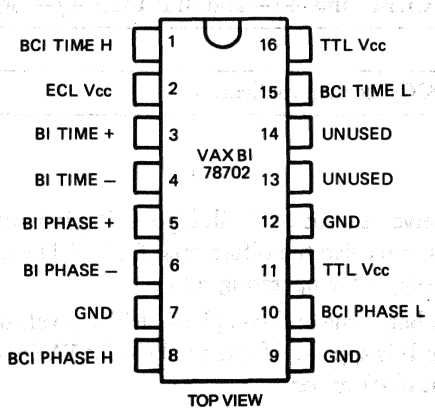


Figure 2 • VAXBI 78702 Pin Assignments

Table 1 • VAXBI 78702 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
3 4	BI TIME + BI TIME –	inputs	BI TIME +/– (20 MHz)—Differential ECL inputs from the VAXBI 78701 clock driver.
5 6	BI PHASE + BI PHASE –	inputs	BI PHASE +/– (5 MHz)—Differential ECL inputs from the VAXBI 78701 clock driver.
1 15	BCI TIME H BCI TIME L	outputs	BCI TIME and $\overline{\text{BCI TIME}}$ (20 MHz)—TTL square-wave outputs to the BIIC and master and slave interfaces.
8 10	BCI PHASE H BCI PHASE L	outputs	BCI PHASE and $\overline{\text{BCI PHASE}}$ (5 MHz)—TTL square-wave outputs to the BIIC and master and slave interfaces.
13,14	Not used		
2	ECL V_{cc}	input	ECL voltage—Power supply voltage to ECL logic.
16,11	TTL V_{cc}	input	TTL voltage—Power supply voltage to TTL logic.
7,9,12	GND*	input	Ground—Common ground reference.

*GND pins are connected internally.

• Application

A typical clock driver and clock receiver configuration is shown in Figure 3. The clock driver receives the 40 MHz TTL input from an external clock oscillator and supplies the differential ECL BI TIME +/± and BI PHASE +/± inputs to the clock receiver and to the VAXBI bus. Refer to the VAXBI 78701 clock driver for a description of the clock driver functions. The differential ECL signals are received by an internal gate structure in the clock receiver. The ouputs are converted to TTL levels and provide the BCI PHASE H and BCI PHASE L and BCI TIME H and BCI TIME L signal timing information to the VAXBI 78732 Bus Interconnect Interface (BIIC) and master and slave interfaces.

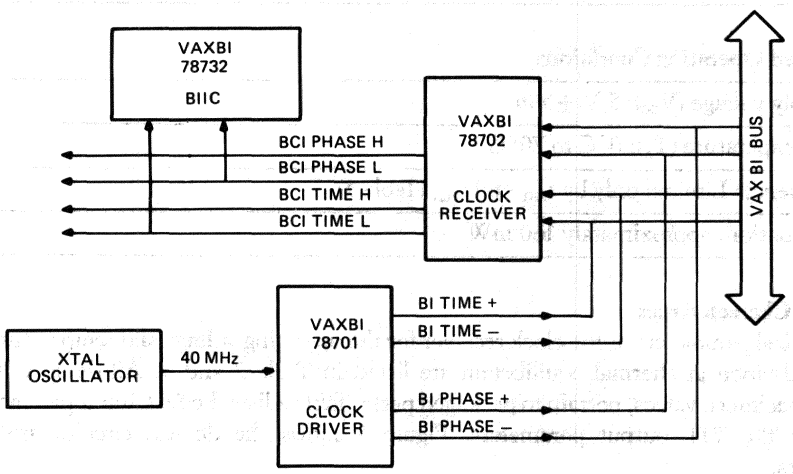


Figure 3 • VAXBI 78702 Clock Receiver with VAXBI 78701 Clock Driver Configuration

• Specifications

The mechanical, electrical, and environmental characteristics and specifications for the clock driver are described in the following paragraphs. The test conditions for the electrical values are as follows and are performed when the chip is at thermal equilibrium unless specified otherwise.

- Power supply voltage (V_{CC}): 5.0 V ±5%
- Temperature range (T_A): 0°C to 70°C

Mechanical Configuration

The physical dimensions of the 16-pin dual-inline (DIP) package are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Power supply voltage (V_{CC}): 7.0 V
- Input voltage applied: –1.5 V to 7.0 V
- TTL output applied current: 40 mA (low state)
- Operating junction temperature (T_J): 125°C
- Storage temperature (T_S): –55°C to 150°C

Recommended Operating Conditions

- Power supply voltage (V_{CC}): 5 V ±5%
- Ambient temperature (T_A): 0°C to 70°C
- Input frequency: Limited only by t_{PHL} and t_{PLH} (Table 5)
- Power dissipation: approximataly 160 mW

dc Electrical Characteristics

The dc electrical parameters of the clock receiver for the operating voltage and temperature ranges specified and when at thermal equilibrium are listed in Table 2 and 3. All dc specifications, including capacitance values, pertain to packaged parts. Table 2 lists the ECL input parameters and Table 3 lists the TTL output parameters. Figure 4 shows the dc test circuits use for the measurements.

Table 2 • VAXBI 78702 dc ECL Input Parameters

Symbol	Parameter	Test Conditions	Requirements Min.	Requirements Max.	Unit
I_{IHL}[1]	High-level input current	V_{CC}=5.25 V V_{in}+4.5 V Complement input=4.3 V	—	150	µA
I_{IH2}	High-level input current	V_{CC}=GND V_{in}=4.5 V V_{diff}=1, 0 V	—	3.0	mA
I_{IL}[1]	Low-level input current	V_{CC}=5.25 V V_{in}=2.8 V Complement input=3.0 V	—	150	µA
V_{diff}[2,3]	Differential threshold voltage	Pins 3 to 4, 5 to 6	200	—	mV

Symbol	Parameter	Test Conditions	Requirements Min.	Max.	Unit
V_{CM}[2,4]	Common mode voltage		2.8	4.5	V
C_{MRR}[5]	Common mode rejection ratio		40	—	dB
I_{CC}[6]	Power supply current	$V_{CC}=5.25$ V	—	20	mA
C_{in}[7]	Single-ended input capacitance	to GND	—	6	pF

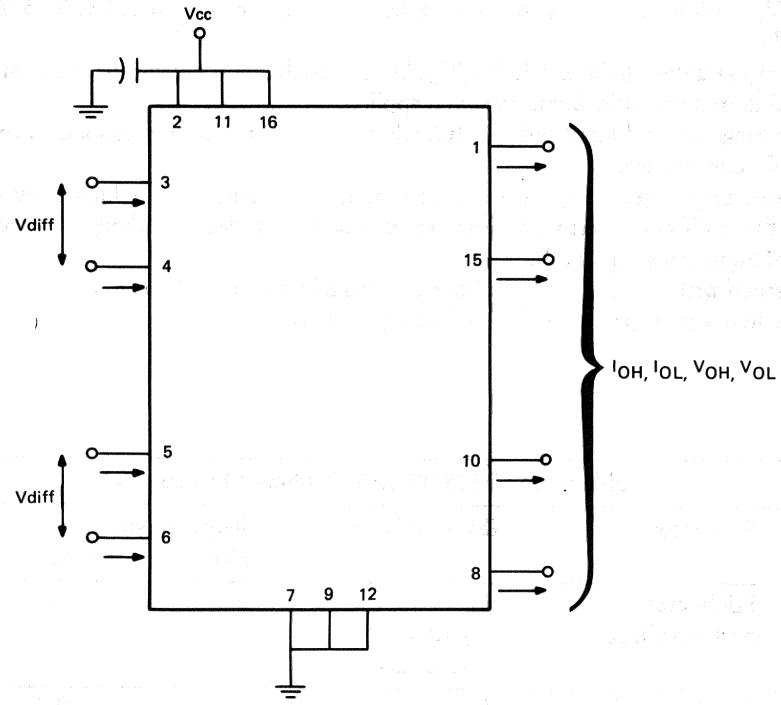
[1]Other gate at valid logic state.
[2]Tested at $V_{in}=2.8$ V, 3.55 V, and 4.3 V with the complement input at 3.0 V, 3.75 V, and 4.5 V, respectively.
[3]Outputs are guaranteed to be within V_{OL}/V_{OH} limits if both inputs are within common mode range and if minimum differential input voltage is applied.
[4]C_{MR} is the range of total input voltage through which the output will respond to the minimum differential input voltage.
[5]C_{MMR} is the ability to reject the effect of voltage applied to both inputs simultaneously. A C_{MRR} of 40 dB means that a 2 V common mode voltage is processed by the device as though it were an additive differential input signal of 20 mV.
[6]I_{CC} is measured with the device in a quiescent state and with no load applied.
[7]Difference in output capacitance ($C_{in}+$ minus $C_{in}-$)=0.7 pF

Table 3 • VAXBI 78702 dc TTL Output Parameters

Symbol	Parameter	Test Conditions	Requirements Min.	Max.	Unit
V_{OH}[1]	High-level output voltage	$V_{CC}=4.75$ V $V_{diff}=200$ mV $I_O=I_{OH}$ max.	2.6	—	V
V_{OL}[1]	Low-level output voltage	$V_{CC}=4.75$ V $V_{diff}=200$ mV $I_O=20$ mA	—	0.5	V
I_{OH}[1]	High-level output current	$V_{CC}=4.75$ V $V_{diff}=200$ mV $V_O=V_{OH}$ min.	—	−1000	A

Symbol	Parameter	Test Conditions	Requirements Min.	Max.	Unit
I_{OL}[1]	Low-level output current	$V_{CC}=4.75$ V $V_{diff}=200$ mV $V_{out}=0.5$ V	20	—	mA
I_{OS}[2]	Output short-circuit current		-25	-150	mA

[1]Testing either V_{OH} and V_{OL} or I_{OH} and I_{OL} is sufficient to satisfy these requirements.
[2]TTL outputs shorted to GND or open do not affect the impedance of differential inputs. I_{OS} is defined as current into output pin when the input conditions cause a logic one at the outputs before application of the short.



NOTES:

1. Vcc = GND FOR I_{IH2} TEST
2. PINS 13 AND 14 ARE UNUSED

Figure 4 ▪ VAXBI 78702 dc Test Circuits

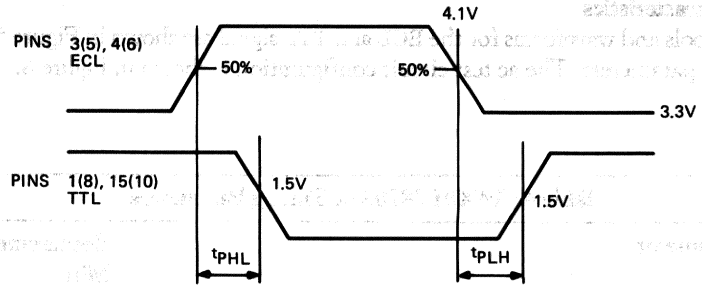
ac Electrical Characteristics

The timing symbols and waveforms for the ECL and TTL signal are shown in Figure 5. Table 4 lists the symbols and parameters. The ac test circuit configuration is shown in Figure 6.

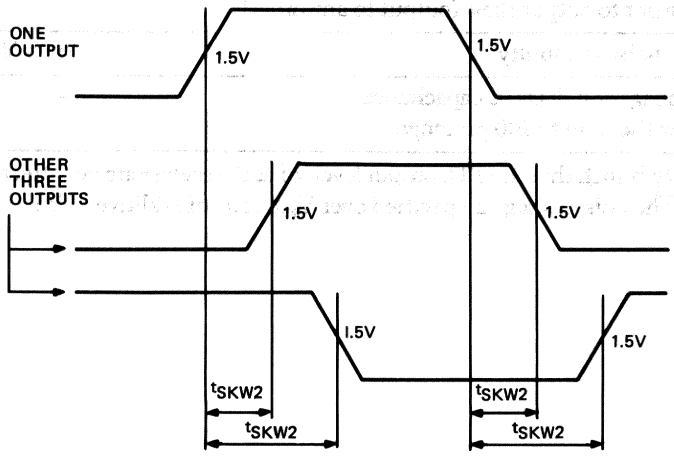
Table 4 • VAXBI 78702 ac Timing Parameters

Symbol	Parameter	Requirements (ns) Min.	Max.
t_{PLH}	Delay ECL to TTL output leading edge	1.5	6.0
t_{PHL}	Delay ECL to TTL trailing edge	1.5	6.0
t_R	Output fall time (V_{OL} max. to V_{OH} max.)	—	5.0
t_F	Output fall time (V_{OH} min. to V_{OL} max.)	—	5.0
t_{SKW1}	Single output skew on individual outputs	—	4.0
t_{SKW2}	Output to output skew (output to any other)	—	4.0
V_{NSE}*	V_{CC} noise immunity	0.25 V	—
P_d/pF	Propagation delay vs capacitance over the 0 pF to 100 pF range	—	40

*Noise immunity is such that the TTL output levels of each receiver are not affected due to additive noise on V_{CC}. The noise voltage is specified over V_{NSE} volts of additive noise.



1. ALSO MAKE MEASUREMENTS REVERSING A AND B.
2. t_{SKW1} = GREATER OF $t_{PHL} - t_{PLH}$ MEASURED FOR PINS 1, 8, 10, AND 15.
3. THIS TEST GUARANTEES WAVEFORM SYMMETRY AT A GIVEN Vcc AND AMBIENT TEMPERATURE.



1. t_{SKW2} IS THE GREATER OF SIX ABSOLUTE VALUE MEASUREMENTS.
2. THIS TEST GUARANTEES MAXIMUM SKEWS ON A GIVEN BOARD (ONE RECEIVER) AND IS FOR A GIVEN Vcc AND AMBIENT TEMPERATURE.

Figure 5 • VAXBI 78702 Timing Waveforms

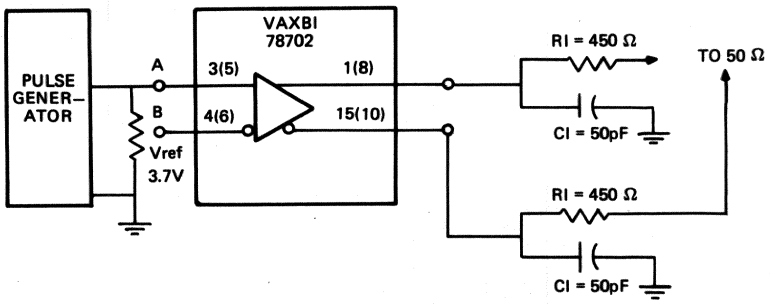


Figure 6 ▪ VAXBI 78702 ac Test Circuit

Features

- Used with the DC004, and DC005 circuits to implement a program control device interface.
- Used with the DC004, DC005, DC006, and DC010 circuits to implement a direct memory access interface.
- Provides two device-interrupt channels.
- Performs a pass-the-pulse interrupt transaction.
- Includes Q-bus drivers and receivers.

Description

The DC003 dual-interrupt circuit, contained in an 18-pin, dual-inline package (DIP), is used in the development of device interfaces for the Q-bus. The simplified logic diagram of the DC003 is shown in Figure 1.

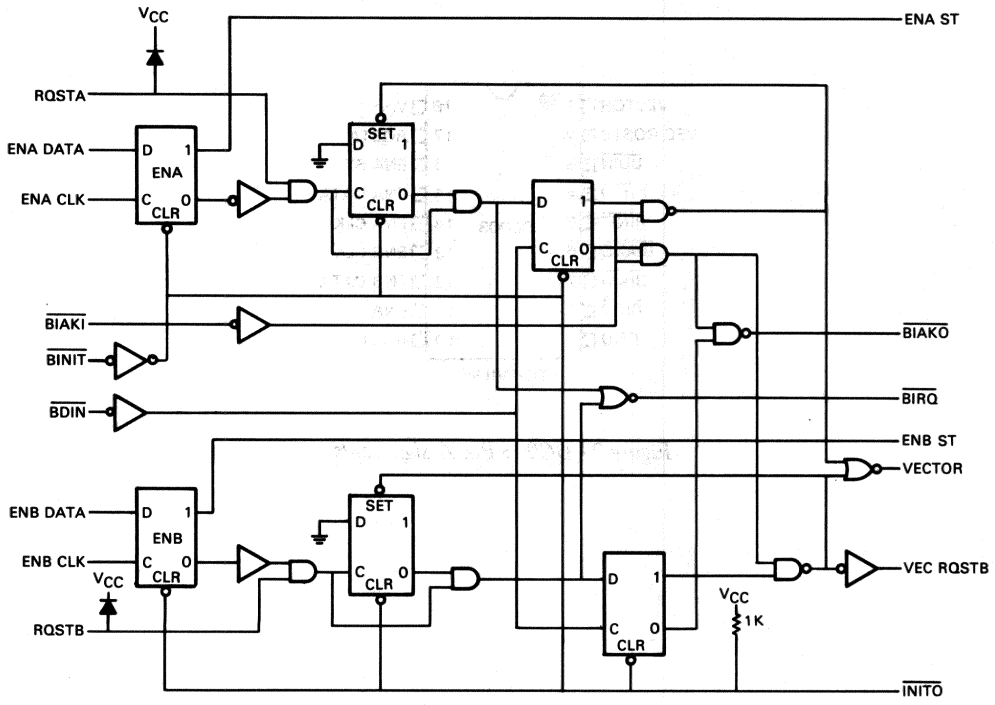


Figure 1 • DC003 Simplified Logic Diagram

It provides the circuits to perform an interrupt transaction in a computer system that uses a daisychain type of arbitration. It includes two interrupt channels; channel A and B, each of which can generate an interrupt request from a device that requires service. The A interrupt logic is assigned a higher priority than the B interrupt logic. Input signals from the bus are received by high-impedance receivers on the DC003 and signals from the DC003 to the bus are supplied by high-current, open-collector driver outputs. The signals levels between the bus and the DC003 are compatible. When a device connected to the interface is not requesting service, the signal from the polling device or processor is passed through the DC003 logic to the next device on the bus. The DC003 circuits includes enable logic and provides interrupt status information to the requesting device.

• Pin and Signal Descriptions

This section provides a brief description of the input and output signals and power and ground connections of the DC003. The pin assignments are identified in Figure 2 and summarized in Table 1.

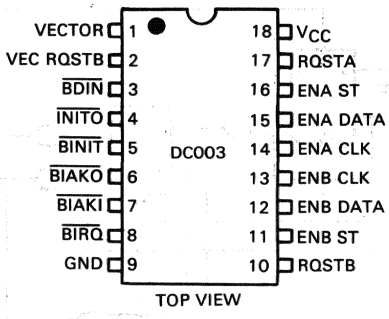


Figure 2 • DC003 Pin Assignments

Table 1 • DC003 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
1	VECTOR	output[1]	Interrupt vector gating—Asserted by the DC003 logic to gate the appropriate vector address onto the bus. This signal forms the bus signal $\overline{\text{BRPLY}}$.
2	VEC RQSTB	output[1]	Vector request B—Asserted by the DC003 logic to indicate that the RQSTB service vector is required and not asserted to indicate that the RQSTA service vector is required. This signal is normally bit 2 of the vector address.
3	$\overline{\text{BDIN}}$	input[2]	Bus data in—Asserted by the master to indicate that an interrupt operation is occurring and always precedes a BIAK signal.
4	$\overline{\text{INITO}}$	output[1]	Initialize out—A buffered $\overline{\text{BINIT}}$ signal from the bus used to initialize the DC003 logic and user device logic
5	$\overline{\text{BINIT}}$	input[2]	Bus initialize—Asserted by the bus to initialize the DC003 logic and device interface logic to a known state.
6	$\overline{\text{BIAKO}}$		Bus interrupt acknowledge output—Asserted by the DC003 logic to pass control to the next device on the bus if the device connected to the DC003 does not require service. Once asserted, it remains passed until the next $\overline{\text{BIAKI}}$ signal is generated.
7	$\overline{\text{BIAKI}}$	input[3]	Bus interrupt acknowledge input—Asserted by the processor in response to the $\overline{\text{BIAKO}}$ signal. The first requesting device prevents the $\overline{\text{BIAKO}}$ signal from being transferred to other devices on the bus and nonrequesting devices will pass the $\overline{\text{BIAKO}}$ signal to the next device. The leading edge of the $\overline{\text{BIAKI}}$ signal clears the $\overline{\text{BIRQ}}$ signal from the DC003.
8	$\overline{\text{BIRQ}}$	output[3]	Bus interrupt request—Asserted by the DC003 logic to indicate that the associated device is requesting an interrupt. This signal is asserted when the RQSTA signal and the ENA DATA signal from the device is asserted. It is cleared when the request signal is removed or on the leading edge of the $\overline{\text{BIAKI}}$ signal after the acceptance of the $\overline{\text{BDIN}}$ signal.
9	GND	input	Ground—Common ground reference
10	RQSTB	input[2]	Device interrupt request B—Asserted by a device requesting an interrupt when the device has asserted the ENB DATA signal. It results in an interrupt request on the $\overline{\text{BIRQ}}$ line that normally remains asserted until the interrupt request is granted.
11	ENB ST	output[1]	Interrupt enable B status—Indicates the status of the interrupt enable B flip-flop on the DC003.

Pin	Signal	Input/Output	Definition/Function
12	ENB DATA	input[1]	Interrupt enable B data—Asserted by a device to enable the transfer of data.
13	ENB CLK	input[1]	A clock pulse from a device that enables the interrupt enable B flip-flop to assume the state of the ENB DATA signal.
14	ENA CLK	input[1]	A clock pulse from a device that enables the interrupt enable A flip-flop to assume the state of the ENA DATA signal.
15	ENA DATA	input[1]	Interrupt enable A data—Asserted by a device to enable the transfer of data.
16	ENA ST	output[1]	Interrupt enable A status—Indicates the status of the interrupt enable A flip-flop on the DC003.
17	RQSTA	input[2]	Device interrupt request A—Asserted by a device requesting an interrupt when the device has asserted the ENA DATA signal. It results in an interrupt request on the $\overline{BIRQ}$ line that normally remains asserted until the interrupt request is granted.
18	V_{CC}	input	Voltage—Power supply dc voltage

[1]TTL level
[2]high-impedance
[3]open-collector

• Application Information

Refer to the *Chipkit Users Manual LSI-11 Bus Interface Chips* (document no. EJ-01387-92) for general application information. The Q-bus is an LSI-11 bus.

• Specifications

The mechanical, electrical, and environmental characteristics and specifications for the DC003 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Operating temperature (T_A): 0°C to 70°C
- Supply voltage (V_{CC}): 5.0 V ±5%

Mechanical Configuration

The physical dimensions of the DC003 18-pin DIP package are contained in Appendix E. The materials and construction of the molded DIP are defined in Digital Specification A-PS-1900002-GS.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Supply voltage (V_{CC}): 7.0 V
- Input voltage (V_I): 5.5 V
- Operating temperature (T_A): 0°C to 70°C
- Storage temperature (T_S): –65°C to 150°C

Recommended Operating Conditions

- Supply voltage (V_{CC}): 4.75 V (minimum), 5.0 V (normal), 5.25 V (maximum)
- Supply current (I_{CC}): 140 mA (maximum)
- Free-air temperature: 0°C to 70°C
- Relative humidity: 10% to 95% (noncondensing)

dc Electrical Characteristics

The dc electrical characteristics of the DC003 for the operating voltage and temperature ranges specified are listed in Tables 2 through 4. Table 2 lists the dc specifications of the TTL input and output circuits that do not connect to the bus. Table 3 lists the dc specifications for the high-impedance receivers that connect to the LSI-11 bus. Table 4 lists the dc specifications for the open-collector driver outputs that connect to the bus. Refer to Appendix C for the test circuit configurations referenced in the tables.

Table 2 • DC003 TTL Input and Output Parameters (nonbus)

Parameter	Symbol	Test Condition	Requirements Min.	Requirements Max.	Units	Test Circuit
High-level input voltage	V_{IH}		2.0	—	V	C1, C2
Low-level input voltage	V_{IL}		—	0.8	V	C1, C2
Input clamp voltage	V_I	$V_{CC}=4.75$ V $I_I=-18$ mA	—	–1.2	V	C3
High-level output voltage	V_{OH}	$V_{CC}=4.7$ V $I_O=-1.0$ mA	2.7	—	V	C1
Low-level output voltage	V_{OL}	$V_{CC}=4.75$ V $I_O=20$ mA	—	0.5	V	C2
Input current at maximum input voltage	I_I	$V_{CC}=5.25$ V $V_I=5.5$ V	—	1.0	mA	C4
High-level input current	I_{IH}	$V_{CC}=5.25$ V $V_I=2.7$ V[1]	—	50	μA	C4

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units	Test Circuit
Low-level input current	I_{IL}	$V_{CC}=5.25$ V $V_I=0.5$ V[2]	—	-0.55	mA	C5
Short-circuit output current	I_{OS}	$V_{CC}=5.25$ V[3]	-40	-100	mA	C6
Supply current	I_{CC}	$V_{CC}=5.25$ V	—	140	mA	C7

[1]$I_{IH}=100$ µA at pins 12 and 15.
[2]I_{IL}–2.0 mA at pins 12 and 15.
[3]Not more than one output shall be short circuited at a time and the duration of the short must not exceed 1 second.
[4]Does not apply to pin 4.

Table 3 • DC003 High-impedance Bus Receiver Parameters

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units	Test Circuit
High-level input voltage	V_{IH}	$V_{CC}=4.75$ V	1.53	—	V	C1,C2
	V_{IH}	$V_{CC}=5.25$ V	1.70	—	V	C1,C2
Low-level input voltage	V_{IL}	$V_{CC}=4.75$ V	—	1.30	V	C1,C2
	V_{IL}	$V_{CC}=5.25$ V	—	1.47	V	C1,C2
Input clamp voltage	V_I	$V_{CC}=4.75$ V $I_I=-18$ mA	—	-1.2	V	C3
	V_I	$V_{CC}=4.75$ V $I_I=18$ mA[1]	—	6.25	V	C3
High-level input current	I_{IH}	$V_{CC}=0$ V $V_I=3.8$ V[2]	—	40	µA	C4
	I_{IH}	$V_{CC}=5.25$ V	—	40	µA	C4
Low-level input current	I_{IL}	$V_{CC}=0$ V $V_I=0.5$ V[2]	—	-10	µA	C5
	I_{IL}	$V_{CC}=5.25$ V	—	-10	µA	C5

[1]Pins 10 and 17 only.
[2]Exclude pins 10 and 17.

Table 4 • DC003 Open-collector Bus Driver Parameters

Parameter	Symbol	Test Condition	Requirements Min.	Requirements Max.	Units	Test Circuit
Output reverse current	I_{OR}	$V_{CC}=4.75$ V $V_{OH}=3.5$ V	—	25	μA	C1
Low-level output voltage	V_{OL}	$V_{CC}=4.75$ V $I_{sink}=70$ mA	—	0.8	V	C2
		$V_{CC}=4.7$ V $I_{sink}=16$ mA	—	0.5	V	C2

ac Electrical Characteristics

The input/output signal timing for the interrupt logic A is shown in Figure 3. The input/output signal timing for interrupt logic A and interrupt logic B is shown in Figure 4. Refer to Appendix D for the standard input voltage waveforms and for the signal propagation delay measurements. The load circuit configurations used in measuring the TTL outputs and open-collector outputs are shown in Figure 5.

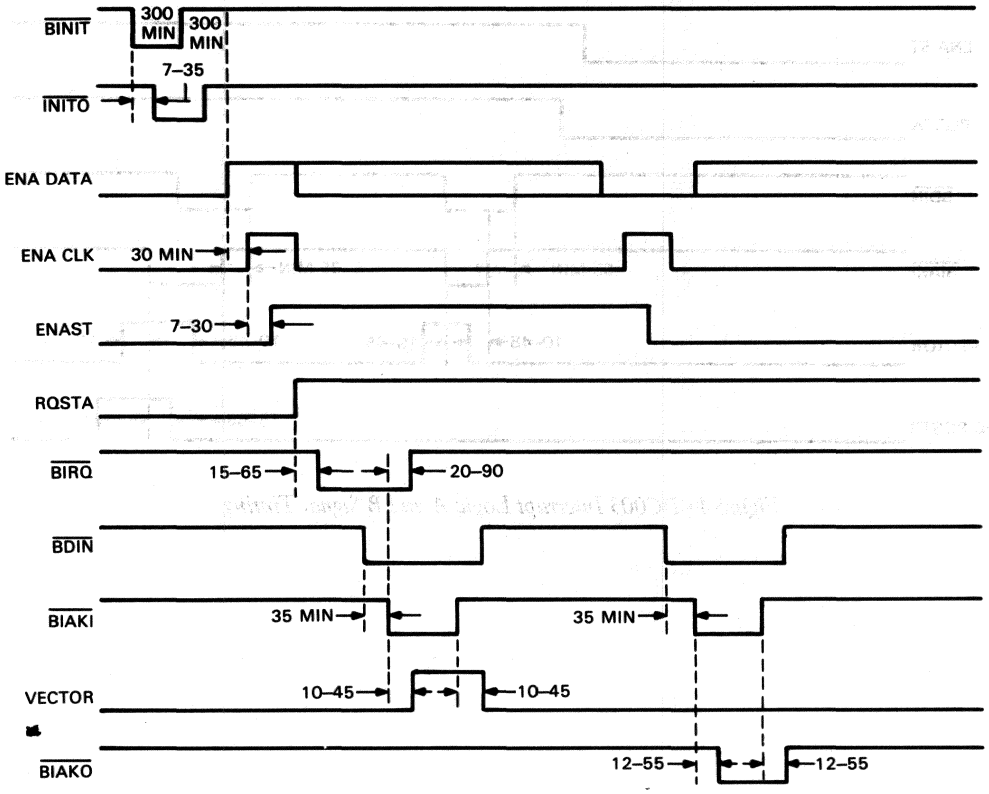


Figure 3 • DC003 Interrupt Logic A Signal Timing

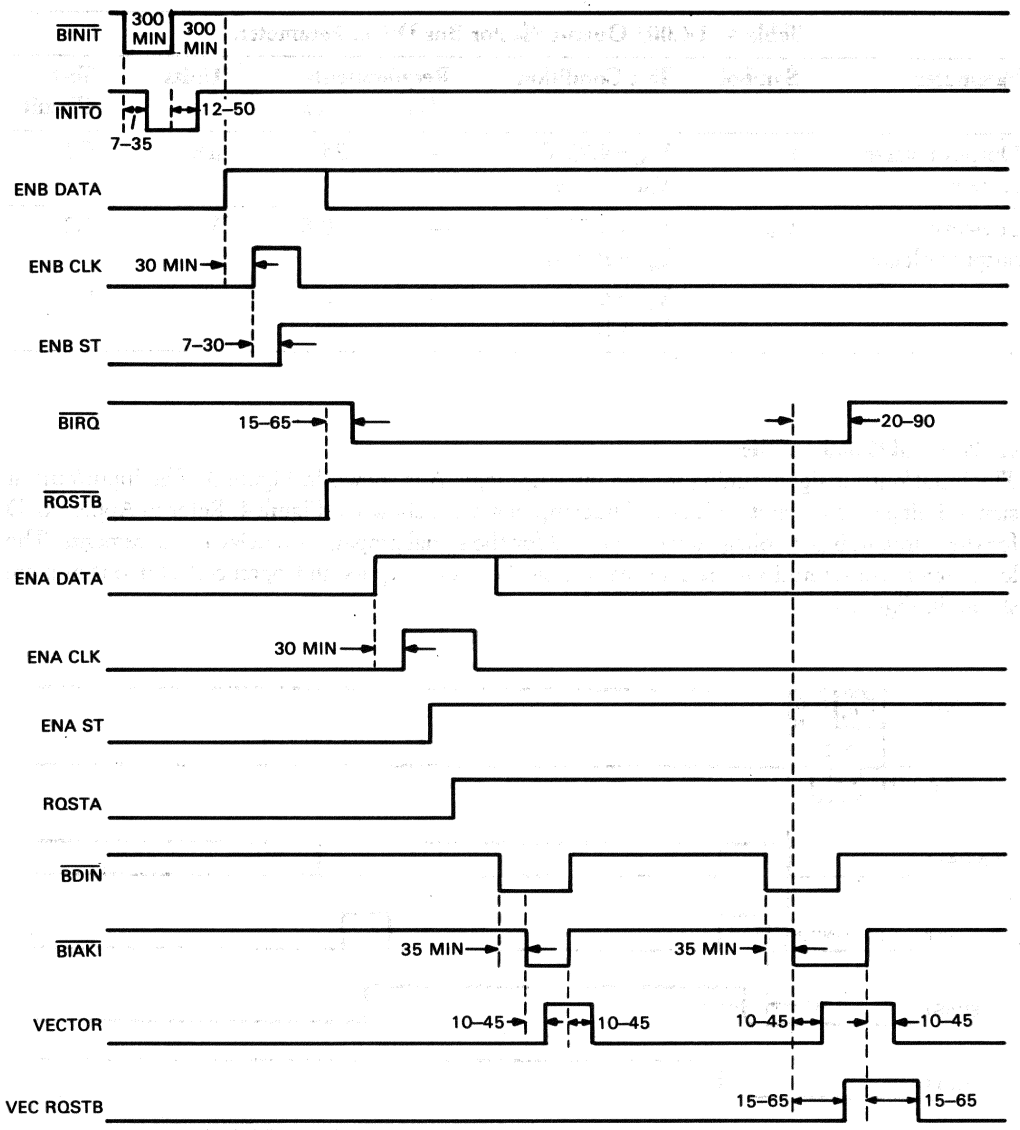


Figure 4 ▪ DC003 Interrupt Logic A and B Signal Timing

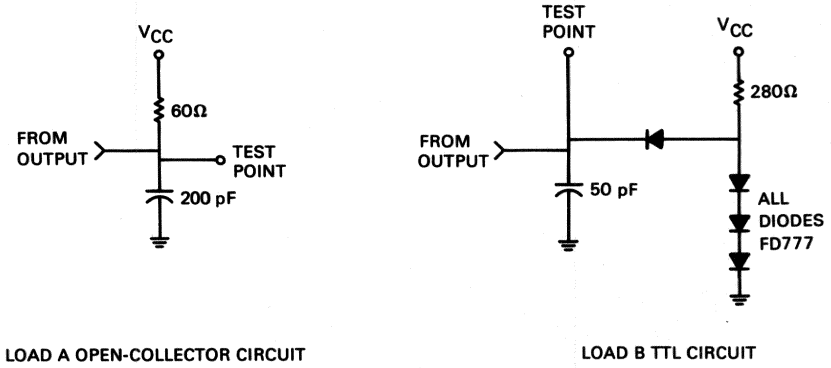


Figure 5 • DC003 Output Load Circuits

Features

- Used with the DC003, DC005, and DC006 circuits to implement a program control device interface.
- Used with the DC003, DC005, DC006, and DC010 circuits to implement a direct memory access interface.
- Controls data transfers to and from as many as four 8-byte word registers.
- Provides variable-delay logic for device response to transfer requests.
- Includes Q-bus drivers and receivers.

Description

The DC004 register selector, contained in a 20-pin dual-inline package (DIP), provides the signals to control the transfer of data to and from as many as four word registers (8-bytes). The Q-bus signals directly connect to high-impedance receivers and to high-current drivers on the DC004. A resistor and capacitor circuit (RC) is included to delay the response of the peripheral interface to the data transfers. An external RC network can be added to vary the delay time to conform to the device requirements. Figure 1 is a simplified logic diagram of the DC004 which includes bus latching circuits, enable logic, and a decoder used to select the registers.

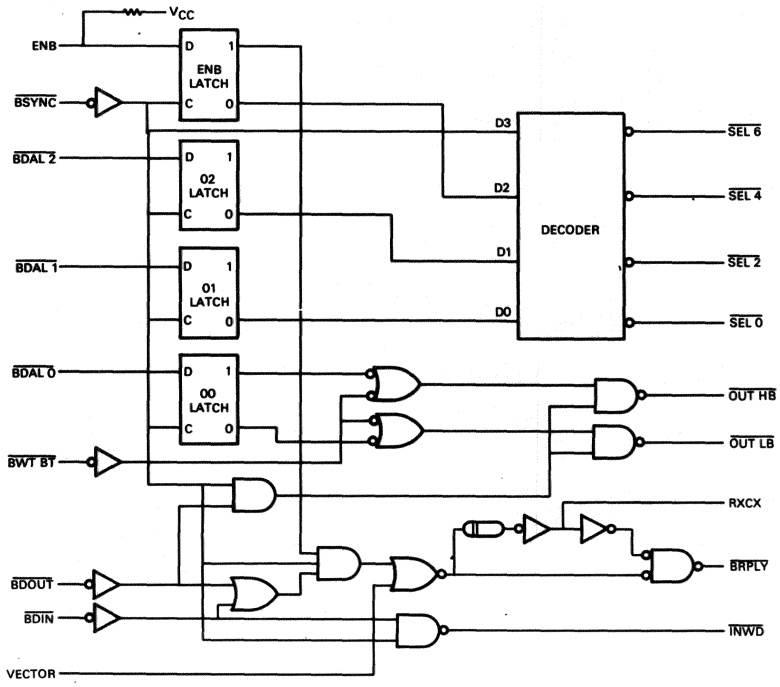


Figure 1 • DC004 Simplified Logic Diagram

• Pin and Signal Descriptions

This section provides a brief description of the input and output signals and power and ground connections of the DC004 20-pin DIP. The pin assignments are identified in Figure 2 and the summarized in Table 1. The signal names shown in the diagram are for the condition where the DC004 is connected to the internal three-state bus of the DC005.

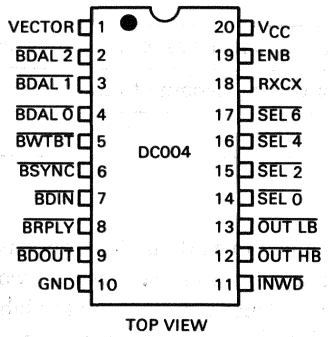


Figure 2 • DC004 Pin Assignments

Table 1 • DC004 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
1	VECTOR	input[1]	Interrupt vector gating—Asserted by the interrupt logic (DC003) to gate the appropriate vector address onto the bus. This signal generates the $\overline{\text{BRPLY}}$ output signal to the bus after a time delay selected by the RC network connected to pin 18 (RXCX).
2-4	$\overline{\text{BDAL<2:0>}}$	input[2]	Bus data/address lines—The information on these lines is latched by the $\overline{\text{BSYNC}}$ signal from the bus and used to select one or more of the 8-bit registers connected to the $\overline{\text{SEL 0}}$, $\overline{\text{SEL 2}}$, $\overline{\text{SEL 4}}$ and $\overline{\text{SEL 6}}$ lines.
5	$\overline{\text{BWTBT}}$	input[2]	Write/Byte—Selects a byte or word operation when the $\overline{\text{BDOUT}}$ signal is asserted: $\overline{\text{BWTBL}}$ asserted = byte, $\overline{\text{BWTBL}}$ negated = word. The latched output of this signal is gated with $\overline{\text{BDOUT}}$ to select a low-byte or high-byte output.
6	$\overline{\text{BSYNC}}$	input[2]	Bus synchronize—Asserted by the bus master to indicate that an address is on the bus. When unasserted, it disables all the outputs except the vector generated output of $\overline{\text{BRPLY}}$.
7	$\overline{\text{BDIN}}$	input[2]	Bus data in—Asserted by the bus master to effect a data input transaction when the $\overline{\text{BSYNC}}$ signal is asserted. It generates the $\overline{\text{BRPLY}}$ output through the delay circuit and the $\overline{\text{INWD}}$ output.
8	$\overline{\text{BRPLY}}$	output[3]	Bus reply—Asserted by the slave device to indicate that data is available on the $\overline{\text{BDAL}}$ bus or that the device has accepted output data from the bus.
9	$\overline{\text{BDOUT}}$	input[2]	Bus data out—Asserted by the device to indicate that device data is available on the $\overline{\text{BDAL}}$ lines.
10	GND	input	Ground—Common ground connection
11	$\overline{\text{INWD}}$	output[1]	In word—Asserted to gate the data from a selected register to the bus.
12 13	$\overline{\text{OUT LB}}$ $\overline{\text{OUT HB}}$	outputs[1]	Out low-byte, Out high-byte—Used to load the low-byte, high-byte, or both bytes of the write data from the selected register onto the bus.
14 15 16 17	$\overline{\text{SEL 0}}$ $\overline{\text{SEL 2}}$ $\overline{\text{SEL 4}}$ $\overline{\text{SEL 6}}$	outputs[1]	Select decoder 0, 2, 4, and 6 output lines—Selects a word register for a data transaction.
18	RXCX	output[3]	External RC node—The value of the resistor and capacitor network connected to this pin determines the delay of the $\overline{\text{BRPLY}}$ signal.

Pin	Signal	Input/Output	Definition/Function
19	ENB	input[1,4]	Enable—Asserted to enable the $\overline{SEL\ 0}$, $\overline{SEL\ 2}$, $\overline{SEL\ 4}$, and $\overline{SEL\ 6}$ outputs of the decoder and the address term of $\overline{BRPLY}$.
20	V_{CC}	input	Voltage—Power supply dc voltage

[1]TTL level
[2]high-impedence
[3]open-collector
[4]connected to V_{CC} through an 850Ω resistor

Application Information

Refer to the *Chipkit Users Manual LSI-11 Bus Interface Chips* (document no. EJ-01387-92) for general application information. The Q-bus is an LSI-11 bus.

Specifications

The mechanical, electrical, and environmental characteristics and specifications for the DC003 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Operating temperature (T_A): 0°C to 70°C
- Supply voltage (V_{CC}): 5.0 V ±5%

Mechanical Configuration

The physical dimensions of the DC004 20-pin DIP are contained in Appendix E. The materials and construction of the molded DIP are defined in Digital Specification A-PS-2100002-GS.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Supply voltage (V_{CC}): 7.0 V
- Input voltage (V_I): 5.5 V
- Operating temperature (T_A): 0°C to 70°C
- Storage temperature (T_S): –65°C to 150°C

Recommended Operating Conditions

- Supply voltage (V_{CC}): 4.75 V (minimum), 5.0 V (normal), 5.25 V (maximum)
- Supply current (I_{CC}): 120 mA (maximum)
- Free-air temperature: 0°C to 70°C
- Relative humidity: 10% to 95% (noncondensing)

dc Electrical Characteristics

The dc electrical characteristics of the DC004 for the operating voltage and temperature ranges specified are listed in Tables 2 through 4. Table 2 lists the dc specifications for the TTL input and outputs that do not connect to the Q-bus bus. Table 3 lists the dc specifications for the high-impedance receiver inputs from the Q-bus bus. Table 4 lists the dc specifications for the open-collector driver outputs that connect to the Q-bus. Refer to Appendix C for test circuit configurations listed in the tables.

- Note: Pin 18 (RXCX) of the DC004 must be connected to V_{CC} through a 1-kΩ ±5% resistor for performing all the dc tests shown in the test circuit diagrams in Appendix C.

Table 2 • DC004 TTL Input and Output Parameters

Parameter	Symbol	Test Condition	Requirements Min	Max	Units	Test Circuit
High-level input voltage	V_{IH}		2.0	—	V	C1,C2
Low-level input voltage	V_{IL}		—	0.8	V	C1,C2
Input clamp voltage	V_I	V_{CC}=4.75 V I_I=−18 mA	—	−1.2	V	C3
High-level output voltage	V_{OH}	V_{CC}=4.75 I_O=−1.0 mA	2.7	—	V	C1
Low-level output voltage	V_{OL}	V_{CC}=4.75 V I_O=20 mA	—	0.5	V	C2
Input current at maximum input voltage	I_I	V_{CC}=5.25 V V_I=5.5 V[1]	—	1.0	mA	C4
High-level input current	I_{IH}	V_{CC}=5.25 V V_I=2.7 V[1]	—	50	μA	C4

Parameter	Symbol	Test Condition	Requirements Min	Max	Units	Test Circuit
Low-level input current	I_{IL}	$V_{CC}=5.25$ V $V_I=0.5$ V	—	−0.70	mA	C5
Short-circuit output current	I_{OS}	$V_{CC}=5.25$ V[2]	−40	−100	mA	C6
Supply current	I_{CC}	$V_{CC}=5.25$ V	—	120	mA	C7

[1]Limits for pin 19 are: $I_I=1.4$ mA, $I_H=-2.25$ mA (minimum) and −3.85 mA (maximum), $I_{IL}=-4.5$ mA (minimum) and −8.0 mA (maximum).
[2]Not more than one output shall be shorted at a time and the duration of the short shall not exceed 1 second.

Table 3 • DC004 High-impedance Bus Receiver Parameters

Parameter	Symbol	Test Condition	Requirements Min	Max	Units	Test Circuit
High-level input voltage	V_{IH}	$V_{CC}=4.75$ V $V_{CC}=5.25$ V	1.53 1.70	— —	V V	C1,C2
Low-level input voltage	V_{IL}	$V_{CC}=4.75$ V $V_{CC}=5.25$ V	— —	1.30 1.47	V V	C1,C2
Input clamp voltage	V_I	$V_{CC}=4.75$ V $I_I=-18$ mA	—	−1.2	V	C3
High-level input current	I_{IH}	$V_I=3.8$ V $V_{CC}=0$ V $V_{CC}=5.25$ V	— —	40 40	µA µA	C4
Low-level input current	I_{IL}	$V_I=0$ V $V_{CC}=0$ V $V_{CC}=5.25$ V	— —	−10 −10	µA µA	C5 C5

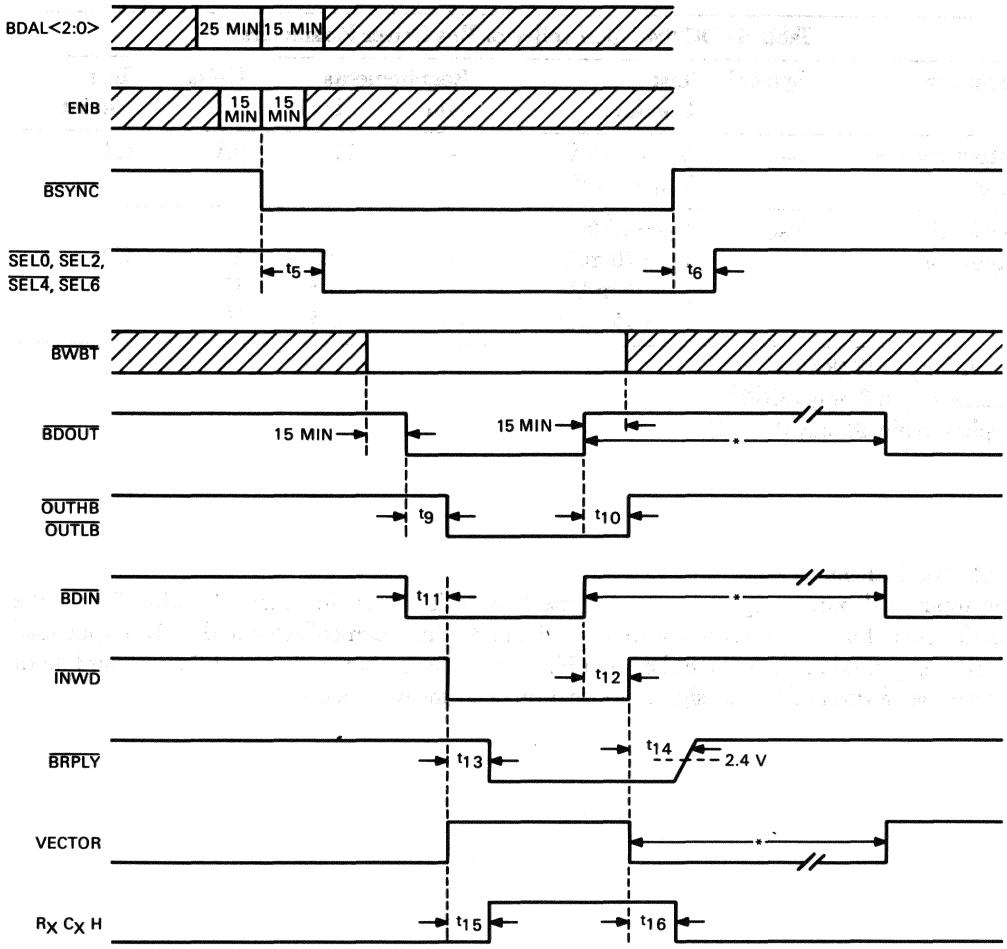
Table 4 • DC004 Open-collector Bus Driver Parameters

Parameter	Symbol	Test Condition	Requirements Min	Requirements Max	Units	Test Circuit
Output reverse current	I_{OH}	$V_{CC}=4.75$ V $V_{OH}=3.5$ V	—	25[1]	μA	C1
Low-level output voltage	V_{OL}	$V_{CC}=4.75$ V				
		$I_{sink}=70$ mA[2]	—	0.8	V	C2
		$_{sink}=16$ mA[3]	—	0.5	V	
		$I_{sink}=15$ mA3	—	0.5	V	

[1]65 μA for pin 18 (RXCX).
[2]Applies to pin 8 only ($\overline{\text{BRPLY}}$).
[3]Applies to pin 18 only (RXCX).

ac Electrical Characteristics

The input and output signal timing for the DC004 is shown in Figure 3. Table 5 lists the specifications for the values referenced on Figure 3. The open-collector and TTL output load circuits referenced in Table 5 are shown in Figure 4. Refer to Appendix D for the standard input voltage waveforms and for the signal propagation delay measurements.



*TIME REQUIRED TO DISCHARGE R_X C_X FROM ANY CONDITION ASSERTED = 150 ns.

NOTE
TIMES ARE IN NANOSECONDS.

Figure 3 ▪ DC004 Signal Timing Sequence

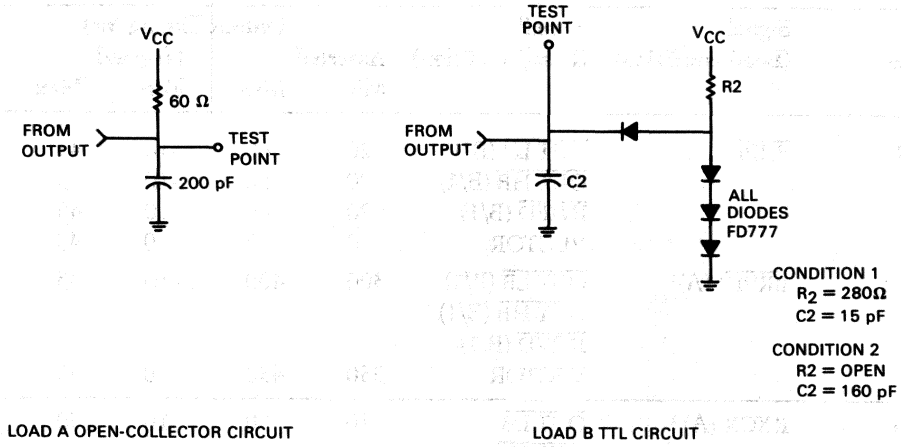


Figure 4 ▪ DC004 Output Load Circuits

Table 5 ▪ DC004 Signal Timing Specifications

Timing Reference	Signal[1] (Load/condition)	Signal[1] (Load/condition)	Output Timing (ns) Asserted Min	Asserted Max	Negated Min	Negated Max
T5,T6	$\overline{\text{SEL 0}}$ (B/2)	$\overline{\text{BSYNC}}$	15	40	5	30
	$\overline{\text{SEL 2}}$ (B/2)		15	40	5	30
	$\overline{\text{SEL 4}}$ (B/2)		15	40	5	30
	$\overline{\text{SEL 6}}$ (B/2)		15	40	5	30
T9, T10	$\overline{\text{OUT LB}}$ (B/2)	$\overline{\text{BDOUT}}$	5	30	5	30
	$\overline{\text{OUT HB}}$ (B/2)		5	30	5	30
T11, T12	$\overline{\text{INWD}}$ (B/2)	$\overline{\text{BDIN}}$	5	30	5	30

Timing Reference	Signal[1] (Load/condition)	Signal[1] (Load/condition)	Output Timing (ns) Asserted Min	Asserted Max	Negated Min	Negated Max
T13, T14	$\overline{\text{BRPLY}}$ (A)[2]	$\overline{\text{OUT LB}}$ (B/1)	20	60	-10	45
		$\overline{\text{OUT HB}}$ (B/1)	20	60	-10	45
		$\overline{\text{INWD}}$ (B/1)	20	60	-10	45
		VECTOR	30	70	0	45
	$\overline{\text{BRPLY}}$ (A)[3]	$\overline{\text{OUT LB}}$ (B/1)	300	400	-10	45
		$\overline{\text{OUT HB}}$ (B/1)				
		$\overline{\text{INWD}}$ (B/1)				
		VECTOR	330	430	0	45
T15, T16	RXCX (A)[2]	$\overline{\text{OUT LB}}$	10	50	10	50
		$\overline{\text{OUT HB}}$				
		$\overline{\text{INWD}}$				
		VECTOR				

[1]Refer to Figure 4 for load circuit configurations.
[2]Pin 18 connections: Rx = 330Ω 5%, Cx = 15 pF 5%.
[3]Pin 18 connections: Rx = 4.64 kΩ 5%, Cx = 220 pF 5%.

• Features

- Used with the DC003 and DC004 circuits to implement a program control device interface.
- Used with the DC003, DC004, DC006, and DC010 circuits to implement a direct memory access interface.
- Functions as a bidirectional buffer between the device logic and computer bus.
- Includes comparison circuit for device address selection.
- Includes constant generator for interrupt vector address generation.
- Includes Q-bus drivers and receivers.

• Description

The DC005 4-bit transceiver, contained in a 20-pin dual-inline package (DIP), implements low-power Schottky technology and functions as a bidirectional buffer between a data bus and peripheral device logic bus. It includes a comparison circuit for device address selection and a constant generator for interrupt-vector address generation. It provides high-impedance inputs and high-drive, open-collector outputs to allow direct connection to a computer data bus structure. The bidirectional device port includes TTL inputs and three-state driver outputs. Figure 1 is a simplified logic diagram of the DC005.

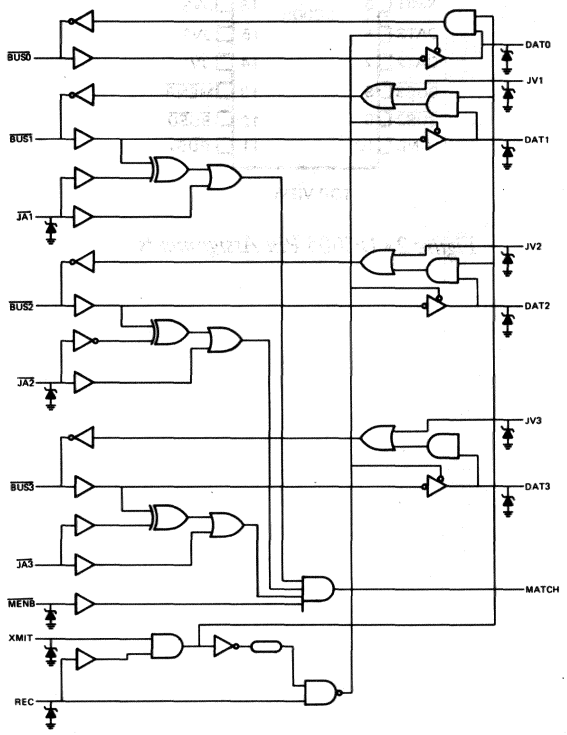


Figure 1 • DC005 Simplified Logic Diagram

Three address select inputs can be configured by jumper leads to enable a comparison to be made between the jumper connections and three bus inputs. An open-collector MATCH output allows several transceiver outputs to be gated to form a composite address match signal. The address jumpers can also be configured to disable the address match condition. The address match condition is controlled by an input line that can enable or disable the MATCH output.

Three vector inputs can also be configured by jumper leads to generate a constant vector value that is transferred to the computer bus. The vector inputs directly drive three of the bus lines to override the function of the control lines.

Two control signals are decoded to select the receive data and transmit data operation and to disable the operation of the device.

▪ Pin and Signal Descriptions

This section provides a brief description of the input and output signals and power and ground connections of the DC005 20-pin DIP. The pin assignments are identified in Figure 2 and the summarized in Table 1.

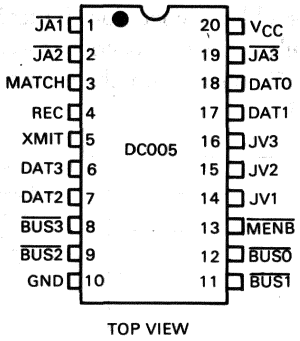


Figure 2 ▪ DC005 Pin Assignments

Table 1 • DC005 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
8-12	$\overline{BUS<3:0>}$	input[1]/output[2]	Bus 3-0 lines—These lines are bidirectional and connect to the BDAL lines of the Q-bus. A low is equal to 1.
6,7,17,18	DAT<3:0>	inputs/outputs[3]	Data 3-0 lines—These lines transfer the inverted data from the bus to the device in receive mode and from the device to the bus in transmit mode. During disable mode, the outputs are at a high impedance.
16-14	JV<3:1>	inputs[4]	Vector jumpers 3-1—These lines directly drive the $\overline{BUS<3:1>}$ lines. A ground connection or an open jumper pin causes an open condition on the corresponding bus pin if the $\overline{XMIT}$ signal is asserted. A high (5 V) connection to the jumper pin will cause a one (low) on the corresponding bus pin. The $\overline{BUS<0>}$ line is not controlled by a jumper.
13	$\overline{MENB}$	input[1]	Match enable—A low input will enable the MATCH output when a match occurs between the level of $\overline{BUS<3:1>}$ signals and the address jumpers JA3 through JA1.
3	MATCH	output[2]	Address match—This output is open when a match occurs between the level of the $\overline{BUS<3:1>}$ lines and the levels selected by the address jumpers JV3 through JV1. The output is low when no match occurs.
19,2,1	$\overline{JA<3:1>}$	inputs[5]	Address jumpers—When connected to ground, these pins allow a match to occur with a one (low) on the corresponding $\overline{BUS<3:1>}$ lines. When these lines are not connected, a match can occur with a zero (high) level on the corresponding bus lines. When connected to 5 V, the corresponding address bit is disconnected.
5 4	XMIT REC	inputs[3]	Transmit/Receive control—Controls the operation of the transceiver as follows: REC XMIT Function 0 0 Disable(open): BUS and DAT lines 0 1 Transmit: DAT to BUS lines 1 0 Receive: BUS to DAT lines 1 1 Receive: BUS to DAT lines
20	V_{cc}	input	Voltage—Power supply dc voltage
10	GND	input	Ground—Common ground reference

[1]high-impedence
[2]open-collector
[3]TTL level
[4]TTL level with pull-down circuit
[5]three-state

Application Information

Refer to the *Chipkit Users Manual LSI-11 Bus Interface Chips* (document no. EJ-01387-92) for general application information. The Q-bus is an LSI-11 bus.

Specifications

The mechanical, electrical, and environmental characteristics and specifications for the DC005 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Operating temperature (T_A): 0°C to 70°C
- Supply voltage (V_{CC}): 5.0 V ±5%

Mechanical Configuration

The physical dimensions of the DC005 20-pin DIP are contained in Appendix E. The materials and construction of the molded DIP are defined in Digital Specification A-PS-2100002-GS.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Supply voltage (V_{CC}): 7.0 V
- Input voltage (V_I): 5.5 V
- Operating temperature (T_A): 0°C to 70°C (32°F to 158°F)
- Storage temperature (T_S): –65°C to 150°C (–149°F to 302°F)

Recommended Operating Conditions

- Supply voltage (V_{CC}): 4.75 V (minimum), 5.0 V (normal), 5.25 V (maximum)
- Supply current (I_{CC}): 120 mA (maximum)
- Free-air temperature: 0°C to 70°C (32°F to 158°F)
- Relative humidity: 10% to 95% (noncondensing)

dc Electrical Characteristics

The dc electrical characteristics of the DC005 for the operating voltage and temperature ranges specified are listed in Tables 2 through 6. Table 2 lists the dc specifications for the TTL input and outputs that do not connect to the bus. Table 3 lists the dc specifications for the high-impedance receiver inputs from the bus. Table 4 lists the dc specifications for the open-collector driver outputs that connect to the bus. Table 5 lists the dc specifications for the three-state jumper lead connections inputs used for the address comparison logic. Table 6 lists the dc specifications for the TTL jumper leads connections used to select a vector address. Refer to Appendix C for test circuit configurations listed in the tables.

Table 2 • DC005 TTL Input and Output Parameters (nonbus)

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units	Test Circuit
High-level input voltage	V_{IH}		2.0	—	V	C1,C2
Low-level input voltage	V_{IL}		—	0.8	V	C1,C2
Input clamp voltage	V_I	$V_{CC}=4.75$ V $I_I=-18$ mA	—	−1.2	V	C3
High-level output voltage	V_{OH}	$V_{CC}=4.7$ V $I_O=-1.0$ mA	3.65	—	V	C1
Low-level output voltage	V_{OL}	$V_{CC}=4.75$ V $I_O=20$ mA	—	0.5	V	C2
Input current at maximum input voltage	I_I	$V_{CC}=5.25$ V $V_I=5.5$ V	—	1.0	mA	C4
High-level input current	I_{IH}	$V_{CC}=5.25$ V $V_I=2.7$ V				C4
		Receive:	—	100	μA	
		Transmit:	—	50	μA	
Low-level input current	I_{IL}	$V_{CC}=5.25$ V $V_I=0.5$ V				C5
		Receive:	—	−2.2	mA	
		Transmit:	—	−1.1	mA	
Short-circuit output current	I_{OS}	$V_{CC}=5.25$ V[1]	−40	−100	mA	C6
Supply current	I_{CC}	$V_{CC}=5.25$ V	—	120	mA	C7
High-impedance state output current[2]	I_O	$V_{CC}=5.25$ V				
		$V_I=3.65$ V	—	100	μA	
		$V_I=0.5$ V	—	−0.36	mA	

[1]Not more than one output shall be short circuited at a time and the duration of the short shall not exceed 1 second.
[2]Off state, DAT<3:0> pins only.

Table 3 • DC005 High-impedance Bus Receiver Parameters

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units	Test Circuit
High-level input voltage	V_{IH}	$V_{CC}=4.75$ V	1.53	—	V	C1,C2
		$V_{CC}=5.25$ V	1.70	—	V	
Low-level input voltage	V_{IL}	$V_{CC}=4.75$ V	—	1.30	V	C1,C2
		$V_{CC}=5.25$ V	—	1.47	V	
Input clamp voltage	V_I	$I_I=-18$ mA $V_{CC}=4.75$ V	—	-1.2	V	C3
High-level input current	I_{IH}*	$V_I=3.8$ V [2]				C4
$\overline{\text{MENB}}$		$V_{CC}=0$ V	—	40	µA	
		$V_{CC}=5.25$ V	—	40	µA	
$\overline{\text{BUS}}$		$V_{CC}=0$ V	—	65	µA	
		$V_{CC}=5.5$ V	—	65	µA	
Low-level input current	I_{IL}	$V_I=0.5$ V				C5
		$V_{CC}=0$ V	—	-10	µA	
		$V_{CC}=5.25$ V	—	-10	µA	

*Includes open-collector leakage current on bus

Table 4 • DC005 Open-collector Bus Driver Parameters

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units	Test Circuit
Output reverse current*	I_{OH}	$V_{CC}=4.75$ V $V_{OH}=5.25$ V	—	25	µA	C1
Low-level output voltage	V_{OL}	$V_{CC}=4.75$ V				C2
MATCH		$I_{sink}=8$ mA	—	0.5	V	
$\overline{\text{BUS}<3:0>}$		$I_{sink}=70$ mA	—	0.8	V	
		$I_{sink}=16$ mA	—	0.5	V	

*Pin 3 only (MATCH). For BUS<3:0> pins, refer to I_{IH} Table 2.

Table 5 • DC005 TTL Three-state Address Input Parameters

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units	Test Circuit
High-level input voltage	V_{IH}		4.75	—	V	C1
Low-level input voltage	V_{IL}		—	0.3	V	C1
Open-circuit input voltage	V_{OP}	$4.75 < V_{CC} > 5.25$	1	2	V	

Table 6 • DC005 TTL Vector Input Parameters

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units	Test Circuit
High-level input voltage	V_{IH}		2.0	—	V	C1
Low-level input voltage	V_{IL}		—	0.8	V	C1
Input clamp voltage	V_I	$V_{CC}=4.75$ V $I_I=-18$ mA	—	-1.2	V	C3
High-level input current	I_{IH}	$V_{CC}=5.25$ V $V_I=2.4$ V	—	1.2	V	C4
Low-level input voltage forcing current	V_{II}	$V_{CC}=4.75$ V $I_I=0.1$ mA	—	0.8	V	C4*
Input current at maximum input voltage	I_{IL}	$V_{CC}=5$ V $V_I=0.4$ V	50	200	μA	C5

*Remaining inputs open.

ac Electrical Characteristics

The input and output signal timing parameters for the DC005 are grouped by functions and shown in Figure 3. Figure 4 shows the load circuits used to measure the signal timing for the open-collector and three-state outputs. Figure 5 shows the three-state voltage waveform parameters. Refer to Appendix D for the standard input and output voltage waveforms parameters used to measure propagation delay.

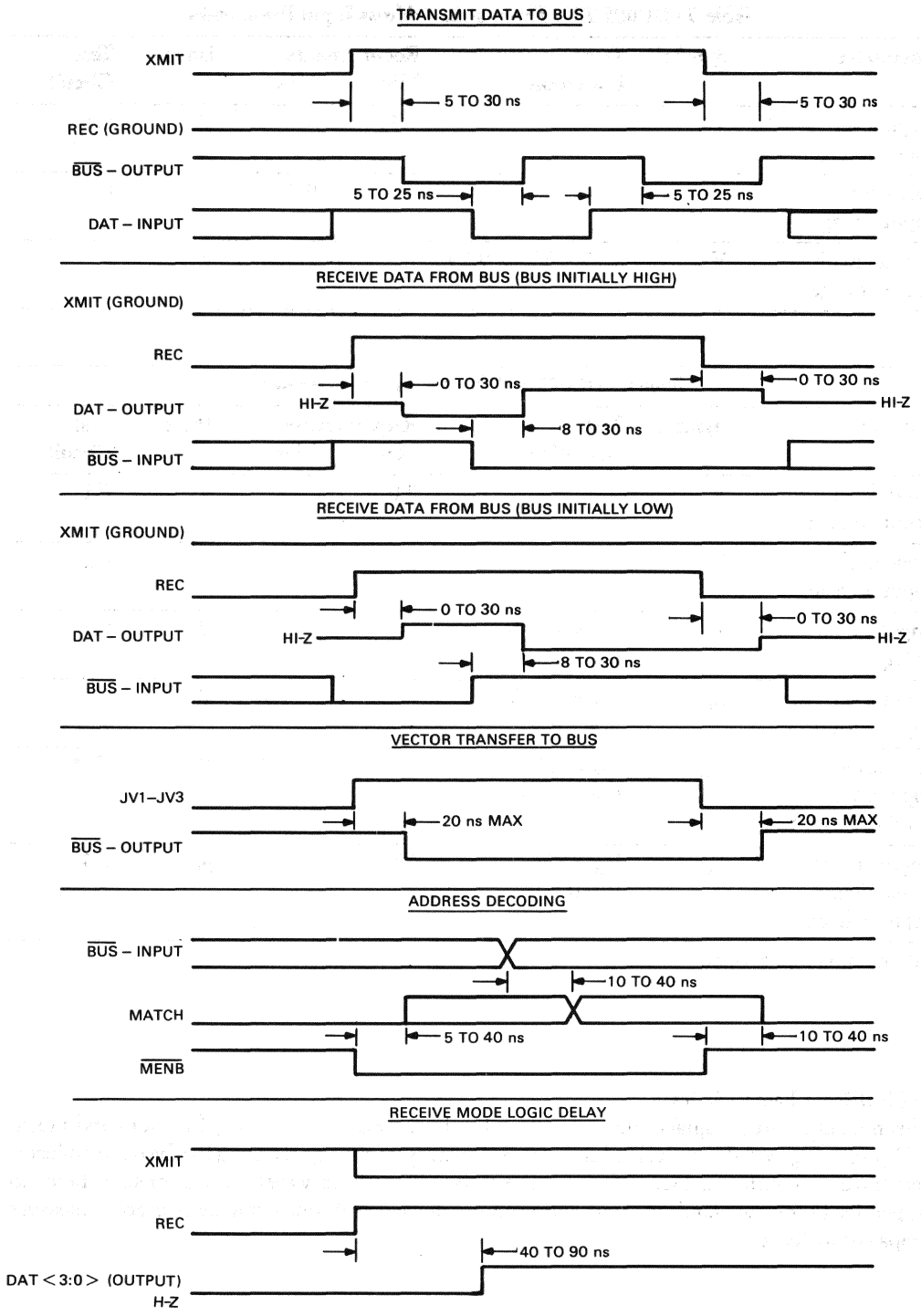


Figure 3 • DC005 Signal Timing Sequence

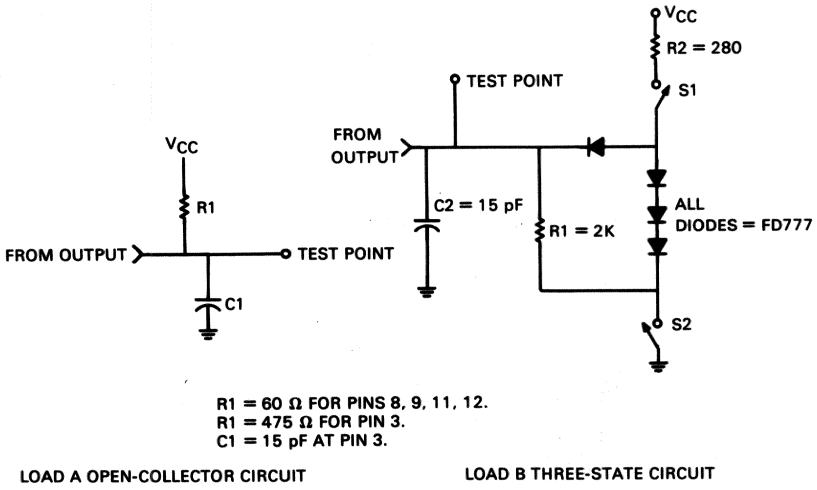


Figure 4 • DC005 Output Load Circuits

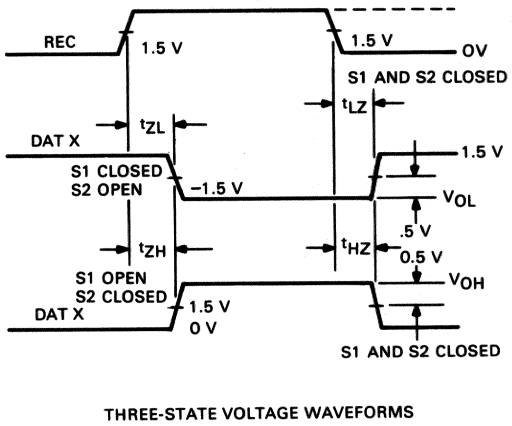


Figure 5 • DC005 Three-state Voltage Waveforms

Features

- Used with the DC003, DC004, DC005, and DC010 to implement a direct memory access interface.
- Contains two 8-bit binary up counters for word byte count and bus address.
- Implements low-power Schottky circuits.
- Includes read and write control logic.
- Includes comparison circuit for device address selection.
- Two DC006s can be cascaded for 16-bit register implementation.
- Contains internal 8-line bus and three-state bus drivers.

Description

The DC006 word count/bus address logic, contained in a 20-pin dual-inline (DIP) package, is designed for use in a direct memory access (DMA) device interface. The DC006 is a low-power Schottky device that connects to the three-state outputs of the DC005 transceiver. The DC006 is controlled by the DC004 register selector, the DC010 direct memory access, and ancillary logic. Figure 1 is a simplified block diagram of the DC006.

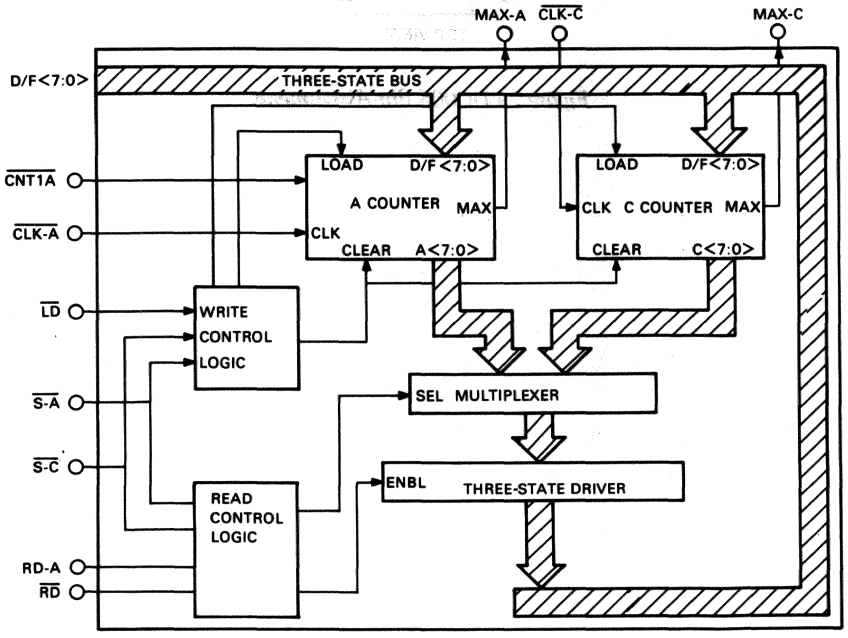


Figure 1 • DC006 Simplified Block Diagram

It includes two separately controlled 8-bit binary up-counters, one for word (byte) count and one for bus address. Each counter can be separately loaded and cleared. The word counter (C counter) is incremented by a count of one and the address counter (A counter) is incremented by a count of one for byte addressing and by a count of two for word addressing. Each counter is read separately. Data from the DC006 is transferred to the three-state bus through the internal drivers.

• Pin and Signal Descriptions

This section provides a brief description of the input and output signals and power and ground connections of the DC005 20-pin DIP. The pin assignments are identified in Figure 2 and then summarized in Table 1.

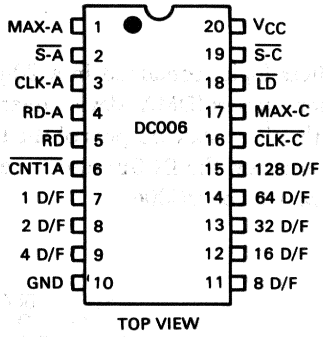


Figure 2 • DC006 Pin Assignments

Table 1 • DC006 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
6	$\overline{\text{CNT1A}}$	input	Increment counter A—Controls the least significant bit of the A counter . When low the A counter is incremented by one. When high, the least-significant bit of the A counter is disabled and the count is incremented by two. When two counters are cascaded, this input should be connected to ground.
3	$\overline{\text{CLK-A}}$	input	Clock A counter—The negative edge of this signal increments the A counter by one or two counts depending on the $\overline{\text{CNT1A}}$ level. The $\overline{\text{CNT1A}}$ and $\overline{\text{LD}}$ signals must be stable while the $\overline{\text{CLK-A}}$ signal is high.
16	$\overline{\text{CLK-C}}$	input	Clock C counter—The negative edge of this signal increments the C counter by a count of one. The $\overline{\text{LD}}$ signal must be stable while the $\overline{\text{CLK-C}}$ signal is high.
2	$\overline{\text{S-A}}$	input	Select A counter—Selects the outputs and functions of the A counter during read and write operations as specified in Tables 2 and 3.
19	$\overline{\text{S-C}}$	input	Select C counter—Selects the outputs and functions of the C counter during read and write operations as specified in Tables 2 and 3.
4	RD-A	input	Read A counter—Selects the outputs and functions of the A counter during read and write operations as specified in Tables 2 and 3.
5	$\overline{\text{RD}}$	input	Read—Enables a read operation as specified in Tables 2 and 3.
18	$\overline{\text{LD}}$	input	Load—A high-to-low transition of this signal enables a load operation to be performed as specified in Table 3. When $\overline{\text{LD}}$ is low, data changes will not occur.
7 8 9 11 12 13 14 15	1D/F 2D/F 4D/F 8D/F 16D/F 32D/F 64D/F 128D/F	outputs*	Data bus—Eight bidirectional lines used to transfer data into or out of the A counter or C counter.
1	MAX-A	output	Maximum A count—Indicates that the A counter has reached a maximum count. The maximum count is 376 when counting by two and 377 when counting by one. The signal is generated by gating $\overline{\text{CLK-A}}$ and the maximum count condition of the counter.

Pin	Signal	Input/Output	Definition/Function
17	MAX-C	output	Maximum C count—Indicates that the C counter has reached a maximum count of 377. This signal is generated by gating $\overline{\text{CLK-C}}$ and the maximum count condition.
20	V_{CC}	input	Voltage—Power supply dc voltage
10	GND	input	Ground—Common ground connection

*TTL three-state

Functional Description

Figure 3 is a simplified logic diagram of the DC006 that shows the read and write control logic, the inputs and outputs of the A and C counters, and the multiplexer logic. Table 2 lists the read and select inputs required to read the counter outputs. Table 3 lists the write-control inputs required to load and clear the counters.

Table 2 • DC006 Read-control Functions

Input Levels*				Output Functions
RD-A	$\overline{\text{RD}}$	$\overline{\text{S-A}}$	$\overline{\text{S-C}}$	D/F<7:0>
L	L	L	L	Clear A and C counters, read C counter
L	L	L	H	A<7:0>
L	L	H	H	C<7:0>
L	H	H	H	High-impedance
L	H	X	X	High-impedance
H	L	L	L	Clear A and C counters, read A counter
H	L	L	H	A<7:0>
H	L	H	L	A<7:0>
H	L	H	H	A<7:0>
H	H	L	L	Clear A and C counters, read A counter
H	H	L	H	A<7:0>
H	H	H	L	A<7:0>
H	H	H	H	A<7:0>

*L = TTL low, H = TTL high, X = TTL low or high

Table 3 • DC006 Write-control Functions

Inputs Levels[1] RD-A	$\overline{RD}$	$\overline{LD}$	$\overline{S\text{-}A}$	$\overline{S\text{-}C}$	Function
L	H	H-L	L	L	Illegal[2]
		H-L	L	H	Load A<7:0>
		H-L	H	L	Load A<7:0>
		X	H	H	Word count/Bus address not selected
		H	L	L	Clear A and B counters
		H	L	H	Loading disabled
		H	H	L	Loading disabled

[1]L = TTL low, H = TTL high, X = TTL low or high, H-L = high-to-low transition.
[2]Simultaneous load and clear operation results in a clear.

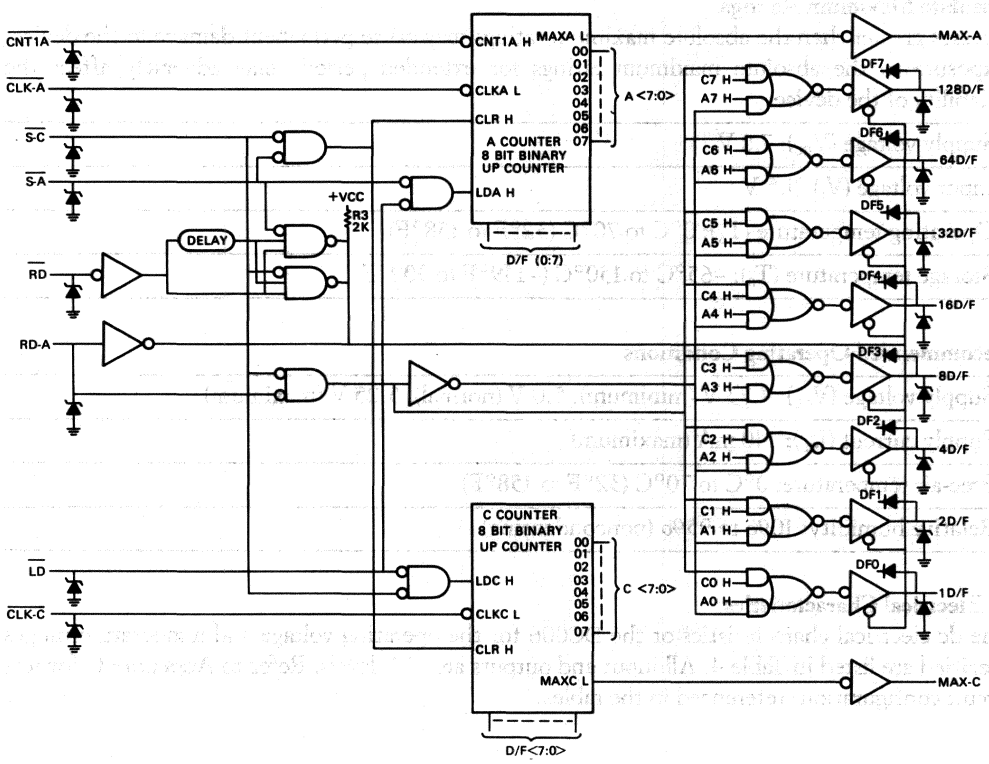


Figure 3 • DC006 Simplified Logic Diagram

Application Information

Refer to the *Chipkit Users Manual LSI-11 Bus Interface Chips* (document no. EJ-01387-92) for general application information. The Q-bus is an LSI-11 bus.

Specifications

The mechanical, electrical, and environmental characteristics and specifications for the DC005 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Operating temperature (T_A): 0°C to 70°C
- Supply voltage (V_{CC}): 5.0 V ±5%

Mechanical Configuration

The physical dimensions of the DC005 20-pin DIP are contained in Appendix E. The materials and construction of the molded DIP are defined in Digital Specification A-PS-1900002-GS.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Supply voltage (V_{CC}): 7.0 V
- Input voltage (V_I): 5.5 V
- Operating temperature (T_A): 0°C to 70°C (32°F to 158°F)
- Storage temperature (T_S): –65°C to 150°C (–149°F to 302°F)

Recommended Operating Conditions

- Supply voltage (V_{CC}): 4.75 V (minimum), 5.0 V (normal), 5.25 V (maximum)
- Supply current (I_{CC}): 140 mA (maximum)
- Free-air temperature: 0°C to 70°C (32°F to 158°F)
- Relative humidity: 10% to 95% (noncondensing)

dc Electrical Characteristics

The dc electrical characteristics of the DC006 for the operating voltage and temperature ranges specified are listed in Table 4. All input and outputs are TTL levels. Refer to Appendix C for test circuit configurations referenced in the tables.

Table 4 • DC006 TTL Input and Output Parameters (nonbus)

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units	Test Circuit
High-level input voltage	V_{IH}		2.0	—	V	C1,C2
Low-level input voltage	V_{IL}		—	0.8	V	C1,C2
Input clamp voltage	V_I	$V_{CC}=4.75$ V $I_I=-18$ mA	—	-1.2	V	C3
High-level output voltage	V_{OH}	$V_{CC}=4.75$ $I_O=-1.0$ mA	2.7	—	V	C1
Low-level output voltage	V_{OL}	$V_{CC}=4.75$ V $I_O=20$ mA	—	0.5	V	C2
Input current at maximum input voltage	I_I	$V_{CC}=5.25$ V $V_I=5.5$ V[1]	—	1.0	mA	C4
High-level input current	I_{IH}	$V_{CC}=5.25$ V $V_I=2.7$ V				C4
Three-state			—	50	μA	
Not three-state			—	55	μA	
Low-level input current	I_{IL}	$V_{CC}=5.25$ V $V_I=0.5$ V				C5
$\overline{\text{CLK-A}}$, $\overline{\text{CLK-C}}$			—	-1.1	mA	
$\overline{\text{CNT1A}}$			—	-1.7	mA	
D/F<0:7> $\overline{\text{LD}}$, $\overline{\text{RD}}$, $\overline{\text{S-C}}$, $\overline{\text{S-A}}$			—	100	μA	
RD-A			—	200	μA	
High-impedance output current	I_O[1]	$V_{CC}=5.25$ V $V_O=3.75$ V	—	100	μA	C1
Short-circuit output current	I_{OS}	$V_{CC}=5.25$ V[2]	-40	-100	mA	C6
Supply current	I_{CC}	$V_{CC}=5.25$ V	—	170	mA	C7

[1]Three-state TTL output in off state.

[2]Not more than one output shall be short circuited at a time and the duration of the short must not exceed 1 second.

ac Electrical Characteristics

The input and output signal timing for the DC006 is shown in Figure 4. Table 5 lists the setup time and pulse characteristics for the times specified in Figure 4. Table 6 lists the propagation delays for the signal timing references in Figure 4. Refer to Appendix D for the standard input and output voltage waveforms parameters used to measure propagation delay. The output loading circuit for the open collectors and the three-state output loading circuit and voltage waveforms are shown in Figure 5. These circuits are referenced in Table 6.

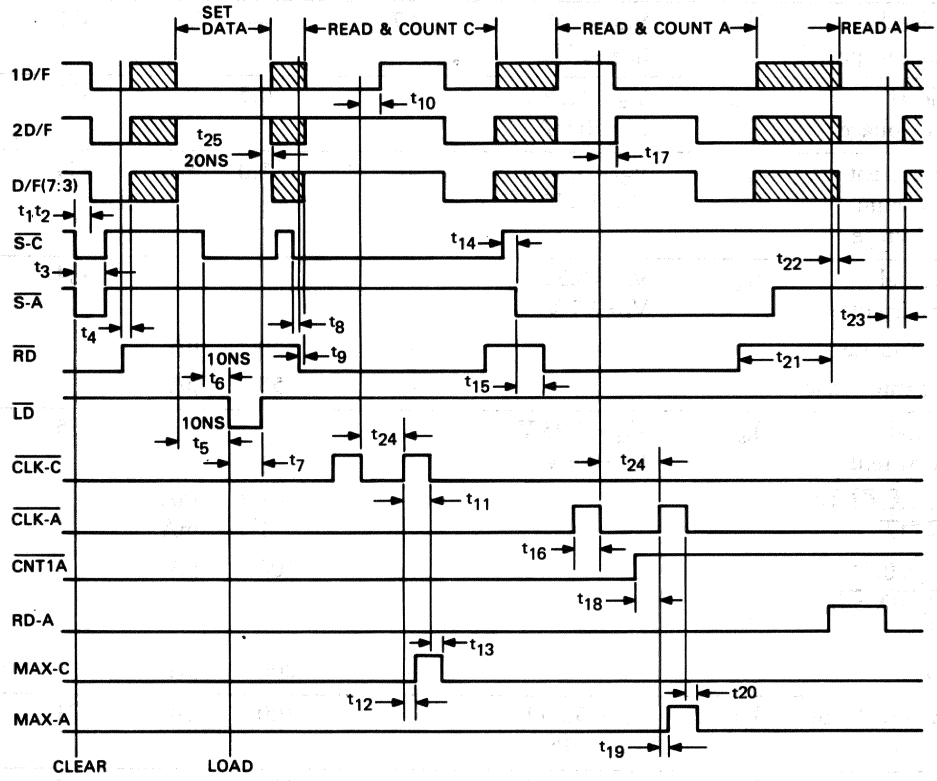


Figure 4 • DC006 Signal Timing Sequence

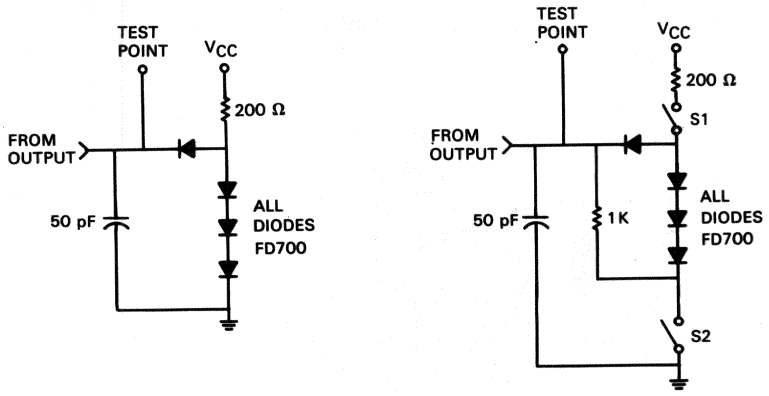
Table 5 • DC006 Pulse-width and Setup time

Time Reference	Signals Reference	Minimum Duration (ns)
t_3 (pulse width)	$\overline{\text{S-C}}$ to $\overline{\text{S-A}}$	50
t_5 (setup)	D/F<7:0> H to $\overline{\text{LD}}$	10
t_6 (setup)	$\overline{\text{S-C}}$ to $\overline{\text{LD}}$	10
t_7 (pulse width)	$\overline{\text{LD}}$	90
t_8 (setup)	$\overline{\text{S-C}}$ to RD	20
t_{11} (pulse width)	$\overline{\text{CLK-C}}$	40
t_{14} (setup)	S-C to S-A	20
t_{15} (setup)	S-A to RD	10
t_{16} (pulse width)	$\overline{\text{CLK-A}}$	40
t_{18} (setup)	$\overline{\text{CNT1A}}$ to $\overline{\text{CLK-A}}$	45
t_{21} (setup)	$\overline{\text{RD}}$ to RD-A	15
t_{24} (off)	$\overline{\text{CLK-A}}$, $\overline{\text{CLK-C}}$	40
t_{25} (data hold)	$\overline{\text{LD}}$ to data in 20	20

Table 6 • DC006 Signal Propagation Delay

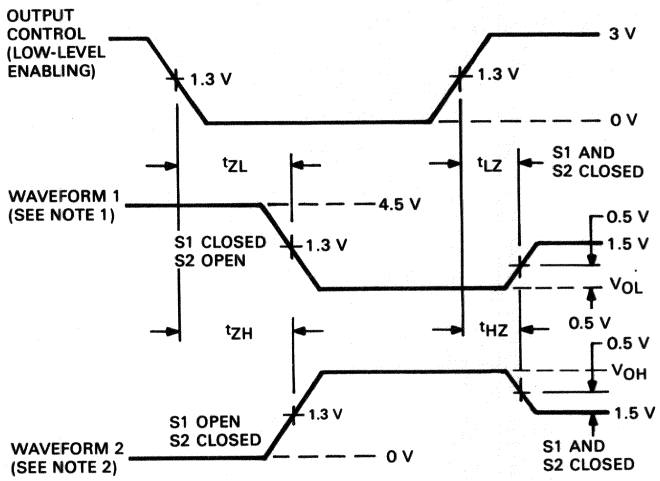
Timing Reference	Input Signal (Transition)	Output Signal (Transition)*	Test Condition	Propagation Delay (ns) Min.	Max.
t_1	$\overline{S\text{-}C}$ (H-L) $\overline{S\text{-}A}$ (H-L)	D/F<7:0> (X-L)	Load A RD-A=0.4 V (C counter)	15	80
t_2	$\overline{S\text{-}C}$ (H-L) $\overline{S\text{-}A}$ (H-L)	D/F<7:0> (X-L)	Load A RD-A=0.4 V (A counter)	15	80
t_4	$\overline{RD}$ (L-H)	D/F<7:0>-(Z)	Load A	10	30
t_9	$\overline{RD}$ (H-L)	(Z)-D/F<7:0>	Load A	34	80
t_{10}	$\overline{CLK\text{-}C}$ (H-L)	MAX-C (L-H)	Load A	18	55
t_{12}	$\overline{CLK\text{-}C}$ (L-H)	MAX-C (L-H)	Load B	10	35
t_{19}	$\overline{CLK\text{-}A}$ (L-H)	MAX-A (L-H)	Load B	10	35
t_{13}	$\overline{CLK\text{-}C}$ (H-L)	MAX-C (L-H)	Load B	10	35
t_{17}	$\overline{CLK\text{-}A}$ (H-L)	D/F2 (L-H)	Load A	18	55
t_{20}	$\overline{CLK\text{-}A}$ (H-L)	MAX-A (H-L)	Load B	10	35
t_{22}	RD-A (L-H)	D/F<7:0> (Z-L) (Z-H)	Load A Load A	10 10	30 30
t_{23}	RD-A (H-L)	D/F<7:0> (L-Z) (H-Z)	Load A Load A	8 8	25 25

*Z = high impedance



LOAD A TTL OUTPUT CIRCUIT

LOAD B THREE STATE CIRCUIT



NOTE

1 WAVEFORM 1 IS FOR AN OUTPUT WITH INTERNAL CONDITIONS SUCH THAT THE OUTPUT IS LOW EXCEPT WHEN DISABLED BY THE OUTPUT CONTROL.

2 WAVEFORM 2 IS FOR AN OUTPUT WITH INTERNAL CONDITIONS SUCH THAT THE OUTPUT IS HIGH EXCEPT WHEN DISABLED BY OUTPUT CONTROL

Figure 5 ▪ DC003 Output Load Circuits

Features

- Used with the DC006 to implement a direct memory access interface.
- Contains logic to request and gain control of the Q-bus.
- Allows four-word or multiple-word transfers.
- Contains bus receivers and drivers for direct connection to the computer bus.
- Uses external free-running clock and internally generates enable and disable clock controls.

Description

The DC010 direct memory access (DMA), contained in a 20-pin dual-inline package (DIP), is a low-power Schottky device used primarily in a DMA device interface to perform the handshake operations required to request and gain control of the Q-bus. Once bus mastership has been established, the DC010 generates the required signals to perform a data-in (DATI) transfer, data-out (DATO) transfer, or a data-in/data-out (DATIO) transfer as selected by the control lines to the DC010. Four words or multiple words can be transferred before control of the bus is relinquished. Figure 1 is a simplified diagram of the logic contained on the DC010 chip. The signals in parenthesis connect internally.

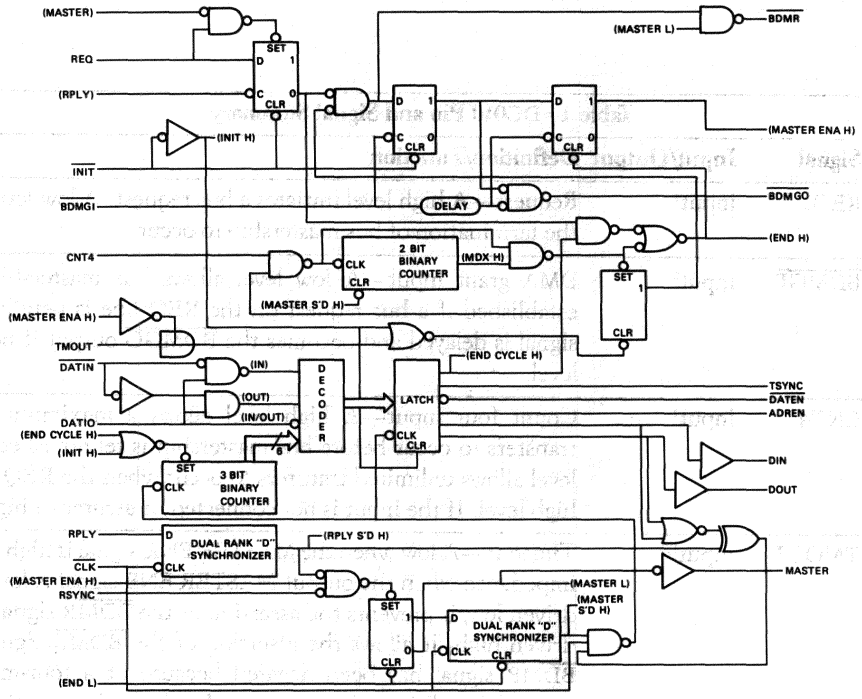


Figure 1 • DC010 Simplified Logic Diagram

• Pin and Signal Descriptions

This section provides a brief description of the input and output signals and power and ground connections of the DC010 20-pin DIP. The pin assignments are identified in Figure 2 and the summarized in Table 1.

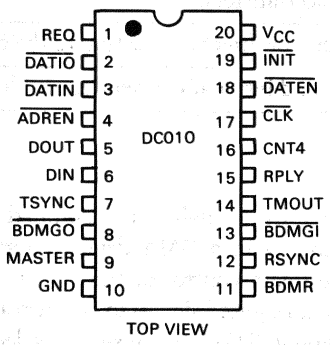


Figure 2 • DC010 Pin Assignments

Table 1 • DC010 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
1	REQ	input[1]	Request—A high level initiates a bus request. A low level allows the termination of bus mastership to occur.
13	$\overline{\text{BDMGI}}$	input[2]	DMA grant input—A low level allows bus mastership to be established if a bus request on the REQ line is pending. This signal is delayed and becomes the $\overline{\text{BDMGO}}$ output if not a low level.
16	CNT4	input[1]	Count four input—A high level allows a maximum of four transfers to occur before bus mastership is relinquished. A low level allows unlimited transfers to occur when the REQ line is a high level. If the input is not connected, it assumes a high state.
14	TMOUT	input[1,3]	Timeout—A low when the MASTER ENA signal is high. A high-impedance when the output MASTER ENB signal is low. When driven low, it prevents the assertion of the $\overline{\text{BDMR}}$ signal. When driven high, it allows the assertion of the $\overline{\text{BDMR}}$ signal if the $\overline{\text{BDMR}}$ signal has been negated because of a four-maximum transfer condition. A resistor and capacitor network can be connected to this pin to delay the assertion of the $\overline{\text{BDMR}}$ signal.

Pin	Signal	Input/Output	Definition/Function
3	$\overline{\text{DATIN}}$	input[1]	Data-in—Used with the DATIO signal to select the type of transfer as specified in Table 3.
2	$\overline{\text{DATIO}}$	input[1]	Data-in/Data-out—Used with the DATIN signal to select the type of transfer as specified in Table 3. If the input is not connected, it assumes a high state.
12	RSYNC	input[1]	Receive synchronize—Allows the device to become bus master according to the relationship of the following signals. $\overline{\text{RSYNC}} + \overline{\text{RPLY}} + \text{MASTER ENA} = \text{MASTER}$
17	$\overline{\text{CLK}}$	input[1]	Clock—Used to generate all transfer timing sequences.
15	RPLY	input[1]	Reply—Enables or disables the clock signal and allows the device to become bus master according to the relationship of the following signals: $\overline{\text{RSYNC}} + \text{RPLY} + \text{MASTER ENA} = \text{MASTER}$
19	$\overline{\text{INIT}}$	input[1]	Initialize—Used to initialize the logic so that the REQ signal will start a bus request transaction. A low level negates the $\overline{\text{BDMR}}$, MASTER, $\overline{\text{DATEN}}$, ADREN, RSYNC, DIN, and DOUT signals.
11	$\overline{\text{BDMR}}$	output[3]	DMA request—A low level indicates that the device is requesting bus mastership. May be connected directly to the Q-bus.
9	MASTER	output[1]	Master—Indicates that a device is bus master and that a transfer sequence is in progress.
8	$\overline{\text{BDMGO}}$	output[3]	DMA grant output—If no request is pending, this signal is the delayed output of the $\overline{\text{BDMGI}}$ input (pin 13). If a request is pending, this signal is not asserted. May be connected directly to the Q-bus.
7	TSYNC	output[1]	Transmit synchronize—Asserted by the device to indicate that a transfer is in progress.
18	$\overline{\text{DATEN}}$	output[1]	Data enable—Asserted to indicate that data may be transferred to the bus.
4	ADREN	output[1]	Address enable—Asserted to indicate that an address may be transferred to the bus.
6	DIN	output[1]	Data in—Asserted to indicate that the bus master device can accept data.
5	DOUT	output[1]	Data out—Asserted to indicate that the bus master device has transferred data to the bus.
20	V_{CC}	input	Voltage—Power supply dc voltage
10	GND	input	Ground—Common ground connection

[1]TTL level
[2]high-impedance
[3]open-collector

Functional Description

The DATIN and DATIO are TTL input levels that select the type of DMA transfer that will occur. Table 2 lists the input levels and the transfer selected.

Table 2 • DC010 Transfer Selection

Inputs Levels* DATIN	DATIO	Transfer Type
X	L	DATIO
L	H	DATI
H	H	DATO

*L = TTL low, H = TTL high, X = TTL low or high

• Application Information

Refer to the *Chipkit Users Manual LSI-11 Bus Interface Chips* (document no. EJ-01387-92) for general application information. The Q-bus is an LSI-11 bus.

• Specifications

The mechanical, electrical, and environmental characteristics and specifications for the DC010 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Operating temperature (T_A): 0°C to 70°C
- Supply voltage (V_{CC}): 5.0 V ±5%

Mechanical Configuration

The physical dimensions of the DC010 20-pin DIP are contained in Appendix E. The materials and construction of the DIP are defined in Digital Specification A-PS-1900002-GS.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Supply voltage (V_{CC}): 7.0 V
- Input voltage (V_I): 5.5 V
- Operating temperature (T_A): 0°C to 70°C (32°F to 158°F)
- Storage temperature (T_S): −65°C to 150°C (−149°F to 302°F)

Recommended Operating Conditions

- Supply voltage (V_{CC}): 4.75 V (minimum), 5.0 V (normal), 5.25 V (maximum)
- Supply current (I_{CC}): 160 mA (maximum)
- Free-air temperature: 0°C to 70°C (32 F to 158°F)
- Relative humidity: 10% to 95% (noncondensing)

dc Electrical Characteristics

The dc electrical parameters of the DC010 for the operating voltage and temperature ranges specified are listed in Table 3. Refer to Appendix C for the test circuit configurations referenced in the table.

Table 3 • DC010 Input and Output Parameters

Parameter	Symbol	Test Condition	Requirements Min	Requirements Max	Units	Test Circuit
High-level input voltage	V_{IH}		2.0		V[1]	C1,C2
		$V_{CC}=4.75$	1.53		V[2,3]	
		$V_{CC}=5.25$	1.70		V[1]	
Low-level input voltage	V_{IL}			0.8	V[1]	C1,C2
		$V_{CC}=4.75$		1.30	V[2,3]	
		$V_{CC}=5.25$		1.47	V[2,3]	
Input clamp voltage	V_I	V_{CC}=open $I_I=-18$ ma	—	−1.2	V[1,2,3]	C3
High-level output voltage	V_{OH}	$V_{CC}=4.75$ $I_O=-1.0$ mA	2.7	—	V[1]	C1
Low-level output voltage	V_{OL}	$V_{CC}=4.75$ V				C2
		$I_O=8$ mA	—	0.5	V[1]	
		$I_O=70$ mA	—	0.8	V[3,4]	
Input current at maximum input voltage	I_I	$V_{CC}=5.25$ V	—	1.0	mA[5]	C4
		$V_I=5.5$ V	—	1.5	mA[6]	
High-level input current	I_{IH}	$V_{CC}=5.25$ V[5]				C4
		$V_I=2.7$ V	—	50	μA	
		$V_I=2.7$ V[6]	—	300	μA	
		$V_I=3.8$ V	—	40	μA[2]	
		$V_I=3.8$ V	—	65	μA[3]	
Low-level input current	I_{IL}	$V_I=0.5$ V[5]				C5
		$V_{CC}=5.25$ V	—	−1.4	mA	
		$V_{CC}=5.25$ V	—	−2.0	mA	
		$V_{CC}=0$–5.25 V	—	−10	μA[2]	
		$V_{CC}=0$–5.25 V	—	−10	μA[3]	

Parameter	Symbol	Test Condition	Requirements Min	Max	Units	Test Circuit
Output leakage	I_{OH}	$V_{CC}=4.75$ V $V_O=5.25$ V		25	μA[4]	C1
Short-circuit output current	I_{OS}	$V_{CC}=5.25$ V[7]	−15	−60	mA[1]	C6
Supply current	I_{CC}	$V_{CC}=5.25$ V	125	160	mA	C7

[1]TTL input
[2]high-impedance input
[3]high-impedance (Schmidt) input, open-collector output
[4]open-collector output
[5]except pin 16 and pin 2
[6]pin 16 and pin 2
[7]Not more than one output shall be short circuited at a time and the duration of the short must not exceed 1 second.

ac Electrical Characteristics

The input and output signal timing for the DC010 is shown in Figures 3 through 7. The setup time, pulse-widths, and switching characteristics referenced in the timing diagrams are listed in Table 4. Table 5 lists the signal propagation delays also referenced in the timing diagrams. Figure 8 shows the load circuits used for measuring for the open-collector and TTL outputs. Refer to Appendix D for the input and output voltage waveforms used for measuring the signal propagation delays.

Figure 3 shows the signal timing required for the DMA bus-request and bus-grant logic. Figure 4 shows the signal timing required for one data-in (DIN) transfer to the DMA interface. The signal timing required for one data-out transfer is shown in Figure 5. Figure 6 shows the signal timing required for multiple-data transfers in and out of the DMA interface. The signal timing for the timeout sequence is shown in Figure 7. The values of the resistor (Rx) and capacitor (Cx) used in the timeout circuit shown in Figure 9 are selected for the proper delay for the next DMA request from the interface. The delay circuit connects to pin 14 (TMOUT).

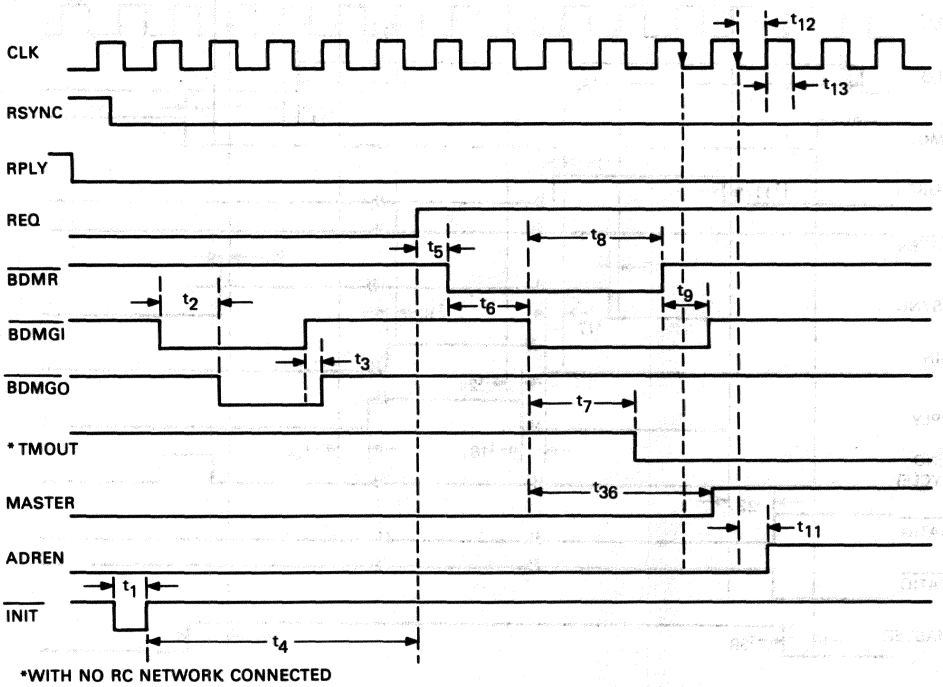


Figure 3 • DC010 DMA Bus Request and Grant Signal Timing Sequence

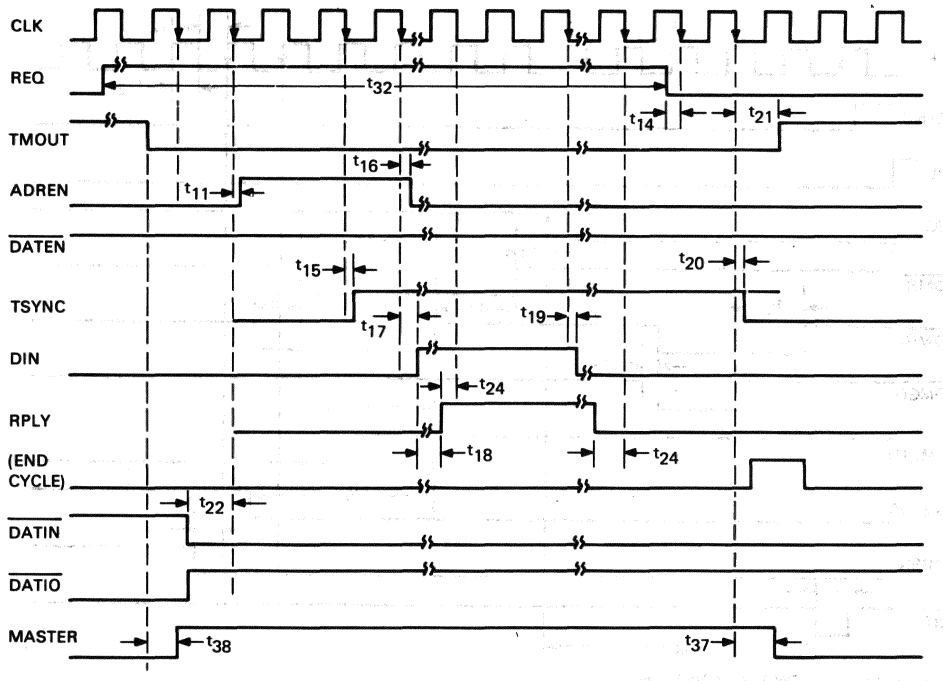
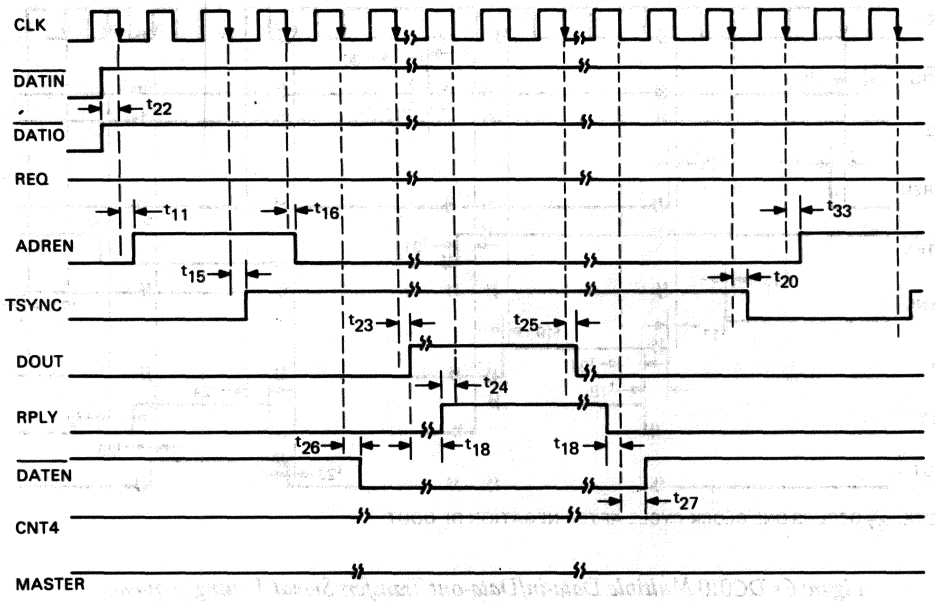
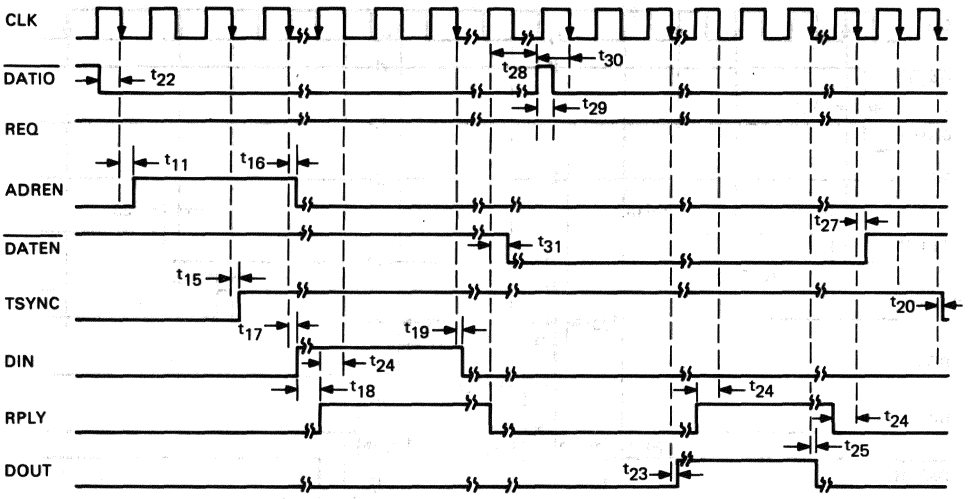


Figure 4 • DC010 One Data-in Transfer Signal Timing Sequence



NOTE: t_{27} OCCURS ONE CLOCK CYCLE AFTER NEGATION OF DOUT

Figure 5 ▪ DC010 Data-out Transfer Signal Timing Sequence



NOTE: t_{27} OCCURS ONE CLOCK CYCLE AFTER NEGATION OF DOUT

Figure 6 ▪ DC010 Multiple Data-in/Data-out Transfers Signal Timing Sequence

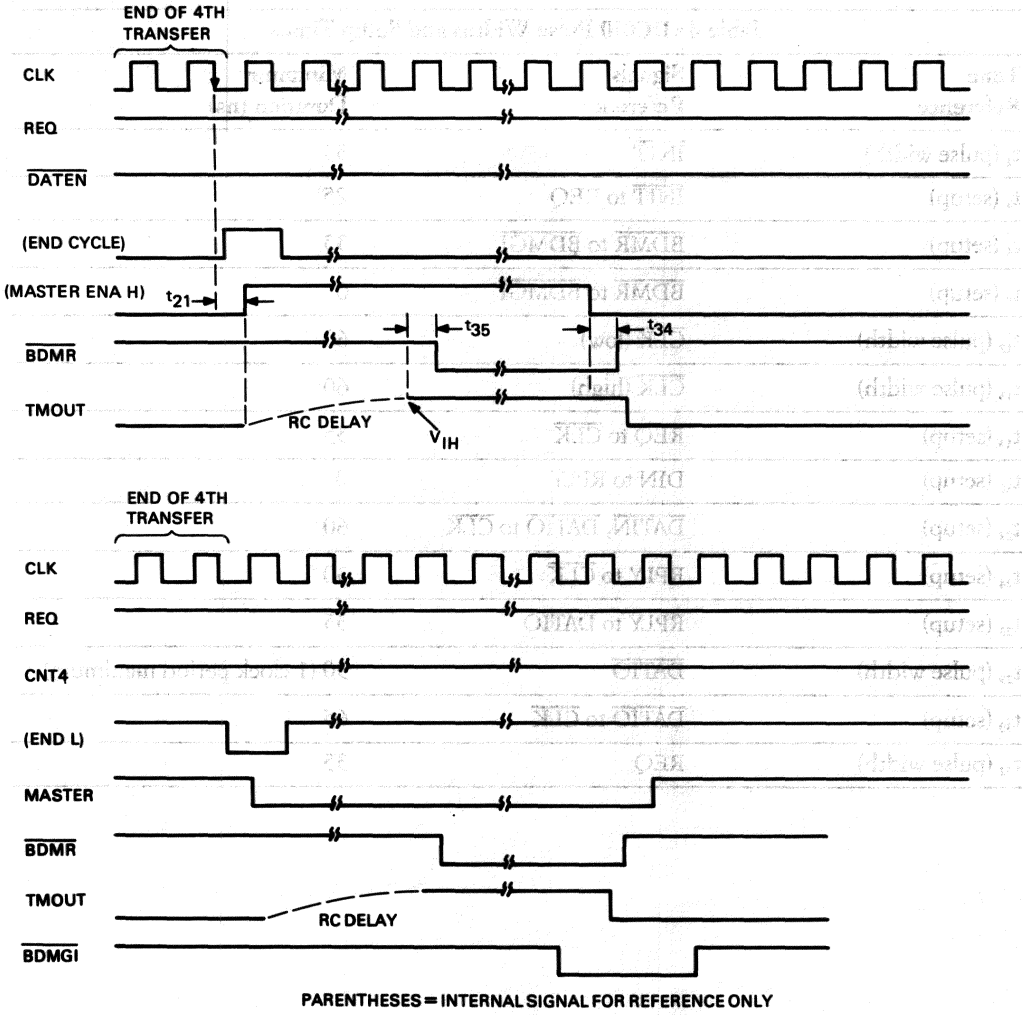


Figure 7 • DC010 Timeout Signal Timing Sequence

Table 4 • DC010 Pulse Widths and Setup Times

Time Reference	Signals Reference	Minimum Duration (ns)
t_1 (pulse width)	$\overline{INIT}$	35
t_4 (setup)	$\overline{INIT}$ to REQ	25
t_6 (setup)	$\overline{BDMR}$ to $\overline{BDMGI}$	35
t_9 (setup)	$\overline{BDMR}$ to $\overline{BDMGI}$	0
t_{12} (pulse width)	$\overline{CLK}$ (low)	60
t_{13} (pulse width)	$\overline{CLK}$ (high)	60
t_{14} (setup)	REQ to $\overline{CLK}$	35
t_{18} (setup)	DIN to RPLY	0
t_{22} (setup)	$\overline{DATIN}$, $\overline{DATIO}$ to $\overline{CLK}$	60
t_{24} (setup)	RPLY to $\overline{CLK}$	30
t_{28} (setup)	RPLY to $\overline{DATIO}$	35
t_{29} (pulse width)	$\overline{DATIO}$	30 (1 clock period maximum)
t_{30} (setup)	$\overline{DATIO}$ to $\overline{CLK}$	65
t_{32} (pulse width)	REQ	35

Table 5 • DC010 Signal Propagation Delays

Timing Reference	Input Signal (Transition)	Output Signal (Transition)	Test Condition[1,2]	Propagation Delay (ns) Min.	Max.
t_2	$\overline{BDMGI}$ (H-L)	$\overline{BDMGO}$ (H-L)	Load A	95	220
t_3	$\overline{BDMGI}$ (L-H)	$\overline{BDMGO}$ (L-H)	Load A	15	60
t_5	REQ (L-H)	$\overline{BDMR}$ (H-L)	Load A	25	70
t_7	$\overline{BDMGI}$ (H-L)	TMOUT (H-L)	Load A	85	230
t_8	$\overline{BDMGI}$ (H-L)	$\overline{BDMR}$ (L-H)	Load A	117	306
t_{11}[2]	$\overline{CLK}$ (H-L)	ADREN (L-H)	Load B	15	60
t_{15}	$\overline{CLK}$ (H-L)	TSYNC (L-H)	Load B	18	60[3]
t_{16}	$\overline{CLK}$ (H-L)	ADREN (H-L)	Load B	20	65[3]
t_{17}	$\overline{CLK}$ (H-L)	DIN (L-H)	Load B	18	60[3]
t_{19}	$\overline{CLK}$ (H-L)	$\overline{DIN}$ (H-L)	Load B	18	60
t_{20}	$\overline{CLK}$ (H-L)	TSYNC (H-L)	Load B	18	60
t_{21}	$\overline{CLK}$ (H-L)	TMOUT (L-H)	Load B	30	90
t_{23}	$\overline{CLK}$ (H-L)	DOUT (L-H)	Load B	60	175
t_{25}	$\overline{CLK}$ (H-L)	DOUT (H-L)	Load B	20	65[3]
t_{26}	CLK (H-L)	$\overline{DATEN}$ (H-L)	Load B	20	65[3]
t_{27}	$\overline{CLK}$ (H-L)	$\overline{DATEN}$ (L-H)	Load B	20	65[3]
t_{31}	RPLY (H-L)	$\overline{DATEN}$ (H-L)	Load B	20	65
t_{33}[2]	$\overline{CLK}$ (H-L)	ADREN (L-H)	Load B	18	60
t_{35}	TMOUT (L-H)	$\overline{BDMR}$ (H-L)	Load B	20	75
t_{36}	$\overline{BDMGI}$ (H-L)	MASTER (L-H)	Load B	90	242
t_{37}	$\overline{CLK}$ (H-L)	MASTER (H-L)	Load B	18	66
t_{38}	RSYNC (H-L)	MASTER (L-H)	Load B	10	58

[1]See Figure 8 for output load circuit configurations.
[2]t_{11} represents the first time ADREN is asserted. t_{33} represents the subsequent assertion of ADREN.
[3]These propagation delays meet the following requirements:

t_{15} to $t_{16} \leq 10$ ns t_{25} to $t_{27} \leq 20$ ns
t_{15} to $t_{17} \leq 10$ ns t_{23} to $t_{26} \leq 40$ ns
t_{16} to $t_{26} \leq 10$ ns t_8 to $t_{36} \leq 27$ ns

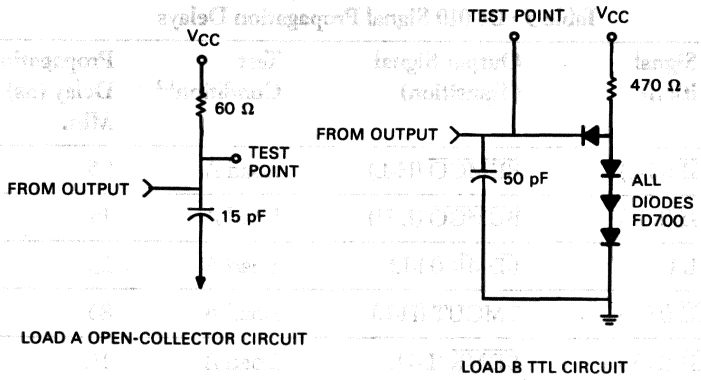


Figure 8 • DC010 Output Load Circuits

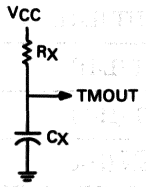


Figure 9 • DC010 Timeout Delay Circuit

Features

- Provides the logic to request and gain control of the UNIBUS
- Used to arbitrate for DMA or interrupt mastership
- Bus receivers and drivers compatible with UNIBUS
- Used to develop UNIBUS interfaces for peripheral devices

Description

The DC013 is a 16-pin, dual-inline package (DIP) used in the development of device interfaces for the UNIBUS. It contains the logic required to perform interrupt bus requests (BR) and nonprocessor direct memory access (DMA) requests to gain control of the UNIBUS.

Input signals from the UNIBUS are received by high-impedance receivers on the DC013 and signals from the DC013 to the UNIBUS are supplied by high-current, open-collector driver outputs. The signals levels between the UNIBUS and the DC013 are compatible. The input and output signals between the device and DC013 are TTL levels.

The DC013 circuits includes bus grant logic, bus busy logic, and slave acknowledge logic. The simplified logic diagram of the DC013 is shown in Figure 1.

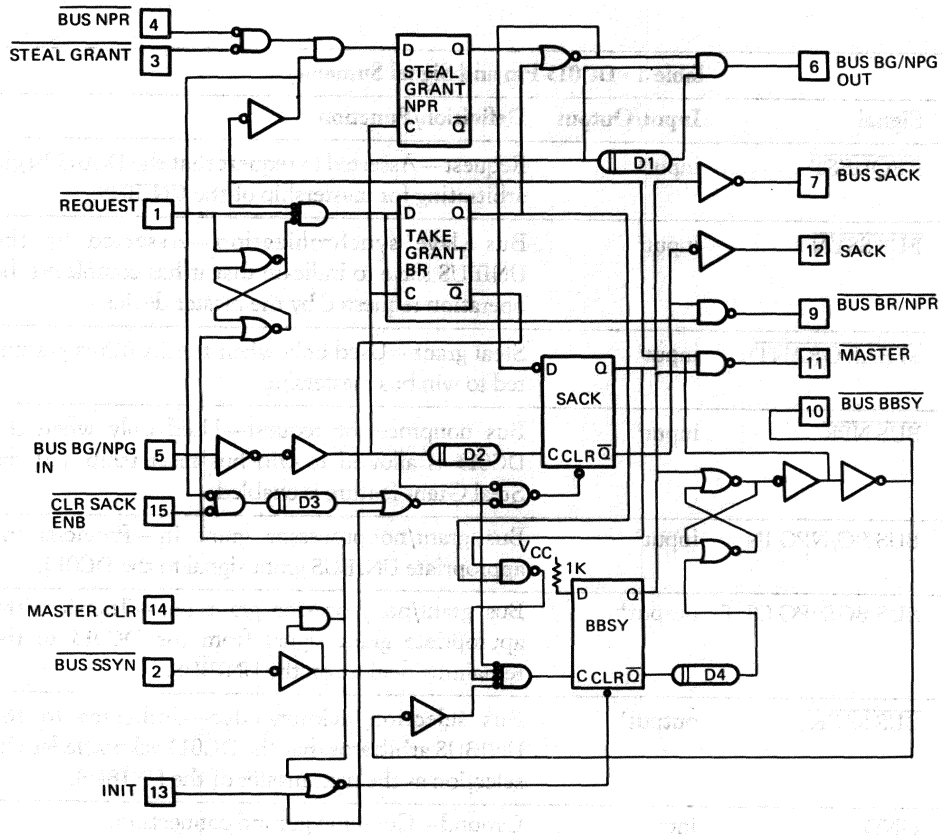


Figure 1 • DC013 Simplified Logic Diagram

• Pin and Signal Definitions

The input and output pins and power and ground connections of the DC013 are shown in Figure 2. Table 1 provides a summary of the signals defined in the following paragraphs.

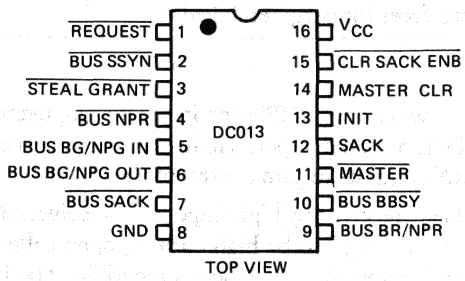


Figure 2 • DC013 Pin Assignments

Table 1 • DC013 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
1	REQUEST	input[1]	Request—Asserted to request that the DC013 begin arbitrating for mastership of the UNIBUS.
2	BUS SSYN	input[2]	Bus slave synchronization—Asserted by the UNIBUS slave to indicate that it has completed the operation requested by the master device.
3	STEAL GRANT	input[2]	Steal grant—Used only when the DC013 is permitted to win bus mastership.
4	BUS NPR	input[2]	Bus nonprocessor request—Used only when the DC013 is allowed to win bus mastership and the Steal Grant feature is enabled.
5	BUS BG/NPG IN	input[2]	Bus grant/nonprocessor grant in—Provides the appropriate UNIBUS grant signal to the DC013.
6	BUS BG/NPG OUT	output[3]	Bus grant/nonprocessor grant out—Transfers the appropriate grant signal from the DC013 to the remaining devices on the UNIBUS.
7	BUS SACK	output[3]	Bus selection acknowledge—Indicates to the UNIBUS arbitrator that the DC013 acknowledges its selection as the next master of the UNIBUS.
8	GND	input	Ground—Common ground connection.

Pin	Signal	Input/Output	Definition/Function
9	$\overline{\text{BUS BR/NPR}}$	output[3]	Bus request/nonprocessor request out—Indicates that the DC013 has been requested by the $\overline{\text{REQUEST}}$ input signal to arbitrate for mastership of the UNIBUS.
10	$\overline{\text{BUS BBSY}}$	input[2]/output[3]	Bus busy—As an input, it informs the DC013 that the current master has completed its use of the UNIBUS. As an output, it allows the DC013 to indicate that it has become the current bus master.
11	$\overline{\text{MASTER}}$	output[1]	Master—Asserted when the DC013 is the current bus master and is asserting the $\overline{\text{BUS BBSY}}$ signal.
12	SACK	output[1]	Selection acknowledge—Asserted when the DC013 is asserting the $\overline{\text{BUS SACK}}$ signal to acknowledge its selection as the next master of the bus.
13	INIT	input[1]	Initialize—Asserted to initialize the DC013.
14	MASTER CLR	input[1]	Master clear—Asserted to allow the $\overline{\text{BUS SSYN}}$ input to clear the DC013.
15	$\overline{\text{CLR SACK ENB}}$	input[1]	Clear selection acknowledge enable—Asserted to allow the UNIBUS arbitration to resume after the DC013 has become bus master.
16	V_{CC}	input	Voltage—Power supply dc voltage.

[1]TTL level
[2]high-impedance UNIBUS input
[3]open-collector UNIBUS output

UNIBUS Signals

$\overline{\text{BUS SSYN}}$—Asserted by the UNIBUS slave to indicate that it has completed the operation reqested by the master device. When the DC013 is used to win interrupt mastership, this input signal indicates that the processor has accepted the interrupt vector and allows the DC013 to release the bus. When the DC013 is used to win DMA mastership, this input may be directly connected to the UNIBUS or to other logic within the master device.

$\overline{\text{STEAL GRANT}}$—When the DC013 is used to win interrupt mastership, this input signal allows it to steal and reply to an interrupt grant intended for another device connected to the UNIBUS. This feature can reduce the overall DMA latency.

$\overline{\text{BUS NPR}}$—When the steal grant feature is enabled and the DC013 requests interrupt mastership, this input informs the DC013 that a DMA device is requesting use of the bus.

$\overline{\text{BUS BG/NPG IN}}$—When the DC013 requests interrupt mastership, this input connects to the appropriate BUS BG7 through BUS BG4 input. When requesting DMA mastership, it connects to the BUS NPG input.

$\overline{\text{BUS BG/NPG OUT}}$—When the DC031 requests interrupt mastership, this output connects to the appropriate BUS BG7 through BUS BG4 output. When the DC013 requests DMA mastership, it connects to BUS NPG output.

BUS SACK—This output indicates to the UNIBUS arbitrator that the DC013 acknowledges its selection as the next master of the UNIBUS.

BUS BR/NPR—This output indicates that the DC013 has been requested by the REQUEST input to arbitrate for mastership of the UNIBUS. If the DC013 is requesting interrupt mastership, this output connects to the appropriate UNIBUS line BUS BR7 through BUS BR4 output. If the DC013 requests DMA mastership, the output connects to the UNIBUS BUS NPR line.

BUS BBSY—After the DC013 has been granted next bus mastership, it must wait for the current master to complete its data transfers. This input informs the DC013 that the current master has completed its use of the UNIBUS and provides an output from the DC013 to indicate that it has become the current master of the UNIBUS.

Device Signals

REQUEST—An input from the device to request that the DC013 begin arbitrating for mastership of the UNIBUS.

MASTER—This output is asserted when the DC013 is asserting the BUS BBSY signal indicating that the DC013 is the current master of the UNIBUS. If the DC013 is requesting interrupt mastership, this output indicates that the interrupt vector from the device should be transferred to the UNIBUS data lines. If the DC013 is requesting DMA mastership, the output is used to trigger tha data transfer logic of the master device.

SACK—This output is asserted when the DC013 is asserting the BUS SACK output to acknowledge its selection as the next bus master.

INIT—This input initializes the logic in the DC013 to end the bus cycle initiated by the DC013.

MASTER CLR—This input allows the BUS SSYN input to clear the DC013 logic. The use of this input is optional when the DC013 is used for DMA transfers.

CLR SACK ENB—This input causes the the DC013 to deassert the BUS SACK output if the DC013 is also asserting the BUS BBSY output. It allows UNIBUS arbitration to resume when the DC013 becomes bus master. When the DC013 is requesting DMA mastership, this output may be used to delay UNIBUS arbitration during multiple data cycles. During the last data cycle, the the master logic should assert the CLR SACK ENB input to the DC031 to allow the arbitration to be resumed.

Power and Ground Connections

Supply voltage (V_{cc}): Connects to the 5-Vdc power supply

Ground (GND): Common signal and voltage ground

• Functional Operation

The following descriptions assume that the reader has a knowledge of the operation of the UNIBUS. Refer to the *PDP-11 UNIBUS Processor Handbook PDP-11/84, PDP-11/44, and PDP-11/24* (Digital document no. EB 26077-41) for a more detailed description of the UNIBUS operation.

Typical UNIBUS interface implementation of the DC013 for interrupt and DMA data transfers is shown in Figure 3. Request and grant signals that are not used are wired through the interface and can be used by subsequent devices.

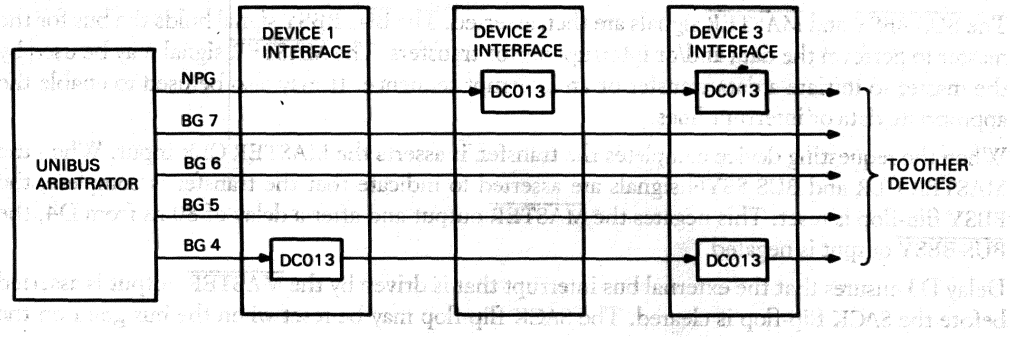


Figure 3 ▪ DC013 Typical DMA and Interrupt Request Configuration

The signals designations, shown in the DC013 control logic Figure 1, that are preceded by a "BUS" designation connect to the UNIBUS lines. The DC013 contains the control logic required for a device to become bus master. A bus master can request and gain control of the UNIBUS to transfer interrupt vectors or data.

The two types of requests that can be initiated are the interrupt bus request (BR) and direct memory access (DMA) nonprocessor request (NPR). A device that requires use of the bus to transfer data or vectors through an interrupt initiates a BR to the arbitrator in a CPU or controller. A BR results in a processor interrupt and the request, grant, and data transfer constitute an interrupt transaction. Only one interrupt transaction may be executed under a single grant.

The NPR is assigned the highest priority and solicits the use of the data section of the bus to transfer data between a device and memory without active participation of the processor. A device that requests the use of the bus sends an NPR to the arbitrator in the CPU when the device is ready to transfer data.

The assertion of the $\overline{\text{REQUEST}}$ input initiates a priority transfer sequence and is applied to the D input of the Take Grant flip-flop. This flip-flop ensures that only one priority transfer sequence is initiated for each $\overline{\text{REQUEST}}$ input. If the requesting device is not presently bus master, the BBSY flip-flop is not set. For each request the $\overline{\text{REQUEST}}$ signal must be negated and again asserted before another cycle can begin.

The state of the $\overline{\text{BUS BBSY}}$ signal determines whether the received bus grant signal (BUS BG/NPG IN) is transferred to the next device or is accepted by the requesting device. If $\overline{\text{BUS BBSY}}$ is not asserted, the logic accepts the grant. If $\overline{\text{BUS BUSY}}$ is asserted, the grant is passed to another device of the same priority level.

The BUS BG/NPG IN signal clocks the Take Grant and Steal Grant flip-flops. With the $\overline{\text{REQUEST}}$ input asserted, the Take Grant flip-flop is set or if another device is asserting the $\overline{\text{BUS NPR}}$ input and the $\overline{\text{STEAL GRANT}}$ input is asserted, the Steal Grant flip-flop is set. Either flip-flop when set disables the bus grant driver and negates the BUS BG/NPG OUT signal. When the grant is accepted, the SACK flip-flop is set after the 75 ns delay from D2. The delay D2 ensures that the Take Grant and Steal Grant flip-flops have time to respond to the BUS BG/NPG IN signal before the SACK flip-flop is clocked. When the Sack flip-flop is set, the $\overline{\text{BUS SACK}}$ output is asserted and the arbitrator is allowed to negate the grant after a minimum of 75 ns and the bus request signal $\overline{\text{BUS BR/NPR}}$ is negated. The BBSY flip-flop is set when the clock input conditions have been satisfied. This occurs when the Take Grant and Sack flip-flops are set and the BUS BG/NPG IN, $\overline{\text{BUS SYNC}}$, and $\overline{\text{BUS BBSY}}$ signals are negated.

The $\overline{\text{BUS BBSY}}$ and $\overline{\text{MASTER}}$ signals are then asserted. The $\overline{\text{BUS BBSY}}$ signal holds the bus for the master to perform the data and/or interrupt vector transfers. The $\overline{\text{MASTER}}$ signal may be used by the master to initiate a data transfer or an interrupt sequence. It may also be used to enable the appropriate data or interrupt lines.

When the requesting device completes the transfer, it asserts the MASTER CLR input. When the MASTER CLR and $\overline{\text{BUS SSYN}}$ signals are asserted to indicate that the transfer is complete, the BBSY flip-flop is reset. This negates the $\overline{\text{MASTER}}$ output and after a delay of 80 ns from D4, the $\overline{\text{BUS BBSY}}$ output is negated.

Delay D3 ensures that the external bus interrupt that is driven by the $\overline{\text{MASTER}}$ output is asserted before the SACK flip-flop is cleared. The SACK flip-flop may be reset when the bus grant on the BUS BG/NPG IN input is negated.

Asserting the INIT input clears the BBSY flip-flop if the bus grant signal BUS BG/NPG input is negated.

A device not requesting the bus may assert the $\overline{\text{STEAL GRANT}}$ input to improve NPR latency. It then receives an $\overline{\text{BUS NPR}}$ input followed by a bus grant signal BUS NPG IN. It blocks the bus grant intended for a another device and asserts the $\overline{\text{BUS SACK}}$ signal that causes the arbitrator to negate the bus grant signal and stop the arbitration. The negation of the BUS BG/NPG IN signal resets the SACK flip-flop.

A device closer to the arbitrator may assert a $\overline{\text{BUS NPR}}$ signal before a grant is issued to a device that previously made a bus request. The arbitrator first honors the NPR and issues an NPG. When the NPR device has completed its transaction, the arbitrator issues a bus grant to the original device.

Interfacing Techniques

The DC013 control logic is used to develop device interfaces to generate bus interrupt and nonprocessor request. The following paragraphs describe the use of the DC013 and standard circuits for typical interface applications.

Bus Request Logic

A typical BR UNIBUS interface using the DC013 and standard ICs for the bus drivers and receivers is shown in Figure 4. This circuit generates one interrupt vector address and transfers the address and data to the UNIBUS. The 8641 ICs can be replaced with two DC021 Octal Bus Transceivers. The interrupt vector from the 74157 ICs is multiplexed with the data lines from the device. The address is selected by inserting and removing jumper leads (V2 through V8). Multiplexing the inputs before the bus drivers reduces the total UNIBUS loading of the device.

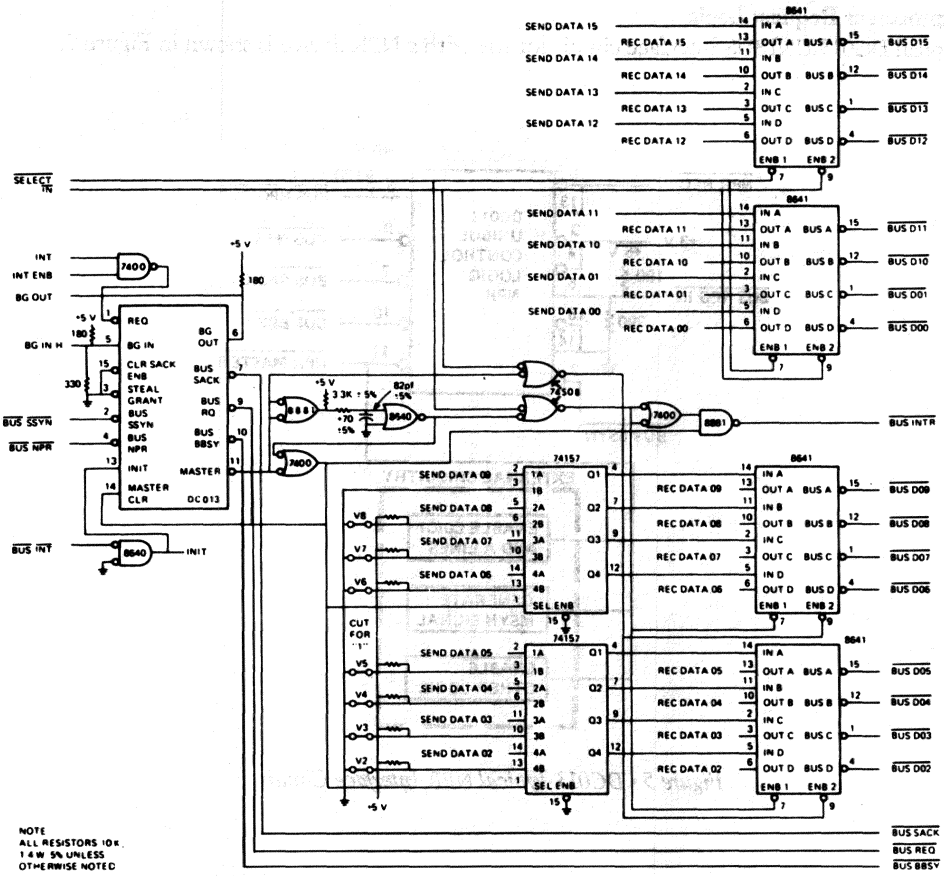


Figure 4 • DC013 Typical BR Interface Circuit

A bus request is initiated by a device when an interrupt is required. The INT ENB signal is normally produced by a bit in the control and status register of the device. When the INT ENB and the INT inputs from the device are asserted, the $\overline{\text{REQUEST}}$ input to the DC013 is asserted to initiate a priority transfer sequence on the UNIBUS. When the device receives the grant, the $\overline{\text{MASTER}}$ output is asserted and begins the interrupt sequence.

The master device negates both the $\overline{\text{SELECT}}$ and $\overline{\text{IN}}$ inputs to disable the UNIBUS transceivers for the D<01:00> and D<15:10> lines.

When the $\overline{\text{SELECT}}$ input to the multiplexer is negated, the A inputs are transferred to the bus. When the SELECT input is asserted, the B inputs are transferred to the bus.

The assertion of the $\overline{\text{MASTER}}$ output enables the SEL inputs to the multiplexers, enables the outputs of the bus transceivers lines BUS D<09:02> and, after a delay, it enables the remaining input to the gate that generates the $\overline{\text{BUS INTR}}$ output. The interrupt vector is therefore available before the interrupt is requested that complies with the bus specifications. When the processor receives the $\overline{\text{BUS INTR}}$ signal, it reads the vector information on the bus and asserts the $\overline{\text{BUS BBSY}}$ signal. The device receives this signal, deasserts the $\overline{\text{MASTER}}$ output to terminate the vector, and deasserts the $\overline{\text{BUS INTR}}$ output. After a delay of 80 ns, it deasserts the $\overline{\text{BUS BBSY}}$ output.

Nonprocessor Request Logic

A typical DC013 UNIBUS interface circuit for use with a NPR device is shown in Figure 5.

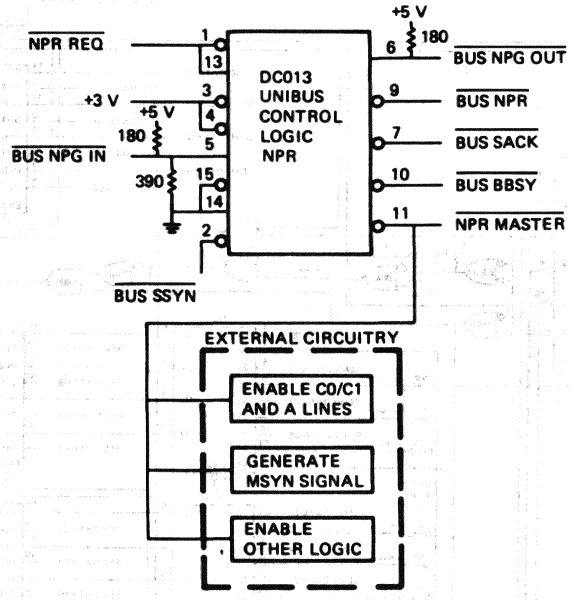


Figure 5 • DC013 Typical NPR Interface Circuit

When an NPR transfer is required, the requesting device asserts the NPR REQ input. The arbitrator recognizes this request and issues a nonprocessor grant by asserting the BUS NPG IN input. The requesting device blocks the grant from being transferred to the next device interface and acknowledges the grant by asserting the BUS SACK output. With this signal asserted, the arbitrator negates the grant and stops the arbitration process. With BUS SACK asserted, the arbitrator negates the grant and stops arbitrating. When the requesting device receives the negation of the BUS BBSY, BUS SSYN, and BUS NPG signals, it asserts the BUS BBSY output, becomes bus master, and initiates the data transfer.

The NPR MASTER output, asserted when the BUS BBSY signal was asserted, is used to drive the external circuits. When the data transfer is complete, the device negates the NPR REQ input to relinquish the bus mastership. The NPR REQ input connects to the INIT input (13). Once a request is initiated, the NPR REQ input must remain asserted until the data transfer is complete to prevent the premature termination of bus mastership.

The Steal Grant flip-flop is disabled by connecting the STEAL GRANT (3) and BUS NPR (4) inputs to 3 V. The BUS NPG input is held asserted by the resistor network. The CLR SACK ENB (15) is asserted and the MASTER CLR input is negated by the ground connection that allows only one bus cycle for each request. For more than one data cycle, the CLR SACK ENB input can be held negated until the beginning of the last bus cycle.

An interrupt request cannot be performed by a device that has become bus master through an NPR. In most NPR applications, an interrupt request usually follows the completion of a set of NPR transfers. This interrupt may be used to notify the processor that the NPR transfers are complete or that an error has occurred during the data transfer.

Bus Loading Configuration

A typical device interface that uses a DC013 for NPR control and a DC013 for BR transfers is shown in Figure 6. The UNIBUS loading for the $\overline{\text{BUS SACK}}$ and $\overline{\text{BUS BBSY}}$ outputs can be reduced by using the wired-OR configurations from the DC013 outputs as shown.

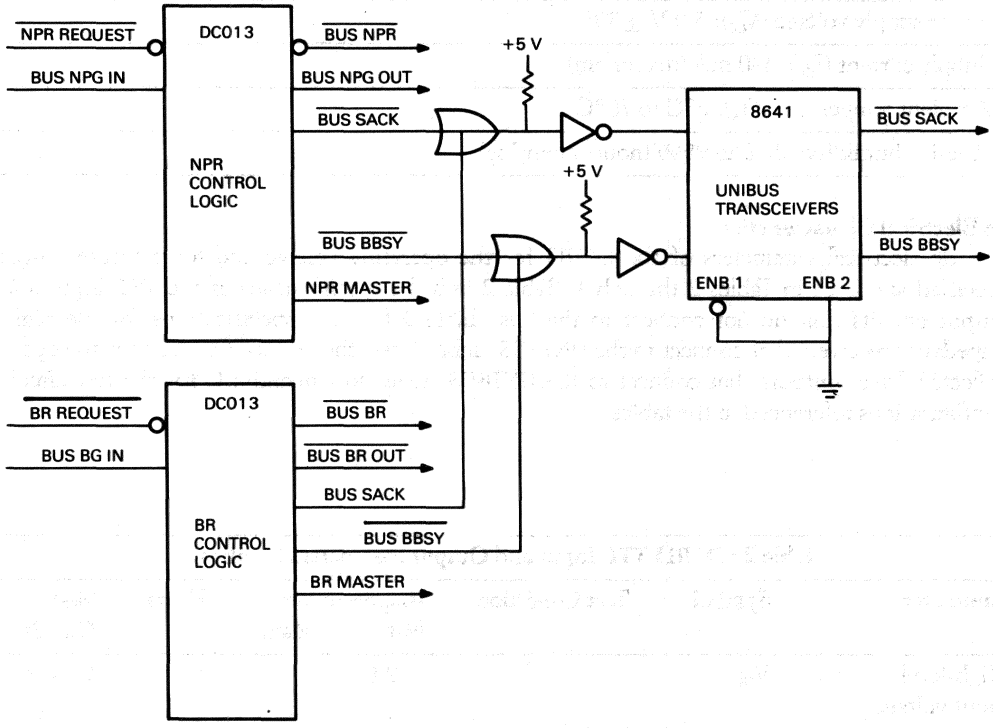


Figure 6 ▪ DC013 BR and NPR Wiring Configuration

▪ Specifications

The mechanical, electrical, and environmental characteristics and specifications for the DC013 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Ambient temperature (T_A): 0°C to 70°C
- Supply voltage (V_{CC}): 5.0 V ±5%

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Supply voltage (V_{CC}): 7.0 V
- Input voltage (V_I): 5.5 V
- Ambient temperature (T_A): 0°C to 70°C
- Storage temperature (T_S): -65°C to 125°C

Recommended Operating Conditions

- Power supply voltage (V_{CC}): 5.0 V ±5%
- Supply current (I_{CC}): 140 mA (maximum)
- Ambient temperature (T_A): 0°C to 70°C
- Relative humidity: 10% to 95% (noncondensing)

dc Electrical Characteristics

The dc electrical parameters of the DC013 for the operating voltage and temperature ranges specified are listed in Tables 2 through 4. Table 2 lists the specifications of the TTL input and output circuits that do not connect to the bus. Table 3 lists the specifications for the high-impedance receivers that connect to the UNIBUS. Table 4 lists the dc specifications for the open-collector driver outputs that connect to the UNIBUS. Refer to Appendix C for the test circuit configurations referenced in the tables.

Table 2 • DC013 TTL Input and Output Parameters (nonbus)

Parameter	Symbol	Test Condition	Requirements Min.	Requirements Max.	Units	Test Circuit
High-level input voltage	V_{IH}		2.0	—	V	C1,C2
Low-level input voltage	V_{IL}		—	0.8	V	C1,C2
Input clamp voltage	V_I	V_{CC}=open I_I=-18 mA	—	-1.2	V	C3
High-level output voltage	V_{OH}	V_{CC}=4.7 V I_O=-1.0 mA	2.7	—	V	C1
Low-level output voltage	V_{OL}	V_{CC}=4.75 V				C2
pin 11		I_O=20 mA[1]	—	0.5	V	
pin 12		I_O=2 mA	—	0.5	V	

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units	Test Circuit
Input current at maximum input voltage	I_I	$V_{CC}=5.25$ V $V_I=5.5$ V	—	1.0	mA	C3
High-level input current	I_{IH}	$V_{CC}=5.25$ V $V_I=2.7$ V				C4
pins 13,15			—	50	μA	
pins 1,14			—	100	μA	
Low-level input current	I_{IL}	$V_{CC}=5.25$ V $V_I=0.5$ V				C5
pin 15			—	-0.55	mA	
pin 1,13,14			—	-1.1	mA	
Short-circuit output current	I_{OS}	$V_{CC}=5.25$ V[2]				C6
pin 11			-40	-100	mA	
pin 12			-5.0	-45	mA	
Supply current	I_{CC}	$V_{CC}=5.25$ V	—	140	mA	C7

[1]Requires a load current of 70 mA at pin 10.
[2]Not more than one output shall be short circuited at a time and the duration of the short shall not exceed 1 second.

Table 3 • DC013 High-impedance Bus Receiver Parameters

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units	Test Circuit
High-level input voltage	V_{IH}	$V_{CC}=4.75$ V $V_{CC}=5.25$ V	1.53 1.70	— —	V V	C1,C2
Low-level input voltage	V_{IL}	$V_{CC}=4.75$ V $V_{CC}=5.25$ V	— —	1.30 1.47	V V	C1,C2
Input clamp voltage	V_I	$V_{CC}=4.75$ V $I_I=-18$ mA	—	-1.2	V	C3
High-level input current*	I_{IH}	$V_{CC}=0$ V $V_I=3.8$ V	—	40	μA	C4
		$V_{CC}=5.25$ V	—	40	μA	
pin 10		$V_{CC}=0$ V	—	40	μA	
pin 10		$V_{CC}=5.25$ V	—	65	μA	

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units	Test Circuit
Low-level input current*	I_{IL}	$V_{CC}=0$ V $V_I=0$ $V_{CC}=5.25$ V $V_I=0$	— —	-10 -10	μA μA	C5
pin 10		$V_{CC}=0$ V $V_I=0.5$ V	—	-10	μA	
pin 10		$V_{CC}=5.25$ V $V_I=0.5$ V	—	-10	μA	

*All pins except pin 10.

Table 4 • DC013 Open-collector Bus Driver Parameters

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units	Test Circuit
Output reverse current*	I_{OH}	$V_{CC}=4.75$ V $V_{OH}=3.5$ V	—	25	μA	C1
pin 10			—	65	μA	
Low-level output voltage	V_{OL}	$V_{CC}=4.75$ V $I_{sink}=70$ mA $I_{sink}=16$ mA	— —	0.8 0.5	V V	C2

*All pins except pin 10.

ac Electrical Characteristics

The input/output signal timing for the UNIBUS request logic is shown in Figure 7. The transient specifications for the signals are listed in Table 5. Refer to Figure 8 for the load circuits used in measuring the TTL outputs and open-collector outputs. Refer to Appendix D for the voltage waveforms used in measuring the propagation delays.

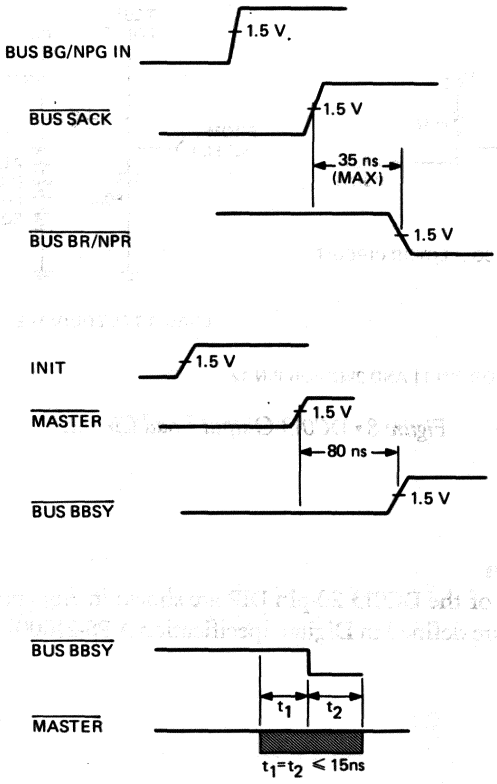


Figure 7 • DC013 Signal Timing Sequence

Table 5 • DC013 ac Signal Transient Specifications

Signal	Input Voltage	Parameters (ns) Rise Time	Fall Time
REQUEST	0 to 3	≤15	≤ 6
CLR SACK ENB	0 to 3	≤15	≤ 6
MASTER CLR	0 to 3	≤15	≤ 6
INIT	0 to 3	≤15	≤ 6
BUS NPR	1 to 2	≤10	≤10
STEAL GRANT	1 to 2	≤10	≤10
BUS BG/NPG IN	1 to 2	≤10	≤10
BUS SSYN	1 to 2	≤10	≤10
BUS BBSY	1 to 2	≤10	≤10

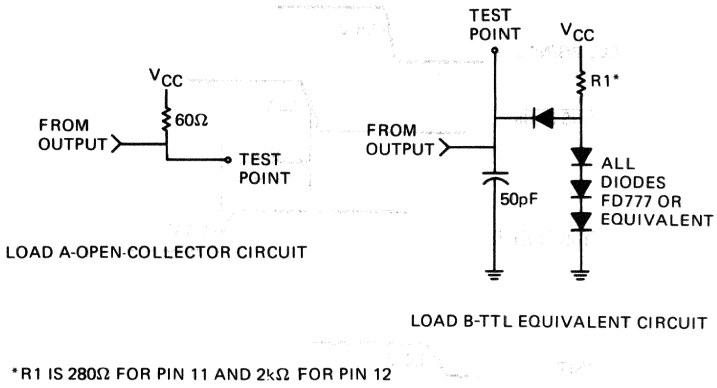


Figure 8 ▪ DC013 Output Load Circuits

Mechanical Configuration

The physical dimensions of the DC013 20-pin DIP are shown in Appendix E. The materials and construction of the DIP are defined in Digital Specification A-PS-210002-GS.

Features

- Bidirectional bus transceivers
- Incorporates receiver noise immunity
- Input and output levels compatible with UNIBUS and Q-bus
- Provides three-state TTL outputs to interface
- Open-collector bus drivers

Description

The DC021 is an octal bus transceiver contained in a 20-pin dual-inline package (DIP) and is compatible with both the UNIBUS and Q-bus. The DC021 provides eight bidirectional channels that transfer information between a wired-OR bus and a user interface. It provides high-impedance receiver inputs and high-current, open-collector driver outputs. The DC021 transfers TTL-level signals between the device logic and the bus. The Select and Enable inputs to the DC021 are used to control the direction of information transfer. The simplified logic diagram of the DC021 is shown in Figure 1.

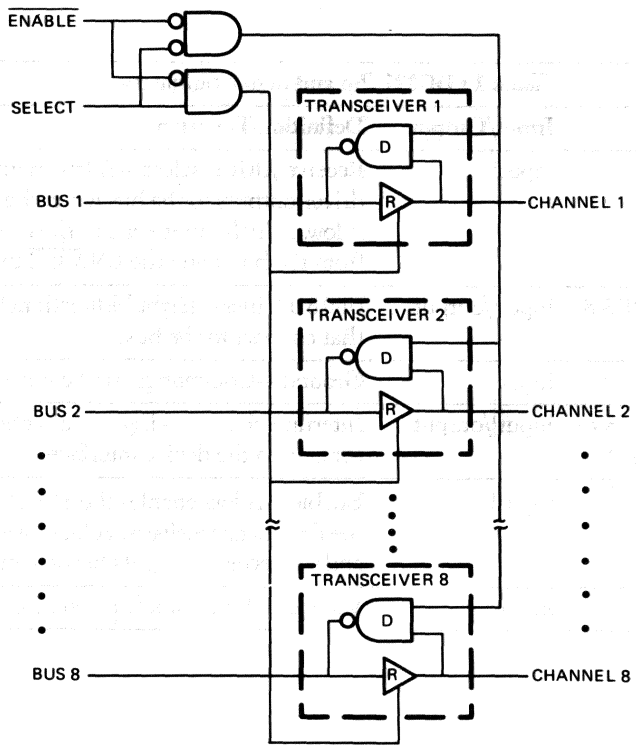


Figure 1 • DC021 Simplified Logic Diagram

• Pin and Signal Definitions

This section provides a brief description of the input and output signals and power and ground connections of the DC021 20-pin DIP. The pin assignments are shown in Figure 2 and the signals are summarized in Table 1.

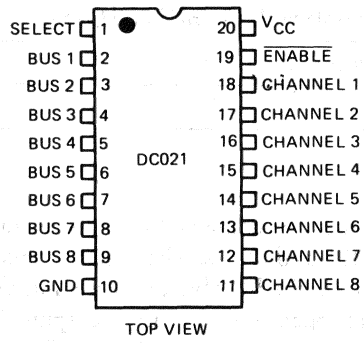


Figure 2 • DC021 Pin Assignments

Table 1 • DC021 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
1	SELECT	input[1]	Receiver/driver select—A low input enables the driver outputs to the bus wnen the $\overline{\text{ENABLE}}$ input is low. A high input enables the receiver inputs from the bus when the $\overline{\text{ENABLE}}$ input is low.
2-9	BUS 1 to BUS 8	input[2]/output[3]	UNIBUS lines—Eight bidirectional line receivers that connect to the bus.
10	GND	input	Ground—Common ground connection.
11-18	CHANNEL 8 to CHANNEL 1	input[1]/output[1]	Interface channel—Eight bidirectional lines that connect to the device interface.
19	$\overline{\text{ENABLE}}$	input[1]	Enable—A low enables the SELECT input to select the drivers or receivers. A high disables the drivers and the receiver outputs become high impedance.
20	V_{CC}	input	Voltage—Power supply dc voltage.

[1]TTL level
[2]high-impedance
[3]open-collector

Enable and Select Function

Table 2 lists the function of the transceiver as related to the $\overline{\text{ENABLE}}$ and SELECT control inputs.

Table 2 • DC021 Control Signal Functions

Signal Inputs*		Function	Driver	Receiver
$\overline{\text{ENABLE}}$	SELECT			
L	L	TTL-to-bus	enabled	high impedance
L	H	bus-to-TTL	disabled	enabled
H	X	no transfer	disabled	high impedance

*H = high level
L = low level
X = high or low level

• Specifications

The mechanical, electrical, and environmental characteristics and specifications for the DC021 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise.

- Ambient temperature (T_A): 0°C to 70°C
- Supply voltage (V_{CC}): 5.0 V ±5%

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Supply voltage (V_{CC}): 7.0 V
- Input voltage (V_{in}) and output voltage (V_{out}): 5.5 V
- Ambient temperature (T_A): 0°C to 70°C
- Storage temperature (T_S): −65°C to 150°C

Recommended Operating Conditions

- Power supply voltage (V_{CC}): 5.0 V ±5%
- Supply current (I_{CC}): 240 mA (maximum)
- Ambient temperature (T_A): 0°C to 70°C
- Relative humidity: 10% to 95% (noncondensing)

dc Electrical Characteristics

The dc electrical specifications of the DC021 for the operating voltage and temperature ranges specified are listed in Tables 3 through 5. Table 3 lists the dc parameters for the bus-to-channel receiver input and outputs. Table 4 lists the dc parameters for the channel-to-bus driver inputs and outputs. Table 5 lists the dc parameters for the enable and select inputs. Refer to Appendix C for the test circuit configurations referenced in Tables.

Table 3 • DC021 Receiver Input and Output dc Parameters

Parameter	Symbol	Test Condition	Requirements Min.	Requirements Max.	Units	Test Circuit
High-level input voltage	V_{IH}	Enable/Select[1] Receiver out = 20 mA Receiver out < 0.5 V				C1,C2
		V_{CC} = 4.75 V	1.72	—	V	
		V_{CC} = 5.25 V	1.9	—	V	
Low-level input voltage	V_{IL}	Enable/Select[1] Receiver out = −2.0 mA Receiver out > 2.4 V				C1,C2
		V_{CC} = 4.75 V	—	1.50	V	
		V_{CC} = 5.25 V	—	1.66	V	
High-level output voltage	V_{OH}	Enable/Select[2] Receiver in = 0.4 mA Receiver out = −2.0 mA				C1
		V_{CC} = 4.75 V	2.4	—	V	
Low-level output voltage	V_{OL}	Enable/Select[2] Receiver in = 0.4 V Receiver out = 20 mA				C2
		V_{CC} = 4.75 V	—	0.5	V	
Short-circuit output current	I_{OS}	Enable/Select[3] Receiver in = 0.5 V Receiver out = 0 V				C6
		V_{CC} = 5.25 V	−40	−100	mA	
Three-state output leakage current	I_{OZL}	Enable/Select[4] Receiver in = 0.8 V Receiver out = 0.4 V	—	50	μA	

[1]Enable = 0.8 V, Select = 2.0 V
[2]Enable = 0.8 V, Select = 2.4 V
[3]Enable = 0 V, Select = 2.0 V
[4]Enable = 2.0 V, Select = 2.0 V

Table 4 • DC021 Driver Input and Output dc Parameters

Parameter	Symbol	Test Condition	Requirements Min.	Requirements Max.	Units	Test Circuit
High-level output leakage current	I_{OZH}	Enable/Select[1] Driver in = 0.8 V Driver out = 4.0 V V_{CC} = 5.25 V	—	100	µA	C9
		Enable/Select[2] Driver in = 0.8 V Driver out = 4.0 V V_{CC} = 5.25 V	—	100	µA	
		Enable/Select[1] Driver in = 0.8 V Driver out = 0.4 V V_{CC} = 0 V	—	100	µA	
		Enable/Select[2] Driver in = 0.8 V Driver out = 4.0 V V_{CC} = 0 V	—	100	µA	
High-level input current and receiver output leakage current	I_{IH}	Enable/Select[3] Driver in = 0.4 V Receiver in = 5.5 V V_{CC} = 5.25 V	—	200	µA	C4
	I_{OZH}[7]	Enable/Select[4] Driver in = 2.0 V Receiver in = 4.75 V	—	60	µA	C9
Low-level output leakage current	I_{OZL}	Enable/Select[1] Driver in = 0.8 V Driver out = 0.4 V V_{CC} = 5.25 V	—	−85	mA	C8
		Enable/Select[5] Driver in = 2.0 V Driver out = 0.4 V V_{CC} = 5.25 V	—	−85	mA	
Low-level output voltage	V_{OL}	Enable/Select[5] Driver in = 2.0 V Driver out = 100 mA V_{CC} = 4.75 V	—	0.7	V	C9
		Driver out = 130 mA	—	0.75	V	
Low-level input current	I_{IL}	Enable/Select[6] Driver in = 0.4 V Receiver in = 0.8 V V_{CC} = 5.25 V	—	−1.6	mA	C5

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units	Test Circuit
High-level input voltage	V_{IH}	[8]	2.0	—	V	C1,C2
Low-level input voltage	V_{IL}	[8]	—	0.8	V	C1,C2

[1]Enable = 0.8 V, Select = 0.8 V
[2]Enable = 2.0 V, Select = 2.0 V
[3]Enable = 5.5 V, Select = 5.5 V
[4]Enable = 2.0 V, Select = 2.4 V
[5]Enable = 0.8 V, Select = 2.0 V
[6]Enable = 0.4 V, Select = 0.4 V
[7]I_{IH} total consists of 40 μA (maximum) and I_{OZ} 20 μA (maximum) leakage current in the high-impedance state.
[8]Applies to all possible combinations of V_{IH} and V_{IL} at 0.8 V or 2.0 V

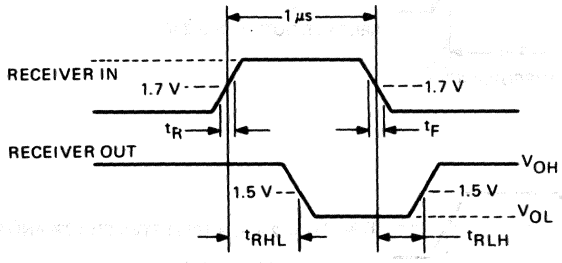
Table 5 • DC021 Enable and Select Input dc Parameters

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units	Test Circuit
High-level input voltage	V_{IH}	Enable or Select input[1]	2.0	—	V	C1,C2
Low-level input voltage	V_{IL}	Enable or Select input[1]	—	0.8	V	C1,C2
High-level input current	I_{IH}	V_{CC} = 5.25 V Enable = 2.4 V Enable input	—	80	mA	C4
		Enable = 5.5 V Enable input	—	100	mA	
		Select = 2.4 V Select Input	—	80	mA	
		Select = 5.5 V Select input	—	100	mA	
Low-level input current	I_{IL}	V_{CC} = 5.25 V Enable or Select input = 0.4 V	—	−0.4	mA	C5
Input diode-clamp voltage	V_I	V_{CC} = 5.0 V ±5%[2] Enable = −18 mA Select = −18 mA Driver in = −18 mA	—	−1.2	V	C3

[1]Applies to all possible combinations of V_{IH} and V_{IL} at 0.8 V or 2.0 V.
[2]Ambient temperature is 25°C. One input at a time.

ac Electrical Characteristics

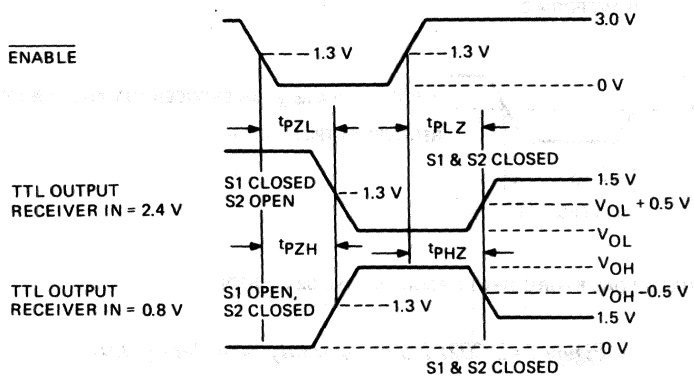
The propagation delays for the receiver input and output signals are listed in Table 6 and the waveforms referenced in the table are shown in Figures 3 and 4. The load circuits referenced in the table and used in the delay measurements are shown in Figure 8.



t_R AND t_F = 10 ns BETWEEN 10% AND 90% LEVELS.

REFER TO FIGURE 8, LOAD A FOR OUTPUT CIRCUIT.

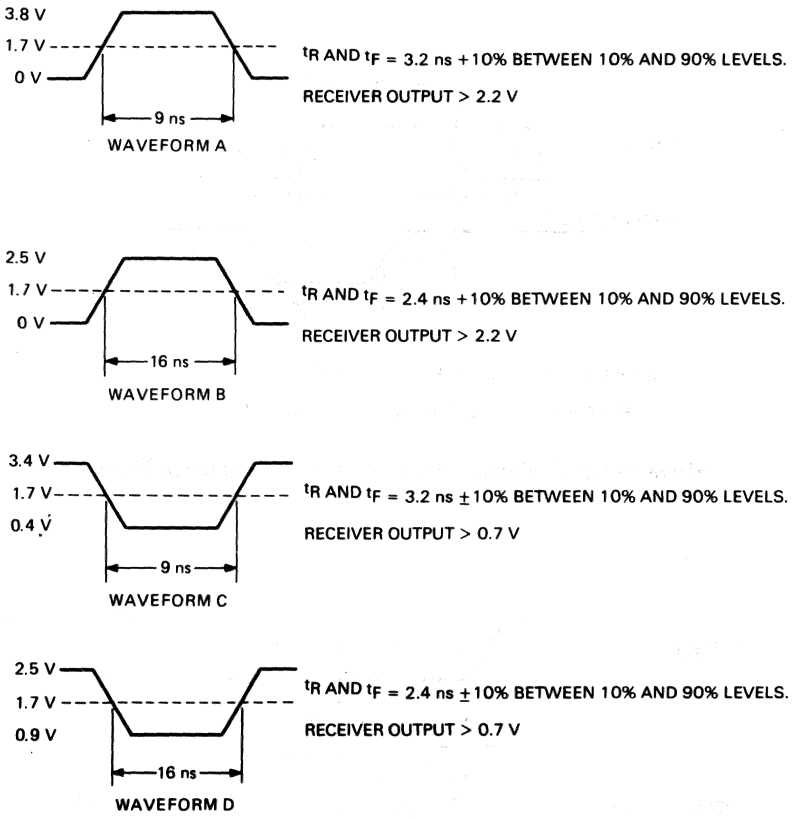
Figure 3 • DC021 Receiver Input to Output Propagation Delays



REFER TO FIGURE 8, LOAD B FOR OUTPUT CIRCUIT

Figure 4 • DC021 Receiver Enable and Disable Propagation Delays

The receiver input waveforms for the noise immunity test are shown in Figure 5. The values indicated are for no response at the output. The load circuit referenced in Figure 5 and used in the delay measurements is shown in Figure 8.



REFER TO FIGURE 8, LOAD B FOR OUTPUT CIRCUIT. (S1 AND S2 CLOSED)

Figure 5 • DC021 Noise Immunity Input Waveforms

Table 6 • DC021 Receiver ac Propagation Delays

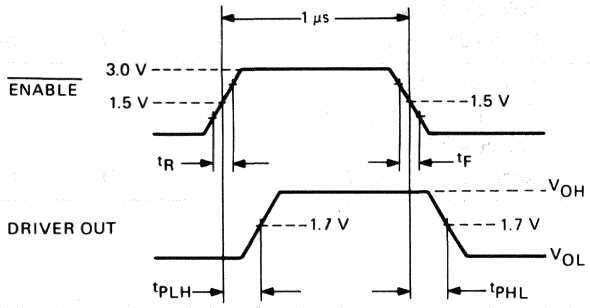
Symbol	Voltage Waveform	Test Condition	Requirements Min.	Requirements Max.	Units	Load Circuit
t_{RLH}	Figure 3	V_{CC} = 4.75 V Enable = 0.8 V Select = 2.0 V				Figure 8A
		V_{in} = 0.8 V to 2.6 V	—	35	ns	
		V_{in} = 1.2 V to 2.2 V	—	39	ns	
t_{RHL}	Figure 3	V_{CC} = 4.75 V Enable = 0.8 V Select = 2.0 V				Figure 8A
		V_{in} = 0.8 V to 2.6 V	—	35	ns	
		V_{in} = 1.2 V to 2.2 V	—	39	ns	
t_{PZL}	Figure 4	V_{CC} = 4.75 V Enable = 0 V to 3.0 V Select = 2.0 V				Figure 8B
		Receiver in = 2.2 V	—	37	ns	
t_{PZH}	Figure 4	V_{CC} = 4.75 V Enable = 0 V to 3.0 V Select = 2.0 V				Figure 8B
		Receiver in = 2.6 V	—	30	ns	
t_{PLZ}	Figure 4	V_{CC} = 4.75 V Enable = 0 V to 3.0 V Select = 2.0 V				Figure 8B
		Receiver in = 2.2 V	—	30	ns	
t_{PHZ}	Figure 4	V_{CC} = 4.75 V Enable = 0 V to 3.0 V Select = 2.0 V				Figure 8B
		Receiver in = 2.2 V	—	30	ns	

The propagation delays for the input signals to output signals are listed in Table 7 and the delay waveforms referenced in the table are shown in Figures 6 and 7. The load circuit referenced in the table and used in the delay measurements are shown in Figure 8.

Table 7 • DC021 Driver ac Propagation Delays

Symbol	Voltage Waveform	Test Condition	Requirements Min.	Requirements Max.	Units	Load Circuit
t_{PLH}	Figure 6	V_{CC} = 4.75 V Select = 0.8 V Receiver in = 0.5 V	—	35	ns	Figure 8C
t_{PHL}	Figure 6	V_{CC} = 4.75 V Select = 0.8 V Receiver in = 2.4 V	—	35	ns	Figure 8C
t_{DLH}	Figure 7	V_{CC} = 4.75 V Enable = 0.8 V Select = 0.8 V	—	25	ns	Figure 8C
t_{DHL}	Figure 7	V_{CC} = 4.75 V Enable = 0.8 V Select = 0.8 V	—	25	ns	Figure 8C
t_R/t_F	Figure 7	[1]	8.0		ns	Figure 8C
C_{DO}[2]		0 V ≤ V_{DO} ≤ 4.0 V	—	20	pF	

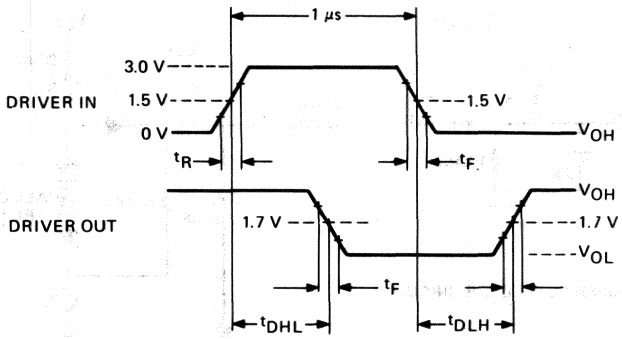
[1] C_L = 15 pF including probe and jig capacitance.
[2] Driver output node capacitance.



t_R AND t_F = 2.5 ns BETWEEN 10% AND 90% LEVELS.

REFER TO FIGURE 8, LOAD C OUTPUT CIRCUIT.

Figure 6 • DC021 Enable Input to Driver Output Propagation Delays



t_R AND t_F = 2.5 ns BETWEEN 10% AND 90% LEVELS.

REFER TO FIGURE 8, LOAD C FOR OUTPUT CIRCUIT.

Figure 7 • DC021 Driver Input-to-Output Propagation Delays

Table 8 lists the current requirements of the DC021 for various input signal conditions.

Table 8 • DC021 Current Requirements

Test Condition*	Function	Requirements Min.	Requirements Max.	Units
Select = 0.8 V Enable = 0.8 V Driver in = 3.0 V	TTL-to-bus	—	240	mA
Select = 0.8 V Enable = 0.8 V Driver in = 0 V	TTL-to-bus	—	150	mA
Select = 2.0 V Enable = 0.8 V Receiver in = 2.4 V	bus-to-TTL	—	165	mA
Select = 2.0 V Enable = 0.8 V Receiver in = 0.5 V	bus-to-TTL	—	110	mA
Select = 2.0 V Enable = 2.0 V Receiver in = 2.4 V Driver in = 0.8 V	disabled	—	165	mA

*V_{cc} = 5.25 V

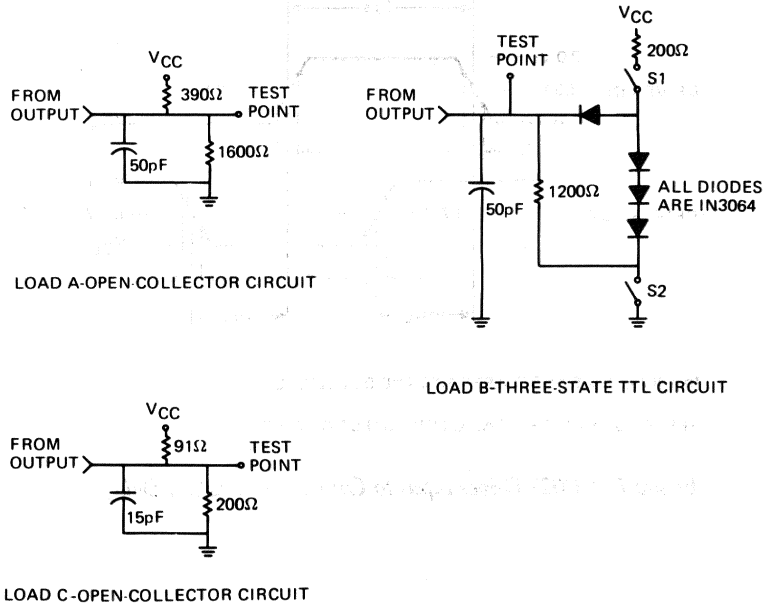


Figure 8 ▪ DC021 Output Load Circuits

Mechanical Configuration

The physical dimensions of the DC021 20-pin DIP are shown in Appendix E.

• Section 5—Mass-storage devices

The mass-storage devices are used in the development of disk and tape drive interfaces.

DC018 Serializer/Deserializer Logic—The DC018 is a 40-pin DIP bipolar device that converts serial data from a drive into parallel data and parallel data into serial data.

DC024 Encoder/Decoder Logic—The DC024 is a 28-pin DIP device that encodes and decodes the information between the head electronics of a mass-storage device and the logic of the DC016 serializer/deserializer.

DC309 Reed Solomon Generator for ECC—The DC309 is a 28-pin DIP dynamic NMOS device that implements a Reed Solomon error correction code (ECC).

• Features

- Converts serial data into parallel data and parallel data into serial data.
- Correlation logic provides autostart capabilities.
- Two speed variations available.
- Selectable word length of 8, 10, or 16 bits.

• Description

The DC018 serializer/deserializer, contained in a 40-pin dual-inline package (DIP), is a bipolar LSI device developed for use in disk-drive controllers. It provides the circuits and logic required to convert serial data into parallel data and parallel data into serial data. The DC018 performs the conversion at a minimum serial-bit time of 36 nanoseconds. Figure 1 is a simplified block diagram of DC018.

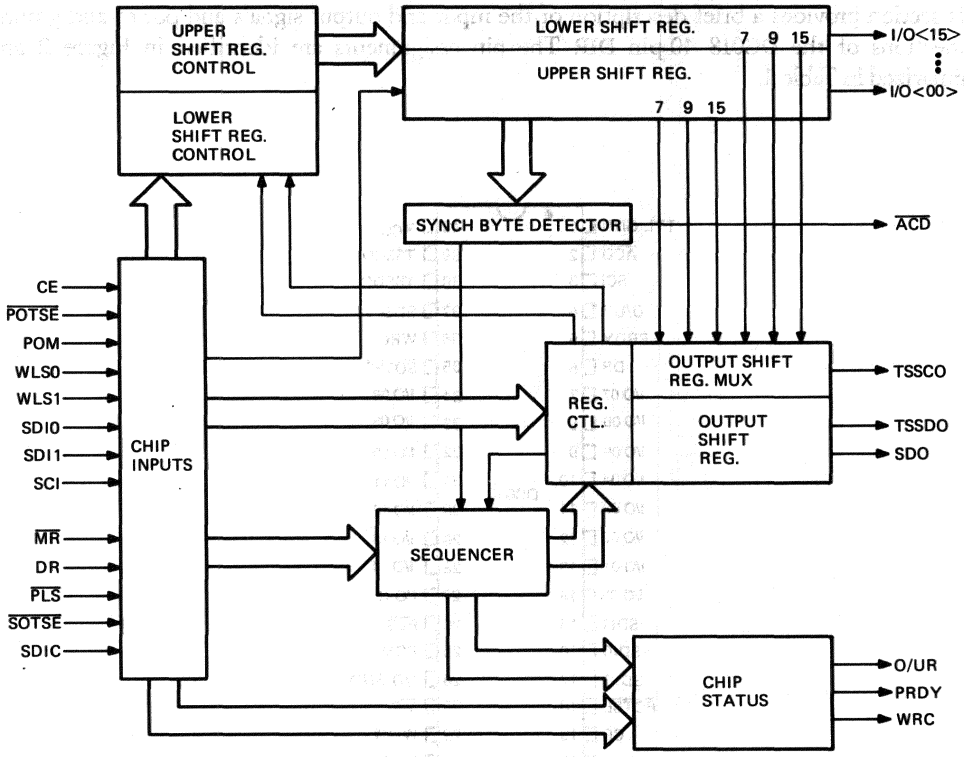


Figure 1 • DC018 Simplified Block Diagram

The DC018 is available in the following variations depending on the clock speed required.

Clock Frequency	Digital Part No.
14.3 Megahertz (maximum)	19-17043-00
27.7 Megahertz (maximum)	19-17043-01

The DC018 contains input logic that selects the serial data input line and parallel data word length, and controls the conversion process. An upper and lower 16-bit shift register and associated logic control the parallel I/O data transfers to and from the chip. The parallel data is available from three-state drivers circuits. The serial data outputs are available from the output shift register. One output provides TTL levels and is used for serially cascading two DC018 chips. The remaining output provides three-state driver output levels that are externally controlled for the parallel port and serial port operation. An autostart feature allows the serial-to-parallel conversion process to begin upon recognition of a serial-input bit pattern.

Autocorrelation logic detects 9 serial-input bits out of 12 bits at clock frequencies up to 14.3 Megahertz. At clock frequencies from 14.3 to 27.7 Megahertz, the autocorrelation logic detects 10 serial-input bits out of 12 bits. During the shifting operations, the logic searches for a maximum of 5 bits that are in agreement. When the correlation is detected, the conversion process occurs in any one of the selectable word lengths of 8 bits, 10 bits, or 12 bits.

• Pin and Signal Description

This section provides a brief description of the input and output signals and power and ground connections of the DC018 40-pin DIP. The pin assignments are identified in Figure 2 and summarized in Table 1.

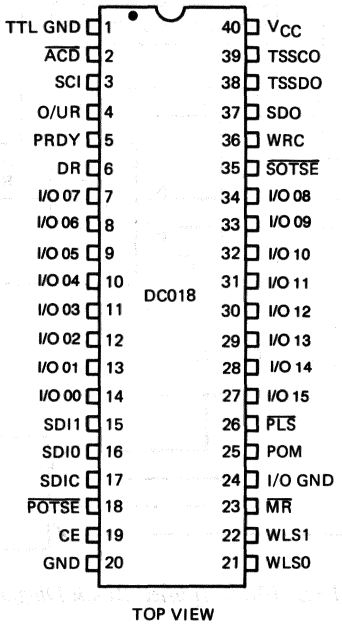


Figure 2 • DC018 Pin Assignments

Table 1 • DC018 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
1	TTL GND	input	TTL ground—A TTL ground reference. Pin 1 connects to pin 20 in the shortest possible path.
2	$\overline{ACD}$	output[1]	Autocorrelator detect—Asserted by the sync byte detector when a sync byte is present. Used for test purposes only.
3	SCI	input[1]	Serial clock—Data is shifted into the shift register on the positive edge of this signal.
4	O/UR	output[1]	Overrun/Underrun—Indicates that data has been missed during a transfer on the parallel port, as follows: Cleared by the master reset pulse ($\overline{MR}$). 1. Parallel in mode—Asserted to indicate an overrun condition in the parallel out mode when the POM signal and parallel ready (PRDY) signal are asserted and the data ready (DR) signal is not asserted within the word time. 2. Parallel out mode—Asserted to indicate an underrun condition when POM signal is negated and PRDY signal is asserted, if the parallel input strobe ($\overline{PLS}$) is not asserted within the word time.
5	PRDY	output[1]	Parallel ready—Indicates that the DC018 is ready to transfer or receive data as follows: 1. Asserted in the parallel out mode (POH asserted) when data is available on pins I/O<15:00> and if the parallel out three-state enable ($\overline{POTSE}$) signal is asserted. It is negated by the assertion of data received (DR) if $\overline{POTSE}$ is asserted. 2. Asserted in the parallel in mode (POM negated) when the DC018 is ready to receive parallel data and the master reset (MR) is negated. It is negated by the negative edge of the parallel input strobe ($\overline{PLS}$). 3. If conditions 1 and 2 do not occur, PRDY will be negated before the parallel data becomes invalid and an overrun or underrun condition (O/UR) will be indicated.
6	DR	input[1]	Data received—Asserted by the user to indicate that the parallel data on pins I/O<15:00> has been received. In the parallel out mode (POM asserted) when the data is available and the POTSE signal is asserted, the DC018 asserts the PRDY line. After the data is received, the assertion edge of the DR signal negates the PRDY signal.

Pin	Signal	Input/Output	Definition/Function
7-14 27-34	I/O<15:00>	inputs[1]/output[2]	Parallel data input/output—Sixteen parallel data lines that provide controllable three-state level outputs and include TTL level gates for the parallel input signals. The POM input selects the function of the lines.
15	SDI1	input[1]	Serial data in 1—Serial data input channel 1 selected when the serial data input control (SDIC) signal is high.
16	SDI0	input[1]	Serial data in 0—Serial data input channel 0 selected when the the SDIC signal is low.
17	SDIC	input[1]	Serial data input control—When high, this input selects serial data input channel 1 (SDI0) and when low it selects serial data input channel 0 (SDI0).
18	$\overline{\text{POTSE}}$	input[1]	Parallel output three-state enable—Asserted to enable and negated to disable the parallel three-state outputs I/O<15:00>).
19	CE	input[1]	Counter enable—In the parallel-in, serial-out mode, this line is asserted to allow the next positive edge of the clock input (SCI) to enable the internal bit-rate counter. In the serial-in, parallel-out mode, this line is asserted to permit the detection of a sync pattern of the serial data to enable the bit-rate counter.
20	GND	input	Ground—A ground reference. Pin 20 connects to pin 1 in the shortest connection possible.
21	WLSO	input[1]	Word length select 0—Used with word length select 1 to select the length of the parallel data output word, as follows: **WLS0** **WLS1** **Word length** H H 16 bits H L 10 bits L H 8 bits
22	WLS1	input[1]	Word length select 1—Refer to word length select 0 description.
23	$\overline{\text{MR}}$	input[1]	Master reset—Asserted to reset all internal registers and counters in the DC018.
24	I/O GND	input	Input/Output ground—An isolated ground reference for parallel outputs I/O<15:00>.
25	POM	input[1]	Parallel out mode—Asserted to select the serial-in, parallel-out mode and negated to select the parallel-in, serial-out mode.

Pin	Signal	Input/Output	Definition/Function
26	$\overline{PLS}$	input[1]	Parallel strobe—In the parallel-in, serial-out mode, this signal is asserted after the parallel ready (PRDY) is asserted, to load the parallel data on lines I/O<15:00> into the parallel registers of the DC018. The PRDY signal negated when this signal becomes low before the next cycle can occur.
35	$\overline{SOTSE}$	input[1]	Serial-out three-state enable—Asserted to enable and negated to disable the outputs of three-state serial data out (TSSDO) and serial clock out (TSSCO) lines.
36	WRC	output[1]	Word rate clock—A clock output with a 50 percent duty cycle that is decoded from the modulo *n* (*n*=8 bit, 10 bit, or 16 bit) counter with a clock period of *n* bit times. The WRC signal is asserted with a count of zero and negated at *n*/2 bit time.
37	SDO	output	Serial data output—Used to cascade the serial data from two DC018 chips operating with the same serial clock input (SCI). It exhibits the same delay characteristics as the serial data input (SDI0 and SDI1) and the I/O<15:00> outputs as described for the TSSDO signal. The minimum propagation delay from when the serial clock SCI signal input becomes a high level until a change in the SDO level, is greater than the minimum hold time on the SDI0 or SDI1 inputs with respect to the SCI input.
38	TSSDO	output[1]	Three-state serial data out—Controlled by the $\overline{SOTSE}$ signal. Data transitions on this line are synchronized with the positive edges of the TSSCO signal.
39	TSSCO	output[2]	Three-state serial clock out—A serial clock output that is a delayed serial clock input SCI signal. The delay conforms to the propagation time incurred by the TSSDO output; therefore, the positive-going edge of the clock pulse is synchronized with TSSDO output bits. The clock output is controlled be the $\overline{SOTSE}$ signal.
40	V_{CC}	input	Voltage—Power supply dc voltage

[1]TTL levels.
[2]Three-state.

• Specifications

The mechanical, electrical, and environmental characteristics and specifications of the DC018 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise. Refer to Digital specification A-PS-2100002-GS for detailed specifications of the DC018.

- Operating temperature (T_A): 0°C to 70°C
- Supply voltage (V_{CC}): 5.0 V ±5%

Mechanical Configuration

The physical dimensions of the DC018 40-pin DIP are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Supply voltage (V_{CC}): 7.0 V
- Input voltage (V_I): 5.5 V
- Operating temperature (T_A): 0°C to 70°C
- Storage temperature (T_S): –65°C to 125°C

Recommended Operating Conditions

- Supply voltage (V_{CC}): 5.0 V ±5%
- Supply current (I_{CC}): 425 mA (maximum)
- Free-air temperature (T_A): 0°C to 70°C
- Relative humidity: 10% to 95% (noncondensing)

dc Electrical Characteristics

The dc electrical parameters of the DC018 for the operating voltage and temperature ranges specified are listed in Table 2. Refer to Appendix C for the test circuit configurations referenced in the table.

Table 2 • DC018 dc Input and Output Parameters

Parameter	Symbol	Test Condition	Requirements Min.	Max.	Units	Test Circuit
High-level input voltage	V_{IH}	Table 4	2.0	—	V	C1
Low-level input voltage	V_{IL}	Table 4	—	0.8	V	C1
Input clamp voltage	V_I	V_{CC}=open I_I=−12 ma	−1.5	—	V	C3
High-level output voltage	V_{OH}	V_{CC}=4.7 V I_O[1] Table 4	2.4	—	V	C1
Low-level output voltage	V_{OL}	V_{CC}=4.75 V I_O[2] Table 4	—	0.5	V	C2
Input current at maximum input voltage	I_I	V_{CC}=5.25 V V_I 5.5 V	—	1.0	mA	C4
High-level input current	I_{IH}	V_{CC}=5.25 V V_I=2.4	—	[3]	μA	C4
Low-level input current	I_{IL}	V_{CC}=5.25 V_I=0.4	—	[4]	mA	C5
Short-circuit output current	I_{OS}	V_{CC}=5.25 V[5]	−100[6]	−5.0[6]	mA	C6
High-impedence state output current	I_{OZL}	V_{CC}=5.25 V V_O=0.4 V Table 4	—	[7]	μA	C8
	I_{OZH}	V_{CC}=5.25 V V_O=2.4 V Table 4	—	[7]	μA	C9
Supply current	I_{CC}	V_{CC}=5.25 V	—	425	mA	C7
Input capacitance	C_{in}		—	8.0	pF	

[1] I_O=−1 mA for SDO, −6.5 mA for TSCCO and TSSDO, −400 mA for all other outputs.

[2] I_O=1 mA for ACD, 20 mA for TSSCO, TSSDO, and SDO, 5 mA for all other outputs.

[3] I_{IH}=20 μA for SDI0, SDI1, $\overline{MR}$, and $\overline{SOTSE}$ 40 μA for SCI, SDIC, CE, POM, DR, $\overline{PLS}$, WLS0, and WLS1, 80 μA for $\overline{POTSE}$, −140 μA for I<15:00>.

[4] I_{IL}=−400 μA for SDI0, SCI1, SOTSE, I<15:00>, and DR, −800 μA for SCI, CE, POM, $\overline{PLS}$, WLS0, WLS1, $\overline{MR}$, and $\overline{POTSE}$, −1.0 mA for SDIC.

[5] Not more than one output shall be short-circuited at a time and the duration of the short shall not exceed 1 second.

[6](I$_{OS}$ for WRC) Failure to perform this test where indicated on Table 3 will cause the WRC output to change states. I$_{OS}$ for $\overline{ACD}$ = –20 mA (minimum) and –50 mA (maximum).

[7]I$_{OZL}$ = 50 μA, I$_{OZH}$ = –50 μA for TSSCO and TSSDO, I$_{OZL}$ = I$_{IL}$, I$_{OZH}$ = I$_{L}$ for I < 15:00 >.

ac Electrical Characteristics

The input and output signal timing for the DC018 is shown in Figures 3 through 10. The propagation delays and conditions for the symbols on the timing diagrams are described in Tables 3 through 8. Refer to Appendix D for the input and output voltage waveform parameters used for measuring the signal propagation delays. Figure 11 shows the load circuit for the TTL outputs and the load circuit and waveforms for the three-state outputs.

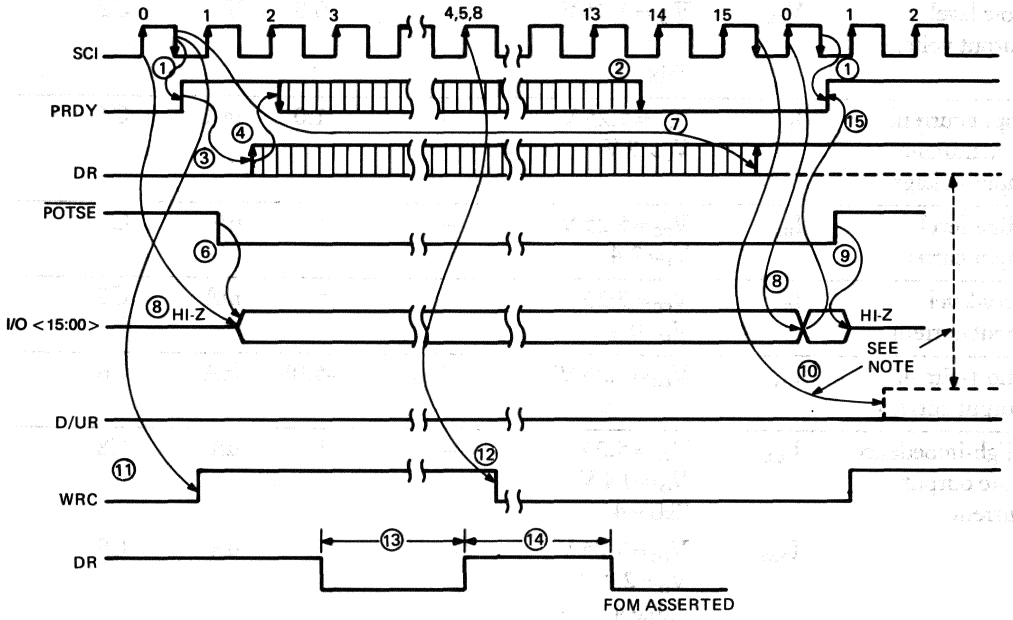


Figure 3 • DC018 Signal Timing A

Table 3 • DC018 Signal Timing A Parameters

Symbol	Conditions	Propagation Delay (ns) $C_L = 15$ pF Min.	Max.	$C_L = 50$ pF Min.	Max.
1		—	—	—	55
2		—	—	—	30
3		0	—	0	—
4		—	—	—	30
6		—	—	—	36
7	DR can be asserted up to $(m-1)(t_{CP})-t_{setup}$ (nanoseconds) after the negative-going edge of SCI where; m = 8, 10, or 16 bits, t_{CP} = input clock period, t_{setup} = 18 ns (as specified)	18	—	18	—
8		—	—	—	65
9	See Figure 11 t_{HZ}				18
	t_{LZ}	—	—	—	25
10	If DR is not asserted during the previous word cycle, O/UR will be asserted by the DC018.	—	—	—	70
11		—	—	—	50
12	WRC will be negated by DC018 in the middle of the 8 (4th bit), 10 (5th bit), or 16 (8th bit) word cycle.	—	45	—	50
13	Pulse width low	40	—	40	—
14	Pulse width high	40	—	40	—
15	Valid data will be present on I/O<15:00> before PRDY is asserted.	—	—	5.0	—

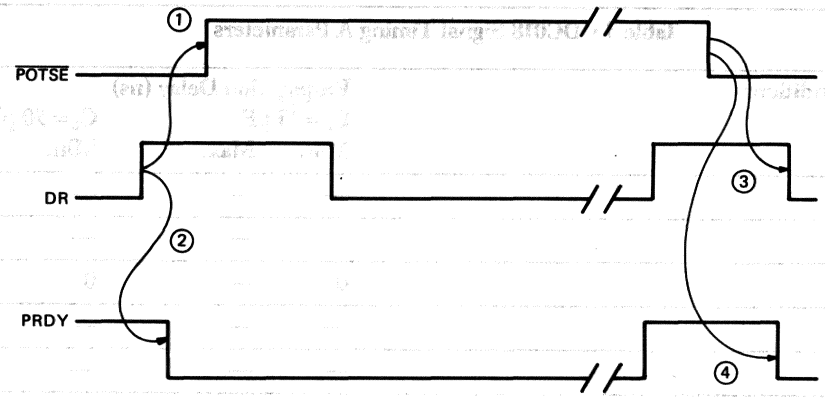


Figure 4 ▪ DC018 Signal Timing B

Table 4 ▪ DC018 Signal Timing B Parameters

Symbol	Conditions	Propagation Delay (ns) $C_L = 15$ pF Min.	Max.	$C_L = 50$ pF Min.	Max.
1	Setup time	25	—	25	—
2		—	—	—	30
3	Setup time	40	—	40	—
4		—	—	—	45

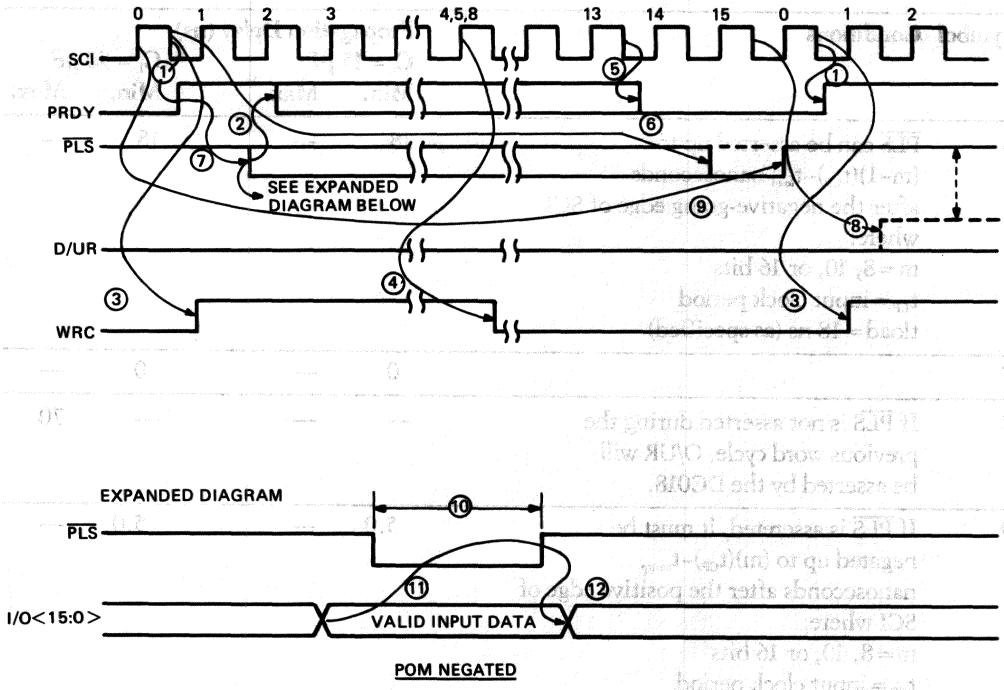


Figure 5 • DC018 Signal Timing C

Table 5 • DC018 Signal Timing C Parameters

Symbol	Conditions	Propagation Delay (ns) $C_L = 15$ pF Min.	Max.	$C_L = 50$ pF Min.	Max.
1		—	—	—	55
2		—	—	—	35
3		—	—	—	50
4	WRC will be negated by DC018 in the middle of the 4th bit, 5th bit, or 8th bit word cycle depending on the 8, 10, or 16 bit (respectively) word length selected.	—	45	—	50
5		—	—	—	30

Symbol	Conditions	Propagation Delay (ns) $C_L = 15$ pF Min.	Max.	$C_L = 50$ pF Min.	Max.
6	$\overline{PLS}$ can be asserted up to $(m-1)(t_{CP})-t_{load}$ nanoseconds after the negative-going edge of SCI where: m = 8, 10, or 16 bits t_{CP} = input clock period tload = 18 ns (as specified)	18	—	18	—
7		0	—	0	—
8	If $\overline{PLS}$ is not asserted during the previous word cycle, O/UR will be asserted by the DC018.	—	—	—	70
9	If $\overline{PLS}$ is assereted, it must be negated up to $(m)(t_{CP})-t_{setup}$ nanoseconds after the positive edge of SCI where: m = 8, 10, or 16 bits t_{CP} = input clock period tsetup = 5 ns (as specified)	5.0	—	5.0	—
10	Pulse-width low	25	—	25	—
11	Setup time	40	—	40	—
12	Hold time	5.0	—	5.0	—

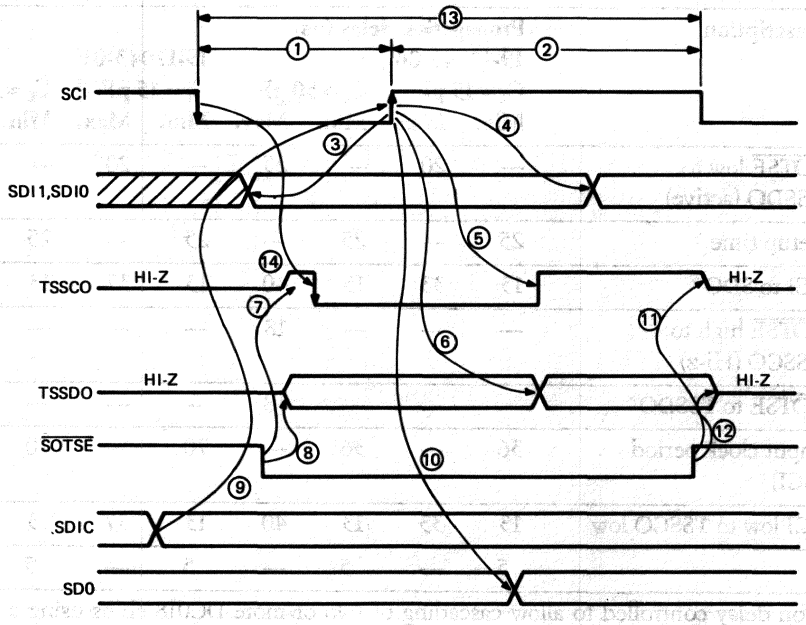


Figure 6 • DC018 Signal Timing D

Table 6 • DC018 Signal Timing D Parameters

Symbol	Description	Propagation delay (ns) 19-17043-00 $C_L = 15$ pF Min.	Max.	$C_L = 50$ pF Min.	Max.	19-17043-01 $C_L = 15$ pF Min.	Max.	$C_L = 50$ pF Min.	Max.
1	SCIH pulse-width low	15	—	15	—	30	—	30	—
2	SCIH pulse-width high	17	—	17	—	30	—	30	—
3	SDI0 or SDI1 setup time	5	—	5	—	5	—	5	—
4	SDI0 or SDI1 hold time	12	—	12	—	12	—	12	—
5	SCI high to TSSCO high	13	33	13	40	13	37	13	45
5a		—	8	—	8	—	10		10
6		—	8	—	8	—	10	—	10
7	$\overline{\text{SOTSE}}$ low to TSSCO (active)	—	20	—	25	—	20	—	25

Symbol	Description	Propagation delay (ns) 19-17043-00 $C_L = 15$ pF Min.	Max.	$C_L = 50$ pF Min.	Max.	19-17043-01 $C_L = 15$ pF Min.	Max.	$C_L = 50$ pF Min.	Max.
8	$\overline{\text{SOTSE}}$ low to TSSDO (active)	—	20	—	25	—	20	—	25
9	Setup time	25	—	25	—	25	—	25	—
10	SCI to SDO*	13	33	13	40	13	37	13	45
11	$\overline{\text{SOTSE}}$ high to TSSCO (Hi-z)	—	—	—	18	—	—	—	18
12	$\overline{\text{SOTSE}}$ to TSSDO	—	—	—	18	—	—	—	18
13	Input clock period (SCI)	36	—	36	—	70	—	70	—
14	SCI low to TSSCO low	13	33	13	40	13	37	13	45
14a		5	—	5	—	5	—	5	—

*Propagation delay controlled to allow cascading of two or more DC018 chips using a common clock.

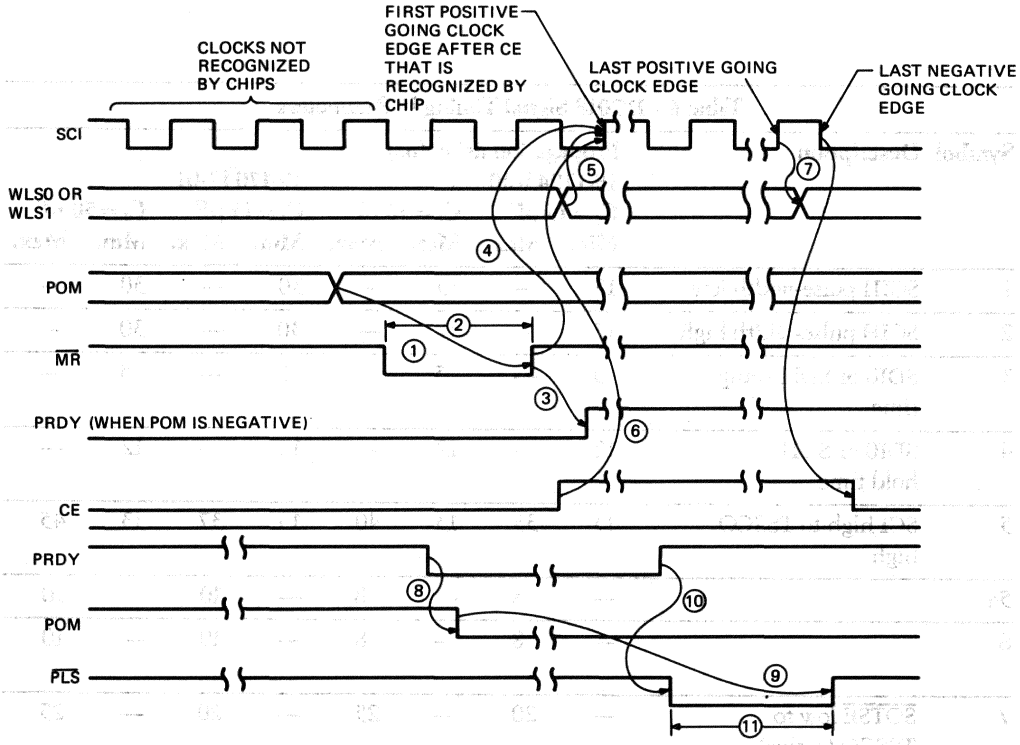


Figure 7 • DC018 Signal Timing E

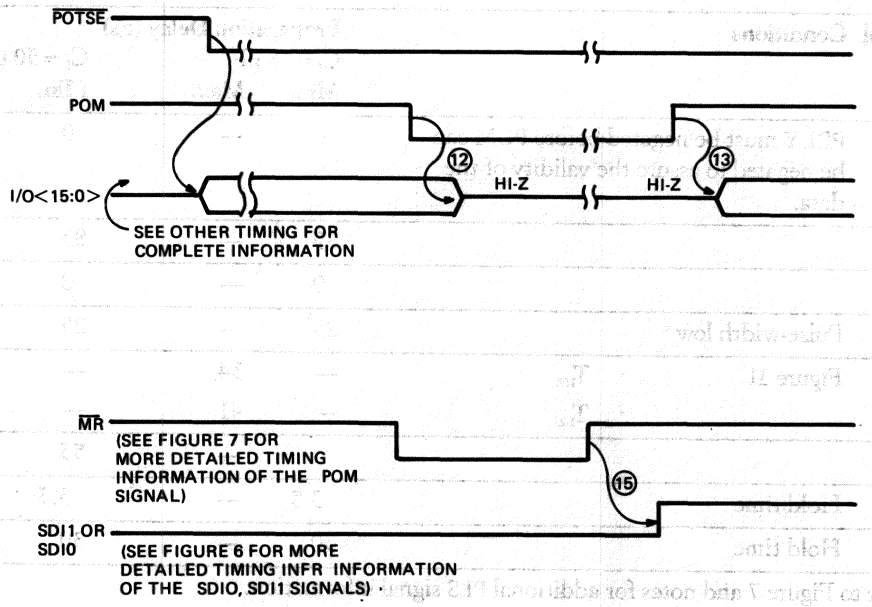


Figure 8 ▪ DC018 Signal Timing E

Table 7 ▪ DC018 Signal Timing E Parameters

Symbol	Conditions	Propagation Delay (ns) $C_L = 15$ pF Min.	Max.	$C_L = 50$ pF Min.	Max.
1	For initialization	60	—	60	—
2	Pulse-width low	80	—	80	—
3	If POM is negated, PRDY becomes high after MRL is negated. If POM is asserted, PRDY remains negated until the sync byte is detected.	—	—	—	50
4		75	—	75	—
5	Setup time	50	—	50	—
6	Setup time	25	—	25	—
7	Hold time	15	—	15	—

Symbol	Conditions		Propagation Delay (ns) $C_L = 15$ pF Min.	Max.	$C_L = 50$ pF Min.	Max.
8	PRDY must be negated before POM can be negated to assure the validity of the data.		0	—	0	—
9			85	—	85	—
10			0	—	0	—
11	Pulse-width low*		25	—	25	—
12	Figure 11	T_{HZ}	—	34	—	40
		T_{LZ}	—	41	—	50
13			—	—	55	—
14	Hold time		3.5	—	3.5	—
15	Hold time		30	—	30	—

*Refer to Figure 7 and notes for additional PLS signal information.

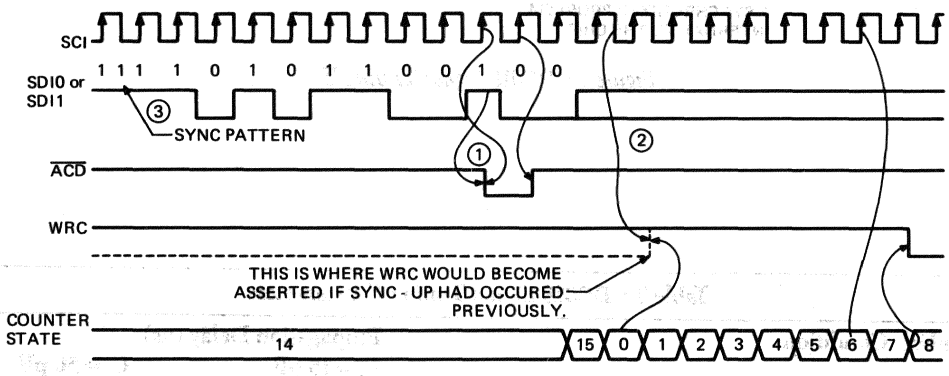


Figure 9 ▪ DC018 Signal Timing F

Table 8 • DC018 Signal Timing F Parameters

Symbol	Conditions	Propagation Delay (ns) $C_L = 15$ pF Min.	Max.	$C_L = 50$ pF Min.	Max.
1	Signal $\overline{ACD}$ will be asserted after the 12th bit of the sync byte is shifted in.	—	—	—	70
2		—	—	—	70
3	The sync byte detector will trigger when a minimum of 9, 10, 11, or 12 bits are in agreement or when 10, 11, or 12 are in agreement for the specified clock rate.	9	—	—	10

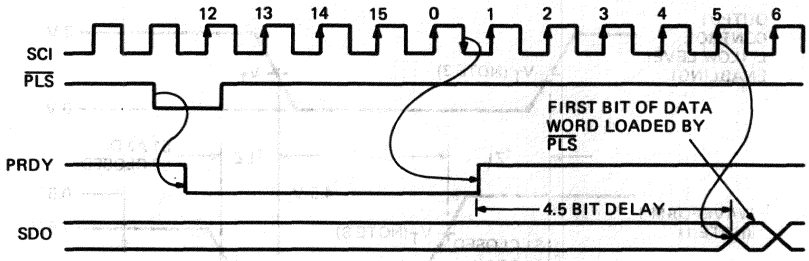
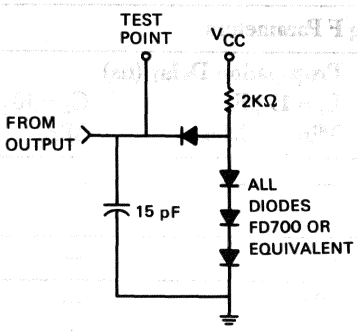
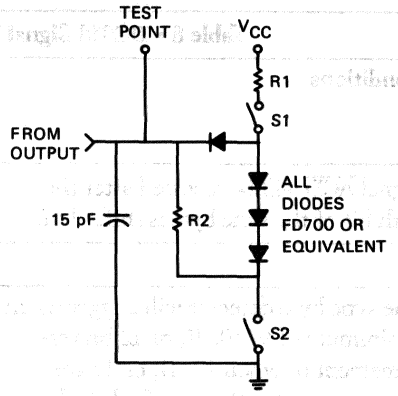


Figure 10 • DC018 Signal Timing G



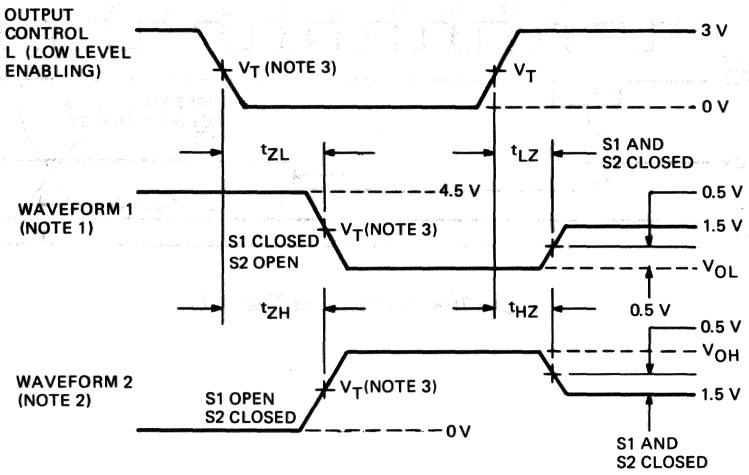
LOAD A TTL OUTPUTS



R1=2KΩ, R2=1KΩ FOR PINS 7-14, 27-34

R1=260Ω, R2=1KΩ FOR PIN PINS 38, 39.

LOAD B THREE-STATE OUTPUT



NOTES:

1 WAVEFORM 1 IS FOR AN OUTPUT WITH INTERNAL CONDITIONS SUCH THAT THE OUTPUT IS LOW EXCEPT WHEN DISABLED BY THE OUTPUT CONTROL

2 WAVEFORM 2 IS FOR AN OUTPUT WITH INTERNAL CONDITIONS SUCH THAT THE OUTPUT IS HIGH EXCEPT WHEN DISABLED BY THE OUTPUT CONTROL

3 V_T=1.3 V FOR ALL OUTPUT PINS EXCEPT 38 AND 39

V_T= 1.5 V FOR PINS 38 AND 39

THREE-STATE OUTPUT WAVEFORMS

Figure 11 • DC018 Output Load Circuits and Waveforms

▪ Features

- ⅔-rate encoder/decoder algoithm for mass-storage devices
- Compatible with the DC018 Serializer/Deserializer
- TTL and ECL inputs and outputs
- Phase detector and counter/divider
- Diagnostic mode operation

▪ Description

The DC024 encoder/decoder, contained in a 28-pin dual-inline package (DIP), provides the functions required to encode and decode the data between the head electronics of a mass-storage device and the logic of the DC018 serializer/deserializer. In addition to the encoder and decoder logic, the DC024 contains phase detectors, dividers, and control logic for a phased-locked loop. A test mode is provided to allow the DC024 and DC018 Serializer/Deserializer to be operated in an end-around configuration for diagnostic purposes. Figure 1 is a simplified block diagram of DC024.

The DC024 performs the ⅔-rate encoding/decoding algorithm for mass storage products. It operates with TTL and ECL level inputs and outputs and requires a 5 V and –5.2 V power supply.

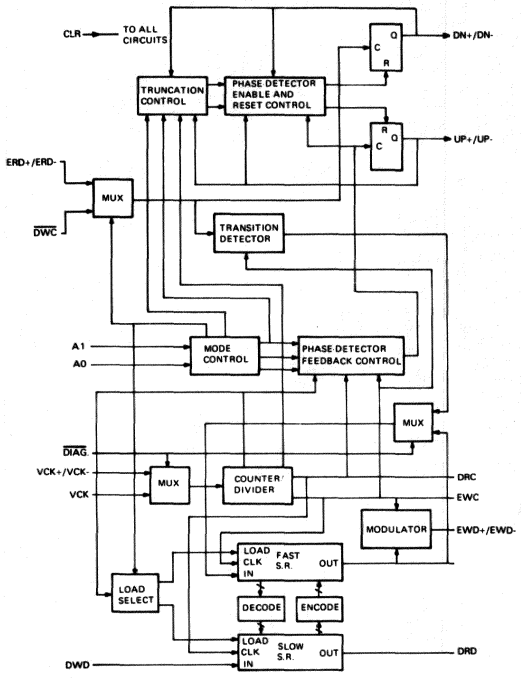


Figure 1 ▪ DC024 Simplified Block Diagram

• Pin and Signal Description

This section provides a brief description of the input and output signals and power and ground connections of the DC024 28-pin DIP. The pin assignments are identified in Figure 2 and summarized in Table 1.

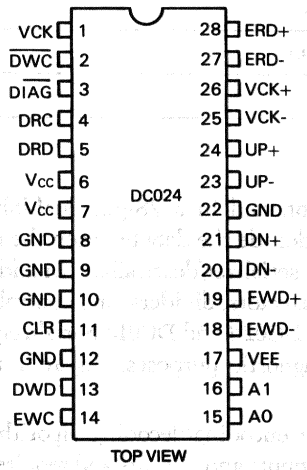


Figure 2 • DC024 Pin Assignments

Table 1 • DC024 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
28 27	ERD+ ERD–	inputs[1]	Encoded read data—A signal from the read circuits. Each flux reversal to the read circuits results in a pulse input on this line. During mode C, the asserting edge of the pulse enables the phase detector of the DC024 to perform a phase comparison of the deasserting edge. During modes A and B, the phase detector is internally enabled and a phase comparison is performed on the deasserting edge of these pulses.
13	DWD	input[2]	Decoded write data—This signal is an input from the deserializer of the system and is used during write operations. The data is in NRZ format and will be encoded by the DC024 before being written on the media.
2	$\overline{\text{DWC}}$	input[2]	Decoded write clock—An input with a clock frequency of one cycle for each decoded write data bit. When writing, this signal is one of the inputs to the phase detector. An internal loop signal of the same frequency and phase when locked clocks the decoded write data into the chip. This input must be connected directly to the write frequency source or indirectly through the deserializer. It cannot be used directly to clock the decoded write data into the DC024.
15 16	A0 A1	inputs[2]	Mode select—Determines the mode of operation as follows: A0 A1 Mode 0 0 A (preamble) 1 0 B (preamble-truncate) 1 1 C (decode) 0 1 D (encode)
11	CLEAR	input[2]	Clear—When asserted, all DC024 storage elements are cleared and the logic is set to a known state.
26 25	VCK+ VCK–	inputs[1]	Voltage control clock—The differential input from an external voltage controlled oscillator. Provides a frequency of two cycles for an encoded data bit (three cycles per decoded data bit).
1	VCK	input[2]	Voltage clock—Used during diagnostic mode as the phase-locked loop feedback clock.
3	$\overline{\text{DIAG}}$	input[2]	Diagnostic mode—When asserted, the DC024 enters the diagnostic mode. The VCK clock signal is used as a substitute for the VCK+/VCK– clock input as the feedback clock for the phase-locked loop.
24 23	UP+ UP–	outputs[1]	Up frequency—The phase detector output signals that causes the voltage controlled oscillator to increase its frequency. The duration of the assertion of this signal is proportional to the magnitude of the phase error.

Pin	Signal	Input/Output	Definition/Function
21 20	DN+ DN–	outputs[1]	Down frequency—The phase detector output that causes the voltage controlled oscillator to decrease its frequency. The duration of the asserton of this signal is proportional to the magnitude of the phase error.
14	EWC	output[2]	Encoded write clock—A clock output from the phase-locked loop circuit with one cycle for each encoded write data bit.
19 18	EWD+ EWD–	outputs[1]	Encoded write data—A combined output of the NRZ encoded write data and the encoded write clock. Every "one" input of the NRZ encoded write data signal causes the outputs to change states.
4	DRC	output[2]	Decoded read clock—This signal is a clock output that is phase locked to the external input of phase-locked loop (ERD+/ERD– when decoding or $\overline{DWC}$ when encoding) at a frequency of one cycle for a decoded data bit. Used to clock the decoded read data from the DC024 during read operations and as a clock source into a deserializer during write operations.
5	DRD	output[2]	Decoded read data—This signal is an NRZ serial output of the DC024 that is decoded from the serial input to the when reading encoded 2/3 rate) data from the peripheral. This signal is normally used as an input to a deserializer. While writing to the media, this signal is not disabled and shifts the decoded data (DWD) out delayed by 10.5 VCK+/VCK– cycles.
6,7	V_{CC}	input	Voltage—Power supply 5 Vdc.
8,9	GND	input	Ground—Ground reference for 5 Vdc.
17	V_{EE}	input	Voltage—Power supply –5.2 V.
1-,12, 22	GND	input	Ground—Ground reference for the –2.5 V.

[1]ECL levels
[2]TTL levels
[3]The multiple power and ground pins of the DC024 must be properly connected to assure proper operation.

• Operation

The DC024 encoder/decoder consists of three main logic sections—a phase detector, a counter/divider, and an encoder/decoder. It implements the ⅔-rate modulation code shown in Table 2. In the bit sequences, the bits at the left are the first in time.

Note: Signals referenced in the descriptions that are enclosed in brackets [ ] are internal signals only.

Table 2 • DC024 2/3-Rate Modulation Code

Data Bits	Encoded sequence*
00	X00
01	X01
10	010
1100	X00000
1101	X00001
1110	010000
1111	010001

*X = The complement of the last bit of the previously encoded data.

Phase Detector

The phase detector operates in one of the four modes selected by the A0 and A1 inputs. These inputs are decoded by the counter/divider logic to select the modes listed in Table 3.

Table 3 • DC024 Modes of Operation

Mode	Mode control A1	A0	Phase detector*	Input	Operation Feedback
A	0	0	ERD+/ERD–	[VCK/6]	Preamble
B	0	1	ERD+/ERD–	[VCK/6]	Preamble-Truncate
C	1	1	ERD+/ERD–	[VCK/2]	Read/decode
D	1	0	$\overline{DWC}$	[VCK/3]	Write/encode

*Signals enclosed in brackets [ ] are internal to the DC024.

Mode A—This mode is a positive locking mode. The phase detector, shown in Figure 3, consists of flip-flops E108 and E107. Flip-flop E108 is set by the rising edge of the phase-locked-loop output divided by 6. Flip-flop E107 is set by the trailing edge of the phased-locked-loop input [ERDL]. Both flip-flops are cleared immediately after they are both set. The output of flip-flop E108 indicates that the VCK clock frequency is low and is used to increase the charge of a filter network. Flip-flop E107 indicates that the VCK frequency is high and is used to decrease the charge on a filter network. At the start of a read operation, a special data pattern designated as the "preamble" is decoded to establish a phase lock before the normal data is read. The decoded preamble consists of all zeros. When encoded, the bit pattern is 100, 100, 100 etc. A lower frequency clock from the counter/divider is used as feedback to the phase detector when the preamble is expected.

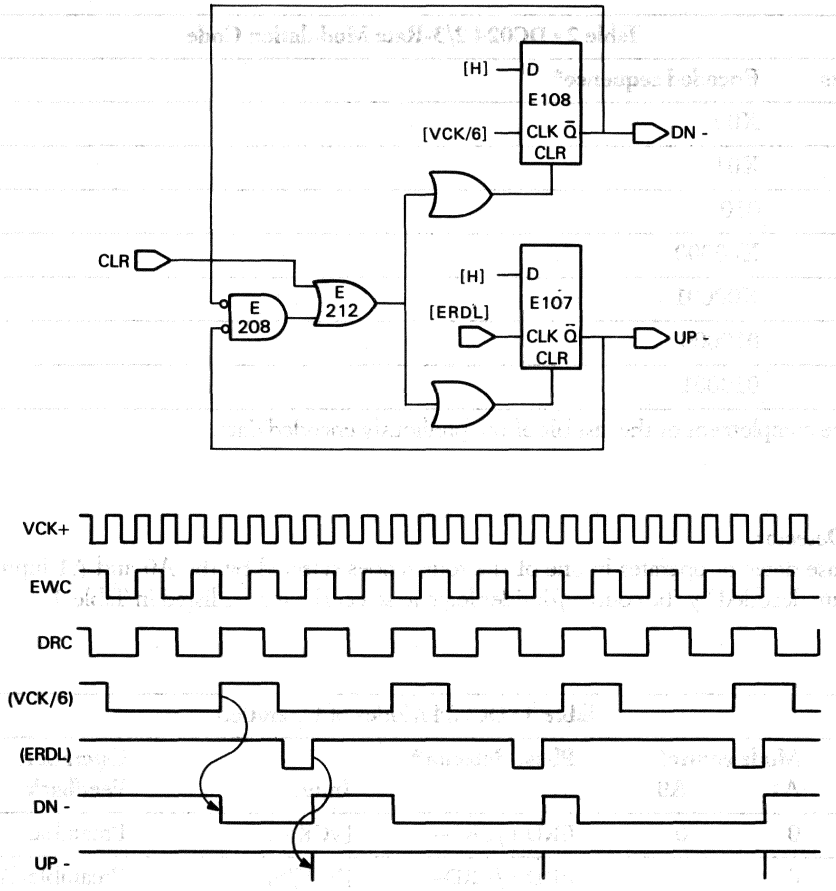


Figure 3 ▪ Mode A Phase Detector Operation

Mode B—Figure 4 shows the logic for mode B. When the phase of the loop is locked, a lost bit in the preamble data, indicating an error condition, could cause the loop to differ by one or more bits. A truncate function is provided so the phase detector can be reset by a signal from the counter/divider after one-half cycle of the feedback signal. This function should not be used until frequency lock is established as it may extend time to frequency lock.

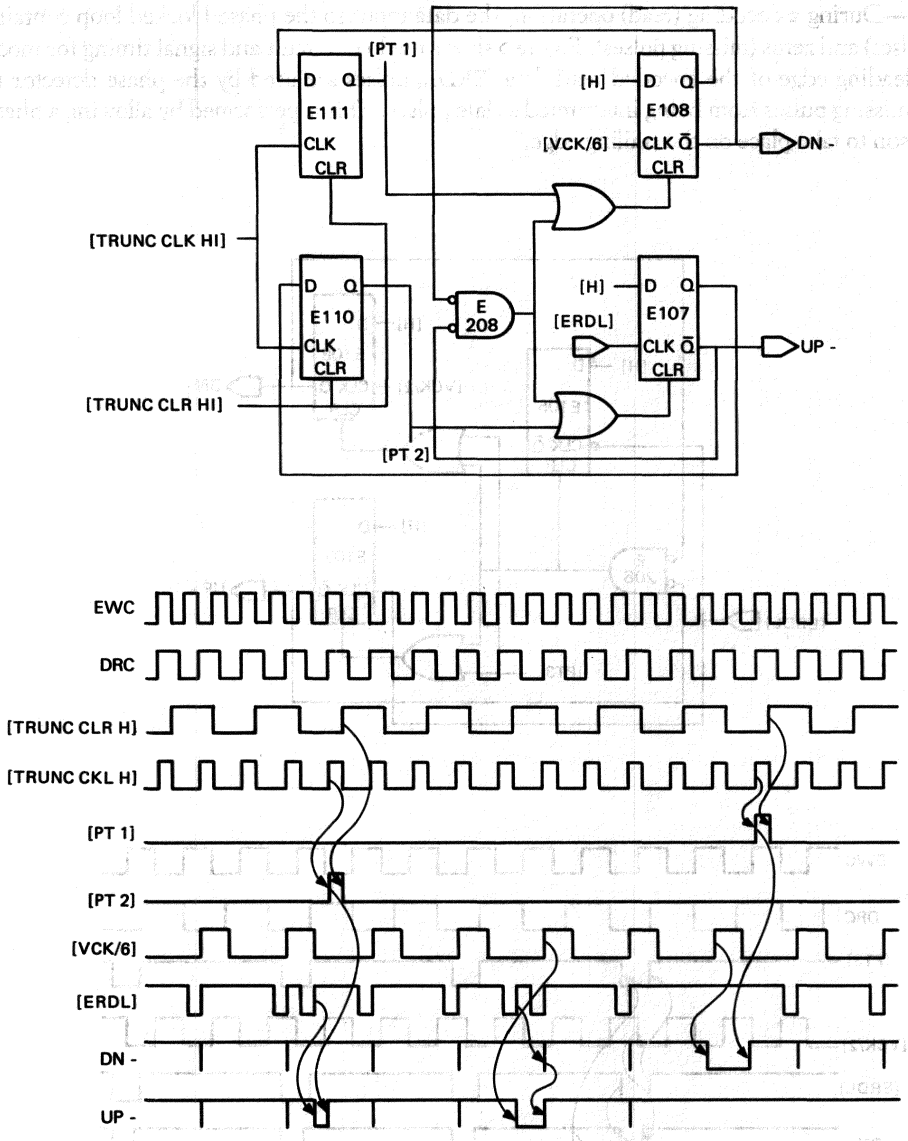


Figure 4 • Mode B Phase Detector Operation

Mode C—During a decoding (read) operation, the data input to the phased locked loop contains ones (pulses) and zeros (missing pulses). Figure 5 shows the logic circuit and signal timing for mode B. The leading edge of the encoded read data [ERDL] pulses are used by the phase detector to prevent missing pulses from being interpreted as late pulses. This is performed by allowing a phase comparison to take place on the trailing edge.

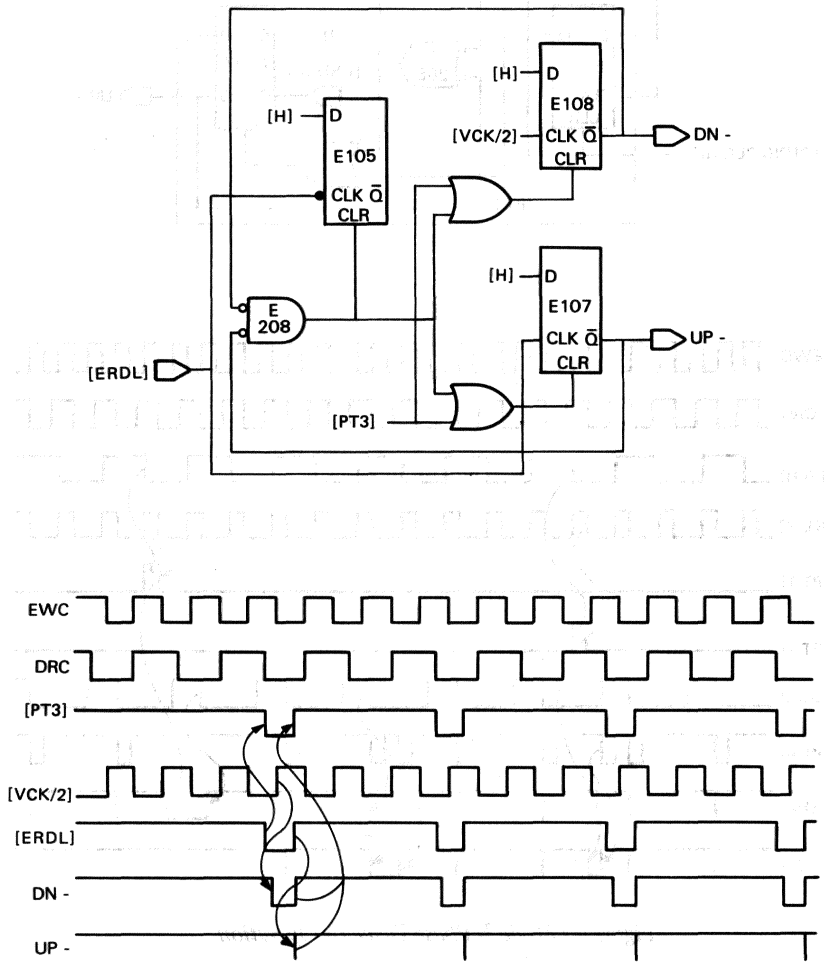
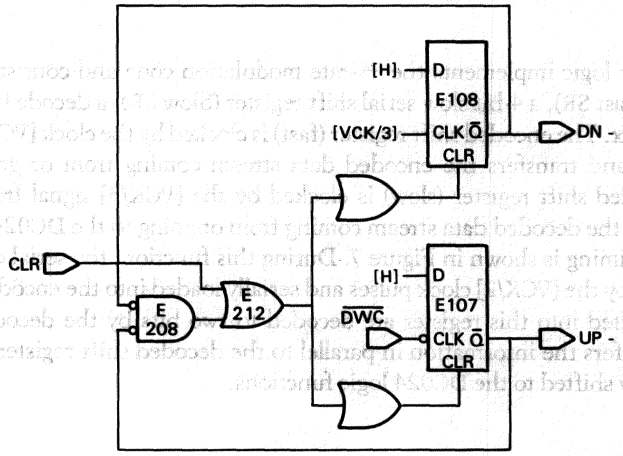


Figure 5 ▪ Mode C Phase Detector Operation

Mode D—The operation of the phase detector during the encoding (write) operation is similar to the decoding (read) operation except that the encoded read data (ERD+/ERD–) signal is replaced with the decoded write clock ($\overline{DWC}$) signal. Figure 6 shows the logic circuit and signal timing for mode D. Feedback to the phase detector is provided by a [VCK/3] signal from the counter/divider logic.



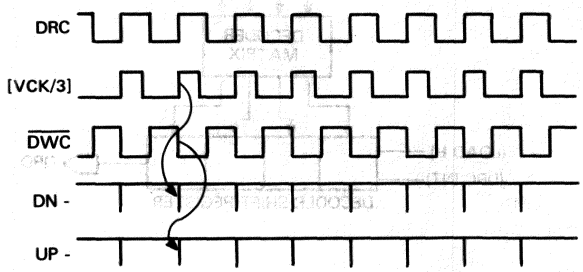


Figure 6 ▪ Mode D Phase Detector Operation

Counter/Divider

The counter/divider provides the timing and clock signals to control the operation of the DC024 and feedback to the phase detector. The logic receives a differential input (VCK+ and VCK–) from an external voltage controlled oscillator during normal modes of operation. The counter/divider provides divide-by-two [VCK/2], divide-by-three [VCK/3], and divide-by-six [VCK/6] clock pulse to the phase detector depending on the operating mode selected. During diagnostic mode, the internal VCK signal is used as the feedback clock to the phase detector and the external signals are inhibited.

Encoder/Decoder

The encoder/decoder logic implements the ⅔-rate modulation code and consists of a 4-bit fast serial shift register (Fast SR), a 4-bit slow serial shift register (Slow SR), a decode logic matrix, and an encode logic matrix. The encoded shift register (fast) is clocked by the clock [VCK/2] signal from the counter/divider and transfers the encoded data stream coming from or going to the disk interface. The decoded shift register (slow) is clocked by the [VCK/3] signal from the counter/divider and transfers the decoded data stream coming from or going to the DC024. The decoding function and signal timing is shown in Figure 7. During this function, the serial data read from a disk is synchronized by the [VCK/2] clock pulses and serially loaded into the encoded shift register. Every three bits shifted into this register are decoded to two bits by the decoder matrix. The decoder matrix transfers the information in parallel to the decoded shift register. The data from this register is serially shifted to the DC024 logic functions.

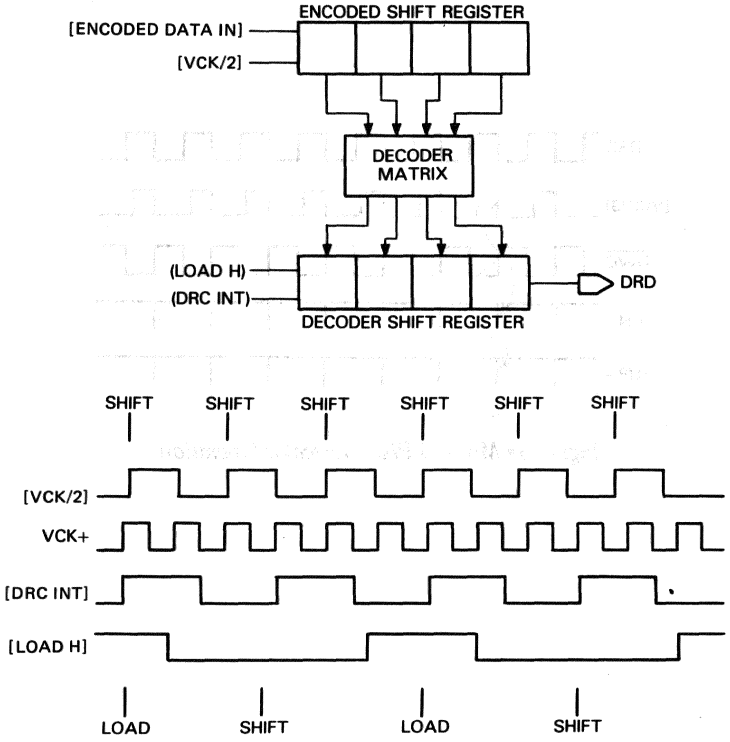
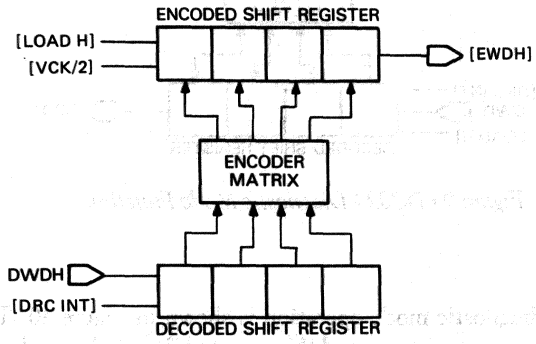


Figure 7 • DC024 Decode Function and Signal Timing

The encoding operation and signal timing are shown in Figure 8. A serial data stream is transferred serially to the decoded shift register. Every two bits shifted into this register are encoded to three bits by the encode logic matrix. This information is then transferred in parallel to the encoded shift register that serially shifts the data to the disk interface.



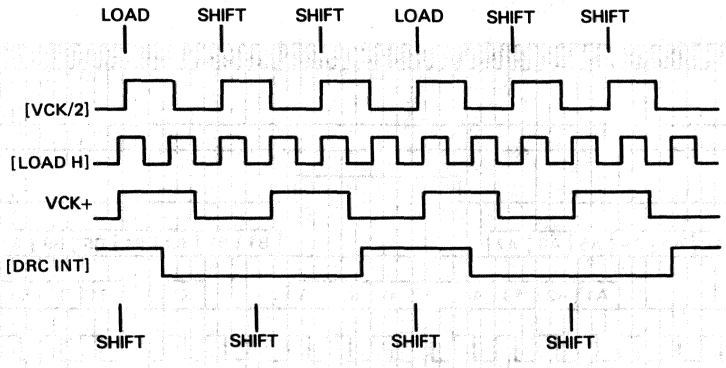


Figure 8 • DC024 Encode Function and Signal Timing

When the $\overline{\text{DIAG}}$ line is asserted, a special diagnostic function is enabled by connecting the encoded shift register in an end-around configuration as shown in Figure 9. This function may be used as an open-loop test of the DC024 by bypassing the VCO controlled phase-locked loop. By selecting the proper operating modes through A0 and A1 lines, two serial bits on the DWD input can be shifted into the decoded register, encoded through the encoded matrix, and loaded into the encoded shift register. The end-around connection reshifts the encoded write data into the encoded register which is then decoded through the decoded logic matrix. This information is then transferred in parallel to the decoded shift register that serially shifts the data to the DRD output.

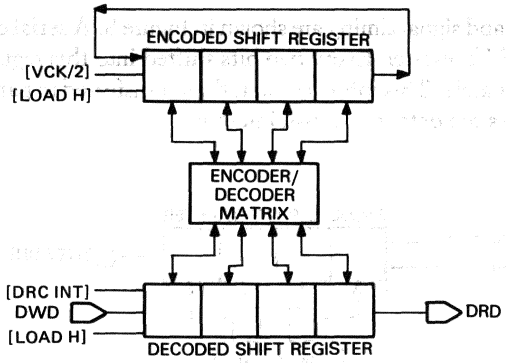


Figure 9 ▪ DC024 Diagnostic Mode Function

The timing for a typical diagnostic mode operation is shown in Figure 10. Table 4 contains the possible decoded write data input patterns and the expected decoded read data output patterns.

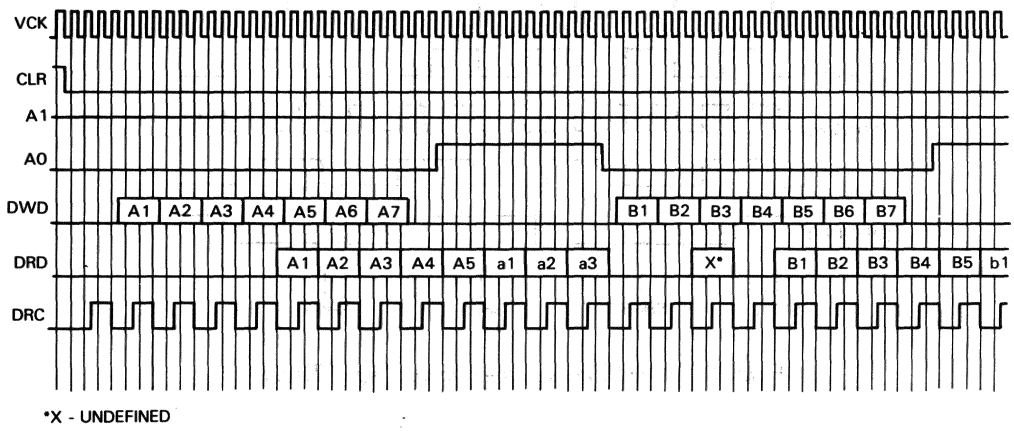


Figure 10 ▪ DC024 Typical Diagnostic Mode Signal Timing

Table 4 • DC024 Typical Diagnostic Mode Decoded Data Patterns

Write Data Input*							Read Data Output		
A1	A2	A3	A4	A5	A6	A7	a1	a2	a3
0	1	1	1	0	0	X	1	1	0
0	1	1	X	1	0	X	1	1	1
X	0	1	0	1	0	X	1	1	1
X	0	1	1	0	0	X	0	0	1
X	0	1	1	0	1	0	0	0	1
X	0	1	1	0	1	1	1	0	1
X	0	1	1	1	0	X	0	1	0
X	0	1	1	1	1	0	0	1	1
X	0	1	1	1	1	1	1	1	1
X	X	0	0	0	0	X	0	0	0
X	X	0	0	0	1	0	0	0	0
X	X	0	0	0	1	1	1	0	0
X	X	0	0	1	0	X	1	0	0
X	X	0	1	0	0	X	0	0	1
X	X	0	1	0	1	0	0	0	1
X	X	0	1	0	1	1	1	0	1
X	X	0	1	1	0	X	0	1	0
X	X	0	1	1	1	0	0	1	1
X	X	0	1	1	1	1	1	1	1
X	X	1	0	0	0	X	1	1	0

[1]X represents either a 1 or 0 state.
[2]Application of all input states listed will exercise 93 percent of the legal decode input sequences and 92 percent of the legal encode input sequences.

Read and Write Signal Timing

Figure 11 shows the timing relationship of the signals during read and write functions. During read operations, the encoded read data (ERD) from the head electronics of the storage device results in a decoded read clock (DRC) signal and the decoded read data (DRD) output. This output is normally transferred to a deserializer. During write operations, the decoded write data input from the serializer results in an encoded write clock (EWC) signal and the encoded write data (EWD+/EWD–) output to the storage device electronics. This output provides a nonreturn-to-zero signal to the head electronics of the storage device.

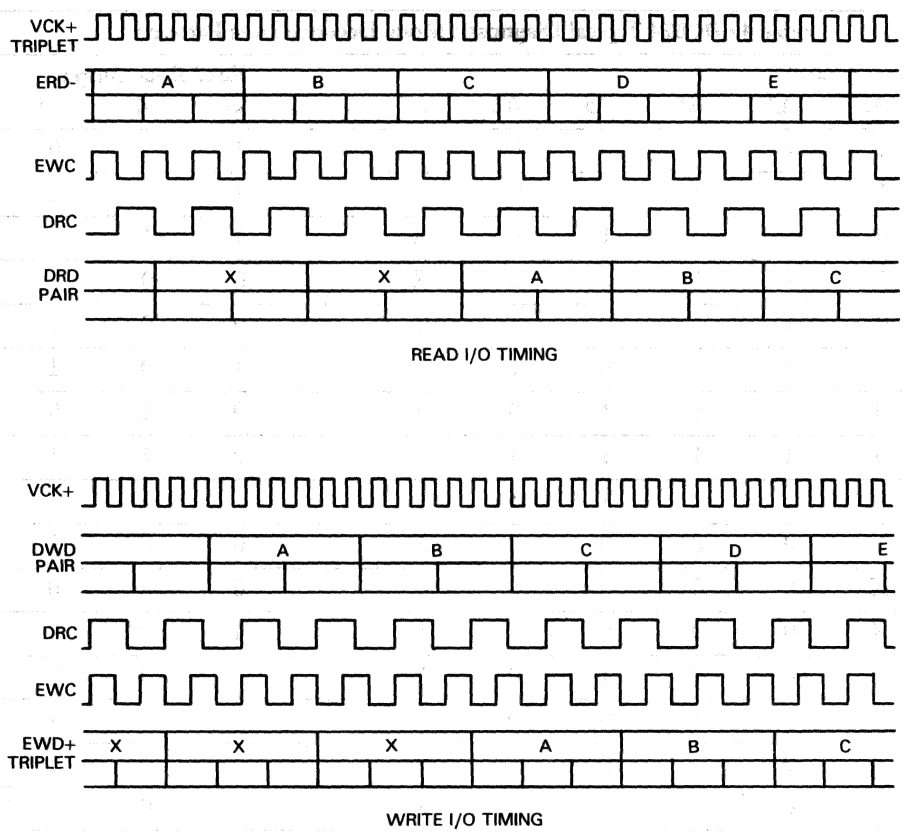


Figure 11 ▪ DC024 Read and Write Operation Timing

Counter/Divider Functions

The counter divider provides all the timing and clock signals, feedback to the phase detector and control for the encoding and decoding functions. The DC024 uses a ⅔-rate modulation code described in Table 2.

Transition Detector

During read/decode operation (Mode C), the encoded read data from the device is tranformed from a pulse format to a non-return-to-zero (NRZ) signal and synchronized with the voltage controlled oscillator feedback clock to allow loading and shifting into the encoded shift register. Figure 12 shows the logic and signal timing that performs this function.

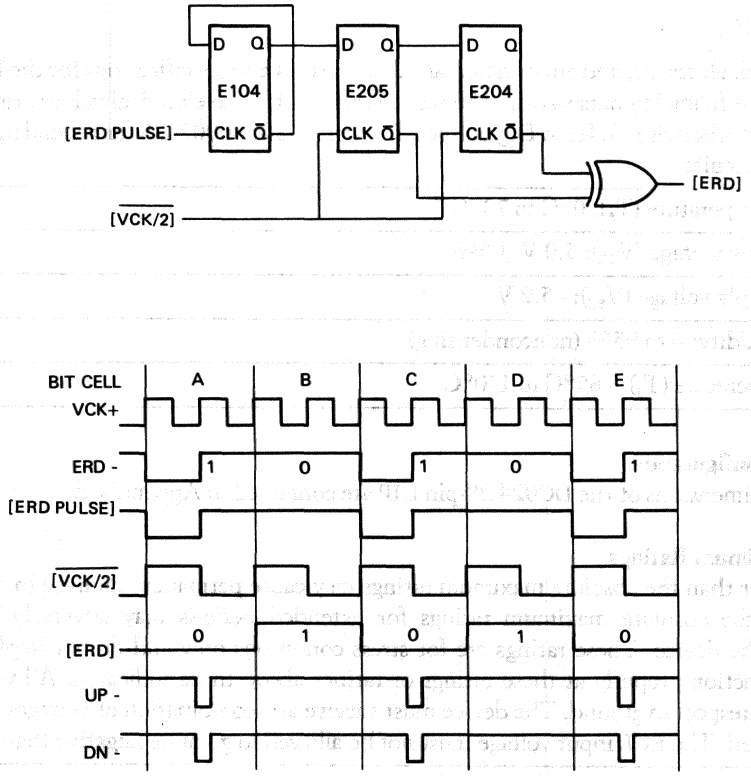


Figure 12 • DC024 Transition Detector and Signal Timing

Flip-flop E104 changes state on the rising edge of the encoded read data [ERD PULSE] input from the multiplexer. This indicates that a logic one of the data read has been detected. Flip-flop E205 synchronizes the data and, together with flip-flop E206 and the exclusive OR gate, provides the separation of the synchronized signal. To assure that transitions at E104 are recognized and clocked during the proper bit cell for correct decoding, the encoded read data and VCO clock input must be near a "phase-locked" condition.

A read data capture window defines a period of time during which a high-to-low transition of the ERD signal can occur and be recognized in the proper bit cell and therefore become correctly decoded. The probability that the data transition will not be recognized in the proper bit cell increases as the phase lock between the data and the VCO signal is degraded. As the rising edge of the ERD– signal passes the right or left window boundaries, the transition of flip-flop E104 will be recognized by E205 either a bit cell too late or a bit cell too early. Refer to Figure 15 and Table 7 for the the parameters associated with this window. Table 7 specifies the absolute limits for these paramenters to assures the correct decoding of encoded read data.

• Specifications

The mechanical, electrical, and environmental characteristics and specifications for the DC024 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise. Refer to Digital specification A-PS-2100002-GS for general information on integrated circuits.

- Operating temperature (T_A): 0°C to 70°C
- Positive supply voltage (V_{CC}): 5.0 V ±5%
- Negative supply voltage (V_{EE}): –5.2 V
- Relative humidity: 0 to 95% (noncondensing)
- Storage temperature (T_S): –65°C to 150°C

Mechanical Configuration

The physical dimensions of the DC024 28-pin DIP are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device. These ratings are for stress conditions only and do not imply that the device will function properly at these ratings or ratings above those indicated. All voltages are specified with respect to ground. The device must survive any combination of voltages within the ratings indicated. The ECL input voltage must not be allowed to go more negative than V_{EE}.

- Power supply voltage (V_{CC}): –0.3 V to 7.0 V
- Power supply voltage (V_{EE}): 0.3 V to –8.0 V
- ECL input voltage (V_{in}): 0.3 V to V_{EE}
- TTL input voltage (V_{in}): –0.5 V to 7.0 V

Recommended Operating Conditions

- Power supply voltage (V_{CC}): 4.75 V to 5.25 V
- Power supply voltage (V_{EE}): –5.46 V to –4.94 V
- Positive power supply current (I_{CC}): 213 mA (maximum)
- Negative power supply current (I_{EE}): –37 mA (maximum)
- Linear air flow 500 feet/minute (minimum)

dc Electrical Characteristics

The dc electrical parameters of the DC024 for the operating voltage and temperature ranges specified are listed in Table 5. Refer to Figures 13 and 14 for the test circuit configurations referenced in the table. The conditions listed in Table 6 apply to the test circuits.

Table 5 • DC024 dc Input and Output Parameters

Parameter	Symbol	Test Conditions[1,2,3]	Requirements Min.	Requirements Max.	Unit	Figure/Test
High-voltage input level	V_{IH}	V_{CC}=5.25 V	2.0	5.25	V	13/A
		Inputs: $\overline{DWC}$, DWD, VCK AO, A1, $\overline{DIAG}$, CLR,				
		Differential inputs:				
		V_{EE}=–4.92 V				
		ERD–=–3.0 V	–2.8	0	V	
		ERD+=0 V	–3.0	–0.2	V	
		VCK–=–3 V	–2.8	0	V	
		VCK+=0 V	–3.0	–0.2	V	
Low-level input voltage	V_{IL}	V_{CC}=4.75 V				13/A
		Inputs: $\overline{DWC}$, DWD, VCK, AO, A1, $\overline{DIAG}$, CLR	–0.1	0.8	V	
		Differential inputs:				
		V_{EE}=–5.46 V				
		ERD–=0 V	–3.0	–0.2	V	
		ERD+=–3 V	–2.8	0	V	
		VCK–=0 V	–3.0	–0.2	V	
		VCK+=–3 V	–2.8	0	V	
High-level input current	I_{IH}	V_{CC}=5.25 V				
		V_{in}=2.75 V				
		Inputs: $\overline{DWC}$, DWD, VCK, AO, A1, CLR	—	50	µA	13/D
		Input: $\overline{DIAG}$	—	150	µA	13/D
		Differential inputs:				14/E
		V_{EE}=–5.46 V				
		V_{in}=0 V[1]				
		ERD–=0 V	0	60	µA	
		ERD+=–3 V	0	60	µA	
		VCK–=0 V	0	60	µA	
		VCK+=–3 V	0	60	µA	
Low-level input current	I_{IL}	V_{CC}=5.25 V				
		V_{in}=0.5 V[2]				
		Inputs: $\overline{DWC}$, DWD, VCK, AO, A1, CLR	0	–500	µA	13/D
		Input: $\overline{DIAG}$	0	–1.5	mA	13/D
		Differential inputs:				14/E
		V_{EE}=–5.20 V				
		V_{in}=–3.0 V				
		ERD–=0 V	–5.0	5.0	µA	
		ERD+=–3 V	–5.0	5.0	µA	
		VCK–=0 V	–5.0	5.0	µA	
		VCK+=–3 V	–5.0	5.0	µA	

Parameter	Symbol	Test Conditions[1,2,3]	Requirements Min.	Max.	Unit	Figure/Test
High-level output voltage	V_{OH}	V_{CC}=4.75 V I_{out}=–2.6 mA Outputs: EWC, DRD, DRC	—	2.4	V	13/B
		Differential outputs: V_{EE}=–5.2 V UP+, UP–, DN+, DN–, EWD+, EWD–				13/C
		0°C	–1.0	–0.84	V	
		25°C	–0.96	–0.81	V	
		70°C	–0.905	–0.73	V	
Low-level output voltage	V_{OL}	V_{CC}=4.75 V I_{out}=8.0 mA Outputs: EWC, DRD, DRC	0	0.5	V	13/B
		Differential outputs: V_{EE}=–5.2 V UP+, UP–, DN+, DN–, EWD+, EWD–				13/C
		0°C	–1.87	–1.665	V	
		25°C	–1.85	–1.65	V	
		70°C	–1.832	–1.63	V	
Short-circuit output current	I_{OS}	V_{CC}=5.25 V[3] Output: EWC, DRD, DRC	–15	–60	mA	13/B
Positive power supply current	I_{CC}	V_{CC}=5.25 V[4] V_{EE}=–5.46 V	—	213	mA	14/F
Negative power supply current	I_{EE}	V_{EE}=–5.4 V[5] V_{CC}=5.25 V	—	–37	mA	14/G
Input clamp diode voltage	V_{IC}	V_{CC}=5.0 V V_{EE}=–5.2 V I_{in}=–18 mA Inputs: $\overline{DWC}$, DWD, VCK AO, A1, CLR $\overline{DIAG}$	–1.5	—	V	14/H
Input leakage current	I_I	V_{CC}=5.25 V V_{EE}=–5.46 V V_{in}=5.5 V Inputs: $\overline{DWC}$, DWH, VCK, A0, A1, CLR, $\overline{DIAG}$	–0.05	1.0	mA	14/H

[1]Voltages are specified with respect to ground.
[2]Positive values of current flow into the device and negative values flow out of the device.
[3]Not more than one output shall be short circuited to ground at a time and the duration of the short shall not exceed one second.
[4]This test is performed with V_{CC} at 5.2 V.
[5]This test shall be performed with V_{EE} at –4.2 V.

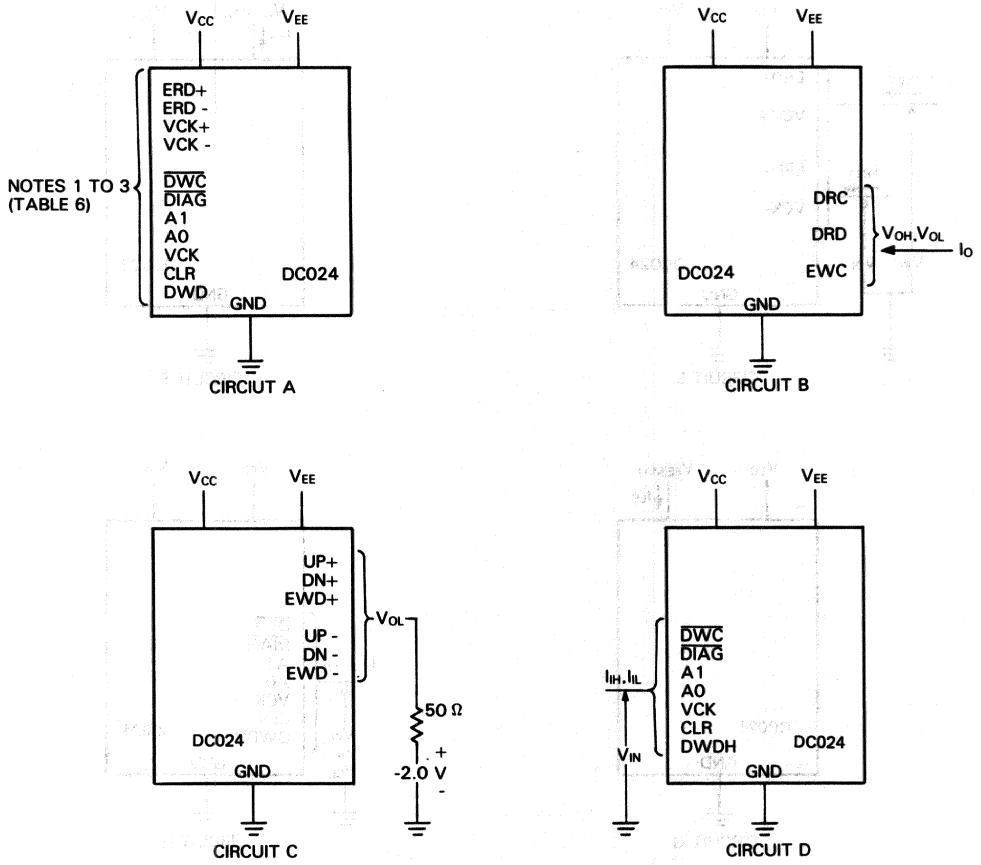


Figure 13 • DC024 dc Test Circuits

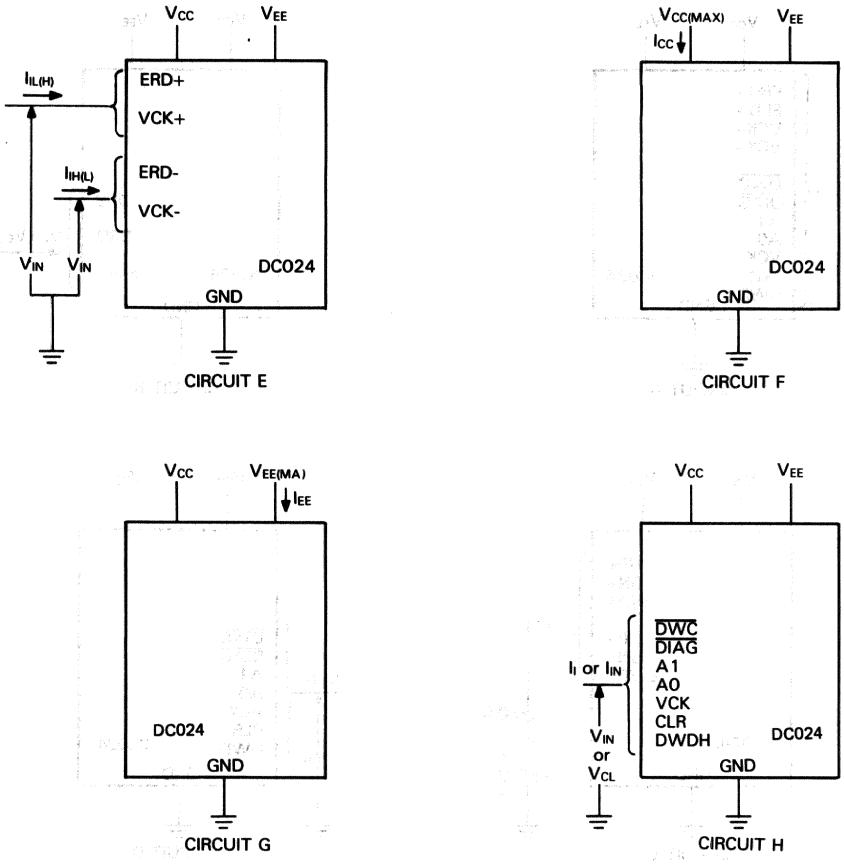


Figure 14 ▪ DC024 dc Test Circuits

Table 6 • DC024 dc Test Circuit Conditions

Condition	Figure/Circuit Reference
1. The V_{IH} and V_{OL} for the TTL inputs are determined by the test circuit logic.	13/A
2. Refer to Table 5 for ECL input conditions. V_{IH} and V_{IL} applied as indicated by the test sequence.	13/A
3. Each input is tested separately. All other inputs must meet the test requirements at the voltage specified in Table 5 for V_{IL} and V_{IH}.	13/A
4. The ECL inputs are tested by differential pairs and the input voltages are applied to meet the V_{IH} and V_{OL} specifications of Table 5.	13/A,13/B,13C 13/D,14E,14/F 14/G
5. The test conditions are listed in Table 5.	13/B
6. The input conditions are determined by the test sequence at any voltage within the range specified for V_{IH} and V_{OL}.	13/B,13/C
7. The input conditions are indicated in Table 5.	13/D,14/E
8. Each input is tested separately with the remaining TTL inputs connected to GND.	13/D,14/H
9. Each differential input pair is tested separately.	14/E
10. The TTL outputs are open.	13/D,14/E,14/F
11. The power supply current must not exceed the limits specified in Table 5 for any combination of input logic levels.	14/F,14/G

ac Electrical Characteristics

The input and output signal timing for the DC024 is shown in Figures 15 through 19. The propagation delays and conditions for the symbols on the timing diagrams are described in Tables 7 and 8. Refer to Appendix D for the standard TTL input and output voltage waveform parameters used for measuring the signal propagation delays. Figure 20 shows the TTL and ECL output loading circuits used for measuring the ac parameters. The time measurements of TTL are at 1.5 V and the ECL measurements are from the voltage crossing of the differential signals.

Table 7 • DC024 Read Data Capture Window ac Timing Parameters

Symbol	Definition	Specification (ns)* Min.	Nom.	Max.
t_{VCP}	Voltage control pulse	—	—	25
t_{EDM}	Early data window margin	—	t_{VCP} −2.0	t_{VCP} − 3.0
t_{LDM}	Late data window margin	—	t_{VCP} −2.5	t_{VCP} − 3.0

*Specifications are $T_A = 30°C$, $V_{CC} = 5.0$ V and $V_{EE} = -5.2$ V

Table 8 • DC024 Signal Timing Parameters

Symbol	Description	Requirements(ns) Min.	Max.	Figure/Mode
t_{DSV}	DWD setup to VCK+(+)	5.0	—	18/D
t_{DHV}	DWD hold to VCK+(+)	20	—	18/D
t_{ASV}	A1, AO setup to VCK+(+)	10	—	18/A
t_{AHV}	A1, AO hold to VCK+(+)	15	—	18/A
t_{ASV}	A1, AO setup to VCK(+)	10	—	18/A
t_{AHV}	A1, AO hold to VCK(+)	15	—	18/A
t_{DSD}	DWD setup to VCK(+)	5.0	—	19/DIAG-B
t_{DHD}	DWD hold to VCK(+)	20	—	19/DIAG-B
t_{CRV}	CLR release to VCK+(+)	25	—	16/A
t_{CRV}	CLR release to VCK(+)	25	—	16/A
t_{CRH}	CLR release to ERD–(+)	25	—	16/A
t_{CPW}	CLR pulse width high level	20	—	16/A
t_{EPH}	ERD– pulse width high (Mode C)[1]	45	—	17/A,B,C
t_{EPL}	ERD– pulse width low[1]	20	—	17/A,B,C
t_{VPH}	VCK+ pulse width high	12	—	17/A,B,C 17/D
t_{VPL}	VCK+ pulse width low	8.0	—	17/A,B,C 17/D
t_{VHD}	VCK pulse width high in diagnostic mode	18	—	19/DIAG-A
t_{VLD}	VCK pulse width low in diagnostic mode	18	—	19/DIAG-A
t_{DWH}	$\overline{DWC}$ pulse width high	20	—	17/D
t_{DPL}	$\overline{DWC}$ pulse width low	20	—	17/D
t_{DPR}	$\overline{DWC}$ period	75	—	17/D
t_{VPC}	VCK+/VCK– period	25	— —	17/A,B,C,D 17/D
t_{VPD}	VCK period in diagnostic mode	40	—	19/DIAG-A
t_{CDP}	CLR to DRC(–) propagation delay:			16/A
	$C_L = 15$ pF	5.0	—	
	$C_L = 70$ pF	—	32	
t_{CPD}	CLR to DRD(–) propagation delay:			16/A
	$C_L = 15$ pF	5.0	—	
	$C_L = 70$ pF	—	33	

Symbol	Description	Requirements(ns) Min.	Max.	Figure/Mode
t_{CEP}	CLR to EWD+(–) propagation delay:			16/A
	C_L=15 pF	5.0	—	
	C_L=70 pF	—	30	
t_{CDN}	CLR to DN–(+) propagation delay:			16/A
	C_L=15 pF	5.0	—	
	C_L=70 pF	—	30	
t_{CUP}	CLR to UP–(+) propagation delay			16/A
	C_L=15 pF	5.0	—	
	C_L=70 pF	—	30	
t_{EUP}	ERD–(+) to UP–(–) propagation delay:			17/A,B,C
	C_L=15 pF	5.0	—	
	C_L=70 pF	—	25	
t_{DUP}	$\overline{DWC}$(–) to –UP–(–) propagation delay:			17/D
	C_L=15 pF	5.0	—	
	C_L=70 pF	—	25	
t_{VDP}	VCK+(+) to DN–(–) proagation delay:			17/A,B,C
	C_L=15 pF	5.0	—	17/D
	C_L=70 pF	—	25	
t_{VTU}	VCK+(+) to –TRUNC UP–(+) propagation delay:			17/B
	C_L=15 pF	5.0	—	
	C_L=70 pF	—	35	
t_{VTD}	VCK+(+) to TRUNC DN–(+) propagation delay:			17/B
	C_L=15 pF	5.0	—	
	C_L=70 pF	—	35	
t_{DCU}	VCK+(+) to DRC(+) propagation delay:			16/C
	C_L=15 pF	9.0	—	
	C_L=70 pF	—	30	
t_{DCD}	VCK+(–) to DRC(–) propagation delay:			16/C
	C_L=15 pF	5.0	—	
	C_L=70 pF	—	25	
t_{DPU}	VCK+(–) to DRD(+) propagation delay:			16/C
	C_L=15 pF	5.0	—	
	C_L=70 pF	—	30	
t_{DPD}	VCK+(–) to DRD(–) propagation delay:			16/C
	C_L=15 pF	5.0	—	
	C_L=70 pF	—	32	
t_{EDU}	VCK+(+) to EWD+(+) propagation delay	5.0	25	16/D

Symbol	Description	Requirements(ns) Min.	Max.	Figure/Mode
t_{EDD}	VCK+(+) to EWD+(–) propagation delay	5.0	25	16/D
t_{VCR}	VCK(+) to DRC(+) propagation delay in diagnostic mode:			19/DIAG-A
	$C_L = 15$ pF	5.0	—	
	$C_L = 70$ pF	—	35	
t_{VCF}	VCK(–) to DRC(–) propagation delay in diagnostic mode:			19/DIAG-A
	$C_L = 15$ pF	5.0	—	
	$C_L = 70$ pF	—	30	
t_{VDR}	VCK(–) to DRD(+) propagation delay in diagnostic mode:			19/DIAG-A
	$C_L = 15$ pF	5.0	—	
	$C_L = 70$ pF	—	33	
t_{VDF}	VCK(–) to DN–(–) propagation delay in diagnostic mode:			19/DIAG-A
	$C_L = 15$ pF	5.0	—	
	$C_L = 70$ pF	—	33	
t_{VDN}	VCK(+) to DN–(–) propagation delay in diagnostic mode:			19/DIAG-A
	$C_L = 15$ pF	5.0	—	
	$C_L = 70$ pF	—	30	
t_{VTU}	VCK(+) to TRUNC UP–(+) propagation delay:			17/B
	$C_L = 15$ pF	5.0	—	
	$C_L = 70$ pF	—	40	
t_{VTD}	VCK(+) to TRUNC DN–(+) propagation delay:			17/B
	$C_L = 15$ pF	5.0	—	
	$C_L = 70$ pF	—	40	
t_{UPL}	UP– pulse width low	5.0	25	17/D
t_{DNL}	DN– pulse width low	5.0	25	17/A,B,C
t_{DNL}-t_{UPL}	UP– pulse to DN– pulse skew	–2.5	2.5	
t_{DCU}-t_{DCD}	VCK+(–) to DRC(–) to VCK+(+) to DRC(+) skew:			
	$C_L = 15$ pF	–5.0	12	
	$C_L = 70$ pF	–5.0	10	

Symbol	Description	Requirements(ns) Min.	Max.	Figure/Mode
t_{DPU}-t_{DCD}	VCK+(–) to DRC(–) to VCK+(–) to DRD(+) skew:			
	C_L=15 pF	–5.0	12	
	C_L=70 pF	–5.0	12	
t_{DPD}-t_{DCD}	VCK+(+) to DRC(–) to VCK+(–) to DRD(–) skew:			
	C_L=15 pF	–5.0	12	
	C_L=70 pF	–5.0	12	
t_{EDD}-t_{EDU}	VCK+(+) to EWD+(+) to VCK+(+)	–1.0	1.0	
t_{ECD}	VCK+(+) to EWC(–) propagation delay:			18/A,B,C,D
	C_L=15 pF	5.0	—	
	C_L=70 pF	—	27	
t_{ECU}	VCK+(+) to EWC(+) propagation delay:			18/A,B,C,D
	C_L=15 pF	5.0	—	
	C_L=70 pF	—	25	
t_{CEU}	CLR to EWC(+) propagation delay:			19/A,B,C,D
	C_L=15 pF	5.0	—	
	C_L=70 pF	—	26	
t_{CUP}-t_{CDN}	CLR to DN–(+) to CLR to UP–(+) skew	–2.0	+2.0	

[1] Optimum ERD+/ERD– pulse width is t_{VCP}+3.5 ns with nominal V_{CC}, V_{EE}, T_A.
[2] UP+, UP–, DN+, DN– must drive matched loads of 15 pF to 70 pF.
[3] EWD+ and EWD– must drive matched loads of 15 pF to 70 pF.

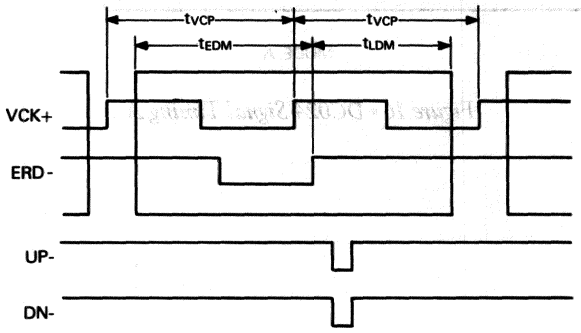
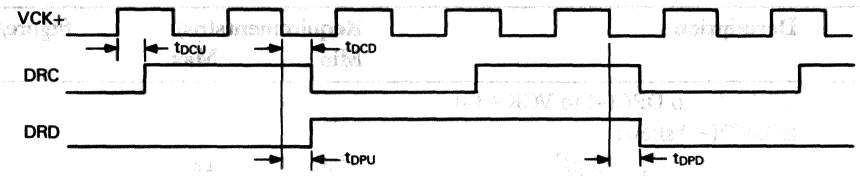
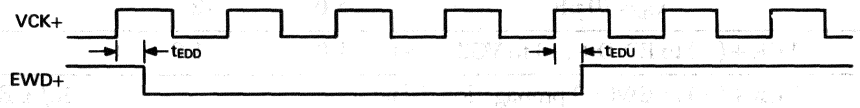


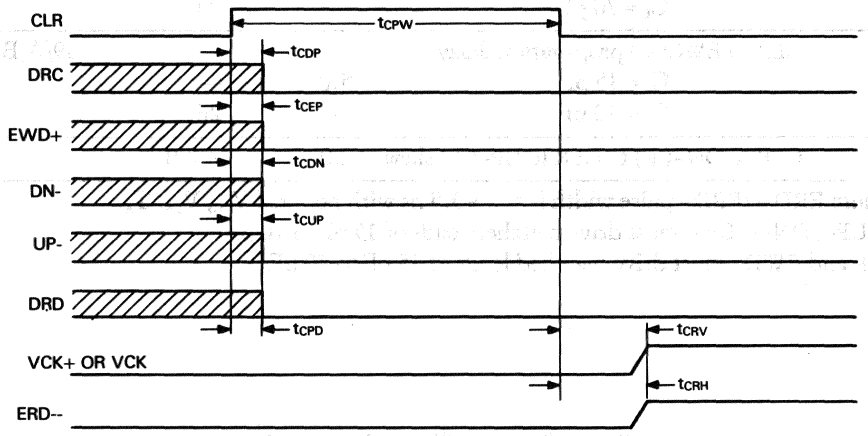
Figure 15 • DC024 Read Data Capture Window Signal Timing



MODE C

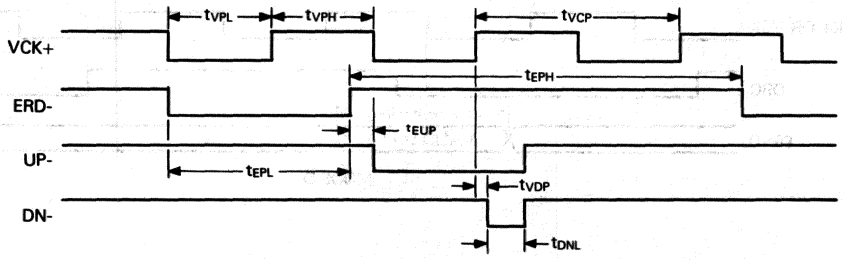


MODE D

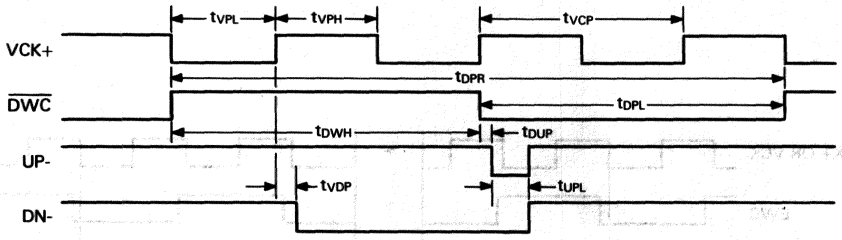


MODE A

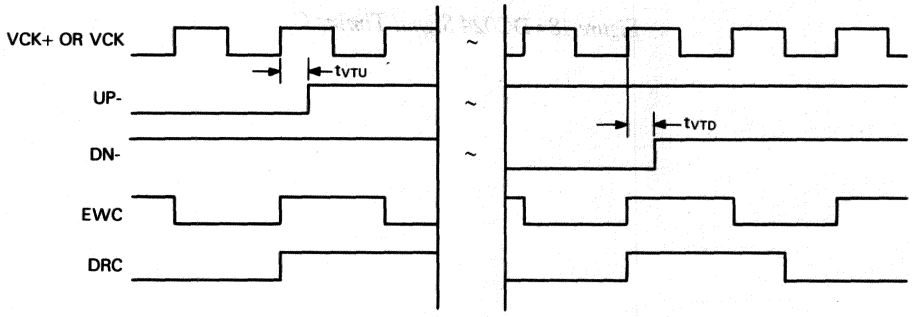
Figure 16 ▪ DC024 Signal Timing A



MODES A, B, C



MODE D



MODE B

Figure 17 • DC024 Signal Timing B

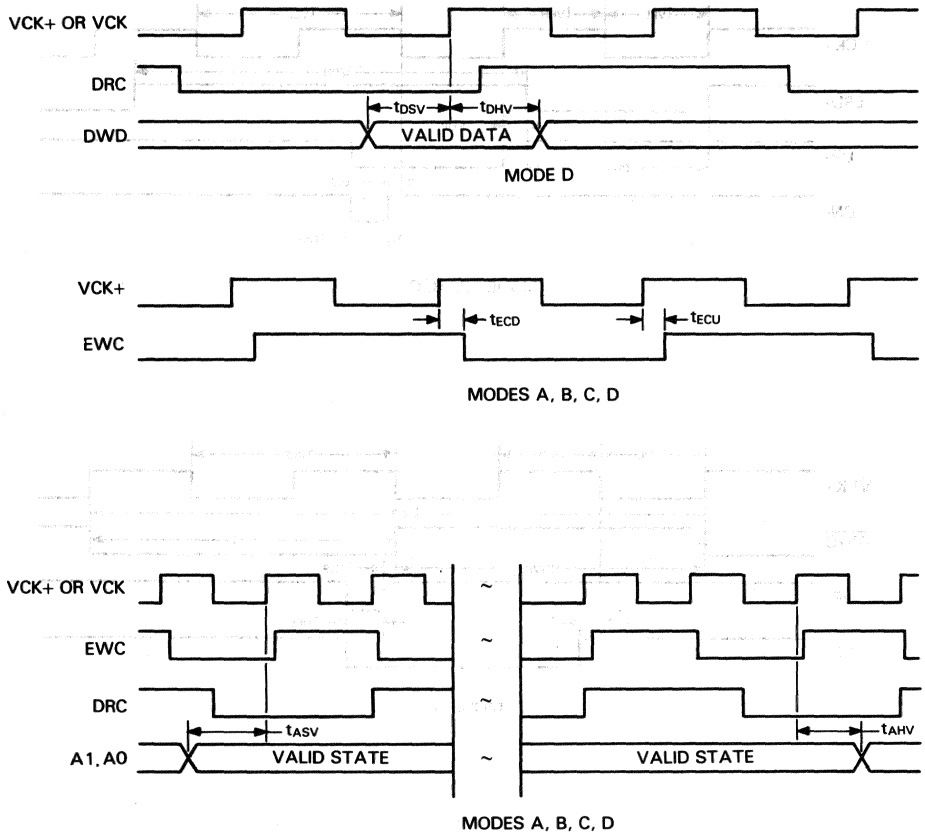


Figure 18 • DC024 Signal Timing C

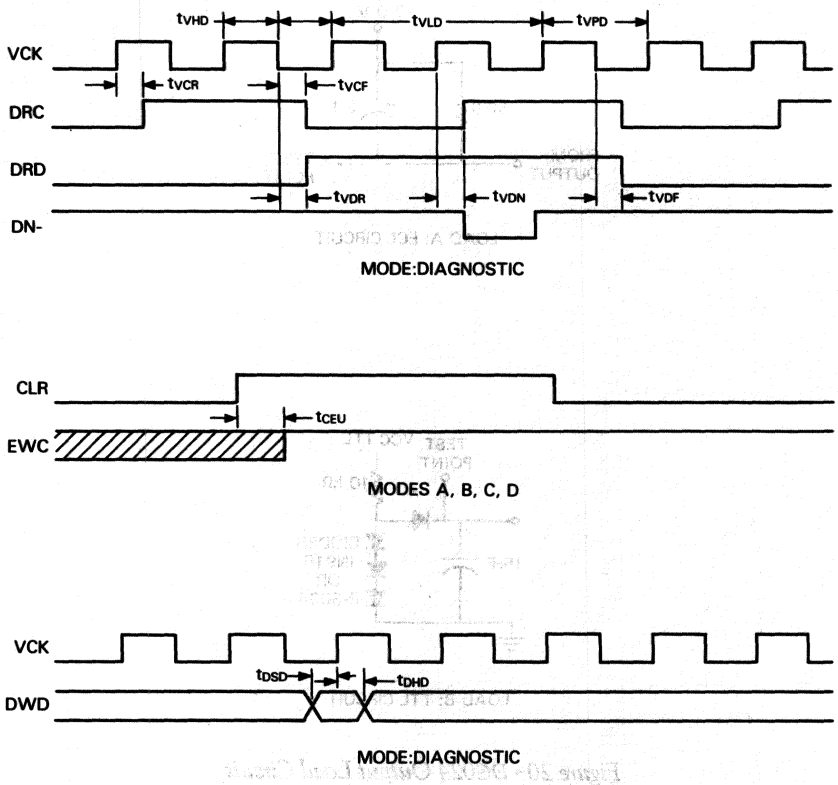
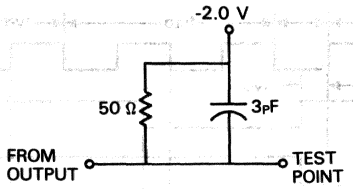
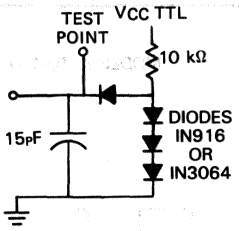


Figure 19 ▪ DC024 Signal Timing D



LOAD A: ECL CIRCUIT



LOAD B: TTL CIRCUIT

Figure 20 ▪ DC024 Output Load Circuit

• Features

- Implements error correction code (ECC)
- Accepts up to 10,060 bits of data in a 10-bit parallel form
- Corrects any number of errors contained in up to any eight of the 10-bit symbols in the data field
- Generates 170-bits of ECC information from parallel forms
- Compares 170-bits of calculated ECC with 170-bits of read ECC

• Description

The DC309 Reed Solomon generator is contained in a 28-pin dual-inline package (DIP) and is a dynamic NMOS LSI chip that implements a Reed Solomon error correction code (ECC). The DC309 accepts up to 10060 bits of of ECC information in 10-bit parallel form . It contains the logic to compare 170-bits of calculated ECC information with 170-bits of read ECC information. This is performed by generating in 10-bit parallel form the exclusive-OR result of the calculated ECC and read ECC. This result permits external correction of invalid data. The algorithm is capable of correcting any number of errors contained in up to any eight of the 10-bit symbols in the data field. Figure 1 is a block diagram of the DC309.

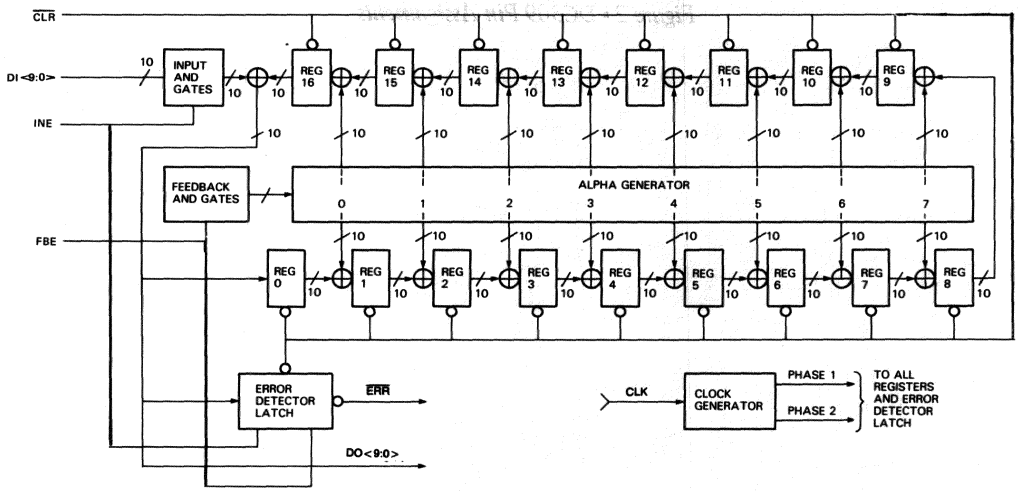


Figure 1 • DC309 Block Diagram

• Pin and Signal Description

This section provides a brief description of the input and output signals and power and ground connections of the DC309 28-pin DIP. The pin assignments are identified in Figure 2 and summarized in Table 1.

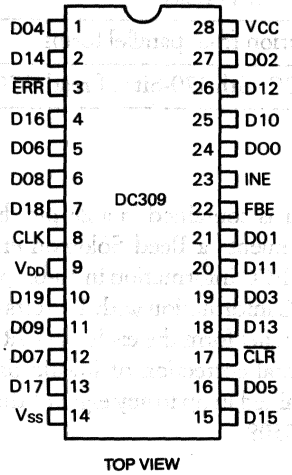


Figure 2 • DC309 Pin Assignments

Table 1 • DC309 Pin and Signal Summary

Pin	Symbol	Input/Output	Definition/Function
10,7,13, 4,15,2,18 26,20,25	DI<9:0>	input	Data in <9:0>—Data input lines that receive the data input in 10-bit parallel form.
11,6,12, 5,16,1, 19,27,21 24	DO<9:0>	output	Data out <9:0>—Data output lines that transfer 170 bits of 10-bit parallel (10 by 17) ECC following the calculation from input data and where 170 bits of 10-bit parallel (10 by 17) residue are transferred for correction.
22	FBE	input	Feedback enable—Asserted when ECC is being calculated and negated when ECC is being transferred or compared.
23	INE	input	Input enable—Asserted when data is received for ECC calculation and when the ECC is received to generate the output residue. This signal is negated when the DC309 transfers the ECC out.
17	$\overline{CLR}$	input	Clear—Asserted for a full clock time to initialize the DC309 before starting any ECC process.
8	CLK	input	Clock—The main timing signal that controls the DC309 operations. During normal operation, the input signals are latched and the outputs signals are available on the falling edge of clock signal. Timing variations can be implemented provided the switching times as specified are valid.
3	$\overline{ERR}$	output	Error out—A statically stored signal that is negated by the clear input for initialization purposes and asserted when the residue is found to be nonzero. The latch is enabled when the FBE signal is negated and the INE signal is asserted.
28	V_{CC}	input	Voltage—Power supply 5 Vdc
9	V_{DD}	input	Voltage—Power supply 12 Vdc
14	V_{SS}	input	Ground—Ground reference

Functional Operation

The DC309 contains 17 10-bit register stages, an alpha generator and feedback logic, an error detector latch, and a clock generator. The register stages and error detector are cleared by the clear signal. A two-phase clock output from the clock generator is distributed to each of the registers.

To generate the ECC, 10 bits of parallel data on lines DI<9:0> are entered through the input AND gates when enabled by the input enable (INE) signal. The parallel data from the input gates is transferred through the feedback gate to the alpha generator when enabled by the feedback enable (FBE) signal. The alpha generator produces eight 10-bit alpha-terms. Each bit of the alpha-term is composed of an exclusive-OR gated combination of from three to seven input bits and is OR gated with the register outputs. The results are clocked into the next register stage. This process continues until all data words are entered. The ECC is then stored at the outputs of the 17 register stages. The INE and FBE signals to the error detector latch are negated to shift out the 17-word ECC register data.

To compare the internally generated ECC with an external ECC, the INE signal is asserted and the FBE signal is negated. The internally generated ECC is shifted through the register stages and exclusive-OR gated with the input ECC. The resulting comparison data is transferred on the data out lines DO<9:0> and to the error detector. The error detector output ($\overline{ERR}$) is asserted when the input ECC does not equal the internally generated ECC.

Table 2 lists the logic levels required to select the DC309 functions.

Table 2 • DC309 Logic Function Selection

Function	Interface Signals* DI<9:0>	DO<9:0>	FBE	INE	$\overline{CLR}$	$\overline{ERR}$
Clear	X	L	X	X	L	H
Calculate ECC	M	X	H	H	H	H
Output ECC	X	M	L	L	H	H
Output remainder	M	M	L	H	H	M

*H = High logic level, L = Low logic level, X = High or low, M = Meaningful data.

• Specifications

The mechanical, electrical, and environmental characteristics and specifications for the DC309 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise. Refer to Digital specification A-PS-2100002-GS for the general specifications for integrated circuits.

- Operating temperature (T_A): 0°C to 75°C
- Power supply voltage (V_{CC}): 5.0 V ±5%
- Power supply voltage (V_{DD}): 12.0 V ±5%

Mechanical Configuration

The physical dimensions of the DC309 28-pin DIP are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device. Functional operation of the device at these or other conditions greater than indicated is not implied.

- Power supply voltage (V_{CC}): 7.0 V
- Power supply voltage (V_{DD}): 14 V
- All other pin voltages (V_T): –1.0 V to 10 V
- Operating temperature (T_A): 0°C to 70°C
- Storage temperature (T_{STG}): –65°C to 150°C
- Relative humidity: 10% to 95% (noncondensing)

Recommended Operating Conditions

- Power supply voltage (V_{CC}): 5 V ±5%
- Power supply voltage (V_{DD}): –5.46 V to –4.94 V
- Supply current (I_{CC}): 75 mA (maximum)
- Supply current (I_{DD}): 30 mA (maximum)
- Operating temperature (T_A): 0°C to 75°C

dc Electrical Characteristics

The dc electrical parameters of the DC309 for the operating voltage and temperature ranges specified are listed in Table 3. Refer to Appendix C for the test circuit configurations referenced in Table 3.

Table 3 • DC309 dc Input and Output Parameters

Parameter	Symbol	Test Condition	Requirements Min.	Requirements Max.	Units	Test Circuit
High-level input voltage	V_{IH}		2.0	—	V	C1
Low-level input voltage	V_{IL}		—	0.8	V	C1
High-level output voltage	V_{OH}	$V_{CC}=4.75$ V $I_O=-50$ µA	2.4	—	V	C1
Low-Input output voltage	V_{OL}	$V_{CC}=4.75$ V $I_O=2$ mA	—	0.45	V	C2
High-level input current	I_{IH}	$V_{CC}=5.25$ V $V_I=5.25$ V	-10	10	µA	C4
Low-level input current	I_{IL}	$V_I=0.8$ V $V_{CC}=5.25$ V	-10	10	µA	C5
Short-circuit output current	I_{OS}	$V_{CC}=5.25$ V	-25	-65	mA	C6
Supply current	I_{CC}	$V_{CC}=5.25$ V	—	75	mA	C7
	I_{DD}	$V_{DD}=12.6$ V	—	30	mA	C7

ac Electrical Characteristics

Table 4 lists ac timing parameters for the input and outputs signals shown in Figures 3 and 4. Figure 3 shows the setup and hold timing sequence and Figure 4 defines the propagation delays for the input and signals. The symbols referenced in the figures and tables are defined in Table 4. The output load circuit used for the ac measurements is shown in Figure 5. Refer to Appendix D for the input and output voltage waveform parameters used for measuring the signal propagation delays.

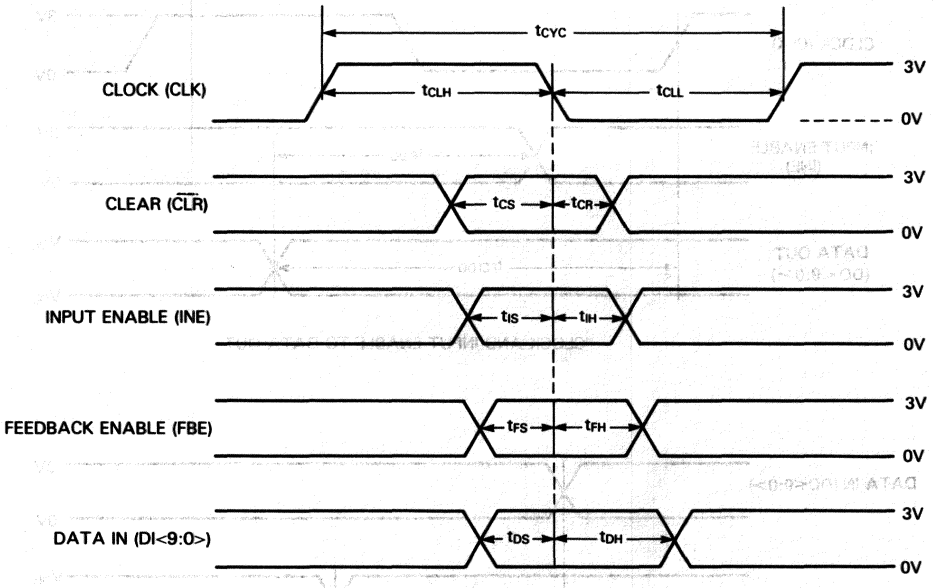


Figure 3 ▪ DC309 Setup and Hold Signal Timing

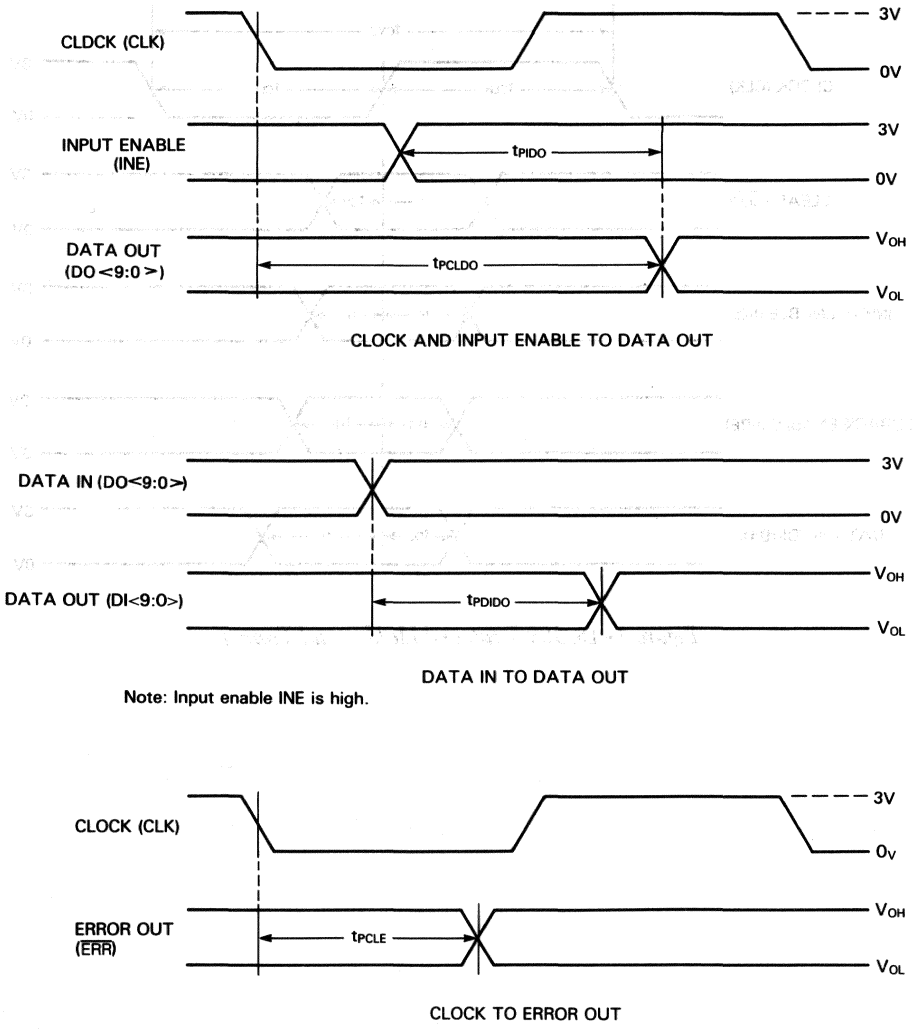


Figure 4 ▪ DC309 ac Signal Propagation Delays

Table 4 • DC309 ac Signal Timing Parameters

Symbol	Definition*	Requirements (ns) Min.	Max.
t_{CYC}	Clock cycle time	375	100,000
t_{CLH}	Clock high time	75	—
t_{CLL}	Clock low time	75	—
t_{IS}	Input enable setup time	30	—
t_{FS}	Feedback enable setup time	40	—
t_{DS}	Data input setup time	45	—
t_{CS}	Clear setup time	115	—
t_{IH}	Input enable hold time	25	—
t_{FH}	Feedback enable hold	35	—
t_{DH}	Data input hold time	50	-
t_{CH}	Clear hold time	25	—
t_{PCLDO}	Clock to output data propagation delay	30	200
t_{PIDO}	Input enable to output data propagation delay	—	125
t_{PDIDO}	Data input to output data propagation delay	—	80
t_{CLE}	Clock to error propagation delay	—	160
t_{PCE}	Clear to error propagation delay	—	160

*All time measurements shall be made from the 1.5 V level of the appropriate signals. Input rise and fall times shall be 15 ns maximum measured at the 10% and 90% levels. Valid output data prior to low-going transition must remain valid for a minimum of 30 ns after the clock transition.

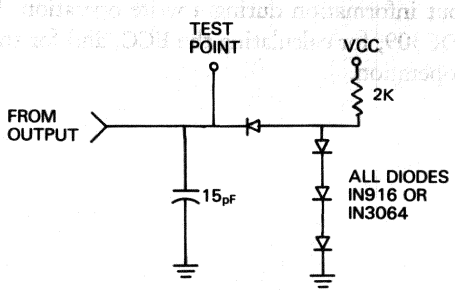


Figure 5 • DC309 ac Load Circuit

• Typical Application

Figure 6 shows a typical application of the DC309. During a media write sequence, data is read from main memory, an ECC is calculated, and the data and ECC are written into the serial memory. During the following read-back data sequence, the data and ECC are read from the serial memory, and an ECC is again calculated from the data read. If the ECC that accompanies the data read is different from the calculated ECC, an error condition is indicated.

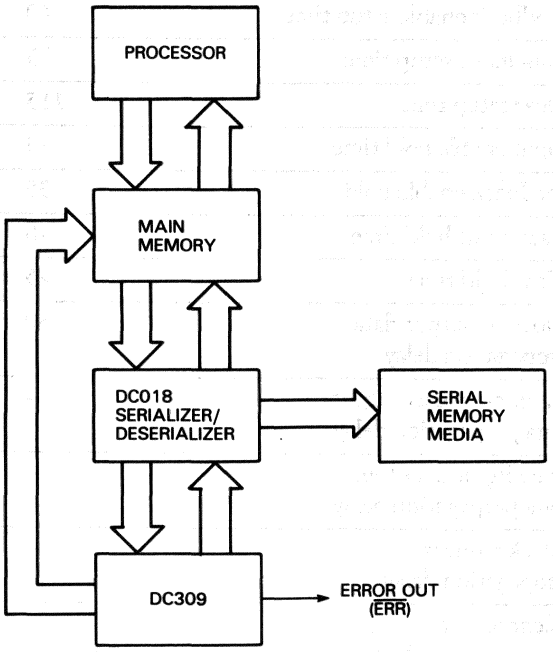


Figure 6 • DC309 Typical ECC Application Block Diagram

Figure 7 shows the timing sequence for clearing the DC309, for calculating the ECC, and for transferring the ECC output information during a write operation. Figure 8 shows the timing sequence for clearing the DC309, for calculating the ECC, and for transferring the ECC output information during a read operation.

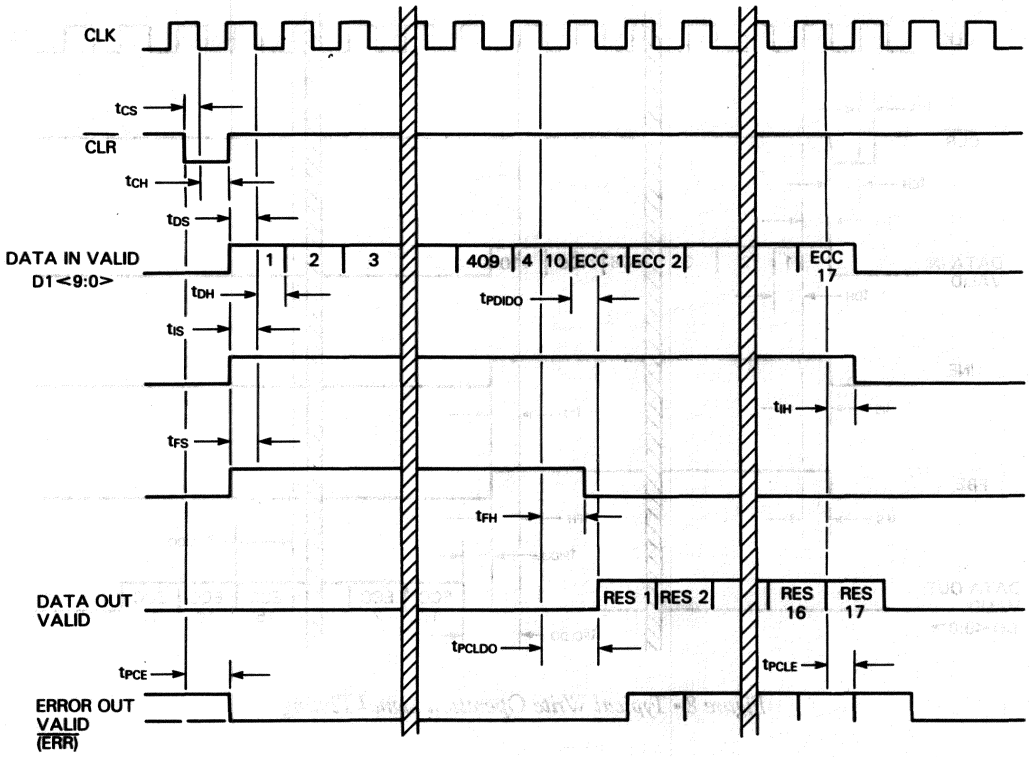


Figure 7 • DC309 Typical Read Operation Signal Timing

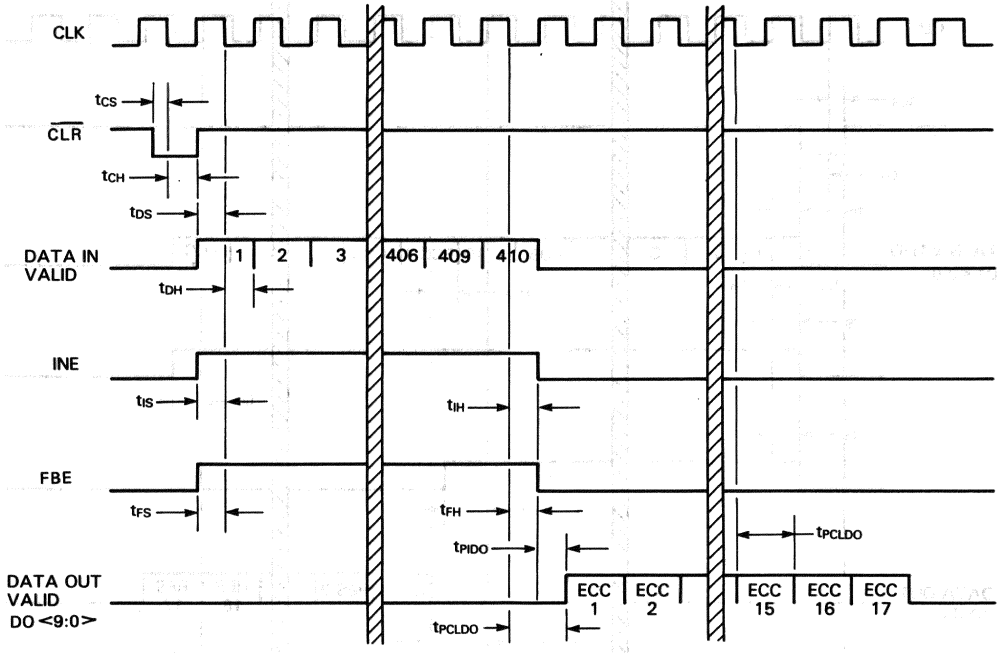


Figure 8 • Typical Write Operation Signal Timing

During a media write sequence, 16 bits of parallel data read from main memory are transferred to the serializer. The data from memory can be transferred in blocks consisting of up to 10,060-bits. The serializer converts the parallel data to serial data and transfers it to the serial memory. The parallel data is also converted into 10-bit parallel symbol form and transferred to the DC309. The DC309 generates an ECC from the data received and transfers the ECC to the serializer where it is transferred to the media.

During a read-back operation following the write operation, the serial data and ECC from the media are read and transferred to the deserializer. The data read is changed into a 10-bit parallel symbol form and transferred with the ECC to the DC309. After all the data is read, if the data is the same as previously written, the 170 bits of ECC will be the same. The DC309 generates the exclusive-OR function of the 170 bits of read-back ECC and the 170 bits of recently calculated ECC and the results (residue) are transferred to the main memory.

The DC309 checks the 170 bits of residue 10 bits at a time for all zero content. If the residue is all zeros, the read-back ECC is equal to the calculated ECC and no data errors are indicated. If the residue is not all zeros, the read-back ECC was not equal to the calculated ECC and the 170-bits of nonzero residue allow external mathematical calculations to correct any errors contained in any 8 of the 10-bit data symbols. A resulting error indication is transferred to main memory where it can be used to correct certain types of errors if required.

• Section 6—General Purpose Devices

The general purpose devices facilitate the development of processor interfaces for the control and transfer of information from processors to memory and other devices.

DC022 16-Word by 4-Bit Register File—The DC022 is a 28-pin DIP device that provides two independent read/write ports to allow simultaneous asynchronous access to a register file from either port.

DC102 Eight Channel Equals Checker—The DC102 is a 20-pin DIP that provides eight dual-input channels used to compare data in pairs for equality.

DC301 Dual Baud Rate Generator—The DC301 is an 18-pin DIP that provides program selectable baud rates (50 to 38,400) and separate transmit and receive frequency outputs to control the transfer of serial-line information.

▪ Features

- Two independent read/write ports
- Simultaneous asynchronous access to file register
- 10k series, ECL compatible input/output port capable of driving 15Ω-loads
- TTL input/output port capable of sinking 20 mA in a low state
- ECL and TTL address decoder and bidirectional data buffer

▪ Description

The DC022 register file, contained in a 28-pin dual-inline package (DIP), provides storage for up to 16 4-bit words and two independent read/write ports that allow simultaneous asynchronous access to the register file information from either port. Figure 1 is a simplified block diagram of the DC022.

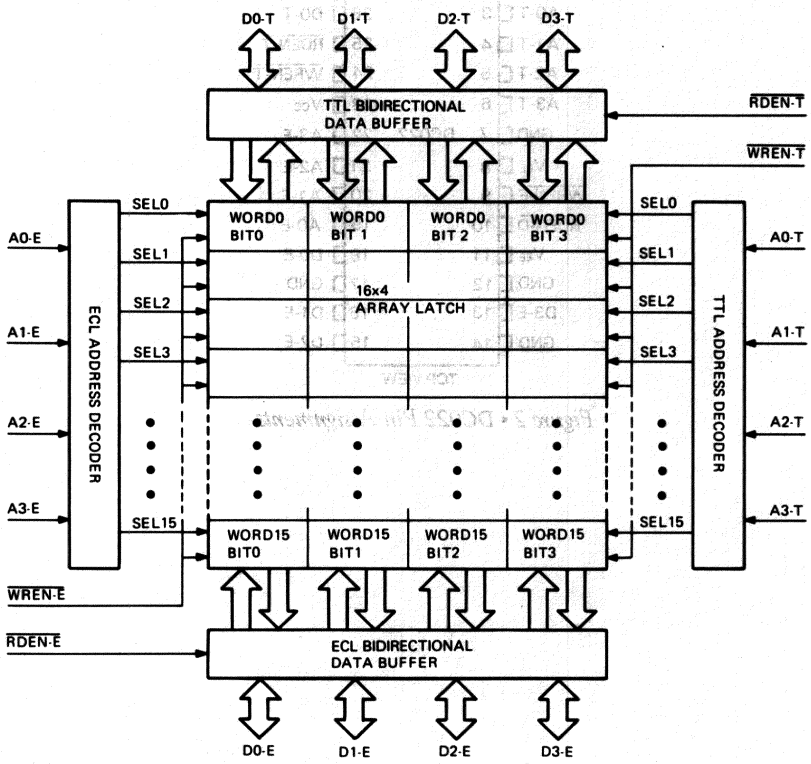


Figure 1 ▪ DC022 Simplified Block Diagram

Parallel data is transferred to and from the file through four sets of bidirectional data lines, one set for each port. Each port receives address inputs and read and write enable lines. One port has ECL 10k series macrocell arrays (MCA) compatible inputs and data outputs capable of driving 25 μA loads. The remaining port has TTL compatible inputs and outputs and is capable of sinking 20 milliamperes in the low state. A location in the file can be written from either port by specifying the address on the address inputs, negating the read enable input, transferring the data to the input/output lines, and asserting the write enable input. A location from the file is read from either port by specifying the address at the address inputs and by asserting the read enable input while the write enable input is negated.

▪ Pin and Signal Description

This section provides a brief description of the input and output signals and power and ground connections of the DC022 28-pin DIP. The pin assignments are identified in Figure 2 and summarized in Table 1.

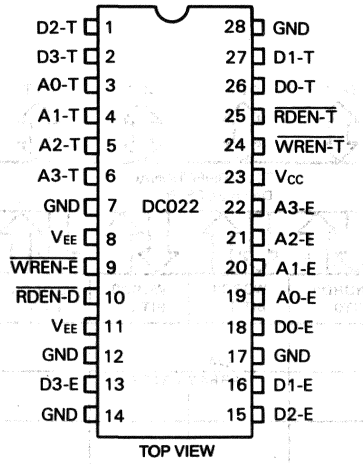


Figure 2 ▪ DC022 Pin Assignments

Table 1 • DC022 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
22-19	A<3:0>-E	inputs[1]	ECL address bits <3:0>—Bits 3 through 0 of the ECL port address.
9	$\overline{\text{WREN-E}}$	input[1]	ECL write enable—When asserted, data from the ECL data lines <3:0> is written into the register file location specified by the ECL address bits <3:0>. When negated, the location retains the data present.
10	$\overline{\text{RDEN-E}}$	input[1]	ECL read enable—The contents of the register file location specified by ECL address lines <3:0> is transferred to the ECL data lines <3:0>.
13-18	D<3:0>-E	inputs[1]/outputs[2]	ECL data lines <3:0>—Bidirectional lines 3 through 0 of the ECL port data with the emitter follower drivers enabled by $\overline{\text{RDEN-E}}$ signal.
6-3	A<3:0>-T	input[3]	TTL address bit <3:0>—Bits 3 through 0 of the TTL port address.
24	$\overline{\text{WREN-T}}$	input[3]	TTL write enable—When asserted, data on the TTL data inputs <3:0> is written to the register file location specified by the TTL address bits <3:0>. When negated, the location retains the data present.
25	$\overline{\text{RDEN-T}}$	input[3]	TTL read enable—The contents of the register file location specified by the TTL address lines <3:0> is transferred to TTL data lines <3:0> through the three-state drivers.
2,1, 27,26	D<3:0>-T	inputs[3]/outputs[4]	TTL data lines <3:0>—Bidirectional lines 3 through 0 of the TTL data port with three-state drivers enabled by $\overline{\text{RDEN-T}}$ signal.
23	V_{CC}	input[6,7]	Voltage—Power supply voltage to TTL input and output buffers.
8,11	V_{EE}	input[6,7]	Voltage—Power supply voltage to the array latch, ECL output buffers, and associated internal logic.
7,14,28	GND	input[5,7]	Ground—Ground reference

[1]ECL levels
[2]Emitter follower drivers
[3]TTL levels
[4]Three-state
[5]Pins 7, 12, 14, 17, and 28 connect to network ground (GND).
[6]Pins 8 and 11 connect to V_{EE}, Pin 23 connects to V_{CC}.
[7]The power and ground pins must be connected externally to the specified power and ground plane.

Operation

The DC022 consists of an ECL and TTL address decoder, an ECL and TTL bidirectional data buffer, and a 16 word by 4-bit array latch. The TTL bidirectional buffer transfers data between the three-state data bus and the array latch. When the read enable $\overline{\text{RDEN-T}}$ signal is asserted, the data from the array latch is transferred to the three-state TTL data lines D<3:0>-T. When the read enable signal is negated, the three-state output buffers are in a high impedance state. The ECL bidirectional data buffer transfers data between the 25 μA emitter OR gate type data bus and the array latch. When the read enable $\overline{\text{RDEN-E}}$ line is asserted, data from the array latch is transferred to the ECL data lines D<3:0>-E. When the read enable signal is negated, the ECL data lines are a high-impedance state. The TTL address decoder decodes the four TTL address bits A<3:0>-T to select one of the 16 4-bit words in the address latch for a read or write transaction. The 16 words are accessed through the 16 TTL address decoder and 16 ECL address decoder select lines. The TTL or ECL input data from the bidirectional data buffer is written into the location selected by the TTL or ECL address bits when the write enable ($\overline{\text{WREN-T}}$ or $\overline{\text{WREN-E}}$) lines are asserted. Data is read from the array latch and transferred to the TTL or ECL bidirectional data buffer by selecting the proper word through the appropriate TTL or ECL address bits. The dual select lines, data paths, and read/write controls allow simultaneous and independent operation of the TTL and ECL ports for any combination of read/write transactions. Simultaneous write operations from the two ports to the same word location, however, will result in undefined logic states for the addressed word. Simultaneous read and write operations to the same location will cause the data read to be the same as the data written.

Specifications

The mechanical, electrical, and environmental characteristics and specifications for the DC022 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise. Refer to Digital specification A-PS-2100002-GS for the general specifications for integrated circuits.

- Operating temperature (T_A): 0°C to 75°C
- Supply voltage (V_{CC}): 5.0 V ±5%
- Relative humidity: 0 to 95% (noncondensing)

Mechanical Configuration

The physical dimensions of the DC022 28-pin DIP are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Positive supply voltage (V_{CC}): –0.3 V to 7.0 V
- Negative supply voltage (V_{EE}): –8.0 V to 0.3 V
- ECL input voltage (V_{in}): 0.3 V to V_{EE}
- TTL input voltage (V_{in}):–0.5 V to 7.0 V
- Relative humidity: 10% to 95% (noncondensing)

Recommended Operating Conditions

- Supply voltage (V_{CC}): 4.75 V to 5.25 V (V_{EE}): –5.46 V to –4.94 V
- Supply current (I_{CC}): 425 mA (maximum)
- Operating temperature (T_A): 0°C to 75°C
- Linear air flow 500 feet/minute (minimum)

dc Electrical Characteristics

The dc electrical parameters of the DC022 for the operating voltage and temperature ranges specified are listed in Table 2. Refer to Figures 3 and 4 for the test circuit configurations referenced in the Table 2. Table 3 lists the conditions that apply to the test circuits.

Table 2 • DC022 dc Electrical Characteristics

Parameter	Symbol	Test Conditions[1,5]	Requirements Min.	Requirements Max.	Units	Figure/Test
High-level input voltage	V_{IH}	D<3:0>-T	2.0	5.25	V	3/A
		A<3:0>-T				
		$\overline{\text{WREN-T}}$				
		$\overline{\text{RDEN-T}}$				
		D<3:0>-E				3/B
		A<3:0>-E				
		$\overline{\text{WREN-E}}$				
		$\overline{\text{RDEN-E}}$				
		0°C	–1.145	–0.74	V	
		25°C	–1.105	–0.71	V	
		75°C	–1.405	–0.62	V	
Low-level input voltage	V_{IL}	D<3:0>-T	–0.1[2]	0.8	V	3/A
		A<3:0>-T				
		$\overline{\text{WREN-T}}$				
		$\overline{\text{RDEN-T}}$				3/B
		D<3:0>-E				
		A<3:0>-E				
		$\overline{\text{WREN-E}}$				
		$\overline{\text{RDEN-E}}$				3/A
		0°C	–2.5	–1.49	V	
		25°C	–2.5	–1.475	V	
		75°C	–2.5	–1.45	V	

Parameter	Symbol	Test Conditions[1,5]	Requirements Min.	Max.	Units	Figure/Test
High-level input current	I_{IH}	D<3:0>-T $V_{in}=2.7$ V	0	100	µA	3/C
		A<3:0>-T $\overline{WREN}$-T $V_{in}=2.7$ V	0	20	µA	
		$\overline{RDEN}$-T $V_{in}=2.7$ V	0	50	µA	
		D<3:0>-E $V_{in}=V_{IH}$ max	0	300	µA	3/D
		A<3:0>-E $\overline{WREN}$-E $\overline{RDEN}$-E $V_{in}=V_{IH}$ max	0	200	µA	3/D
Low-level input current	I_{IL}	D<3:0>-T	-500	0	µA	3/C
		A<3:0>-T $\overline{WREN}$-T $V_{in}=0.5$ V	-400	0	µA	
		$\overline{RDEN}$-T $V_{in}=0.5$ V	-200	0	µA	
		D<3:0>-E $V_{in}=-2.0$	-800	20	µA	3/D
		A<3:0>-E $\overline{WREN}$-E $\overline{RDEN}$-T $V_{in}=-2.0$ V	0	20	µA	
Input clamp voltage	V_{IC}	D<3:0>-T A<3:0>-T $\overline{WREN}$-T $\overline{RDEN}$-T $I_{in}=-18$ mA[3]	-1.2	0	V	3/E
Input current at maximum input voltage	I_{IM}	D<3:0>-T $V_{in}=7.0$ V	0	300	µA	3/C
		A<3:0>-T $\overline{WREN}$-T $\overline{RDEN}$-T $V_{in}=7.0$ V	0	100	µA	

Parameter	Symbol	Test Conditions[1,5]	Requirements Min.	Max.	Units	Figure/Test
High-level output voltage	V_{OH}	D<3:0>-T I_{out}=-2.6 mA	2.4		V	3/B
		D<3:0>-E I_{out}=40 mA I_{out}=50 mA				3/A
		0°C	-1.0	-0.84	V	
		25°C	-0.96	-0.81	V	
		75°C	-0.9	-0.72	V	
Low-level output voltage	V_{OL}	D<3:0>-T I_{out}=-20.0 mA	0	0.5	V	3/B
Low-level output current	I_{OL}	D<3:0>-E V_{out}=-2.0 V	-800	20	µA	3/A
Short circuit output current	I_{OS}	D<3:0>T[4]	-40	-100	mA	4/F
Positive power supply current	I_{CC}		0	45	mA	4/G
Negative power supply current	I_{EE}		-400	0	mA	4/G

[1]All inputs must be at the voltage levels specified to achieve the desired logic state. Open inputs will result in undefined logic states being propagated through the device. All voltages are specified with respect to network ground.

[2]On a transient basis, V_{IL} (min.) for TTL inputs is considered to have a value equal to the input clamp voltage (V_{IC}) as measured for the device under test. The device must meet functional requirements when any combination of inputs is subjected to a V_{IC} transient of up to 100 nanoseconds.

[3]The duration of the specified input current (I_{in}) on a TTL input for the input clamp voltage (V_{IC}) test must not exceed 1.0 milliseconds.

[4]All TTL outputs must meet all dc and ac characteristics after the application of ground for 1.0 seconds. Only one output may be shorted to ground at a time.

[5]The definitions of input and output voltage and current parameters apply to both TTL and ECL inputs and outputs unless stated otherwise.

Table 3 • DC022 dc Test Circuit Conditions

Figure/Test	Condition
3/A,3/B	Each input is tested separately. The test requirement must be met with other inputs at any voltage within the range specified and for any voltage within the range specified for V_{IL} or V_{IH}. Pins D<3:0>-E are terminated to –2.0 V through a 25Ω resistor for input test.
3/C	Each input is tested separately. To test D<3:0>-T, $\overline{\text{RDEN-T}}$ must be driven by a high-level voltage and the input currents specified must be met over the range of V_{IH} specified for $\overline{\text{RDEN-T}}$.
3/D	Each input is tested separately. To test D<3:0>-E, $\overline{\text{RDEN-E}}$ must be driven by a high-level voltage and the input currents specified must be met over the range of V_{IH} specified for $\overline{\text{RDEN-E}}$.
3/E	Each input is tested separately. To test D<3:0>-T, $\overline{\text{RDEN-T}}$ must be driven by a high-level voltage and the input currents specified must be met over the range of V_{IH} specified for $\overline{\text{RDEN-T}}$.
4/F	Each input is tested separately. To test D<3:0>-T, $\overline{\text{RDEN-T}}$ must be driven by a low-level voltage. The short circuit currents (I_{OS}) specified must be met over the range of V_{IH} specified for $\overline{\text{RDEN-T}}$.
4/G	Each input is tested separately. 1. The power supply currents may not exceed the limits specified for any combination of input logic levels. 2. The total power dissipation will increase when the outputs are connected to loads.

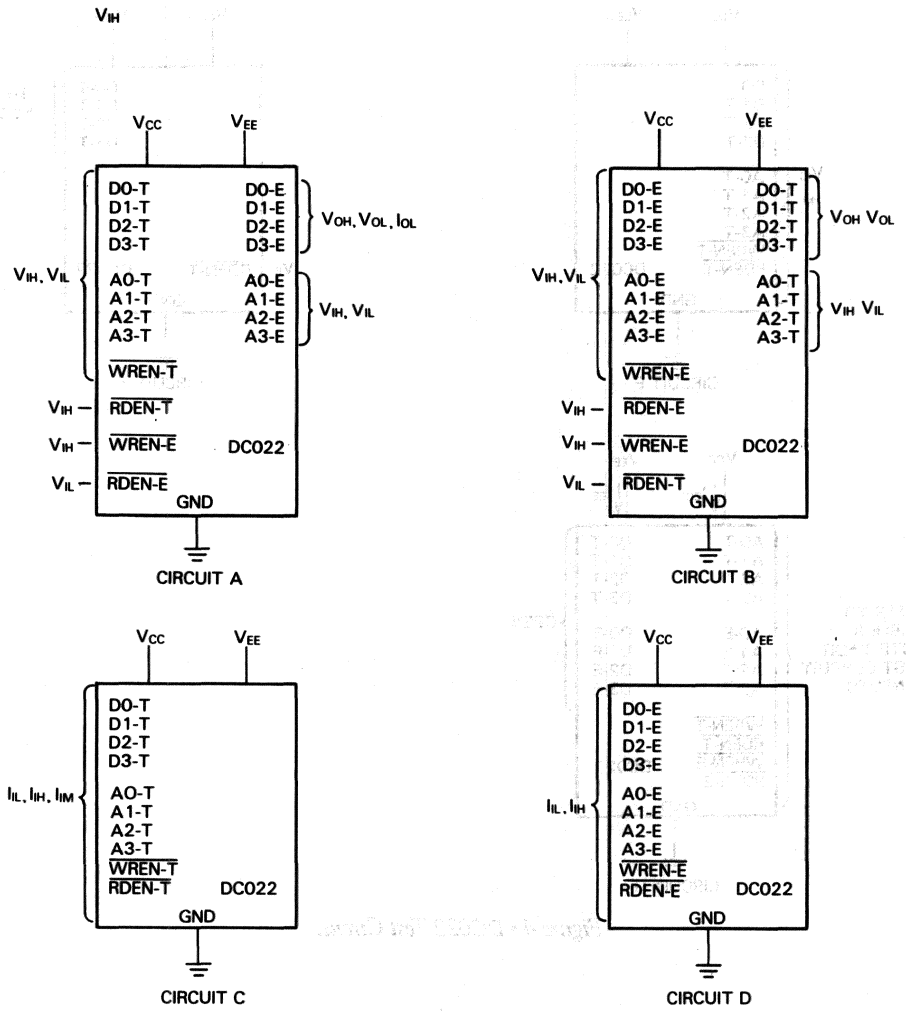


Figure 3 ▪ DC022 Test Circuits

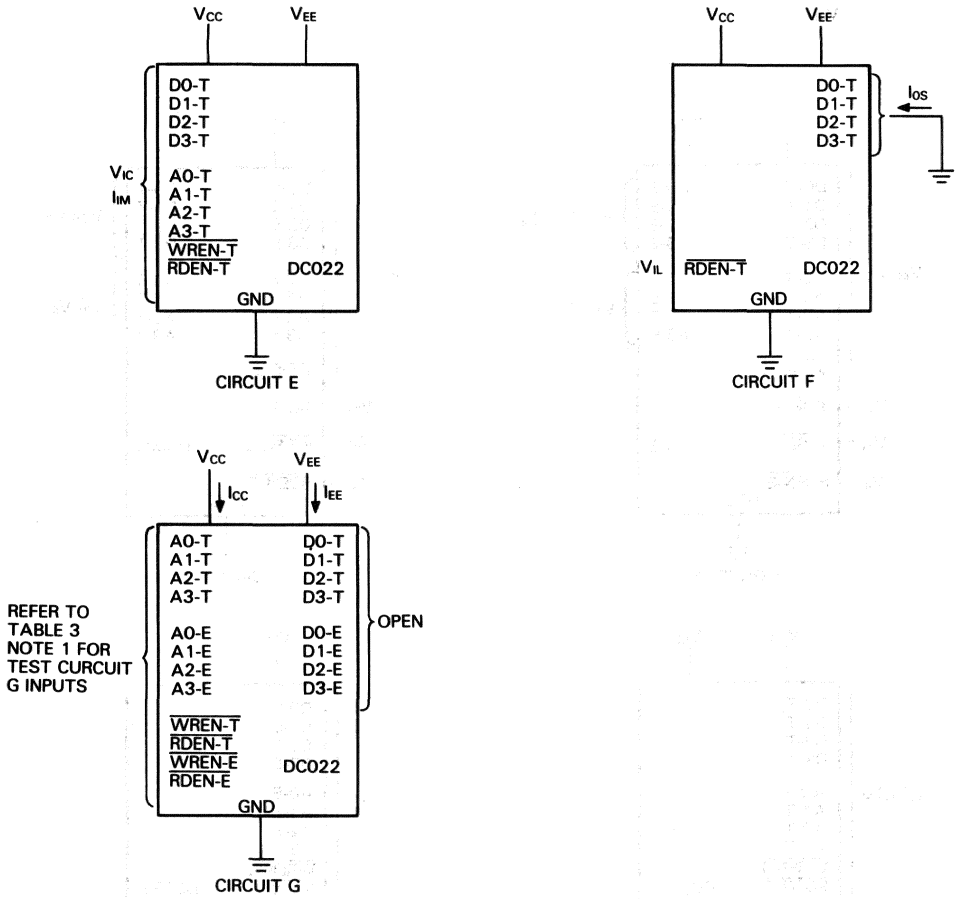


Figure 4 • DC022 Test Circuits

ac Electrical Characteristics

Table 4 lists ac timing parameters for the input and outputs signals shown in Figures 5 through 9. The symbols referenced in the figures and tables are defined in Table 5. The TTL and ECL output load circuits used to measure the ac signal parameters are shown in Figure 10. Refer to Appendix D for the input and output voltage waveform parameters used for measuring the signal propagation delays.

Table 4 • DC022 ac Parameter Definitions

Symbol	Name/Definition*
t_{AA}	Address access time—The time between the specified references on the address input and data output voltage waveforms with the output changing from one defined level (high or low) to the other defined level. (Read enable is asserted and write enable is negated.)
t_{DA}	Data access time—The time between the specified references on the data input voltage waveforms of one port and the data output voltage waveforms of the other port with the output changing from one defined level (high or low) to the other defined level. The first mentioned port is set up to write the array latch. (Write enable is asserted and read enable is negated.) The second port is set up to read the array latch (read enable is asserted and write enable is negated), and both ports are set up to access the same location in the array latch.
t_{WA}	Write access time—The time between the specified reference points on the write enable input voltage waveform of one port and the data output voltage waveforms of the other port with the output changing from one defined level (high or low) to the other defined level. The first mentioned port is set up to write the array latch (read enable is negated and write enable is in the process of being asserted). The second port is set up to read the array latch (read enable is asserted and write enable is negated), and both ports are set up to access the same location in the array latch.
t_{OE}	Output enable time—The propagation delay time between the specified reference points on the read enable input and the data output voltage waveforms with the output changing from the high-impedance (off) state to either of the defined active levels (high or low). For the three-state TTL outputs, both the high and low levels are active. For the open-emitter ECL outputs, only the high level is active.
T_{OD}	Output disable time—The propagation delay time between the specified reference points on the read enable input and the data output voltage waveforms with the output changing from either of the active levels (high or low) to the high-impedance (off) state. For the three-state TTL outputs, both the high and low levels are active. For the open-emitter ECL outputs, only the high level is active.
T_{AS}	Address setup time—The time interval between the specified reference points on the address input and write enable input waveforms that defines the earliest point in time that the address inputs must assume stable logic states with respect to the write enable input pulse. The logic levels at the address inputs must remain stable between the points in time defined by the address setup and hold times to ensure proper transfer of logic levels at the data inputs into the array location specified by the address without affecting any other location in the array.
T_{AH}	Address hold time—The time interval between the specified reference points on the write enable input and address input waveforms that defines the latest point in time that the address inputs must maintain stable logic states with respect to the write enable input pulse. The logic levels at the address inputs must remain stable between the points in time defined by the address setup and hold times to ensure proper transfer of logic levels at the data inputs into the array location specified by the address without affecting any other location in the array.

Symbol	Name/Definition*
t_{DS}	Data setup time—The time interval between the specified reference points on the data input and write enable input waveforms that defines the earliest point in time that the data inputs must assume stable logic states with respect to the write enable input pulse. The logic levels at the data inputs must remain stable between the points in time defined by the data setup and hold times to ensure proper transfer of logic levels at the data inputs into the array location specified by the address.
t_{DH}	Data hold time—The time interval between the specified reference points on the write enable input and data input waveforms that defines the latest point in time that the data inputs must maintain stable logic states with respect to the write enable input pulse. The logic levels at the data inputs must remain stable between the points in time defined by the data setup and hold times to ensure proper transfer of logic levels at the data inputs into the array location specified by the address.
t_{WP}	Write pulse width—The time interval between the specified reference points on the leading and trailing edges of the waveform that must be applied to the write enable input to ensure proper transfer of logic levels at the data inputs into the array location specified by the address.
t_R	Rise-time—The time between a specified low-level voltage and a specified high-level voltage on a waveform that is changing from the low level to the high level.
t_F	Fall-time—The time between a specified high-level voltage and a specified low-level voltage on a waveform that is changing from the high level to the low level.
C_{in}	Input capacitance—The capacitance measured at the specified pins with power applied to the device.
R_{in}	Real input impedance—The real portion of the input impedance measured at the specified pins with power applied to the device.

*The definition of the input and output voltage and current parameters and the timing parameters apply to the TTL and ECL ports unless stated otherwise.

Table 5 • DC022 ac Electrical Characteristics

Symbol	Definition	Load (C_L)	Requirements (ns) Min.	Max.	Figure/Level
t_{AA}	Address access time[1]				
	A<3:0>-T input to	15 pF	2.0	30	5/TTL
	D<3:0>-T output	50 pF	2.0	30	
		150 pF	2.0	40	
	Address access time[2]				
	A<3:0>-E input to	30 pF	2.0	30	5/ECL
	D<3:0>-E output	90 pF	2.0	30	
t_{DA}	Data access time[3]				
	D<3:0>-E input	15 pF	6.0	30	6/TTL
	D<3:0>-T output	50 pF	6.0	30	
		150 pF	6.0	33	
	Data access time[4]				
	D<3:0>-T input	30 pF	6.0	30	6/ECL
	D<3:0>-E output	90 pF	6.0	30	
t_{WA}	Write access time[5]				
	$\overline{\text{WREN-E}}$ input	15 pF	6.0	32	7/TTL
	D<3:0>-T output	50 pF	6.0	32	
		150 pF	6.0	36	
	Write access time[6]				
	$\overline{\text{WREN-T}}$ input	30 pF	6.0	32	7/ECL
	D<3:0>-E output	90 pF	6.0	32	
t_{OE}	Output enable time				8/TTL
	$\overline{\text{RDEN-T}}$ input	15 pF	5.0	22	
	D<3:0>-T output	50 pF	5.0	22	
		150 pF	5.0	28	
	$\overline{\text{RDEN-E}}$ input	3.5 pF	5.0	22	8/ECL
	D<3:0>-E output	30 pF	5.0	23	
		90 pF	5.0	25	
t_{OD}	Output disable time				8/TTL
	$\overline{\text{RDEN-T}}$ input	15 pF	3.0	20	
	D<3:0>-T output	50 pF	3.0	20	
		150 pF	3.0	40	
	$\overline{\text{RDEN-E}}$ input	3.5 pF	2.0		8/ECL
	D<3:0>-E output	30 pF		18	
		90 pF		20	
t_{AS}	Address setup time[7]				
	A<3:0>-T input[8]		5.0		9/TTL
	A<3:0>-E input[9]		8.0		9/ECL

Symbol	Definition	Load (C_L)	Requirements (ns) Min.	Max.	Figure/Level
t_{AH}	Address hold time[7]				
	A<3:0>-T input[8]		8.0		9/TTL
	A<3:0>-E input[9]		9.0		9/ECL
t_{DS}	Data setup time[7]				
	D<3:0>-T input[8]		11.0		9/TTL
	D<3:0>-E input[9]		3.0		9/ECL
t_{DH}	Data hold time[7]		12.0		
	D<3:0>-T input[8]		18.0		9/TTL
	D<3:0>-E input[9]		19.0		9/ECL
t_{WP}	Write pulse width[7]				
	$\overline{\text{WREN-T}}$ input[8]		18.0		9/TTL
	$\overline{\text{WREN-E}}$ input[9]		15.0		9/ECL
t_R	Rise time[10]				
	D<3:0>-T output	15 pF	3.0		
		50 pF		8.0	
	D<3:0>-E output	3.5 pF	1.0		
		90 pF		5.0	
t_F	Fall time[11]				
	D<3:0>-T output	15 pF	1.0		
		50 pF		8.0	
	D<3:0>-T output	3.5 pF	0.5		
		90 pF		4.0	
C_{in}	Input capacitance				
	D<3:0>-T = 10 pF (max)[12]				
	A<3:0>-T = 10 pF (max)				
	$\overline{\text{WREN-T}}$ = 10 pF				
	$\overline{\text{RDEN-T}}$ = 10 pF (max)				
	D<3:0>-E = 10 pF (max)[13]				
	A<3:0>-E = 8 pF				
	$\overline{\text{WREN-E}}$ = 8 pF				
	$\overline{\text{RDEN-E}}$ = 8 pF				
R_{in}	Real input impedance[14]				
	D<3:0>-T = 0Ω				
	A<3:0>-T = 0Ω				
	$\overline{\text{WREN-T}}$ = 0Ω				
	$\overline{\text{RDEN-T}}$ = 0Ω				
	D<3:0>-E = 0Ω				
	A<3:0>-E = 0Ω				
	$\overline{\text{WREN-E}}$ = 0Ω				
	$\overline{\text{RDEN-E}}$ = 0Ω				

[1]$\overline{\text{WREN-T}}$ and $\overline{\text{WREN-E}} = V_{IH}$, $\overline{\text{RDEN-T}} = V_{IL}$, A<3:0>-E are stable.

[2]$\overline{\text{WREN-T}}$ and $\overline{\text{WREN-E}} = V_{IH}$, $\overline{\text{RDEN-E}} = V_{IL}$, A<3:0>-T are stable.

[3]$\overline{\text{WREN-T}}$ and $\overline{\text{RDEN-E}} = V_{IH}$, $\overline{\text{RDEN-T}}$ and $\overline{\text{WREN-E}} = V_{IL}$, A<3:0>-T and A<3:0>-E are stable with the same address.

[4]$\overline{\text{WREN-T}}$ and $\overline{\text{RDEN-E}} = V_{IL}$, $\overline{\text{RDEN-T}}$ and $\overline{\text{WREN-E}} = V_{IH}$, A<3:0>-E and A<3:0>-T are stable with the same address.

[5]$\overline{\text{WREN-T}}$ and $\overline{\text{RDEN-E}} = V_{IH}$, $\overline{\text{RDEN-T}} = V_{IL}$, D<3:0>-E are stable, A<3:0>-T and A<3:0>-E are stable with the same address.

[6]$\overline{\text{WREN-E}}$ and $\overline{\text{RDEN-T}} = V_{IH}$, $\overline{\text{RDEN-E}} = V_{IL}$, D<3:0>-T are stable and A<3:0>-T and A<3:0>-E are stable with the same address.

[7]Setup times, hold times, and input pulse widths are specified as minimum values and represent the requirements imposed by the device on the input signal timing relationships.

[8]$\overline{\text{RDEN-T}} = V_{IH}$.

[9]$\overline{\text{RDEN-E}} = V_{IH}$.

[10]For D<3:0>-T, the rise time is measured from 0.8 to 2.0 volts. For D<3:0>-E, the rise time is measured from –1.5 to –1.1 volts.

[11]For D<3:0>-T, the fall time is measured from 2.0 to 0.8 volts. For D3:0-E, the fall time is measured from –1.1 to –1.5 volts.

[12]Capacitance is measured at the D<3:0>-T pins with the three-state buffers disabled. The requirement specified must be met for an V_{IH} applied to the $\overline{\text{RDEN-T}}$ input that meets the requirements of Table 2.

[13]Capacitance is measured at the D<0:3>-E pins with the emitter follower outputs in the off state. The requirement specified must be met for any V_{IH} applied to the $\overline{\text{RDEN-E}}$ input that meets the requirements of Table 2.

[14]This parameter specifies the real portion of the input impedance to be positive for all frequencies that will ensure that the device will not cause oscillations in a system environment.

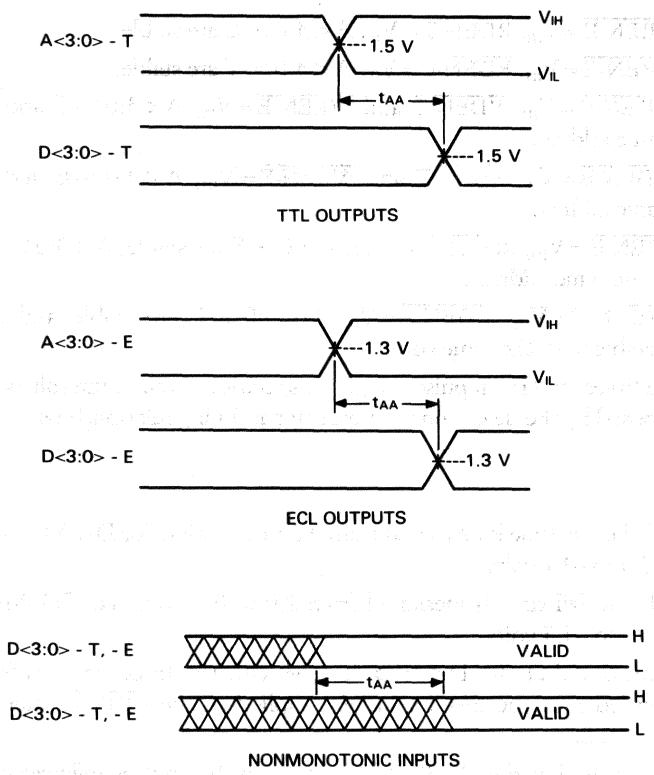
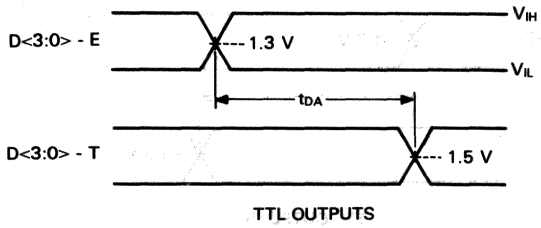
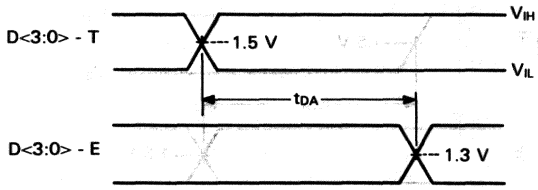


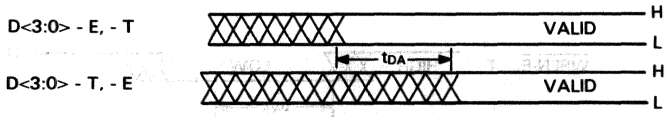
Figure 5 • DC022 Address Access Timing



TTL OUTPUTS



ECL OUTPUTS



NONMONOTONIC INPUTS

Figure 6 ▪ DC022 Data Access Timing

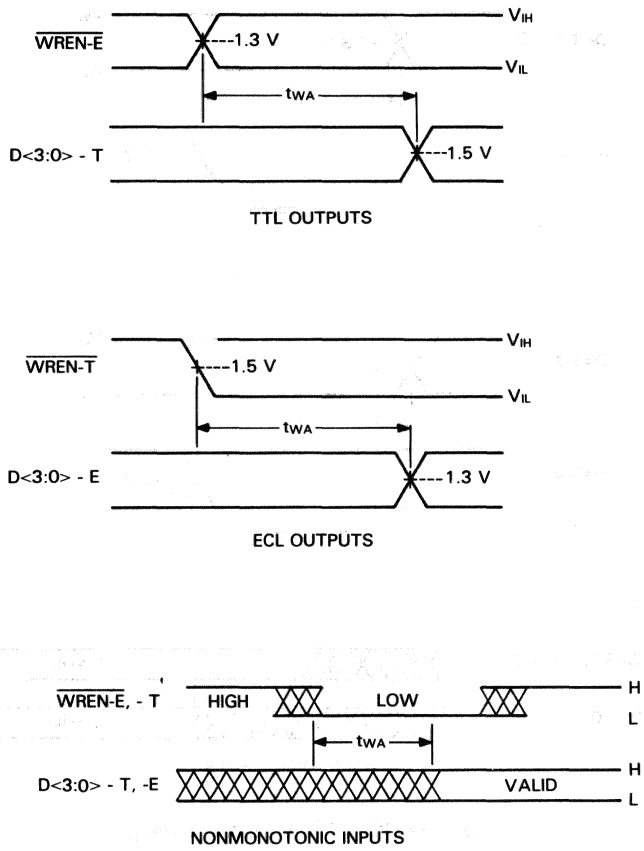


Figure 7 • DC022 Write Access Timing

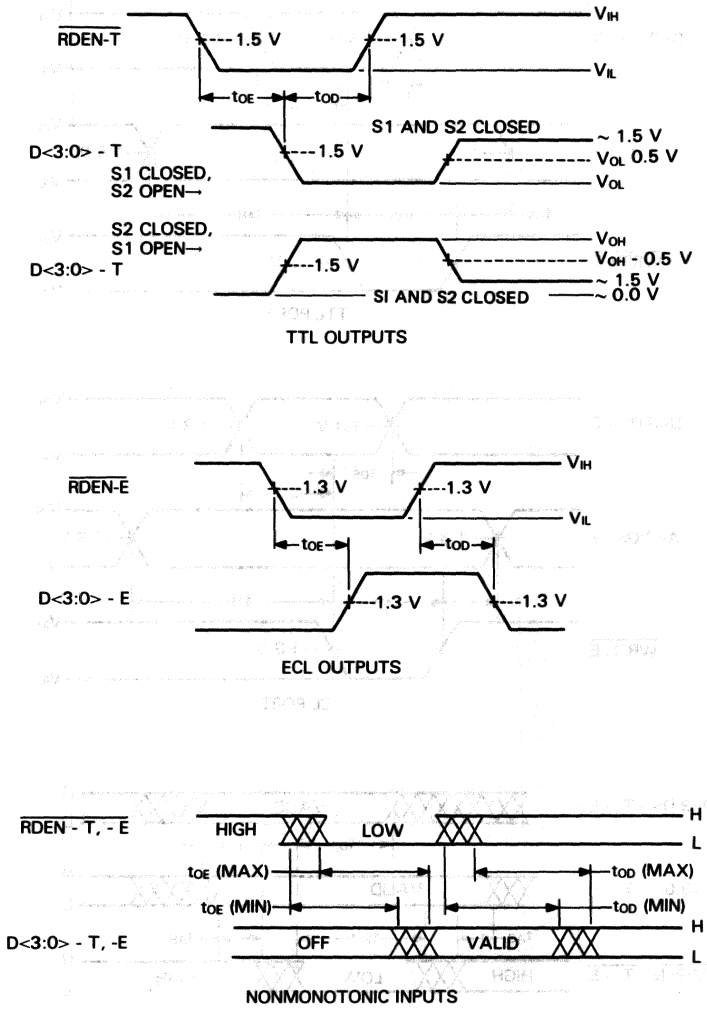


Figure 8 ▪ DC022 Enable and Disable Signal Timing

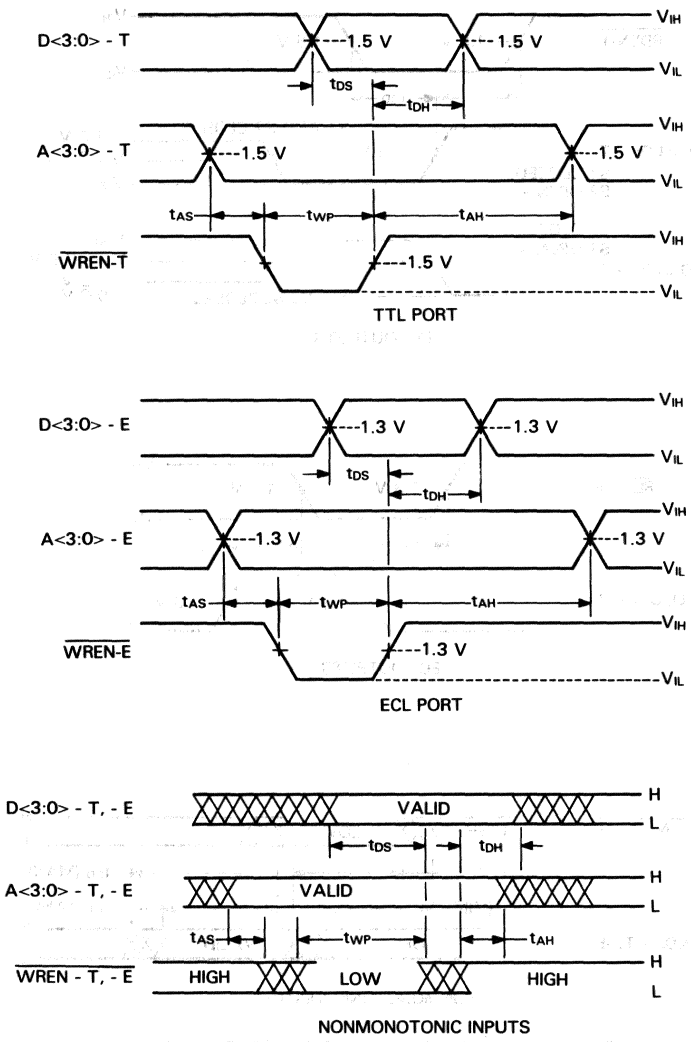
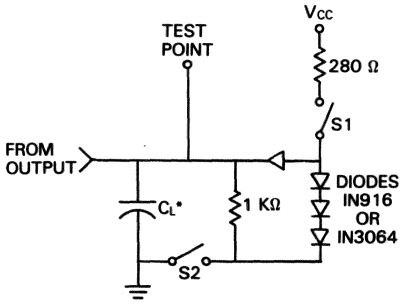
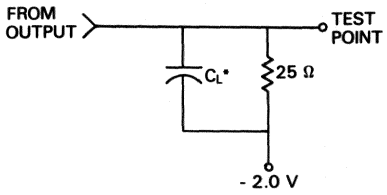


Figure 9 ▪ DC022 Write Sequence Timing



* C_L INCLUDES PROBE CAPACITANCE

Load A TTL Circuit



* C_L INCLUDES PROBE CAPACITANCE

Load B TTL Circuit

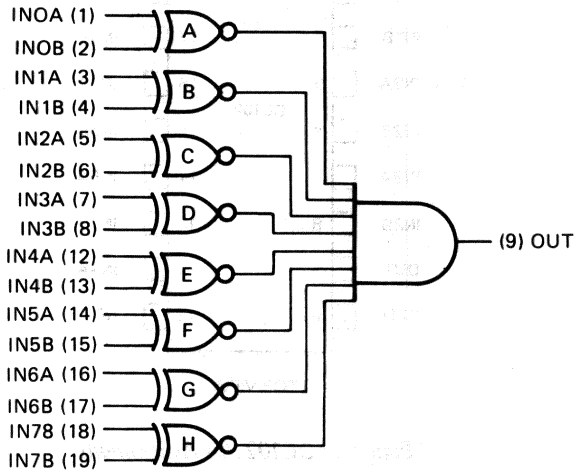
Figure 10 ▪ DC022 ac Output Load Circuits

Features

- High-speed, 8-channel equals checker function
- TTL compatible inputs and output

Description

The DC102 equals checker, contained in a 20-pin dual-inline package (DIP), provides eight dual-input channels to compare data in pairs for equality. The equals checker output is a low level when any one or more input pairs are not at equal logic levels. Figure 1 is a simplified logic diagram of the DC102.



NOTE:
Numbers in ( ) Denote Terminal Numbers.

Figure 1 • DC102 Simplified Logic Diagram

The DC102 is provided in the following two variations. Refer to Table 3 for the ac parameters for each variation.

- DC102: Digital part no. 1913888-00 and 1913888-01

Pin and Signal Description

The input and output signals and the power and ground connections for the DC102 20-pin DIP are shown in Figure 2 and summarized in Table 1.

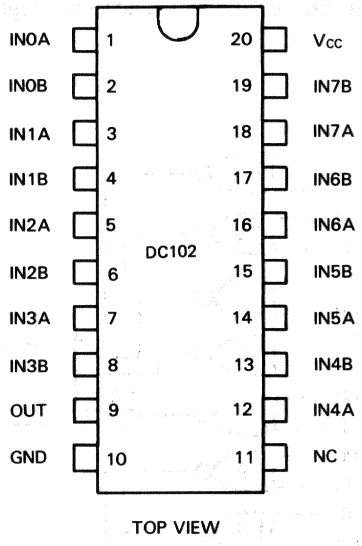


Figure 2 ▪ DC102 Pin Assignments

Table 1 • DC102 Pin and Signal Summary

Pin	Signal	Input/Output*	Definition/Function
1,2	IN0A,IN0B	input	Input A0 and B0—Dual data inputs to gate A
3,4	IN1A,IN1B	input	Input A1 and B1—Dual data inputs to gate B
5,6	IN2A,IN2B	input	Inputs A2 and B2—Dual data inputs to gate C
7,8	IN3A,IN3B	input	Inputs A3 and B3—Dual data inputs to gate D
12,13	IN4A,IN4B	input	Inputs A4 and B4—Dual data inputs to gate E
14,15	IN5A,IN5B	input	Inputs A5 and B5—Dual data inputs to gate F
16,17	IN6A,IN6B	input	Inputs A6 and B6—Dual data inputs to gate G
18,19	IN7A,IN7B	input	Input A7 and B7—Dual data inputs to gate H
9	OUT	output**	Equals function output
11	NC		No connection
10	GND	input	Ground—Ground reference
20	V	input	Voltage—Power supply voltage

*All signals are TTL levels
**Open collector

Functional Description

Figure 3 is a functional symbol of the DC102 that shows the input signal groups.

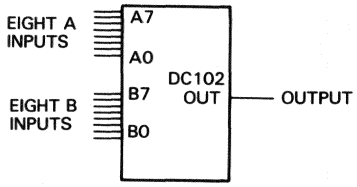


Figure 3 • DC102 Functional Symbol

Specifications

The mechanical, electrical, and environmental characteristics and specifications for the DC102 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise. Refer to Digital specification A-PS-2100002-GS for the general specifications of integrated circuits.

- Power supply voltage (V_{CC}): 5.0 V ±5%
- Temperature range (T_A): 0°C to 70°C

Mechanical Configuration

The physical dimensions of the DC102 20-pin DIP are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Supply voltage (V_{CC}): 7.0 V
- Operating temperature (T_A): 0°C to 70°C
- Relative humidity: 0 to 95% (noncondensing)

Recommended Operating Conditions

- Supply voltage (V_{CC}): 5 V ±5%
- Supply current (I_{CC}): 160 mA

dc Electrical Characteristics

The dc electrical parameters of the DC102 for the operating voltage and temperature ranges specified are listed in Table 2. Refer to Appendix C for the test circuits configurations referenced in the tables.

Table 2 • DC102 dc Input and Output Parameters

Parameter	Symbol	Test Condition	Requirements Min.	Requirements Max.	Units	Test Circuit
Low-level input voltage	V_{IL}		—	0.8	V	C1,C2
High-level input voltage	V_{IH}		2.0		V	C1
Low-level input current	I_{IL}	$V_{IL}=0.4$ V $V_{CC}=5.25$ V	0.1	−0.2	mA	C4
High-level input current	I_{IH}	$V_{IH}=2.7$ V $V_{CC}=5.25$ V	−0.1	0.2	mA	C4
Input current at maximum input voltage	I_I	$V_{IH}=7$ V $V_{CC}=5.25$ V	—	10	mA	C4
Low-level output voltage	V_{OL}	$V_{CC}=4.75$ V $I_{OL}=150$ mA Pin 1 = 0.8 V All other inputs = 2.0 V	—	0.55	V	C2
		$V_{CC}=4.75$ V $I_{OL}=120$ mA Pin 1 = 0.8 V All other inputs = 2.0 V	—	0.5	V	C2
Output reverse current	I_{OH}	$V_{CC}=4.75$ V $V_{OH}=3.75$ V All inputs = 2.0 V	—	0.1	mA	C1
Input clamp voltage	V_I	$V_{CC}=4.75$ V $I_I=-18$ mA	—	−1.2	V	C3
Supply current	I_{CC}	$V_{CC}=5.25$ V Pin 1 = V_{IH}, all other inputs = V_{IL}.*	—	160	mA	C7

*Connect 33Ω resistor from V_{CC} to OUT (Pin 9)

ac Electrical Characteristics

The voltage waveforms and propagation delays symbols for the input and output signals are shown in Figure 4 and defined in Table 3. Refer to Figure 5 for the load circuit used in measuring the propagation delays. The following test conditions apply to the ac propagation delay measurements.

- V_{IL} = 0 V to 0.8 V, V_{IH} = 2.0 V to 3.5 V for inputs not being tested.
- The DC102 must meet the speed requirements for the specified input voltage range and performance must not be degraded by momentary negative voltage spikes on the inputs.
- The nominal test conditions are V_{in} (0) = 0 V, V_{in} (1) = 3.0 V, V_{IL} = 0.5 V and V_{IH} = 2.7 V.

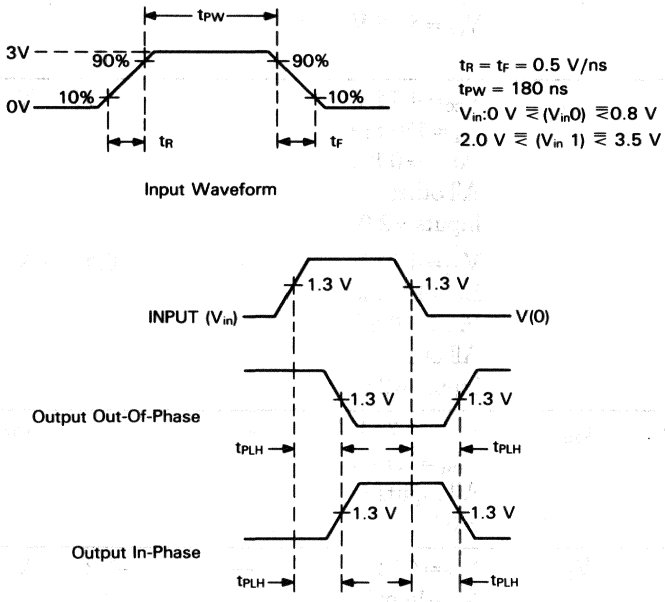


Figure 4 • DC102 ac Signal Timing Delay

Table 3 • DC102 ac Propagation Delay

Symbol	Definition	Requirements 1913888-00* Min.	Max.	1913888-01** Min.	Max.	Units
t_{PLH}	Low-to-high level output	—	20	—	13	ns
t_{PHL}	High-to-low level output	—	20	—	13	ns

*$C_L = 150pF$
**$C_L = 100pF$

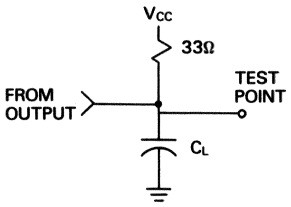


Figure 5 • DC102 ac Load Circuit

• Features

- Separate transmit and receive frequency outputs
- Program selectable baud rates from 50 to 38,400
- External TTL clock or crystal inputs
- TTL compatible inputs and outputs

• Description

The DC301 dual baud-rate generator, contained in a 18-pin dual-inline package (DIP), provides a transmit and receive output to control the transfer of serial line information. Figure 1 is a simplified block diagram of the DC301 baud rate generator.

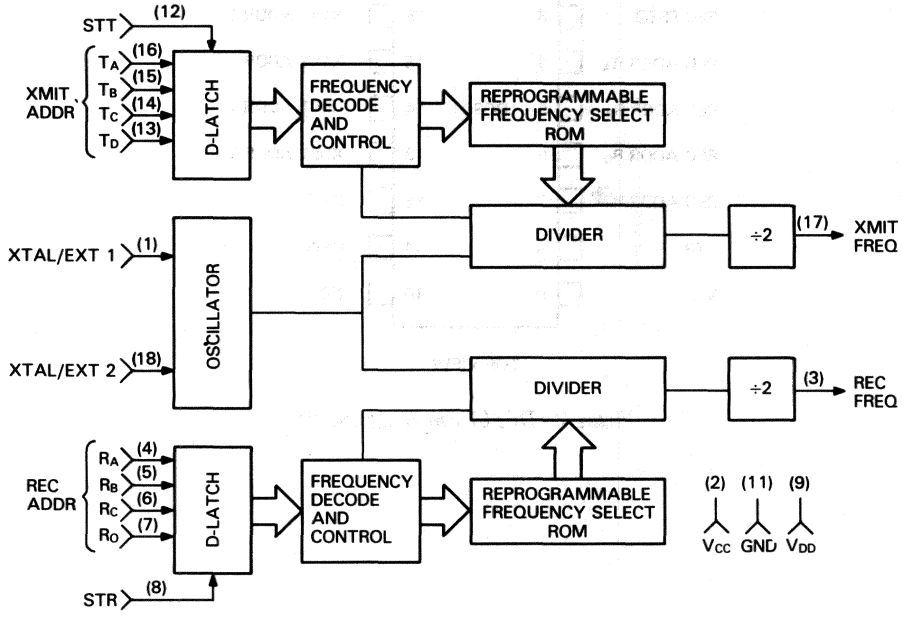


Figure 1 • DC301 Simplified Block Diagram

The DC301 consists of a crystal-controlled oscillator, two programmable frequency dividers, and two divide-by-two counters. The input frequency to the oscillator can be supplied by an external TTL generator or controlled by a crystal connected directly to the DC301. Separate programmable inputs are provided for both the transmit and receive channels. These inputs select their respective baud rates that may deviate depending on the frequency of the crystal of TTL input. The frequency decode and control logic is ROM-based and controls the functions of the divider.

▪ Pin and Signal Description

The input and output signals and the power and ground connections for the DC301 18-pin DIP are shown in Figure 2 and summarized in Table 1.

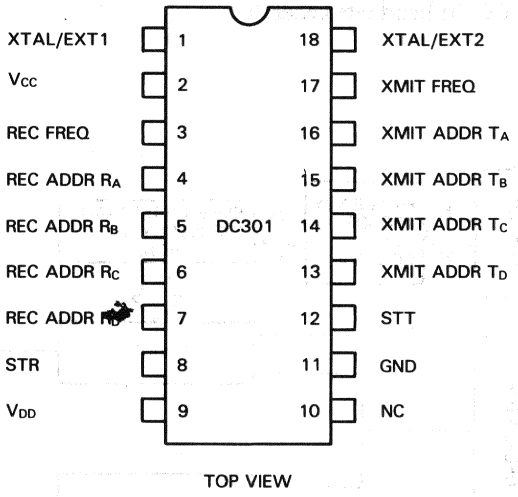


Figure 2 ▪ DC301 Pin Assignments

Table 1 • DC301 Pin and Signal Summary

Pin	Signal	Input/Output	Definition/Function
1	XTAL/EXT1	input[1]	Crystal/external 1—Connects to one lead of the crystal or to a TTL input.
18	XTAL/EXT2	input[1]	Crystal/external 2—Connects to the other lead of the crystal, the opposite polarity a TTL input, or not connected.
17	XMIT FREQ	output[2]	Transmit frequency (f_T)—The frequency output selected by the transmitter address.
3	REC FREQ	output[2]	Receive frequency(f_R)—The frequency output selected by the receiver address.
4-7	REC ADDR R_A, R_B,R_C,R_D	input[2]	Receiver address—Four receiver address inputs that select the output frequency of the receiver (RECV OUT).
8	STR	input[2]	Strobe receiver address—Asserted to load the receiver address into the receiver address register.
16-13	XMIT ADDR T_A, T_B,T_C,T_D	input[2]	Transmit address—Four transmit address inputs that select the output frequency of the transmitter (XMIT OUT).
12	STT	input[2]	Strobe transmitter address—Asserted to load the transmitter address into the transmitter address register.
10	NC		No connection
11	GND	input	Ground—Common ground reference
2	V_{CC}	input	Voltage—Power supply 5 V connection
9	V_{DD}	input	Voltage—Power supply 12 V connection

[1]TTL levels or crystal input
[2]TTL level

Frequency Selection

The DC301 is available in several variations depending on the baud rates required and the external clock connections as listed in Table 2.

Table 2 • DC301 Electrical Variations

Digital Part Number	External Clock	Baud Rates	Power Supplies
2112623-00	TTL	50 to 19,200 (Table 3)	12 V, 5 V or 5 V only
2112623-01	TTL or crystal	50 to 19,200 (Table 3)	12 V, 5 V or 5 V only
2112623-02	TTL	50 to 19,200 (Table 4)	12 V, 5 V or 5 V only
2112623-03	TTL	50 to 10,200 (Table 5)	12 V, 5 V or 5 V only
2112623-04	TTL	100 to 38,400 (Table 6)	12 V, 5 V or 5 V only
2112623-05	TTL	50 to 19,200 (Table 3)	5 V only

Tables 3 through 6 list the transmit and receive addresses required to select the desired baud rates for the crystal frequency specified. The tables also contain the characteristics of the DC301 signals for the selected baud rates.

Table 3 • DC301 Baud Rate Selection and Characteristics (Crystal Frequency 5.0688 MHz)

Transmit/Receive Address D	C	B	A	Baud Rates	Theoretical Frequency 16X Clock (kHz)	Actual Frequency 16X Clock (kHz)	Percent Error	Duty Cycle %	Divisor
0	0	0	0	50	0.8	0.8	—	50/50	6336
0	0	0	1	75	1.2	1.2	—	50/50	4224
0	0	1	0	110	1.76	1.76	—	50/50	2880
0	0	1	1	134.5	2.152	2.1523	0.016	50/50	2355
0	1	0	0	150	2.4	2.4	—	50/50	2112
0	1	0	1	300	4.8	4.8	—	50/50	1056
0	1	1	0	600	9.6	9.6	—	50/50	528
0	1	1	1	1200	19.2	19.2	—	50/50	264
1	0	0	0	1800	28.8	28.8	—	50/50	176
1	0	0	1	2000	32.0	32.081	0.253	50/50	158
1	0	1	0	2400	38.4	38.4	—	50/50	132
1	0	1	1	3600	57.6	57.6	—	50/50	88
1	1	0	0	4800	76.8	76.8	—	50/50	66
1	1	0	1	7200	115.2	115.2	—	50/50	44
1	1	1	0	9600	153.6	153.6	—	48/52	33
1	1	1	1	19200	307.2	316.8	3.125	50/50	16

Table 4 • DC301 Baud Rate Selection and Characteristics (Crystal Frequency 2.76480 MHz)

Transmit/Receive Address D	C	B	A	Baud Rate	Theoretical Frequency 16X Clock (kHz)	Actual Frequency 16X Clock (kHz)	Percent Error	Duty Cycle %	Divisor
0	0	0	0	50	0.8	0.8	0	50/50	3456
0	0	0	1	75	1.2	1.2	0	50/50	2304
0	0	1	0	110	1.76	1.76	-.006	50/50	1571
0	0	1	1	134.5	2.152	2.1523	-.019	50/50	1285
0	1	0	0	150	2.4	2.4	0	50/50	1152
0	1	0	1	200	3.2	3.2	0	50/50	864
0	1	1	0	300	4.8	4.8	0	50/50	576
0	1	1	1	600	9.6	9.6	0	50/50	288
1	0	0	0	1200	19.2	19.2	0	50/50	144
1	0	0	1	1800	28.8	28.8	0	50/50	96
1	0	1	0	2000	32.0	32.149	+.465	50/50	86
1	0	1	1	2400	38.4	38.4	0	50/50	72
1	1	0	0	3600	57.6	57.6	0	50/50	48
1	1	0	1	4800	76.8	76.8	0	50/50	36
1	1	1	0	9600	153.6	153.6	0	50/50	18
1	1	1	1	19200	307.2	307.2	0	44/56	9

Table 5 • DC301 Baud Rate Selection and Characteristics (Crystal Frequency 6.01835 MHz)

Transmit/Receive Address D	C	B	A	Baud Rate	Theoretical Frequency 16X Clock (kHz)	Actual Frequency 16X Clock (kHz)	Percent Error	Duty Cycle %	Divisor
0	0	0	0	50	0.8	0.8	0	50/50	7523
0	0	0	1	75	1.2	1.20	0	50/50	5015
0	0	1	0	110	1.76	1.76	0	50/50	3420
0	0	1	1	134.5	2.152	2.152	0	50/50	2797
0	1	0	0	150	2.4	2.4	0	50/50	2508
0	1	0	1	200	3.2	3.2	0	50/50	1881
0	1	1	0	300	4.8	4.8	0	50/50	1254
0	1	1	1	600	9.6	9.6	0	50/50	627
1	0	0	0	1200	19.2	19.23	+.14	50/50	313
1	0	0	1	1800	28.8	28.8	0	50/50	209
1	0	1	0	2000	32.0	32.0	0	50/50	188
1	0	1	1	2400	38.4	38.33	–.17	50/50	157
1	1	0	0	3600	57.6	57.6	0	50/50	104
1	1	0	1	4800	76.8	76.8	0	50/50	78
1	1	1	0	9600	153.6	153.6	0	50/50	39
1	1	1	1	19200	307.2	307.2	0	50/50	20

Table 6 • DC301 Baud Rate Selection and Characteristics (Crystal Frequency 5.52960 MHz)

Transmit/Receive Address D	C	B	A	Baud Rate	Theoretical Frequency 16X Clock (kHz)	Actual Frequency 16X Clock (kHz)	Percent Error	Duty Cycle %	Divisor
0	0	0	0	100	1.6	1.6	0	50/50	3456
0	0	0	1	150	2.4	2.4	0	50/50	2304
0	0	1	0	220	3.52	3.5197	–.006	50/50	1571
0	0	1	1	269	4.304	4.3032	–.019	50/50	1285
0	1	0	0	300	4.8	4.8	0	50/50	1152
0	1	0	1	400	6.4	6.4	0	50/50	864
0	1	1	0	600	9.6	9.6	0	50/50	576
0	1	1	1	1200	19.2	19.2	0	50/50	288
1	0	0	0	2400	38.4	38.4	0	50/50	144
1	0	0	1	3600	57.6	57.6	0	50/50	96
1	0	1	0	4000	64.0	64.298	+.466	50/50	86
1	0	1	1	4800	76.8	76.8	0	50/50	72
1	1	0	0	7200	115.2	115.2	0	50/50	48
1	1	0	1	9600	153.6	153.6	0	50/50	36
1	1	1	0	19200	307.2	307.2	0	50/50	18
1	1	1	1	38400	614.8	614.8	0	44/56	9

• Specifications

The mechanical, electrical, and environmental characteristics and specifications for the DC301 are described in the following paragraphs. The test conditions for the electrical values are as follows unless specified otherwise. Refer to Digital specification A-PS-2100002-GS for the general specifications of integrated circuits.

- Power supply voltage (V_{CC}): 5.0 V ±5%
- Power supply voltage (V_{DD}): 12 V ±5%
- Temperature range (T_A): 0°C to 70°C

Mechanical Configuration

The physical dimensions of the DC301 18-pin DIP are contained in Appendix E.

Absolute Maximum Ratings

Stresses greater than the absolute maximum ratings may cause permanent damage to the device. Exposure to the absolute maximum ratings for extended periods may adversely affect the reliability of the device.

- Positive voltage on any pin with respect to ground (V_{CC}): 20 V
- Negative voltage on any pin with respect to ground (V_{CC}): –0.3 V
- Operating temperature (T_A): 0°C to 70°C
- Relative humidity: 0 to 95% (noncondensing)

Recommended Operating Conditions

- Supply voltage (V_{CC}): 5 V ±5%
- Supply voltage (V_{DD}): 12 V ±5%
- Total power supply current: 105 mA

dc Electrical Characteristics

The dc electrical parameters of the DC301 for the operating voltage and temperature ranges specified are listed in Table 7. Refer to Appendix C for the test circuit configurations referenced in the table.

Table 7 • DC301 dc Input and Output Parameters

Parameter	Symbol	Test condition	Requirements Min:	Requirements Max:	Units	Test Circuit
Low-level input voltage	V_{IL}	Excluding crystal input levels	—	0.8	V	A,B
High-level input voltage	V_{IH}		3.5	V_{CC}	V	A,B
Low-level output voltage	V_{OL}	$I_{OL}=3.2$ mA	—	0.4	V	B
High-level output voltage	V_{OH}	$I_{OH}=100$ µA	2.4	—	V	A
Input current	I_{IH}	$V_{in}=5$ V, Excluding crystal input levels	—	10	µA	D
Input capacitance (all inputs)	C_{in}	$V_{in}=2.0$ V Excluding crystal inputs	—	10	pF	
External input load	1 TTL load		—	5.0		
Power supply current	I_{CC}	$V_{CC}=5.25$ V $V_{in}=$open $V_{out}=$open	—	75	mA	G
	I_{DD}	$V_{DD}=12.6$ V*	—	30	mA	
Total power dissipation			—	500	mW	

*Test not required when only 5 Vdc supply is used.

ac Electrical Characteristics

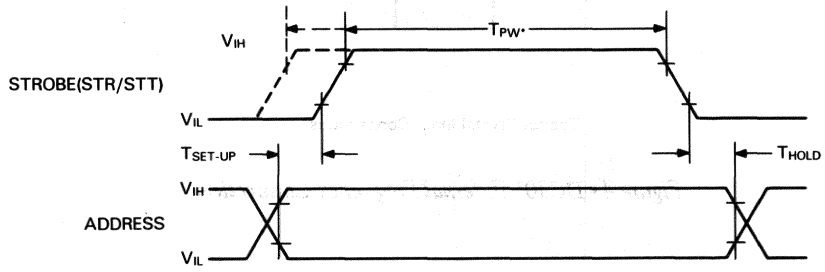
The input and output signal timing for the DC301 is shown in Figure 3. and the parameters are defined in Table 8. Refer to Appendix D for the standard TTL input and output voltage waveform parameters used for measuring the signal timing.

Table 8 ▪ DC301 ac Timing Parameters

Symbol	Definition	Requirements[1] Min.	Max.	Units
PW	Receiver strobe pulse width[2]	150	dc	ns
	Transmitter strobe pulse width[2]	150	dc	ns
t_{setup}	Address input setup time[2]	250	—	ns
t_{hold}	Address input hold time	50	—	ns
	Strobe to new frequency delay	—	2.5	µs

[1]$C_L = 30$ pF under normal operating conditions. Unit(s) shall meet $C_L = 80$ pF with derated conditions.

[2]Input setup time can be decreased to > 0 ns by increasing the minimum strobe width by 50 ns to a total of 200 ns.

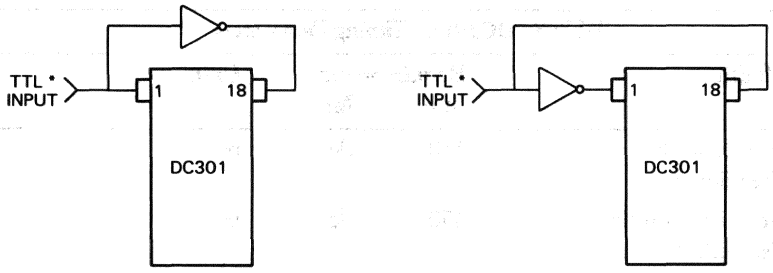


* Address need only be valid during the last TPW, Min time of the input strobe.

Figure 3 ▪ DC301 ac Timing Parameters

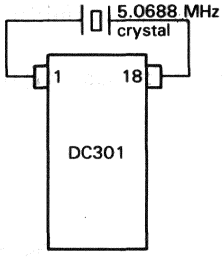
External Frequency

Figure 4 shows typical external circuits used to connect the TTL frequency inputs and crystal-controlled inputs to the DC301.



* totem pole or open collector output

Typical TTL Input Circuit



Typical Crystal Input Connections

Figure 4 ▪ DC301 External Frequency Connections

Instruction Set

This appendix provides a summary of the VAX-11 instructions implemented by the MicroVAX 78032, the floating-point instructions supported by the floating-point unit, and the emulated instructions that are assisted by the MicroVAX 78032's microcode. The standard notation for operand specifiers is

<name> <access type> <data type>

1. Name—A suggestive name for the operand in the context of the instruction. It is the capitalized name of a register or block for implied operands.
2. Access type—A letter denoting the operand specifier access type.
   a = address operand
   b = branch displacement
   m = modified operand (both read and written)
   r = read only operand
   v = if not "Rn," same as address operand, otherwise R[n+1]'R[n]
   w = write only operand
3. Data type—A letter denoting the data type of the operand.
   b = byte
   d = D_floating
   f = F_floating
   g = G_floating
   l = longword
   q = quadword
   v = field (used only in implied operands)
   w = word
   * = multiple longwords (used only in implied operands)
4. Implied operands—Locations that are accessed by the instruction, but not specified in an operand, are denoted by braces { }. The abbreviations for condition codes are
   * = conditionally set/cleared
   – = not affected
   0 = cleared
   1 = set

   The abbreviations for exceptions are
   rsv = reserved operand fault
   iov = integer overflow trap
   idvz = integer divide by zero trap
   fov = floating overflow fault
   fuv = floating underflow fault
   fdvz = floating divide by zero fault
   dov = decimal overflow trap
   ddvz = decimal divide by zero trap
   sub = subscript range trap
   prv = privileged instruction fault

   Opcode values are given in hexadecimal.

• Integer Arithmetic and Logical Instructions

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
58	ADAW1 add.rw, sum.mw	Add aligned word interlocked	*	*	*	*	iov
80	ADDB2 add.rb, sum.mb	Add byte 2-operand	*	*	*	*	iov
C0	ADDL2 add.rl, sum.ml	Add long 2-operand	*	*	*	*	iov
A0	ADDW2 add.rw, sum.mw	Add word 2-operand	*	*	*	*	iov
81	ADDB3 add1.rb, add2.rb, sum.wb	Add byte 3-operand	*	*	*	*	iov
C1	ADDL3 add1.rl, add2.rl, sum.wl	Add long 3-operand	*	*	*	*	iov
A1	ADDW3 add1.rw, add2.rw, sum.ww	Add word 3-operand	*	*	*	*	iov
D8	ADWC add.rl, sum.ml	Add with carry	*	*	*	*	iov
78	ASHL cnt.rb src.rl, dst.wl	Arithmetic shift left	*	*	*	0	iov
79	ASHQ cnt.rb src.rq, dst.wq	Arithmetic shift quad	*	*	*	0	iov
8A	BICB2 mask.rb, dst.mb	Bit clear byte 2-operand	*	*	0	–	
CA	BICL2 mask.rl, dst.ml	Bit clear long 2-operand	*	*	0	–	
AA	BICW2 mask.rw, dst.mw	Bit clear word 2-operand	*	*	0	–	
8B	BICB3 mask.rb, src.rb, dst.wb	Bit clear byte 3-operand	*	*	0	–	
CB	BICL3 mask.rl, src.rl, dst.ml	Bit clear long 3-operand	*	*	0	–	
AB	BICW3 mask.rw, src.rw, dst.mw	Bit clear word 3-operand	*	*	0	–	
88	BISB2 mask.rb, dst.mb	Bit set byte 2-operand	*	*	0	–	
C8	BISL2 mask.rl, dst.ml	Bit set long 2-operand	*	*	0	–	
A8	BISW2 mask.rw, dst.mw	Bit set word 2-operand	*	*	0	–	
89	BISB3 mask.rb, src.rb, dst.mb	Bit set byte 3-operand	*	*	0	–	
C9	BISL3 mask.rl, src.rl, dst.ml	Bit set long 3-operand	*	*	0	–	
A9	BISW3 mask.rw, src.rw, dst.mw	Bit set word 3-operand	*	*	0	–	
93	BITB mask.rb, src.rb	Bit test byte	*	*	0	–	
D3	BITL mask.rl, src.rl	Bit test long	*	*	0	–	
B3	BITW mask.rw, src.rw	Bit test word	*	*	0	–	
94	CLRB dst.wb	Clear byte	0	1	0	–	
D4	CLRL dst.wl	Clear long	0	1	0	–	
7C	CLRQ dst.wq	Clear quad	0	1	0	–	
B4	CLRW dst.ww	Clear word	0	1	0	–	
91	CMPB src1.rb, src2.rb	Compare byte	*	*	0	*	
D1	CMPL src1.rl, src2.rl	Compare long	*	*	0	*	
B1	CMPW src1.rw, src2.rw	Compare word	*	*	0	*	
98	CVTBL src.rb, dst.wl	Convert byte to long	*	*	0	0	
99	CVTBW src.rb, dst.wl	Convert byte to word	*	*	0	0	
F6	CVTLB src.rl, dst.wb	Convert long to byte	*	*	*	0	iov
F7	CVTLW src.rl, dst.ww	Convert long to word	*	*	*	0	iov
33	CVTWB src.rw, dst.wb	Convert word to byte	*	*	*	0	iov
32	CVTWL src.rw, dst.wl	Convert word to long	*	*	0	0	
97	DECB dif.mb	Decrement byte	*	*	*	*	iov
D7	DECL dif.l	Decrement long	*	*	*	*	iov
97	DECW dif.mw	Decrement word	*	*	*	*	iov
86	DIVB2 divr.rb, quo.mb	Divide byte 2-operand	*	*	*	0	iov, idvz
C6[2]	DIVL2 divr.rl, quo.ml	Divide long 2-operand	*	*	*	0	iov, idvz
A6	DIVW2 divr.rw, quo.mw	Divide word 2-operand	*	*	*	0	iov, idvz

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
87	DIVB3 divr.rb, divd.rb, quo.wb	Divide byte 3-operand	*	*	*	0	iov, idvz
C7[2]	DIVL3 divr.rl, divd.rl, quo.wl	Divide long 3-operand	*	*	*	0	iov, idvz
A7	DIVW3 divr.rw, divd.rw, quo.ww	Divide word 3-operand	*	*	*	0	iov, idvz
7B[2]	EDIV divr.rl, divd.rq, quo.wl, rem.wl	Extended divide	*	*	*	0	iov, idvz
7A[1]	EMUL mulr.rl, muld.rl, add.rl, prod.wq	Extended multiply	*	*	0	0	
96	INCB sum.mb	Increment byte	*	*	*	*	iov
D6	INCL sum.ml	Increment long	*	*	*	*	iov
B6	INCW sum.mw	Increment word	*	*	*	*	iov
92	MCOMB src.rb, dst.wb	Move complemented byte	*	*	0	–	
D2	MCOML src.rl, dst.wl	Move complemented long	*	*	0	–	
B2	MCOMW src.rw, dst.ww	Move complemented word	*	*	0	–	
8E	MNEGB src.rb, dst.wb	Move negated byte	*	*	*	*	iov
CE	MNEGL src.rl, dst.wl	Move negated long	*	*	*	*	iov
AE	MNEGW src.rw, dst.ww	Move negated word	*	*	*	*	iov
90	MOVB src.rb, dst.wb	Move byte	*	*	0	–	
D0	MOVL src.rl, dst.wl	Move long	*	*	0	–	
B0	MOVW src.rw, dst.ww	Move word	*	*	0	–	
9A	MOVZBW src.rb, dst.wb	Move zero-extended byte to word	0	*	0	–	
9B	MOVZBL src.rb, dst.wl	Move zero-extended byte to long	0	*	0	–	
3C	MOVZWL src.rw, dst.ww	Move zero-extended word to long	0	*	0	–	
84	MULB2 mulr.rb, prod.mb	Multiply byte 2-operand	*	*	*	0	iov
C4[1]	MULL2 mulr.rl, prod.ml	Multiply long 2-operand	*	*	*	0	iov
A4	MULW2 mulr.rw, prod.mw	Multiply word 2-operand	*	*	*	0	iov
85	MULB3 mulr.rb, muld.rb, prod.mb	Multiply byte 3-operand	*	*	*	0	iov
C5[1]	MULL3 mulr.rl, muld.rl, prod.ml	Multiply long 3-operand	*	*	*	0	iov
A5	MULW3 mulr.rw, muld.rw, prod.mw	Multiply word 3-operand	*	*	*	0	iov
DD	PUSHL src.rl,	Push long	*	*	0	–	
9C	ROTL cnt.rb, src.rl, dst.wl	Rotate long	*	*	0	–	
D9	SBWC sub.rl, dif.ml	Subtract with carry	*	*	*	*	iov
82	SUBB2 sub.rb, dif.mb	Subtract byte 2-operand	*	*	*	*	iov
C2	SUBL2 sub.rl, dif.ml	Subtract long 2-operand	*	*	*	*	iov
A2	SUBW2 sub.rw, dif.mw	Subtract word 2-operand	*	*	*	*	iov
83	SUBB3 sub.rb, min.rb, dif.mb	Subtract byte 3-operand	*	*	*	*	iov
C3	SUBL3 sub.rl, min.rl, dif.ml	Subtract long 3-operand	*	*	*	*	iov
A3	SUBW3 sub.rw, min.rw, dif.mw	Subtract word 3-operand	*	*	*	*	iov
95	TSTB src.rb	Test byte	*	*	0	0	
D5	TSTL src.rl	Test long	*	*	0	0	
B5	TSTW src.rw	Test word	*	*	0	0	
8C	XORB2 mask.rb, dst.mb	Exclusive or byte 2-operand	*	*	0	–	
CC	XORL2 mask.rl, dst.ml	Exclusive or long 2-operand	*	*	0	–	
AC	XORW2 mask.rw, dst.mw	Exclusive or word 2-operand	*	*	0	–	
8D	XORB3 mask.rb, src.rb, dst.wb	Exclusive or byte 3-operand	*	*	0	–	
CD	XORL3 mask.rl, src.rl, dst.wl	Exclusive or long 3-operand	*	*	0	–	
AD	XORW3 mask.rw, src.rw, dst.ww	Exclusive or word 3-operand	*	*	0	–	

1. The MicroVAX 78132 FPU, when present, accelerates execution of these integer multiplication instructions.

2. The MicroVAX 78132 FPU, when present, accelerates execution of these integer division instructions.

• Address Instructions

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
9E	MOVAB src.ab, dst.wl	Move address of byte	*	*	0	–	
DE	MOVAL {=F} src.al, dst.wl	Move address of long	*	*	0	–	
7E	MOVAQ {=D=G} src.aq, dst.wl	Move address of quad	*	*	0	–	
3E	MOVAW src.aw, dst.wl	Move address of word	*	*	0	–	
9F	PUSHAB src.ab, {–(SP).wl}	Push address of byte	*	*	0	–	
DF	PUSHAL {=F} src.al, {–(SP).wl}	Push address of long	*	*	0	–	
7F	PUSHAQ {=D=G} src.aq, {–(SP).wl}	Push address of quad	*	*	0	–	
3F	PUSHAW src.aw, {–(SP).wl}	Push address of word	*	*	0	–	

• Variable-length Bit Field Instructions

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
EC	CMPV pos.rl, size.rb, base.rb, {field.rv}, src.rl	Compare field	*	*	0	*	rsv
ED	CMPZV pos.rl, size.rb, base.vb, {field.rv}, src.rl	Compare zero-extended field	*	*	0	*	rsv
EE	EXTV pos.rl, size.rb, base.vb, {field.rv}, dst.wl	Extract field	*	*	0	–	rsv
EF	EXTZV pos.rl, size.rb, base.vb, {field.rv}, dst.wl	Extract zero-extended field	*	*	0	–	rsv
F0	INSV src.rl, pos.rl, size.rb, base.vb, {field.wv}	Insert field	–	–	–	–	rsv
EB	FFC startpos.rl size.rb, base.vb, {field.rv}, findpos.wl	Find first clear bit	*	*	0	–	rsv
EA	FFS startpos.rl, size.rb, base.vb, {field.rv}, findpos.wl	Find first set bit	*	*	0	–	rsv

• Control Instructions

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
9D	ACBB limit.rb, add.rb, index.mb, displ.bw	Add compare and branch byte	*	*	*	–	iov
F1	ACBL limit.rl, add.rl, index.ml, displ.bw	Add compare and branch long	*	*	*	–	iov
3D	ACBW limit.rw, add.rw, index.mw, displ.bw	Add compare and branch word	*	*	*	–	iov
F3	AOBLEQ limit.rl, index.ml, displ.bb	Add one and branch on less or equal	*	*	*	–	iov
F2	AOBLSS limit.rl, index.ml, displ.bb	Add one and branch on less	*	*	*	–	iov

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
1E	BCC{=BGEQU} displ.bb	Branch on carry clear	–	–	–	–	–
1F	BCS{=BLSSU} displ.bb	Branch on carry set	–	–	–	–	–
13	BEQL{=BEQLU} displ.bb	Branch on equal	–	–	–	–	–
18	BGEQ displ.bb	Branch on greater or equal	–	–	–	–	–
14	BGTR displ.bb	Branch on greater	–	–	–	–	–
1A	BGTRU displ.bb	Branch on greater unsigned	–	–	–	–	–
15	BLEQ displ.bb	Branch on less or equal	–	–	–	–	–
1B	BLEQU displ.bb	Branch on less or equal unsigned	–	–	–	–	–
19	BLSS displ.bb	Branch on less	–	–	–	–	–
12	BNEQ {=BNEQU} displ.bb	Branch on not equal	–	–	–	–	–
1C	BVC displ.bb	Branch on overflow clear	–	–	–	–	–
1D	BVS displ.bb	Branch on overflow set	–	–	–	–	–
E1	BBC pos.rl, base.vb, displ.bb, {field.rv}	Branch on bit clear	–	–	–	–	rsv
E0	BBS pos.rl, base.vb, displ.bb, {field.rv}	Branch on bit set	–	–	–	–	rsv
E5	BBCC pos.rl, base.vb, displ.bb, {field.mv}	Branch on bit clear and clear	–	–	–	–	rsv
E3	BBCS pos.rl, base.vb, displ.bb, {field.mv}	Branch on bit clear and set	–	–	–	–	rsv
E4	BBSC pos.rl, base.vb, displ.bb, {field.mv}	Branch on bit set and clear	–	–	–	–	rsv
E2	BBSS pos.rl, base.vb, displ.bb, {field.mv}	Branch on bit set and set	–	–	–	–	rsv
E7	BBCCI pos.rl, base.vb, dislp.bb, {field.mv}	Branch on bit clear and clear interlocked	–	–	–	–	rsv
E6	BBSSI pos.rl, base.vb, dislp.bb, {field.mv}	Branch on bit set and set interlocked	–	–	–	–	rsv
E9	BLBC src.rl, displ.bb	Branch on low bit clear	–	–	–	–	–
E8	BLBS src.rl, displ.bb	Branch on low bit set	–	–	–	–	–
11	BRB displ.bb	Branch with byte displacement	–	–	–	–	–
31	BRW displ.bw	Branch with word displacement	–	–	–	–	–
10	BSBB displ.bb {–(SP).wl}	Branch to subroutine with byte displacement	–	–	–	–	–
30	BSBW displ.bw {–(SP).wl}	Branch to subroutine with word displacement	–	–	–	–	–
8F	CASEB selector.rb, base.rb, limit.rb, displ.bw-list	Case byte	*	*	0	*	
CF	CASEL selector.rl, base.rl, limit.rl, displ.bw-list	Case long	*	*	0	*	
AF	CASEW selector.rw, base.rw, limit.rw, displ.bw-list	Case word	*	*	0	*	
17	JMP dst.ab	Jump	–	–	–	–	–
16	JSB dst.ab, {–(SP).wl}	Jump to subroutine	–	–	–	–	–
05	RSB {(SP)+.rl}	Return from subroutine	–	–	–	–	–
F4	SOBGEQ index.ml, displ.bb	Subtract one and branch on greater or equal	*	*	*	–	iov
F5	SOBGTR index.ml, displ.bb	Subtract one and branch on greater	*	*	*	–	iov

Procedure Call Instructions

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
FA	CALLG arglist.ab, dst.ab, {–(SP).w*}	Call with general argument list	0	0	0	0	rsv
FB	CALLS numarg.rl, dst.ab, {–(SP).w*}	Call with argument list on stack	0	0	0	0	rsv
04	RET {(SP)+.r*}	Return from procedure	*	*	*	*	rsv

Miscellaneous Instructions

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
B9	BICPSW mask.rw	Bit clear processor status word	*	*	*	*	rsv
B8	BISPSW mask.rw	Bit set processor status word	*	*	*	*	rsv
03	BPT {–(KSP).w*}	Break point fault	0	0	0	0	
00	HALT {–(KSP).w*}	Halt (kernel mode only)	–	–	–	–	prv
0A[1]	INDEX subscript.rl, low.rl, high.rl, size.rl, indexin.rl, indexout.wl	Index calculation	*	*	0	0	sub
DC	MOVPSL dst.wl	Move processor status longword	–	–	–	–	
01	NOP	No operation	–	–	–	–	
BA	POPR mask.rw, ;{(SP)+.r*}	Pop registers	–	–	–	–	
BB	PUSHR mask.rw, ;{–(SP)+.w*}	Push registers	–	–	–	–	
FC	XFC {unspecified oeprands}	Extended function call	0	0	0	0	

Queue Instructions

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
5C	INSQHI entry.ab header.aq	Insert at head of queue, interlocked	0	*	0	*	rsv
5D	INSQTI entry.ab header.aq	Insert at tail of queue, interlocked	0	*	0	*	rsv
0E	INSQUE entry.ab, pred.ab	Insert into queue	*	*	0	*	
5E	REMQHI header.aq, addr.wl	Remove from head of queue, interlocked	0	*	*	*	rsv
5F	REMQTI header.aq, addr.wl	Remove from tail of queue, interlocked	0	*	*	*	rsv
0E	REMQUE entry.ab, addr.wl	Remove from queue	*	*	*	*	

Character String Instructions

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
28	MOVC3 len.rw, srcaddr.ab, dstaddr.ab, {R0-5.wl}	Move character 3-operand	0	1	0	0	
2C	MOVC5 srclen.rw, srcaddr.ab, fill.rb, dstlen.rw, dstaddr.ab, {R0-5.wl}	Move character 5-operand	*	*	0	*	

System Support Instructions

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
BD	CHME param.rw, {–(ySP).w*}	Change mode to executive	0	0	0	0	
BC	CHMK param.rw, {–(ySP).w*}	Change mode to kernel	0	0	0	0	
BE	CHMS param.rw, {–(ySP).w*}	Change mode to supervisor	0	0	0	0	
BF	CHMU param.rw, {–(ySP).w*}	Change mode to user	0	0	0	0	
	Where y = MINU(x.PSL < current_mode <)						
06	LDPCTX {PCB.r*, –(KSP).w*}	Load process context (kernel mode only)	–	–	–	–	rsv, prv
DB	MFPR procreg.rl, dst.wl	Move from processor register (kernel mode only)	*	*	0	-	rsv, prv
DA	MTPR src.rl, procreg.rl	Move to processor register (kernel mode only)	*	*	0	–	rsv, prv
0C	PROBER mode.rb, len.rw, base.ab	Probe read access	0	*	0	–	
0D	PROBEW mode.rb, len.rw, base.ab	Probe write access	0	*	0	–	
02	REI {(SP)+.r*}	Return from exception or interrupt	*	*	*	*	rsv
07	SVPCTX {(SP)+.r*, PCB.w*}	Save process context (kernel mode only)	–	–	–	–	prv

Microcode-assisted Emulated Instructions

The MicroVAX 78032 provides microcode assistance for the emulation of these instructions by system software. The processor processes the operand specifiers, creates a standard argument list, and takes an emulated instruction fault. For a description of emulated instruction processing, refer to the *MicroVAX 78032 User's Guide*.

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
20	ADDP4 addlen.rw, addaddr.ab, sumlen.rw, sumaddr.ab	Add packed 4-operand	*	*	*	0	rsv, dov
21	ADDP6 add1len.rw, add1addr.ab, add2len.rw, add2addr.ab, sumlen.rw, sumaddr.ab	Add packed 6-operand	*	*	*	0	rsv, dov
F8	ASHP cnt.rb, srclen.rw, srcaddr.ab, round.rb, dstlen.rw, dstaddr.ab	Arithmetic shift and round packed	*	*	*	0	rsv, dov
29	CMPC3 len.rw, src1addr.ab, src2addr.ab	Compare character 3-operand	*	*	0	*	
2D	CMPC5 src1len.rw, src1addr.ab, fill.rb, src2len.rw, src2addr.ab	Compare character 5-operand	*	*	0	*	
35	CMPP3 len.rw, src1addr.ab, src2addr.ab	Compare packed 3-operand	*	*	0	0	
37	CMPP4 src1len.rw, src1addr.ab, src2len.rw, src2add.ab	Compare packed 3-operand	*	*	0	*	
0B	CRC tbl.ab, inicrc.rl, strien.rw, stream.ab	Calculate cyclic redundancy check	*	*	0	0	
F9	CVTLP src.rl, dstlen.rw, dstaddr.ab	Convert long to packed	*	*	*	0	rsv, dov
36	CVTPL srclen.rw, srcaddr.ab, dst.wl	Convert packed to long	*	*	*	0	rsv, iov
08	CVTPS, srclen.rw, srcaddr.ab, dstlen.rw, dstaddr.ab	Convert packed to leading separate	*	*	*	0	rsv, dov
09	CVTSP, srclen.rw, srcaddr., dstlen.rw, dstaddr.ab	Convert leading separate to packed	*	*	*	0	rsv, dov

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
24	CVTPT srclen.rw, srcaddr.ab, tbladdr.ab, dstlen.rw, dstaddr.ab	Convert packed to trailing	*	*	*	0	rsv, dov
26	CVTTP srclen.rw, srcaddr.ab, tbladdr.ab, dstlen.rw, dstaddr.ab	Convert packed to trailing	*	*	*	0	rsv, dov
27	DIVP divrien.rw, divraddr.ab, divdien.rw, quolen.rw, quoaddr.ab	Divide packed	*	*	*	0	rsv, dov, ddvz
38	EDITPC srclen.rw, srcaddr.ab, pattern.ab, dstaddr.ab	Edit packed to character string	*	*	*	*	rsv, dov
3A	LOCC char.rb, len.rw, addr.ab	Locate character	0	*	0	0	
39	MATCHC objlen.rw, objaddr.ab, srclen.rw, srcaddr.ab	Match characters	0	*	0	0	
34	MOVP len.rw, srcaddr.ab, dstaddr.ab	Move packed	*	*	0	0	
2E	MOVTC srclen.rw, srcaddr.ab, fill.rb, tbladdr.ab, dstlen.rw, dstaddr.ab	Move translated characters	*	*	0	*	
2F	MOVTUC srclen.rw, srcaddr.ab, esc.rb, tbladdr.ab, dstlen.rw, dstaddr.ab	Move translated until character	*	*	*	*	
25	MULP mulrien.rw, mulraddr.ab, muldlen.rw, muldaddr.ab, prodlen.rw, prodaddr.ab	Multiply packed	*	*	*	0	rsv, dov
2A	SCANC len.rw, addr.ab, tbladdr.ab, mask.rb	Scan for character	0	*	0	0	
3B	SKPC char.rb, len.rw, addr.ab	Skip character	0	*	0	0	
2B	SPANC len.rw, len.rw, tbladdr.ab, mask.rb	Scan characters	0	*	0	0	
22	SUBP4 sublen.rw, subaddr.ab, diflen.rw, difaddr.ab	Subtract packed 4-operand	*	*	*	0	rsv, dov
23	SUBP6 sublen.rw, subaddr.ab, minlen.rw, minaddr.ab, diflen.rw, difaddr.ab	Subtract packed 6-operand	*	*	*	0	rsv, dov

• MicroVAX 78032 Foating-point Instructions

These instructions are implemented in hardware if the optional MicroVAX 78132 floating-point unit is present in the system. They must be software emulated if the MicroVAX 78132 is not included.

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
06F	ACBD limit.rd, add.rd, index.md	Add compare and branch D_floating	*	*	0	–	rsv, fov, fuv
04F	ACBF limit.rf, add.rf, index.rf	Add compare and branch F_floating	*	*	0	–	rsv, fov, fuv
4FFD	ACBG limit.rg, add.rg, index.mg	Add compare and branch G_floating	*	*	0	–	rsv, fov, fuv
060	ADDD2 add.rd, sum.md	Add D_floating 2-operand	*	*	0	0	rsv, fov, fuv
040	ADDF2 add.rf, sum.mf	Add F_floating 2-operand	*	*	0	0	rsv, fov, fuv
40FD	ADDG2 add.rg, sum.mg	Add G_floating 2-operand	*	*	0	0	rsv, fov, fuv
061	ADDD3 add1.rd, add2.rd, sum.wd	Add D_floating 3-operand	*	*	*	0	rsv, fov, fuv
041	ADDF3 add1.rf, add2.rf, sum.wf	Add F_floating 3-operand	*	*	*	0	rsv, fov, fuv
41FD	ADDG3 add1.rg, add2.rg, sum.wg	Add G_floating 3-operand	*	*	*	0	rsv, fov, fuv

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
071	CMPD src1.rd, src2.rd	Compare D_floating	*	*	0	0	rsv
051	CMPF src1.rf, src2.rf	Compare F_floating	*	*	0	0	rsv
51FD	CMPG src1.rg, src2.rg	Compare G_floating	*	*	0	0	rsv
06C	CVTBD src.rb, dst.wd	Convert byte to D_floating	*	*	0	0	
04C	CVTBF src.rb, dst.wf	Convert byte to F_floating	*	*	0	0	
4CFD	CVTBG src.rb, dst.wg	Convert byte to G_floating	*	*	0	0	
068	CVTDB src.rb, dst.wb	Convert D_floating to byte	*	*	*	0	rsv, iov
076	CVTDF src.rd, dst.wf	Convert D_floating to F_float	*	*	0	0	rsv, fov
06A	CVTDL src.rd, dst.wl	Convert D_floating to long	*	*	*	0	rsv, iov
069	CVTDW src.rd, dst.ww	Convert D_floating to word	*	*	*	0	rsv, iov
048	CVTFB src.rf, dst.wb	Convert F_floating to byte	*	*	*	0	rsv, iov
056	CVTFD src.rf, dst.wg	Convert F_floating to D_float	*	*	0	0	rsv
99FD	CVTFG src.rf, dst.wg	Convert F_floating to G_float	*	*	0	0	rsv
04A	CVTFL src.rf, dst.wl	Convert F_floating to long	*	*	*	0	rsv, iov
049	CVTFW src.rf, dst.ww	Convert F_floating to word	*	*	*	0	rsv, iov
48FD	CVTGB src.rg, dst.wb	Convert G_floating to byte	*	*	*	0	rsv, iov
33FD	CVTGF src.rg, dst.wf	Convert G_floating to F_float	*	*	0	0	rsv, fov, fuv
4AFD	CVTGL src.rg, dst.wl	Convert G_floating to long	*	*	*	0	rsv, iov
49FD	CVTGW src.rg, dst.ww	Convert G_floating to word	*	*	*	0	rsv, iov
06E	CVTLD src.rl, dst.wb	Convert long to D_floating	*	*	0	0	
04E	CVTLF src.rl, dst.wf	Convert long to F_floating	*	*	0	0	
4EFD	CVTLG src.rl, dst.wg	Convert long to G_floating	*	*	0	0	
06D	CVTWD src.rw, dst.wd	Convert word to D_floating	*	*	0	0	
04D	CVTWF src.rw, dst.wf	Convert word to F_floating	*	*	0	0	
4DFD	CVTWG src.rw, dst.wg	Convert word to G_floating	*	*	0	0	
06B	CVTRDL src.rd, dst.wl	Convert rounded D_floating to long	*	*	0	0	rsv, iov
04B	CVTRFL src.rf, dst.wl	Convert rounded F_floating to long	*	*	*	0	rsv, iov
4BFD	CVTRGL src.rg, dst.wl	Convert rounded G_floating to long	*	*	*	0	rsv, iov
066	DIVD2 divr.rd, quo.md	Divide D_floating 2-operand	*	*	0	0	rsv fov fuv fdvz
046	DIVF2 divr.rf, quo.mf	Divide F_floating 2-operand	*	*	0	0	rsv fov fuv fdvz
46FD	DIVG2 divr.rg, quo.mg	Divide G_floating 2-operand	*	*	0	0	rsv fov fuv fdvz
067	DIVD3 divr.rd, divr.rd, quo.wd	Divide D_floating 3-operand	*	*	0	0	rsv fov fuv fdvz
047	DIVD3 divr.rf, divr.rf, quo.wf	Divide F_floating 3-operand	*	*	0	0	rsv fov fuv fdvz
47FD	DIVD3 divr.rg, divr.rg, quo.wg	Divide G_floating 3-operand	*	*	0	0	rsv fov fuv fdvz
074	EMODD muir.rd, mulrx.rd, muld.rd, int.wl, fract.wd	Extended modulus D_floating	*	*	*	0	rsv fov fuv iov
054	EMODF muir.rf, mulrx.rb, muld.rd int.wl, fract.wf	Extended modulus F_floating	*	*	*	0	rsv fov fuv iov
54FD	EMODG muir.rg, mulrx.rw, muld.rg int.wl, fract.wg	Extended modulus G_floating	*	*	*	0	rsv fov fuv iov
072	*MNEGD src.rd, dst.wd	Move negated D_floating	*	*	0	0	rsv
052	*MNEGF src.rf, dst.wf	Move negated F_floating	*	*	0	0	rsv
52FD	*MNEGG src.rg, dst.wg	Move negated G_floating	*	*	0	0	rsv
070	*MOVD src.rd, dst.wd	Move D_floating	*	*	0	–	rsv
050	*MOVF src.rf, dst.wf	Move F_floating	*	*	0	–	rsv
50FD	*MOVG src.rg, dst.wg	Move G_floating	*	*	0	–	rsv

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
064	MULD2 mulr.rd, prod.md	Multiply D_floating 2-operand	*	*	0	0	rsv, fov, fuv
044	MULF2 mulr.rf, prod.mf	Multiply F_floating 2-operand	*	*	0	0	rsv, fov, fuv
44FD	MULG2 mulr.rg, prod.mg	Multiply G_floating 2-operand	*	*	0	0	rsv, fov, fuv
065	MULD3 mulr.rd, muld.rd, prod.wd	Multiply D_floating 3-operand	*	*	0	0	rsv, fov, fuv
045	MULF3 mulr.rf, muld.rf, prod.wf	Multiply F_floating 3-operand	*	*	0	0	rsv, fov, fuv
45FD	MULG3 mulr.rf, muld.rg, prod.wg	Multiply G_floating 3-operand	*	*	0	0	rsv, fov, fuv
075	POLYD arg.rd, degree rw, tbladder.ab	Evaluate polynomial D_floating	*	*	0	0	rsv, fov, fuv
055	POLYF arg.rf, degree rw, tbladder.ab	Evaluate polynomial F_floating	*	*	0	0	rsv, fov, fuv
55FD	POLYD arg.rg, degree rw, tbladder.ab	Evaluate polynomial G_floating	*	*	0	0	rsv, fov, fuv
062	SUBD2 sub.rd, dif.md	Subtract D_floating 2-operand	*	*	0	0	rsv, fov, fuv
042	SUBF2 sub.rf, dif.mf	Subtract F_floating 2-operand	*	*	0	0	rsv, fov, fuv
42FD	SUBG2 sub.rg, dif.mg	Subtract G_floating 2-operand	*	*	0	0	rsv, fov, fuv
063	SUBD3 sub.rd, min rd, dif.md	Subtract D_floating 3-operand	*	*	*	0	rsv fov fuv
043	SUBF2 sub.rf, min rf, dif.mf	Subtract F_floating 3-operand	*	*	0	0	rsv fov fuv
43FD	SUBG2 sub.rg, min rg, dif.mg	Subtract G_floating 3-operand	*	*	0	0	rsv fov fuv
073	*TSTD src.rd	Test D_floating	*	*	0	0	rsv
053	*TSTF src.rf	Test F_floating	*	*	0	0	rsv
53FD	*TSTG src.rg	Test G_floating	*	*	0	0	rsv

• MicroVAX 78132 FPU Integer Multiplication Instructions

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
07A	Extended multiply	EMUL mulr.rl, muld.rl, add.rl, prod.wq	*	*	0	0	
00A	Index calculation	INDEX subscript.rl, low.rl, high.rl, size.rl, indexin.rl, indexout.wl	*	*	0	0	subscript range
0C4	Multiply long 2-operand	MULL2 mulr.rl, prod.ml	*	*	*	0	iov
0C5	Multiply long 3-operand	MULL3 mulr.rl, muld.rl, prod.wl	*	*	*	0	iov

• MicroVAX 78132 FPU Integer Division Instructions

OP	Mnemonic and Arguments	Description	N	Z	V	C	Exceptions
0C6	Divide long 2-operand	DIVL2 divr.rl, quo.ml	*	*	*	0	iov, fdvz
0C7	Divide long 3-operand	DIVL3 divr.rl, divd.rl, quo.wl	*	*	*	0	iov, fdvz
07B	Extended divide	EDIV divr.rl, divd.rq, quo.wl, rem.wl	*	*	*	0	iov, fdvz

• MicroVAX 78132 FPU Operand Transfer

The integer divide instructions require that the lower 32-bits of the dividend be transferred first and then the upper 32-bits.

VAX Mnemonic	FPU Opcode	First Transfer	Second Transfer	Third Transfer	Operation	Result 1	Result 2
ACBD	06F	limit.d	add.d	index.d	(index+add):limit	(index.d[R])	---
ACBF	04F	limit.f	add.f	index.f	(index+add):limit	(index.f[R])	---
ACBG	14F	limit.g	add.g	index.g	(index+add):limit	(index.g[R])	---
ADDDx	060,061	add2.d	add1.d	---	add1+add2	sum.d[R]	---
ADDFx	040,041	add2.f	add1.f	---	add1+add2	sum.f[R]	---
ADDGx	140,141	add2.g	add1.g	---	add1+add2	sum.g[R]	---
CMPD	071	src2.d	src1.d	---	src1-src2	---	---
CMPF	051	src2.f	src1.f	---	src1-src2	---	---
CMPG	071	src2.g	src1.g	---	src1-src2	---	---
CVTBD	06C	src.b	---	---	fit cvrt	d[rf4/.6]float[E]	---
CVTBF	04C	src.b	---	---	fit cvrt	f[rf4/.6]float[E]	---
CVTBG	14C	src.b	---	---	fit cvrt	g[rf4/.6]float[E]	---
CVTDB	068	src.d	---	---	fit cvrt	byte[T]	---
CVTDF	076	src.d	---	---	fit change	f[rf4/.6]float[R]	---
CVTDL	06A	src.d	---	---	int cvrt	longword[E]	---
CVTDW	069	src.d	---	---	int cvrt	word[T]	---
CVTFB	048	src.f	---	---	int cvrt	byte[T]	---
CVTFD	056	src.f	---	---	fit change	d[rf4/.6]float[E]	---
CVTFG	199	src.f	---	---	int cvrt	g[rf4/.6]float[E]	---
CVTFL	04A	src.f	---	---	int cvrt	longword[E]	---
CVTFW	049	src.f	---	---	int cvrt	word[T]	---
CVTGB	148	src.g	---	---	int cvrt	byte[T]	---
CVTGF	133	src.g	---	---	fit change	f[rf4/.6]float[R]	---
CVTGL	14A	src.g	---	---	int cvrt	longword[T]	---
CVTGW	149	src.g	---	---	int cvrt	word[T]	---
CVTLD	06E	src.l	---	---	fit cvrt	d[rf4/.6]float[E]	---
CVTLF	04E	src.l	---	---	fit cvrt	f[rf4/.6]float[R]	---
CVTLG	14E	src.l	---	---	fit cvrt	g[rf4/.6]float[E]	---
CVTWD	06D	src.w	---	---	fit cvrt	d[rf4/.6]float[E]	---
CVTWF	04D	src.w	---	---	fit cvrt	f[rf4/.6]float[E]	---
CVTWG	14E	src.w	---	---	fit cvrt	g[rf4/.6]float[E]	---
CVTRDL	14A	src.d	---	---	md, int cvrt	longword[R]	---
CVTRFL	14A	src.f	---	---	md, int cvrt	longword[R]	---
CVTRGL	14A	src.g	---	---	md, int cvrt	longword[R]	---
DIVDx	066,067	divd.d	divr.d	---	divd/divr	quo.d[R]	---
DIVFx	046,047	divd.f	divr.f	---	divd/divr	quo.f[R]	---
DIVGx	146,147	divd.g	divr.g	---	divd/divr	quo.g[R]	---
EMODD	074	mulrx.b	mulr.d	muld.d	muld*(mulr'mulrx)	int.l[E]	fract.d[R]
EMODF	054	mulrx.b	mulr.f	muld.f	muld*(mulr'mulrx)	int.l[E]	fract.f[R]
EMODG	154	mulrx.b	mulr.g	muld.g	muld*(mulr'mulrx)	int.l[E]	fract.g[R]
MULDx	064,065	mulr.d	muld.d	---	mulr*muld	prod.d[R]	---
MULFx	044,045	mulr.f	muld.f	---	mulr*muld	prod.f[R]	---
MULGx	144,145	mulr.g	muld.g	---	mulr*muld	prod.g[R]	---

VAX Mnemonic	FPU Opcode	First Transfer	Second Transfer	Third Transfer	Operation	Result 1	Result 2
POLYD	075	arg.d	int1.d	#coeff.d	(arg*int1)+coeff	#int2.d[R]	#int3.d[R]
POLYF	055	arg.f	int1.f	#coeff.f	(arg*int1)+coeff	#int2.f[R]	#int3.f[R]
POLYG	155	arg.g	int1.g	#coeff.g	(arg*int1)+coeff	#int2.g[R]	#int3.g[R]
SUBDx	062,063	min.d	sub.d	---	min–sub	diff.d[R]	---
SUBFx	042,043	min.f	sub.f	---	min–sub	diff.f[R]	---
SUBGx	142,143	min.g	sub.g	---	min–sub	diff.g[R]	---
EMUL	07A	mulr.rl	muld.rl	---	mulr*muld	prod.wq[E]	---
INDEX	00A	(mulr.rl)	size.rl	---	mulr*size	indexout.wq[E]	---
MULLx	0C4,0C5	mulr.rl	muld.rl	---	mulr*muld	prod.wq[E]	---
DIVLx	0C6,0C7	divr.rl	divd.rq		divd/divr	quo.wl[E]	rem.wl[E]
EDIVx	07B	divr.rl	divd.rq		divd/divr	quo.wl[E]	rem.wl[E]

The DCT11 and DCJ11 16-bit microprocessors share the PDP-11 architecture and operate with a common instruction set and similar addressing techniques. The following table list the instruction set for both processors. The instructions that do not apply to the DCT11 microprocessor are indicated by a dagger (†) preceding the instruction mnemonic term. The DCT11 microprocessor does not perform floating-point arithmetical operations and some of the DCT11 instruction executions are performed differently from those of the DCJ11 microprocessor.

The abbreviations for the condition codes (N,Z,V,C) listed in the table are as follows. For specific condition code information, refer to the *μT11 User's Guide.*

* = conditionally set/cleared
– = not affected
0 = cleared
1 = set

The block □ preceding the opcode or base code = 0 for word/1 for byte

• Single Operand

General

Opcode	Mnemonic	Description	N	Z	V	C
□050DD	CLR (B)	Clear destination	0	1	0	0
□051DD	COM (B)	Complement destination	*	*	0	1
□052DD	INC (B)	Increment destination	*	*	*	–
□053DD	DEC (B)	Decrement destination	*	*	*	–
□054DD	NEG (B)	Negate destination	*	*	*	*
□057DD	TST (B)	Test destination	*	*	0	0
0073DD	†WRTLCK	Read/lock destination, write/unlock RO into destination	*	*	0	–
0072DD	†TSTSET	Test destination, set low bit	*	*	0	*

Shift and Rotate

Opcode	Mnemonic	Description	N	Z	V	C
□062DD	ASR (B)	Arithmetic shift right	*	*	*	*
□063DD	ASL (B)	Arithmetic shift left	*	*	*	*
□060DD	ROR (B)	Rotate right	*	*	*	*
□061DD	ROL (B)	Rotate left	*	*	*	*
0003DD	SWAB	Swap bytes	*	*	0	0

Multiple-precision

Opcode	Mnemonic	Description	N	Z	V	C
□055DD	ADC (B)	Add carry	*	*	*	*
□056DD	SBC (B)	Subtract carry	*	*	*	*
0067DD	SXT	Sign extend	–	*	0	–

Processor Status (PS) Word Operators

Opcode	Mnemonic	Description	N	Z	V	C
1067DD	MFPS	Move byte from PS	*	*	0	-
1064SS	MTPS	Move byte to PS	*	*	*	*

• Double Operand

General

Opcode	Mnemonic	Description	N	Z	V	C
□1SSDD	MOV (B)	Move source to destination	*	*	0	–
□2SSDD	CMP (B)	Compare source to destination	*	*	*	*
06SSDD	ADD	Add source to destination	*	*	*	*
16SSDD	SUB	Subtract source from destination	*	*	*	*
072RSS	†ASH	Arithmetic shift	*	*	*	*
073RSS	†ASHC	Arithmetic shift combined	*	*	*	*
070RSS	†MUL	Multiply	*	*	0	*
071RSS	†DIV	Divide	*	*	*	*

Logical

Opcode	Mnemonic	Description	N	Z	V	C
□3SSDD	BIT (B)	Bit test	*	*	0	–
□4SSDD	BIC (B)	Bit clear	*	*	0	–
□5SSDD	BIS (B)	Bit set	*	*	0	–
074RDD	XOR	Exclusive OR	*	*	0	–

• Program Control

Branch

Opcode or Base Code	Mnemonic	Description	N	Z	V	C
000400	BR	Branch unconditional	–	–	–	–
001000	BNE	Branch if not equal to zero	–	–	–	–
001400	BEQ	Branch if equal to zero	–	–	–	–
100000	BPL	Branch if plus	–	–	–	–
100400	BMI	Branch if minus	–	–	–	–
102000	BVC	Branch if overflow is clear	–	–	–	–
102400	BVS	Branch if overflow is set	–	–	–	–
103000	BCC	Branch if carry is clear	–	–	–	–
103400	BCS	Branch if carry is set	–	–	–	–

Signed Conditional Branch

Opcode or Base Code	Mnemonic	Description	N	Z	V	C
002000	BGE	Branch if greater than or equal to zero	–	–	–	–
002400	BLT	Branch if less than zero	–	–	–	–
003000	BGT	Branch if greater than zero	–	–	–	–
003400	BLE	Branch if less than or equal to zero	–	–	–	–

Unsigned Conditional Branch

Opcode or Base Code	Mnemonic	Description	N	Z	V	C
101000	BHI	Branch if higher	–	–	–	–
101400	BLOS	Branch if lower or same	–	–	–	–
103000	BHIS	Branch if higher or same	–	–	–	–
103400	BLO	Branch if lower	–	–	–	–

Jump and Subroutine

Opcode or Base Code	Mnemonic	Description	N	Z	V	C
0001DD	JMP	Jump	–	–	–	–
004RDD	JSR	Jump to subroutine	–	–	–	–
00020R	RTS	Return from subroutine	–	–	–	–
077R00	SOB	Subtract one and branch if not equal to zero	–	–	–	–

Trap and Interrupt

Opcode or Base Code	Mnemonic	Description	N	Z	V	C
104000 to 104377	EMT	Emulator trap	*	*	*	*
104400 to 104777	TRAP	Trap	*	*	*	*
000003	BPT	Breakpoint trap	*	*	*	*
000004	IOT	Input/output trap	*	*	*	*
000002	RTI	Return from interrupt	*	*	*	*
000006	RTT	Return from interrupt	*	*	*	*

Miscellaneous Program Control

Opcode or Base Code	Mnemonic	Description	N	Z	V	C
0070DD	†CSM	Call to supervisor mode	–	–	–	–
0064NN	†MARK	Mark	–	–	–	–
00023N	†SPL	Set priority level	–	–	–	–

Miscellaneous

Opcode or Base Code	Mnemonic	Description	N	Z	V	C
000000	HALT	Halt	–	–	–	–
000001	WAIT	Wait for interrupt	–	–	–	–
000005	RESET	Reset external bus	–	–	–	–
000007	MFPT	Move processor type	–	–	–	–
1066SS	†MTPD	Move to previous data space	*	*	0	–
0066SS	†MTPI	Move to previous instruction space	*	*	0	–
0065SS	†MFPD	Move from previous data space	*	*	0	–
1065SS	†MFPI	Move from previous instruction space	*	*	0	–

Condition Code Operators

Opcode or Base Code	Mnemonic	Description	N	Z	V	C
000241	CLC	Clear C	–	–	–	0
000242	CLV	Clear V	–	–	0	–
000244	CLZ	Clear Z	–	0	–	–
000250	CLN	Clear N	0	–	–	–
000257	CCC	Clear all CC bits	0	0	0	0
000261	SEC	Set C	–	–	–	1
000262	SEV	Set V	–	–	1	–
000264	SEZ	Set Z	–	1	–	–
000270	SEN	Set N	1	–	–	–
000277	SCC	Set all CC bits	1	1	1	1
000240	NOP	No operation	–	–	–	–

The following electrical circuit configurations are used in performing the dc tests. A definition of the parameters used in the tests is contained in Table C.1.

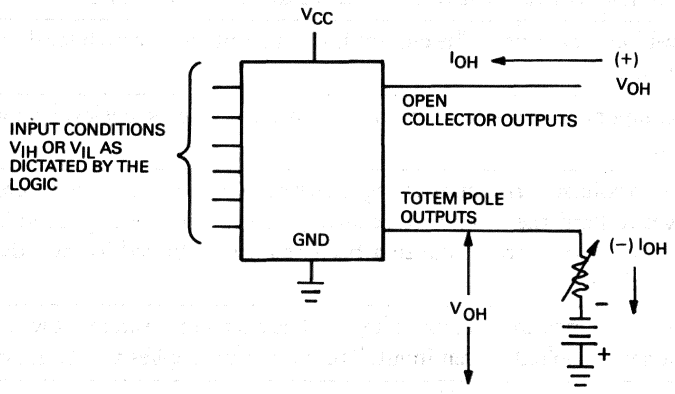
Current flow into the device is a negative value and current flow out of the device is a positive value.

High (H) assertion signals are true or asserted for high-level voltages and false or negated for low-level voltages. Low (L) assertion signals are true or asserted for low-level voltages and false or negated for high-level voltages.

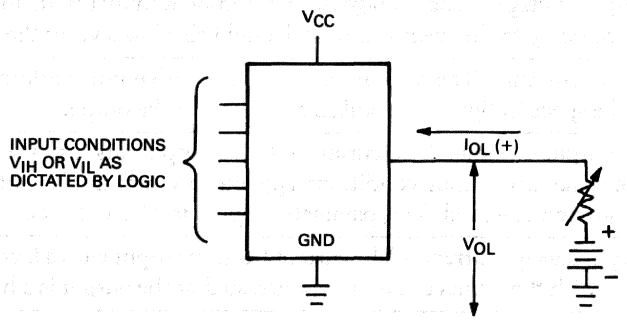
Table C.1 • dc Test Specification Parameters

Symbol	Name/Definition
V_{IH}	High-level input voltage—An input voltage level within the more positive (less negative) of the two ranges of values used to represent the binary variables.
V_{IL}	Low-level input voltage—An input voltage level within the less positive (more negative) of the two ranges of values used to represent the binary variables.
I_{IH}	High-level input current—The current into an input when a high level voltage is applied to that input.
I_{IL}	Low-level input current—The current into an input when a low level voltage is applied to that input.
V_{IC}	Input clamp voltage—An input voltage in a region of relatively low differential resistance that serves to limit the input voltage swing. This parameter applies to TTL inputs that have clamping diodes to ground that become forward biased for negative excursions of the input voltage.
I_{IM}	Input current at maximum input voltage—The current into an input when the maximum input voltage is applied to that input. This parameter applies to TTL inputs.
V_{OH}	High-level output voltage—The voltage at an output terminal with input conditions applied that according to the specification will establish a high level at the output.
V_{OL}	Low-level output voltage—The voltage at the output terminal with input conditions applied that according to the specification will establish a low level at the output.
I_{OL}	Low-level output current—The current into an output with input conditions applied that according to the specification will establish a low level at the output.
I_{OS}	Short circuit output current—The current into an output when that output is short-circuited to ground with input conditions applied to establish the output logic level farthest from ground potential. This parameter applies to TTL outputs.
I_{OZL}	Low-level output leakage current—The current from an output with a low level applied to the output and with the input conditions applied so that the output is a high impedance.

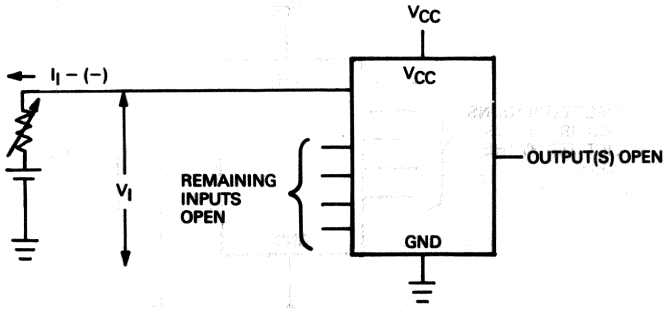
Symbol	Name/Definition
I_{OZH}	High-level output leakage current—The current from an output with a high-level applied to the output and with the input conditions applied so that the output is a high impedance.
I_{CC}	Positive power supply current—The current into the V_{CC} input from the power supply. V_{CC} represents the positive power supply voltage applied to the device.
I_{EE}	Negative power supply current—The current into the V_{EE} supply terminal of the device. V_{EE} represents the negative power supply voltage applied to the device.
C_{in}	Input capacitance—The capacitance measured at the specified pins with power applied to the device.
R_{in}	Real input impedance—The real portion of the input impedance measured at the specified pins with power applied to the device.



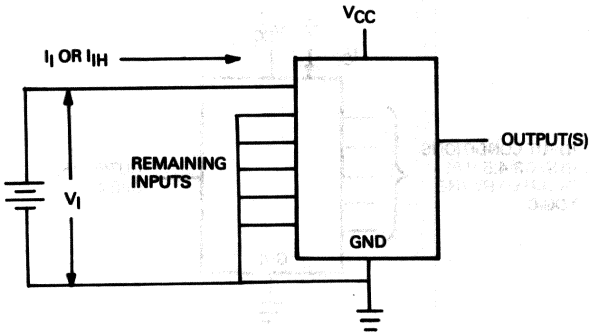
DC TEST CIRCUIT C1



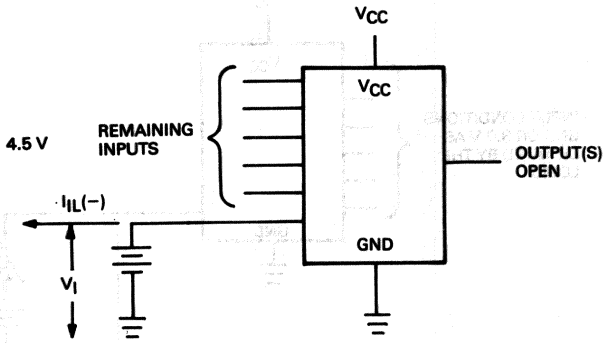
DC TEST CIRCUIT C2



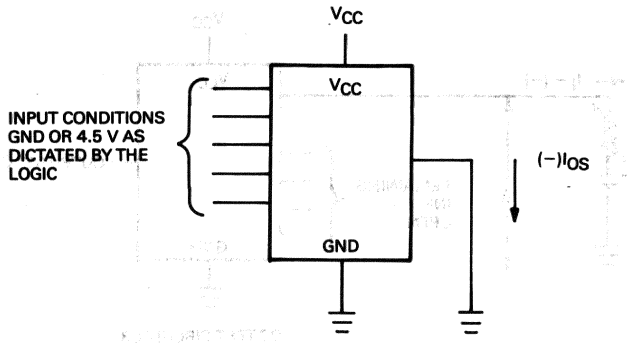
DC TEST CIRCUIT C3



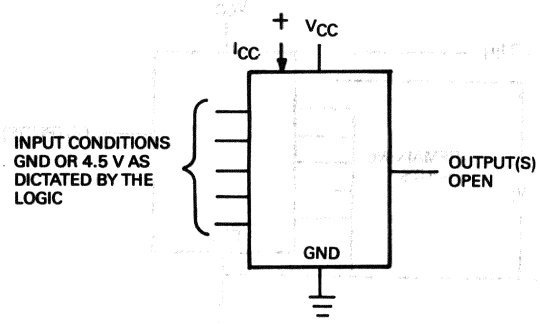
DC TEST CIRCUIT C4



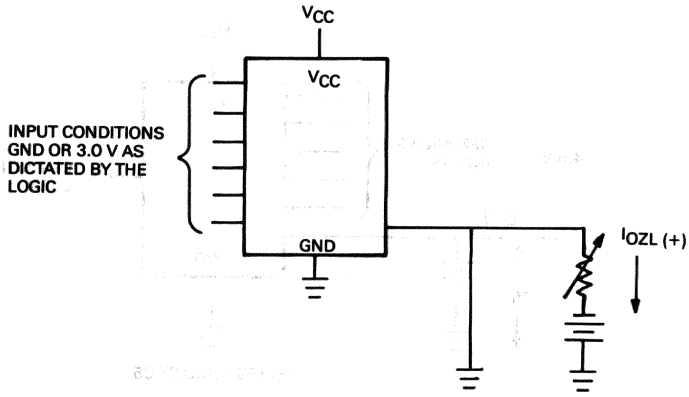
DC TEST CIRCUIT C5



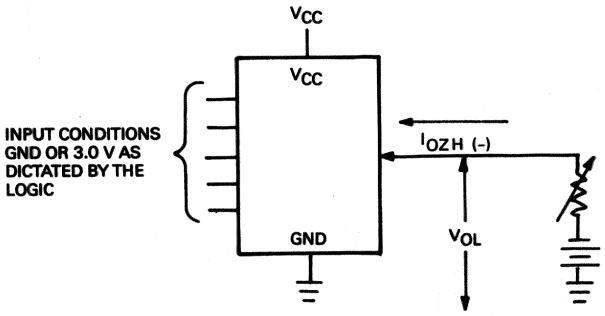
DC TEST CIRCUIT C6



DC TEST CIRCUIT C7



DC TEST CIRCUIT C8



DC TEST CIRCUIT C9

Figure D.1 shows the waveforms and symbols used to measure the propagation delay for some input and output voltages.

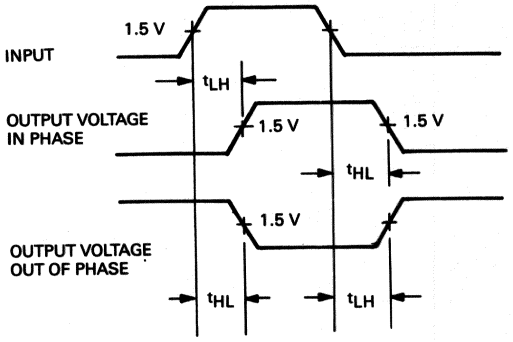


Figure D.1 ▪ Input/Output Propagation Delay Symbols

Figure E.1 and E.2 show the configurations of the molded plastic/ceramic dual-inline packages. Figure E.3 shows the configuration of the DCT11 40-pin ceramic package. Table E.1 lists the pin and package dimensions.

Figure E.4 shows the dimensions for the DCJ11 60-pin ceramic package and Figure E.5 shows the dimensions for the FPJ11 40-pin ceramic package.

Figure E.6 shows the configuration and dimensions of the cerquad inline package.

Figure E.7 shows the configuration of the pin grid array (PGA) package and Table E.2 lists the package dimensions.

Table E.1 • Molded DIP Package Dimensions

Package	14-pin		16-pin		18-pin		20-pin		28-pin		40-pin	
Dimension	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
A	0.28	0.29	0.28	0.29	0.28	0.29	0.28	0.29	0.58	0.59	0.58	0.59
B	0.605	0.705	0.705	0.805	0.805	0.905	0.905	1.05	1.305	1.405	1.905	2.05
C	0.08	0.18	0.08	0.18	0.08	0.18	0.08	0.18	0.08	0.18	0.08	0.18
D	0.12	—	0.12	—	0.12	—	0.12	—	0.12	—	0.12	—
E	0.1 ± 0.01*											
F	0.014	0.022	0.014	0.022	0.014	0.022	0.014	0.022	0.014	0.022	0.014	0.022
G	0.015	0.06	0.015	0.06	0.015	0.06	0.015	0.06	0.015	0.06	0.015	0.06
H	0.044	0.07	0.044	0.07	0.044	0.07	0.044	0.07	0.044	0.07	0.044	0.07
I	0.008	0.075	0.008	0.075	0.008	0.075	0.008	0.075	0.008	0.075	0.008	0.075
J	0.008	0.014	0.008	0.014	0.008	0.014	0.008	0.014	0.012	0.02	0.012	0.02
K	—	0.4	—	0.4	—	0.4	—	0.4	—	0.7	—	0.7
L	—	0.35	—	0.35	—	0.35	—	0.35	—	0.35	—	0.35

*Non-cumulative at seating plane

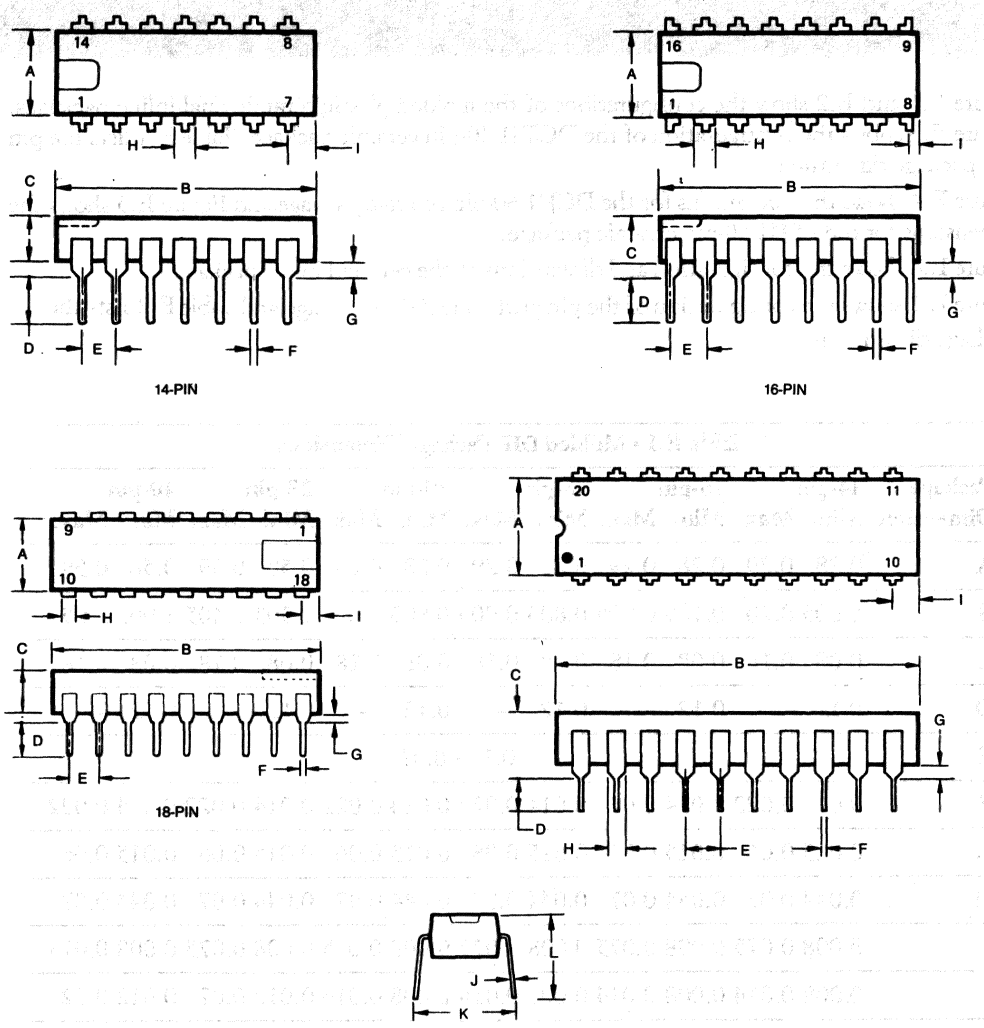


Figure E.1 ▪ 14-, 16-, 18- and 20-pin Plastic/Ceramic DIP Configurations

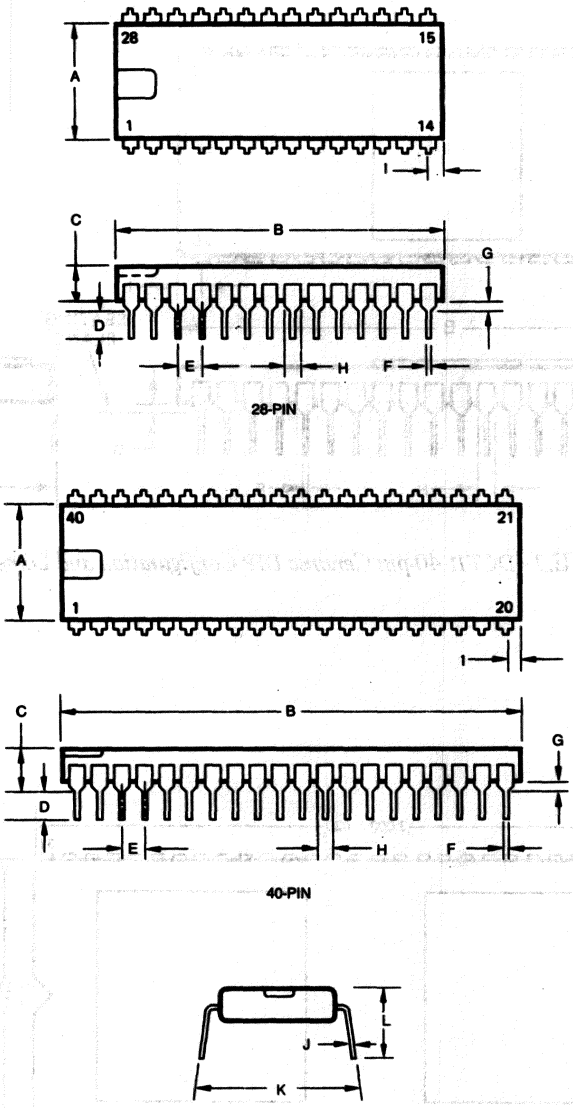


Figure E.2 ▪ 28- and 40-pin Plastic/Ceramic DIP Configurations

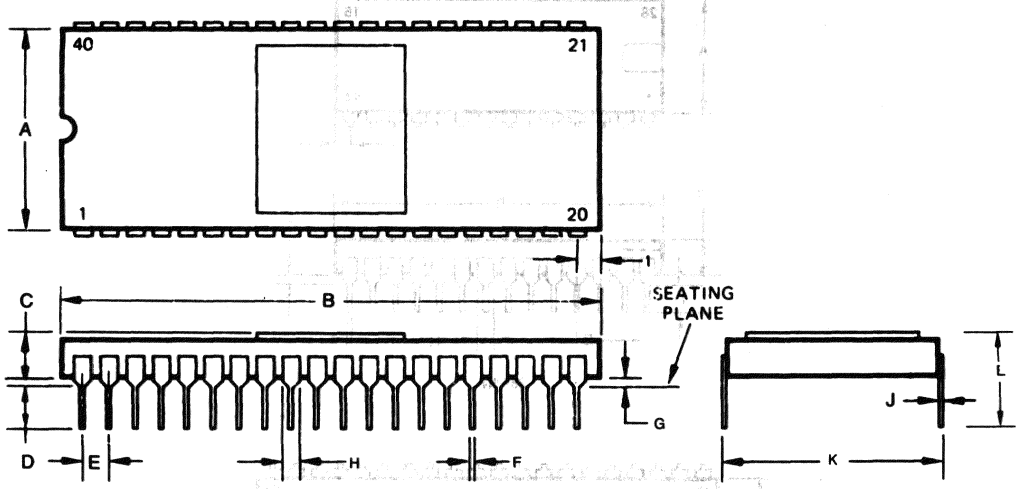


Figure E.3 ▪ DCT11 40-pin Ceramic DIP Configuration and Dimensions

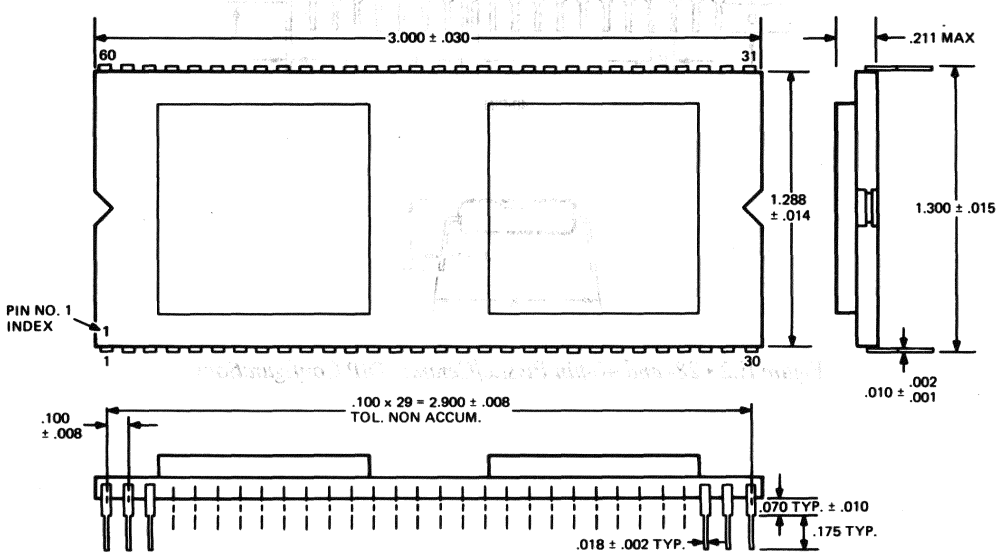


Figure E.4 ▪ DCJ11 60-pin Ceramic DIP Configuration and Dimensions

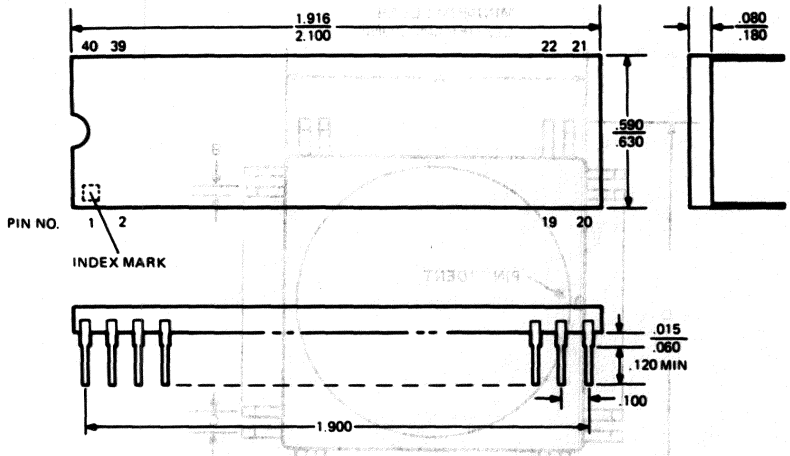
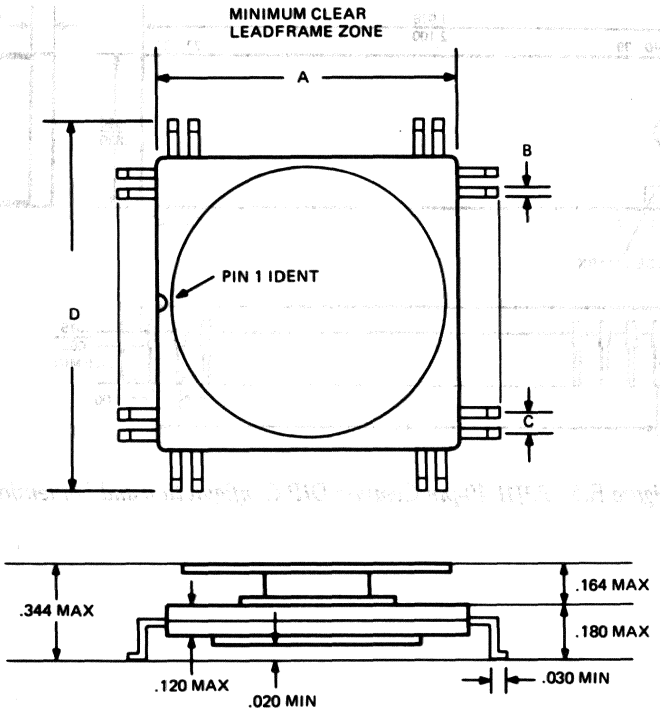


Figure E.5 ▪ FPJ11 40-pin Ceramic DIP Configuration and Dimensions



Number of Leads	Dimensions A	B	C	D
44	0.6	0.02	0.05	0.825
68	0.9	0.02	0.05	1.125
84	1.1	0.02	0.05	1.325
132	0.9	0.012	0.025	1.125
164	1.1	0.012	0.025	1.325

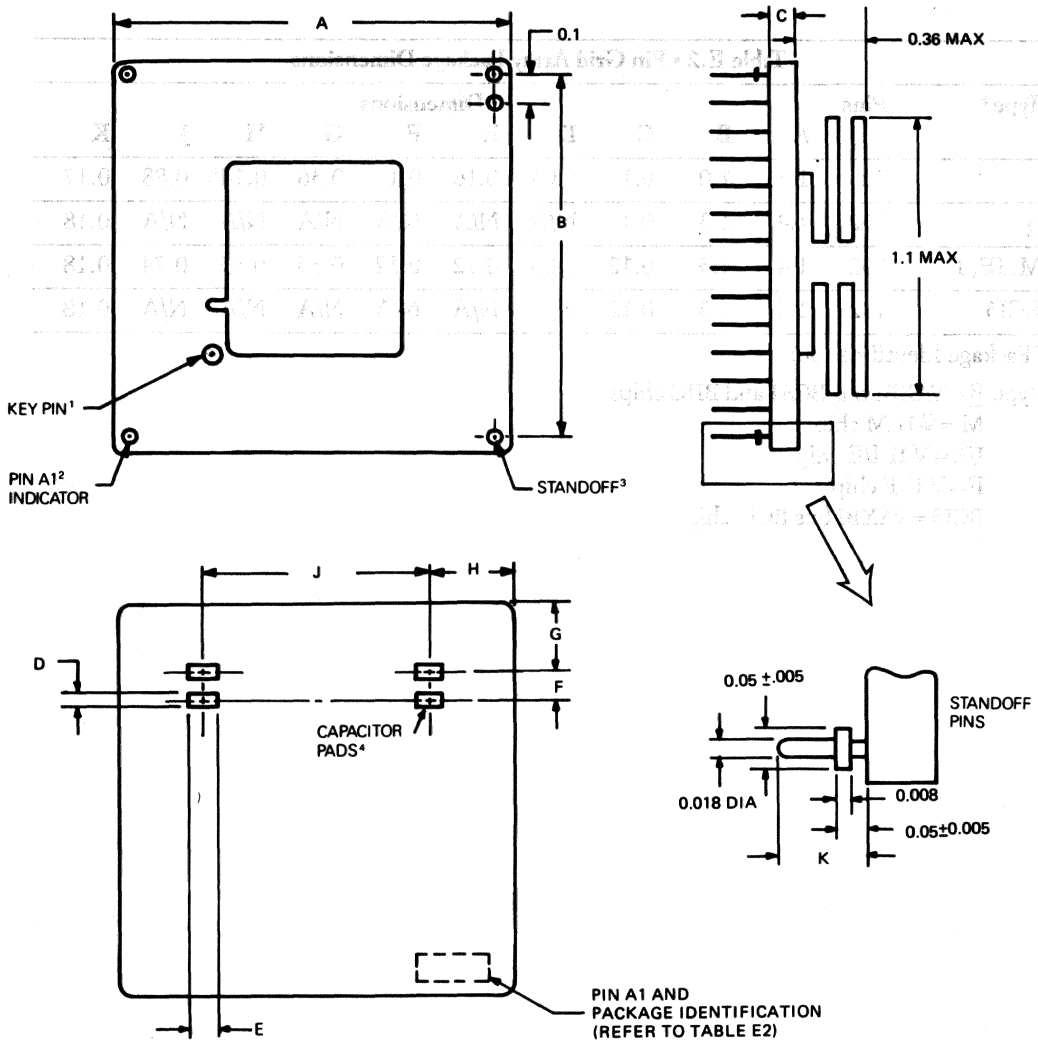
Figure E.6 ▪ Cerquad Package Configuration and Dimensions

Table E.2 • Pin Grid Array Package Dimensions

Type*	Pins	Dimensions									
		A	B	C	D	E	F	G	H	J	K
	72	1.17	1.0	0.1	0.05	0.16	0.1	0.36	0.145	0.88	0.17
B	132	1.4	1.3	0.1	N/A	N/A	N/A	N/A	N/A	N/A	0.18
M, IE, F	132	1.4	1.3	0.12	0.05	0.12	0.12	0.35	0.33	0.74	0.18
BCI3	132	1.4	1.3	0.12	N/A	N/A	N/A	N/A	N/A	N/A	0.18

*Package Identification:

Type B = VAXBI bus BCAI and BIIC chips
M = V-11 M chip
I/E = V-11 I/E chip
F = V-11 F chip
BCI3 = VAXBI bus BCI3 chip



[1]Key pin is nonelectrical and is for alignment on type B chips only.
[2]Pin A1 is indicated by a protrusion on the standoff collar.
[3]Standoff pins are positioned at the four exterior corners of the 132-pin PGA and at the four interior corners of the 72-pin PGA.
[4]Capacitor pads not available on the B and BCI3 PGA versions.

Figure E.7 ▪ PGA Package Configuration and Dimensions