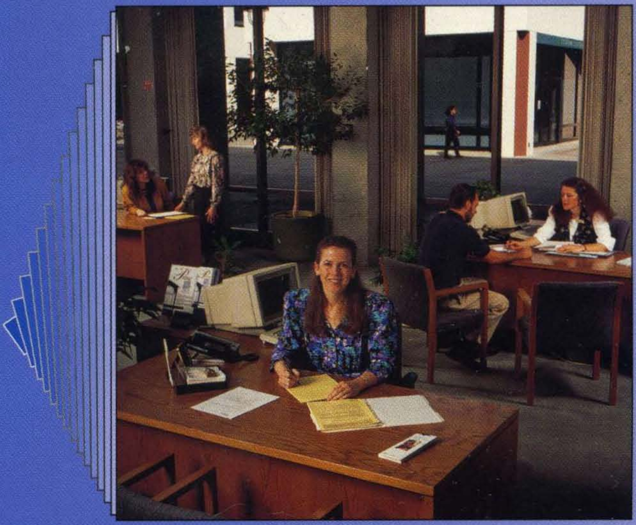
ZiLOG

High-Speed Serial Communication Controllers



Includes Specifications for the following parts:

Z16C30
Z16C32

Product Specifications Databook

High-Speed Serial Communication Controllers

Includes Specifications for the following parts:

- Z16C30
- Z16C32

Databook

HIGH-SPEED SCC DATABOOK

TABLE OF CONTENTS

TITLE PAGE

ZiLOG

INTRODUCTION

Zilog's Focus on Application Specific Products Helps You Maintain Your Technological Edge

Zilog's High-Speed SCC products are suitable for a broad range of applications, from general-purpose use through high-end LANs and WANs. Whichever device you choose, you'll find a comprehensive feature set and easy-to-use development tools to speed your design time to production.

Z16C30 CMOS USC™ Universal Serial Controller

The USC Universal Serial Controller (USC) is a dual-channel multi-protocol datacommunications peripheral designed for use with any conventional multiplexed or non-multiplexed bus. The two independent full duplex channels can each operate up to 10 Mbps. The USC functions as a powerful serial-to-parallel, parallel-to-serial converter/controller and may be software configured to satisfy a wide variety of serial communications applications. The device contains a variety of new, sophisticated internal functions including two baud rate generators per channel, a digital phase-locked loop per channel, character counters for both receive and transmit in each channel, and 32-byte data FIFOs for each receiver and transmitter.

The USC handles asynchronous formats, synchronous byte-oriented formats such as BISYNC, and synchronous bit-oriented formats such as HDLC. This device supports virtually any serial data transfer application.

High-speed data transfers through DMA are supported by a Request/Acknowledge signal pair for each receiver and transmitter. The device supports automatic status transfer through DMA and allows device initialization under DMA control.

Z16C32 CMOS IUSC™ Integrated Universal Serial Controller

The Integrated Universal Serial Controller (IUSC) is a single-channel multiple protocol datacommunications device with on-chip DMA. The integration of a high speed serial communications channel with high performance DMA facilitates higher data throughput than is possible with discrete serial/DMA chip combinations. The buffer chaining capabilities combined with features like character counters, frame status block and buffer termination at the end of the frame facilitate sophisticated buffer management that can significantly reduce CPU overhead.

The IUSC is software configurable to satisfy a wide variety of serial communications applications. The 10 Mbit per second data rate and multiple protocol support make it ideal for applications in todays dynamic environment of changing specifications and ever increasing speed. The many programmable features allow the user to tune the device response to meet system requirements and adapt to future changes with software instead of redesigning hardware.

The on-chip DMA channels allow high-speed data transfers for both the receiver and the transmitter. The device supports automatic status transfer through DMA and allows device initialization under DMA control. Each DMA channel can transfer data words in as little as three 50 ns clock cycles and can generate addresses compatible with 32-, 24- or 16-bit memory ranges. The DMA channels may operate in any of four modes: single buffer, pipelined, array-chained, or linked-list. The array-chained and linked-list modes improve memory management and allow the IUSC to receive and transmit multiple frames without CPU intervention. To prevent the DMA from holding bus mastership too long, mastership time may be limited by counting the absolute number of clock cycles, the number of bus transactions, or both.

PRODUCT SPECIFICATION

Z16C30

CMOS USC™ UNIVERSAL SERIAL CONTROLLER

FEATURES

- Two Independent, 0 to 10 Mbit/sec, Full Duplex Channels, Each with Two Baud Rate Generators and One Digital Phase-Locked Loop for Clock Recovery.
- 32-Byte Data FIFO's for Each Receiver and Transmitter
- 110 ns Bus Cycle Time, 16-Bit Data Bus Bandwidth
- Multi-Protocol Operation Under Program Control with Independent Mode Selection for Receiver and Transmitter.
- Async Mode with One to Eight Bits/Character, 1/16 to 2 Stop Bits/Character in 1/16 Bit Increments; Programmable Clock Factor; Break Detect and Generation; Odd, Even, Mark, Space or no Parity and Framing Error Detection. Supports One Address/Data Bit and MIL STD 1553B Protocols.
- Byte Oriented Synchronous Mode with One to Eight Bits/Character; Programmable Idle Line Condition; Optional Receive Sync Stripping; Optional Preamble Transmission; 16- or 32-Bit CRC and Transmit-to-Receive Slaving (for X.21).
- Bisync Mode with 2- to 16-Bit Programmable Sync Character; Programmable Idle Line Condition; Optional Receive Sync Stripping; Optional Preamble Transmission; 16- or 32-Bit CRC.
- Transparent Bisync Mode with EBCDIC or ASCII Character Code; Automatic CRC Handling; Programmable Idle Line Condition; Optional Preamble Transmission; Automatic Recognition of DLE, SYN, SOH, ITX, ETX, ETB, EOT, ENQ and ITB.
- External Character Sync Mode for Receive
- HDLC/SDLC Mode with Eight Bit Address Compare; Extended Address Field Option; 16- or 32-Bit CRC; Programmable Idle Line Condition; Optional Preamble Transmission and Loop Mode.
- DMA Interface with Separate Request and Acknowledge for Each Receiver and Transmitter.
- Channel Load Command for DMA Controlled Initialization.
- Flexible Bus Interface for Direct Connection to Most Microprocessors; User Programmable for 8 or 16 Bits Wide. Directly Supports 680X0 Family or 8X86 Family Bus Interfaces.
- Low Power CMOS
- 68-Pin PLCC Package

GENERAL DESCRIPTION

The USC™ Universal Serial Controller is a dual-channel multi-protocol data communications peripheral designed for use with any conventional multiplexed or non-multiplexed bus. The USC functions as a serial-to-parallel, parallel-to-serial converter/controller and may be software configured to satisfy a wide variety of serial communications applications. The device contains a variety of new, sophisticated internal functions including two baud rate generators per channel, a digital phase-locked loop per channel, character counters for both receive and transmit in each channel and 32-byte data FIFO's for each receiver and transmitter.

Zilog now offers a high speed version of the USC with improved bus bandwidth. CPU bus accesses have been shortened from 160 ns per access to 110 ns per access. The USC has a transmit and receive clock range of up to 10 MHz (20 MHz when using the DPLL, BRG, or CTR) and data transfer rates as high as 10 Mbits/sec full duplex.

The USC handles asynchronous formats, synchronous byte-oriented formats such as BISYNC and synchronous bit-oriented formats such as HDLC. This device supports virtually any serial data transfer application.

GENERAL DESCRIPTION (Continued)

The device can generate and check CRC in any synchronous mode and can be programmed to check data integrity in various modes. The USC also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls may be used for general-purpose I/O. The same is true for most of the other pins in each channel.

Interrupts are supported with a daisy-chain hierarchy, with the two channels having completely separate interrupt structures.

High-speed data transfers through DMA are supported by a Request/Acknowledge signal pair for each receiver and transmitter. The device supports automatic status transfer through DMA and also allows device initialization under DMA control.

To aid the designer in efficiently programming the USC, support tools are available. The Technical Manual describes in detail all features presented in this Product Specification and gives programming sequence hints. The Programmer's Assistant is a MS-DOS disk-based programming initialization tool to be used in conjunction with the Technical Manual. There are also available assorted application notes and development boards to assist the designer in the hardware/software development.

Note: All reserved bits must be programmed to zero when written to.

Notes:
All Signals with a preceding front slash, "/", are active Low, e.g., B//W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

Connection	Circuit	Device
Power	V_{CC}	V_{DD}
Ground	GND	V_{SS}

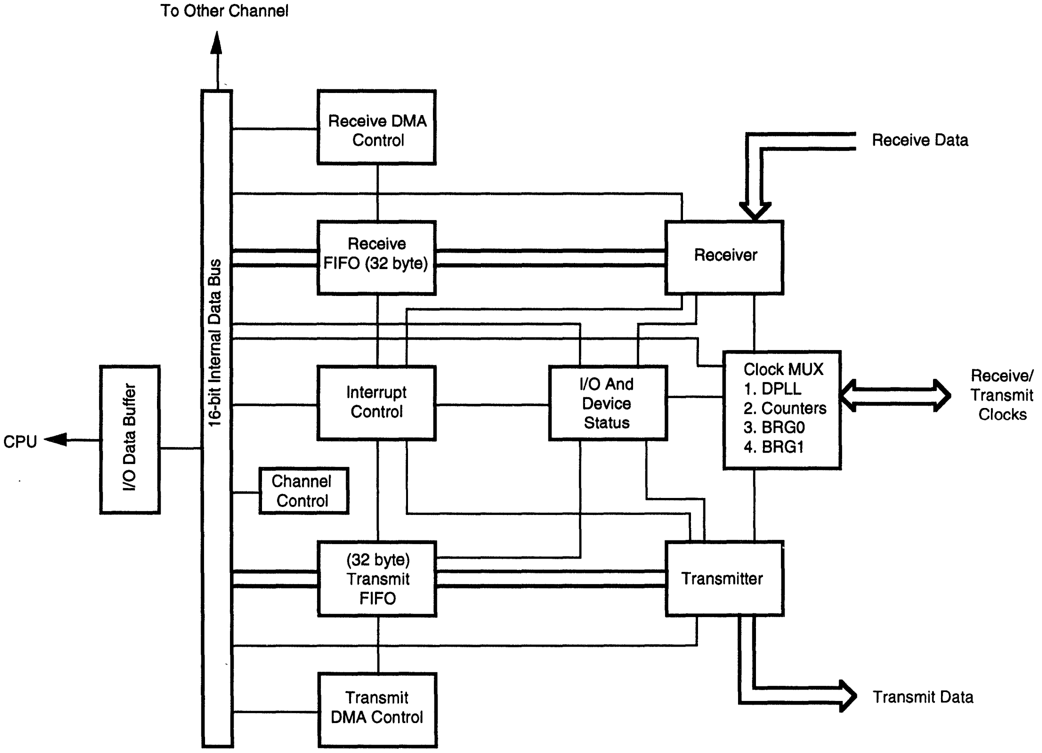


Figure 1. USC Block Diagram

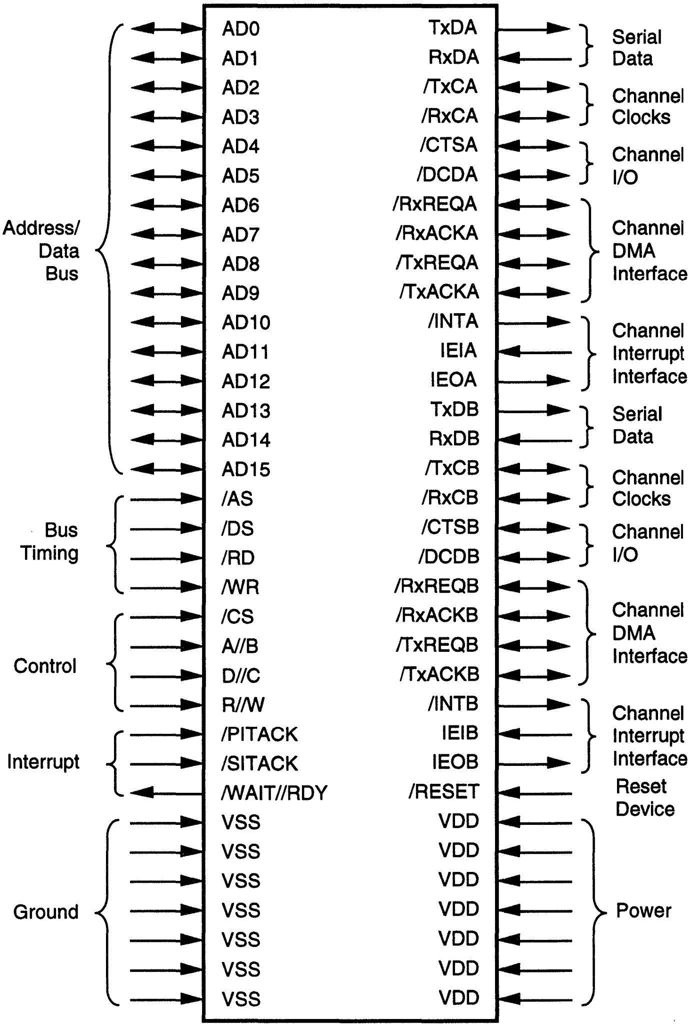


Figure 2. Pin Functions

GENERAL DESCRIPTION (Continued)

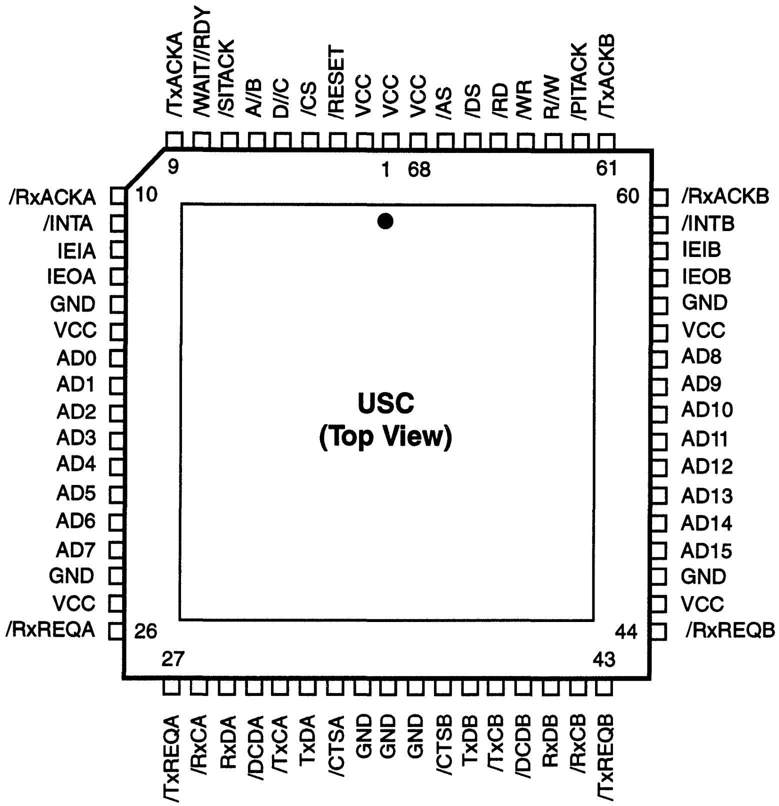


Figure 3. Pin Assignments

PIN DESCRIPTION

The device contains 13 pins per channel for channel I/O, 16 pins for address and data, 12 pins for CPU handshake and 14 pins for power and ground.

Three separate bus interface types are available for the device. The Bus Configuration Register (BCR) and external connections to the AD bus control selection of the bus type. A 16-bit bus is selected by setting BCR bit 2 to a 1. The 8-bit bus is selected by setting BCR bit 2 to zero and tying AD15 - AD8 to V_{SS}.

The 8-bit bus with separate address is selected by setting BCR bit 2 to zero and, during the BCR write, forcing AD15 to a 1 and forcing AD14-AD8 to zero.

The multiplexed bus is selected for the USC if there is an Address Strobe prior to or during the transaction which writes the BCR. If no Address Strobe is present prior to or during the transaction which writes the BCR, a non-multiplexed bus is selected (See Figure 6).

/RESET *Reset* (input, active Low). This signal resets the device to a known state. The first write to the USC after a reset accesses the BCR to select additional bus options for the device.

/AS *Address Strobe* (input, active Low). This signal is used in the multiplexed bus modes to latch the address on the AD lines. The /AS signal is not used in the non-multiplexed bus modes and should be tied to V_{DD}.

/DS *Data Strobe* (input, active Low). This signal strobes data out of the device during a read and may strobe an interrupt vector out of the device during an interrupt acknowledge cycle. /DS also strobes data into the device on the state of R//W.

/RD *Read Strobe* (input, active Low). This signal strobes data out of the device during a read and may strobe an interrupt vector out of the device during an interrupt acknowledge cycle.

/WR *Write Strobe* (input, active Low). This signal strobes data into the device during a write.

R//W *Read/Write* (input). This signal determines the direction of data transfer for a read or write cycle in conjunction with /DS.

/CS *Chip Select* (input, active Low). This signal selects the device for access and must be asserted for read and write cycles, but is ignored during interrupt acknowledge and fly-by DMA transfers. In the case of a multiplexed bus interface, /CS is latched by the rising edge of /AS.

A//B *Channel A/Channel B Select* (input). This signal selects between the two channels in the device. High selects channel A and Low selects channel B. This signal is sampled and the result is latched during the BCR (Bus Configuration Register) write. It programs the sense of the /WAIT//RDY signal appropriate for different bus interfaces.

D//C *Data/Control Select* (input). This signal, when High, provides for direct access to the RDR and TDR. In the case of a multiplexed bus interface, D//C High overrides the address provided to the device.

/SITACK *Status Interrupt Acknowledge* (input, active Low). This signal is a status signal that indicates that an interrupt acknowledge cycle is in progress. The device is capable of returning an interrupt vector that may be encoded with the type of interrupt pending during this acknowledge cycle. This signal is compatible with 680X0 family microprocessors.

/PITACK *Pulsed Interrupt Acknowledge* (input, active Low). This signal is a strobe signal that indicates that an interrupt acknowledge cycle is in progress. The device is capable of returning an interrupt vector that may be encoded with the type of interrupt pending during this acknowledge cycle. /PITACK may be programmed to accept a single pulse or double pulse acknowledge type. This programming is done in the BCR. With the double pulse type selected, the first /PITACK is recognized but no action takes place. The interrupt vector is returned on the second pulse if the no vector option is not selected. The double pulse type is compatible with 8X86 family microprocessors.

/WAIT//RDY *Wait/Data Ready* (output, active Low). This signal serves to indicate when the data is available during a read cycle, when the device is ready to receive data during a write cycle, and when a valid vector is available during an interrupt acknowledge cycle. It may be programmed to function either as a Wait signal or a Ready signal using the state of the A//B pin during the BCR write. When A//B is High during the BCR write, this signal functions as a wait output and thus supports the READY function of 8X86 family microprocessors. When A//B is Low during the BCR write, this signal functions as a ready output and thus supports the DTACK function of 680X0 family microprocessors.

AD15-AD0 *Address/Data Bus* (bidirectional, active High, tri-state). The AD signals carry addresses to, and data to and from, the device. When the 16-bit non-multiplexed bus is selected, AD15-AD0 carry data to and from the device. Addresses are provided using a pointer within the device that is loaded with the desired register address. When selecting the 8-bit non-multiplexed bus (without separate address) only AD7-AD0 are used to transfer data. The pointer is used for addressing, with AD15-AD8 unused. When selecting the 8-bit non-multiplexed bus (with separate address), AD7-AD0 are used to transfer data with AD15-AD8 used as address bus. When the 16-bit multiplexed bus is selected, addresses are latched from AD7-AD0 and data transfers are sixteen bits wide. When selecting the 8-bit multiplexed bus (without separate address) only AD7-AD0 are used to transfer addresses and data, with AD15-AD8 unused. When the 8-bit multiplexed bus with separate address is selected, only AD7-AD0 are used to transfer data, while AD15-AD8 are used as an address bus.

/INTA, /INTB *Interrupt Request* (outputs, active Low). These signals indicate that the channel has an interrupt condition pending and is requesting service. These outputs are NOT open-drain.

PIN DESCRIPTION (Continued)

IEIA, IEIB *Interrupt Enable In* (inputs, active High). The IEI signal for each channel is used with the accompanying IEO signal to form an interrupt daisy chain. An active IEI indicates that no device having higher priority is requesting or servicing an interrupt.

IEOA, IEOB *Interrupt Enable Out* (outputs, active High). The IEO signal for each channel is used with the accompanying IEI signal to form an interrupt daisy chain. IEO is Low if IEI is Low, an interrupt is under service in the channel, or an interrupt is pending during an interrupt acknowledge cycle.

/TxACKA, /TxACKB *Transmit Acknowledge* (inputs or outputs, active Low). The primary function of these signals is to perform fly-by DMA transfers to the transmit FIFOs. They may also be used as bit inputs or outputs.

/RxACKA, /RxACKB *Receive Acknowledge* (inputs or outputs, active Low). The primary function of these signals is to perform fly-by DMA transfers from the receive FIFOs. They may also be used as bit inputs or outputs.

TxDA, TxDB *Transmit Data* (outputs, active High, tri-state). These signals carry the serial transmit data for each channel.

RxDA, RxDB *Receive Data* (inputs, active High). These signals carry the serial receive data for each channel.

/TxCA, /TxCB *Transmit Clock* (inputs or outputs, active Low). These signals are used as clock inputs for any of the functional blocks within the device. They may also be used as outputs for various transmitter signals or internal clock signals.

/RxCA, /RxCB *Receive Clock* (inputs or outputs, active Low). These signals are used as clock inputs for any of the functional blocks within the device. They may also be used as outputs for various receiver signals or internal clock signals.

/TxREQA, /TxREQB *Transmit Request* (inputs or outputs, active Low). The primary function of these signals is to request DMA transfers to the transmit FIFOs. They may also be used as simple inputs or outputs.

/RxREQA, /RxREQB *Receive Request* (inputs or outputs, active Low). The primary function of these signals is to request DMA transfers from the receive FIFOs. They may also be used as simple inputs or outputs.

/CTSA, /CTSB *Clear To Send* (inputs or outputs, active Low). These signals are used as enables for the respective transmitters. They may also be programmed to generate interrupts on either transition or used as simple inputs or outputs.

/DCDA, /DCDB *Data Carrier Detect* (inputs or outputs, active Low). These signals are used as enables for the respective receivers. They may also be programmed to generate interrupts on either transition or used as simple inputs or outputs.

ARCHITECTURE

The USC internal structure includes two completely independent full-duplex serial channels, each with two baud rate generators, a digital phase-locked loop for clock recovery, transmit and receive character counters and a full-duplex DMA interface. The two serial channels share a common bus interface. The bus interface is designed to provide easy interface to most microprocessors, whether they employ a multiplexed or non-multiplexed, 8-bit or 16-bit bus structure. Each channel is controlled by a set of thirty 16-bit registers, nearly all of which are readable and writable. There is one additional 16-bit register in the bus interface used to configure the nature of the bus interface. The BCR functions are shown in Figure 4.

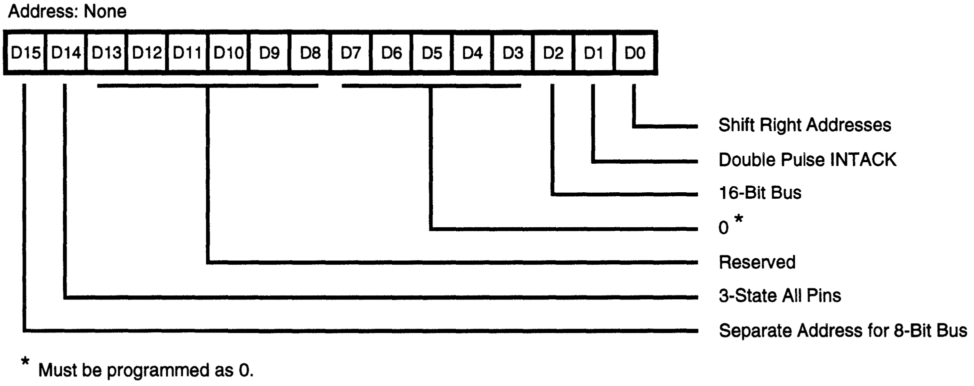


Figure 4. Bus Configuration Register

DATA PATH

Both the transmitter and the receiver in the channel are actually microcoded serial processors. As the data shifts through the transmit or receive shift register, the microcode watches for specific bit patterns, counts bits, and at the appropriate time transfers data to or from the FIFOs. The microcode also checks status and generates status interrupts as appropriate.

FUNCTIONAL DESCRIPTION

The functional capabilities of the USC are described from two different points of view: as a data communications device, it transmits and receives data in a wide variety of data communications protocols; as a microprocessor peripheral, the USC offers such features as read/write registers, a flexible bus interface, DMA interface support and vectored interrupts.

Data Communications Capabilities

The USC provides two independent full-duplex channels programmable for use in any common data communication protocol. The receiver and transmitter modes are completely independent, as are the two channels. Each receiver and transmitter is supported by a 32-byte deep FIFO and a 16-bit message length counter. All modes allow optional even, odd, mark or space parity. Synchronous modes allow the choice of two 16-bit or one 32-bit CRC polynomial. Selection of from one to eight bits-per-character is available in both receiver and transmitter, independently. Error and status conditions are carried with the data in the receive and transmit FIFOs to greatly reduce the CPU overhead required to send or receive a message. Specific, appropriately timed interrupts are available to signal such conditions as overrun, parity error, framing error, end-of-frame, idle line received, sync acquired, transmit underrun, CRC sent, closing sync/flag sent, abort sent, idle line sent and preamble sent. In addition, several useful internal signals such as receive FIFO load, received sync, transmit FIFO read and transmission complete may be sent to pins for use by external circuitry.

Asynchronous Mode. The receiver and transmitter can handle data at a rate of 1/16, 1/32, or 1/64 the clock rate. The receiver rejects start bits less than one-half a bit time and will not erroneously assemble characters following a framing error. The transmitter is capable of sending one, two, or anywhere in the range of 1/16th to two stop bits per character in 1/16 bit increments.

External Sync Mode. The receiver is synchronized to the receive data stream by an externally-supplied signal on a pin for custom protocol applications.

Isochronous Mode. Both transmitter and receiver may operate on start-stop (async) data using a 1x clock. The transmitter can send one or two stop bits.

Asynchronous With Code Violations. This is similar to Isochronous mode except that the start bit is replaced by a three bit-time code violation pattern as in MIL-STD 1553B. The transmitter can send zero, one or two stop bits.

Monosync Mode. In this mode, a single character is used for synchronization. The sync character can be either eight bits long with an arbitrary data character length, or programmed to match the data character length. The receiver is capable of automatically stripping sync characters from the received data stream. The transmitter may be programmed to automatically send CRC on either an underrun or at the end of a programmed message length.

Bisync Mode. This mode is identical to monosync mode except that character synchronization requires two successive characters for synchronization. The two characters need not be identical.

HDLC Mode. In this mode, the receiver recognizes flags, performs optional address matching, accommodates extended address fields, 8- or 16-bit control fields and logical control fields, performs zero deletion and CRC checking. The receiver is capable of receiving shared-zero flags, recognizes the abort sequence and can receive arbitrary length messages. The transmitter automatically sends opening and closing flags, performs zero insertion and can be programmed to send an abort, an extended abort, a flag or CRC and a flag on transmit underrun. The transmitter can also automatically send the closing flag with optional CRC at the end of a programmed message length. Shared-zero flags are selected in the transmitter and a separate character length may be programmed for the last character in the frame.

Bisync Transparent Mode. In this mode, the synchronization pattern is DLE-SYN, programmable selected from either ASCII or EBCDIC encoding. The receiver recognizes control character sequences and automatically handles CRC calculation without CPU intervention. The transmitter can be programmed to send either SYN, DLE-SYN, CRC-SYN, or CRC-DLE-SYN upon underrun and can automatically send the closing DLE-SYN with optional CRC at the end of a programmed message length.

NBIP Mode. This mode is identical to async except that the receiver checks for the status of an additional address/data bit between the parity bit and the stop bit. The value of this bit is FIFO'ed along with the data. This bit is automatically inserted in the transmitter with the value that is FIFO'ed with the transmit data.

802.3 Mode. This mode implements the data format of IEEE 802.3 with 16-bit address compare. In this mode, /DCD and /CTS are used to implement the carrier sense and collision detect interactions with the receiver and transmitter.

Slaved Monosync Mode. This mode is available only in the transmitter and allows the transmitter (operating as though it were in monosync mode) to send data that is byte-synchronous to the data being received by the receiver.

HDLC Loop Mode. This mode is also available only in the transmitter and allows the USC to be used in an HDLC loop configuration. In this mode, the receiver is programmed to operate in HDLC mode so that the transmitter echoes received messages. Upon receipt of a particular bit pattern (actually a sequence of seven consecutive ones) the transmitter breaks the loop and inserts its own frame(s).

Data Encoding

The USC may be programmed to encode and decode the serial data in any of eight different ways as shown in Figure 5. The transmitter encoding method is selected independently of the receiver decoding method.

NRZ. In NRZ, a 1 is represented by a High level for the duration of the bit cell and a 0 is represented by a Low level for the duration of the bit cell.

NRZB. Data is inverted from NRZ.

NRZI-Mark. In NRZI-Mark, a 1 is represented by a transition at the beginning of the bit cell. That is, the level present in the preceding bit cell is reversed. A 0 is represented by the absence of a transition at the beginning of the bit cell.

NRZI-Space. In NRZI-Space, a 1 is represented by the absence of a transition at the beginning of the bit cell. That is, the level present in the preceding bit cell is maintained. A 0 is represented by a transition at the beginning of the bit cell.

Biphase-Mark. In Biphase-Mark, a 1 is represented by a transition at the beginning of the bit cell and another transition at the center of the bit cell. A 0 is represented by a transition at the beginning of the bit cell only.

Biphase-Space. In Biphase-Space, a 1 is represented by a transition at the beginning of the bit cell only. A 0 is represented by a transition at the beginning of the bit cell and another transition at the center of the bit cell.

Biphase-Level. In Biphase-Level, a 1 is represented by a High during the first half of the bit cell and a Low during the second half of the bit cell. A 0 is represented by a Low during the first half of the bit cell and a High during the second half of the bit cell.

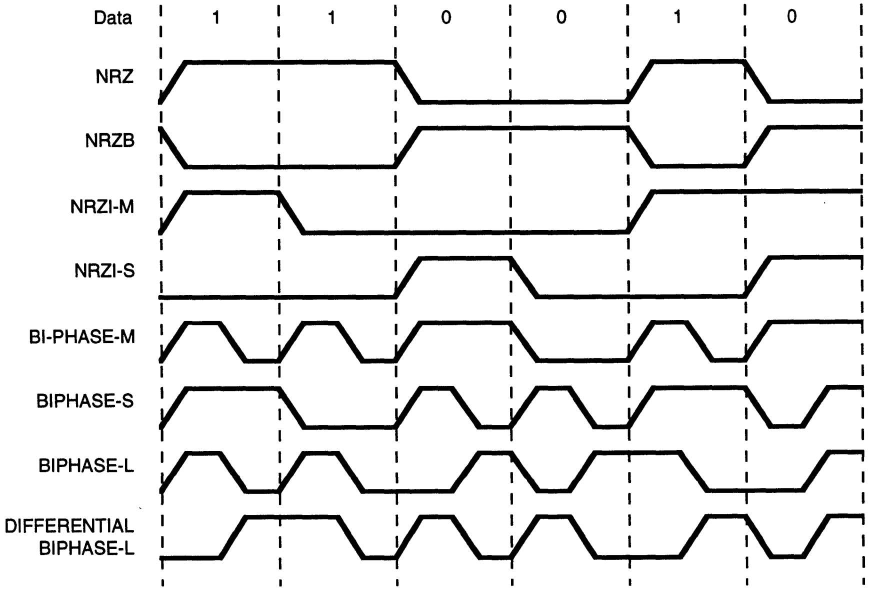


Figure 5. Data Encoding

FUNCTIONAL DESCRIPTION (Continued)

Differential Biphase-Level. In Differential Biphase-Level, a 1 is represented by a transition at the center of the bit cell, with the opposite polarity from the transition at the center of the preceding bit cell. A 0 is represented by a transition at the center of the bit cell with the same polarity as the transition at the center of the preceding bit cell. In both cases there may be transitions at the beginning of the bit cell to set up the level required to make the correct center transition.

Character Counters

Each channel in the USC contains a 16-bit character counter for both receiver and transmitter. The receive character counter may be preset either under software control or automatically at the beginning of a receive message. The counter decrements with each receive character and at the end of the receive message the current value in the counter is automatically loaded into a four-deep FIFO. This allows DMA transfer of data to proceed without CPU intervention at the end of a received message, as the values in the FIFO allow the CPU to determine message boundaries in memory. Similarly, the transmit character counter is loaded either under software control or automatically at the beginning of a transmit message. The counter is decremented with each write to the transmit FIFO. When the counter has decremented to zero, and that byte is sent, the transmitter automatically terminates the message in the appropriate fashion (usually CRC and the closing flag or sync character) without requiring CPU intervention.

Baud Rate Generators

Each channel in the USC contains two baud rate generators. Each generator consists of a 16-bit time constant register and a 16-bit down counter. In operation, the counter decrements with each baud rate generator clock, with the time constant automatically reloaded when the count reaches zero. The output of the baud rate generator toggles when the counter reaches a count of one-half of the time constant and again when the counter reaches zero. A new time constant may be written at any time but the new value will not take effect until the next load of the counter. The outputs of both baud rate generators are sent to the clock multiplexer for use internally or externally. The baud rate generator output frequency is related to the baud rate generator input clock frequency by the following formula:

Output frequency = Input frequency/(time constant + 1)

This allows an output frequency in the range of 1 to 1/65536 of the input frequency, inclusive.

Digital Phase-Locked Loop

Each channel in the USC contains a Digital Phase-Locked Loop (DPLL) to recover clock information from a data stream with NRZI or Biphase encoding. The DPLL is driven by a clock that is nominally 8, 16 or 32 times the receive data rate. The DPLL uses this clock, along the data stream, to construct a clock for the data. This clock may then be routed to the receiver, transmitter, or both, or to a pin for use externally. In all modes, the DPLL counts the input clock to create nominal bit times. As the clock is counted, the DPLL watches the incoming data stream for transitions. Whenever a transition is detected, the DPLL makes a count adjustment (during the next counting cycle), to produce an output clock which tracks the incoming bit cells. The DPLL provides properly phased transmit and receive clocks to the clock multiplexer.

Counters

Each channel contains two 5-bit counters, which are programmed to divide an input clock by 4, 8, 16 or 32. The inputs of these two counters are sent to the clock multiplexer. The counters are used as prescalers for the baud rate generators, or to provide a stable transmit clock from a common source when the DPLL is providing the receive clock.

Clock Multiplexer

The clock multiplexer in each channel selects the clock source for the various blocks in the channel and selects an internal clock signal to potentially be sent to either the /RxC or /TxC pin.

Test Modes

The USC can be programmed for local loopback or auto echo operation. In local loopback, the output of the transmitter is internally routed to the input of the receiver. This allows testing of the USC data paths without any external logic. Auto echo connects the RxD pin directly to the TxD pin. This is useful for testing serial links external to the USC.

I/O INTERFACE CAPABILITIES

The USC offers the choice of polling, interrupt (vectored or non-vectored) and block transfer modes to transfer data, status and control information to and from the CPU.

Polling

All interrupts are disabled. The registers in the USC are automatically updated to reflect current status. The CPU polls the Daisy Chain Control Register (DCCR) to determine status changes and then reads the appropriate status register to find and respond to the change in status. USC status bits are grouped according to function to simplify this software action.

Interrupt

When a USC responds to an interrupt acknowledge from the CPU, an interrupt vector may be placed on the data bus. This vector is held in the Interrupt Vector Register (IVR). To speed interrupt response time, the USC modifies three bits in this vector to indicate which type of interrupt is being requested.

Each of the six sources of interrupts in each channel of the USC (Receive Status, Receive Data, Transmit Status, Transmit Data, I/O Status and Device Status) has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt-Under-Service (IUS) and Interrupt Enable (IE). If the IE bit for a given source is set, that source can request interrupts. Note that individual sources within the six groups also have interrupt enable bits which are set for the particular source. In addition, there is a Master Interrupt Enable (MIE) bit in each channel which globally enables or disables interrupts within the channel.

The other two bits are related to the interrupt priority chain. A channel in the USC may request an interrupt only when no higher priority interrupt source is requesting one, e.g., when IEI is High for the channel. In this case the channel activates the /INT signal. The CPU then responds with an interrupt acknowledge cycle, and the interrupting channel places a vector on the data bus.

In the USC, the IP bit signals that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority within the channel and external to the channel are prevented from requesting interrupts. The internal interrupt sources are inhibited by the state of the internal daisy chain, while lower priority devices are inhibited by the IEO output of the channel being pulled Low and propagated to subsequent peripherals. An IUS bit is set during an interrupt acknowledge cycle if there are no higher priority devices requesting interrupts.

There are six sources of interrupt in each channel: Receive Status, Receive Data, Transmit Status, Transmit Data, I/O Status and Device Status, prioritized in that order within the channel. There are six sources of Receive Status interrupt, each individually enabled: exited hunt, idle line, break/abort, code violation/end-of-transmission/end-of-frame, parity error and overrun error. The Receive Data interrupt is generated whenever the receive FIFO fills with data beyond the level programmed in the Receive Interrupt Control Register (RICR).

There are six sources of Transmit Status interrupt, each individually enabled: preamble sent, idle line sent, abort sent, end-of-frame/end-of-transmission sent, CRC sent and underrun error. The Transmit Data interrupt is generated whenever the transmit FIFO empties below the level programmed in the Transmit Interrupt Control Register (TICR). The I/O Status interrupt serves to report transitions on any of six pins. Interrupts are generated on either or both edges with separate selection and enables for each pin. The pins programmed to generate I/O Status interrupts are /RxC, /TxC, /RxREQ, /TxREQ, /DCD and /CTS. These interrupts are independent of the programmed function of the pins. The Device Status interrupt has four separately enabled sources: receive character count FIFO overflow, DPLL sync acquired, BRG1 zero count and BRGO zero count.

Block Transfer Mode

The USC accommodates block transfers through DMA through the /RxREQ, /TxREQ, /RxACK and /TxACK pins. The /RxREQ signal is activated when the fill level of the receive FIFO exceeds the value programmed in the RICR. The DMA may respond with either a normal bus transaction or by activating the /RxACK pin to read the data directly (fly-by transfer). The /TxREQ signal is activated when the empty level of the transmit FIFO falls below the value programmed in the TICR. The DMA may respond either with a normal bus transaction or by activating the /TxACK pin to write the data directly (fly-by transfer). The /RxACK and /TxACK pin functions for this mode are controlled by the Hardware Configuration Register (HCR). Then using the /RxACK and /TxACK pins to transfer data, no chip select is necessary; these are dedicated strobes for the appropriate FIFO.

PROGRAMMING

The registers in each USC channel are programmed by the system to configure the channels. Before this can occur, however, the system must program the bus interface by writing to the Bus Configuration Register (BCR). The BCR has no specific address and is only accessible immediately after a hardware reset of the device. The first write to the USC, after a hardware reset, programs the BCR. From that time on the normal channel registers may be accessed. No specific address need be presented to the USC for the BCR write; the USC knows that the first write after a hardware reset is destined for the BCR.

In the multiplexed bus case, all registers are directly addressable through the address latched by /AS at the beginning of a bus transaction. The address is decoded from either AD6-AD0 or AD7-AD1. This is controlled by the Shift Right/Shift Left bit in the BCR. The address maps for these two cases are shown in Table 1. The D//C pin is still used to directly access the receive and transmit data registers (RDR and TDR) in the multiplexed bus; if D//C is High the address latched by /AS is ignored and an access of RDR or TDR is performed.

In the non-multiplexed bus case, the registers in each channel are accessed indirectly using the address pointer in the Channel Command/Address Register (CCAR) in each channel. The address of the desired register is first written to the CCAR and then the selected register is accessed; the pointer in the CCAR is automatically cleared after this access. The RDR and TDR are accessed directly using the D//C pin, without disturbing the contents of the pointer in the CCAR.

Table 1. Multiplexed Bus Address Assignments

Address Signal	Shift Left	Shift Right
Byte//Word Access	AD7	AD6
Address 4	AD6	AD5
Address 3	AD5	AD4
Address 2	AD4	AD3
Address 1	AD3	AD2
Address 0	AD2	AD1
Upper//Lower Byte Select	AD1	AD0

There are two important things to note about the USC. First, the Channel Reset bit in the CCAR places the channel in the reset state. To exit this reset state either a word of all zeros must be written to the CCAR (16-bit bus) or a byte of all zeros must be written to the lower byte of the CCAR (8-bit bus). The second thing to note is that after reset, the transmit and receive clocks are not connected. The first thing that should be done in any initialization sequence is a write to the Clock Mode Control Register (CMCR) to select a clock source for the receiver and transmitter.

The register addressing is shown in Table 2. and the bit assignments for the registers are shown in Figure 6.

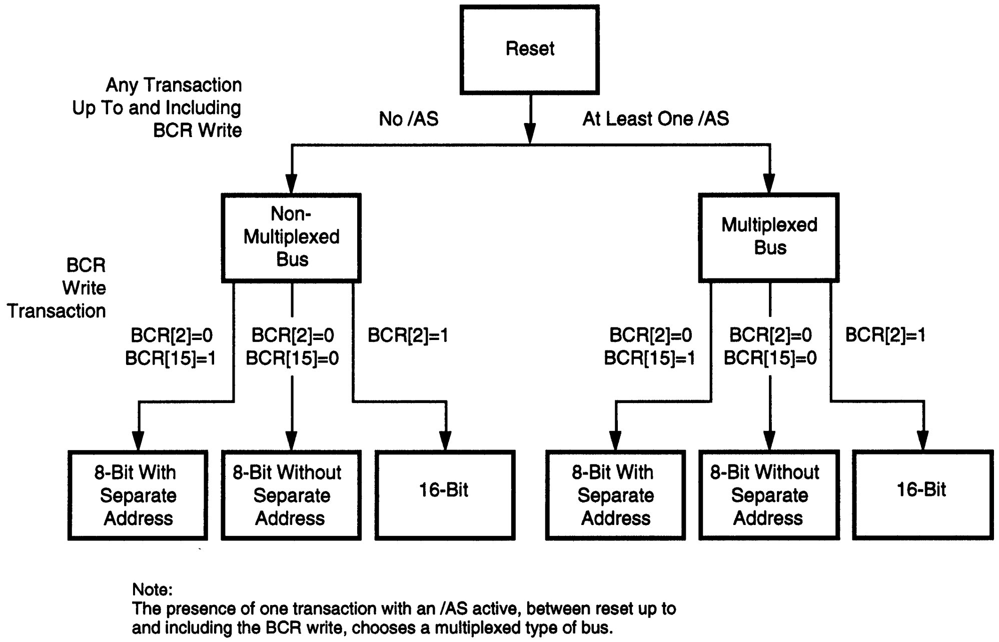

Reset
Any Transaction
Up To and Including
BCR Write
No /AS
At Least One /AS
Non-Multiplexed Bus
Multiplexed Bus
BCR
Write
Transaction
BCR[2]=0
BCR[15]=1
BCR[2]=0
BCR[15]=0
BCR[2]=1
BCR[2]=0
BCR[15]=1
BCR[2]=0
BCR[15]=0
BCR[2]=1
8-Bit With Separate Address
8-Bit Without Separate Address
16-Bit
8-Bit With Separate Address
8-Bit Without Separate Address
16-Bit
Note:
The presence of one transaction with an /AS active, between reset up to and including the BCR write, chooses a multiplexed type of bus.

Figure 6. BCR Reset Sequence and Bit Assignments

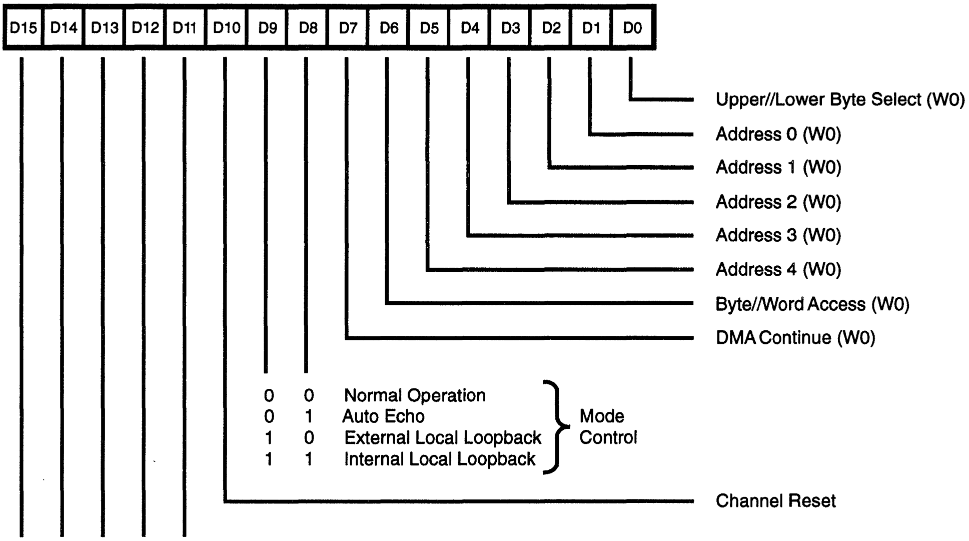
Table 2. Register Address List

Address A4-A0		
00000	CCAR	Channel Command/Address Register
00001	CMR	Channel Mode Register
00010	CCSR	Channel Command/Status Register
00011	CCR	Channel Control Register
00110	TMDR	Test Mode Data Register
00111	TMCR	Test Mode Control Register
01000	CMCR	Clock Mode Control Register
01001	HCR	Hardware Configuration Register
01010	IVR	Interrupt Vector Register
01011	IOCR	I/O Control Register
01100	ICR	Interrupt Control Register
01101	DCCR	Daisy-Chain Control Register
01110	MISR	Misc Interrupt Status Register
01111	SICR	Status Interrupt Control Register
1X000	RDR	Receive Data Register (Read Only)
10001	RMR	Receive Mode Register

Address A4-A0		
10010	RCSR	Receive Command/Status Register
10011	RICR	Receive Interrupt Control Register
10100	RSR	Receive Sync Register
10101	RCLR	Receive Count Limit Register
10110	RCCR	Receive Character Count Register
10111	TC0R	Time Constant 0 Register
1X000	TDR	Transmit Data Register (Write Only)
11001	TMR	Transmit Mode Register
11010	TCSR	Transmit Command/Status Register
11011	TICR	Transmit Interrupt Control Register
11100	TSR	Transmit Sync Register
11101	TCLR	Transmit Count Limit Register
11110	TCCR	Transmit Character Count Register
11111	TC1R	Time Constant 1 Register
XXXXX	BCR	Bus Configuration Register

CONTROL REGISTERS

Address: 00000



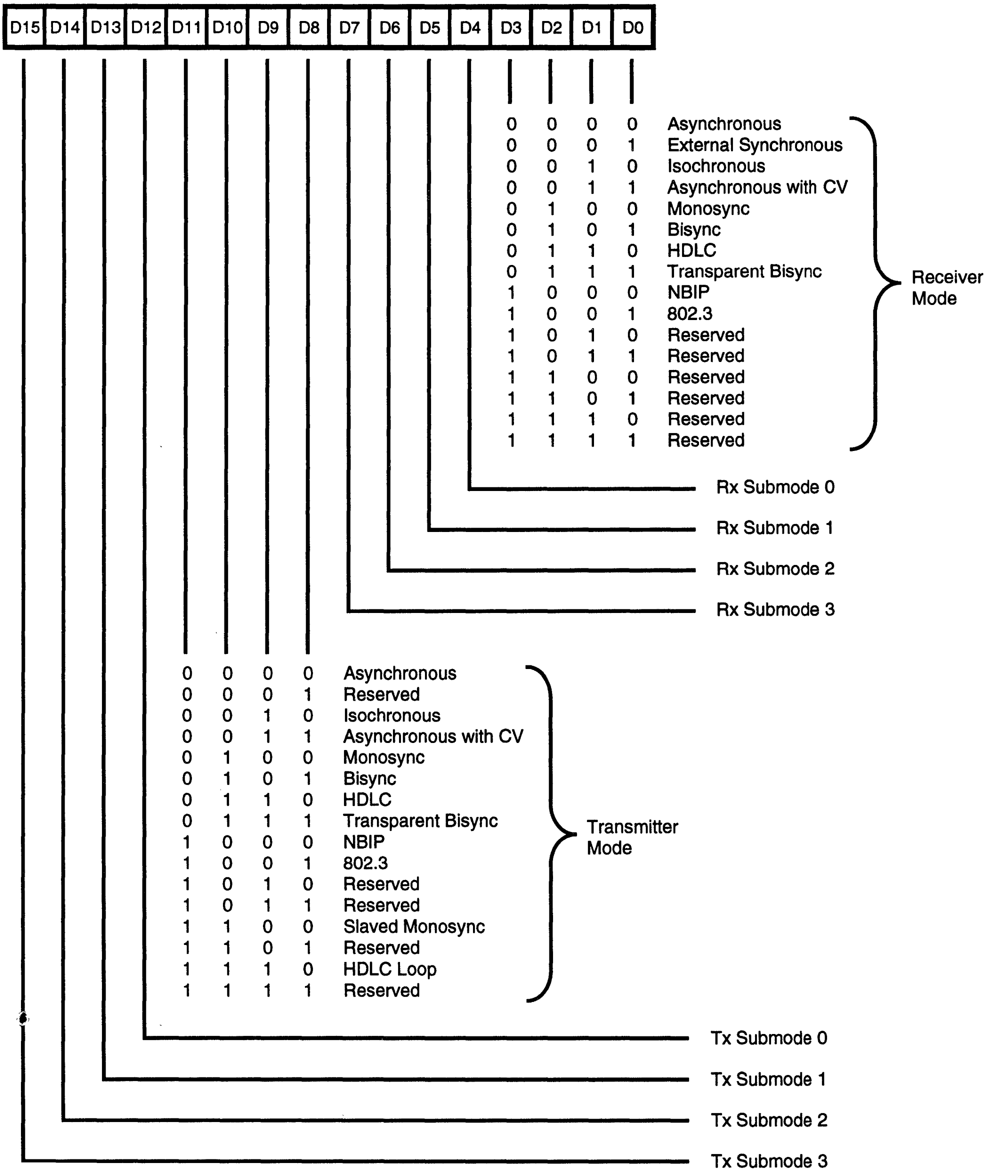
D9	D8	Mode Control
0	0	Normal Operation
0	1	Auto Echo
1	0	External Local Loopback
1	1	Internal Local Loopback

D15	D14	D13	D12	D11	Channel Command (WO)
0	0	0	0	0	Null Command
0	0	0	0	1	Reserved
0	0	0	1	0	Reset Highest IUS
0	0	0	1	1	Reserved
0	0	1	0	0	Trigger Channel Load DMA
0	0	1	0	1	Trigger Rx DMA
0	0	1	1	0	Trigger Tx DMA
0	0	1	1	1	Trigger Rx & Tx DMA
0	1	0	0	0	Reserved
0	1	0	0	1	Rx FIFO Purge
0	1	0	1	0	Tx FIFO Purge
0	1	0	1	1	Rx & Tx FIFO Purge
0	1	1	0	0	Reserved
0	1	1	0	1	Load Rx Character Count
0	1	1	1	0	Load Tx Character Count
0	1	1	1	1	Load Rx & Tx Character Count
1	0	0	0	0	Reserved
1	0	0	0	1	Load TC0
1	0	0	1	0	Load TC1
1	0	0	1	1	Load TC0 & TC1
1	0	1	0	0	Select Serial Data LSB First *
1	0	1	0	1	Select Serial Data MSB First
1	0	1	1	0	Select Straight Memory Data *
1	0	1	1	1	Select Swapped Memory Data
1	1	0	0	0	Reserved
1	1	0	0	1	Rx Purge
1	1	0	1	0	Reserved
1	1	0	1	1	Reserved
1	1	1	0	0	Reserved
1	1	1	0	1	Reserved
1	1	1	1	0	Reserved
1	1	1	1	1	Reserved

* Selected Upon Reset

Figure 7. Channel Command/Address Register

Address: 00001



Receiver Mode

D3	D2	D1	D0	
0	0	0	0	Asynchronous
0	0	0	1	External Synchronous
0	0	1	0	Isochronous
0	0	1	1	Asynchronous with CV
0	1	0	0	Monosync
0	1	0	1	Bisync
0	1	1	0	HDLC
0	1	1	1	Transparent Bisync
1	0	0	0	NBIP
1	0	0	1	802.3
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved

D4 Rx Submode 0
D5 Rx Submode 1
D6 Rx Submode 2
D7 Rx Submode 3

Transmitter Mode

D11	D10	D9	D8	
0	0	0	0	Asynchronous
0	0	0	1	Reserved
0	0	1	0	Isochronous
0	0	1	1	Asynchronous with CV
0	1	0	0	Monosync
0	1	0	1	Bisync
0	1	1	0	HDLC
0	1	1	1	Transparent Bisync
1	0	0	0	NBIP
1	0	0	1	802.3
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Slaved Monosync
1	1	0	1	Reserved
1	1	1	0	HDLC Loop
1	1	1	1	Reserved

D12 Tx Submode 0
D13 Tx Submode 1
D14 Tx Submode 2
D15 Tx Submode 3

Figure 8. Channel Mode Register

CONTROL REGISTERS (Continued)

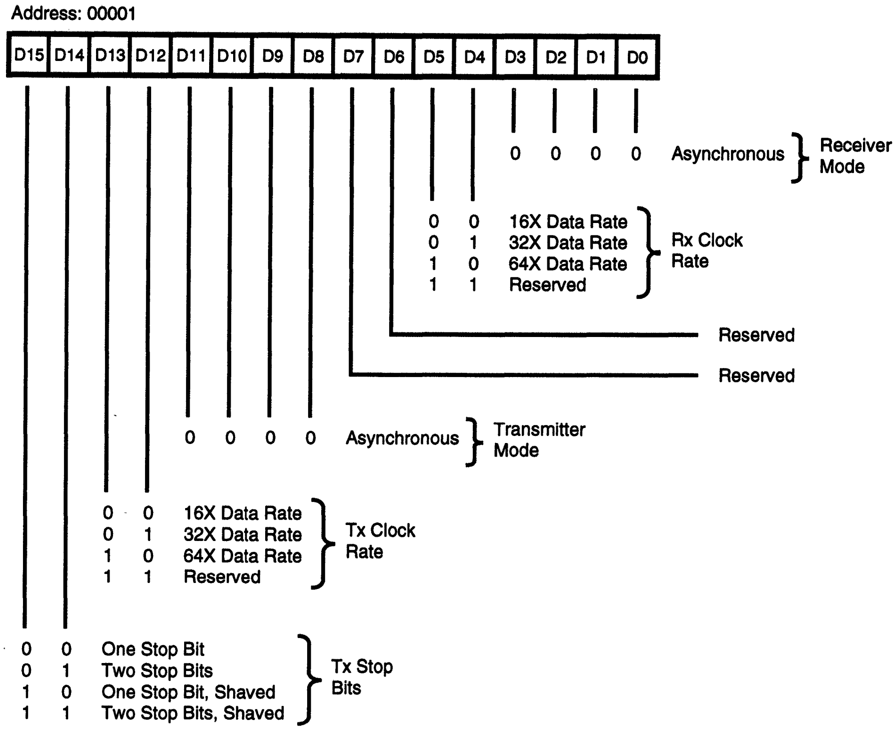


Figure 9. Channel Mode Register, Asynchronous Mode

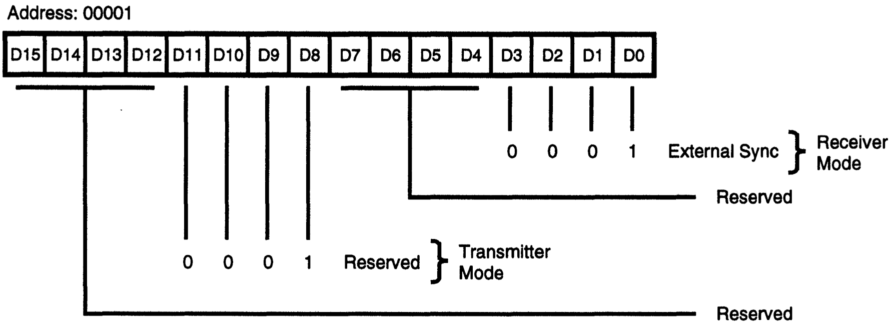


Figure 10. Channel Mode Register, External Sync Mode

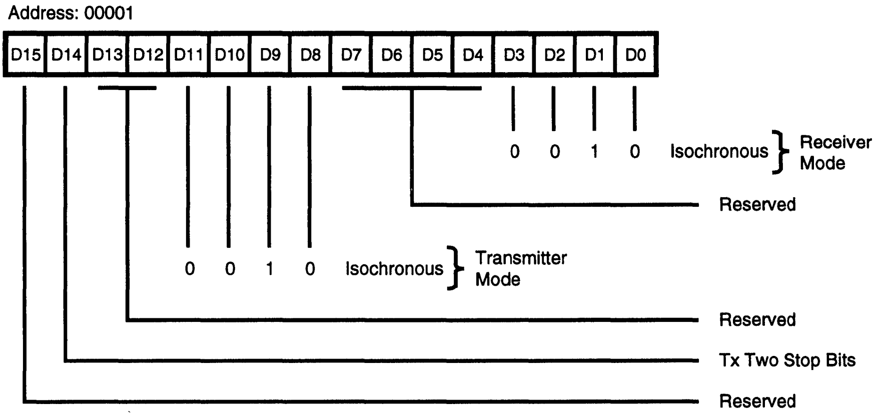


Figure 11. Channel Mode Register, Isochronous Mode

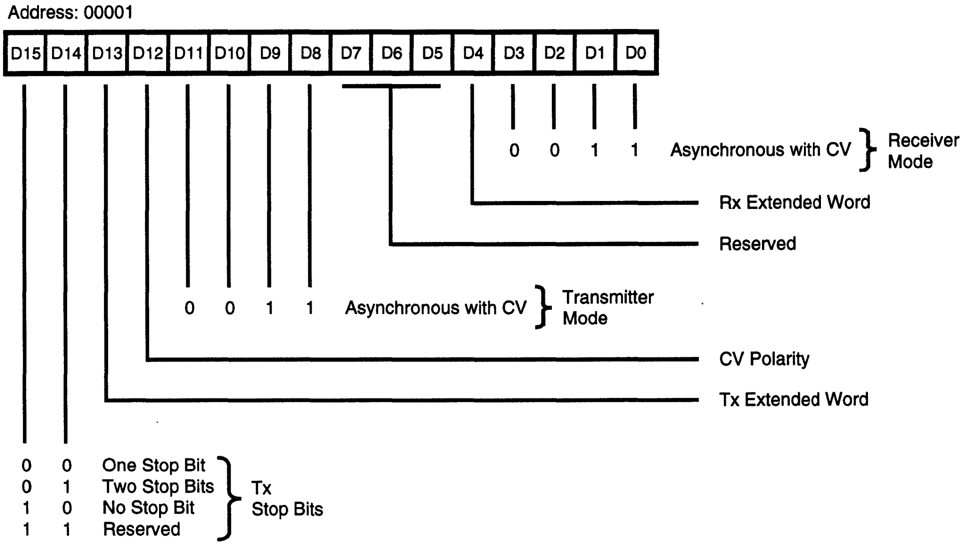


Figure 12. Channel Mode Register, Asynchronous Mode with Code Violation (MIL STD 1553)

CONTROL REGISTERS (Continued)

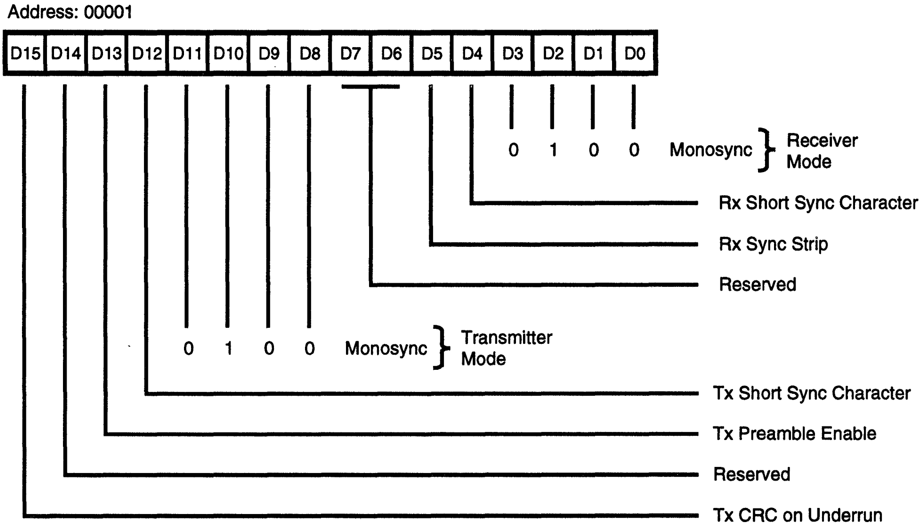


Figure 13. Channel Mode Register, Monosync Mode

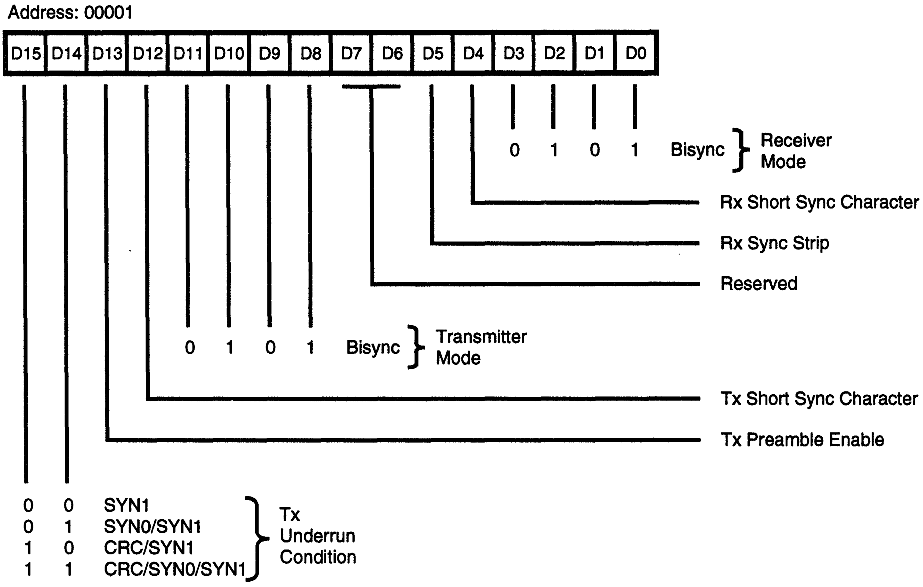


Figure 14. Channel Mode Register, Bisync Mode

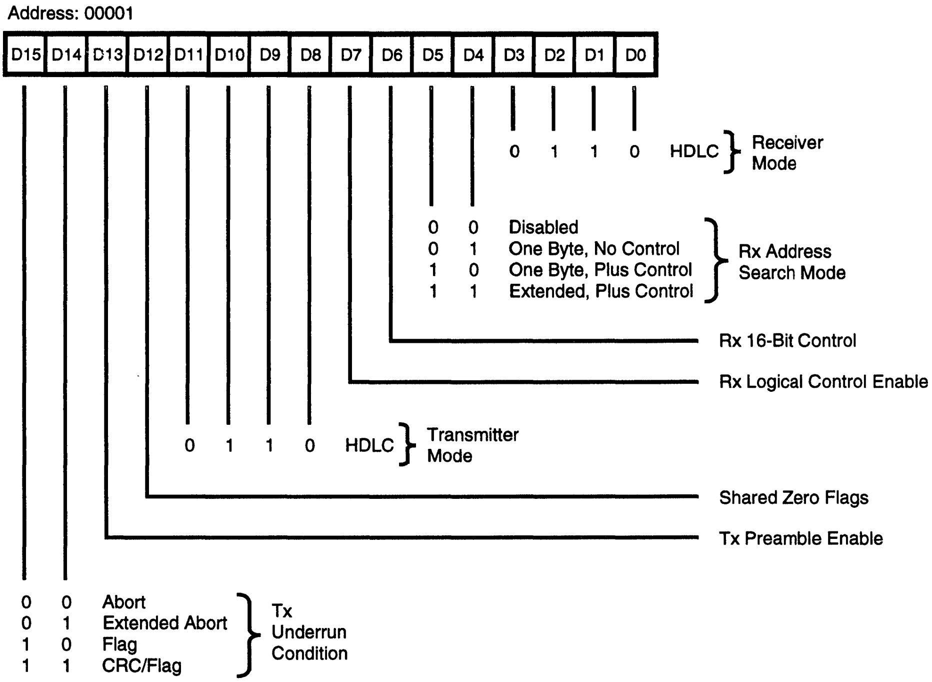


Figure 15. Channel Mode Register, HDLC Mode

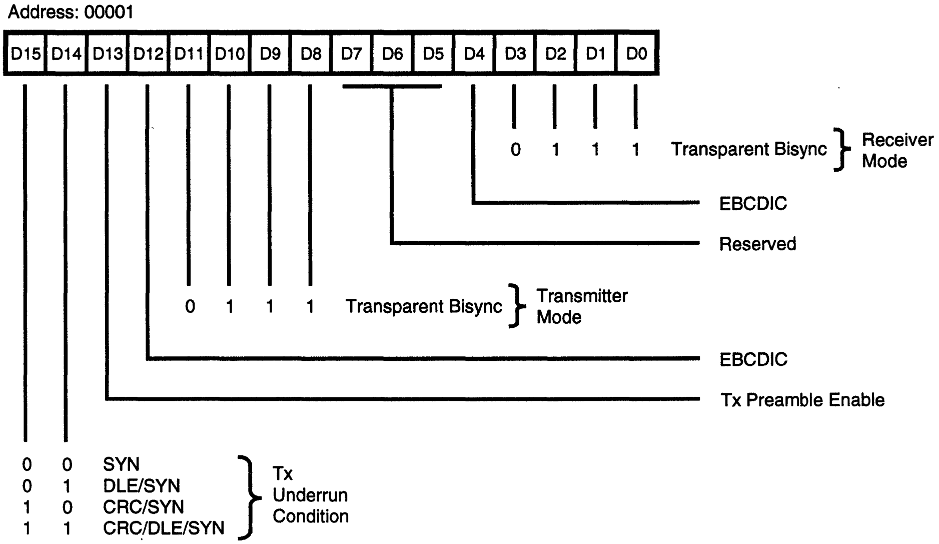


Figure 16. Channel Mode Register, Transparent Bisync Mode

CONTROL REGISTERS (Continued)

Address: 00001

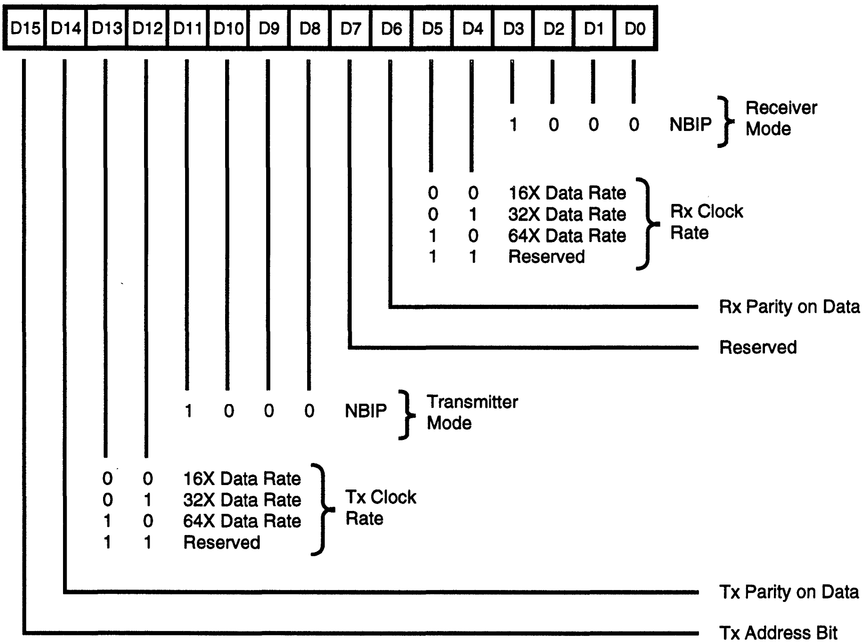


Figure 17. Channel Mode Register, NBIP Mode

Address: 00001

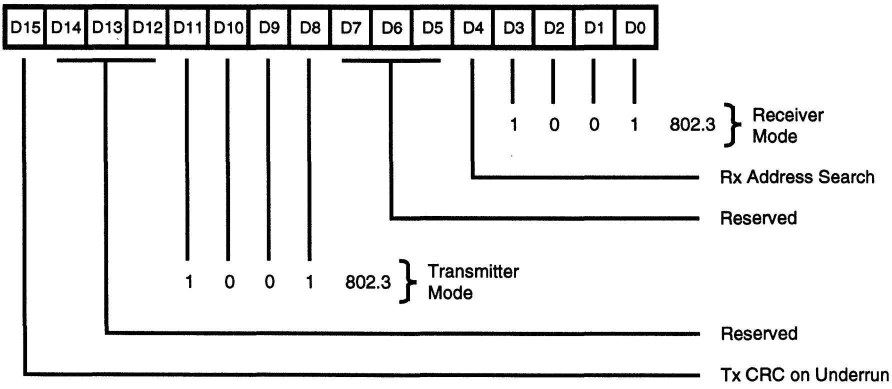


Figure 18. Channel Mode Register, 802.3 Mode

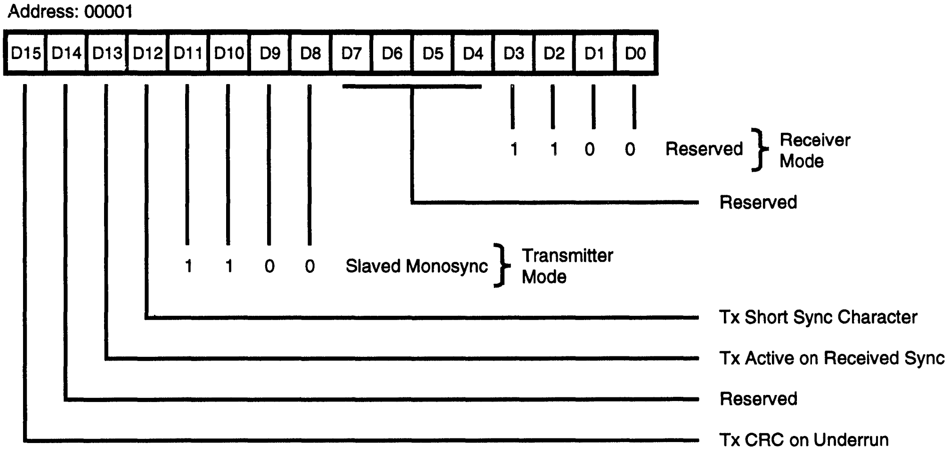


Figure 19. Channel Mode Register, Slaved Monosync Mode

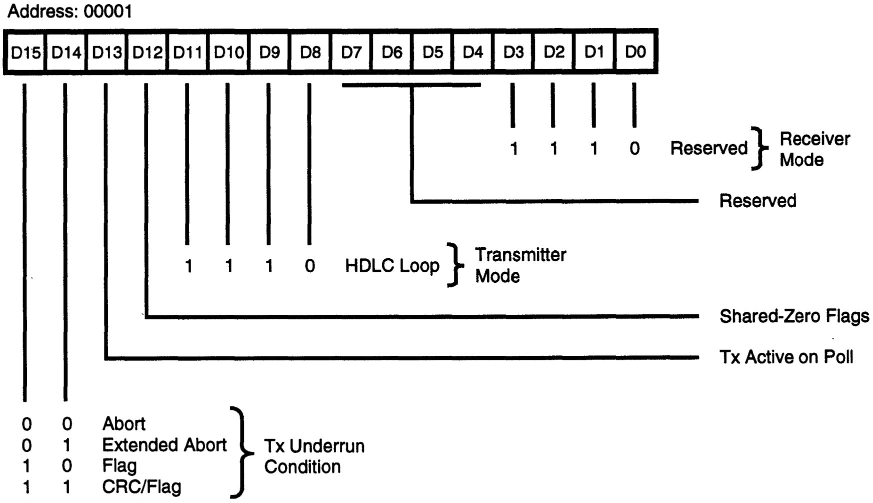


Figure 20. Channel Mode Register, HDLC Loop Mode

CONTROL REGISTERS (Continued)

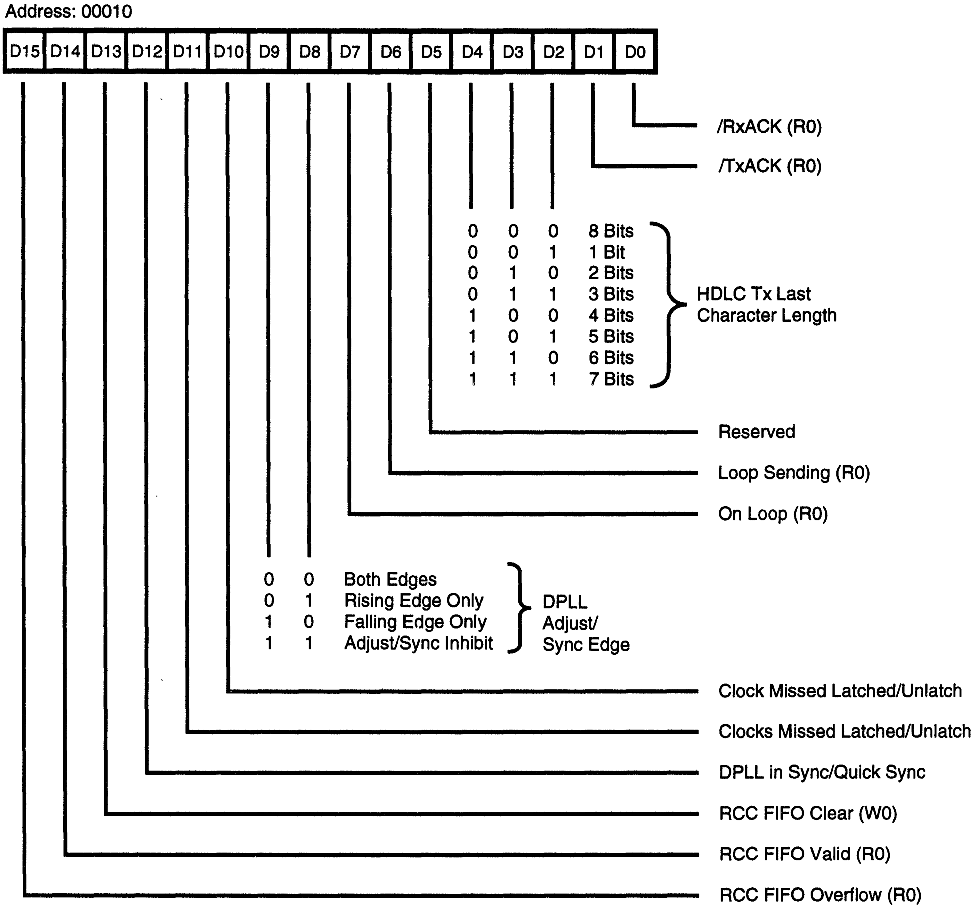


Figure 21. Channel Command/Status Register

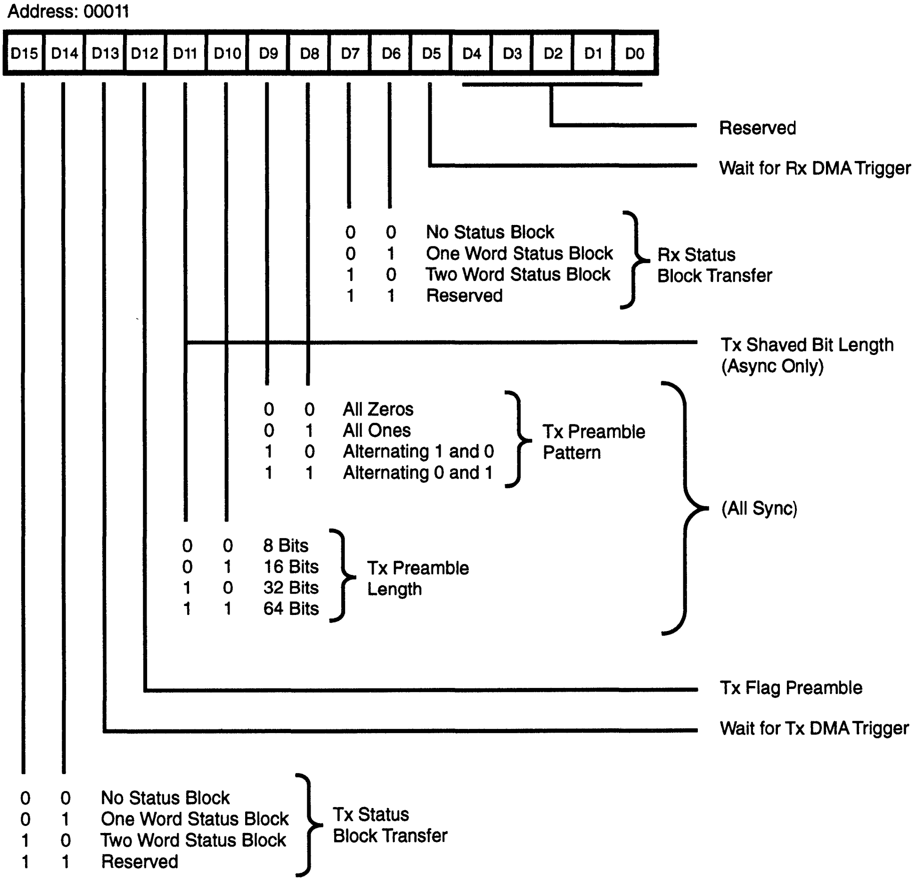


Figure 22. Channel Control Register

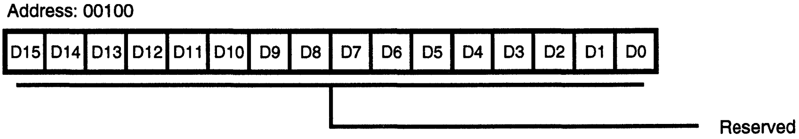


Figure 23. Primary Reserved Register

CONTROL REGISTERS (Continued)

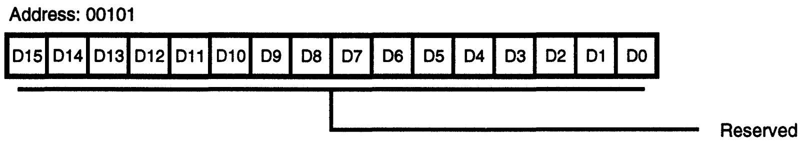


Figure 24. Secondary Reserved Register

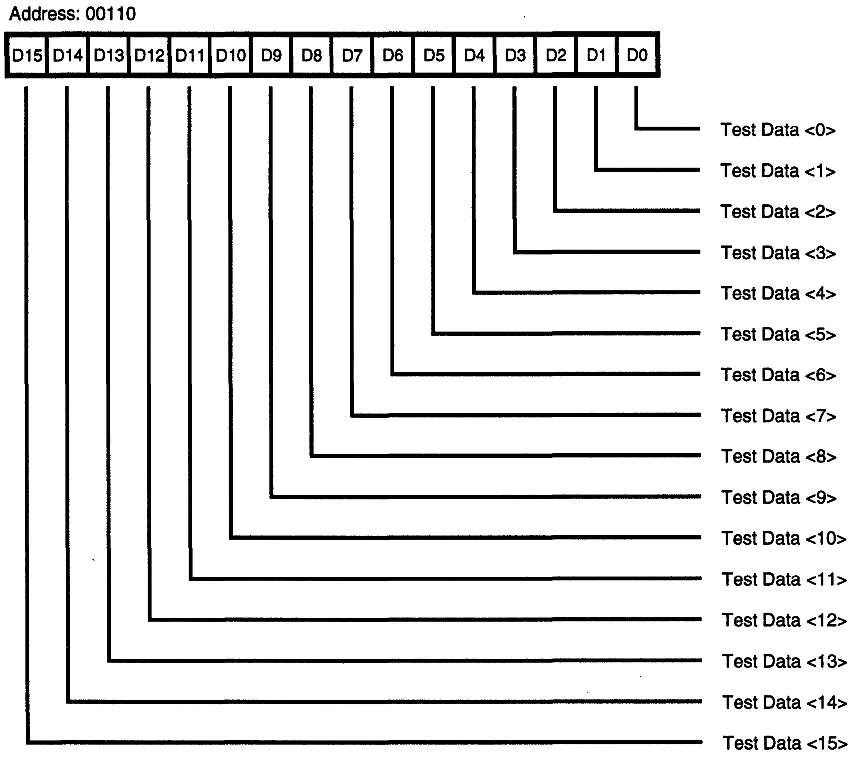
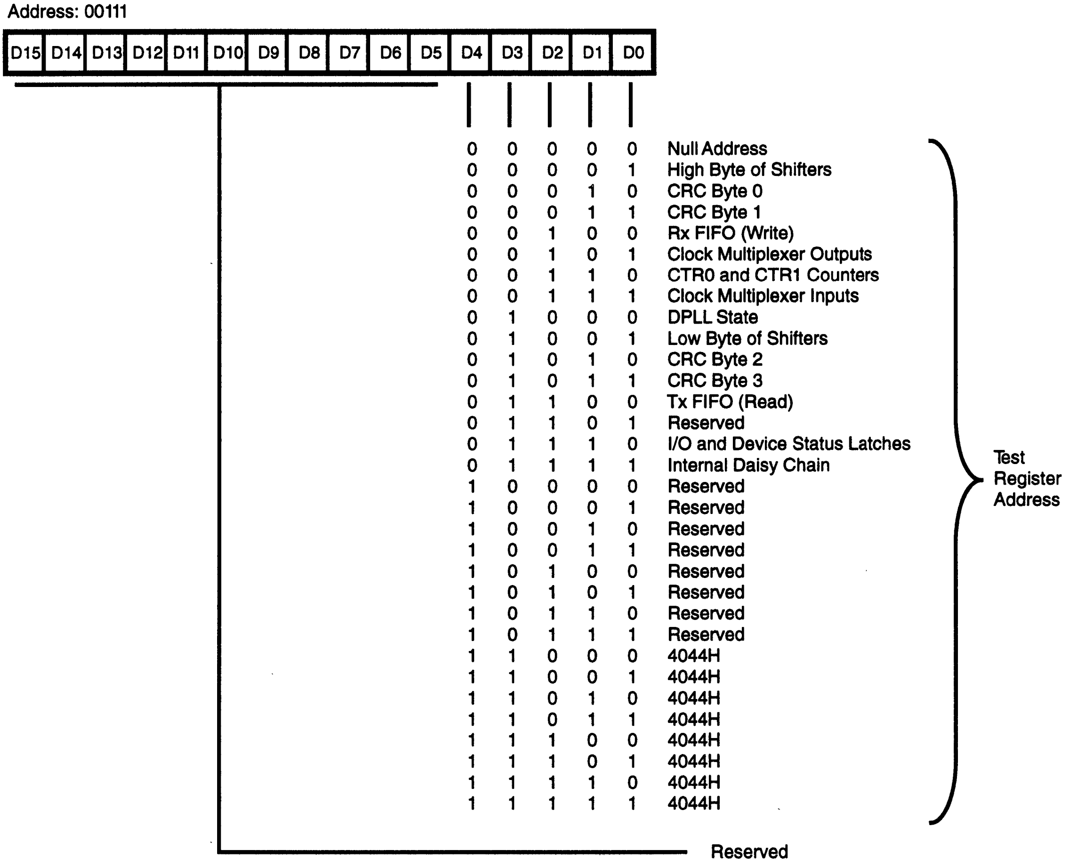


Figure 25. Test Mode Data Register

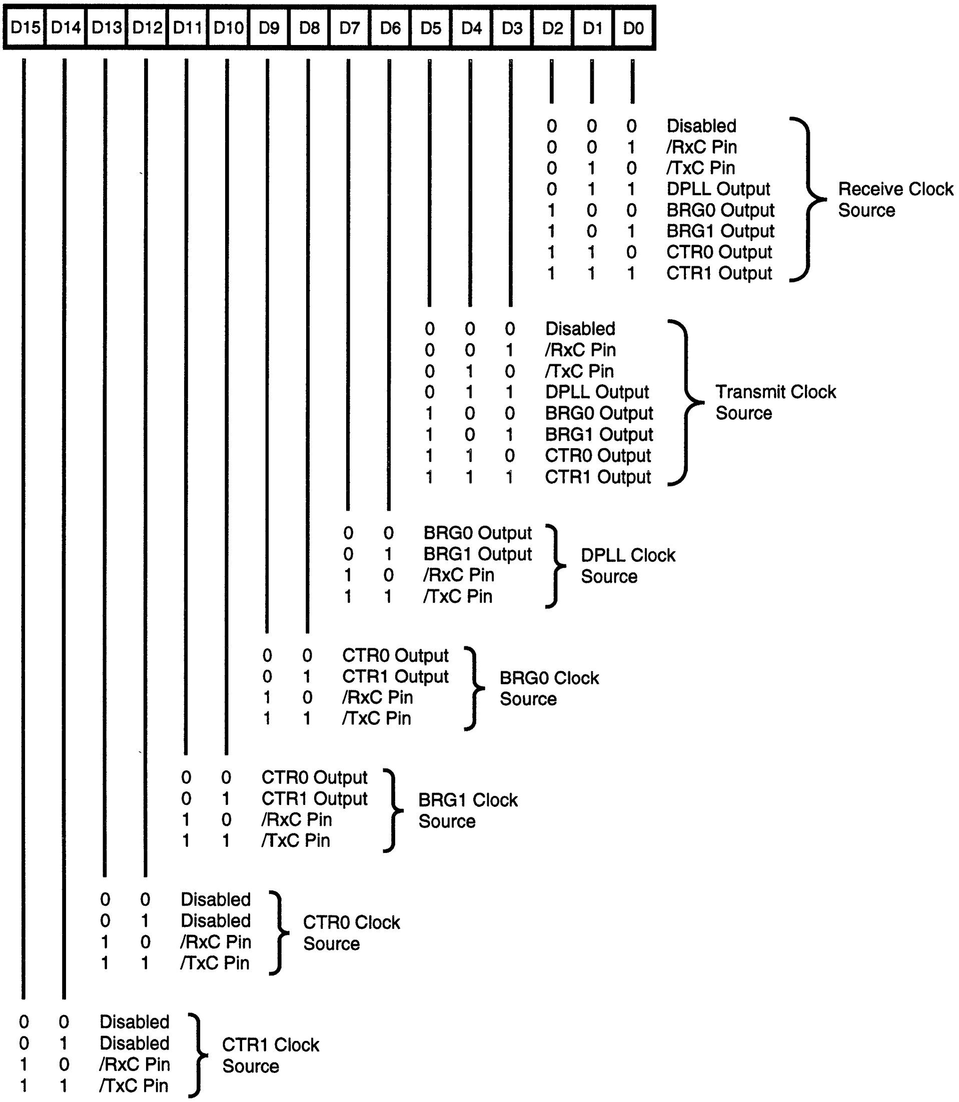


D4	D3	D2	D1	D0	Test Register Address
0	0	0	0	0	Null Address
0	0	0	0	1	High Byte of Shifters
0	0	0	1	0	CRC Byte 0
0	0	0	1	1	CRC Byte 1
0	0	1	0	0	Rx FIFO (Write)
0	0	1	0	1	Clock Multiplexer Outputs
0	0	1	1	0	CTR0 and CTR1 Counters
0	0	1	1	1	Clock Multiplexer Inputs
0	1	0	0	0	DPLL State
0	1	0	0	1	Low Byte of Shifters
0	1	0	1	0	CRC Byte 2
0	1	0	1	1	CRC Byte 3
0	1	1	0	0	Tx FIFO (Read)
0	1	1	0	1	Reserved
0	1	1	1	0	I/O and Device Status Latches
0	1	1	1	1	Internal Daisy Chain
1	0	0	0	0	Reserved
1	0	0	0	1	Reserved
1	0	0	1	0	Reserved
1	0	0	1	1	Reserved
1	0	1	0	0	Reserved
1	0	1	0	1	Reserved
1	0	1	1	0	Reserved
1	0	1	1	1	Reserved
1	1	0	0	0	4044H
1	1	0	0	1	4044H
1	1	0	1	0	4044H
1	1	0	1	1	4044H
1	1	1	0	0	4044H
1	1	1	0	1	4044H
1	1	1	1	0	4044H
1	1	1	1	1	4044H

Figure 26. Test Mode Control Register

CONTROL REGISTERS (Continued)

Address: 01000



D2	D1	D0	Receive Clock Source
0	0	0	Disabled
0	0	1	/RxC Pin
0	1	0	/TxC Pin
0	1	1	DPLL Output
1	0	0	BRG0 Output
1	0	1	BRG1 Output
1	1	0	CTR0 Output
1	1	1	CTR1 Output

D5	D4	D3	Transmit Clock Source
0	0	0	Disabled
0	0	1	/RxC Pin
0	1	0	/TxC Pin
0	1	1	DPLL Output
1	0	0	BRG0 Output
1	0	1	BRG1 Output
1	1	0	CTR0 Output
1	1	1	CTR1 Output

D7	D6	DPLL Clock Source
0	0	BRG0 Output
0	1	BRG1 Output
1	0	/RxC Pin
1	1	/TxC Pin

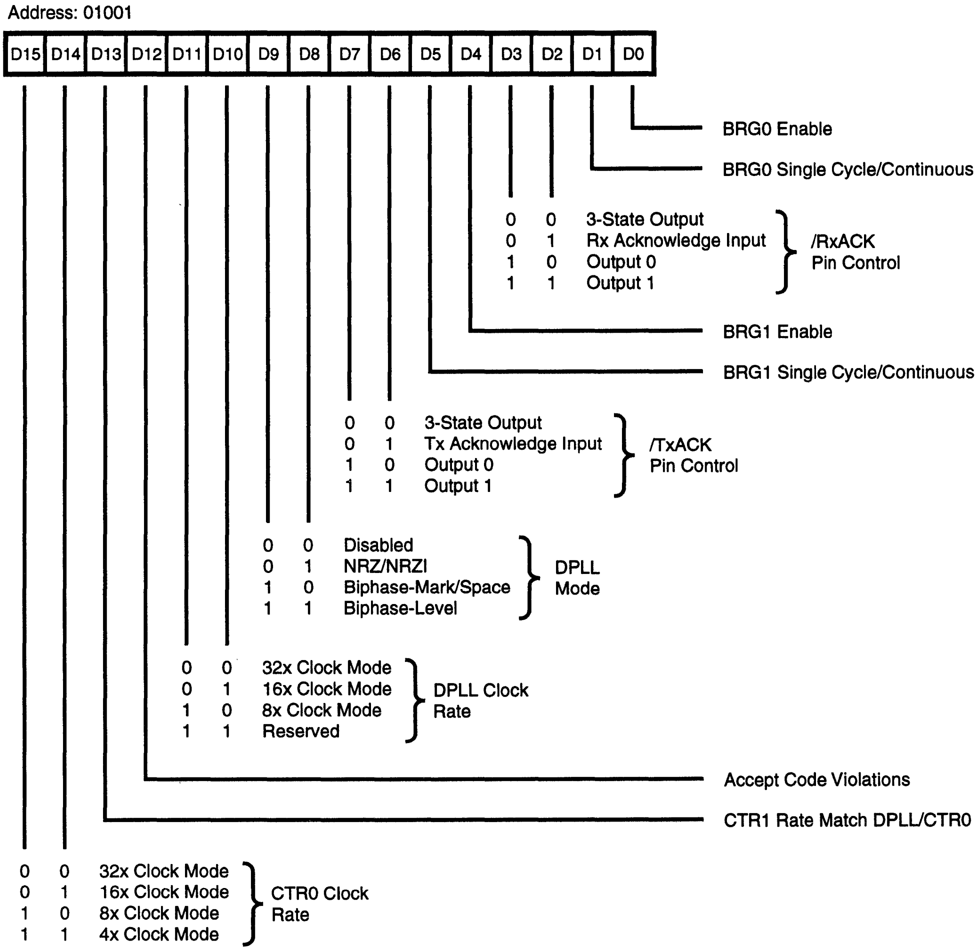
D9	D8	BRG0 Clock Source
0	0	CTR0 Output
0	1	CTR1 Output
1	0	/RxC Pin
1	1	/TxC Pin

D11	D10	BRG1 Clock Source
0	0	CTR0 Output
0	1	CTR1 Output
1	0	/RxC Pin
1	1	/TxC Pin

D13	D12	CTR0 Clock Source
0	0	Disabled
0	1	Disabled
1	0	/RxC Pin
1	1	/TxC Pin

D15	D14	CTR1 Clock Source
0	0	Disabled
0	1	Disabled
1	0	/RxC Pin
1	1	/TxC Pin

Figure 27. Clock Mode Control Register



Address: 01001

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

D0: BRG0 Enable

D1: BRG0 Single Cycle/Continuous

D3 D2: /RxACK Pin Control

D3	D2	
0	0	3-State Output
0	1	Rx Acknowledge Input
1	0	Output 0
1	1	Output 1

D4: BRG1 Enable

D5: BRG1 Single Cycle/Continuous

D7 D6: /TxACK Pin Control

D7	D6	
0	0	3-State Output
0	1	Tx Acknowledge Input
1	0	Output 0
1	1	Output 1

D9 D8: DPLL Mode

D9	D8	
0	0	Disabled
0	1	NRZ/NRZI
1	0	Biphase-Mark/Space
1	1	Biphase-Level

D11 D10: DPLL Clock Rate

D11	D10	
0	0	32x Clock Mode
0	1	16x Clock Mode
1	0	8x Clock Mode
1	1	Reserved

D12: Accept Code Violations

D13: CTR1 Rate Match DPLL/CTR0

D15 D14: CTR0 Clock Rate

D15	D14	
0	0	32x Clock Mode
0	1	16x Clock Mode
1	0	8x Clock Mode
1	1	4x Clock Mode

Figure 28. Hardware Configuration Register

CONTROL REGISTERS (Continued)

Address: 01010

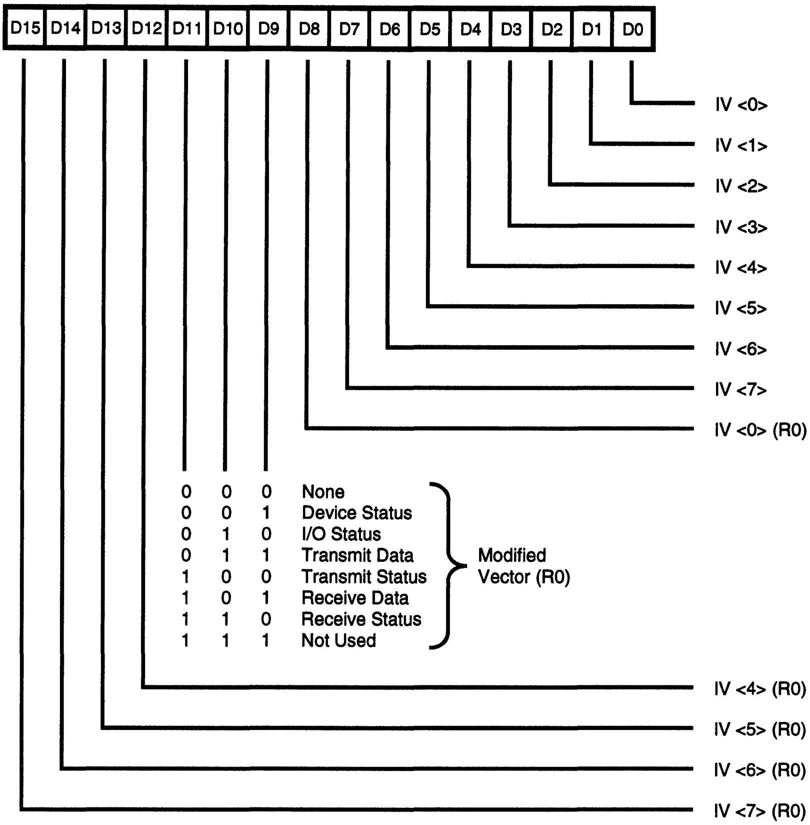
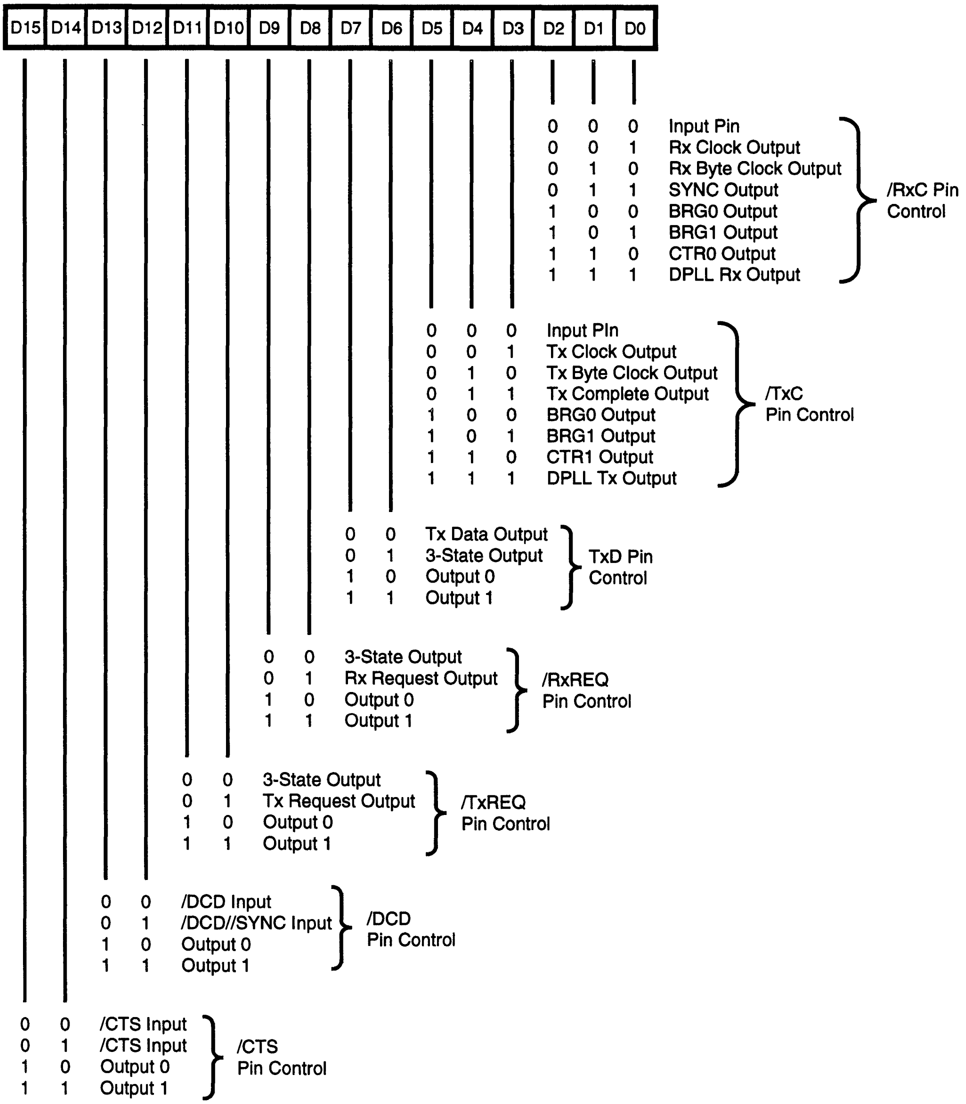


Figure 29. Interrupt Vector Register

Address: 01011



D2	D1	D0		
0	0	0	Input Pin	/RxC Pin Control
0	0	1	Rx Clock Output	
0	1	0	Rx Byte Clock Output	
0	1	1	SYNC Output	
1	0	0	BRG0 Output	
1	0	1	BRG1 Output	
1	1	0	CTR0 Output	
1	1	1	DPLL Rx Output	

D5	D4	D3		
0	0	0	Input PIn	/TxC Pin Control
0	0	1	Tx Clock Output	
0	1	0	Tx Byte Clock Output	
0	1	1	Tx Complete Output	
1	0	0	BRG0 Output	
1	0	1	BRG1 Output	
1	1	0	CTR1 Output	
1	1	1	DPLL Tx Output	

D7	D6		
0	0	Tx Data Output	TxD Pin Control
0	1	3-State Output	
1	0	Output 0	
1	1	Output 1	

D9	D8		
0	0	3-State Output	/RxREQ Pin Control
0	1	Rx Request Output	
1	0	Output 0	
1	1	Output 1	

D11	D10		
0	0	3-State Output	/TxREQ Pin Control
0	1	Tx Request Output	
1	0	Output 0	
1	1	Output 1	

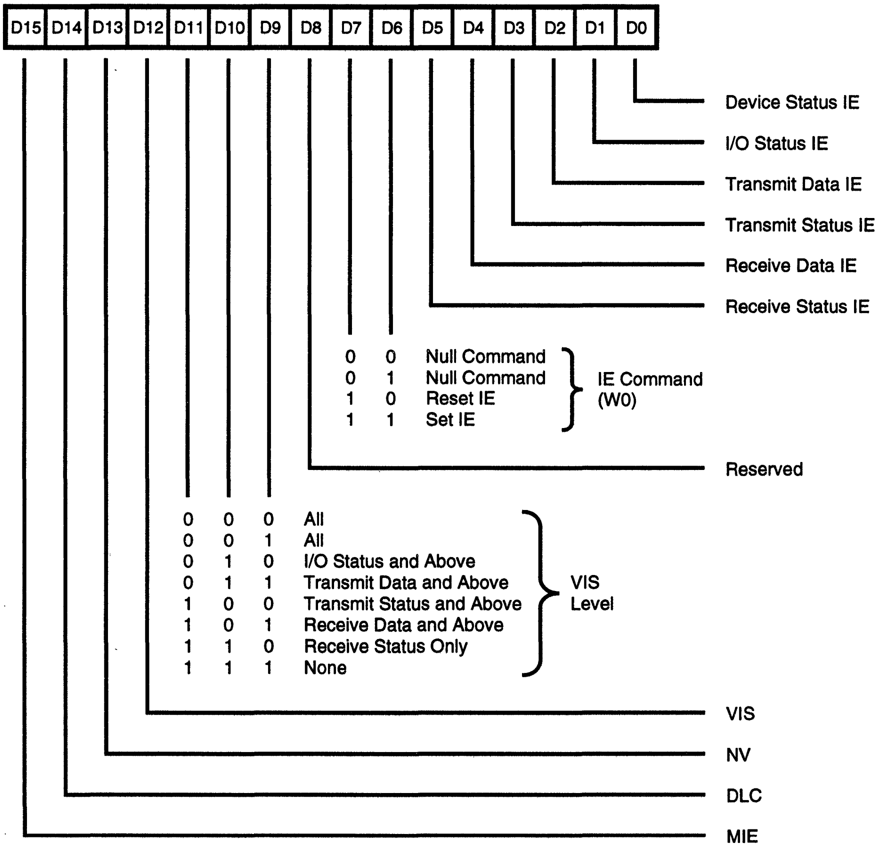
D13	D12		
0	0	/DCD Input	/DCD Pin Control
0	1	/DCD//SYNC Input	
1	0	Output 0	
1	1	Output 1	

D15	D14		
0	0	/CTS Input	/CTS Pin Control
0	1	/CTS Input	
1	0	Output 0	
1	1	Output 1	

Figure 30. I/O Control Register

CONTROL REGISTERS (Continued)

Address: 01100



D7	D6	IE Command (W0)
0	0	Null Command
0	1	Null Command
1	0	Reset IE
1	1	Set IE

D11	D10	D9	VIS Level
0	0	0	All
0	0	1	All
0	1	0	I/O Status and Above
0	1	1	Transmit Data and Above
1	0	0	Transmit Status and Above
1	0	1	Receive Data and Above
1	1	0	Receive Status Only
1	1	1	None

Figure 31. Interrupt Control Register

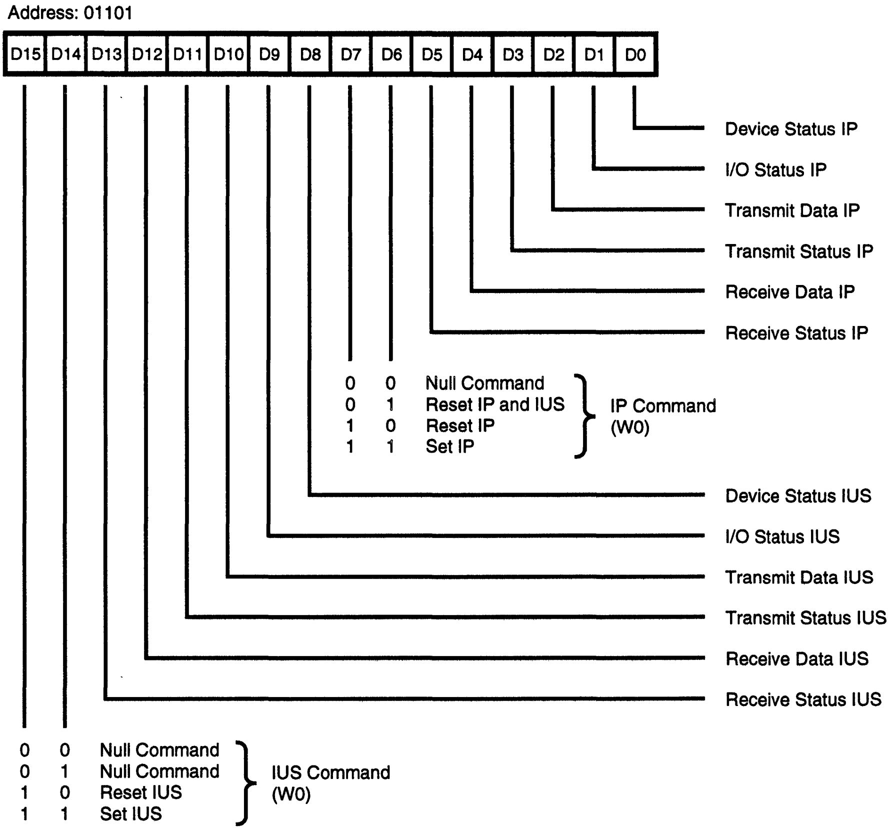


Figure 32. Daisy-Chain Control Register

CONTROL REGISTERS (Continued)

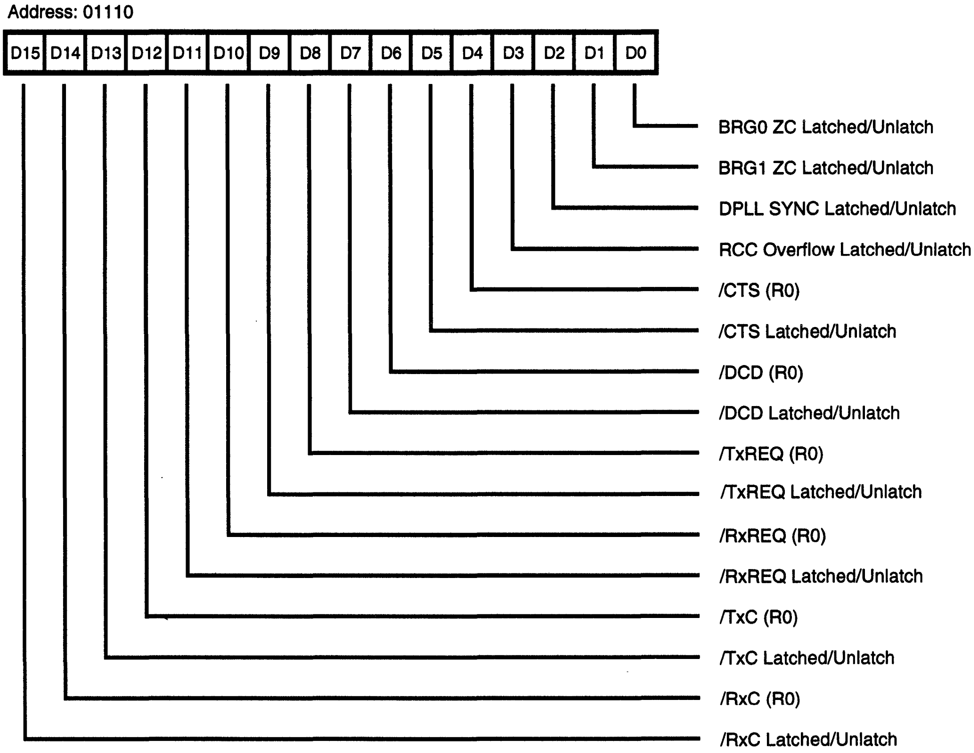


Figure 33. Miscellaneous Interrupt Status Register

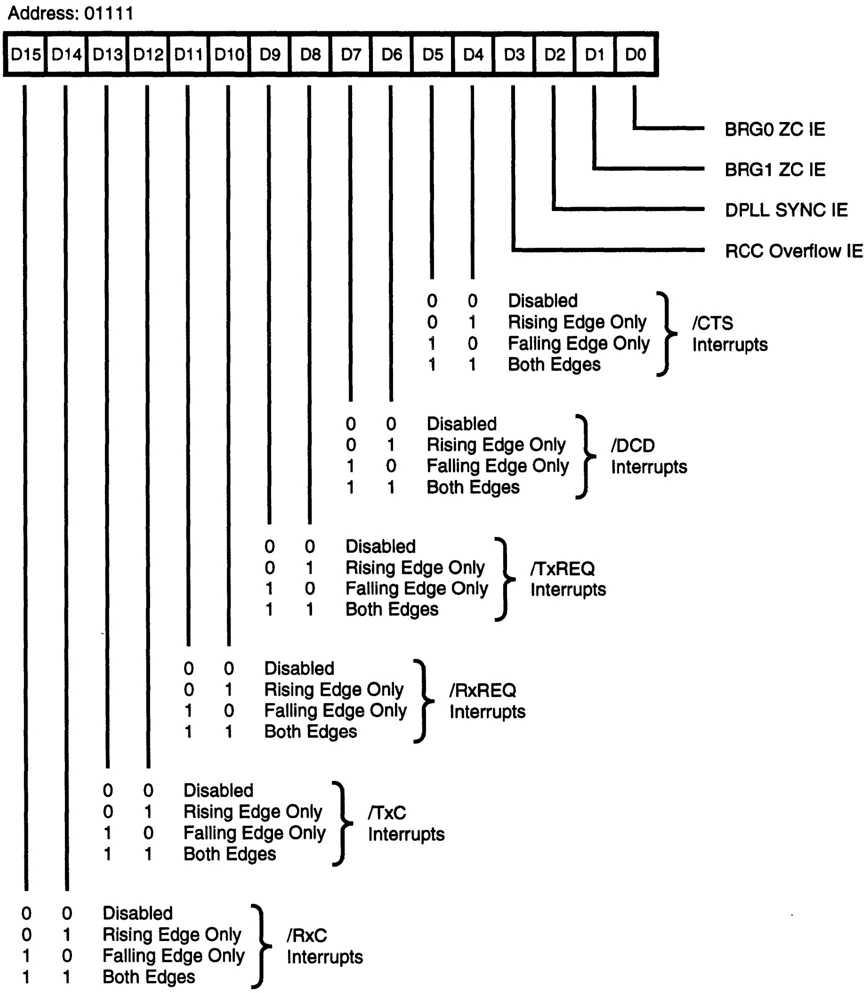


Figure 34. Status Interrupt Control Register

CONTROL REGISTERS (Continued)

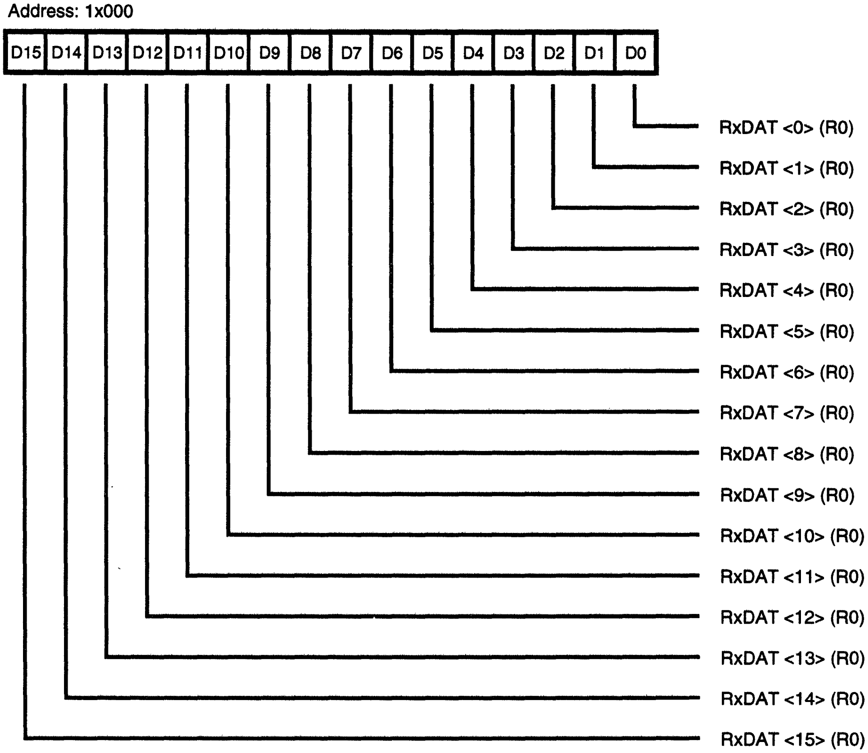
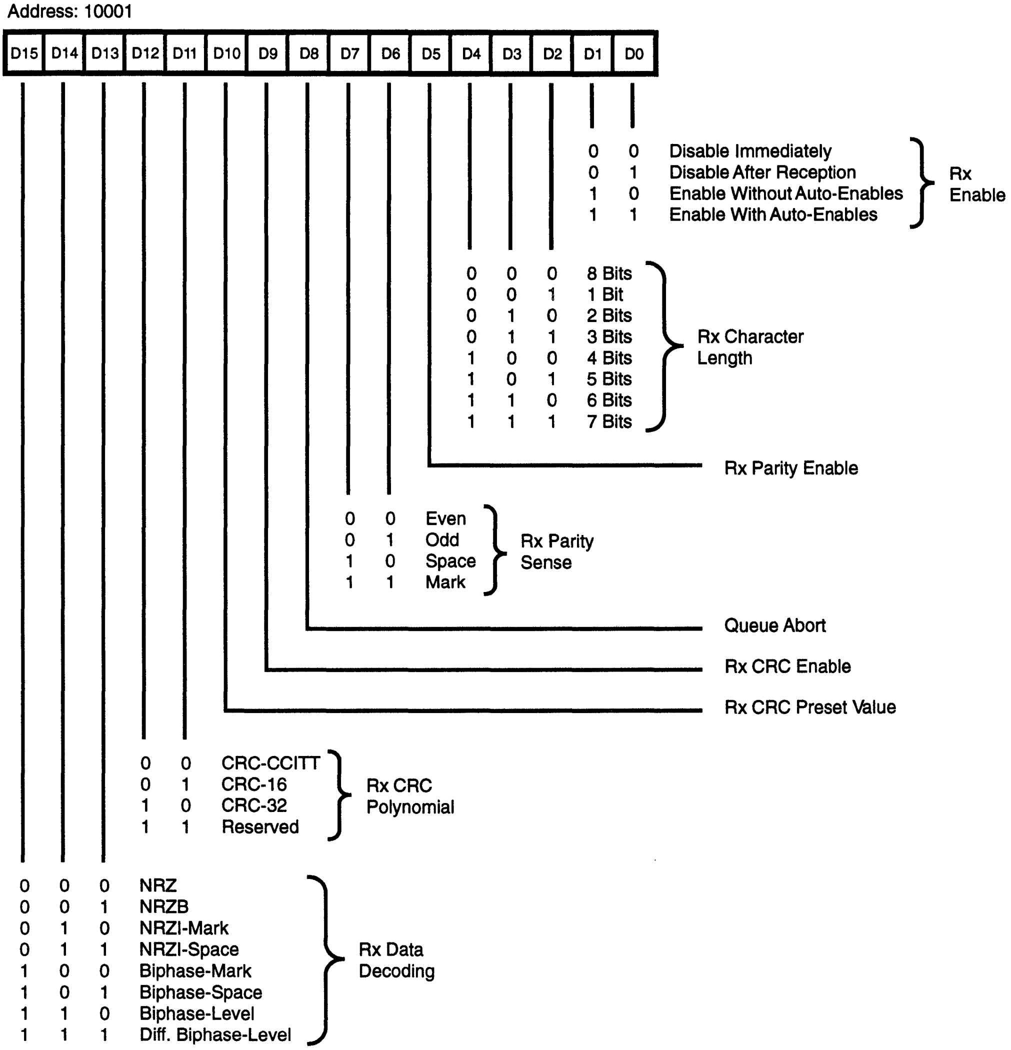


Figure 35. Receive Data Register



D1	D0		
0	0	Disable Immediately	Rx Enable
0	1	Disable After Reception	
1	0	Enable Without Auto-Enables	
1	1	Enable With Auto-Enables	

D4	D3	D2		
0	0	0	8 Bits	Rx Character Length
0	0	1	1 Bit	
0	1	0	2 Bits	
0	1	1	3 Bits	
1	0	0	4 Bits	
1	0	1	5 Bits	
1	1	0	6 Bits	
1	1	1	7 Bits	

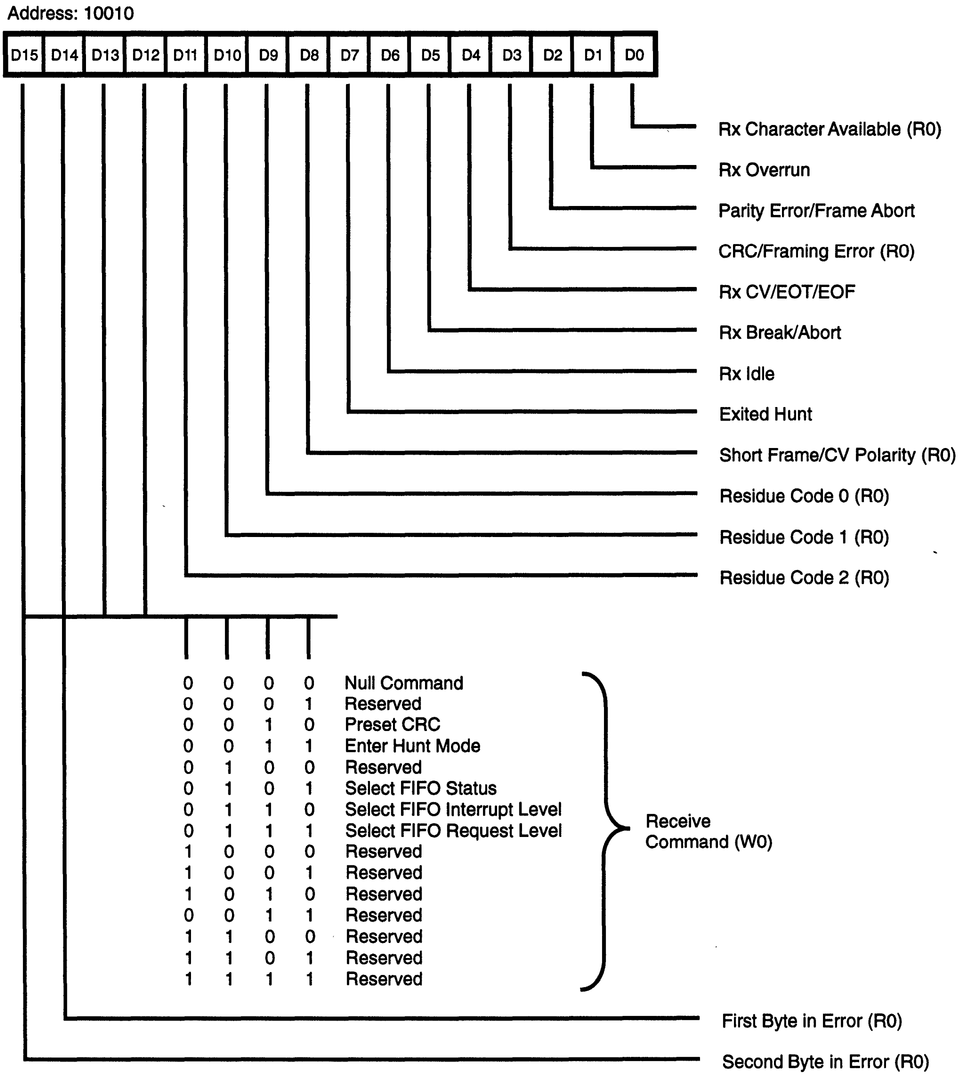
D7	D6		
0	0	Even	Rx Parity Sense
0	1	Odd	
1	0	Space	
1	1	Mark	

D12	D11		
0	0	CRC-CCITT	Rx CRC Polynomial
0	1	CRC-16	
1	0	CRC-32	
1	1	Reserved	

D15	D14	D13		
0	0	0	NRZ	Rx Data Decoding
0	0	1	NRZB	
0	1	0	NRZI-Mark	
0	1	1	NRZI-Space	
1	0	0	Biphase-Mark	
1	0	1	Biphase-Space	
1	1	0	Biphase-Level	
1	1	1	Diff. Biphase-Level	

Figure 36. Receive Mode Register

CONTROL REGISTERS (Continued)



D15	D14	D13	D12	
0	0	0	0	Null Command
0	0	0	1	Reserved
0	0	1	0	Preset CRC
0	0	1	1	Enter Hunt Mode
0	1	0	0	Reserved
0	1	0	1	Select FIFO Status
0	1	1	0	Select FIFO Interrupt Level
0	1	1	1	Select FIFO Request Level
1	0	0	0	Reserved
1	0	0	1	Reserved
1	0	1	0	Reserved
0	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	1	Reserved

Figure 37. Receive Command Status Register

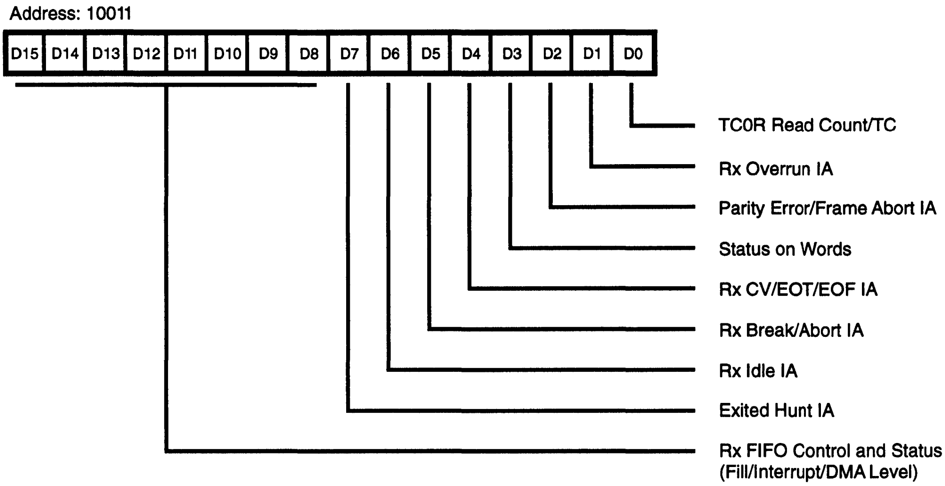


Figure 38. Receive Interrupt Control Register

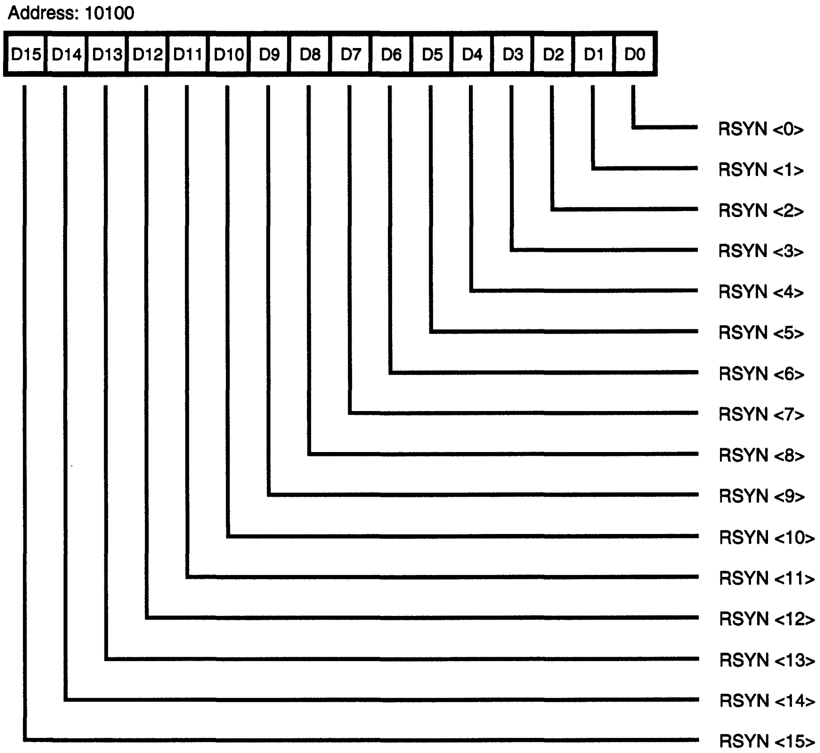


Figure 39. Receive Sync Register

CONTROL REGISTERS (Continued)

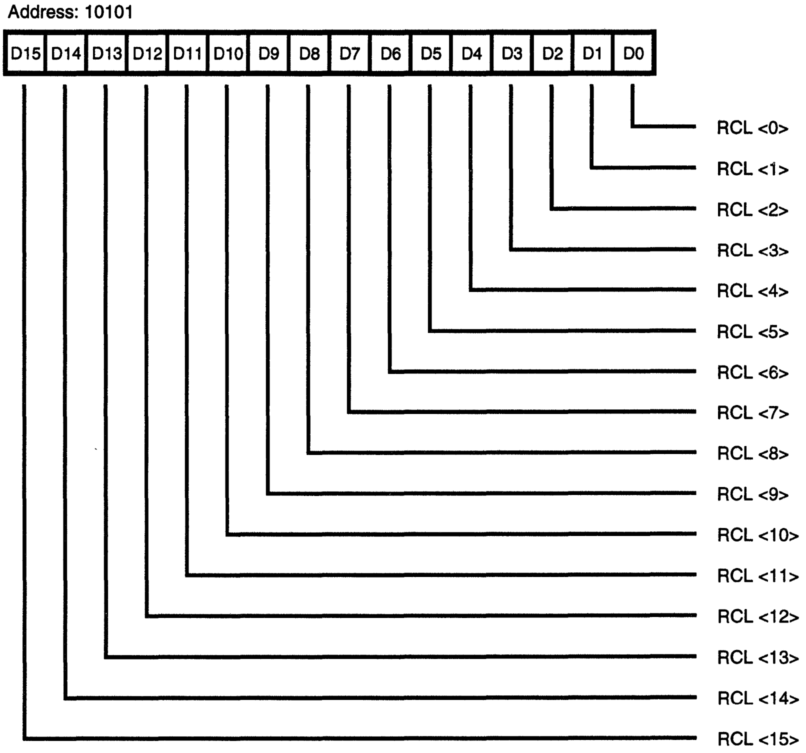


Figure 40. Receive Count Limit Register

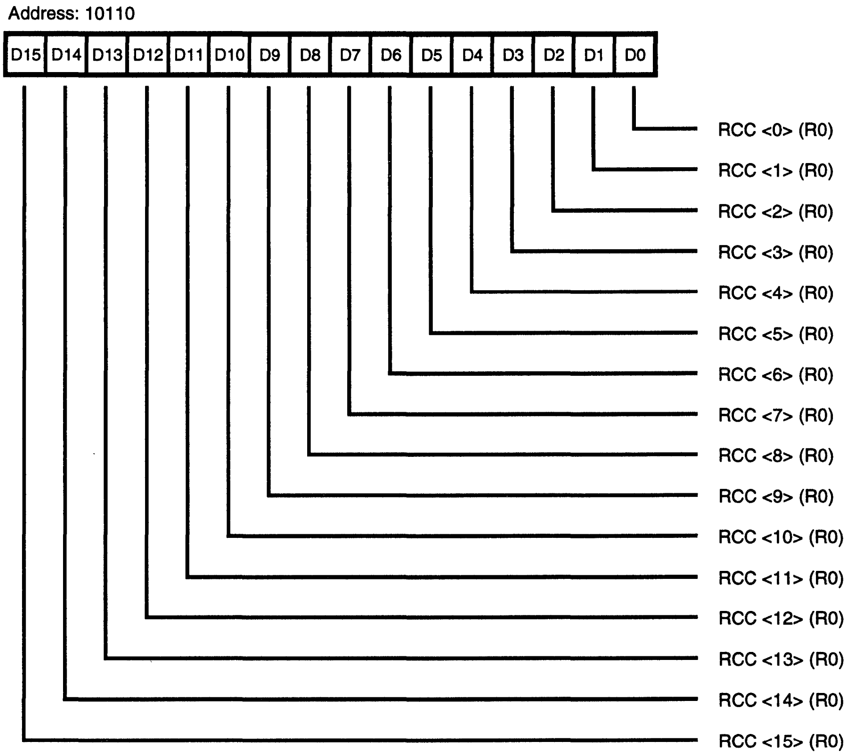


Figure 41. Receive Character Count Register

CONTROL REGISTERS (Continued)

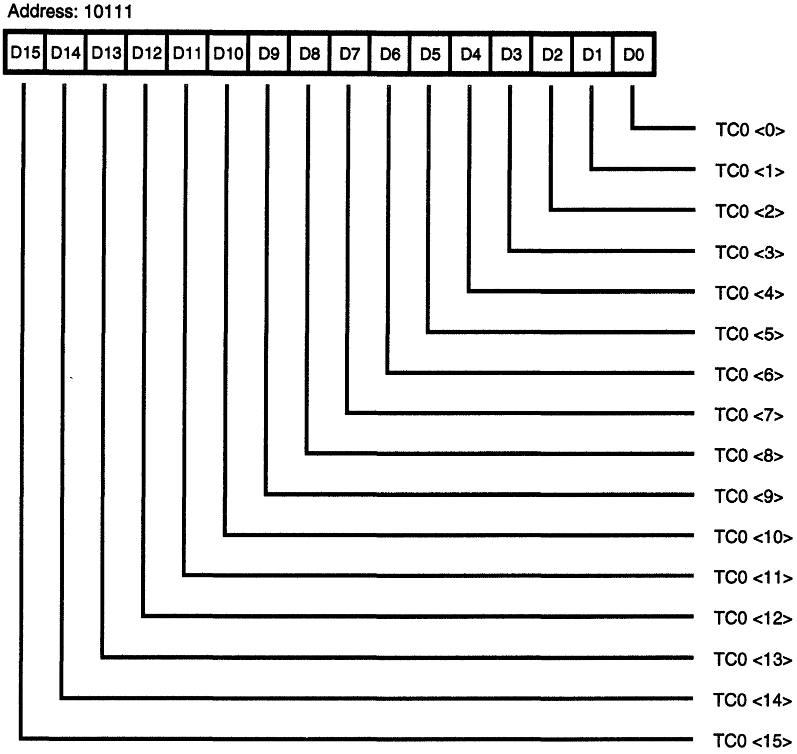


Figure 42. Time Constant 0 Register

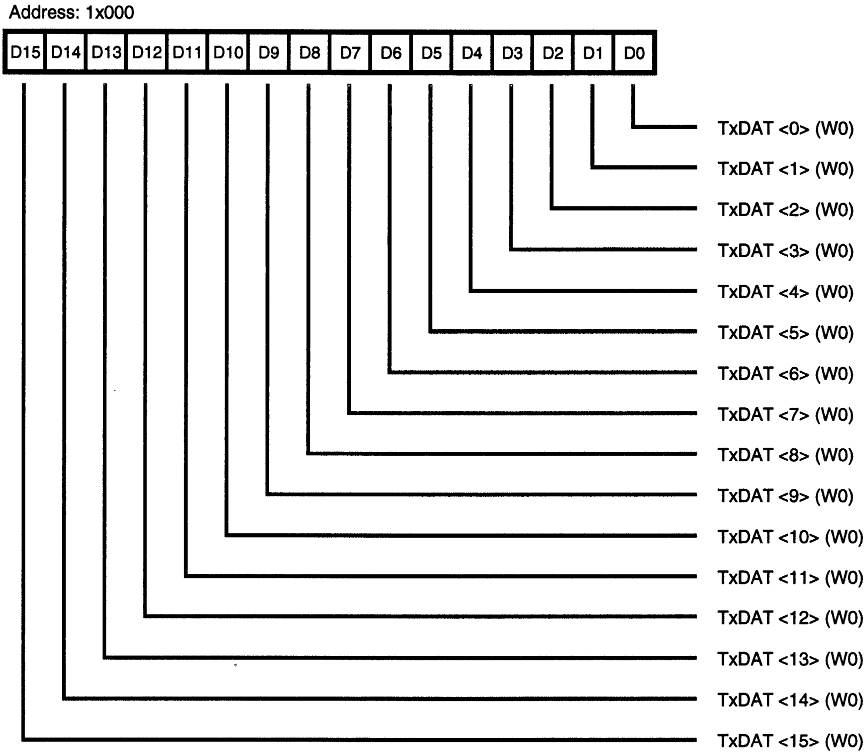
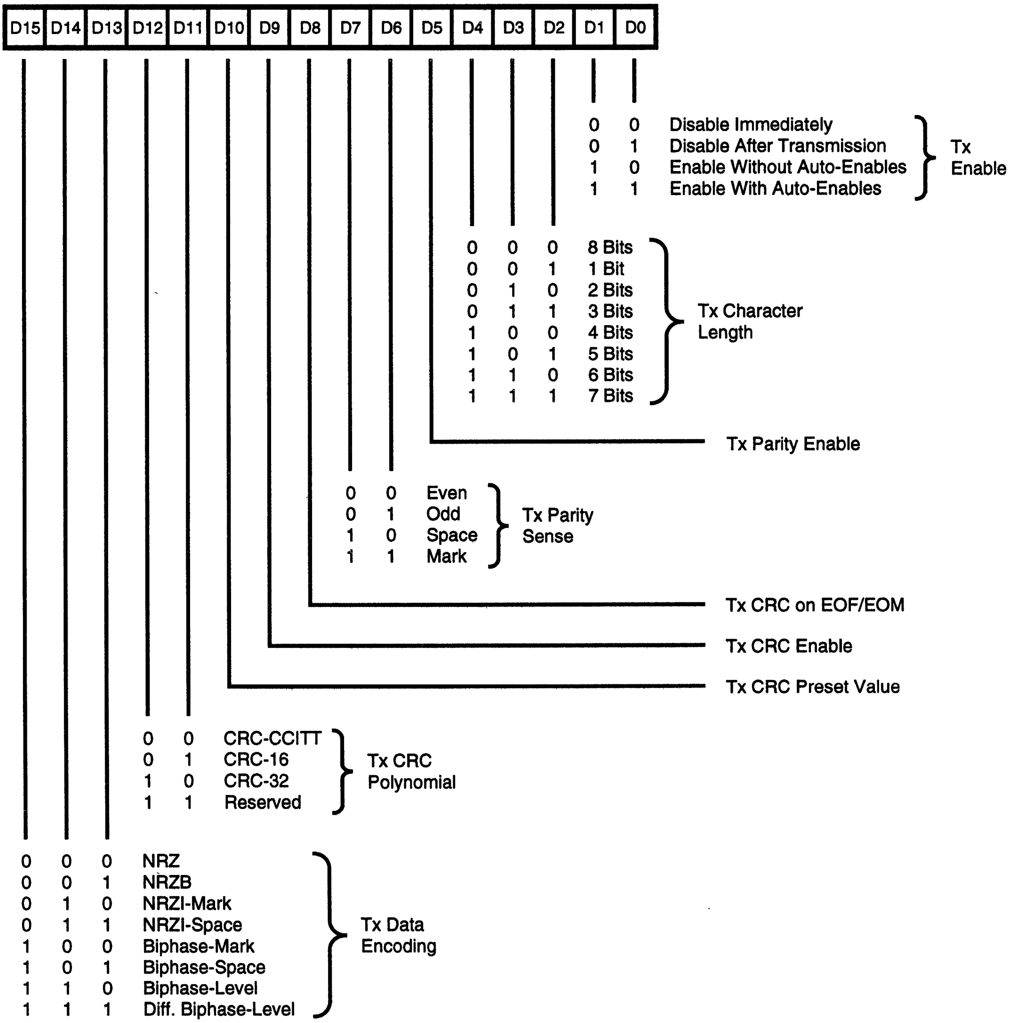


Figure 43. Transmit Data Register

CONTROL REGISTERS (Continued)

Address: 11001



D1	D0		
0	0	Disable Immediately	Tx Enable
0	1	Disable After Transmission	
1	0	Enable Without Auto-Enables	
1	1	Enable With Auto-Enables	

D4	D3	D2		
0	0	0	8 Bits	Tx Character Length
0	0	1	1 Bit	
0	1	0	2 Bits	
0	1	1	3 Bits	
1	0	0	4 Bits	
1	0	1	5 Bits	
1	1	0	6 Bits	
1	1	1	7 Bits	

D5: Tx Parity Enable

D7	D6		
0	0	Even	Tx Parity Sense
0	1	Odd	
1	0	Space	
1	1	Mark	

D8: Tx CRC on EOF/EOM

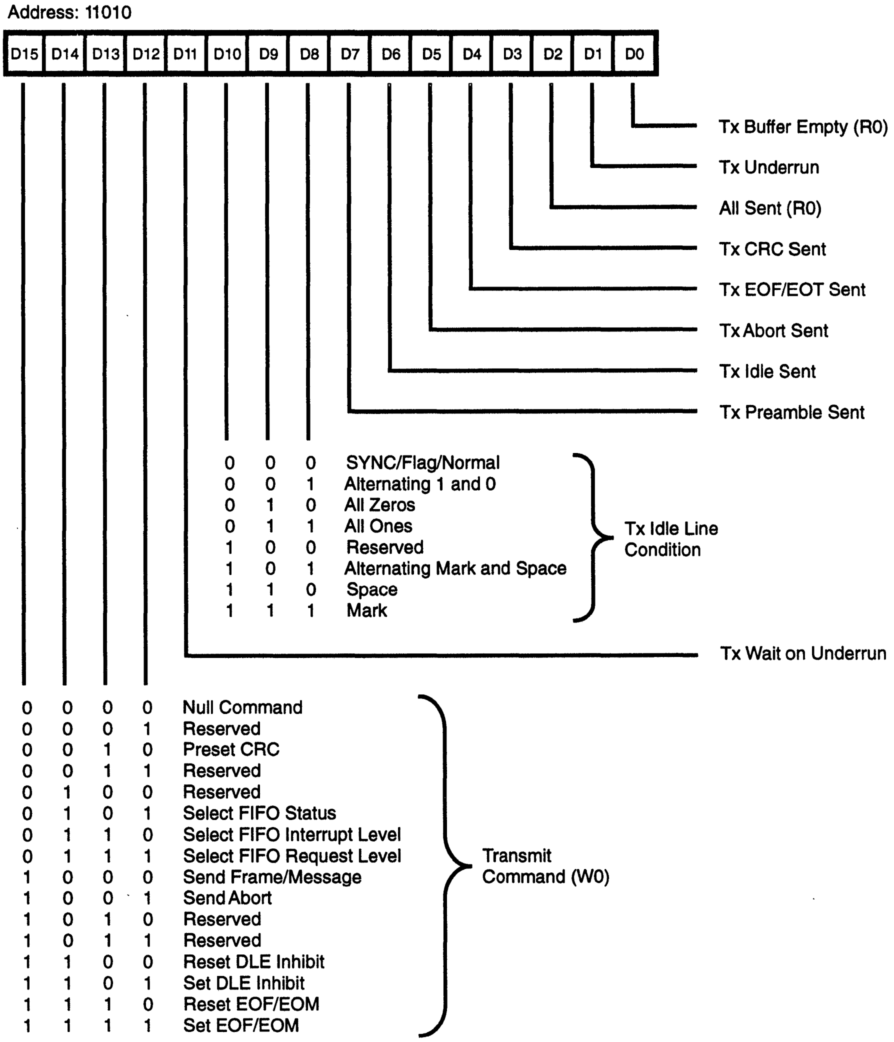
D9: Tx CRC Enable

D10: Tx CRC Preset Value

D12	D11		
0	0	CRC-CCITT	Tx CRC Polynomial
0	1	CRC-16	
1	0	CRC-32	
1	1	Reserved	

D15	D14	D13		
0	0	0	NRZ	Tx Data Encoding
0	0	1	NRZB	
0	1	0	NRZI-Mark	
0	1	1	NRZI-Space	
1	0	0	Biphase-Mark	
1	0	1	Biphase-Space	
1	1	0	Biphase-Level	
1	1	1	Diff. Biphase-Level	

Figure 44. Transmit Mode Register



D10	D9	D8	Tx Idle Line Condition
0	0	0	SYNC/Flag/Normal
0	0	1	Alternating 1 and 0
0	1	0	All Zeros
0	1	1	All Ones
1	0	0	Reserved
1	0	1	Alternating Mark and Space
1	1	0	Space
1	1	1	Mark

D15	D14	D13	D12	Transmit Command (W0)
0	0	0	0	Null Command
0	0	0	1	Reserved
0	0	1	0	Preset CRC
0	0	1	1	Reserved
0	1	0	0	Reserved
0	1	0	1	Select FIFO Status
0	1	1	0	Select FIFO Interrupt Level
0	1	1	1	Select FIFO Request Level
1	0	0	0	Send Frame/Message
1	0	0	1	Send Abort
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reset DLE Inhibit
1	1	0	1	Set DLE Inhibit
1	1	1	0	Reset EOF/EOM
1	1	1	1	Set EOF/EOM

Figure 45. Transmit Command/Status Register

CONTROL REGISTERS (Continued)

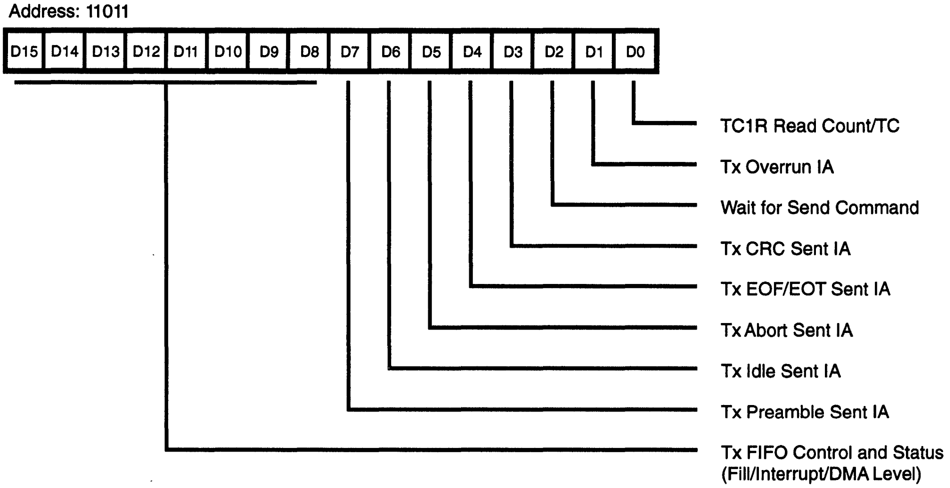


Figure 46. Transmit Interrupt Control Register

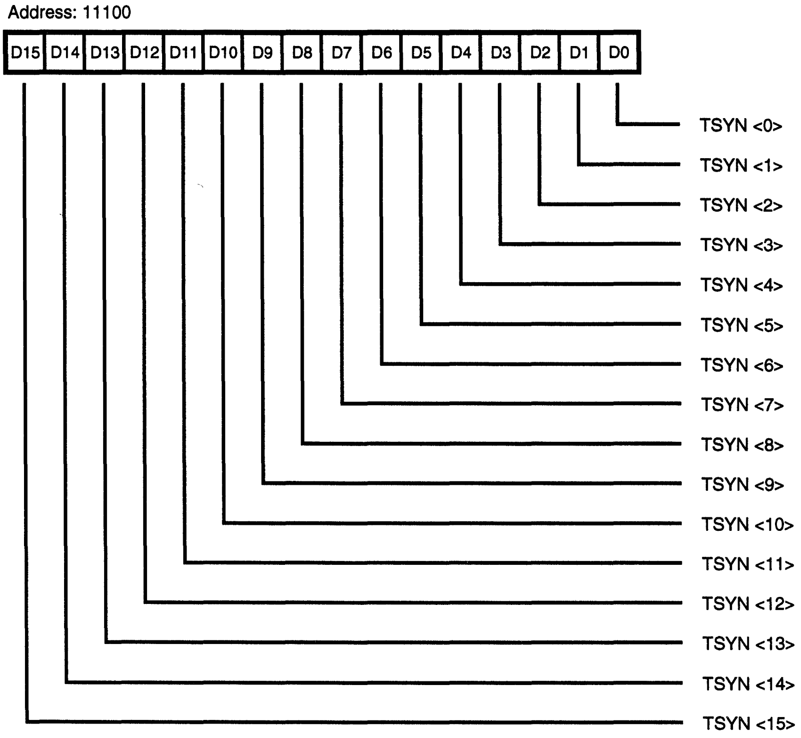


Figure 47. Transmit Sync Register

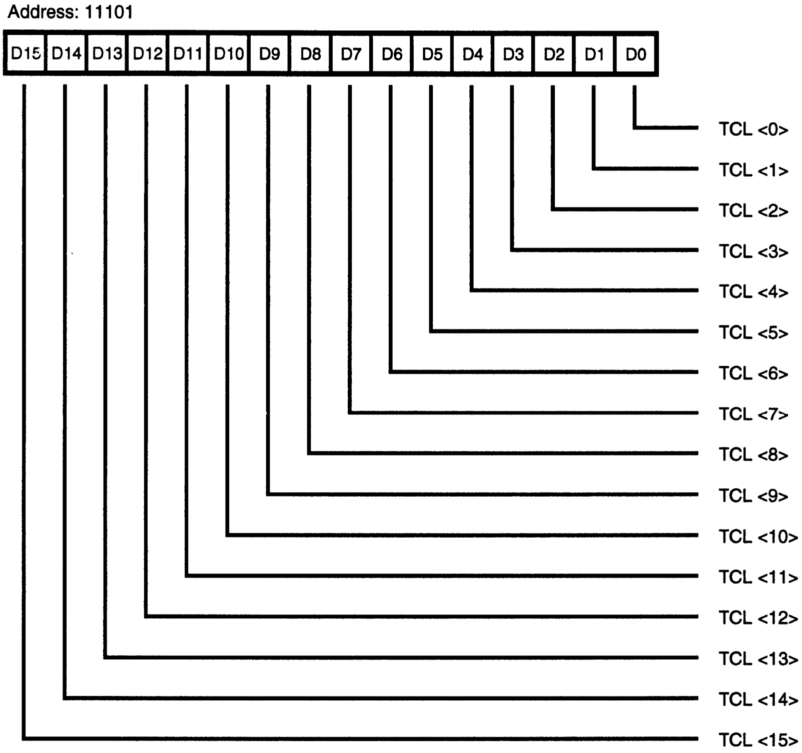


Figure 48. Transmit Count Limit Register

CONTROL REGISTERS (Continued)

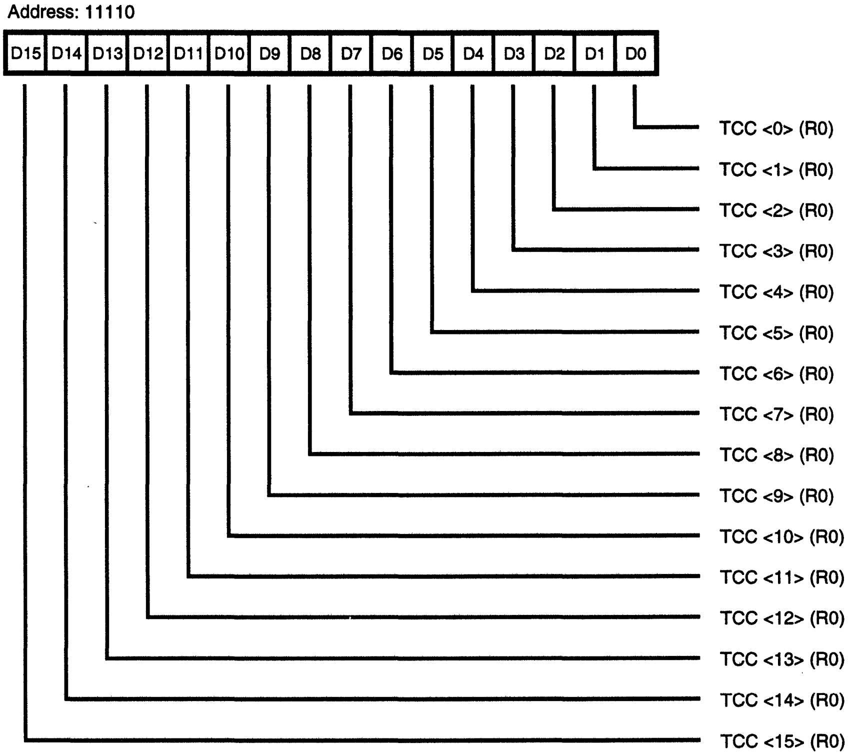


Figure 49. Transmit Character Count Register

Address: 11111

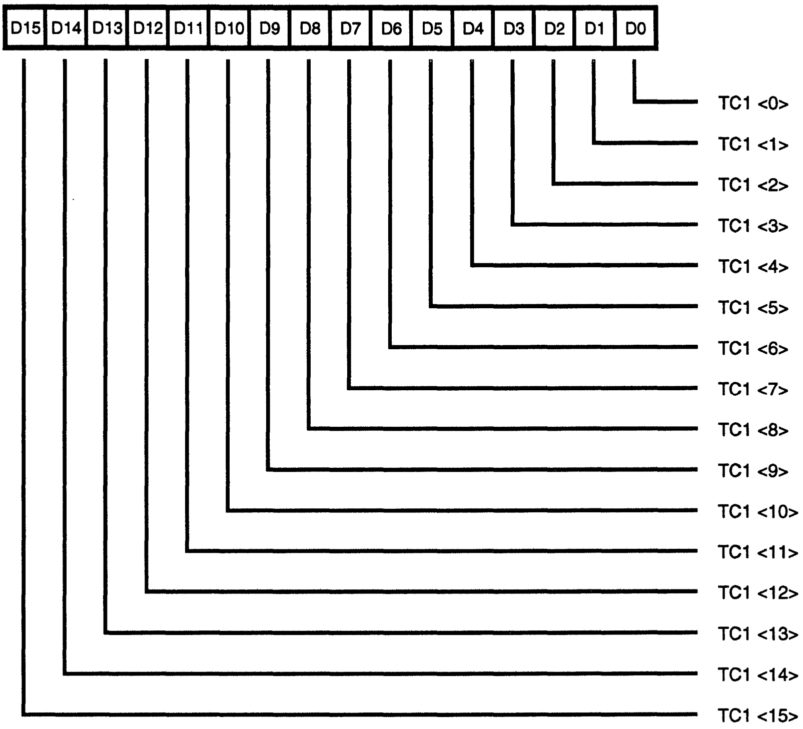
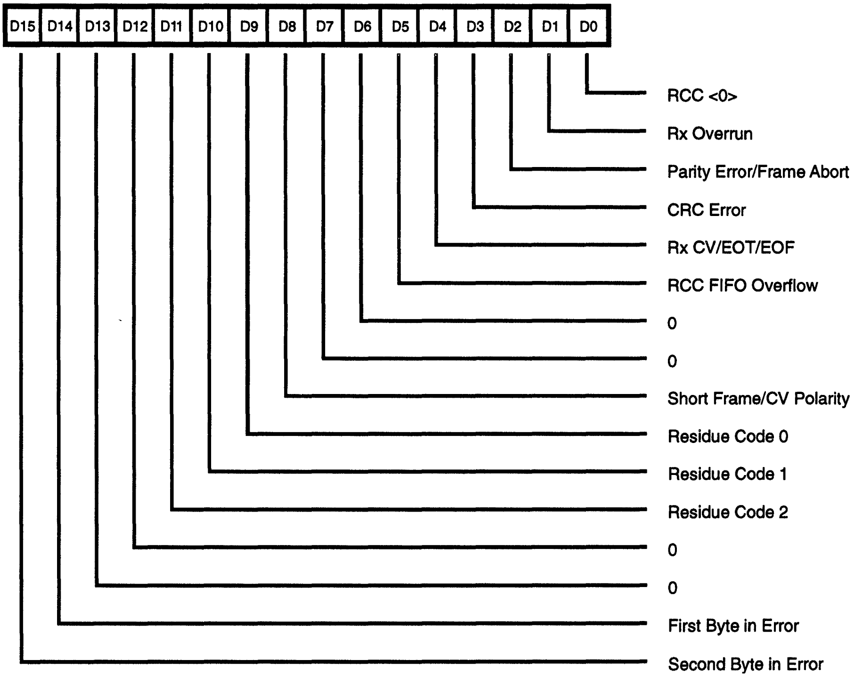


Figure 50. Time Constant 1 Register

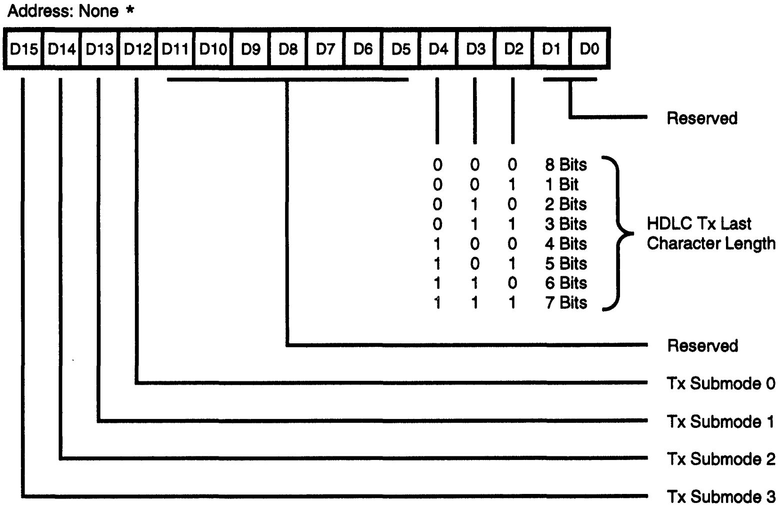
CONTROL REGISTERS (Continued)

Address: None *



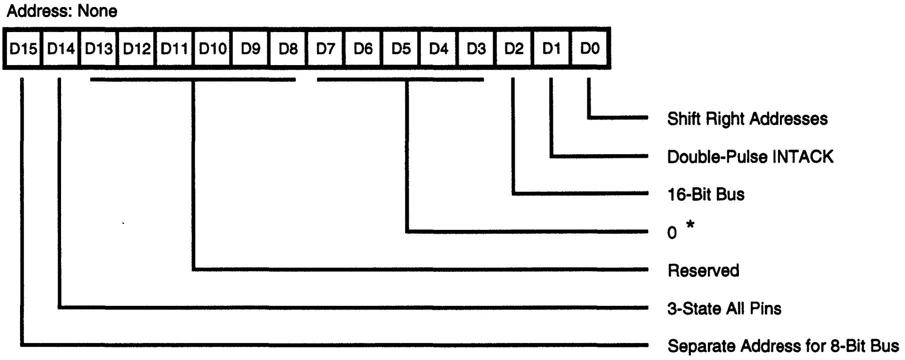
* Refer to Figure 22 (Channel Control Register)
Bits 6-7 for Access Method

Figure 51. Receive Status Block Register



* Refer to Figure 22 (Channel Control Register) Bits15-14 for Access Method

Figure 52. Transmit Status Block Register



* Must be programmed as zero.

Figure 53. Bus Configuration Register

USC TIMING

The USC interface timing is similar to that found on a static RAM, except that it is much more flexible. Up to eight separate timing strobe signals may be present on the interface: /DS, /RD, /WR, /PITACK, /RxACKA, /RxACKB, /TxACKA and /TxACKB. Only one of these timing strobes may be active at any time. Should the external logic activate more than one of these strobes at the same time the USC will enter a pre-reset state that is only exited by a hardware reset. Do not allow overlap of timing strobes. The timing diagrams, beginning on the next page, illustrate the different bus transactions possible, with the necessary setup, hold and delay times.

ABSOLUTE MAXIMUM RATINGS

Symbol	Description	Min	Max	Units
V_{CC}	Supply Voltage (*)	–0.3	+7.0	V
T_{STG}	Storage Temp	–65°	+150°	C
T_A	Oper Ambient Temp		†	C
	Power Dissipation		2.2	W

Notes:
* Voltage on all pins with respect to GND.
† See Ordering Information.

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

STANDARD TEST CONDITIONS

The DC Characteristics and Capacitance section below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin (Figure 54). Standard conditions are as follows:

- +4.5 V < V_{CC} < +5.5 V
- GND = 0 V
- T_A as specified in Ordering Information

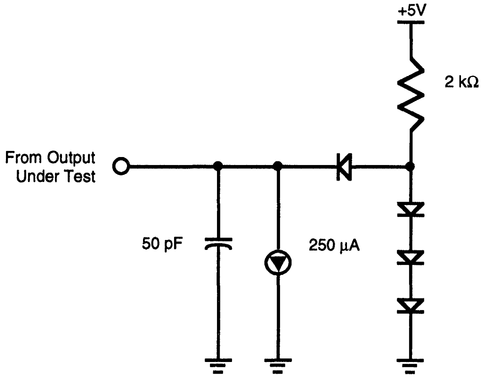


Figure 54. Standard Test Load

CAPACITANCE

Symbol	Parameter	Min	Max	Unit	Condition
C_{IN}	Input Capacitance		10	pf	Unmeasured pins
C_{OUT}	Output Capacitance		15	pf	returned to Ground.
$C_{I/O}$	Bidirectional Capacitance		20	pf	

Note:
f = 1 MHz, over specified temperature range.

MISCELLANEOUS

Transistor Count - 174,000

TEMPERATURE RATINGS

Standard = 0°C to ±70°C
Extended = –40°C to +85°C

DC CHARACTERISTICS
Z16C30

Symbol	Parameter	Min	Typ	Max	Unit	Condition
V_{IH}	Input High Voltage	2.2		V_{CC}+0.3	V	
V_{IL}	Input Low Voltage	–0.3		0.8	V	
$V_{OH}1$	Output High Voltage	2.4			V	I_{OH} = –1.6 mA
$V_{OH}2$	Output High Voltage	V_{CC}-0.8			V	I_{OH} = –250 µA
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = +2.0 mA
I_{IL}	Input Leakage			±10.00	µA	0.4 < V_{IN} < +2.4V
I_{OL}	Output Leakage			±10.00	µA	0.4 < V_{OUT} < +2.4V
I_{CCI}	V_{CC} Supply Current		7	50	mA	V_{CC} = 5V V_{IH} = 4.8V V_{IL} = 0.2V

Note:
V_{CC}= 5V ± 10% unless otherwise specified, over specified temperature range.

AC CHARACTERISTICS
Z16C30

No	Symbol	Parameter	Min	Max	Units	Note
1	Tcyc	Bus Cycle Time	110		ns	
2	TwASl	/AS Low Width	30		ns	
3	TwASh	/AS High Width	60		ns	
4	TwDSl	/DS Low Width	60		ns	
5	TwDSh	/DS High Width	50		ns	
6	TdAS(DS)	/AS Rise to /DS Fall Delay Time	5		ns	
7	TdDS(AS)	/DS Rise to /AS Fall Delay Time	5		ns	
8	TdDS(DRa)	/DS Fall to Data Active Delay	0		ns	
9	TdDS(DRv)	/DS Fall to Data Valid Delay		60	ns	
10	TdDS(DRn)	/DS Rise to Data Not Valid Delay	0		ns	
11	TdDS(DRz)	/DS Rise to Data Float Delay		20	ns	
12	TsCS(AS)	/CS to /AS Rise Setup Time	15		ns	
13	ThCS(AS)	/CS to /AS Rise Hold Time	5		ns	
14	TsADD(AS)	Direct Address to /AS Rise Setup Time	15		ns	[1]
15	ThADD(AS)	Direct Address to /AS Rise Hold Time	5		ns	[1]
16	TsSIA(AS)	/SITACK to /AS Rise Setup Time	15		ns	
17	ThSIA(AS)	/SITACK to /AS Rise Hold Time	5		ns	
18	TsAD(AS)	Address to /AS Rise Setup Time	15		ns	
19	ThAD(AS)	Address to /AS Rise Hold Time	5		ns	
20	TsRW(DS)	R//W to /DS Fall Setup Time	0		ns	

AC CHARACTERISTICS
Z16C30

No	Symbol	Parameter	Min	Max	Units	Notes
21	ThRW(DS)	R//W to /DS Fall Hold Time	25		ns	
22	TsDSf(RRQ)	/DS Fall to /RxREQ Inactive Delay		60	ns	[4]
23	TdDSr(RRQ)	/DS Rise to /RxREQ Active Delay	0		ns	
24	TsDW(DS)	Write Data to /DS Rise Setup Time	30		ns	
25	ThDW(DS)	Write Data to /DS Rise Hold Time	0		ns	
26	TdDSf(TRQ)	/DS Fall to /TxREQ Inactive Delay		65	ns	[5,6]
27	TdDSr(TRQ)	/DS Rise to /TxREQ Active Delay	0		ns	
28	TwRDl	/RD Low Width	60		ns	
29	TwRDh	/RD High Width	50		ns	
30	TdAS(RD)	/AS Rise to /RD Fall Delay Time	5		ns	
31	TdRD(AS)	/RD Rise to /AS Fall Delay Time	5		ns	
32	TdRD(DRa)	/RD Fall to Data Active Delay	0		ns	
33	TdRD(DRv)	/RD Fall to Data Valid Delay		60	ns	
34	TdRD(DRn)	/RD Rise to Data Not Valid Delay	0		ns	
35	TdRD(DRz)	/RD Rise to Data Float Delay		20	ns	
36	TdRDf(RRQ)	/RD Fall to /RxREQ Inactive Delay		60	ns	[4]
37	TdRDr(RRQ)	/RD Rise to /RxREQ Active Delay	0		ns	
38	TwWRl	/WR Low Width	60		ns	
39	TwWRh	/WR High Width	50		ns	
40	TdAS(WR)	/AS Rise to /WR Fall Delay Time	5		ns	
41	TdWR(AS)	/WR Rise to /AS Fall Delay Time	5		ns	
42	TsDW(WR)	Write Data to /WR Rise Setup Time	30		ns	
43	ThDW(WR)	Write Data to /WR Rise Hold Time	0		ns	
44	TdWRf(TRQ)	/WR Fall to /TxREQ Inactive Delay		65	ns	[5]
45	TdWRr(TRQ)	/WR Rise to /TxREQ Active Delay	0		ns	
46	TsCS(DS)	/CS to /DS Fall Setup Time	0		ns	[2]
47	ThCS(DS)	/CS to /DS Fall Hold Time	25		ns	[2]
48	TsADD(DS)	Direct Address to /DS Fall Setup Time	5		ns	[1,2]
49	ThADD(DS)	Direct Address to /DS Fall Hold Time	25		ns	[1,2]
50	TsSIA(DS)	/SITACK to /DS Fall Setup Time	5		ns	[2]
51	ThSIA(DS)	/SITACK to /DS Fall Hold Time	25		ns	[2]
52	TsCS(RD)	/CS to /RD Fall Setup Time	0		ns	[2]
53	ThCS(RD)	/CS to /RD Fall Hold Time	25		ns	[2]
54	TsADD(RD)	Direct Address to /RD Fall Setup Time	5		ns	[1,2]
55	ThADD(RD)	Direct Address to /RD Fall Hold Time	25		ns	[1,2]
56	TsSIA(RD)	/SITACK to /RD Fall Setup Time	5		ns	[2]
57	ThSIA(RD)	/SITACK to /RD Fall Hold Time	25		ns	[2]
58	TsCS(WR)	/CS to /WR Fall Setup Time	0		ns	[2]
59	ThCS(WR)	/CS to /WR Fall Hold Time	25		ns	[2]
60	TsADD(WR)	Direct Address to /WR Fall Setup Time	5		ns	[1,2]

AC CHARACTERISTICS
Z16C30

No	Symbol	Parameter	Min	Max	Units	Notes
61	ThADD(WR)	Direct Address to /WR Fall Hold Time	25		ns	[1,2]
62	TsSIA(WR)	/SITACK to /WR Fall Setup Time	5		ns	[2]
63	ThSIA(WR)	/SITACK to /WR Fall Hold Time	25		ns	[2]
64	TwRAKl	/RxACK Low Width	60		ns	
65	TwRAKh	/RxACK High Width	50		ns	
66	TdRAK(DRa)	/RxACK Fall to Data Active Delay	0		ns	
67	TdRAK(DRv)	/RxACK Fall to Data Valid Delay		60	ns	
68	TdRAK(DRn)	/RxACK Rise to Data Not Valid Delay	0		ns	
69	TdRAK(DRz)	/RxACK Rise to Data Float Delay		20	ns	
70	TdRAKf(RRQ)	/RxACK Fall to /RxREQ Inactive Delay		60	ns	[4]
71	TdRAKr(RRQ)	/RxACK Rise to /RxREQ Active Delay	0		ns	
72	TwTAKl	/TxACK Low Width	60		ns	
73	TwTAKh	/TxACK High Width	50		ns	
74	TsDW(TAK)	Write Data to /TxACK Rise Setup Time	30		ns	
75	ThDW(TAK)	Write Data to /TxACK Rise Hold Time	0		ns	
76	TdTAKf(TRQ)	/TxACK Fall to /TxREQ Inactive Delay		65	ns	[5]
77	TdTAKr(TRQ)	/TxACK Rise to /TxREQ Active Delay	0		ns	
78	TdDSf(RDY)	/DS Fall (INTACK) to /RDY Fall Delay		200	ns	
79	TdRDY(DRv)	/RDY Fall to Data Valid Delay		40	ns	
80	TdDSr(RDY)	/DS Rise to /RDY Rise Delay		40	ns	
81	TsIEI(DSI)	IEI to /DS Fall (INTACK) Setup Time	10		ns	
82	ThIEI(DSI)	IEI to /DS Rise (INTACK) Hold Time	0		ns	
83	TdIEI(IEO)	IEI to IEO Delay		30	ns	
84	TdAS(IEO)	/AS Rise (Intack) to IEO Delay		60	ns	
85	TdDSI(INT)	/DS Fall (INTACK) to /INT Inactive Delay		200	ns	[7]
86	TdDSI(Wf)	/DS Fall (INTACK) to /WAIT Fall Delay		40	ns	
87	TdDSI(Wr)	/DS Fall (INTACK) to /WAIT Rise Delay		200	ns	
88	TdW(DRv)	/WAIT Rise to Data Valid Delay		40	ns	
89	TdRDf(RDY)	/RD Fall (INTACK) to /RDY Fall Delay		200	ns	
90	TdRDr(RDY)	/RD Rise to /RDY Rise Delay		40	ns	
91	TsIEI(RDI)	IEI to /RD Fall (INTACK) Setup Time	10		ns	
92	ThIEI(RDI)	IEI to /RD Rise (INTACK) Hold Time	0		ns	
93	TdRDI(INT)	/RD Fall (INTACK) to /INT Inactive Delay		200	ns	
94	TdRDI(Wf)	/RD Fall (INTACK) to /WAIT Fall Delay		40	ns	
95	TdRDI(Wr)	/RD Fall (INTACK) to /WAIT Rise Delay		200	ns	
96	TwPIAl	/PITACK Low Width	60		ns	
97	TwPIAh	/PITACK High Width	50		ns	
98	TdAS(PIA)	/AS Rise to /PITACK Fall Delay Time	5		ns	
99	TdPIA(AS)	/PITACK Rise to /AS Fall Delay Time	5		ns	
100	TdPIA(DRa)	/PITACK Fall to Data Active Delay	0		ns	

AC CHARACTERISTICS
Z16C30

No	Symbol	Parameter	Min	Max	Units	Notes
101	TdPIA(DRn)	/PITACK Rise to Data Not Valid Delay	0		ns	
102	TdPIA(DRz)	/PITACK Rise to Data Float Delay		20	ns	
103	TsIEI(PIA)	IEI to /PITACK Fall Setup Time	10		ns	
104	ThIEI(PIA)	IEI to /PITACK Rise Hold Time	0		ns	
105	TdPIA(IEO)	/PITACK Fall to IEO Delay		60	ns	
106	TdPIA(INT)	/PITACK Fall to /INT Inactive Delay		200	ns	
107	TdPIAf(RDY)	/PITACK Fall to /RDY Fall Delay		200	ns	
108	TdPIAr(RDY)	/PITACK Rise to /RDY Rise Delay		40	ns	
109	TdPIA(Wf)	/PITACK Fall to /WAIT Fall Delay		40	ns	
110	TdPIA(Wr)	/PITACK Fall to /WAIT Rise Delay		200	ns	
111	TdSIA(INT)	/SITACK Fall to IEO Inactive Delay		200	ns	[2]
112	TwSTBh	/Strobe High Width	50		ns	[3]
113	TwRESl	/RESET Low Width	170		ns	
114	TwRESh	/RESET High Width	60		ns	
115	Tdres(STB)	/RESET Rise to /STB Fall	60		ns	[3]
116	TdDSf(RDY)	/DS Fall to /RDY Fall Delay		50	ns	
117	TdWRf(RDY)	/WR Fall to /RDY Fall Delay		50	ns	
118	TdWRr(RDY)	/WR Rise to /RDY Rise Delay		40	ns	
119	TdRDf(RDY)	/RD Fall to /RDY Fall Delay		50	ns	
120	TdRAKf(RDY)	/RxACK Fall to /RDY Fall Delay		50	ns	
121	TdRAKr(RDY)	/RxACK Rise to /RDY Rise Delay		40	ns	
122	TdTAKf(RDY)	/TxACK Fall to /RDY Fall Delay		50	ns	
123	TdTAKr(RDY)	/TxACK Rise to /RDY Rise Delay		40	ns	

Notes:

[1] Direct address is any of A//B, D//C or AD15-AD8 used as an address bus.

[2] The parameter applies only when /AS is not present.

[3] Strobe (/STB) is any of /DS, /RD, /WR, /PITACK, /RxACK or /TxACK.

[4] Parameter applies only if read empties the receive FIFO.

[5] Parameter applies only if write fills the transmit FIFO.

[6] For extended temperature part TdDSI(Wf) max = 220 ns.

[7] For extended temperature part TdDSF(TRQ) max = 75 ns.

TIMING DIAGRAMS

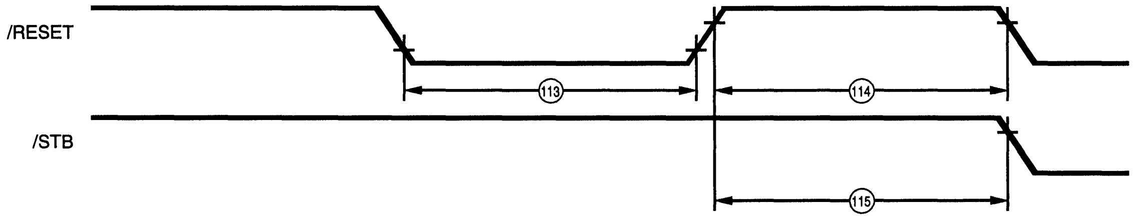


Figure 55. Reset Timing

Note:
/STB is any of /DS, /RD, /WR, /PITACK, /RxACK, or /TxACK.

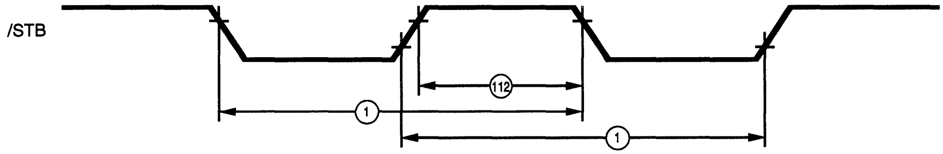


Figure 56. Bus Cycle Timing

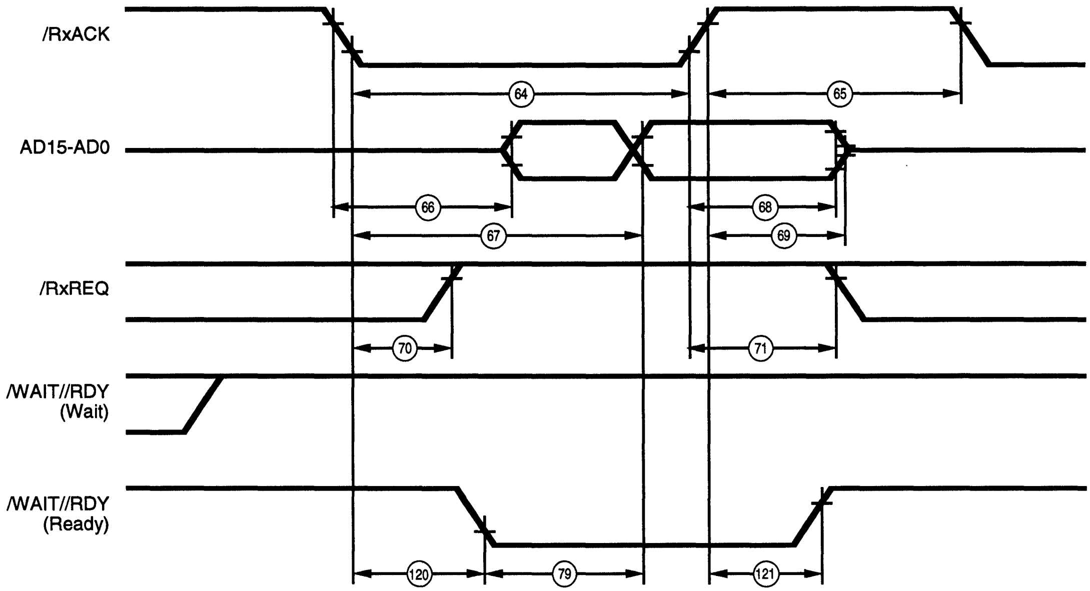


Figure 57. DMA Read Cycle

TIMING DIAGRAMS (Continued)

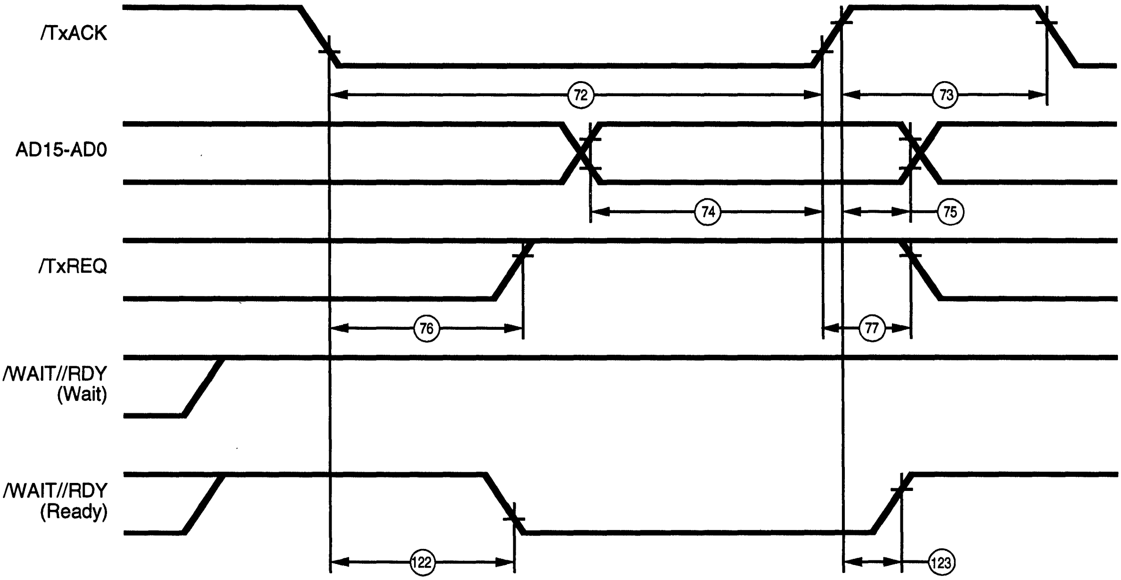


Figure 58. DMA Write Cycle

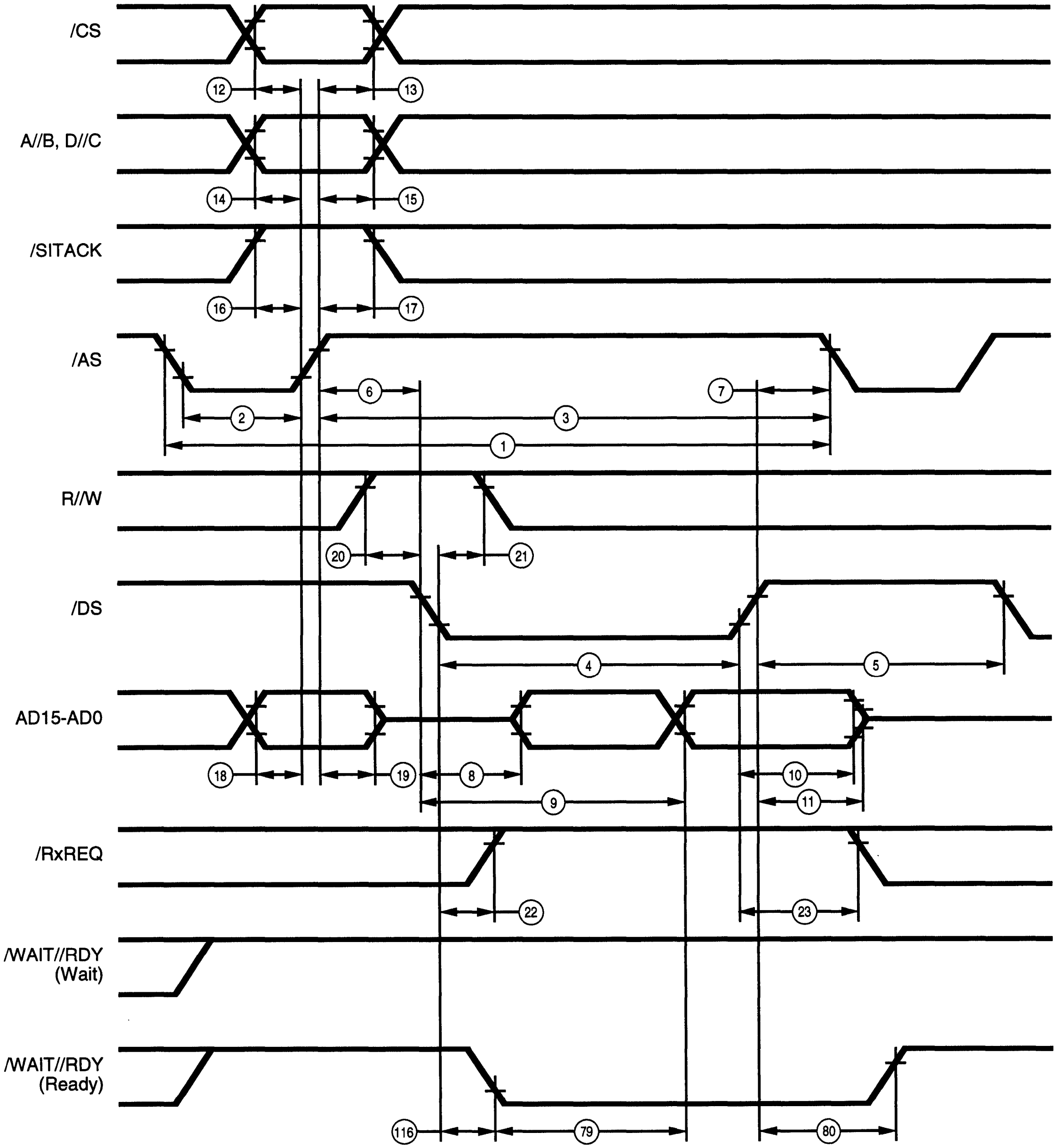


Figure 59. Multiplexed /DS Read Cycle

TIMING DIAGRAMS (Continued)

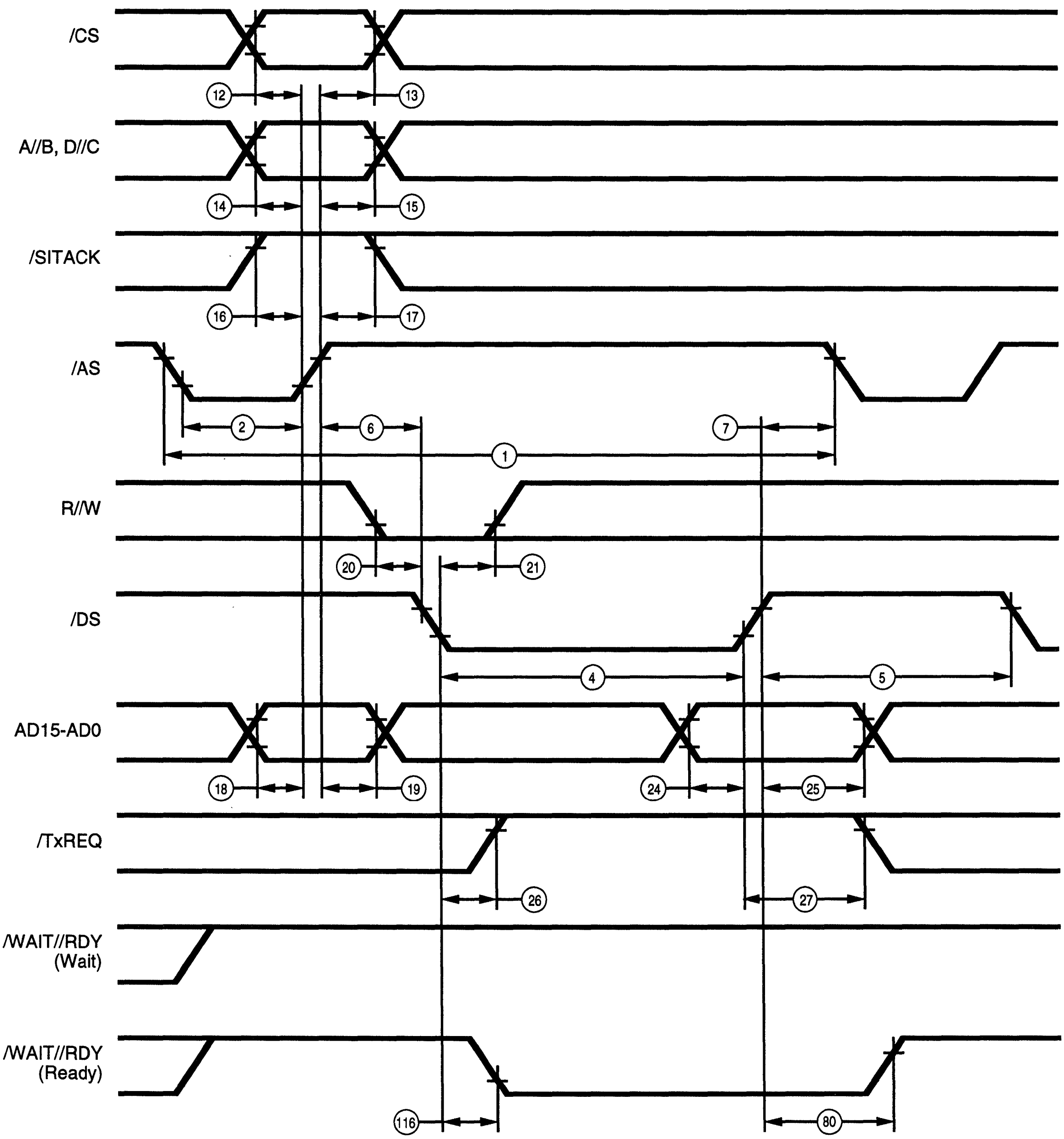


Figure 60. Multiplexed /DS Write Cycle

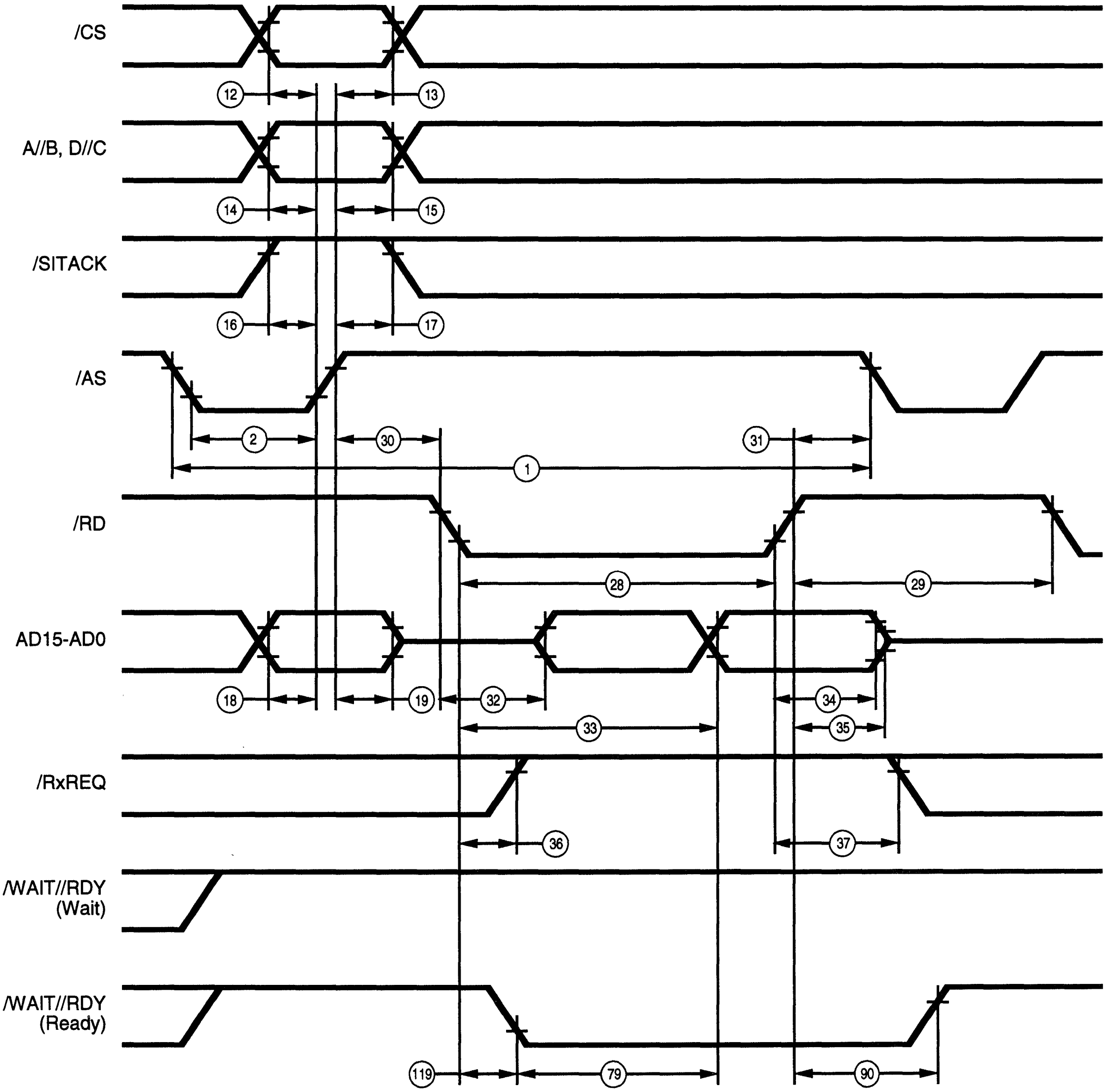


Figure 61. Multiplexed /RD Read Cycle

TIMING DIAGRAMS (Continued)

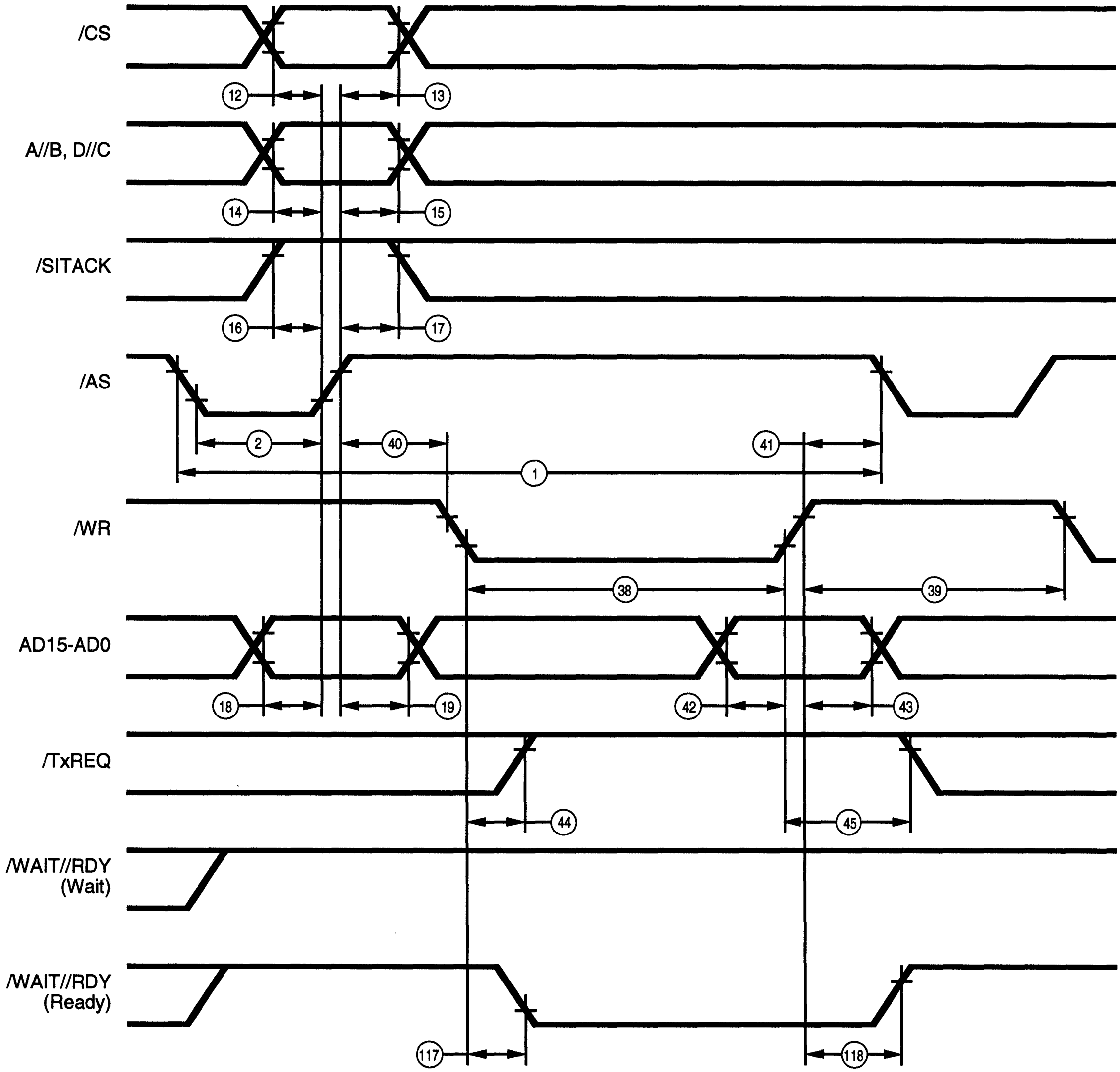


Figure 62. Multiplexed /WR Write Cycle

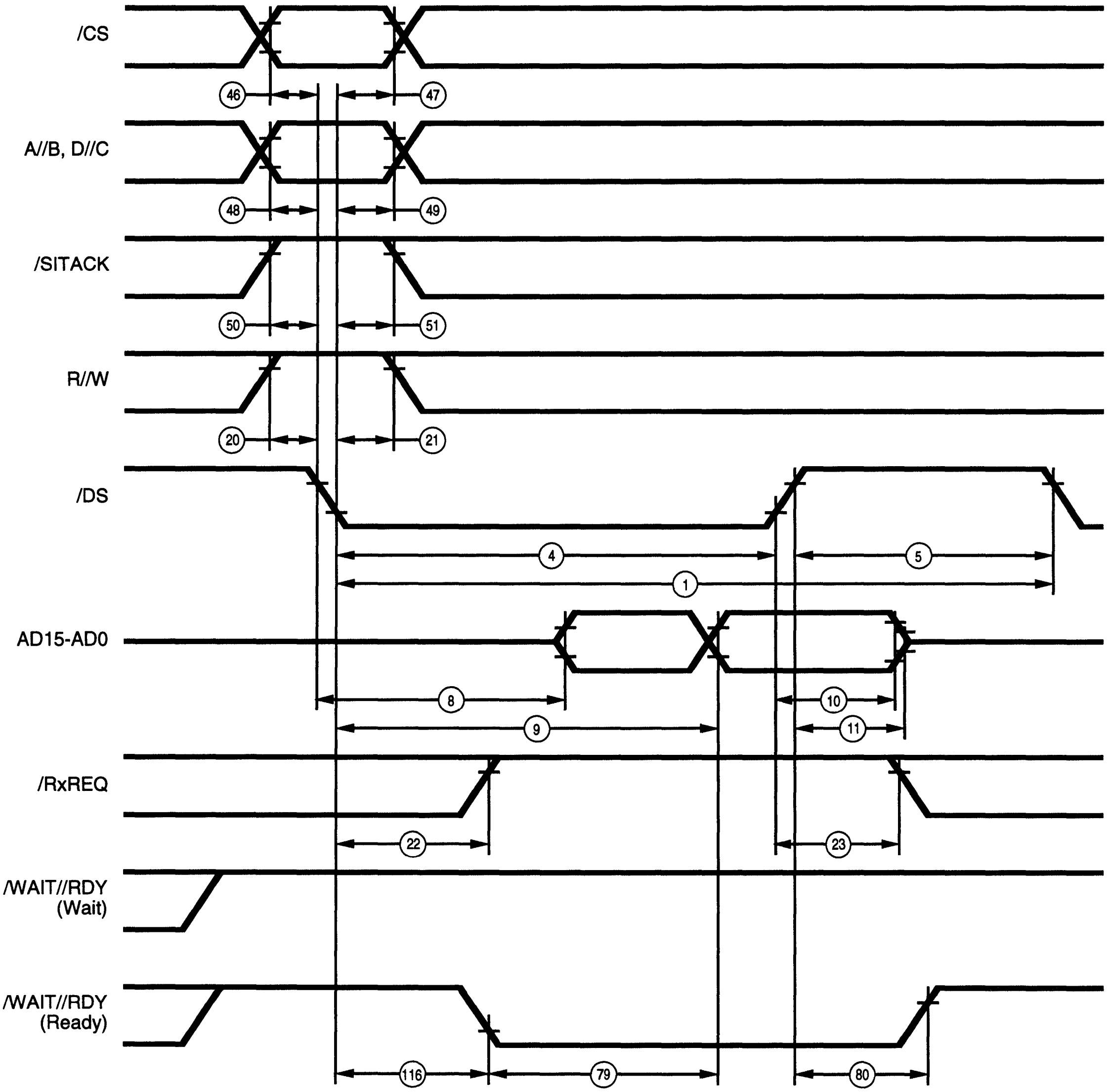


Figure 63. Non-Multiplexed /DS Read Cycle

TIMING DIAGRAMS (Continued)

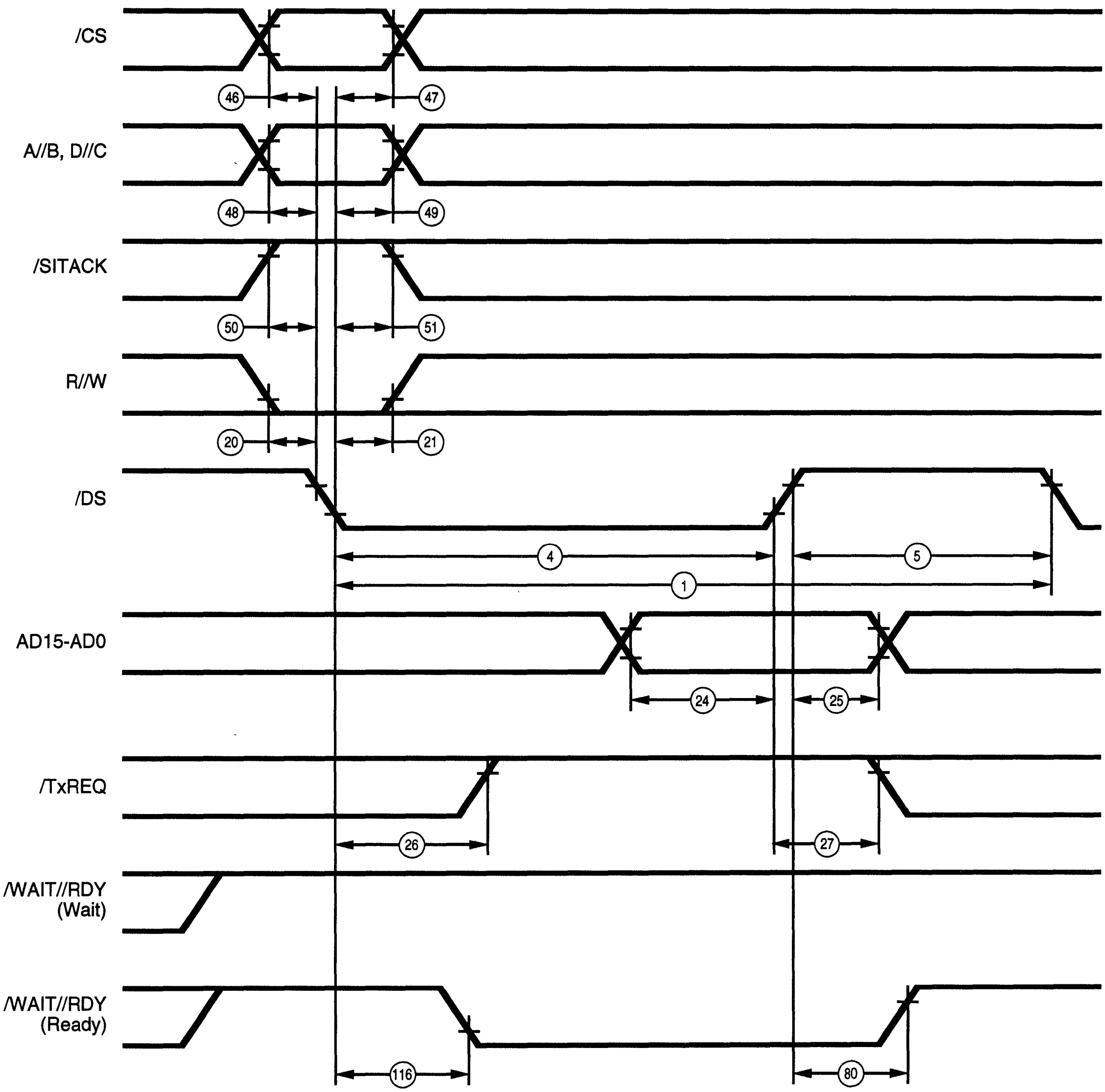


Figure 64. Non-Multiplexed /DS Write Cycle

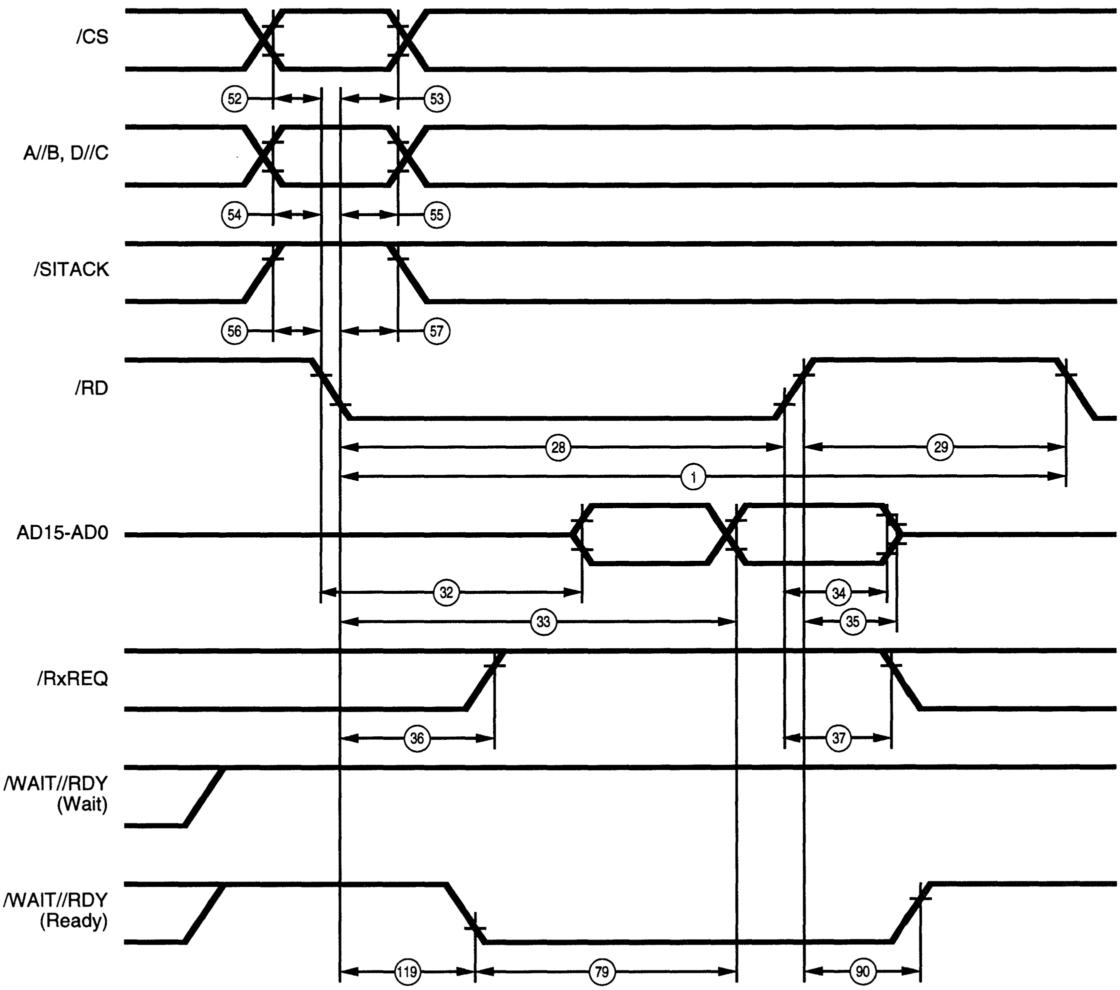


Figure 65. Non-Multiplexed /RD Read Cycle

TIMING DIAGRAMS (Continued)

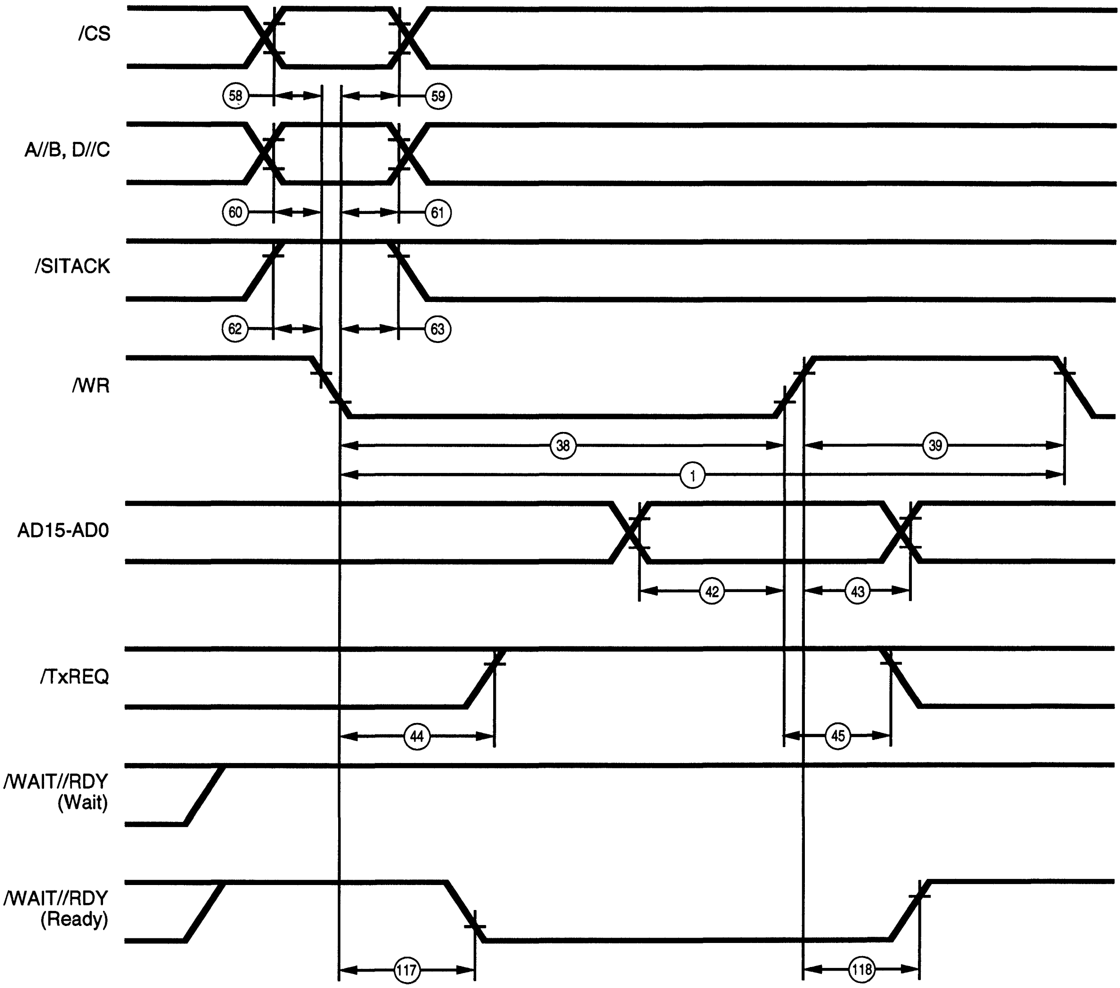


Figure 66. Non-Multiplexed /WR Write Cycle

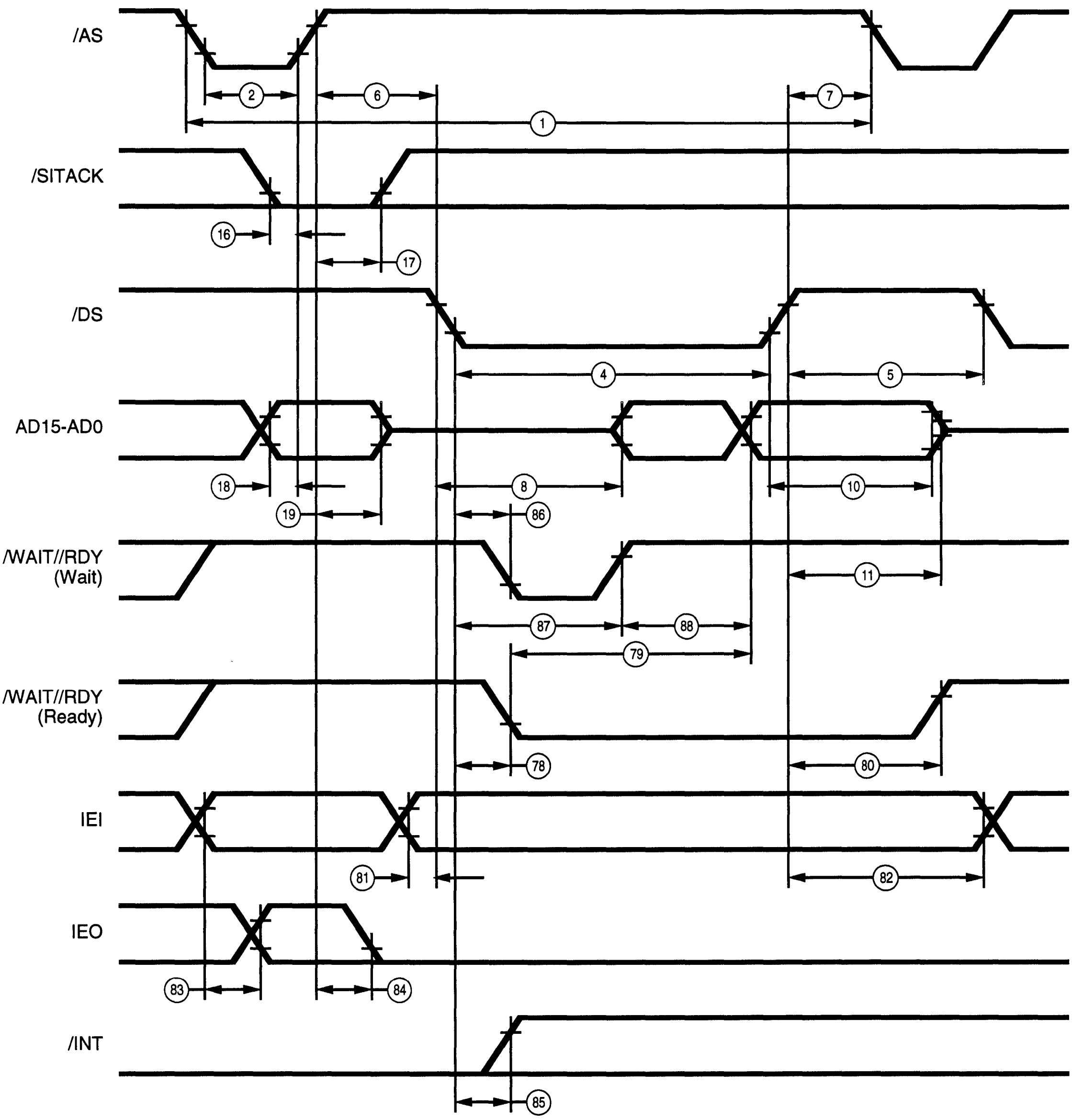


Figure 67. Multiplexed /DS Interrupt Acknowledged Cycle

TIMING DIAGRAMS (Continued)

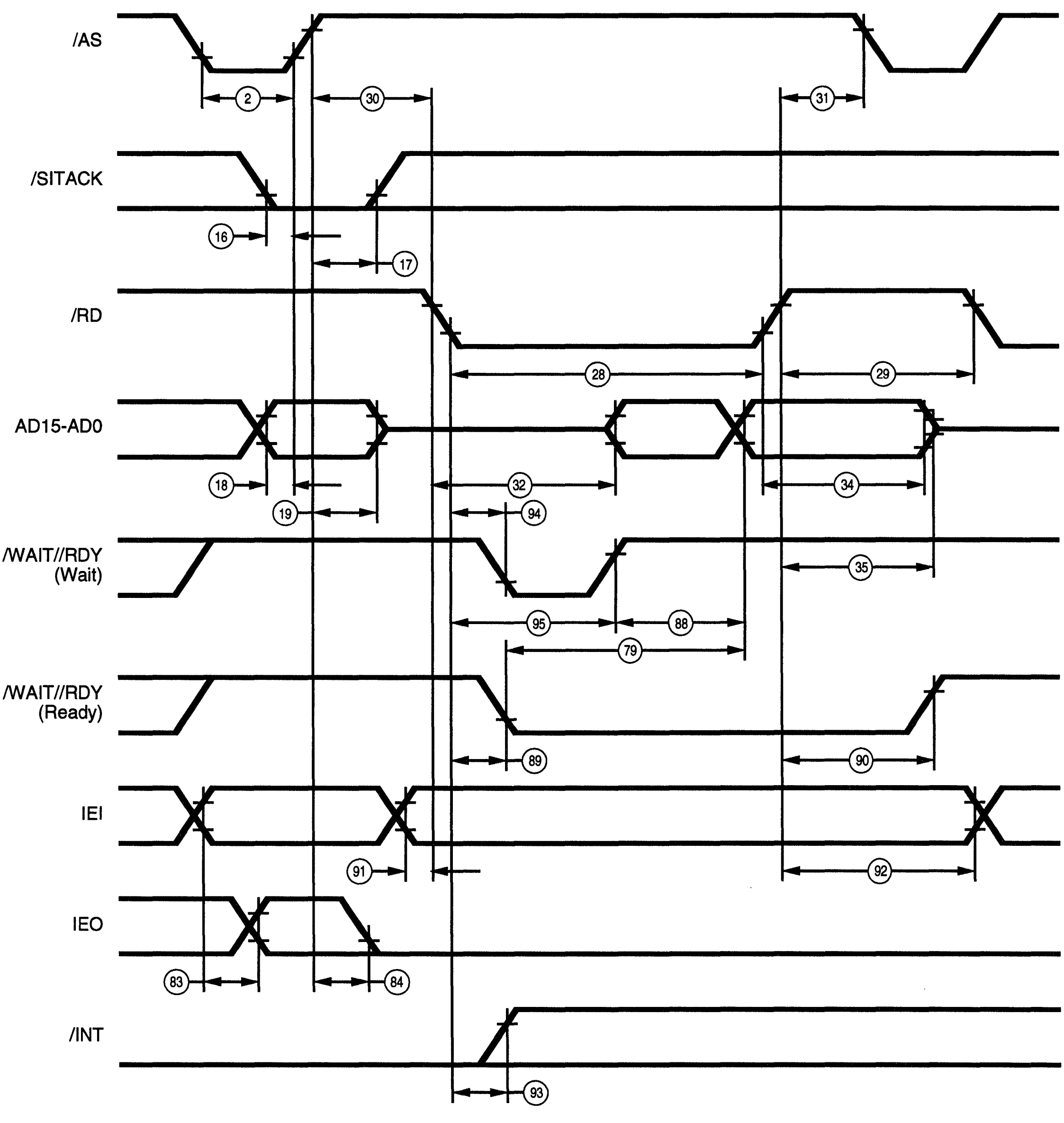


Figure 68. Multiplexed /RD Interrupt Acknowledge Cycle

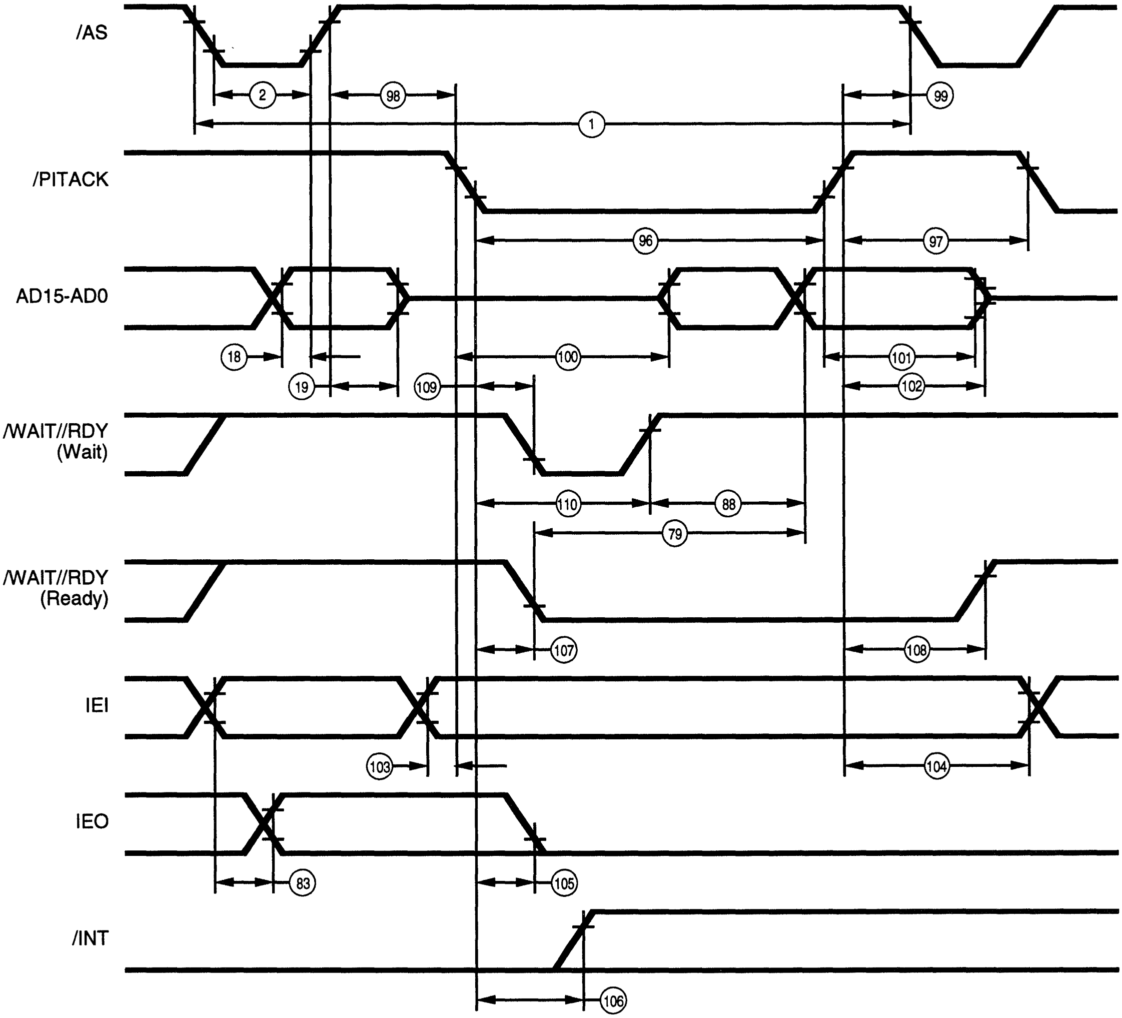


Figure 69. Multiplexed Pulsed Interrupt Acknowledge Cycle

TIMING DIAGRAMS (Continued)

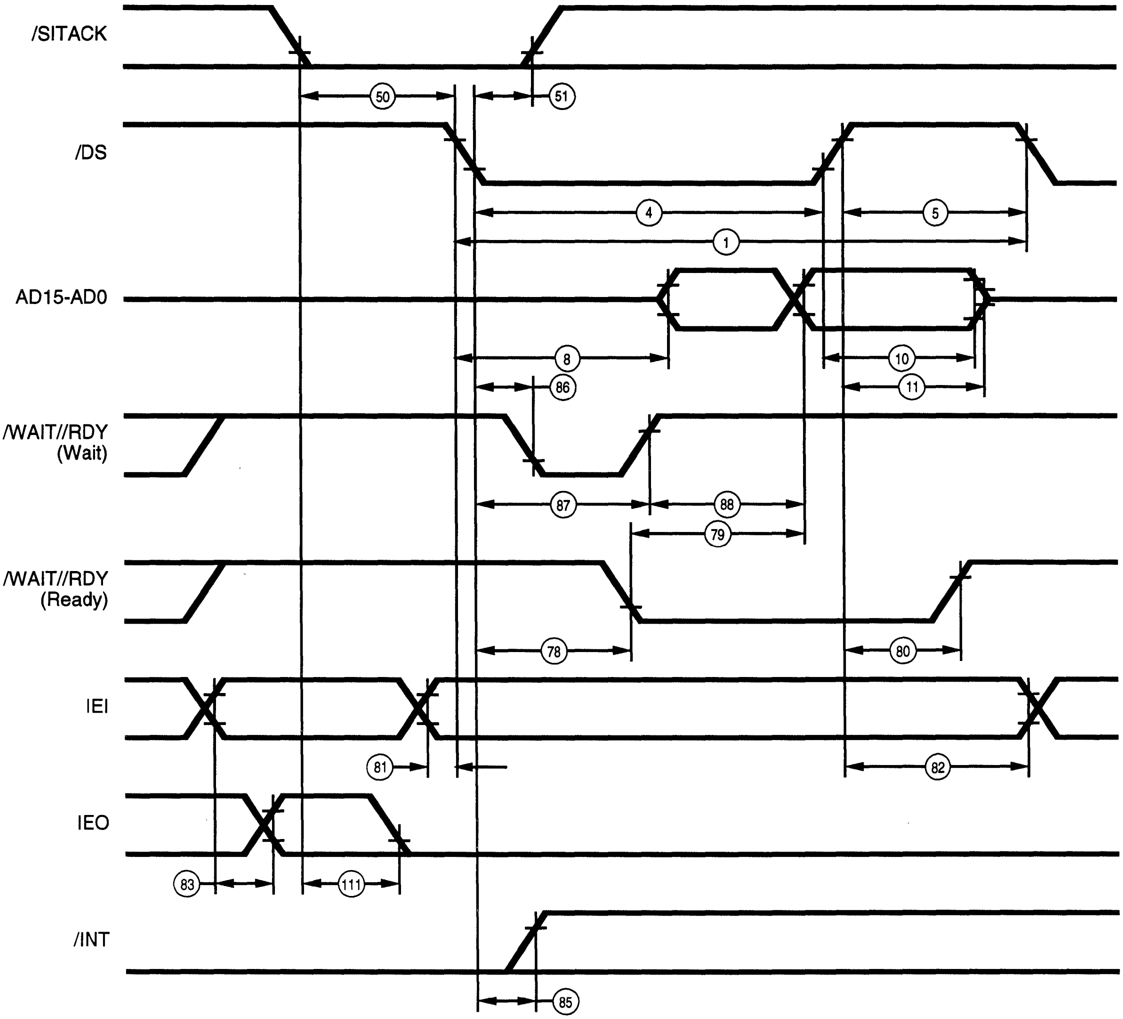


Figure 70. Non-Multiplexed /DS Interrupt Acknowledge Cycle

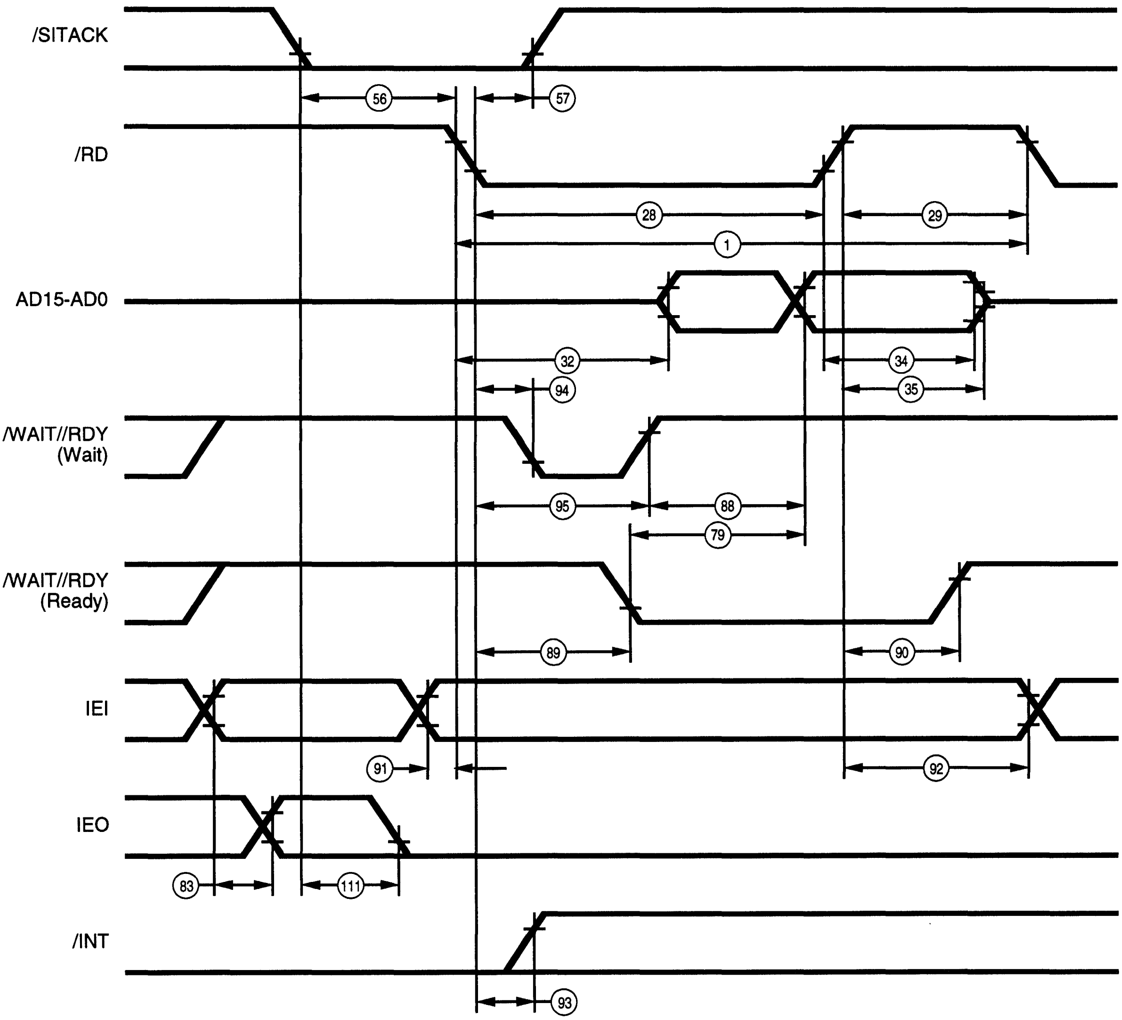


Figure 71. Non-Multiplexed /RD Interrupt Acknowledge Cycle

TIMING DIAGRAMS (Continued)

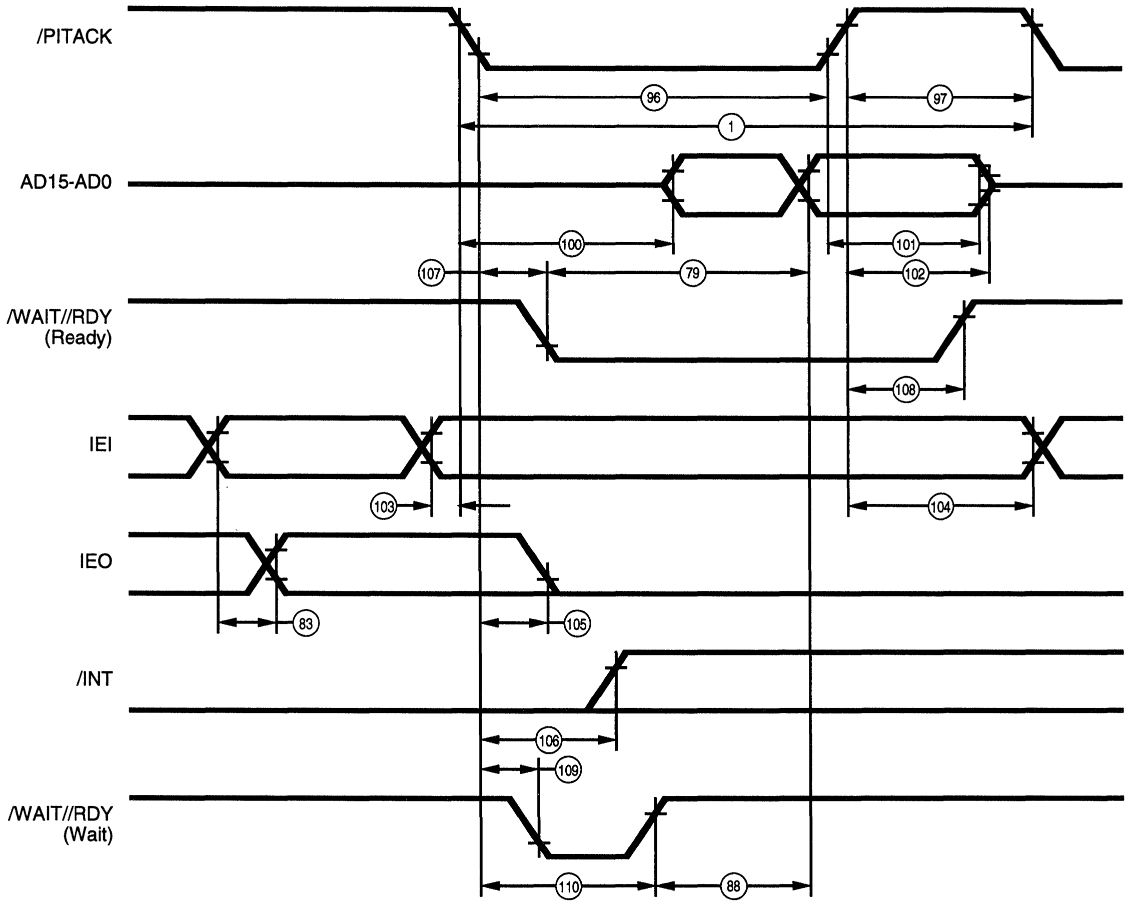


Figure 72. Non-Multiplexed Pulsed Interrupt Acknowledge Cycle

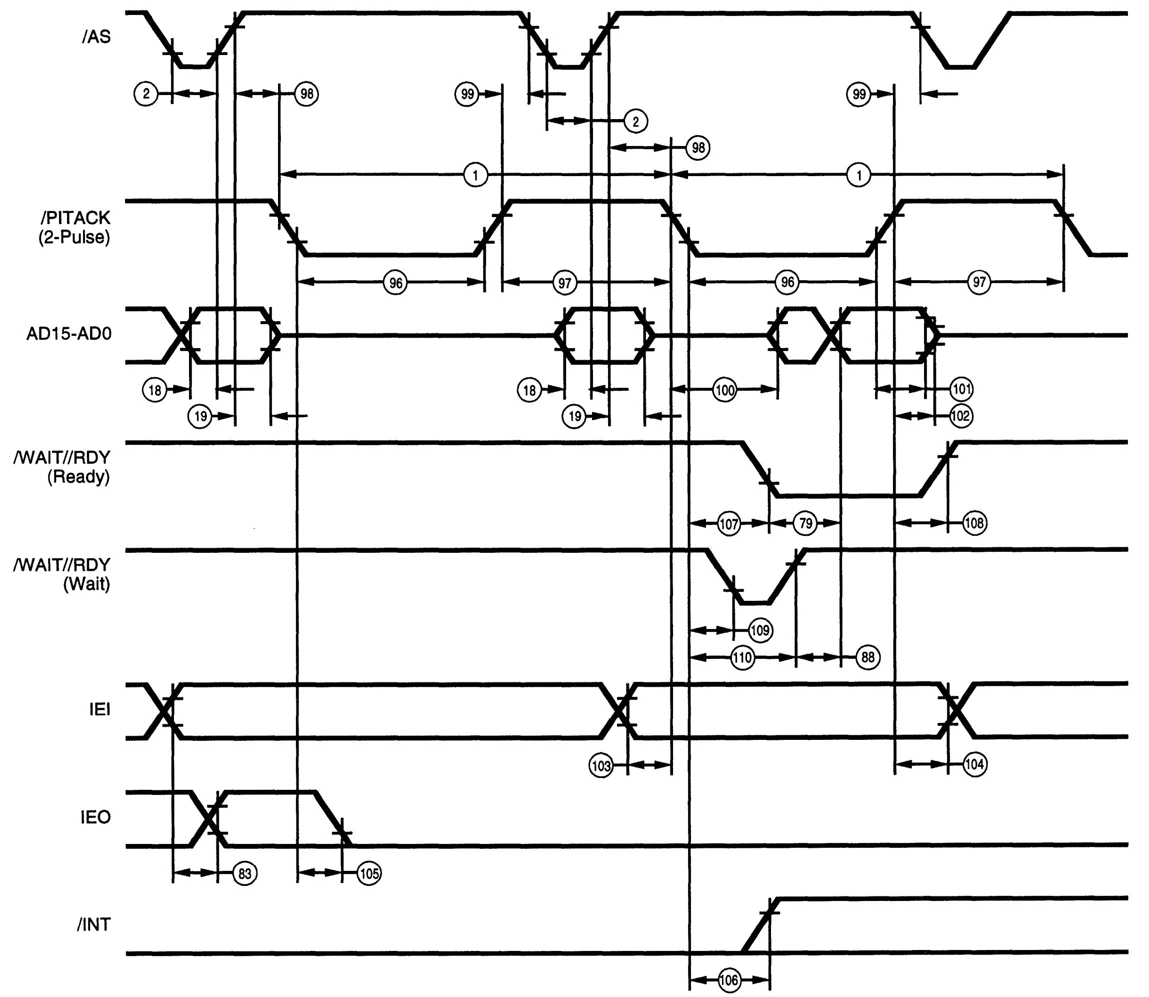


Figure 73. Multiplexed Double-Pulse Intack Cycle

TIMING DIAGRAMS (Continued)

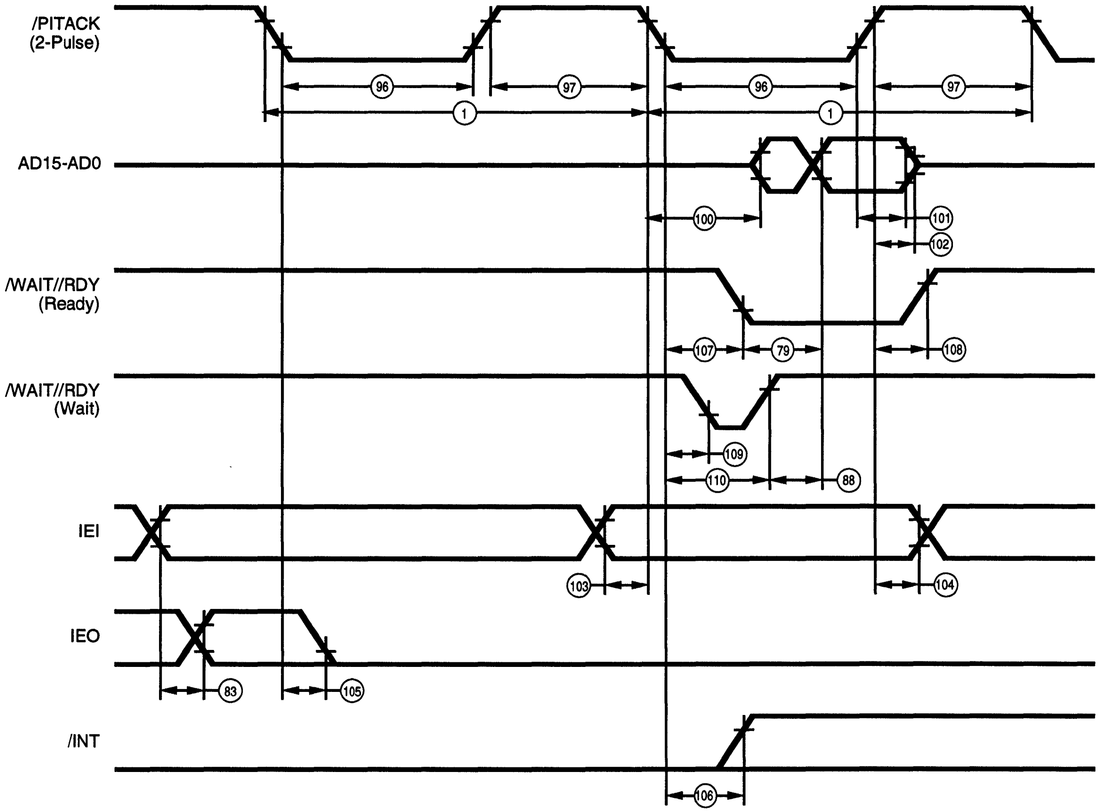


Figure 74. Non-Multiplexed Double-Pulse Intack Cycle

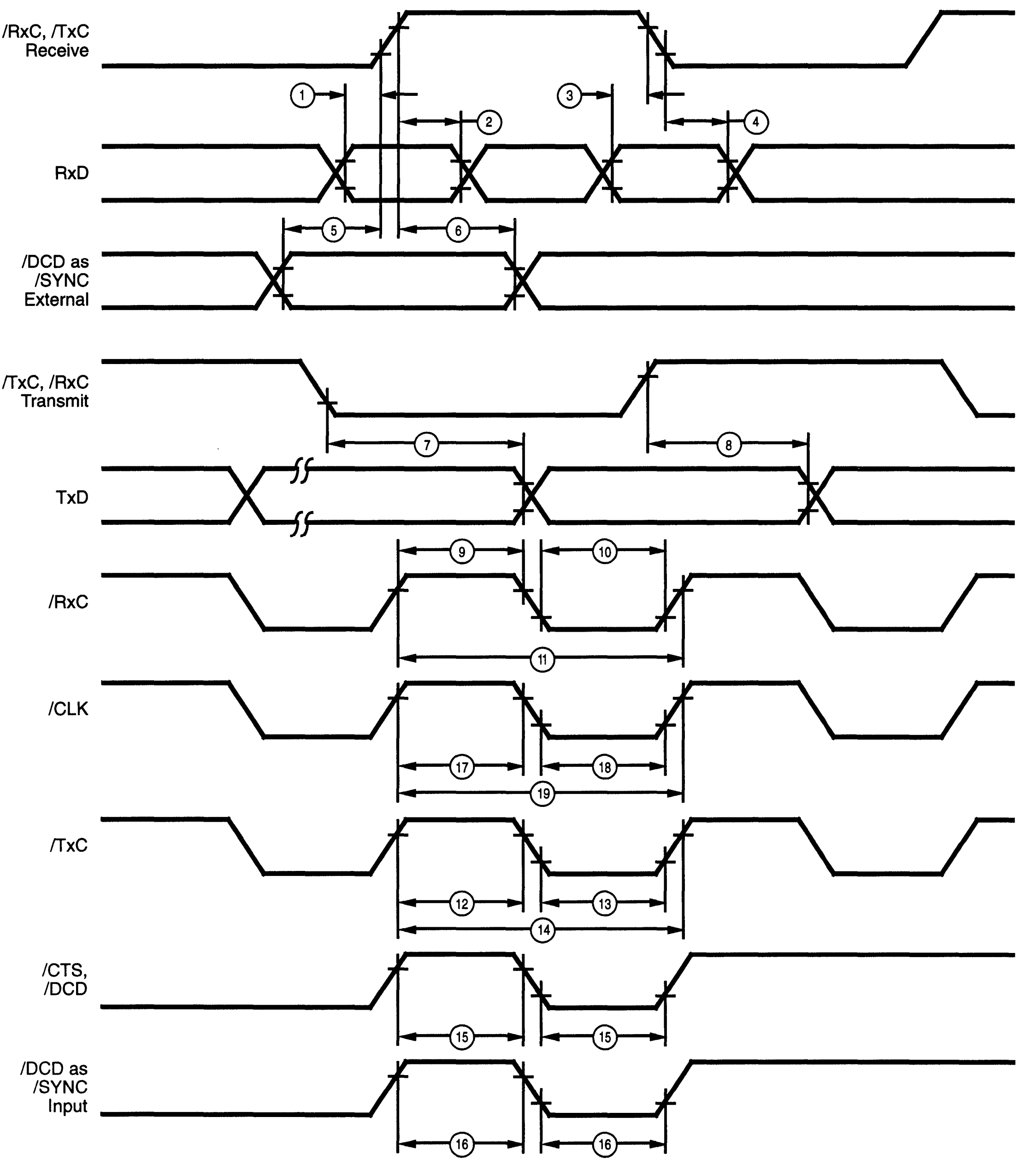


Figure 75. Z16C30 General Timing

Note:
/CLK is /RxC or /TxC when supplying DPLL, BRG, or CTR clock.

AC CHARACTERISTICS
Z16C30 General Timing

No	Symbol	Parameter	Min	Max	Units	Notes
1	TsRxD(RxCr)	RxD to /RxC Rise Setup Time (x1 Mode)	0		ns	[1]
2	ThRxD(RxCr)	RxD to /RxC Rise Hold Time (x1 Mode)	40		ns	[1]
3	TsRxd(RxCf)	RxD to /RxC Fall Setup Time (x1 Mode)	0		ns	[1,3]
4	ThRxD(RxCf)	RxD to /RxC Fall Hold Time (x1 Mode)	40		ns	[1,3]
5	TsSy(RxC)	/DCD as /SYNC to /RxC Rise Setup Time	0		ns	[1]
6	ThSy(RxC)	/DCD as /SYNC to /RxC Rise Hold Time (x1 Mode)	40		ns	[1]
7	TdTxCf(TxD)	/TxC Fall to TxD Delay		50	ns	[2]
8	TdTxCr(TxD)	/TxC Rise to TxD Delay		50	ns	[2,3]
9	TwRxCh	/RxC High Width	40		ns	[1]
10	TwRxCl	/RxC Low Width	40		ns	[1]
11	TcRxC	/RxC Cycle Time	100		ns	[1]
12	TwTxCh	/TxC High Width	40		ns	[2]
13	TwTxCl	/TxC Low Width	40		ns	[2]
14	TcTxC	/TxC Cycle Time	100		ns	[2]
15	TwExT	/DCD or /CTS Pulse Width	70		ns	
16	TWSY	/DCD as /SYNC Input Pulse Width	70		ns	
17	TwCLKh	/CLK High Width	20		ns	[4]
18	TwCLKl	/CLK High Width	20		ns	[4]
19	TcCLK	/CLK Cycle Time	50		ns	[4]

Notes:

[1] /RxC is /RxC or /TxC, whichever is supplying the receive clock.

[2] /TxC is /TxC or /RxC, whichever is supplying the transmit clock.

[3] Parameter applies only to FM encoding/decoding.

[4] /CLK is /RxC or /TxC, when supplying DPLL, BRG, or CTR clock.

TIMING DIAGRAMS (Continued)

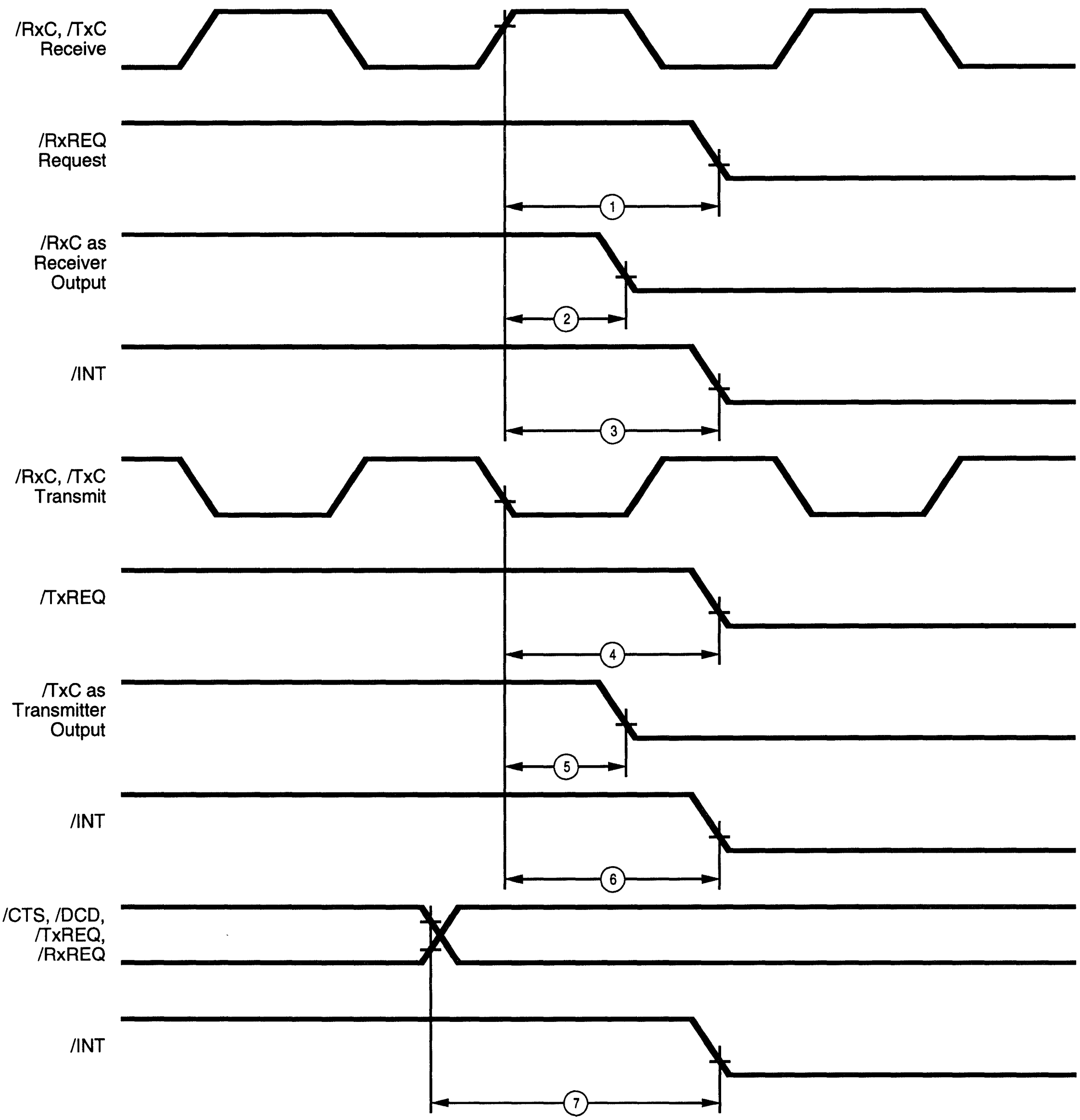


Figure 76. Z16C30 System Timing

AC CHARACTERISTICS

Z16C30 System Timing

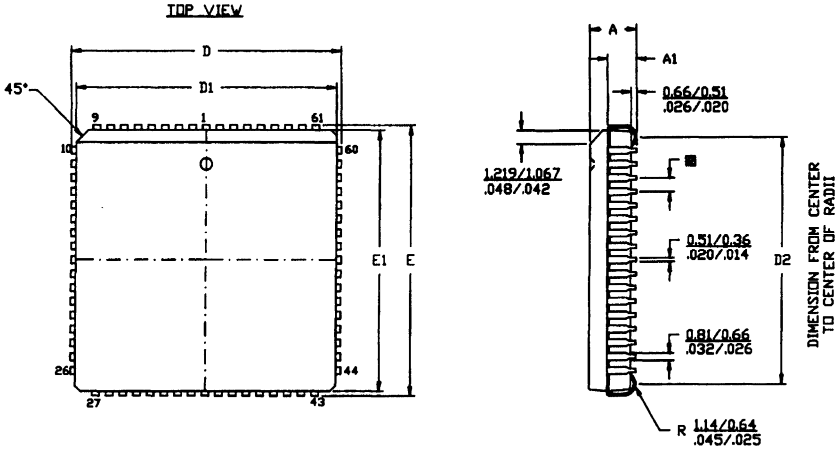
No	Symbol	Parameter	Min	Max	Units	Notes
1	TdRxC(REQ)	/RxC Rise to /RxREQ Valid Delay		100	ns	[1]
2	TdRxC(RxC)	/TxC Rise to /RxC as Receiver Output Valid Delay		100	ns	[1]
3	TdRxC(INT)	/RxC Rise to /INT Valid Delay		100	ns	[1]
4	TdTxC(REQ)	/TxC Fall to /TxREQ Valid Delay		100	ns	[2]
5	TdTxC(TxC)	/RxC Fall to /TxC as transmitter Output Valid Delay		100	ns	[2]
6	TdTxC(INT)	/TxC Fall to /INT Valid Delay		100	ns	[2]
7	TdEXT(INT)	/CTS, /DCD, /TxREQ, /RxREQ transition to /INT Valid Delay		100	ns	

Notes:

[1] /RxC is /RxC or /TxC, whichever is supplying the receive clock.

[2] /TxC is /TxC or /RxC, whichever is supplying the transmit clock.

PACKAGE INFORMATION



NOTES:

1. CONTROLLING DIMENSIONS : INCH
2. LEADS ARE COPLANAR WITHIN .004 IN.
3. DIMENSION : $\frac{\text{MM}}{\text{INCH}}$

SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	4.32	4.57	.170	.180
A1	2.67	2.92	.105	.115
D/E	25.02	25.40	.985	1.000
D1/E1	24.13	24.33	.950	.958
D2	22.86	23.62	.900	.930
■	1.27 TYP		.050 TYP	

68-Pin PLCC Package Diagram

ORDERING INFORMATION

Z16C30

10 MHz
68-Pin PLCC
Z16C3010VSC

For fast results, contact your local Zilog sales office for assistance in ordering the part desired.

Package
V = Plastic Leaded Chip Carrier

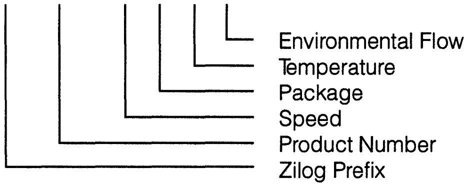
Temperature
S = 0°C to +70°C

Speed
10 = 10 MHz

Environmental
C = Plastic Standard

Example:

Z 16C30 10 V S C is a Z16C30, 10 MHz, PLCC, 0°C to +70°C, Plastic Standard Flow



Environmental Flow
Temperature
Package
Speed
Product Number
Zilog Prefix

PRELIMINARY PRODUCT SPECIFICATION

Z16C32

IUSC™ INTEGRATED UNIVERSAL SERIAL CONTROLLER

FEATURES

- Two Full-Capacity 20 MHz DMA Channels, Each with 32-Bit Addressing and 16-Bit Data Transfers.
- DMA Modes Include Single Buffer, Pipelined, Array-Chained and Linked-Array Chained.
- Ring Buffer Feature Supports Circular Queue of Buffers in Memory.
- Linked Frame Status Transfer Feature Writes Status Information for Received Frames and Reads Control Information for Transmit Frames to the DMA Channel's Array or Linked List to Significantly Simplify Processing Frame Status and Control Information.
- Programmable Throttling of DMA Bus Occupancy in Burst Mode with Bus Occupancy Time Limitation.
- 0 to 20 Mbit/sec, Full-Duplex Channel, with Two Baud Rate Generators and a Digital Phase-Locked Loop for Clock Recovery.
- 32-Byte Data FIFOs for Receiver and Transmitter
- Up to 12.5 MByte/sec (16-Bit) Data Bus Bandwidth
- Multiprotocol Operation Under Program Control with Independent Mode Selection for Receiver and Transmitter.
- Async Mode with One-to-Eight Bits/Character, 1/16 to Two Stop Bits/Character in 1/16 Bit Increments; 16x, 32x, or 64x Oversampling; Break Detect and Generation; Odd, Even, Mark, Space or No Parity and Framing Error Detection. Supports Nine-Bit and MIL-STD-1553B Protocols.
- HDLC/SDLC Mode with 8-Bit Address Compare; Extended Address Field Option; 16- or 32-Bit CRC; Programmable Idle Line Condition; Optional Preamble Transmission and Loop Mode. Selectable Number of Flags Between Back-to-Back Frames.
- Byte Oriented Synchronous Mode with One-to-Eight Bits/Character; Programmable Sync and Idle Line Conditions; Optional Receive Sync Stripping; Optional Preamble Transmission; 16- or 32-Bit CRC; Transmit-to-Receive Slaving (for X.21).
- External Character Sync Mode for Receive
- Transparent Bisync Mode with EBCDIC or ASCII Character Code; Automatic CRC Handling; Programmable Idle Line Condition; Optional Preamble Transmission; Automatic Recognition of DLE, SYN, SOH, ITX, ETX, ETB, EOT, ENQ and ITB.
- Flexible Bus Interface for Direct Connection to Most Microprocessors; User Programmable for 8 or 16 Bits Wide. Directly Supports 680X0 Family or 8X86 Family Bus Interfaces.
- Receive and Transmit Time Slot Assigners for ISDN, T1 and E1 (CEPT) Applications.
- 8-Bit General-Purpose Port with Transition Detection
- Low Power CMOS
- 68-Pin PLCC Package
- Electronic Programmer's Manual Support Tool and Software Drivers are Available.

GENERAL DESCRIPTION

The IUSC™ (Integrated Universal Serial Controller) is a multiprotocol datacommunications device with on-chip dual-channel DMA. The integration of a high-speed serial communications channel with high-performance DMA facilitates higher data throughput than is likely to be achieved with discrete serial/DMA chip combinations.

GENERAL DESCRIPTION (Continued)

The IUSC's value is more than just reduced chip count and saving board space. The DMA and serial channel intercommunication with each other provides real application benefits. For example, events like the reception of the end of a HDLC frame is internally communicated from the serial controller to the DMA so that each frame can be written into a separate memory buffer. The buffer chaining capabilities, ring buffer support, automated frame status/control blocks, and buffer termination at the end of the frame combine to significantly reduce CPU overhead (Figure 1).

The IUSC is software configurable to satisfy a wide variety of serial communication applications. The 20 Mbit/second data rate and multiple protocol support make it ideal for applications in today's dynamic environment of changing specifications and increasing speed. The many programmable features allow the user to tune the device response to meet system requirements and adapt to future requirements. The IUSC contains a variety of sophisticated internal functions including two baud rate generators, a digital phase-locked loop, character counters, and 32-byte FIFOs for both the receiver and the transmitter.

The on-chip DMA channels allow high speed data transfers for both the receiver and the transmitter. The IUSC supports automatic status and control transfer through DMA and allows initialization of the serial controller under DMA control. Each DMA channel can do a 16-bit transfer in as little as three 50 ns clock cycles and can generate addresses compatible with 32-, 24- or 16-bit memory ranges. The DMA channels operate in any of four modes: single buffer, pipelined, array-chained, or linked-list. The array-chained and linked-list modes provide scatter-read and gather-write capabilities with minimal software intervention. To prevent the DMA from holding bus mastership too long, mastership time may be limited by counting the absolute number of clock cycles, the number of bus transactions, or both.

The CPU bus interface is designed for use with any conventional multiplexed or non-multiplexed bus from manufacturers of CISC and RISC processors including Intel, Motorola, and Zilog. The bus interface is configurable for 16-bit data, 8-bit data with separate address or 8-bit data without separate address to support multiplexed or non-multiplexed busses.

The IUSC handles asynchronous formats, synchronous bit-oriented formats such as HDLC and synchronous byte-oriented formats (e.g., BISYNC and DDCMP). This device supports virtually any serial data transfer application.

The IUSC can generate and check CRC in any synchronous mode. Complete access to the CRC value allows system software to resend or manipulate the CRC as needed in various applications. The IUSC also provides facilities for modem control signals. In applications where these controls are not needed, the modem controls can be used for general-purpose I/O.

Interrupts are supported by a daisy-chain hierarchy within the serial channel and between the serial channel and the DMA. Separate interrupt vectors for each type of interrupt within the serial controller and the DMA facilitate fast discrimination of the interrupt source. The IUSC supports Pulsed, Double Pulsed, and Status Interrupt Acknowledge cycles.

Support tools are available to aid the designer in efficiently programming the IUSC. The Technical Manual describes in detail all the features and gives programming sequence hints. The Electronic Programmer's Manual, DC #8287-02, is an MS-DOS, disk-based programming initialization tool that can generate custom sequences. Also, Zilog offers assorted application notes and development boards to assist the designer in hardware and software development. Contact your nearest Zilog representative for additional information.

Notes:
All Signals with a preceding front slash, "/", are active Low, e.g.: B//W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

Connection	Circuit	Device
Power	V_{CC}	V_{DD}
Ground	GND	V_{SS}

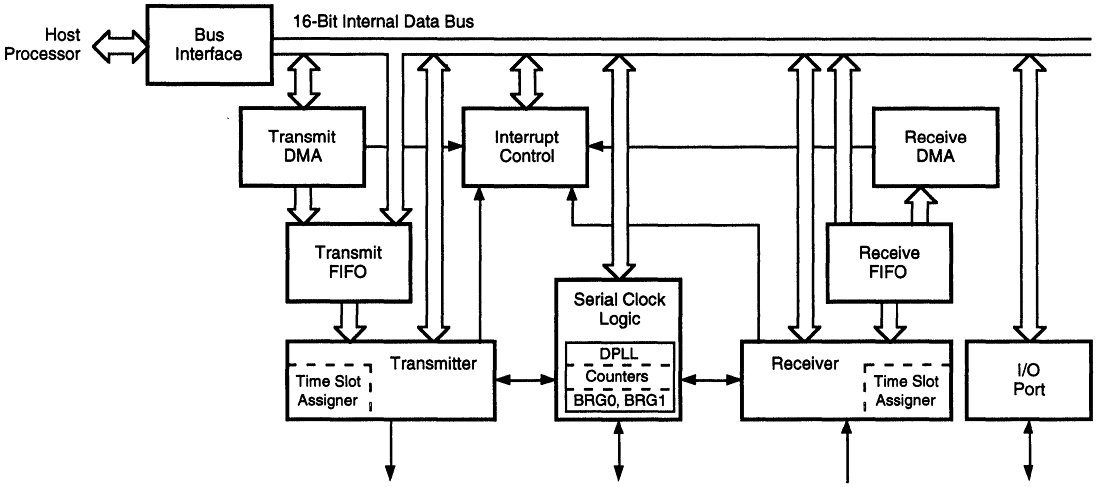


Figure 1. IUSC Block Diagram

GENERAL DESCRIPTION (Continued)

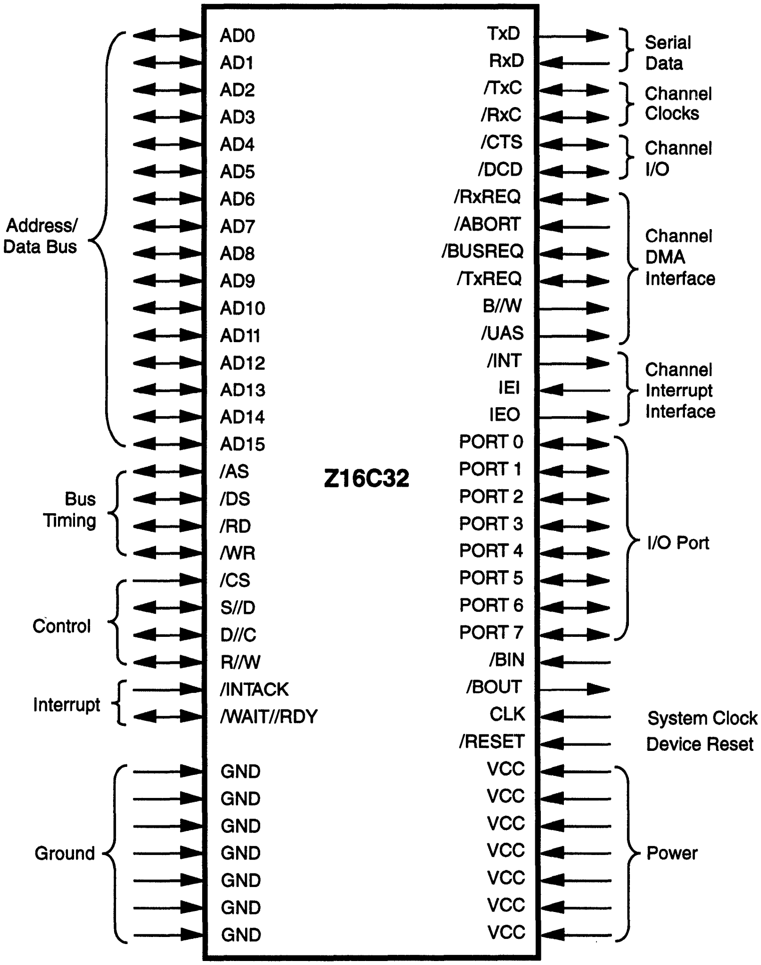


Figure 2. Pin Functions

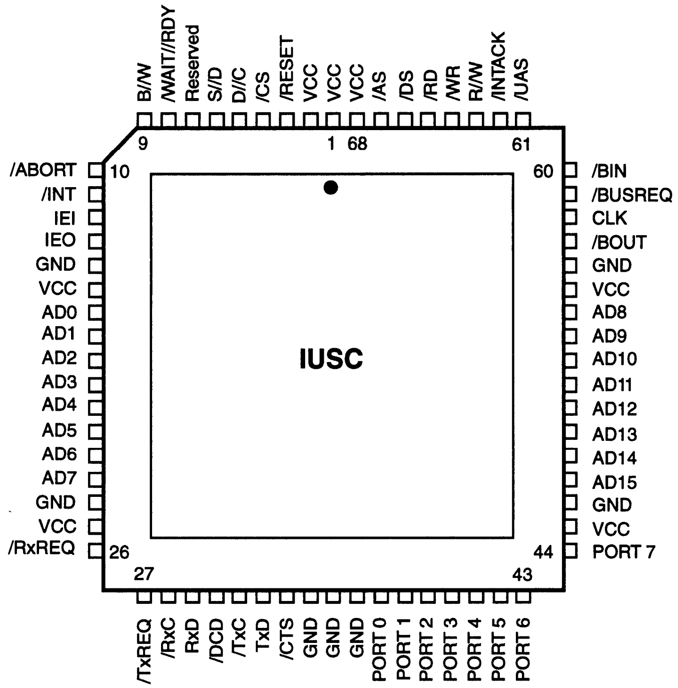


Figure 3. 68-Pin PLCC Pin Assignments

PIN DESCRIPTION

Figure 2 shows the logical pin groupings of the IUSC's pins, and Figure 3 shows the physical pin assignments.

Only one strobe pin (/DS, /RD, /WR or Pulsed INTACK) should ever be active at one time. Any unused input pin (if an input when the IUSC is bus master or slave) must be pulled up to its inactive state.

/RESET *Reset* (input, active Low). A Low on this line places the IUSC in a known, inactive state, and conditions it so that the data, from the next write operation that asserts the /CS pin, goes into the Bus Configuration Register (BCR) regardless of register addressing. /RESET should be driven Low as soon as possible during power-up, and as needed when restarting the overall system or the communications subsystem.

CLK *System Clock* (input). This signal is the timing reference for the DMA and bus interface logic. (The serial controller section is clocked by the selected sources of receive and transmit clocking.)

AD15-0 *Address/Data Bus* (inputs/tri-state outputs). After Reset, these lines carry data between the controlling microprocessor and the IUSC, and may also carry multiplexed addresses of registers within the IUSC. Such operation, between the host processor and the IUSC, is often called slave mode. Once the software has set up the device and placed it into operation, these lines also carry multiplexed addresses and data between the IUSC and system memory; such operation is called master mode. AD15-0 can be used in a variety of ways based on whether the IUSC senses activity on /AS after Reset, and on the data written to the Bus Configuration Register (BCR).

/CS *Chip Select* (input, active Low). A Low on this line indicates that the controlling microprocessor's current bus cycle refers to a register in the IUSC. The IUSC ignores /CS when a Low on /INTACK indicates that the current bus operation is an interrupt acknowledge cycle. On a multiplexed bus the IUSC latches the state of this pin at rising edges on /AS; on a non-multiplexed bus, it latches /CS at leading/falling edges on /DS, /RD, or /WR.

PIN DESCRIPTION (Continued)

S//D *Serial/DMA* (input/tri-state output, input High indicates serial). Cycles with /CS Low, and /INTACK and this pin both High, access registers in the serial controller section. Cycles with /INTACK High, and /CS and this pin both Low, access registers in the DMA controller section. The state of this line when the Bus Configuration Register is written determines wait vs acknowledge operation, as described in the text. On a multiplexed bus, the IUSC latches the state of this pin at rising edges on /AS; on a non-multiplexed bus, it latches the state at leading/falling edges on /DS, /RD, or /WR.

Software can program the IUSC so that when it is acting as a bus master, it drives this line High to indicate a DMA cycle for serial data and Low to indicate an "array" or "list" access. (Array/list accesses read the address and length of the next memory buffer.)

D//C *Data/Control* (input/tri-state output, input High indicates Data). A slave read cycle with /CS Low , and all three of /INTACK, S//D, and this pin High, fetches data from the serial controller's receive FIFO through the Receive Data Register (RDR). A slave write cycle with the same conditions writes data into the transmit FIFO through its Transmit Data Register (TDR). Slave cycles with /INTACK and S//D High, and /CS and this pin Low , read or write registers in the serial controller. On a multiplexed bus, the IUSC determines which register to access from the low-order AD lines at the rising edge of /AS; on a non-multiplexed bus it typically selects the register based on the Least Significant Bits of the serial controller's Channel Command/Address Register. On a multiplexed bus, the IUSC latches the state of this pin at rising edges on /AS; on a non-multiplexed bus it latches the state at leading/ falling edges on /DS, /RD, or /WR.

For slave cycles on a multiplexed bus, with /INTACK High and both /CS and S//D Low, the state of this line at the rising edge of /AS selects between the registers of the transmit DMA channel (Low) and those of the receive DMA channel (High). On a non-multiplexed bus, with /INTACK High and /CS and S//D both Low, the IUSC can take the DMA channel selection from this line or from the DMA Command/ Address Register.

Software can program the IUSC so that when it is acting as a bus master, it drives this line High to indicate a DMA cycle for the receiver and Low to indicate a cycle for the transmitter.

/AS *Address Strobe* (input/tri-state output, active Low). After a reset, the IUSC's bus interface logic monitors this signal to see if the host bus multiplexes addresses and data on AD15-0. If the logic sees activity on /AS before (or during) software writes to the Bus Configuration Register, then in subsequent slave cycles directed to the IUSC, it captures register selection from the AD lines, S//D, and C//D on rising edges of /AS.

When the IUSC takes control of the bus and operates as a master, it always uses the bus in a multiplexed fashion, driving /AS Low when it places the least significant 16 bits of an address on the AD15-0 lines. External devices can be used to de-multiplex the address and data, if this is necessary to match the characteristics of the host processor or host bus.

For a non-multiplexed bus, this pin should be pulled up to +5V using a resistor of about 10 kOhms. If a processor uses a non-multiplexed bus, yet has an output called Address Strobe (e.g., 680x0 devices), this pin should not be tied to the output.

/UAS *Upper Address Strobe* (tri-state output, active Low). When the IUSC takes control of the bus and operates as a master, it drives /UAS Low when it places the more significant 16 bits of an address on AD15-0. External memory and other slave devices (or de-multiplexing latches) should capture the MS address at each rising edge on this line.

R//W *Read/Write control* (input/tri-state output, Low signifies "write"). R//W and /DS indicate read and write cycles on the bus, for host processors/buses having this kind of signalling. When the IUSC has taken control of the bus and is operating in master mode, this pin is an output that remains valid throughout the Low time of /DS. In slave cycles, the IUSC samples R//W at each leading/falling edge on /DS.

/DS *Data Strobe* (input/tri-state output, active Low). R//W and /DS indicate read and write cycles on the bus, for host processors/buses having this kind of signalling. It is an output when the IUSC has taken control of the bus and is operating in master mode, otherwise, it is an input that is qualified by /CS Low or /INTACK Low. In master mode, the R//W line remains valid throughout the Low time of this line. In slave mode, the IUSC samples R//W at each leading/ falling edge on this line. For slave write cycles and master read cycles, the IUSC captures data at the rising (trailing) edge on this line. For slave read cycles the IUSC provides valid data on the AD lines within the specified access time after this line goes Low , and keeps the data valid until after the master releases this line to High. For master write cycles, the IUSC places valid data on the AD lines before it drives this signal to Low, and keeps the data valid until after it drives this line back to High.

/RD *Read Strobe* (input/tri-state output, active Low). This line indicates a read cycle on the bus, for host processors/ buses having this kind of signalling. It is an output when the IUSC has taken control of the bus and is operating in master mode, otherwise, it is an input that is qualified by /CS Low or /INTACK Low. For master read cycles, the IUSC captures data at the rising (trailing) edge of this line. For slave read cycles the IUSC provides valid data on the AD lines within the specified access time after this line goes Low, and keeps the data valid until after the master releases this line to High.

/WR *Write Strobe* (input/tri-state output, active Low). This line indicates write cycles on the bus, for host processors/ buses having this kind of signalling. It is an output when the IUSC has taken control of the bus and is operating in master mode, otherwise it is an input that is qualified by /CS Low. For slave write cycles, the IUSC captures write data at the rising (trailing) edge of this line. For master write cycles, the IUSC places valid data on the AD lines before it drives this signal to Low , and keeps the data valid until after it drives this line back to High.

B//W *Byte / Word Select* (tri-state output, High indicates 8-bit transfer). When the IUSC takes control of the bus and operates as a master, a High on this line indicates that a byte is to be transferred, and a Low indicates that 16 bits are to be transferred. The IUSC ignores this signal during slave cycles: it takes the byte/word distinction from an AD line at the rising edge of /AS, or from a bit in the serial or DMA Command/Address Register.

/WAIT//RDY *Wait, Ready, or Acknowledge handshaking* (input/tri-state output, active Low). This line is an input when the IUSC has taken control of the bus and is operating in master mode. For slave cycles, the IUSC activates this line as an output. In both directions, the line can carry wait or acknowledge signalling depending on the state of the S//D input during the initial BCR write. If S//D is High when the BCR is written, this line operates as a Ready/Wait line for Zilog and most Intel processors. In this mode, the IUSC will not complete a master cycle while this line is Low, and it asserts this line Low until it's ready to complete an interrupt acknowledge cycle; it never asserts this line when the host accesses one of the IUSC registers.

If S//D is Low when the BCR is written, this line operates thereafter as an Acknowledge line for Motorola and some Intel processors. In this mode, the IUSC will not complete a master cycle until this line is Low. It asserts this line Low for register read and write cycles, and when it is ready to complete an interrupt acknowledge cycle. For slave cycles, this is a full time (totem pole) output. The board designer can combine this signal with similar signals from other slaves, by means of an external logic gate or a tri-state or open-collector driver.

/INT *Interrupt Request* (output, active Low). The IUSC drives this line Low when (1) its IEI pin is High, (2) one or more of its interrupt condition(s) is (are) enabled and pending, and (3) the Under Service flag is not set for its highest priority enabled/pending condition, nor for any higher-priority internal condition. Software can program whether the bus interface drives this pin in a totem-pole or an open-drain fashion.

/INTACK *Interrupt Acknowledge* (input, active Low). A Low on this line indicates that the host processor is performing an interrupt acknowledge cycle. In some systems, a Low on this line may further indicate that external logic has selected this IUSC as the device to be acknowledged, or as a potential device to be acknowledged. A field in the Bus Configuration Register selects whether this line carries a level-sensitive "status" signal that the IUSC should sample at the leading edge of /AS or /DS, or a single-pulse or double-pulse protocol. The IUSC responds to an interrupt acknowledge cycle in a variety of ways depending on this programming and the state of the /INT and IEI lines, as described in the text.

IEI *Interrupt Enable In* (input, active High). This signal and the IEO pin can be part of an interrupt-acknowledge daisy chain with other devices that may request interrupts. If IEI is High outside of an interrupt acknowledge cycle, one or more IUSC interrupt condition(s) is (are) enabled and pending, and the Under Service flag isn't set for the highest priority condition nor for any higher-priority one, then the IUSC requests an interrupt by driving its /INT pin Low. If the IEI pin is High during an interrupt acknowledge cycle, one or more IUSC interrupt condition(s) is (are) enabled and pending, and the Under Service flag isn't set for the highest priority condition nor for any higher-priority, then the IUSC keeps IEO Low and responds to the cycle.

IEO *Interrupt Enable Out* (output, active High). This signal and/or IEI can be part of an interrupt acknowledge daisy chain with other devices that may request interrupts. The IUSC drives its IEO pin Low whenever its IEI pin is Low, and/or if the Under Service flag is set for any condition. This IUSC drives this signal slightly differently during an interrupt acknowledge cycle, in that it also forces IEO Low if it is (has been) requesting an interrupt.

PIN DESCRIPTION (Continued)

/BUSREQ *Bus Request* (output, active Low). The DMA controller section drives this line Low to request control of the host bus. /BUSREQ can be an open-drain or totem-pole output depending on a bit in the Bus Configuration Register. In open-drain mode the IUSC samples the pin as an input and only drives it Low after sampling it high.

/BIN *Bus Acknowledge In* (input, active Low). When the IUSC receives a falling edge on this input, it samples whether it has been driving (or has just begun to drive) /BUSREQ. If so, it keeps /BOUT High and takes control of the host bus. If not, it passes the bus grant by driving /BOUT Low. This signal can be used with /BOUT to form a bus-grant daisy chain for arbitration of bus control. Alternatively, it can be connected to a direct, positive grant from an external arbiter, and the /BOUT pin can be left unconnected.

/BOUT *Bus Acknowledge Out* (output, active Low). As noted above, this signal can be used with /BIN to form a bus-grant daisy chain for arbitration of bus control.

/ABORT *Abort Master Cycle* (input, active Low). A Low on this line during a master cycle makes the currently active DMA channel terminate its activity and enter a disabled state. Note that /ABORT is only effective during a DMA cycle, so that the IUSC knows which channel should be aborted. Also note that external logic must set /WAIT//RDY to the right state for the cycle to complete, before /ABORT becomes effective.

RxD *Received Data* (input, positive logic). The serial input.

TxD *Transmit Data* (output, positive logic). The serial output.

/RxC *Receive Clock* (input or output). This signal can be used as a clock input for any of the functional blocks in the serial controller. Or, software can program the IUSC so that this pin is an output carrying any of several receiver or internal clock signals, a general-purpose input or output, or an interrupt input.

/TxC *Transmit Clock* (input or output). This signal can be used as a clock input for any of the functional blocks in the serial controller. Or, software can program the IUSC so that this pin is an output carrying any of several transmitter or internal clock signals, a general purpose input or output, or an interrupt input.

/RxREQ *Receive DMA Request* (input or output). In device testing or in applications not using the serial controller and DMA controller sections together in the usual way, this pin can carry an active Low DMA Request from the receive FIFO. On the IUSC this request is internally routed to the on-chip Receive DMA channel; it is more typical to use the RxREQ pin as a general-purpose output or as an interrupt input.

/TxREQ *Transmit DMA Request* (input or output). In device testing or in applications not using the serial controller and DMA controller sections together in the usual way, this pin can carry an active Low DMA Request from the transmit FIFO. On the IUSC this request is internally routed to the on-chip Transmit DMA channel, and it's more typical to use the RxREQ pin as a general-purpose output or as an interrupt input.

/DCD *Data Carrier Detect* (input or output, active Low). Software can program the IUSC so that this signal enables/disables the receiver. In addition, software can program the device to request interrupts in response to transitions on this line. The pin can also be used as a simple input or output.

/CTS *Clear to Send* (input or output, active Low). Software can program the IUSC so that this signal enables/disables the transmitter. In addition, software can program the device to request interrupts in response to transitions on this line. The pin can also be used as a simple input or output.

PORT7/TxCOMPLT *General-Purpose I/O or Transmit Complete* (input or output). Software can program the IUSC so that this pin is a general-purpose input or output, or so that it carries a Transmit Complete signal from the Transmitter, that can control an external driver. The IUSC captures transitions on this pin in internal latches.

PORT6/FSYNC *General-Purpose I/O or Frame Sync* (input or output). Software can program the IUSC so that this pin is a general-purpose input or output, or a Frame Sync input for the IUSC's Time Slot Assigner circuits. The IUSC captures transitions on this pin in internal latches.

PORT5/RxSYNC *General-Purpose I/O or Receive Sync* (input or output). Software can program the IUSC so that this pin is a general-purpose input or output, or so that it carries a Receive Sync output from the Receiver. The IUSC captures transitions on this pin in internal latches.

PORT4/TxTSA *General-Purpose I/O or Transmit Time Slot Assigner Gate* (input or output). Software can program the IUSC so that this pin is a general-purpose input or output, or so that it carries the Gate output of the Transmit Time Slot Assigner, that can enable an external TxD driver in time-slotted ISDN or Fractional T1 applications. The IUSC captures transitions on this pin in internal latches, as described in the text.

PORT 3/RxTSA *General-Purpose I/O or Receive Time Slot Assigner Gate* (input or output). Software can program the IUSC so that this pin is a general-purpose input or output, or so that it carries the Gate output of the Receive Time Slot Assigner. The IUSC captures transitions on this pin in internal latches.

PORT 2 *General-Purpose I/O* (input or output). Software can program the IUSC so that this pin is a general-purpose input or output. The IUSC captures transitions on this pin in internal latches.

PORT 1-0/CLK 1-0 *General-Purpose I/Os or Reference Clocks* (inputs or outputs). Software can program the IUSC so that either of these pins is a general-purpose input or output, or a reference clock that can be divided down to derive clocking for the Receiver and/or Transmitter. When one of these pins is a general-purpose I/O, the IUSC captures transitions on it in internal latches.

V_{CC}, V_{SS} *Power and Ground.* The inclusion of seven pins for each power rail ensures good signal integrity, prevents transients on outputs, and improves noise margins on inputs. The IUSC's internal power distribution network requires that all these pins be connected appropriately.

ARCHITECTURE

The IUSC integrates a fast and efficient dual-channel DMA with a highly versatile serial communications controller. The functional capabilities of the IUSC are described from two different points of view; as a datacommunications device, it transmits and receives data in a wide variety of datacommunications protocols; as a microprocessor peripheral with two DMA channels that offer such features as four DMA transfer types, a flexible bus interface, and vectored interrupts. The architecture is described in three sections, DMA and Bus Interface Capabilities, Communication between the DMA and Serial Channel, and Serial Communication Capabilities. The structure of the IUSC is shown in Figure 1.

DMA AND BUS INTERFACE CAPABILITIES

The IUSC's two versatile DMA channels combined with a flexible bus interface gives it the ability to meet a wide variety of application requirements. The time required to move data into and out of the transmitter and receiver is minimized by the IUSC's speed (20 MHz clock, three clock cycles per word, typical); two buffer-chaining modes with linked-frame status transfer; early buffer termination to keep received frames in separate memory buffers; and vectored interrupts. Some of the many features are briefly described below. See the IUSC Technical Manual for details.

DMA Modes

The IUSC contains two DMA channels, one for the transmitter and one for the receiver. Each channel supports a 32-bit address and a 16-bit byte count. The channels operate in one of four modes. In normal mode, the processor must reload the address and length at the end of each buffer. In Pipelined mode, the processor can load the address and length of the next buffer at any time during the DMA transfer to the first buffer. In Array-Chained mode the processor creates a table of address/length pairs in memory for automatic transfer by the channel. In Linked List mode the processor creates a linked list of address and length pairs in memory to be automatically transferred by the channel.

Single Buffer Mode is the most basic of the four data transfer types. The starting address of each memory buffer and the maximum number of characters to be transferred to or from memory are programmed into the IUSC registers. When the DMA is enabled, it transfers all data between system memory and the transmit and receive FIFOs.

Pipelined Mode is similar to Single Buffer Mode with the addition of an extra set of registers into which the processor can load to reload the DMA with the address and count of the next memory buffer. Therefore, when a buffer is complete, the IUSC is pre-programmed with the address and count of the next buffer so the DMA need not stop between each buffer as long as software stays one step ahead of memory buffer usage.

In Array Mode, one of the two chaining modes, software sets up a table of memory buffer information. The length of the array is only limited by the amount of system memory available for buffers. The IUSC is programmed with the location of the array of buffer addresses and sizes. This mode has the advantage that a burst of short frames is less likely to overrun the systems ability to keep up. The use of receive status block and transmit control block along with the early buffer termination feature simplifies the segmentation and re-assembly of serial messages in memory buffers. When a DMA channel fetches a buffer count of zero, it stops and can create an End-Of-Array interrupt.

Linked List Mode is the most versatile of DMA modes. It has the Array Mode's ability to switch buffers rapidly without the requirement for the buffer information to be in a continuous table. Each link entry contains: The starting address to write or read the data; the size of the buffer; optional status or control information; and a pointer to the next link. Memory buffers can easily be added and removed from the list by changing the links in list entries.

DMA Features

In Linked List Mode, the IUSC has a programmable feature to facilitate the use of buffers in a ring. When this feature is enabled, the DMA writes a zero back to the buffer length field of each array or list entry after it is read. Therefore, if a linked list wraps around on itself, a DMA channel will not reuse a buffer until software has processed the buffer, and indicated that its eligible for reuse by writing a non-zero value in the count field (fetching a count value of zero deactivates the DMA channel). This feature can also be used in array mode to track buffer use.

In both Bus Slave and Master Modes, the IUSC can read and write data words in either byte order. It supports the Little Endian convention used by many Intel microprocessors and the Big Endian convention used by many Motorola microprocessors. When the IUSC is bus master, it can be programmed to only generate the upper 16-bit address when required and, consequently, save a clock cycle on each transfer (three clocks per transfer instead of four). When using the IUSC on a 16-bit bus and the starting address of the message is on an odd address, the IUSC automatically re-orients itself onto even word boundaries by first fetching a byte. This is especially valuable when retransmitting a frame with a different size header than was received. Two pins are available as status signals of the type of transfer in progress.

There are a variety of command and status registers to control and monitor the DMA channels. A DMA channel can be aborted with either the /ABORT pin or by software command. A pause command is also available to temporarily suspend transfers.

Bus Interface & Utilization

The bus interface module stands between the external bus pins and an on-chip 16-bit data bus that interconnects the other functional modules. It includes several flexible bus interfacing options that are controlled by the contents of the Bus Configuration Register (BCR). The BCR is automatically the destination of the first write to the IUSC by the host processor after a reset.

The IUSC is compatible with both multiplexed and non-multiplexed bus interfaces and can transfer either 8 or 16 bits. It supports data transfers with /RD and /WR or R//W and /DS strobe pins and either format of byte ordering. The IUSC generates the Wait or Ready acknowledge handshaking used by Intel or Motorola microprocessors. Also, three styles of interrupt acknowledge signals are supported for automated return of an interrupt vector to any common microprocessor.

There are several options that control how the IUSC uses the bus. The /BIN and /BOUT pins are available to form a bus-grant daisy chain. The IUSC has several options on how it arbitrates requests for bus mastership between channels and how long it stays off the bus between requests. The priority of the two DMA channels is programmable and can alternate between requests to allow both channels equal access to the bus. Once one of the channels has mastership of the bus, control can be passed to the other channel if it is requesting or the IUSC can be forced off the bus. A programmable preempt feature selects whether the higher priority channel can take over control of the bus if it starts requesting control while the lower priority channel is using the bus.

The IUSC maximizes the use of its 32-byte FIFOs by holding /BUSREQ active until the transmit FIFO is full, the receive FIFO is empty, or both. The programmable dwell timers can be used to limit how long the IUSC holds bus mastership by counting either bus transfers, clock cycles or both. Therefore, the combination of programmable FIFO request levels, channel arbitration options, and programmable dwell timer features provide application software the flexibility to optimize the IUSCs bus occupancy to meet system throughput and bus response requirements.

Interrupts

The interrupt subsystem of the IUSC derives from Zilog's experience in providing the most advanced interrupt capabilities in the microprocessor field. These capabilities are at their best when used with a Zilog microprocessor, but it is easy to interface the IUSC to work well with other microprocessors as well. Four pins are dedicated to create an interrupt daisy-chain hierarchy within the Serial Channel and between the Serial Channel and the DMA.

When an IUSC responds to an interrupt acknowledge from the CPU, it places an interrupt vector on the data bus. To speed interrupt response time, the IUSC modifies three bits in the vector to indicate which type of interrupt is being requested. Separate vectors are provided for the serial channel and DMA to easily discriminate the interrupt source.

The DMA has four interrupt sources each for the receive and transmit channels. Each interrupt source is independently enabled and there is a master enable for all DMA interrupts. The four interrupt sources are End Of Array/ End of Link, End Of Buffer, Hardware Abort, and Software Abort.

Each of the six types of interrupts in the serial portion IUSC (Receive Status, Receive Data, Transmit Status, Transmit Data, I/O Status and Device Status) has three bits associated with it: Interrupt Pending (IP), Interrupt-Under-Service (IUS) and Interrupt Enable (IE). If the IE bit for a given source is set, then that bit can source request interrupts. Note that individual sources within the six types also have their own interrupt arm bits. Finally, there is a Master Interrupt Enable (MIE) bit which globally enables or disables all interrupts from the serial channel.

The Interrupt (/INT), Interrupt Acknowledge (/INTACK), Interrupt Enable In (IEI) and Interrupt Enable Out (IEO) pins are provided to create an automated mechanism to place the vector on the bus among the highest priority pending interrupts from multiple devices. The device with the highest pending interrupt (/INT Low, IEI High) places a vector on the bus in response to an interrupt acknowledge cycle.

In the IUSC, the IP bit signals that an interrupt is pending. If an IUS bit is set, this interrupt is being serviced and all interrupt sources of lower priority are prevented from requesting interrupts. An IUS bit is set during an interrupt acknowledge cycle if there are no higher priority devices requesting interrupts.

DMA AND BUS INTERFACE CAPABILITIES (Continued)

There are six sources of Receive Status interrupt. Each one is individually armed: Receiver exited hunt, received idle line, received break/abort, received code violation/end-of-transmission/end-of-message, parity error/abort and overrun error. The Receive Data interrupt is generated whenever the receive FIFO fills with data beyond the level programmed in the Receive Interrupt Control Register (RICR). There are six sources of Transmit Status interrupt. Each one is individually armed: Preamble sent, idle line sent, abort sent, end-of-frame/end-of-message sent, CRC sent and underrun error. The Transmit Data interrupt is generated whenever the transmit FIFO empties below the level programmed in the Transmit Interrupt Control Register (TICR).

The I/O Status interrupt serves to report transitions on any of six pins. Interrupts are generated on either or both edges with individual edge selection and arming for each pin. The pins that can be programmed to generate I/O Status interrupts are /RxC, /TxC, /RxREQ, /TxREQ, /DCD and /CTS. These interrupts are independent of the programmed function of the pins.

The Device Status interrupt has four individually enabled sources: Receive character counter underflow, DPLL sync acquired, BRG1 zero count and BRGO zero count. Refer to the IUSC Technical Manual for more details.

COMMUNICATION BETWEEN THE DMA AND SERIAL CHANNELS

The IUSC's intra-chip communication between the DMA and serial communications controller gives it the power to achieve higher efficiency than is possible with a separate DMA controller. The Linked Frame Status Transfer feature writes the status and byte count of each received frame to memory as part of an array or linked list. This provides a simple and easy to use mechanism for storing the results of a received message without arbitrary restrictions on how quickly the host software must examine the results. Similarly, control information for transmit frames can be automatically read by the DMA from the array or link and transferred into registers in the serial logic.

In all modes, the DMA can accept a signal from the serial channel for early buffer termination. When the end of a message is received, the data is transferred to the buffer and the status is written to memory. The status is written after the data in single buffer and pipelined modes or to the array/link in array and linked-list modes if Linked-Frame Status Transfer is enabled. This early buffer termination is treated identically to the terminal count condition in the DMA. Therefore, the receipt of the end of a message is a seamless transition from one memory buffer to the next.

An example of using these inter-communication features using linked list mode is shown in Figure 4. This example shows the format of a ring of memory buffers with the linked frame status transfer and ring buffer features enabled. Any protocol that sets the "RxBound" bit (RCSR4 = 1), like HDLC or 802.3, is appropriate to this example. The linked list is shown in Figure 4 with three links for simplicity and may be as large as memory allows. The sixth word in each list entry is reserved and should not be used (it keeps the list entries on 32-bit boundaries). If the end of the buffer is reached and it is not the end of the frame, the IUSC writes zeros as the status and count. Also, if the transmit channel needs to start a new memory buffer other than at the beginning of a frame, the DMA ignores the transmit control block.

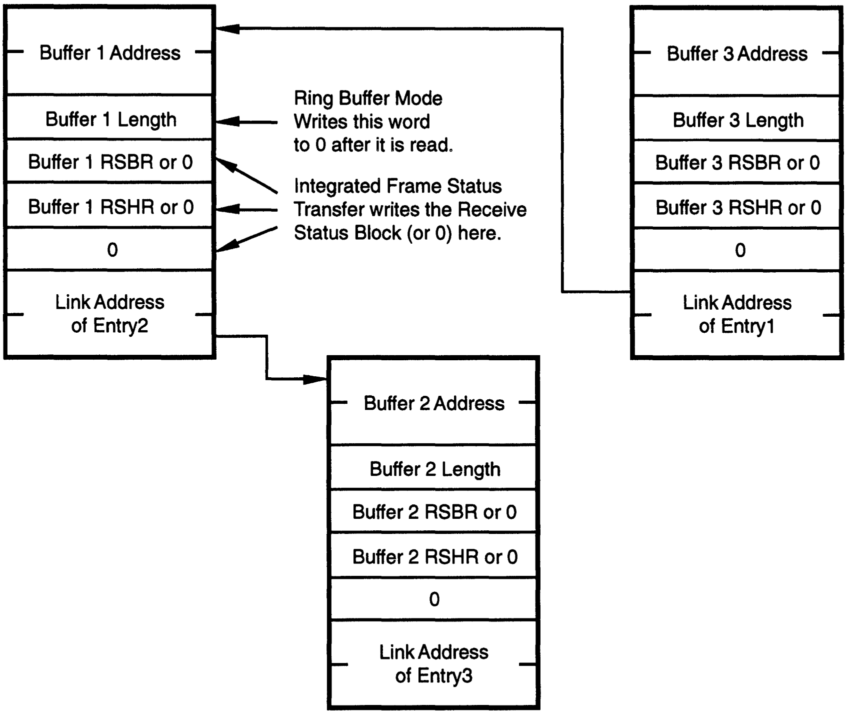


Figure 4. Linked List Mode with Linked Frame Status Transfer and Ring Buffer Features

Another method by which the DMA and serial channel work together is using the Transmit Character Counter to break a large block of data into a number of fixed length frames. For example, it is desired to transmit a large file which is located in several memory buffers as fixed length smaller frames. With the IUSC, the serial channel is programmed to send the end-of-frame sequence each time the set number of bytes is transmitted. Therefore, DMA transfers are not interrupted, nor is system response required to break the large file into frames.

The IUSC provides higher throughput than discrete serial and DMA chip solutions because discrete chips do not directly communicate with each other and, therefore, the status of one device must be read by the CPU and communicated to the other. This typically requires interrupts and the suspension of activity until status/control information is updated. This uses precious time and bus bandwidth, which can limit total throughput.

DATA COMMUNICATIONS CAPABILITIES

The IUSC provides a full-duplex channel programmable for use in any common data communication protocol. The receiver and transmitter are completely independent and each is supported by a 32-byte deep FIFO and a 16-bit frame length counter. All modes allow optional even, odd, mark or space parity. Synchronous modes allow the choice of either of two 16-bit or a 32-bit CRC polynomials. Character length of up to 8 bits can be programmed for the receiver and transmitter independently. Error and status conditions are carried with the data in the receive FIFO to greatly reduce the CPU overhead required to send or receive a message, while key control parameters accompany transmit characters through the Tx FIFO. Interrupts can be individually armed to signal such conditions as overrun, parity error, framing error, end-of-frame, idle line received, sync acquired, transmit underrun, CRC sent, closing sync/flag sent, abort sent, idle line sent and preamble sent. In addition, several useful internal signals like receive character boundary, received sync, transmit character boundary and transmission complete can be sent to pins for use by external circuitry.

Protocols

Asynchronous Mode. The receiver and transmitter handle data at a rate of 1/16, 1/32, or 1/64 the clock rate. The receiver rejects start bits less than one-half a bit time and includes recovery logic following a framing error. The transmitter is capable of sending one, two, or anywhere in the range of 9/16th to two stop bits per character in 1/16 bit increments.

Nine-Bit Mode. This mode is identical to async except that the receiver checks for the status of an additional address/data bit between the parity bit and the stop bit. The value of this bit is FIFO'ed along with the data. In the transmitter, this bit is automatically inserted with the value that is FIFO'ed from the transmit data.

Isochronous Mode. Both transmitter and receiver operate on start-stop (async) data using a 1x clock. The transmitter sends one or two stop bits.

Asynchronous With Code Violations. This is similar to Isochronous mode except that the start bit is replaced by a three bit-time code violation pattern as in MIL-STD-1553B. The transmitter sends zero, one or two stop bits.

HDLC Mode. In this mode, the receiver recognizes flags, performs optional address matching, accommodates extended address fields, and performs zero deletion and CRC checking. The receiver is capable of receiving shared-zero flags, recognizes abort sequences and can receive arbitrary length frames. The transmitter automatically sends opening and closing flags, performs zero insertion and can be programmed to send an abort, an extended abort, a flag or CRC and a flag on transmit underrun. The transmitter automatically sends a closing flag with optional CRC at the end of a programmed message length. Shared-zero flags are selected in the transmitter and a separate character length is programmed for the last character in the frame.

Frames terminated with an ABORT can be marked with a status bit on the preceding character in addition to the status interrupt that can be enabled. Abort is only detected in-frame and, therefore, eliminates false detection due to an idle line. The IUSC provides four choices (flag, all 1s, all 0s, or alternating 1s and 0s) of line preamble to condition the line before beginning data transmission. This feature is valuable to get the receiver DPLL in sync and as a flow control mechanism to slow down frame transmission without slowing down the clock or disabling the transmitter.

HDLC Loop Mode. This mode is available only in the transmitter and allows the IUSC to be used in an HDLC Loop configuration. In this mode, the receiver is programmed to operate in HDLC mode to allow the transmitter to echo received messages. Upon receipt of a particular bit pattern (actually a sequence of seven consecutive ones) the transmitter stops repeating data and inserts its own frame(s).

802.3 Mode. This mode implements the data format of IEEE 802.3 with a 16-bit address compare. In this mode, /DCD and /CTS are used to implement the carrier sense and collision detect interactions with the receiver and transmitter. Back-off timing must be provided externally.

Monosync Mode. In this mode, a single character is used for synchronization. The sync character can be either eight bits long or the same length as the data characters. The receiver can automatically strip sync characters from the received data stream. The transmitter is programmed to automatically send CRC on either an underrun or at the end of a programmed message length.

Slaved Monosync Mode. This mode is available only in the transmitter and allows the transmitter (operating just as though it were in monosync mode) to send data with its byte boundaries synchronized to those of the received data.

Bisync Mode. This mode is identical to monosync mode except that character synchronization requires two successive characters. The two characters need not be identical.

Transparent Bisync Mode. In this mode, the synchronization pattern is DLE-SYN, programmable selected from either ASCII or EBCDIC encoding. The receiver recognizes control character sequences and automatically handles CRC calculations without CPU intervention. The transmitter is programmed to send either SYN, DLE-SYN, CRC-SYN, or CRC-DLE-SYN upon underrun and automatically sends the closing DLE-SYN with optional CRC at the end of a programmed message length.

External Sync Mode. The receiver is synchronized to the receive data by an externally-supplied signal on a pin for custom protocol applications.

Data Encoding

The IUSC is programmed to encode and decode the serial data in any of eight different ways (Figure 5). The transmitter encoding method is selected independently of the receiver decoding method.

NRZ. In NRZ, a 1 is represented by a High level for the duration of the bit cell and a 0 is represented by a Low level for the duration of the bit cell.

NRZB. NRZB is inverted from NRZ.

NRZI-Mark. In NRZI-Mark, a 1 is represented by a transition at the beginning of a bit cell, i.e., the level present in the preceding bit cell is reversed. A 0 is represented by the absence of a transition at the beginning of the bit cell.

NRZI-Space. In NRZI-Space, a 1 is represented by the absence of a transition at the beginning of a bit cell, i.e., the level present in the preceding bit cell is maintained. A 0 is represented by a transition at the beginning of the bit cell.

Biphase-Mark. In Biphase-Mark, a 1 is represented by a transition at the beginning of the bit cell and another transition at the center of the bit cell. A 0 is represented by a transition at the beginning of the bit cell only.

Biphase-Space. In Biphase-Space, a 1 is represented by a transition at the beginning of the bit cell only. A 0 is represented by a transition at the beginning of the bit cell and another transition at the center of the bit cell.

Biphase-Level. In Biphase-Level, a 1 is represented by a High during the first half of the bit cell and a Low during the second half of the bit cell. A 0 is represented by a Low during the first half of the bit cell and a High during the second half of the bit cell.

Differential Biphase-Level. In Differential Biphase-Level, a 1 is represented by a transition at the center of the bit cell, with the opposite polarity from the transition at the center of the preceding bit cell. A 0 is represented by a transition at the center of the bit cell with the same polarity as the transition at the center of the preceding bit cell. In both cases, there are transitions at the beginning of the bit cell to set up the level required to make the correct center transition.

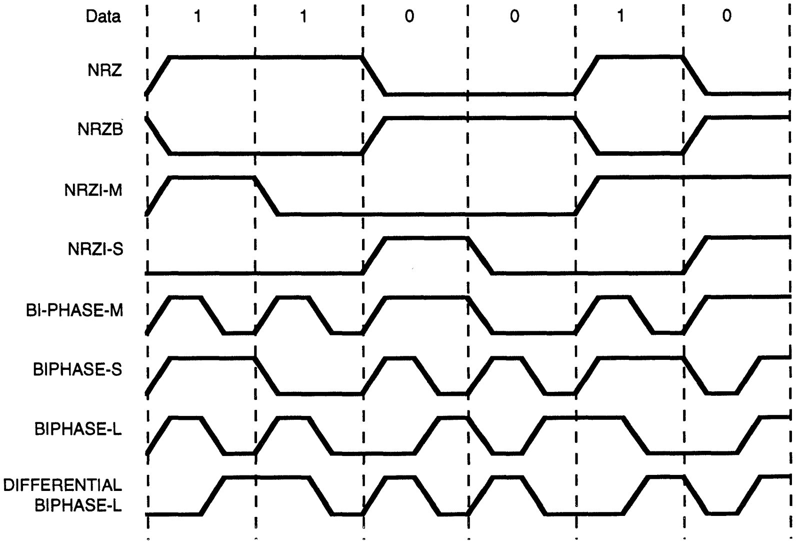


Figure 5. Data Encoding

DATA COMMUNICATIONS CAPABILITIES (Continued)

Character Counters

The IUSC contains separate 16-bit character counters for the receiver and transmitter. The receive character counter is set to a programmable starting value or automatically at the beginning of each received frame and can be reloaded under software control during a frame. The counter decrements with each receive character. At the end of the receive message the current value in the counter is automatically loaded into a four-deep FIFO. With the Receive Status Block (RSB) feature enabled, the counter value and the status (RCSR) can be automatically transferred to memory following the data. In array and linked list modes, the RSB can be transferred to the array or list entry for easy software access. This allows DMA transfer of data to proceed without CPU intervention at the end of a received frame, as the values in the FIFO allow the CPU to determine the status and length of each frame.

Similarly, the transmit character counter is loaded automatically at the beginning of each transmit frame and can be reloaded under software control during a frame. The counter is decremented with each write to the transmit FIFO. When the counter reaches zero, and that byte is sent, the transmitter automatically terminates the message in the appropriate fashion (usually by sending the CRC and the closing flag or sync character) without requiring CPU intervention. In linked list and array modes, the transmit character count and frame control word can be fetched from the linked list or array.

Baud Rate Generators

The IUSC contains two Baud Rate Generators. Each generator consists of a 16-bit time constant register and a 16-bit down counter. In operation, the counter decrements with each cycle of its selected input clock, and the time constant can be automatically reloaded when the count reaches zero. The output of the Baud Rate Generator toggles when the counter reaches a count of one-half of the time constant and again when the counter reaches zero. A new time constant can be written at any time but the new value does not take effect until the next load of the counter. The outputs of both baud rate generators are sent to the clock multiplexer for use internally or externally. The input to the Baud Rate Generator can be the /TxC pin, the /RxC pin, a PORT pin, or the output of either counter. The baud rate generator output frequency is related to the baud rate generator input clock frequency by the following formula:

Output frequency = Input frequency/time constant + 1.

Note: This allows an output frequency in the range of 1 to 1/65536 of the input frequency, inclusive.

The output of either Baud Rate Generator can be used as the transmit or receive clock, the reference clock input to the DPLL circuit, and/or can be output on the /RxC or /TxC pin.

Digital Phase-Locked Loop

The IUSC contains a DPLL (Digital Phase-Locked Loop) to recover clock information from a data stream with NRZI or Biphase encoding. The DPLL is driven by a clock that is nominally 8, 16 or 32 times the receive data rate. The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock can be routed to the receiver, transmitter, or both, or to a pin for use externally. In all modes, the DPLL counts the input clock to create nominal bit times. While counting, the DPLL watches the incoming data stream for transitions. When a transition is detected, the DPLL may make a count adjustment (during the next counting cycle) to produce an output clock which tracks the incoming bit cells. The DPLL provides properly phased transmit and receive clocks to the clock multiplexer.

Counters

The IUSC contains two 5-bit counters, which are programmed to divide an input clock by 4, 8, 16 or 32. The outputs of these two counters are sent to the clock multiplexer. The counters can be used as prescalers for the Baud Rate Generators. They also provide a stable transmit clock from a common source when the DPLL is providing the receive clock. The PORT0 and PORT1 pins can be used as inputs to the counters.

Clock Multiplexers

The clock multiplexer logic selects the receive and transmit clocks and optional outputs on the /RxC and/or /TxC pin(s). In the Z16C32, the PORT0 and PORT1 pins can be used directly as receive and transmit clocks, as well as being used as inputs to the counters.

Time Slot Assigner

The IUSC is equipped with two Time Slot Assigners to support ISDN and Fractional T1 communications. There is one assigner for the receiver. Each time slot assigner selects one or more time slots within a frame, however, the selected time slots must be contiguous. The first selected time slot is programmable from slot 0 (the first slot) to slot 127 of the frame. The number of concatenated slots is programmable from 1 to 15 (total slots). The time of the first slot can be offset an integral number of clocks. This offset is a delay and is programmable from 0 (no offset) to 7 clocks in increments of one clock (one bit cell). This offset can be used to compensate for delays in frame sync detection logic.

Test Modes

The IUSC can be programmed for local loopback or auto echo operation. In local loopback, the output of the transmitter is internally routed to the input of the receiver. This allows testing of the IUSC data paths without any external logic. Auto echo connects the RxD pin directly to the TxD pin. This is useful for testing serial links external to the IUSC.

I/O Port

The Port pins are general-purpose I/O pins. They are used as additional modem control lines or other I/O functions. Each port bit is individually programmable as general-purpose input, as an output, or for a dedicated input or output function. This programming is done in the Port Control Register. Whether used as inputs or outputs, the port pins can be read at any time.

The dedicated functions of the port pins include Time Slot Assigner gate outputs, transmit complete output, clock inputs, receive sync output, or frame sync input.

The port pins capture edge transitions. Programming for the capture is done using the Port Latched/Unlatch command bits in the Port Status Register. Each port bit is individually controlled. The Latched/Unlatch bit is used as a status signal to indicate that a transition has occurred on the port pin and as a command to open the latches that capture this transition. Both rising edge and falling edge are detected. When a transition is detected, the latch closes, holding the post transition state of the input.

The Latched/Unlatch bit is held at 0 if no transitions occur on the port pin; this bit is set to a 1 when a rising edge or falling edge transition is detected, or immediately after the latch is opened if one or more transitions occurred while the latch was closed. Writing a 0 to the Latched/Unlatch bit has no effect on the latch. Writing a 1 to this bit resets the status bit and opens the latch. To use the port as an input without edge detection, a 1 would be written to the Latched/Unlatch bit to open the latch and then the Port Status Register would be read to obtain the current pin input status.

PROGRAMMING

An Electronic Programmer's Manual (MS DOS based) and a Technical Manual are available to provide details about programming the IUSC. Also included are explanations and features of all registers in the IUSC.

The registers in the IUSC are programmed by the system to configure the channel. Before this can occur, the system must set up the bus interface by writing to the Bus Configuration Register (BCR). The BCR has no specific address and is only accessible after a hardware reset of the device. The first write to the IUSC, after a hardware reset, programs the BCR. From that time on other channel registers can be accessed. No specific address need be presented to the IUSC for the BCR write; the IUSC knows that the first write after a hardware reset is destined for the BCR.

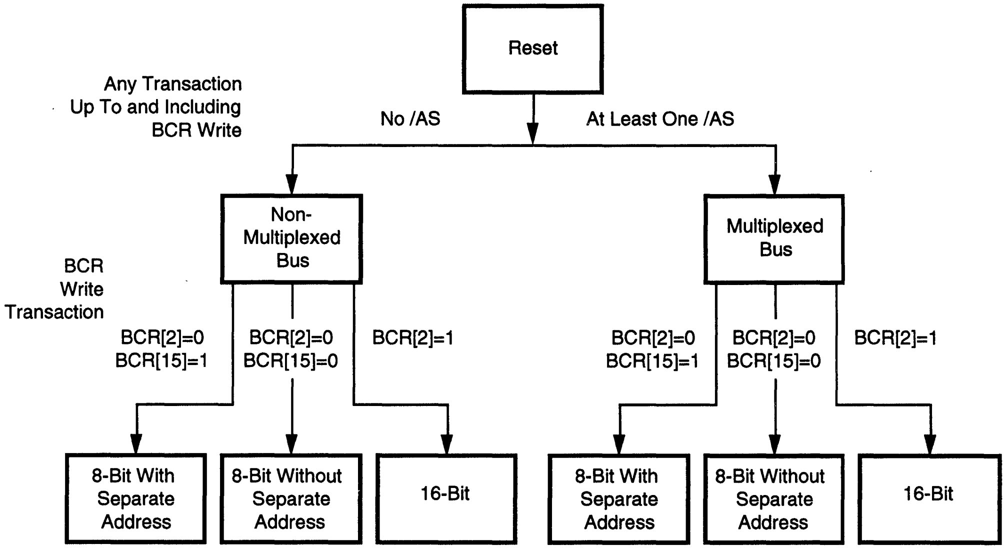
In the multiplexed bus case, all registers are directly addressable through the address latched by /AS at the beginning of each bus cycle. The D//C pin is still used to directly access the receive and send data registers (RDR and TDR) with a multiplexed bus; if D//C is High, the address latched by /AS is ignored and an access of RDR or TDR is performed.

In the non-multiplexed bus case, the channel registers are accessed indirectly using the address pointer in the Channel Command/Address Register (CCAR). The address of the desired register is first written to the CCAR and then the selected register is accessed; the pointer in the CCAR is automatically cleared after this access.

Two more points about the IUSC should be noted here. Channel Reset bit in the CCAR places the channel in the reset state. To exit this reset state either a word of all zeros is written to the CCAR (16-bit bus) or a byte of all zeros is written to the lower byte of the CCAR (8-bit bus). Secondly, after reset, the transmit and receive clocks are disabled. The first thing that should be done in any initialization sequence is a write to the Clock Mode Control Register (CMCR) to select a clock source for the receiver and transmitter.

The Serial/DMA (S//D) pin is used to differentiate between the serial channel and the DMA registers. The DMA registers fall into three logic groupings; common registers that apply to both transmit and receive, transmit registers, and receive registers. The registers for DMA transmit functions and receive functions are symmetric and therefore, a single diagram is shown for each in the following pages. When addressing the DMA registers, the Data/Control (D//C) pin selects between the transmit and receive registers. For example, there is a DMA byte count register for transmit and receive (TBCR and RBCR) at address 10101 with S//D pin Low. The TBCR is selected with the D//C pin Low, and the RBCR is selected with the D//C pin High. The format of these two registers is shown in Figure 20.

The register addressing is shown in Table 2 and the table assumes that the BCR register bit 0 is set to 1. The A5-A1 column in the Table reflects the state of AD5-AD1, AD13-AD9, CCAR5-CCAR1 or DCAR5-DCAR1 as applicable. The bit assignments of the registers are shown in Figures 7 through 80. See the IUSC Technical Manual for details. The register addressing is shown in Table 2 and the bit assignments for the registers are shown in Figure 6.



Note:
The presence of one transaction with an /AS active, between reset up to and including the BCR write, chooses a multiplexed type of bus.

Figure 6. BCR Reset Sequence and Bit Assignments

PROGRAMMING (Continued)

Table 1. Register Address List

S//D	D//C	Address A5-A1	Name	Description
1	0	00000	CCAR	Channel Command/Address Register
1	0	00001	CMR	Channel Mode Register
1	0	00010	CCSR	Channel Command/Status Register
1	0	00011	CCR	Channel Control Register
1	0	00100	PSR	Port Status Register
1	0	00101	PCR	Port Control Register
1	0	00110	TMDR	Test Mode Data Register
1	0	00111	TMCR	Test Mode Control Register
1	0	01000	CMCR	Clock Mode Control Register
1	0	01001	HCR	Hardware Configuration Register
1	0	01010	IVR	Interrupt Vector Register
1	0	01011	IOCR	I/O Control Register
1	0	01100	ICR	Interrupt Control Register
1	0	01101	DCCR	Daisy-Chain Control Register
1	0	01110	MISR	Misc. Interrupt Status Register
1	0	01111	SICR	Status Interrupt Control Register
1	1	XXXXX	RDR	Receive Data Register (Read Only)
1	0	1X000	RDR	Receive Data Register (Read Only)
1	0	10001	RMR	Receive Mode Register
1	0	10010	RCSR	Receive Command/Status Register
1	0	10011	RICR	Receive Interrupt Control Register
1	0	10100	RSR	Receive Sync Register
1	0	10101	RCLR	Receive Count Limit Register
1	0	10110	RCCR	Receive Character Count Register
1	0	10111	TC0R	Time Constant 0 Register
1	1	XXXXX	TDR	Transmit Data Register (Write Only)
1	0	1X000	TDR	Transmit Data Register (Write Only)
1	0	11001	TMR	Transmit Mode Register
1	0	11010	TCSR	Transmit Command/Status Register
1	0	11011	TICR	Transmit Interrupt Control Register
1	0	11100	TSR	Transmit Sync Register
1	0	11101	TCLR	Transmit Count Limit Register
1	0	11110	TCCR	Transmit Character Count Register
1	0	11111	TC1R	Time Constant 1 Register

Table 1. Register Address List (Continued)

S//D	D//C	Address A5-A1	Name	Description
X	0	XXXXX	BCR	Bus Configuration Register
0	X	00000	DCAR	DMA Command/Address Register
0	0	00001	TDCMR	Transmit DMA Channel Mode Register
0	X	00011	DCR	DMA Control Register
0	X	00100	DACR	DMA Array Count Register
0	X	01001	BDCR	Burst Dwell Control Register
0	X	01010	DIVR	DMA Interrupt Vector Register
0	X	01100	DICR	DMA Interrupt Control Register
0	X	01101	CDIR	Clear DMA Interrupt Register
0	X	01110	SDIR	Set DMA Interrupt Register
0	0	01111	TDIAR	Transmit DMA Interrupt Arm
0	0	10101	TBCR	Transmit Byte Count Register
0	0	10110	TARL	Transmit Address Register (Lower)
0	0	10111	TARU	Transmit Address Register (Upper)
0	0	11101	NTBCR	Next Transmit Byte Count Register
0	0	11110	NTARL	Next Transmit Address Register (Lower)
0	0	11111	NTARU	Next Transmit Address Register (Upper)
0	1	00001	RDMR	Receive DMA Mode Register
0	1	01111	RDIAR	Receive DMA Interrupt Arm
0	1	10101	RBCR	Receive DMA Byte Count Register
0	1	10110	RARL	Receive Address Register (Lower)
0	1	10111	RARU	Receive Address Register (Upper)
0	1	11101	NRBCR	Next Receive Byte Count Register
0	1	11110	NRARL	Next Receive Address Register (Lower)
0	1	11111	NRARU	Next Receive Address Register (Upper)

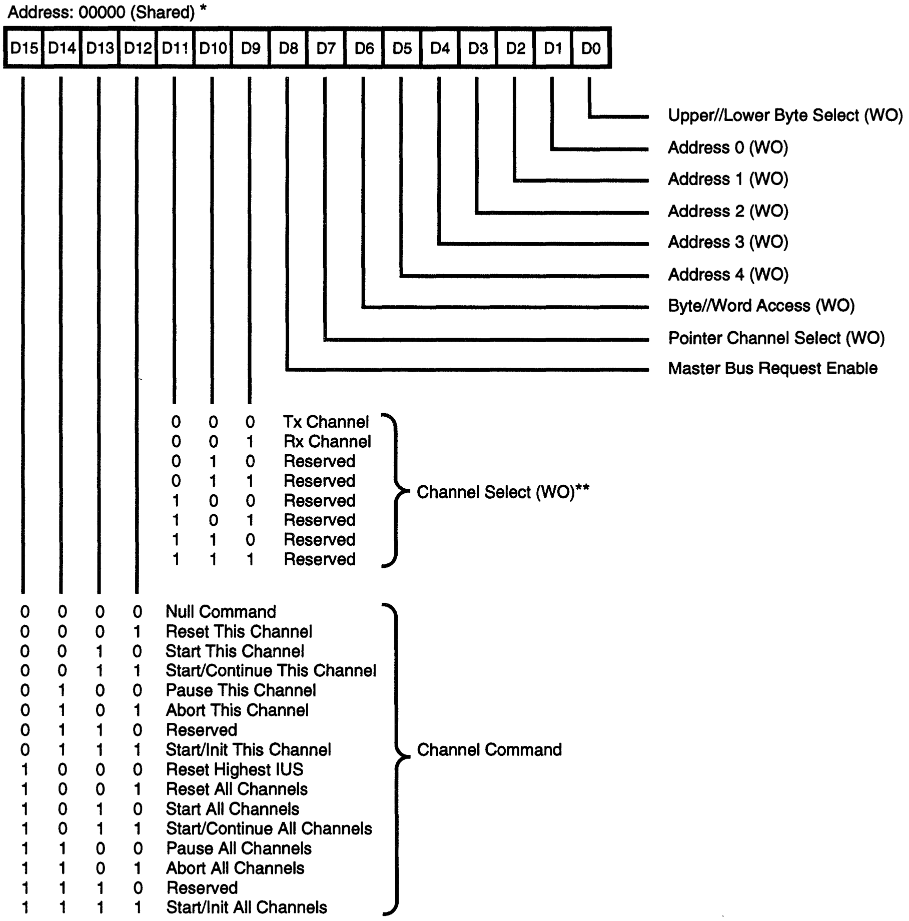
REGISTER DESCRIPTION

This section describes the function of the various bits in the registers of the device. Throughout this section the following conventions are discussed:

Control bits are written and read by the CPU and are not modified by the device. Command bits are written by the CPU to initiate an action in the device and are read as zeros. Status bits are controlled by the device and are read to check device status. Any writes to status bits are ignored by the device. Command/Status bits are controlled by both the device and the CPU. They may be written and read by the CPU and may also be modified by the device.

Reserved bits are not used in this implementation of the device and may or may not be physically present in the device. Any reserved bits that are physically present are readable and writable, but reserved bits that are not present are always read as zeros. To ensure compatibility with future versions of the device, reserved bits should always be written with zeros. Reserved commands should not be used for the same reason.

First, the DMA registers unique to the IUSC are described in the following pages (Figures 7-16) and then the serial channel registers are described (Figures 17-80).



Address: 00000 (Shared) *

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Bit	Function
D0	Upper//Lower Byte Select (WO)
D1	Address 0 (WO)
D2	Address 1 (WO)
D3	Address 2 (WO)
D4	Address 3 (WO)
D5	Address 4 (WO)
D6	Byte//Word Access (WO)
D7	Pointer Channel Select (WO)
D8	Master Bus Request Enable

D11	D10	D9	Channel Select (WO)**
0	0	0	Tx Channel
0	0	1	Rx Channel
0	1	0	Reserved
0	1	1	Reserved
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

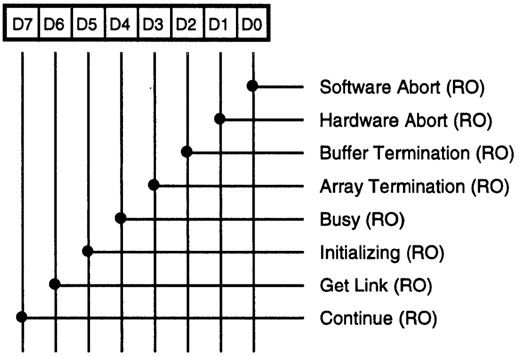
D15	D14	D13	D12	Channel Command
0	0	0	0	Null Command
0	0	0	1	Reset This Channel
0	0	1	0	Start This Channel
0	0	1	1	Start/Continue This Channel
0	1	0	0	Pause This Channel
0	1	0	1	Abort This Channel
0	1	1	0	Reserved
0	1	1	1	Start/Init This Channel
1	0	0	0	Reset Highest IUS
1	0	0	1	Reset All Channels
1	0	1	0	Start All Channels
1	0	1	1	Start/Continue All Channels
1	1	0	0	Pause All Channels
1	1	0	1	Abort All Channels
1	1	1	0	Reserved
1	1	1	1	Start/Init All Channels

Notes:

* (Shared) means, shared between DMA Channels

** (WO) means Write Only

Figure 7. DMA Command/Address Register



0	0	0	0	0	0	0	0	Reset	Command
X	X	X	1	X	X	X	X	Start	
1	X	X	1	X	X	X	X	Start/Continue (Buffered Mode)	
X	X	X	1	X	X	X	X	Start/Continue (Other Than Buffered Mode)	
X	X	X	0	X	X	X	X	Pause	
X	X	X	0	X	X	X	1	Abort	
X	X	1	1	X	X	X	X	Start/Init (Array or Linked Array Modes)	
X	X	X	1	X	X	X	X	Start/Init (Normal or Buffered Modes)	

Notes:
X = Previous Value
(RO) = Read Only

Figure 8. Affect of Commands on Status Bits

CONTROL REGISTERS (Continued)

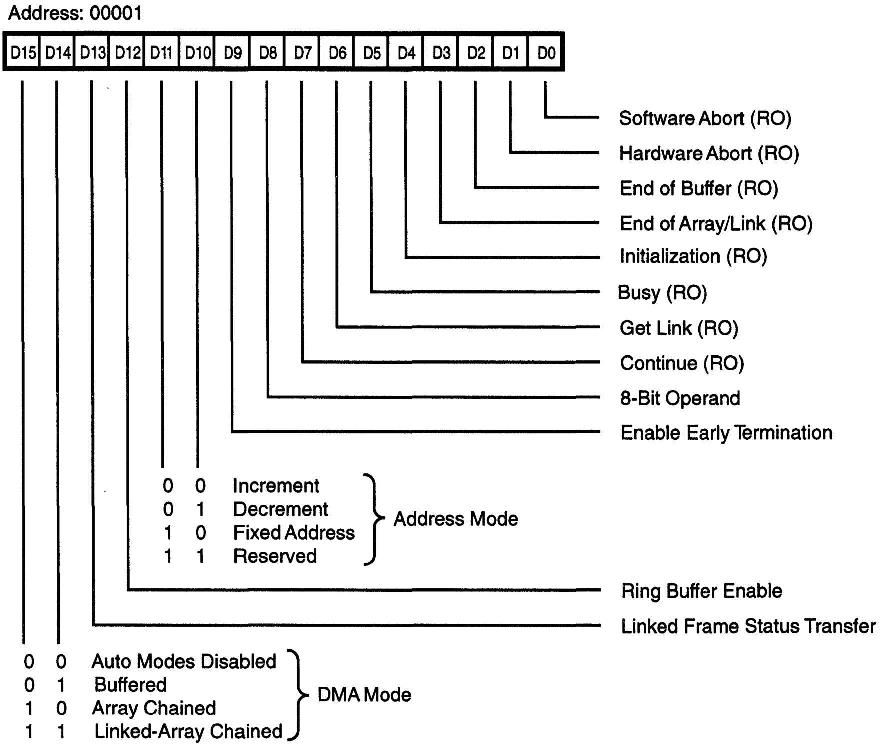
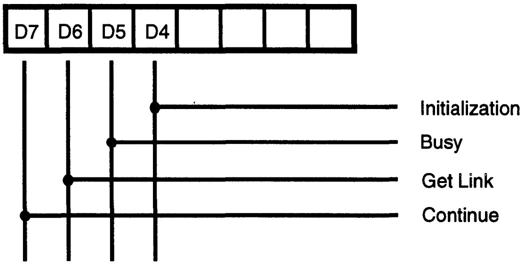


Figure 9. Tx/Rx DMA Mode Register (TDMR) (RDMR)



D7	D6	D5	D4		
0	0	0	0	Channel Disabled	Normal Mode
0	0	1	0	Channel Enabled	
X	X	X	1	Not Possible	
X	1	X	X	Not Possible	
1	X	X	X	Not Possible	
0	0	0	0	Channel Disabled, Base Registers Invalid	Buffered Mode
0	0	1	0	Channel Enabled, Base Registers Invalid	
1	0	0	0	Channel Disabled, Base Registers Valid	
1	0	1	0	Channel Enabled, Base Registers Valid	
X	X	X	1	Not Possible	
X	1	X	X	Not Possible	
0	0	0	0	Channel Disabled, Data Transfer Phase	Array-Chained Mode
0	0	1	0	Channel Enabled, Data Transfer Phase	
0	0	0	1	Channel Disabled, Array Transfer Phase	
0	0	1	1	Channel Enabled, Array Transfer Phase	
X	1	X	X	Not Possible	
1	X	X	X	Not Possible	
0	0	0	0	Channel Disabled, Data Transfer Phase	Linked Array-Chained Mode
0	0	1	0	Channel Enabled, Data Transfer Phase	
0	0	0	1	Channel Disabled, Array Transfer Phase	
0	0	1	1	Channel Enabled, Array Transfer Phase	
0	1	X	0	Not Possible	
0	1	0	1	Channel Disabled, Link Transfer Phase	
0	1	1	1	Channel Enabled, Link Transfer Phase	
1	X	X	X	Not Possible	

Figure 10. Status Bit Combinations

CONTROL REGISTERS (Continued)

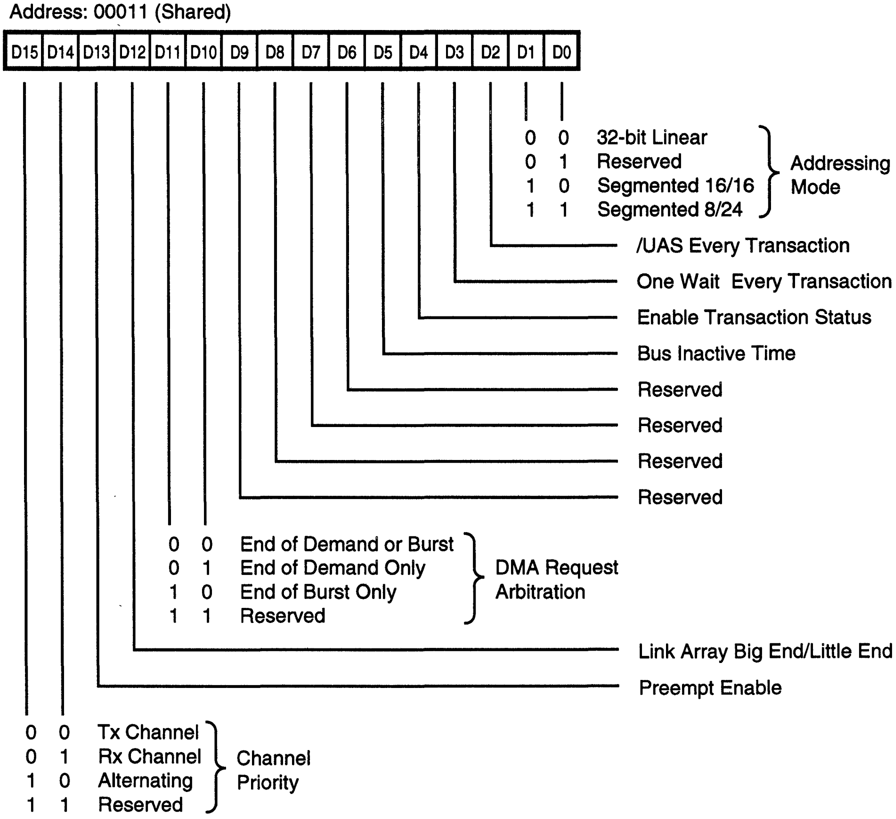


Figure 11. DMA Control Register (DCR)

Big End Array
(16-Bit bus)

	AD15															AD0
Address n	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Address n+2	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00

Little End Array
(16-Bit bus)

	AD15															AD0
Address n	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Address n+2	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Big End Array
(8-Bit bus)

	AD7							AD0
Address n	31	30	29	28	27	26	25	24
Address n+1	23	22	21	20	19	18	17	16
Address n+2	15	14	13	12	11	10	09	08
Address n+3	07	06	05	04	03	02	01	00

Little End Array
(8-Bit bus)

	AD7							AD0
Address n	07	06	05	04	03	02	01	00
Address n+1	15	14	13	12	11	10	09	08
Address n+2	23	22	21	20	19	18	17	16
Address n+3	31	30	29	28	27	26	25	24

Figure 12. Array-Chained Bit Ordering

Note:
Bit 12 in DCR is used to control the byte ordering of addresses and counts stored in memory in the Array and Linked Array Modes. The above figure shows the two cases for both bus bandwidths.

CONTROL REGISTERS (Continued)

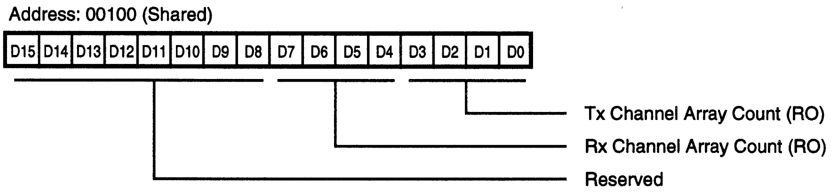

Address: 00100 (Shared)
D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
Tx Channel Array Count (RO)
Rx Channel Array Count (RO)
Reserved

Figure 13a. DMA Array Count Register (DACR)

Channel Array Count 3	Channel Array Count 2	Channel Array Count 1	Channel Array Count 0		
0	0	0	0	Array Operation Not Started	Array-Chained
0	0	0	1	Fetched Last Byte of Array	
0	0	1	0	Fetched Fifth Byte of Array	
0	0	1	1	Fetched Fourth Byte of Array	
0	1	0	0	Fetched Third Byte of Array	
0	1	0	1	Fetched Second Byte of Array	
0	1	1	0	Fetched First Byte of Array	
0	1	1	1	Fetched Last Byte of Buffer	

Getlink (in DCMR)	Channel Array Count 3	Channel Array Count 2	Channel Array Count 1	Channel Array Count 0		
0	0	0	0	0	Array Operation Not Started	Linked Array-Chained
0	0	0	0	1	Fetched Last Byte of Array	
0	0	0	1	0	Fetched Fifth Byte of Array	
0	0	0	1	1	Fetched Fourth Byte of Array	
0	0	1	0	0	Fetched Third Byte of Array	
0	0	1	0	1	Fetched Second Byte of Array	
0	0	1	1	0	Fetched First Byte of Array	
0	0	1	1	1	Fetched Last Byte of Linked Address	
1	0	0	0	0	Invalid	
1	0	0	0	1	Invalid	
1	0	0	1	0	Invalid	
1	0	0	1	1	Invalid	
1	0	1	0	0	Fetched Third Byte of Linked Address	
1	0	1	0	1	Fetched Second Byte of Linked Address	
1	0	1	1	0	Fetched First Byte of Linked Address	
1	0	1	1	1	Fetched Last Byte of Buffer	

Figure 13b. Channel Array Count Bit Combinations

Note: See the Z16C32 Technical Manual for the appropriate table with Linked Status Transfer feature enabled.

Address: 01001 (Shared)

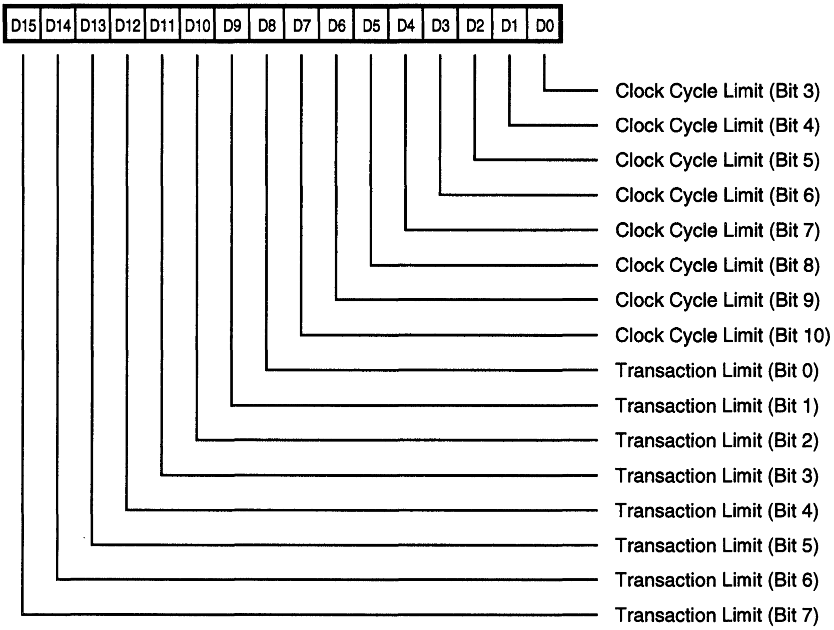

D15
D14
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
Clock Cycle Limit (Bit 3)
Clock Cycle Limit (Bit 4)
Clock Cycle Limit (Bit 5)
Clock Cycle Limit (Bit 6)
Clock Cycle Limit (Bit 7)
Clock Cycle Limit (Bit 8)
Clock Cycle Limit (Bit 9)
Clock Cycle Limit (Bit 10)
Transaction Limit (Bit 0)
Transaction Limit (Bit 1)
Transaction Limit (Bit 2)
Transaction Limit (Bit 3)
Transaction Limit (Bit 4)
Transaction Limit (Bit 5)
Transaction Limit (Bit 6)
Transaction Limit (Bit 7)

Figure 14. Burst Dwell Control Register (BDCR)

Notes:
BDCR Controls the amount of time that DMA may remain bus master.

Bits 15 through 8 are used to select a limit for the number of DMA transfers on the Bus while the DMA is bus master. This limit is a binary number, a value of zero disables the transaction limit function.

Bits 7 through 0 are used to select a limit for the number of clock cycles that the DMA may remain on the bus as bus master.

Bus transaction will always complete, even if the clock cycle limit is exceeded during the bus cycle, and even if the cycle is extended by external hardware signalling through /WAIT//RDY.

CONTROL REGISTERS (Continued)

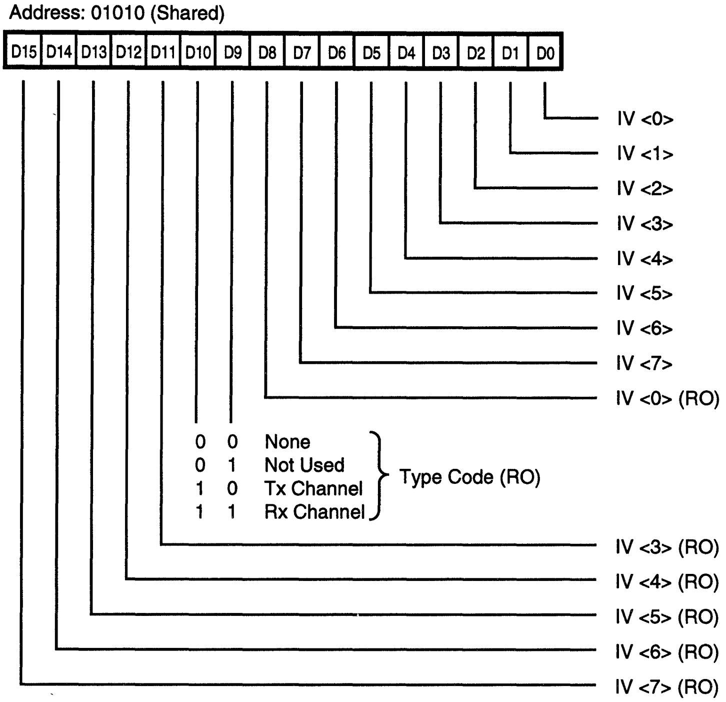


Figure 15. DMA Interrupt Vector Register (DIVR)

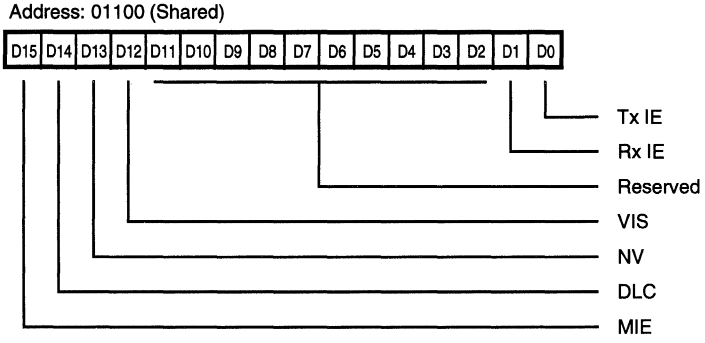


Figure 16. DMA Interrupt Control Register (DICR)

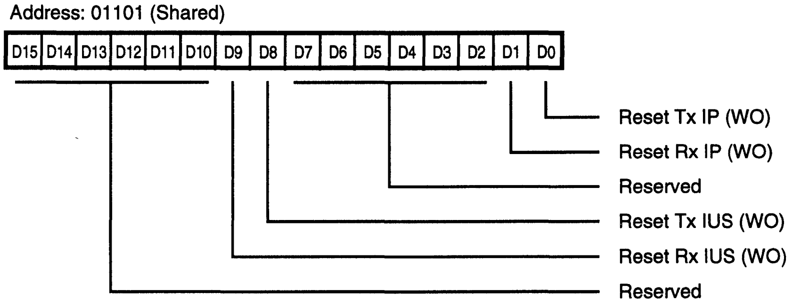


Figure 17. Clear DMA Interrupt Register (CDIR)

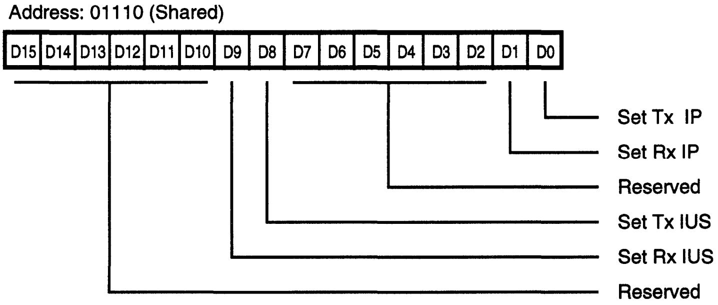


Figure 18. Set DMA Interrupt Register (SDIR)

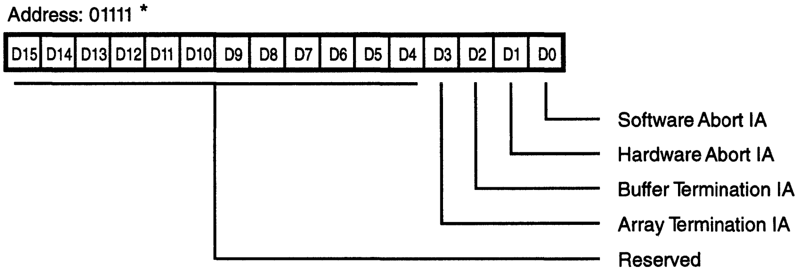


Figure 19. Tx/Rx DMA Interrupt Arm (TDIAR)/(RDIAR)

* The format of this register is the same for the receiver and transmitter. The transmit register is accessed by addressing it with the D//C pin Low (0). The receive register is accessed by addressing it with the D//C pin High (1). This applies to Figures 19 through 25.

CONTROL REGISTERS (Continued)

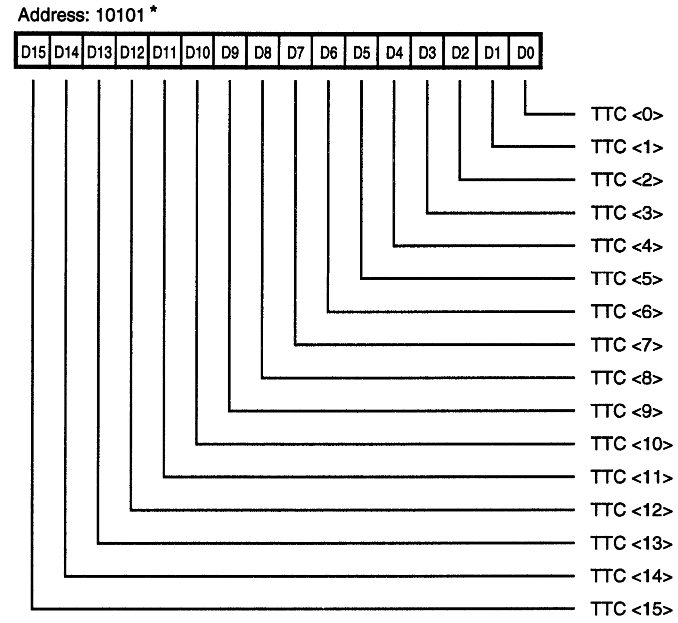


Figure 20. Tx/Rx Byte Count Register (TBCR)/(RBCR)

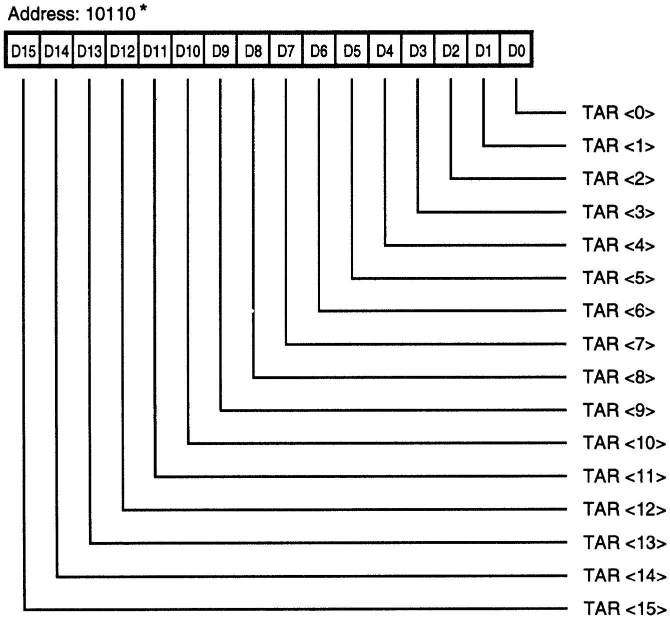


Figure 21. Tx/Rx Address Register (lower) (TARL)/(RARL)

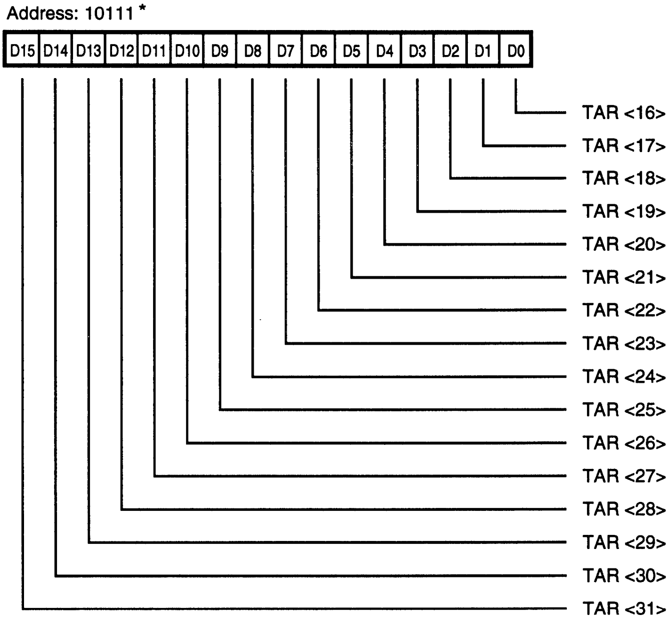


Figure 22. Tx/Rx Address Register (Upper) (TARU)/(RARU)

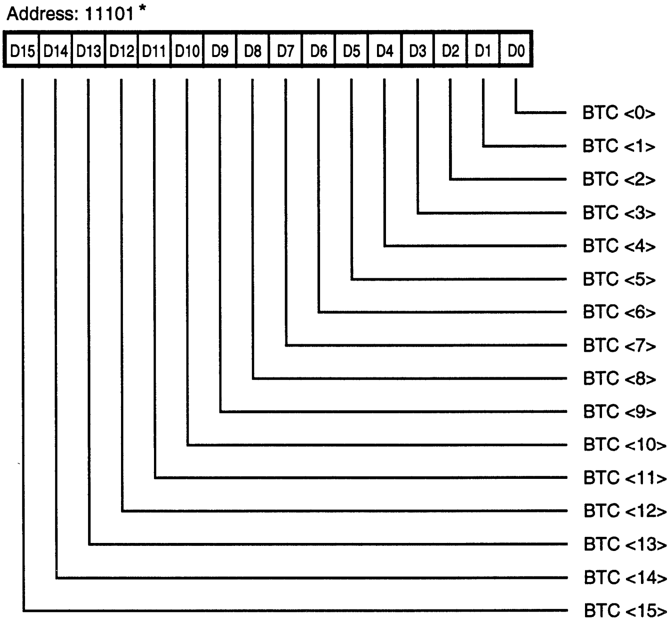


Figure 23. Next Tx/Rx Byte Counter Register (NTBCR)/(RTBCR)

CONTROL REGISTERS (Continued)

Address: 11110*

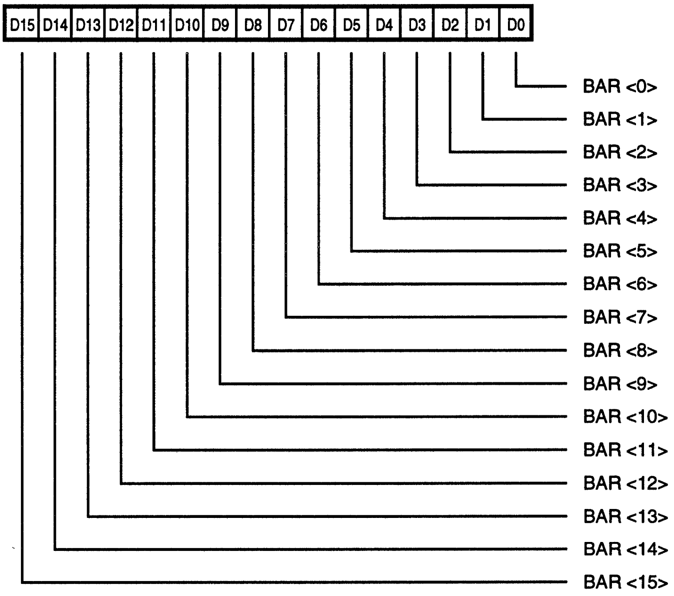


Figure 24. Next Tx/Rx Address Register (Lower) (NTARL)/(RTARL)

Address: 11111 *

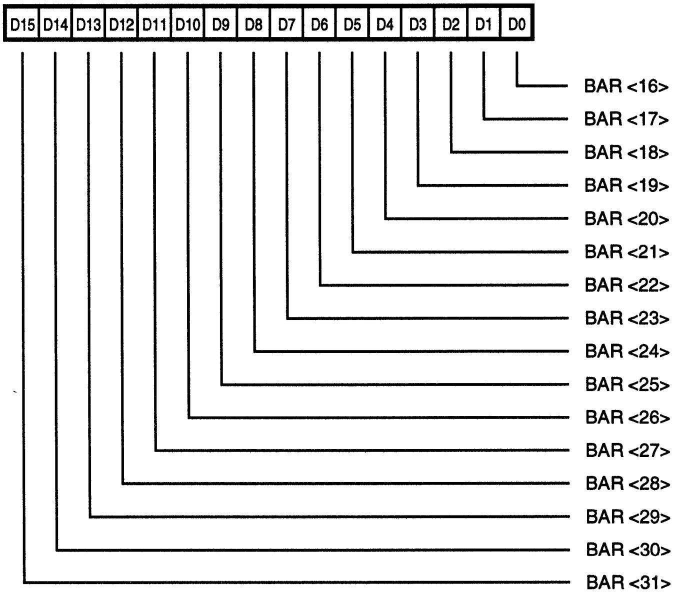


Figure 25. Next Tx/Rx Address Register (Upper) (NTARU)/(RTARU)

		AD15	AD0
Base Address	Buffer #1	AD<31-24>	AD<23-16>
Base Address + 2		AD<15-8>	AD<7-0>
Base Address + 4		CNT<15-8>	CNT<7-0>
Base Address + 6	Buffer #2	AD<31-24>	AD<23-16>
Base Address + 8		AD<15-8>	AD<7-0>
Base Address + 10		CNT<15-8>	CNT<7-0>
Base Address + 12	Buffer #3	AD<31-24>	AD<23-16>
Base Address + 14		AD<15-8>	AD<7-0>
Base Address + 16		CNT<15-8>	CNT<7-0>

Last Base Address	Dummy	Ignored	Ignored
Last Base Address + 2		Ignored	Ignored
Last Base Address + 4		0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
Base Address Register After Termination			

Figure 26a. Array-Chained, 16-Bit Bus, Big End Array

Note:
The addition of frame status/control information in the array with Linked Frame Status Transfer Enabled is similar for Big and Little End Array. See Figure 26b.

CONTROL REGISTERS (Continued)

		AD15	AD0
Base Address	Buffer #1	AD<31-24>	AD<23-16>
Base Address + 2		AD<15-8>	AD<7-0>
Base Address + 4		CNT <15-8>	CNT <7-0>
Base Address + 6		RSB/TCB <15-8>	RSB/TCB <7-0>
Base Address + 8		RCHR/TCLR <15-8>	RCHR/TCLR <7-0>
Base Address + 10		0	0
Base Address + 12	Buffer #2	AD<31-24>	AD<23-16>
Base Address + 14		AD<15-8>	AD<7-0>
Base Address + 16		CNT <15-8 >	CNT <7-0>
Base Address + 18		RSB/TCB <15-8>	RSB/TCB <7-0>
Base Address + 20		RCHR/TCLR <15-8>	RCHR/TCLR <7-0>
Base Address + 22		0	0
Base Address + 24	Buffer #3	AD<31-24>	AD<23-16>
Base Address + 26		AD<15-8>	AD<7-0>
Base Address + 28		CNT <15-8>	CNT <7-0>
Base Address + 30		RSB/TCB <15-8>	RSB/TCB <7-0>
Base Address + 32		RCHR/TCLR <15-8>	RCHR/TCLR <7-0>
Base Address + 34		0	0
Last Base Address	Dummy	Ignored	Ignored
Last Base Address + 2		Ignored	Ignored
Last Base Address + 4		0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
Base Address Register After Termination			

Figure 26b. Array-Chained, 16-Bit Bus, Big End Array Linked Frame Status Transfer Enabled

		AD15	AD0
Base Address	Buffer #1	AD<15-8>	AD<7-0>
Base Address + 2		AD<31-24>	AD<23-16>
Base Address + 4		CNT<15-8>	CNT<7-0>
Base Address + 6	Buffer #2	AD<15-8>	AD<7-0>
Base Address + 8		AD<31-24>	AD<23-16>
Base Address + 10		CNT<15-8>	CNT<7-0>
Base Address + 12	Buffer #3	AD<15-8>	AD<7-0>
Base Address + 14		AD<31-24>	AD<23-16>
Base Address + 16		CNT<15-8>	CNT<7-0>

Last Base Address	Dummy	Ignored	Ignored
Last Base Address + 2		Ignored	Ignored
Last Base Address + 4		0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
Base Address Register After Termination			

Figure 27. Array-Chained, 16-Bit Bus, Little End Array

CONTROL REGISTERS (Continued)

		AD7 AD0
Base Address	Buffer #1	AD<31-24>
Base Address + 1		AD<23-16>
Base Address + 2		AD<15-8>
Base Address + 3		AD<7-0>
Base Address + 4		CNT<15-8>
Base Address + 5		CNT<7-0>
Base Address + 6	Buffer #2	AD<31-24>
Base Address + 7		AD<23-16>
Base Address + 8		AD<15-8>
Base Address + 9		AD<7-0>
Base Address + 10		CNT<15-8>
Base Address + 11		CNT<7-0>

Last Base Address	Dummy	Ignored
Last Base Address + 1		Ignored
Last Base Address + 2		Ignored
Last Base Address + 3		Ignored
Last Base Address + 4		0 0 0 0 0 0 0 0
Last Base Address + 5		0 0 0 0 0 0 0 0
Base Address Register After Termination		

Figure 28a. Array-Chained, 8-Bit Bus, Big End Array

Note:
The addition of frame status/control information in the array with Linked Frame Status Transfer Enabled is similar for Big and Little End Array. See Figure 28b.

		AD7 AD0
Base Address	Buffer #1	AD<31-24>
Base Address + 1		AD<23-16>
Base Address + 2		AD<15-8>
Base Address + 3		AD<7-0>
Base Address + 4		CNT<15-8>
Base Address + 5		CNT<7-0>
Base Address + 6		RSB/TCB <15-8>
Base Address + 7		RSB/TCB <7-0>
Base Address + 8		RCHR/TCLR <15-8>
Base Address + 9		RSHR/TCLR <7-0>
Base Address + 10		0
Base Address + 11		0
Base Address + 12	Buffer #2	AD<31-24>
Base Address + 13		AD<23-16>
Base Address + 14		AD<15-8>
Base Address + 15		AD<7-0>
Base Address + 16		CNT<15-8>
Base Address + 17		CNT<7-0>
Base Address + 18		RSB/TCB <15-8>
Base Address + 19		RSB/TCB <7-0>
Base Address + 20		RCHR/TCLR <15-8>
Base Address + 21		RSHR/TCLR <7-0>
Base Address + 22		0
Base Address + 23		0

Last Base Address	Dummy	Ignored
Last Base Address + 1		Ignored
Last Base Address + 2		Ignored
Last Base Address + 3		Ignored
Last Base Address + 4		0 0 0 0 0 0 0 0
Last Base Address + 5		0 0 0 0 0 0 0 0
Base Address Register After Termination		

Figure 28b. Array-Chained, 8-Bit Bus, Big End Array, Linked Frame Status Transfer Enabled

CONTROL REGISTERS (Continued)

		AD7 AD0
Base Address	Buffer #1	AD<7-0>
Base Address + 1		AD<15-8>
Base Address + 2		AD<23-16>
Base Address + 3		AD<31-24>
Base Address + 4		CNT<7-0>
Base Address + 5		CNT<15-8>
Base Address + 6	Buffer #2	AD<7-0>
Base Address + 7		AD<15-8>
Base Address + 8		AD<23-16>
Base Address + 9		AD<31-24>
Base Address + 10		CNT<7-0>
Base Address + 11		CNT<15-8>

Last Base Address	Dummy	Ignored
Last Base Address + 1		Ignored
Last Base Address + 2		Ignored
Last Base Address + 3		Ignored
Last Base Address + 4		0 0 0 0 0 0 0 0
Last Base Address + 5		0 0 0 0 0 0 0 0
Base Address Register After Termination		

Figure 29. Array-Chained, 8-Bit Bus, Little End Array

		AD15	AD0
Base Address	Buffer #1	AD<31-24>	AD<23-16>
Base Address + 2		AD<15-8>	AD<7-0>
Base Address + 4		CNT<15-8>	CNT<7-0>
Base Address + 6	Base #2	AD<31-24>	AD<23-16>
Base Address + 8		AD<15-8>	AD<7-0>
#2 Base Address	Buffer #2	AD<31-24>	AD<23-16>
#2 Base Address + 2		AD<15-8>	AD<7-0>
#2 Base Address + 4		CNT<15-8>	CNT<7-0>
#2 Base Address + 6	Base #3	AD<31-24>	AD<23-16>
#2 Base Address + 8		AD<15-8>	AD<7-0>
#3 Base Address	Buffer #3	AD<31-24>	AD<23-16>
#3 Base Address + 2		AD<15-8>	AD<7-0>
#3 Base Address + 4		CNT<15-8>	CNT<7-0>
#n - 1 Base Address + 6	Base #n	AD<31-24>	AD<23-16>
#n - 1 Base Address + 8		AD<15-8>	AD<7-0>
#n Base Address	Buffer #n	Ignored	Ignored
#n Base Address + 2		Ignored	Ignored
#n Base Address + 4		0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Figure 30a. Linked Array-Chained, 16-Bit Bus, Big End Array

Note:
The addition of frame status/control information in the array with Linked Frame Status Transfer Enabled is similar for Big and Little End Array. See Figure 30b.

CONTROL REGISTERS (Continued)

		AD15	AD0
Base Address	Buffer #1	AD<31-24>	AD<23-16>
Base Address + 2		AD <15-8>	AD<7-0>
Base Address + 4		CNT<15-8>	CNT<7-0>
Base Address + 6		RSB/TCB <15-8>	RSB/TCB <7-0>
Base Address + 8		RCHR/TCLR <15-8>	RCHR/TCLR <7-0>
Base Address + 10		0	0
Base Address + 12	Base #2	AD<31-24>	AD<23-16>
Base Address + 14		AD <15-8>	AD<7-0>
#2 Base Address	Buffer #2	AD<31-24>	AD<23-16>
#2 Base Address + 2		AD <15-8>	AD<7-0>
#2 Base Address + 4		CNT<15-8>	CNT<7-0>
#2 Base Address + 6		RSB/TCB <15-8>	RSB/TCB <7-0>
#2 Base Address + 8		RCHR/TCLR <15-8>	RCHR/TCLR <7-0>
#2 Base Address + 10		0	0
#2 Base Address + 12	Base #3	AD<31-24>	AD<23-16>
#2 Base Address + 14		AD <15-8>	AD<7-0>
#3 Base Address	Buffer #3	AD<31-24>	AD<23-16>
#3 Base Address + 2		AD <15-8>	AD<7-0>
#3 Base Address + 4		CNT<15-8>	CNT<7-0>
#3 Base Address + 6		RSB/TCB <15-8>	RSB/TCB <7-0>
#3 Base Address + 8		RCHR/TCLR <15-8>	RCHR/TCLR <7-0>
#3 Base Address + 10		0	0

Figure 30b. Linked Array-Chained, 16-Bit Bus, Big End Array

		AD15	AD0
Base Address	Buffer #1	AD<15-8>	AD<7-0>
Base Address + 2		AD<31-24>	AD<23-16>
Base Address + 4		CNT<15-8>	CNT<7-0>
Base Address + 6	Base #2	AD<15-8>	AD<7-0>
Base Address + 8		AD<31-24>	AD<23-16>
#2 Base Address	Buffer #2	AD<15-8>	AD<7-0>
#2 Base Address + 2		AD<31-24>	AD<23-16>
#2 Base Address + 4		CNT<15-8>	CNT<7-0>
#2 Base Address + 6	Base #3	AD<15-8>	AD<7-0>
#2 Base Address + 8		AD<31-24>	AD<23-16>
#3 Base Address	Buffer #3	AD<15-8>	AD<7-0>
#3 Base Address + 2		AD<31-24>	AD<23-16>
#3 Base Address + 4		CNT<15-8>	CNT<7-0>
#n - 1 Base Address + 6	Base #n	AD<15-8>	AD<7-0>
#n - 1 Base Address + 8		AD<31-24>	AD<23-16>
#n Base Address	Buffer #n	Ignored	Ignored
#n Base Address + 2		Ignored	Ignored
#n Base Address + 4		0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Figure 31. Linked Array-Chained, 16-Bit Bus, Little End Array

CONTROL REGISTERS (Continued)

		AD7 AD0
Base Address	Buffer #1	AD<7-0>
Base Address + 1		AD<15-8>
Base Address + 2		AD<23-16>
Base Address + 3		AD<31-24>
Base Address + 4		CNT<7-0>
Base Address + 5		CNT<15-8>
Base Address + 6	Base #2	AD<7-0>
Base Address + 7		AD<15-8>
Base Address + 8		AD<23-16>
Base Address + 9		AD<31-24>
#2 Base Address	Buffer #2	AD<7-0>
#2 Base Address + 1		AD<15-8>
#2 Base Address + 2		AD<23-16>
#2 Base Address + 3		AD<31-24>
#2 Base Address + 4		CNT<7-0>
#2 Base Address + 5		CNT<15-8>
#2 Base Address + 6	Base #3	AD<7-0>
#2 Base Address + 7		AD<15-8>
#2 Base Address + 8		AD<23-16>
#2 Base Address + 9		AD<31-24>

Figure 32a. Linked Array-Chained, 8-Bit Bus, Big End Array

Note:
The addition of frame status/control information in the array with Linked Frame Status Transfer Enabled is similar to Big End Array. See Figure 32b.

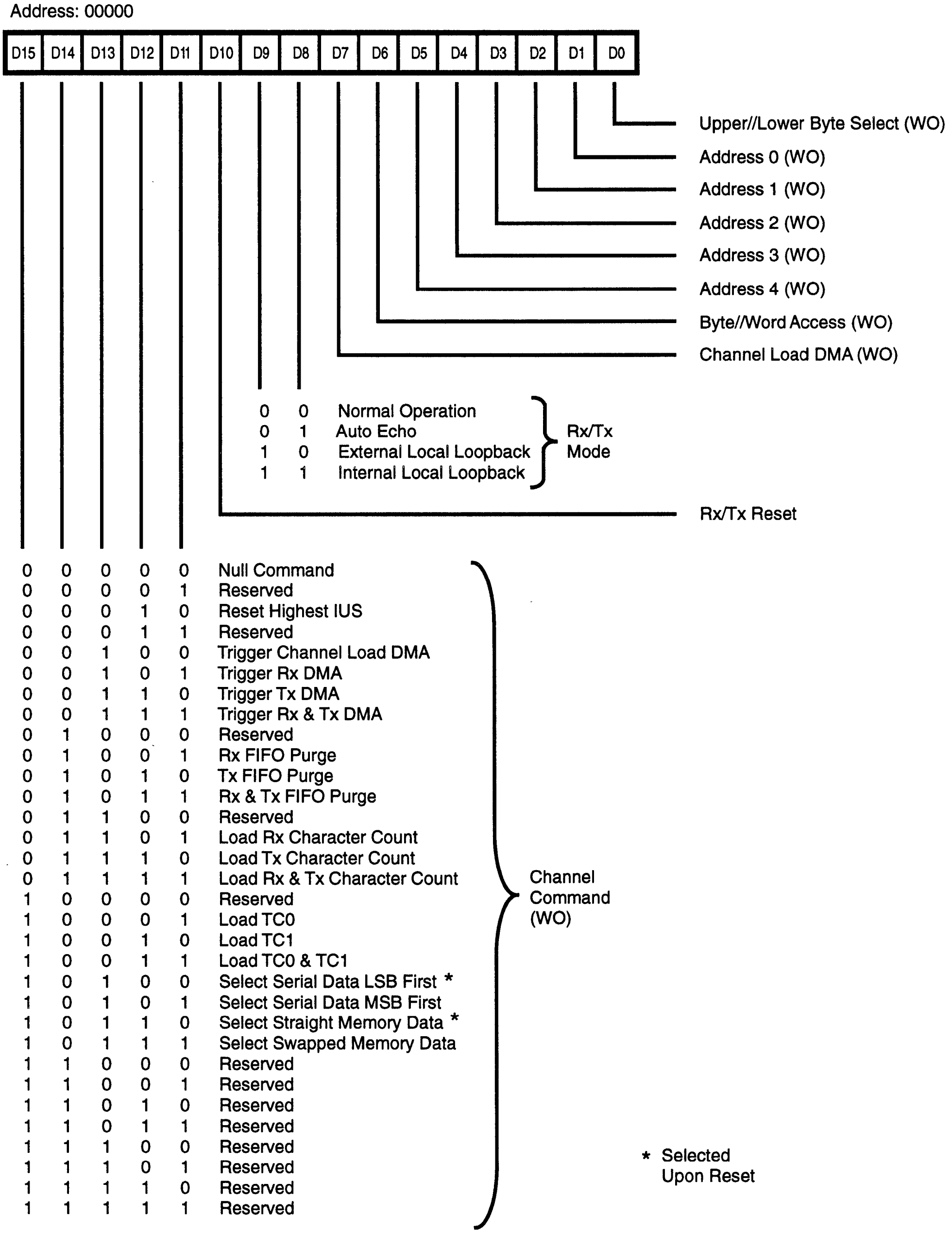
		AD7 AD0
Base Address	Buffer #1	AD<31-24>
Base Address + 1		AD<23-16>
Base Address + 2		AD<15-8>
Base Address + 3		AD<7-0>
Base Address + 4		CNT<15-8>
Base Address + 5		CNT<7-0>
Base Address + 6		RSB/TCB <15-8>
Base Address + 7		RSB/TCB <7-0>
Base Address + 8		RCHR/TCLR <15-8>
Base Address + 9		RCHR/TCLR <7-0>
Base Address + 10		0
Base Address + 11		0
Base Address + 12	Base #2	AD<31-24>
Base Address + 13		AD<23-16>
Base Address + 14		AD<15-8>
Base Address + 15		AD<7-0>
#2 Base Address	Buffer #2	AD<31-24>
#2 Base Address + 1		AD<23-16>
#2 Base Address + 2		AD<15-8>
#2 Base Address + 3		AD<7-0>
#2 Base Address + 4		CNT<15-8>
#2 Base Address + 5		CNT<7-0>
#2 Base Address + 6		RSB/TCB <15-8>
#2 Base Address + 7		RSB/TCB <7-0>
#2 Base Address + 8		RCHR/TCLR <15-8>
#2 Base Address + 9		RCHR/TCLR <7-0>
#2 Base Address + 10		0
#2 Base Address + 11		0

Figure 32b. Linked Frame Status Transfer Enables

CONTROL REGISTERS (Continued)

		AD7 AD0
Base Address	Buffer #1	AD<31-24>
Base Address + 1		AD<23-16>
Base Address + 2		AD<15-8>
Base Address + 3		AD<7-0>
Base Address + 4		CNT<15-8>
Base Address + 5		CNT<7-0>
Base Address + 6	Base #2	AD<31-24>
Base Address + 7		AD<23-16>
Base Address + 8		AD<15-8>
Base Address + 9		AD<7-0>
#2 Base Address	Buffer #2	AD<31-24>
#2 Base Address + 1		AD<23-16>
#2 Base Address + 2		AD<15-8>
#2 Base Address + 3		AD<7-0>
#2 Base Address + 4		CNT<15-8>
#2 Base Address + 5		CNT<7-0>
#2 Base Address + 6	Base #3	AD<31-24>
#2 Base Address + 7		AD<23-16>
#2 Base Address + 8		AD<15-8>
#2 Base Address + 9		AD<7-0>

Figure 33. Linked Array-Chained 8-Bit Bus, Little End Array



Address: 00000

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

- D0: Upper//Lower Byte Select (WO)
- D1: Address 0 (WO)
- D2: Address 1 (WO)
- D3: Address 2 (WO)
- D4: Address 3 (WO)
- D5: Address 4 (WO)
- D6: Byte//Word Access (WO)
- D7: Channel Load DMA (WO)
- D10: Rx/Tx Reset

Rx/Tx Mode (D9, D8):

D9	D8	
0	0	Normal Operation
0	1	Auto Echo
1	0	External Local Loopback
1	1	Internal Local Loopback

Channel Command (WO) (D15–D11):

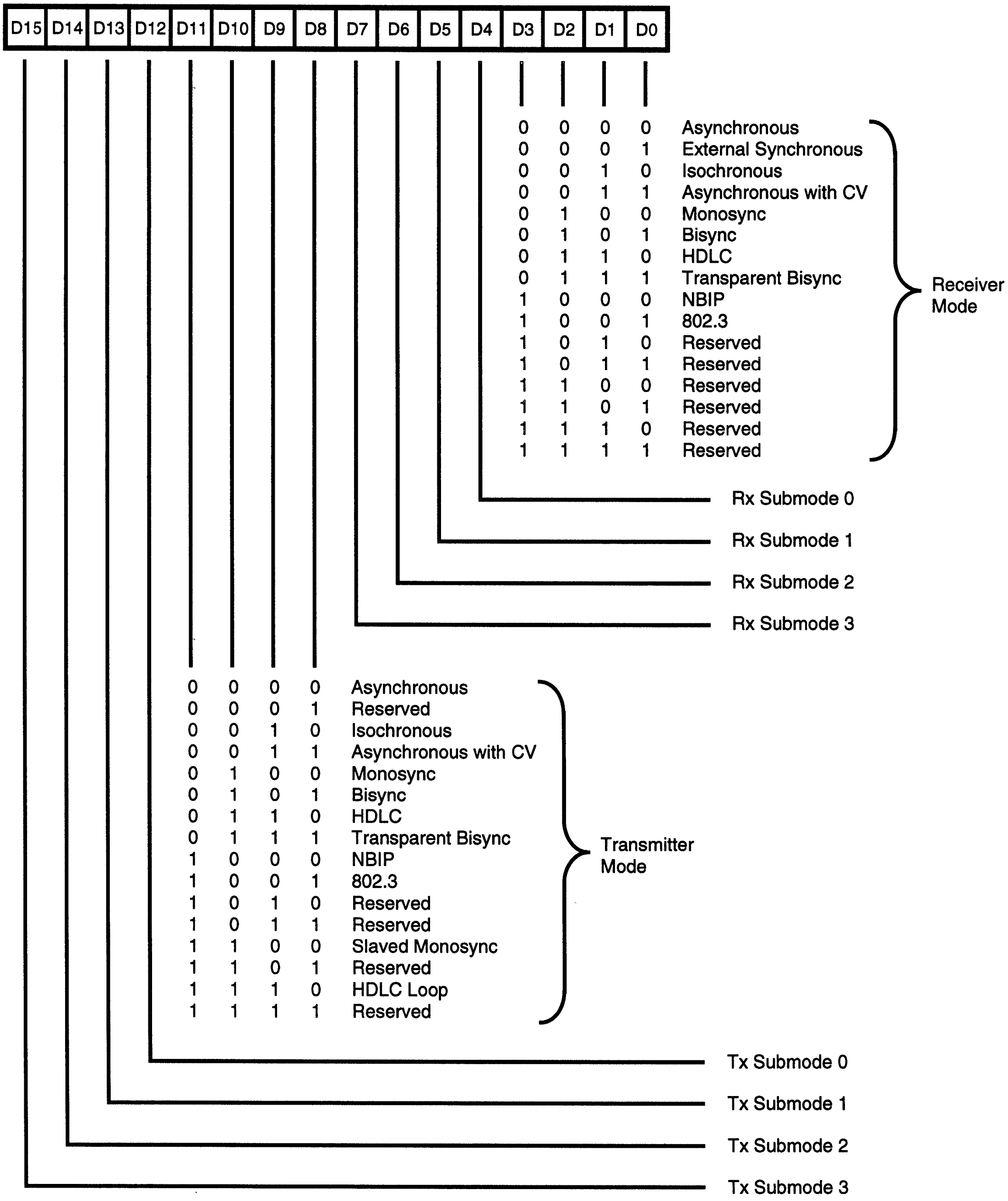
D15	D14	D13	D12	D11	
0	0	0	0	0	Null Command
0	0	0	0	1	Reserved
0	0	0	1	0	Reset Highest IUS
0	0	0	1	1	Reserved
0	0	1	0	0	Trigger Channel Load DMA
0	0	1	0	1	Trigger Rx DMA
0	0	1	1	0	Trigger Tx DMA
0	0	1	1	1	Trigger Rx & Tx DMA
0	1	0	0	0	Reserved
0	1	0	0	1	Rx FIFO Purge
0	1	0	1	0	Tx FIFO Purge
0	1	0	1	1	Rx & Tx FIFO Purge
0	1	1	0	0	Reserved
0	1	1	0	1	Load Rx Character Count
0	1	1	1	0	Load Tx Character Count
0	1	1	1	1	Load Rx & Tx Character Count
1	0	0	0	0	Reserved
1	0	0	0	1	Load TC0
1	0	0	1	0	Load TC1
1	0	0	1	1	Load TC0 & TC1
1	0	1	0	0	Select Serial Data LSB First *
1	0	1	0	1	Select Serial Data MSB First
1	0	1	1	0	Select Straight Memory Data *
1	0	1	1	1	Select Swapped Memory Data
1	1	0	0	0	Reserved
1	1	0	0	1	Reserved
1	1	0	1	0	Reserved
1	1	0	1	1	Reserved
1	1	1	0	0	Reserved
1	1	1	0	1	Reserved
1	1	1	1	0	Reserved
1	1	1	1	1	Reserved

* Selected Upon Reset

Figure 34. Channel Command/Address Register (CCAR)

CONTROL REGISTERS (Continued)

Address: 00001



D3	D2	D1	D0	Receiver Mode
0	0	0	0	Asynchronous
0	0	0	1	External Synchronous
0	0	1	0	Isochronous
0	0	1	1	Asynchronous with CV
0	1	0	0	Monosync
0	1	0	1	Bisync
0	1	1	0	HDLC
0	1	1	1	Transparent Bisync
1	0	0	0	NBIP
1	0	0	1	802.3
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved

D11	D10	D9	D8	Transmitter Mode
0	0	0	0	Asynchronous
0	0	0	1	Reserved
0	0	1	0	Isochronous
0	0	1	1	Asynchronous with CV
0	1	0	0	Monosync
0	1	0	1	Bisync
0	1	1	0	HDLC
0	1	1	1	Transparent Bisync
1	0	0	0	NBIP
1	0	0	1	802.3
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Slaved Monosync
1	1	0	1	Reserved
1	1	1	0	HDLC Loop
1	1	1	1	Reserved

Figure 35. Channel Mode Register (CMR)

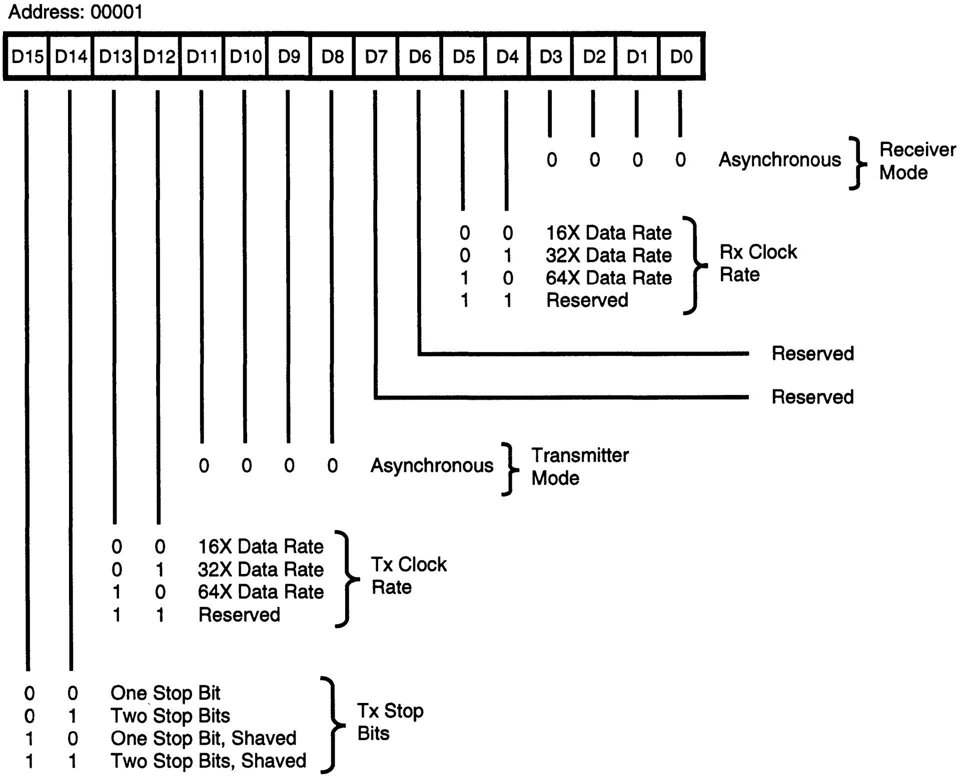


Figure 36. Channel Mode Register, Asynchronous Mode

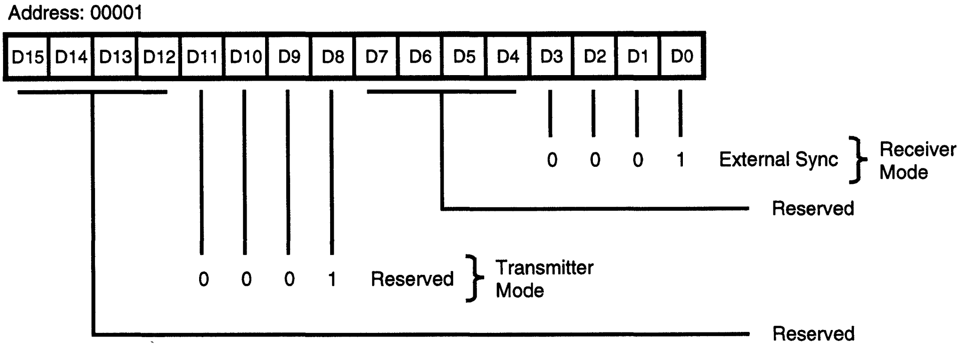


Figure 37. Channel Mode Register, External Sync Mode

CONTROL REGISTERS (Continued)

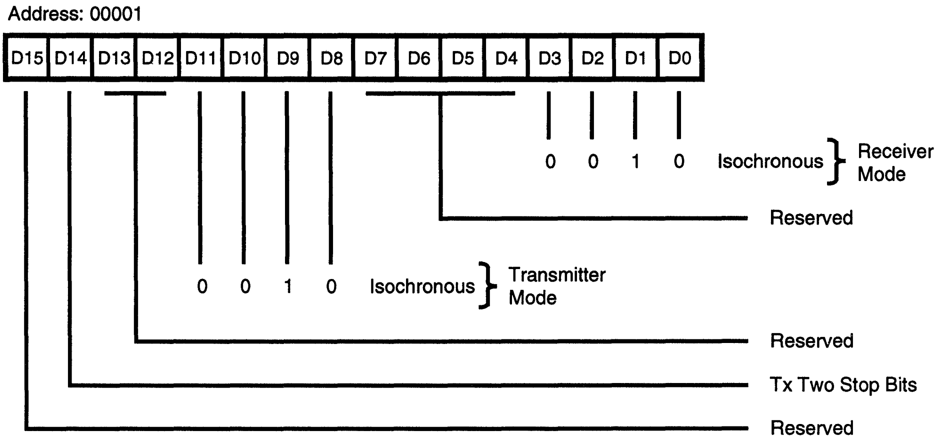


Figure 38. Channel Mode Register, Isochronous Mode

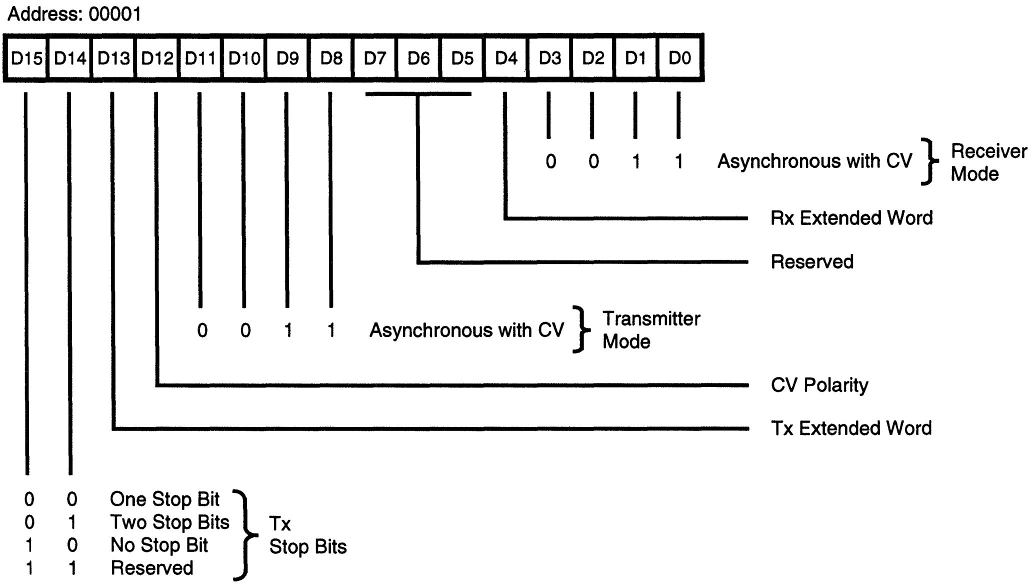


Figure 39. Channel Mode Register, Asynchronous Mode with Code Violation (MIL STD 1553)

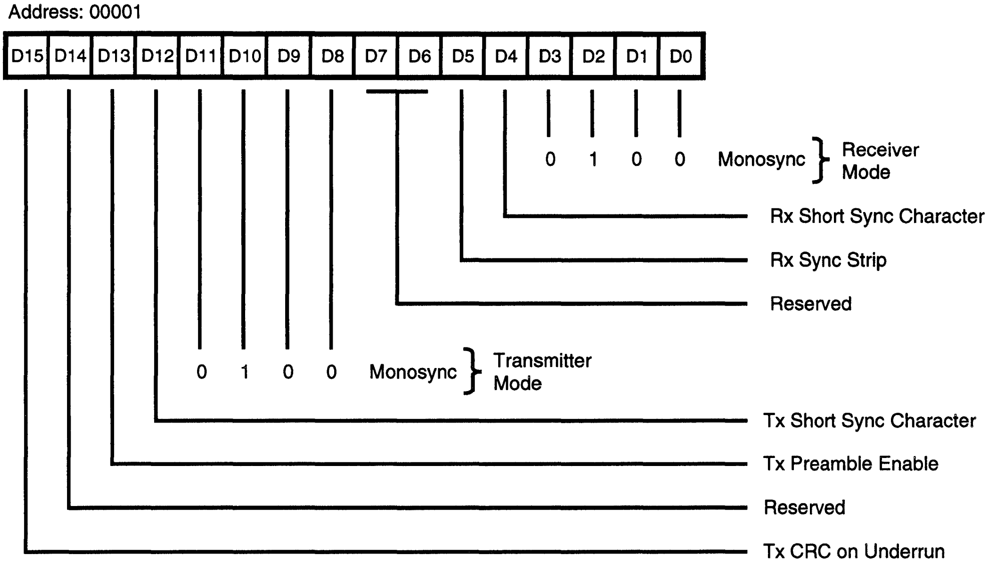


Figure 40. Channel Mode Register, Monosync Mode

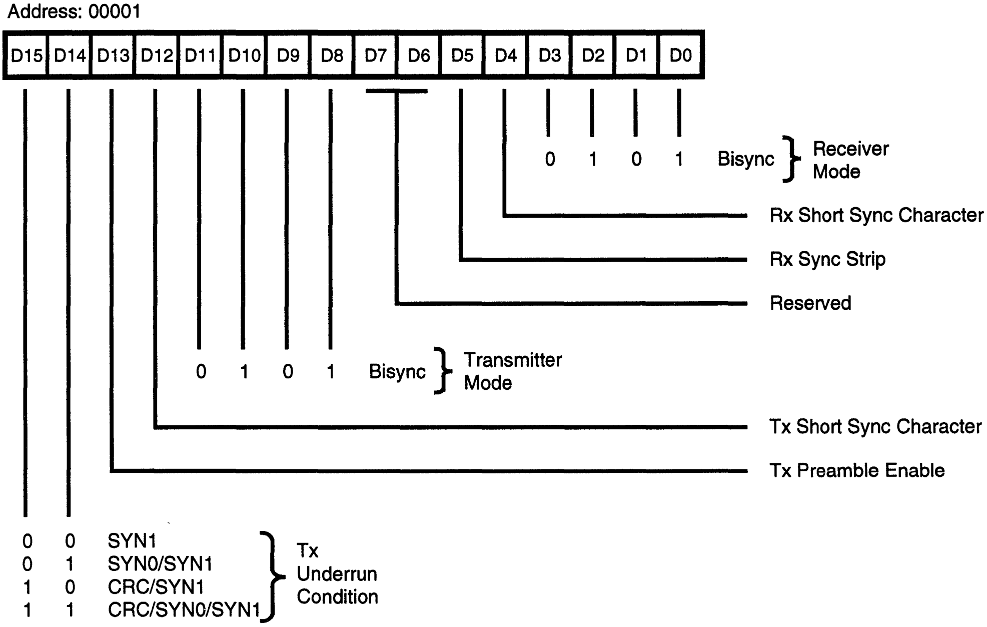


Figure 41. Channel Mode Register, Bisync Mode

CONTROL REGISTERS (Continued)

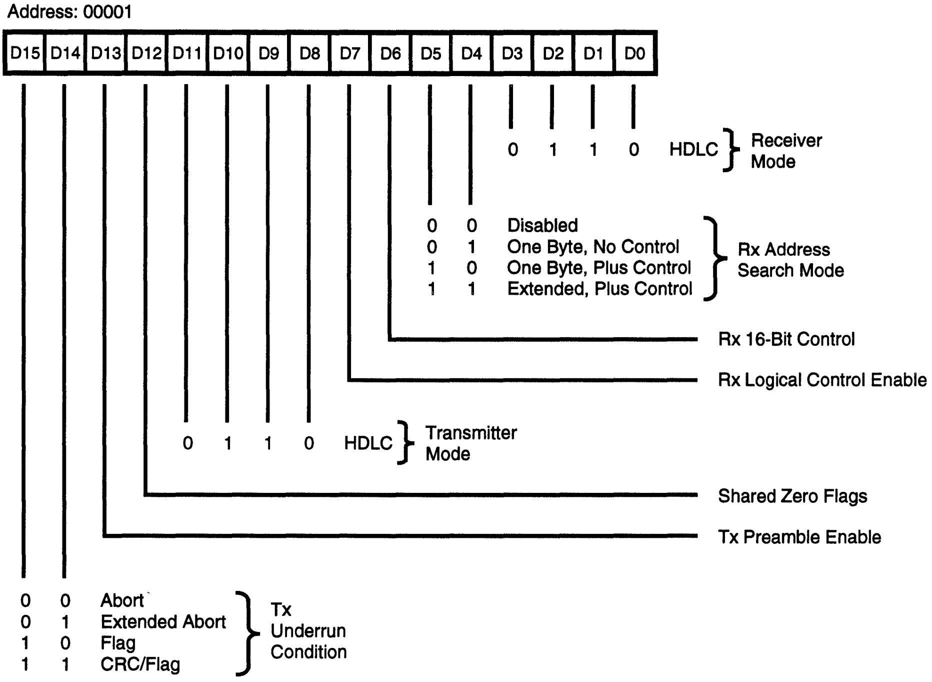


Figure 42. Channel Mode Register, HDLC Mode

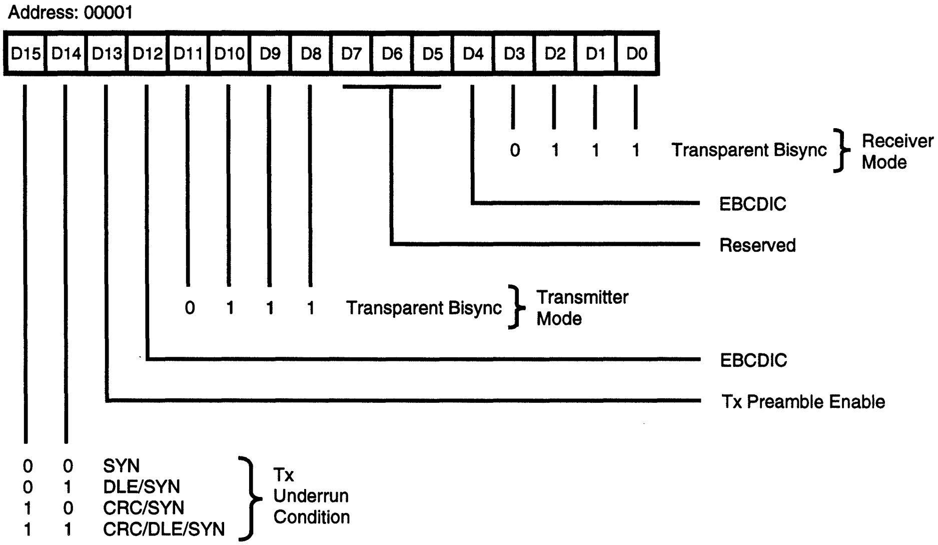

Address: 00001
D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
0 1 1 1 Transparent Bisync Receiver Mode
EBCDIC
Reserved
0 1 1 1 Transparent Bisync Transmitter Mode
EBCDIC
Tx Preamble Enable
0 0 SYN
0 1 DLE/SYN
1 0 CRC/SYN
1 1 CRC/DLE/SYN
Tx Underrun Condition

Figure 43. Channel Mode Register, Transparent Bisync Mode

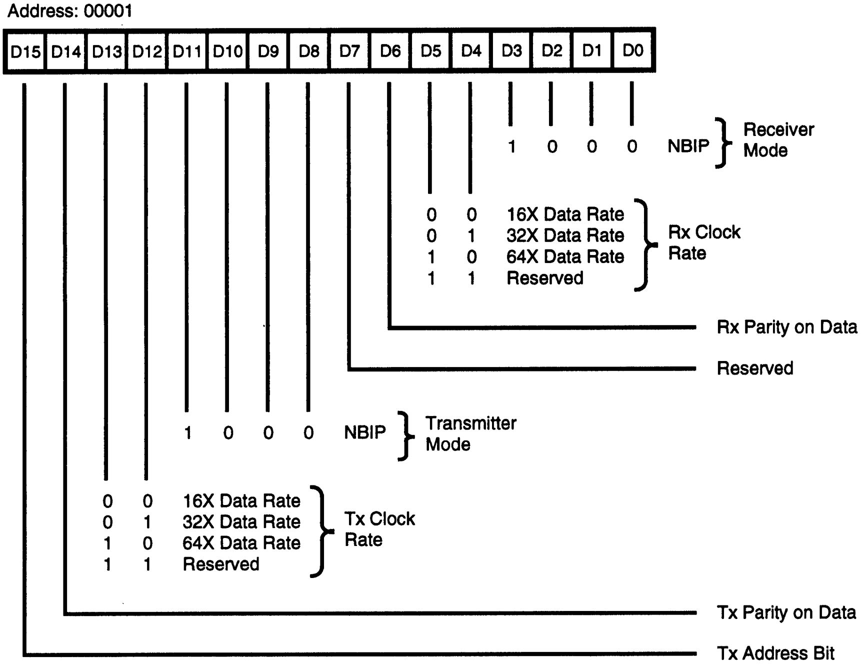


Figure 44. Channel Mode Register, NBIP Mode

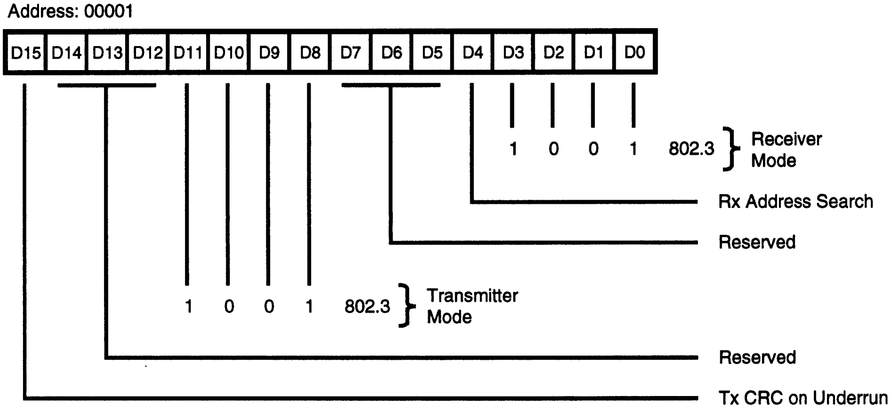


Figure 45. Channel Mode Register, 802.3 Mode

CONTROL REGISTERS (Continued)

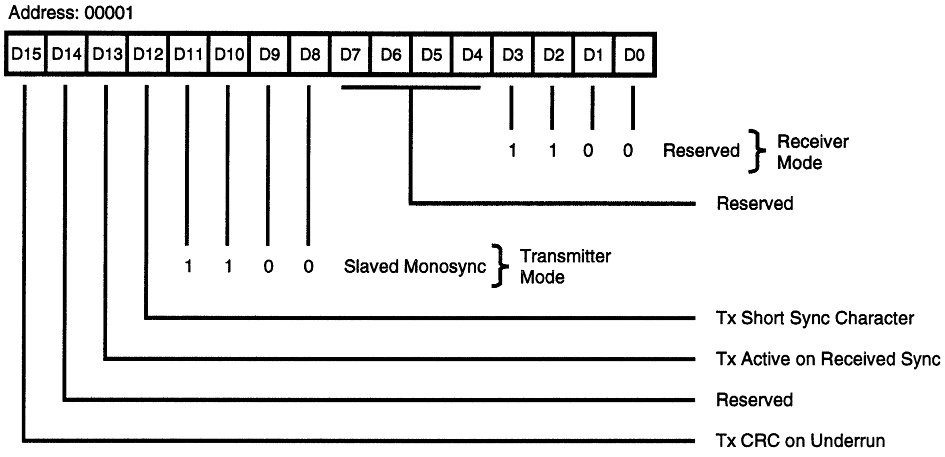


Figure 46. Channel Mode Register, Slaved Monosync Mode

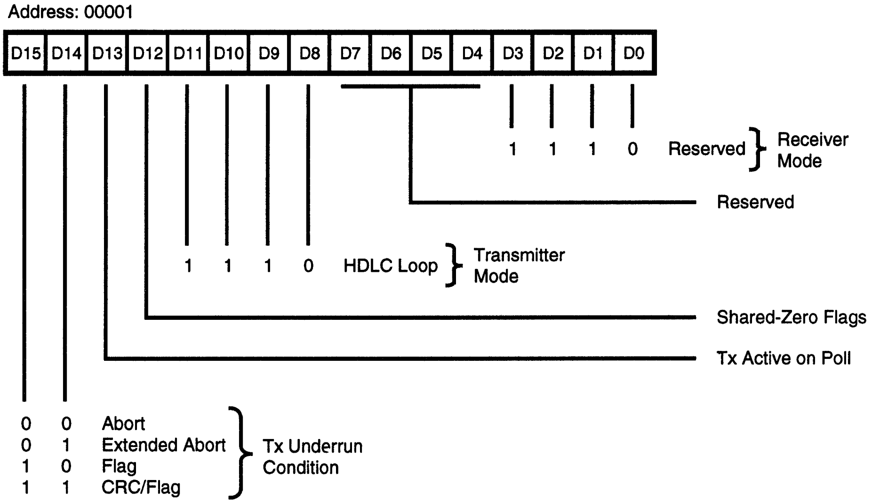


Figure 47. Channel Mode Register, HDLC Loop Mode

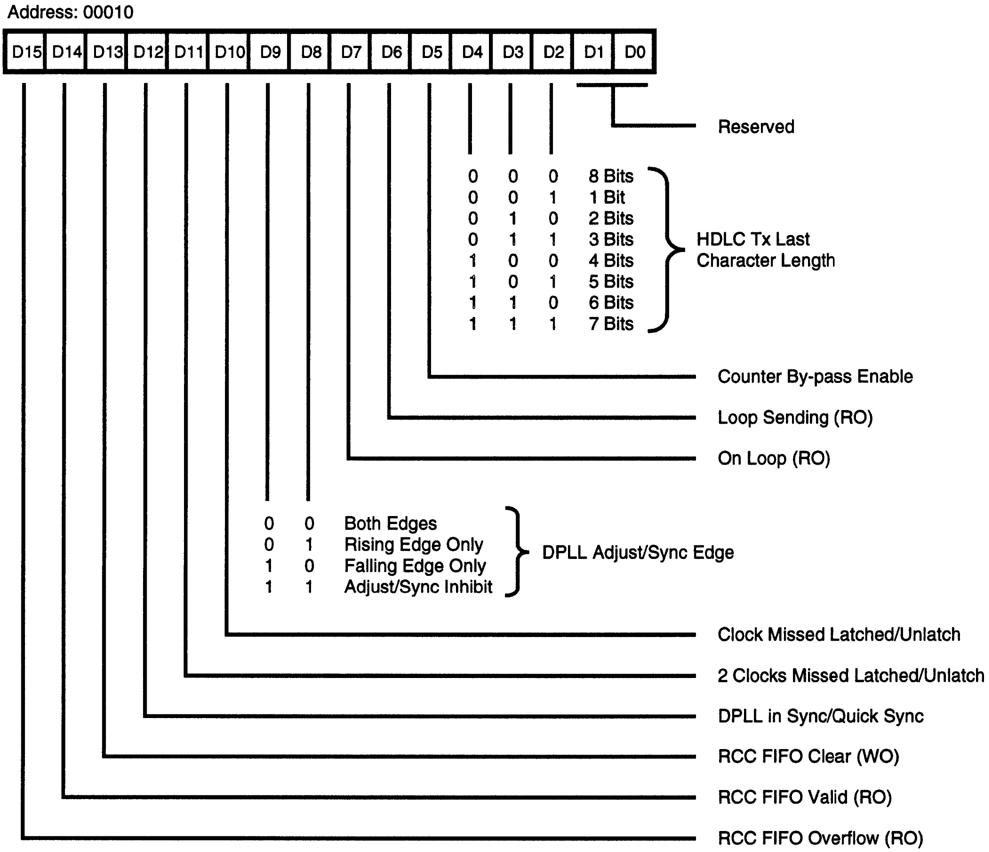


Figure 48. Channel Command/Status Register (CCSR)

CONTROL REGISTERS (Continued)

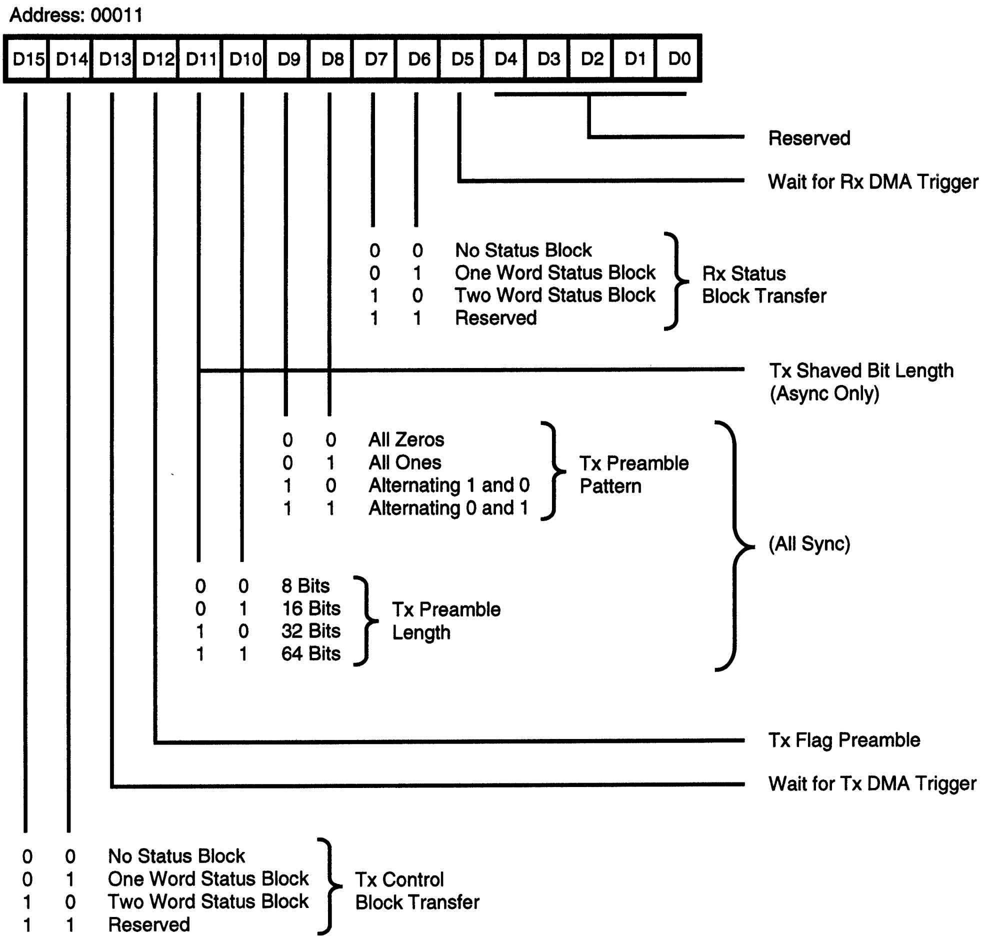


Figure 49. Channel Control Register (CCR)

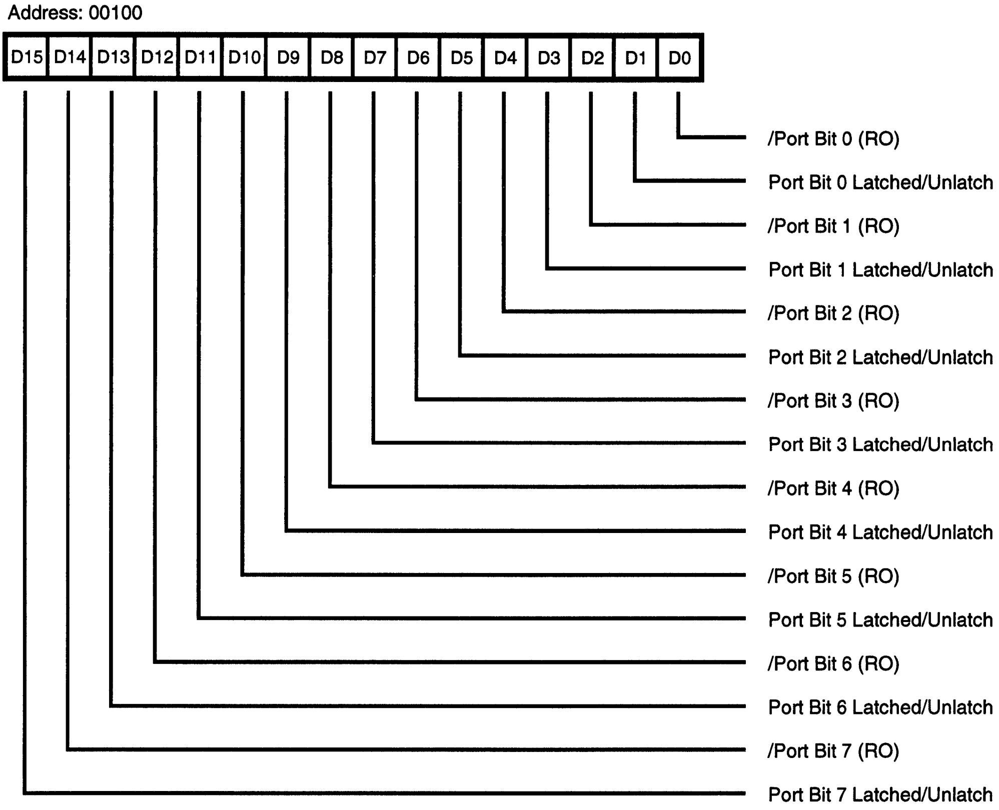

Address: 00100
D15
D14
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
/Port Bit 0 (RO)
Port Bit 0 Latched/Unlatch
/Port Bit 1 (RO)
Port Bit 1 Latched/Unlatch
/Port Bit 2 (RO)
Port Bit 2 Latched/Unlatch
/Port Bit 3 (RO)
Port Bit 3 Latched/Unlatch
/Port Bit 4 (RO)
Port Bit 4 Latched/Unlatch
/Port Bit 5 (RO)
Port Bit 5 Latched/Unlatch
/Port Bit 6 (RO)
Port Bit 6 Latched/Unlatch
/Port Bit 7 (RO)
Port Bit 7 Latched/Unlatch

Figure 50. Port Status Register (PSR)

CONTROL REGISTERS (Continued)

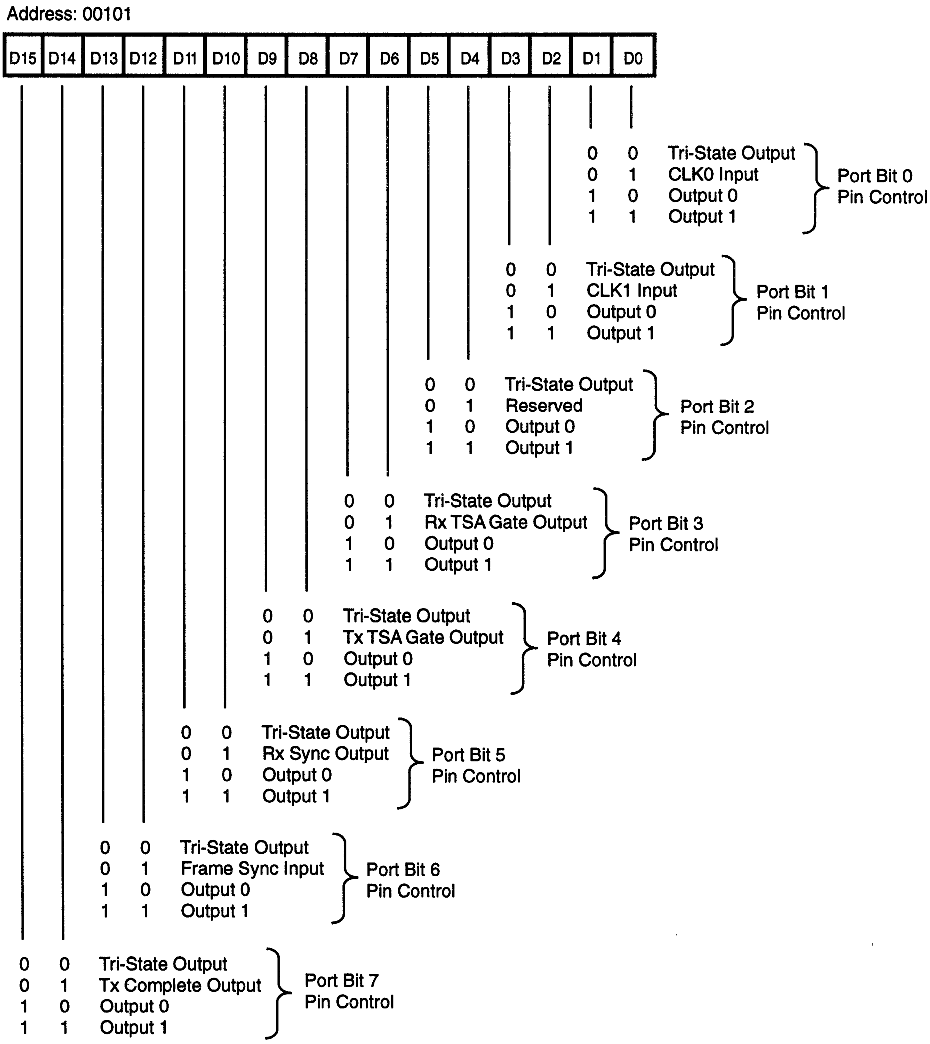


Figure 51. Port Control Register (PCR)

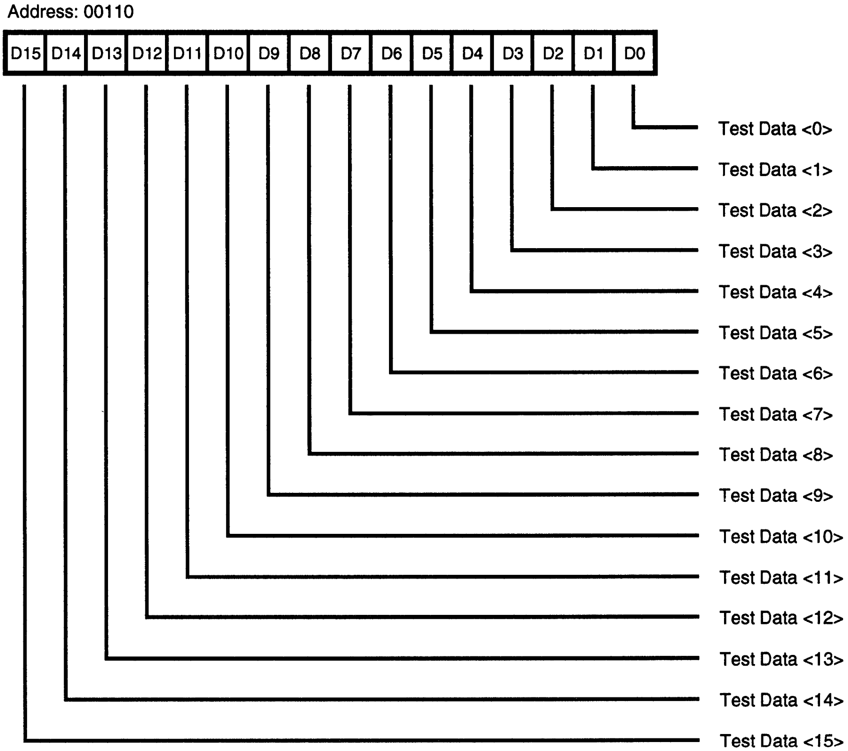
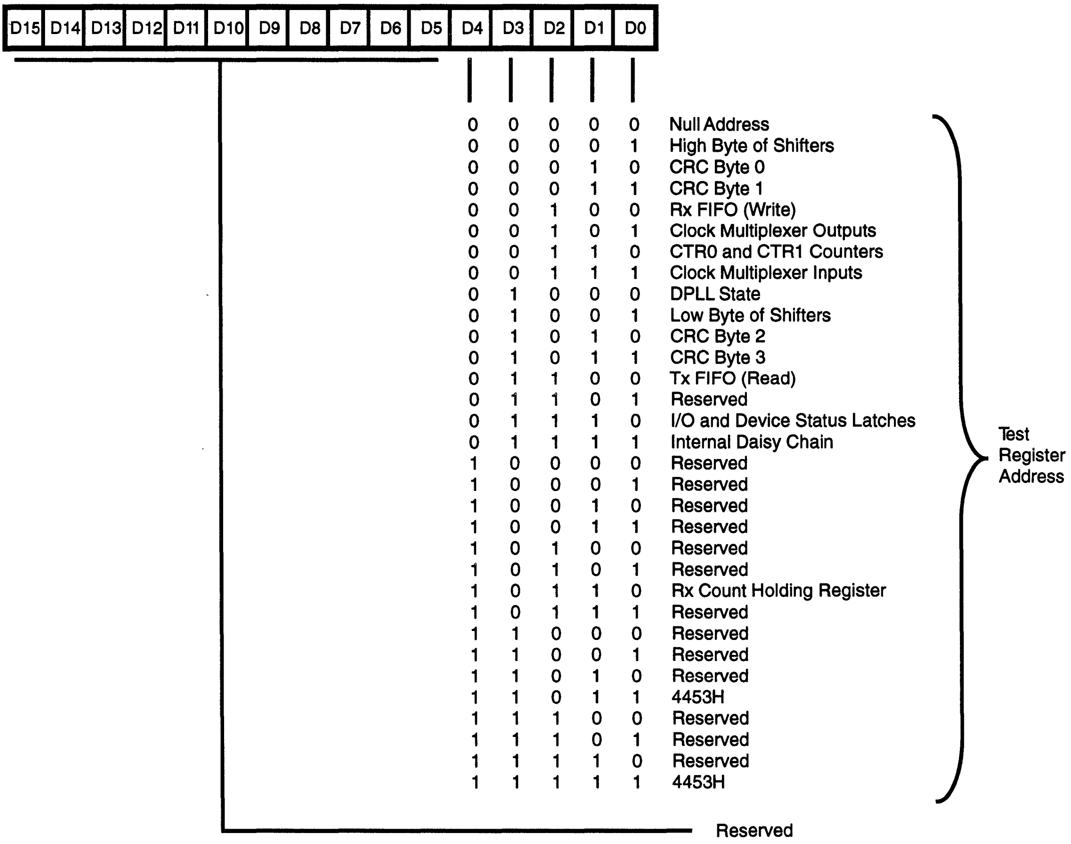


Figure 52. Test Mode Data Register (TMDR)

CONTROL REGISTERS (Continued)

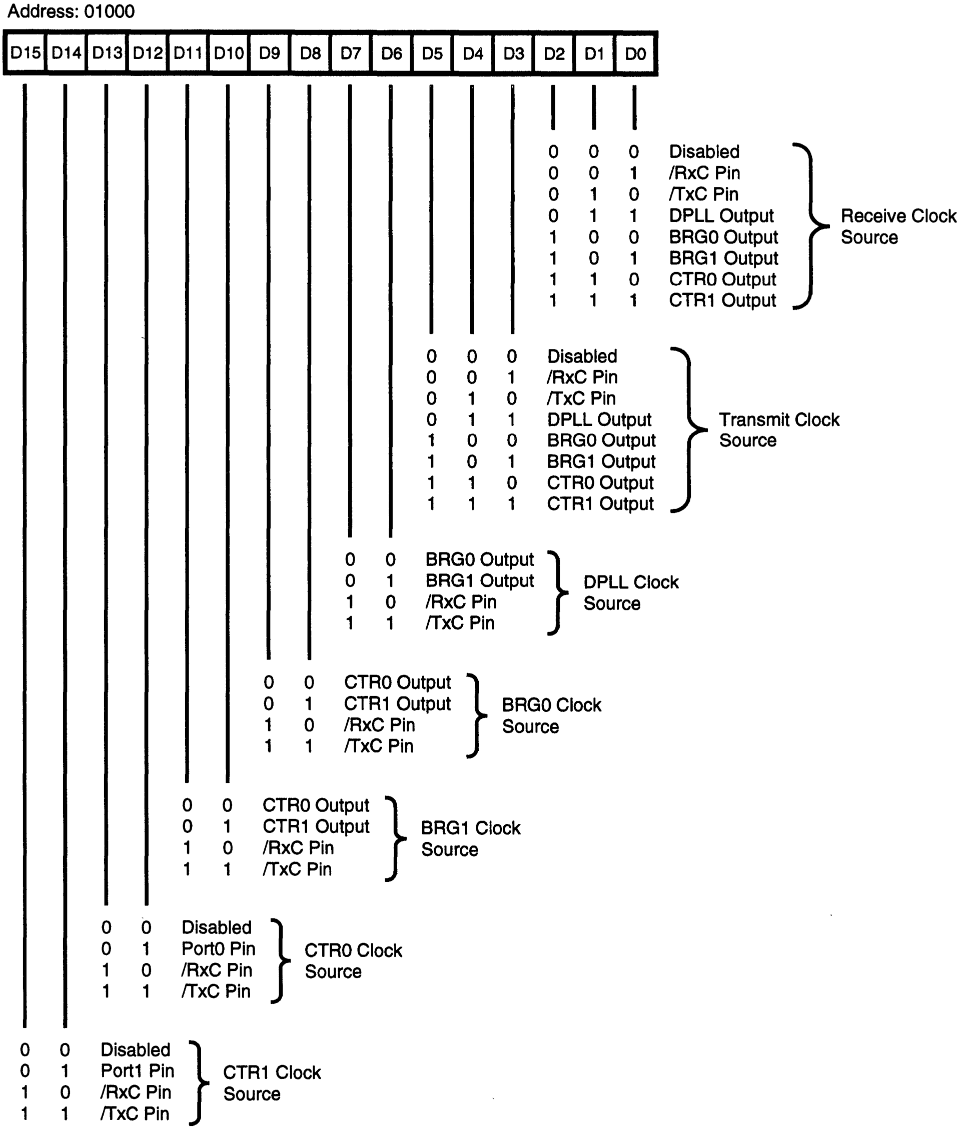
Address: 00111



D4	D3	D2	D1	D0	
0	0	0	0	0	Null Address
0	0	0	0	1	High Byte of Shifters
0	0	0	1	0	CRC Byte 0
0	0	0	1	1	CRC Byte 1
0	0	1	0	0	Rx FIFO (Write)
0	0	1	0	1	Clock Multiplexer Outputs
0	0	1	1	0	CTR0 and CTR1 Counters
0	0	1	1	1	Clock Multiplexer Inputs
0	1	0	0	0	DPLL State
0	1	0	0	1	Low Byte of Shifters
0	1	0	1	0	CRC Byte 2
0	1	0	1	1	CRC Byte 3
0	1	1	0	0	Tx FIFO (Read)
0	1	1	0	1	Reserved
0	1	1	1	0	I/O and Device Status Latches
0	1	1	1	1	Internal Daisy Chain
1	0	0	0	0	Reserved
1	0	0	0	1	Reserved
1	0	0	1	0	Reserved
1	0	0	1	1	Reserved
1	0	1	0	0	Reserved
1	0	1	0	1	Reserved
1	0	1	1	0	Rx Count Holding Register
1	0	1	1	1	Reserved
1	1	0	0	0	Reserved
1	1	0	0	1	Reserved
1	1	0	1	0	Reserved
1	1	0	1	1	4453H
1	1	1	0	0	Reserved
1	1	1	0	1	Reserved
1	1	1	1	0	Reserved
1	1	1	1	1	4453H

Figure 53. Test Mode Control Register (TMCR)

Note:
When software writes the value 1F to the LS byte of the Test Mode Control Register (TMCR), and then reads the Test Mode Data Register (TMDR), current versions of the Z16C32 will return hex 4453. Future revisions, if any, will return other values.



D2	D1	D0	Receive Clock Source
0	0	0	Disabled
0	0	1	/RxC Pin
0	1	0	/TxC Pin
0	1	1	DPLL Output
1	0	0	BRG0 Output
1	0	1	BRG1 Output
1	1	0	CTR0 Output
1	1	1	CTR1 Output

D5	D4	D3	Transmit Clock Source
0	0	0	Disabled
0	0	1	/RxC Pin
0	1	0	/TxC Pin
0	1	1	DPLL Output
1	0	0	BRG0 Output
1	0	1	BRG1 Output
1	1	0	CTR0 Output
1	1	1	CTR1 Output

D7	D6	DPLL Clock Source
0	0	BRG0 Output
0	1	BRG1 Output
1	0	/RxC Pin
1	1	/TxC Pin

D9	D8	BRG0 Clock Source
0	0	CTR0 Output
0	1	CTR1 Output
1	0	/RxC Pin
1	1	/TxC Pin

D11	D10	BRG1 Clock Source
0	0	CTR0 Output
0	1	CTR1 Output
1	0	/RxC Pin
1	1	/TxC Pin

D13	D12	CTR0 Clock Source
0	0	Disabled
0	1	Port0 Pin
1	0	/RxC Pin
1	1	/TxC Pin

D15	D14	CTR1 Clock Source
0	0	Disabled
0	1	Port1 Pin
1	0	/RxC Pin
1	1	/TxC Pin

Figure 54. Clock Mode Control Register (CMCR)

CONTROL REGISTERS (Continued)

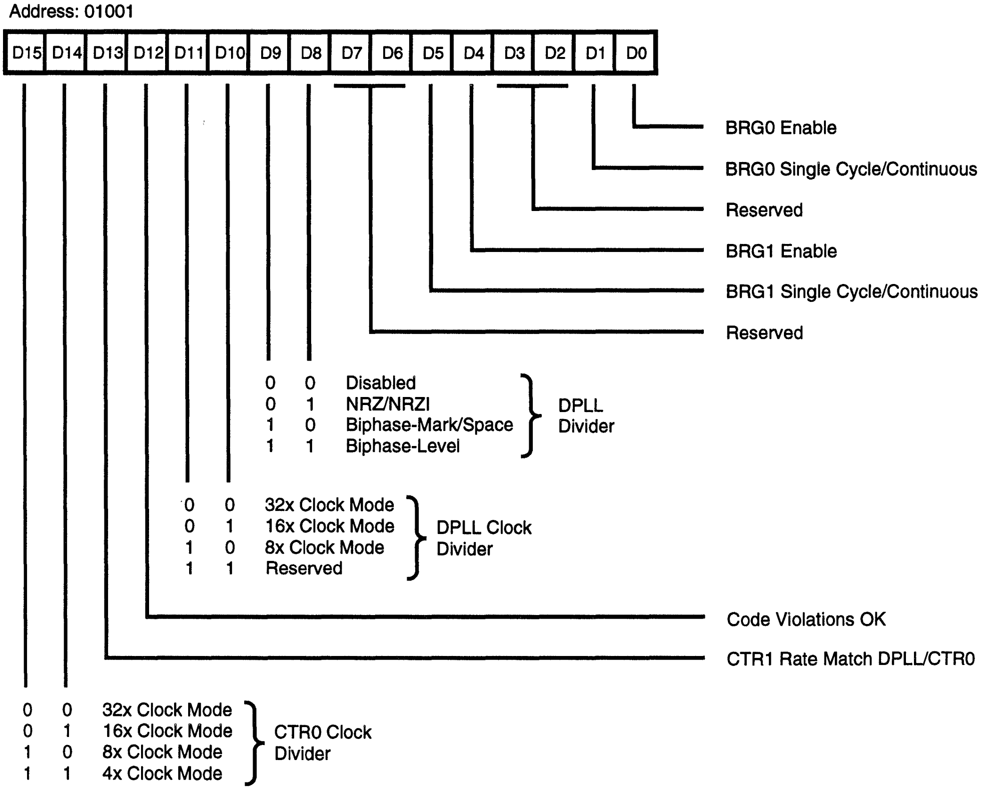


Figure 55. Hardware Configuration Register (HCR)

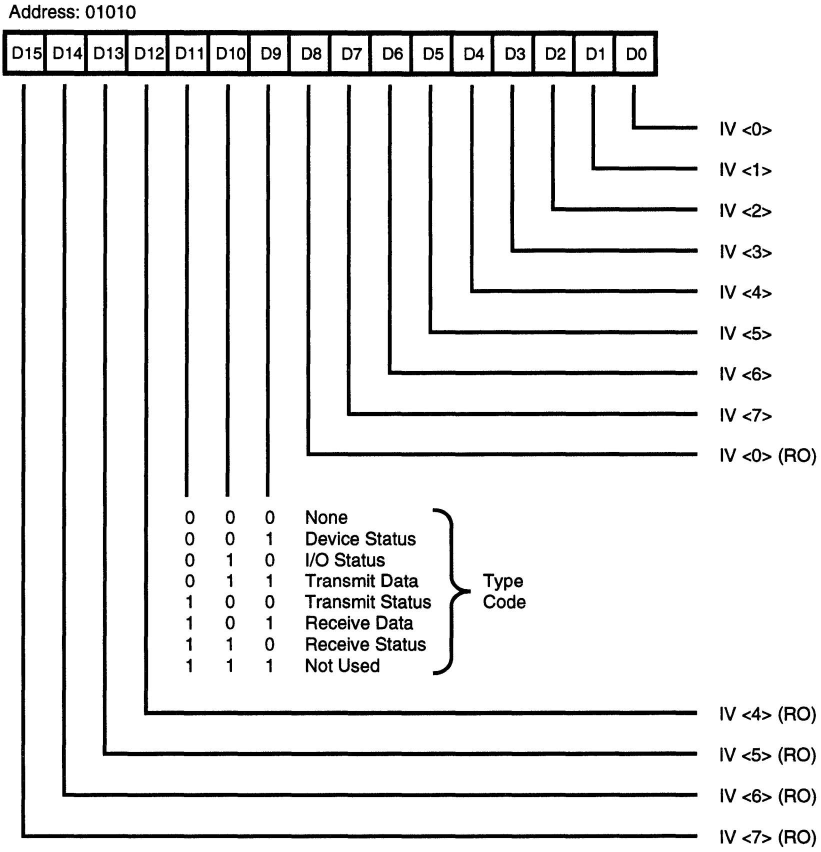
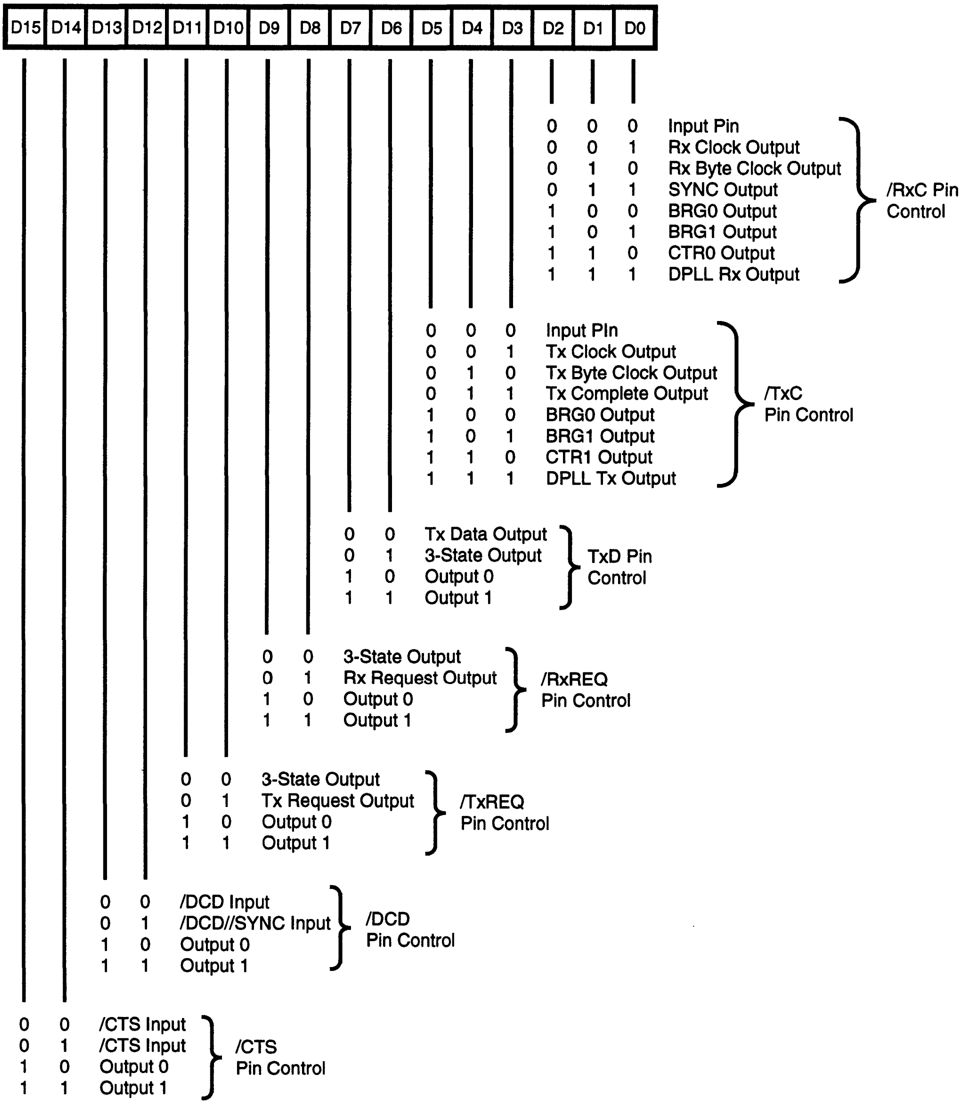


Figure 56. Interrupt Vector Register (IVR)

CONTROL REGISTERS (Continued)

Address: 01011



D2	D1	D0	/RxC Pin Control
0	0	0	Input Pin
0	0	1	Rx Clock Output
0	1	0	Rx Byte Clock Output
0	1	1	SYNC Output
1	0	0	BRG0 Output
1	0	1	BRG1 Output
1	1	0	CTR0 Output
1	1	1	DPLL Rx Output

D5	D4	D3	/TxC Pin Control
0	0	0	Input PIn
0	0	1	Tx Clock Output
0	1	0	Tx Byte Clock Output
0	1	1	Tx Complete Output
1	0	0	BRG0 Output
1	0	1	BRG1 Output
1	1	0	CTR1 Output
1	1	1	DPLL Tx Output

D7	D6	TxD Pin Control
0	0	Tx Data Output
0	1	3-State Output
1	0	Output 0
1	1	Output 1

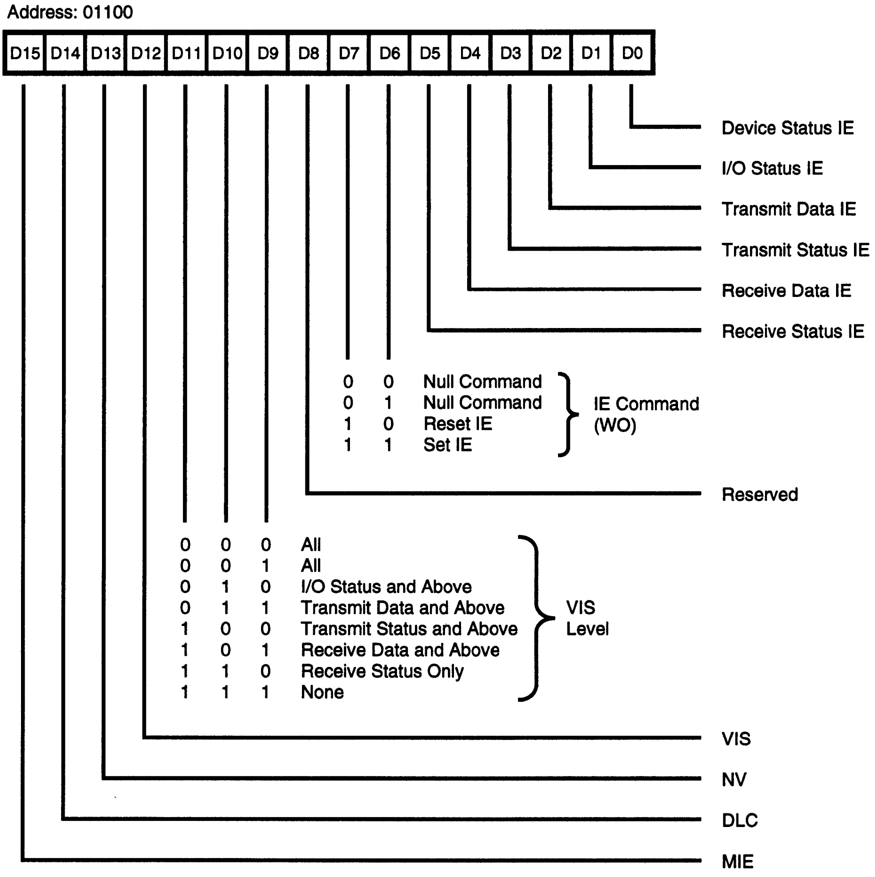
D9	D8	/RxREQ Pin Control
0	0	3-State Output
0	1	Rx Request Output
1	0	Output 0
1	1	Output 1

D11	D10	/TxREQ Pin Control
0	0	3-State Output
0	1	Tx Request Output
1	0	Output 0
1	1	Output 1

D13	D12	/DCD Pin Control
0	0	/DCD Input
0	1	/DCD//SYNC Input
1	0	Output 0
1	1	Output 1

D15	D14	/CTS Pin Control
0	0	/CTS Input
0	1	/CTS Input
1	0	Output 0
1	1	Output 1

Figure 57. I/O Control Register (IOCR)



D7	D6	
0	0	Null Command
0	1	Null Command
1	0	Reset IE
1	1	Set IE

D11	D10	D9	
0	0	0	All
0	0	1	All
0	1	0	I/O Status and Above
0	1	1	Transmit Data and Above
1	0	0	Transmit Status and Above
1	0	1	Receive Data and Above
1	1	0	Receive Status Only
1	1	1	None

Figure 58. Interrupt Control Register (ICR)

CONTROL REGISTERS (Continued)

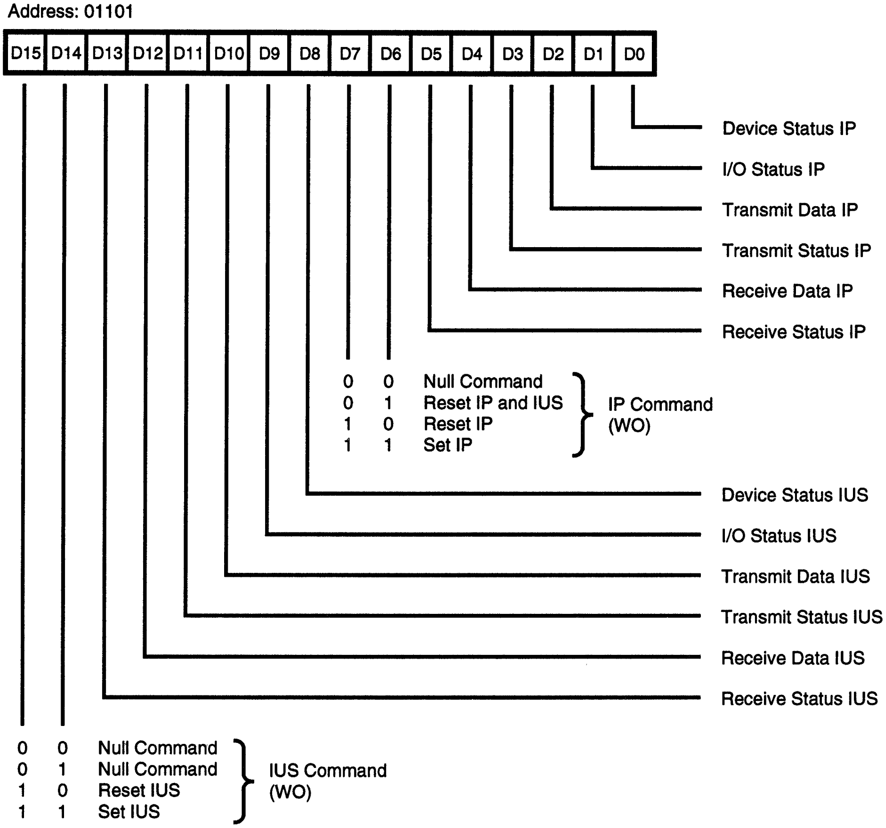
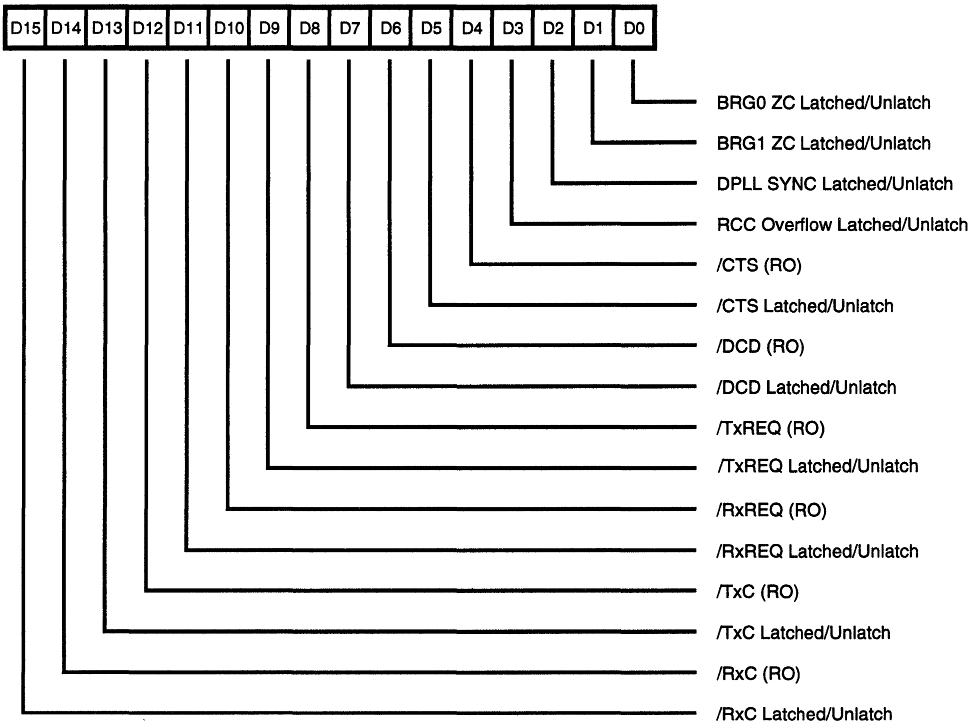


Figure 59. Daisy-Chain Control Register (DCCR)

Address: 01110



D0	BRG0 ZC Latched/Unlatch
D1	BRG1 ZC Latched/Unlatch
D2	DPLL SYNC Latched/Unlatch
D3	RCC Overflow Latched/Unlatch
D4	/CTS (RO)
D5	/CTS Latched/Unlatch
D6	/DCD (RO)
D7	/DCD Latched/Unlatch
D8	/TxREQ (RO)
D9	/TxREQ Latched/Unlatch
D10	/RxREQ (RO)
D11	/RxREQ Latched/Unlatch
D12	/TxC (RO)
D13	/TxC Latched/Unlatch
D14	/RxC (RO)
D15	/RxC Latched/Unlatch

Figure 60. Miscellaneous Interrupt Status Register (MISR)

CONTROL REGISTERS (Continued)

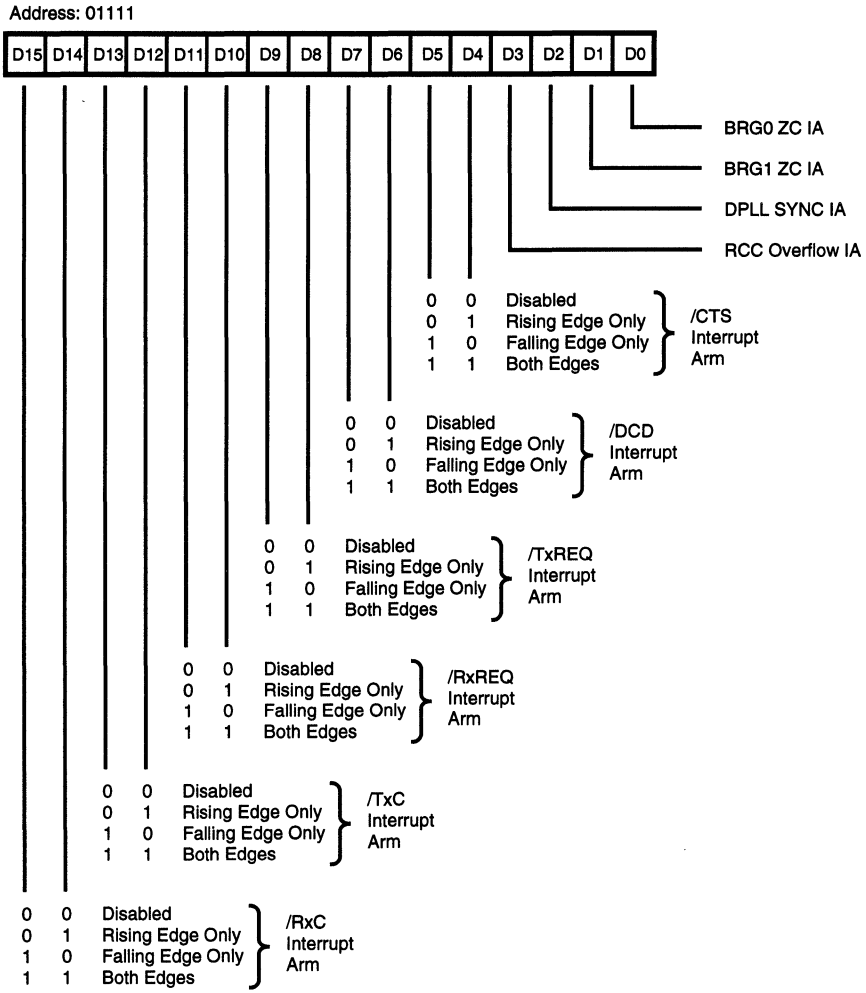


Figure 61. Status Interrupt Control Register (SICR)

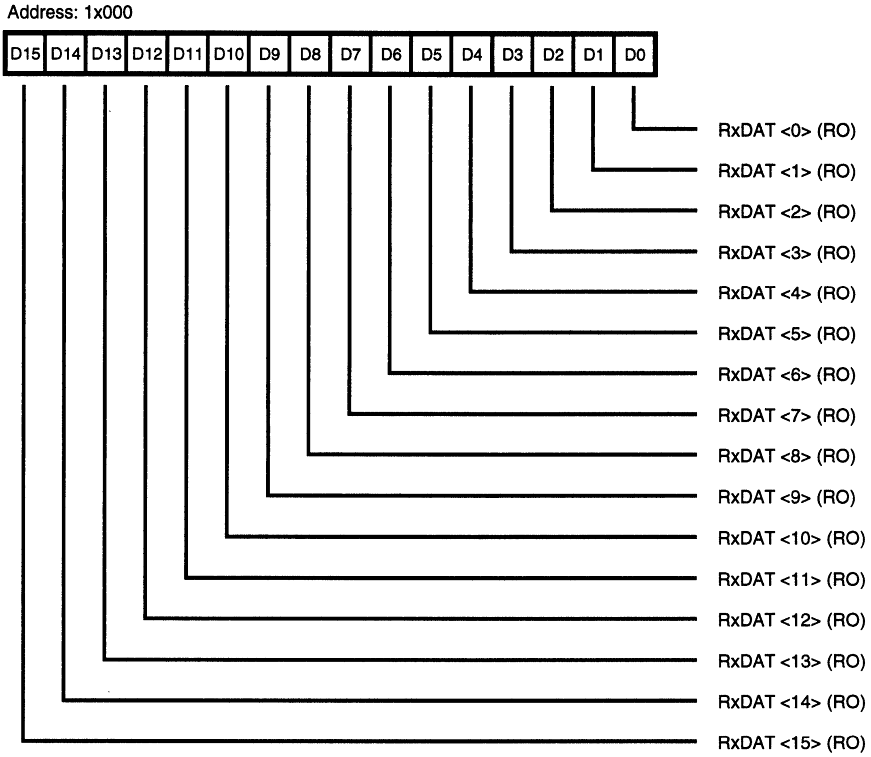
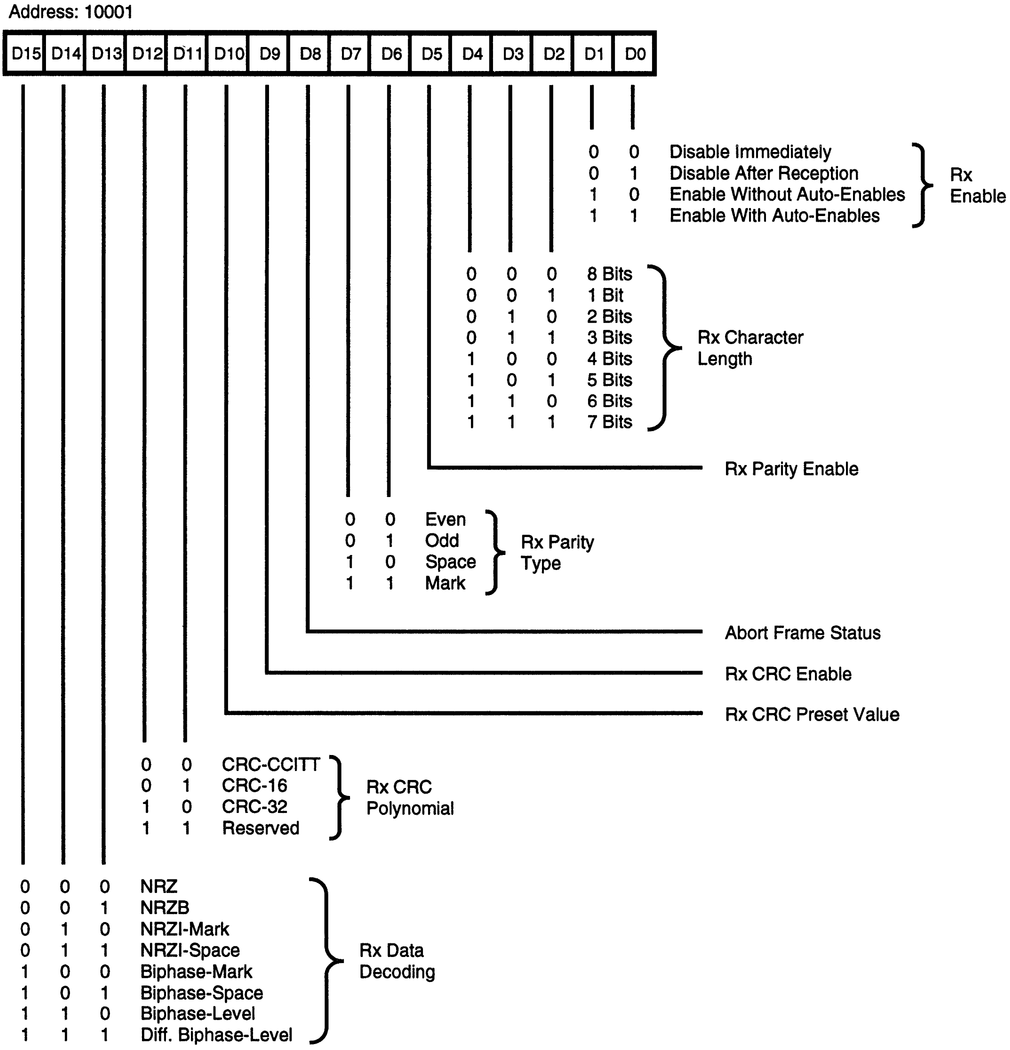


Figure 62. Receive Data Register (RDR)

CONTROL REGISTERS (Continued)



D1	D0	Rx Enable
0	0	Disable Immediately
0	1	Disable After Reception
1	0	Enable Without Auto-Enables
1	1	Enable With Auto-Enables

D4	D3	D2	Rx Character Length
0	0	0	8 Bits
0	0	1	1 Bit
0	1	0	2 Bits
0	1	1	3 Bits
1	0	0	4 Bits
1	0	1	5 Bits
1	1	0	6 Bits
1	1	1	7 Bits

D5: Rx Parity Enable

D7	D6	Rx Parity Type
0	0	Even
0	1	Odd
1	0	Space
1	1	Mark

D8: Abort Frame Status

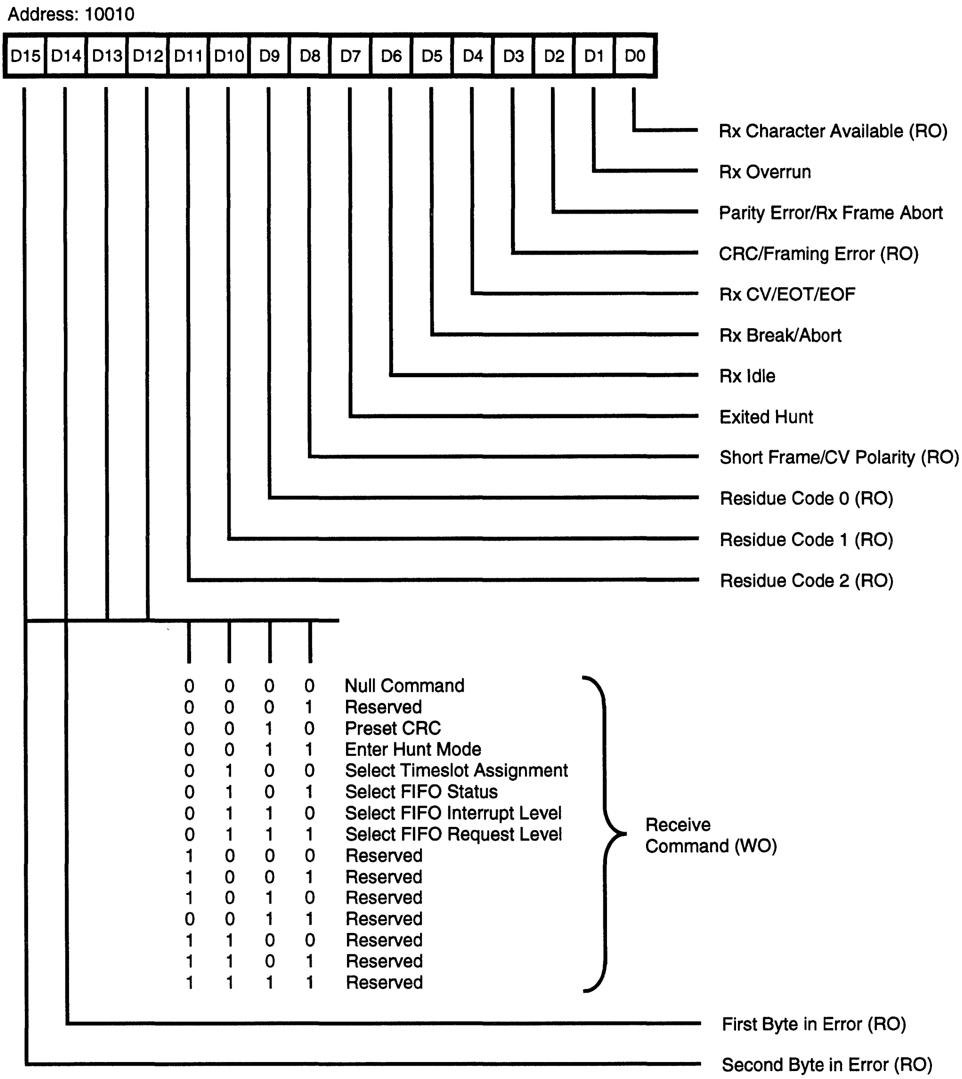
D9: Rx CRC Enable

D10: Rx CRC Preset Value

D12	D11	Rx CRC Polynomial
0	0	CRC-CCITT
0	1	CRC-16
1	0	CRC-32
1	1	Reserved

D15	D14	D13	Rx Data Decoding
0	0	0	NRZ
0	0	1	NRZB
0	1	0	NRZI-Mark
0	1	1	NRZI-Space
1	0	0	Biphase-Mark
1	0	1	Biphase-Space
1	1	0	Biphase-Level
1	1	1	Diff. Biphase-Level

Figure 63. Receive Mode Register (RMR)



0	0	0	0	Null Command
0	0	0	1	Reserved
0	0	1	0	Preset CRC
0	0	1	1	Enter Hunt Mode
0	1	0	0	Select Timeslot Assignment
0	1	0	1	Select FIFO Status
0	1	1	0	Select FIFO Interrupt Level
0	1	1	1	Select FIFO Request Level
1	0	0	0	Reserved
1	0	0	1	Reserved
1	0	1	0	Reserved
0	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	1	Reserved

Figure 64. Receive Command Status Register (RCSR)

CONTROL REGISTERS (Continued)

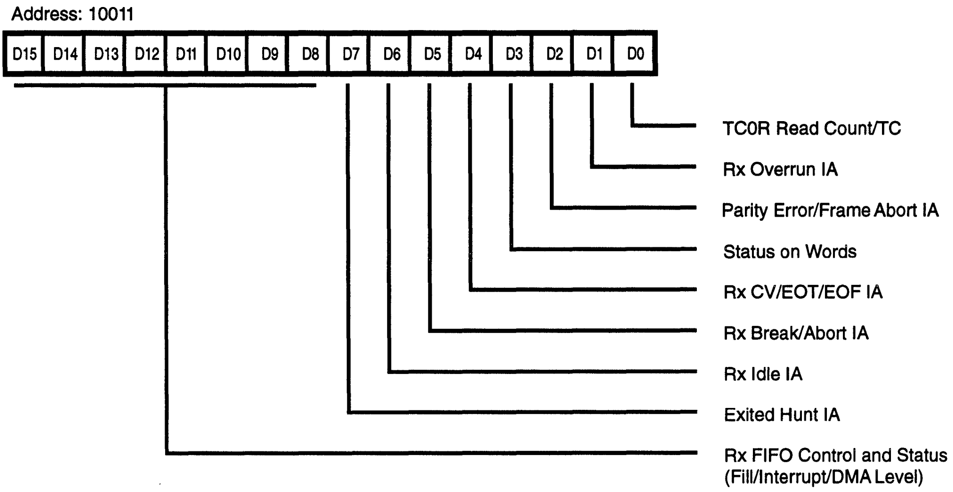


Figure 65a. Receive Interrupt Control Register (RICR)

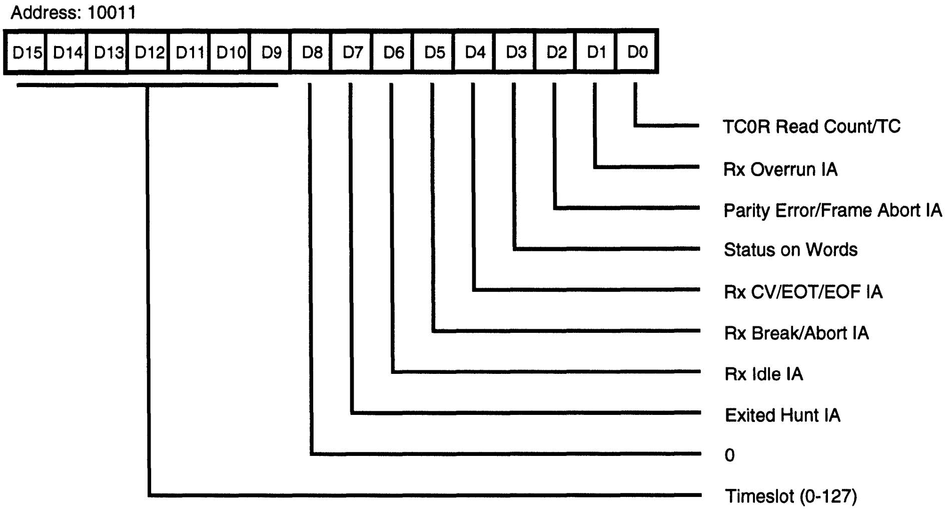
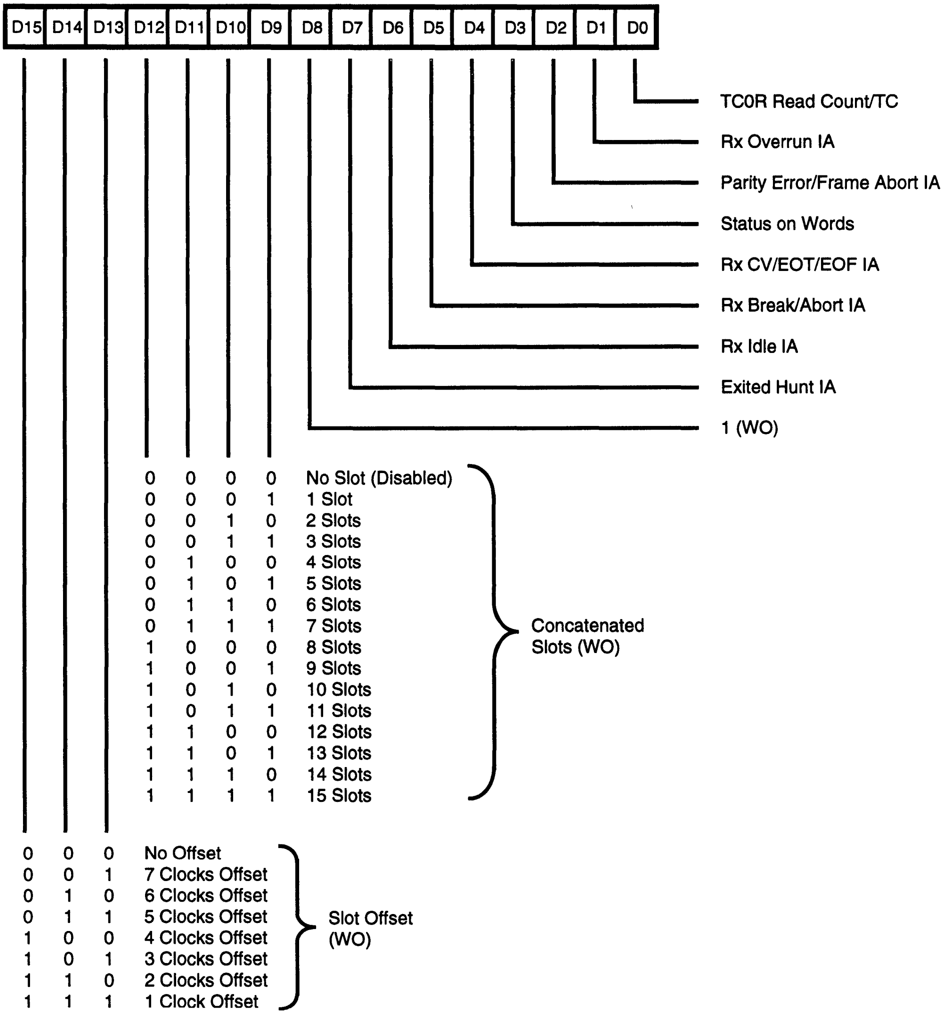


Figure 65b. Receive Interrupt Control Register (RICR)

Address: 10011



Bit	Function
D0	TCOR Read Count/TC
D1	Rx Overrun IA
D2	Parity Error/Frame Abort IA
D3	Status on Words
D4	Rx CV/EOT/EOF IA
D5	Rx Break/Abort IA
D6	Rx Idle IA
D7	Exited Hunt IA
D8	1 (WO)

Concatenated Slots (WO)

D12	D11	D10	D9	
0	0	0	0	No Slot (Disabled)
0	0	0	1	1 Slot
0	0	1	0	2 Slots
0	0	1	1	3 Slots
0	1	0	0	4 Slots
0	1	0	1	5 Slots
0	1	1	0	6 Slots
0	1	1	1	7 Slots
1	0	0	0	8 Slots
1	0	0	1	9 Slots
1	0	1	0	10 Slots
1	0	1	1	11 Slots
1	1	0	0	12 Slots
1	1	0	1	13 Slots
1	1	1	0	14 Slots
1	1	1	1	15 Slots

Slot Offset (WO)

D15	D14	D13	
0	0	0	No Offset
0	0	1	7 Clocks Offset
0	1	0	6 Clocks Offset
0	1	1	5 Clocks Offset
1	0	0	4 Clocks Offset
1	0	1	3 Clocks Offset
1	1	0	2 Clocks Offset
1	1	1	1 Clock Offset

Figure 65c. Receive Interrupt Control Register (RICR)

CONTROL REGISTERS (Continued)

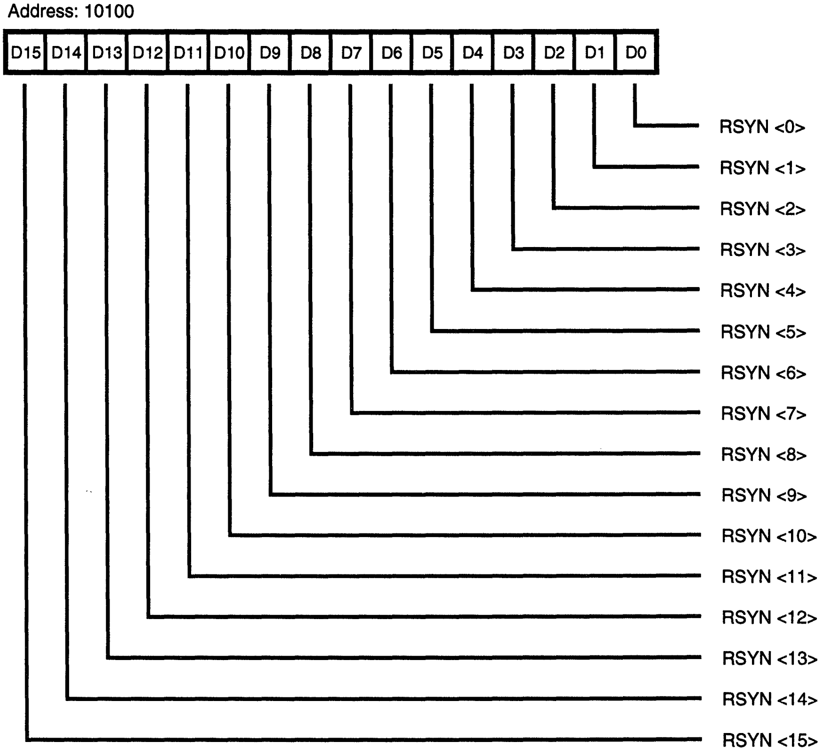


Figure 66. Receive Sync Register (RSR)

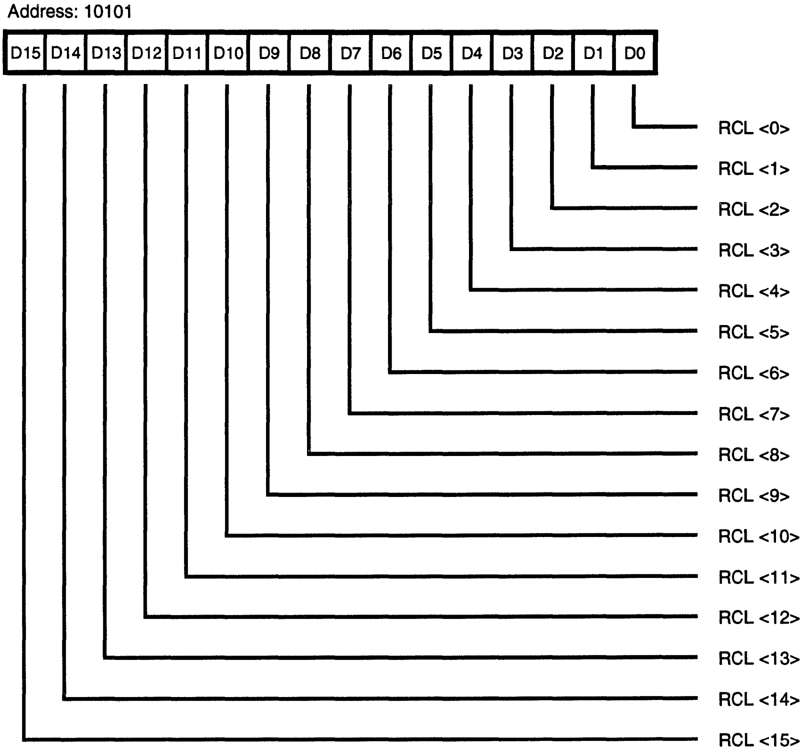


Figure 67. Receive Count Limit Register (RCLR)

CONTROL REGISTERS (Continued)

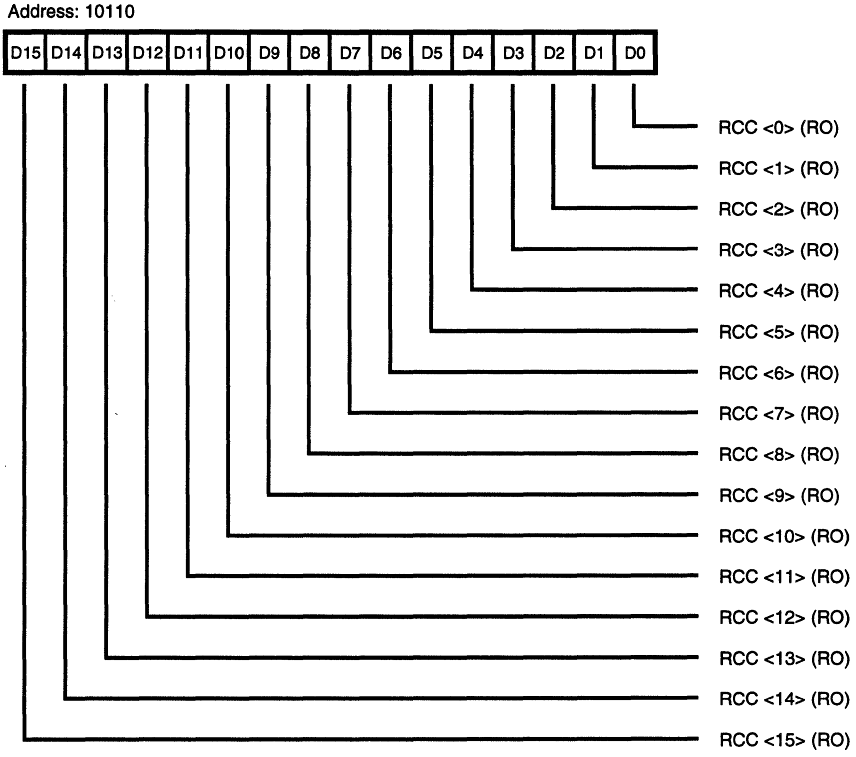


Figure 68. Receive Character Count Register (RCCR)

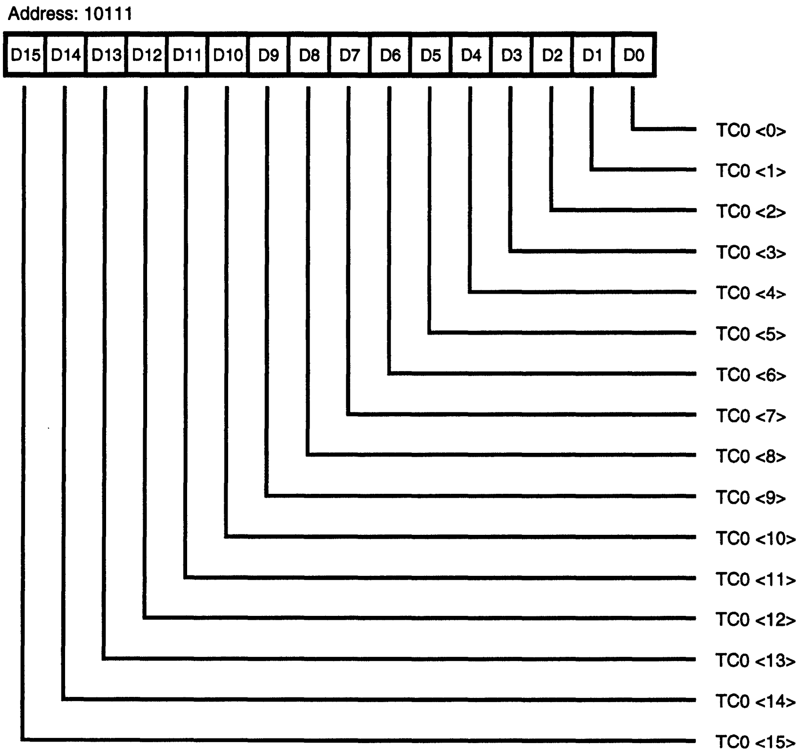


Figure 69. Time Constant 0 Register (TC0R)

CONTROL REGISTERS (Continued)

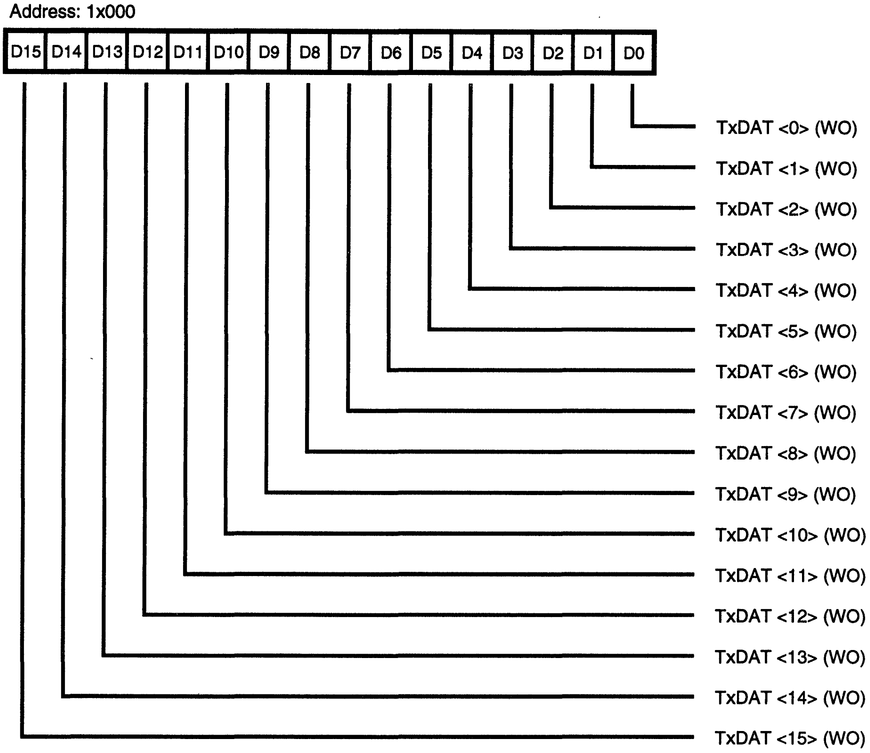


Figure 70. Transmit Data Register (TDR)

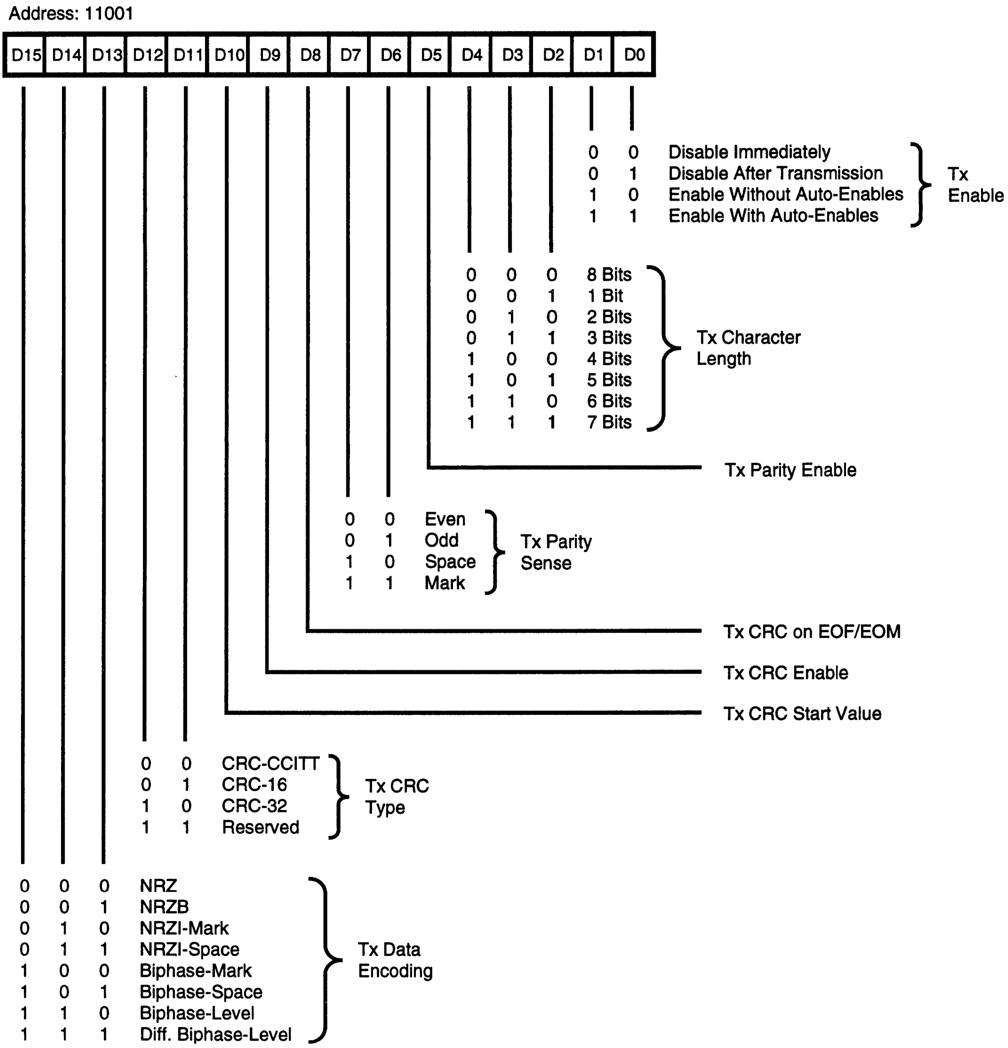
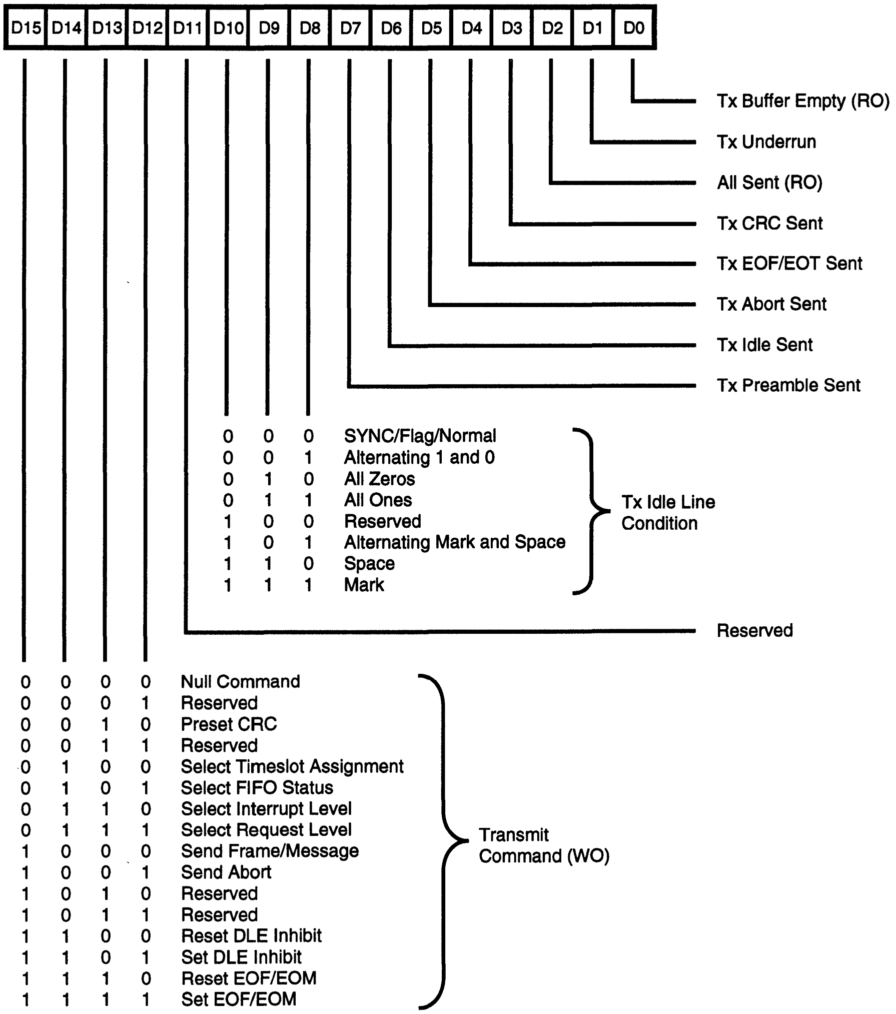


Figure 71. Transmit Mode Register (TMR)

CONTROL REGISTERS (Continued)

Address: 11010



D10	D9	D8	Tx Idle Line Condition
0	0	0	SYNC/Flag/Normal
0	0	1	Alternating 1 and 0
0	1	0	All Zeros
0	1	1	All Ones
1	0	0	Reserved
1	0	1	Alternating Mark and Space
1	1	0	Space
1	1	1	Mark

D15	D14	D13	D12	Transmit Command (WO)
0	0	0	0	Null Command
0	0	0	1	Reserved
0	0	1	0	Preset CRC
0	0	1	1	Reserved
0	1	0	0	Select Timeslot Assignment
0	1	0	1	Select FIFO Status
0	1	1	0	Select Interrupt Level
0	1	1	1	Select Request Level
1	0	0	0	Send Frame/Message
1	0	0	1	Send Abort
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reset DLE Inhibit
1	1	0	1	Set DLE Inhibit
1	1	1	0	Reset EOF/EOM
1	1	1	1	Set EOF/EOM

Figure 72. Transmit Command/Status Register (TCSR)

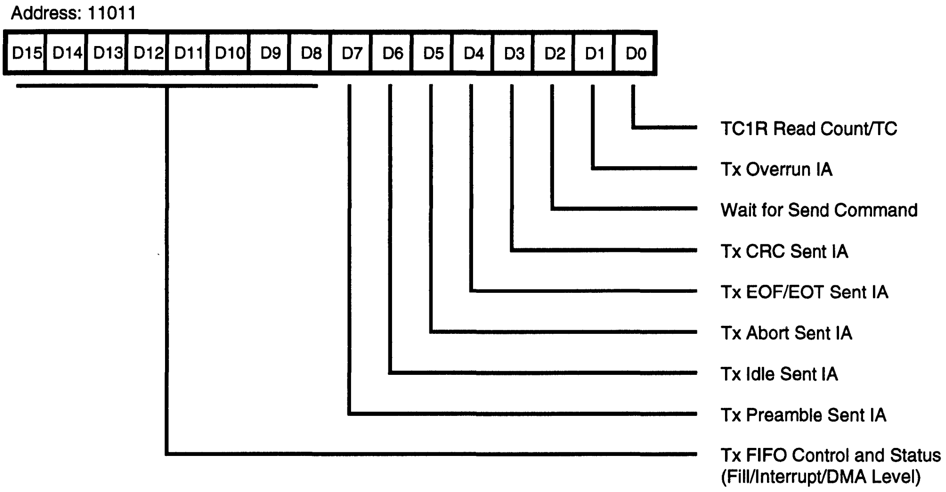


Figure 73a. Transmit Interrupt Control Register (TICR)

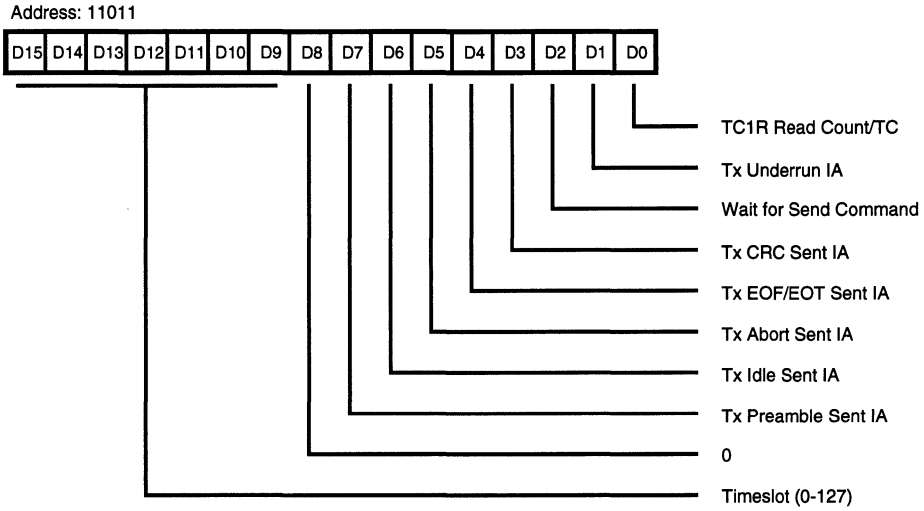
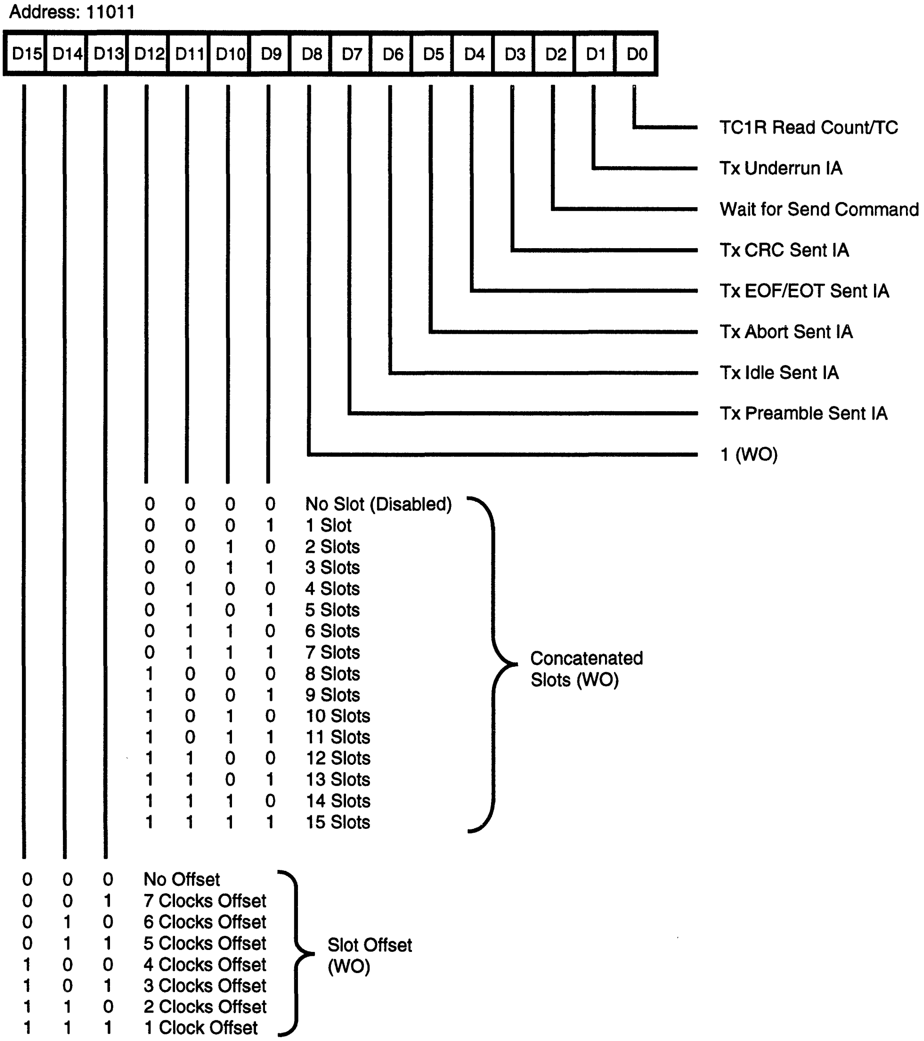


Figure 73b. Transmit Interrupt Control Register (TICR)

CONTROL REGISTERS (Continued)



D12	D11	D10	D9	Concatenated Slots (WO)
0	0	0	0	No Slot (Disabled)
0	0	0	1	1 Slot
0	0	1	0	2 Slots
0	0	1	1	3 Slots
0	1	0	0	4 Slots
0	1	0	1	5 Slots
0	1	1	0	6 Slots
0	1	1	1	7 Slots
1	0	0	0	8 Slots
1	0	0	1	9 Slots
1	0	1	0	10 Slots
1	0	1	1	11 Slots
1	1	0	0	12 Slots
1	1	0	1	13 Slots
1	1	1	0	14 Slots
1	1	1	1	15 Slots

D15	D14	D13	Slot Offset (WO)
0	0	0	No Offset
0	0	1	7 Clocks Offset
0	1	0	6 Clocks Offset
0	1	1	5 Clocks Offset
1	0	0	4 Clocks Offset
1	0	1	3 Clocks Offset
1	1	0	2 Clocks Offset
1	1	1	1 Clock Offset

Figure 73c. Transmit Interrupt Control Register (TICR)

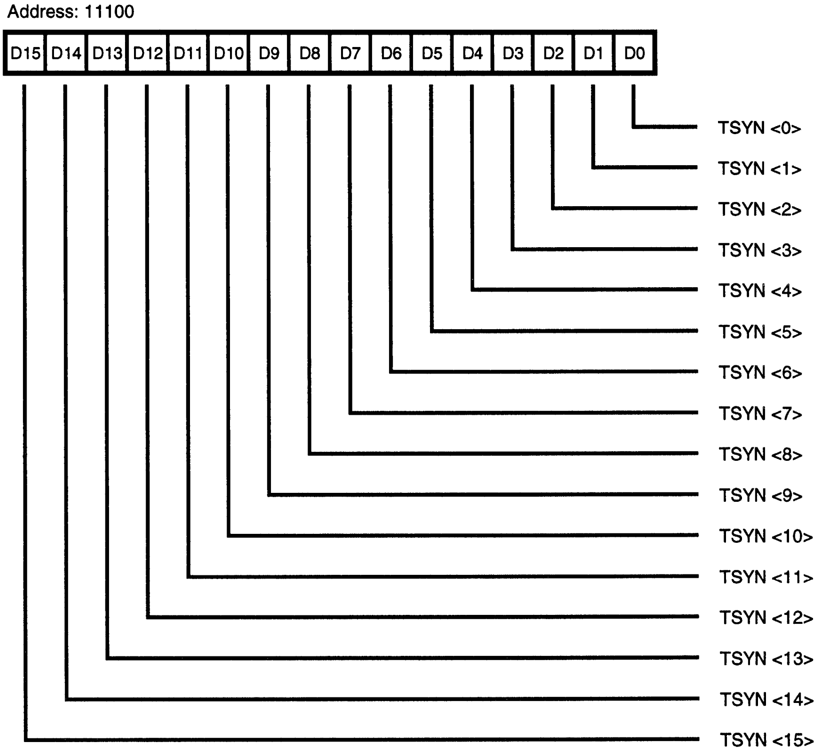


Figure 74. Transmit Sync Register (TSR)

CONTROL REGISTERS (Continued)

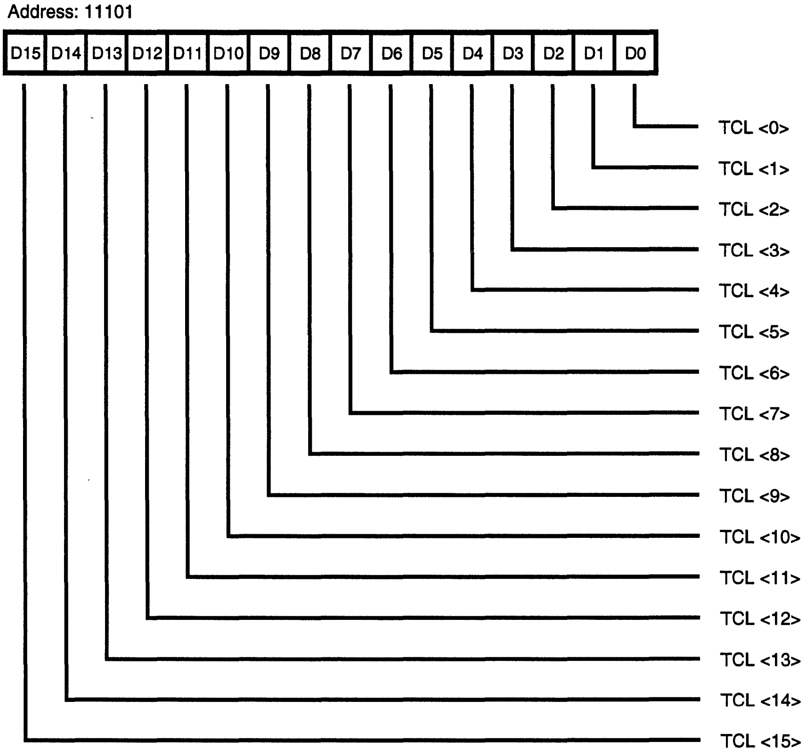


Figure 75. Transmit Count Limit Register (TCLR)

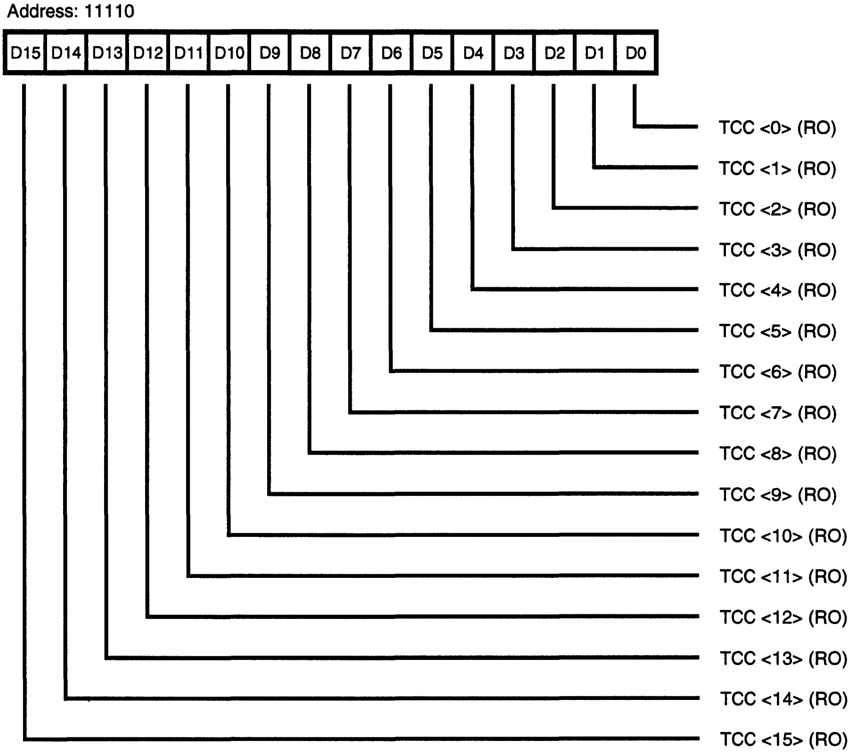


Figure 76. Transmit Character Count Register (TCCR)

CONTROL REGISTERS (Continued)

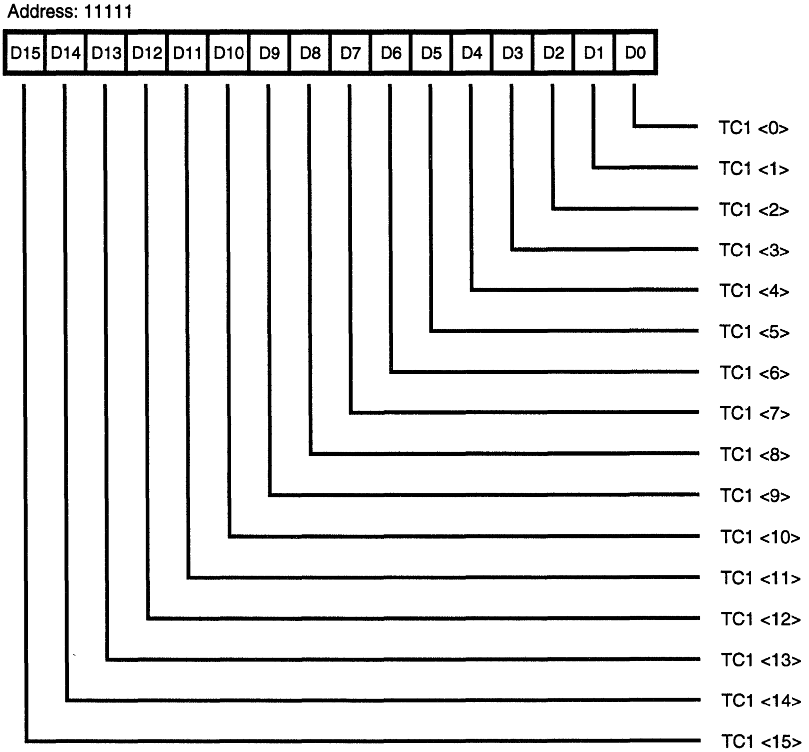
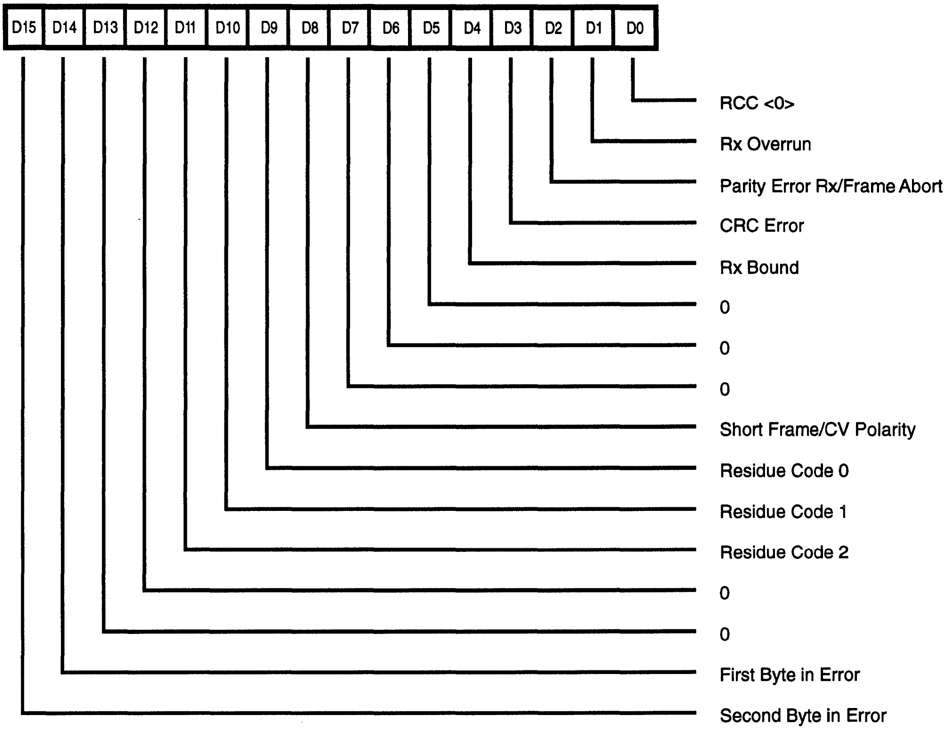


Figure 77. Time Constant 1 Register (TC1R)

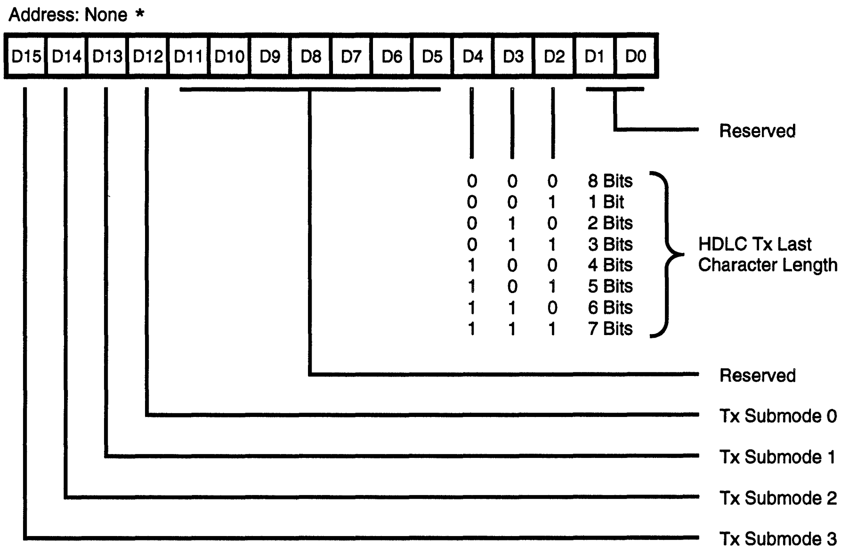
Address: None *



* Refer to Figure 22 (Channel Control Register)
Bits 6-7 for Access Method

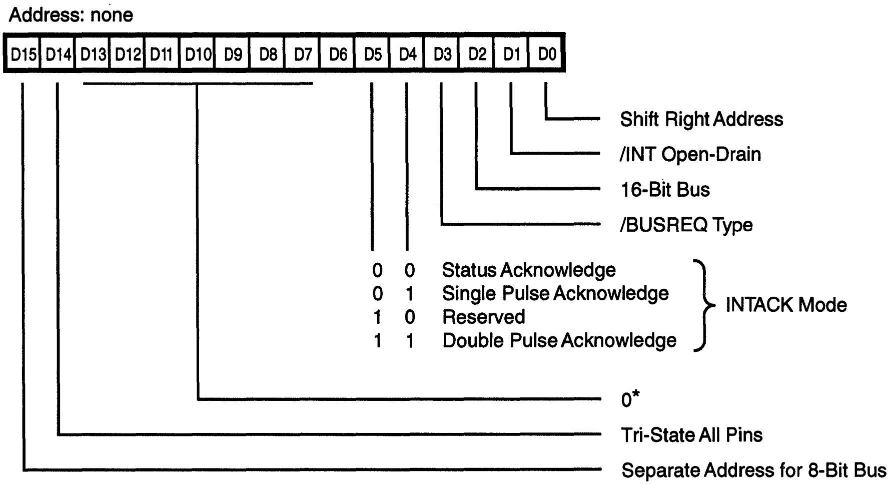
Figure 78. Receive Status Block Register (RSBR)

CONTROL REGISTERS (Continued)



* Refer to Figure 22 (Channel Control Register) Bits15-14 for Access Method

Figure 79. Transmit Status Block Register (TSBR)


Address: none
D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
Shift Right Address
/INT Open-Drain
16-Bit Bus
/BUSREQ Type
0 0 Status Acknowledge
0 1 Single Pulse Acknowledge
1 0 Reserved
1 1 Double Pulse Acknowledge
INTACK Mode
0*
Tri-State All Pins
Separate Address for 8-Bit Bus

* Must be programmed as 0.

Figure 80. Bus Configuration Register (BCR)

ABSOLUTE MAXIMUM RATINGS

Symbol	Description	Min	Max	Units
V_{CC}	Supply Voltage (*)	–0.3	+7.0	V
T_{STG}	Storage Temp	–65°	+150°	C
T_A	Oper Ambient Temp		†	C
	Power Dissipation		2.2	W

Notes:
* Voltage on all pins with respect to GND.
† See Ordering Information.

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

STANDARD TEST CONDITIONS

The DC Characteristics and Capacitance Section below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to Ground. Positive current flows into the referenced pin (Figure 81). Standard conditions are as follows:

- +4.5 V < V_{CC} < +5.5 V
- GND = 0 V
- T_A as specified in Ordering Information

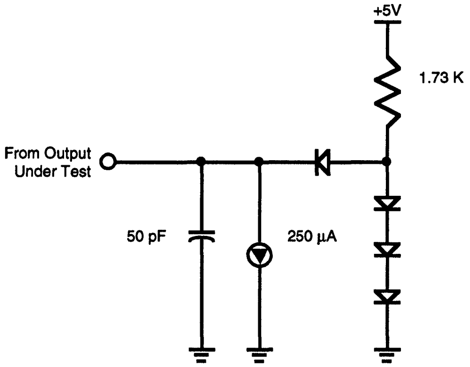


Figure 81. Standard Test Load

CAPACITANCE

Symbol	Parameter	Min	Max	Unit	Condition
C_{IN}	Input Capacitance		10	pF	*
C_{OUT}	Output Capacitance		15	pF	*
$C_{I/O}$	Bidirectional Capacitance		20	pF	

Notes:
F = 1 MHz, over specified temperature range.
* Unmeasured pins returned to Ground.

MISCELLANEOUS

Transistor Count - 100,000

TEMPERATURE RANGE

Standard = 0°C to 70°C

IUSC TIMING

The IUSC interface timing is similar to that found on a static RAM, except that it is much more flexible. Up to four separate timing strobe signals are present on the interface: /DS, /RD, /WR and /INTACK. Only one of these timing strobes is active at any time. Should the external logic activate more than one of these strobes at the same time, the IUSC will enter a pre-reset state. This state is only exited by a hardware reset. Do not allow overlap of timing strobes. The timing diagrams, beginning on the next page, illustrate the different bus transactions possible with the necessary setup, hold, and delay times. IUSC Timing diagrams are shown from Figure 82 through Figure 106.

DC CHARACTERISTICS

Symbol	Parameter	Min	Typ	Max	Unit	Condition
V_{IH}	Input High Voltage	2.2		V_{CC} +0.3	V	
V_{IL}	Input Low Voltage	–0.3		0.8	V	
V_{OH1}	Output High Voltage	2.4			V	I_{OH} = –1.6 mA
V_{OH2}	Output High Voltage	V_{CC}–0.8			V	I_{OH} = –250 µA
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = +2.0 mA
I_{IL}	Input Leakage			+10.00	µA	0.4 < V_{IN} < +2.4V
I_{OL}	Output Leakage			+10.00	µA	0.4 < V_{OUT} < +2.4V
I_{CC1}	V_{CC} Supply Current		7	50	mA	V_{CC} = 5V V_{IH} = 4.8V V_{IL} = 0.2V

Note:
V_{CC} = 5V ± 10% unless otherwise specified, over specified temperature range.

AC CHARACTERISTICS
Timing Diagrams (Figures 82-104)

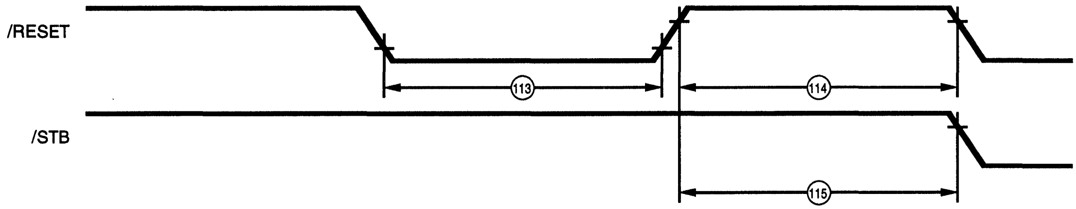


Figure 82. Reset Timing

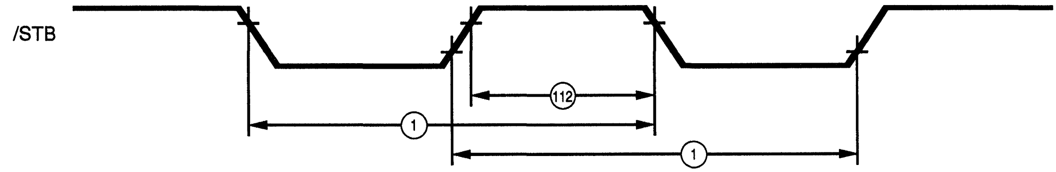


Figure 83. Bus Cycle Timing

Note:
/STB is any of the following: /DS, /RD, /WR or Pulsed /INTACK.

AC CHARACTERISTICS
Timing Diagrams (Continued)

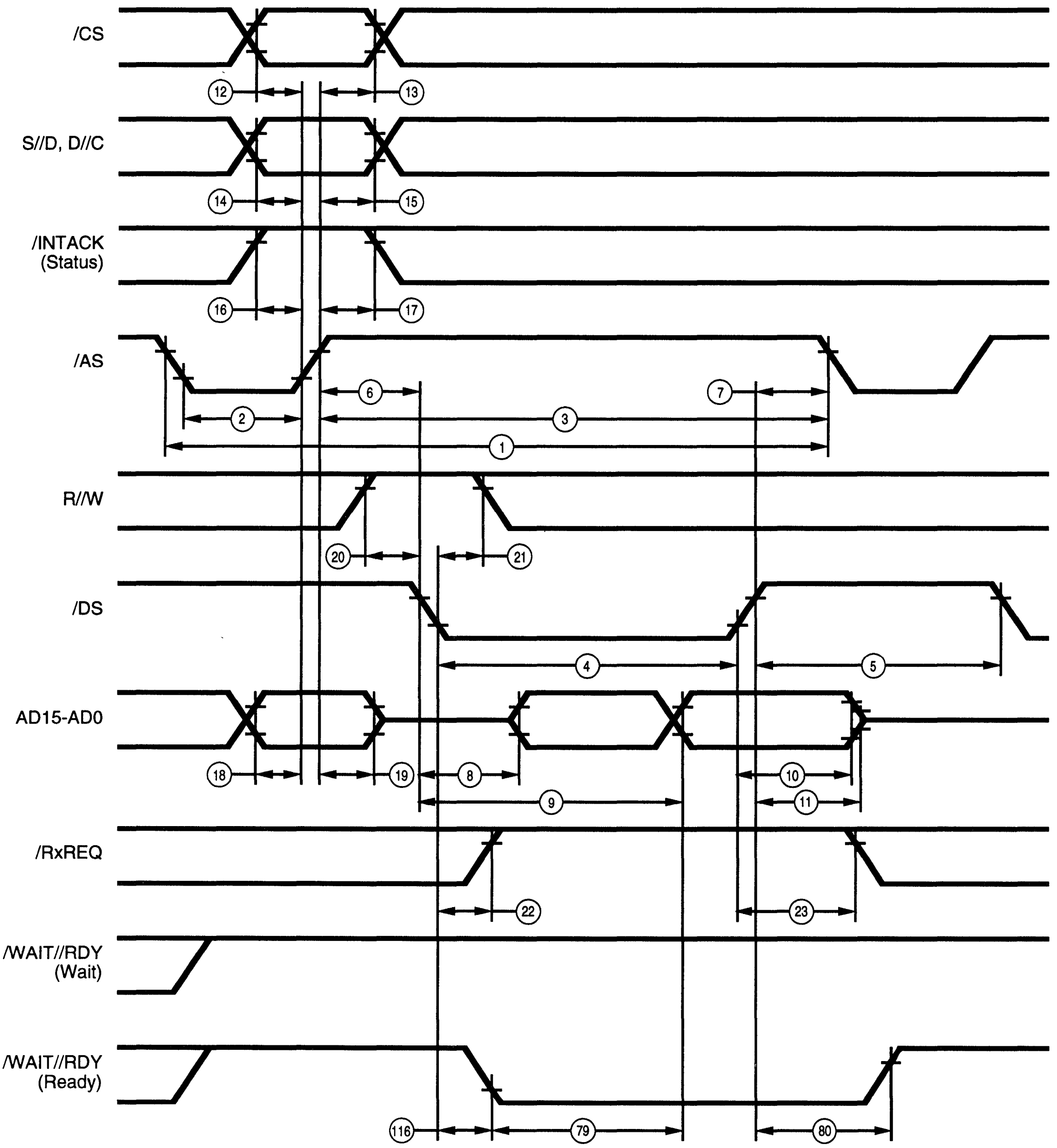


Figure 84. Multiplexed /DS Read Cycle

AC CHARACTERISTICS
Timing Diagrams (Continued)

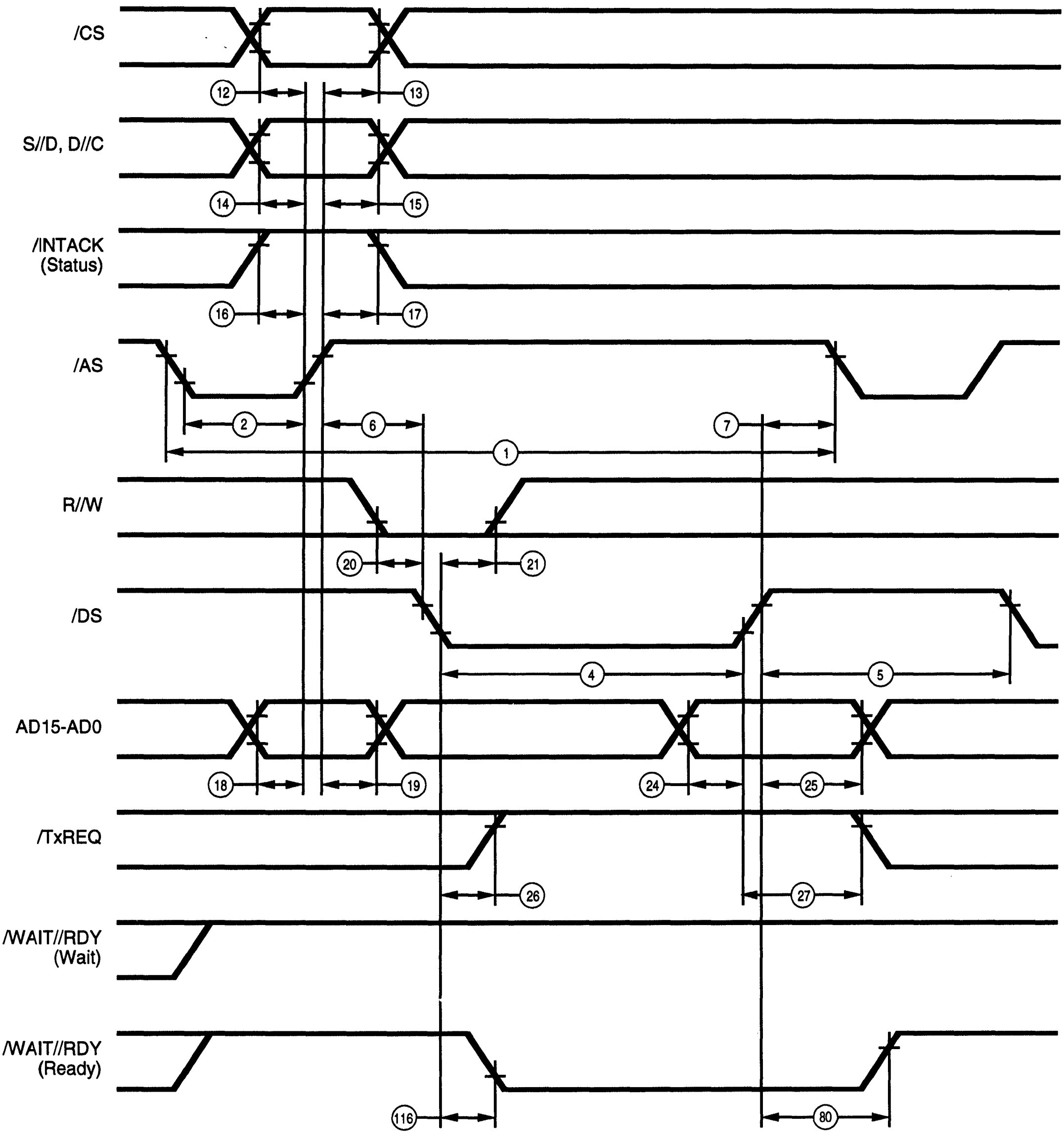


Figure 85. Multiplexed /DS Write Cycle

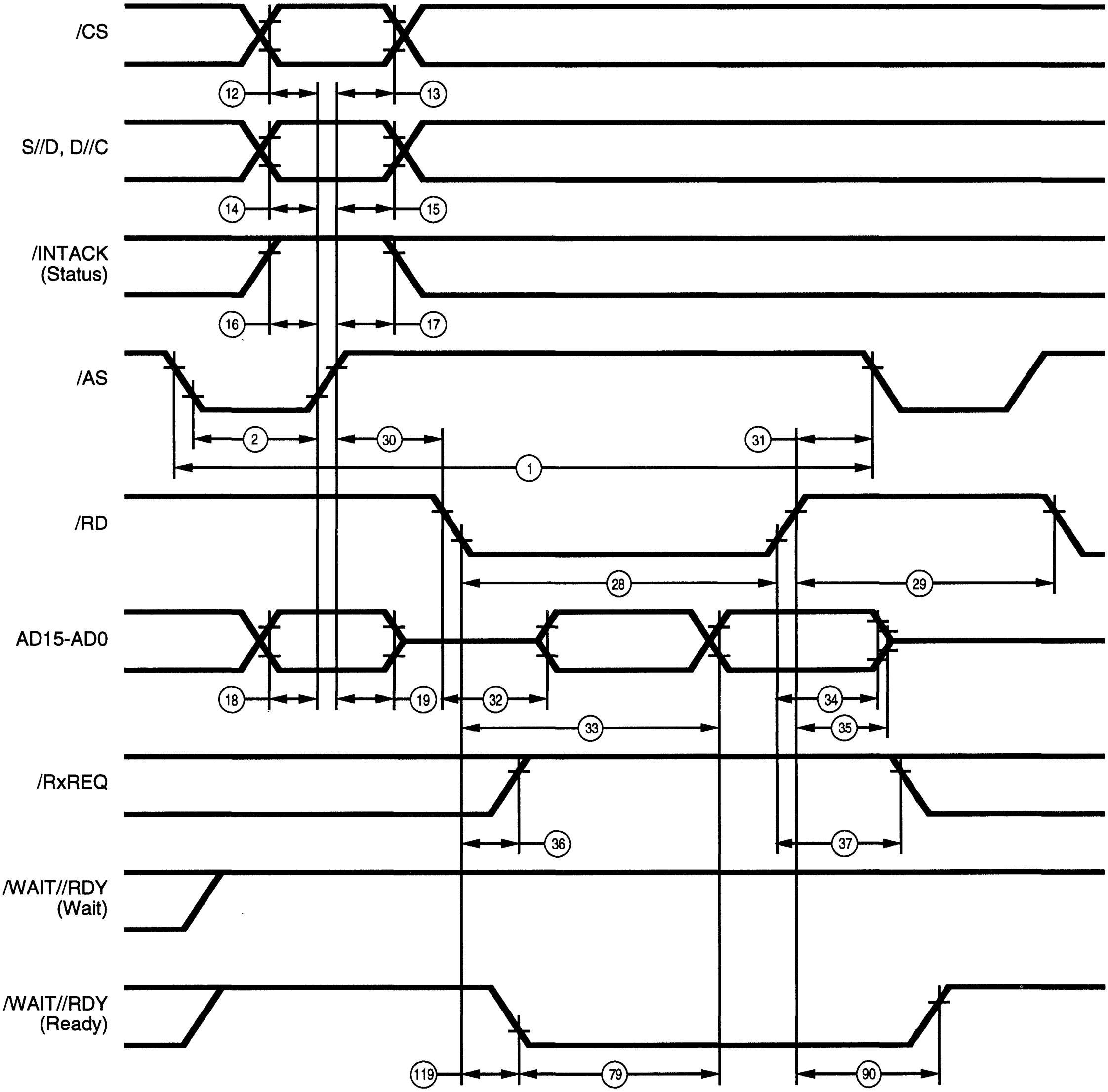


Figure 86. Multiplexed /RD Read Cycle

AC CHARACTERISTICS

Timing Diagrams (Continued)

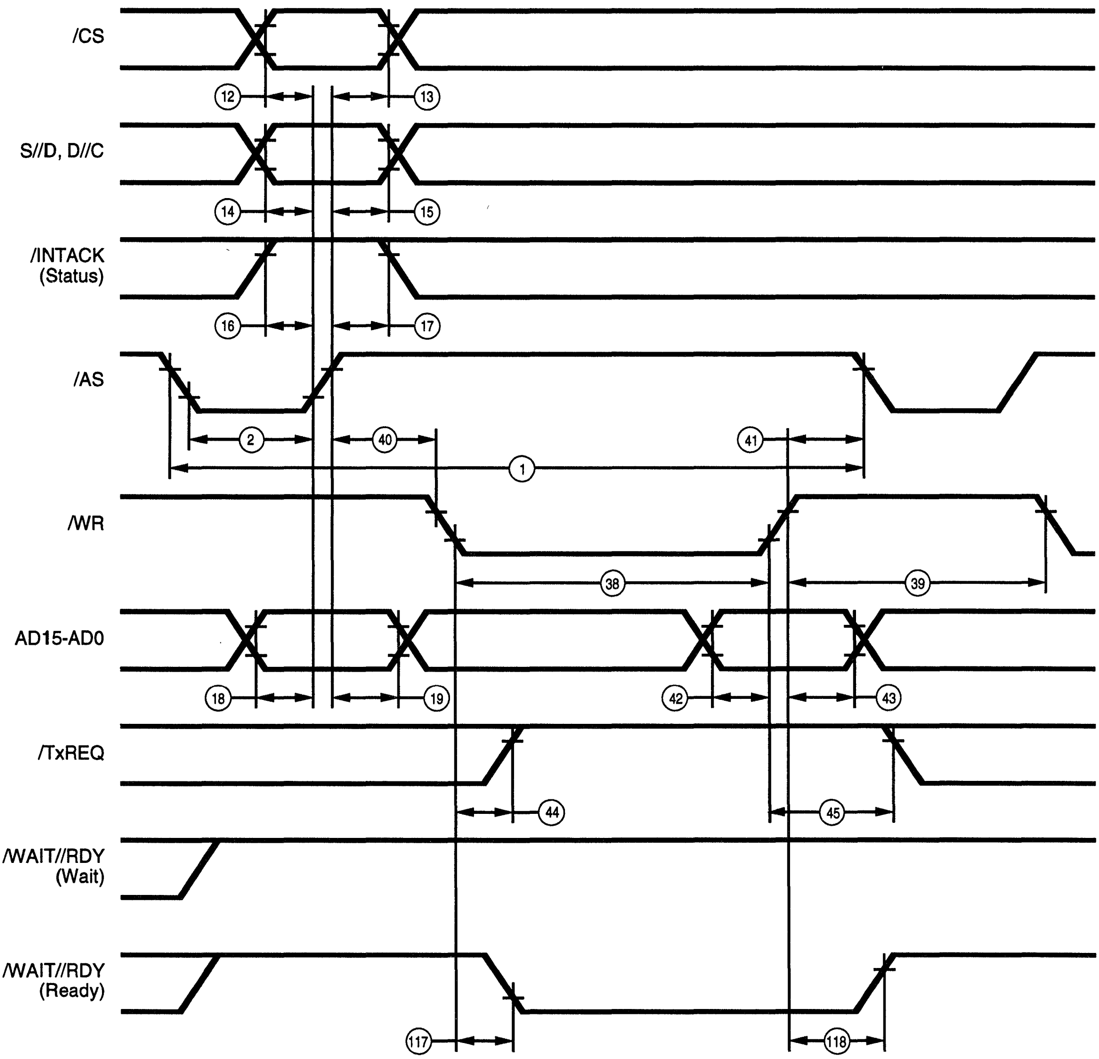


Figure 87. Multiplexed /WR Write Cycle

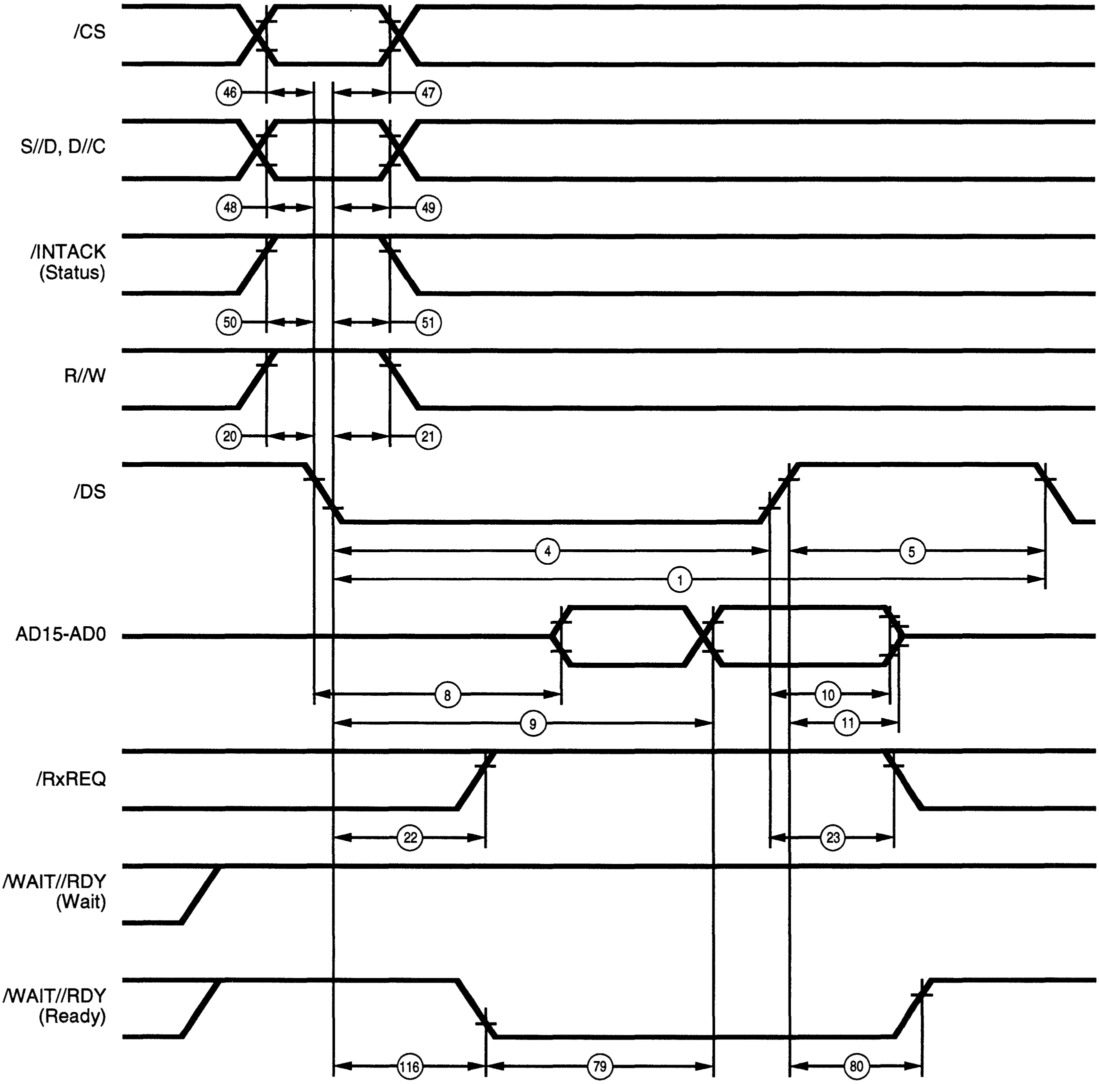


Figure 88. Non-Multiplexed /DS Read Cycle

AC CHARACTERISTICS
Timing Diagrams (Continued)

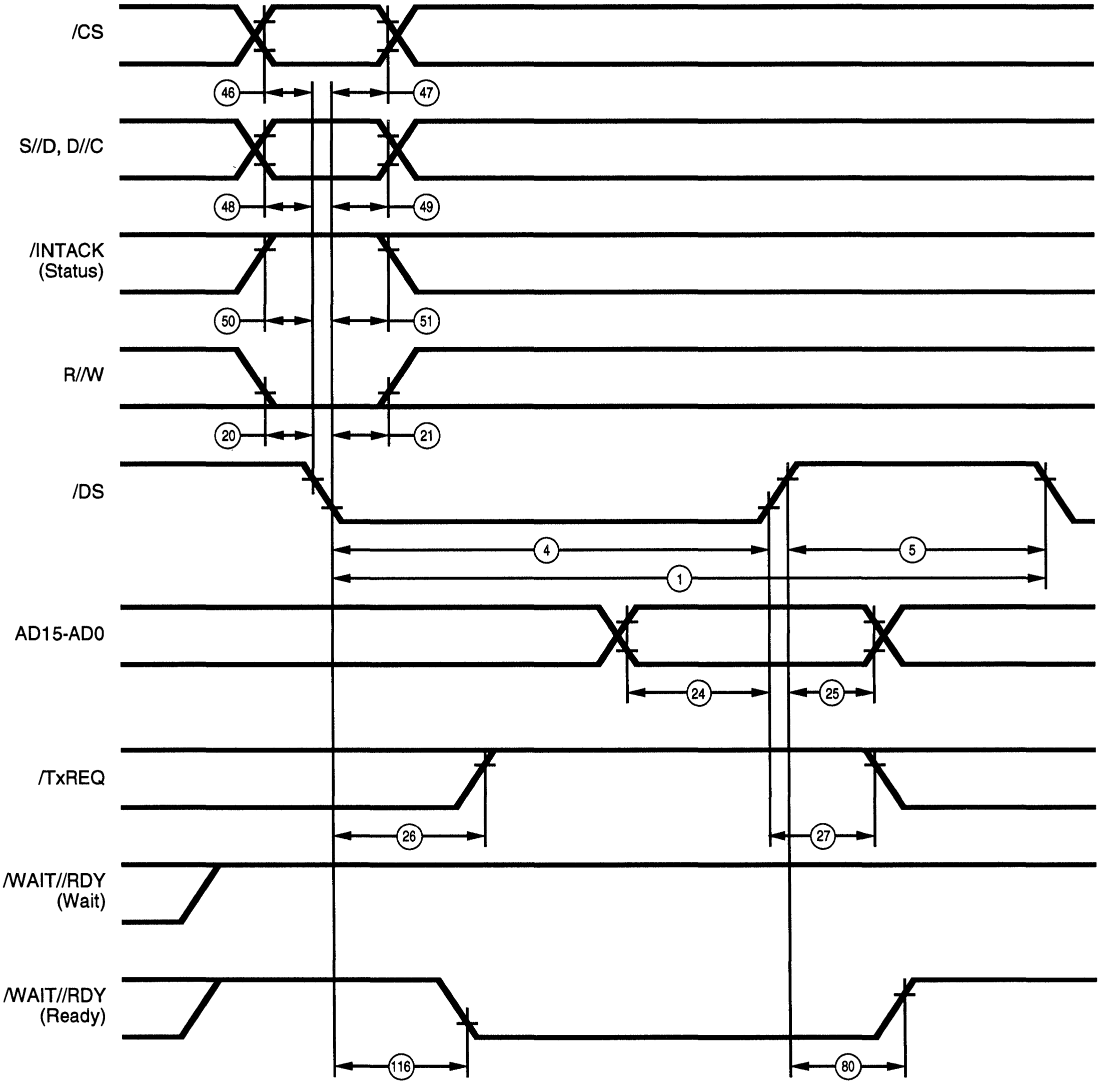


Figure 89. Non-Multiplexed /DS Write Cycle

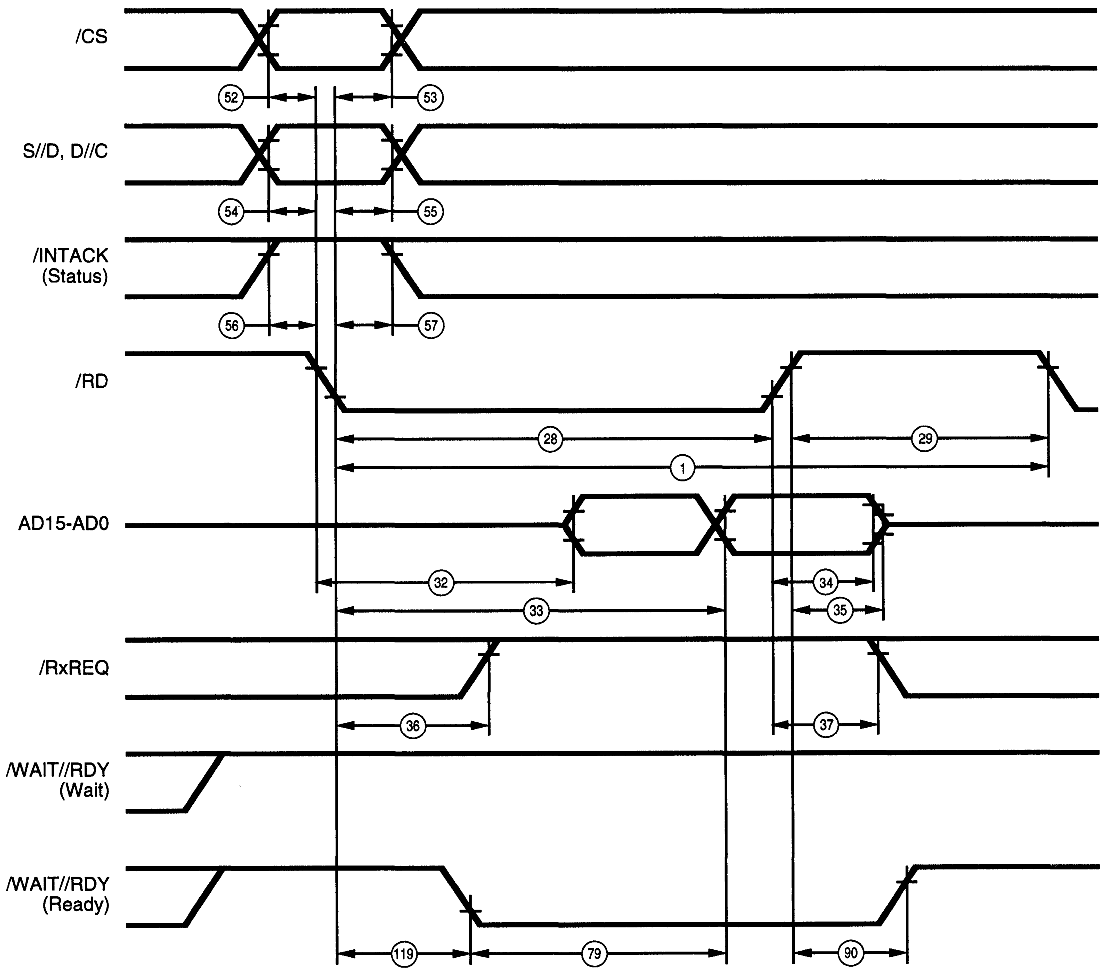


Figure 90. Non-Multiplexed /RD Read Cycle

AC CHARACTERISTICS
Timing Diagrams (Continued)

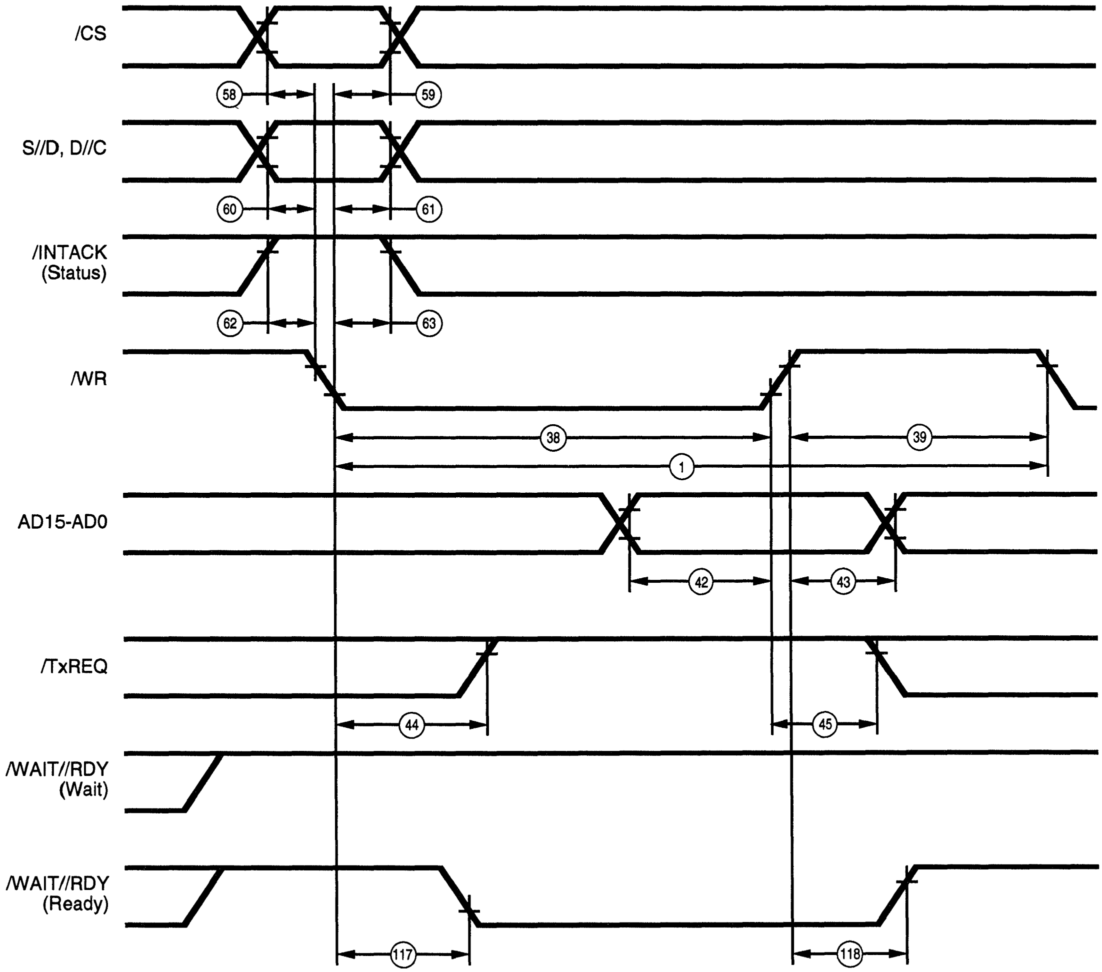


Figure 91. Non-Multiplexed /WR Write Cycle

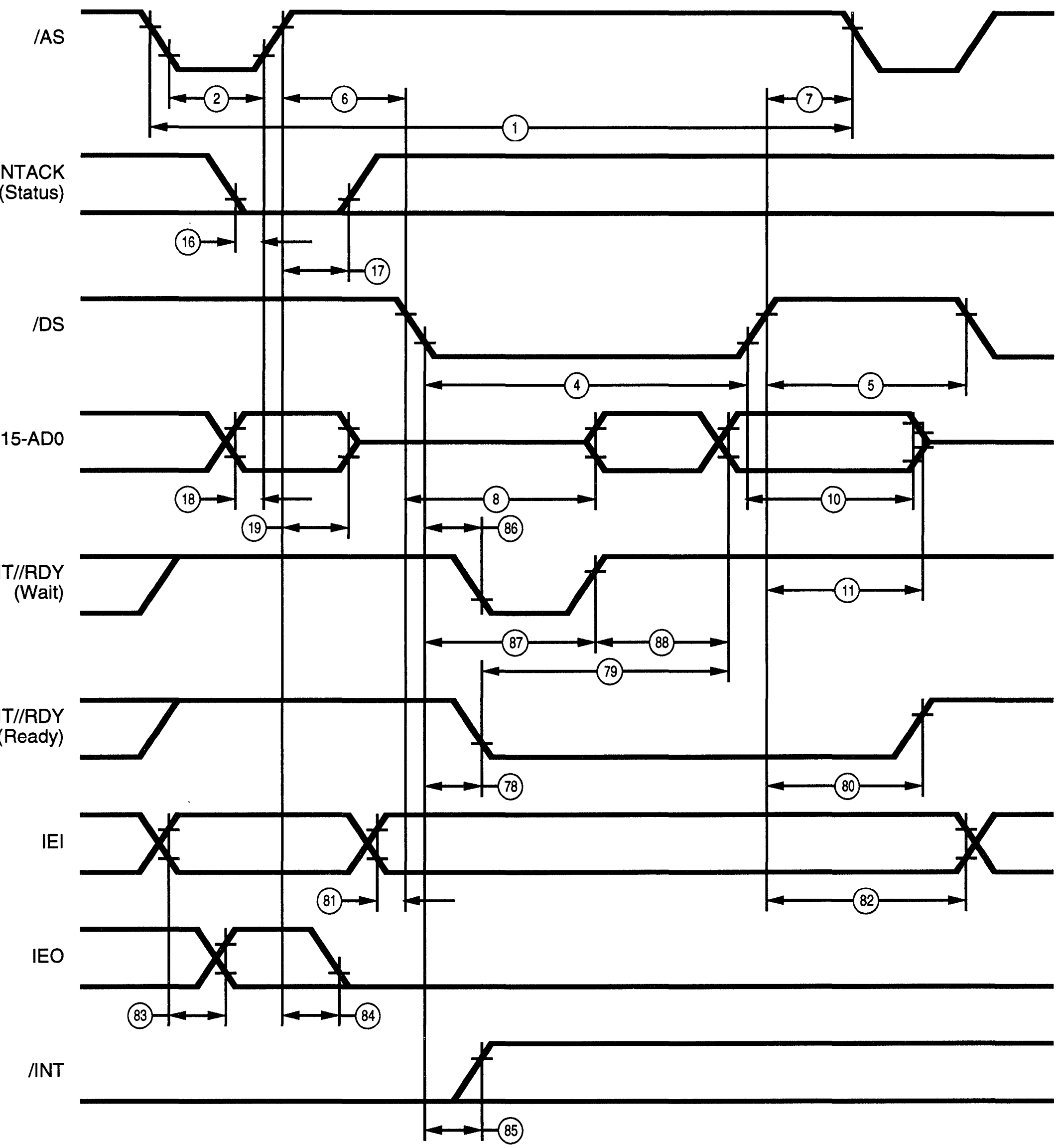


Figure 92. Multiplexed /DS Interrupt Acknowledge Cycle

AC CHARACTERISTICS
Timing Diagrams (Continued)

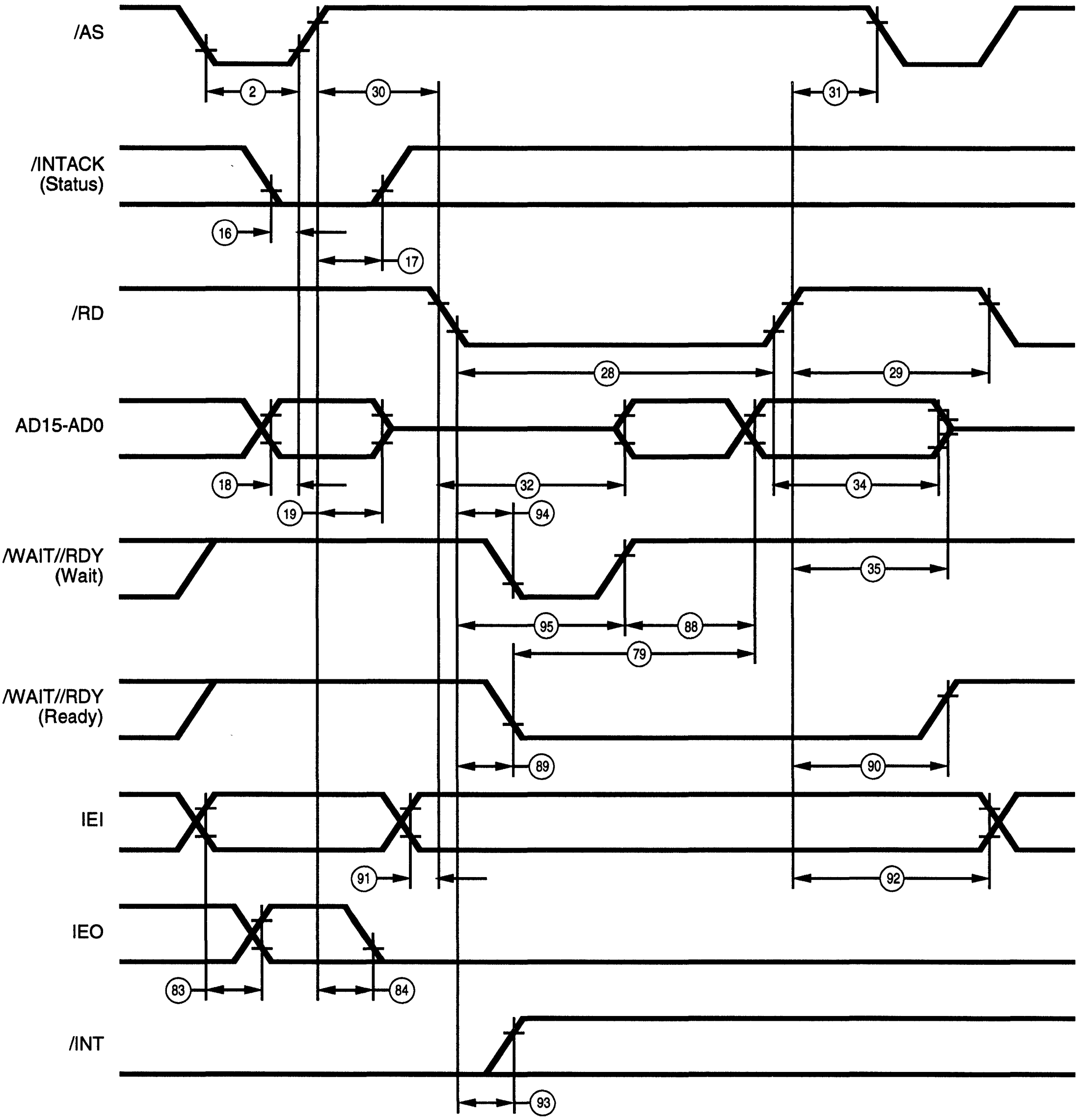


Figure 93. Multiplexed /RD Interrupt Acknowledge Cycle

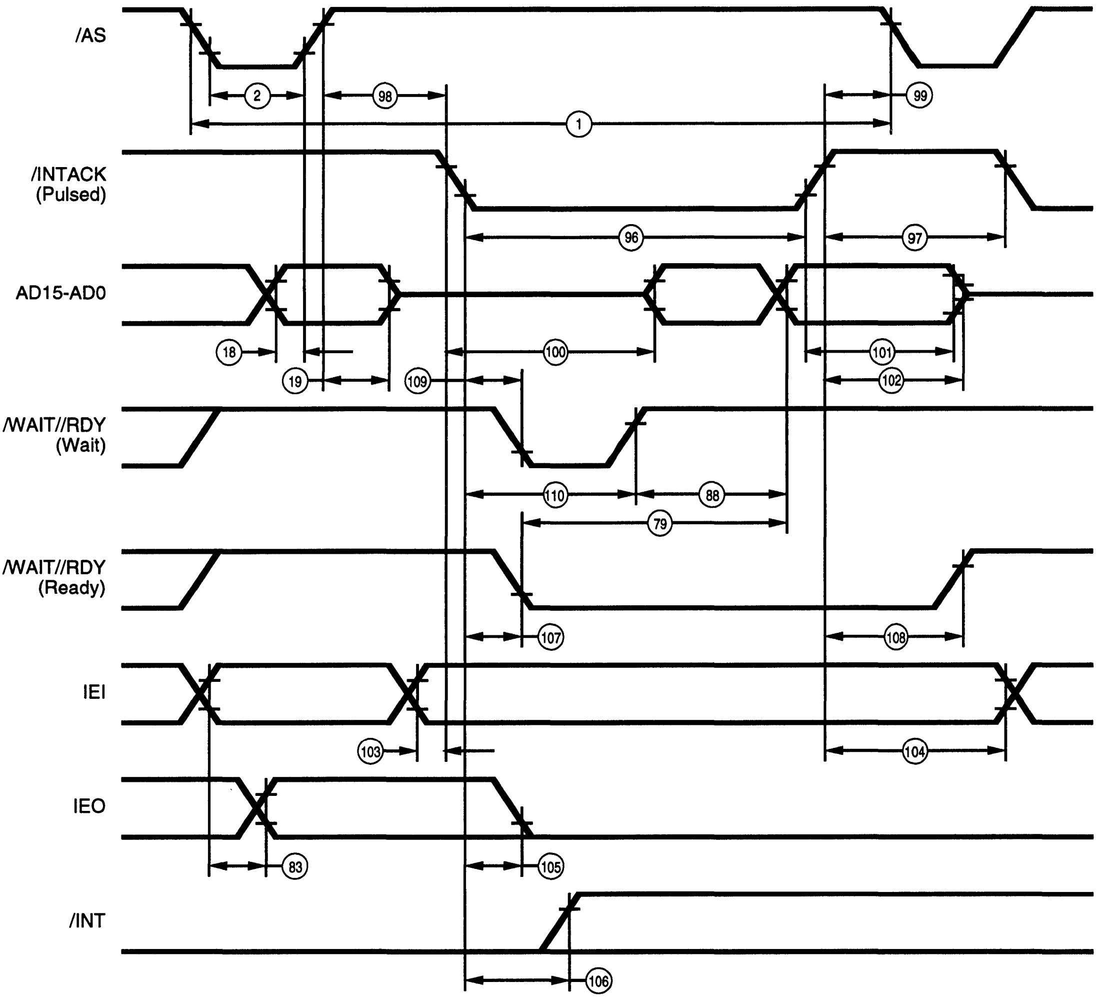


Figure 94. Multiplexed Pulsed Interrupt Acknowledge Cycle

AC CHARACTERISTICS
Timing Diagrams (Continued)

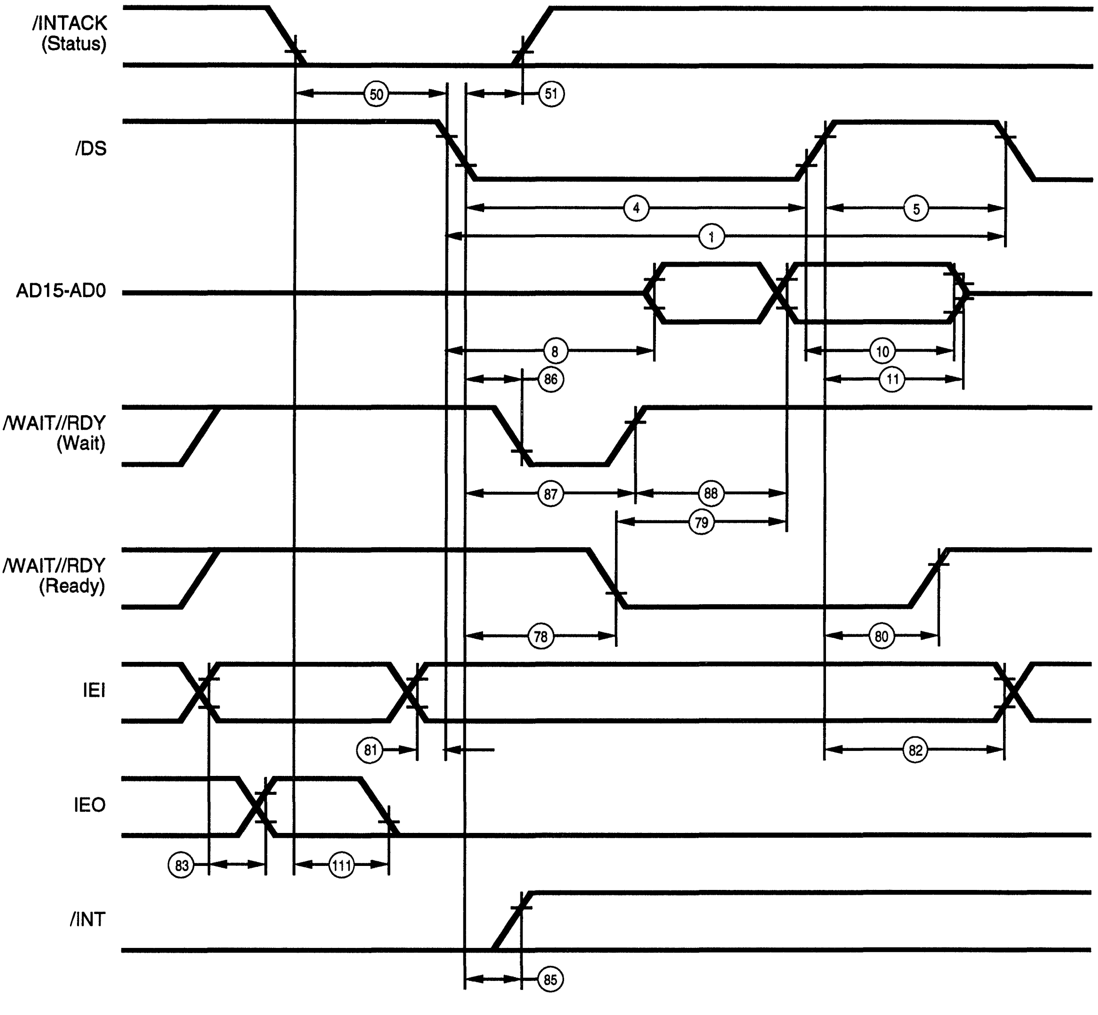


Figure 95. Non-Multiplexed /DS Interrupt Acknowledge Cycle

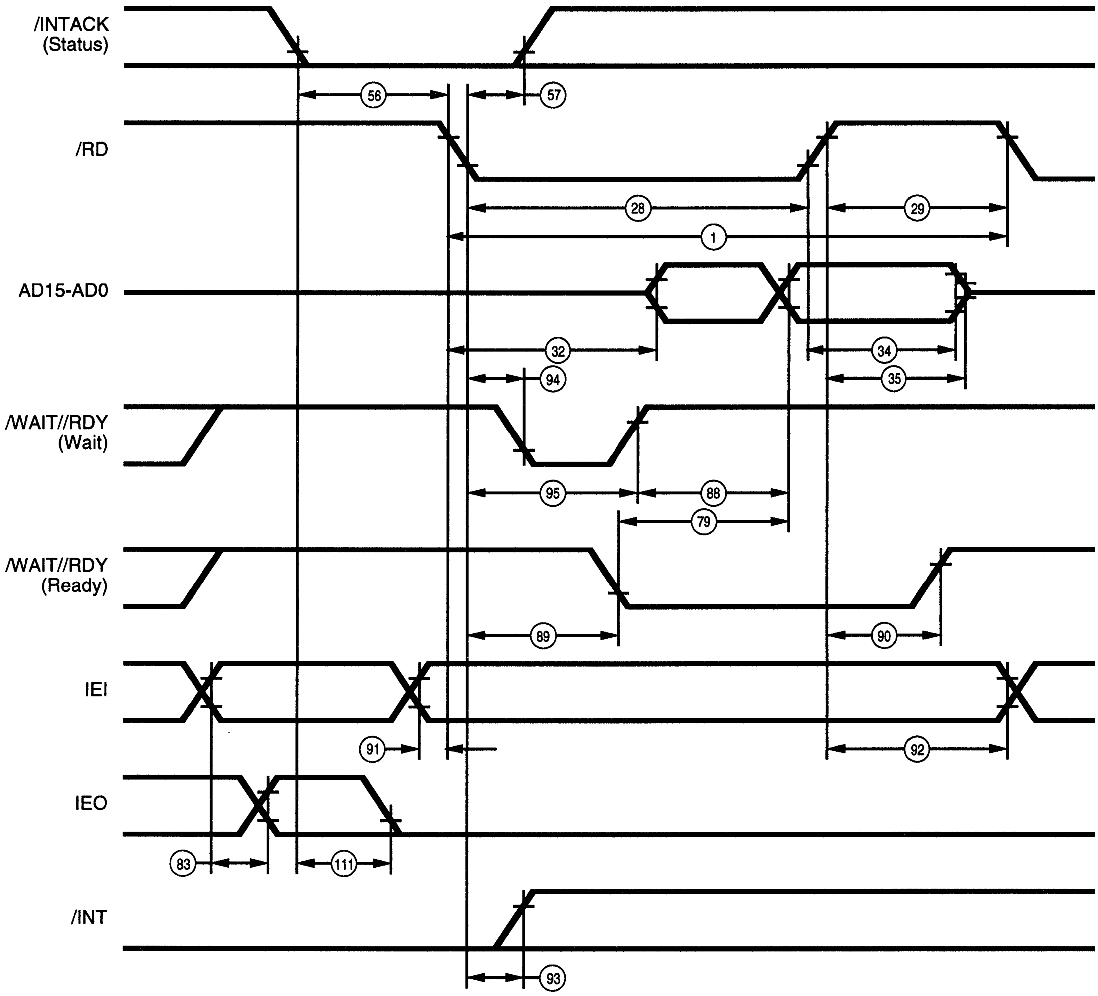


Figure 96. Non-Multiplexed /RD Pulsed Interrupt Acknowledge Cycle

AC CHARACTERISTICS

Timing Diagrams (Continued)

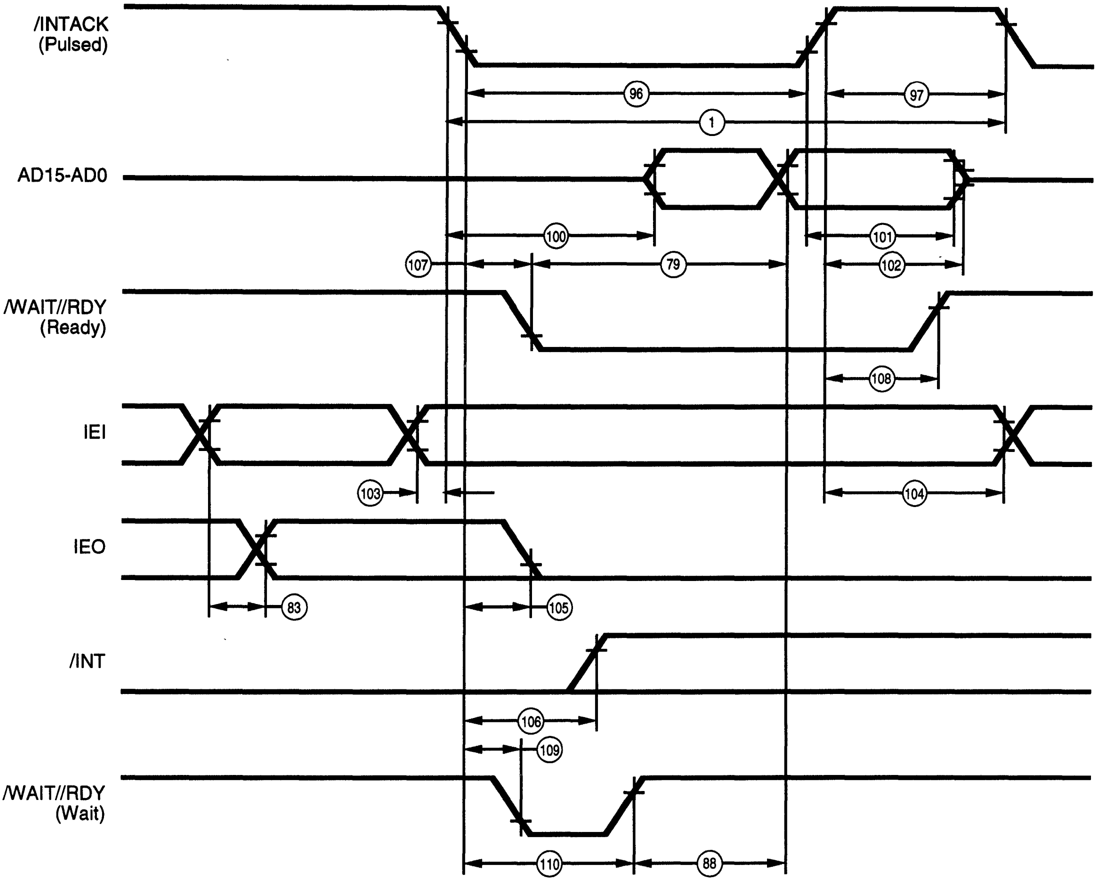


Figure 97. Non-Multiplexed Pulsed Interrupt Acknowledge Cycle

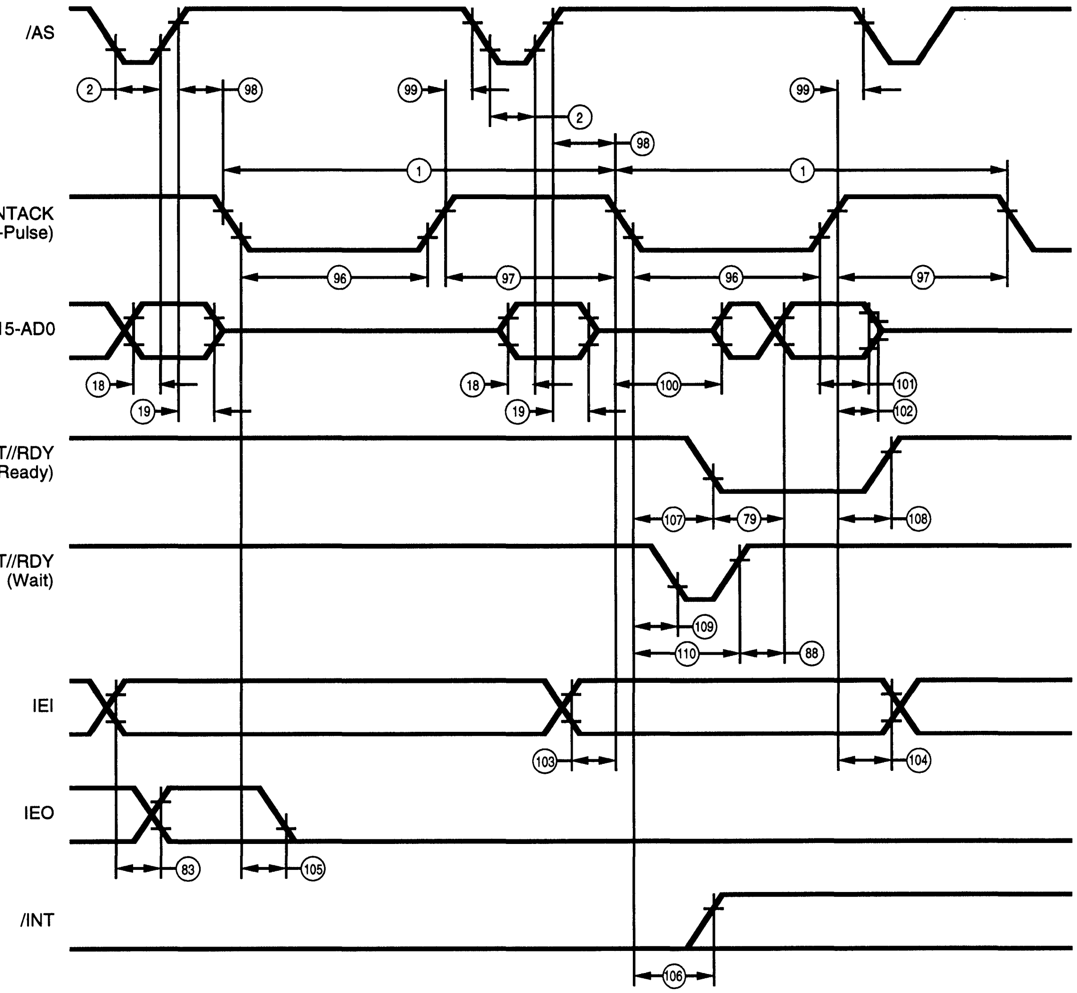


Figure 98. Multiplexed Double-Pulse Intack Cycle

AC CHARACTERISTICS
Timing Diagrams (Continued)

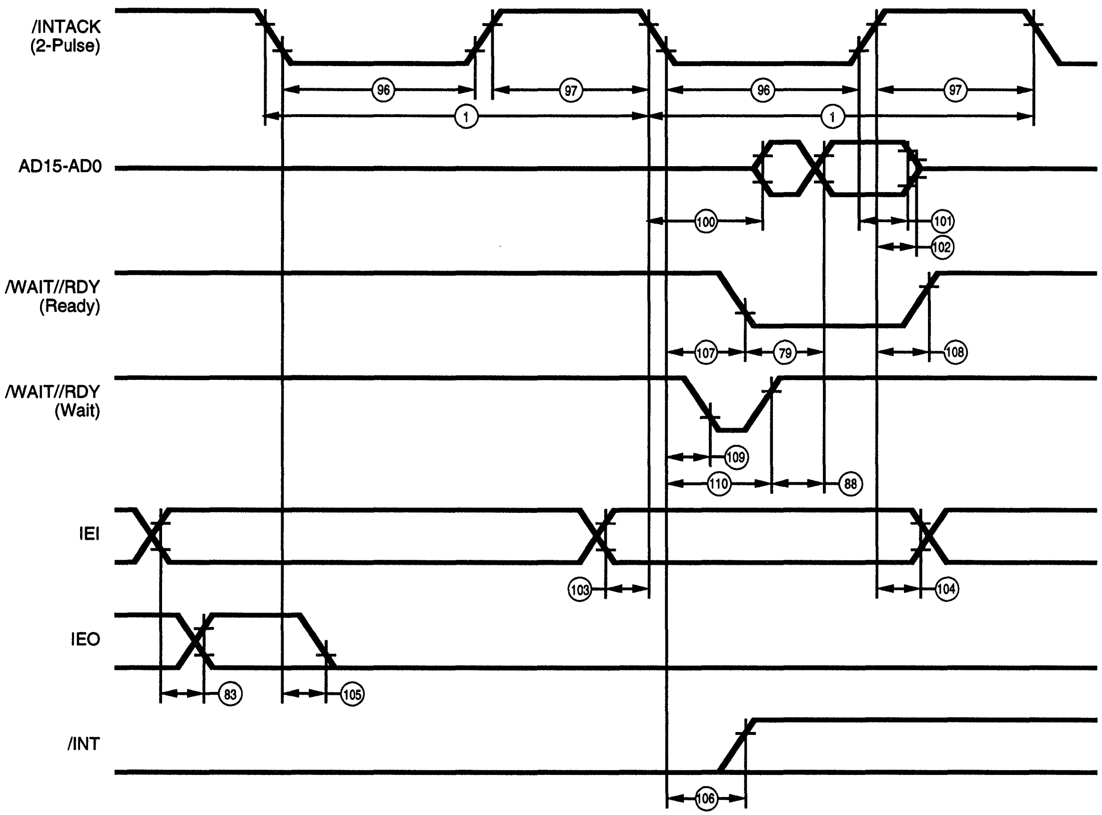


Figure 99. Non-Multiplexed Double-Pulse Intack Cycle

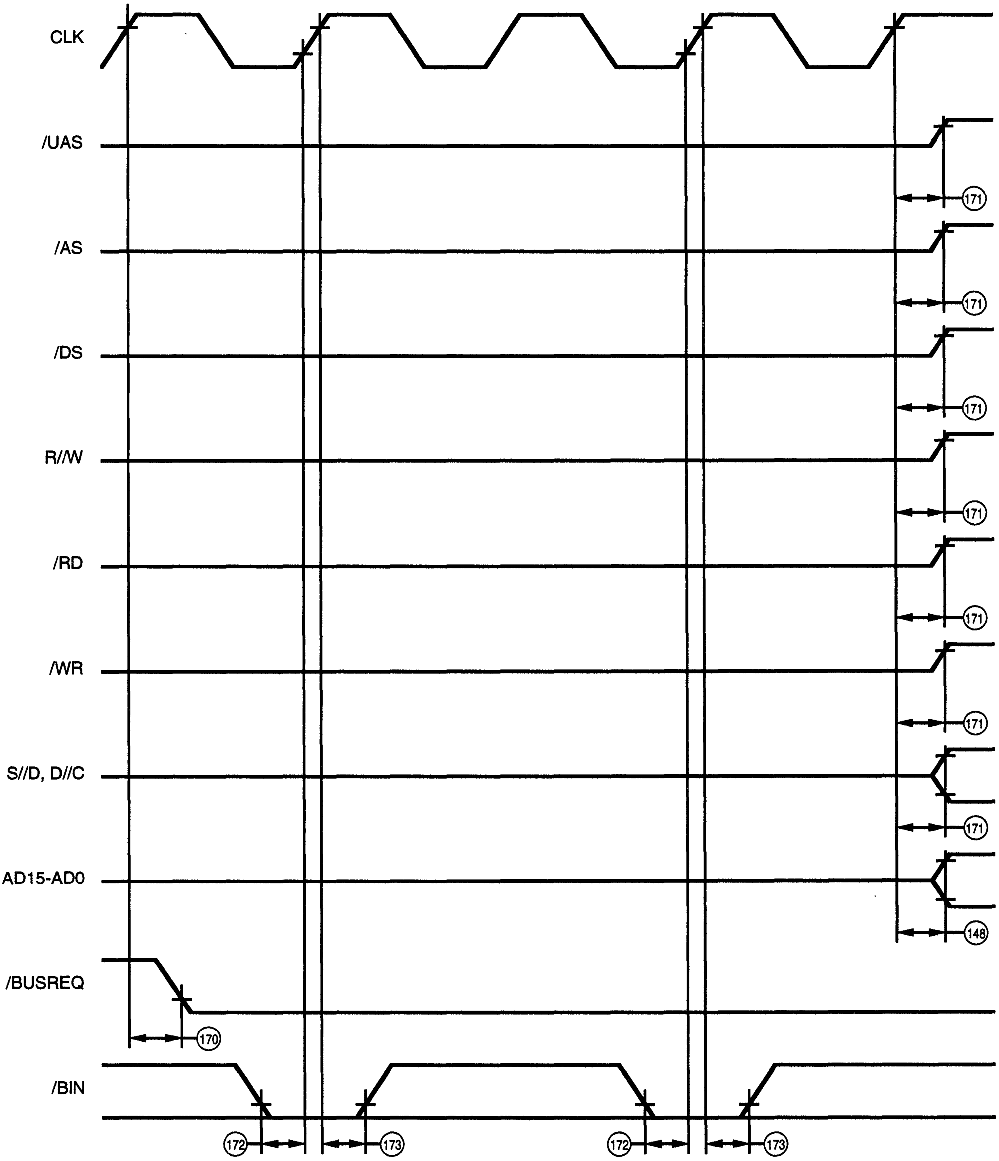


Figure 100. DMA Start-Up

AC CHARACTERISTICS
Timing Diagrams (Continued)

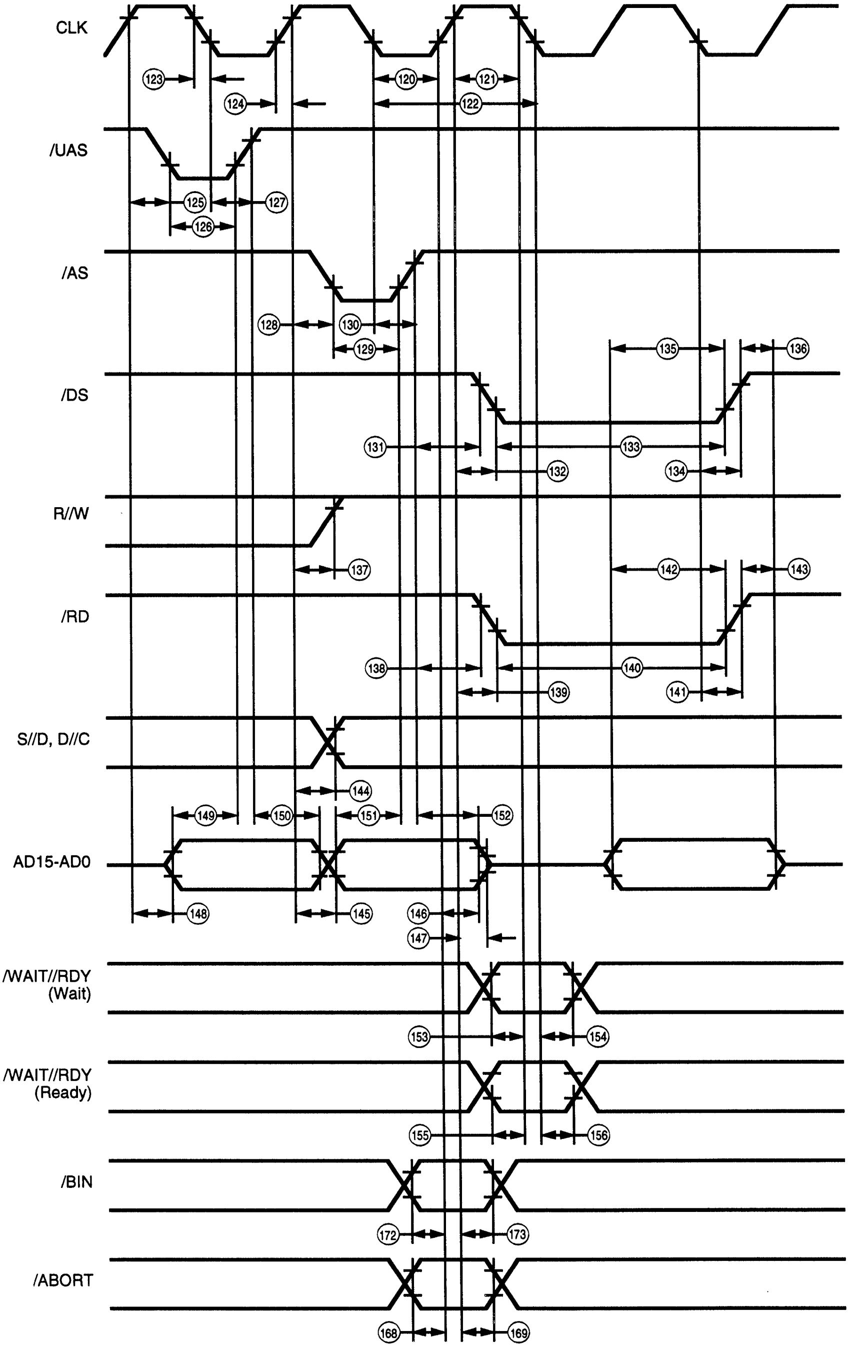


Figure 101. Memory Read

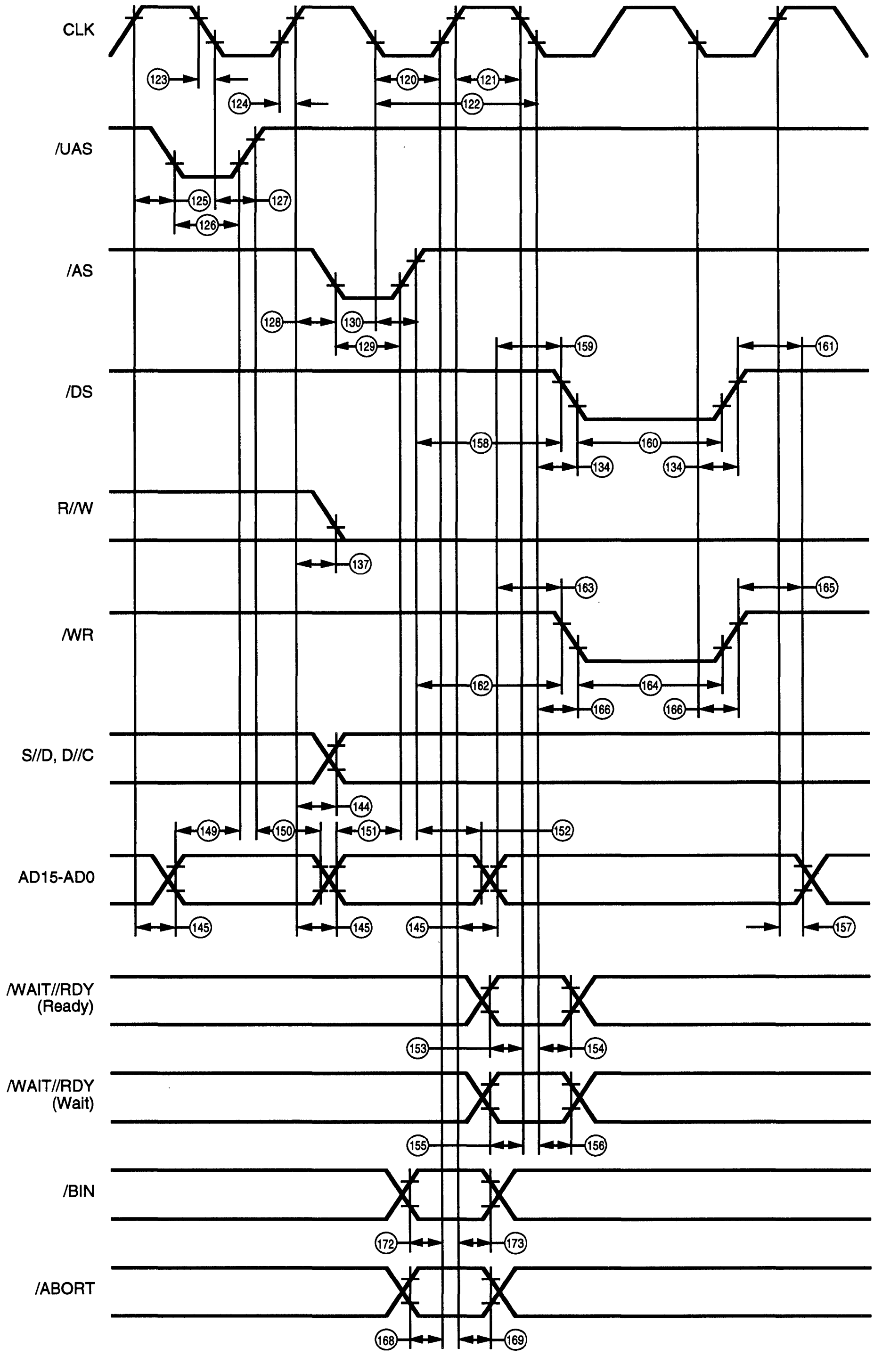


Figure 102. Memory Write

AC CHARACTERISTICS
Timing Diagrams (Continued)

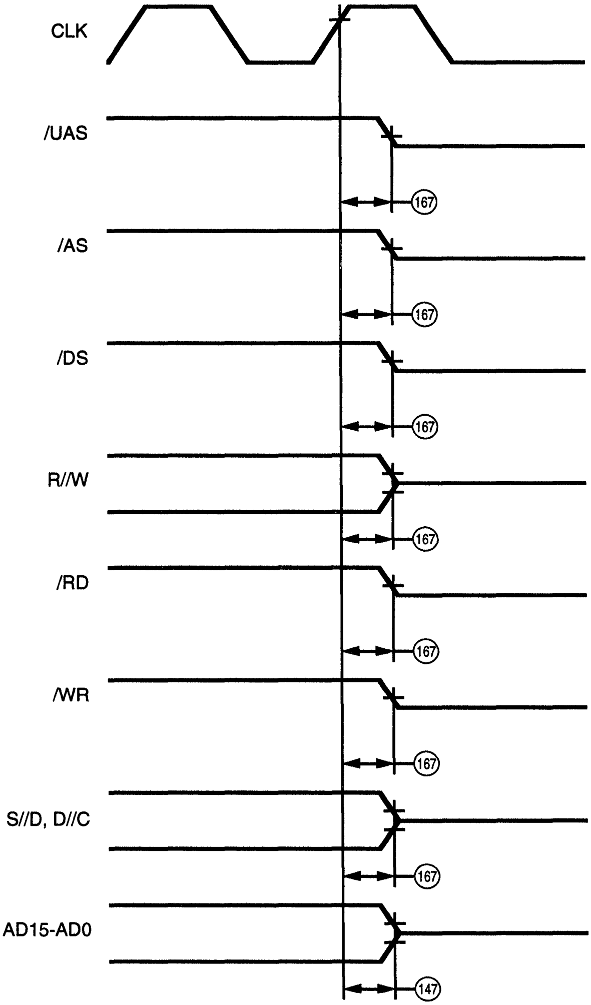


Figure 103. Bus Release

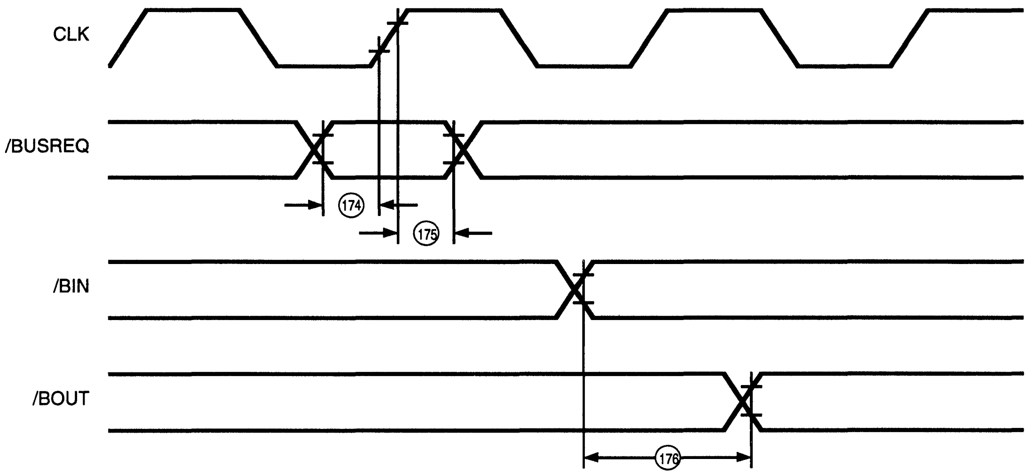


Figure 104. Request Timing

AC CHARACTERISTICS
Timing Table

No	Symbol	Parameter	Min	Max	Units	Note
1	Tcyc	Bus Cycle Time	160		ns	
2	TwASl	/AS Low Width	40		ns	
3	TwASh	/AS High Width	90		ns	
4	TwDSl	/DS Low Width	70		ns	
5	TwDSh	/DS High Width	60		ns	
6	TdAS(DS)	/AS Rise to /DS Fall Delay Time	5		ns	
7	TdDS(AS)	/DS Rise to /AS Fall Delay Time	5		ns	
8	TdDS(DRa)	/DS Fall to Data Active Delay	0		ns	
9	TdDS(DRv)	/DS Fall to Data Valid Delay		85	ns	
10	TdDS(DRn)	/DS Rise to Data Not Valid Delay	0		ns	
11	TdDS(DRz)	/DS Rise to Data Float Delay		20	ns	
12	TsCS(AS)	/CS to /AS Rise Setup Time	15		ns	
13	ThCS(AS)	/CS to /AS Rise Hold Time	5		ns	
14	TsADD(AS)	Direct Address to /AS Rise Setup Time	15		ns	[1]
15	ThADD(AS)	Direct Address to /AS Rise Hold Time	5		ns	[1]
16	TsSIA(AS)	Status /INTACK to /AS Rise Setup Time	15		ns	
17	ThSIA(AS)	Status /INTACK to /AS Rise Hold Time	5		ns	
18	TsAD(AS)	Address to /AS Rise Setup Time	15		ns	
19	ThAD(AS)	Address to /AS Rise Hold Time	5		ns	
20	TsRW(DS)	R/W to /DS Fall Setup Time	0		ns	
21	ThRW(DS)	R/W to /DS Fall Hold Time	25		ns	
22	TsDSf(RRQ)	/DS Fall to /RxREQ Inactive Delay		60	ns	[4]
23	TdDSr(RRQ)	/DS Rise to /RxREQ Active Delay	0		ns	
24	TsDW(DS)	Write Data to /DS Rise Setup Time	30		ns	
25	ThDW(DS)	Write Data to DS Rise Hold Time	0		ns	
26	TdDSf(TRQ)	/DS Fall to /TxREQ Inactive Delay		60	ns	[5]
27	TdDSr(TRQ)	/DS Rise to /TxREQ Active Delay	0		ns	
28	TwRDl	/RD Low Width	70		ns	
29	TwRDh	/RD High Width	60		ns	
30	TdAS(RD)	/AS Rise to /RD Fall Delay Time	5		ns	
31	TdRD(AS)	/RD Rise to /AS Fall Delay Time	5		ns	
32	TdRD(DRa)	/RD Fall to Data Active Delay	0		ns	
33	TdRD(DRv)	/RD Fall to Data Valid Delay		85	ns	
34	TdRD(DRn)	/RD Rise to Data Not Valid Delay	0		ns	
35	TdRD(DRz)	/RD Rise to Data Float Delay		20	ns	
36	TdRDf(RRQ)	/RD Fall to /RxREQ Inactive Delay		60	ns	[4]
37	TdRDr(RRQ)	/RD Rise to /RxREQ Active Delay	0		ns	
38	TwWRl	/WR Low Width	70		ns	
39	TwWRh	/WR High Width	60		ns	
40	TdAS(WR)	/AS Rise to /WR Fall Delay Time	5		ns	
41	TdWR(AS)	/WR Rise to /AS Fall Delay Time	5		ns	
42	TsDW(WR)	Write Data to /WR Rise Setup Time	30		ns	
43	ThDW(WR)	Write Data to /WR Rise Hold Time	0		ns	
44	TdWRf(TRQ)	/WR Fall to /TxREQ Inactive Delay		60	ns	[5]

No	Symbol	Parameter	Min	Max	Units	Note
45	TdWRr(TRQ)	/WR Rise to /TxREQ Active Delay	0		ns	
46	TsCS(DS)	/CS to /DS Fall Setup Time	0		ns	[2]
47	ThCS(DS)	/CS to /DS Fall Hold Time	25		ns	[2]
48	TsADD(DS)	Direct Address to /DS Fall Setup Time	5		ns	[1,2]
49	ThADD(DS)	Direct Address to /DS Fall Hold Time	25		ns	[1,2]
50	TsSIA(DS)	Status /INTACK to /DS Fall Setup time	5		ns	[2]
51	ThSIA(DS)	Status /INTACK to /DS Fall Hold Time	25		ns	[2]
52	TsCS(RD)	/CS to /RD Fall Setup Time	0		ns	[2]
53	ThCS(RD)	/CS to /RD Fall Hold Time	25		ns	[2]
54	TsADD(RD)	Direct Address to /RD Fall Setup Time	5		ns	[1,2]
55	ThADD(RD)	Direct Address to /RD Fall Hold Time	25		ns	[1,2]
56	TsSIA(RD)	Status /INTACK to /RD Fall Setup Time	5		ns	[2]
57	ThSIA(RD)	Status /INTACK to /RD Fall Hold Time	25		ns	[2]
58	TsCS(WR)	/CS to /WR Fall Setup Time	0		ns	[2]
59	ThCS(WR)	/CS to /WR Fall Hold Time	25		ns	[2]
60	TsADD(WR)	Direct Address to /WR Fall Setup Time	5		ns	[1,2]
61	ThADD(WR)	Direct Address to /WR Fall Hold Time	25		ns	[1,2]
62	TsSIA(WR)	Status /INTACK to /WR Fall Setup Time	5		ns	[2]
63	ThSIA(WR)	Status /INTACK to /WR Fall Hold Time	25		ns	[2]
78	TdDSf(RDY)	/DS Fall (Intack) to /RDY Fall Delay		200	ns	
79	TdRDY(DRv)	/RDY Fall to Data Valid Delay		40	ns	
80	TdDSr(RDY)	/DS Rise to /RDY Rise Delay		40	ns	
81	TsIEI(DSI)	IEI to /DS Fall (Intack) Setup Time	60		ns	
82	ThIEI(DSI)	IEI to /DS Rise (Intack) Hold Time	0		ns	
83	TdIEI(IEO)	IEI to IEO Delay		60	ns	
84	TdAS(IEO)	/AS Rise (Intack) to IEO Delay		60	ns	
85	TdDSI(INT)	/DS Fall to /INT Inactive Delay		200	ns	
86	TdDSI(Wf)	/DS Fall (Intack) to /WAIT Fall Delay		40	ns	
87	TdDSI(Wr)	/DS Fall (Intack) to /WAIT Rise Delay		200	ns	
88	TdW(DRv)	/WAIT Rise to Data Valid Delay		40	ns	
89	TdRDf(RDY)	/RD Fall (Intack) to /RDY Fall Delay		200	ns	
90	TdRDr(RDY)	/RD Rise to /RDY Rise Delay		40	ns	
91	TsIEI(RDI)	IEI to /RD Fall (Intack) Setup Time	60		ns	
92	ThIEI(RDI)	IEI to /RD Rise (Intack) Hold Time	0		ns	
93	TdRDI(INT)	/RD Fall (Intack) to /INT Inactive Delay		200	ns	
94	TdRDI(Wf)	/RD Fall (Intack) to /WAIT Fall Delay		40	ns	
95	TdRDI(Wr)	/RD Fall (Intack) to /WAIT Rise Delay		200	ns	
96	TwPIAl	Pulsed /INTACK Low Width	70		ns	
97	TwPIAh	Pulsed /INTACK High Width	60		ns	
98	TdAS(PIA)	/AS Rise to Pulsed /INTACK Fall Delay Time	5		ns	
99	TdPIA(AS)	Pulsed /INTACK Rise to /AS Fall Delay Time	5		ns	
100	TdPIA(DRa)	Pulsed /INTACK Fall to Data Active Delay	0		ns	
101	TdPIA(DRn)	Pulsed /INTACK Rise to Data Not Valid Delay	0		ns	
102	TdPIA(DRz)	Pulsed /INTACK Rise to Data Float Delay		20	ns	

AC CHARACTERISTICS (Continued)

Timing Table

No	Symbol	Parameter	Min	Max	Units	Note
103	TsIEI(PIA)	IEI to Pulsed /INTACK Fall Setup Time	10		ns	
104	ThIEI(PIA)	IEI to Pulsed /INTACK Rise Hold Time	0		ns	
105	TdPIA(IEO)	Pulsed /INTACK Fall to IEO Delay		60	ns	
106	TdPIA(INT)	Pulsed /INTACK Fall to /INT Inactive Delay		200	ns	
107	TdPIAf(RDY)	Pulsed /INTACK Fall to /RDY Fall Delay		200	ns	
108	TdPIAr(RDY)	Pulsed /INTACK Rise to /RDY Rise Delay		40	ns	
109	TdPIA(Wf)	Pulsed /INTACK Fall to /WAIT Fall Delay		40	ns	
110	TdPIA(Wr)	Pulsed /INTACK Fall to /WAIT Rise Delay		200	ns	
111	TdSIA(INT)	Status /INTACK Fall to IEO Inactive Delay		200	ns	[2]
112	TwSTBh	/Strobe High Width	50		ns	[3]
113	TwRESl	/RESET Low Width	170		ns	
114	TwRESh	/RESET High Width	60		ns	
115	TdRES(STB)	/RESET Rise to /STB Fall	60		ns	[3]
116	TdDSf(RDY)	/DS Fall to /RDY Fall Delay		50	ns	
117	TdWRf(RDY)	/WR Fall to /RDY Fall Delay		50	ns	
118	TdWRr(RDY)	/WR Rise to /RDY Rise Delay		40	ns	
119	TdRDf(RDY)	/RD Fall to /RDY Fall Delay		50	ns	
120	TwCLKl	CLK Low Width	25		ns	
121a	TwCLKh	CLK High Width	25		ns	
121b	TwCLKh	CLK High Width (Linked List Mode)	35		ns	[12]
122a	TcCLK	CLK Cycle Time	50		ns	
122b	TcCLK	CLK Cycle Time (Linked List Mode)	60		ns	[12]
123	TfCLK	CLK Fall Time		5	ns	
124	TrCLK	CLK Rise Time		5	ns	
125	TdCLKr (UAS)	CLK Rise to /UAS Fall Delay		30	ns	[6]
126	TwUASl	/UAS Low Width	25		ns	[6,7,13]
127	TdCLKf(UAS)	CLK Fall to /UAS Rise Delay		30	ns	[6]
128	TdCLKr(AS)	CLK Rise to /AS Fall Delay		30	ns	[6]
129	TwASl	/AS Low Width	25		ns	[6,7,13]
130	TdCLKf(AS)	CLK Fall to /AS Rise Delay		30	ns	[6]
131	TdAS(DSr)	/AS Rise to /DS Fall (Read) Delay	25		ns	[6,8]
132	TdCLKr(DS)	CLK Rise to /DS Delay		30	ns	[6]
133	TwDSlr	/DS (Read) Low Width	75		ns	[6,9,13]
134	TdCLKf(DS)	CLK Fall to /DS Delay		30	ns	[6]
135	TsDR(DS)	Read Data to /DS Rise Setup Time	30		ns	[6]
136	ThDR(DS)	Read Data to /DS Rise Hold Time	0		ns	[6]
137	TdCLK(RW)	CLK Rise to R//W Delay		30	ns	[6]
138	TdAS(RD)	/AS Rise to /RD Fall Delay	25		ns	[6,8]
139	TdCLKr(RD)	CLK Rise to /RD Delay		30	ns	[6]
140	TwRDl	/RD Low Width	75		ns	[6,9]
141	TdCLKf(RD)	CLK Fall to /RD Delay		30	ns	[6]
142	TsDR(RD)	Read Data to /RD Rise Setup Time	30		ns	[6]
143	ThDR(RD)	Read Data to /RD Rise Hold Time	0		ns	[6]
144	TdCLK(ADD)	CLK Rise to Direct Address Delay		30	ns	[1,6]
145	TdCLK(AD)	CLK Rise to Address Delay	TdCLKf(DS)	35	ns	[6]
146	ThAD(PC)	Address to CLK Rise Hold Time	0		ns	[6]

No	Symbol	Parameter	Min	Max	Units	Note
147	TdCLK(ADz)	CLK Rise to Address Float Delay		35	ns	[6]
148	TdCLK(ADa)	CLK Rise to Address Active Delay		35	ns	[6]
149	TsAD(UAS)	Address to /UAS Rise Setup Time	10		ns	[6]
150	ThAD(UAS)	Address to /UAS Rise Hold Time	10		ns	[6]
151	TsAD(AS)	Address to /AS Rise Setup Time	10		ns	[6]
152	ThAD(AS)	Address to /AS Rise Hold Time	10		ns	[6]
153	TsW(CLK)	/WAIT to CLK Fall Setup Time	10		ns	[6]
154	ThW(CLK)	/WAIT to CLK Fall Hold Time	15		ns	[6]
155	TsRDY(CLK)	/READY to CLK Fall Setup Time	10		ns	[6]
156	ThRDY(CLK)	/READY to CLK Fall Hold Time	15		ns	[6]
157	ThDW(CLK)	Write Data to CLK Rise Hold Time	0		ns	[6]
158	TdAS(DSw)	/AS Rise to /DS Fall (Write) Delay	40		ns	[6,10,13]
159	TsDW(DS)	Write Data to /DS Fall Setup Time	25		ns	[6,7,13]
160	TwDSlw	/DS (Write) Low Width	45		ns	[6,11,13]
161	ThDW(DS)	Write Data to /DS Rise Hold Time	25		ns	[6,8]
162	TdAS(WR)	/AS Rise to /WR Fall Delay	40		ns	[6,10,13]
163	TsDW(WR)	Write Data to /WR Fall Setup Time	25		ns	[6,7,13]
164	TwWRl	/WR Low Width	45		ns	[6,11,13]
165	ThDW(WR)	Write Data to /WR Rise Hold Time	25		ns	[6,8]
166	TdCLK(WR)	CLK Fall to /WR Delay		30	ns	[6]
167	TdCLK(BUSz)	CLK Rise to Bus Float Delay		30	ns	[6]
168	TsABT(CLK)	/ABORT to CLK Rise Setup Time	20		ns	[6]
169	ThABT(CLK)	/ABORT to CLK Rise Hold Time	15		ns	[6]
170	TdCLK(BRQ)	CLK Rise to /BUSREQ Delay		30	ns	[6]
171	TdCLK(BUSa)	CLK Rise to Bus Active Delay		30	ns	[6]
172	TsBIN(CLK)	/BIN to CLK Rise Setup Time	20		ns	[6]
173	ThBIN(CLK)	/BIN to CLK Rise Hold Time	15		ns	[6]
174	TsBRQ(CLK)	/BUSREQ to CLK Rise Setup Time	25		ns	[6]
175	ThBRQ(CLK)	/BUSREQ to CLK Rise Hold Time	0		ns	[6]
176	TdBIN(BOT)	/BIN to /BOUT Delay		60	ns	

Notes:

AC Test Conditions:

V_{CC} = 5V ±5% unless otherwise specified,
over specified temperature range.
V_{IH} = 2.0V V_{OH} = 2.0V
V_{IL} = 0.8V V_{OL} = 0.8V
Float = +0.5V

[1] Direct Address is any of S//D, D//C or AD15-AD8 used as an address bus.

[2] The parameter applies only when /AS is not present.

[3] Strobe is any of /DS, /RD, /WR or Pulsed /INTACK.

[4] Parameter applies only if read empties the receive FIFO.

[5] Parameter applies only if write fills the transmit FIFO.

[6] Parameter applies only while the IUSC is bus master.

[7] Parameter is clock-cycle dependent, TwCLKh + TfCLK – 5.

[8] Parameter is clock-cycle dependent, TwCLKl + TrCLK –5

[9] Parameter is clock-cycle dependent, TcCLK + TwCLKh + TfCLK –5.

[10] Parameter is clock-cycle dependent, TcCLK –10.

[11] Parameter is clock-cycle dependent, TcCLK –5.

[12] Clock cycle parameters TwCLKh and TcCLK have unique values for Linked List Mode. In Linked List Mode, the system clock cycle is extended to 60 ns, and the system clock High pulse width is extended to 35 ns. This is due to the internal timing paths unique to the Linked List Mode. The transmit and receive bit rates are not affected.

[13] For Linked List Mode, the minimum for these values should be calculated using TwCLKh = 35 ns and TcCLK = 60 ns.

AC CHARACTERISTICS

General Timing Diagram

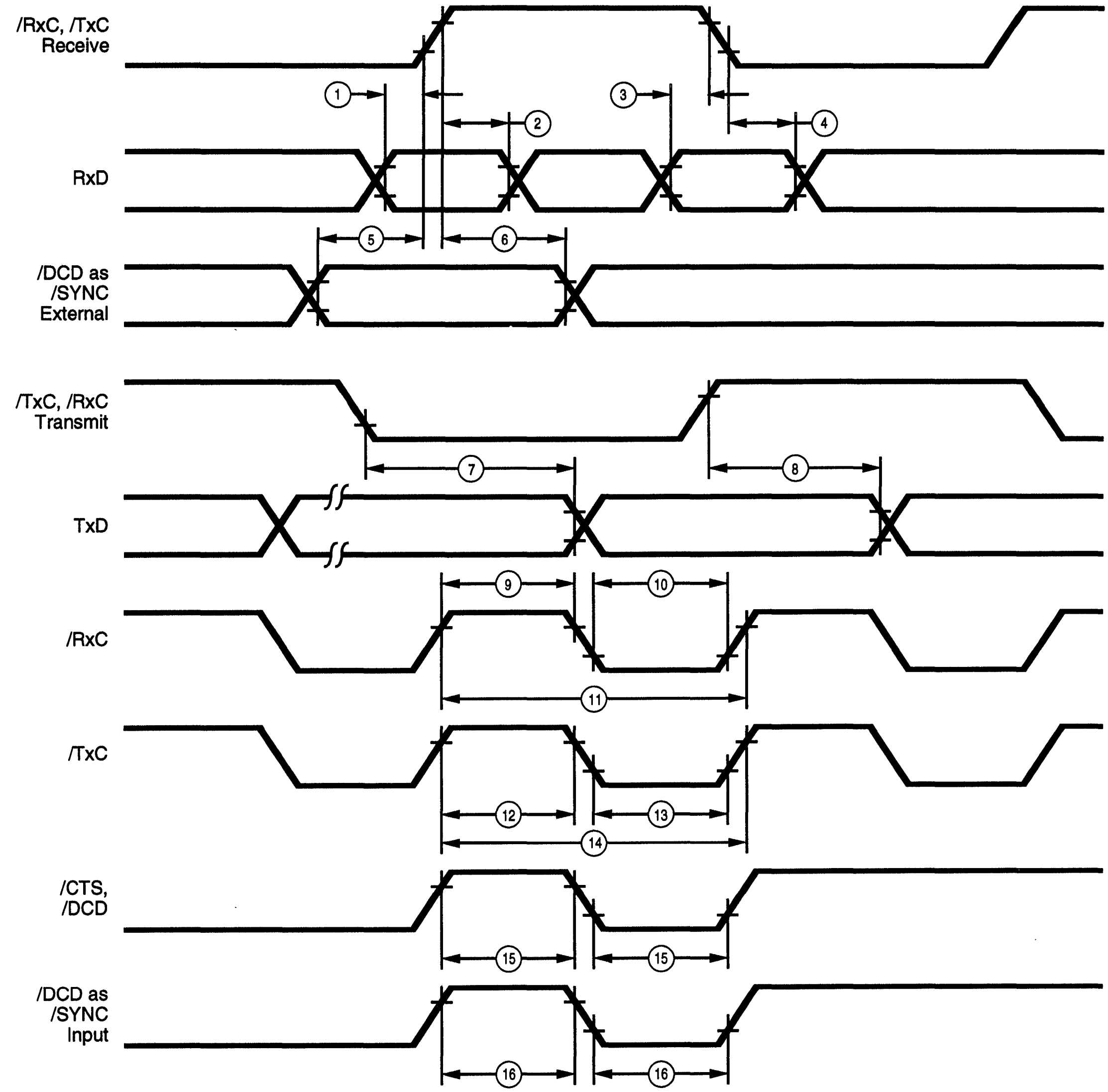


Figure 105. General Timing

AC CHARACTERISTICS

General Timing Table

No	Symbol	Parameter	Min	Max	Units	Note
1	TsRxD(RxCr)	RxD to /RxC Rise Setup Time (x1 Mode)	0		ns	[1]
2	ThRxD(RxCr)	RxD to /RxC Rise Hold Time (x1 Mode)	20		ns	[1]
3	TsRxd(RxCf)	RxD to /RxC Fall Setup Time (x1 Mode)	0		ns	[1,3]
4	ThRxD(RxCf)	RxD to /RxC Fall Hold Time (x1 Mode)	20		ns	[1,3]
5	TsSy(RxC)	/DCD as /SYNC to /RxC Rise Setup Time	0		ns	[1]
6	ThSy(RxC)	/DCD as /SYNC to /RxC Rise Hold Time (x1 Mode)	20		ns	[1]
7	TdTxCf(TxD)	/TxC Fall to TxD Delay		25	ns	[2]
8	TdTxCr(TxD)	/TxC Rise to TxD Delay		25	ns	[2,3]
9	TwRxCh	/RxC High Width	20		ns	
10	TwRxCl	/RxC Low Width	20		ns	
11	TcRxC	/RxC Cycle Time	50		ns	
12	TwTxCh	/TxC High Width	20		ns	
13	TwTxCl	/TxC Low Width	20		ns	
14	TcTxC	/TxC Cycle Time	50		ns	
15	TwExT	/DCD or /CTS Pulse Width	35		ns	
16	TWSY	/DCD as /SYNC Input Pulse Width	35		ns	

AC CHARACTERISTICS
System Timing Diagram

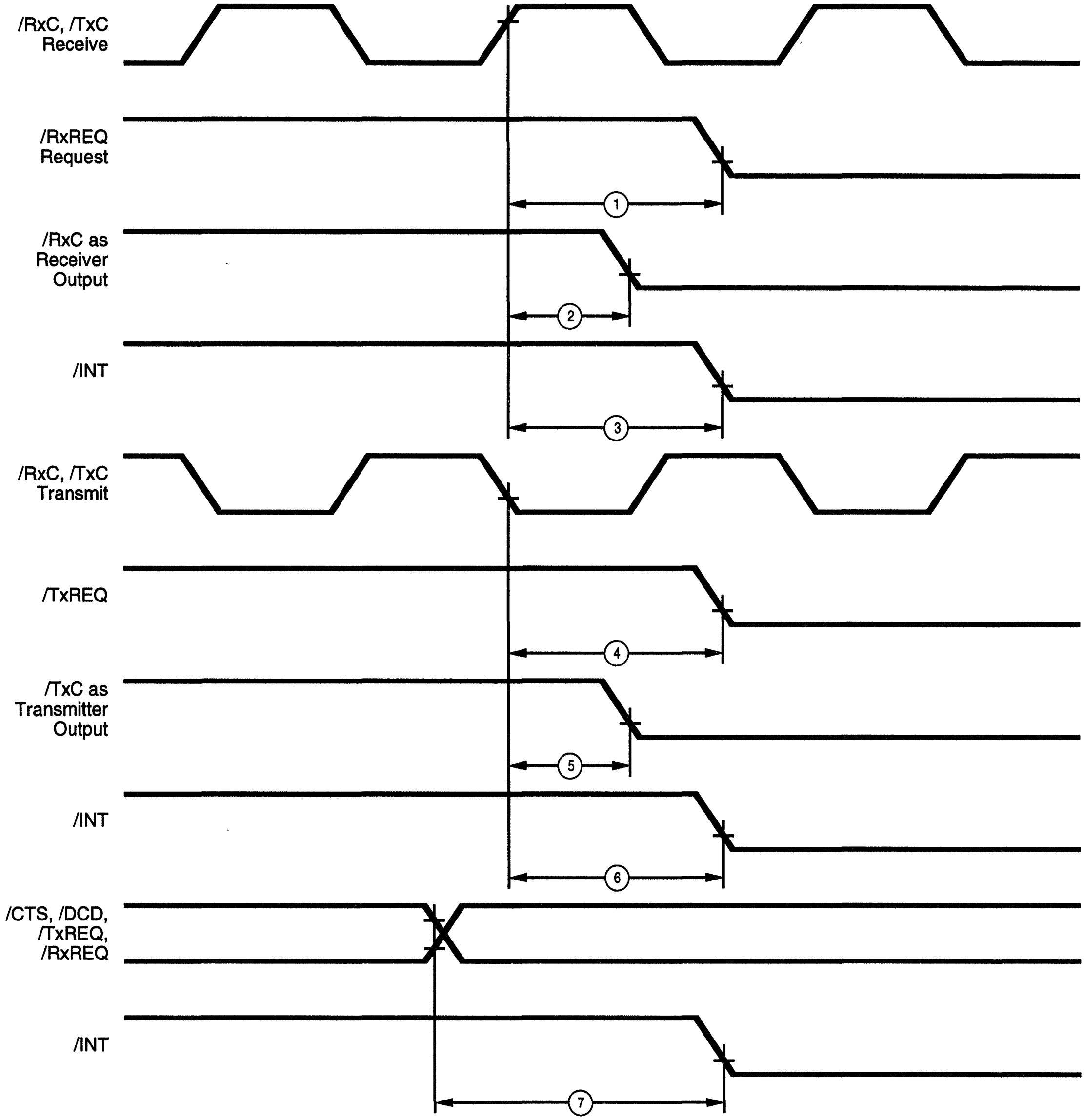


Figure 106. Z16C32 System Timing

AC CHARACTERISTICS
System Timing Table

No	Symbol	Parameter	Min	Max	Units	Note
1	TdRxC(REQ)	/RxC Rise to /RxREQ Valid Delay		50	ns	[2]
2	TdRxC(RxC)	/TxC Rise to /RxC as Receiver Output Valid Delay		50	ns	[2]
3	TdRxC(INT)	/RxC Rise to /INT Valid Delay		50	ns	[2]
4	TdTxC(REQ)	/TxC Fall to /TxREQ Valid Delay		50	ns	[2]
5	TdTxC(TxC)	/RxC Fall to /TxC as transmitter Output Valid Delay		50	ns	
6	TdTxC(INT)	/TxC Fall to /INT Valid Delay		50	ns	[2]
7	TdEXT(INT)	/CTS, /DCD, /TxREQ, /RxREQ transition to /INT Valid Delay		50	ns	

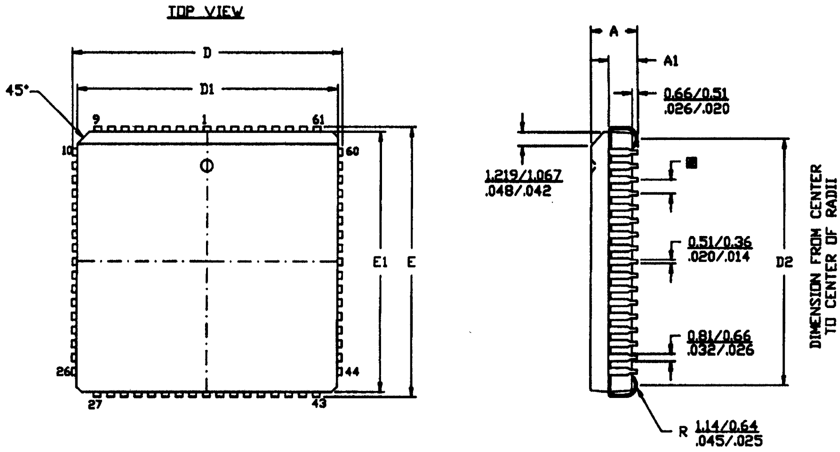
Notes:

[1] /RxC is /RxC or /TxC, whichever is supplying the receive clock.

[2] /TxC is /TxC or /RxC, whichever is supplying the transmit clock.

[3] Parameter applies only to FM encoding/decoding.

PACKAGE INFORMATION



NOTES:

1. CONTROLLING DIMENSIONS : INCH
2. LEADS ARE COPLANAR WITHIN .004 IN.
3. DIMENSION : MM
   INCH

SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	4.32	4.57	.170	.180
A1	2.67	2.92	.105	.115
D/E	25.02	25.40	.985	1.000
D1/E1	24.13	24.33	.950	.958
D2	22.86	23.62	.900	.930
■	1.27 TYP		.050 TYP	

68-Pin Plastic Lead Chip Carrier (PLCC)

ORDERING INFORMATION

Z16C32 IUSC

20 MHz

68-Pin PLCC

Z16C3120VSC

For fast results, contact your local Zilog sales office for assistance in ordering the part desired.

Package

V = Plastic Chip Carrier

Temperature

S = 0°C to 70°C

Speed

20 = 20 MHz

Environmental

C = Plastic Standard

Example:

Z 16C32 20 V S C is a Z16C32, 20 MHz, PLCC, 0°C to +70°C, Plastic Standard Flow

Environmental Flow
Temperature
Package
Speed
Product Number
Zilog Prefix

Using the Z16C30 Universal Serial Controller With MIL-STD-1553B

INTRODUCTION

Zilog's Z16C30 Universal Serial Controller (USC™) is a dual-channel multi-protocol data communications peripheral that supports virtually any serial data transfer application. However, because the USC is so flexible, it may be confusing to a user interested in a particular application of the device. This Application Note will describe the use of the USC in a MIL-STD-1553B environment.

MIL-STD-1553B

MIL-STD-1553B defines a serial data bus that was originally intended for use in aircraft. However, several attributes of 1553B make it suitable, not only for other military systems, but industrial and process-control environments as well. Chief among these attributes is employment of a command/response protocol. This guarantees a response within a certain amount of time. Other attractive attributes include high noise immunity and provision for redundant buses.

Devices attached to a 1553B bus can operate as either a Bus Controller (BC), a Remote Terminal (RT) or Bus Monitor (BM). Both the BC and RT are capable of receiving and transmitting on the bus, while the BM is a receive-only device. Allowed transfers on the bus are BC-to-RT, RT-to-RT, RT-to-BC, and broadcast. All transfers on the bus occur at the request of the current bus controller, even though the BC may not be the source or destination of the data (as in an RT-to-RT transfer).

The standard allows either a single bus controller or multiple bus controllers, although only one may be active at any one time. Control transfer from one bus controller to another is done either by polling, where the current bus controller polls other potential bus controllers before transferring mastership, or in a round-robin fashion. Round-robin means bus mastership passes from BC to BC in a predetermined fashion after a fixed amount of time.

Messages on the bus must be acknowledged within a fixed amount of time, which is an attractive feature of 1553B. A typical sequence could be: first, the BC sends a command word, which contains the destination address and a byte count of the data words to follow; then it sends the data words. The RT responds with a status word showing proper receipt of the data, within a fixed amount of time.

The 1553B uses a unique word format (Figure 1). Each word on the bus is 20 bits long. The first three bits are synchronization bits, which also identify the word as either command, status, or data. The next 16 bits carry the actual information, followed by a parity bit in bit position 20. Because 1553B uses Manchester encoding, shown in Figure 2, the sync patterns are unique and easily identifiable by a receiver. Both the receiver and the transmitter in the USC explicitly support this word format and data encoding.

Notes:
All Signals with a preceding front slash, "/", are active Low, e.g., B//W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

Connection	Circuit	Device
Power	V_{CC}	V_{DD}
Ground	GND	V_{SS}

MIL-STD-1553B (Continued)

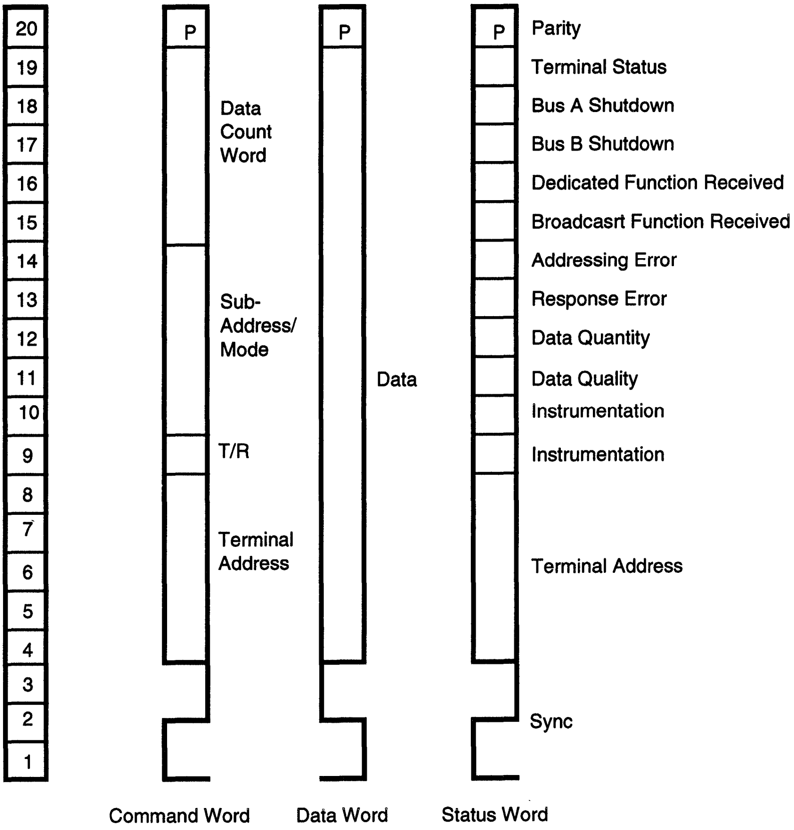


Figure 1. Message Formats

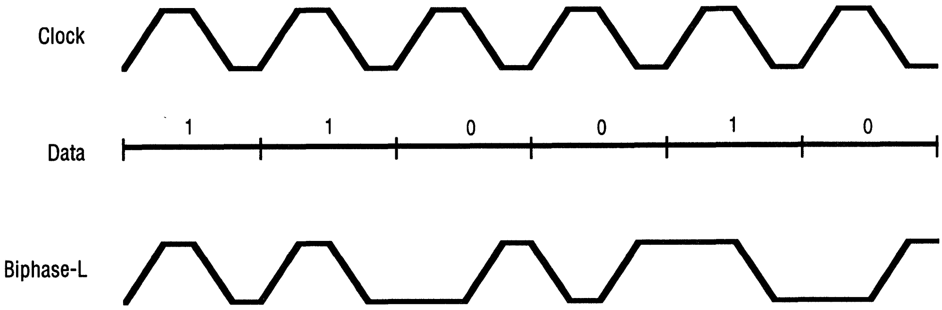


Figure 2. Manchester II (Bi-phase-Level) Data Encoding

USC Initialization for 1553B Operation

Before the USC can be initialized, it is properly connected to the remainder of the system and the bus interface programmed appropriately. While this Application Note does not cover the hardware interfacing aspects of the USC, it is easily interfaced to either multiplexed or non-multiplexed 8-bit, 16-bit or 32-bit systems. This document assumes a multiplexed 16-bit interface.

After hardware reset, the USC watches the bus interface to determine whether it is multiplexed or non-multiplexed. A multiplexed bus interface is selected when an Address Strobe (/AS) is detected. After a hardware reset, the first write to the USC accesses the Bus Configuration Register (BCR) to select the bus width, addressing method, and Wait or Ready function. This is the only time that the BCR may be accessed, and once the BCR is programmed the remainder of the USC may be configured.

Both hardware reset and software reset clear all of the registers in the USC. One by-product of this is that the clocks are disconnected from both the receiver and transmitter, so the first thing that should be programmed is the clock sources in the Clock Mode Control Register (CMCR). The CMCR should be programmed (Figure 3), with the RxC pin feeding CTR1 and the DPLL, which in turn feed the transmitter and the receiver, respectively. All unused functional blocks (CTR0, BRG0 and BRG1) are either disabled or programmed with static clock sources to minimize power dissipation.

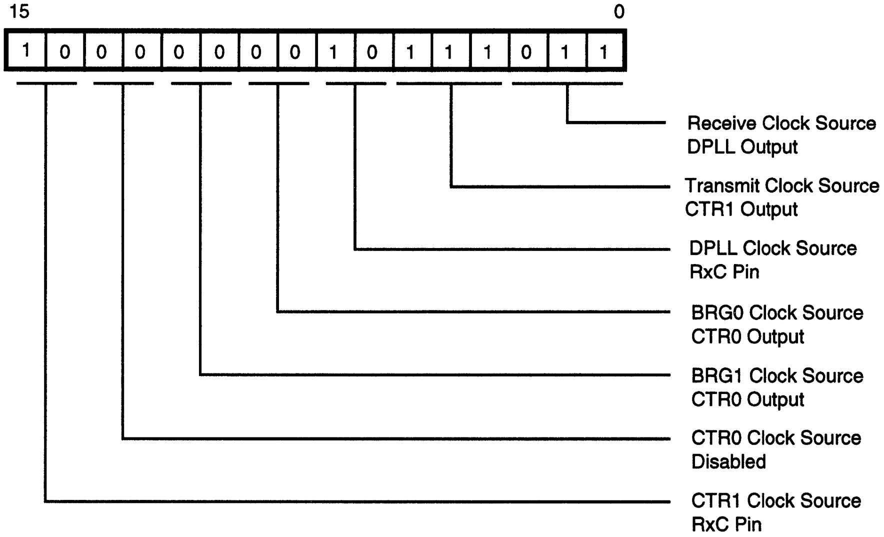


Figure 3. Clock Mode Control Register

USC Initialization for 1553B Operation (Continued)

Once the clocks are connected the remainder of the hardware interface should be programmed. This includes the DMA interface, with the /RxREQ, /TxREQ, /RxACK and /TxACK signals, and the /RxC and /TxC pins. The lower byte of the Hardware Configuration Register (HCR) controls the /RxACK and /TxACK signals, while the I/O Control Register (IOCR) controls /RxREQ, /TxREQ, /RxC, /TxC and TxD signals. Both of these registers contain bits that are not used in this application; these are programmed with zeros as shown in Figures 4 and 5.

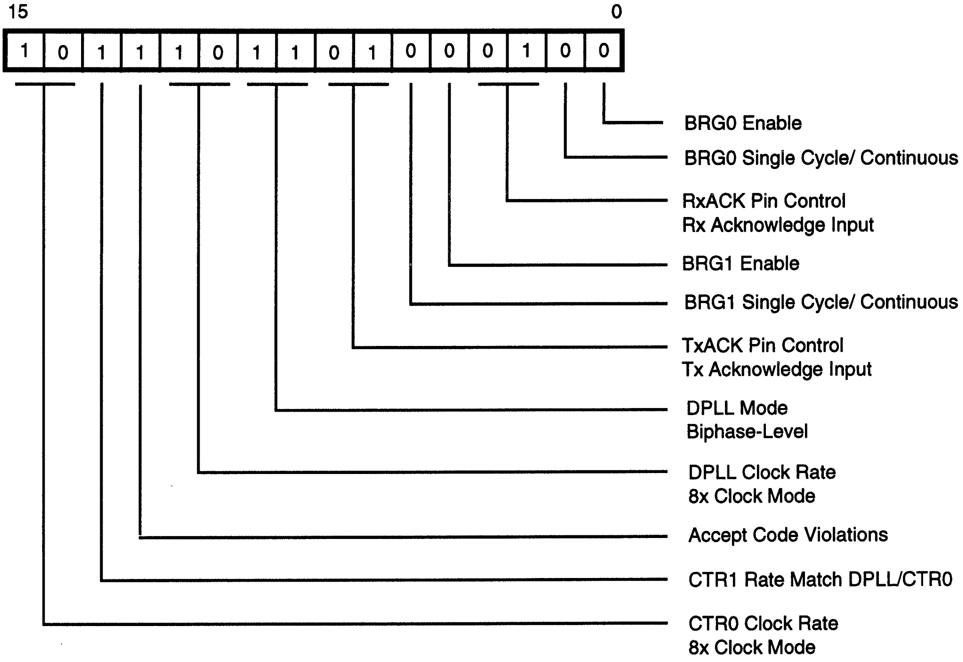


Figure 4. Hardware Configuration Register

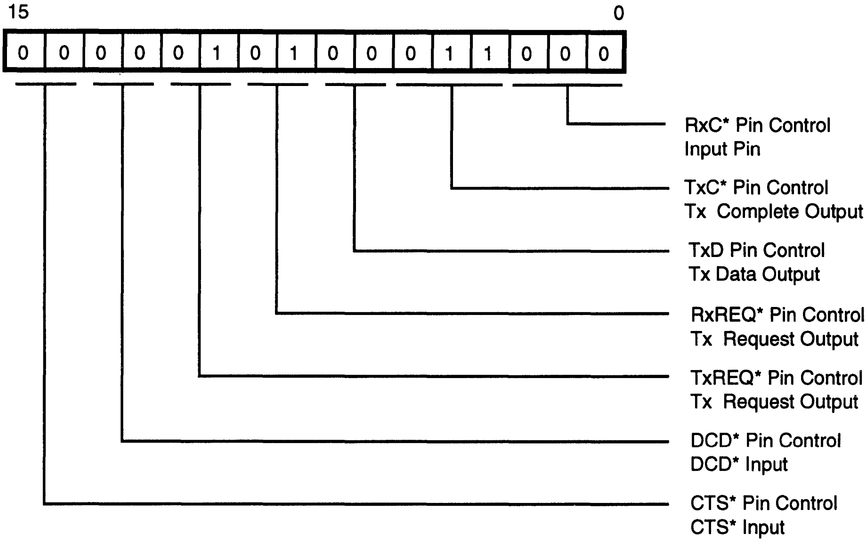


Figure 5. I/O Control Register

Figure 5 also shows that the upper byte of the HCR controls the DPLL. In this example the DPLL and CTR1 will both operate in 8X mode, requiring an 8.000 MHz clock to be supplied to the /RxC pin. This gives the required 1.000 MHz 1553B data rate. The DPLL is configured to operate in Bi-phase-Level mode, with the Accept Code Violations option enabled because of the code violations inherent in the 1553B sync patterns. Design considerations for the DPLL are discussed in a later section.

Once the clocks are configured, the receiver and transmitter may be programmed, but not enabled. First, the Receive Mode Register (RMR) and Transmit Mode Register (TMR) should be written as shown in Figures 6 and 7, respectively. These registers control data encoding, parity, and character length. Note that a character length of 8 bits is selected; this will be extended to 16 bits by a control bit in the Channel Mode Register (CMR) - Figure 8. The CMR is where the receiver and transmitter modes are actually selected. Note that the 1553B sync polarity for transmit is controlled by bit 12 of the CMR (the CV Polarity bit). The state of this bit is FIFO'ed with the data by the transmitter and may be automatically loaded with the data as part of the transmit status block. This is explained later when the transmitter operation is covered in more detail.

USC Initialization for 1553B Operation (Continued)

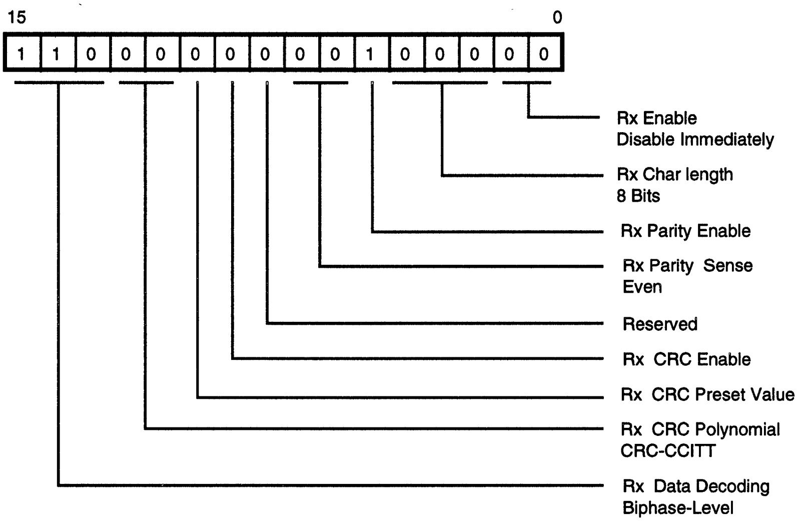


Figure 6. Receive Mode Register

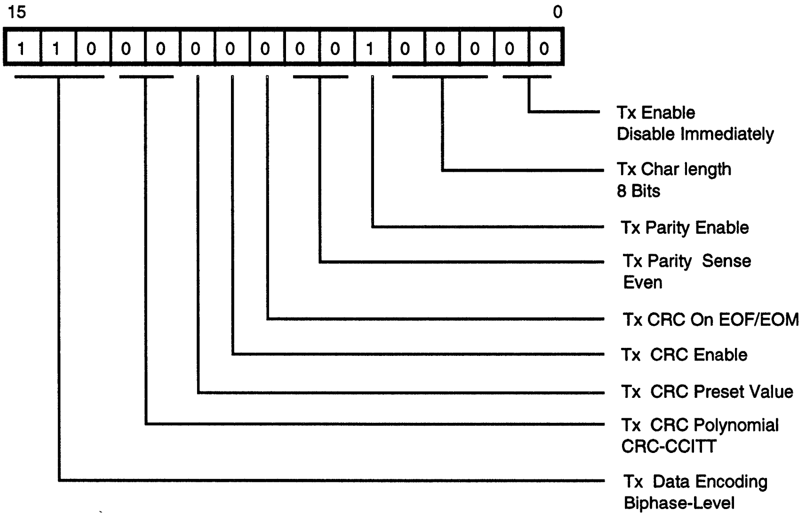


Figure 7. Transmit Mode Register

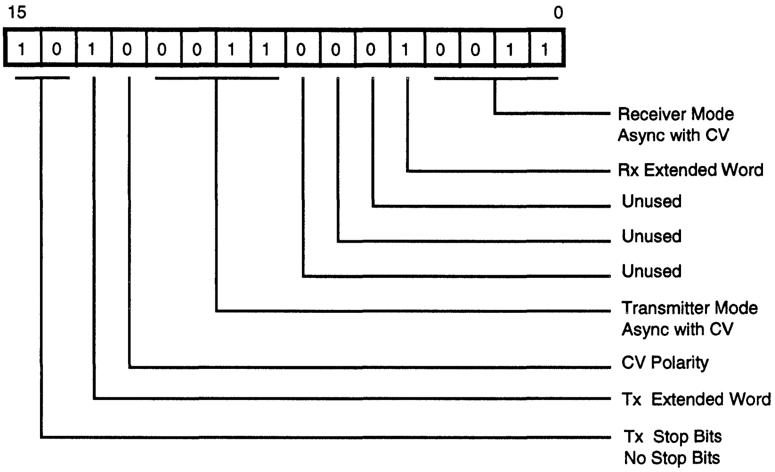


Figure 8. Channel Mode Register

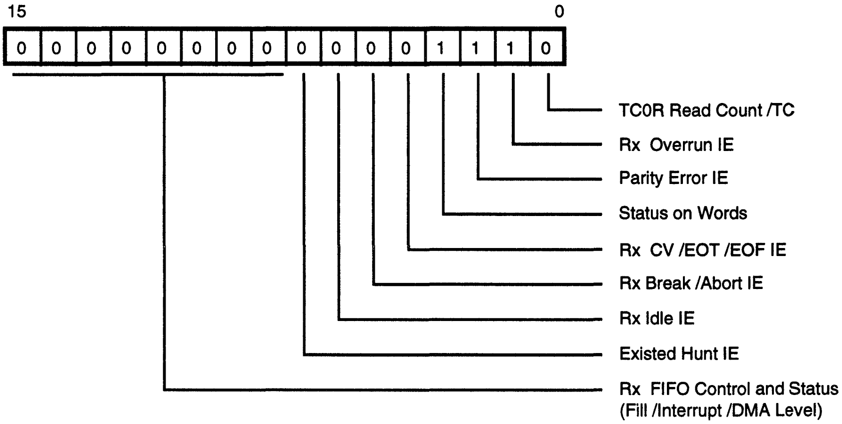


Figure 9. Receive Interrupt Control Register

USC Initialization for 1553B Operation (Continued)

At this point the interrupt and DMA details are programmed. Data transfer through DMA is assumed in this document, but some interrupts are still enabled for error conditions. In the receiver, Parity Error and Overrun Error status interrupts are enabled to collect receive errors on a word-by-word basis. In the transmitter, no status interrupts are enabled and the Wait For Send Command option is not enabled. This means that the DMA controls the data flow, as the USC is always requesting data to fill the transmit FIFO.

Alternatively, with Wait For Send Command enabled, the CPU controls the data flow since the USC only requests data after this command is issued and until a word marked as EOF is written to the FIFO. The Transmit Interrupt Control Register selects these options (Figure 10). The transmitter automatically idles with continuous ones, but because of the Bi-phase-Level data encoding it is preferred to have the transmitter idle with a marking line. This is controlled by bits in the Transmit Command/Status Register (Figure 11).

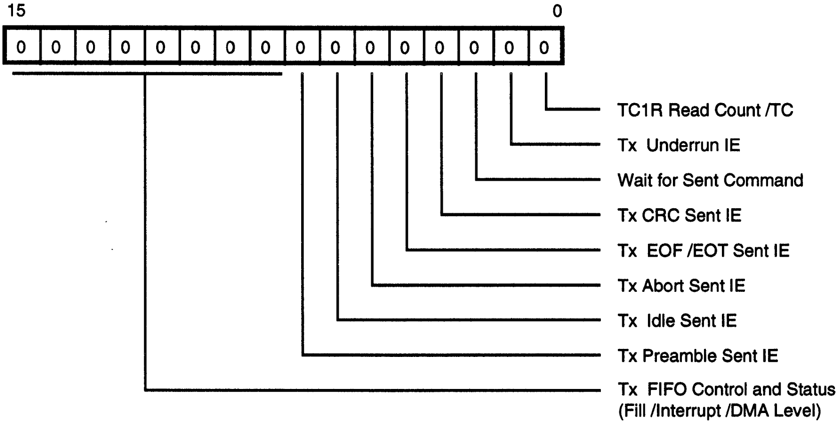


Figure 10. Transmit Interrupt Control Register

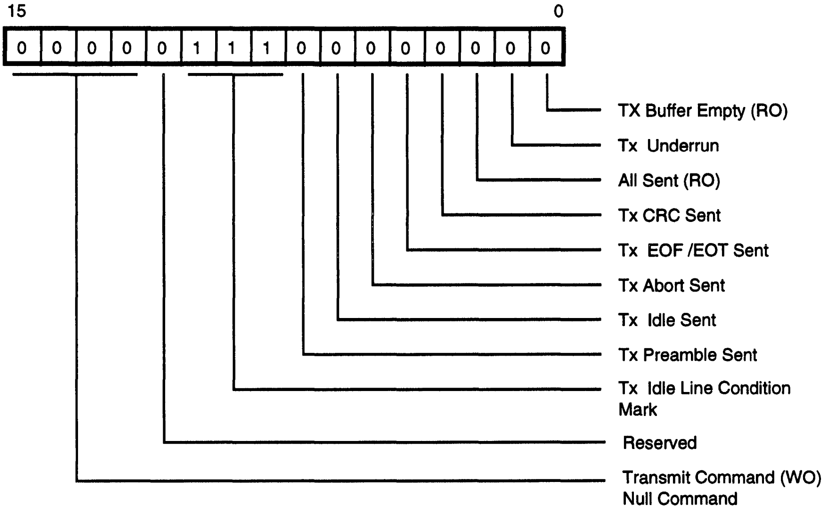


Figure 11. Transmit Command/Status Register

USC Initialization for 1553B Operation (Continued)

The only remaining task is to write the Interrupt Vector Register with the interrupt vector and enable the interrupts, the receiver and the transmitter. Interrupts are enabled in the Interrupt Control Register (ICR) shown in Figure 12. Note that both the individual Interrupt Enable (IE) and the Master Interrupt Enable (MIE) bits must be set. In this example the Vector Includes Status (VIS) option will not be enabled because only receive status interrupts are enabled. The receiver is enabled by setting bit 1 in the RMR and the transmitter is enabled by setting bit 1 in the TMR. The /DCD and /CTS pins may be used as enables for the receiver and transmitter respectively, by setting bit 0 in the RMR or TMR.

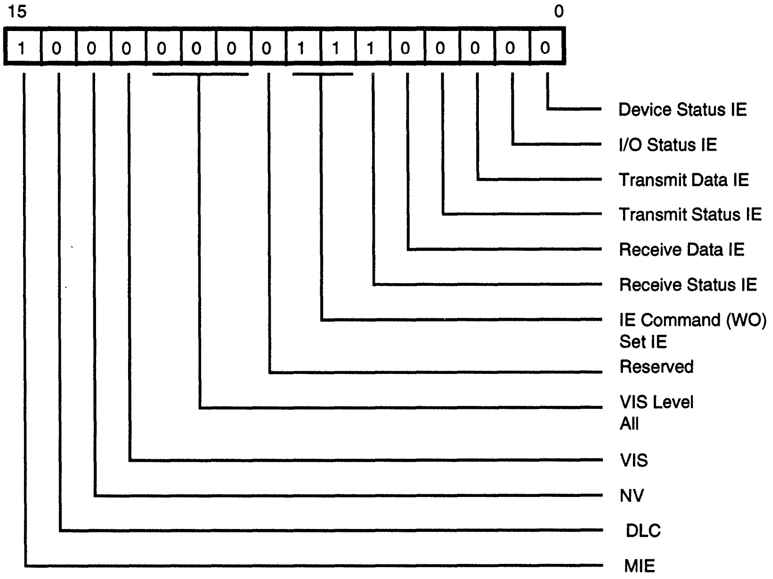


Figure 12. Interrupt Control Register

The USC provides for both byte and bit shuffling within the receiver and transmitter to allow easy interface to any processor bus. The four options are shown in Table 1. These options are changed by commands in the CCAR as shown in Figure 13. Note that the default (reset) case is straight and LSB first.

Table 1. CPU Bus to Transmit and Receive Bit Ordering

First Sent/Received (1553B Bit 4)													Last Sent/Received (1553B Bit 19)			
LSB First Straight	AD8	AD9	AD10	AD11	AD12	AD13	AD14	AD15	AD0	AD1	AD2	AD3	AD4	AD5	AD6	AD7
MSB First Straight	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
LSB First Swapped	AD0	AD1	AD2	AD3	AD4	AD5	AD6	AD7	AD8	AD9	AD10	AD11	AD12	AD13	AD14	AD15
MSB First Swapped	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8

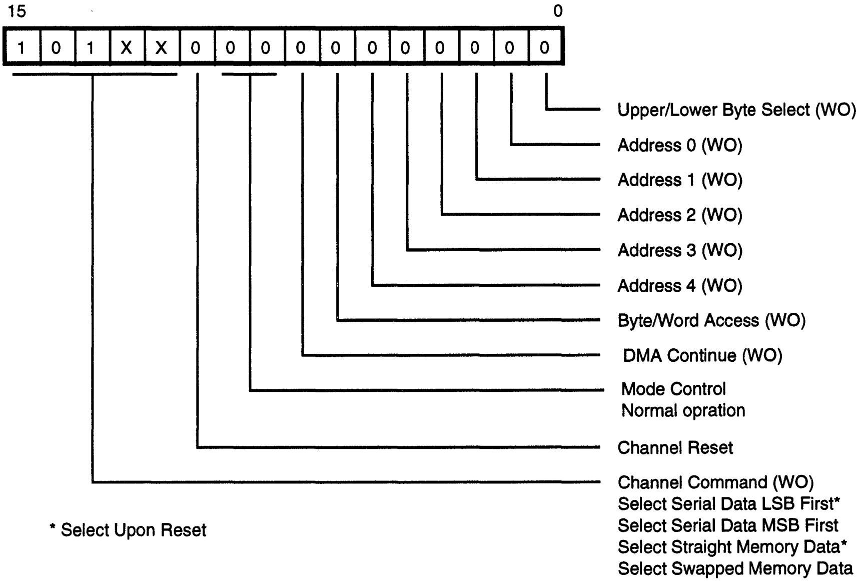


Figure 13. Channel Command/Address Register

Transmitter: Detailed Operation

The polarity of the transmitted sync signal is controlled by bit 12 in the CMR. The state of this bit is FIFO'ed along with the data so the full depth of the transmit FIFO is available for buffering. Note that this bit need not be written with every word transmitted; the current value is FIFO'ed with the data written to the FIFO. The sync polarity may be written by the CPU, either with or without the DMA transfer of data, or the USC may be configured so that the DMA transfers this information to the USC with the data. With this method, the DMA can either transfer a word at a time or a block with the same sync polarity at a time. The method takes advantage of the send character counter and the transmit status block features of the USC.

The send character counter keeps track of bytes written to the transmit FIFO by counting down from a programmed value. This works even when a 16-bit bus is employed. The byte written to the FIFO with a count of zero is marked in the FIFO as the "End-Of-Frame". While this has no meaning in a 1553B protocol, it does tell the logic in the USC to fetch another send status block before asking for any more data. In fact, the DMA request is deactivated until this "last byte" is extracted from the FIFO by the transmitter. This is important because it directly influences bus activity and the availability of the transmit FIFO.

If the sync polarity bit is written (in the transmit status block) with every word, the transmit FIFO is effectively not available. Thus, the preferred method of operation is to use one transmit status block for sequential words of the same type (command or data). To do this, a two word transmit status block must be selected in the CCR with the send data organized in memory. The first word of the transmit status block contains the sync polarity, and as a by-product of the design, controls the number of stop bits to be sent. The second word is the byte count of the words to be sent.

Transmit Data Memory Organization

1. Transmit Status Block First Word (Sync Polarity).
2. Transmit Status Block Second Word (Number of bytes to be sent).
3. First Word for Transmission.
4. Second Word for Transmission.

●
●
●

Note: It is probably best to send status word replies (when acting as an RT) under CPU control because they are so short.

Further, the transmitter normally idles, sending continuous ones which are encoded. This application has programmed an idle line condition of marking ones. In addition, the /TxC pin is programmed to provide the Transmit Complete signal output. Transmit Complete will be active (High) for those bit times where the transmitter is sending the idle line condition. This allows external logic to create a double-ended signal for transformer drive from the single-ended transmit data output.

Receiver: Detailed Operation

The polarity of the received sync pattern is reported in bit 8 of the Receive Command/Status register (RCSR). A zero in this bit indicates a data sync, while a one in this bit indicates a command/status sync. The state of this bit is FIFO'ed along with the receive data so the full depth of the receive FIFO is available for buffering. The sync polarity is read by the CPU, either with or without DMA transfer of data, or the USC is configured so the DMA transfers this information to memory with the data. With this method, the DMA transfers a word (plus status) at a time. Note that this is necessary to meet the response time restrictions of 1553B. Only a single-word receive status block should be used because the second word is meaningless in this word-oriented protocol.

The second byte of each received word is marked as End-Of-Frame in the receive FIFO. As in the transmitter case, this designation has no meaning in the 1553B protocol but tells the logic in the USC to transfer a receive status block to memory. The received data will be organized in memory.

Note that because each word is marked in the FIFO as End-Of-Frame, the status for the received word may be read from the RCSR after the data is read from the FIFO. Thus, when transferring data under CPU control, the data may be read first followed by a read of the RCSR to determine command/status/data as well as parity error information.

Receive Data Memory Organization

1. First Word Received.
2. First Receive Status Block (Sync Polarity).
3. Second Word Received.
4. Second Receive Status Block (Sync Polarity).

●
●
●

Timing requirements for the receiver are shown in Figure 14, for both start-up and end of a word.

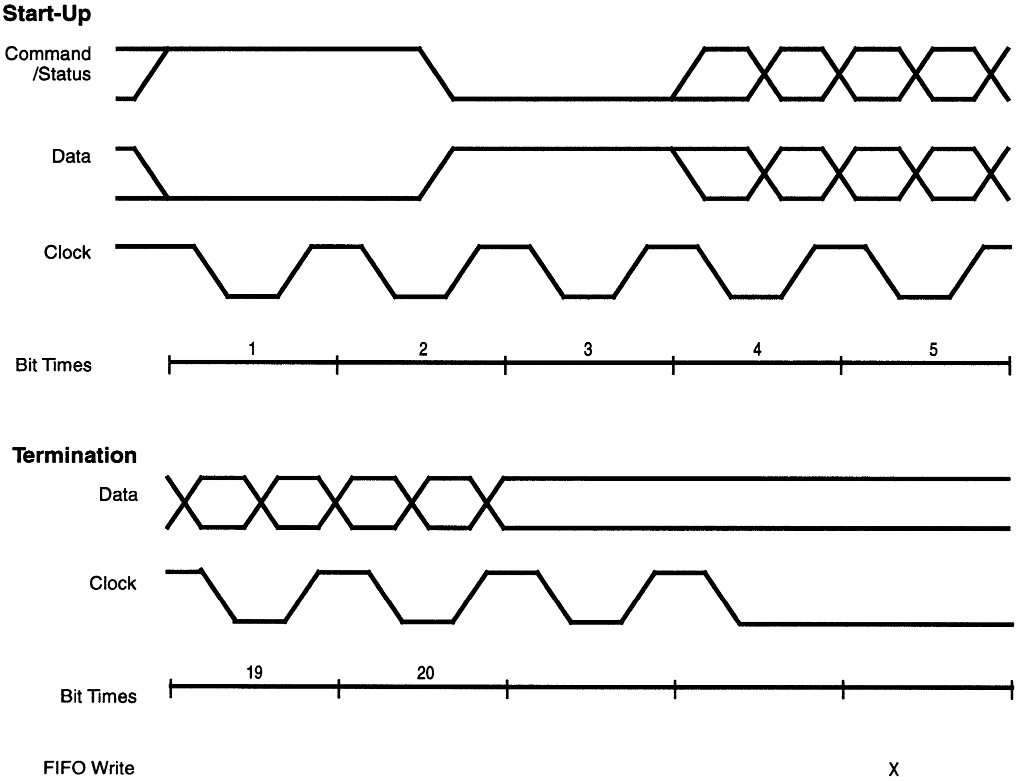


Figure 14. Receiver Clocking Requirements

DPLL: Detailed Operation

The Digital Phase-Locked Loop (DPLL) in the USC is used to generate a clock for the receiver. As mentioned before, the DPLL must be programmed in Bi-phase-Level mode, accepting code violations. In this mode, the DPLL generates a clock which is properly phased to sample the receive data. Also, it persists in time long enough to transfer received data to memory (Figure 15).

The one case that is not handled well by the DPLL is start-up after an idle line. The DPLL assumes that the first edge it sees when in search mode (waiting to sync up) is a valid clock edge. In Bi-phase-Level the valid clock edge is at the center of the bit cell, but notice that the first edge in the sync pattern is a code violation at the bit-cell boundary. Thus, if the DPLL uses this first edge to sync up, the DPLL output is 90 degrees out of phase.

The solution is to create, externally, a fake mid-bit change which precedes the sync pattern into the Receiver/DPLL combination. The required waveforms are shown in Figure 15. The desired waveform is created with a simple state machine running off the 8X clock and using the differential receive signals from the 1553B transceiver.

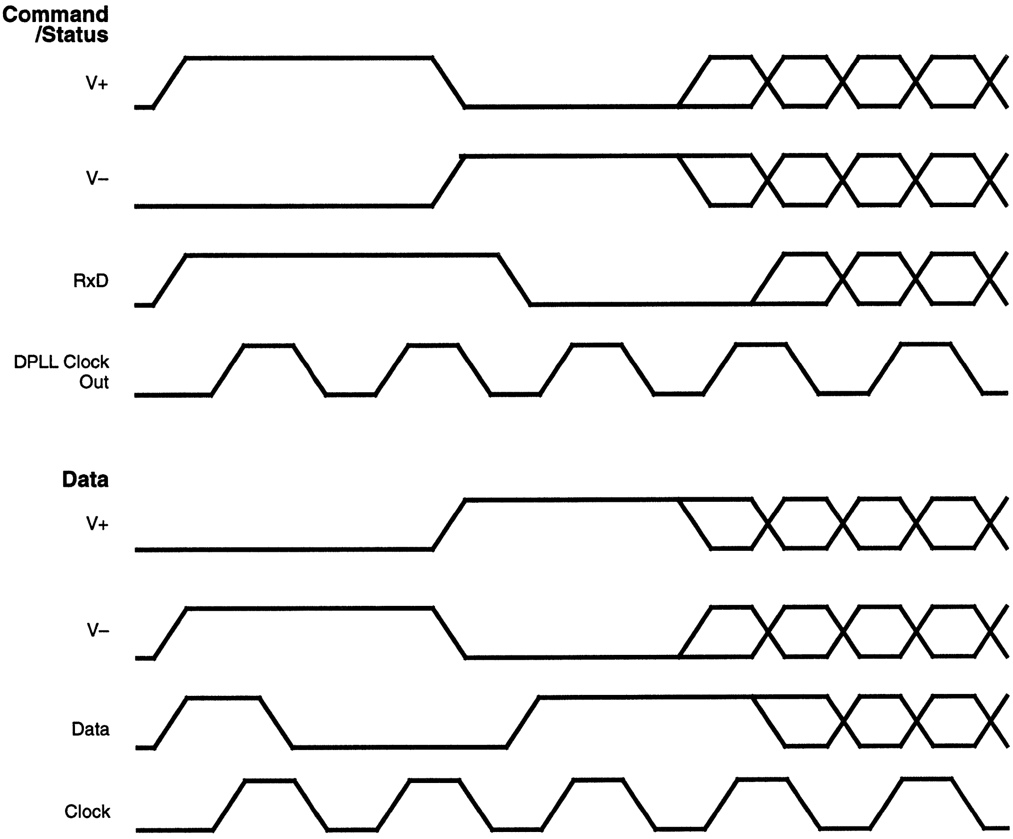


Figure 15. DPLL Idle Line Start-Up Requirements

CONCLUSION

The USC provides a cost-effective, single-chip solution for 1553B communication by providing Manchester encoding and decoding, sync pattern recognition and generation, and parity generation and checking. The DMA transfer of both data and status is supported, along with block data transfers and 32-byte FIFOs. The flexible bus interface and both bit and byte shuffling allow connection to any type of system bus. Connection to a Z16C00 (CMOS Z8000 family CPU) is shown in Figure 16.

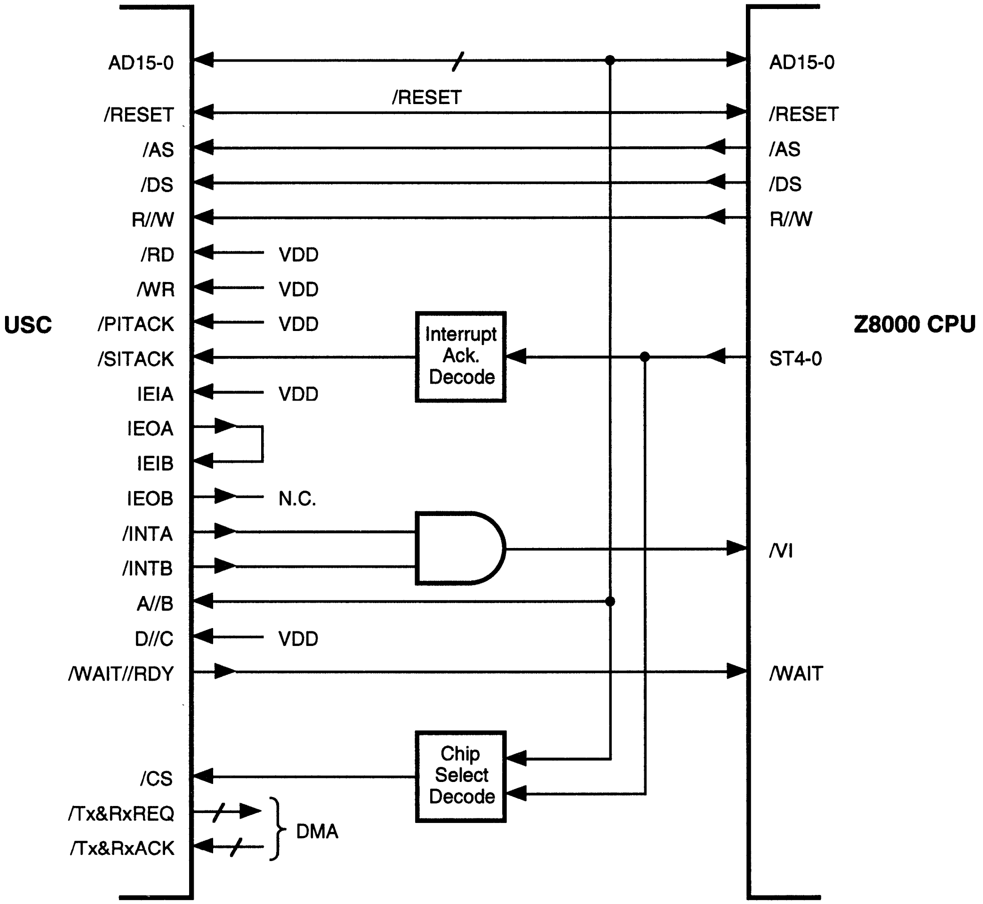


Figure 16. USC to Z8000 CPU Connection (Example)

Design a Serial Board to Handle Multiple Protocols

Eliminate the dedicated protocol controller by creating a systems communication board(s) that minimizes swapping parts and maximizes easy access networking. The Z16C30 USC™ fills the bill with Protocol flexibility and greater throughput.

INTRODUCTION

A new approach to handling multiple serial communications protocols is to eliminate the dedicated protocol controller. Overcoming this more expensive communications method is done by creating a systems' communications board(s) that minimizes swapping parts when changing protocols and maximizes easy access networking. The need for greater protocol flexibility and higher throughput is satisfied by the Z16C30 USC™ (Universal Serial Controller). The USC is a full-duplex, two independent channel communications controller with a 10 Mbit/sec data transfer rate and 12 M byte/sec bus bandwidth for interfacing a network to a systems' communication board (Figure 1).

On the network side of the interface, the only changes necessary to switch from one protocol to another are swapping (per protocol) one receive and one transmit driver, one connector, and one cable. Ten different protocols and eight data encoding formats are supported by the USC. For illustration purposes, this particular system's communication board (hereinafter referred to as Syscom) is set for two protocols. There can be more protocols per board(s), depending upon the total system design needs.

For example, a board for an 80386 PC can simultaneously connect the host PC to another work station and to a laser printer. The board connects to the work station through a synchronous serial port, using the IEEE 802.3 Ethernet Protocol, and to the laser printer through an RS-232 asynchronous port, as shown in Figure 1.

The receivers and drivers used depend upon the type of protocol and signal levels. The cables and connectors are of the standard hardware variety.

On the system's side of the interface, the USC (Figure 2) connects to two DMA controllers (one receive and one transmit per controller for each channel) which in turn connect to the 16-bit bus. The USC accesses the Z8002 CPU (or other microcontroller with its own ROM) and 4K bit x 8 Static RAM through this bus. Another microcontroller with its own ROM can be used in place of the Z8002 chip and ROM.

Notes:
All Signals with a preceding front slash, "/", are active Low, e.g., B//W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

Connection	Circuit	Device
Power	V_{CC}	V_{DD}
Ground	GND	V_{SS}

3

INTRODUCTION (Continued)

A Dual-Channel Communications Board for an IBM 80386 PC consists of four main sections: The communications controller (a Z16C30 Universal Serial Controller), an interface to the PC's bus, buffer memory, and two communications channels. In this application, Channel A drives an Ethernet interface and Channel B drives a laser printer through an RS-232 port (Figure 1).

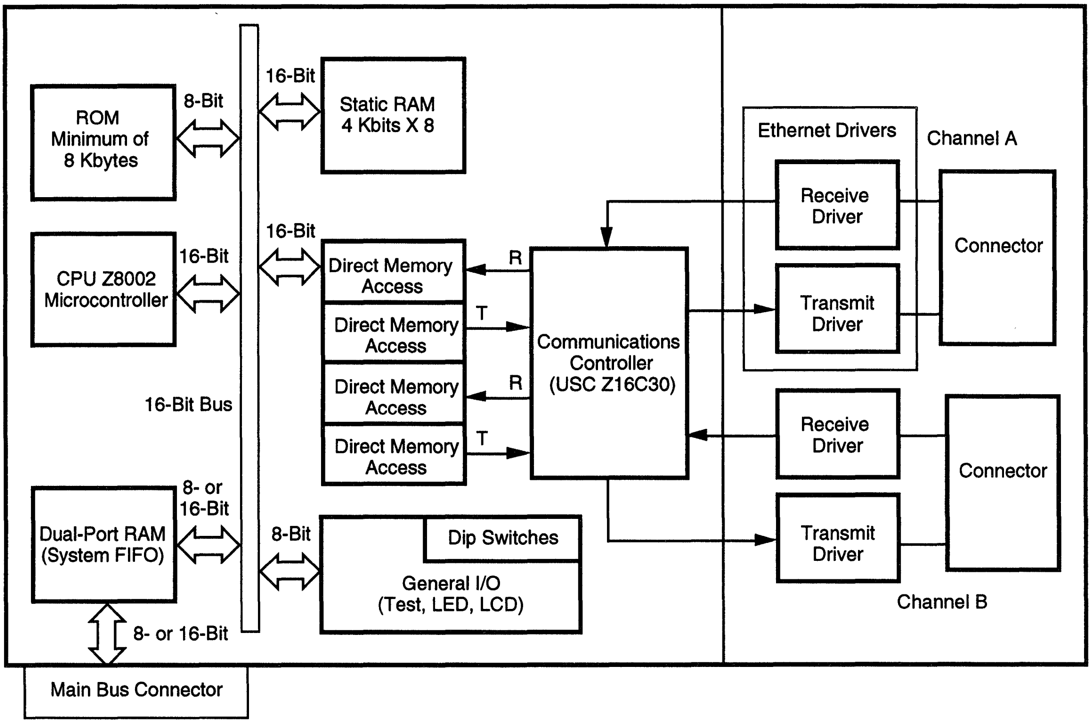


Figure 1. IBM 80386 PC Dual-Channel Communications Board

At the heart of the dual-channel communications board is the Z16C30 Universal Serial Controller chip, which supports 10 different protocols and eight encoding formats. In the upper portion of the diagram, serial data from the receiver is placed in a 32-byte FIFO buffer prior to being placed onto the 16-bit internal bus under DMA control. Similarly, data to be transmitted from the bus is placed in the FIFO transmit buffer and then sent in serial form by the transmitter (Figure 2).

The software is arranged to have the CPU execute from the ROM (minimum of 8K bytes) by first checking the two different protocol settings in the DIP switches which have been set for these particular protocols (Ethernet and RS-232). The protocol initialization procedures are then executed from ROM.

The Dual Port RAM (or System FIFO) buffers the Syscom transactions with the IBM 386 PC host system through the Main Bus connector. The size (byte depth) design of the Dual Port RAM (hereinafter referred to as DPR) depends on the difference in speeds between the host and Syscom. The greater the speed difference, the larger the byte buffering needed. In this case, 32 each of transmit and receive buffering are enough to handle the 80386's 12 MHz speed.

One of the main features of a global system network arrangement is the varying amount of host/slave (source/target) dissimilar nodes (IBM PCs or their clones, MACs, Work Stations, printers, modems, terminals, etc.) communicating with each other. Based on design complexity (state of the art speed, distance between nodes, data crunching needs, etc.), the systems' communication board(s) can carry 4, 8, 16, or even 32 channels—and with differing protocols and formats all working simultaneously. As the number of channels increases, the speed and bandwidth of the serial controller plays an increasingly important role in helping the Syscom's CPU by reducing bus impact which translates into keeping data throughput from being derated.

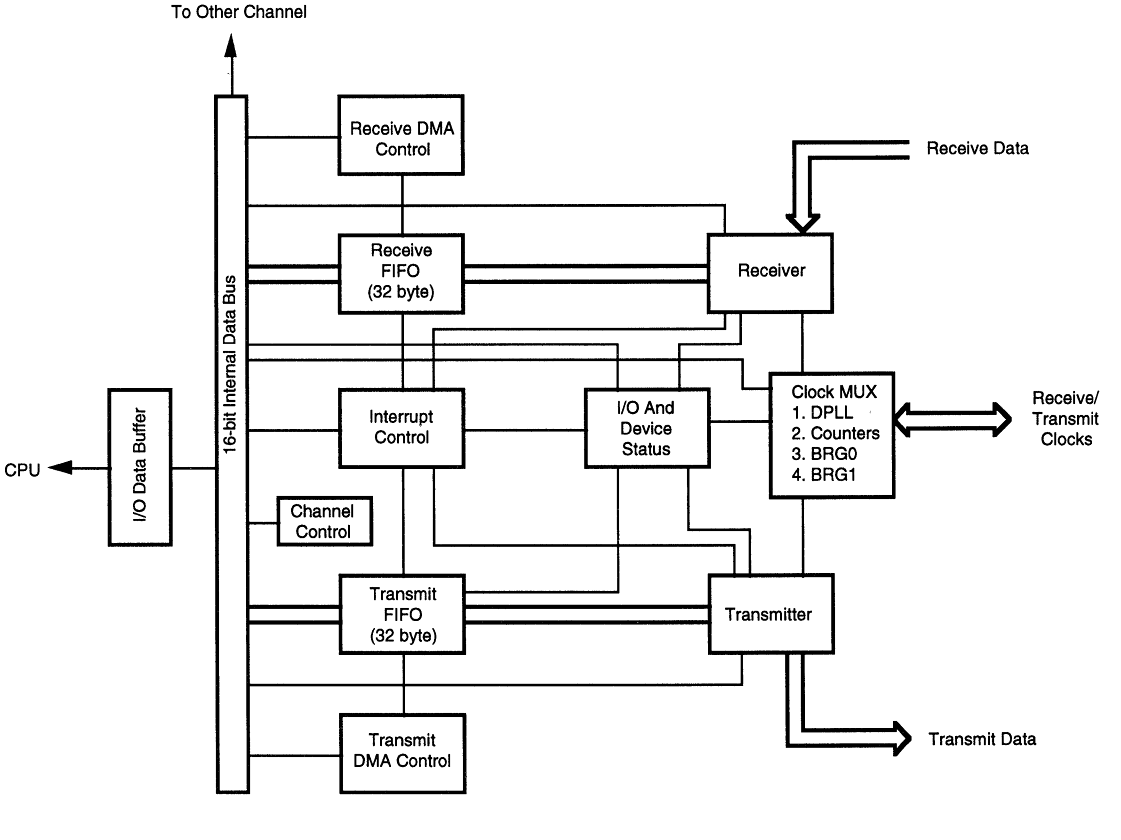


Figure 2. Z16C30 Universal Serial Controller

TWO CHANNEL COMMUNICATIONS OPERATION

The following example demonstrates and illustrates a two channel design for data throughput events. This hypothetical example uses an IBM 386 PC as the host system which is communicating over a network to a slave Work Station through Syscom and an IEEE 802.3 Ethernet protocol (through Channel A) using a synchronous serial port. At the same time, the host is multitasking by sending data through the Syscom (through Channel B) to an RS232 asynchronous port to a laser printer (Figure 3). The Syscom board is the interfacing link for control and data movement between the host and slaves.

Before starting the dynamic interchange of data, all initialization procedures of the Syscom are accomplished. These include Power-On Reset, reading the setting of the DIP switches for the 802.3 Ethernet protocol, and writing to the USC's Clock Mode Control Register to select a clock source for the receiver and transmitter.

While the host performs its normal application functions, the Syscom remains in an open mode waiting for either the network device or the host to request its services. In this example, the IBM 386 PC becomes the host by requesting data from the Work Station. At the same time it's sending data to the laser printer. The first part of the scenario begins with the Work Station (slave) responding to the host's request. The second part explains how the host sends data to the laser printer while simultaneously receiving data from the Work Station (throughout this article refer to Figure 1 and the timing diagrams in Figure 3 and 4).

Part I. Channel A

After initialization procedures, which included the host having control and requesting data from the Work Station, the scenario begins with Channel A receiving serial data from the network. When the network has a message from the Work Station, the network alerts the Syscom CPU that it has data for the target host. The Syscom CPU alerts the host that a message is coming from the network and then acknowledges the request from the network (Figure 4). The network begins to send the IEEE 802.3 raw data (ones and zeros) from the target system.

Ethernet Receive/Transmit Driver Interface

The 802.3 mode implements the data format with a 16-bit address compare. In this mode, DCD (Data Carrier Detect) and CTS (Clear To Send) implement the carrier sense and collision detect interactions with the receiver and transmitter. Figure 3 shows this hardware interface and related timing.

Bit combination 1001 selects Ethernet 802.3 mode through the channel mode register where each message is preceded by a preamble (protocol) and a start bit of one and is terminated with CRC (Cyclic Redundancy Check), without any trailing delimiter. To meet the 802.3 standard, Bi-phase-level (Bi-phase-L) data encoding must be programmed in the Transmit Mode Register (TMR). An idle line condition of mark or space, selected in the TCSR, will allow external logic to terminate the transmit signal due to the absence of a mid-bit transition. In this mode, the CTS input is then used to signal a collision detect to the CPU and disable the transmitter.

The Ethernet interface consists of an ethernet driver, which connects to the communications controller through clock and data lines (Figure 3a). All that's required to change to another communications protocol is replacing the receive and transmit drivers, connectors, and cable. Transfer of Ethernet data to the driver is controlled by the communications controller (Figure 3b). Serial data on the TxD pin is sent on the falling edge of the TxC.

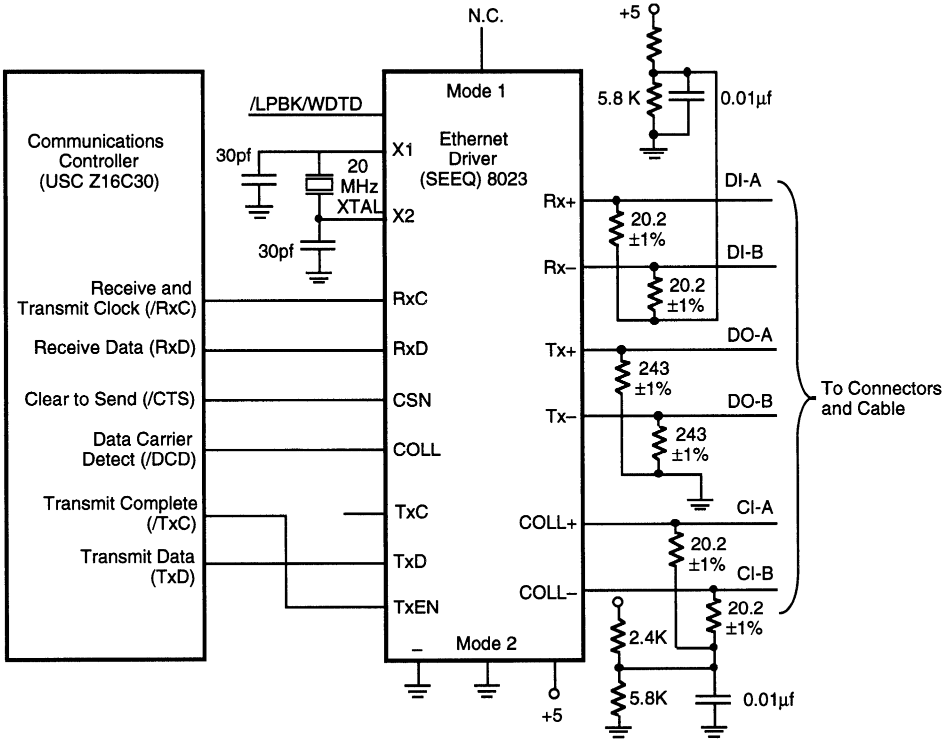
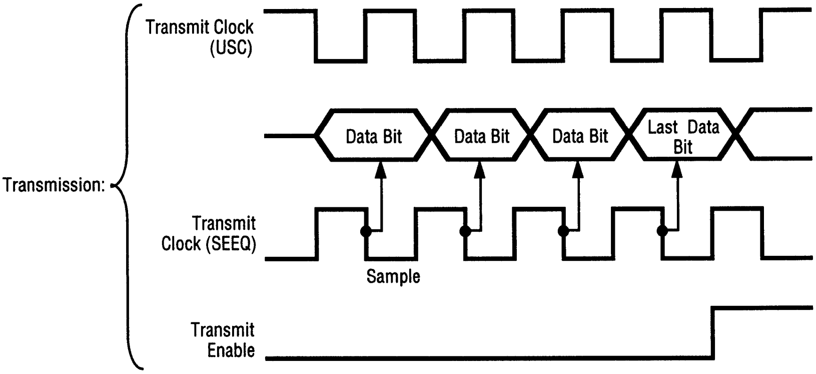


Figure 3a. Ethernet Interface



Note: /Transmit Clock Pin of USC is Programmed to be used as a Signal to Flag the End of Transmit.

Figure 3b. Transfer of Ethernet Data

TWO CHANNEL COMMUNICATIONS OPERATION (Continued)

The USC/Ethernet Interface handles reading and writing of data through the two serial channels. Here, after the controller acknowledges a network request, raw IEEE 802.3 data is read from an Ethernet serial link (Figure 4).

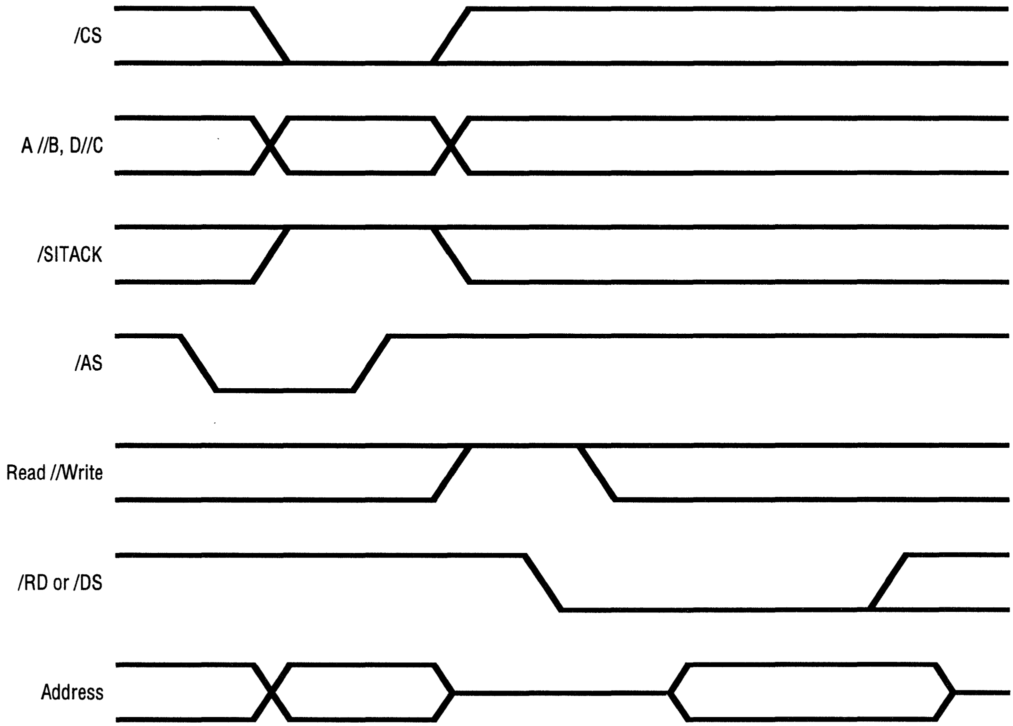


Figure 4. USC/Ethernet Interface and Two Channel Read/Write Data

Bi-phase Encoding

The USC can be programmed to encode and decode the serial data in any of the following seven ways; NRZ, NRZI-M, NRZI-S, BIPHASE-M, BIPHASE-S, BIPHASE-L, and DIFFERENTIAL BIPHASE-L. The transmitter encoding (TMR) method is selected independently of the receiver encoding method (RMR). The DPLL (Data Phase-Locked Loop) is used to decode the receiving data.

Digital Phase-Locked Loop

Each channel in the USC contains a Digital Phase-Locked Loop to recover clock information from a data stream with NRZI or Biphase encoding. The DPLL is driven by a clock that is nominally 8, 16, or 32 times the receive data rate. The DPLL uses this clock, along the data stream, to construct a clock for the data. This clock is then routed to the receiver, transmitter, or both, or to a pin for external use. In all modes, the DPLL counts the input clock to create nominal bit times.

As the clock is counted, the DPLL monitors the incoming data stream for transitions. Whenever a transition is detected, the DPLL makes a counter adjustment (during the next counting cycle), to produce an output clock which tracks the incoming bit cells. The DPLL provides properly phased transmit and receive clocks to the clock multiplexer.

The data is then examined by the USC for the Ethernet protocol preamble of binary bit stream 0101...01011. Any other combination is not recognized. Once recognized, the preamble is stripped off and the USC looks for a 16-bit address which is matched against a preprogrammed address in the USC. When matched, the USC proceeds to accept the rest of the data (If no match, data is not recognized). The first word of the address is stripped off and the following data is stored in the USC's 32 byte receive FIFO buffers.

As the data shifts through the Receive Shift Register, the USC code watches for specific bit patterns, counts bits, and at the appropriate time, transfers data to the receive FIFO. Also, the microcode checks status and generates status interrupts when appropriate.

USC Receiver

The receiver performs all of the functions necessary to convert serial data back to parallel for the processor. Serial data on the RxD is sampled on the rising edge of the /RxC (receive clock) pin for all formats and encoding modes, except for Biphase encoding modes where both /RxC edges are used for data sampling. The serial data is received with the LSB (Least Significant Bit) first. The data, however, can be stored in LSB or MSB first format in the output FIFO's (LSB after reset). It is controlled by the CCAR D15-D11. In addition, the serial receiver can read the serial data as words and put the byte first received in bits 15-8 of the word and the byte which followed to occupy bits 7-0 of the same word (default setting after reset). This order can be swapped, and both arrangements are controlled by the CCAR D15-D11.

Error and status conditions are carried with the data in the receive (and transmit) FIFOs to greatly reduce the Syscom CPU overhead required to receive (or send) a message. Specific, appropriately timed interrupts are available to signal such conditions as overrun, parity error, framing error, end-of-frame, idle-line-received, sync acquired, transmit under run, and others. Such internal signals as receive FIFO load, received sync, transmit FIFO read and transmission complete, can be sent to pins for use by external circuitry.

Interrupts

The Master Interrupt Enable (MIE) bit in the ICR D15 globally enables or disables interrupts within a channel. When the USC responds to an interrupt acknowledge from the CPU, an interrupt vector is placed on the data bus. The vector is held in the Interrupt Vector Register (IVR). To speed interrupt response time, the USC can modify three bits in this vector to indicate which type of interrupt is being requested (IVR). These three bits are maskable by the vector and includes the Status Level Control Field (ICR).

Each of the six sources of interrupts in each USC channel (Receive Status, Receive Data, Transmit Status, Transmit Data, I/O Status and Device Status) has three bits associated with the interrupt source: Interrupt Pending (IP). Interrupt Under Service (IUS) and Interrupt Enable (IE). If the IE bit for a given source is set, then that source can request interrupts. Note that individual sources within the six groups also have interrupt enable bits which must be set (ICR D7-D0). Even though the IE bits can be reset, it will not affect the IP bit being set by the interrupt condition.

The other two bits are related to the interrupt priority chain. A channel in the USC may request an interrupt only when no higher priority interrupt source is requesting one, e.g., when IEI is High for the channel. In this case, the channel activates the /INT signal. The CPU then responds with an interrupt acknowledge cycle and the interrupting channel places a vector on the data bus.

DMA Control

At the trigger point of the USC Receive FIFO buffers (preprogrammed anywhere from 1 to 32 bytes), a DMA request is generated. The DMA controller then requests, through a standard request/acknowledge protocol, the Syscom CPU for the 16-bit bus. Once recognized, the DMA controller begins sending data to the Static RAM or the DPR (system FIFO) in "Flyby Transfer" mode (all 32 bytes sent in one burst).

Interrupt Acknowledge Handshake

USC interrupts occur asynchronously, and to allow time for the internal prioritization of interrupts during an interrupt acknowledge, the USC responds to the interrupt acknowledge with the /IVACK signal when this prioritization is complete. Two different types of response on /IVACK are available, selected by bit one in the BCR. When this bit is reset to 0, the /IVACK signal operates as a /READY signal. When it is set to 1, the /IVACK acts as a /WAIT signal.

The DMA activates the RxACK signal for the Flyby mode. If the SRAM is full, data can go directly to the DPR. If Flyby mode is not required, normal flow-through transfers can be used.

TWO CHANNEL COMMUNICATIONS OPERATION (Continued)

Syscom CPU/Host Handshake

After the Syscom CPU acknowledges (and releases) the DMA request for the 16-bit bus, the CPU then requests the host to receive data. When the host receives the request, it sends an interrupt acknowledgment back to the CPU and enables its own bus and memory for data reception. During this time, the Syscom SRAM has been loading the DPR, which when enabled, transfers data to the host memory.

This chain of events continues uninterrupted (except for Non-Maskable Interrupts) until the Work Station has completed its message to the host. When the host receives the preloaded data decoded by the USC, to produce the terminate transmission signal, the host acknowledges the termination and frees the Syscom CPU. This puts Channel A on the Syscom board back to an idle mode for any new receive/transmit messages.

USC Bus Interface

The USC is unique in that the bus interface for the device provides the resources necessary to interface the USC to virtually any type of bus. The USC directly supports either a 16-bit bus or an 8-bit bus, but may be easily connected to a 32-bit bus as well. The control signals provided allow connection to either a multiplexed address/data type bus or to a separate (non-multiplexed) address and data type bus. Interrupt acknowledge is signaled either through a status line or through a dedicated interrupt acknowledge strobe signal. In addition, fly-by DMA transfers are supported for both receive and transmit.

It is important to note that the USC does not have a clock input. The USC "looks" like a memory rather than a traditional peripheral device. All bus timing information is carried in the control signals for the bus and interrupt. DMA requests are generated asynchronously.

Multiplexed Case

The multiplexed address/data bus carry 16-bit addresses and data to and from the USC (16-bit case). The addresses on AD7-AD0 are latched into the USC on the rising edge of the /AS signal along with chip select and the Interrupt Acknowledge status on the /SITACK signal. Because the register address is latched on every bus transaction, all of the registers in the USC are directly accessible. This bus interface provides the highest bus bandwidth capability for the USC.

The 8-bit multiplexed bus is the same as the 16-bit bus except that AD7-AD0 carry addresses to the USC and 8-bit data to and from the USC (AD15-AD8 is not used). Fly-by data transfers are 8-bits in this case.

Non-Multiplexed Case

AD15-AD0 carry 16-bit data to and from the USC. Only the receive and transmit data registers and the Channel Command/Address Register (CCAR) are directly accessible, with all other registers being accessed by first writing a register address to the CCAR and then accessing the desired register. This bus interface provides the same high bus bandwidth as the multiplexed 16-bit bus when accessing either the transmit or receive data registers. These registers are still accessed directly, in this case using the D/C signal, or through fly-by DMA transfers using /TXACK or /RXACK.

The non-multiplexed 8-bit case enables AD7-AD0 to carry 8-bit data to and from the USC with AD15-AD8 unused. The rest of the explanation is the same as in the 16-bit case except that fly-by DMA transfers are 8-bits.

USC Bus Transactions

The following multiplexed and non-multiplexed bus transactions are described in a general sense without referring to 8- or 16-bit address/data size.

Multiplexed Transactions

During a read transaction the /DS (or /RD) signal, along with R//W selecting a read, strobes the data out of the USC. The register address is latched by the rising edge of the /AS signal, along with an active /CS (Chip Select) and an inactive /SITACK (Status Interrupt Acknowledge) signal.

During a write transaction the /DS (or /WR), along with the R//W selecting a write, strobes the data into the USC. The register address is latched by the rising edge of the /AS signal, along with an active /CS and an inactive /SITACK signal.

There are three different signals available to strobe (read) multiplexed interrupt acknowledge vector transactions from the USC. Signals /DS or /RD read the vector from the USC and set the IUS (Interrupt under Service) bit in the interrupt section. The third signal, /PITACK (dedicated interrupt acknowledge strobe), performs this same function. The kind of interrupt acknowledge function needed determines which one of the three strobes are used.

Non-Multiplexed Transactions

During a non-multiplexed read transaction, the /DS (or /RD) signal, along with R//W selecting a read, strobes the data out of the USC. The leading edge of the strobe signal latches both an active /CS and an inactive /SITACK signal.

During a non-multiplexed write transaction, the /DS (or /WR) signal, along with R//W selecting a write, strobes the data into the USC. The leading edge of the strobe signal latches both an active /CS and an inactive /SITACK signal. In both the multiplexed and non-multiplexed cases of DMA fly-by transactions of reads and writes, the /CS and /SITACK signals are inactive.

Prior to this transmission example, when the host had requested the message from the Work Station, the exact reverse chain of events took place. The host had established the handshake with the Syscom CPU, which in turn requested the DMA controllers. The DMA controllers then requested the USC, which enabled its Transmit FIFO buffers and the data was received and encoded by the USC into the 802.3 protocol and sent out (transmit) to the Work Station through the Ethernet.

USC Transmitter

The transmitter performs all of the necessary functions to convert parallel data from the processor into the appropriate serial bit stream (Figure 3b). Serial data on the TxD pin is sent on the falling edge of the /TxC (transmit clock) pin for all formats and encoding modes except for Biphase encoding mode where both /TxC edges are used for data transmission. The serial data is transmitted with the LSB first.

The data, however, can be stored in LSB or MSB first format in the input FIFO's (LSB after reset). It is controlled by the CCAR D15-D11. Also, the transmitter can be programmed to shift bits 15-8 of a word first (after reset), followed by byte 7-0 of the same word, or vice versa. This function is controlled by the CCAR D15-D11.

Now for a look at what Channel B was doing during Channel A's flow through or Flyby.

PART II. Channel B

The multitasking host required laser printing while it was communicating with the Work Station. Whenever the host is ready to send the message to be printed, it interrupts the Syscom CPU. The Syscom CPU acknowledges the interrupt and enables the DPR receive buffer latches. The host loads the DPR through the main bus connector. When the DPR is full (or when the host sends a finished message signal) and the USC is ready, the CPU enables the DPR data out latches and the message is put onto the 16-bit bus in either byte or word format.

The DMA controller now accepts the data and enables the USC transmit FIFO buffers to store up to 32 bytes. The FIFO byte level is programmed into the Transmit Interrupt Control Register (TICR) which generates a DMA request whenever the level drops below this predetermined mark. The USC then performs the necessary functions of preparing the data to be transmitted out Channel B to the network. The preparations include transmit status interrupts, I/O status interrupts (which are independent of the programmed functions), and the device status interrupts. The device status interrupts have four individually enabled sources; receive character count FIFO overflow, DPLL (Digitally Phased Locked Loop) sync acquired, plus BRG1 and BRG2 zero count.

Transmission Preparation

Two other areas of data manipulation by the USC, needed to prepare for network transmission, are parallel to serial conversion and encoding the data for asynchronous transmission.

When the data is ready, the USC sends it to the Channel B transmit driver which is set for RS232 asynchronous mode. The RS232 has the proper voltage levels and transmits the serial data onto the network. The laser printer queue's the data in its buffers. If the printer buffers are full, an error message is sent back to the USC. The USC can either periodically continue to send the data or wait for a buffer's empty signal from the printer.

The USC's Programming Particulars

The registers in each USC channel must be programmed by the Syscom CPU to configure the channels. Before this happens, Syscom program's the bus interface by writing to the Bus Configuration Register (BCR) shown in Figure 5. Each channel is controlled by a set of thirty 16-bit registers which are almost all readable and writable. The BCR is the 16-bit register in the bus interface which configures the type of bus interface. It has no specific address and is only accessible immediately after a hardware reset of the device.

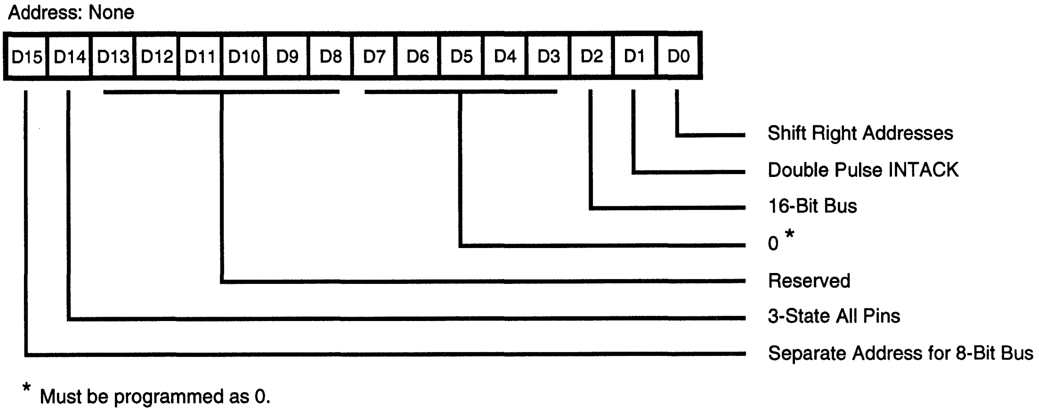


Figure 5. Bus Configuration Register (BCR)

Register Access

USC registers (Table 1) are accessed explicitly, directly, or indirectly, depending on the bus type and the specific control signals used. In all cases, the B//W signal selects between a byte access (B//W High) and a word access (B//W Low). When an 8-bit bus is selected, the B//W signal is always forced to a 1, selecting a byte access. The U//L signal chooses between the upper byte (U//L High) and the lower byte (U//L Low) in the case of a byte access. U//L is always Low for a word access.

Only three registers in the USC have explicit addressing; These are the BCR, for the first write after a hardware reset; the RDR, either through a read with the D//C signal high, or by a fly-by DMA read; and the TDR, either by a write with the D//C signal high, or by a fly-by DMA write.

In the non-multiplexed bus case, only the CCAR is accessed directly, while in the multiplexed bus case all USC registers are accessed directly. Further, when a separate address bus is used, all registers are directly addressed.

The first write to the USC, after a hardware reset, programs the BCR. After that, the normal channel registers are accessed.

Multiplexed Bus

In the multiplexed bus case, all registers are directly addressable, through the address latched by the Address Strobe (AS) at the beginning of a bus transaction. The address is decoded from either AD6-AD0 (Shift Right) or AD7-AD1 (Shift Left). This is controlled by the Shift Right/ Shift Left bit in the BCR. The D/C pin is still used to directly access the receive and transmit data registers (RDR and TDR) in the multiplexed bus; if D/C is High, the address latched by AS is ignored and an access of RDR or TDR is performed.

Multiplexing data and address onto the same lines makes more efficient use of pins and facilitates expansion of the number of data and address bits. Multiplexing also allows straight-forward addressing of a peripheral's internal registers, which greatly simplifies I/O programming.

Non-Multiplexed Bus

In the non-multiplexed bus case, the registers in each channel are accessed indirectly using the address pointer in the Channel Command/Address Register (CCAR) of each channel. The address of the desired register is first written to the CCAR and then the selected register is accessed; the pointer in the CCAR is automatically cleared after this access. The RDR and TDR are still accessed directly using the D/C pin, without disturbing the contents of the pointer in the CCAR.

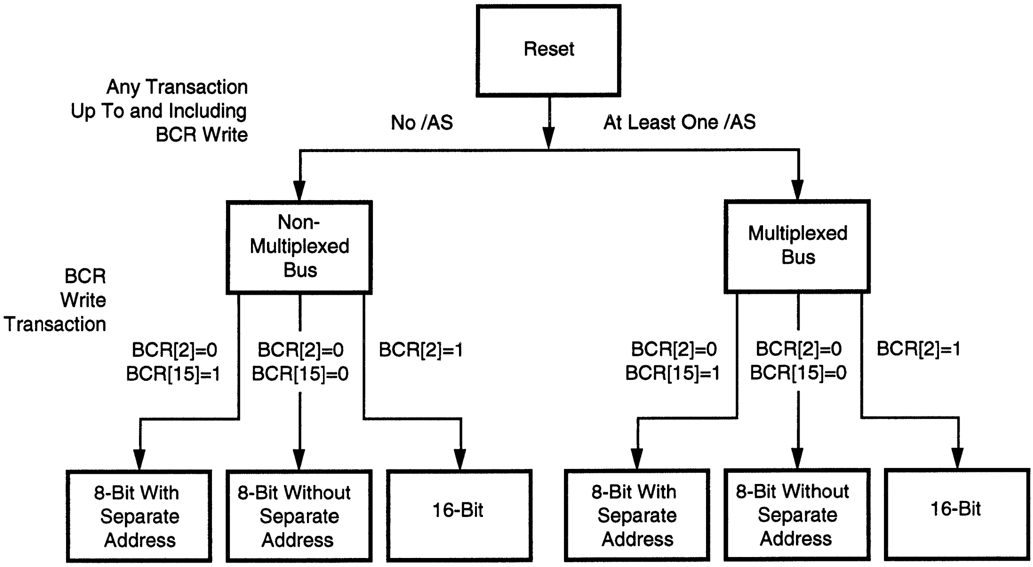
Polling

Polling is a software method of avoiding interrupts and is the simplest mode to implement. In this mode, the software must poll the USC to determine when data is to be written or read to or from the USC. All interrupts have to be disabled (ICR D15). The registers in the USC are automatically updated to reflect current status. The CPU polls the Daisy-Chain Control Register (DCCR) to determine status changes and then reads the appropriate status register to find and respond to the change in status. USC status bits are grouped according to function to simplify this software action.

Two Important Points

There are two important points to note about the USC. First, the Channel Reset bit in the CCAR places the channel in the reset state. To exit this reset state, either a word or all zeros is written to the CCAR (16-bit bus) or a byte of all zeros is written to the lower byte of the CCAR (8-bit). Secondly, after reset, the transmit and receive clocks are not connected. Therefore, upon initialization, a write to the Clock Mode Control Register (CMCR) establishes a clock source for the receiver and transmitter.

Register addresses are shown in Table 1 and the bit assignments for the registers are shown in Figure 6.



Note:
The presence of one transaction with an /AS active, between reset up to and including the BCR write, chooses a multiplexed type of bus.

Figure 6. BCR Reset Sequence and Bit Assignments

Table 1. Register Address List

Address A4-A0	Register	Function
00000	CCAR	Channel Command/Address Register
00001	CMR	Channel Mode Register
00010	CCSR	Channel Command/Status Register
00011	CCR	Channel Control Register
00110	TMDR	Test Mode Data Register
00111	TMCR	Test Mode Control Register
01000	CMCR	Clock Mode Control Register
01001	HCR	Hardware Configuration Register
01010	IVR	Interrupt Vector Register
01011	IOCR	I/O Control Register
01100	ICR	Interrupt Control Register
01101	DCCR	Daisy-Chain Control Register
01110	MISR	Misc Interrupt Status Register
01111	SICR	Status Interrupt Control Register
1X000	RDR	Receive Data Register (Read Only)
10001	RMR	Receive Mode Register
10010	RCSR	Receive Command/Status Register
10011	RICR	Receive Interrupt Control Register
10100	RSR	Receive Sync Register
10101	RCLR	Receive Count Limit Register
10110	RCCR	Receive Character Count Register
10111	TC0R	Time Constant 0 Register
1X000	TDR	Transmit Data Register (Write Only)
11001	TMR	Transmit Mode Register
11010	TCSR	Transmit Command/Status Register
11011	TICR	Transmit Interrupt Control Register
11100	TSR	Transmit Sync Register
11101	TCLR	Transmit Count Limit Register
11110	TCCR	Transmit Character Count Register
11111	TC1R	Time Constant 1 Register
XXXXX	BCR	Bus Configuration Register

DATACOMMUNICATIONS IUSC™/MUSC™ TIME SLOT ASSIGNER

Use the IUSC™/MUSC™ for ISDN and Fractional T1 in High Speed Time-Multiplexed Datacommunications to Process One of a Set of Independent Voice and Data Streams.

INTRODUCTION

The networking of electronic equipment is becoming an increasingly important area of datacommunications/telecommunications. Each new technical innovation in these expanding fields enhances overall performance. This App Note explains the functions of the TSA (Time Slot Assigner) of Zilog's MUSC/IUSC as another technical building block for the networking world.

In applications such as ISDN and Fractional T1, a high speed link is time-multiplexed among a set of independent voice and data streams. The IUSC can send and/or receive such a data stream with the aid of its Transmit and Receive TSA (Time Slot Assigner) logic (TTSA and RTSA).

Many telecom applications employ time division multiplexing to combine lower-speed serial bit streams into higher-speed serial bit streams for long distance transmission. The common technique used is to transmit a fixed amount of data from each of the lower speed serial bit streams, one after the other, to create a continuous higher-speed bit stream.

For example, to create a 1.544 MHz T1 serial bit stream, one 8 kHz serial channel, and twenty-four 64 kHz serial channels, are combined using time-division multiplexing. One bit from the 8 kHz channel is combined with eight bits from each of the serial channels to form a 193 bit "frame," with 8000 frames sent per second.

Hence, to transmit or receive on a given serial channel in this format, the boundaries of the frame must be known. The 8 kHz serial channel, through repeating bit patterns, provides this "framing" information. Once this "frame sync" information has been acquired and provided to the IUSC/MUSC, the time slot assigners within the device can be programmed to selectively receive or transmit only during the programmed time slots.

Notes:
All Signals with a preceding front slash, "/", are active Low, e.g., B//W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

Connection	Circuit	Device
Power	V_{CC}	V_{DD}
Ground	GND	V_{SS}

3

TSA APPLICATION

To use the IUSC in such an application, external logic must determine the start point of (or at least a consistent point in) each cycle during the overall data stream. Then, the external logic signals the IUSC when this point occurs, using a pulse on the PORT6/FSYNC pin that is Low for one period of RxCLK and/or TxCLK. This pulse is used by both the Receive and Transmit Time Slot Assigners. Note that if both the Receiver and Transmitter are operating simultaneously in such an application, the IUSC assumes that they are both operating in (different parts of) the same overall data stream. This means that RxCLK and TxCLK is selected from the same source.

Figure 1 shows how the Time Slot Assigners determine when to start receiving and/or transmitting in each cycle. After sensing the /FSYNC pulse, the RTSA waits for a number of RxCLK cycles (bit times) that are determined by the values programmed into the RTSASlot and RTSAOff fields in the Receive Interrupt Control Register (RICR). The number of RxCLK cycles (bits) waited is eight times the value in RTSASlot, plus the value in RTSAOff.

After blocking RxCLKs to the Receiver for this number of bits (or right after the FSYNC pulse if both fields contain zero), the RTSA allows RxCLK to reach the Receiver for the number of consecutive bytes/octets/slots programmed into the RTSACnt field in RICR. That is, it allows eight (RTSACnt) RxCLKs to reach the Receiver (Figure 2). (If the RTSACnt field is zero, the whole RTSA feature is disabled.) Now, the RTSA again blocks RxCLKs to the Receiver until after the next pulse on /FSYNC.

The net result of this clock gating is that the IUSC can receive up to 15 consecutive bytes/octets out of each cycle on the serial link, starting at any point within the (first) 128 octets of each cycle. It also allows for possible delays in sensing and signalling the frame sync.

In ISDN circles it is common parlance to refer to the 128 octets in each frame as "slots" which are numbered from 0 through 127. Given this definition of slot number, if the frame sync detection logic allows /FSYNC to be sampled Low in the bit time before RxD is sampled for the first bit of the first slot, then RTSAOff is programmed with zero; RTSASlot is programmed with the slot number of the first octet that is received. Otherwise, call the FSync delay, one, if /FSYNC is sampled Low in the same bit time that the first bit of the first slot is available on RxD; two, if /FSYNC is Low in the bit time after the first bit appears on RxD; and so on up through the maximum value of seven, i.e., if /FSYNC is Low six bit times after the first bit of the first slot appears on RxD. In cases where the first slot cannot be received, program the RTSAOff field with eight minus the FSync delay; program RTSASlot with the slot number of the first octet that is received, minus one.

Figure 1 applies equally to the transmit side. The TTSA blocks TxCLKs to the Transmitter for a number of TxCLK cycles that are determined from the TTSASlot and TTSAOff fields in the Transmit Interrupt Control Register (TICR).

After blocking TxCLKs for eight (TTSASlot) + (TTSAOff) bits, the TTSA allows TxCLK to reach the Transmitter for the number of consecutive bytes/octets/slots programmed into the TTSACnt field in the Transmit Interrupt Control Register (TICR). That is, it allows eight (TTSACnt) TxCLKs to reach the Transmitter. (Figure 2 - on the receive side, if the TTSACnt field contains zero, the whole TTSA feature is disabled.) The TTSA again blocks TxCLKs to the Transmitter until after the next pulse on /FSYNC.

Thus, symmetrically with the receive side, the IUSC transmits up to 15 consecutive bytes/octets/slots in each cycle on the serial link. This occurs while starting at any point within the (first) 128 octets of each cycle, plus allowing for possible delays in sensing and signalling the frame sync.

Since the IUSC maintains output drive on TxD throughout each cycle on the serial link, an external driver with an enable/disable input is needed to transmit in this kind of time-multiplexed environment. The IUSC provides the required Transmit Gate signal on the PORT 4 pin. This signal goes Low while the TTSA is enabling the Transmitter in each frame. There is also a similar facility whereby the RTSA's active Low Receive Gate signal can be output on the PORT 3 pin, but the application of this signal is less obvious. As already noted in the section on the PORT pins, these options are enabled by programming the P4 MODE and/or P3 MODE fields of the Port Control Register (PCR9-PCR8 and/or PCR7-PCR6, respectively) as 01.

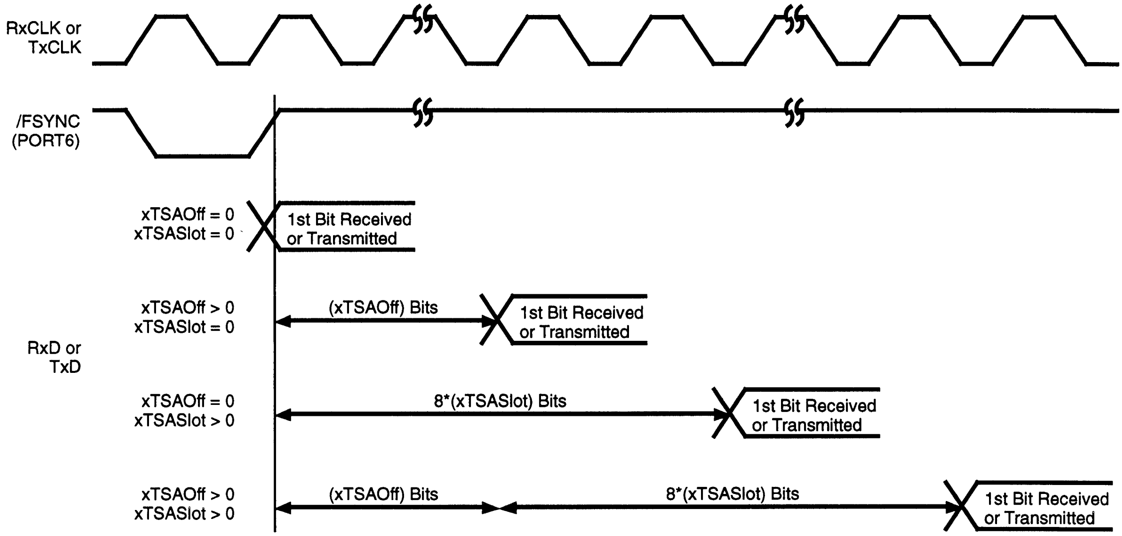


Figure 1. Start of Received or Transmitted Data in a TSA Application

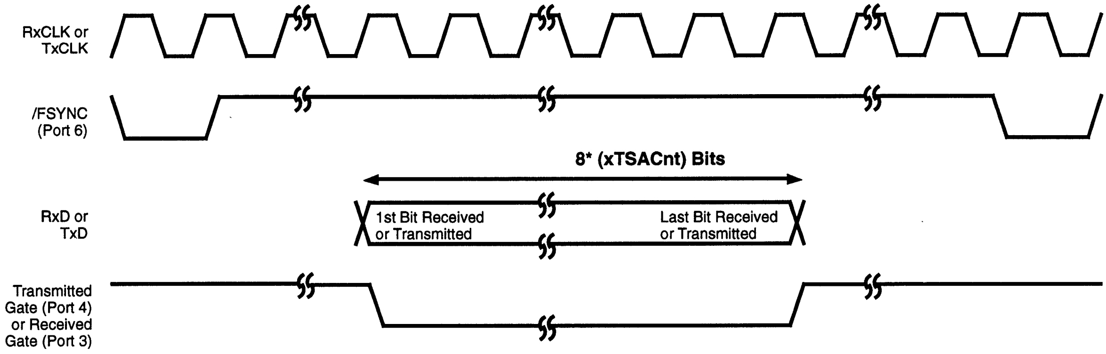

RxCLK or TxCLK
/FSYNC (Port 6)
8* (xTSACnt) Bits
RxD or TxD
1st Bit Received or Transmitted
Last Bit Received or Transmitted
Transmitted Gate (Port 4) or Received Gate (Port 3)

Figure 2. Length of Received or Transmitted Data in a TSA Application

PROGRAMMING THE TIME SLOT ASSIGNERS

There is an intentional vagueness in the preceding description of the Time Slot Assigner control fields as being "in" the Receive and Transmit Interrupt Control Registers (RICR and TICR). These two registers are somewhat more complex than other IUSC registers. This section describes how to access the TSA fields.

The Least Significant Byte (bits 7-0) of both the RICR and TICR contain fixed data, but any of five different internal registers can be selected as the Most Significant Byte of each register (Figure 3). At the first level of data structure, the contents of RICR15-RICR8 are selected by means of four of the commands that are written to the RCMD field of the Receive Command/Status Register (RCSR15-RCSR12); the contents of TICR15-TICR8 are selected by four of the commands written to the TCMD field of the Transmit Command/Status Register (TCSR15-TCSR12). The encoding of both sets of commands is the same:

xCMD	Contents of xICR15-xICR8
0100	xTSA data
0101	Current xFIFO Level
0110	xFIFO Level for Interrupt
0111	xFIFO Level for DMA Request

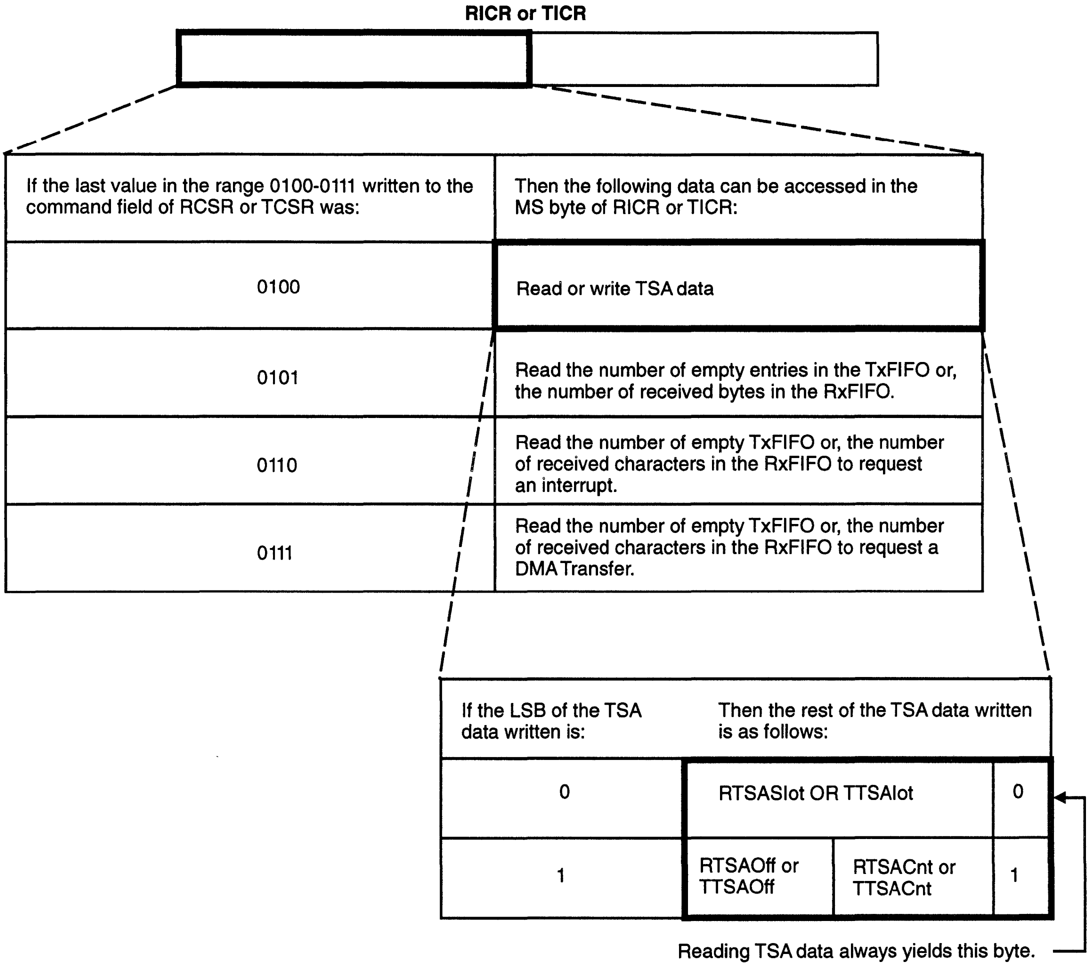
Note:
Where "x" stands for either "R" or "T".

The other options are discussed in subsequent chapters; for our purposes it is sufficient to note that TSA data is read and written as xICR15-xICR8, if the 0100 command has been written to xSCR15-xSCR12 more recently than 0101, 0110, or 0111. The IUSC resets to read the Current FIFO level in both the RICR and TICR.

Figure 3 also shows how a second level of data structuring determines the meaning of TSA data. For write operations, the destination of the data is determined by the Least Significant Bit of the Most Significant Byte of data written:

xICR8 Value	Destination of xICR15-xICR9
0	xICR15-9 -> xTSASlot
1	xICR15-13 -> xTSAOff
1	xICR12-9 -> xTSACnt

Reading TSA data from RICR or TICR always yields the xTSASlot value, with the Least Significant Bit of the Most Significant Byte equal to zero.



If the last value in the range 0100-0111 written to the command field of RCSR or TCSR was:	Then the following data can be accessed in the MS byte of RICR or TICR:
0100	Read or write TSA data
0101	Read the number of empty entries in the TxFIFO or, the number of received bytes in the RxFIFO.
0110	Read the number of empty TxFIFO or, the number of received characters in the RxFIFO to request an interrupt.
0111	Read the number of empty TxFIFO or, the number of received characters in the RxFIFO to request a DMA Transfer.

If the LSB of the TSA data written is:	Then the rest of the TSA data written is as follows:		
0	RTSASlot OR TTSAlot		0
1	RTSAOff or TTSAOff	RTSACnt or TTSACnt	1

Figure 3. Structure of RICR and TICR

SUMMARY

To set up xTSA, first write the 0100 command to the xCMD field of the xSCR; write the xTSASlot value to the Most Significant Byte of xICR with the Least Significant Bit of the byte equal to 0; write the xTSAOff and xTSACnt values to the Most Significant Byte of xICR with the Least Significant Bit of the byte equal to 1.

Z16C3001ZCO EVALUATION BOARD
PRODUCT SPECIFICATION

SUPPORTED DEVICE: Z16C30

DESCRIPTION

The kit contains an assembled PC/XT/AT circuit board with two high-speed serial connections, DB9 and DB25 connectors selectively driven by RS-232 or RS-422 line drivers. The kit also contains software and documentation to support software and hardware development for Zilog's USC™.

The board illustrates the use of Zilog's USC device in a variety of communication applications such as ASYNC, SDLC/HDLC and high-speed ASYNC.

SPECIFICATIONS

Power Requirements

+5 Vdc @ .5 A

Dimensions

Width: 4.5 in. (11.43 cm)
Length: 6.5 in. (16.51 cm)

Serial Interface

DB9 and DB25 connectors selectively driven by RS-232C or RS-422 at selectable baud rates.

KIT CONTENTS

Z16C30 Evaluation Board

CMOS Z16C30 USC
RS-232C and RS-422 line drivers
DB9 and DB25 Interfaces

Software (IBM® PC Platform)

Source and executable codes to run the USC in SDLC/HDLC or ASYNC mode. All codes are written and compiled using Microsoft® Quick C 2.5.

Documentation

Z16C30 Product Specifications
Z16C30 Technical Manual
Z16C3001ZCO Kit User Guide

ORDERING INFORMATION

Part No: Z16C3001ZCO

Z8018600ZCO EVALUATION BOARD
PRODUCT SPECIFICATION

SUPPORTED DEVICES: Z8X30, Z85230, Z85233, Z8XC30, Z16C30, Z16C32

DESCRIPTION

The kit contains an assembled circuit board, software, and documentation to support the evaluation and development of code for Zilog's Z85C30 SCC, Z85230 ESCC™, Z85233 EMSCC™, Z16C30 USC™, Z16C32 IUSC™, and the Z16C35 ISCC™. The purpose of the board is to illustrate how the Datacom family interfaces and communicates with the 80186 CPU. This will help potential customers evaluate Zilog's datacommunication controllers in an Intel® environment. A board-resident monitor program allows code to be downloaded and executed.

SPECIFICATIONS

Power Requirements

+5 Vdc @ .50A

Dimensions

Width: 8.4 in. (21.34 cm)
Length: 9.3 in. (23.62 cm)

Serial Interfaces

RS-232C, RS-422

KIT CONTENTS

Z8018600ZCO Evaluation Board

Intel 80186 Integrated 16-bit MPU @ 16 MHz
CMOS Z85230 ESCC
CMOS Z16C30 USC
CMOS Z16C32 IUSC
CMOS Z16C35 ISCC
2 (64K) 8Kx8 EPROMs
6 (256K) 32Kx8 SRAMs
RS-232C, RS-422, and Apple® LocalTalk™ line drivers
DB9, DB25, and DIN 8 Interfaces

Cables

1 25-pin RS-232C Cable
14 Jumper Wires

Software (IBM® PC Platform)

Resident Monitor for download and execution (80186 Assembler source code)
PC-board terminal emulator
Z85230, Z16C32, and Z16C35
Examples Software (All codes written in "C" and compiled using the Microtec® C compiler.)

Documentation

Z85230 ESCC Product Specification and Technical Manual
Z16C30 USC Product Specification and Technical Manual
Z16C32 IUSC Product Specification and Technical Manual
Z16C35 ISCC Product Specification and Technical Manual
Datacom Evaluation Board Application Note

ORDERING INFORMATION

Part No: Z8018600ZCO

Introduction I

Z16C30 CMOS Universal Serial Controller (USC™) 1

Z16C32 Integrated Universal Serial Controller (IUSC™) 2

Application Notes and Support Products

Superintegration™ Products Guide **S**

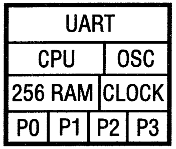
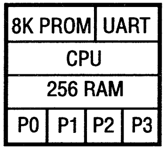
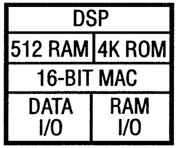
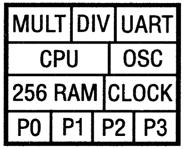
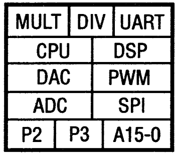
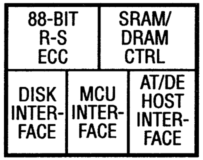
Literature Guide and Third Party Support L

Zilog Sales Offices Representatives and Distributors Z

ZiLOG Fax/Modem Superintegration™ Products Guide

	Data Pump	Single Chip		Controllers			
Block Diagram	DSP 512 RAM | 4K ROM 16-BIT MAC DATA I/O | RAM I/O	Z8 | DSP 24K ROM | 4K WORD ROM 256 BYTES RAM | 512 WORD RAM 8-Bit A/D | 10-Bit D/A	Z8 | DSP 4K WORD ROM 256 BYTES RAM | 512 WORD RAM 8-BIT A/D | 10-BIT D/A	PIO | CGC WDT SIO | CTC Z80 CPU	24 I/O ESCC (2 CH) | 16550 MIMIC S180	Z80 CPU | 2 DMA 2 UART 2 C/T C/Ser MMU | OSC	ESCC
Part #	**Z89C00**	**Z89120**	**Z89920**	**Z84C15**	**Z80182**	**Z80180**	**Z85230**
Description	16-Bit Digital Signal Processor	Zilog Modem/Fax Controller (ZMFC)	Zilog Modem/Fax Controller (ZMFC)	IPC/EIPC Controller	Zilog Intelligent Peripheral (ZIP™)	High-performance Z80® CPU with peripherals	Enhanced Serial Com. Controller
Process/Speed	CMOS 10, 15 MHz	CMOS 20 MHz	CMOS 20 MHz	CMOS 6, 10,16 MHz	CMOS 16, 20 MHz	6, 8, 10, 16*, 20* *Z8S180 only	CMOS 8, 10,16, 20 MHz
Features	16-bit Mac 75 ns 2 data RAMs (256 words each) 4K word ROM 64Kx16 Ext. ROM 16-bit I/O Port 74 instructions Most single cycle Two conditional branch inputs, two user outputs Library of software macros available zero overhead pointers	Z8® controller with 24 Kbyte ROM 16-bit DSP with 4K word ROM 8-bit A/D 10-bit D/A (PWM) Library of software macros available 47 I/O pins Two comparators Independent Z8® and DSP Operations Power-Down Mode	Z8 w/64K external memory DSP w/4K word ROM 8-bit A/D 10-bit D/A Library of macros 47 I/O pins Two comparators Independent Z8® and DSP Operations Power-Down Mode	Z80® CPU, SIO, CTC WDT, CGC The Z80 Family in one device Power-On Reset Two chip selects 32-bit CRC WSG EV mode[1] 3 and 5 Volt Version	Complete Static Version of Z180™ plus ESCC (2 channels of Z85230) 16550 MIMIC 24 Parallel I/O Emulation Modes[1]	Enhanced Z80® CPU MMU 1 Mbyte 2 DMAs 2 UARTs with BRGs C/Serial I/O Port Oscillator Z8S180 includes; Pwr dwn, Prgmble EMI, divide-by-one clock option	Full dual-channel SCC plus deeper FIFOs: 4 bytes on Tx 8 bytes on Rx DPLL counter per channel Software compatible to SCC
Package	68-pin PLCC 60-pin VQFP	68-pin PLCC	68-pin PLCC	100-pin QFP 100-pin VQFP	100-pin QFP 100-pin VQFP	64-pin DIP 68-pin PLCC 80-pin QFP	40-pin DIP 44-pin PLCC
Other Applications	16-bit General-Purpose DSP TMS 32010/20/25 applications	Multimedia-Audio Voicemail Speech Storage and Transmission Modems FAXes, Sonabouys	Multimedia-Audio Voicemail Speech Storage and Transmission Modems FAXes, Sonabouys	Intelligent peripheral controllers Modems	General-Purpose Embedded Control Modem, Fax, Data Communications	Embedded Control	General-Purpose datacom. High performance SCC software compatible upgrade

Zilog Mass Storage Superintegration™ Products Guide

Block Diagram	 UART CPU OSC 256 RAM CLOCK P0 P1 P2 P3		 DSP 512 RAM 4K ROM 16-BIT MAC DATA I/O RAM I/O	 MULT DIV UART CPU OSC 256 RAM CLOCK P0 P1 P2 P3	 MULT DIV UART CPU DSP DAC PWM ADC SPI P2 P3 A15-0	 88-BIT R-S ECC SRAM/ DRAM CTRL DISK INTER-FACE MCU INTER-FACE AT/DE HOST INTER-FACE
Part #	**Z86C91/Z8691**	**Z86E21**	**Z89C00**	**Z86C93**	**Z86C95**	**Z86018**
Description	ROMless Z8®	Z8® 8K OTP	16-Bit Digital Signal Processor	Enhanced Z8®	Enhanced Z8® with DSP	Zilog Datapath Controller (ZDPC)
Process/Speed	CMOS 16 MHz (C91) NMOS 12 MHz (91)	CMOS 12, 16 MHz	CMOS 10, 15 MHz	CMOS 20, 25 MHz	CMOS 24 MHz	CMOS 40 MHz
Features	Full duplex UART 2 Standby Modes (STOP and HALT) 2x8 bit Counter/Timer	8K OTP ROM 256 Byte RAM Full-duplex UART 2 Standby Modes (STOP and HALT) 2 Counter/Timers ROM Protect option RAM Protect option Low EMI option	16-bit Mac 75 ns 2 data RAMs (256 words each) 4K word ROM 64Kx16 Ext. ROM 16-bit I/O Port 74 instructions Most single cycle Two conditional branch inputs, two user outputs Library of software macros available zero overhead pointers	16x16 Multiply 1.7 µs 32x16 Divide 2.0 µs Full duplex UART 2 Standby Modes (STOP and HALT) 3 16-bit Counter/Timers Pin compatible to Z86C91 (PDIP)	8 channel 8-bit ADC, 8-bit DAC 16-bit Multiply/Divide Full duplex UART SPI (Serial Peripheral Interface) 3 Standby Modes (STOP/HALT/PAUSE) Pulse Width Modulator 3x16-bit timer 16-bit DSP slave processor 83 ns Mult./Accum.	Full track read Automatic data transfer (Point & Go®) 88-bit Reed Solomon ECC "on the fly" Full AT/IDE bus interface 64 KB SRAM buffer 1 MB DRAM buffer Split data field support 100-pin VQFP package JTAG boundary scan option Up to 8 KB buffer RAM reserved for MCU
Package	40-pin DIP 44-pin PLCC 44-pin QFP	40-pin DIP 44-pin PLCC 44-pin QFP	68-pin PLCC 60-pin VQFP	40-pin DIP 44-pin PLCC 44-pin QFP 48-pin VQFP	80-pin QFP 84-pin PLCC 100-pin VQFP	100-pin VQFP 100-pin QFP
Application	Disk Drives Modems Tape Drives	Software Debug Z8® prototyping Z8® production runs Card Reader	Disk Drives Tape Drives Servo Control Motor Control	Disk Drives Tape Drives Modems	Disk Drives Tape Drives Servo Control Motor Control	Hard Disk Drives

Zilog Telephone Answering Devices Superintegration™ Products Guide

Block Diagram	ROM UART 8611 | CPU COUNTER/ TIMERS | RAM P0 | P1 | P2 | P3	4K ROM CPU WDT | 236 RAM | P1 P2 | P3 | P0	Z8 | DSP 24K* ROM | 4K ROM A/D | D/A 31*/47 DIGITAL I/O	Z8 | DSP 24K ROM* | 6K ROM RAM PORT | CODEC INTF. RAM REFRESH | PWM 27*/43 DIGITAL I/O	Z8 | DSP 32K ROM | 6K ROM RAM PORT | CODEC INTF. RAM REFRESH | PWM 43 DIGITAL I/O	Z8 | DSP 24K ROM* | 8K ROM RAM PORT | CODEC INTF. RAM REFRESH | CODEC INTF. 27*/43 DIGITAL I/O	Z8 | DSP 32K ROM | 8K ROM RAM PORT | CODEC INTF. RAM REFRESH | CODEC INTF. 27*/43 DIGITAL I/O
Part #	**Z08600/Z08611**	**Z86C30/E30** **Z86C40/E40**	**Z89C65/C66**	**Z89C67/C68**	**Z89C69**	**Z89167/168**	**Z89169**
Description	Z8® NMOS (CCP™) 8600 = 2K ROM 8611 = 4K ROM	Z8® Consumer Controller Processor (CCP™) with 4K ROM C30 = 28-pin C40 = 40-pin E30/E40 = OTP version	Telephone Answering Controller with DSP LPC voice synthesis and DTMF detection, External RAM /RAM Interface (C66)	Telephone Answering Controller with digital voice encode and decode DTMF detection and full memory control interface. Ext. ROM/RAM Intfc. (C68)	Telephone Answering Controller with digital voice encode and decode DTMF detection and full memory control interface	Enhanced telephone answering controller with digital voice encode and decode DTMF detection and full memory controller intfc. ext. ROM/RAM intfc. (168)	Enhanced telephone answering controller with digital voice encode and decode DTMF detection and full memory controller interface
Process/Speed	NMOS 8,12 MHz	CMOS 12 MHz	CMOS 20 MHz	CMOS 20 MHz	CMOS 20 MHz	CMOS 24 MHz	CMOS 24 MHz
Features	2K/4K ROM 128 Bytes RAM 22/32 I/O lines On-chip oscillator 2 Counter/Timers 6 vectored, priority interrupts UART (Z8611)	4K ROM, 236 RAM 2 Standby Modes 2 Counter/Timers ROM Protect RAM Protect 4 Ports (86C40/E40) 3 Ports (86C30/E30) Brown-Out Protection 2 Analog Comparators Low EMI Watch-Dog Timer Auto Power-On Reset Low Power option	Z8® Controller 24K ROM (C65) 16-bit DSP 4K Word ROM 8-bit A/D with AGC DTMF macro available LPC macro available 10-bit PWM D/A Other DSP software options available 47 I/O Pins (C65) * = Note Z89C66 is ROMless (Z8) with 31 I/O pins.	Z8® Controller 24K ROM (C67) 16-bit DSP 6K Word ROM DTMF macro available LPC macro available 10-bit PWM D/A Other DSP S/W opt. avail. ARAM/DRAM/ROM Controller & Interface Dual Codec Interface 43 I/O (C67) * = Note Z89C68 is ROMless (Z8) with 27 I/O pins.	Z8® Controller 32K ROM 16-bit DSP 6K Word ROM DTMF macro available LPC macro available 10-bit PWM D/A Other DSP software options available ARAM/DRAM/ROM Controller & Interface Dual Codec Interface 43 I/O	Z8® Controller 24K ROM (167) 16-bit DSP 8K Word ROM DTMF Macro available LPC Macro available 10-bit PWM D/A Other DSP software options available ARAM/DRAM/ROM Dual Codec Interface 43 I/O (167) * = Note Z89168 is ROMless (Z8) with 27 I/O pins.	Z8® Controller 32K ROM 16-bit DSP 8K Word ROM DTMF Macro available LPC Macro available 10-bit PWM D/A Other DSP software options available ARAM/DRAM/ROM Dual Codec Interface 43 I/O
Package	28-pin DIP 40-pin DIP 44-pin PLCC	28-pin DIP 40-pin DIP 44-pin PLCC, QFP	68-pin PLCC	84-pin PLCC	84-pin PLCC	84-pin PLCC 80-pin QFP	84-pin PLCC 80-pin QFP
Application	Low cost tape board TAD	Window Control Wiper Control Sunroof Control Security Systems TAD	Fully featured cassette answering machines with voice prompts and DTMF signaling Digital OGM available	Voice Processing, DSP applications in tapeless TAD and other high-performance voice processors	Voice Processing, DSP applications in tapeless TAD and other high-performance voice processors	Voice Processing, DSP applications in tapeless TAD and other high-performance voice processors	Voice Processing, DSP applications in tapeless TAD and other high-performance voice processors

ZiLOG Video Products Superintegration™ Products Guide

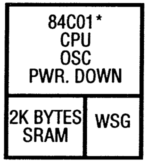
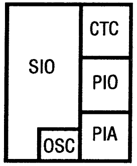
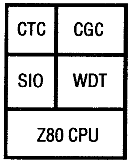
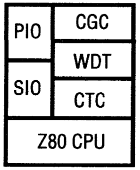
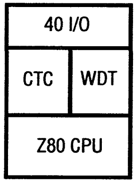
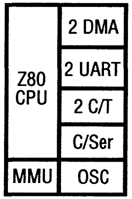
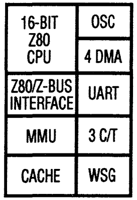
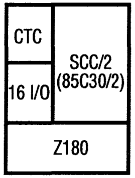
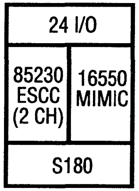
	TV Controller			IR Controller		Cable TV	
Block Diagram	8K ROM 4K CHAR ROM Z8 CPU | RAM OSD 13 PWM | TIMER WDT | 5 PORTS	6K ROM 3K CHAR ROM Z8 CPU | RAM OSD 7 PWM | TIMER WDT | 3 PORTS	CHAR ROM COMMAND INTERPRETER ANALOG SYNC/DATA SLICER | OSD CTRL	1K/6K ROM Z8 CPU WDT | 124 RAM P2 | P3	2K/8K/16K ROM Z8 CPU WDT | 128,256, 768 RAM P0 | P1 | P2 | P3	4K ROM CPU WDT | 236 RAM | P1 P2 | P3 | P0	16K ROM | UART CPU | 236 RAM P0 | P1 | P2 P3 | P4 | P5 | P6
Part #	**Z86C27/127/97**	**Z86227**	**Z86128**	**Z86L06/L29**	**Z86L70/71/72** (Q193)	**Z86C40/E40**	**Z86C61/62**
Description	Z8® Digital Television Controller MCU with logic functions needed for Television Controller, VCRs and Cable	Standard DTC features with reduced ROM, RAM, PWM outputs for greater economy	Line 21 Controller (L21C™) for Closed Caption Television	18-pin Z8® Consumer Controller Processor (CCP™) low-voltage and low-current battery operation 1K-6K ROM	Z8® (CCP™) low-voltage parts that have more ROM, RAM and special Counter/Timers for automated output drive capabilities	Z8® Consumer Controller Processor (CCP™) with 4K ROM (C40) E40 = OTP version	Z8® MCU with Expanded I/O's and 16K ROM
Process/Speed	CMOS 4 MHz	CMOS 4 MHz	CMOS 12 MHz	Low Voltage CMOS 8 MHz	Low Voltage CMOS 8 MHz	CMOS 12 MHz	CMOS 16, 20 MHz
Features	Z8/DTC Architecture 8K ROM, 256-byte RAM 160x7-bit video RAM On-Screen Display (OSD) video controller Programmable color, size, position attributes 13 PWMs for D/A conversion 128-character set 4Kx6-bit char. Gen. ROM Watch-Dog Timer (WDT) Brown-Out Protection 5 Ports/36 pins 2 Standby Modes Low EMI Mode	Z8/DTC Architecture 6K ROM, 256-byte RAM 120x7-bit video RAM OSD on board Programmable color, size, position attributes 7 PWMs 96-character set 3Kx6-bit character generator ROM Watch-Dog Timer (WDT) Brown-Out Protection 3 Ports/20 pins 2 Standby Modes Low EMI Mode	Conforms to FCC Line 21 format Parallel or serial modes Stand-alone operation On-board data sync and slicer On-board character generator - Color - Blinking - Italic - Underline	Z8® Architecture 1K ROM & 6K ROM Watch-Dog Timer 2 Analog Comparators with output option 2 Standby Modes 2 Counter/Timers Auto Power-On Reset 2 volt operation RC OSC option Low Noise option Brown-Out Protection High current drivers (2, 4)	Z8® Architecture 2K/8K/16K ROM Watch-Dog Timer 2 Analog Comparators with output option 2 Standby Modes 2 Enhanced Counter/ Timers, Auto Pulse Reception/Generation Auto Power-On Reset 2 volt operation RC OSC option Brown-Out Protection High current drivers (4)	4K ROM, 236 RAM 2 Standby Modes 2 Counter/Timers ROM Protect RAM Protect 4 Ports Brown-Out Protection 2 Analog Comparators Low EMI Watch-Dog Timer Auto Power-On Reset Low Power option	16K ROM Full duplex UART 2 Standby Modes (STOP and HALT) 2 Counter/Timers ROM Protect option RAM Protect option Pin compatible to Z86C21 C61 = 4 Ports C62 = 7 Ports
Package	64-pin DIP 52-pin active (127)	40-pin DIP	18-pin DIP	18-pin DIP 18-pin SOIC	20-pin DIP (L71), 18-pin DIP, SOIC (L70) 40,44-pin DIP, PLCC, QFP (L72)	40-pin DIP	40-pin DIP (C61) 44-pin PLCC,QFP (C61) 68-pin PLCC (C62)
Application	Low-end Television Cable/Satellite Receiver	Low-end Television Cable/Satellite Receiver	TVs, VCRs, Decoders	I.R. Controller Portable battery operations	I.R. Controller Portable battery operations	Window Control Wiper Control Sunroof Control Security Systems TAD	Cable Television Remote Control Security

ZiLOG Datacommunications Superintegration™ Products Guide

Block Diagram	SCC	ESCC	SCC DMA DMA DMA DMA BIU	PIO CGC WDT SIO CTC Z80 CPU	CTC SCC/2 (85C30/2) 16 I/O Z180	24 I/O 85230 ESCC (2 CH) 16550 MIMIC S180	USC	USC/2 TSA	USC/2 DMA DMA
Part #	**Z8030/Z80C30 Z8530/Z85C30**	**Z85230/Z80230 Z85233***	**Z16C35**	**Z84C15**	**Z80181**	**Z80182**	**Z16C30**	**Z16C33**	**Z16C32**
Description	Serial Com. Controller	Enhanced Serial Com. Controller	Integrated Serial Com. Controller	Intelligent Peripheral Controller	Smart Access Controller	Zilog Intelligent Peripheral	Universal Serial Controller	Mono-channel Universal Serial Controller	Integrated Universal Serial Controller
Process/ Speed/ Clock Data Rate	NMOS: 4, 6, 8 MHZ CMOS: 8,10 16 MHz 2, 2.5, 4 Mb/s	CMOS: 10, 16 20 MHz 2.5, 4.0, 5.0 Mb/s	CMOS: 10, 16 MHz 2.5, 4.0 Mb/s	CMOS 6, 10,16 MHz	10, 12.5	CMOS 16, 20 MHz	CMOS: 20 MHz CPU Bus 10 Mb/s 20 Mb/s	CMOS: 10 MHz CPU Bus 10 Mb/s	CMOS:20 MHz CPU Bus 16 Mb/s 20 Mb/s
Features	Two independent full-duplex channels Enhanced DMA support: 10x19 status FIFO 14-bit byte counter NRZ/NRZI/FM	Full dual-channel SCC plus deeper FIFOs: 4 bytes on Tx 8 bytes on Rx DPLL counter per channel Software compatible to SCC *One channel of Z85230	Full dual-channel SCC plus 4 DMA controllers and a bus interface unit	Z80® CPU, SIO, CTC WDT, CGC The Z80 Family in one device Power-On Reset Two chip selects 32-bit CRC WSG EV mode[1] 3 and 5 Volt Version	Complete Z180™ plus SCC/2 CTC 16 I/O lines Emulation Mode[1]	Complete Static version of Z180 plus ESCC (2 channels of 85230) 16550 MIMIC 24 Parallel I/O Emulation Mode[1]	Two dual-channel 32-byte receive & transmit FIFOs 16-bit bus B/W: 18.2 Mb/s 2 BRGs per channel Flexible 8/16-bit bus interface	Single-channel (half of USC™) plus Time Slot Assigner functions for ISDN	Single-channel (half of USC) plus two DMA controllers Array chained and linked-list modes with ring buffer support
Package	40-pin DIP 44-pin CERDIP 44-pin PLCC	40-pin DIP 44-pin PLCC *44-pin QFP (85233)	68-pin PLCC	100-pin QFP 100-pin VQFP	100-pin QFP	100-pin QFP 100-pin VQFP	68-pin PLCC	68-pin PLCC	68-pin PLCC
Application	General-Purpose datacom.	General-Purpose datacom. High performance SCC software	High performance datacom. SCC upgrades	Intelligent peripheral controllers Modems	Intelligent peripheral controllers Printers, Faxes, Modems, Terminals	General-Purpose Embedded Control Modem, Fax, Data Communications	General-Purpose high-end datacom. Ethernet HDLC X.25 Frame Relay	General-Purpose high-end datacom. Ethernet HDLC X.25 Frame Relay	General-Purpose high-end datacom. Ethernet HDLC X.25 Frame Relay

AppleTalk® A Registered Trademark of Apple Computer, Inc.

Z80® Embedded Controllers Superintegration™ Products Guide

Block Diagram	 84C01* CPU OSC PWR. DOWN 2K BYTES SRAM WSG	 SIO CTC PIO OSC PIA	 CTC CGC SIO WDT Z80 CPU	 PIO CGC WDT SIO CTC Z80 CPU	 40 I/O CTC WDT Z80 CPU	 Z80 CPU 2 DMA 2 UART 2 C/T C/Ser MMU OSC	 16-BIT Z80 CPU OSC 4 DMA Z80/Z-BUS INTERFACE UART MMU 3 C/T CACHE WSG	 CTC 16 I/O SCC/2 (85C30/2) Z180	 24 I/O 85230 ESCC (2 CH) 16550 MIMIC S180
Part #	**Z84C50**	**Z84C90**	**Z84013/C13**	**Z84015/C15**	**Z84011/C11**	**Z80180/S180**	**Z80280**	**Z80181**	**Z80182**
Description	Z80/84C01 with 2K SRAM	Killer I/O (3 Z80 peripherals)	Intelligent Peripheral Controller	Intelligent Peripheral Controller	Parallel I/O Controller	High-performance Z80® CPU with peripherals	16-bit Z80® code compatible CPU with peripherals	Smart Access Controller	Zilog Intelligent Peripheral
Speed MHz	10	8, 10, 12.5	6, 10	6, 10, 16	6, 10	6, 8, 10, 16*, 20* *Z8S180 only	10, 12	10, 12.5	16, 20
Features	Z80® CPU 2 Kbytes SRAM WSG Oscillator Pin compatible with Z84C00 DIP & PLCC EV mode[1] *84C01 is available as a separate part	SIO, PIO, CTC plus 8 I/O lines	Z80® CPU, SIO, CTC WDT, CGC, WSG, Power-On Reset 2 chip selects EV mode[1]	Z80® CPU, SIO, CTC WDT, CGC The Z80 Family in one device Power-On Reset Two chip selects 32-bit CRC WSG EV mode[1]	Z80® CPU, CTC, WDT 40 I/O lines bit programmable Power-On Reset EV mode[1]	Enhanced Z80 CPU MMU 1 Mbyte 2 DMAs 2 UARTs with BRGs C/Serial I/O Port Oscillator Z8S180 includes; Pwr dwn, Prgmble EMI, divide-by-one clock option	16-bit code compatible Z80® CPU Three stage pipeline MMU 16 Mbyte CACHE 256 byte Inst. & Data Peripherals 4 DMAs, UART, 3 16-bit C/T, WSG Z80/Z-BUS® interface	Complete Z180 plus SCC/2 CTC 16 I/O lines Emulation Mode[1]	Complete Static Version of Z180™ plus ESCC (2 channels of Z85230) 16550 MIMIC 24 Parallel I/O Emulation Modes[1]
Package	40-pin DIP 44-pin PLCC 44-pin QFP	84-pin PLCC	84-pin PLCC	100-pin QFP 100-pin VQFP	100-pin QFP	64-pin DIP 68-pin PLCC 80-pin QFP	68-pin PLCC	100-pin QFP	100-pin QFP 100-pin VQFP
Application	Embedded Controllers	General-purpose peripheral that can be used with Z80 and other CPU's	Intelligent datacom controllers	Intelligent peripheral controllers Modems	Intelligent parallel-I/O controllers Industrial display terminals	Embedded Control	Embedded Control Terminals Printers	Intelligent peripheral controllers Printers, Faxes, Modems, Terminals	General-Purpose Embedded Control Modem, Fax, Data Communications

[1] Allows use of existing development systems.

Zilog Peripherals Superintegration™ Products Guide

	Z8036 Z8536	Z32H00	Z5380 Z53C80	Z85C80
Description	Counter/Timer & parallel I/O Unit (CIO)	Hyperstone Enhanced Fast Instruction Set Computer (EFISC) Embedded (RISC) Processor	Small Computer System Interface (SCSI)	Serial Communication Controller and Small Computer System Interface
Process/ Speed	NMOS 4,6 MHz	CMOS 25 MHz	CMOS Z5380: 1.5 MB/s Z53C80: 3.0 MB/s	CMOS SCC - 10, 16 MHz SCSI - 3.0 MB/s
Features	Three 16-bit Counter/Timers, Three I/O ports with bit catching, pattern matching interrupts and handshake I/O	32-bit MPU 4 Gbytes address space 19 global and 64 local registers of 32 bits each 128 bytes instruction cache 1.2μ CMOS 42 mm^2 die	ANSI X3.131-1986 Direct SCSI bus interface On-board 48 mA drivers Normal or Block mode DMA transfers Bus interface, target and initiator	Full dual-channel SCC plus SCSI sharing databus and read/write functions
Package	40-pin PDIP 44-pin PLCC	144-pin PGA 132-pin QFP	Z5380: 40-pin DIP 44-pin PLCC Z53C80: 48-pin DIP 44-pin PLCC	68-pin PLCC
Application	General-Purpose Counter/Timers and I/O system designs	Embedded high-performance industrial controller Workstations	Bus host adapters, formatters, host ports	AppleTalk® networking SCSI disk drives

[2] Software and hardware compatible with discrete devices.

ZiLOG

ZILOG LITERATURE GUIDE

Z8®/SUPER8™ MICROCONTROLLER FAMILY

Databooks	Part No	Unit Cost
Z8 Microcontrollers Databook (includes the following documents)	**DC-8275-04**	**5.00**

Z8 CMOS Microcontrollers

Z86C00/C10/C20 MCU OTP Product Specification
Z86C06 Z8 CCP™ Preliminary Product Specification
Z86C08 8-Bit MCU Product Specification
Z86E08 Z8 OTP MCU Product Specification
Z86C09/19 Z8 CCP Product Specification
Z86E19 Z8 OTP MCU Advance Information Specification
Z86C11 Z8 MCU Product Specification
Z86C12 Z8 ICE Product Specification
Z86C21 Z8 MCU Product Specification
Z86E21/Z86E22 OTP Product Specification
Z86C30 Z8 CCP Product Specification
Z86E30 Z8 OTP CCP Product Specification
Z86C40 Z8 CCP Product Specification
Z86E40 Z8 OTP CCP Product Specification
Z86C27/97 Z8 DTC™ Product Specification
Z86127 Low-Cost Digital Television Controller Adv. Info. Spec.
Z86C50 Z8 CCP ICE Advance Information Specification
Z86C61 Z8 MCU Advance Information Specification
Z86C62 Z8 MCU Advance Information Specification
Z86C89/C90 CMOS Z8 CCP Product Specification
Z86C91 Z8 ROMless MCU Product Specification
Z86C93 Z8 ROMless MCU Preliminary Product Specification
Z86C94 Z8 ROMless MCU Product Specification
Z86C96 Z8 ROMless MCU Advance Information Specification
Z88C00 CMOS Super8 MCU Advance Information Specification

Z8 NMOS Microcontrollers

Z8600 Z8 MCU Product Specification
Z8601/03/11/13 Z8 MCU Product Specification
Z8602 8-Bit Keyboard Controller Preliminary Product Spec.
Z8604 8-Bit MCU Product Specification
Z8612 Z8 ICE Product Specification
Z8671 Z8 MCU With BASIC/Debug Interpreter Product Spec.
Z8681/82 Z8 MCU ROMless Product Specification
Z8691 Z8 MCU ROMless Product Specification
Z8800/01/20/22 Super8 ROMless/ROM Product Specification

Peripheral Products

Z86128 Closed-Captioned Controller Adv. Info. Specification
Z765A Floppy Disk Controller Product Specification
Z5380 SCSI Product Specification
Z53C80 SCSI Advance Information Specification

Z8 Application Notes and Technical Articles

Zilog Family On-Chip Oscillator Design
Z86E21 Z8 Low Cost Thermal Printer
Z8 Applications for I/O Port Expansions
Z86C09/19 Low Cost Z8 MCU Emulator
Z8602 Controls A 101/102 PC/Keyboard
The Z8 MCU Dual Analog Comparator
The Z8 MCU In Telephone Answering Systems
Z8 Subroutine Library
A Comparison of MCU Units
Z86xx Interrupt Request Registers
Z8 Family Framing
A Programmer's Guide to the Z8 MCU
Memory Space and Register Organization

Super8 Application Notes and Technical Articles

Getting Started with the Zilog Super8
Polled Async Serial Operations with the Super8
Using the Super8 Interrupt Driven Communications
Using the Super8 Serial Port with DMA
Generating Sine Waves with Super8
Generating DTMF Tones with Super8
A Simple Serial Parallel Converter Using the Super8

Additional Information

Z8 Support Products
Zilog Quality and Reliability Report
Literature List
Package Information
Ordering Information

Z8®/SUPER8™ MICROCONTROLLER FAMILY (Continued)

Databooks By Market Niche	Part No	Unit Cost
Digital Signal Processor Databook (includes the following documents) Z86C95 Z8® Digital Signal Processor Preliminary Product Specification Z89C00 16-Bit Digital Signal Processor Preliminary Product Specification Z89C00 DSP Application Note "Understanding Q15 Two's Complement Fractional Multiplication" Z89120, Z89920 (ROMless) 16-Bit Mixed Signal Processor Preliminary Product Specification Z89121, Z89921 (ROMless) 16-Bit Mixed Signal Processor Preliminary Product Specification Z89320 16-Bit Digital Signal Processor Preliminary Product Specification Z89321 16-Bit Digital Signal Processor Advance Information Specification	DC-8299-02	3.00
Telephone Answering Device Databook (includes the following documents) Z89C65, Z89C66 (ROMless) Dual Processor T.A.M. Controller Preliminary Product Specification Z89C67, Z89C68/C69 (ROMless) Dual Processor Tapeless T.A.M. Controller Preliminary Product Specification Z89C65 Software Development Guide Z89C67/C69 Software Development Guide	DC-8300-02	3.00
Infrared Remote (IR) Control Databook (includes the following documents) Z86L06 Low Voltage CMOS Consumer Controller Processor Preliminary Product Specification Z86L29 6K Infrared (IR) Remote (ZIRC™) Controller Advance Information Specification Z86L70/L71/L72, Z86E72 Zilog IR (ZIRC™) CCP™ Controller Family Preliminary Product Specification	DC-8301-03	3.00
Z8 Microcontrollers (includes the following documents) Z86C07 CMOS Z8 8-Bit Microcontroller Product Specification Z86C08 CMOS Z8 8-Bit Microcontroller Product Specification Z86E08 CMOS Z8 8-Bit OTP Microcontroller Product Specification Z86C11 CMOS Z8 Microcontroller Product Specification Z86C12 CMOS Z8 In-Circuit Microcontroller Emulator Product Specification Z86C21 8K ROM Z8 CMOS Microcontroller Product Specification Z86E21 CMOS Z8 8K OTP Microcontroller Product Specification Z86C61/62/96 CMOS Z8 Microcontroller Product Specification Z86C63/64 32K ROM Z8 CMOS Microcontroller Product Specification Z86C91 CMOS Z8 ROMless Microcontroller Product Specification Z86C93 CMOS Z8 Multiply/Divide Microcontroller Product Specification	DC-8305-01	3.00
Z8 Microcontrollers (includes the following documents) Z86C04 CMOS Z8 8-Bit Low Cost 1K ROM Microcontroller Product Specification Z86E04 CMOS Z8 OTP 8-Bit Low Cost Microcontroller Product Specification	DC-6018-01	3.00
Mass Storage (includes the following documents) Z86C21 8K ROM Z8 CMOS Microcontroller Product Specification Z86E21 CMOS Z8 8K OTP Microcontroller Product Specification Z86C91 CMOS Z8 ROMless Microcontroller Product Specification Z86C93 CMOS Z8 Multiply/Divide Microcontroller Product Specification Z86C95 Z8 Digital Signal Processor Product Specification Z89C00 16-Bit Digital Signal Processor Product Specification Z89C00 DSP Application Note - "Understanding Q15 Two's Complement Fractional Multiplication"	DC-8303-00	3.00

Z8®/SUPER8™ MICROCONTROLLER FAMILY (Continued)

Databooks By Market Niche	Part No	Unit Cost
Digital Television Controllers (includes the following documents)	DC-8308-00	3.00
Z86C27/97 CMOS Z8® Digital Signal Processor Product Specification		
Z86C61/62/96 CMOS Z8 Microcontroller Product Specification		
Z86C63/64 32K ROM CMOS Z8 Microcontroller Product Specification		
Z86127 Low Cost Digital Television Controller Product Specification		
Z86128 Line 21 Closed-Caption Controller (L21C™) Digital Television Controller Product Specification		
Z86227 40-Pin Low Cost (4LDTC™) Digital Television Controller Product Specification		
Keyboard/Mouse/Pointing Devices Databook (includes the following documents)	DC-8304-00	3.00
Z8602 NMOS Z8® 8-Bit Keyboard Controller Product Specification		
Z8614 NMOS Z8® 8-Bit Keyboard Controller Product Specification		
Z8615 NMOS Z8® 8-Bit Keyboard Controller Product Specification		
Z86E23 Z8® 8-Bit Keyboard Controller with 8K OTP Product Specification		
Z86C04 CMOS Z8® 8-Bit Microcontroller Product Specification		
Z86C08 CMOS Z8® 8-Bit Microcontroller Product Specification		
Z88C17 CMOS Z8® 8-Bit Microcontroller Product Specification		
PC Audio Databook (includes the following documents)	DC-8317-00	3.00
Z86321 Digital Audio Processor Preliminary Product Specification		
Z89320 16-Bit Digital Signal Processor Preliminary Product Specification		
Z89321/371 16-Bit Digital Signal Processor Preliminary Product Specification		
Z89331 16-Bit PC ISA Bus Interface Advance Information Specification		
Z89341/42/43 Wave Synthesis Chip Set Advance Information Specification		
Z5380 Small Computer System Interface Product Specification		
PCMCIA/SCSI Interface Controllers (includes the following documents)	DC 8313-00	3.00
Z5380 Small Computer System Interface Product Specification		
Z53C80 Small Computer System Interface Product Specification		
Z85C80 SCSCI™ Serial Communications and Small Computer Interface Product Specification		
Z86017 PCMCIA Interface Preliminary Product Specification		
Z86015 PCMCIA Interface with DMA Support Advance Product Specification		
Z86020 CardBus/PCI Interface Advance Product Specification		
Z8® Low End MCU Databook (includes the following documents)	DC-8318-00	3.00
Z86C04		

L

Z8®/SUPER8™ MICROCONTROLLER FAMILY (Continued)

Z8 Product Specifications, Technical Manuals and Users Guides	Part No	Unit Cost
Z86E23 CMOS Z8 OTP Microcontroller Preliminary Product Specification	DC-2598-00	N/C
Z8614 NMOS Z8 8-Bit MCU Keyboard Controller Preliminary Product Specification	DC-2576-00	N/C
Z8 OTP CMOS One-Time-Programmable Microcontrollers Addendum	DC-2614-AA	N/C
Z8 Microcontrollers Technical Manual	DC-8291-02	5.00
Z86018 Preliminary User's Manual	DC-8296-00	N/C
Digital TV Controller User's Manual	DC-8284-01	3.00
Z89C00 16-Bit Digital Signal Processor User's Manual/DSP Software Manual	DC-8294-02	3.00
PLC Z89C00 Cross Development Tools Brochure	DC-5538-01	N/C
Z86C95 16-Bit Digital Signal Processor User Manual	DC-8595-00	3.00
Z86017 PCMCIA Adaptor Chip User's Manual	DC-8298-01	3.00
Z89C65/C67/C69 Software Manual	DC-8310-00	3.00

Z8 Application Notes	Part No	Unit Cost
Z8602 Controls A 101/102 PC/Keyboard	DC-2601-01	N/C
The Z8 MCU Dual Analog Comparator	DC-2516-01	N/C
Z8 Applications for I/O Port Expansions	DC-2539-01	N/C
Z86E21 Z8 Low Cost Thermal Printer	DC-2541-01	N/C
Zilog Family On-Chip Oscillator Design	DC-2496-01	N/C
Using the Zilog Z86C06 SPI Bus	DC-2584-01	N/C
Interfacing LCDs to the Z8	DC-2592-01	N/C
X-10 Compatible Infrared (IR) Remote Control	DC-2591-01	N/C
Z86C17 In-Mouse Applications	DC-3001-01	N/C
Z86C40/E40 MCU Applications Evaluation Board	DC-2604-01	N/C
Z86C08/C17 Controls A Scrolling LED Message Display	DC-2605-01	N/C
Z86C95 Hard Disk Controller Flash EPROM Interface	DC-2639-01	N/C
Timekeeping with Z8; DTMF Tone Generation; Serial Communication Using the CCP Software UART	DC-2645-01	N/C

Z80®/Z8000® CLASSIC FAMILY OF PRODUCTS

Databooks By Market Niche	Part No	Unit Cost
High-Speed Serial Communication Controllers	DC-8314-00	3.00
Z16C30 CMOS Universal Serial Controller (USC™) Preliminary Product Specification		
Z16C32 Integrated Universal Serial Controller (IUSC™) Preliminary Product Specification		
Application Notes and Support Products		
Zilog's Superintegration™ Products Guide		
Literature Guide and Third Party Support		
Serial Communication Controllers	DC-8316-00	3.00
Z8030/Z8530 Z-Bus® SCC Serial Communication Controller Product Specification		
Z80C30/Z85C30 CMOS Z-Bus® SCC Serial Communication Controller Product Specification		
Z80230 Z-Bus® ESCC™ Enhanced Serial Communication Controller Preliminary Product Specification		
Z85230 ESCC™ Enhanced Serial Communication Controller Product Specification		
Z85233 EMSCC™ Enhanced Mono Serial Communication Controller Product Specification		
Z85C80 SCSCI™ Serial Communications and Small Computer Interface Product Specification		
Z16C35/Z85C35 CMOS ISCC™ Integrated Serial Communications Controller Product Specification		
Application Notes and Support Products		
Zilog's Superintegration™ Products Guide		
Literature Guide and Third Party Support		

Z80/Z180/Z280 Product Specifications, Technical Manuals and Users Guides	Part No	Unit Cost
Z80 Family Technical Manual	DC-8306-00	3.00
Z80180 Z180 MPU Microprocessor Unit Technical Manual	DC-8276-04	3.00
Z280 MPU Microprocessor Unit Technical Manual	DC-8224-03	3.00
Z80180/Z8S180 Z180 Microprocessor Product Specification	DC-2609-03	N/C
Z80182 Zilog Intelligent Peripheral (ZIP™)	DC-2616-03	N/C
Z380™ Preliminary Product Specification	DC-6003-03	N/C
Z380™User's Manual	DC-8297-00	3.00
Z80 Family Programmer's Reference Guide	DC-0012-04	N/C

Z80/Z180/Z280 Application Notes	Part No	Unit Cost
Z180/SCC™ Serial Communications Controller Interface at 10 MHz	DC-2521-02	N/C
Z80 Using the 84C11/C13/C15 in place of the 84011/013/015	DC-2499-02	N/C
A Fast Z80 Embedded Controller	DC-2578-01	N/C

Z80®/Z8000® CLASSIC FAMILY OF PRODUCTS (Continued)

Z8000 Product Specifications, Technical Manuals and Users Guides	Part No	Unit Cost
Z8000 CPU Central Processing Unit Technical Manual	DC-2010-06	3.00
SCC Serial Communication Controller User's Manual	DC-8293-02	3.00
Z8036 Z-CIO/Z8536 CIO Counter/Timer and Parallel Input/Output Technical Manual	DC-2091-02	3.00
Z8038 Z8000 Z-FIO FIFO Input/Output Interface Technical Manual	DC-2051-01	3.00
Z8000 CPU Central Processing Unit Programmer's Pocket Guide	DC-0122-03	3.00
Z85233 EMSCC Enhanced Mono Serial Communication Controller Preliminary Product Specification	DC-2590-00	N/C
Z85C80 SCSCI™ Serial Communication and Small Computer Interface Preliminary Product Specification	DC-2534-02	N/C
Z16C30 USC Universal Serial Controller Preliminary Technical Manual	DC-8280-02	3.00
Z16C32 IUSC Integrated Universal Serial Controller Technical Manual	DC-8292-03	3.00
Z16C35 ISCC Integrated Serial Communication Controller Technical Manual	DC-8286-01	3.00
Z16C35 ISCC Integrated Serial Communication Controller Addendum	DC-8286-01A	N/C
Z53C80 Small Computer System Interface (SCSI) Product Specification	DC-2575-01	N/C
Z80230 Z-BUS® ESCC Enhanced Serial Communication Controller Preliminary Product Specification	DC-2603-01	N/C

Z8000 Application Notes	Part No	Unit Cost
Z16C30 Using the USC in Military Applications	DC-2536-01	N/C
Datacom IUSC/MUSC Time Slot Assigner	DC-2497-02	N/C
Datacom Evaluation Board Using The Zilog Family With The 80186 CPU	DC-2560-03	N/C
Boost Your System Performance Using the Zilog ESCC Controller	DC-2555-02	N/C
Z16C30 USC - Design a Serial Board for Multiple Protocols	DC-2554-01	N/C
Using a SCSI Port for Generalized I/O	DC-2608-01	N/C

MILITARY COMPONENTS FAMILY

Military Specifications	Part No	Unit Cost
Z8681 ROMless Microcomputer Military Product Specification	DC-2392-02	N/C
Z8001/8002 Military Z8000 CPU Central Processing Unit Military Product Specification	DC-2342-03	N/C
Z8581 Military CGC Clock Generator and Controller Military Product Specification	DC-2346-01	N/C
Z8030 Military Z8000 Z-SCC Serial Communications Controller Military Product Specification	DC-2388-02	N/C
Z8530 Military SCC Serial Communications Controller Military Product Specification	DC-2397-02	N/C
Z8036 Military Z8000 Z-CIO Counter/Timer Controller and Parallel I/O Military Electrical Specification	DC-2389-01	N/C
Z8038/8538 Military FIO FIFO Input/Output Interface Unit Military Product Specification	DC-2463-02	N/C
Z8536 Military CIO Counter/Timer Controller and Parallel I/O Military Electrical Specification	DC-2396-01	N/C
Z8400 Military Z80 CPU Central Processing Unit Military Electrical Specification	DC-2351-02	N/C
Z8420 Military PIO Parallel Input/Output Controller Military Product Specification	DC-2384-02	N/C
Z8430 Military CTC Counter/Timer Circuit Military Electrical Specification	DC-2385-01	N/C
Z8440/1/2/4 Z80 SIO Serial Input/Output Controller Military Product Specification	DC-2386-02	N/C
Z80C30/85C30 Military CMOS SCC Serial Communications Controller Military Product Specification	DC-2478-02	N/C
Z84C00 CMOS Z80 CPU Central Processing Unit Military Product Specification	DC-2441-02	N/C
Z84C20 CMOS Z80 PIO Parallel Input/Output Military Product Specification	DC-2384-02	N/C
Z84C30 CMOS Z80 CTC Counter/Timer Circuit Military Product Specification	DC-2481-01	N/C
Z84C40/1/2/4 CMOS Z80 SIO Serial Input/Output Military Product Specification	DC-2482-01	N/C
Z16C30 CMOS USC Universal Serial Controller Military Preliminary Product Specification	DC-2531-01	N/C
Z80180 Z180 MPU Microprocessor Unit Military Product Specification	DC-2538-01	N/C
Z84C90 CMOS KIO Serial/Parallel/Counter Timer Preliminary Military Product Specification	DC-2502-00	N/C
Z85230 ESCC Enhanced Serial Communication Controller Military Product Specification	DC-2595-00	N/C

GENERAL LITERATURE

Catalogs, Handbooks, Product Flyers and Users Guides	Part No	Unit Cost
Superintegration Shortform Catalog 1994	DC-5472-12	N/C
Superintegration Products Guide	DC-5499-07	N/C
ZIA™3.3-5.5V Matched Chip Set for AT Hard Disk Drives Datasheet	DC-5556-01	N/C
ZIA ZIA00ZCO Disk Drive Development Kit Datasheet	DC-5593-01	N/C
Zilog Hard Disk Controllers - Z86C93/C95 Datasheet	DC-5560-01	N/C
Zilog Infrared (IR) Controllers - ZIRC™ Datasheet	DC-5558-01	N/C
Zilog Intelligent Peripheral Controller - ZIP™Z80182 Datasheet	DC-5525-01	N/C
Zilog Digital Signal Processing - Z89320 Datasheet	DC-5547-01	N/C
Zilog Keyboard Controllers Datasheet	DC-5600-01	N/C
Z380™ - Next Generation Z80®/Z180™ Datasheet	DC-5580-02	N/C
Fault Tolerant Z8® Microcontroller Datasheet	DC-5603-01	N/C
32K ROM Z8® Microcontrollers Datasheet	DC-5601-01	N/C
Zilog Datacommunications Brochure	DC-5519-00	N/C
Zilog Digital Signal Processing Brochure	DC-5536-02	N/C
Zilog PCMCIA Adaptor Chip Z86017 Datasheet	DC-5585-01	N/C
Zilog Television/Video Controllers Datasheet	DC-5567-01	N/C
Zilog TAD Controllers - Z89C65/C67/C69 Datasheet	DC-5561-01	N/C
Zilog ASSPs - Partnering With You Product Flyer	DC-5553-01	N/C
Quality and Reliability Report	DC-2475-11	N/C
The Handling and Storage of Surface Mount Devices User's Guide	DC-5500-02	N/C
Universal Object File Utilities User's Guide	DC-8236-04	3.00
Zilog 1991 Annual Report	DC-1991-AR	N/C
Zilog 1992 Annual Report	DC-1992-AR	N/C
Zilog 1993 First Quarter Financial Report	DC-1993-Q1	N/C
Zilog 1993 Second Quarter Financial Report	DC-1993-Q2	N/C
Microcontroller Quick Reference Folder	DC-5508-01	N/C

Z80® & Z80180 HARDWARE AND SOFTWARE SUPPORT

Hardware Support

Company	Product	Phone
American Automation	Emulator	(714) 731-1661
Applied Microsystems	Emulator (symbolic debug)	(206) 882-2000
Hewlett-Packard	Emulator pods for HP 64000/UX/PC	(800) 4HP-DATA
Huntsville Microsystems	Emulator	(205) 881-6005
iSystems (Germany)	Emulator	08131-25083
Micromint	SB180,SB180FX, BCC180,RTC180	(800) 635-3355
MicroTek	Z80 Emulator, RS-232 compatible	(213) 321-2121
MicroWorks	Prototyping board	(408) 997-1644
Orion Instruments	Logic Analyzer and ROM emulator, plus Z182 emulator	(415) 327-8800
Pentica Systems, Inc.	Emulator	(617) 577-1101
Softaid	Emulator, ICEBOX, ICE Analyzer (symbolic debug)	(800) 433-8812
Sophia Systems	Emulator, SA2000	(415) 493-6700
Versalogic	Z80 STD Bus circuit board	(503) 485-8575
Z-World	IBM PC Development Bd.	(916) 753-3722
Zaxtek	Emulator	(714) 474-1170
Zilog	Z180+SCC (Z8018000ZCO) Application Board	*Call Zilog*
Zilog	Z80181 Eval. Kit (Z8018100ZCO)	*Call Zilog*
Zilog	Z84C11 Eval. Kit (Z84C1100ZCO)	*Call Zilog*
Zilog	Z84C15 Eval. Kit (Z84C1500ZCO)	*Call Zilog*
Zilog	Z84C50+KIO Application Bd. (Z84C5000ZCO)	*Call Zilog*
Zilog	Z84C01+KIO Development Bd. (Z84C9000ZCO)	*Call Zilog*
Emulation Technology	Z182 Emulator	(408) 982-0660

68 PLCC Socket Manufacturers:
Methode, TNB, ITT, Cannon, Precicontact

64 Shrink DIP Socket Manufacturers:
TI, Bevar, Yamaichi

44/80/100 Pin QFP:
ZIF (Zero Insertion Force) sockets for prototyping may be obtained from Yamaichi Electronics, (408) 450-0797.

100-Pin QFP Clip:
Emulation Technology, 408-982-0660

Assemblers and Cross Assemblers

Company	Host/Comments	Phone
2500AD	IBM PC, CP/M, VAX	(800) 843-8144
Allen Ashley	IBM PC	(818) 793-5748
American Automation	IBM PC	(714) 731-1661
AnyWare Engineering	IBM PC, Macintosh	(303) 442-0556
Avocet Systems	IBM PC	(800) 448-8500
Enertec, Inc.	IBM PC, VAX	(215) 362-0966
Hewlett-Packard	HP 64000/64000UX/ 64000-PC Systems	(800) 4HP-DATA
Huntsville Microsystems	IBM PC, Sun Micro, Apollo	(205) 881-6005
Logisoft	IBM PC	(408) 773-8465
Micro Computer Tools	IBM PC	(415) 825-4200
Micro Dialects, Inc.	Macintosh	(513) 271-9100
Softaid	IBM PC	(800) 433-8812
Softools	IBM PC	(410) 750-3733
Software Development Systems	Uniware,IBM,VAX Unix/VMS,Sun,Apollo	(708) 971-8170
Z-World	IBM PC	(916) 753-3722

Simulators/Applications Software

Company	Host/Comments	Phone
Avocet Systems	Simulator/IBM PC	(800) 448-8500
Lear Com Company	Simulator/IBM PC	(303) 232-2226
Logisoft	8080 to Z80 Translator	(408) 773-8465
Micromint	Z-System OS	(800) 635-3355
Softaid	Z80180 Guide, IBM PC diskette	(800) 433-8812
The AG Group	LLAP Dvmnt/Apple	(510) 937-7900
Z-World	Simulator/IBM PC	(916) 753-3722

L

Z80, Z80180, Z80280, & Z80380 HARDWARE AND SOFTWARE SUPPORT

Z80 & Z80180 High Level Language Compilers

Company	Language Host	Phone
2500AD	C, IBM PC, CP/M, VAX	(800) 843-8144
American Automation	C, IBM PC	(714) 731-1661
Archimedes	C, IBM PC, Sun, VAX, HP	(415) 567-4010
Avocet Systems	C, IBM PC	(800) 448-8500
Laboratory	Forth IBM PC Microsystems	(213) 306-7412
Microtek Lab, Inc.	C, Pascal IBM PC,	(213) 321-2121
Microtek Research	Sun Micro, VAX	(408) 980-1300
MPE	Forth IBM PC	(716) 461-9187
Softaid	MT-Basic IBM PC	(800) 433-8812
Software Development Systems	C, IBM PC, VAX, Sun, Apollo	(708) 971-8170
Z-World	Dynamic C, IBM PC	(916) 753-3722

Note: Z80/64180 software is also compatible with the Z80180.

Z80280 Hardware Support

Company	System Name	Phone
Computer Design Solutions	STD Buscard & Z280 Dvmnt. Board	(704) 876-2346
Softaid	Z280 ICE Analyzer	(800) 433-8812

68 PLCC Socket Manufacturers:
Methods, TNB, ITT, Cannon, Precicontact

Assemblers and Cross Assemblers:

Company	Host/Comments	Phone
2500AD	IBM PC, CP/M, VAX	(800) 843-8144
Computer Design Solutions	IBM PC	(704) 876-2346

Simulators

Company	Host/Comments	Phone
Micro Methods, Inc.	IBM PC (ZRPM)	(503) 861-1765

High Level Language Compilers

Company	Language Host	Phone
2500AD	C, IBM PC, CP/M, VAX	(800) 843-8144
Computer Design Solutions	C, IBM PC	(704) 876-2346

Note: Z80 software is object code compatible with the Z280.

This is NOT a complete list of hardware and software vendors who support Zilog products. Please contact the Zilog sales office nearest you if what you are looking for is not on this list. This list is for reference only and is not an endorsement for any company.

Communications Software Support

SCC Physical Layer Drivers and Upper Layer Software

Company	Software	Phone
Trillium	SCC Physical Layer X.25, ISDN, Frame Relay Networking Layers.	(310) 479-0500
Probito's	LLAP Applications	(415) 941-2090
AT Barret Assoc.	Real Time Kernel	(713) 728-8688 (800) 525-4303

Z80380 Hardware Support

Company	System Name	Phone
Signum	In-Circuit Emulator	(805) 371-4608

Assemblers and Cross Assemblers:

Company	Host/Comments	Phone
PLC	IBM PC, Unix	(817) 599-8365
2500AD	IBM PC, CP/M, VAX	(800) 843-8144

ZiLOG

LITERATURE GUIDE

ORDERING INFORMATION

Complete the attached literature order form. Be sure to enclose the proper payment or supply a purchase order. Please reference specific order requirements.

MINIMUM ORDER REQUIREMENTS

Orders under $300.00 must be prepaid by check, money order or credit card. Canadian and foreign orders must be accompanied by a cashier's check in U.S. dollars, drawn on a correspondent U.S. bank only.

Orders over $300.00 may be submitted with a Purchase Order.

SHIPMENT

Orders will be shipped after your check is cashed or credit is checked via the most economical method. Please allow four weeks for delivery.

RETURNS ARE NOT ACCEPTED.

PLEASE PRINT OR TYPE

NAME

PHONE ( ) -

COMPANY

Method of Payment (Check One)

□ Check □ Money Order

Credit Card □ VISA □ M/C □ P.O. (over $300.00)

ADDRESS

CITY

STATE

ZIP

COUNTRY

PART NUMBER	DOCUMENT TITLE	UNIT COST	QTY.	TOTAL
		$		$
		$		$
		$		$
		$		$
		$		$
		$		$
		$		$
		$		$
		$		$
		$		$
		$		$
		$		$
		$		$
		$		$

Credit Card or Purchase Order # ____________ SUBTOTAL

Expiration Date ____________ ADD APPLICABLE SALES TAX (CA ONLY)

Signature ____________ ADD 10% SHIPPING AND HANDLING

TOTAL

Mail To:

ZiLOG
210 E. HACIENDA AVE. M/S C1-0
CAMPBELL, CA 95008-6600

Phone: (408)370-8016
Fax: (408)370-8056

Zilog Sales Offices Representatives and Distributors

Z

ZILOG DOMESTIC SALES OFFICES AND TECHNICAL CENTERS

CALIFORNIA	
Agoura	818-707-2160
Campbell	408-370-8120
Irvine	714-453-9701
San Diego	619-658-0391
COLORADO	
Boulder	303-494-2905
FLORIDA	
Clearwater	813-725-8400
GEORGIA	
Duluth	404-931-4022
ILLINOIS	
Schaumburg	708-517-8080
MINNESOTA	
Minneapolis	612-944-0737
NEW HAMPSHIRE	
Nashua	603-888-8590
OHIO	
Independence	216-447-1480
OREGON	
Portland	503-274-6250
PENNSYLVANIA	
Horsham	215-784-0805
TEXAS	
Austin	512-343-8976
Dallas	214-987-9987

INTERNATIONAL SALES OFFICES

CANADA	
Toronto	905-673-0634
CHINA	
Shenzhen	86-755-2236089
GERMANY	
Munich	49-8967-2045
Sömmerda	49-3634-23906
JAPAN	
Tokyo	81-3-3587-0528
HONG KONG	
Kowloon	852-7238979
KOREA	
Seoul	82-2-577-3272
SINGAPORE	
Singapore	65-2357155
TAIWAN	
Taipei	886-2-741-3125
UNITED KINGDOM	
Maidenhead	44-628-392-00

The information in this document is subject to change without notice. Devices sold by Zilog, Inc. are covered by warranty and patent indemnification provisions appearing in Zilog, Inc. Terms and Conditions of Sale only. Zilog, Inc. makes no warranty, express, statutory, implied or by description, regarding the information set forth herein or regarding the freedom of the described devices from intellectual property infringement. Zilog, Inc. makes no warranty of merchantability or fitness for any purpose. Zilog, Inc. shall not be responsible for any errors that may appear in this document. Zilog, Inc. makes no commitment to update or keep current the information contained in this document.

Zilog's products are not authorized for use as critical components in life support devices or systems unless a specific written agreement pertaining to such intended use is executed between the customer and Zilog prior to use. Life support devices or systems are those which are intended for surgical implantation into the body, or which sustains life whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.

Zilog, Inc. 210 East Hacienda Ave.
Campbell, CA 95008-6600
Telephone (408) 370-8000
Telex 910-338-7621
FAX 408 370-8056

SALES REPRESENTATIVES AND DISTRIBUTORS

U.S., CANADIAN & PUERTO RICAN REPRESENTATIVES

ALABAMA

Huntsville
Electronic Sales, Inc. .... (205) 533-1735

ARIZONA

Scottsdale
Thom Luke Sales, Inc. .... (602) 451-5400

COLORADO

Englewood
Thorson Rocky Mountain .... (303) 773-6300

CONNECTICUT

Wallingford
Advanced Technical Sales .... (508) 664-0888

FLORIDA

Altamonte Springs
Semtronic Associates, Inc. .... (407) 831-8233
Clearwater
Semtronic Associates, Inc. .... (813) 461-4675
Fort Lauderdale
Semtronic Associates, Inc. .... (305) 731-2484

GEORGIA

Norcross
Electronic Sales, Inc. .... (404) 448-6554

ILLINOIS

Hoffman Estates
Victory Sales, Inc. .... (708) 490-0300

IOWA

Cedar Rapids
Advanced Technical Sales .... (319) 393-8280

KANSAS

Olathe
Advanced Technical Sales .... (913) 782-8702

MARYLAND

Pasadena
Electronic Engineering & Sales .... (410) 255-9686

MASSACHUSETTS

North Reading
Advanced Technical Sales .... (508) 664-0888

MICHIGAN

Farmington Hills
Rathsburg Associates, Inc. .... (313) 489-1500

MINNESOTA

Minneapolis
Professional Sales for Industry .... (612) 944-8545

MISSOURI

Bridgeton
Advanced Technical Sales .... (314) 291-5003

NEW JERSEY

Cherry Hill
Tritek .... (609) 667-0200

NEW MEXICO

Albuquerque
Quatra & Associates .... (505) 296-6781

NEW YORK

Fairport
L-Mar Associates, Inc. .... (716) 425-9100

NORTH CAROLINA

Cary
Electronic Sales, Inc. .... (919) 467-8486

OHIO

Centerville
Q-Mark, Inc. .... (513) 438-1129
Independence
Rathburg Associates, Inc. .... (216) 447-8825

OKLAHOMA

Tulsa
Nova Marketing, Inc. .... (918) 660-5105

SALES REPRESENTATIVES AND DISTRIBUTORS

U.S., CANADIAN & PUERTO RICAN REPRESENTATIVES

TEXAS

Austin

Nova Marketing, Inc. .... (512) 343-2321

Dallas

Nova Marketing, Inc. .... (214) 265-4630

Houston

Nova Marketing, Inc. .... (713) 988-6082

UTAH

Salt Lake City

Thorson Rocky Mountain .... (801) 942-1683

WISCONSIN

Brookfield

Victory Sales, Inc. .... (414) 789-5770

CANADA

British Columbia

BBD Electronics, Inc. .... (604) 465-4907

Ontario

BBD Electronics, Inc. .... (905) 821-7800

Ottawa

BBD Electronics, Inc. .... (613) 764-1752

Quebec

BBD Electronics, Inc. .... (514) 697-0801

PUERTO RICO

San Juan

Semtronic Associates, Inc. .... (809) 766-0700

SALES REPRESENTATIVES AND DISTRIBUTORS

U.S. AND CANADIAN DISTRIBUTORS

NATIONWIDE

Newark Electronics	1-800-367-3573

ALABAMA

Inside Alabama	800-572-7236
Outside Alabama	800-633-2918

Huntsville

Arrow Electronics	(205) 837-6955
Hamilton Hallmark Electronics	(205) 837-8700

ARIZONA

Inside Arizona	800-352-8489
Outside Arizona	800-528-8471

Phoenix

Hamilton Hallmark Electronics	(602) 437-1200

Tempe

Anthem Electronics	(602) 966-6600
Arrow Electronics	(602) 431-0030

CALIFORNIA

Calabasas

Arrow Electronics	(818) 880-9686

Chatsworth

Anthem Electronics	(818) 700-1000

Costa Mesa

Hamilton Hallmark Electronics	(714) 641-4100

Hayward

Arrow Electronics	(510) 487-8416

Irvine

Anthem Electronics	(714) 768-4444
Arrow Electronics	(714) 587-0404

Rocklin

Anthem Electronics	(916) 624-9744
Hamilton Hallmark Electronics	(916) 961-0922

San Diego

Anthem Electronics	(619) 453-9005
Arrow Electronics	(619) 565-4800
Hamilton Hallmark Electronics	(619) 277-6136

San Jose

Anthem Electronics	(408) 453-1200
Arrow Electronics	(408) 441-9700

Sunnyvale

Hamilton Hallmark Electronics	(408) 743-3300

Woodland Hills

Hamilton Hallmark Electronics	(818) 594-0404

COLORADO

Colorado Springs

Hamilton Hallmark Electronics	(719) 637-0055

Englewood

Anthem Electronics	(303) 790-4500
Arrow Electronics	(303) 799-0258
Hamilton Hallmark Electronics	(303) 790-1662

CONNECTICUT

Cheshire

Hamilton Hallmark Electronics	(203) 271-2844

Wallingford

Arrow Electronics	(203) 265-7741

Waterbury

Anthem Electronics	(203) 596-3200

FLORIDA

Deerfield Beach

Arrow Electronics	(305) 429-8200

Lake Mary

Arrow Electronics	(407) 333-9300

Largo

Hamilton Hallmark Electronics	(813) 541-7440 (800) 282-9350

Fort Lauderdale

Hamilton Hallmark Electronics	(305) 484-5482

Winter Park

Hamilton Hallmark Electronics	(407) 657-3300

GEORGIA

Duluth

Arrow Electronics	(404) 497-1300
Hamilton Hallmark Electronics	(404) 623-5475 (404) 623-4400

ILLINOIS

Bensonville

Hamilton Hallmark Electronics	(708) 860-7780

Itasca

Arrow Electronics	(708) 250-0500

Schaumburg

Anthem Electronics	(708) 884-0200

INDIANA

Indianapolis

Arrow Electronics	(317) 299-2071
Hamilton Hallmark Electronics	(317) 872-8875 (800) 829-0146

IOWA

Cedar Rapids

Arrow Electronics	(319) 395-7230

KANSAS

Lenexa

Arrow Electronics	(913) 541-9542
Hamilton Hallmark Electronics	(913) 888-4747 (800) 332-4375

SALES REPRESENTATIVES AND DISTRIBUTORS

U.S. AND CANADIAN DISTRIBUTORS

KENTUCKY

Lexington	Hamilton Hallmark Electronics	(800) 235-6039 (800) 525-0069

MARYLAND

Columbia	Anthem Electronics	(301) 995-6640
	Arrow Electronics	(301) 596-7800
	Hamilton Hallmark Electronics	(410) 988-9800

MASSACHUSETTS

Peabody	Hamilton Hallmark Electronics	(508) 532-9808
Wilmington	Anthem Electronics	(508) 657-5170
	Arrow Electronics	(508) 658-0900

MICHIGAN

Livonia	Arrow Electronics	(313) 462-2290
Nori	Hamilton Hallmark Electronics	(313) 347-4271
Plymouth	Hamilton Hallmark Electronics	(313) 416-5800 (800) 767-9654

MINNESOTA

Bloomington	Hamilton Hallmark Electronics	(612) 881-2600
Eden Prairie	Anthem Electronics	(612) 944-5454
	Arrow Electronics	(612) 941-5280

MISSOURI

Earth City	Hamilton Hallmark Electronics	(314) 291-5350
St. Louis	Arrow Electronics	(314) 567-6888

NEVADA

Sparks	Arrow Electronics	(702) 331-5000

NEW JERSEY

Cherry Hill	Hamilton Hallmark Electronics	(609) 424-0110
Marlton	Arrow Electronics	(609) 596-8000
Pinebrook	Anthem Electronics	(201) 227-7960
	Arrow Electronics	(201) 227-7880
Parsippany	Hamilton Hallmark Electronics	(201) 515-1601

NEW YORK

Commack	Anthem Electronics	(516) 864-6600
Hauppauge	Arrow Electronics	(516) 231-1000
	Hamilton Hallmark Electronics	(516) 434-7470
Melville	Arrow Electronics	(516) 391-1300
Rochester	Arrow Electronics	(716) 427-0300
	Hamilton Hallmark Electronics	(716) 475-9130
Ronkonkoma	Hamilton Hallmark Electronics	(516) 737-0600

NORTH CAROLINA

Raleigh	Arrow Electronics	(919) 876-3132
	Hamilton Hallmark Electronics	(919) 872-0712

OHIO

Centerville	Arrow Electronics	(513) 435-5563
Dayton	Hamilton Hallmark Electronics	(513) 439-6735 (800) 423-4688
Solon	Arrow Electronics	(216) 248-3990
	Hamilton Hallmark Electronics	(216) 498-1100
Worthington	Hamilton Hallmark Electronics	(614) 888-3313

SALES REPRESENTATIVES AND DISTRIBUTORS

U.S. AND CANADIAN DISTRIBUTORS

OKLAHOMA

Tulsa

Arrow Electronics	(918) 252-7537
Hamilton Hallmark Electronics	(918) 254-6110

OREGON

Beaverton

ALMAC/Arrow Electronics	(503) 629-8090
Hamilton Hallmark Electronics	(503) 526-6200

PENNSYLVANIA

Horsham

Anthem Electronics	(215) 443-5150

Pittsburgh

Arrow Electronics	(412) 963-6807

TEXAS

Austin

Arrow Electronics	(512) 835-4180
Hamilton Hallmark Electronics	(512) 258-8848

Carrollton

Arrow Electronics	(214) 380-6464

Dallas

Hamilton Hallmark Electronics	(214) 553-4300

Houston

Arrow Electronics	(713) 530-4700
Hamilton Hallmark Electronics	(713) 781-6100

Richardson

Anthem Electronics	(214) 238-7100

San Antonio

Hamilton Hallmark Electronics	(210) 828-2246

UTAH

Salt Lake City

Anthem Electronics	(801) 973-8555
Arrow Electronics	(801) 973-6913
Hamilton Hallmark Electronics	(801) 266-2022

WASHINGTON

Bellevue

ALMAC/Arrow Electronics	(206) 643-9992

Bothell

Anthem Electronics	(206) 483-1700

Redmond

Hamilton Hallmark Electronics	(206) 881-6697

Spokane

ALMAC/Arrow Electronics	(509) 924-9500

WISCONSIN

Brookfield

Arrow Electronics	(414) 792-0150

New Berlin

Hamilton Hallmark Electronics	(414) 797-7844

CANADA

Alberta

Future Electronics	(403) 250-5550
Future Electronics	(403) 438-2858

British Columbia

Arrow Electronics	(604) 421-2333
Future Electronics	(604) 294-1166

Manitoba

Future Electronics	(204) 786-7711

Ontario

Arrow Electronics	(613) 226-6903
Arrow Electronics	(905) 670-7769
Future Electronics	(905) 612-9200
Future Electronics	(613) 820-8313

Quebec

Arrow Electronics	(514) 421-7411
Future Electronics	(514) 694-7710

Burnaby

Hamilton Hallmark Electronics	(604) 420-4101

Mississauga

Hamilton Hallmark Electronics	(416) 564-6060

Nepean

Hamilton Hallmark Electronics	(613) 226-1700

Ville St. Laurent

Hamilton Hallmark Electronics	(514) 335-1000

Sales Representatives and Distributors

CENTRAL AND SOUTH AMERICA

MEXICO

Semiconductores Profesionales	525-524-6123
Proyeccion Electronica	525-264-7482

ARGENTINA

Buenos Aires

YEL SRL	011-541-440-1532

BRAZIL

Sao Paulo

Digibyte	011-55-11-581-4100
Nishicom	011-55-11-535-1755

ASIA-PACIFIC

AUSTRALIA

R&D Electronics	61-3-558-0444
GEC Electronics Division	61-2-638-1888

CHINA

Lestina International Ltd.	86-1-849-8888 Rm. 20469
China Electronics Appliance Corp.	86-755-335-4214
TLG Electronics, Ltd.	852-388-7613

HONG KONG

Lestina International Ltd.	852-735-1736
Electrocon Products Ltd.	852-481-6022

INDIA

Bangalore

Zenith Technologies Pvt. Ltd.	91-812-586782

Bombay

Zenith Technologies Pvt. Ltd.	91-22-4947457

JAPAN

Teksel Co., Ltd.	81-44-812-7430
Internix Incorporated	81-3-3369-1101
Kanematsu Electronic Components Corp	81-3-3779-7811

KOREA

ENC-Korea	822-579-3330

MALAYSIA

Eltee Electronics Ltd.	60-3-7038498

NEW ZEALAND

GEC Distributors Ltd.	64-25-971057

PHILIPPINES

Alexan Commercial	63-2-402223

SINGAPORE

Eltee Electronics Ltd.	65-2830888

TAIWAN (ROC)

Acer Sertek, Inc.	886-2-501-0055
Orchard Electronics Co.	886-2-504-7083
Weikeng Ind. Co. Ltd.	886-2-659-0303

THAILAND

Eltee Electronics Ltd.	66-2-530-1739

SALES REPRESENTATIVES AND DISTRIBUTORS

EUROPE

AUSTRIA

Vienna	
EBV Elektronik GMBH	43-222-8941774

BELGIUM

Antwerp	
D & D Electronics PVBA	010-323-82779
Zaventem	
EBV Elektronik	010-322-7209936

DENMARK

Brondby	
Ditz Schweitzer AS	010-4542-453044
Lynge	
Delco A/S	011-45-35-821200

ENGLAND

Berkshire	
Future Electronics	0753-687000
Gothic Crellon	0734-787848
Macro Marketing	0628-604383
Lancastershire	
Complementary Technologies Ltd.	44-942-274731

FINLAND

Espoo	
OY SW Instruments AB	011-358-0-522-122

FRANCE

Cedex	
A2M	010-331-46232425
Massy	
Reptronic SA	010-331-60139300

GERMANY

Berlin	
EBV Elektronik GMBH	030-3421041
Electronic 2000	030-2110761
Burgwedel	
EBV Elektronik GMBH	051-3980870
Dortmund	
Future Electronics	02305-42051
Duesseldorf	
Electronic 2000	021-1920030
Erfurt	
Thesys	036-1476000
Frankfurt	
EBV Elektronik GMBH	069-785037
Electronic 2000	069-973840
Future Electronics	6126-54020
Hamburg	
Electronic 2000	040-64557021
Leonberg	
EBV Elektronik GMBH	071-5230090
Muenchen	
Electronic 2000	089-451101
EBV Elektronik GMBH	089-460960
Future Electronics	089-9571950
Nuernberg	
Electronic 2000	911-9951610
Neuss	
EBV Elektronik GMBH	021-31530072
Stuttgart	
Electronic 2000	071-563560
Future Electronics	711-830830

SALES REPRESENTATIVES AND DISTRIBUTORS

ISRAEL	
RDT	010-972-35483737
ITALY	
Milano	
De Mico S.P.A.	39-295-343600
NETHERLANDS	
Tekelec	010-3179-310100
NORWAY	
Hefro	010-47-22676800
PORTUGAL	
Ibertronics	35-1-471-6587
SPAIN	
Madrid	
Unitronics S.A.	34-1-5428493
SWEDEN	
NC Nordcomp Sweden AB	010-46-87646710
SWITZERLAND	
Dietikon	
EBV Elektronik GMBH	41-1-7401090
Regensdorf	
Moor elektronik AG	41-1-8433111
Lausanne	
EBV Elektronik GMBH	41-21-3112804

DC 8314-01

Zilog, Inc.
210 East Hacienda Ave.
Campbell, CA 95008-6600
408-370-8000