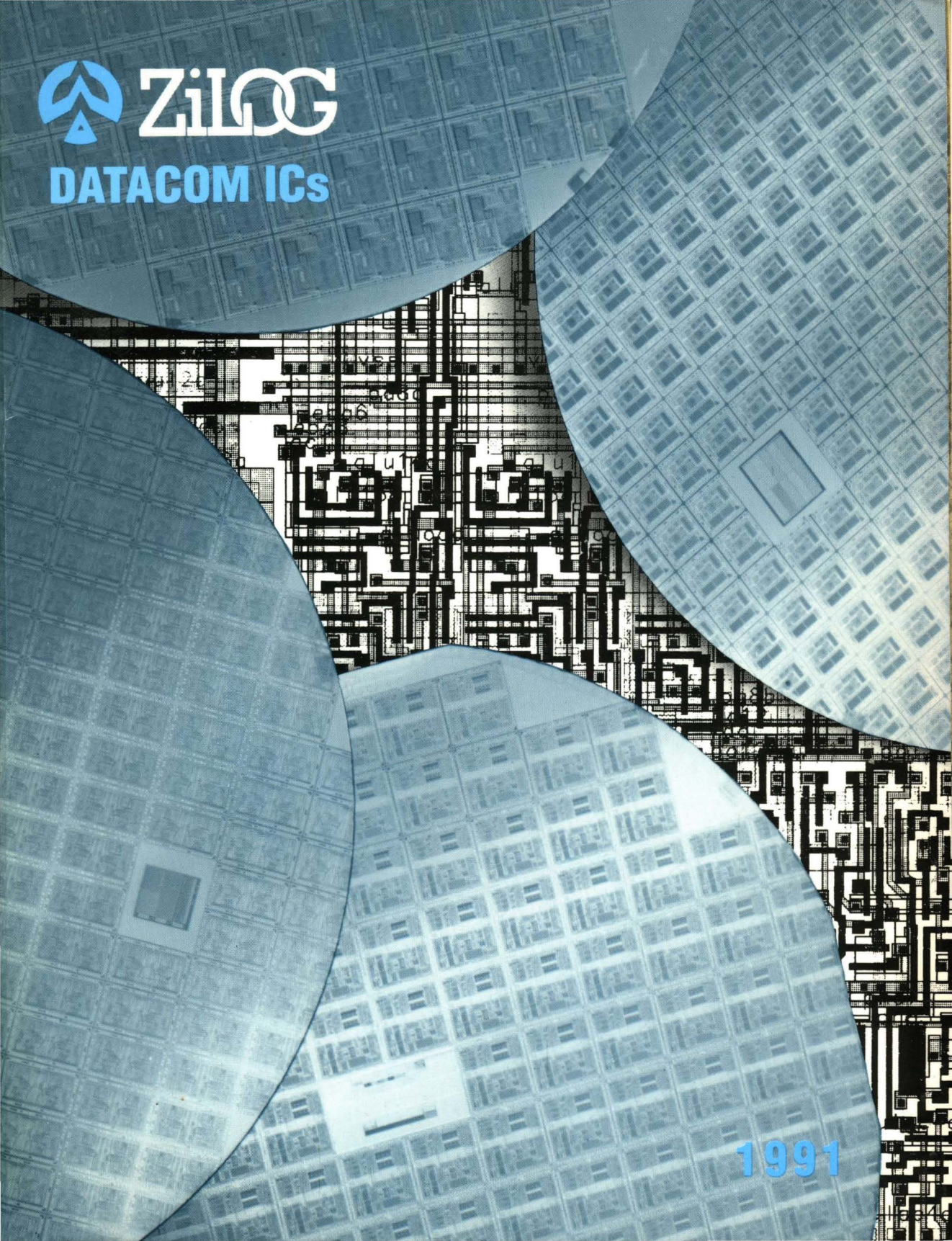
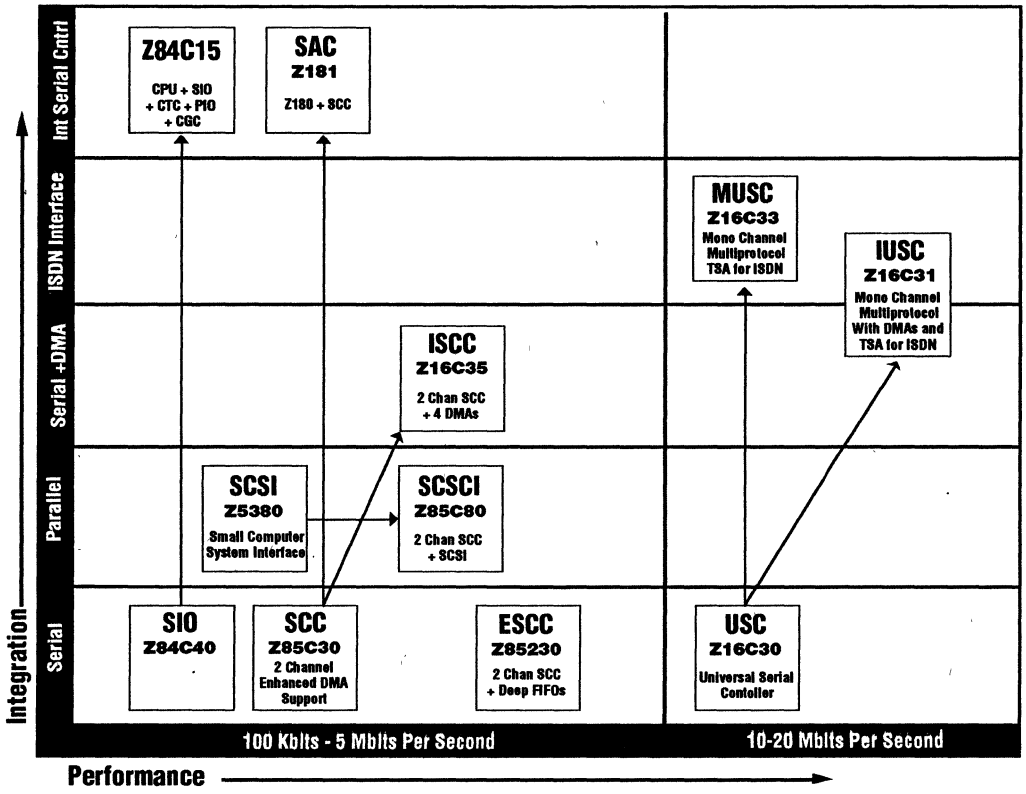

ZiLOG
DATACOM ICs
1991

Table of Contents

DATACOM PRODUCTS



ZiLOG

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
- 12.5 MByte/sec (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.

GENERAL DESCRIPTION (Continued)

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.

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 via DMA are supported by a Request/Acknowledge signal pair for each receiver and transmitter. The device supports automatic status transfer via 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 Signals with a preceding front slash, "/", are active Low, e.g.: B//W (WORD is active Low); /B/W (BYTE is active Low, only); /N//S (NORMAL and SYSTEM are both active Low).

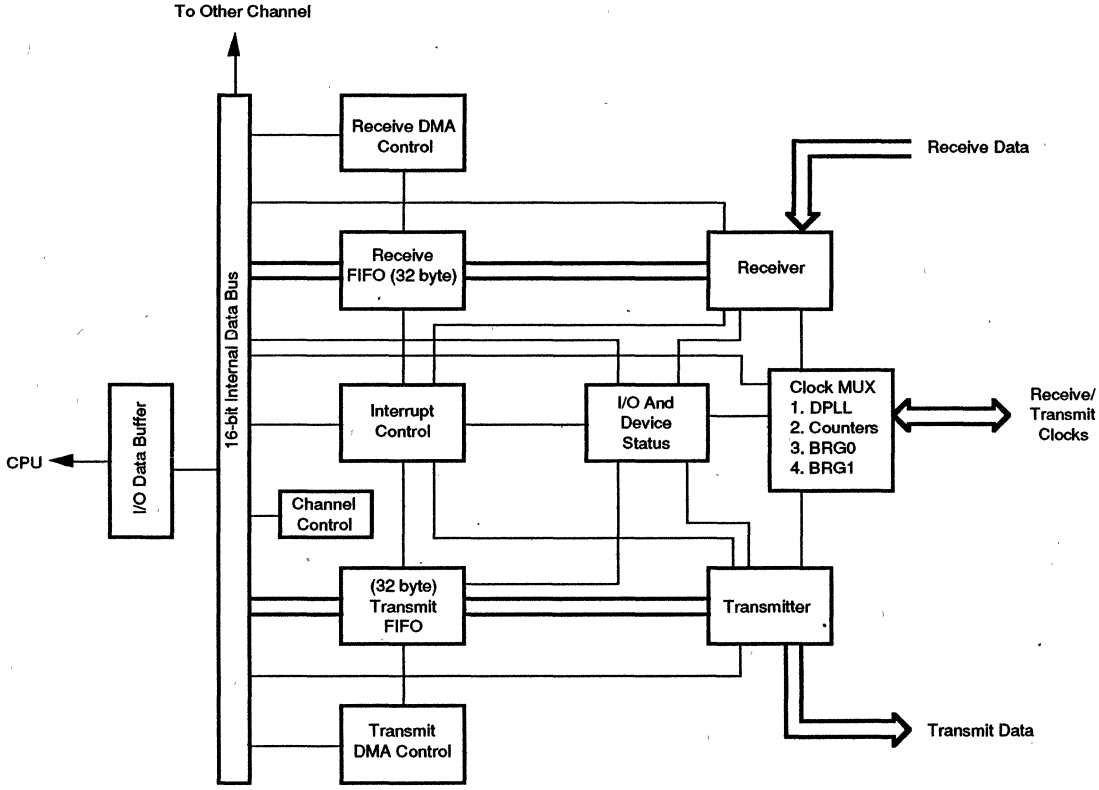


Figure 1. USC Block Diagram

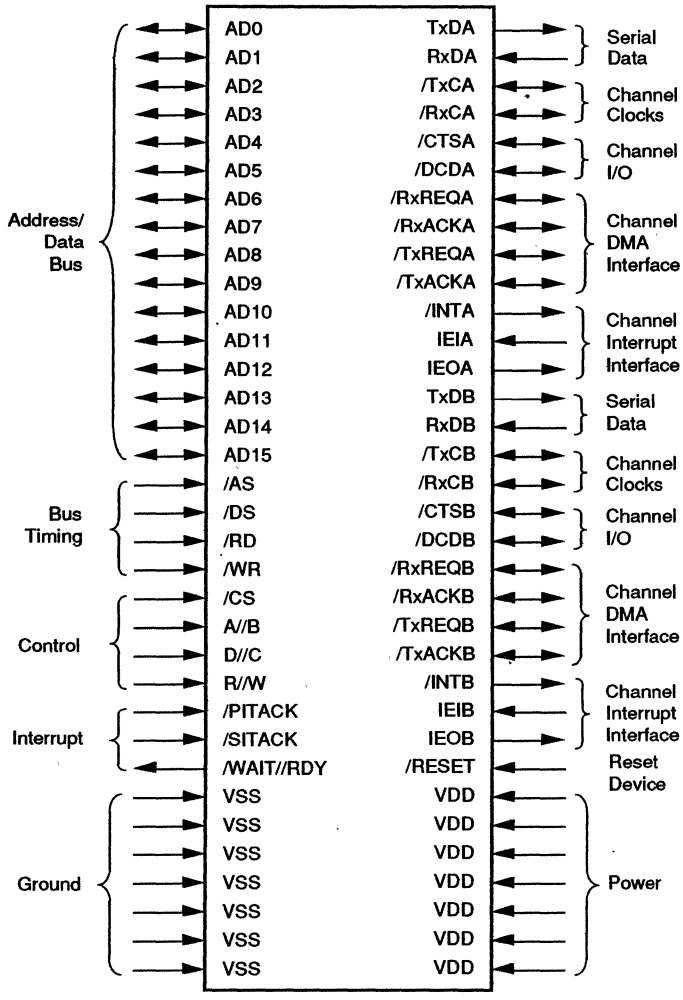
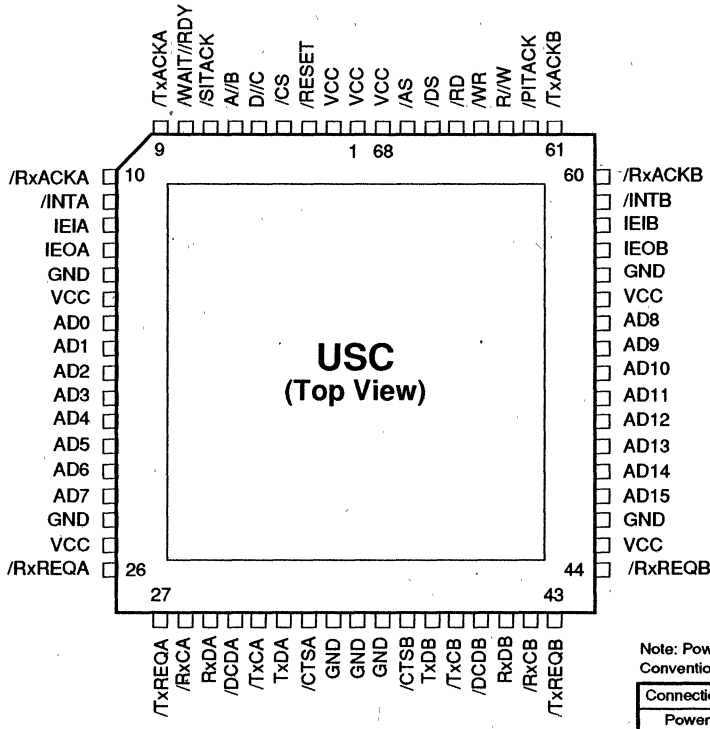


Figure 2. Pin Functions

Note: Power connections follow conventional descriptions below:

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



Note: Power connections follow Conventional descriptions below

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

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 VSS.

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).

The section below describes in detail the USC pin assignment.

/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 VDD.

/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. See /WAIT//RDY below.

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, 3-state). The AD signals carry addresses to, and data to and from, the device. When the 16-bit non-multiplexed bus is selected, AD15-0 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 AD7-0 are used to transfer data. The pointer is used for addressing, with AD15-8 unused. When selecting the 8-bit non-multiplexed bus (with separate address), AD7-0 are used to transfer data with AD15-8 used as address bus. When the 16-bit multiplexed bus is selected, addresses are latched from AD7-0 and data transfers are sixteen bits wide. When selecting the 8-bit multiplexed bus (without separate address) only AD7-0 are used to transfer addresses and data, with AD15-8 unused. When the 8-bit multiplexed bus with separate address is selected, only AD7-0 are used to transfer data, while AD15-8 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.

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, 3-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.

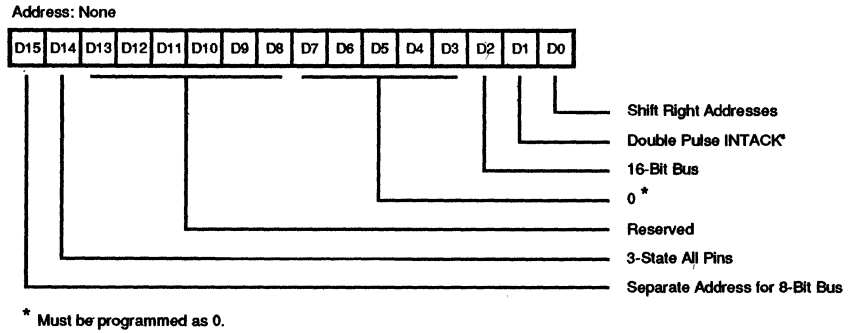


Figure 4. Bus Configuration Register

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 echos 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.

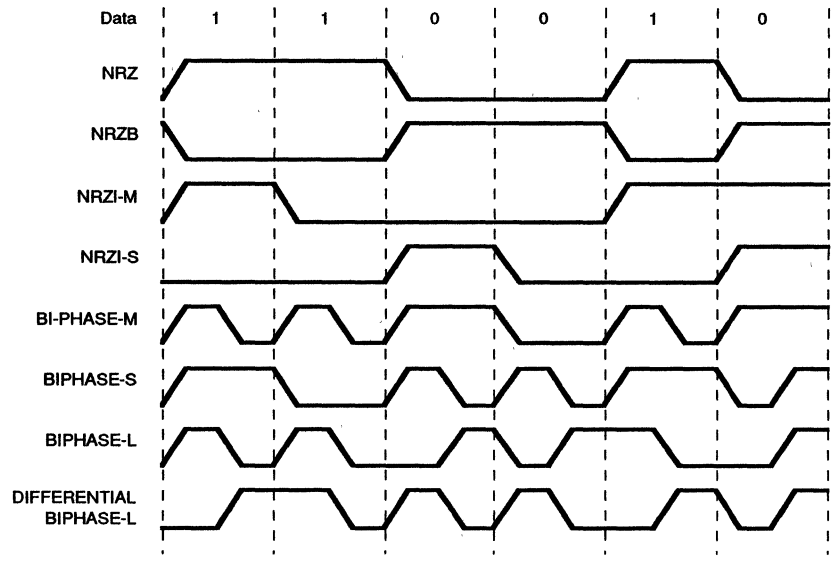


Figure 5. Data Encoding

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 is 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 via 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 Programmers Assistant (MS DOS based) and Technical Manual are available to provide details about programmming the USC. Also included are explanations and features of all registers in the USC

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 via 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.

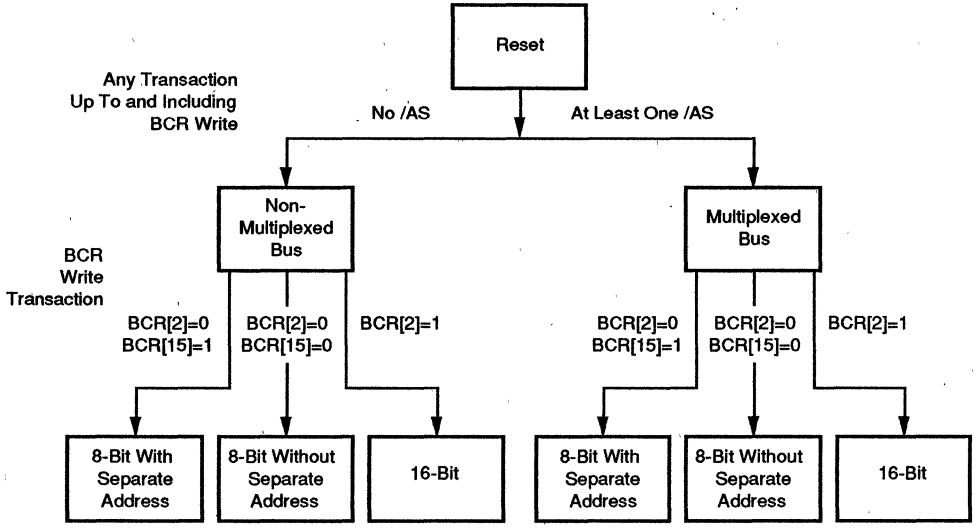
In the non-multiplexed bus case, the registers in each channel are accessed indirectly using the address pointer in the Channel Command/Address Register (CCAR) in each channel. The address of the desired register is first written to the CCAR and then the selected register is accessed; the pointer in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



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

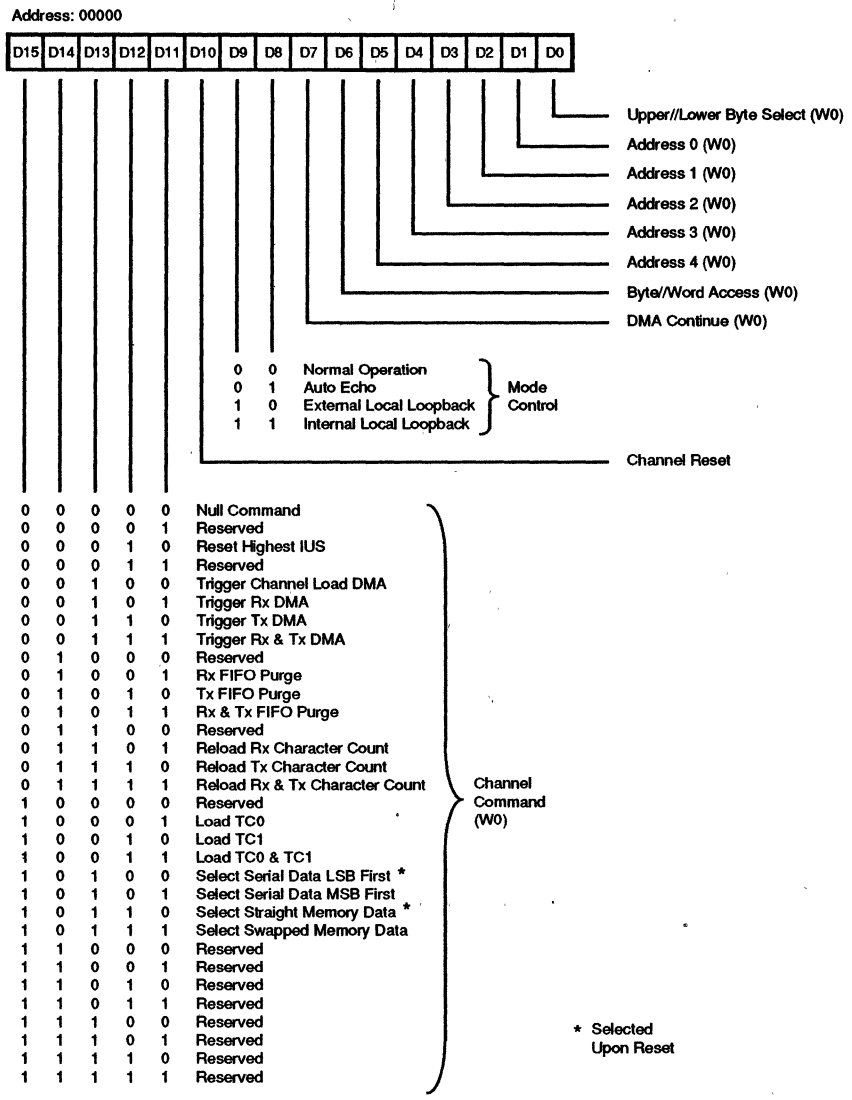


Figure 7. Channel Command/Address Register

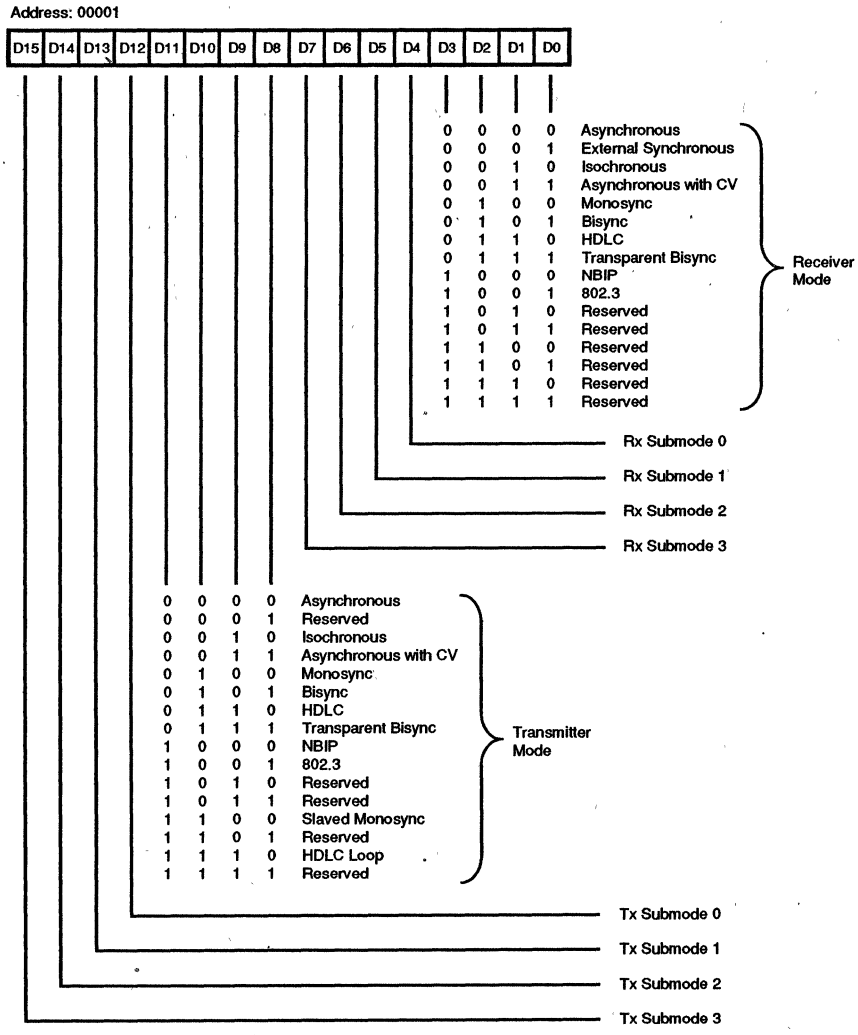


Figure 8. Channel Mode Register

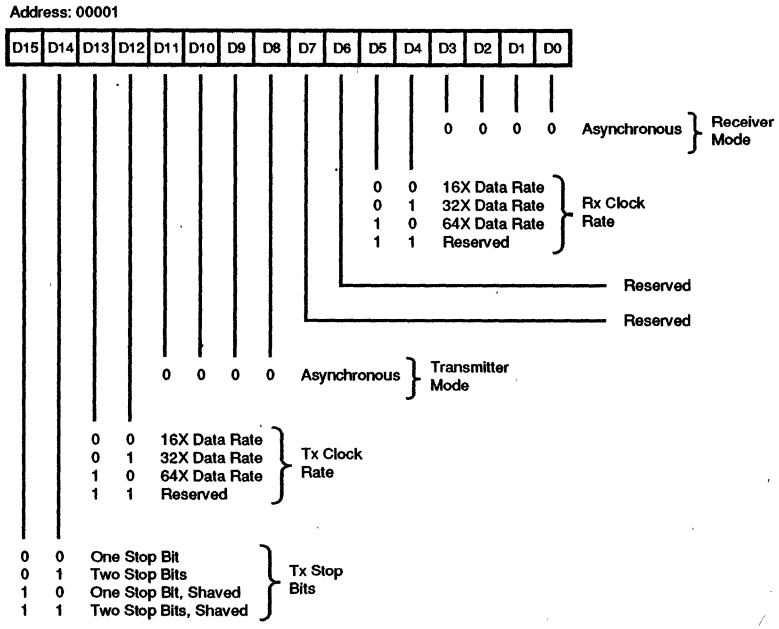


Figure 9. Channel Mode Register, Asynchronous Mode

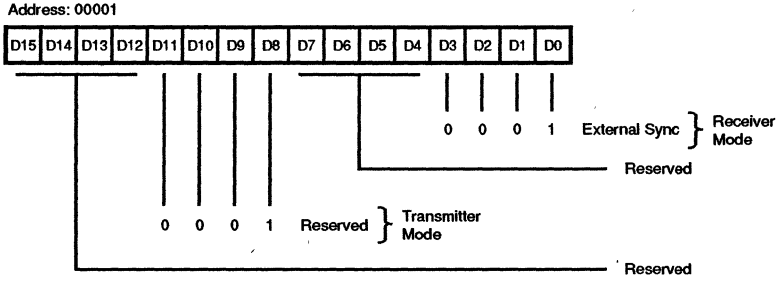


Figure 10. Channel Mode Register, External Sync Mode

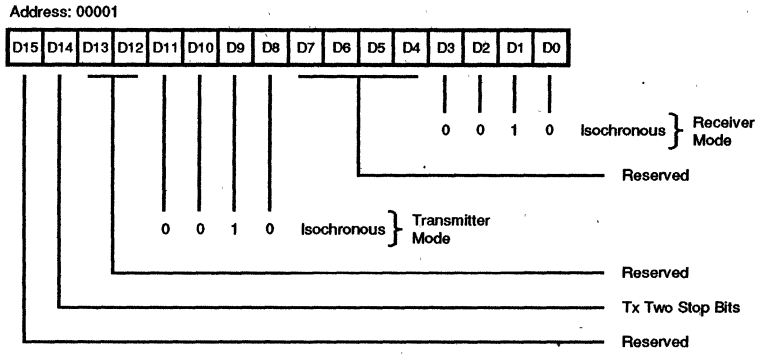


Figure 11. Channel Mode Register, Isochronous Mode

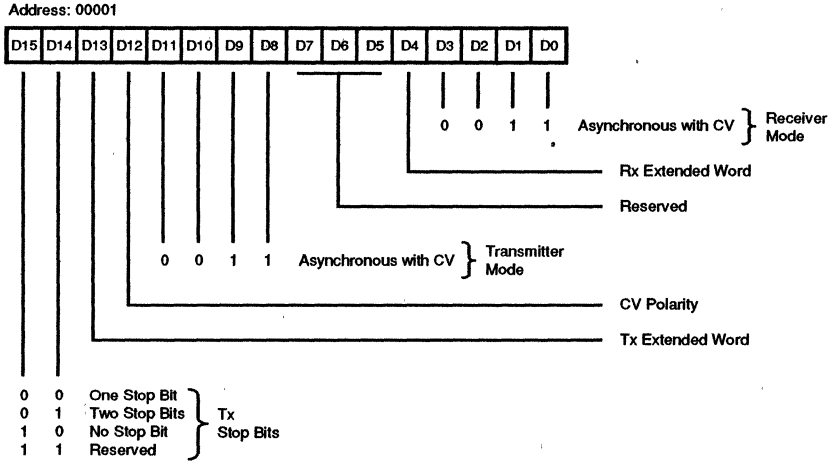


Figure 12. Channel Mode Register, Asynchronous Mode with Code Violation (MIL STD 1553)

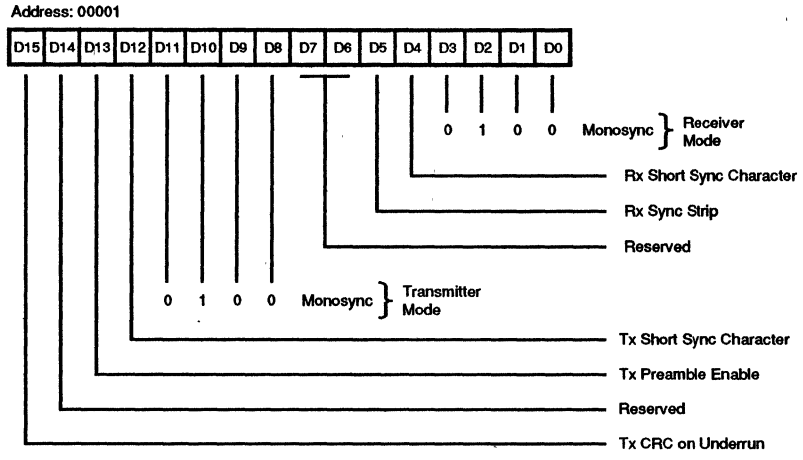


Figure 13. Channel Mode Register, Monosync Mode

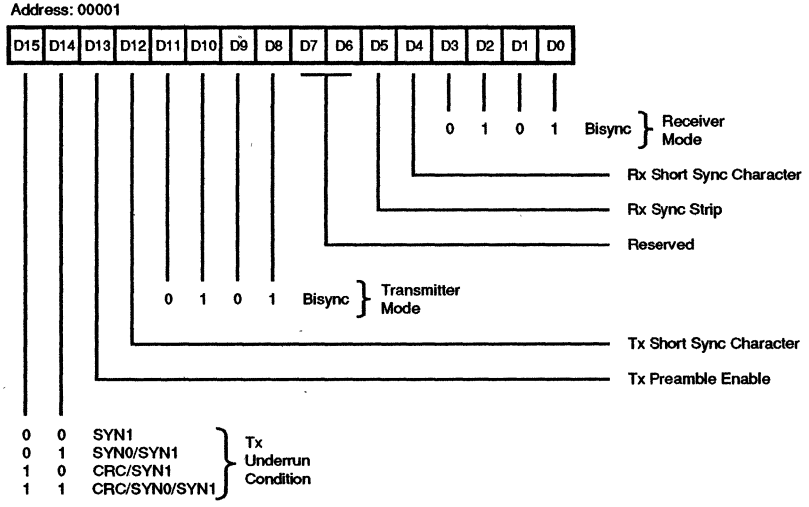


Figure 14. Channel Mode Register, Bisync Mode

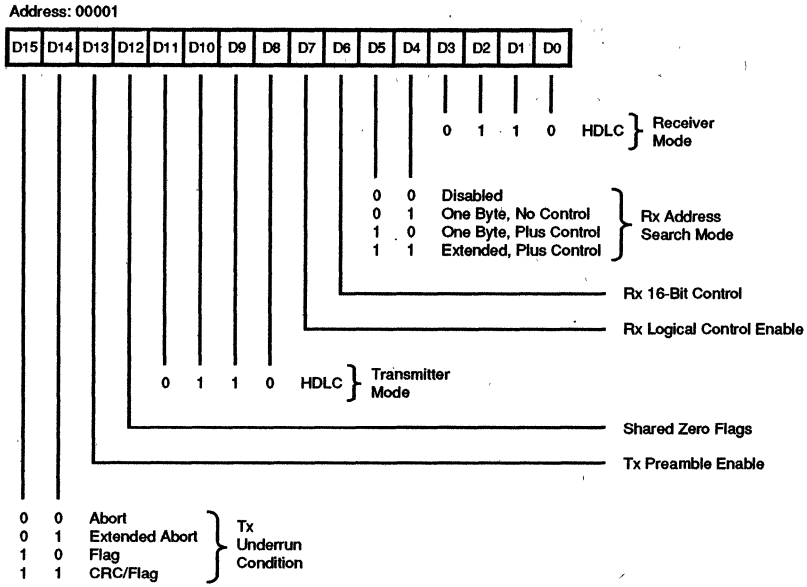


Figure 15. Channel Mode Register, HDLC Mode

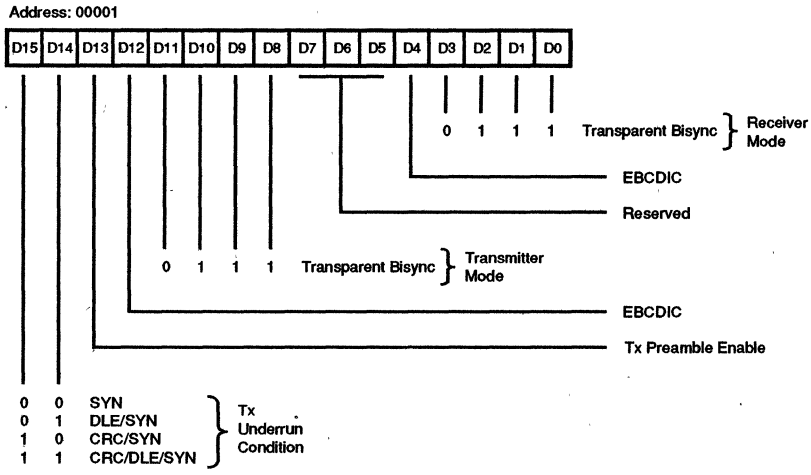


Figure 16. Channel Mode Register, Transparent Bisync Mode

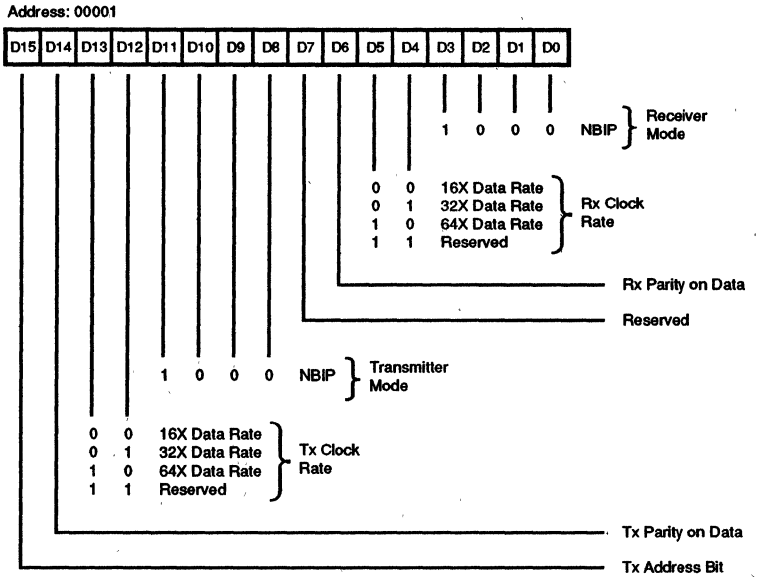


Figure 17. Channel Mode Register, NBIP Mode

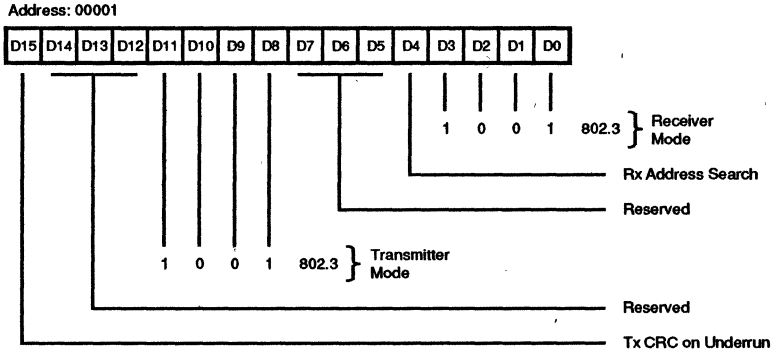


Figure 18. Channel Mode Register, 802.3 Mode

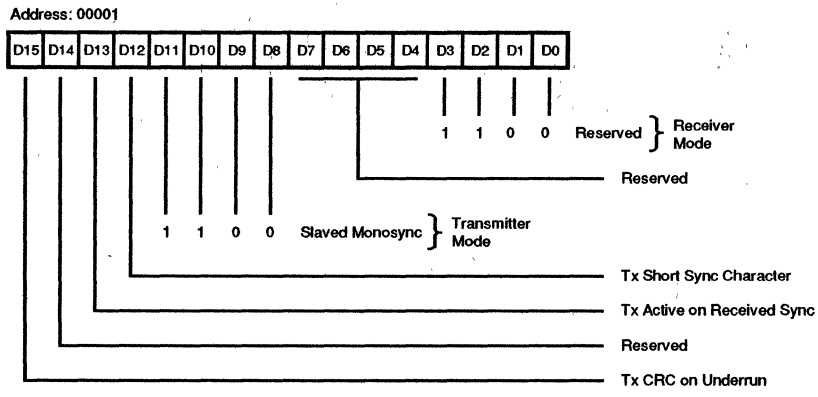


Figure 19. Channel Mode Register, Slaved Monosync Mode

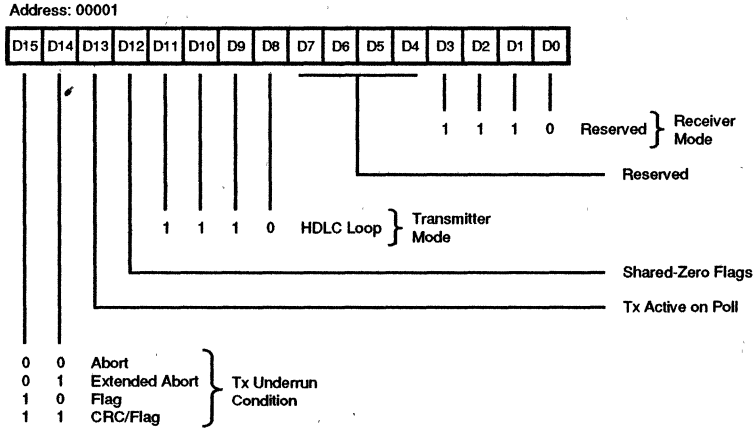


Figure 20. Channel Mode Register, HDLC Loop Mode

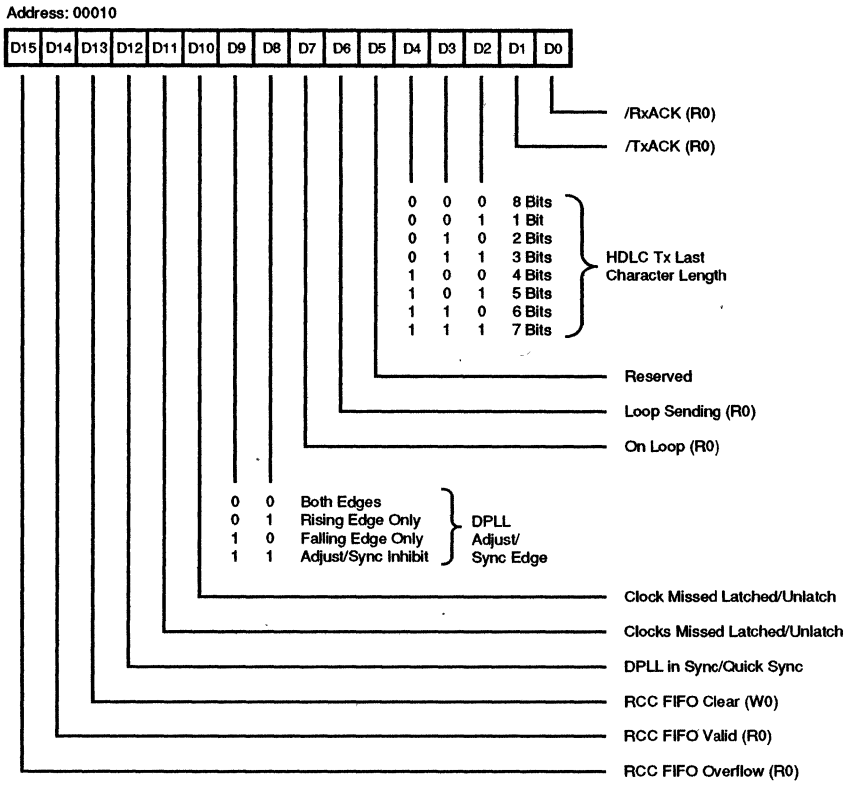


Figure 21. Channel Command/Status Register

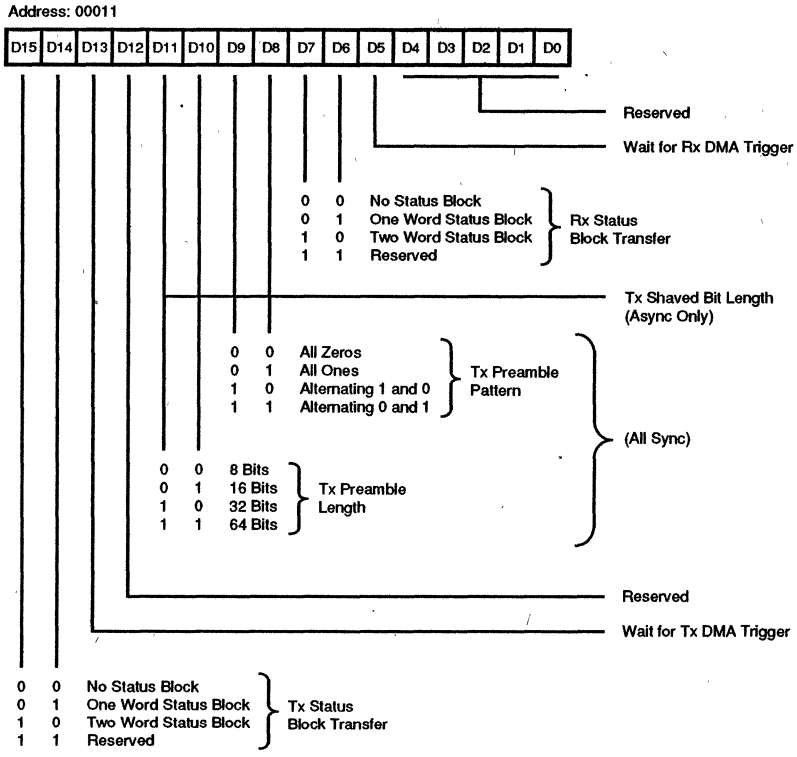


Figure 22. Channel Control Register

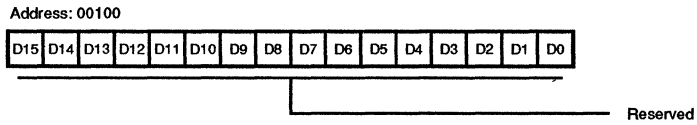


Figure 23. Primary Reserved Register

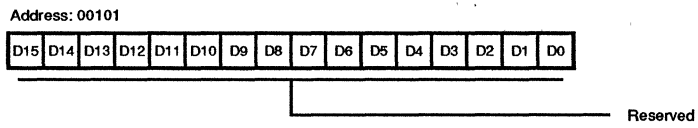


Figure 24. Secondary Reserved Register

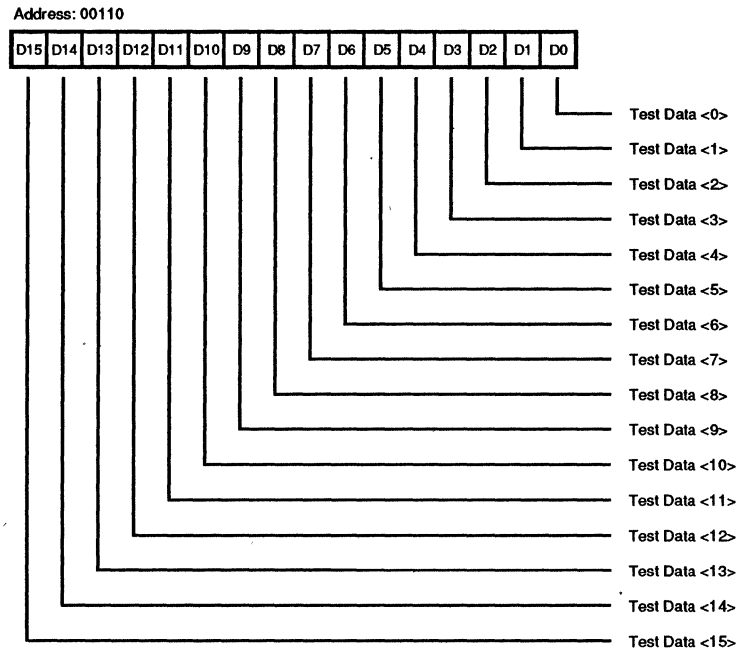


Figure 25. Test Mode Data Register

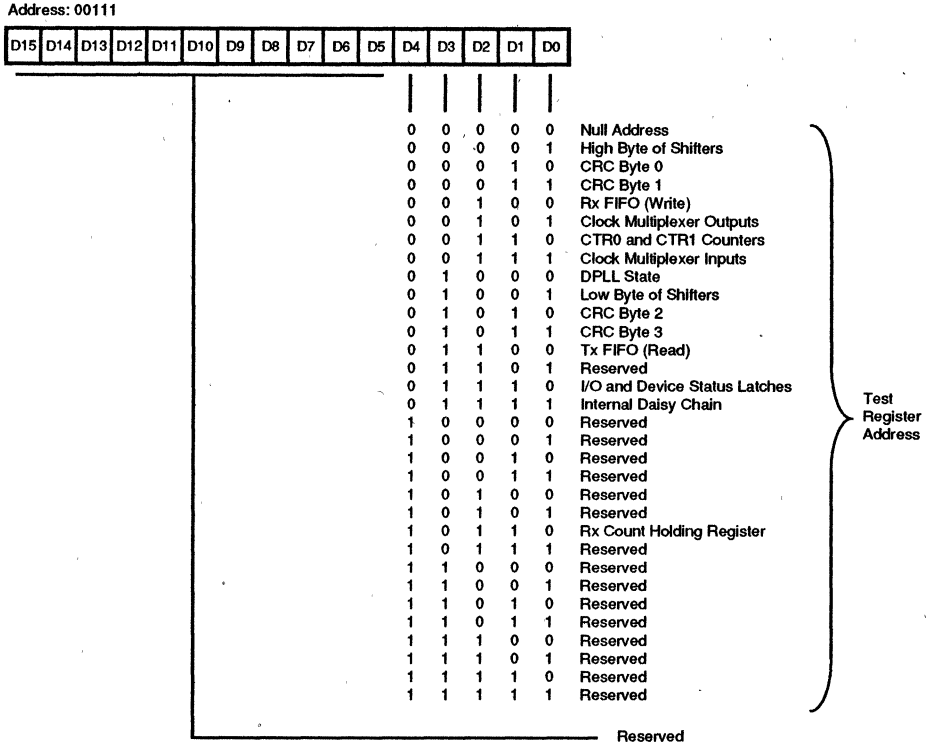


Figure 26. Test Mode Control Register

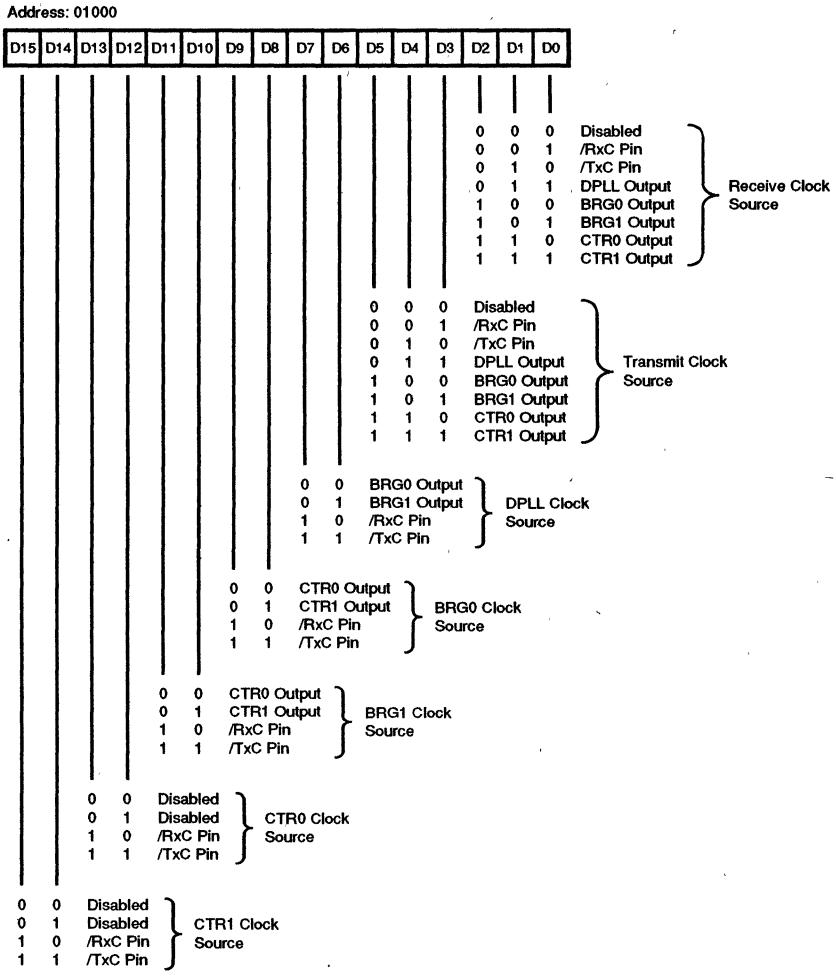


Figure 27. Clock Mode Control Register

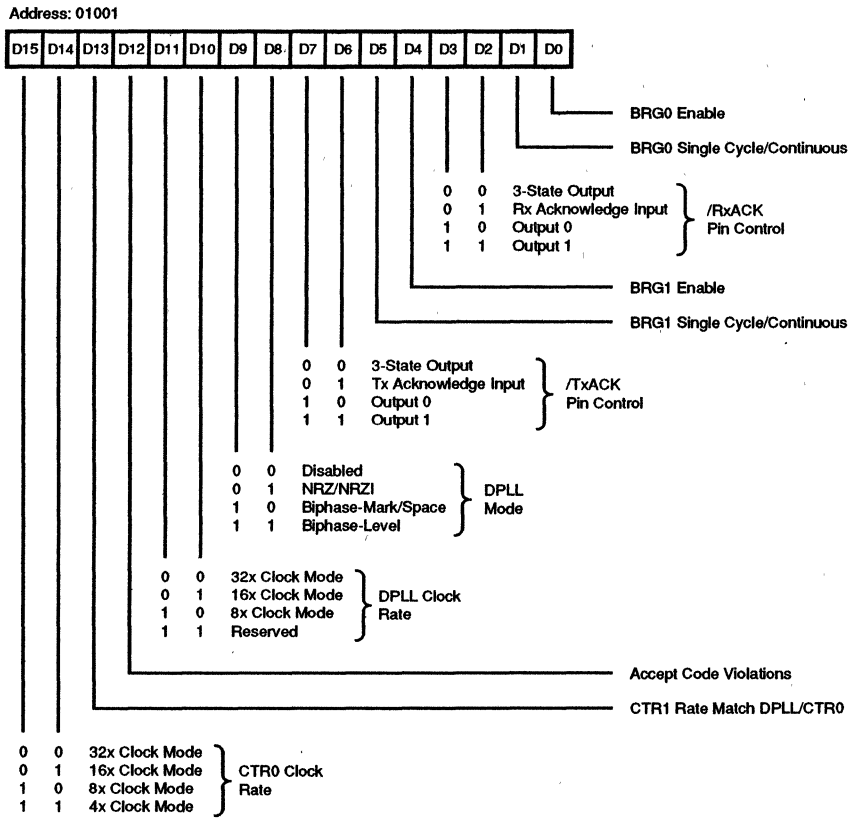


Figure 28. Hardware Configuration Register

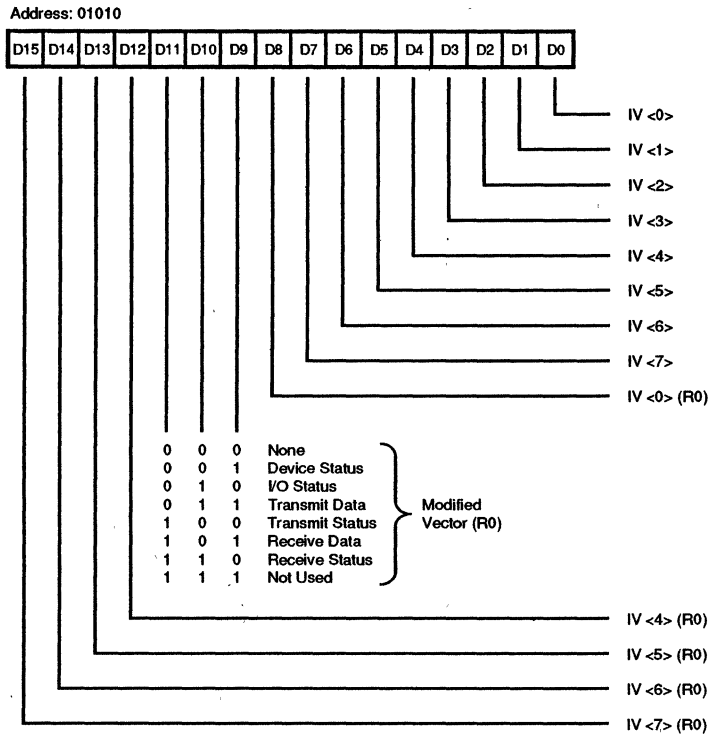


Figure 29. Interrupt Vector Register

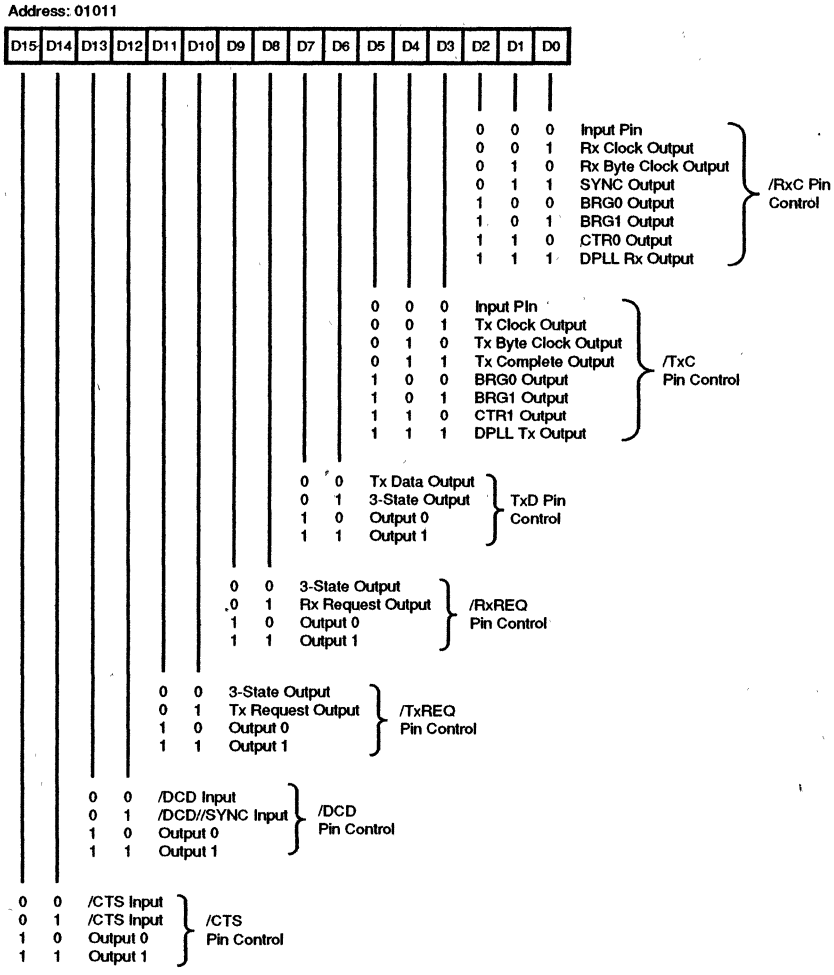


Figure 30. I/O Control Register

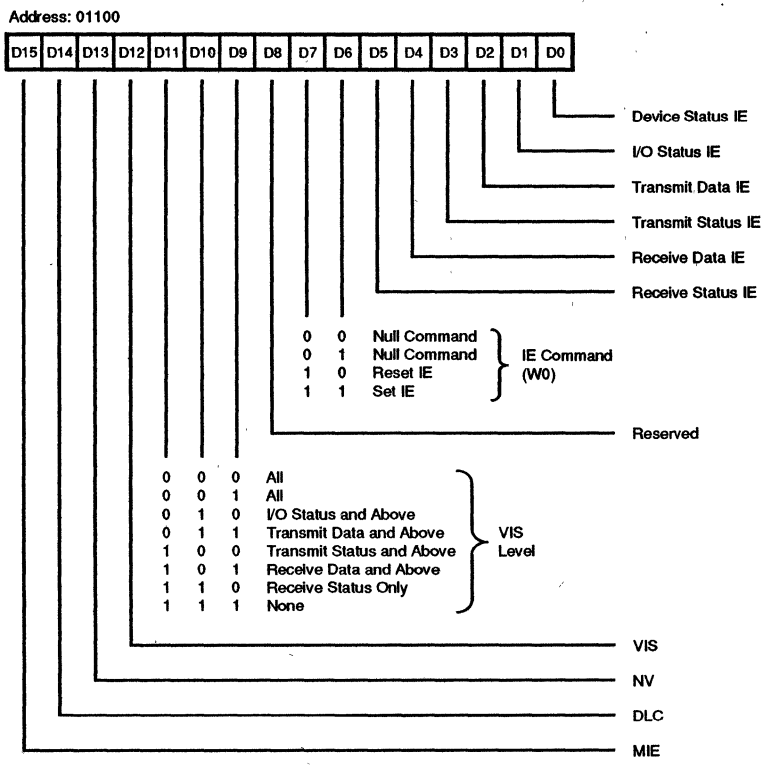


Figure 31. Interrupt Control Register

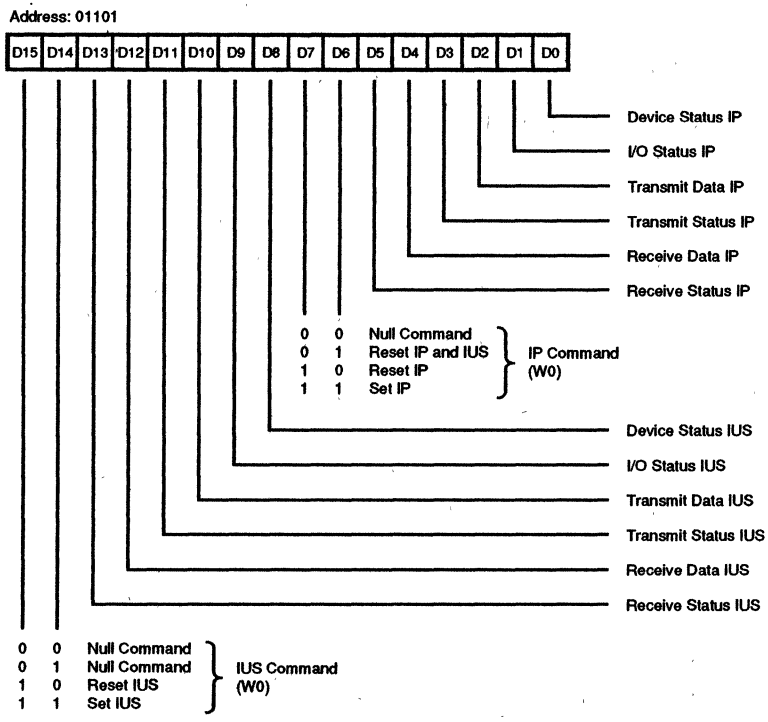


Figure 32. Daisy-Chain Control Register

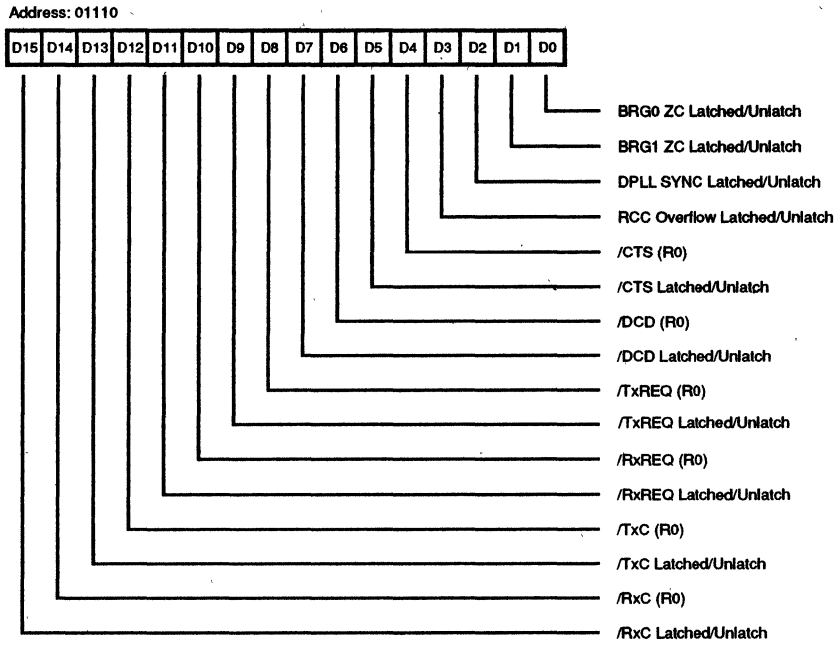


Figure 33. Miscellaneous Interrupt Status Register

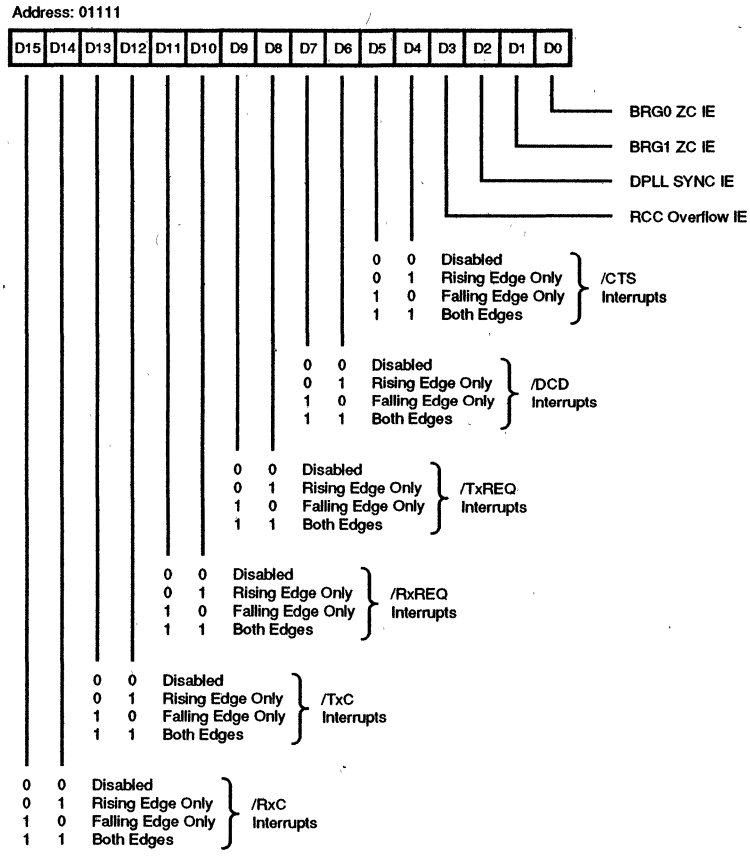


Figure 34. Status Interrupt Control Register

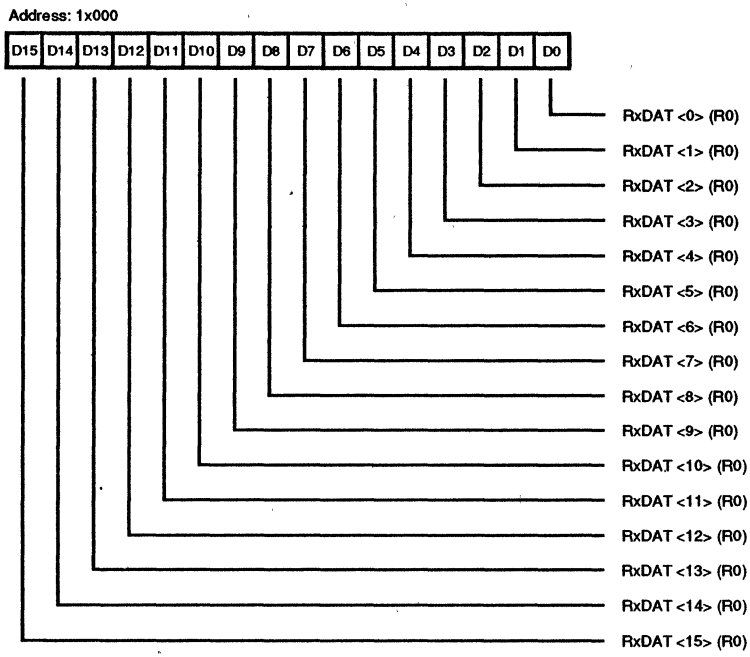


Figure 35. Receive Data Register

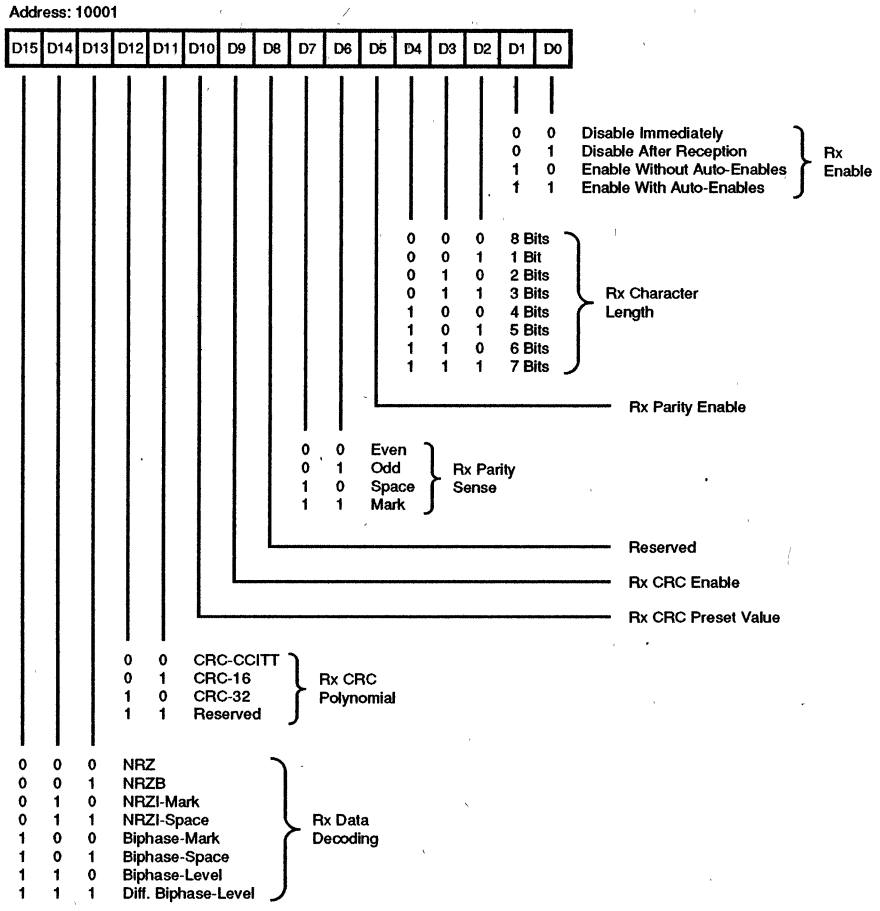


Figure 36. Receive Mode Register

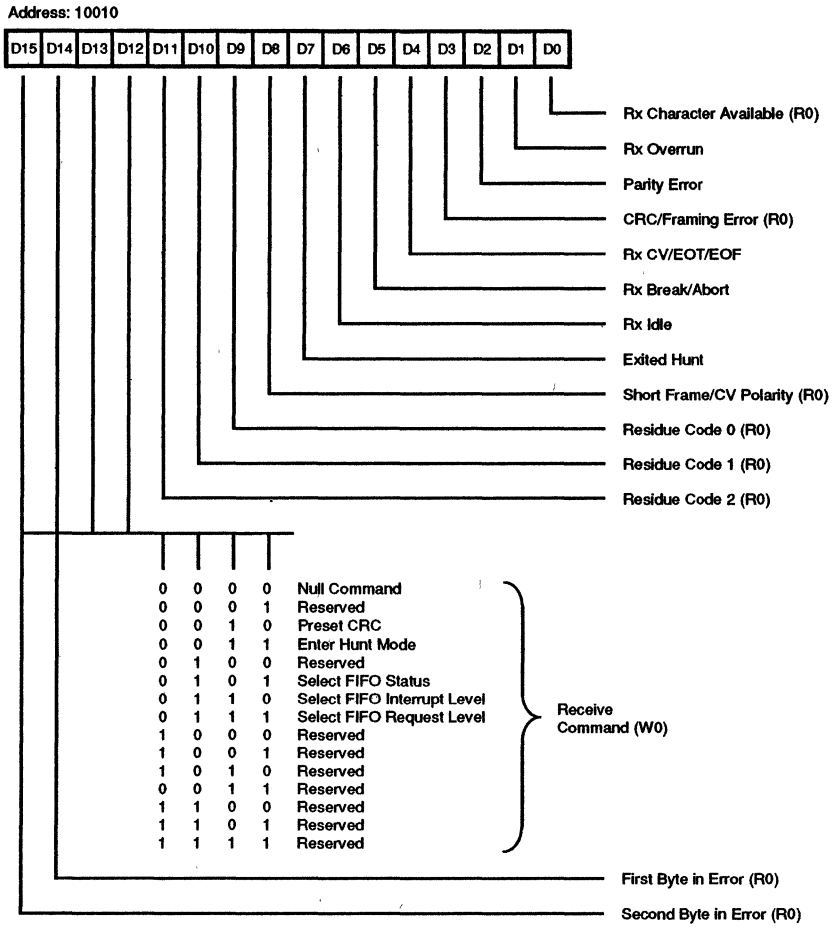


Figure 37. Receive Command Status Register

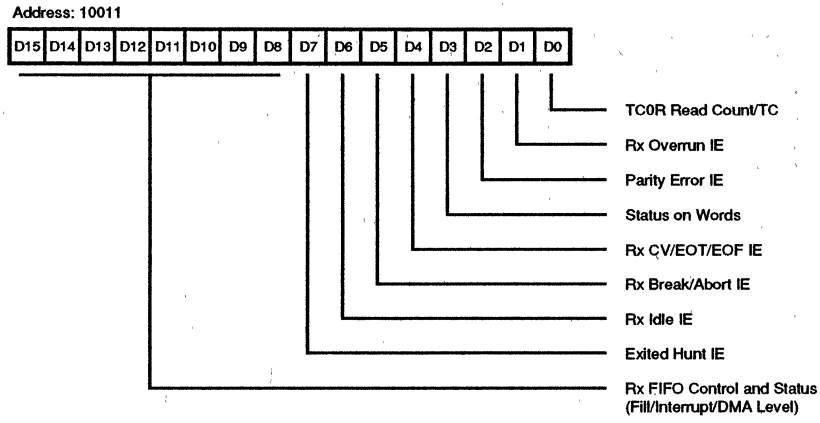


Figure 38. Receive Interrupt Control Register

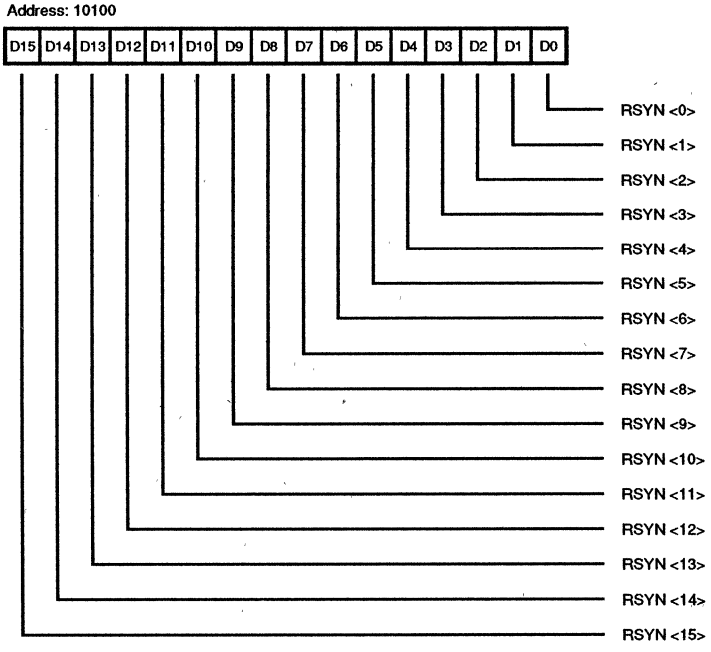


Figure 39. Receive Sync Register

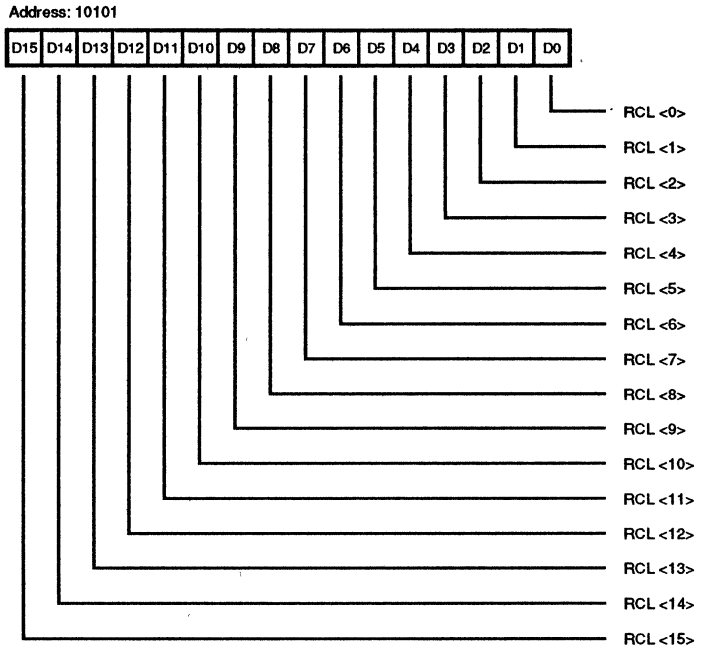


Figure 40. Receive Count Limit Register

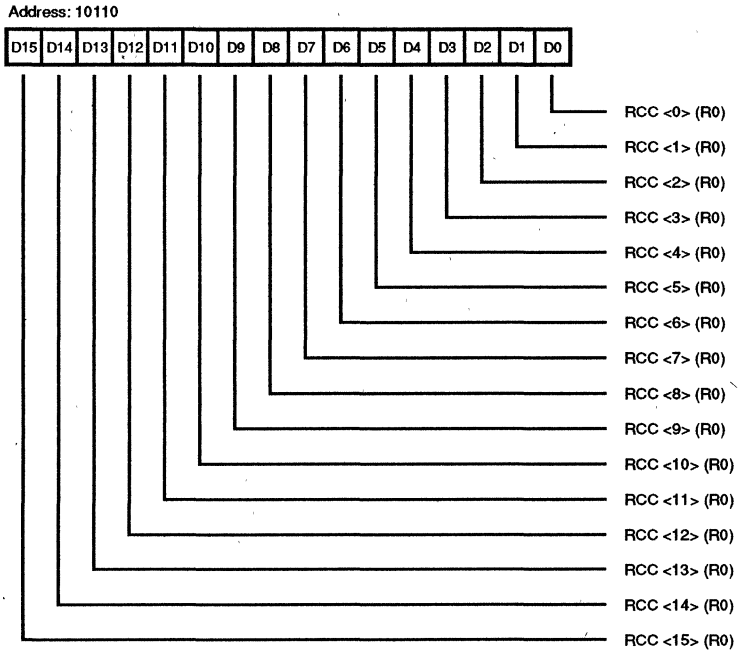


Figure 41. Receive Character Count Register

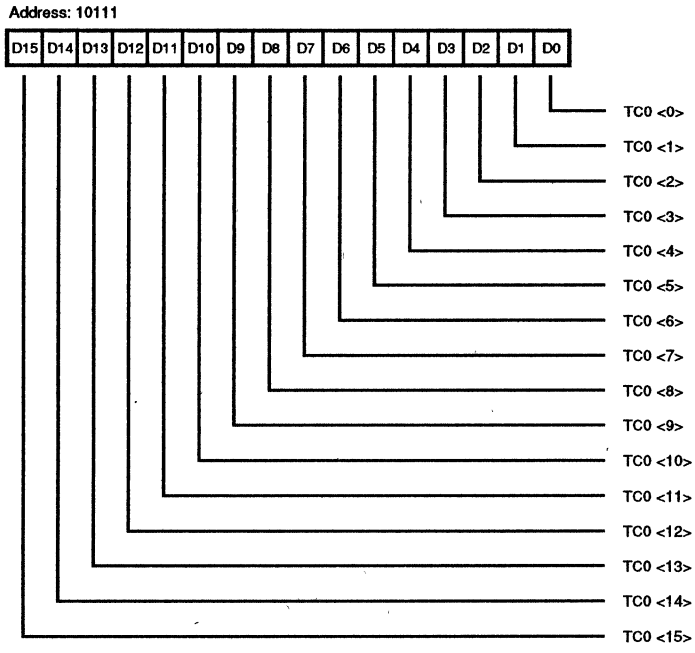


Figure 42. Time Constant 0 Register

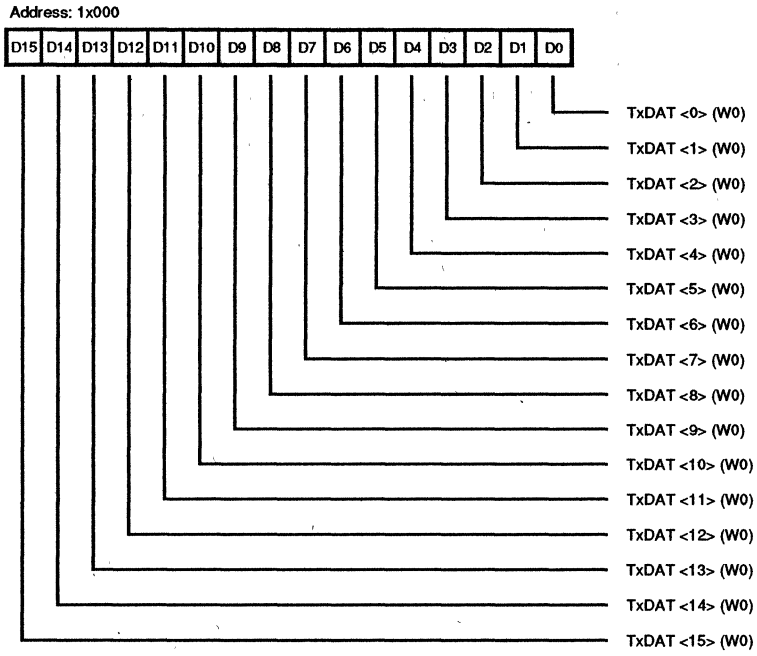


Figure 43. Transmit Data Register

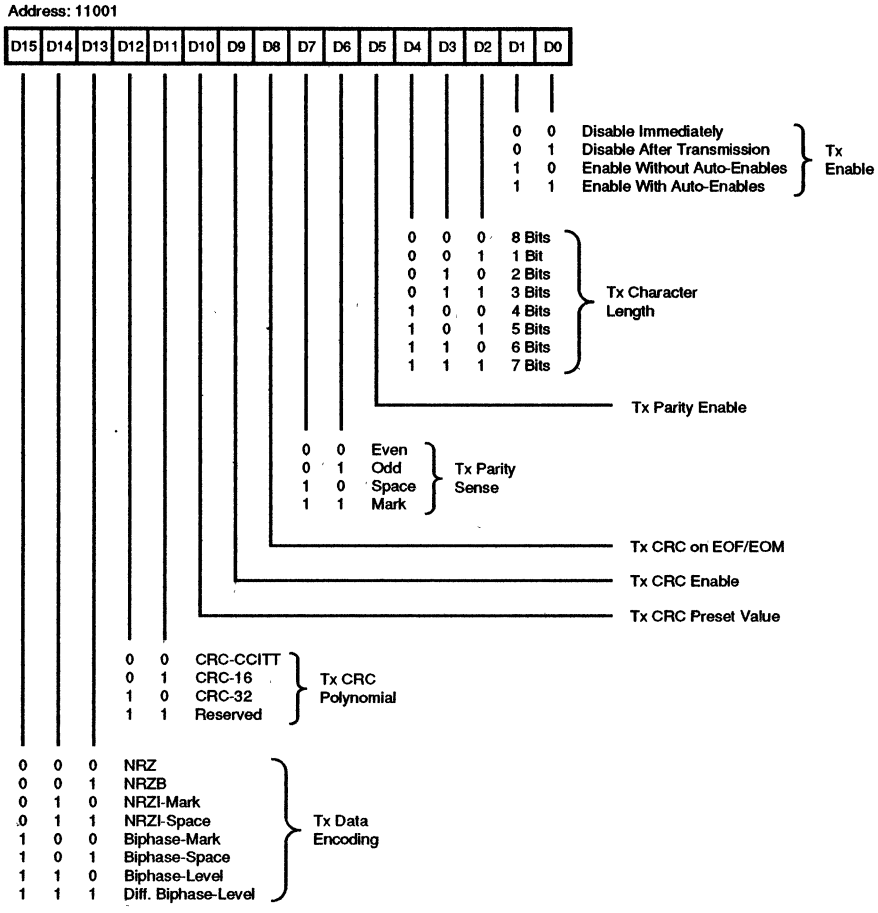


Figure 44. Transmit Mode Register

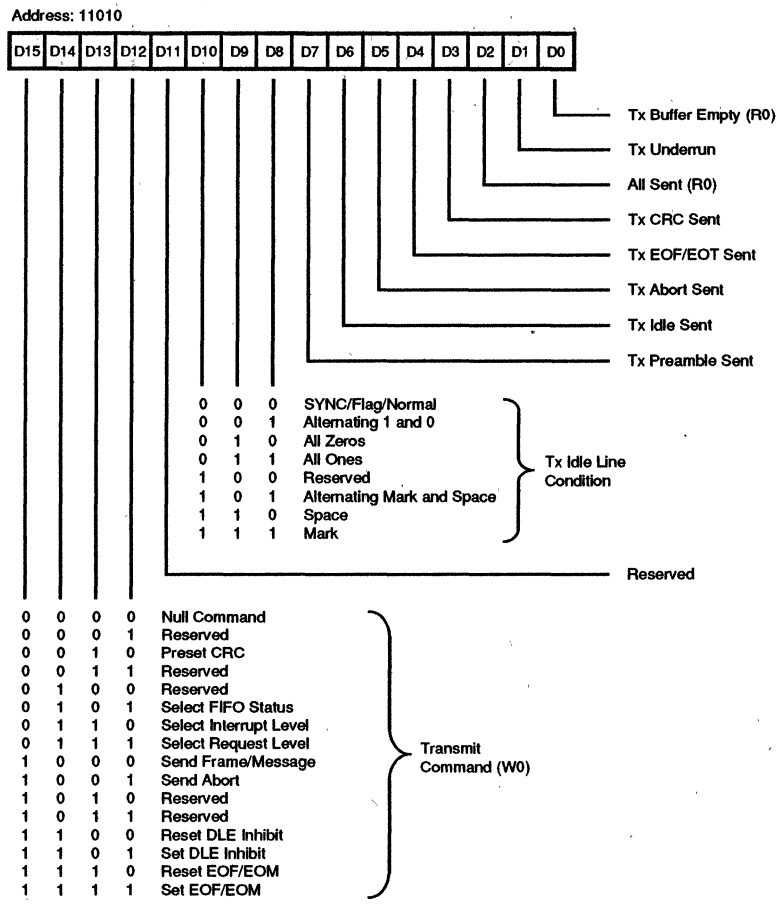


Figure 45. Transmit Command/Status Register

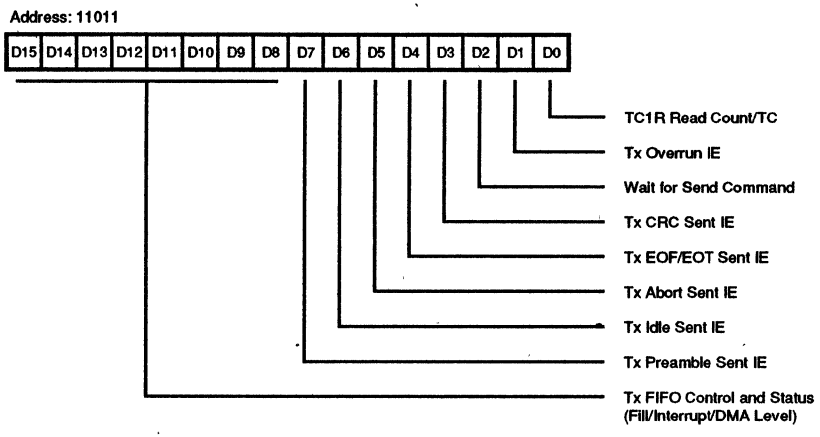


Figure 46. Transmit Interrupt Control Register

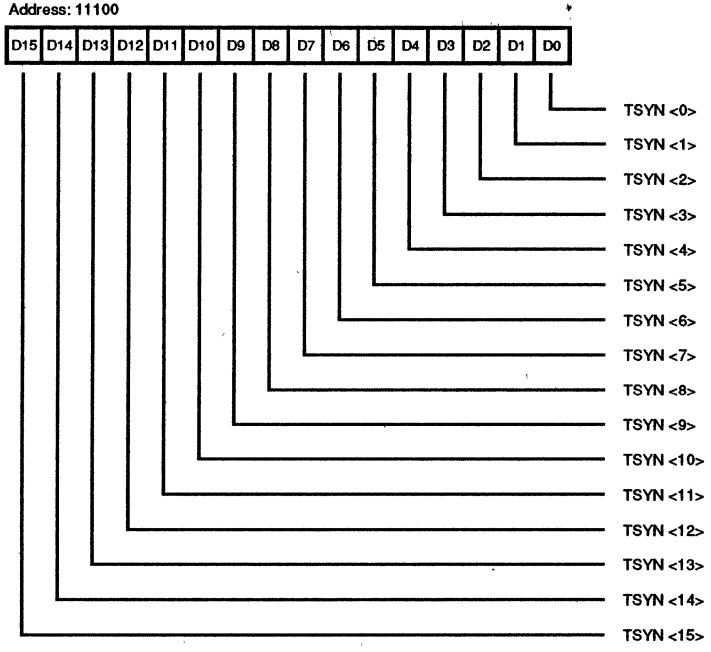


Figure 47. Transmit Sync Register

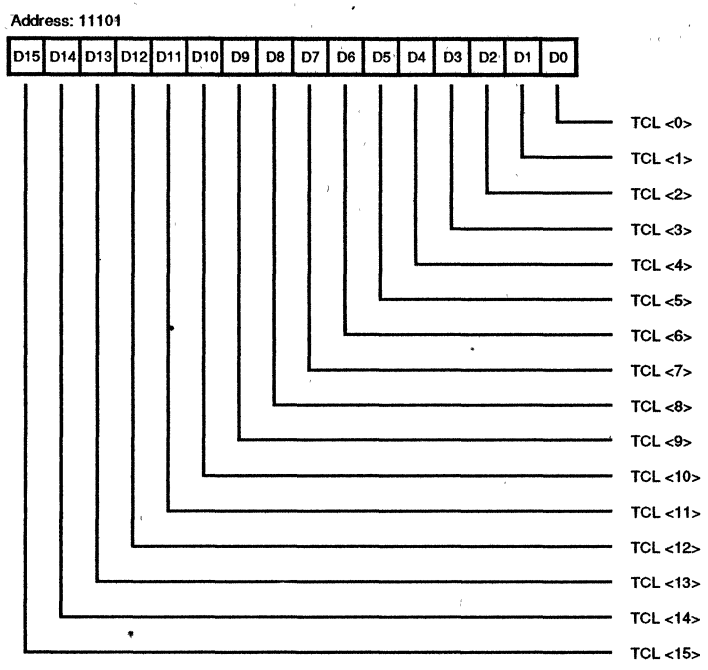


Figure 48. Transmit Count Limit Register

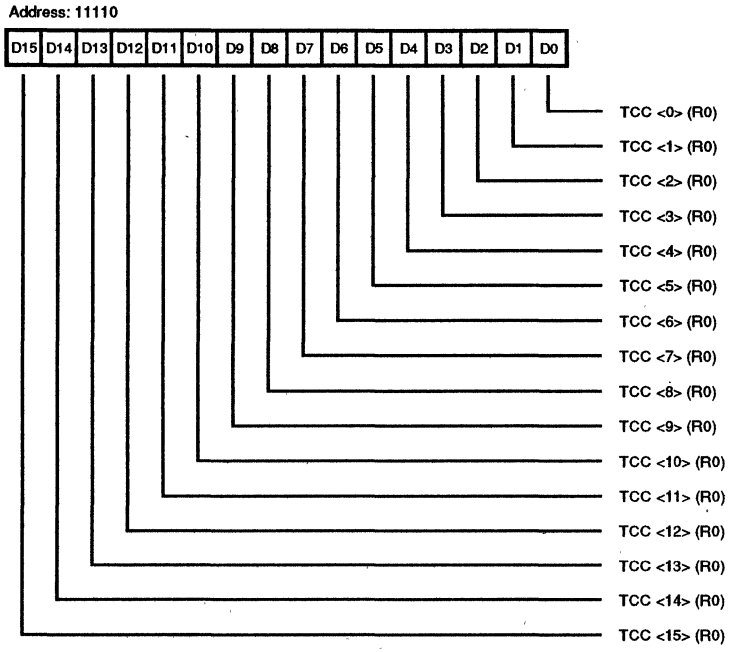


Figure 49. Transmit Character Count Register

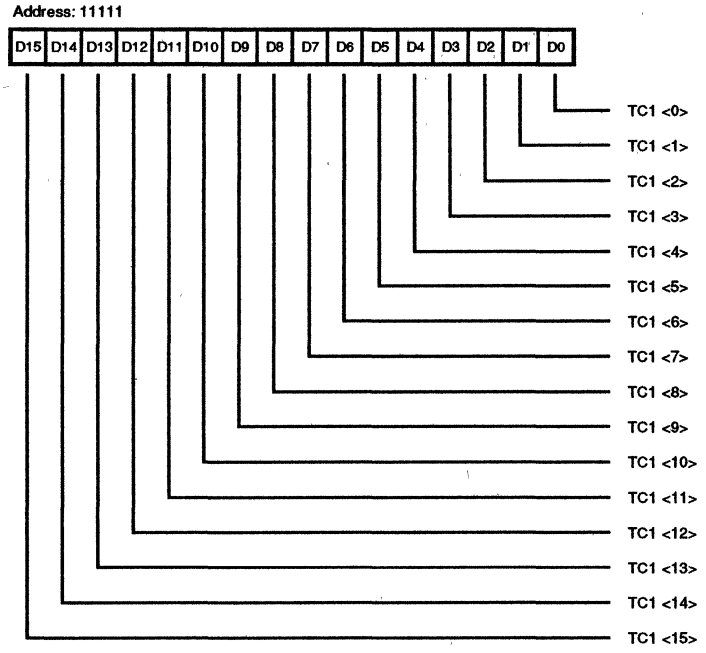
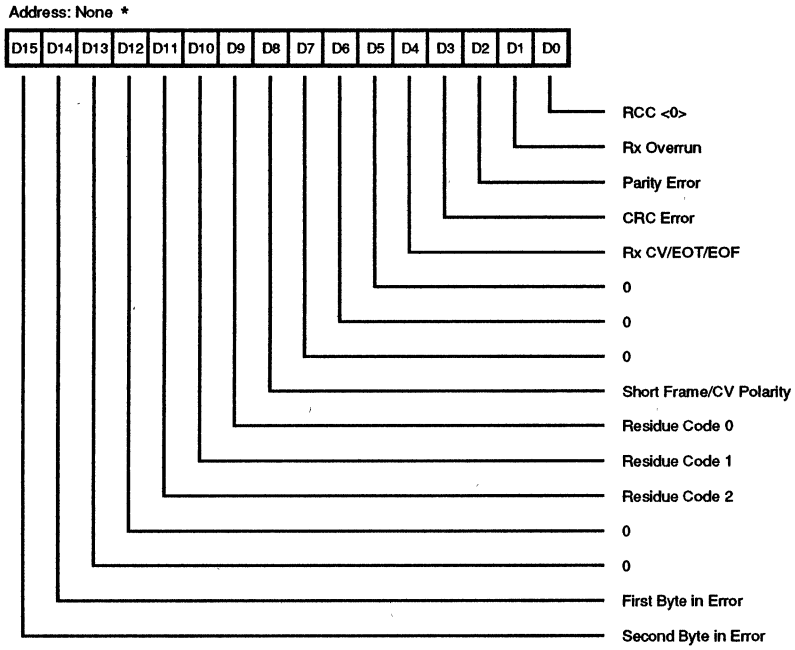


Figure 50. Time Constant 1 Register



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

Figure 51. Receive Status Block Register

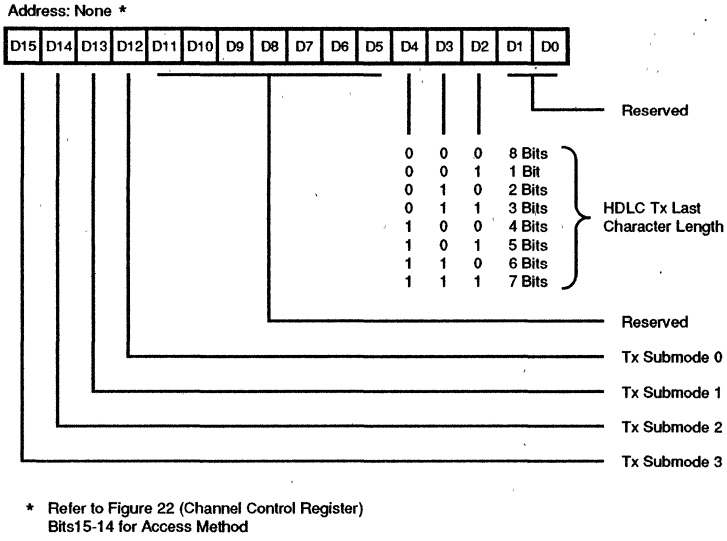


Figure 52. Transmit Status Block Register

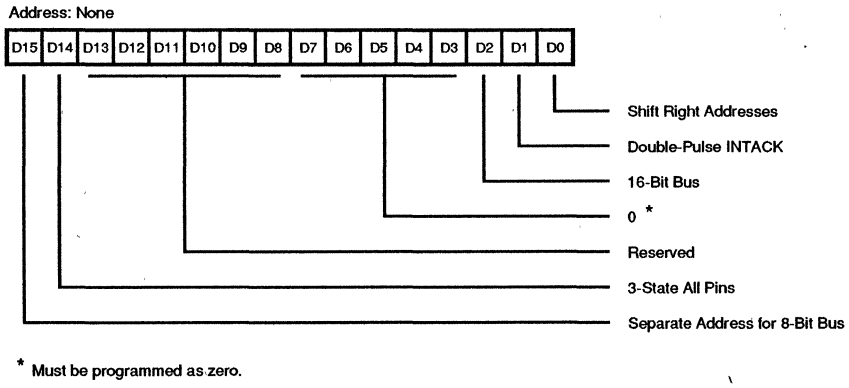


Figure 53. Bus Configuration Register

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

Voltages on all pins
with respect to Vss .....-0.3 V to +7.0 V
Voltages on all inputs
with respect to Vss ....-0.3V to Vcc +0.3V
Operating Ambient
Temperature .....See Ordering Information
Storage Temperature .....-65°C to +150°C

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. Standard conditions are as follows:

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

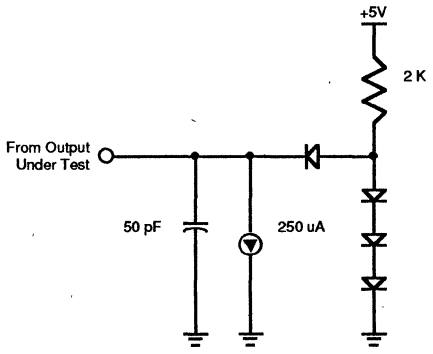


Figure 54. Standard Test Load

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=1MHz, over specified temperature range.
Unmeasured pins returned to ground

MISCELLANEOUS Transistor Count - 174,000

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}= -1.6mA
$V_{OH}2$	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	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	0		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	
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]
27	TdDSr(TRQ)	/DS Rise to /TxREQ Active Delay	0		ns	
28	TwRDl	/RD Low Width	70		ns	

AC CHARACTERISTICS
Z16C30

No	Symbol	Parameter	Min	Max	Units	Note
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		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]
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	70		ns	
65	TwRAKh	/RxACK High Width	60		ns	
66	TdRAK(DRa)	/RxACK Fall to Data Active Delay	0		ns	
67	TdRAK(DRv)	/RxACK Fall to Data Valid Delay		85	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	70		ns	

AC CHARACTERISTICS
Z16C30

No	Symbol	Parameter	Min	Max	Units	Note
73	TwTAKh	/TxACK High Width	60		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	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 (INTACK) 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	/PITACK Low Width	70		ns	
97	TwPIAh	/PITACK High Width	60		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	
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	60		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	60		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	

AC CHARACTERISTICS
Z16C30

No	Symbol	Parameter	Min	Max	Units	Note
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.

TIMING DIAGRAMS

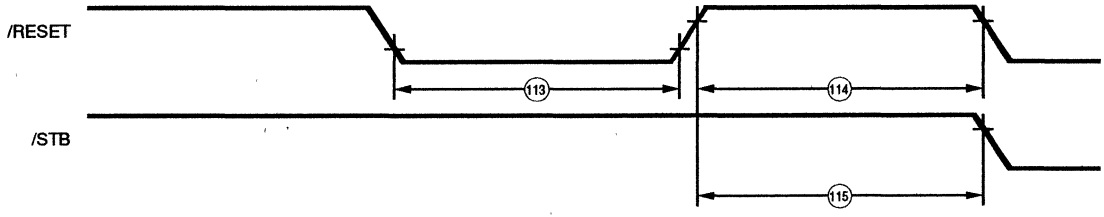
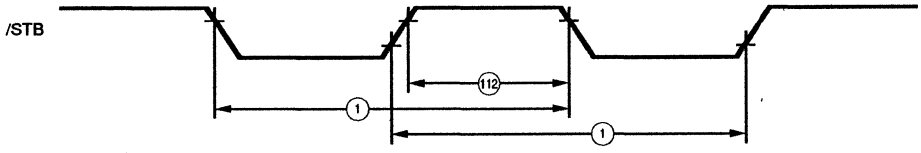


Figure 55. Reset Timing



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

Figure 56. Bus Cycle Timing

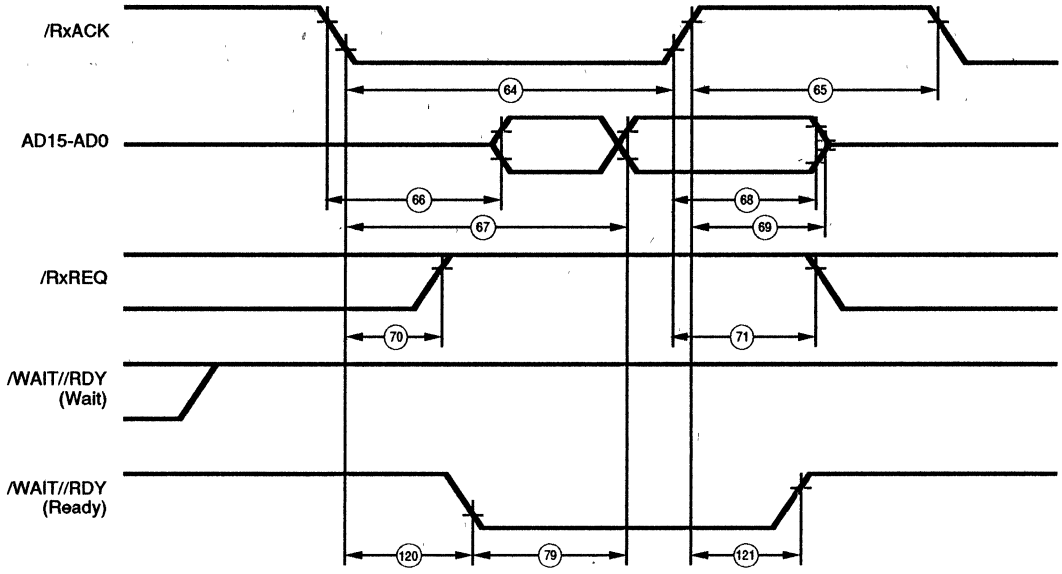


Figure 57. DMA Read Cycle

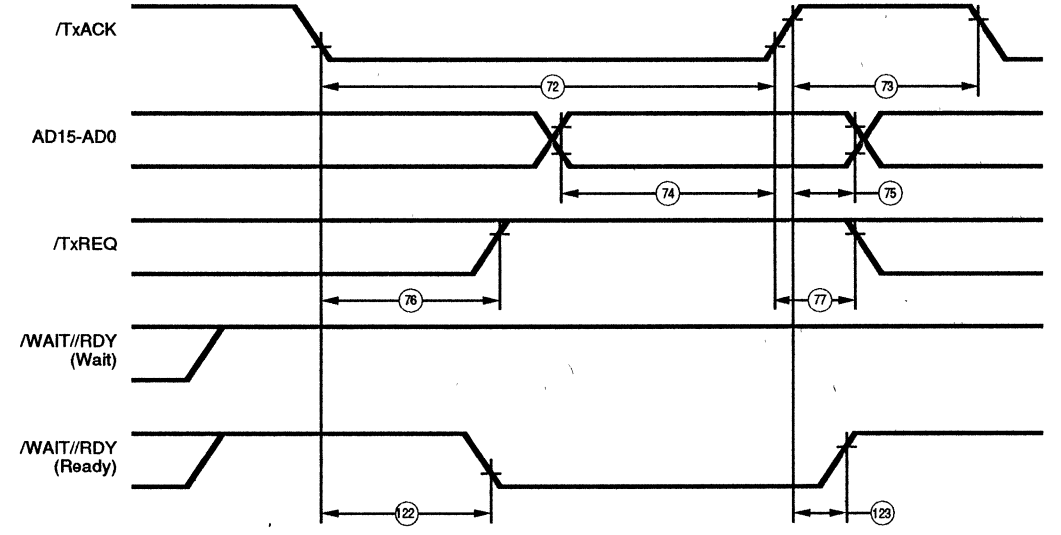


Figure 58. DMA Write Cycle

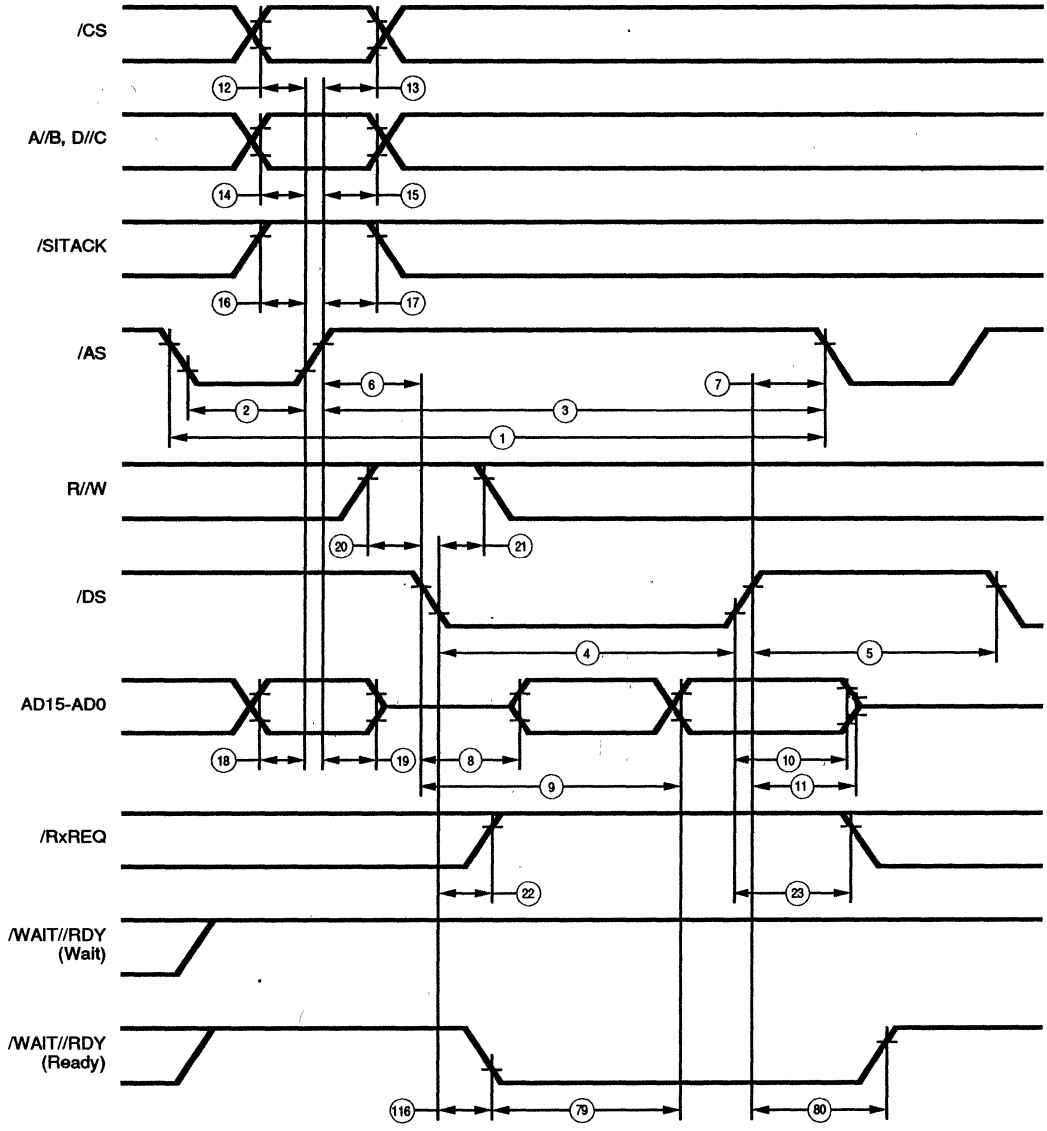


Figure 59. Multiplexed /DS Read Cycle

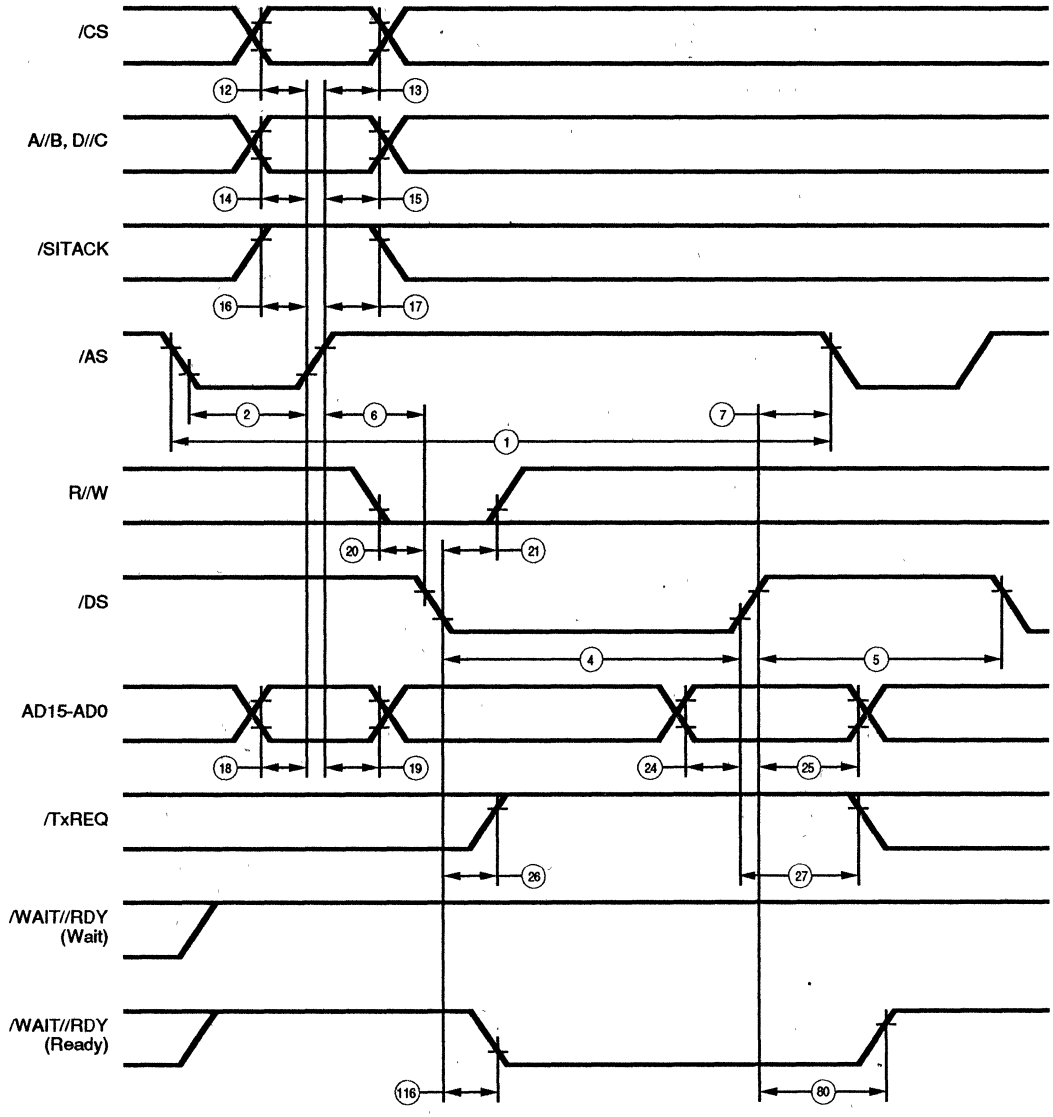


Figure 60. Multiplexed /DS Write Cycle

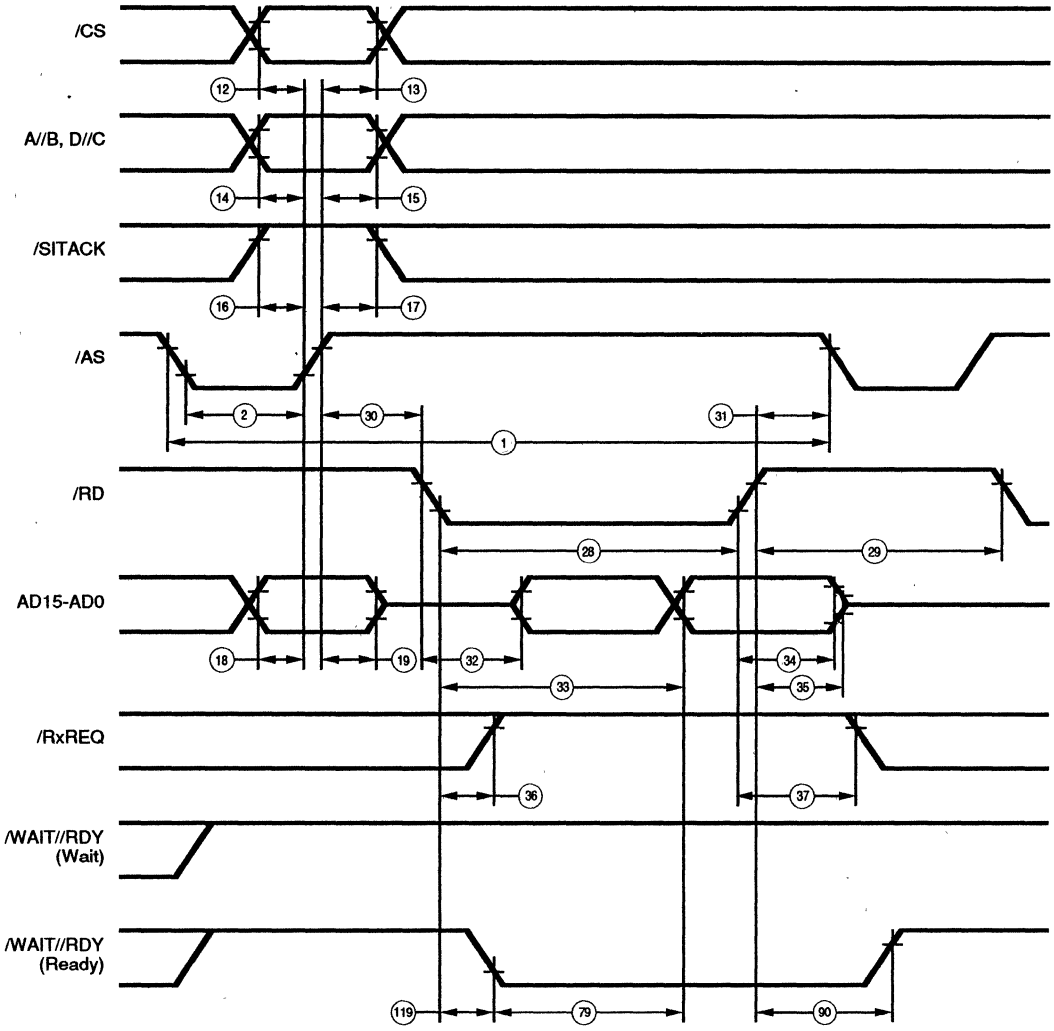


Figure 61. Multiplexed /RD Read Cycle

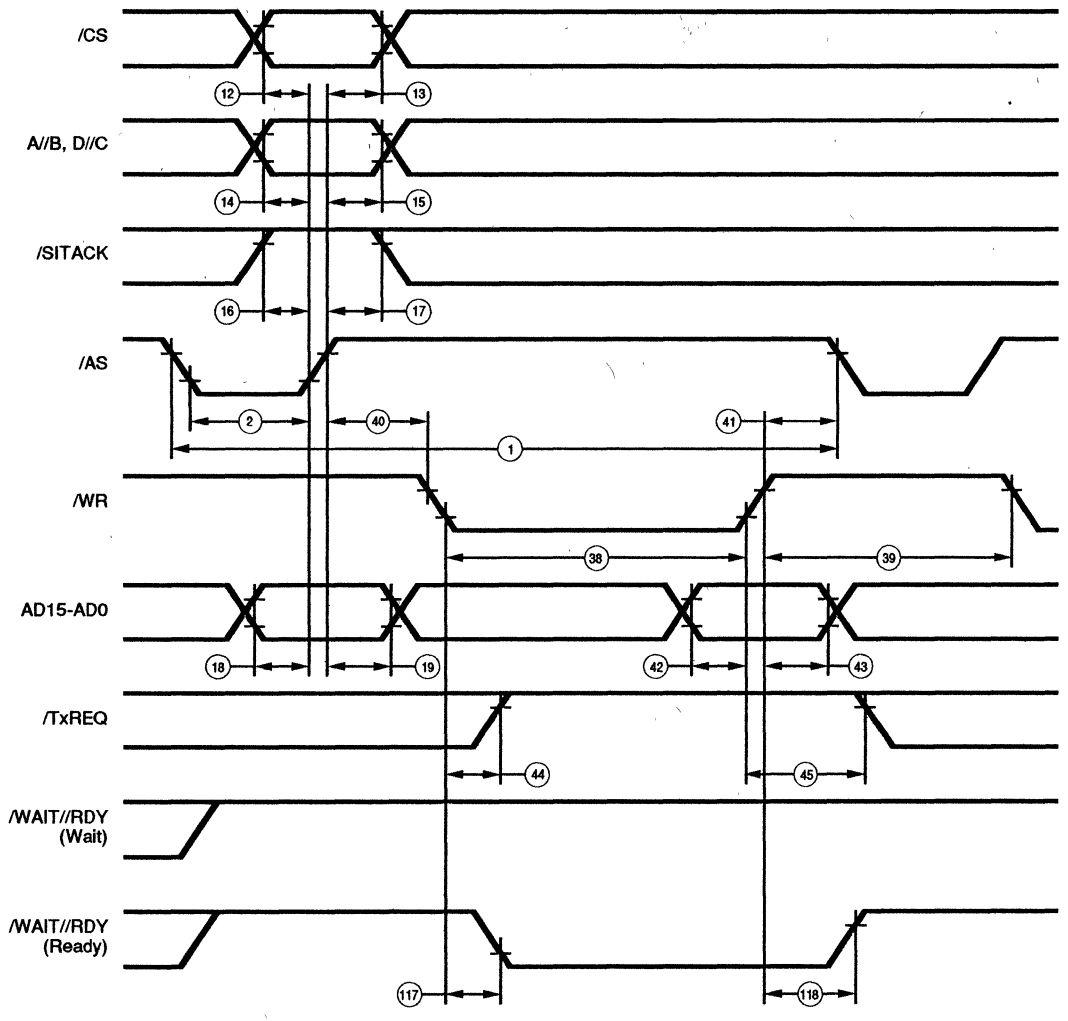


Figure 62. Multiplexed /WR Write Cycle

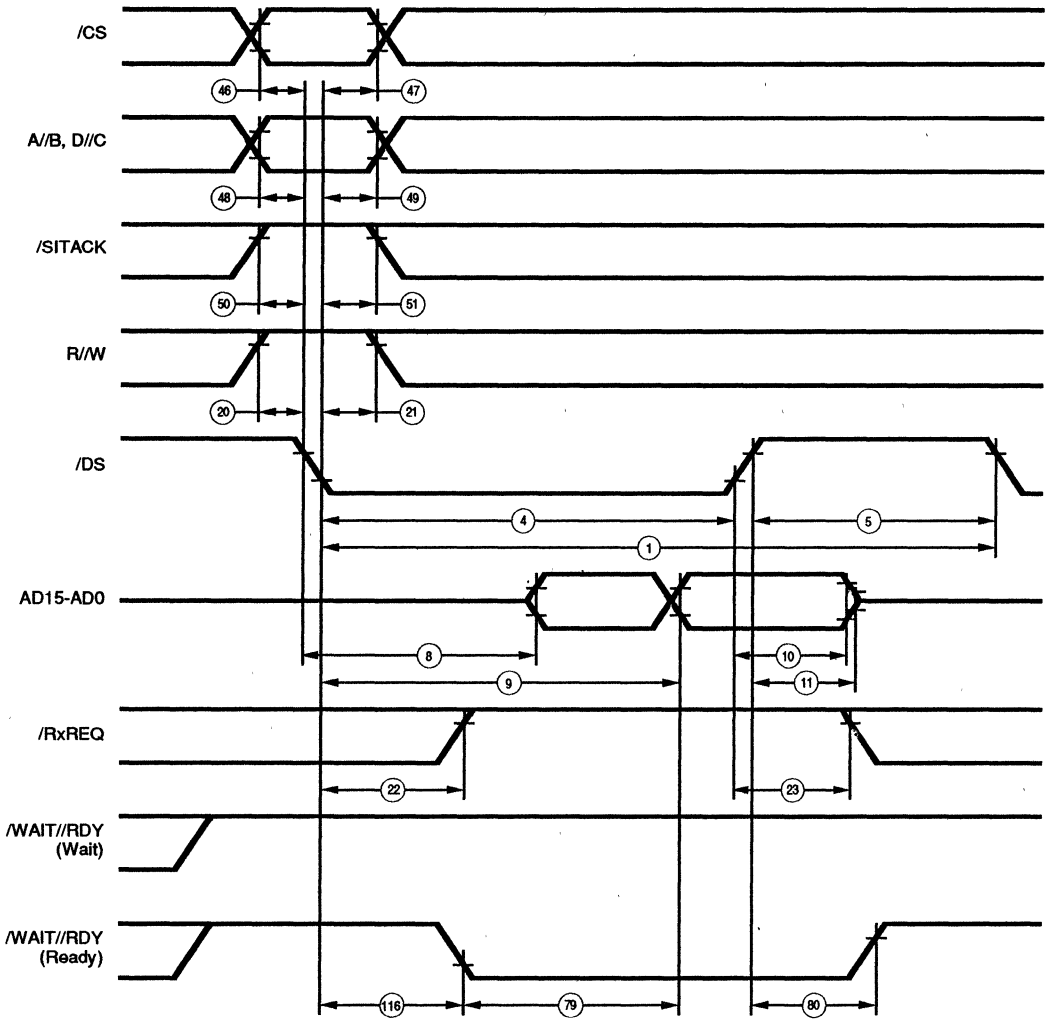


Figure 63. Non-Multiplexed /DS Read Cycle

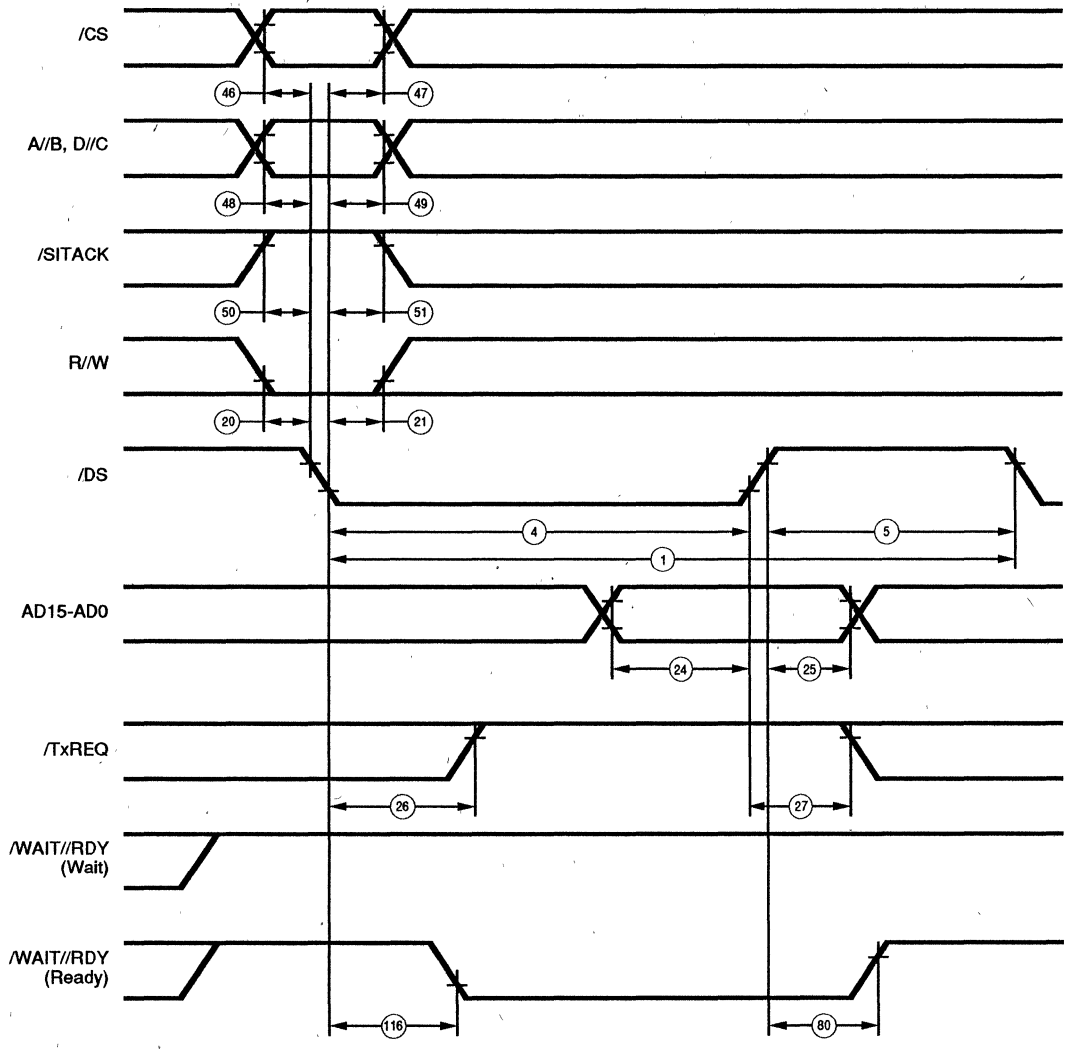


Figure 64. Non-Multiplexed /DS Write Cycle

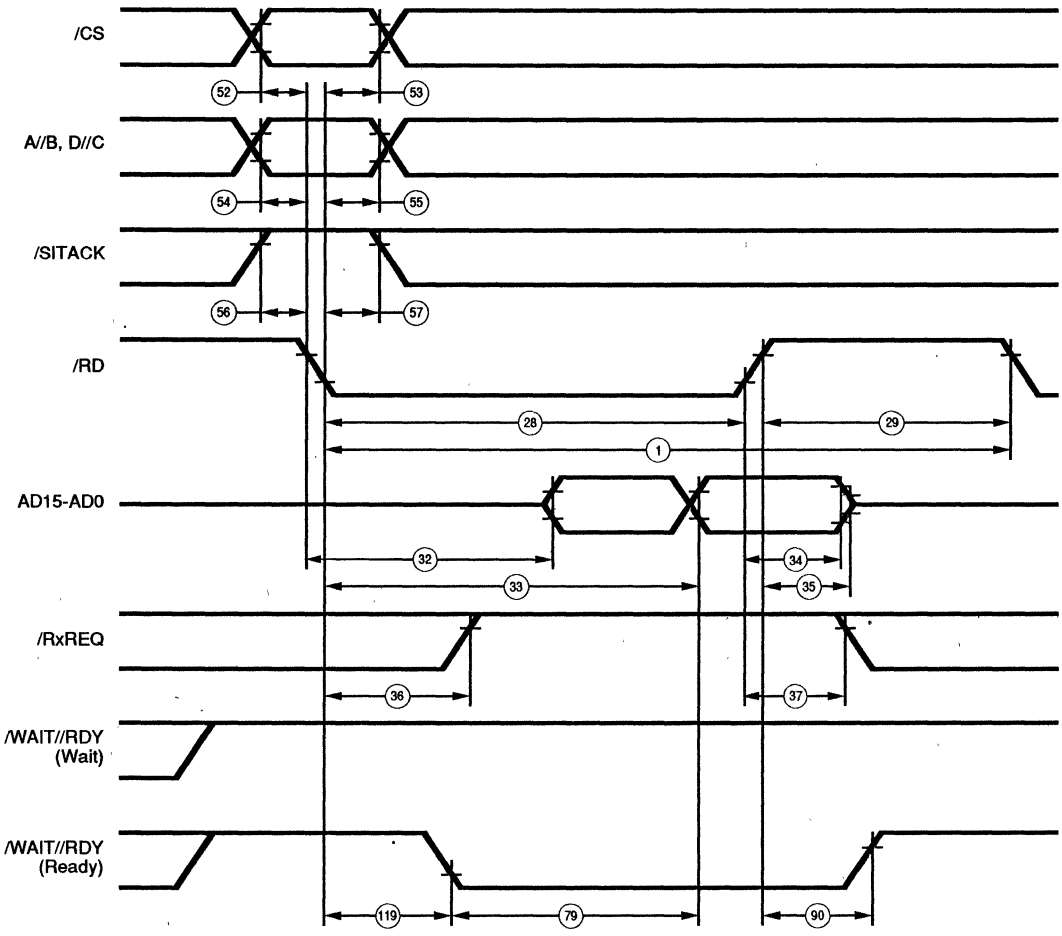


Figure 65. Non-Multiplexed /RD Read Cycle

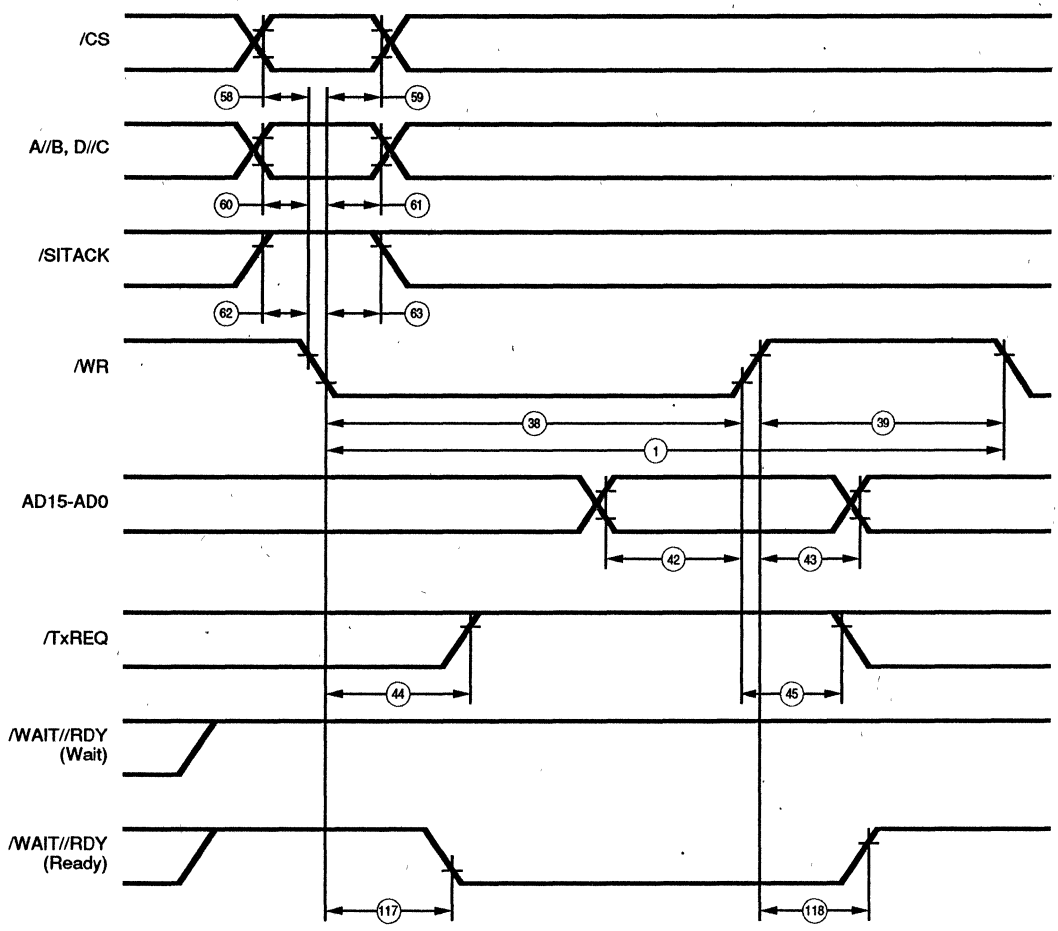


Figure 66. Non-Multiplexed /WR Write Cycle

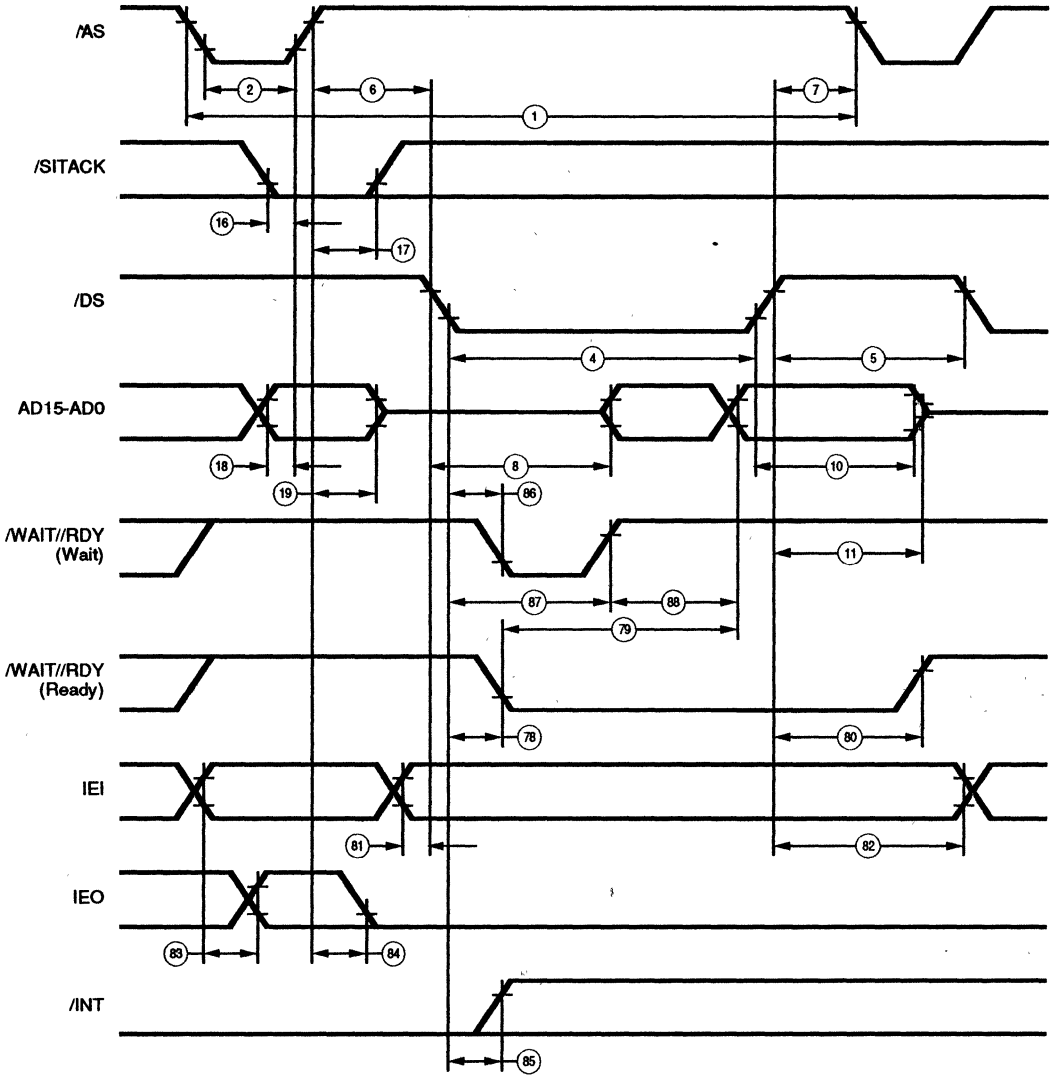


Figure 67. Multiplexed /DS Interrupt Acknowledged Cycle

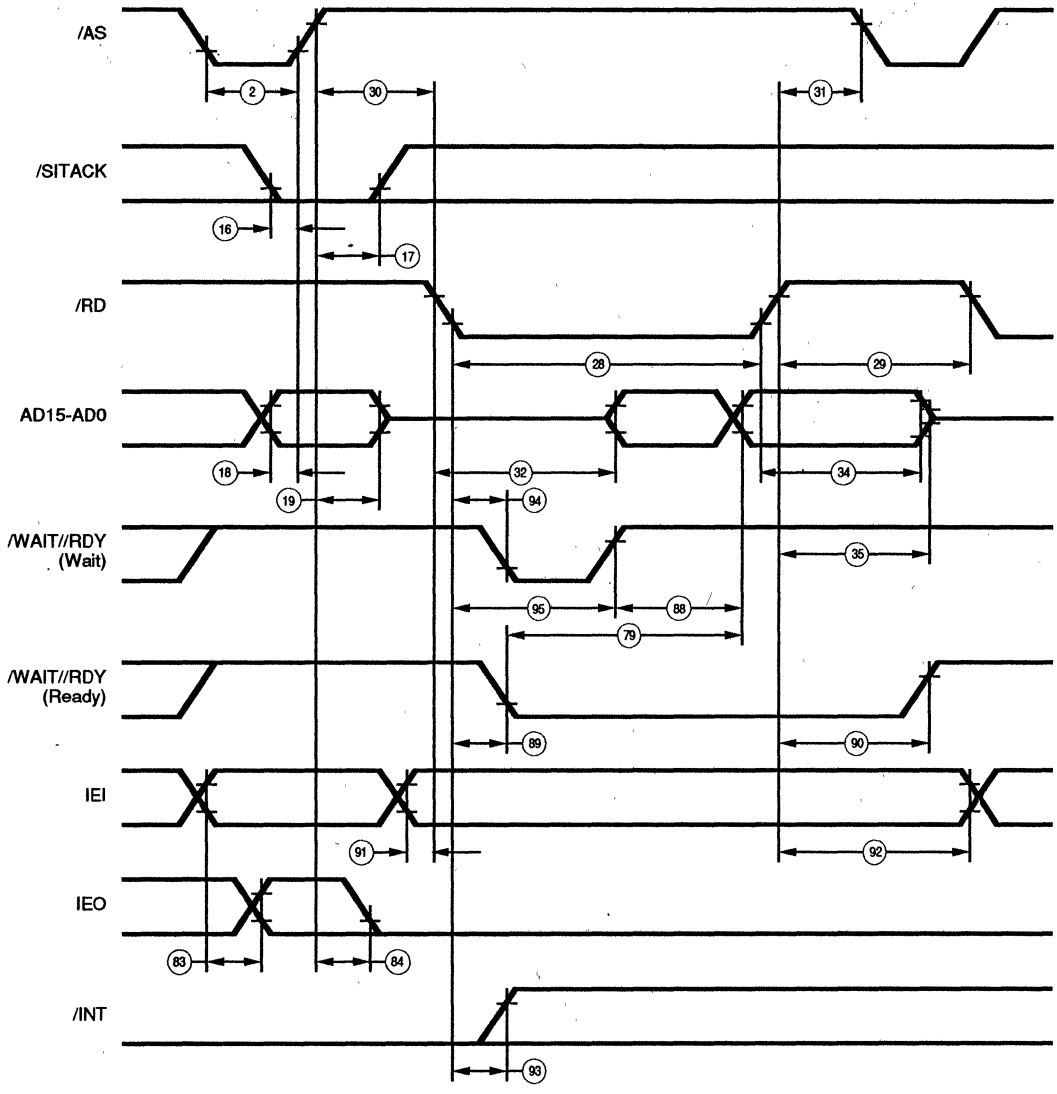


Figure 68. Multiplexed /RD Interrupt Acknowledge Cycle

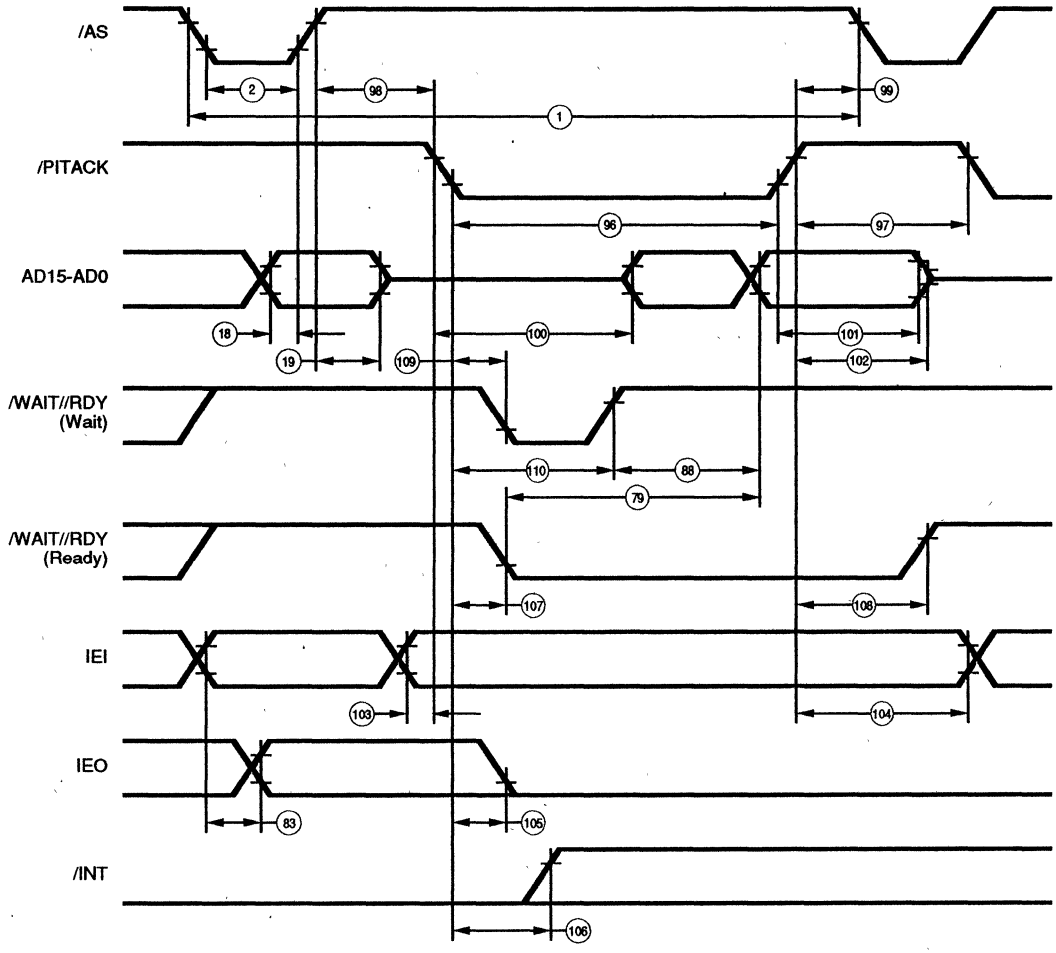


Figure 69. Multiplexed Pulsed Interrupt Acknowledge Cycle

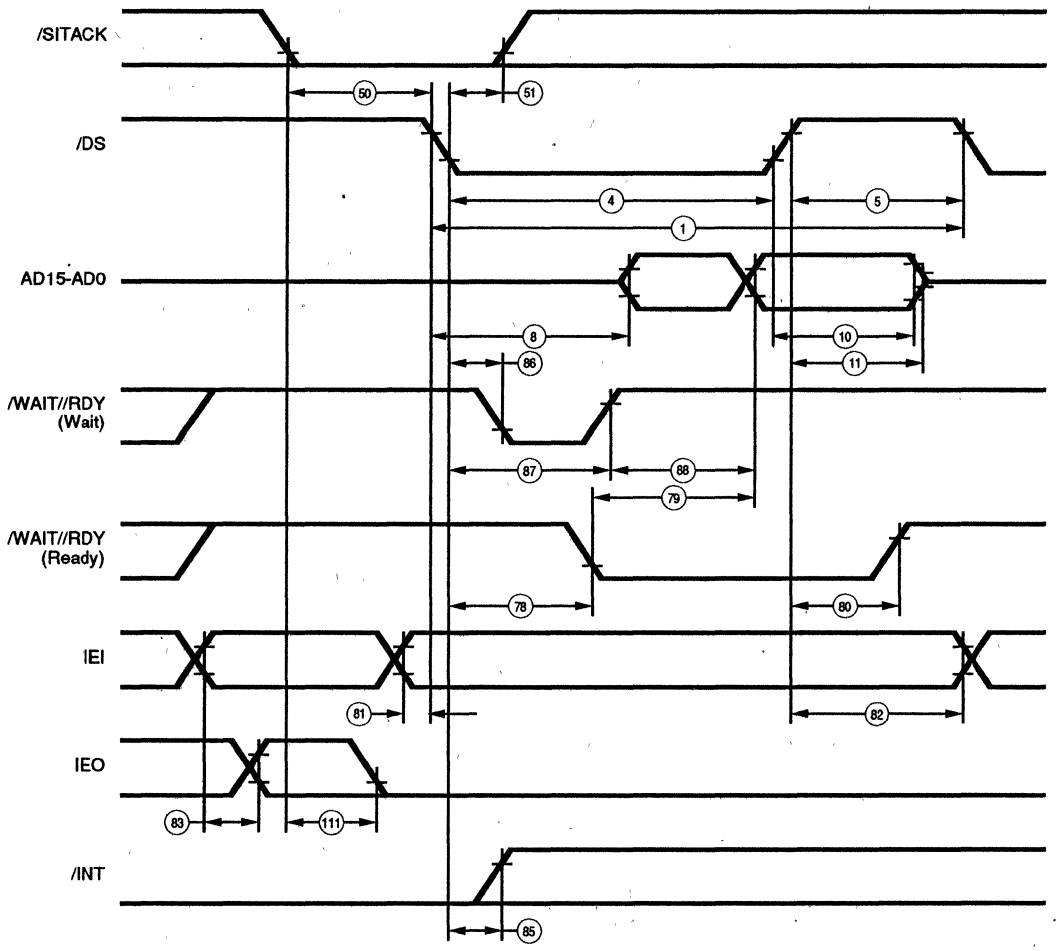


Figure 70. Non-Multiplexed /DS Interrupt Acknowledge Cycle

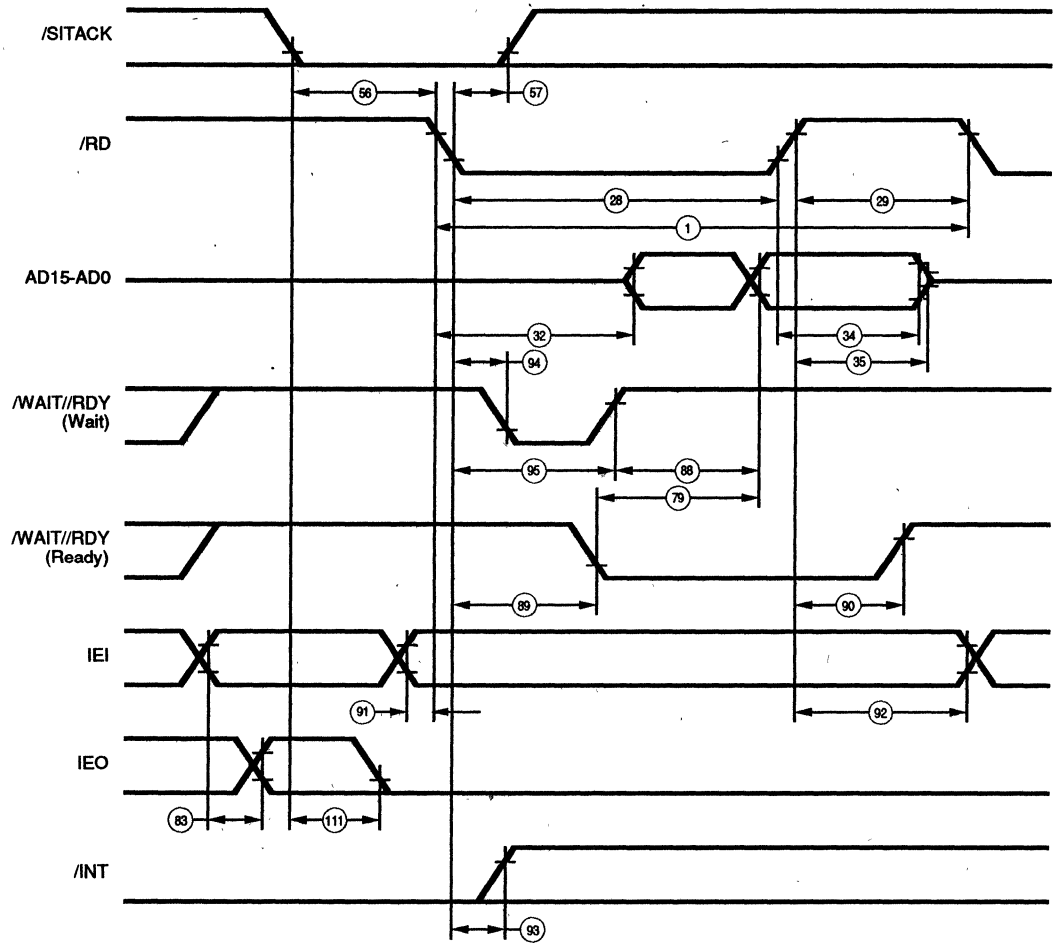


Figure 71. Non-Multiplexed Pulsed Interrupt Acknowledge Cycle

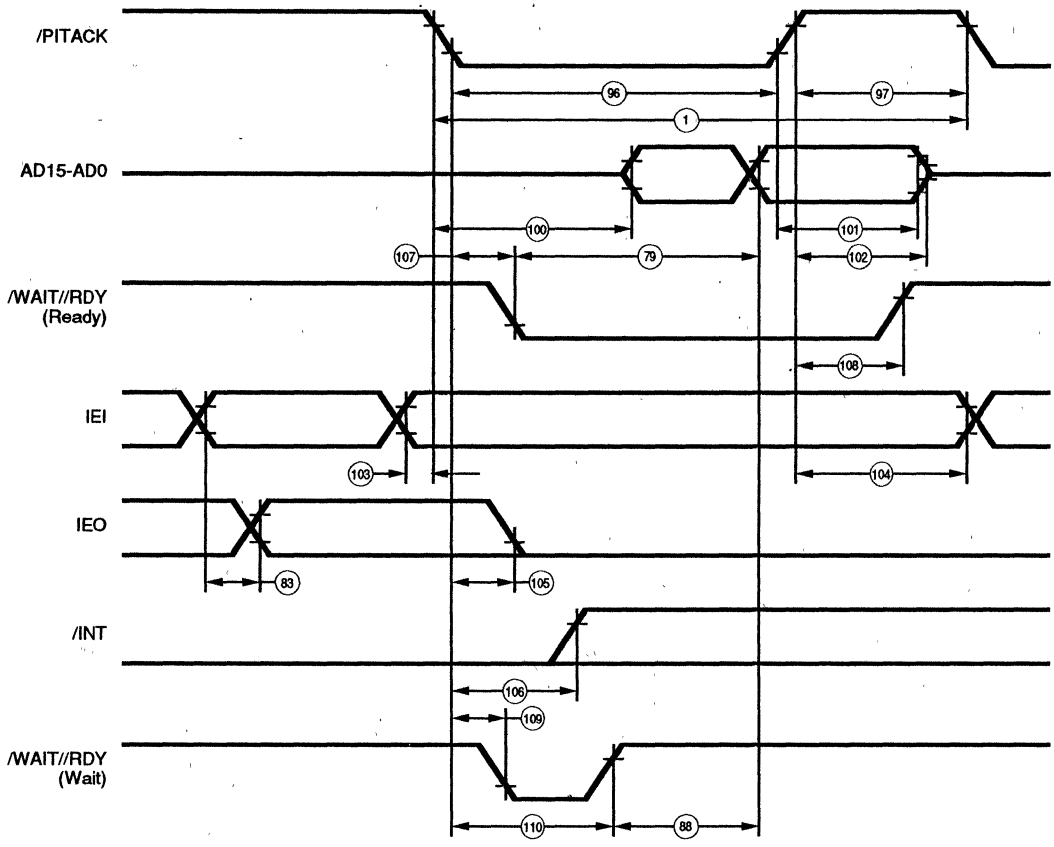


Figure 72. Non-Multiplexed /RD Interrupt Acknowledge Cycle

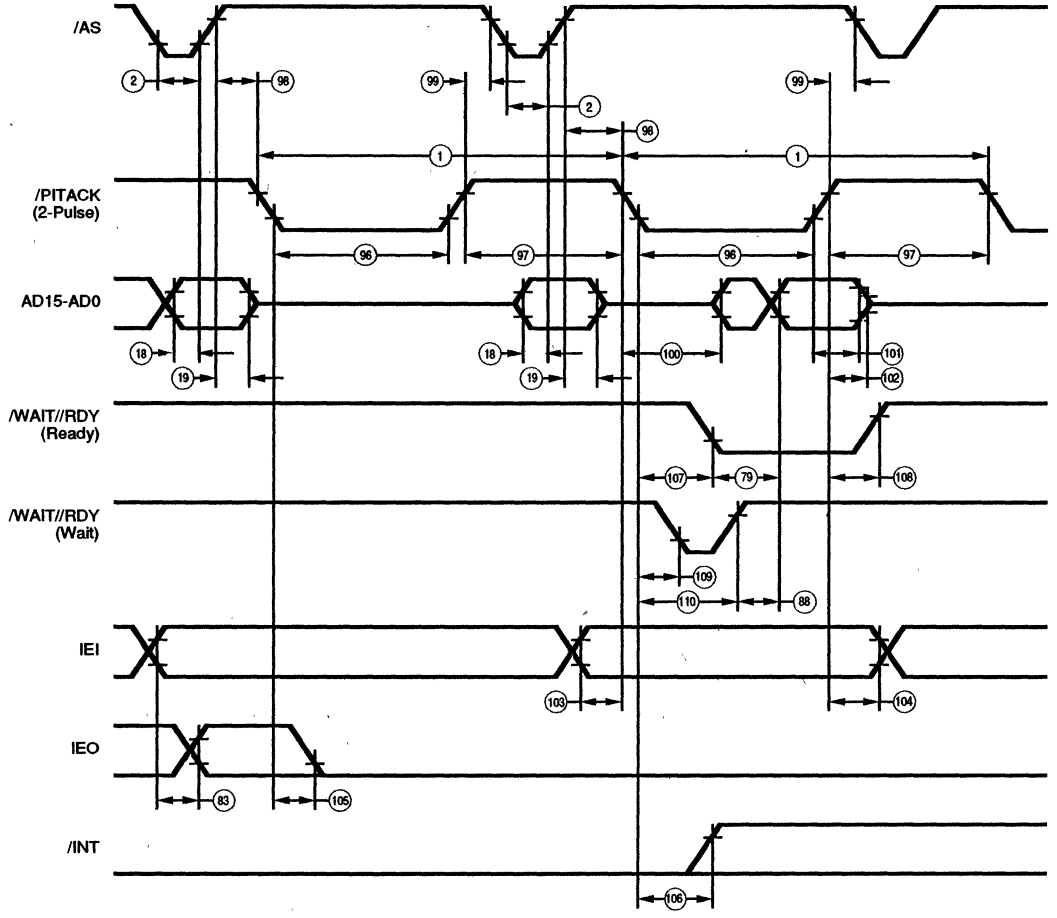


Figure 73. Multiplexed Double-Pulse Intack Cycle

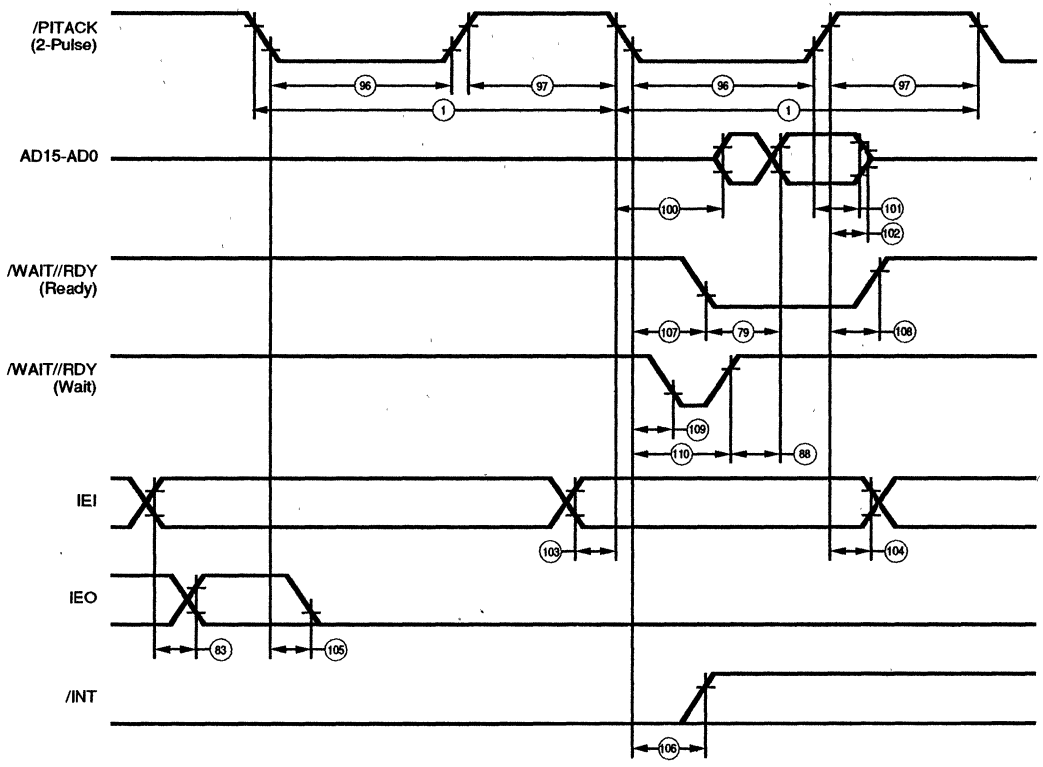


Figure 74. Non-Multiplexed Double-Pulse Intack Cycle

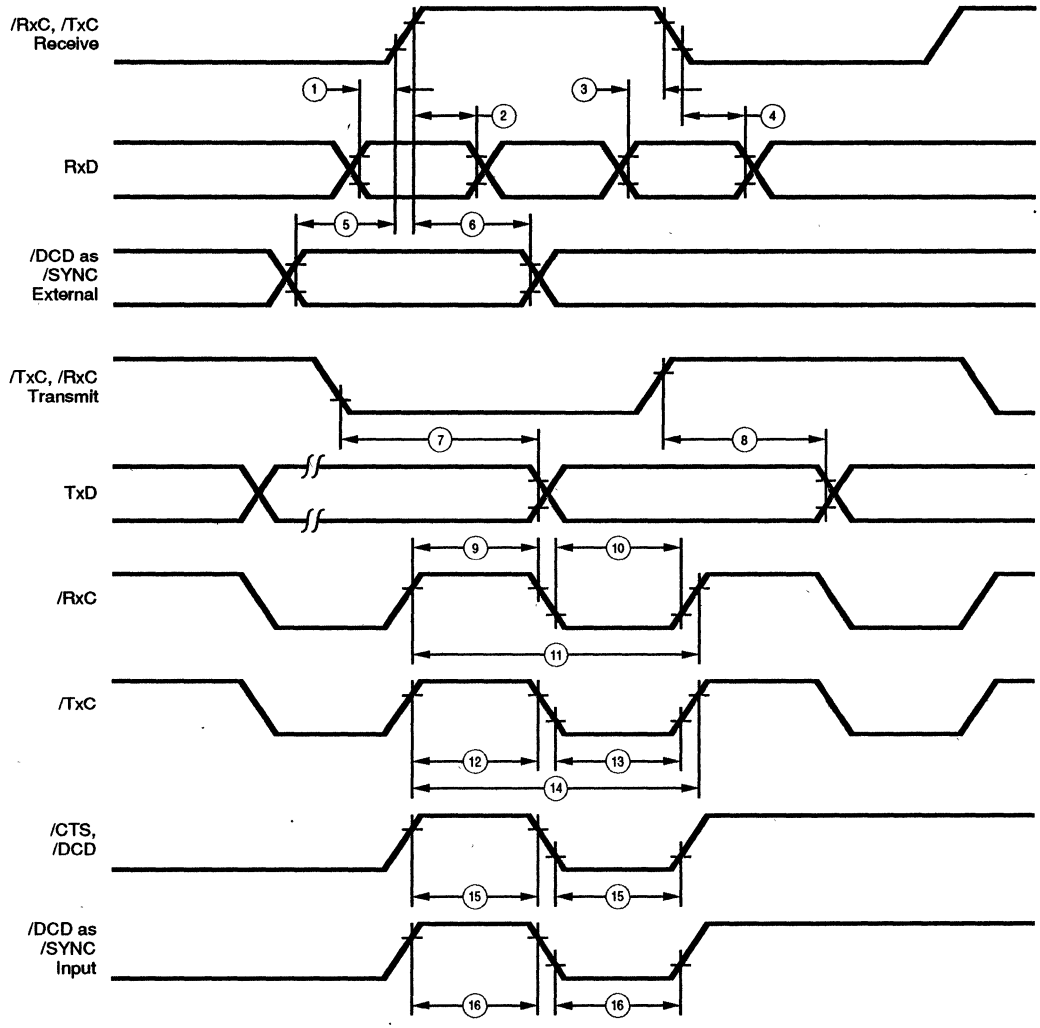


Figure 75. Z16C30 General Timing

AC CHARACTERISTICS

Z16C30 General Timing

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)	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	
10	TwRxCl	/RxC Low Width	40		ns	
11	TcRxC	/RxC Cycle Time	100		ns	
12	TwTxCh	/TxC High Width	40		ns	
13	TwTxCl	/TxC Low Width	40		ns	
14	TcTxC	/TxC Cycle Time	100		ns	
15	TwExT	/DCD or /CTS Pulse Width	70		ns	
16	TWSY	/DCD as /SYNC Input Pulse Width	70		ns	

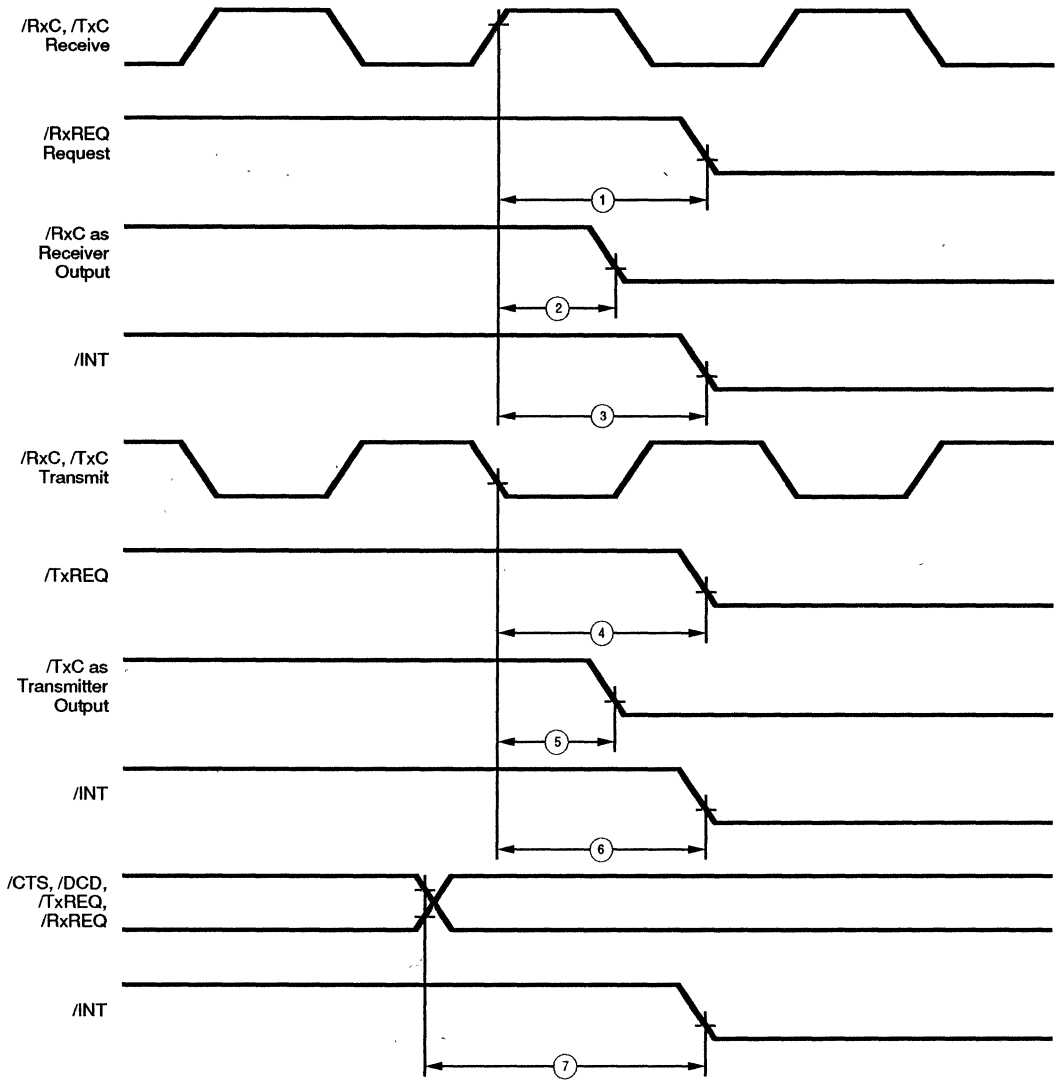


Figure 76. Z16C30 System Timing

AC CHARACTERISTICS

Z16C30 System Timing

No	Symbol	Parameter	Min	Max	Units	Note
1	TdRxC(REQ)	/RxC Rise to /RxREQ Valid Delay		100	ns	[2]
2	TdRxC(RxC)	/TxC Rise to /RxC as Receiver Output Valid Delay		100	ns	[2]
3	TdRxC(INT)	/RxC Rise to /INT Valid Delay		100	ns	[2]
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	
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.

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

ADVANCE INFORMATION
PRELIMINARY PRODUCT SPECIFICATION

Z16C31
IUSC INTEGRATED UNIVERSAL SERIAL CONTROLLER

FEATURES

- Full-duplex multiprotocol serial controller
- Two full-capability DMA channels
- Flexible adaptation to various system buses
- Serial data rates to 10M bits/second
- Serial modes include Asynchronous, Bisync, SDLC, HDLC, Ethernet, 1553B, and 9-Bit.
- Two baud rate generators
- Digital phase-locked loop for clock recovery
- Receive and Transmit Time Slot Assigners for ISDN applications.
- Eight general-purpose I/O lines plus Carrier Detect and Clear to Send.
- Transmit and receive frame-length counters, independent of the DMA facility.
- Async features include false-start filtering, stop bit length programmable by 1/16 bit steps, parity generation/checking, break generation/detection.
- HDLC/SDLC features include 8-bit address checking, extended address support, 16/32 bit CRC, programmable idle state, auto preamble option, loop mode.
- Sync features include 2- to 16-bit sync, sync-strip option, 16/32 bit CRC, programmable idle state, auto preamble option, X.21 transmitter/receiver slaving.
- Automatic control character recognition in Transparent Bisync mode.
- 32-byte transmit and receive FIFOs between serial controller and DMA channels.
- DMA modes include single block, buffered, array-chained, and link-chained.
- Programmable throttling of DMA bus occupancy
- 16/32 bit addressing, 8- or 16-bit data
- Flexible interrupt and bus-arbitration modes, interrupt and bus-acknowledge daisy-chains.
- High speed, low power CMOS technology
- 68-pin PLCC

GENERAL DESCRIPTION

The IUSC (Integrated Universal Serial Controller) is a single-channel multiprotocol data communications device with on-chip DMA designed for use with any conventional multiplexed or non-multiplexed bus The IUSC functions as a serial-to-parallel, parallel-to-serial converter/controller and is software configured to satisfy a wide variety of serial communications applications under DMA control. The device contains a variety of sophisticated internal functions including two baud rate generators, a digital phase-locked loop, character counters for both receive and transmit, 32-byte FIFOs for both the receiver and transmitter and a two-channel, 32-bit DMA controller.

The IUSC handles a wide variety of formats including asynchronous, synchronous byte-oriented (e.g. BISYNC), and synchronous bit-oriented formats such as SDLC and HDLC.

The IUSC can generate and check CRC in any synchronous mode and is programmed to check data integrity in various modes The IUSC also has facilities for modem controls. In applications where these controls are not needed, the modem controls may 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. There are no interrupts associated with the 8-bit port.

The on-chip DMA channels allow high-speed data transfers for both the receiver and the transmitter. The device supports automatic status transfer via DMA and allows device initialization under DMA control. Each DMA channel provides for 32-bit addresses and a 16-bit transfer length. The DMA channels may operate in any of four modes: normal, buffered, array-chained, or linked array-chained. The DMA bus mastership time may be limited, under program control, as to the absolute number of clock cycles, or the number of bus transactions, or both. This prevents the IUSC from hogging the bus.

Note: 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); /N//S (NORMAL and SYSTEM are both active Low).

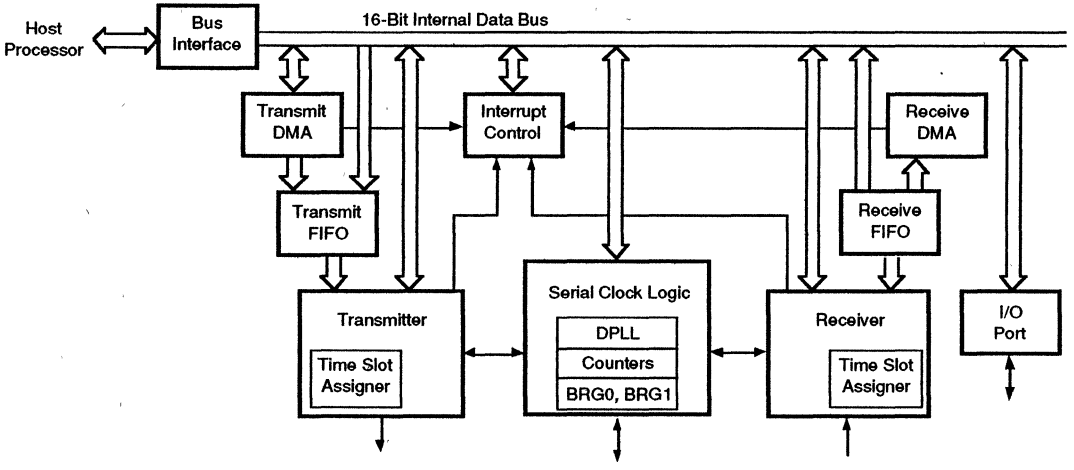


Figure 1. IUSC Block Diagram

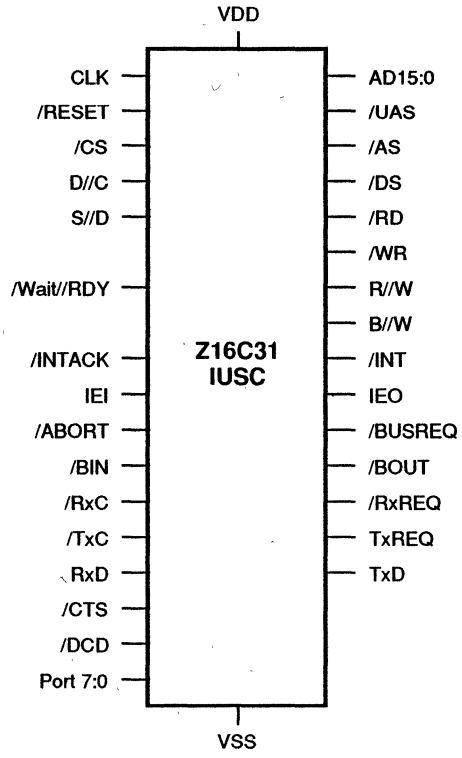
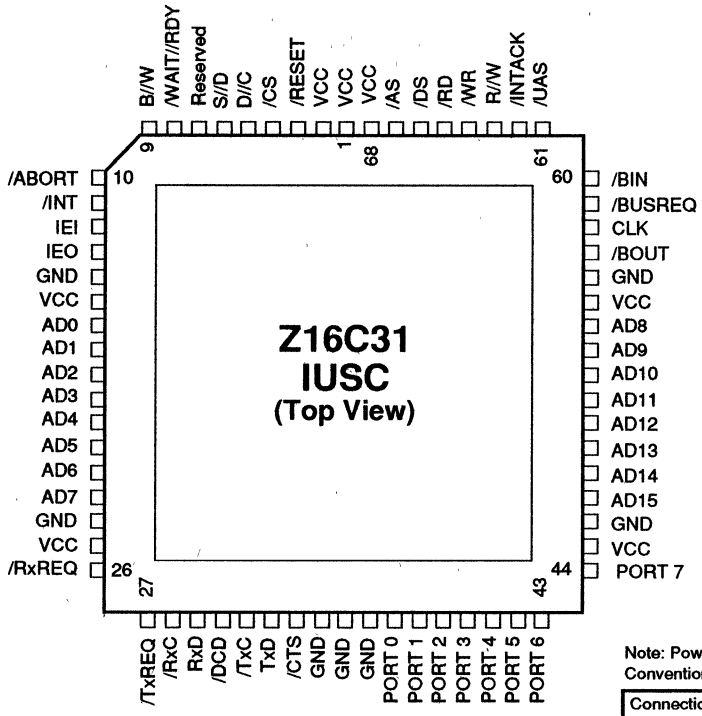


Figure 2. Logic Symbol



Note: Power connections follow Conventional descriptions below

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

Figure 3. Packaging

PIN DESCRIPTIONS

/RESET. *Reset* (input, active Low). A low on this line will place the IUSC in a known, inactive condition. /RESET should be driven low as soon as possible during power-up, and as needed thereafter 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 driven solely by the selected sources of receive and transmit clocking.)

AD15-AD0. *Address/Data Bus* (3-state input/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 MPU software has set up the IUSC 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. These lines are used in a variety of ways based on the value initially written to the Bus Configuration Register (BCR), as described in the text.

/CS. *Chip Select* (input, active Low). In slave mode, a low on this line indicates that the controlling microprocessor's current bus cycle is targeted for a register in the IUSC. /CS is ignored in master mode, and also when a low on /INTACK indicates that the current bus operation is an interrupt acknowledge cycle. On a multiplexed bus, /CS is latched by rising edges on /AS.

/S//D. *Serial/DMA* (input, High indicates serial). Slave cycles with /CS low and /INTACK and this pin both high, access registers in the serial controller section. Slave cycles with /INTACK high and /CS and this pin both low, access.registers in the DMA controller section.

The IUSC can be programmed so that when it is acting as a bus master, it drives this line low to indicate an array access cycle and high to indicate a normal cycle for the transmitter or receiver.

D//C. *Data/Control* (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 Data Register (FIFO). A slave write cycle with the same conditions writes data into the Transmit Data Register (FIFO). Slave cycles with both /INTACK and S//D high and both /CS and this pin low, access a serial controller register; on a multiplexed bus the particular register is selected by the low-order AD lines at the rising edge of /AS; on a non-multiplexed bus the particular register is selected by the LSB's of the serial controller's Channel Command/ Address Register.

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 receive DMA channel (low) and the transmit DMA channel (high). On a non-multiplexed bus with /INTACK high and /CS and S//D both low, the channel selection is taken from the DMA controller's address pointer register, and the state of this line does not affect the cycle.

The IUSC can be programmed so that when it is acting as a bus master, it drives this line low to indicate a DMA cycle for the receiver and high to indicate a cycle for the transmitter.

/AS. *Address Strobe* (input/output, active Low, 3-state). After a reset, the IUSC's bus interface logic monitors this signal to see if the host bus multiplexes address and data on AD15-AD0. If the logic sees activity on /AS before (or as part of) the initial write to the Bus Configuration Register (BCR), then in subsequent slave cycles, the IUSC captures register selection from the low-order 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, and drives /AS low to indicate the presence of the least significant 16 bits of an address on the AD15-AD0 lines. External latches are 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 4.7K ohms. 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 processor's output.

/UAS. *Upper Address Strobe* (3-state output, active Low). When the IUSC takes control of the bus and operates as a master, it drives /UAS low to indicate the presence of the more significant 16 bits of an address on AD15-AD0. External slaves or de-multiplexing latches should capture the MS address at each rising edge on this line.

R/W. *Read / Write control* (3-state input/output, Low signifies Write). This line is used in conjunction with the /DS line for host processors/buses having this kind of signalling, to indicate read and write cycles on the bus. 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. Otherwise, it is an input that is sampled at the leading/falling edge of /DS.

DS. *Data Strobe* (3-state input/output, active Low). This line is used in conjunction with the R/W line for host processors/buses having this kind of signalling, to indicate read and write cycles on the bus. 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. The R/W line remains valid throughout the low time of 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, data is valid on the AD lines after the specified access time and remains 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* (3-state input/output, active Low). This line is used in conjunction with the /WR line for host processors/buses having this kind of signalling, to indicate read and write cycles on the bus. 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, data is valid on the AD lines after the specified access time and remains valid until after the master releases this line to high.

/WR. *Write Strobe* (3-state input/output, active low). This line is used in conjunction with the /RD line for host processors/buses having this kind of signaling, to indicate read and write cycles on the bus. 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 slave write cycles, the IUSC captures data at the rising (trailing) edge of this line. For

PIN DESCRIPTIONS (Continued)

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* (3-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, a low indicates that 16 bits are to be transferred. For slave cycles on a non-multiplexed bus, the byte/word distinction is taken from bit 6 of the Command/Address register of the serial controller or DMA controller as applicable. For slave cycles on a multiplexed bus, the byte distinction is taken from an AD line at the rising edge of /AS.

/WAIT//RDY. *Wait, Ready, or acknowledge handshaking* (3-state input/output, active Low). This line is an input when the IUSC has taken control of the bus and is operating in master mode, otherwise, it is activated as an output when the IUSC detects a cycle with /CS or /INTACK low. In both directions, the way the line is used depends on the state of the S//D input that the IUSC captured during the initial BCR write. If S//D was high when the BCR was written, this line operates as a Wait/Ready line for Zilog and most Intel processors: the IUSC will not complete a master cycle while this line is low, and it may assert this line low until it is ready to complete a slave cycle.

If S//D was low when the BCR was written, this line operates as an Acknowledge line for Motorola and some Intel processors; the IUSC will not complete a master cycle until this line is low, and it asserts this line low when it is ready to complete a slave cycle.

/INT. *Interrupt Request* (output, active Low). This line is driven low by either the serial controller and/or the DMA controller sections, when one or more interrupt condition(s) is(are) enabled, pending, and not being serviced by the host processor. The IUSC can be programmed to drive this pin either totem-pole or open-drain.

/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 a potential device to be acknowledged. The initial write to the BCR includes selection of whether this line carries a level-sensitive "status" protocol, or a single-pulse or double-pulse protocol. Depending on this programming and the state of the /INT and IEI lines, the IUSC will respond in a variety of ways.

IEI. *Interrupt Enable In* (input, active High). This signal can be used with IEO to form an interrupt-acknowledge daisy-chain with other devices that may request interrupts. If IEI is high outside of an interrupt acknowledge cycle, and one or more IUSC interrupt condition(s) is(are) enabled, pending, and not being serviced by the host processor, then the IUSC will request an interrupt by driving /INT low. If IEI is high during an interrupt acknowledge cycle, and one or more IUSC interrupt condition(s) is(are) enabled, pending, and not being serviced by the host processor, then the IUSC will keep the IEO line low and responds to the cycle.

IEO. *Interrupt Enable Out* (output, active High). This signal can be used with IEI to form an interrupt-acknowledge daisy-chain with other devices that may request interrupts. IEO is low whenever IEI is low, and/or whenever an IUSC interrupt is under service. In addition, during an interrupt acknowledge cycle, IEO is forced low if the IUSC is (has been) requesting an interrupt.

/BUSREQ. *Bus Request* (output, active Low). This signal is used by the DMA controller section to request control of the host bus. It is selected as an open-drain or totem-pole output in the initial write to the BCR. If this line is used as open-drain, the IUSC samples the pin as an input and only drives it low after it has sampled 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 to 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 signal is ignored.

/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). The IUSC monitors this input in master mode. A low on this line indicates that the DMA channel terminates its activity and enter a disabled state at the end of the current cycle. Note that · 1) /ABORT is only effective during a DMA cycle, so that the IUSC knows which channel should be "aborted", and 2) external logic sets /WAIT//RDY to the right state for the cycle to complete, before /ABORT becomes effective.

RxD. *Received Data* (input, positive logic). The serial input data to the serial controller section.

TxD. *Transmit Data* (output, positive logic). The serial output data from the serial controller section.

RxC. *Receive Clock* (input or output). As an input, this signal is used as a clock signal for any of the functional blocks within the serial controller section. Or, the IUSC can be programmed so that this pin is an output carrying any of several receiver or internal clock signals.

TxC. *Transmit Clock* (input or output). As an input, this signal is used as a clock signal for any of the functional blocks within the serial controller section. Or, the IUSC can be programmed so that this pin is an output carrying any of several transmitter or internal clock signals.

/RxREQ. *Receive 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 carries the low-active request from the receiver FIFO, which is normally handled by the receive DMA channel. More typically, it is used as a general-purpose input or output.

/TxREQ. *Transmit 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 the low-active request from the transmit FIFO, which is normally handled by the transmit DMA channel. More typically, it is used as a general-purpose input or output.

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

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

PORT0-7. *Input/output Port* (inputs and/or outputs). The IUSC can be programmed so that these lines are inputs and/or outputs in any mixture. These lines are used for additional status and/or control signals for a modem, or for any other purpose. Transitions on input lines are captured by internal latches.

Vcc, Vss. *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.

FUNCTIONAL DESCRIPTION

The basic structure of the IUSC is shown in Figure 1. 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 word written by the host processor after a Reset; thereafter, it is no longer accessible to the host software.

The Transmit DMA channel has the task of fetching data to be transmitted from output buffers in memory on the host processor's bus, and delivering this data to the IUSC's Transmit First-In, First-Out (FIFO) memory. The host software can set up the Transmit DMA channel to operate in any of four major modes. In single-block mode, the channel transfers one block of consecutive bytes with a programmable location and length from host memory to the Transmit FIFO. It then notifies the host processor and stops. The processor has to reprogram the channel before it can transfer another block, but in many applications this is satisfactory because the Transmit FIFO is 32 bytes deep. In ping-pong mode, there are two sets of buffer address and length registers: the processor can be programming one set while the DMA channel is using the other set. When the channel finishes transferring one block, it notifies the host processor and automatically switches to transferring the block described by the other register set. In array-chained mode, the host processor programs the Transmit DMA channel with the address of a table containing the addresses and lengths of the actual memory buffers. When the channel finishes transferring the data from one memory buffer to the Transmit FIFO, it automatically fetches the next buffer address and length from the table and begins to transfer the data in that buffer. Finally, in link-chained mode the host programs the channel with the address of the start of a linked list of buffer addresses and lengths, wherein each entry also includes the address of the next entry. Channel operation is similar to array-chained mode but includes the extra steps of fetching the link addresses.

At any point in time, the Transmit FIFO can be empty or can contain from 1 to 32 characters to be transmitted. Characters written into the FIFO automatically migrate to its other end, where they become available to the Transmitter.

FUNCTIONAL DESCRIPTION (Continued)

While the host processor can itself write characters into the Transmit FIFO, the best use of the IUSC is to use the Transmit DMA channel to do so. The host can program the IUSC so that the Transmit DMA channel swings into operation at varying degrees of FIFO emptiness. Selecting this point involves balancing the probability and consequences of under running the transmitter against the overhead for the DMA channel to repeatedly acquire the host bus.

The serial Transmitter takes characters from the Transmit FIFO and converts them to serial data on the TxD pin. While this function is conceptually simple, the IUSC supports a good number of complex serial protocols, which increases the complexity of the Transmitter dramatically. For example, depending on the serial mode selected, the Transmitter may do any of the following in addition to parallel-serial conversion: start, stop, and/or parity bit generation, CRC calculation and transmission, automatic generation of opening and closing flags, encoding the serial data into any of several formats that guarantee transitions and carry clocking with the data, and/or hardware flow control based on the CTS input pin.

Finally, for ISDN and other time-multiplexed applications, the Transmitter section includes Time Slot Assigner logic that can be programmed to activate the Transmitter only periodically and for certain bits within a multiple-sourced, cyclically time-multiplexed data stream.

In general, the functions of the Receiver section are the inverse to those of the Transmitter. The receiver monitors the serial data on the TxD pin, recognizes its organization according to the programmed serial mode, and converts the data to parallel characters which it puts into the Receive FIFO. Once again, there is much more going on than just serial-parallel conversion - depending on the serial mode the Receiver may have to: start bit detection and synchronization, parity and stop bit checking, CRC calculation and checking, detection of flag, abort and idle sequences, control character and transparency recognition, decoding of the serial data and clock extraction from any of several serial-encoding schemes, and/or disabling/squelching based on the DCD input pin Based on such checking, the Receiver generates several status bits associated with each character, and writes them into the Receive FIFO along with the character.

The Receiver section also includes an optional Time Slot Assigner which is used to activate the rest of the Receiver for certain bits within a multiple-destination, cyclically time-multiplexed data stream such as an ISDN link.

The Receive FIFO can hold up to 32 characters and their associated status bits. As entries are written in by the receiver, they automatically migrate to the output side where they become available to either the host processor or the Receive DMA channel. As for the Transmit FIFO, there is detection logic for various degrees of fullness; it controls when the Receive DMA channel is set into operation and/or when the host processor is interrupted. In addition to the main Receive FIFO there is a 4-entry Frame Status FIFO that is used to hold status related to entire frames rather than to individual characters.

While the host processor can access data directly from the Receive FIFO, the IUSC is used to best advantage when the Receive DMA section is programmed to transfer the received data into buffer areas in memory on the host processor's bus. As described for the transmit side, the Receive DMA channel can be programmed to operate in single-buffer mode, ping-pong mode, array-chained mode, or link-chained mode.

The Serial Clocking Logic section makes up the clocking signals for both the Transmitter and Receiver It can be programmed to do this based on two internal Baud Rate Generators or on external clocks. It includes a Digital Phase-Locked Loop that can recover clocking from an encoded serial stream on RxD.

The Interrupt Control section gathers the various request lines from the Transmitter, Receiver, and the DMA channels, and handles the details of requesting host interrupts and responding to host interrupt-acknowledge cycles or to software equivalents. Interrupt operation is affected by both the initial write to the Bus Configuration Register (BCR) and by several registers in the Receiver, Transmitter, and DMA Channels.

In addition to the Clear to Send (CTS) and Carrier Detect (DCD) inputs, which are handled by the Transmitter and Receiver, respectively, the I/O port section provides eight pins that can be used for additional modem control lines or any other purpose. Each pin can be individually controlled as an input or output, and some of them have optional dedicated functions.

PRODUCT SPECIFICATION

Z16C33

CMOS MUSC
MONO-UNIVERSAL SERIAL CONTROLLER

FEATURES

- 0 to 10 Mbit/sec, full-duplex channel, with two baud rate generators and a digital phase-locked loop for clock recovery.
- 32-byte data FIFO's for receiver and transmitter
- 12.5 MByte/sec (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.
- 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.
- 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 the 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.
- ISDN time slot assigner
- 8-bit general purpose port with transition detection
- Low power CMOS
- 68-pin PLCC package

GENERAL DESCRIPTION

The MUSC (Mono - Universal Serial Controller) is a single-channel multi-protocol data communications peripheral designed for use with any conventional multiplexed or non-multiplexed bus. The MUSC functions as a serial-to-parallel, parallel-to-serial converter/controller and is 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, a digital phase-locked loop, character counters for both receive and transmit, and 32-byte data FIFO's for both the receiver and transmitter.

The MUSC handles asynchronous formats, synchronous byte-oriented formats (e.g. BISYNC), and synchronous bit-oriented formats like 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 is programmed to check data integrity in various modes. The MUSC also has facilities for modem controls. In applications where these controls are not needed, the modem controls are used for general-purpose I/O. The same is true for most of the other pins.

Interrupts are supported by a daisy-chain hierarchy with the serial channel. There are no interrupts associated with the 8-bit Port.

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

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

Note: 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); /N//S (NORMAL and SYSTEM are both active Low).

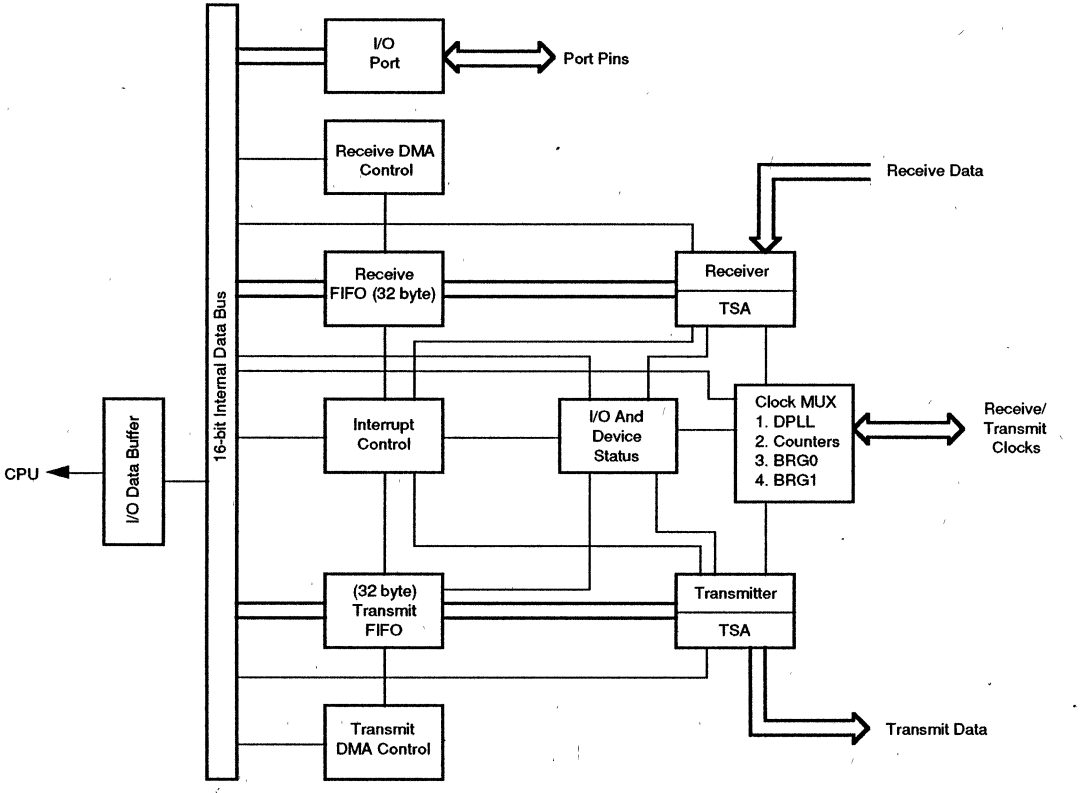


Figure 1. MUSC Block Diagram

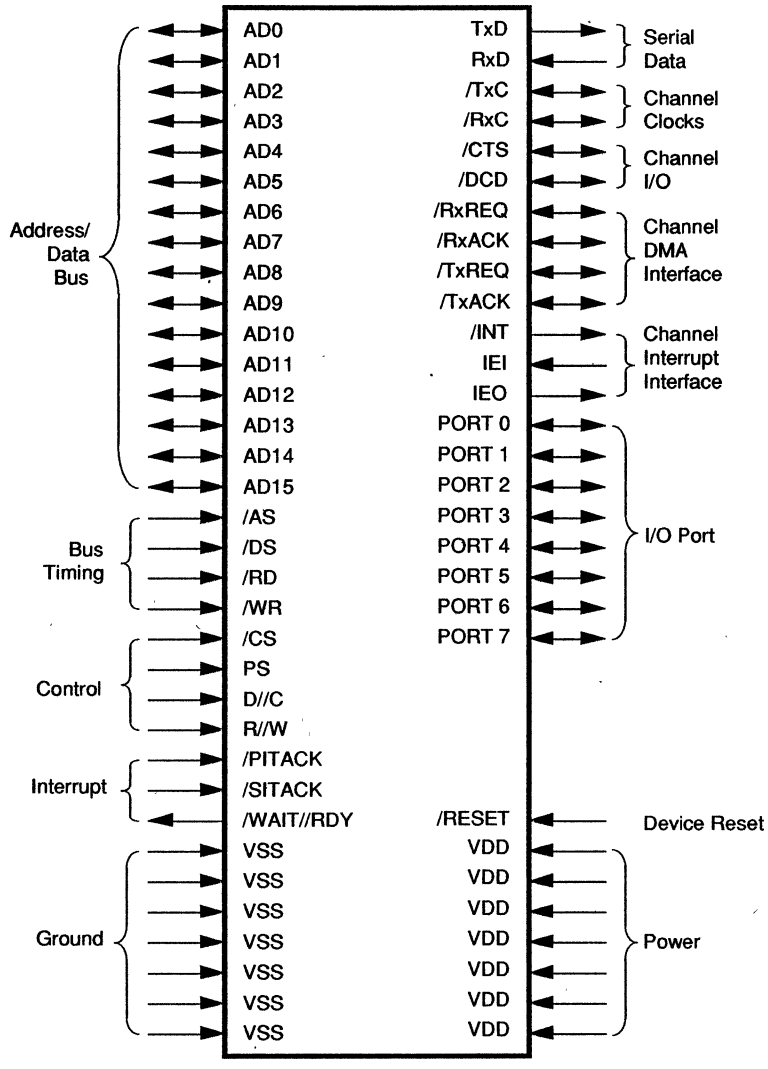
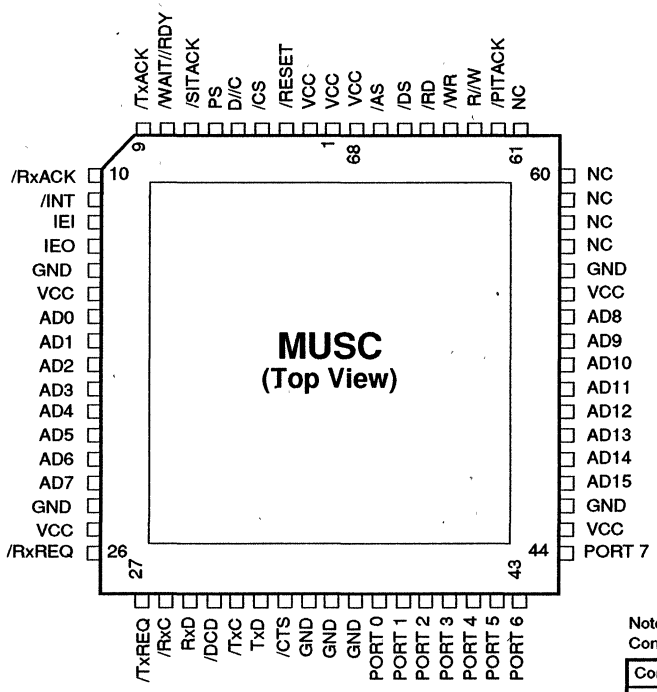


Figure 2. Pin Functions

Note: Power connections follow conventional descriptions below:

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



Note: Power connections follow Conventional descriptions below

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

Figure 3. Pin Assignments

PIN DESCRIPTION

The device contains 13 pins for channel I/O, 16 pins for address and data, 12 pins for CPU handshake, 8 pins for the I/O Port, and 14 pins for power and ground.

Three separate bus interface types are available for the device: The Bus Configuration Register (BCR), the external connection of the Protocol Select (PS) pin, 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 VSS.

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 MUSC 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).

PIN ASSIGNMENTS

/RESET. *Reset* (input, active Low). This signal resets the device to a known state. The first write to the MUSC 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 is tied to VDD in these cases.

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

/RD. *Read Strobe* (input, active Low). This signal strobes data out of the device during a read and strobes 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 is 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.

PS. *Protocol Select* (input, active High). This input is sampled and stored during the BCR (Bus Configuration Register) write It selects the sense of the /WAIT//RDY signal appropriate for different bus interfaces. With PS High, /WAIT//RDY functions as a /WAIT signal and with PS Low, /WAIT//RDY functions as a /READY signal This selection applies to all bus transactions.

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 indicating that an interrupt acknowledge cycle is in progress. The device is capable of returning an interrupt vector that is encoded with the type of interrupt pending during this acknowledge cycle.

/PITACK. *Pulsed Interrupt Acknowledge* (input, active Low) This is a strobe signal indicating that an interrupt acknowledge cycle is in progress. The device is capable of returning an interrupt vector that is encoded with the type of interrupt pending during this acknowledge cycle. /PITACK is programmed to accept either single pulse or double pulse acknowledges. This programming is performed in the BCR. The double pulse acknowledge 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 is programmed to function either as a Wait signal or a Ready signal using the state of the PS pin during the BCR write. When PS is High during the BCR write, this signal functions as a wait output and supports the READY function of 8X86 family microprocessors. When PS is Low during the BCR write, it functions as a ready output and supports the DTACK function of 680X0 family microprocessors.

AD15-AD0. *Address/Data Bus* (bidirectional, active High, 3-state). The AD signals carry addresses to, and data to and from, the device. When the 16 bit non-multiplexed bus is selected, AD15-0 carries data to and from the device. Addresses are provided using a pointer within the device that is loaded with the desired register address. When the 8-bit non-multiplexed bus, without separate address, is selected only AD7-0 is used to transfer data. The pointer is used for addressing, and AD15-8 is unused. When the 8-bit non-multiplexed bus with separate address is selected AD7-0 is used to transfer data, while AD15-8 is used as an address bus. When the 16 bit multiplexed bus is selected, addresses are latched from AD7-0 and data transfers are sixteen bits wide. When the 8-bit multiplexed bus without separate address is selected only AD7-0 is used to transfer addresses and data, and AD15-8 is unused When the 8-bit multiplexed bus with separate address is selected only AD7-0 is used to transfer data, while AD15-8 is used as an address bus.

The non-multiplexed, 16-bit bus interface mode directly supports 680X0 family microprocessors and the multiplexed, 16-bit bus interface mode directly supports the 8X86 family microprocessors. The multiplexed, 8-bit bus interface mode without separate address supports the 8088 family microprocessors.

PIN ASSIGNMENTS (Continued)

/INT. *Interrupt Request* (output, active Low). Indicates that the channel has an interrupt condition pending and is requesting service. This output is NOT open-drain.

IEI. *Interrupt Enable In* (input, active High). The IEI signal 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.

IEO. *Interrupt Enable Out* (output, active High). The IEO signal 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.

/TxACK. *Transmit Acknowledge* (input or output, active Low). The primary function of this signal is to perform fly-by DMA transfers to the transmit FIFO. It also is used as bit input or output.

/RxACK. *Receive Acknowledge* (input or output, active Low). The primary function of this signal is to perform fly-by DMA transfers from the receive FIFO. It also is used as bit input or output.

TxD. *Transmit Data* (output, active High, 3-state). TxD carries the serial transmit data for the channel.

RxD. *Receive Data* (input, active High). RxD carries the serial receive data for the channel.

/TxC. *Transmit Clock* (input or output, active Low). This signal may be used as a clock input for any of the functional blocks within the device. It also is used as an output for various transmitter signals or internal clock signals.

/RxC. *Receive Clock* (input or output, active Low). This signal is used as a clock input for any of the functional blocks within the device. It also is used as an output for various receiver signals or internal clock signals.

/TxREQ. *Transmit Request* (input or output, active Low). The primary function of this signal is to request DMA transfers to the transmit FIFO. It also is used as a simple input or output.

/RxREQ. *Receive Request* (input or output, active Low). The primary function of this signal is to request DMA transfers from the receive FIFO. It also is used as a simple input or output.

/CTS. *Clear To Send* (input or output, active Low) /CTS is used as an enable for the transmitter It also is programmed to generate interrupt on either transition or used as a simple input or output.

/DCD. *Data Carrier Detect* (input or output, active Low). This signal is used as an enable for the receiver. It also is programmed to generate an interrupt on either transition or used as a simple input or output.

PORT7 - PORT0. *Port Signals* (inputs or outputs, active High). These pins are general purpose I/O pins. They are used as additional MODEM control lines or for other I/O functions. When used as inputs, the ports capture transitions on these pins.

ARCHITECTURE

The MUSC internal structure includes a full-duplex serial channel 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 bus interface is designed to provide easy interface to most microprocessors, whether they employ a multiplexed, non-multiplexed, 8-bit or 16-bit bus structure The channel is controlled by a set of thirty-two 16-bit registers, almost 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.

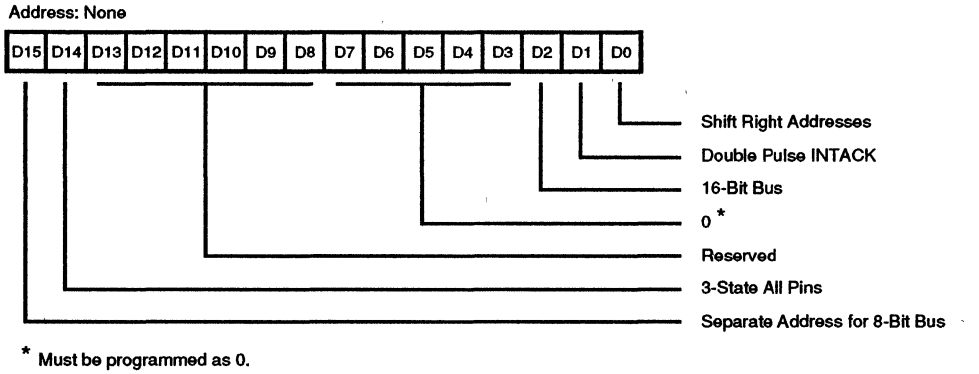


Figure 4. Bus Configuration Register

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 checks status and generates status interrupts as appropriate.

FUNCTIONAL DESCRIPTION

The functional capabilities of the MUSC 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 MUSC offers such features as read/write registers, a flexible bus interface, DMA interface support, vectored interrupts, and an eight-bit I/O port.

Data Communications Capabilities

The MUSC provides a full-duplex channel programmable for use in any common data communication protocol. The receiver and transmitter modes are completely independent. The receiver and transmitter are each 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 like receive FIFO load, received sync, transmit FIFO read and transmission complete are sent to pins for use by external circuitry.

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 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 sends one or two stop bits.

FUNCTIONAL DESCRIPTION (Continued)

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.

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 is 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 is programmed to send an abort, an extended abort, a flag or CRC and a flag on transmit underrun. The transmitter automatically sends 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 is 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 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.

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. In the transmitter, this bit is automatically inserted with the value that is FIFO'ed from 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 just 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 available only in the transmitter and allows the MUSC 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 breaks the loop and inserts its own frame(s).

Data Encoding

The MUSC 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.

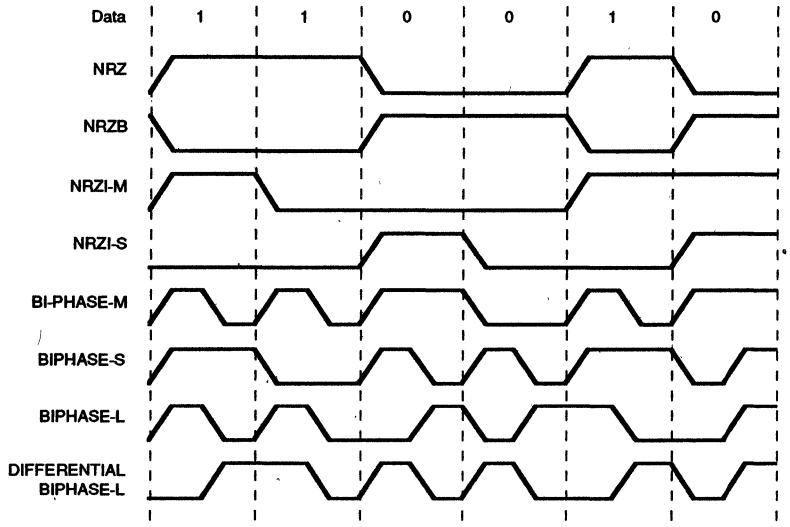


Figure 5. Data Encoding

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.

Character Counters

The MUSC contains a 16-bit character counter for both the 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

The MUSC 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, and the time constant is 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 is 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 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.

Digital Phase-Locked Loop

The MUSC 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 with the data stream, to construct a clock for the data. This clock is 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

The MUSC contains two 5-bit counters, which may be 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. They also provide a stable transmit clock from a common source when the DPLL is providing the receive clock.

Clock Multiplexer

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

Test Modes

The MUSC is 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 MUSC 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 MUSC.

I/O INTERFACE CAPABILITIES

The MUSC 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 MUSC 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. MUSC status bits are grouped according to function to simplify this software action.

Interrupt

When a MUSC 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 MUSC modifies three bits in this vector to indicate which type of interrupt is being requested.

Each of the six sources of interrupts in the MUSC (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 requests interrupts. Note that individual sources within the six groups also have interrupt enable bits which are separately enabled. There is a Master Interrupt Enable (MIE) bit which globally enables or disables interrupts within the serial channel.

The other two bits are related to the interrupt priority chain. The MUSC requests 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 device places a vector on the data bus.

In the MUSC, 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 the following priority. Receive Status, Receive Data, Transmit Status, Transmit Data, I/O Status and Device Status. There are six sources of Receive Status interrupt. Each one is individually enabled receiver exited hunt, received idle line, received break/abort, received 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 one is 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 individual selection and enables for each pin. The pins that are 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 count FIFO overflow, DPLL sync acquired, BRG1 zero count and BRGO zero count.

Block Transfer Mode

The MUSC accommodates block transfers via 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 responds 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 responds 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) When using the /RxACK and /TxACK pins to transfer data, no chip select is necessary; these are dedicated strobes for the appropriate FIFO.

Time Slot Assigner

The MUSC is equipped with two time slot assigners to support ISDN communications. There is one assigner for the receiver and one assigner for the transmitter and the assigners function independently. The time slot assigner selects one or more time slots within a frame, however, all 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 total number of concatenated slots is programmable from 1 to 15 (total slots).

The time of the slot is 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 is used to compensate for delays in frame sync detection logic.

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 for the three-state mode, output a logic 0, or output a logic 1 This programming is done in the Port Control Register. When programmed to be three-stated, the ports are used as inputs. Whether used as inputs or outputs, the port pins can be read at any time.

The port pins capture edge transition's input to the port. This 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 transitions 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

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

The registers in the MUSC must be programmed by the system to configure the channel. Before this can occur, 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 MUSC, 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 MUSC for the BCR write; the MUSC knows that the first write after a hardware reset is destined for the BCR.

In the multiplexed bus case, all registers are directly addressable via the address latched by /AS at the beginning of a bus transaction. The address may be 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 is shown in Table 1. The D//C pin is still used to directly access the receive and send 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

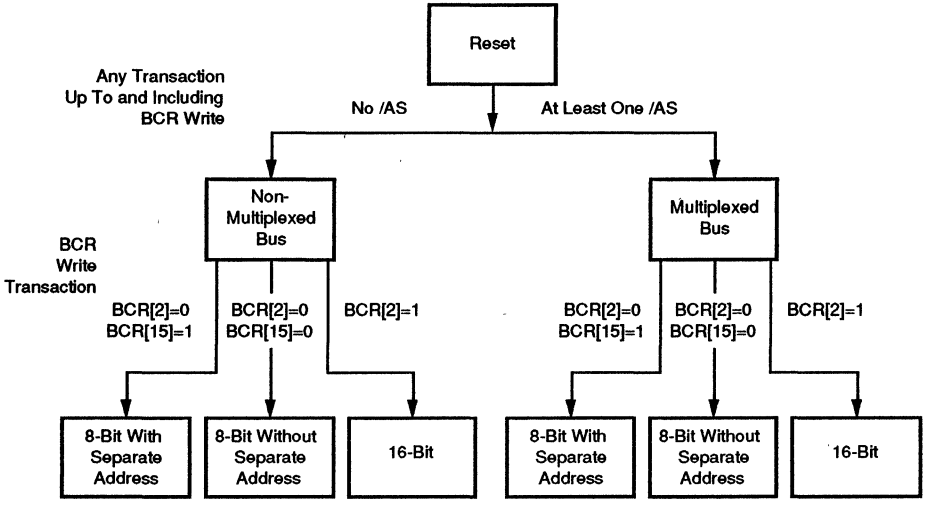
Table 1. Multiplexed Bus Address Assignments

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

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. The RDR and TDR are still accessed directly using the D//C pin, without disturbing the contents of the pointer in the CCAR.

There are two important things to note about the MUSC 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 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) 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.



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-AØ			Address A4-AØ		
00000	CCAR	Channel Command/Address Register	10001	RMR	Receive Mode Register
00001	CMR	Channel Mode Register	10010	RCSR	Receive Command/Status Register
00010	CCSR	Channel Command/Status Register	10011	RICR	Receive Interrupt Control Register
00011	CCR	Channel Control Register	10100	RSR	Receive Sync Register
00100	PSR	Port Status Register	10101	RCLR	Receive Count Limit Register
00101	PCR	Port Control Register	10110	RCCR	Receive Character Count Register
00110	TMDR	Test Mode Data Register	10111	TC0R	Time Constant 0 Register
00111	TMCR	Test Mode Control Register	1X000	TDR	Transmit Data Register (Write Only)
01000	CMCR	Clock Mode Control Register	11001	TMR	Transmit Mode Register
01001	HCR	Hardware Configuration Register	11010	TCSR	Transmit Command/Status Register
01010	IVR	Interrupt Vector Register	11011	TICR	Transmit Interrupt Control Register
01011	IOCR	I/O Control Register	11100	TSR	Transmit Sync Register
01100	ICR	Interrupt Control Register	11101	TCLR	Transmit Count Limit Register
01101	DCCR	Daisy-Chain Control Register	11110	TCCR	Transmit Character Count Register
01110	MISR	Misc Interrupt Status Register	11111	TC1R	Time Constant 1 Register
01111	SICR	Status Interrupt Control Register			
1X000	RDR	Receive Data Register (Read Only)	XXXXX	BCR	Bus Configuration Register

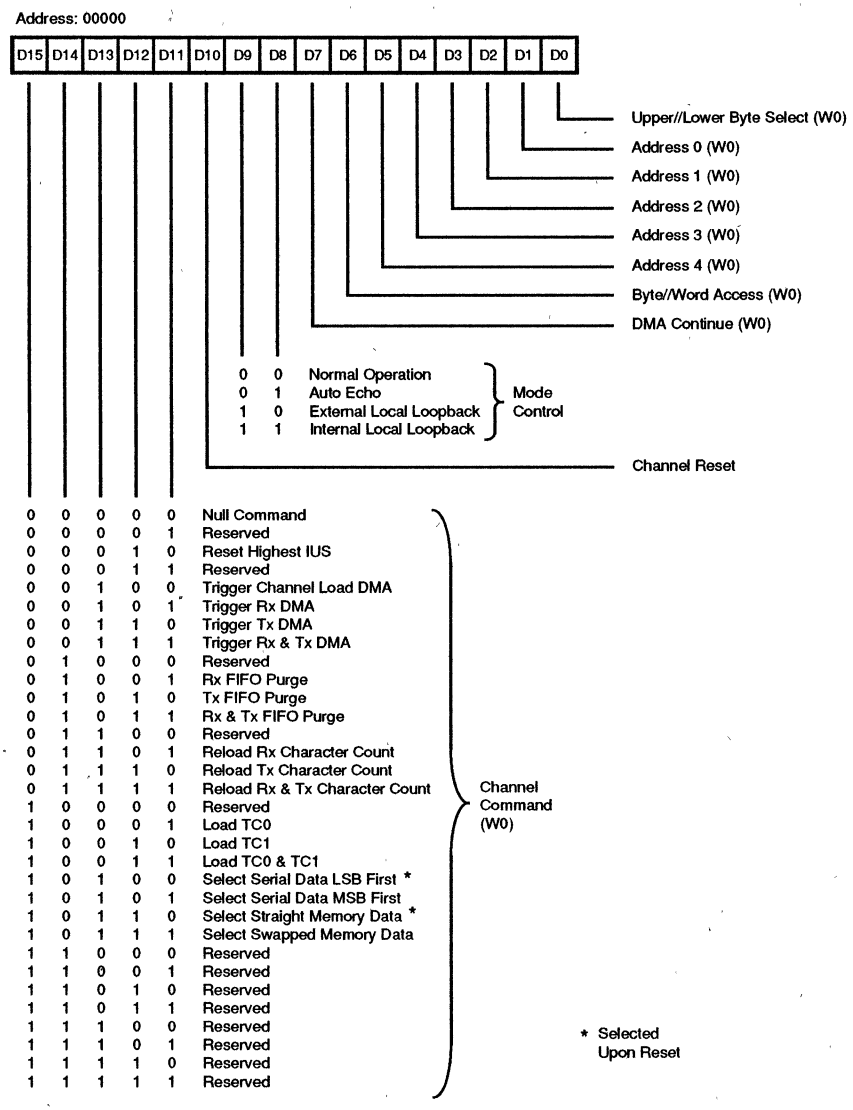


Figure 7. Channel Command/Address Register

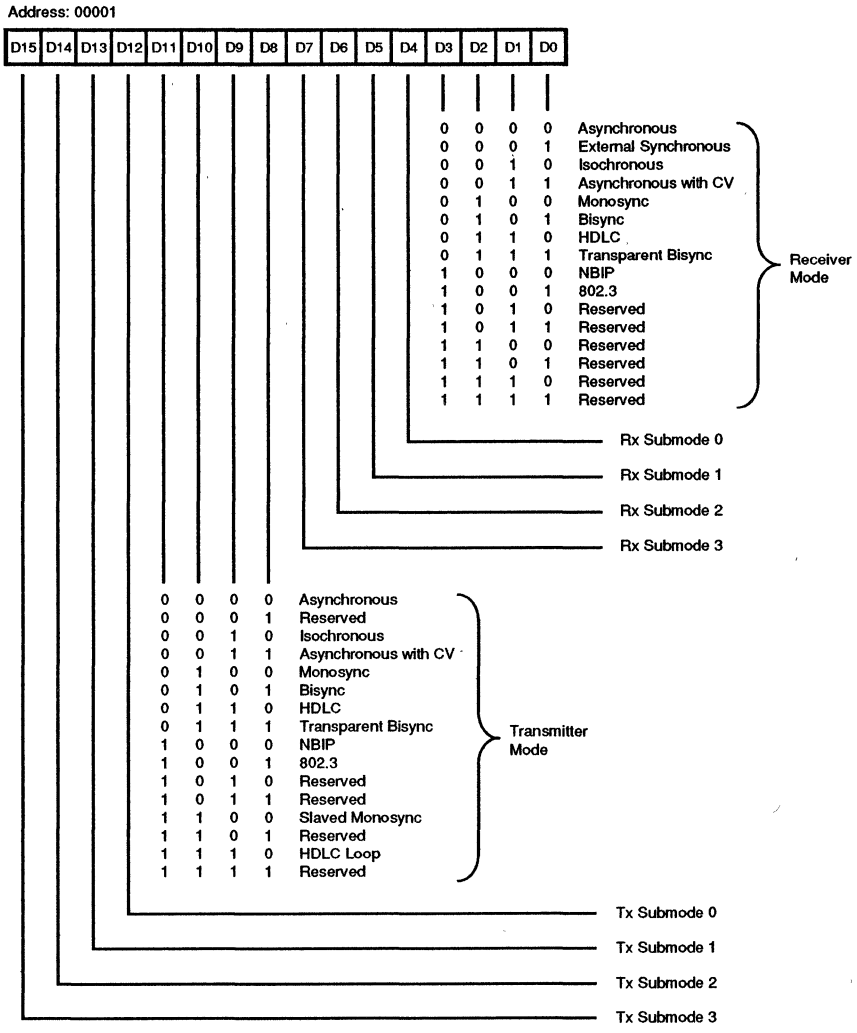


Figure 8. Channel Mode Register

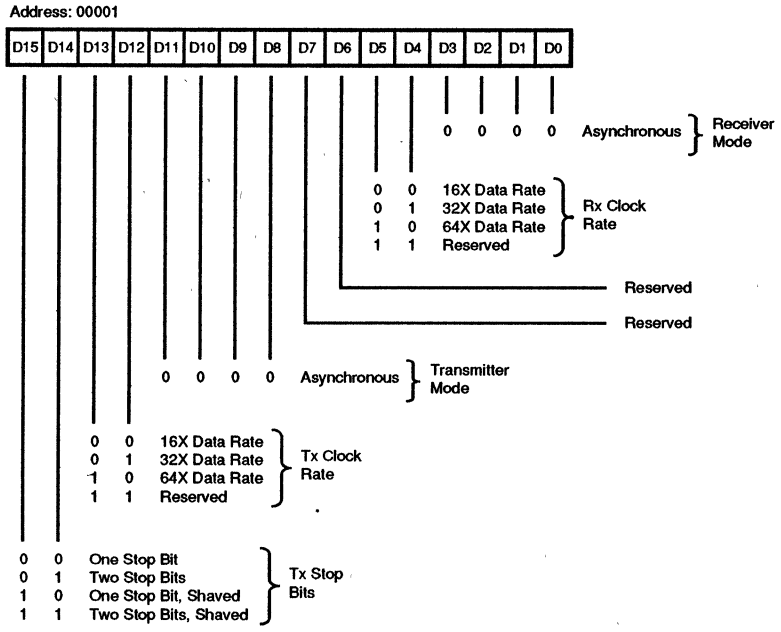


Figure 9. Channel Mode Register, Asynchronous Mode

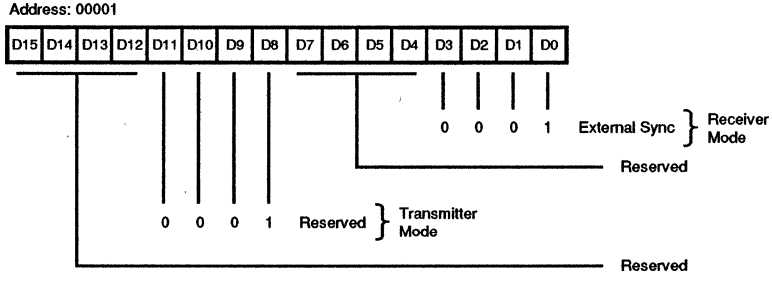


Figure 10. Channel Mode Register, External Sync Mode

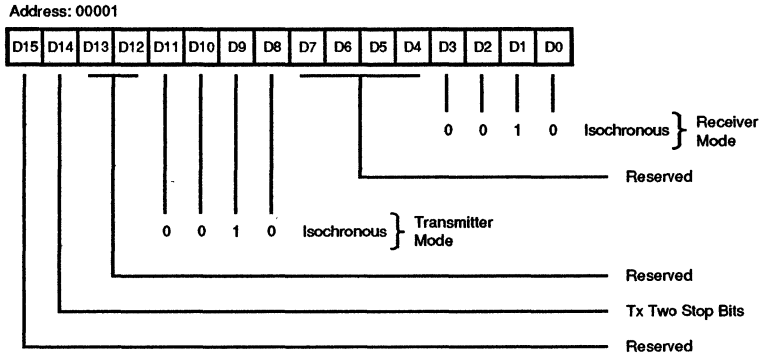


Figure 11. Channel Mode Register, Isochronous Mode

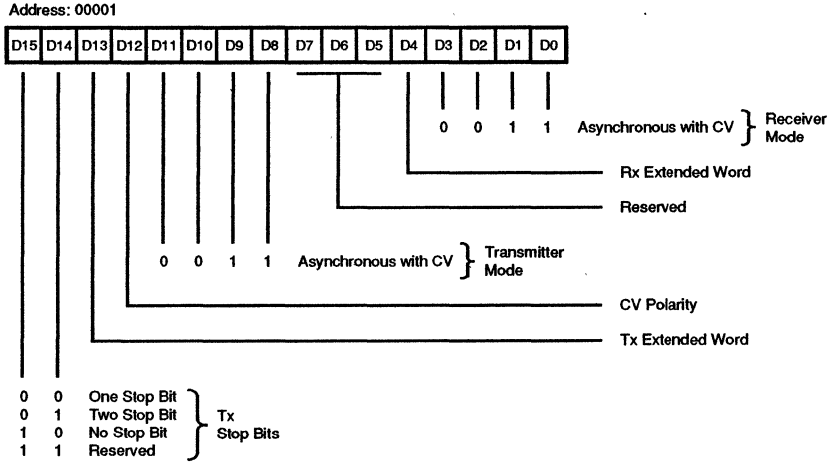


Figure 12. Channel Mode Register, Asynchronous Mode with Code Violation (MIL STD 1553)

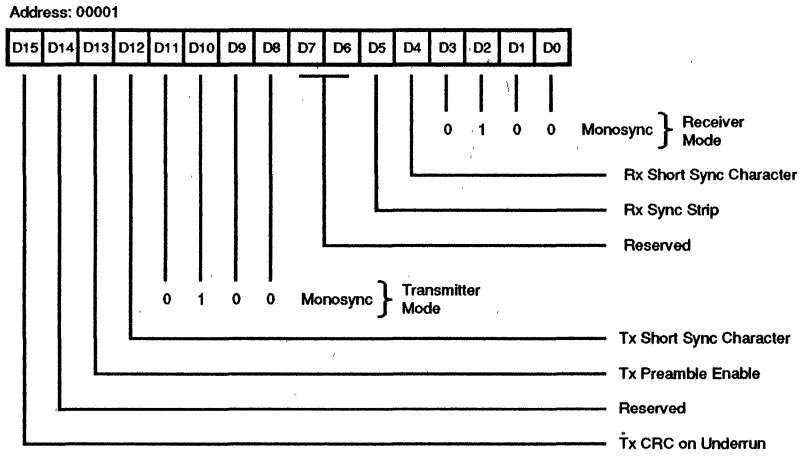


Figure 13. Channel Mode Register, Monosync Mode

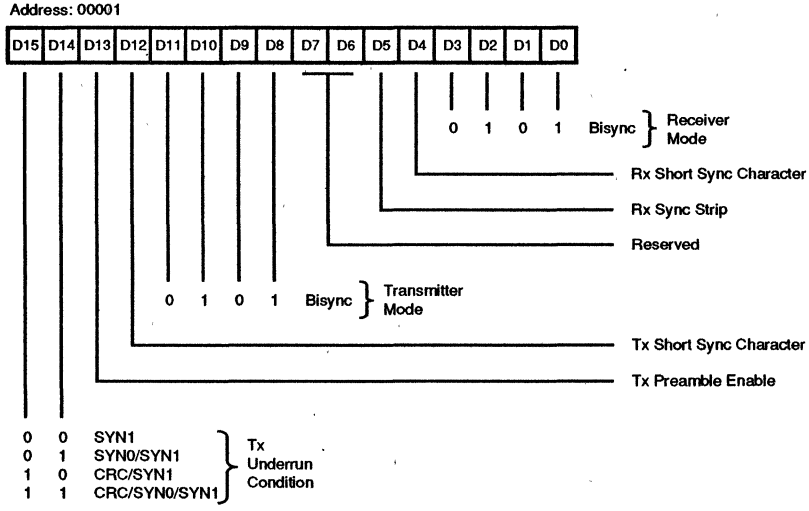


Figure 14. Channel Mode Register, Bisync Mode

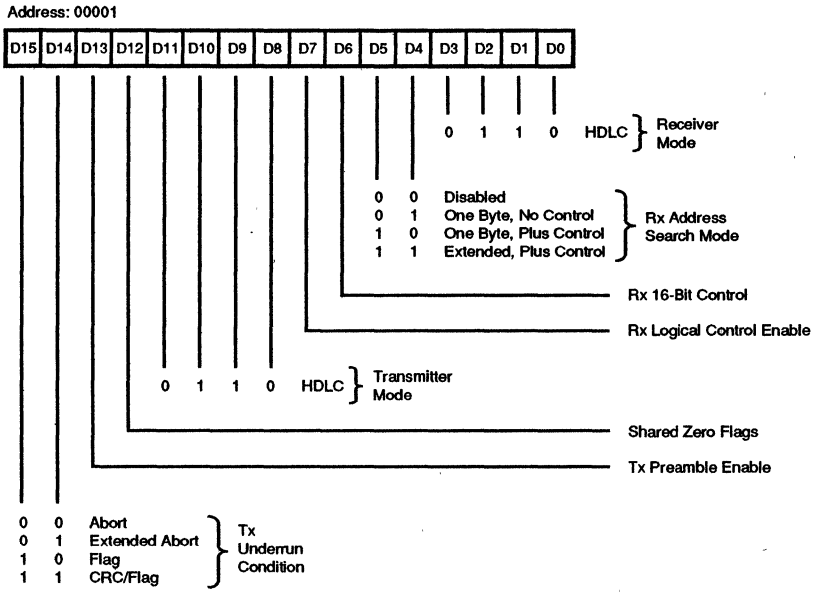


Figure 15. Channel Mode Register, HDLC Mode

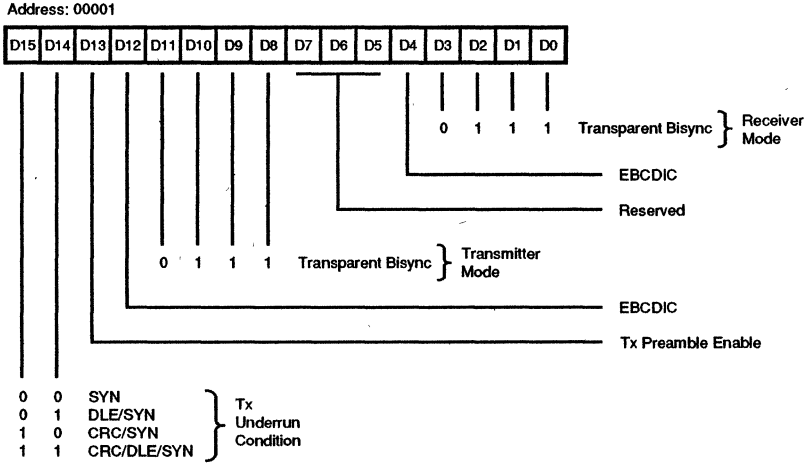


Figure 16. Channel Mode Register, Transparent Bisync Mode

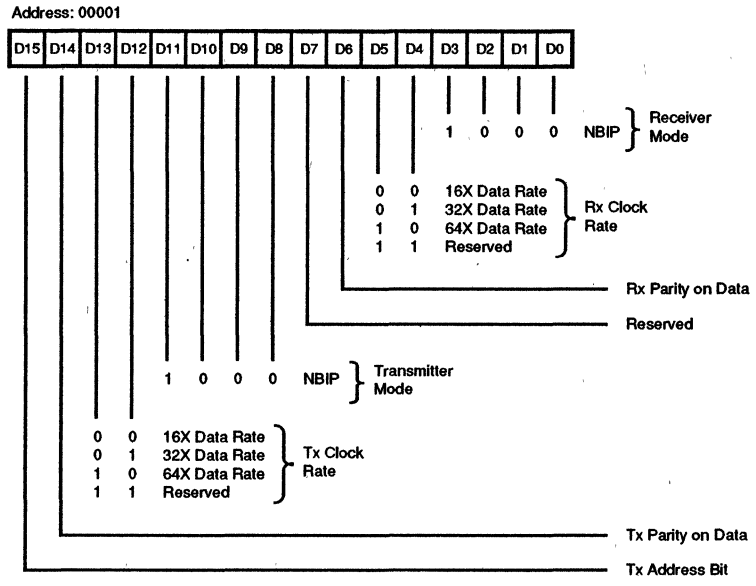


Figure 17. Channel Mode Register, NBIP Mode

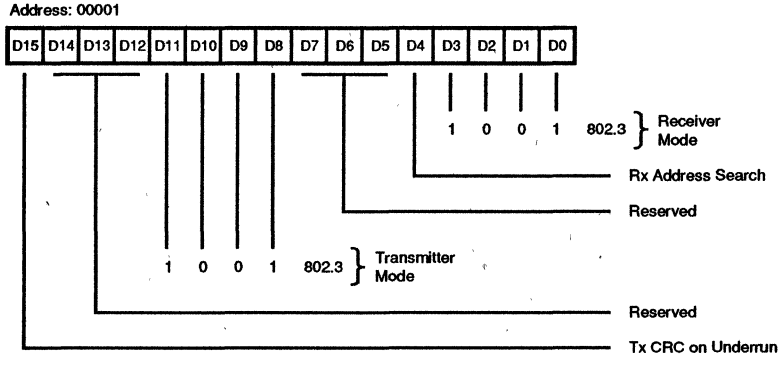


Figure 18. Channel Mode Register, 802.3 Mode

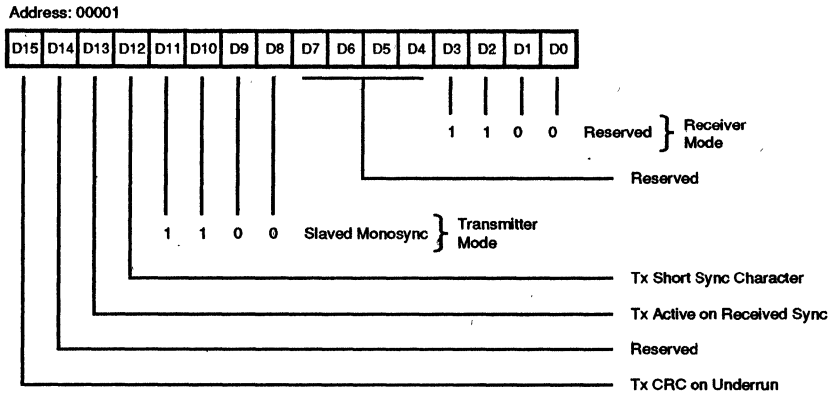


Figure 19. Channel Mode Register, Slaved Monosync Mode

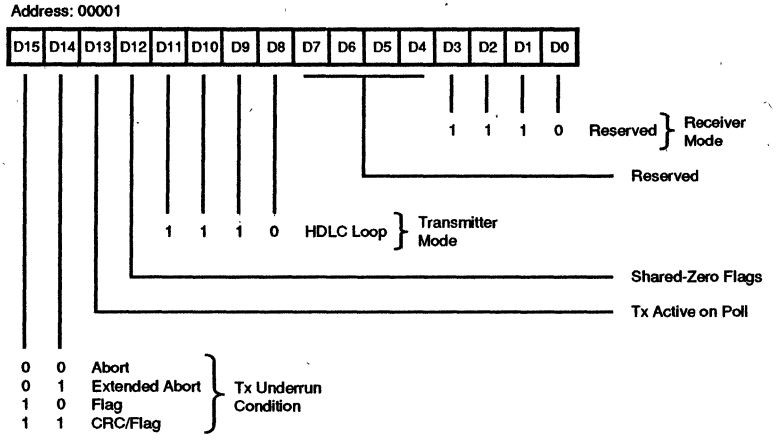


Figure 20. Channel Mode Register, HDLC Loop Mode

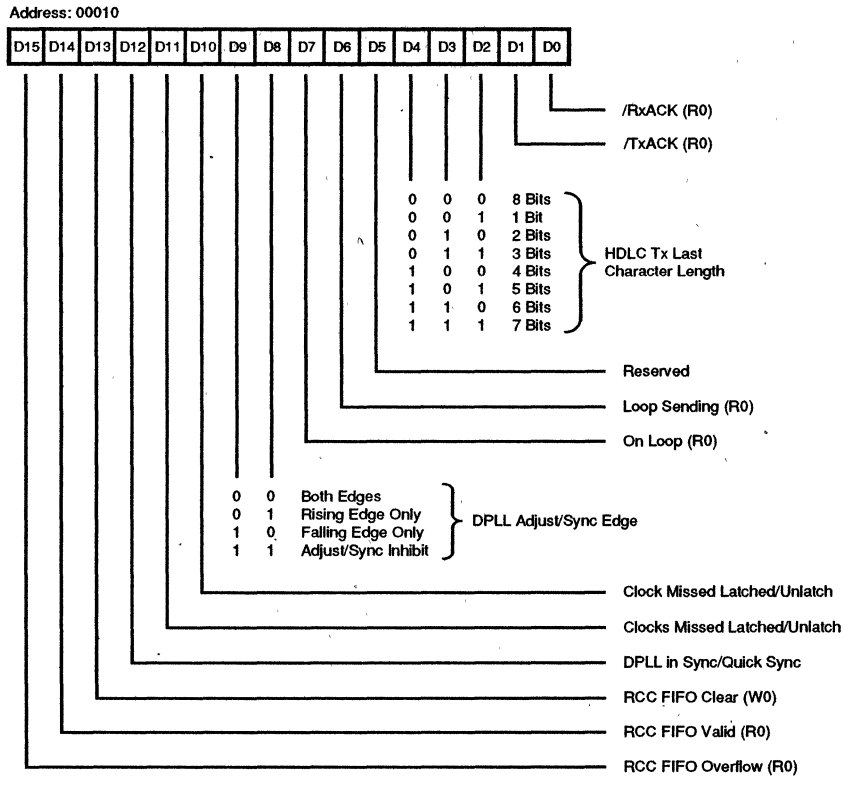


Figure 21. Channel Command/Status Register

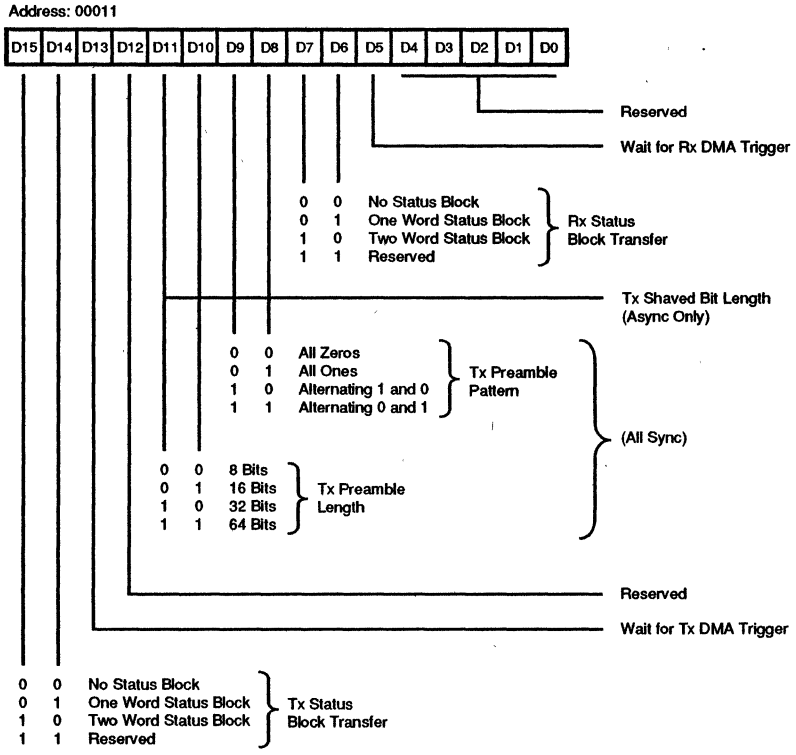


Figure 22. Channel Control Register

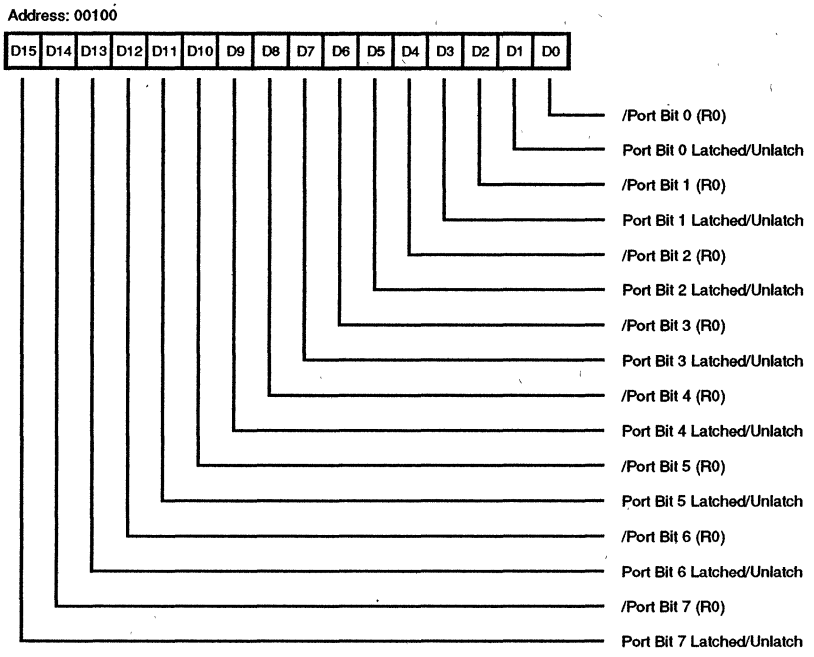


Figure 23. Port Status Register

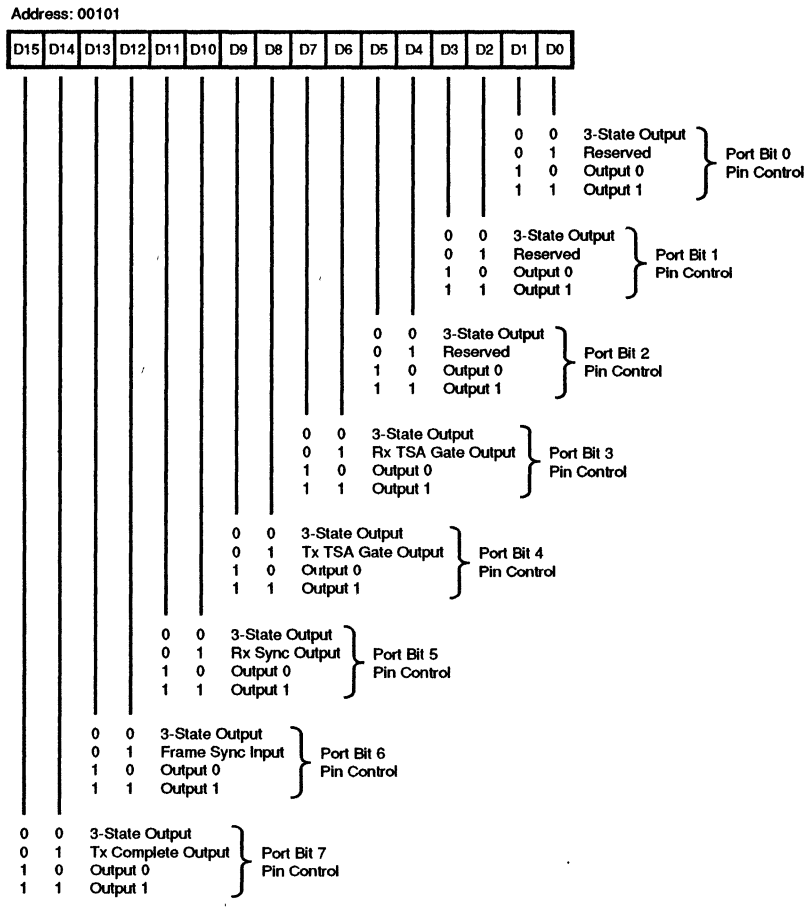


Figure 24. Port Control Register

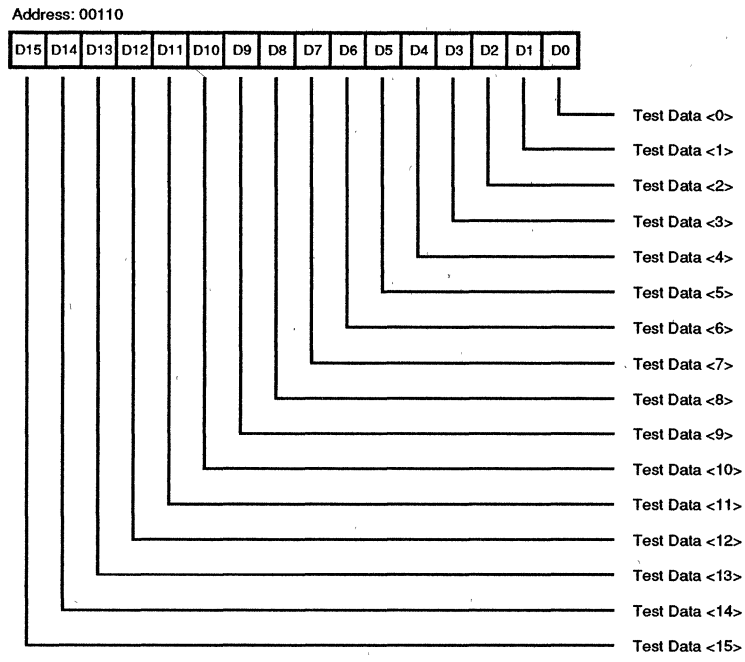


Figure 25. Test Mode Data Register

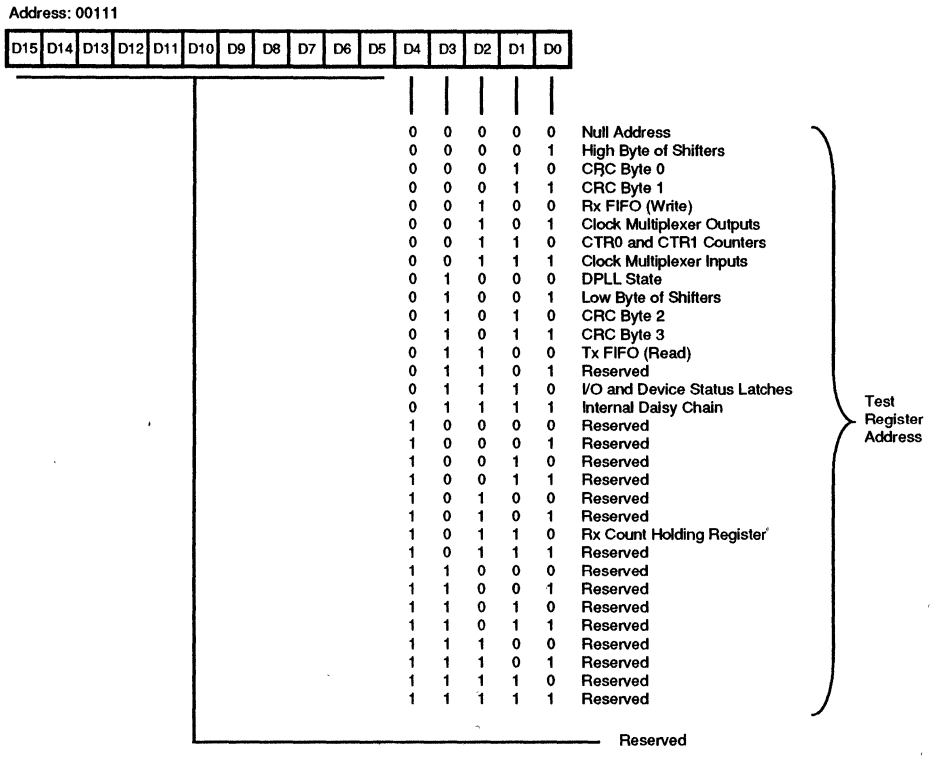


Figure 26. Test Mode Control Register

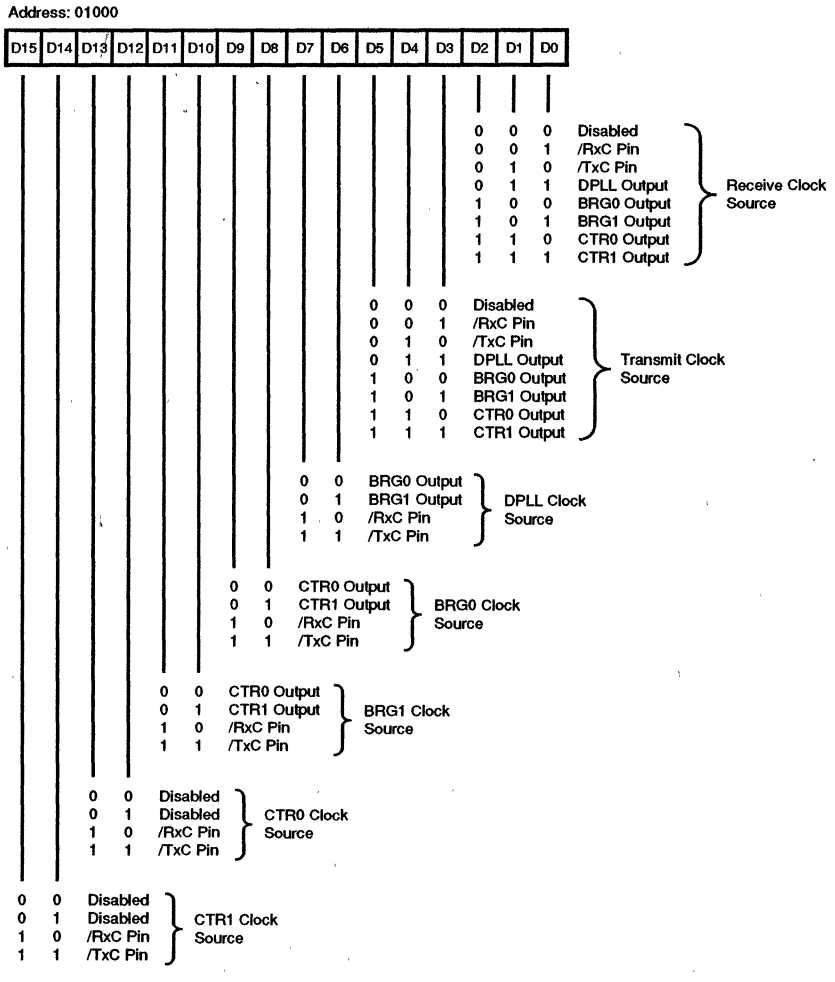


Figure 27. Clock Mode Control Register

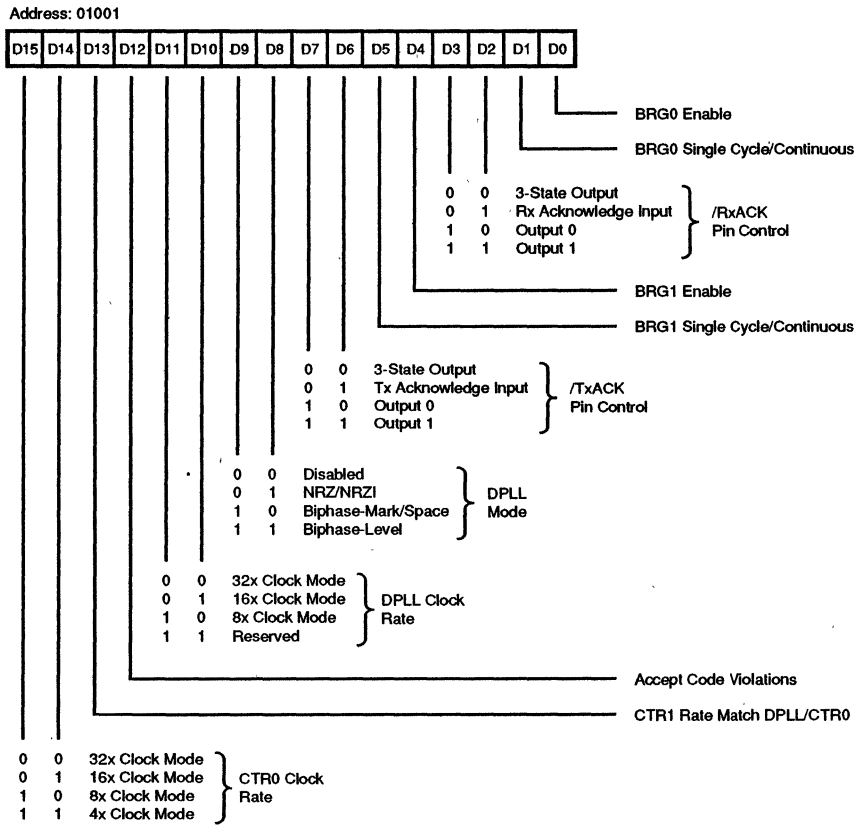


Figure 28. Hardware Configuration Register

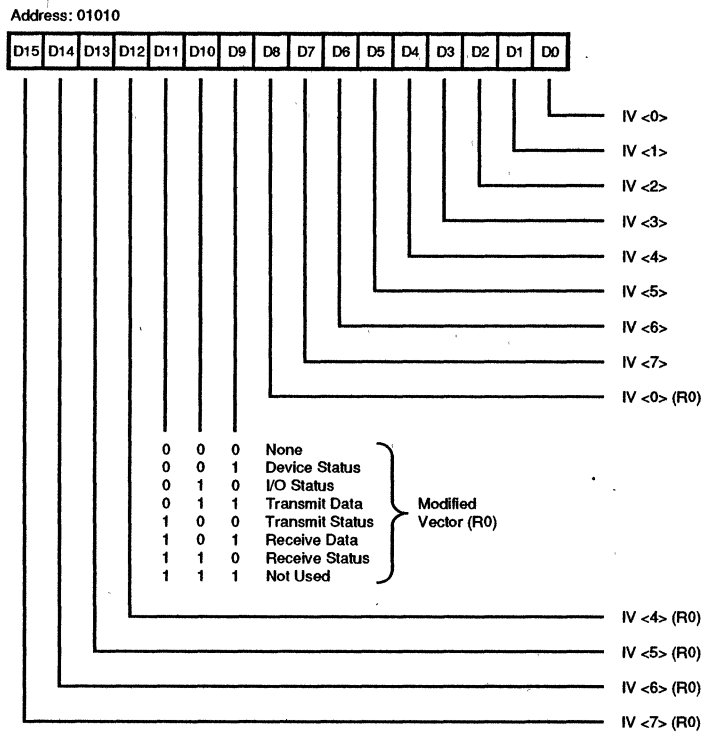


Figure 29. Interrupt Vector Register

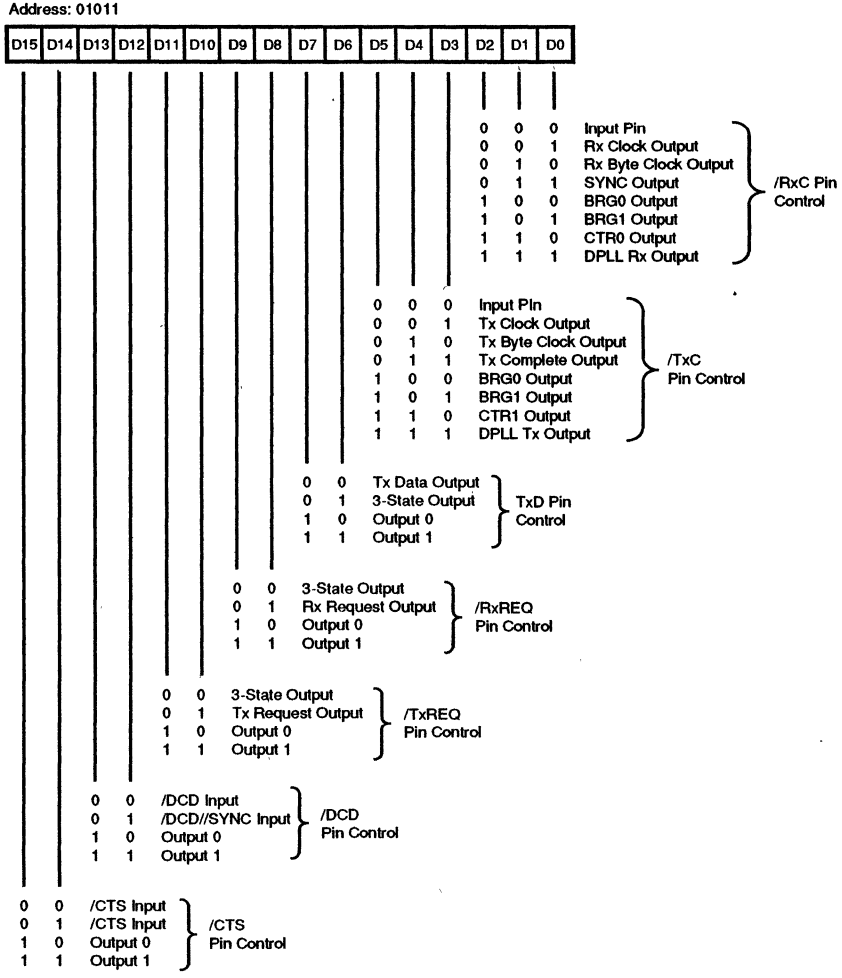


Figure 30. I/O Control Register

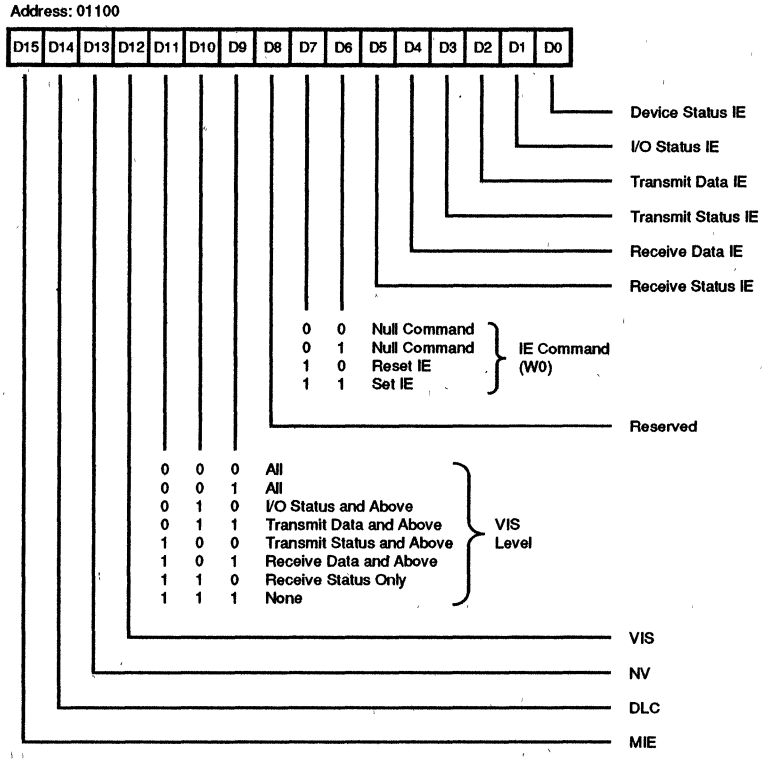


Figure 31. Interrupt Control Register

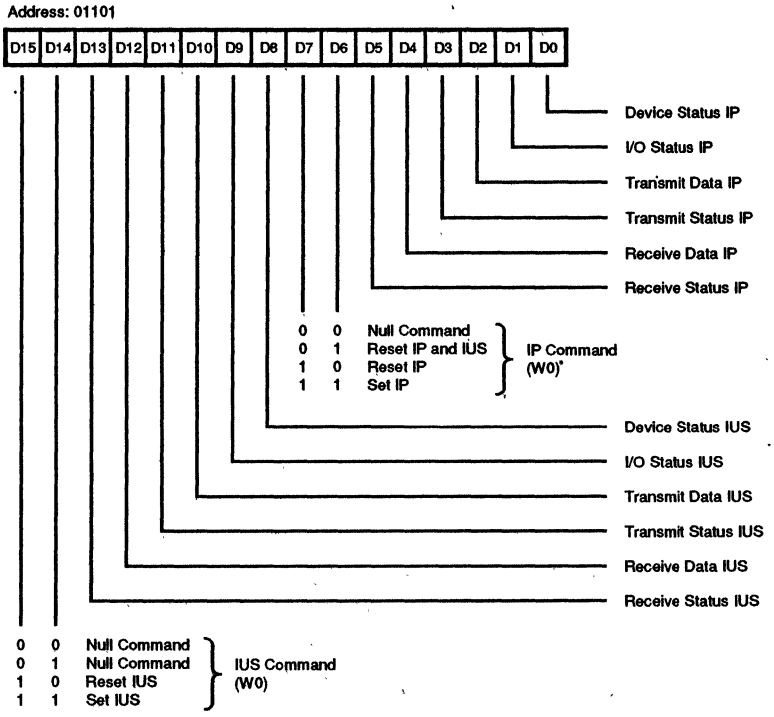


Figure 32. Daisy-Chain Control Register

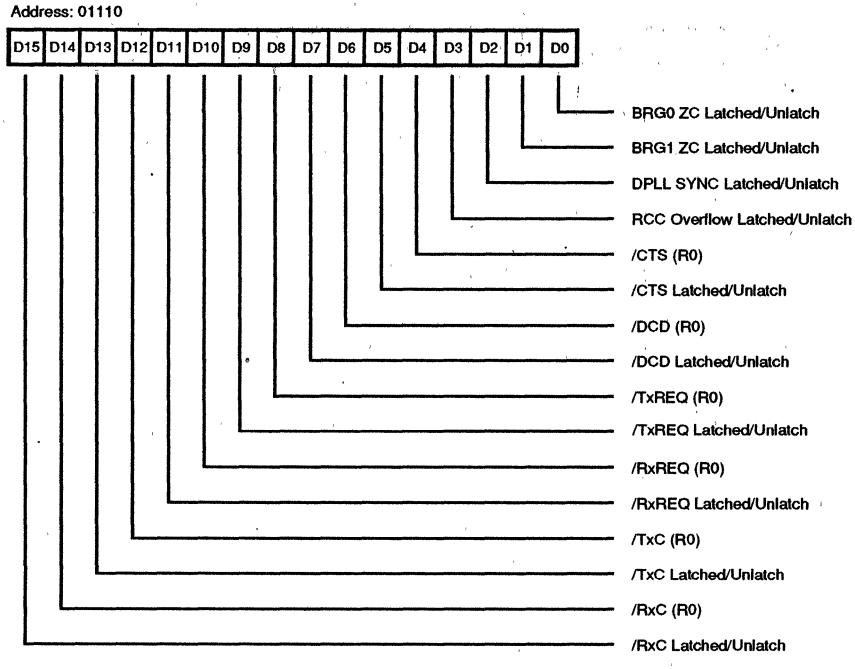


Figure 33. Miscellaneous Interrupt Status Register

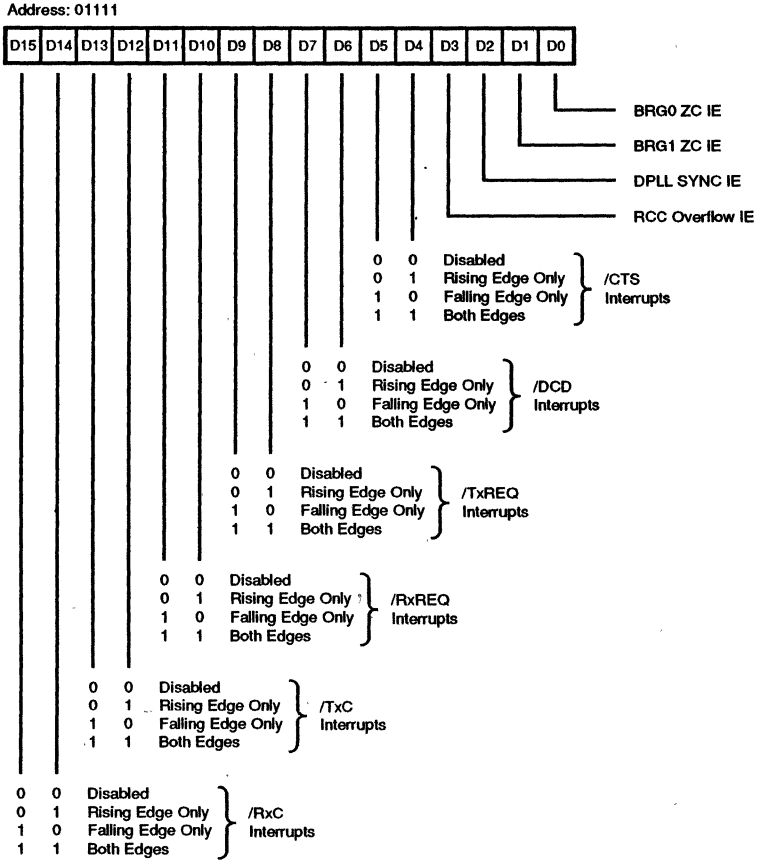


Figure 34. Status Interrupt Control Register

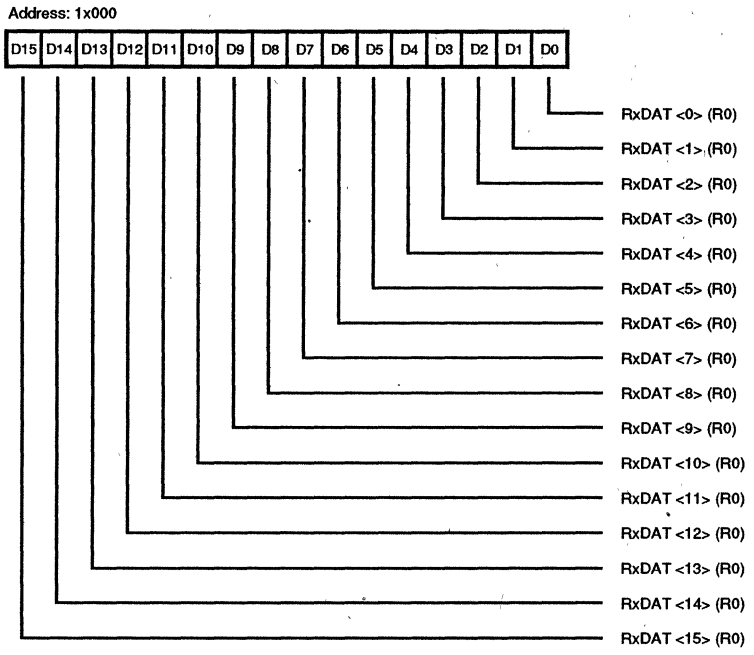


Figure 35. Receive Data Register

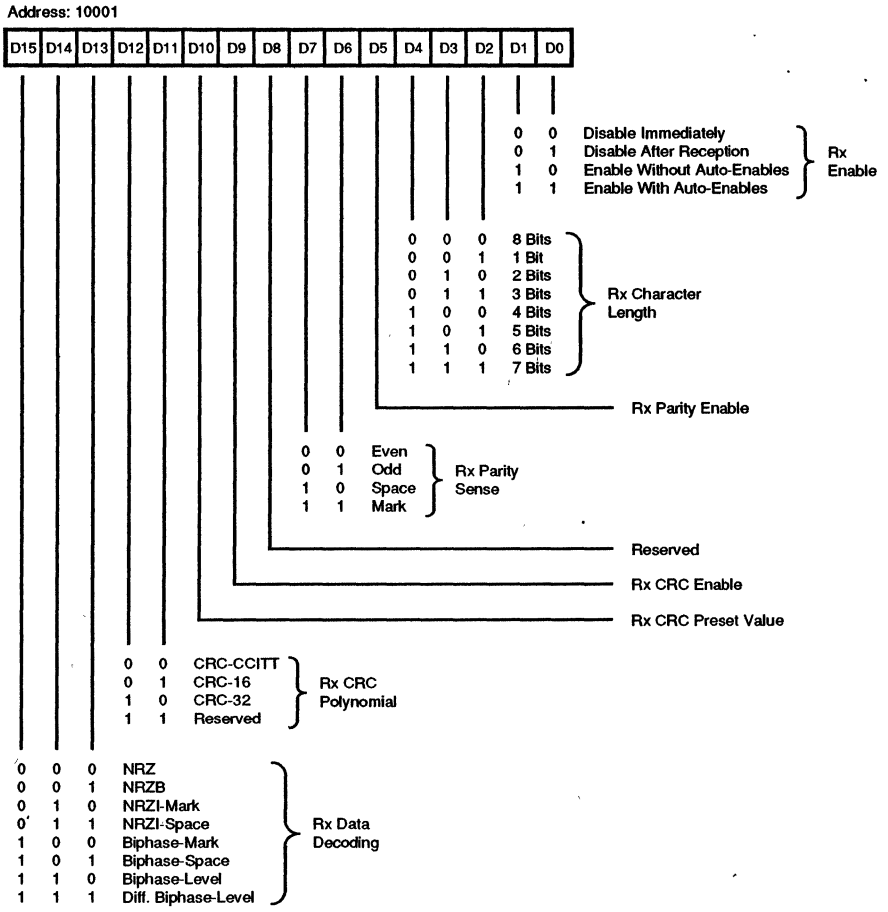


Figure 36. Receive Mode Register

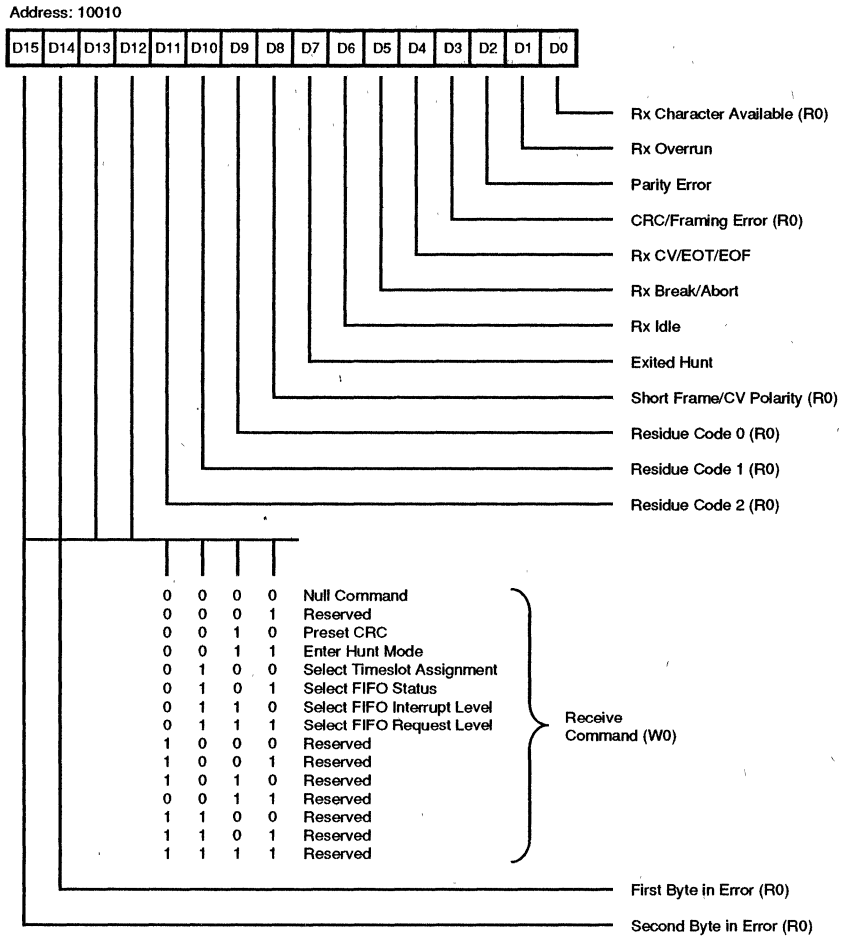


Figure 37. Receive Command Status Register

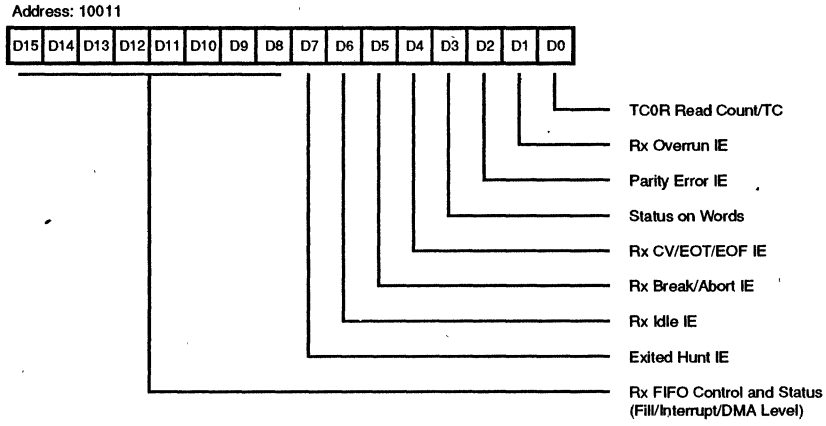


Figure 38a. Receive Interrupt Control Register

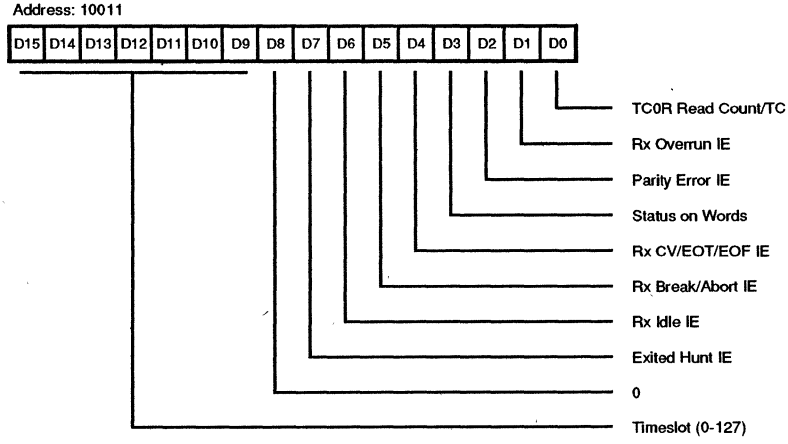


Figure 38b. Receive Interrupt Control Register

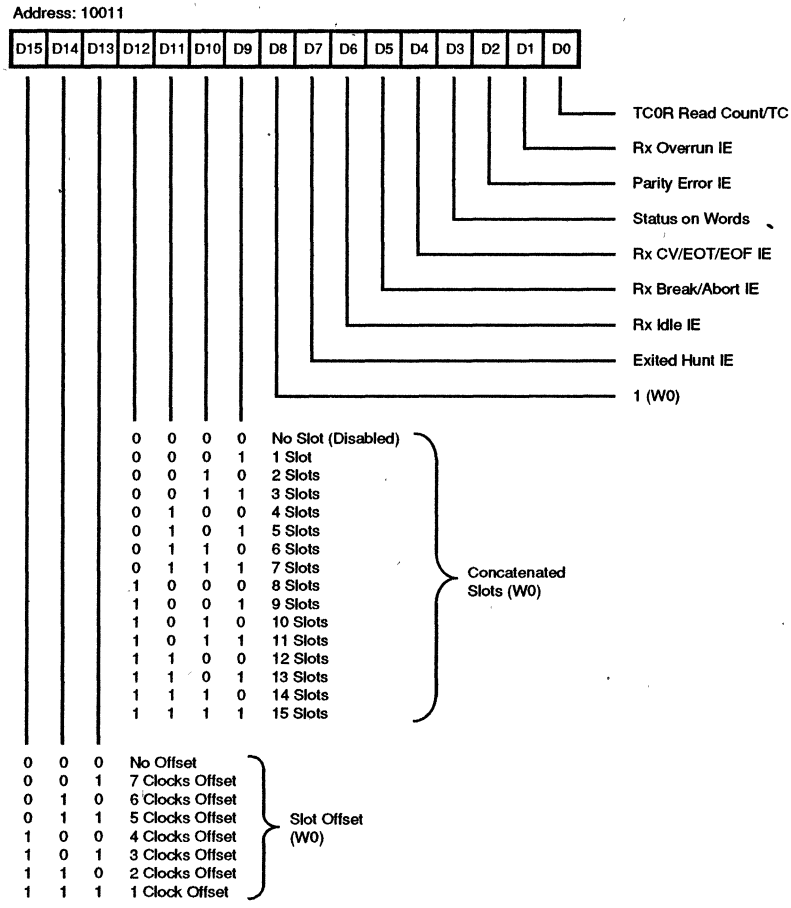


Figure 38c. Receive Interrupt Control Register

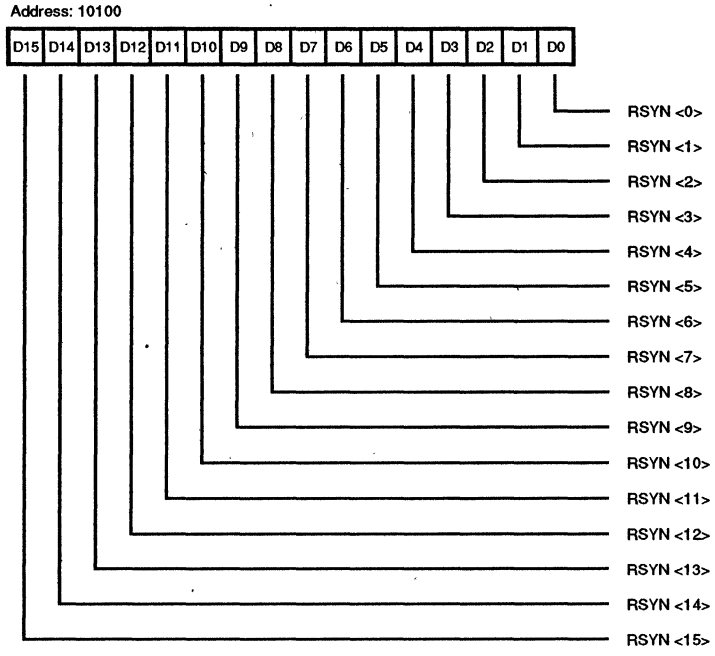


Figure 39. Receive Sync Register

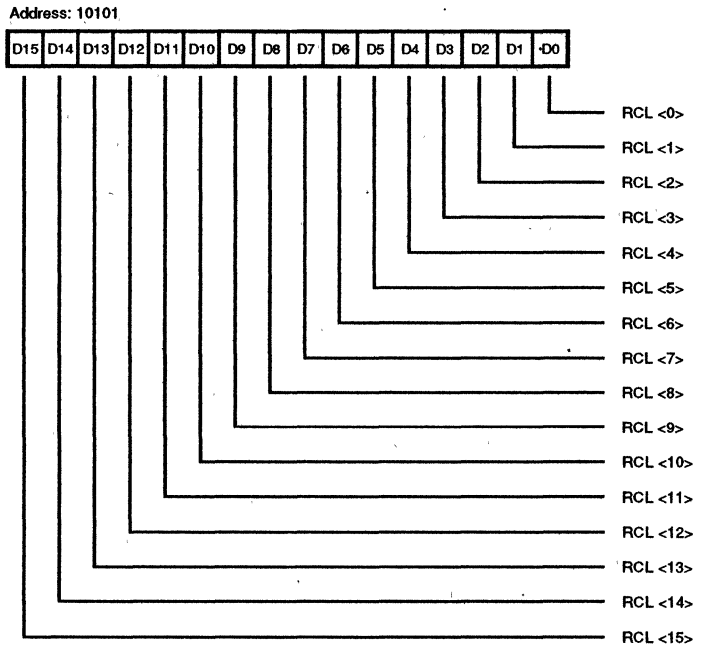


Figure 40. Receive Count Limit Register

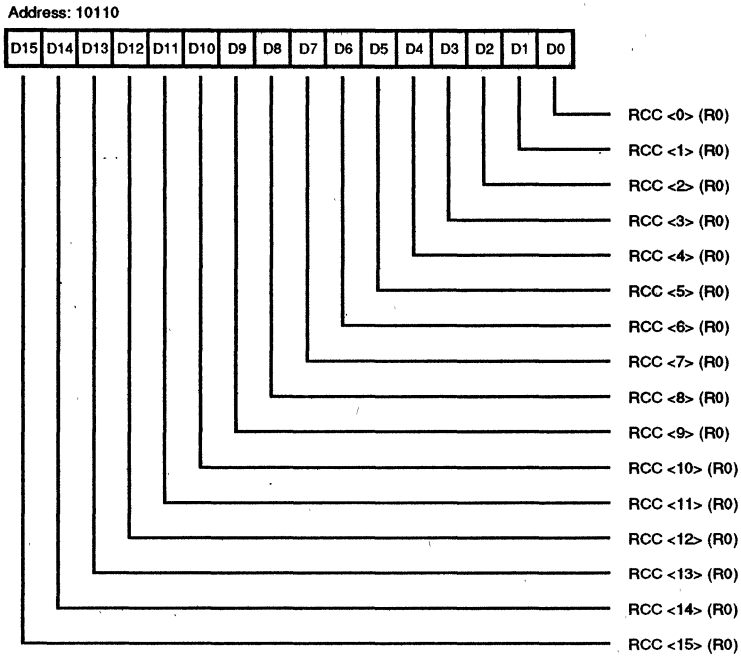


Figure 41. Receive Character Count Register

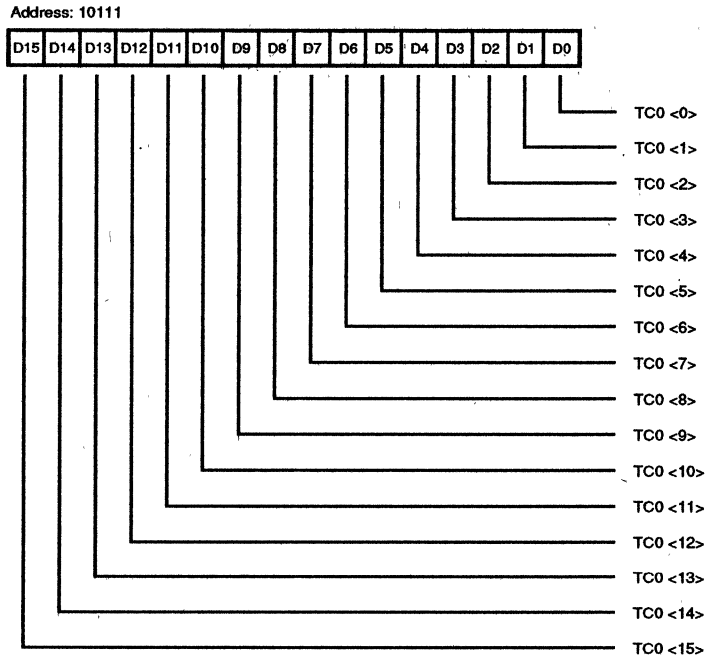


Figure 42. Time Constant 0 Register

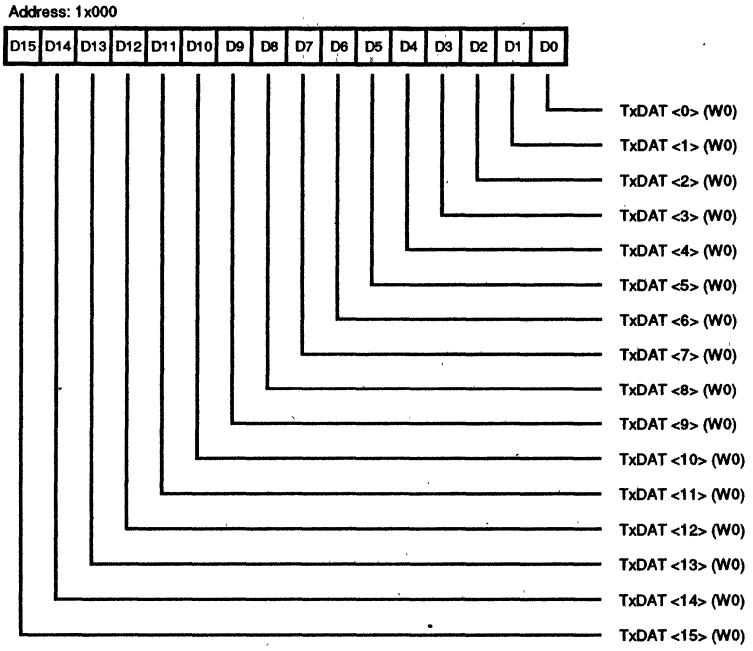


Figure 43. Transmit Data Register

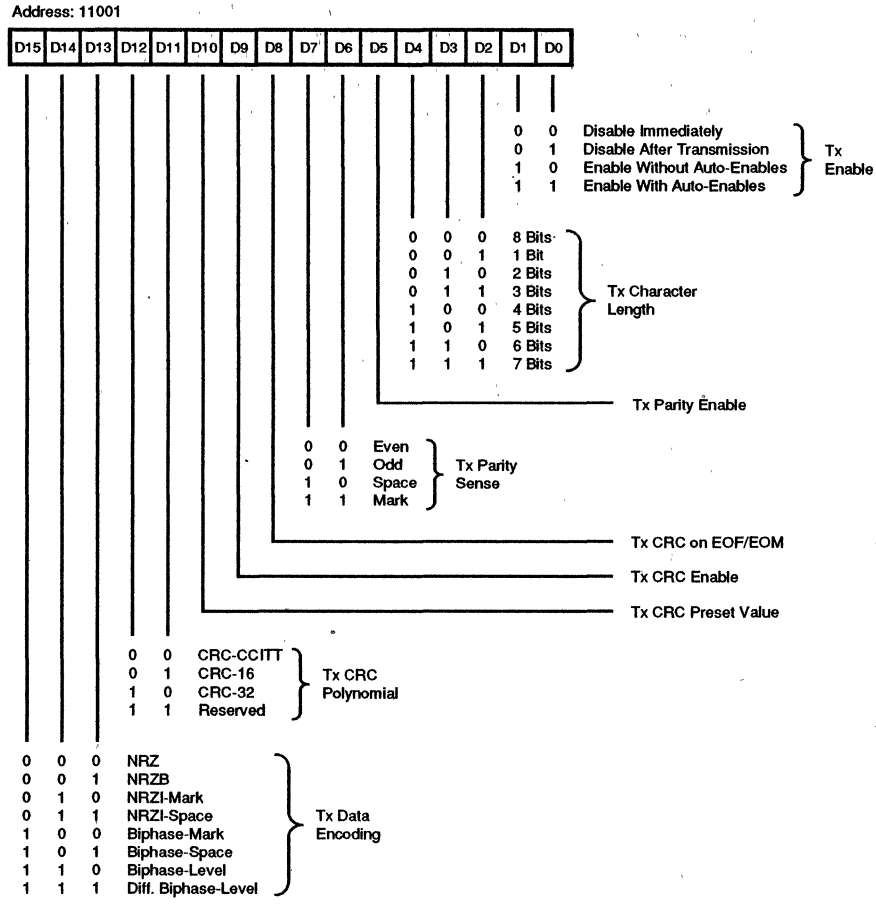


Figure 44. Transmit Mode Register

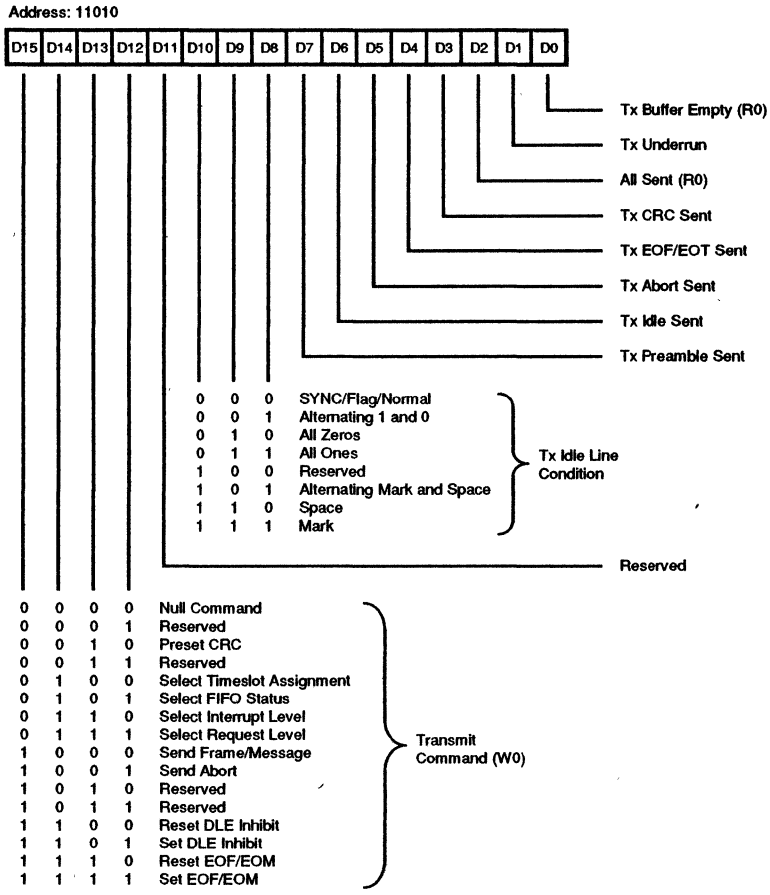


Figure 45. Transmit Command/Status Register

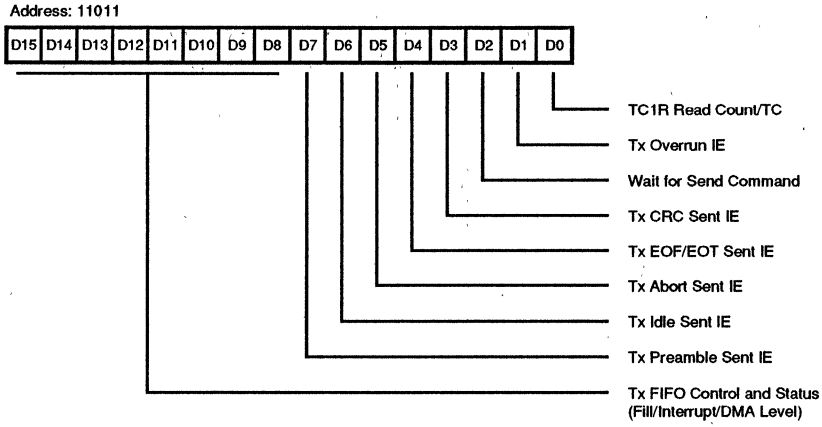


Figure 46a. Transmit Interrupt Control Register

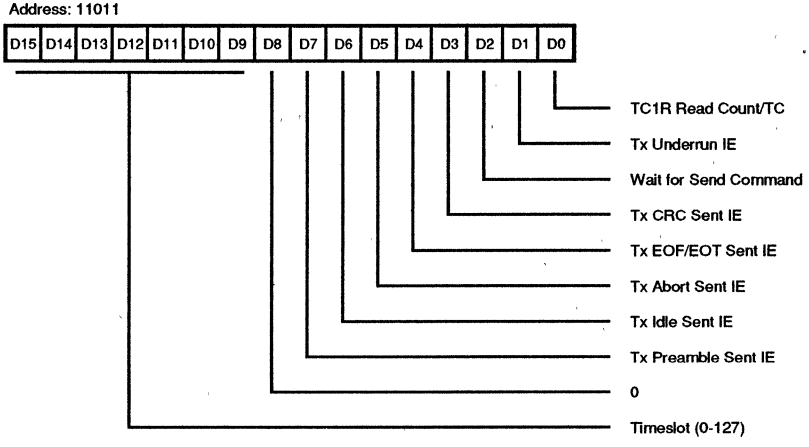


Figure 46b. Transmit Interrupt Control Register

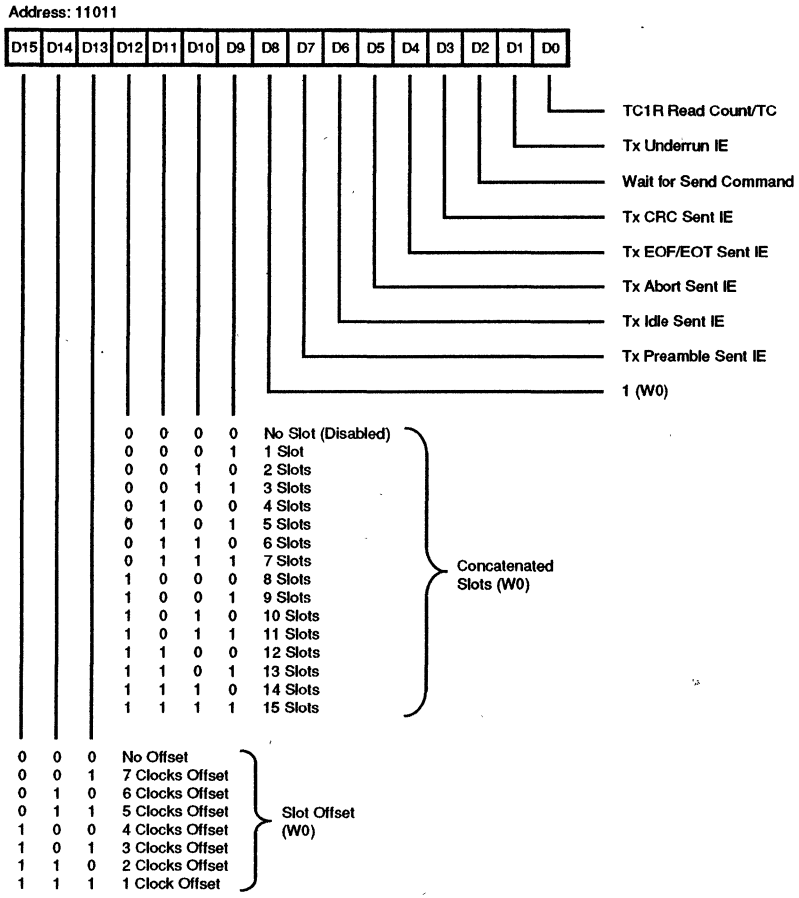


Figure 46c. Transmit Interrupt Control Register

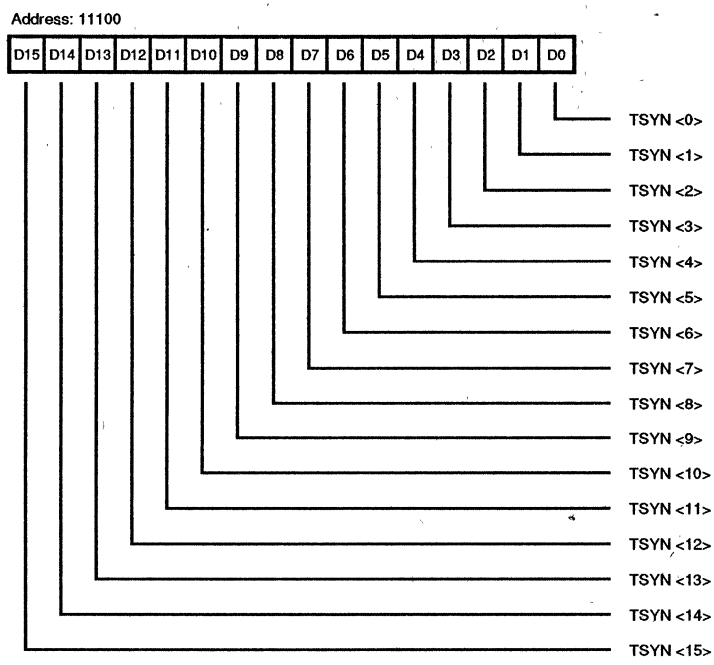


Figure 47. Transmit Sync Register

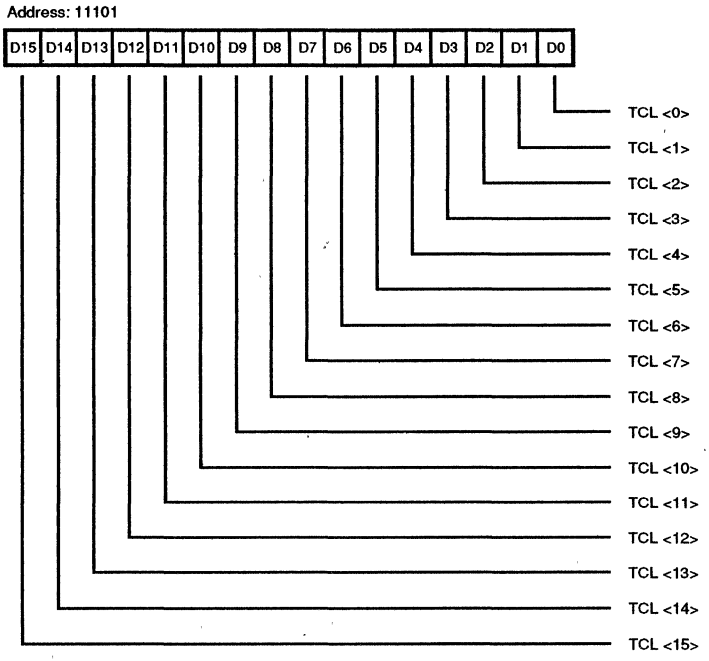


Figure 48. Transmit Count Limit Register

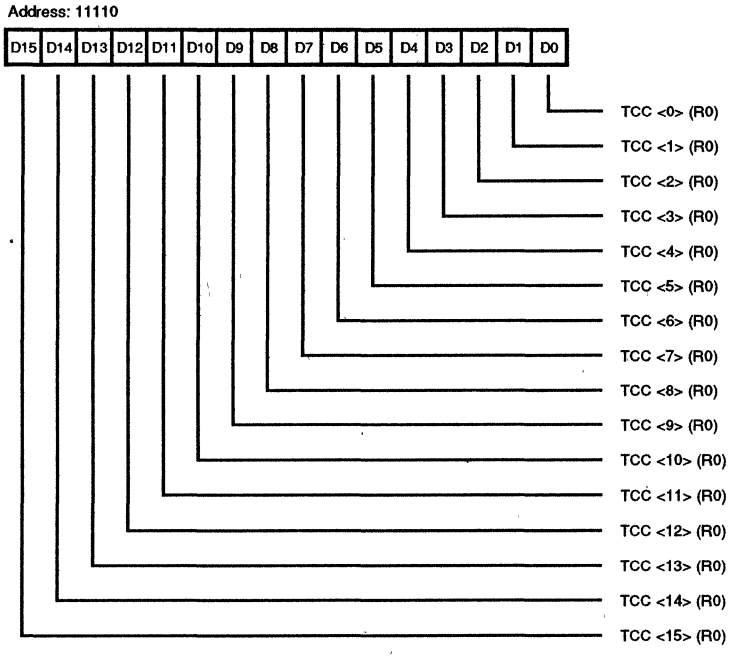


Figure 49. Transmit Character Count Register

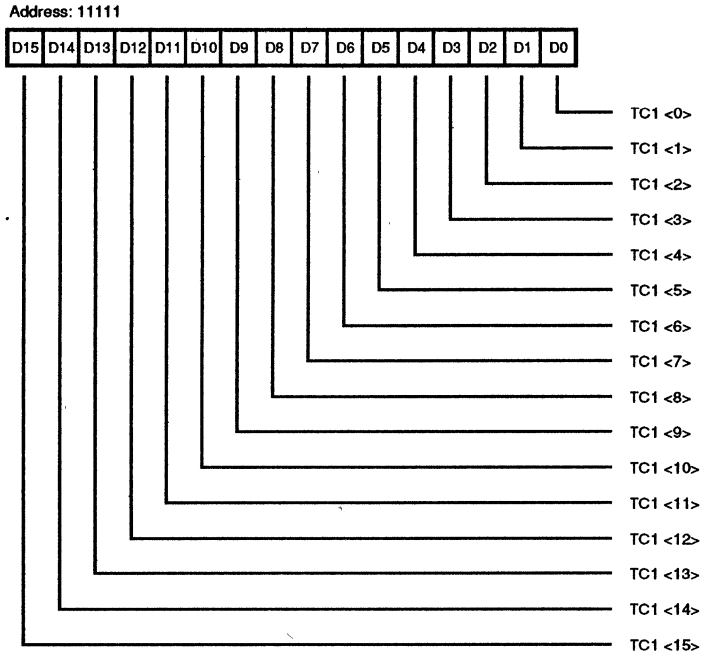
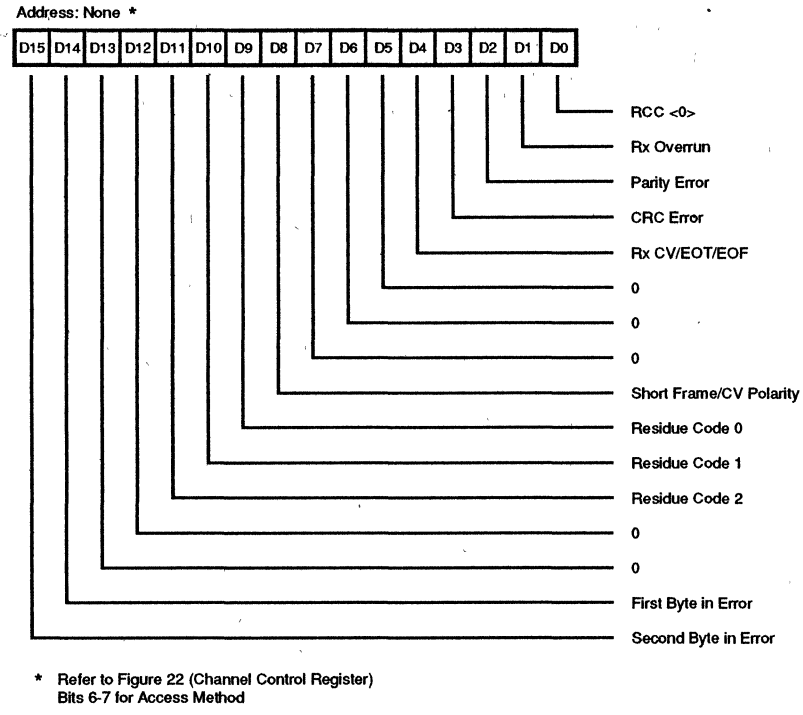
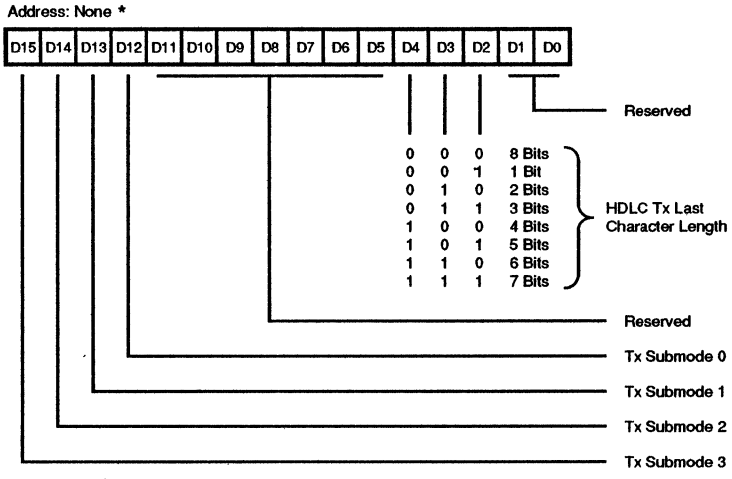


Figure 50. Time Constant 1 Register



* 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

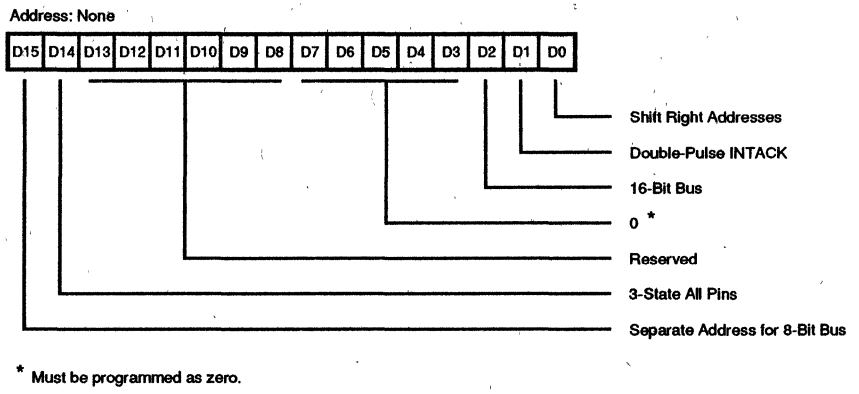


Figure 53. Bus Configuration Register

MUSC TIMING

The MUSC 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 are 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 MUSC 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.

ABSOLUTE MAXIMUM RATINGS

Voltages on all pins
with respect to Vss ..... -0.3 V to +7.0 V
Voltages on all inputs
with respect to Vss ..... -0.3V to Vcc +0.3V
Operating Ambient
Temperature ..... See Ordering Information
Storage Temperature ..... -65°C to +150°C

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. Standard conditions are as follows:

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

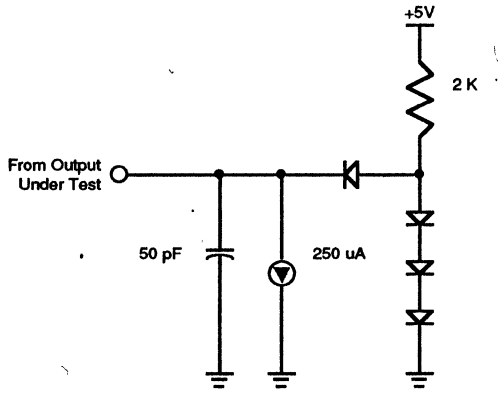


Figure 54. Standard Test Load

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. Unmeasured pins returned to ground.

MISCELLANEOUS Transistor Count - 100,000

DC CHARACTERISTICS

Z16C33

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}= -1.6mA
$V_{OH}2$	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
$I_{CC}1$	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
Z16C33

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	0		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	
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]
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		65	ns	[5]

AC CHARACTERISTICS
Z16C33

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)	/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]
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	70		ns	
65	TwRAKh	/RxACK High Width	60		ns	
66	TdRAK(DRa)	/RxACK Fall to Data Active Delay	0		ns	
67	TdRAK(DRv)	/RxACK Fall to Data Valid Delay		85	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	70		ns	
73	TwTAKh	/TxACK High Width	60		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	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	

AC CHARACTERISTICS
Z16C33

No	Symbol	Parameter	Min	Max	Units	Note
89	TdRDf(RDY)	/RD Fall (Intack) to /RDY Fall Delay		200	ns	
90	TdRDr(RDY)	/RD Rise to /RDY Rise Delay		40	ns	
91	TslEI(RDI)	IEI to /RD Fall (Intack) Setup Time	60		ns	
92	ThlEI(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	70		ns	
97	TwPIAh	/PITACK High Width	60		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	
101	TdPIA(DRn)	/PITACK Rise to Data Not Valid Delay	0		ns	
102	TdPIA(DRz)	/PITACK Rise to Data Float Delay		20	ns	
103	TslEI(PIA)	IEI to /PITACK Fall Setup Time	60		ns	
104	ThlEI(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	60		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 PS, 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.

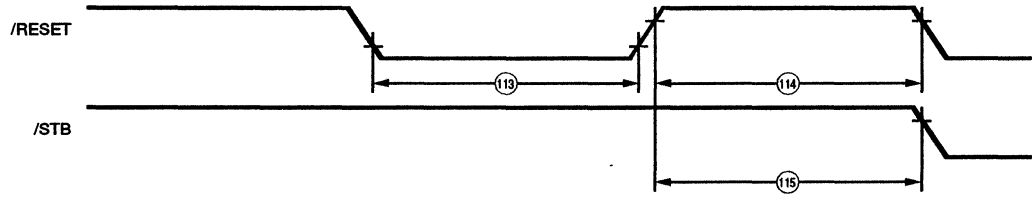


Figure 55. Reset Timing

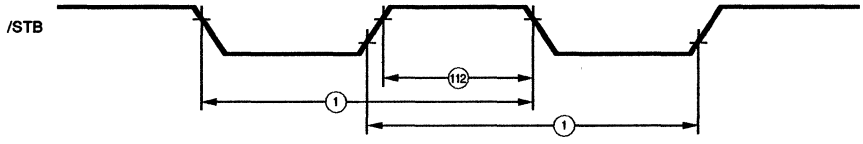


Figure 56. Bus Cycle Timing

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

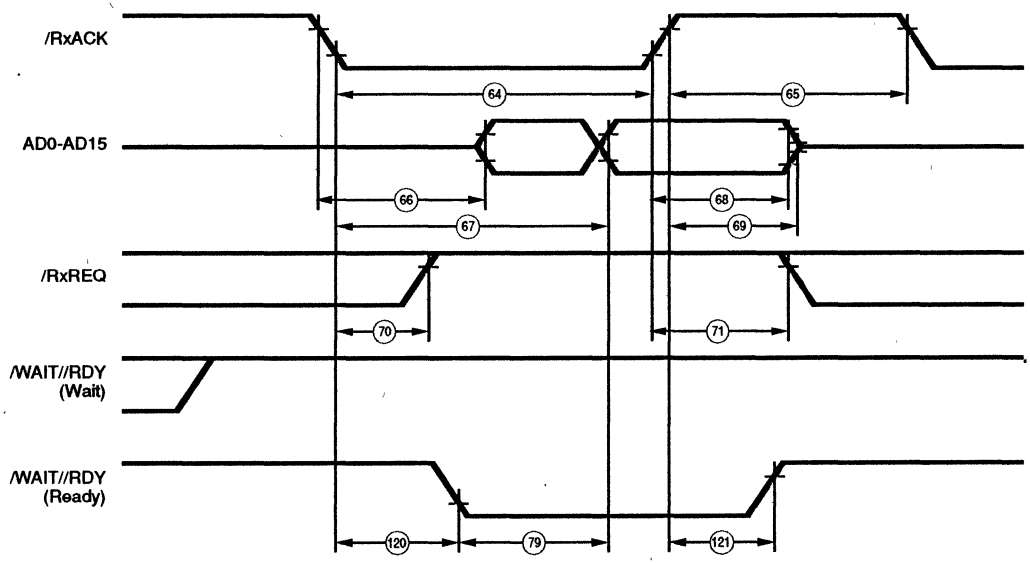


Figure 57. DMA Read Cycle

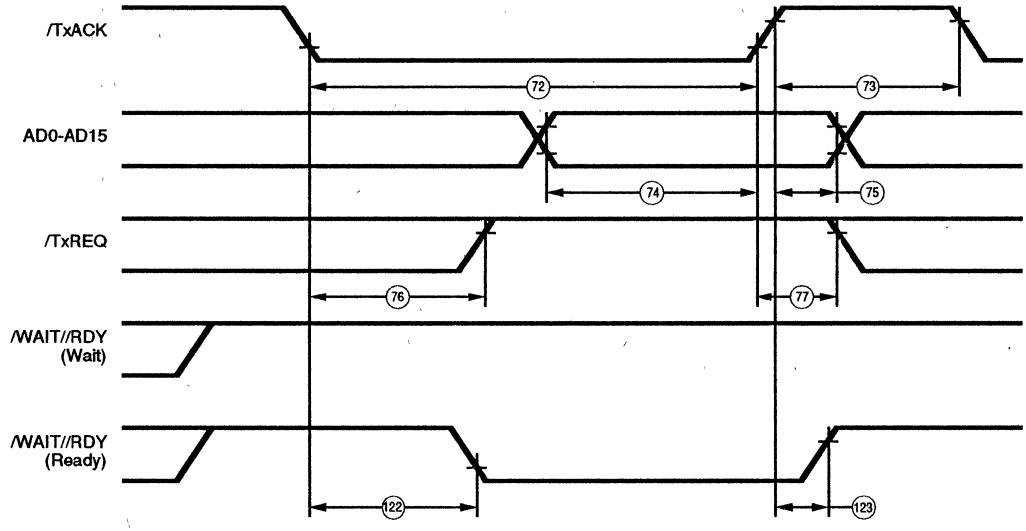


Figure 58. DMA Write Cycle

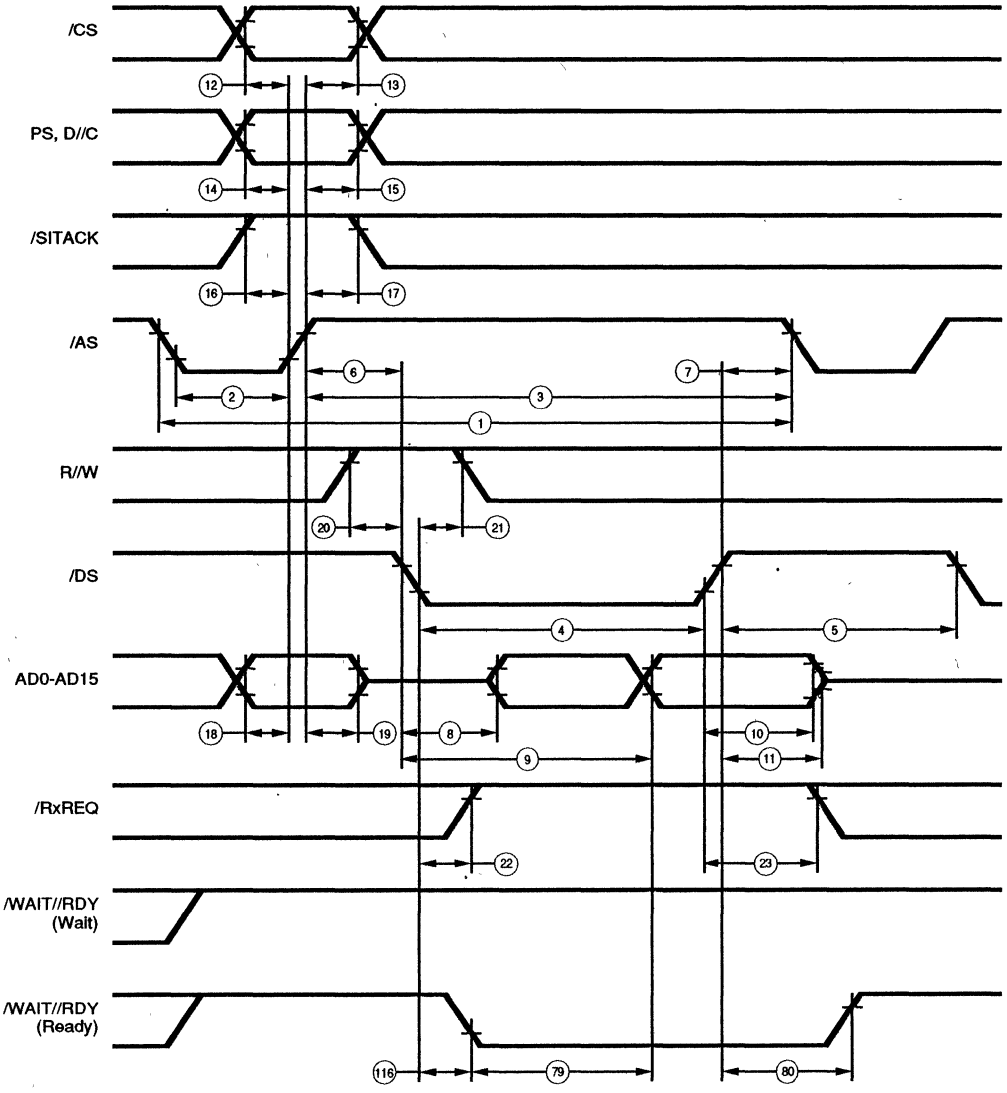


Figure 59. Multiplexed /DS Read Cycle

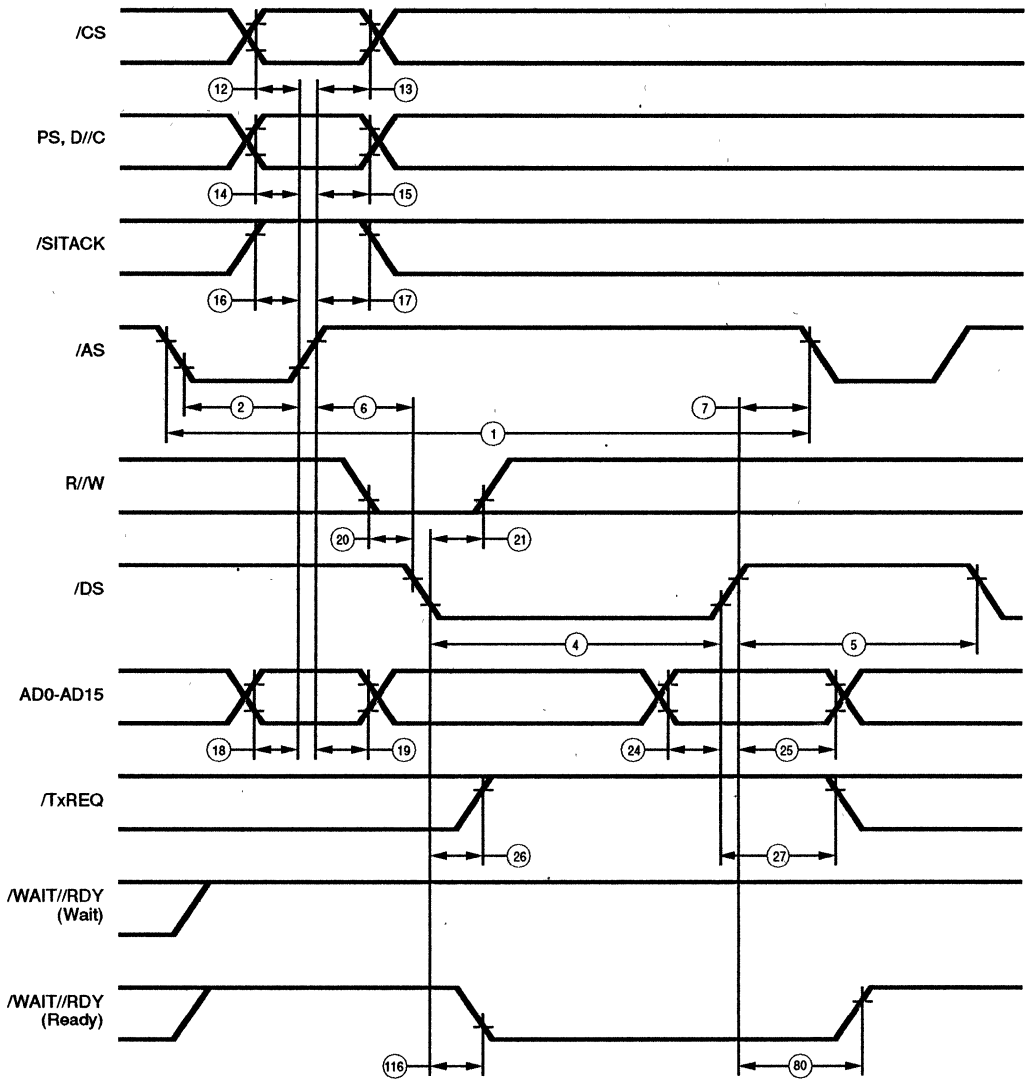


Figure 60. Multiplexed /DS Write Cycle

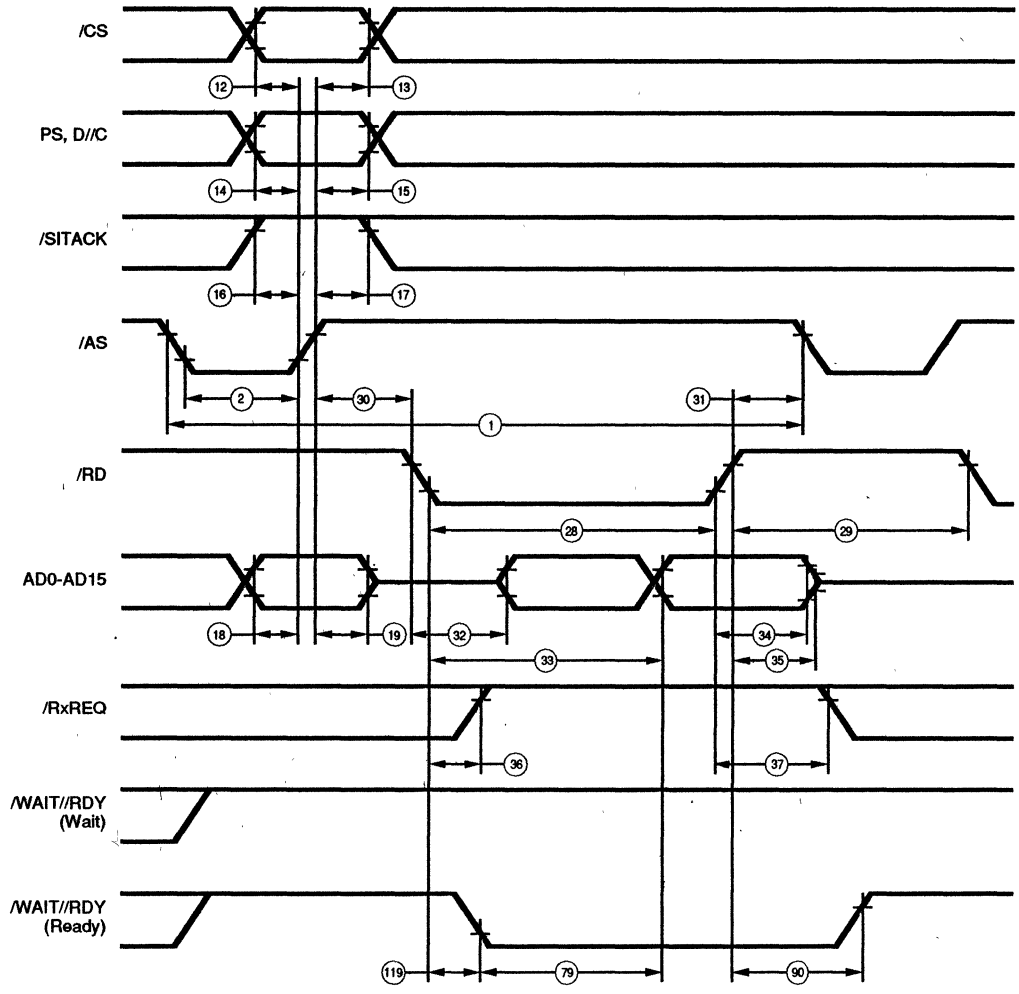


Figure 61. Multiplexed /RD Read Cycle

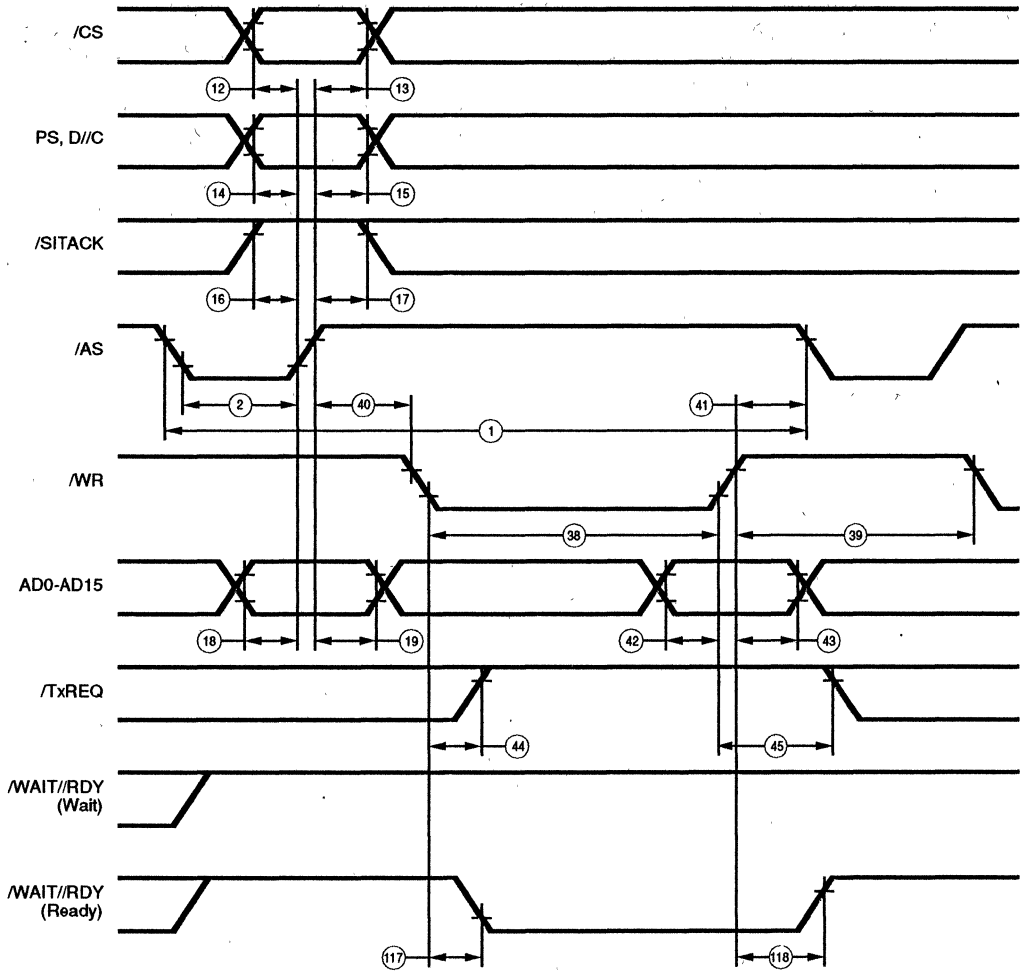


Figure 62. Multiplexed /WR Write Cycle

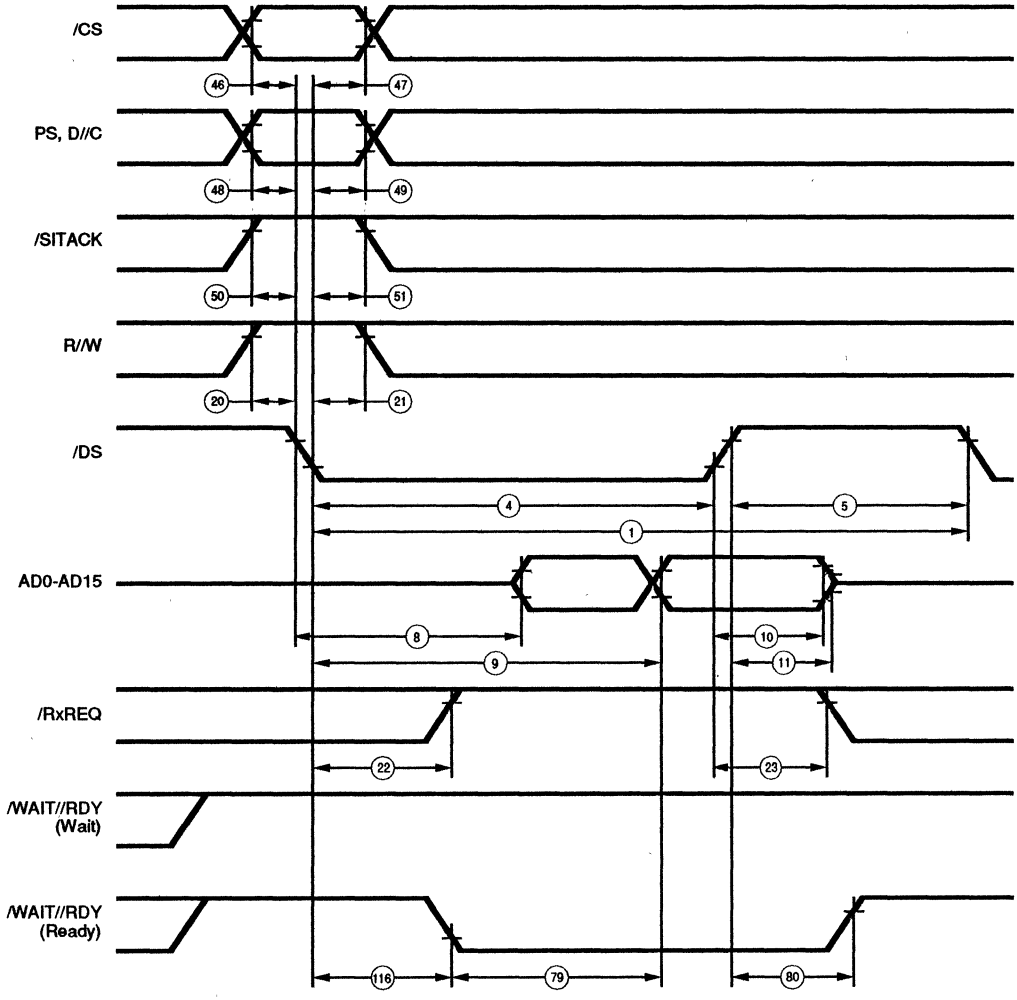


Figure 63. Non-Multiplexed /DS Read Cycle

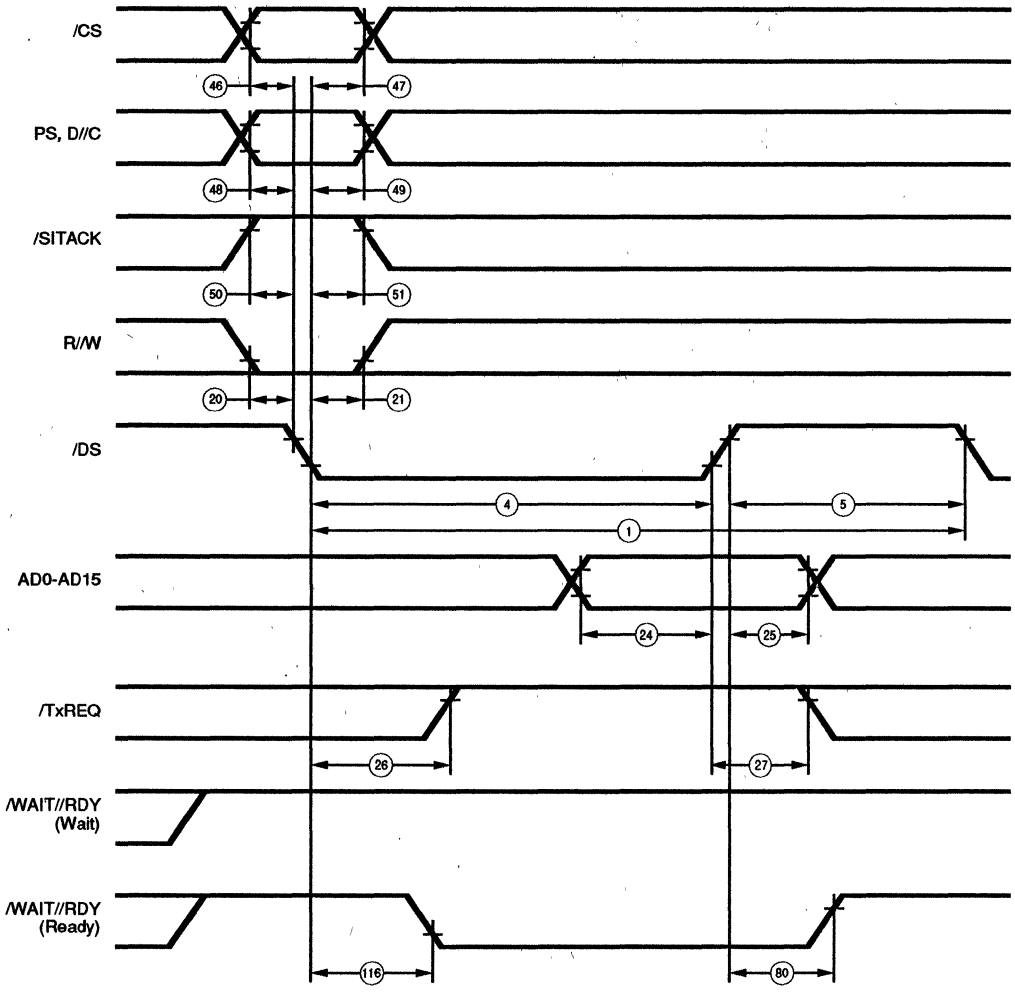


Figure 64. Non-Multiplexed /DS Write Cycle

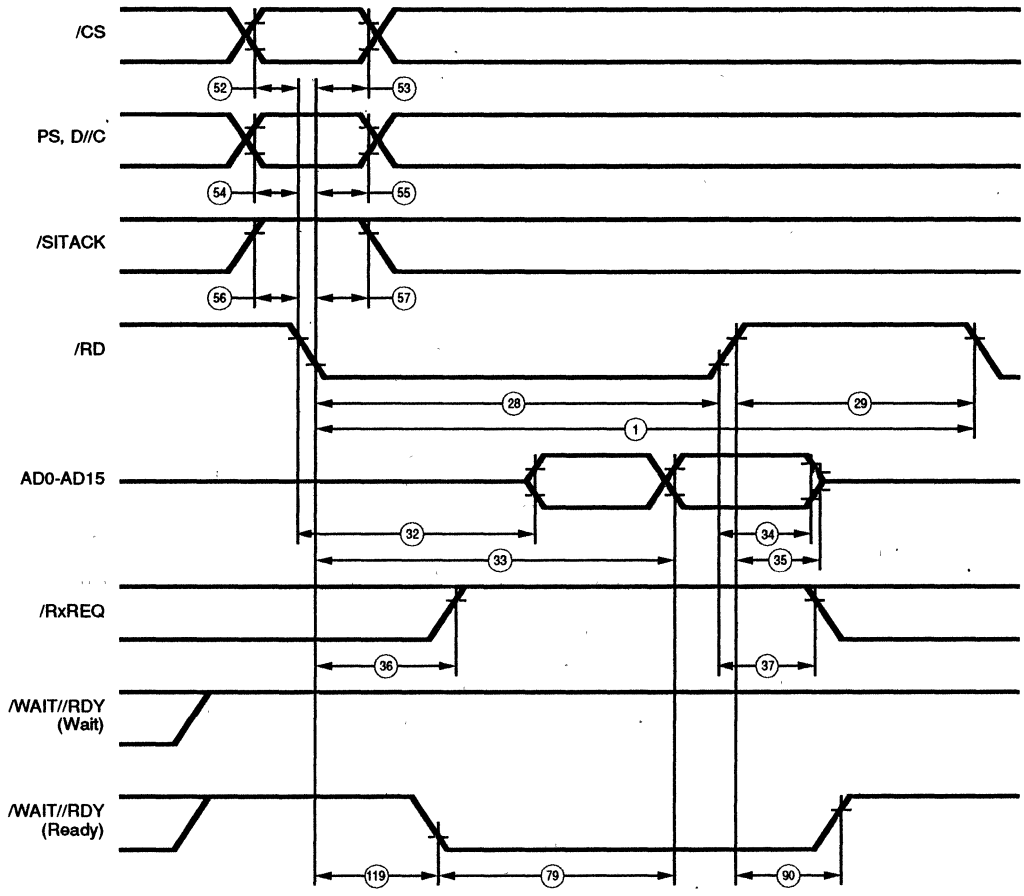


Figure 65. Non-Multiplexed /RD Read Cycle

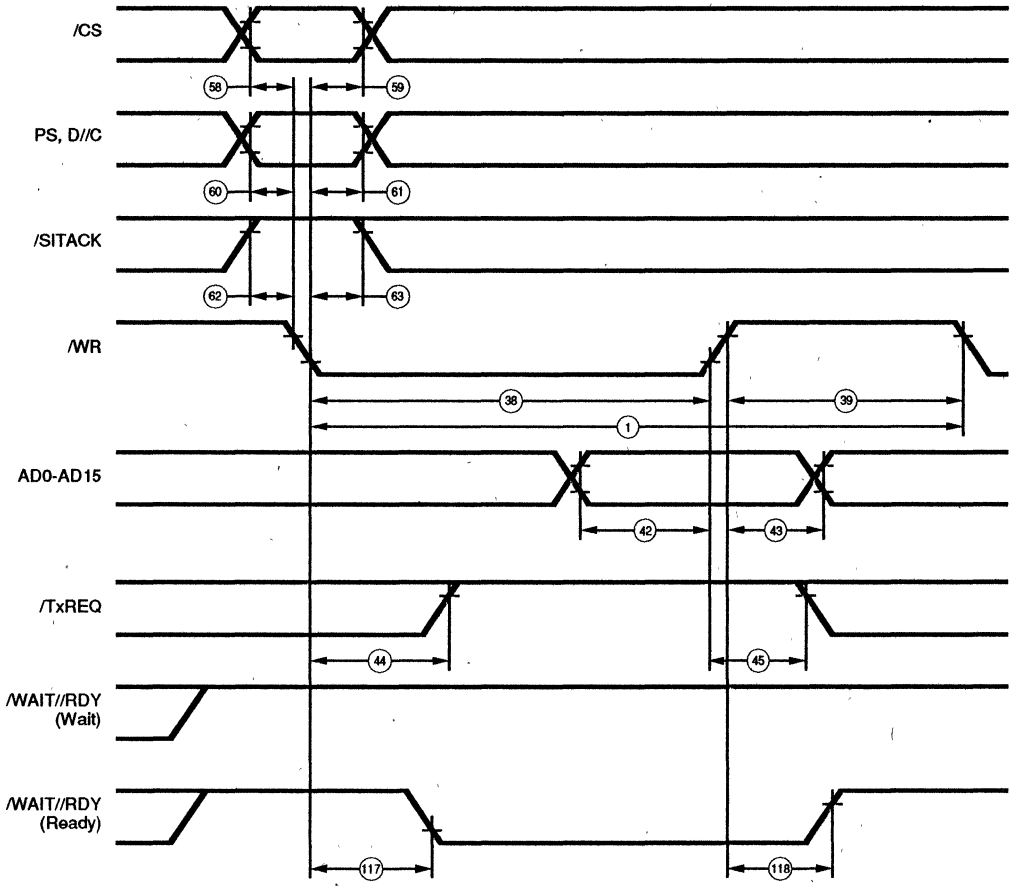


Figure 66. Non-Multiplexed /WR Write Cycle

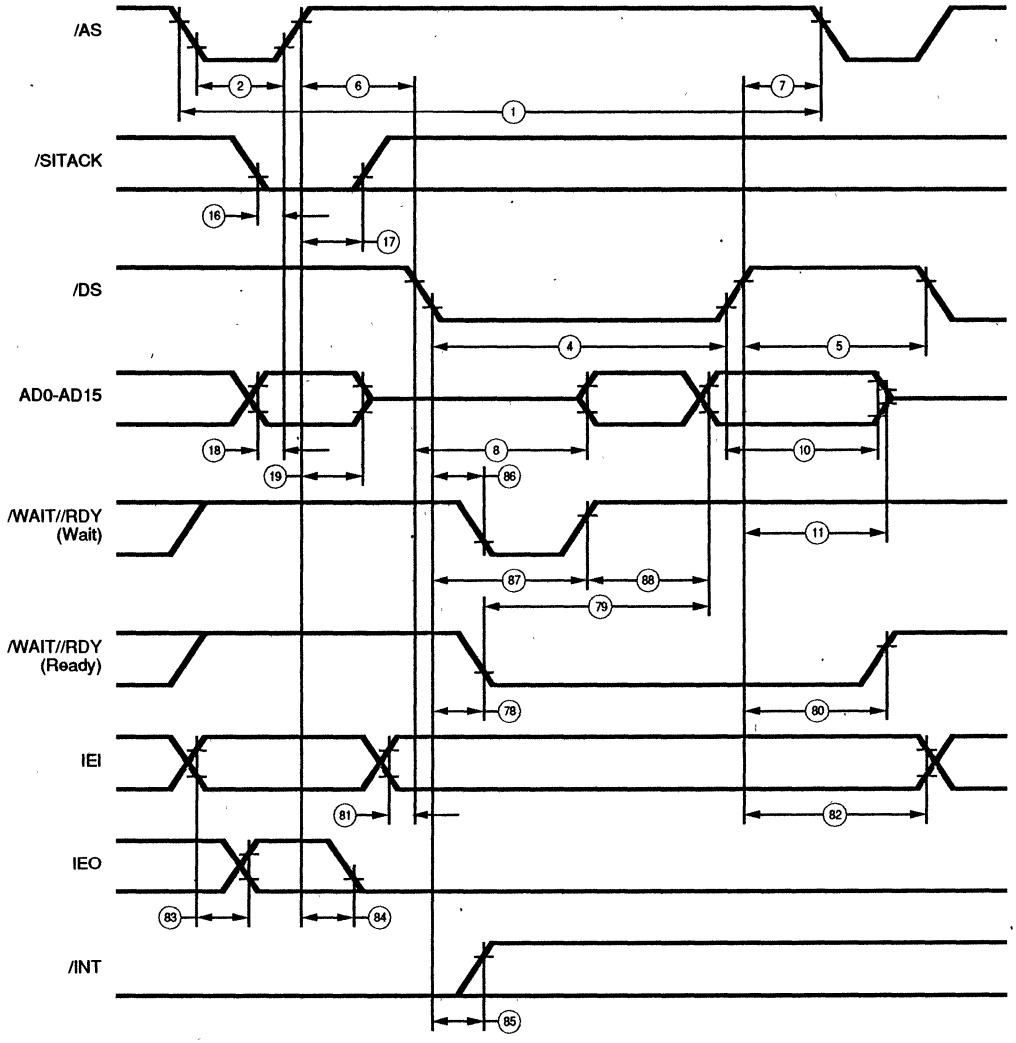


Figure 67. Multiplexed /DS Interrupt Acknowledge Cycle

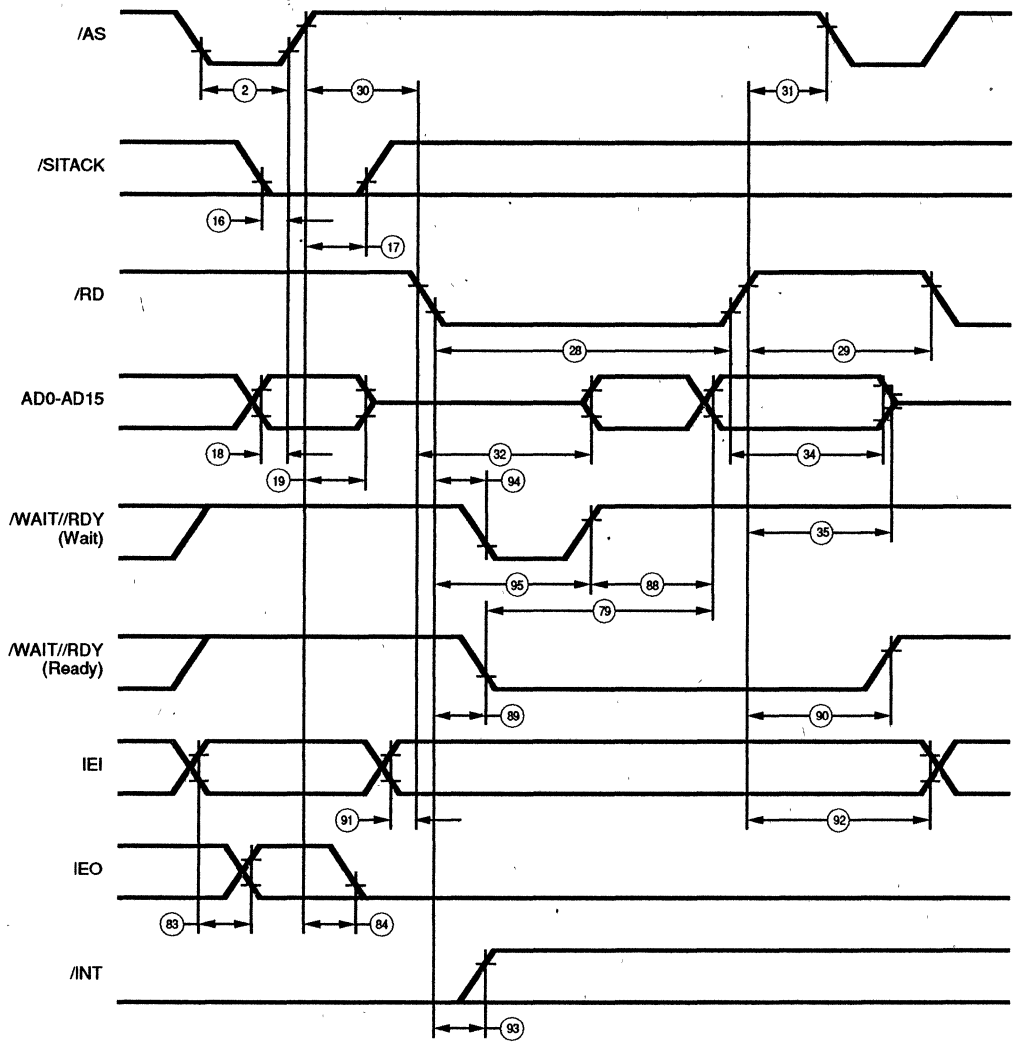


Figure 68. Multiplexed /RD Interrupt Acknowledge Cycle

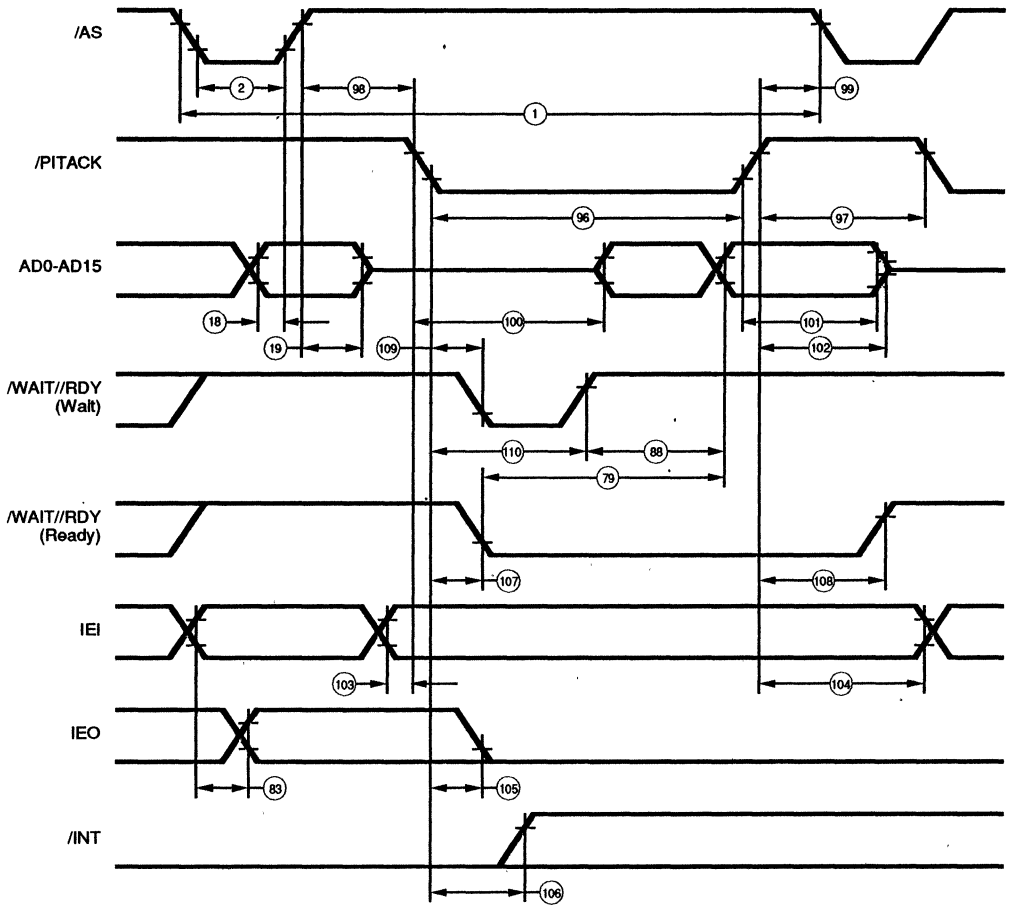


Figure 69. Multiplexed Pulsed Interrupt Acknowledge Cycle

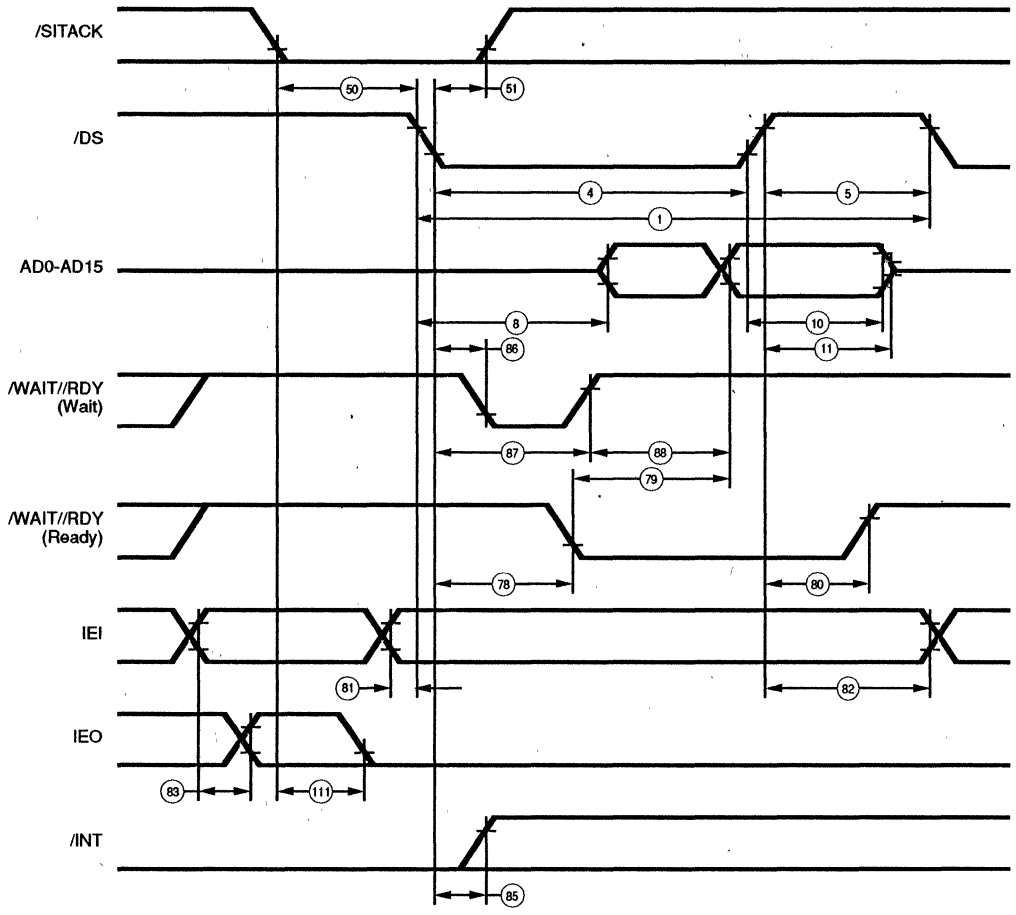


Figure 70. Non-Multiplexed /DS Interrupt Acknowledge Cycle

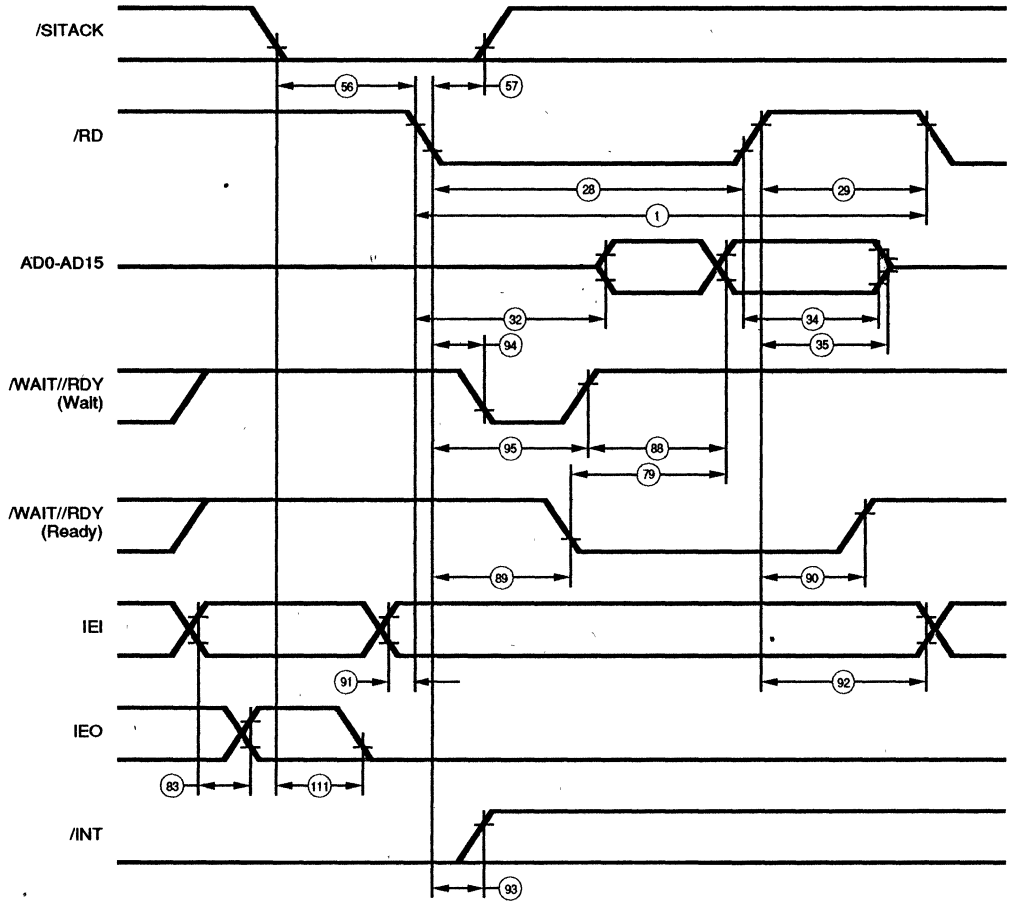


Figure 71. Non-Multiplexed /RD Pulsed Interrupt Acknowledge Cycle

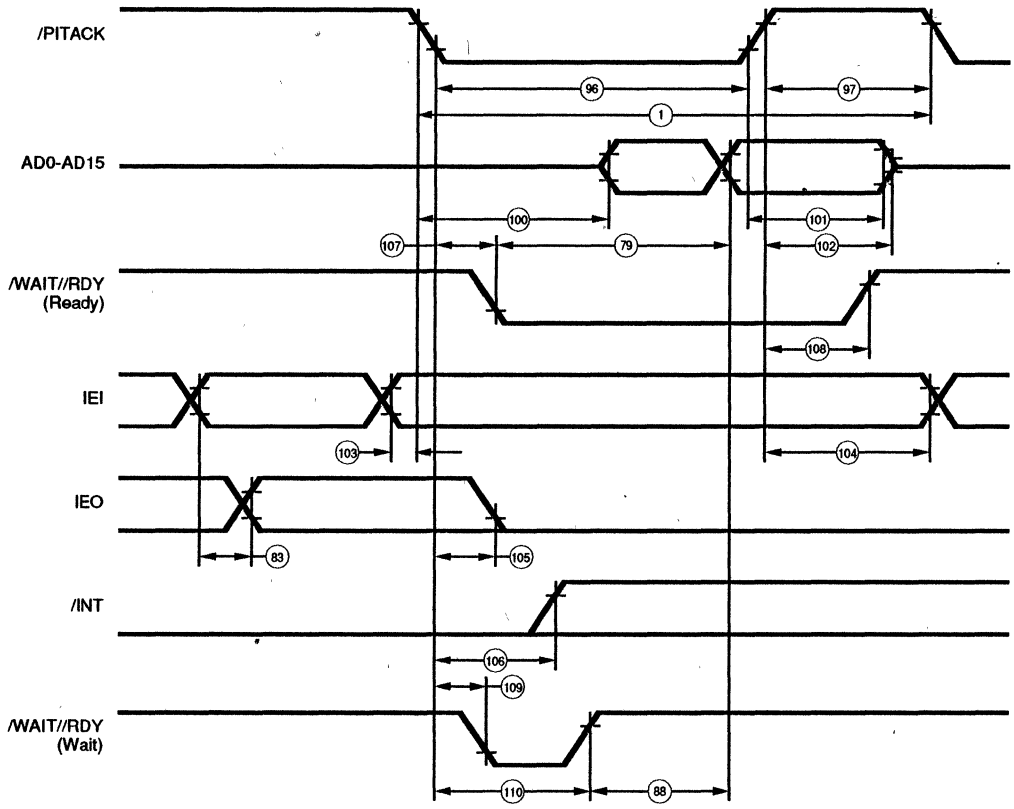


Figure 72. Non-Multiplexed Pulsed Interrupt Acknowledge Cycle

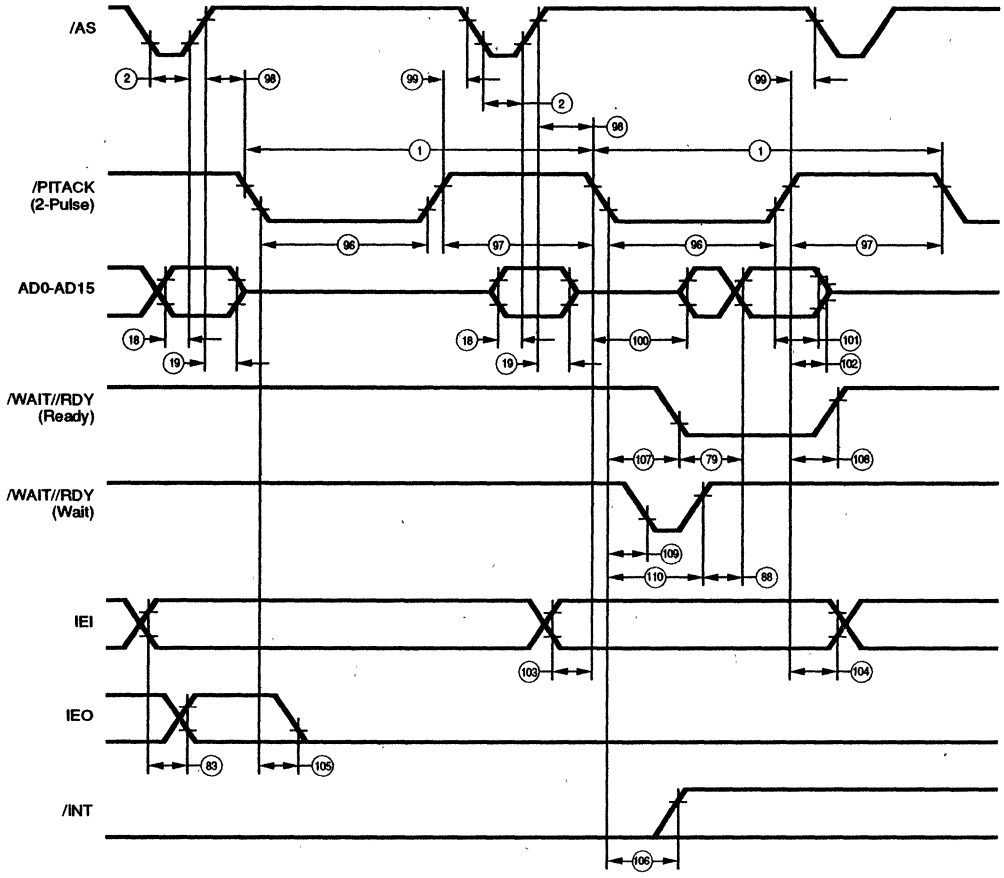


Figure 73. Multiplexed Double-Pulse Intack Cycle

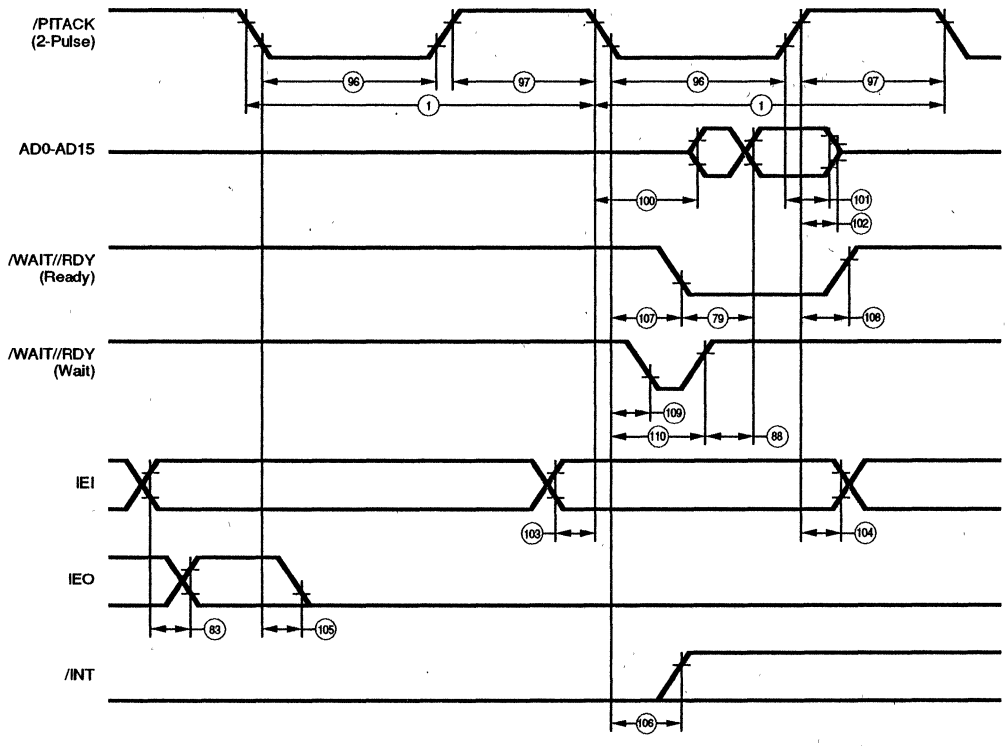


Figure 74. Non-Multiplexed Double-Pulse Intack Cycle

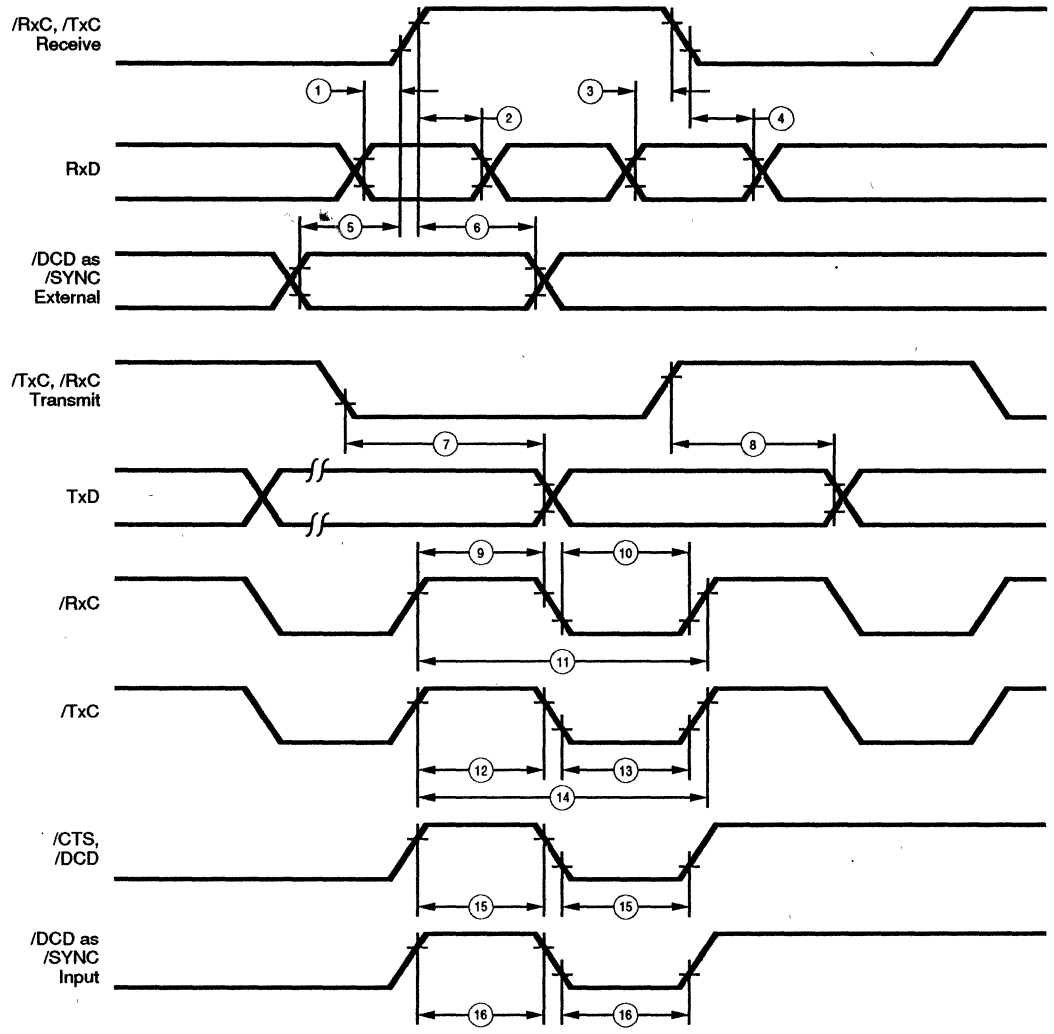


Figure 75. Z16C33 General Timing

AC CHARACTERISTICS

Z16C33 General Timing

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)	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	
10	TwRxCl	/RxC Low Width	40		ns	
11	TcRxC	/RxC Cycle Time	100		ns	
12	TwTxCh	/TxC High Width	40		ns	
13	TwTxCl	/TxC Low Width	40		ns	
14	TcTxC	/TxC Cycle Time	100		ns	
15	TwExT	/DCD or /CTS Pulse Width	70		ns	
16	TWSY	/DCD as /SYNC Input Pulse Width	70		ns	

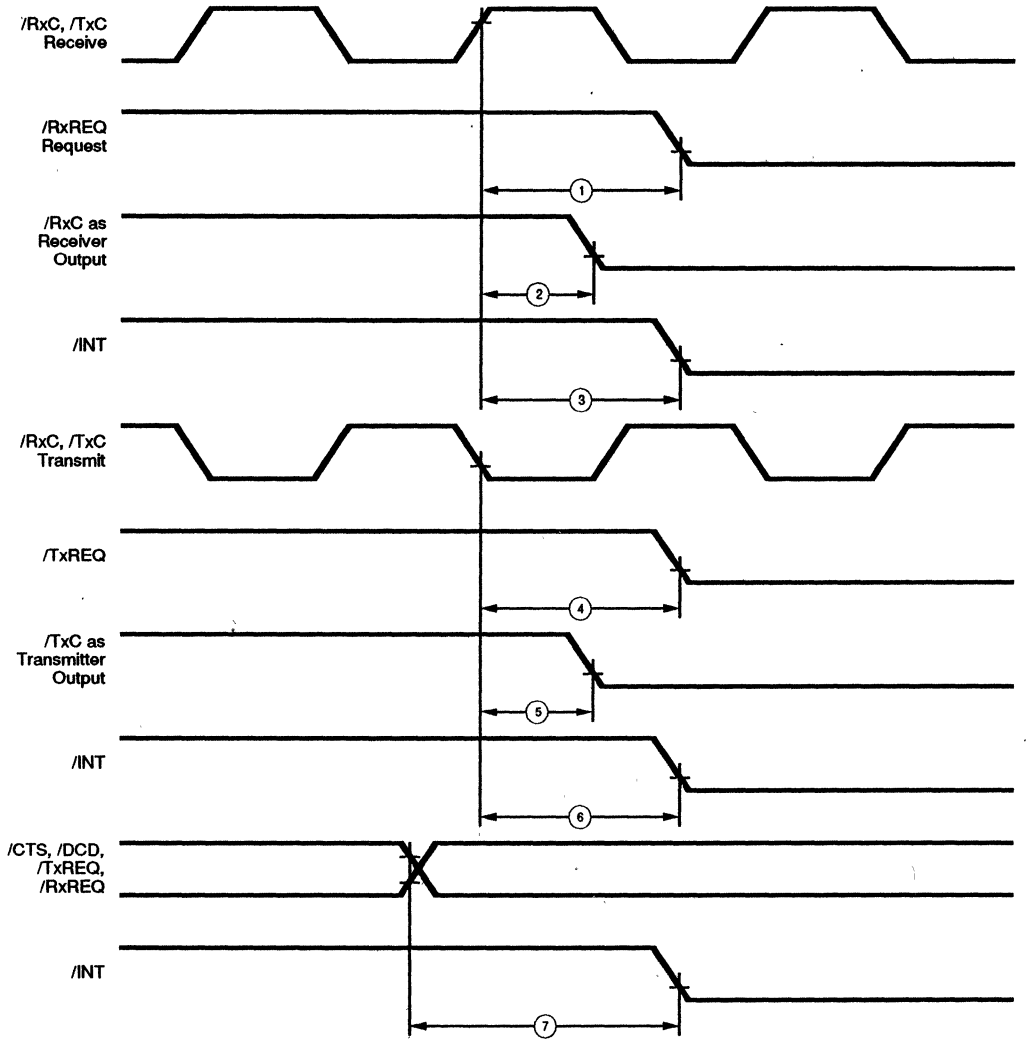


Figure 76. Z16C33 System Timing

AC CHARACTERISTICS

Z16C33 System Timing

No	Symbol	Parameter	Min	Max	Units	Note
1	TdRxC(REQ)	/RxC Rise to /RxREQ Valid Delay		100	ns	[2]
2	TdRxC(RxC)	/TxC Rise to /RxC as Receiver Output Valid Delay		100	ns	[2]
3	TdRxC(INT)	/RxC Rise to /INT Valid Delay		100	ns	[2]
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	
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.

[3] Parameter applies only to FM encoding/decoding

PRELIMINARY PRODUCT SPECIFICATION

Z16C35

CMOS ISCC™ INTEGRATED SERIAL COMMUNICATIONS CONTROLLER

FEATURES

- Low power CMOS technology
- Two general-purpose SCC channels, four DMA channel; and a Universal Bus Interface Unit.
- Software compatible to the Zilog CMOS SCC
- Four DMA channels; two transmit and two receive channels to and from the SCC.
- Four gigabyte address range per DMA channel
- Flyby DMA transfer mode
- Programmable DMA channel priorities
- Independent DMA register set
- A Universal Bus Interface Unit providing a simple interface to most CPUs with a multiplexed or non-multiplexed bus; compatible with 680x0 and 8x86 CPUs.
- 32-bit addresses multiplexed to 16-pin address/data lines
- 8-bit data supporting high/low byte swapping
- 10 and 16 MHz timing
- 68-pin PLCC

Supports all Zilog CMOS SCC features:

- Two independent, 0 to 4.0 Mbit/second, full-duplex channels, each with a separate crystal oscillator, baud rate generator, and digital phase-locked loop circuit for clock recovery.
- Multi-protocol operation under program control; programmable for NRZ, NRZI, or FM data encoding.
- Asynchronous mode with five to eight bits and one, one and one-half, or two stop bits per character; programmable clock factor; break detection and generation; parity, overrun, and framing error detection.
- Synchronous mode with internal or external character synchronization on one or two synchronous characters and CRC generation and checking with CRC-16 or CRC-CCITT preset to either 1's or 0's.
- SDLC/HDLC mode with comprehensive frame-level control, automatic zero insertion and deletion, I-field residue handling, abort generation and detection, CRC generation and checking, and SDLC Loop mode operation.
- Local Loopback and Auto Echo modes
- Supports T1 digital trunk
- Enhanced SDLC 10x19 Status FIFO for DMA support
- Full CMOS SCC register set

GENERAL DESCRIPTION

The Z16C35 ISCC is a CMOS superintegrated device with a flexible Bus Interface Unit (BIU) connecting a built-in Direct Memory Access (DMA) cell to the CMOS Serial Communications Control (SCC) cell.

The ISCC is a dual-channel, multi-protocol data communications peripheral which easily interfaces to CPU's with either multiplexed or non-multiplexed address and data buses. The advanced CMOS process offers lower power consumption, higher performance, and superior noise immunity. The programming flexibility of the internal registers allow the ISCC to be configured for a wide variety of serial communications applications. The many on-chip features such as, streamlined bus interface, four channel

GENERAL DESCRIPTION (Continued)

DMA, baud rate generators, digital phase-locked loops, and crystal oscillators dramatically reduce the need for external logic. Additional features, including a 10x19 bit status FIFO, are added to support high speed SDLC transfers using on-chip DMA controllers (Figure 1).

The ISCC can address up to four gigabytes per DMA channel by using the /UAS and /AS signals to strobe out 32-bit multiplexed addresses.

The ISCC handles asynchronous formats, synchronous byte-oriented protocols such as IBM Bisync, and synchronous bit-oriented protocols such as HDLC and IBM SDLC. This versatile device supports virtually any serial data transfer application (terminals, printers, diskette, tape drives, etc.).

The device can generate and check CRC codes in any synchronous mode and can be programmed to check data integrity in various modes. The ISCC also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls can be used for general-purpose I/O.

The standard Zilog interrupt daisy chain is supported for interrupt hierachy control. Internally, the SCC cell has higher interrupt priority than the DMA cell.

The DMA cell consists of four DMA channels; one for transmit and one for receive to and from each SCC channel, respectively. The cycle time for each DMA transfer is 400 ns for the 10 MHz version. There is no idle cycle between DMA transfers.

The DMA cell adopts a simple fly-by mode DMA transfer, allowing easy programming of the DMA cell and yet providing a powerful and efficient DMA access. The cell does not support memory-to-memory transfer.

Priorities between the four DMA channels are programmable to custom-fit user applications. Arbitration of Bus priority control signals between the ISCC DMA and other system DMA's should be handled outside the ISCC.

The BIU has a universal interface to most system/CPU bus structures and timing. The first write to the ISCC after a hardware reset will confirm the bus interface type being implemented.

Note: 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); /N//S (NORMAL and SYSTEM are both active Low).

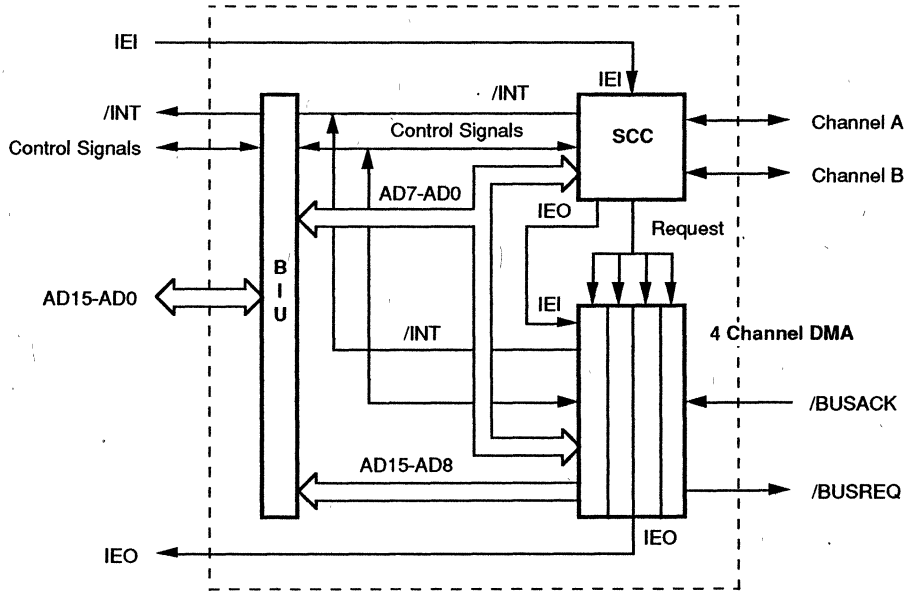


Figure 1. Block Diagram

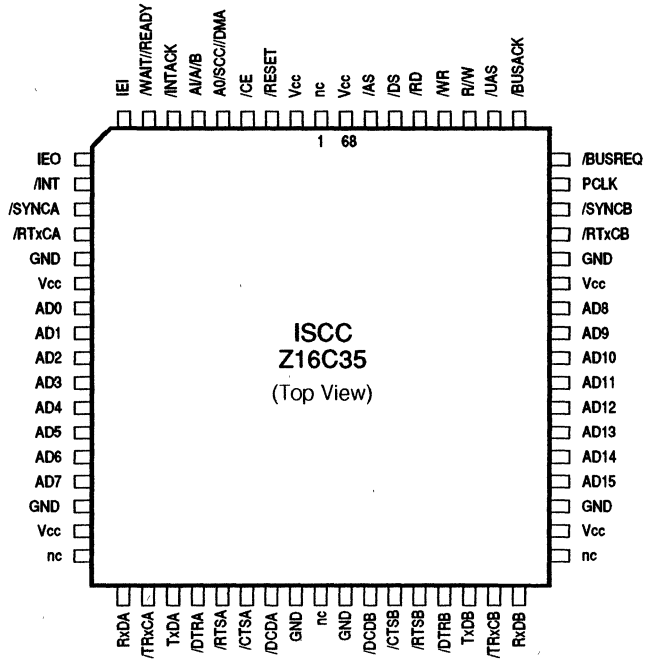


Figure 2. Pin Assignments

PIN DESCRIPTION

The following section describes the Z16C35 pin functions. Figures 2 details the respective pin functions and pin assignments. All references to DMA are internal.

/CTSA, /CTSB. *Clear To Send* (inputs, active Low). If these pins are programmed as Auto Enables, a Low on the inputs enables the respective transmitters. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The SCC cell detects pulses on these inputs and can interrupt the CPU on both logic level transitions.

/DCDA, /DCDB. *Data Carrier Detect* (inputs, active Low). These pins function as receiver enables if they are programmed for Auto Enables; otherwise they are used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accommodate slow rise time signals. The SCC cell detects pulses on these pins and can interrupt the CPU on both logic level transitions.

/DTRA, /DTRB. *Data Terminal Ready* (outputs, active Low). These outputs follow the state programmed into the DTR bit.

IEI. *Interrupt Enable In* (input, active High). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt driven device. A high IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt. The SCC cell has a higher interrupt priority than the DMA cell.

IEO. *Interrupt Enable Out* (output, active High) IEO is High only if IEI is High and the CPU is not servicing the ISCC (SCC or DMA) interrupt, or the ISCC is not requesting an interrupt (Interrupt Acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.

/INT. *Interrupt* (output, active Low). This signal is activated when the SCC or DMA requests an interrupt. Note that /INT is pulled high and is not an open-drain output.

PIN DESCRIPTION (Continued)

/INTACK. *Interrupt Acknowledge* (input, active Low) This is a strobe which indicates that an interrupt acknowledge cycle is in progress. During this cycle, the SCC and DMA interrupt daisy chain is resolved. The device is capable of returning an interrupt vector that may be encoded with the type of interrupt pending during this acknowledge cycle when RD or DS become high. INTACK may be programmed to accept a status acknowledge, a single pulse acknowledge, or a double pulse acknowledge. This is programmed in the Bus Configuration Register (BCR). The double pulse acknowledge is compatible with 8x86 family microprocessors.

PCLK. *Clock* (input). This is the master SCC and DMA clock used to synchronize internal signals. PCLK is a TTL level signal. PCLK is not required to have any phase relationship with the master system clock.

RxDA, RxDB. *Receive Data* (inputs, active High). These input signals receive serial data at standard TTL levels.

/RTxCA, /RTxCB. *Receive/Transmit Clocks* (inputs, active Low). These pins can be programmed to several modes of operation. In each channel, RTxC may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the Digital Phase-Locked Loop. These pins can also be programmed for use with the respective SYNC pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.

/RTSA, /RTSB. *Request To Send* (outputs, active I ow) When the Request To Send (RTS) bit in Write Register 5 is set, the RTS signal goes Low. When the RTS bit is reset in the Asynchronous mode and Auto Enable is on, the signal goes High after the transmitter is empty. In Synchronous mode or in Asynchronous mode with Auto Enable off, the RTS pin strictly follows the state of the RTS bit Both pins can be used as general-purpose outputs.

/SYNCA, /SYNCB. *Synchronization* (inputs or outputs, active Low). These pins can act either as inputs, outputs, or part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to CTS and DCD. In this mode, transitions on these lines affect the state of the Synchronous/Hunt status bits in Read Register 0 but have no other function.

In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, SYNC must be driven Low to receive clock cycles after the last bit in the synchronous character is received Character assembly begins on the rising edge of the receive clock immediately preceding the activation of SYNC.

In the Internal Synchronization mode (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which synchronous condition is not latched These outputs are active each time a synchronization pattern is recognized (regardless of character boundaries). In SDLC mode, the pins act as outputs and are valid on receipt of a flag.

TxDA, TxDB. *Transmit Data* (outputs, active high). These output signals transmit serial data at standard TTL levels.

/TRxCA, /TRxCB. *Transmit/Receive Clocks* (inputs or outputs, active Low). These pins can be programmed in several different modes of operation. TRxC may supply the receive clock or the transmit clock in the input mode or supply the output of the Digital Phase-Locked Loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.

/CE. *Chip Enable* (input, active Low) This signal selects the ISCC for a peripheral read or write operation. This signal is not used when the ISCC is bus master.

AD15-AD0. *Data bus* (bidirectional, 3-state). These lines carry data and commands to and from the ISCC.

/RD. *Read* (bidirectional, active Low). When the ISCC is a peripheral (i.e. bus slave), this signal indicates a read operation and when the ISCC is selected, enables the ISCC's bus drivers. As an input, /RD indicates that the CPU wants to read from the ISCC read registers. During the Interrupt Acknowledge cycle, /RD gates the interrupt vector onto the bus if the ISCC is the highest priority device requesting an interrupt When the ISCC is the bus master, this signal is used to read data. As an output, after the ISCC has taken control of the system buses, /RD indicates a DMA-controlled read from a memory or I/O port address.

/WR. *Write* (bidirectional, active Low) When the ISCC is selected, this signal indicates a write operation As an input, this indicates that the CPU wants to write control or command bytes to the ISCC write registers. As an output, after the ISCC has taken control of the system buses /WR indicates a DMA-controlled write to a memory or I/O port address.

/DS. *Data Strobe* (bidirectional, active Low). A Low on this signal indicates that the AD15-AD0 bus is used for data transfer. When the ISCC is not in control of the system bus and the external system is transferring information to or from the ISCC, /DS is a timing input used by the ISCC to move data to or from the AD15-AD0 bus Data is written into the ISCC by the external system on the Low to High /DS transition. Data is read from the ISCC by the external

system while /DS is Low There are no timing requirements between /DS as an input and ISCC clock, this allows use of the ISCC with a system bus which does not have a bussed clock.

During a DMA operation when the ISCC is in control of the system, DS is an output generated by the ISCC and used by the system to move data to or from the AD15-AD0 bus When the ISCC has bus control, it writes to the external system by placing data on the AD15-AD0 bus before the High-to-Low DS transition and holds the data stable until after the Low-to-High DS transition; while reading from the external system, the Low-to-High transition of DS inputs data from the AD15-AD0 bus into the ISCC.

R//W. *Read/Write* (bidirectional) Read polarity is High and write polarity is Low. When the ISCC is bus master, R//W indicates the data direction of the current bus transaction, and is stable from when AS is High until the bus transaction ends When the ISCC is not in control of the system bus and the external system is transferring information to or from the ISCC, R//W is a status input used by the ISCC to determine if data is entering or leaving on the AD15-AD0 bus during /DS time. In such a case, Read (High) indicates that the system is requesting data from the ISCC and Write (Low) indicates that the system is presenting data to the ISCC. The only timing requirements for R//W as an input are defined relative to DS When the ISCC is in control of the system bus, R//W is an output generated by the ISCC, with Read indicating that data is being requested from the addressed location or device, and Write indicating that data is being presented to the addressed location or device.

/UAS. *Upper Address Strobe* (Output, active Low) This signal is used if the address is more than 16-bit The upper address, A31-A16, can be latched externally by the rising edge of this signal. /UAS is active first before AS becomes active. This signal and AS are used by the DMA cell

/AS. *Lower Address Strobe* (Bidirectional, active Low) When the ISCC is bus master, this signal when an output, is used as a lower address strobe for AD15-AD0 It is used in conjunction with UAS since the address is 32-bits This signal and /UAS are used by the DMA cell when it is bus master. When ISCC is not bus master, 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 Vcc in these cases.

/WAIT//RDY. *Wait/Ready* (bidirectional, active Low) It may be programmed to function either as a Wait signal or Ready signal during the BCR write When the BCR is written to Channel A (A1/A//B High during the BCR write), this signal functions as a WAIT and thus supports the READY function of 8X86 microprocessors family When the BCR writes to Channel B (A1/A//B Low), this signal functions as a READY and supports the DTACK function of the 680X0 microprocessor family.

This signal is an output when the ISCC in not bus master In this case, the Wait/RDY signal indicates 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.

When the ISCC is the bus master (the DMA cell has taken control of the bus), the /Wait//RDY signal functions as a WAIT or READY input. Slow memories and peripheral devices can assert WAIT to extend /DS during bus transfers Similarly, memories and peripherals use READY to indicate that its output is valid or that it is ready to latch input data.

/BUSACK. *Bus Acknowledge* (input, active Low) Signals the bus has been released to the DMA If the /BUSACK is inactive before the DMA transfer is completed, the current DMA transfer is aborted.

/BUSREQ. *Bus Request* (output, active Low) This signal is used by the DMA to obtain the bus from the CPU

A0/SCC//DMA. *DMA Channel/SCC Select/DMA Select* (bidirectional) When this pin is used as input, a high selects the SCC cell and a low selects the DMA cell When this pin is used as output, the signal on this pin is used in conjunction with A1/A//B pin output to identify which DMA channel is active This information can be used by the user to determine whether to issue a DMA abort command A0/SCC//DMA and A1/A//B output encoding is shown below

A1/A//B	A0/SCC//DMA	DMA channel
1	1	RxA
1	0	TxA
0	1	RxB
0	0	TxB

A1/A//B. *DMA Channel/Channel A/Channel B* (bidirectional) This signal, when used as input, selects the SCC channel in which the read and write operation occurs Note that A0/SCC//DMA pin must be held high to select this feature. When this pin is used as an output, it is used in conjunction with the A0/SCC//DMA pin output to identify which DMA channel is active During a DMA peripheral access, the A1/A//B pin is ignored.

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

FUNCTIONAL DESCRIPTION

The functional capabilities of the ISCC are described in three blocks: the SCC cell, the DMA cell, and the Bus Interface Unit (BIU). Each of the blocks are described independently in the following sections with the ISCC architecture shown in Figure 3. Please refer to the ISCC Technical Manual for a detailed description of the functions outlined here.

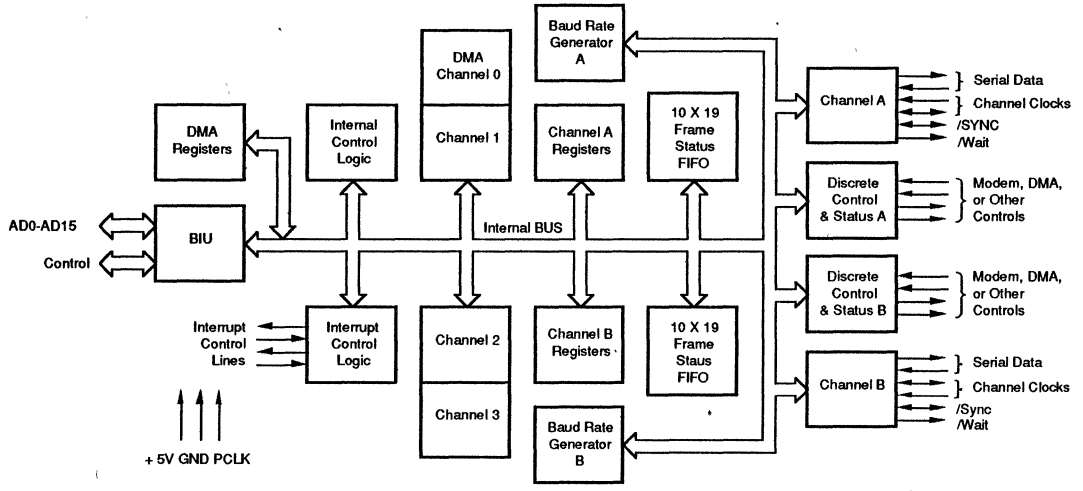


Figure 3. Block Diagram of ISCC Architecture

SCC Cell Data Communications Capabilities. The ISCC provides two independent full-duplex programmable channels for use in any common asynchronous or synchronous data communications protocol. The ISCC is built from Zilog's industry standard SCC core and is compatible with designs using Zilog's SCC to receive and transmit data (Figure 4).

Asynchronous Modes. Send and Receive can be accomplished independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one-and-a-half, or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break. Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxDA or RxDB in Figure 2). If the Low does not persist (e.g., a transient), the character assembly process does not start.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occur. Vectored interrupts allow fast servicing or error conditions using dedicated routines. Furthermore, a built-in checking process avoids the interpretation of a framing error as a new start bit: a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit begins.

The ISCC does not require symmetric transmit and receive clock signals - a feature allowing use of the wide variety of clock sources. The transmitter and receiver can handle data at a rate supplied to the receive and transmit clock inputs. In Asynchronous modes, the SYNC pin may be programmed as an input used for functions such as monitoring a ring indicator.

Synchronous Modes. The ISCC supports both byte-oriented and bit-oriented synchronous communication. Synchronous byte-oriented protocols can be handled in several modes, allowing character synchronization with a 6-bit or 8-bit synchronous character (Monosync), and 12-bit synchronization pattern (Bisync), or with an external synchronous signal Leading sync characters can be removed without interrupting the CPU.

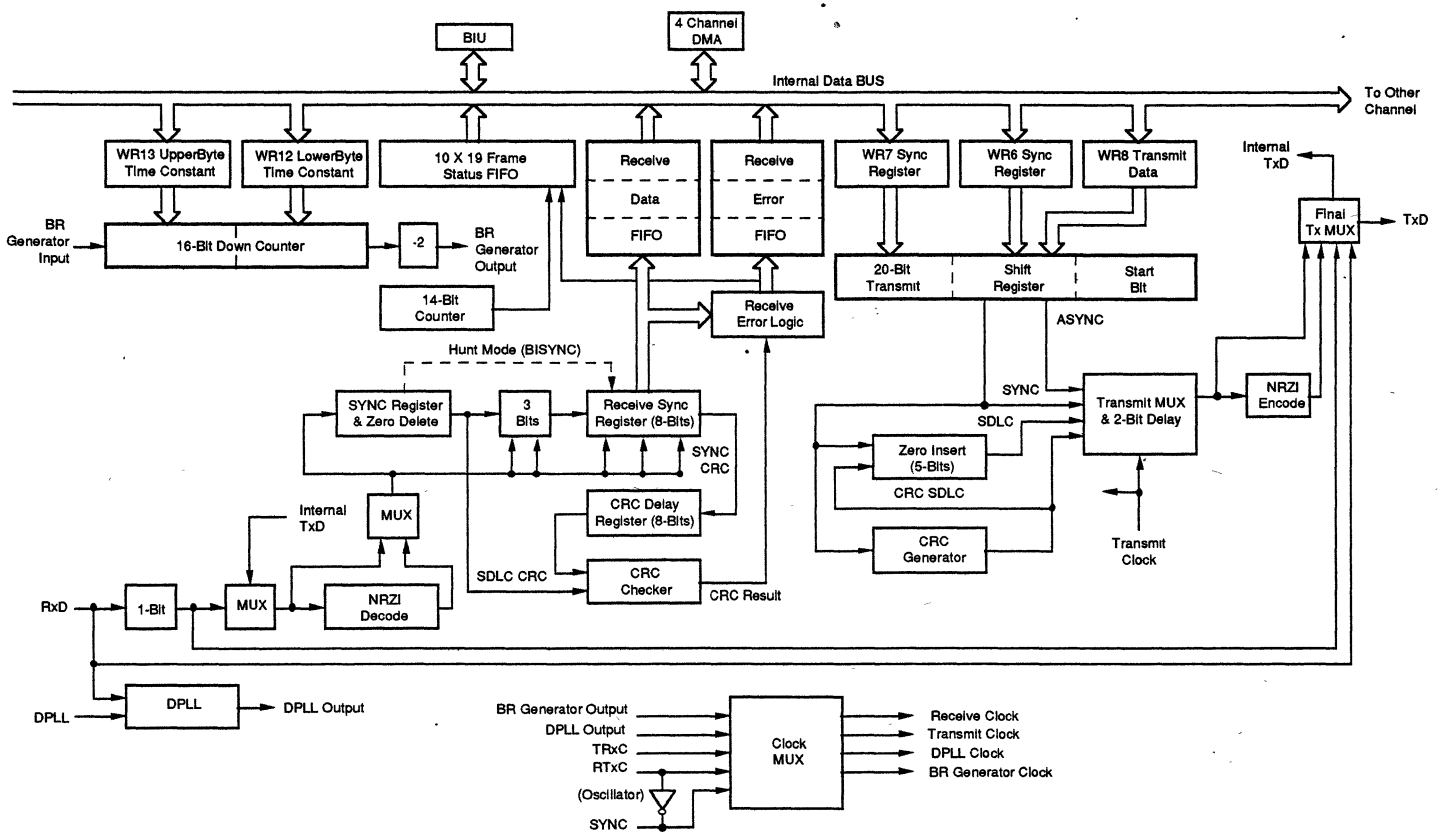


Figure 4. ISCC Data Path

FUNCTIONAL DESCRIPTION (Continued)

Five or 7-bit synchronous characters are detected with 8- or 16-bit patterns in the ISCC by overlapping the larger pattern across multiple incoming synchronous characters as shown in Figure 5.

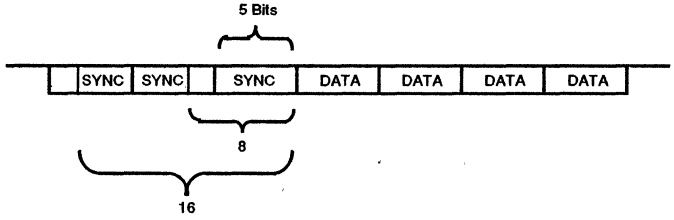


Figure 5. Detecting 5- or 7-Bit Synchronous Characters

CRC checking for Synchronous byte oriented modes is delayed by one character time so that the CPU may disable CRC checking on specific characters. This permits the implementation of protocols such as IBM Bisync

Both CRC-16 (X16 + X15 + X2 +1) and CCITT (X16 + X12 + X5 +1) error checking polynomials are supported. Either polynomial may be selected in all Synchronous modes. Users may preset the CRC generator and checker to all 1's or all 0's. The ISCC also provides a feature that automatically transmits CRC data when no other data is available for transmission. This allows for high speed transmissions under DMA control, with no need for CPU intervention at the end of a message. When there is no data or CRC to send in Synchronous modes, the transmitter inserts 6-, 8-, or 16-bit synchronous characters, regardless of the programmed character length.

The ISCC supports Synchronous bit-oriented protocols, such as SDLC and HDLC, by performing automatic flag sending, zero insertion, and CRC generation. A special command is used to abort a frame in transmission At the end of a message, the ISCC automatically transmits the CRC and trailing flag when the transmitter underruns The transmitter may also be programmed to send an idle line consisting of continuous flag characters or a steady marking condition.

If a transmit underrun occurs in the middle of a message, an external/status interrupt warns the CPU of this status change so that an abort may be issued. The ISCC may also be programmed to send an abort itself in case of an underrun, relieving the CPU of this task. One to eight bits per character can be sent, allowing reception of a message with no prior information about the character structure in the information field of a frame.

The receiver automatically acquires synchronization on the leading flag of a frame in SDLC or HDLC and provides a synchronization signal on the SYNC pin (an interrupt can also be programmed). The receiver can be programmed to search for frames addressed by a single byte (or four bits within a byte) of a user-selected address or to a global broadcast address. In this mode, frames not matching either the user-selected or broadcast address are ignored. The number of address bytes can be extended under software control. For receiving data, an interrupt on the first received character, or an interrupt on every character, or on special condition only (end-of-frame) can be selected The receiver automatically deletes all 0's inserted by the transmitter during character assembly. CRC is also calculated and is automatically checked to validate frame transmission. At the end of transmission, the status of a received frame is available in the status registers. In SDLC mode, the ISCC must be programmed to use the SDLC CRC polynomial, but the generator and checker may be preset to all 1's or all 0's. The CRC is inverted before transmission and the receiver checks against the bit pattern 0001110100001111.

NRZ, NRZI or FM coding may be used in any 1x mode. The parity options available in Asynchronous modes are available in Synchronous modes.

SDLC Loop Mode. The ISCC supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, there is a primary controller station that manages the message traffic flow on the loop and any number of secondary stations In SDLC Loop mode, the ISCC performs the functions of a secondary station while an ISCC operating in regular SDLC mode acts as a controller (Figure 6).

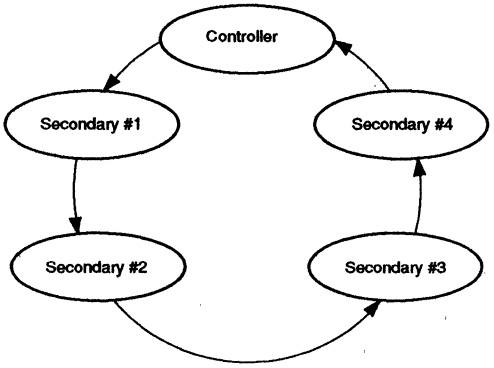


Figure 6. An SDLC Loop

A secondary station in an SDLC Loop is always listening to the messages being sent around the loop and, in fact, passes these messages to the rest of the loop by retransmitting them with a one-bit-time delay. The secondary station places its own message on the loop only at specific times. The controller signals that secondary stations can transmit messages by sending a special character, called an EOP (End Of Poll), around the loop. The EOP character is the bit pattern 11111110. Because of zero insertion during messages, this bit pattern is unique and easily recognized.

When a secondary station has a message to transmit and recognizes an EOP on the line, it changes the last binary 1 of the EOP to a 0 before transmission. This has the effect of turning the EOP into a flag sequence. The secondary station now places its message on the loop and terminates the message with an EOP. Any secondary stations further down the loop with messages to transmit appends their messages to the message of the first secondary station by the same process. Any secondary stations without messages to send merely echo the incoming message and are prohibited from placing messages on the loop (except upon recognizing an EOP.)

SDLC Loop mode is a programmable option in the ISCC. NRZ, NRZI, and FM coding may all be used in SDLC Loop mode

SDLC FIFO. The ISCC's ability to receive high speed back-to-back SDLC frames is maximized by a 10-bit deep by 19-bit wide status FIFO. When enabled (through WR15, bit D2), it provides the DMA the ability to continue to transfer data into memory so that the CPU can examine the message later. For each SDLC frame, a 14-bit byte count and 5 status/error bits are stored. The byte count and status bits are accessed through Read Registers 6 and 7. Read Registers are only accessible when the SDLC FIFO is enabled. The 10x19 status FIFO is separate from the 3 byte receive data FIFO.

Notes on the SDLC FIFO. When using the SDLC FIFO enhancment in channel B, it is necessary to enable the enhancment in channel A. There is no special requirement to enable the enhancement in channel A only , or to use it in both channels. Designs using only one channel should, therefore, use channel A.

When an SDLC frame is received with an abort condition, the byte counter in the FIFO enhancment is not reset. Therefore, after the abort is received, a dummy frame consisting of a flag should be sent by the transmitter. This resets the byte counter for the next frame. The aborted frame has a byte count which includes the byte count of the next dummy frame.

Baud Rate Generator. Each channel in the ISCC contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On startup, the flip-flop on the output is set in a High state, the value in the time constant register is loaded into the counter, and the counter starts counting down The output of the baud rate generator toggles upon reaching 0, the value in the time constant register is loaded into the counter, and the process is repeated. The time constant may be changed at any time, but the new value does not take effect until the next load of the counter.

The output of the baud rate generator may be used as either the transmit clock, the receive clock, or both. It can also drive the Digital Phase-Locked Loop (see next section).

If the receive clock or transmit clock is not programmed to come from the TRxC pin, the output of the baud rate generator may be echoed out via the TRxC pin.

The following formula relates the time constant to the baud rate where PCLK or RTxC is the baud rate generator input frequency in Hertz. The clock mode is 1, 16, 32, or 64, as selected in Write Register 4, bits D6 and D7. Synchronous operation modes should select 1 and Asynchronous should select 16, 32 or 64.

$$\text{Time Constant} = \frac{\text{PCLK or RTxC Frequency}}{2(\text{Baud Rate})(\text{Clock Mode})} - 2$$

Digital Phase-Locked Loop. The ISCC contains a Digital Phase-Locked Loop (DPLL) to recover clock information from a data stream with NRZI or FM encoding. The DPLL is driven by a clock that is nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock is then used as the ISCC receive clock, the transmit clock, or both.

For NRZI encoding, the DPLL counts the 32x clock to create nominal bit times. As the 32x clock is counted, the DPLL is searching the incoming data stream for edges (either 1 to 0, or 0 to 1). Whenever an edge is detected, the DPLL makes a count adjustment (during the next counting cycle), producing a terminal count closer to the center of the bit cell.

For FM encoding, the DPLL still counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream should occur between counts 15 and 16 and between counts 31 and 0. The DPLL looks for edges only during a time centered on the 15 to 16 counting transition.

The 32x clock for the DPLL can be programmed to come from either the RTxC input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the ISCC via the TRxC pin (if this pin is not being used as an input).

Data Encoding. The ISCC may be programmed to encode and decode the serial data in four different ways (Figure 7). In NRZ encoding, a 1 is represented by a High level and a 0 is represented by a Low level. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level. In FM1 (more properly, bi-phase mark), a transition occurs at the beginning of every bit cell. A 1 is represented by an additional transition at the center of the bit cell and a 0 is represented by no additional transition at the center of the bit cell. In FM0 (bi-phase space), a transition occurs at the beginning of every bit cell. A 0 is represented by an additional transition at the center of the bit cell, and a 1 is represented by no additional transition at the center of the bit cell. In addition to these four methods, the ISCC can be used to decode Manchester (bi-phase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0 to 1, the bit is a 0. If the transition is 1 to 0, the bit is a 1.

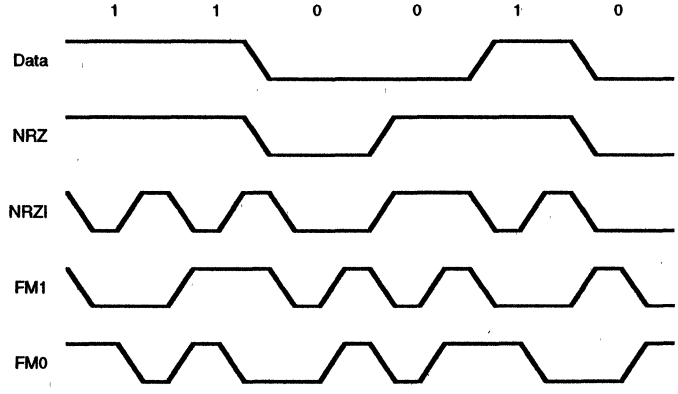


Figure 7. Data Encoding Methods

Auto Echo and Local Loopback. The ISCC is capable of automatically echoing everything it receives. This feature is useful mainly in Asynchronous modes, but works in Synchronous and SDLC modes as well. In Auto Echo mode, TxD is RxD. Auto Echo mode can be used with NRZI or FM encoding with no additional delay because the data stream is not decoded before retransmission. In Auto Echo mode, the /CTS input is ignored as a transmitter enable (although transitions on this input can still cause interrupts if programmed to do so). In this mode, the transmitter is actually bypassed and the programmer is responsible for disabling transmitter interrupts and /WAIT//REQUEST on transmit.

The ISCC is also capable of local loopback. In this mode TxD is RxD is just like Auto Echo mode. However, in Local Loopback mode the internal transmit data is tied to the internal receive data and RxD is ignored (except to be echoed out via TxD). The /CTS and /DCD inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local Loopback works in Asynchronous, Synchronous and SDLC modes with NRZ, NRZI or FM coding of the data stream.

DMA Core. The ISCC contains four independent fly-by mode DMA channels. Each of the ISCC's transmit and receive channels has a DMA channel dedicated to it to move data to-and-from memory. The DMA channels are dedicated to the transmit and receive FIFO's, and therefore, can not be used for device initialization. Each DMA has a 32-bit address and a 16-bit byte counter. The DMA address may be incremented or decremented providing flexibility in doing block transfers.

See the I/O Interface Capabilities Section for more details on the DMA features.

BUS INTERFACE UNIT (BIU) DESCRIPTION

The ISCC contains a flexible bus interface that is compatible with a variety of microprocessors and microcontrollers. The device is designed to work with 8- or 16-bit bus systems and may be used with address/data multiplexed busses or non-multiplexed busses. The multiplexed bus is selected for the ISCC 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.

When the ISCC is initialized for non-multiplexed operation, register addressing for the ISCC cell is (with the exception of WR0 and RR0), accomplished as follows. Programming the write registers requires two write operations and reading the read registers requires both a write and a read operation. The first write is to WR0 which contains four bits that point to the selected register (note point high command). The second write is the actual control word for the selected register. If the second operation is a read, the selected register is accessed. When in the non-multiplexed mode, all of the registers in the SCC cell of the ISCC, including the data registers, are accessed in this fashion. The pointer register is automatically cleared after the second read or write operation so that WR0 (or RR0) is addressed again. Note that when the DMA is not used to address the data, the data registers must be accessed by pointing to Register 8. This is in contrast to the Z8530 which allows direct addressing of the data registers through the C/D pin.

When the ISCC is initialized for non-multiplexed operation, register addressing for the DMA cell (with the exception of CSAR) is accomplished as follows and is completely independent of the SCC cell register addressing. Programming the write registers requires two write operations and reading the read registers requires both a write and a read operation. The first write is to the Command Status Address Register (CSAR) which contains five bits that point to the selected register (CSAR bits 4 - 0). The second write is the actual control word for the selected register. If the second operation is a read, the selected register is accessed. When in the non-multiplexed mode, all of the registers in the DMA cell of the ISCC may be accessed in this fashion. The pointer bits are automatically cleared after the second read or write operation so that CSAR is addressed again.

When the ISCC is initialized for multiplexed bus operation, all registers in the SCC cell are directly addressable with the register address occupying AD5 through AD1, or AD4 through AD0 (Shift Left / Shift Right modes). Two additional pins, A0/SCC//DMA and A1/A//B control the channel A/B register selection and the SCC channel /DMA selection Refer to the A0/SCC//DMA and A1/A//B pin descriptions for the encoding of these signals.

The Shift Left / Shift Right modes for the address decoding for the internal registers (multiplexed bus) are separately programmable for the SCC cell and for the DMA cell. For the SCC cell the programming and operation is identical to that in the SCC; programming is accomplished through Write Register 0 (WR0), bits 1 and 0 (Figure 9-1).

The programming of the Shift Left/Shift Right modes for the DMA cell is accomplished in the BCR, bit 0. In this case, the shift function is similar to that for the SCC cell; with Shift left, the internal register addresses are decoded from bits AD5 through AD1 and with Shift Right, the internal register addresses are decoded from bits AD4 through AD0.

When the multiplexed bus mode is selected, Write Register 0 (WR0) takes on the form of WR0 in the Z8030 (Figure 9).

All data transfers to and from the ISCC are done in bytes even though the data can, at special times, occupy the lower or upper byte of the 16-bit bus. When accessed as a peripheral device (i.e., when the ISCC is not a bus master performing DMA transfers), all bus transactions are on the lower 8 bits of the bus with the following exception

FUNCTIONAL DESCRIPTION (Continued)

When the ISCC registers are read, the byte data is present on both the lower 8 bits of the bus and the upper 8 bits of the bus. Data is accepted only on the lower 8 bits of the bus except in certain DMA transfers.

During DMA transfers, data may be transferred to or from the ISCC on the upper 8 bits of the bus for odd or even byte transfers. During DMA transfers to memory from the ISCC, byte data only is transferred and the data appears on both the lower 8 bits and is replicated on the upper 8 bits of the bus.

During DMA transfers to the ISCC from memory, byte data only is transferred and normally data is accepted only on the lower 8 bits of the bus. However, the byte swapping feature may be used to elect on which byte of the bus the data is accepted. The byte swapping feature is enabled by programming the Byte Swap Enable bit to a 1 in the BCR. The odd/even byte transfer selection is made by programming the Byte Swap Select bit in the BCR. If Byte Swap Select is a 1, then even address bytes (transfers where the DMA address has A0 equal 0) are transferred on the lower 8 bits of the bus and odd address bytes (transfers where the DMA address has A0 equal 1) are transferred on the upper 8 bits of the bus. If Byte Swap Select is a 0, then even address bytes (transfers where the DMA address has A0 equal 0) are transferred on the upper 8 bits of the bus and odd address bytes(transfers where the DMA address has A0 equal 1) are transferred on the lower 8 bits of the bus.

I/O INTERFACE CAPABILITIES

The ISCC offers the choice of Polling, Interrupt (vectored or non-vectored), and DMA Transfer modes to transfer data, status, and control information to and from the CPU.

Polling. In this mode all interrupts and the DMA's are disabled. Three status registers in the SCC are automatically updated whenever any function is performed. For example, end-of-frame in SDLC mode sets a bit in one of these status registers. With polling, the CPU must periodically read a status register until the register contents indicate the need for some CPU action to be taken Only one register in the SCC needs to be read; depending on the contents of the register, the CPU either reads data, writes data, or satisfies an error condition. Two bits in the register indicate the need for data transfer. An alternative is to poll the Interrupt Pending register to determine the source of an interrupt. The status for both SCC channels resides in one register.

Interrupts. When the ISCC responds to an Interrupt Acknowledge signal (INTACK) from the CPU, an interrupt vector is placed on the data bus. Both the SCC and the DMA contain vector registers. Depending on the source of interrupt, one of these vectors is returned. either unmodified or modified by the interrupt status to indicate the exact cause of the interrupt.

Each of the six sources in interrupts in the SCC (Transmit, Receive, and External/Status interrupts in both channels) and each DMA channel has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). If the IE bit is set for any given source of interrupt, then that source can request interrupts. The only exception to this rule is when the associate Master Interrupt Enable (MIE) bit is reset, then no interrupts are requested. Both the SCC and the DMA have an associated MIE bit. The IE bits in the SCC are write only, but the IE bits in the DMA are read write.

The ISCC provides for nesting of interrupt sources with an interrupt daisy chain using the IEI, IEO, and INTACK pins. As a microprocessor peripheral, the ISCC may request an interrupt only when no higher priority device is requesting one, e.g., when IEI is High. If the device in question requests an interrupt, it enables the /INT signal. The CPU then responds with /INTACK, and the interrupting device places the vector on the data bus.

In the ISCC, the IP bit signals a need for interrupt servicing. When an IP bit is 1 and the IEI input is High, the /INT signal is activated, requesting an interrupt. In the SCC, if the IE bit is not set, then the IP for that source can never be set. The IP bits in the DMA are set independent of the IE bit.

The IUS bits signal that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority in the ISCC and external to the ISCC 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 ISCC being pulled Low and propagated to subsequent peripherals. Internally, the SCC is higher priority than the DMA. An IUS bit is set during an Interrupt Acknowledge cycle if there are no higher priority devices requesting interrupts.

Within the SCC portion of the ISCC there are three types of interrupts· Transmit, Receive, and External/Status. Each

interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receive, Transmit, and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted when the transmit buffer becomes empty. This implies that the transmitter had a data character written into it to make it empty. When enabled, the receiver interrupts the CPU in one of three ways:

1. Interrupt on First Receive Character or Special Receive Condition

2. Interrupt on All Receive Characters or Special Receive Condition

3. Interrupt on Special Condition Only

Interrupt on First Character or Special Condition, and Interrupt on Special Condition Only, are typically used when doing block transfers with the DMA. A Special Receive Condition is one of the following: receiver overrun, framing error in Asynchronous mode, end-of-frame in SDLC mode and, optionally, a parity error. The Special Receive Condition interrupt is different from an Ordinary Receive Character Available interrupt only by the status placed in the vector during the Interrupt Acknowledge cycle. In Interrupt on First Receive Character, an interrupt occurs from Special Receive Conditions any time after the First Receive Character interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the /CTS, /DCD, and /SYNC pins; however, an External/Status interrupt is also caused by a Transmit Underrun condition, or a zero count in the baud rate generator, or by the detection of a Break (Asynchronous mode), Abort (SDLC mode) or EOP (SDLC Loop mode) sequence in the data stream. The interrupt caused by the Abort or EOP has a special feature allowing the ISCC to interrupt when the Abort or EOP sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Abort condition in external logic.

Each DMA in the ISCC has two sources of interrupt, which share an IP bit and an IUS bit, but have independent enables: Terminal Count and Abort. The Abort interrupt is generated when an active DMA channel is forced to terminate its transfers because /BUSACK is de-asserted during a transfer. The Terminal Count interrupt is generated when the DMA transfer count reaches zero. The DMA channels themselves are prioritized in a fixed order: Receive A, Transmit A, Receive B, and Transmit B.

DMA Transfer. In this mode, the on-chip DMA channels transfer data directly to the transmit buffers or directly from the receive buffers. No other transfers are possible (for initialization, for example). The request signals from the receivers and transmitters are hard-wired to the request inputs of the DMA channels internally. Each DMA channel provides a 32-bit address which is either incremented or decremented with a 16-bit transfer length. Whenever a DMA channel receives a request from its associated receiver or transmitter and the DMA channel is enabled, the ISCC activates the /BUSREQ signal. Upon receipt of an active /BUSACK, the DMA channel transfers data between memory and the SCC. This transfer continues until the receiver or transmitter stops requesting a transfer, until the terminal count is reached, or /BUSACK is deactivated. The four DMA channels operate independently when the Request Per Channel option is selected; otherwise, all requests pending at the time of bus acquisition will be serviced before the bus is released. Each DMA channel is independently enabled and disabled.

Bus Interface. The ISCC contains a flexible bus interface that provides the resources necessary to interface the ISCC to virtually any type of bus. The ISCC directly supports either an 8-bit or a 16-bit bus, although all transfers to and from the device are limited to 8-bits at a time. The control signals provided allow connection to either a multiplexed address/data type bus or to a separate address and data type bus. While the ISCC is bus master, the upper address, lower address, and data are multiplexed on AD15-0 Interrupt Acknowledge is signaled through the /INTACK signal, which may be programmed as either a status input, a pulsed input, or a double-pulsed input The ISCC also contains a /WAIT//RDY input for synchronizing CPU or DMA and memory accesses This pin may be programmed to act as either a /WAIT signal or a /READY signal The appropriate signal is provided by the ISCC when it is not bus master, and is sampled by the ISCC when it is bus master. The ISCC requests the bus via a /BUSREQ signal and assumes bus mastership upon receipt of a /BUSACK signal.

CONTROL REGISTERS

The ISCC contains separate register sets for the SCC core and the DMA core. Access to each set is controlled by the A0/SCC//DMA pin. When this pin is an input, a High selects the SCC core and a Low selects the DMA core. The first write to the ISCC after reset is always to the Bus Configuration Register (BCR), see Figure 8. If an /AS is present before the BCR is written to, a multiplexed bus is selected. If no /AS is present before the BCR write, a non-multiplexed bus is selected. The BCR cannot be changed without resetting the ISCC.

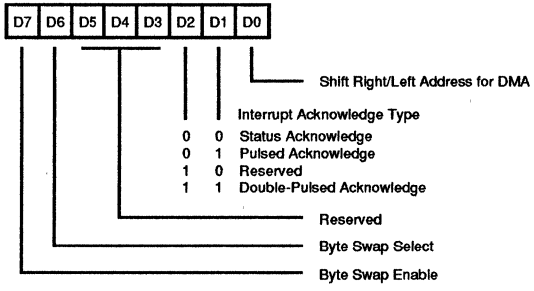


Figure 8. Bus Configuration Register (BCR)

SCC Cell. The SCC core contains 13 write registers (14 counting the transmit buffer) and ten read registers (11 counting the receive buffer) in each channel. Two of the write registers are shared (WR2 and WR9) and are accessed by both channels. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits. Table 1 is a list of the SCC write registers and Table 2 is a list of the SCC read registers. Figures 9 and 10 show the write and read register formats. Read Registers 6 and 7 are only accessible when the SDLC FIFO is enabled. When the SDLC FIFO is not enabled, Read Registers 6 and 7 are images of Read Registers 2 and 3, respectively.

DMA Cell. The DMA cell contains 17 registers (counting the BCR). All of the registers are write/read except the BCR, CCAR and ICSR. The ISCC also has two status registers, the DMA status register (DSR) and the Interrupt Status Register (ISR), which are addressed by reading the CCAR and ICSR. The DMA also reserves two addresses for future use and should not be addressed or should be written with all zeros to prevent unexpected operation and maintain compatibility with future products. Each DMA channel has a 32-bit wide address register providing an addressing range of 4 gigabytes. Each channel also has a 16-bit count register for up to 64K byte data packet sizes (Reference Figures 11-26 and Table 3).

Table 1. SCC Write Registers

Bit	Description
WR0	Register Pointers, various initialization commands
WR1	Transmit and Receive interrupt enables, WAIT/DMA commands
WR2	Interrupt Vector
WR3	Receive parameters and control modes
WR4	Transmit and Receive modes and parameters
WR5	Transmit parameters and control modes
WR6	Sync Character or SDLC address
WR7	Sync Character or SDLC flag
WR8	Transmit buffer
WR9	Master Interrupt control and reset commands
WR10	Miscellaneous transmit and receive control bits
WR11	Clock mode controls for receive and transmit
WR12	Lower byte of baud rate generator
WR13	Upper byte of baud rate generator
WR14	Miscellaneous control bits
WR15	External status interrupt enable control

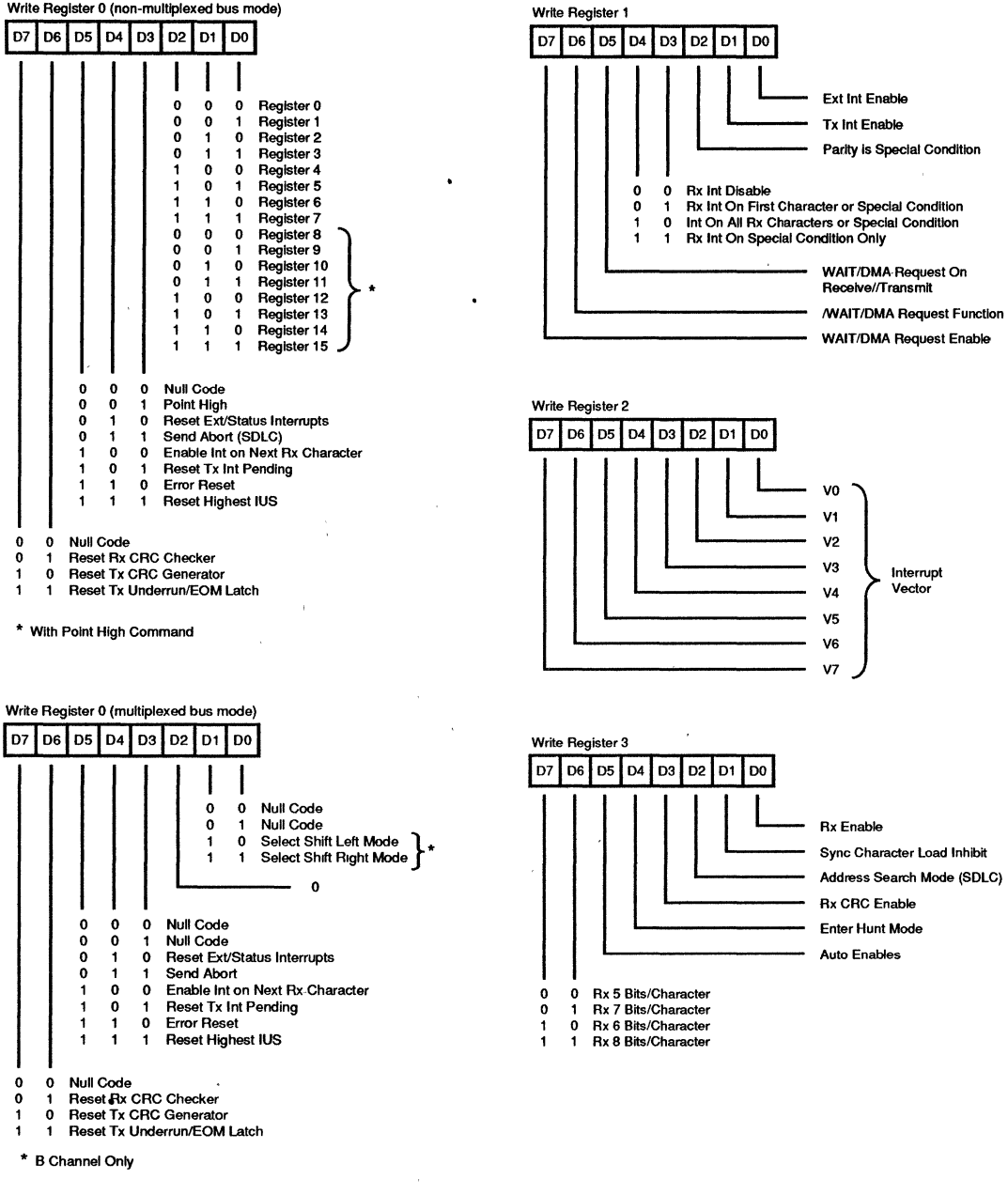


Figure 9. Write Register Bit Functions

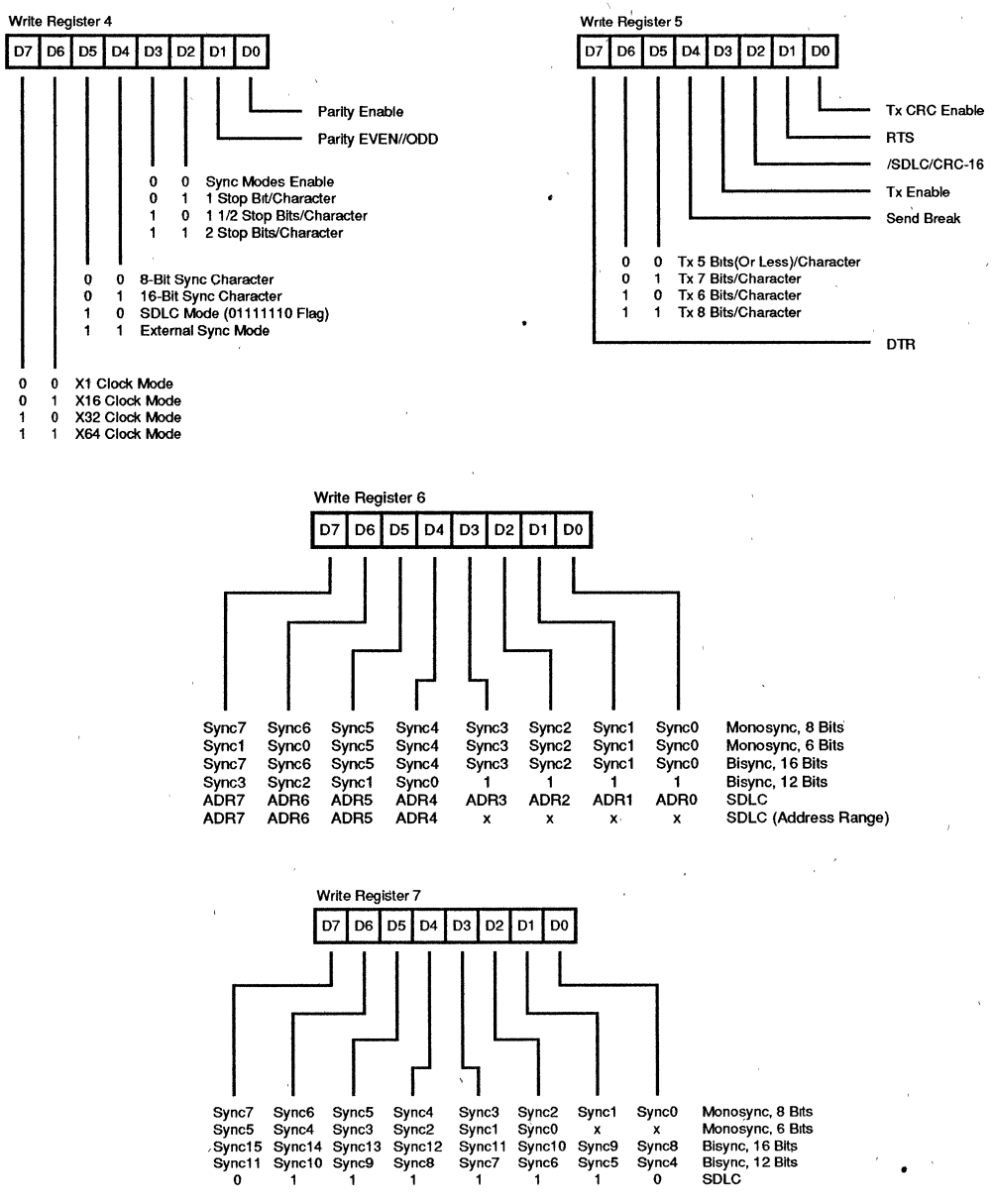


Figure 9. Write Register Bit Functions (Continued)

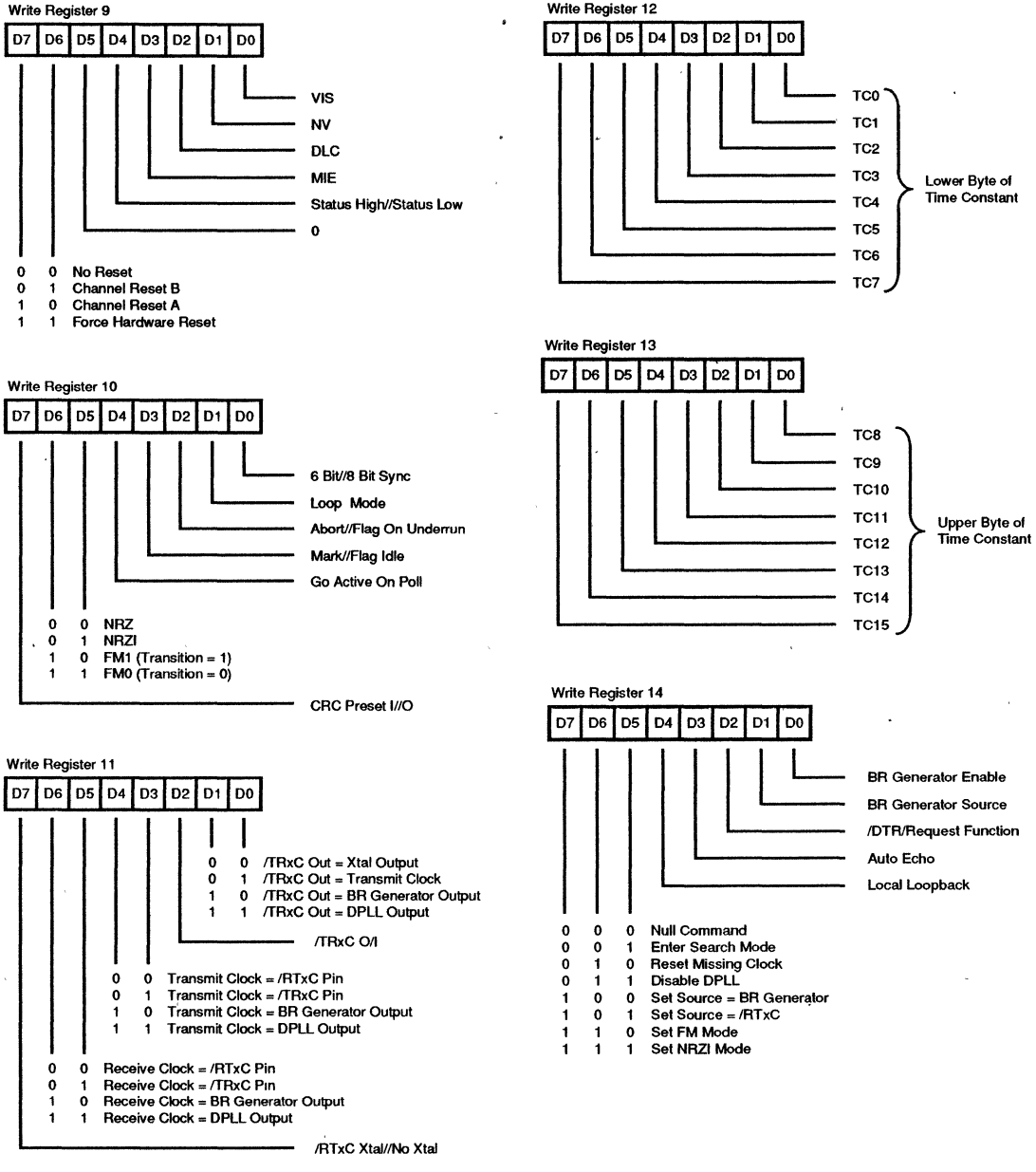


Figure 9. Write Register Bit Functions (Continued)

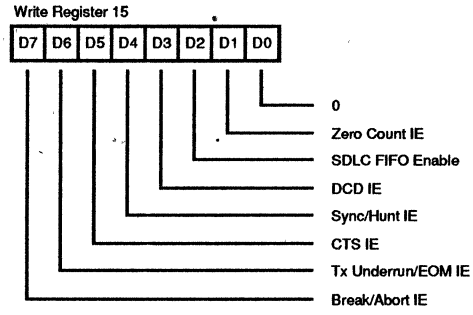


Figure 9. Write Register Bit Functions (Continued)

Table 2. SCC Read Registers

Bit	Description
RR0	Transmit and Receive buffer status and external status
RR1	Special Receive Condition status
RR2	Modified interrupt vector (Channel B only), Unmodified interrupt vector (Channel A only)
RR3	Interrupt pending bits (Channel A only)
RR6	SDLC FIFO byte counter lower byte (only when enabled)
RR7	SDLC FIFO byte count and status (only when enabled)
RR8	Receive buffer
RR10	Miscellaneous status bits
RR12	Lower byte of baud rate generator time constant
RR13	Upper byte of baud rate generator time constant
RR15	External Status interrupt information

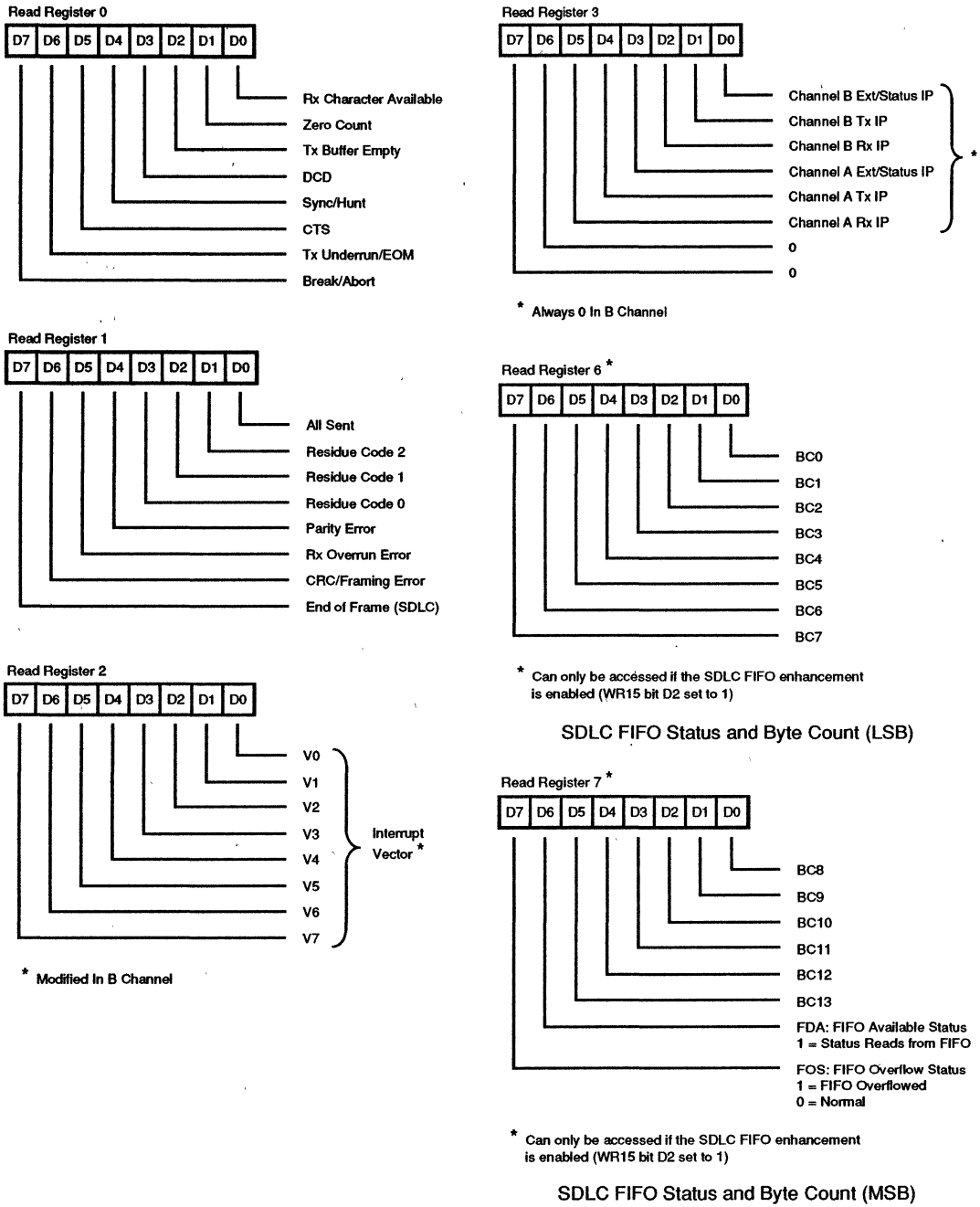


Figure 10. Read Register Bit Functions

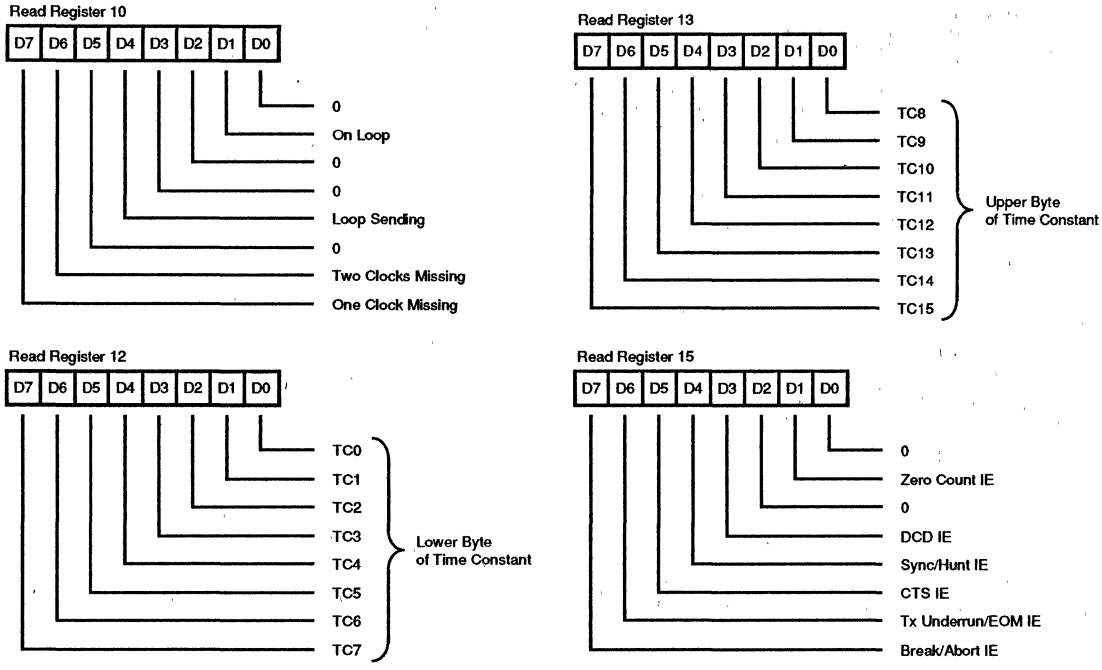


Figure 10. Read Register Bit Functions (Continued)

Table 3. DMA Cell Register Description

Address	Name	Description
xxxxx	BCR	Bus Configuration Register
00000	CCAR	Channel Command/Address Register (WRITE)
00000	DSR	DMA Status Register (READ)
00001	ICR	Interrupt Control Register
00010	IVR	Interrupt Vector Register
00011	ICSR	Interrupt Command Register (WRITE)
00011	ISR	Interrupt Status Register (READ)
00100	DER	DMA Enable/Disable Register
00101	DCR	DMA Control Register
00110		Reserved Address
00111		Reserved Address
01000-01001	RDCRA	Receive DMA Count Register Channel A (Low-high byte)
01010-01011	TDCRA	Transmit DMA Count Register Channel A
01100-01101	RDCRB	Receive DMA Count Register Channel B
01110-01111	TDCRB	Transmit DMA Count Register Channel B
10000-10011	RDARA	Receive DMA Address Register Channel A
10100-10111	TDARA	Transmit DMA Address Register Channel A
11000-11011	RDARB	Receive DMA Address Register Channel B
11100-11111	TDARB	Transmit DMA Address Register Channel B

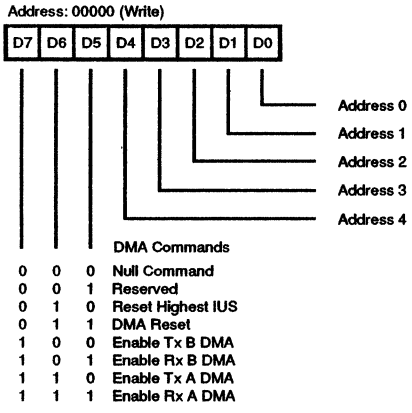


Figure 11. Channel Command/Address Register

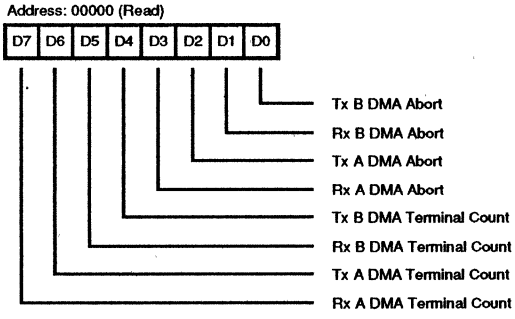


Figure 12. DMA Status Register

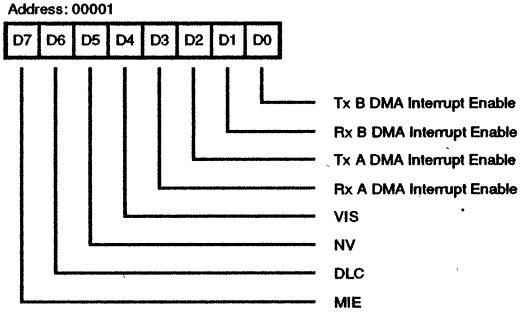


Figure 13. Interrupt Control Register

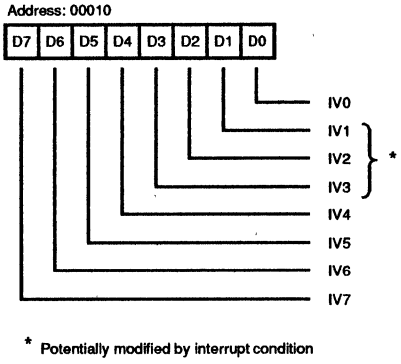


Figure 14. Interrupt Vector Register

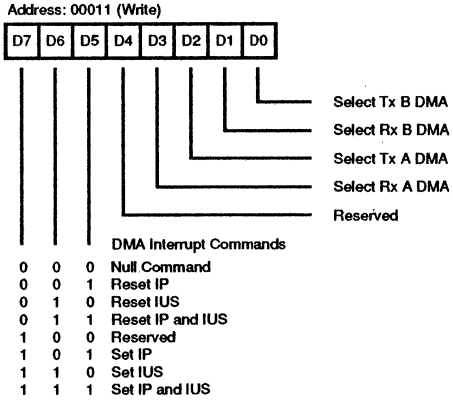


Figure 15. Interrupt Command/Register

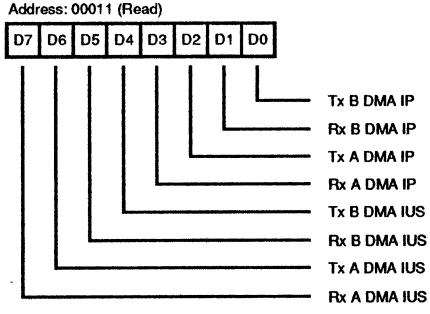


Figure 16. Interrupt Status Register

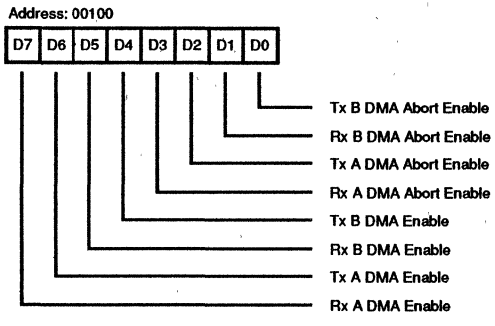


Figure 17. DMA Enable Register

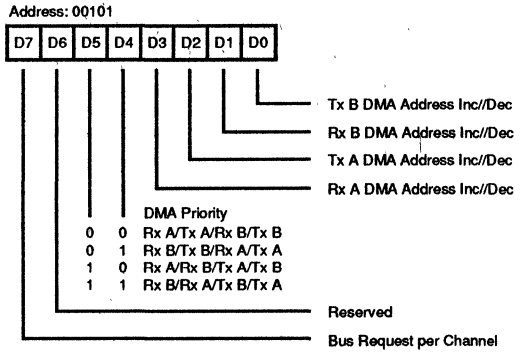


Figure 18. DMA Control Register

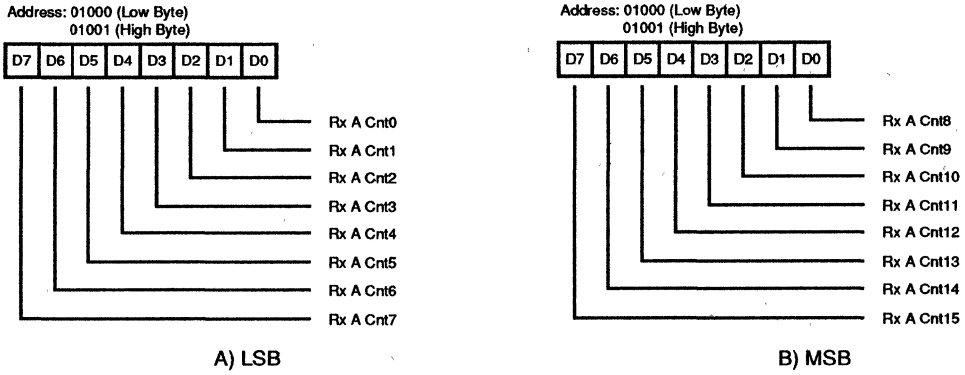


Figure 19. Receive DMA Count Register Channel A

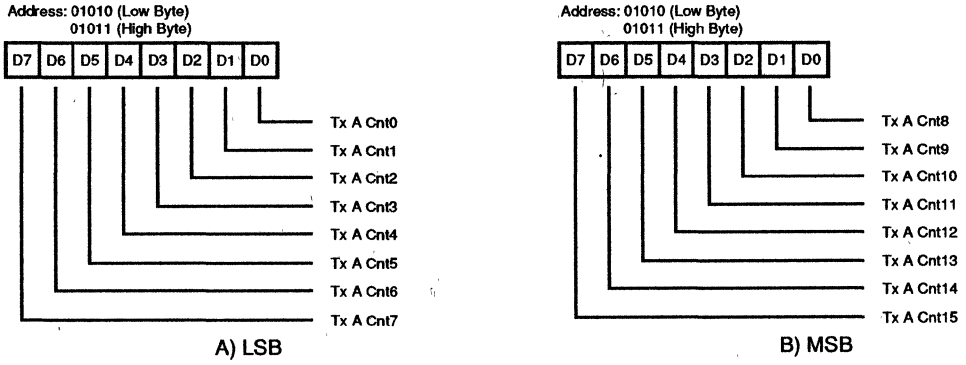


Figure 20. Transmit DMA Count Register Channel A

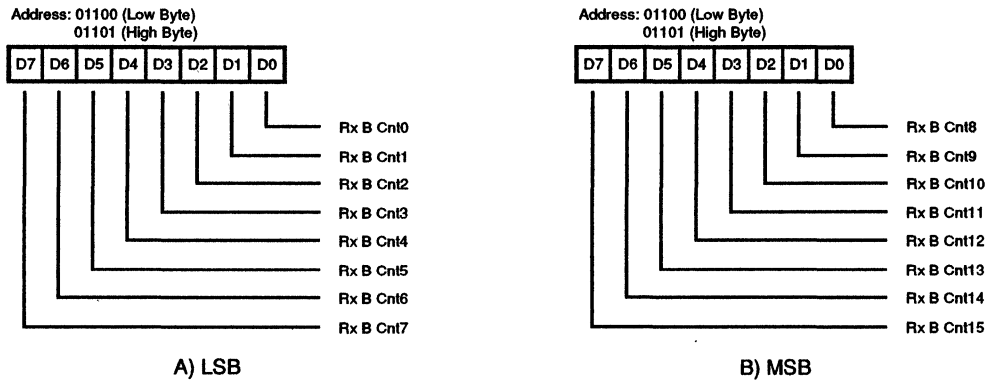


Figure 21. Receive DMA Count Register Channel B

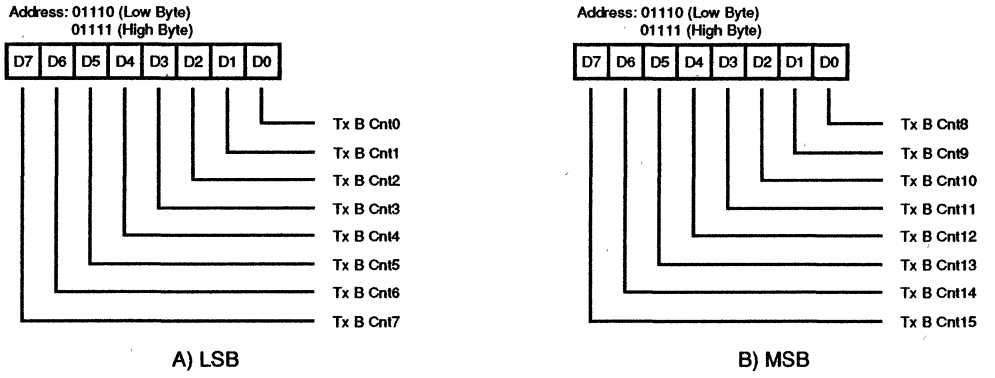


Figure 22. Transmit DMA Count Register Channel B

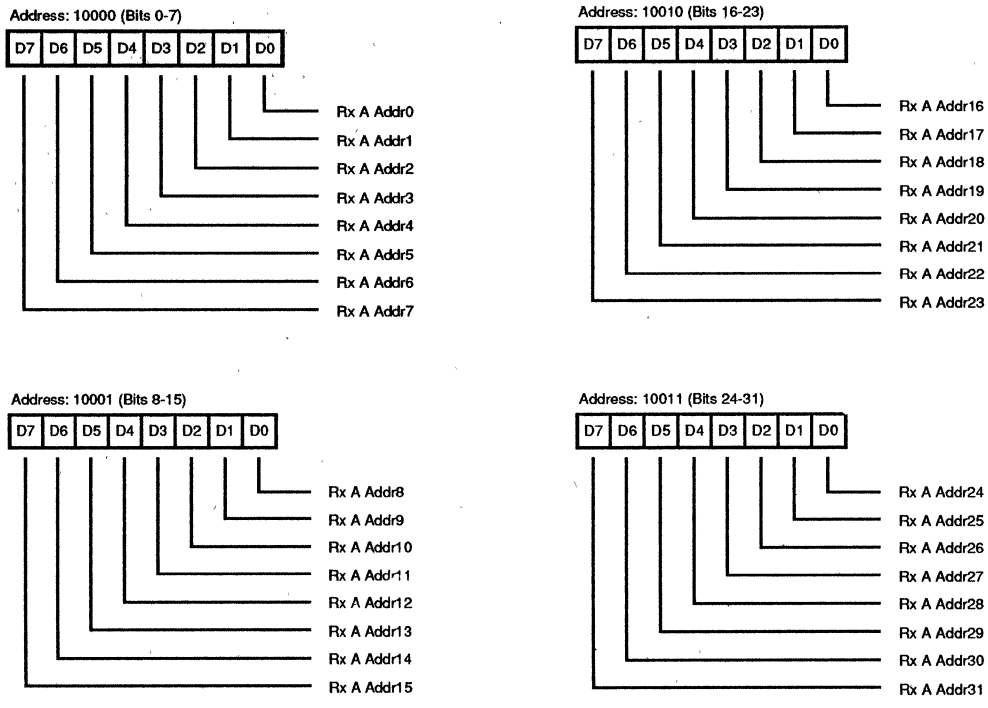


Figure 23. Receive DMA Address Register Channel A

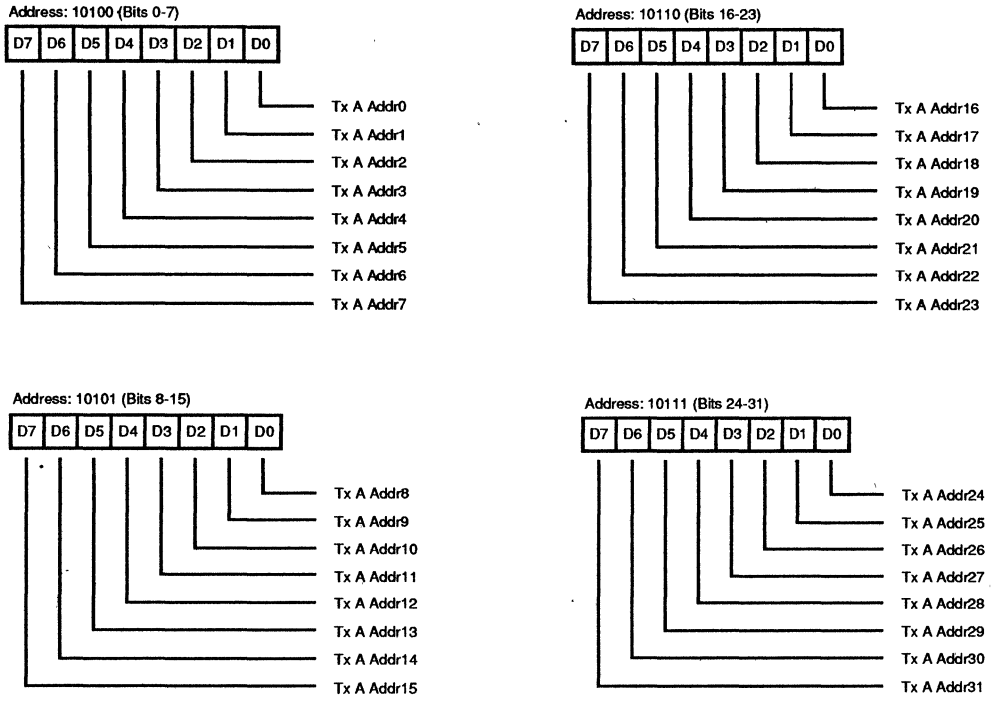


Figure 24. Transmit DMA Address Register Channel A

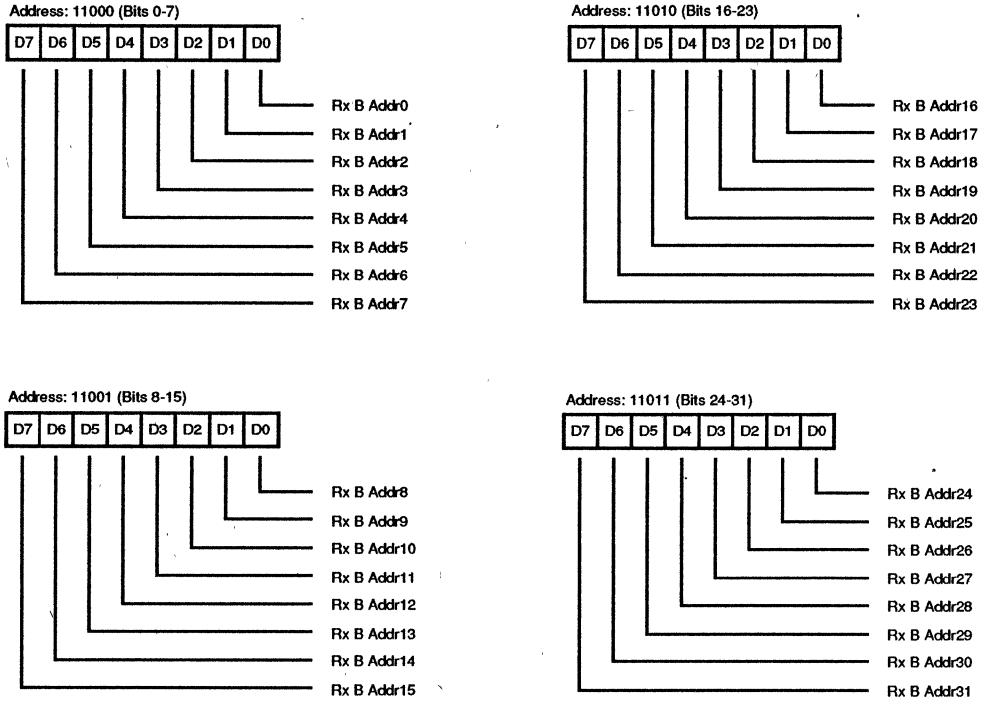


Figure 25. Receive DMA Address Register Channel B

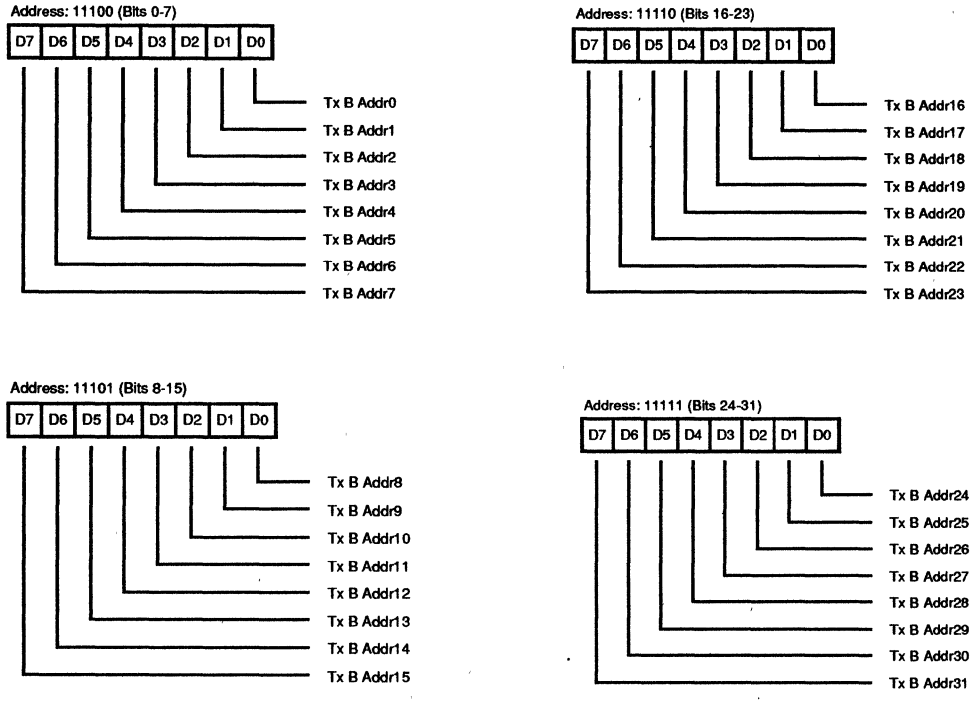


Figure 26. Transmit DMA Address Register Channel B

ABSOLUTE MAXIMUM RATINGS

Voltages on all pins,
with respect to GND .....-0.3 V to +7.0 V
Operating Ambient
Temperature ....See Ordering Information
Storage Temperature .....-85°C to 150°C

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. Standard conditions are as follows:

- +4.75 V ≤ V_{CC} ≤ 5.25 V
- GND = 0 V
- T_A as specified in Ordering Information

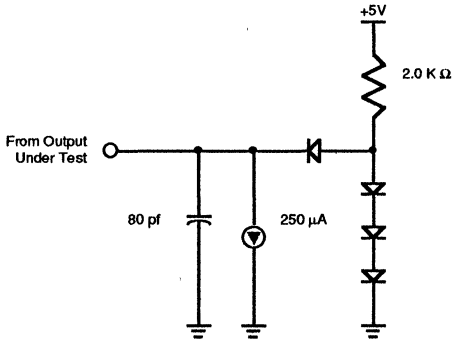


Figure 27. Standard Test Load

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.
Unmeasured pins returned to ground.

MISCELLANEOUS

Transistor Count 52,047

DC CHARACTERISTICS
Z16C35

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} < +2.4 V
I_{OL}	Output Leakage			±10.00 µA		0.4 < V_{OUT} < +2.4 V
I_{CC1}	V_{CC} Supply Current		7	50	mA	V_{CC} = 5 V, V_{IH} = 4.8 V, V_{IL} = 0.2 V

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

AC CHARACTERISTICS

Note: See the corresponding figures following this table (Figures 28-49).

No	Symbol	Parameter	10 MHz * Min	10 MHz * Max	16 MHz† Min	16 MHz† Max	Notes
1	Tcyc	Bus Cycle Time	4TcPC		4TcPC		
2	TwASl	/AS Low Width	40		20		
3	TwASh	/AS High Width	90		55		
4	TwDSl	/DS Low Width	70		50		
5	TwDSh	/DS High Width	60		30		
6	TdAS(DS)	/AS Rise to /DS Fall Delay Time	5		5		
7	TdDS(AS)	/DS Rise to /AS Fall Delay Time	5		5		
8	TdDS(DRa)	/DS Fall to Data Active Delay	0		0		
9	TdDS(DRv)	/DS Fall to Data Valid Delay		85		75	
10	TdDS(DRn)	/DS Rise to Data Not Valid Delay	0		0		
11	TdDS(DRz)	/DS Rise to Data Float Delay		20		15	
12	TsCS(AS)	/CS to /AS Rise Setup Time	15		12		
13	ThCS(AS)	/CS to /AS Rise Hold Time	0		0		
14	TsADD(AS)	Direct Address to /AS Rise Setup Time	15		12		[1]
15	ThADD(AS)	Direct Address to /AS Rise Hold Time	5		5		[1]
16	TsSIA(AS)	Status /INTACK to /AS Rise Setup Time	15		10		
17	ThSIA(AS)	Status /INTACK to /AS Rise Hold Time	5		5		
18	TsAD(AS)	Address to /AS Rise Setup Time	15		10		
19	ThAD(AS)	Address to /AS Rise Hold Time	5		5		
20	TsRW(DS)	R/W to /DS Fall Setup Time	0		0		
21	ThRW(DS)	R/W to /DS Fall Hold Time	25		15		
22	TdDSf(RDY)	/DS Fall to /READY Fall Delay		50		40	
23	TdDSr(RDY)	/DS Rise to /READY Rise Delay		40		20	
24	TsDW(DS)	Write Data to /DS Fall Setup Time	0		0		
25	ThDW(DS)	Write Data to /DS Fall Hold Time	25		15		
26	TdRDY(DRv)	/READY Fall to Data Valid Delay		40		40	
28	TwRDl	/RD Low Width	70		50		
29	TwRDh	/RD High Width	60		30		

AC CHARACTERISTICS (Continued)

No	Symbol	Parameter	10 MHz * Min	10 MHz * Max	16 MHz† Min	16 MHz† Max	Notes
30	TdAS(RD)	/AS Rise to /RD Fall Delay Time	5		5		
31	TdRD(AS)	/RD Rise to /AS Fall Delay Time	5		5		
32	TdRD(DRa)	/RD Fall to Data Active Delay	0		0		
33	TdRD(DRv)	/RD Fall to Data Valid Delay		85		75	
34	TdRD(DRn)	/RD Rise to Data Not Valid Delay	0		0		
35	TdRD(DRz)	/RD Rise to Data Float Delay		20		15	
36	TdRDf(RDY)	/RD Fall to /READY Fall Delay		50		40	
37	TdRDr(RDY)	/RD Rise to /READY Rise Delay		40		20	
38	TwWRl	/WR Low Width	70		50		
39	TwWRh	/WR High Width	60		30		
40	TdAS(WR)	/AS Rise to /WR Fall Delay Time	5		5		
41	TdWR(AS)	/WR Rise to AS Fall Delay Time	5		5		
42	TsDW(WR)	Write Data to /WR Fall Setup Time	0		0		
43	ThDW(WR)	Write Data to /WR Fall Hold Time	25		15		
44	TdWRf(RDY)	/WR Fall to /READY Fall Delay		50		40	
45	TdWRr(RDY)	/WR Rise to /READY Fall Delay		40		20	
46	TsCS(DS)	/CS to /DS Fall Setup Time	0		0		[2]
47	ThCS(DS)	/CS to /DS Fall Hold Time	25		15		[2]
48	TsADD(DS)	Direct Address to /DS Fall Setup Time	0		0		[1,2]
49	ThADD(DS)	Direct Address to /DS Fall Hold Time	25		15		[1,2]
50	TsSIA(DS)	Status /INTACK to /DS Fall Setup Time	0		0		[2]
51	ThSIA(DS)	Status /INTACK to /DS Fall Hold Time	25		15		[2]
52	TsCS(RD)	/CS to /RD Fall Setup Time	0		0		[2]
53	ThCS(RD)	/CS to /RD Fall Hold Time	25		15		[2]
54	TsADD(RD)	Direct Address to /RD Fall Setup Time	0		0		[1,2]
55	ThADD(RD)	Direct Address to /RD Fall Hold Time	25		15		[1,2]
56	TsSIA(RD)	Status /INTACK to /RD Fall Setup Time	0		0		[2]
57	ThSIA(RD)	Status /INTACK to /RD Fall Hold Time	25		15		[2]
58	TsCS(WR)	/CS to /WR Fall Setup Time	0		0		[2]
59	ThCS(WR)	/CS to /WR Fall Hold Time	25		15		[2]
60	TsADD(WR)	Direct Address to /WR Fall Setup Time	0		0		[1,2]
61	ThADD(WR)	Direct Address to /WR Fall Hold Time	25		15		[1,2]
62	TsSIA(WR)	Status /INTACK to /WR Fall Setup Time	0		0		[2]
63	ThSIA(WR)	Status /INTACK to /WR Fall Hold Time	25		15		[2]
78	TdDSI(RDY)	/DS Fall (INTACK) to /READY Fall Delay		300		250	[4]
81	TsIEI(DSI)	IEI to /DS Fall (INTACK) Setup Time	60		40		
82	ThIEI(DSI)	IEI to /DS Rise (INTACK) Hold Time	0		0		
83	TdIEI(IEO)	IEI to IEO Delay		60		40	
84	TdAS(IEO)	/AS Rise or Status INTACK to IEO Delay		60		40	
85	TdDSI(INT)	/DS Fall (INTACK) to /INT Inactive Delay		200		170	
86	TdDSI(Wf)	/DS Fall (INTACK) to /WAIT Fall Delay		40		35	
87	TdDSI(Wr)	/DS Fall (INTACK) to /WAIT Rise Delay		300		175	[4]
88	TdW(DRy)	/WAIT Rise to Data Valid Delay		40		35	
89	TdRDI(RDY)	/RD Fall (INTACK) to /READY Fall Delay		300		175	[4]

AC CHARACTERISTICS (Continued)

No	Symbol	Parameter	10 MHz * Min	10 MHz * Max	16 MHz† Min	16 MHz† Max	Notes
91	TsIEI(RDI)	IEI to /RD Fall (INTACK) Setup Time	60		50		
92	ThIEI(RDI)	IEI to /RD Rise (INTACK) Hold Time	0		0		
93	TdRDI(INT)	/RD Fall (INTACK) to /INT Inactive Delay		200		170	
94	TdRDI(Wf)	/RD Fall (INTACK) to /WAIT Fall Delay		40		35	
95	TdRDI(Wr)	/RD Fall (INTACK) to /WAIT Rise Delay		300		175	[4]
96	TwPIA1	Pulsed /INTACK Low Width	70		55		
97	TwPIAh	Pulsed /INTACK High Width	60		45		
98	TdAS(PIA)	/AS Rise to Pulsed /INTACK Fall Delay Time	5		5		
99	TdPIA(AS)	Pulsed /INTACK Rise to /AS Fall Delay Time	5		5		
100	TdPIA(DRa)	Pulsed /INTACK Fall to Data Active Delay	0		0		
101	TdPEA(DRn)	Pulsed /INTACK Rise to Data Not Valid Delay	0		0		
102	TdPIA(DRz)	Pulsed /INTACK Rise to Data Float Delay		20		15	
103	TsIEI(PIA)	IEI to Pulsed /INTACK Fall Setup Time	60		50		
104	ThIEI(PIA)	IEI to Pulsed /INTACK Rise Hold Time	0		0		
105	TdPIA(IEO)	Pulsed /INTACK Fall to IEO Delay		60		50	
106	TdPIA(INT)	Pulsed /INTACK Fall to /INT Inactive Delay		200		170	
107	TdPIAf(RDY)	Pulsed /INTACK Fall to /READY Fall Delay		300		200	[4]
108	TdPIAr(RDY)	Pulsed /INTACK Rise to /READY Rise Delay		40		35	
109	TdPIA(Wf)	Pulsed /INTACK Fall to /WAIT Fall Delay		40		35	
110	TdPIA(Wr)	Pulsed /INTACK Fall to /WAIT Rise Delay		300		175	[4]
111	TdSIA(INT)	Status /INTACK Fall to /INT Inactive Delay		200		200	[2]
113	TwRESI	/RESET Low Width	170		140		
114	TwRESh	/RESET High Width	60		40		
115	TdRES(STB)	/RESET Rise to /Strobe Fall	60		40		[3]
116	TdPC(BUSa)	PCLK Rise to Bus Active Delay		40		35	[5]
117	TdPC(BRQ)	PCLK Rise to /BUSREQ Delay		40		35	
118	TsBAK(PC)	/BUSACK to PCLK Rise Setup Time	10		10		
119	ThBAK(PC)	/BUSACK to PCLK Rise Hold Time	30		20		
120	TwPCl	PCLK Low Width	35		26		
121	TwPCh	PCLK High Width	35		26		
122	TcPC	PCLK Cycle Time	100		61		
123	TfPC	PCLK Fall Time		10		5	
124	TrPC	PCLK Rise Time		10		5	
125	TdPCr(UAS)	PCLK Rise to /UAS Delay		30		25	[5]
126	TwUASl	/UAS Low Width	30		25		[5,6]
127	TdPCf(UAS)	PCLK Fall to /UAS Delay		30		25	[5]
128	TdPCr(AS)	PCLK Rise to /AS Delay		30		25	[5]
129	TwASl	/AS Low Width	30		25		[5,6]
130	TdPCf(AS)	PCLK Fall to /AS Delay		30		25	[5]
131	TdAS(DSr)	/AS Rise to /DS Fall (READ) Delay	30		25		[5,7]
132	TdDS(PCr)	PCLK Rise to /DS Delay		30		25	[5]
133	TwDSlr	/DS Low Width (READ)	135		90		[5,8]
134	TdPCf(DS)	PCLK Fall to /DS Delay		30		25	[5]
135	TsDR(DS)	Read Data to /DS Rise Setup Time	30		25		[5]

AC CHARACTERISTICS (Continued)

No	Symbol	Parameter	10 MHz * Min	10 MHz * Max	16 MHz† Min	16 MHz† Max	Notes
136	ThDR(DS)	Read Data to /DS Rise Hold Time	0		0		[5]
137	TdPC(RW)	PCLK Rise to R//W Delay		30		25	[5]
138	TdAS(RD)	/AS Rise to /RD Fall Delay	30		25		[5,7]
139	TdPCr(RD)	PCLK Rise to /RD Delay		30		25	[5]
140	TwRDI	/RD Low Width	135		90		[5,8]
141	TdPCf(RD)	PCLK Fall to /RD Delay		30		25	[5]
142	TsDR(RD)	Read Data to /RD Rise Setup Time	30		25		[5]
143	ThDR(RD)	Read Data to /RD Rise Hold Time	0		0		[5]
144	TdPC(ADD)	PCLK Rise to Direct Address Delay		30		25	[1,5]
145	TdPC(AD)	PCLK Rise to Address Delay		40		40	[5]
146	ThAD(PC)	Address to PCLK Rise Hold Time	0		0		[5]
147	TdPC(ADz)	PCLK Rise to Address Float Delay		50		45	[5]
148	TdPC(ADa)	PCLK Rise to Address Active Delay		40		35	[5]
149	TsAD(UAS)	Address to /UAS Rise Setup Time	20		10		[5]
150	ThAD(UAS)	Address to /UAS Rise Hold Time	20		10		[5]
151	TsAD(AS)	Address to /AS Rise Setup Time	20		10		[5]
152	ThAD(AS)	Address to /AS Rise Hold Time	20		10		[5]
153	TsW(PC)	/WAIT to PCLK Fall Setup Time	10		10		[5]
154	ThW(PC)	/WAIT to PCLK Fall Hold Time	30		20		[5]
155	TsRDY(PC)	/READY to PCLK Fall Setup Time	10		10		[5]
156	ThRDY(PC)	/READY to PCLK Fall Hold Time	30		20		[5]
157	ThDW(PC)	Write Data to PCLK Rise Hold Time	0		0		[5]
158	TdAS(DSw)	/AS Rise to /DS Fall (WRITE) Delay	85		45		[5,9]
159	TsDW(DS)	Write Data to /DS Fall Setup Time	30		25		[5,6]
160	TwDSlw	/DS Low Width (WRITE)	90		70		[5,10]
161	ThDW(DS)	Write Data to /DS Rise Hold Time	30		25		[5, 7]
162	TdAS(WR)	/AS Rise to /WR Fall Delay	85		55		[5,9]
163	TsDW(WR)	Write Data to /WR Fall Setup Time	30		25		[5,6]
164	TwWRI	/WR Low Width	90		55		[5,10]
165	ThDW(WR)	Write Data to /WR Rise Hold Time	30		25		[5,7]
166	TdPC(WR)	PCLK Fall to /WR Delay		30		25	[5]
167	TdPC(BUSz)	PCLK Rise to Bus Float Delay		50		40	[5]

Notes:
[1] Direct address is A1/A//B or A0/SCC//DMA.
[2] The parameter applies only when /AS is not present.
[3] /Strobe is any of /DS, /RD, /WR or Pulsed /INTACK.
[4] Clock-cycle dependent, 2TcPC + TwPCI + TfPC + 55
[5] Parameter applies only while ISCC is bus master.
[6] Clock-cycle dependent, TwPCh + TfPC - 15.
[7] Clock-cycle dependent, TwPC1 + TrPC - 15.
[8] Clock-cycle dependent, TcPC + TwPCh + TrPC - 10.
[9] Clock-cycle dependent, TcPC - 15.
[10] Clock-cycle dependent, TcPC - 10.

* Units in nanoseconds
† 16 MHz Timing is Preliminary

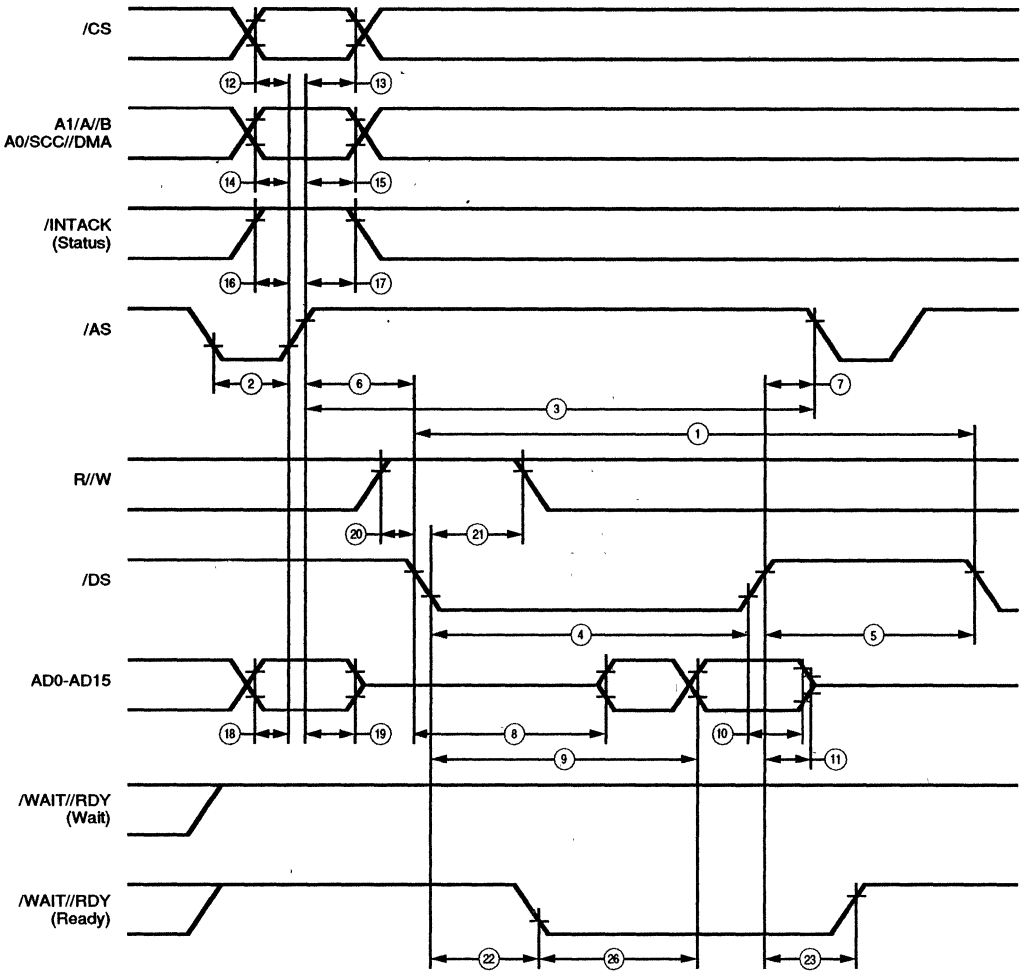


Figure 28. Multiplexed /DS Read Cycle

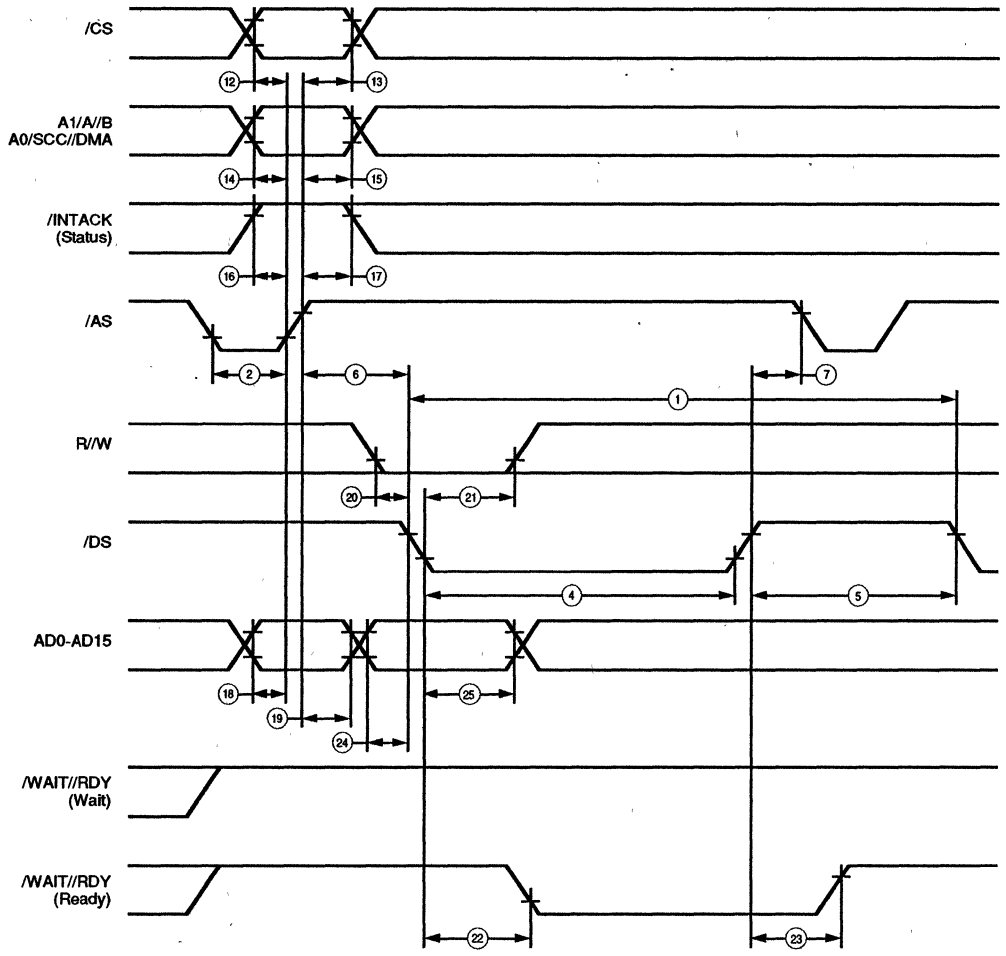


Figure 29. Multiplexed /DS Write Cycle

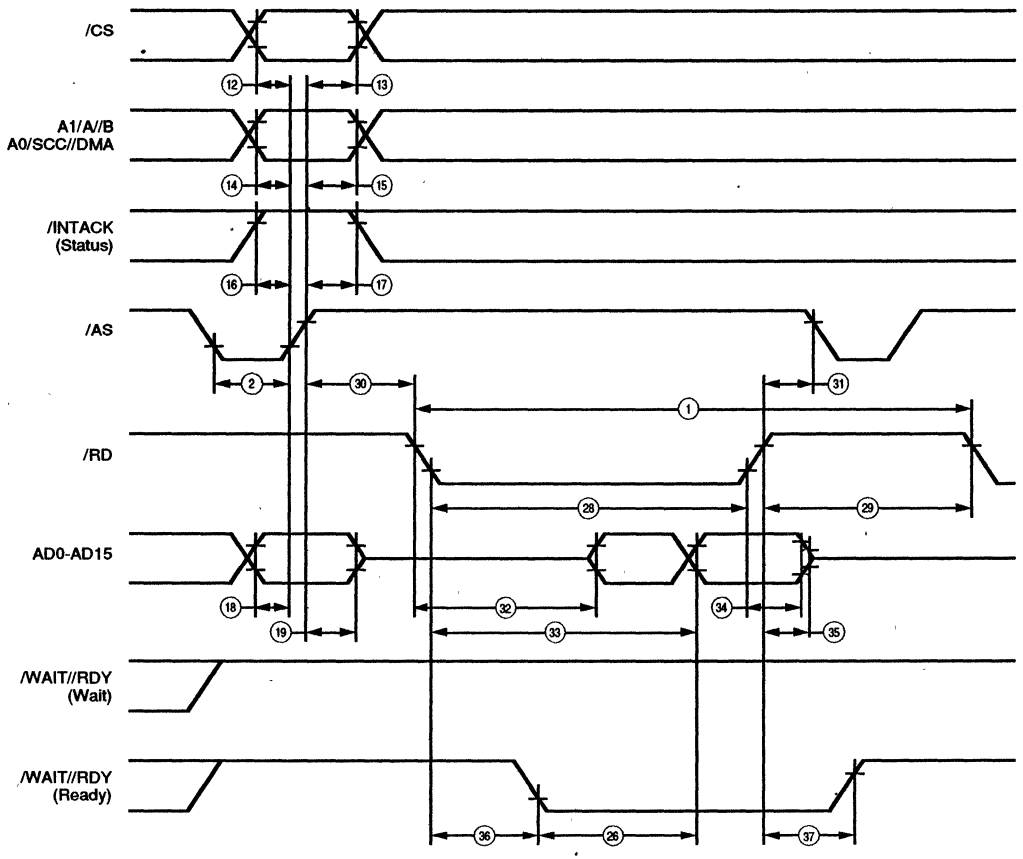


Figure 30. Multiplexed /RD Read Cycle

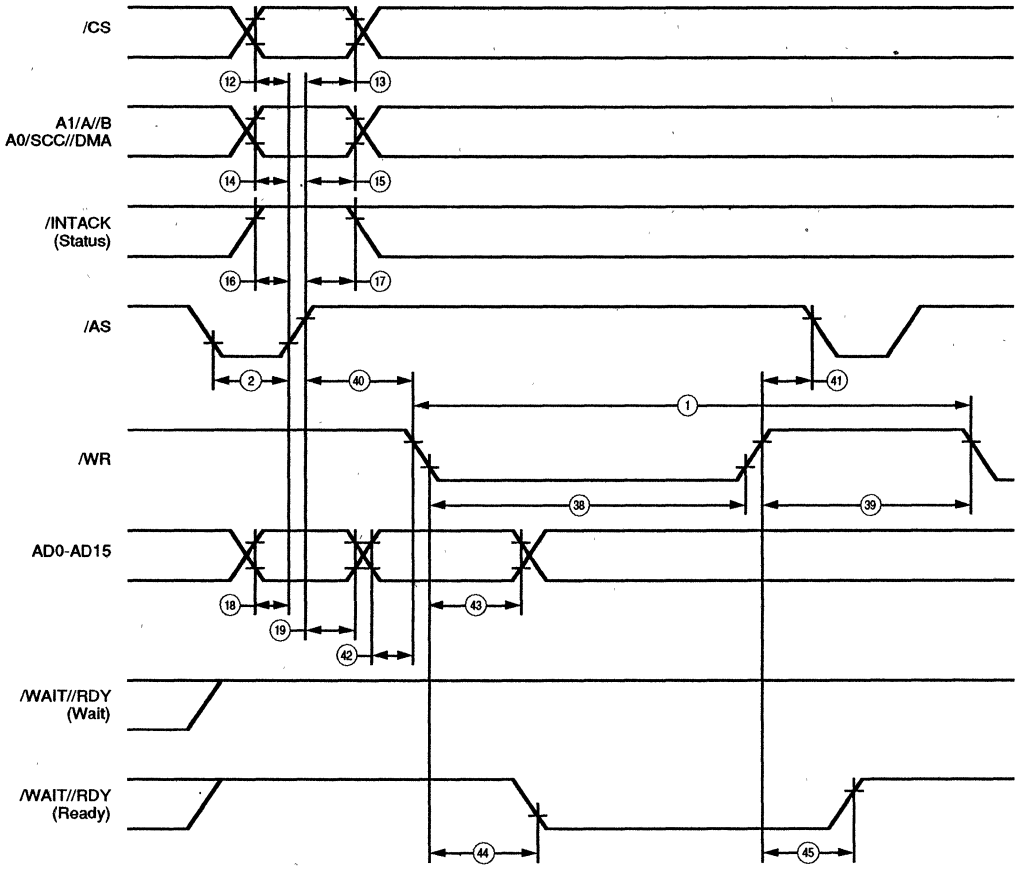


Figure 31. Multiplexed /WR Write Cycle

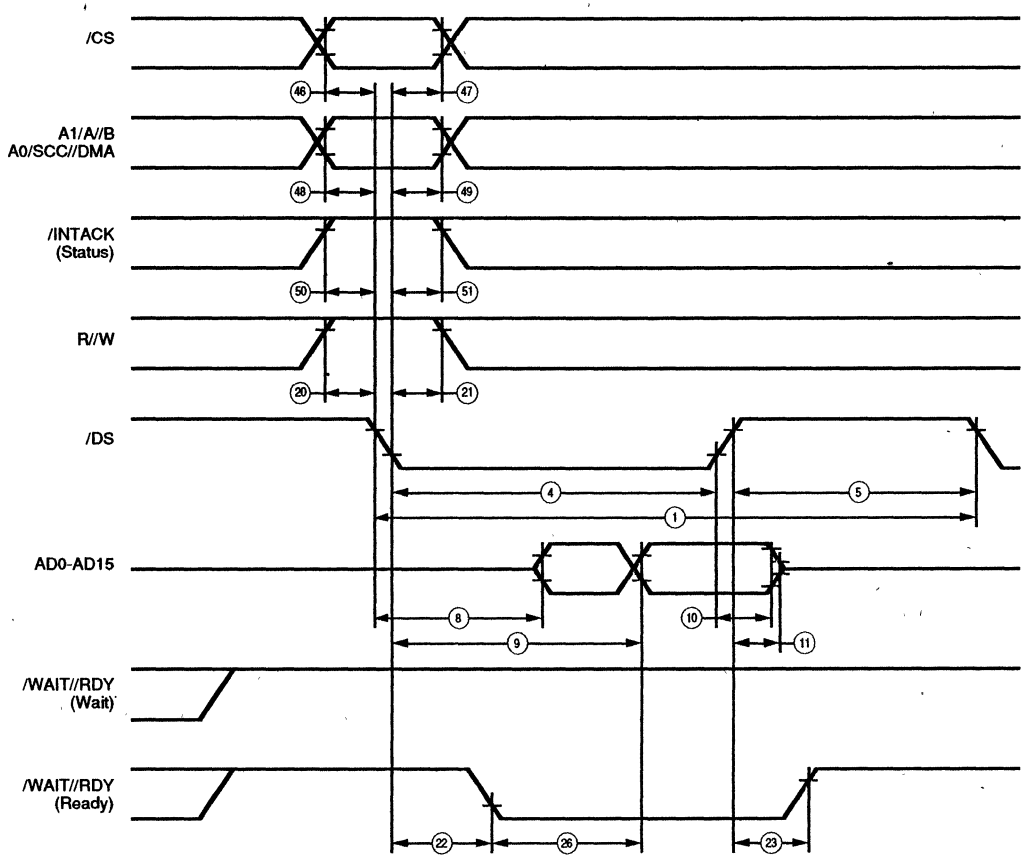


Figure 32. Non-multiplexed /DS Read Cycle

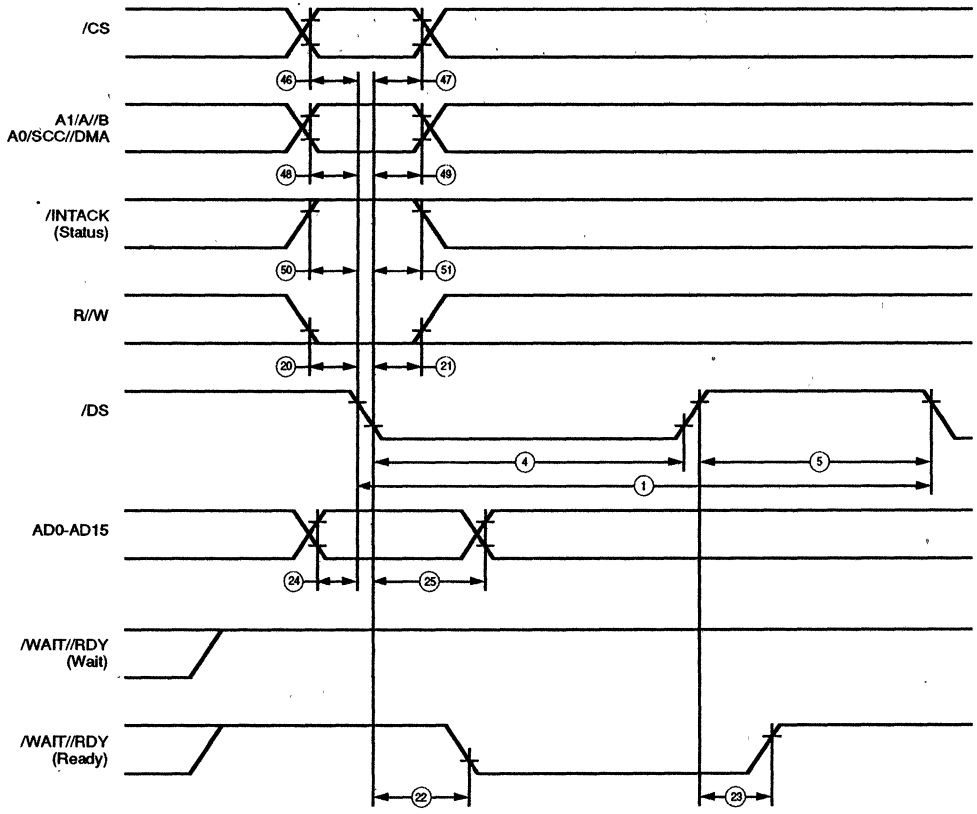


Figure 33. Non-multiplexed /DS Write Cycle

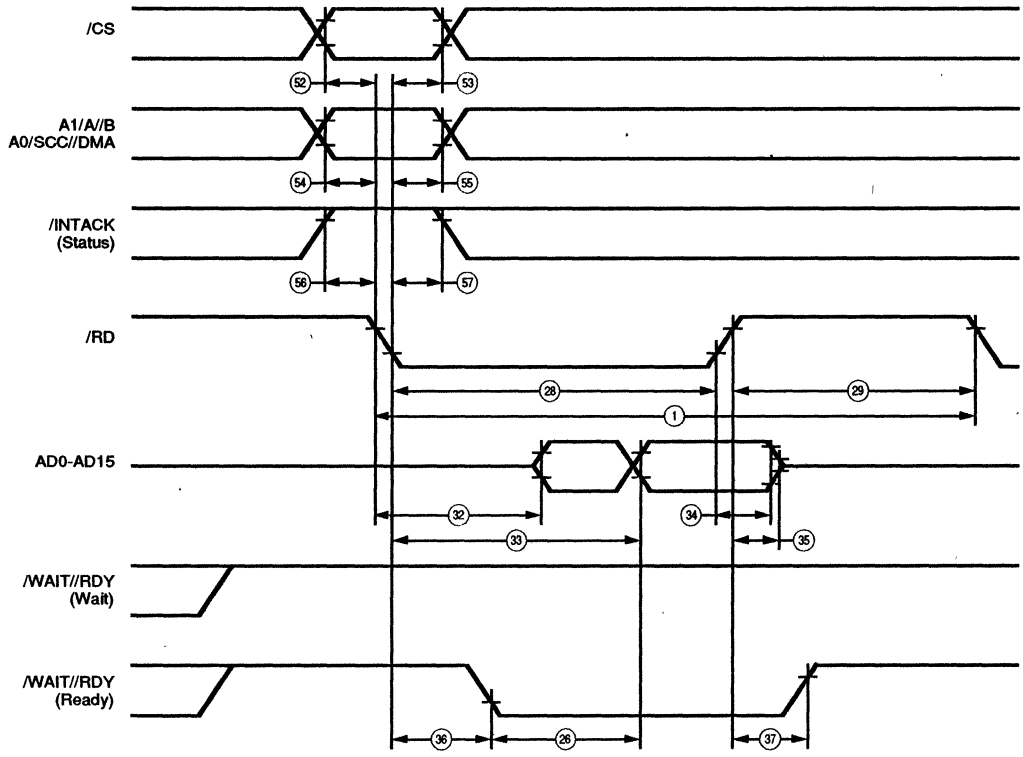


Figure 34. Non-multiplexed /RD Read Cycle

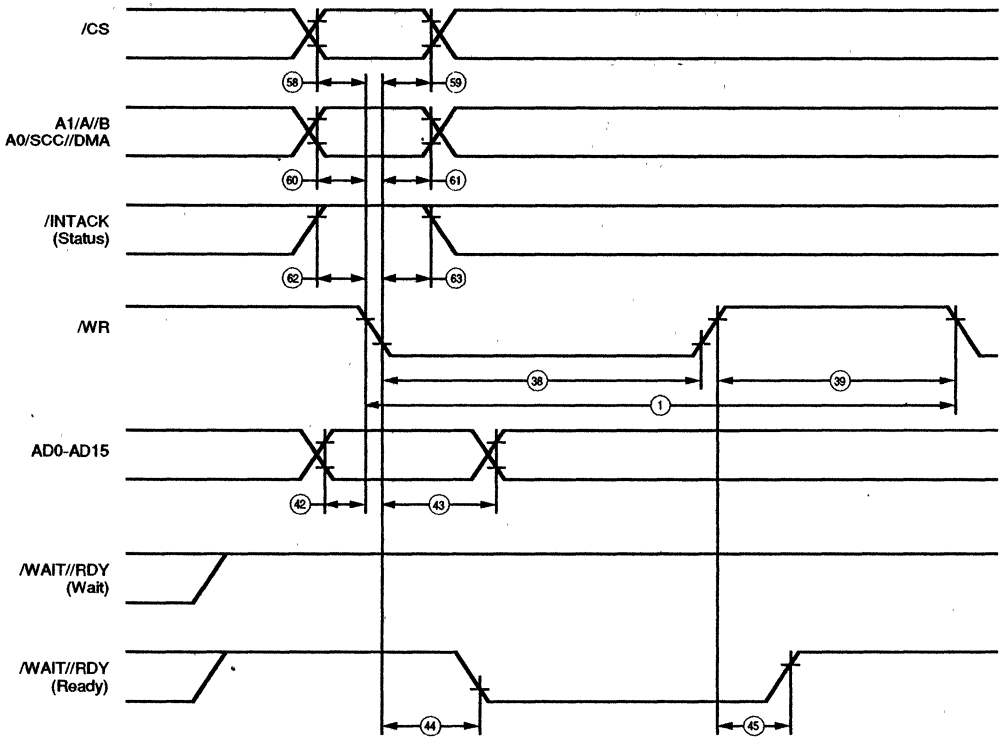


Figure 35. Non-multiplexed /WR Write Cycle

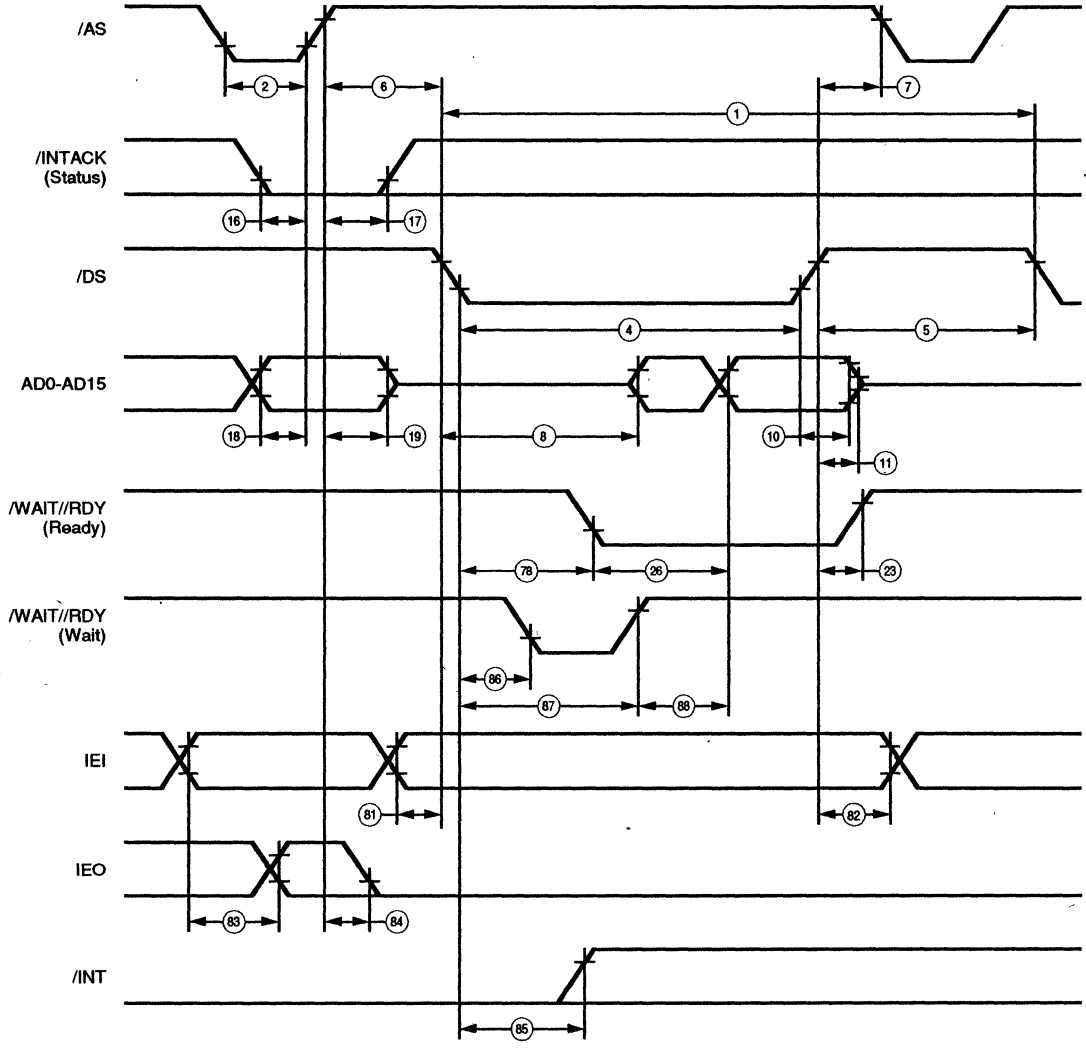


Figure 36. Multiplexed /DS Status INTACK Cycle

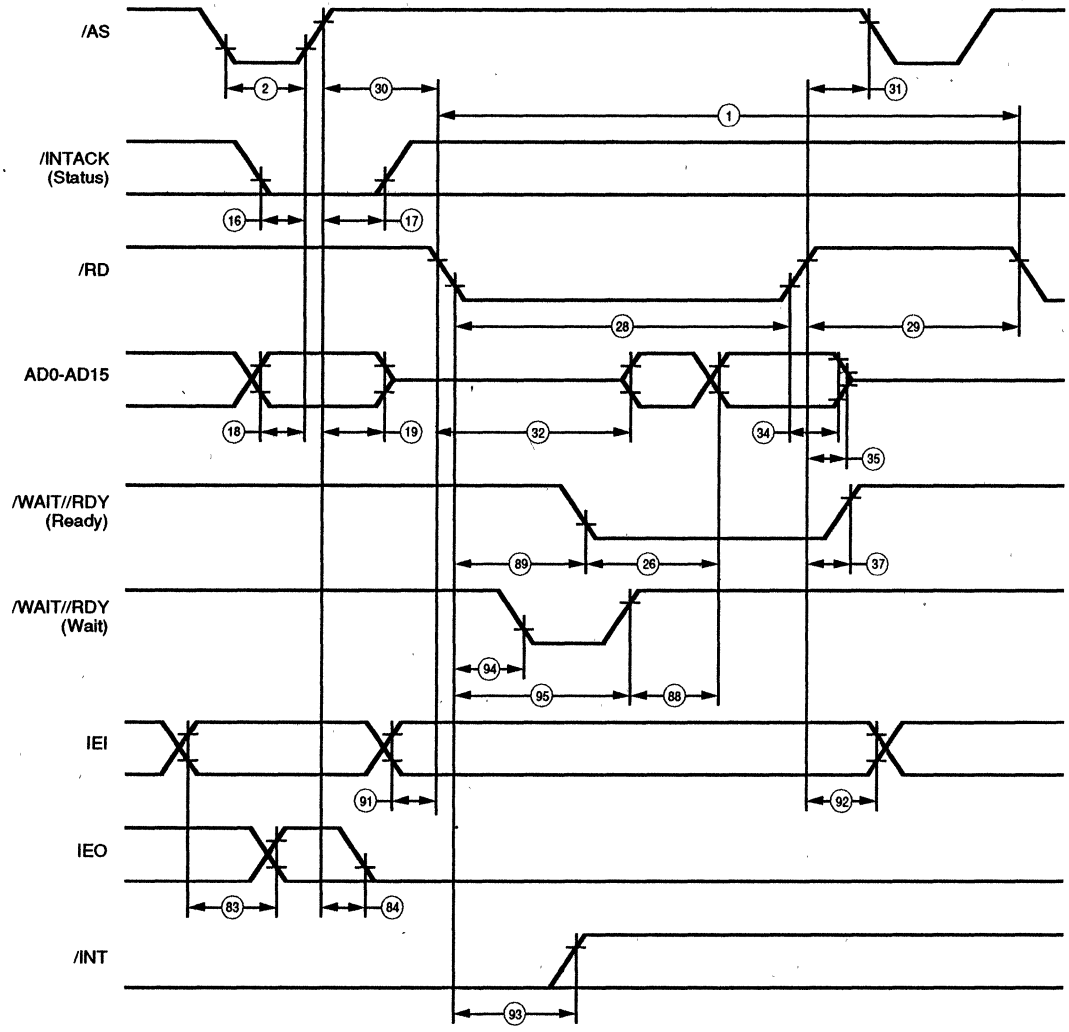


Figure 37. Multiplexed /RD Status INTACK Cycle

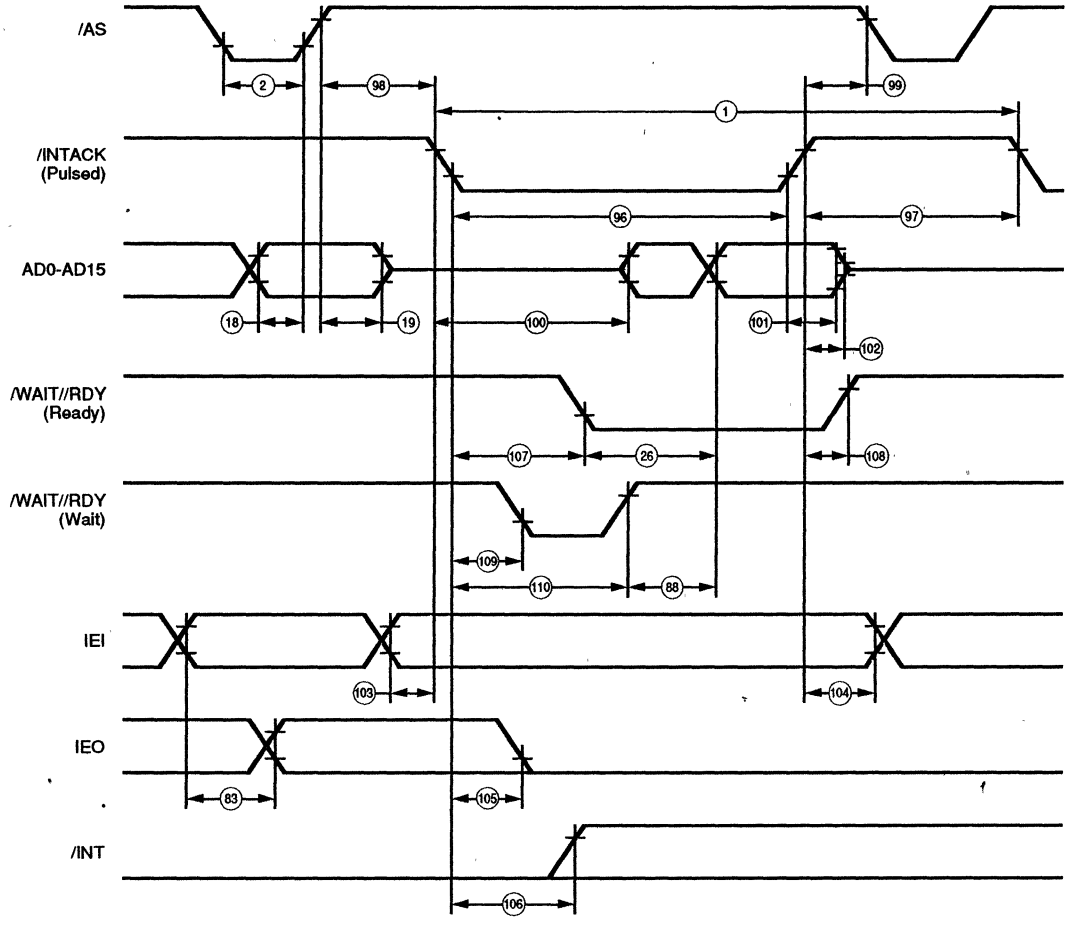


Figure 38. Multiplexed Pulsed INTACK Cycle

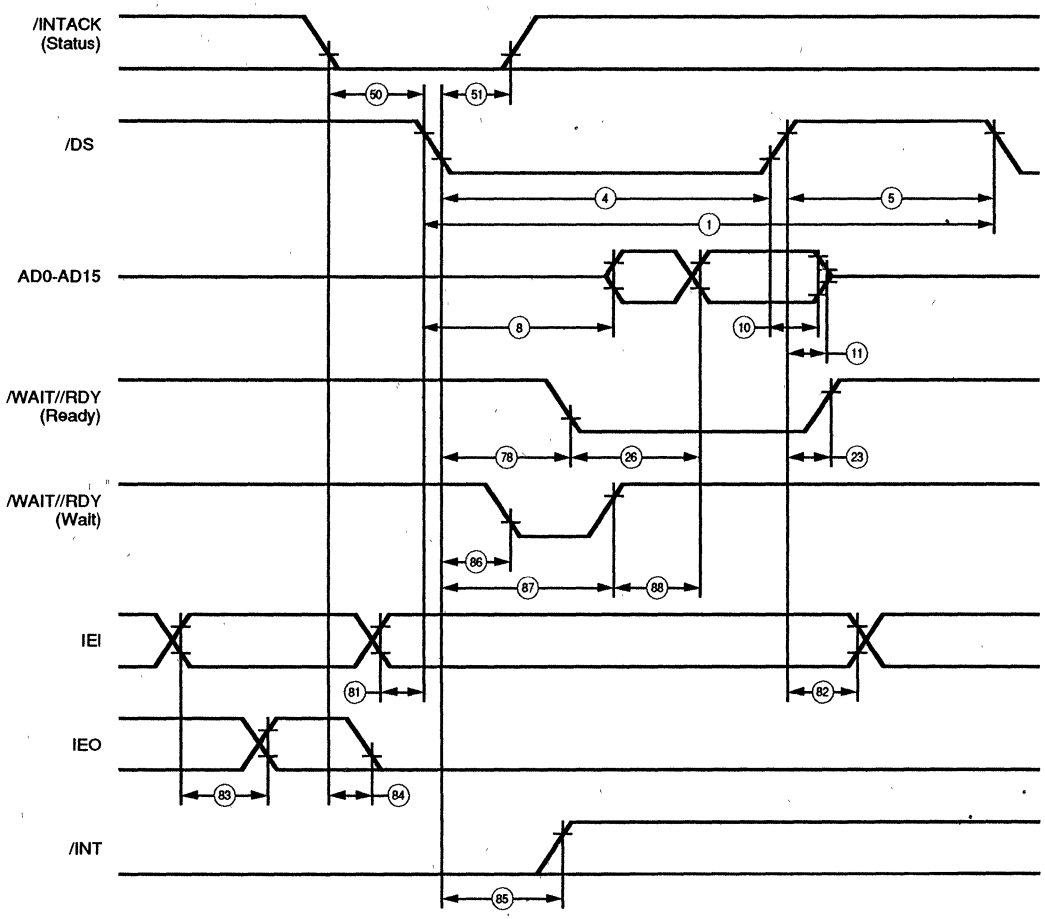


Figure 39. Non-multiplexed /DS INTACK Cycle

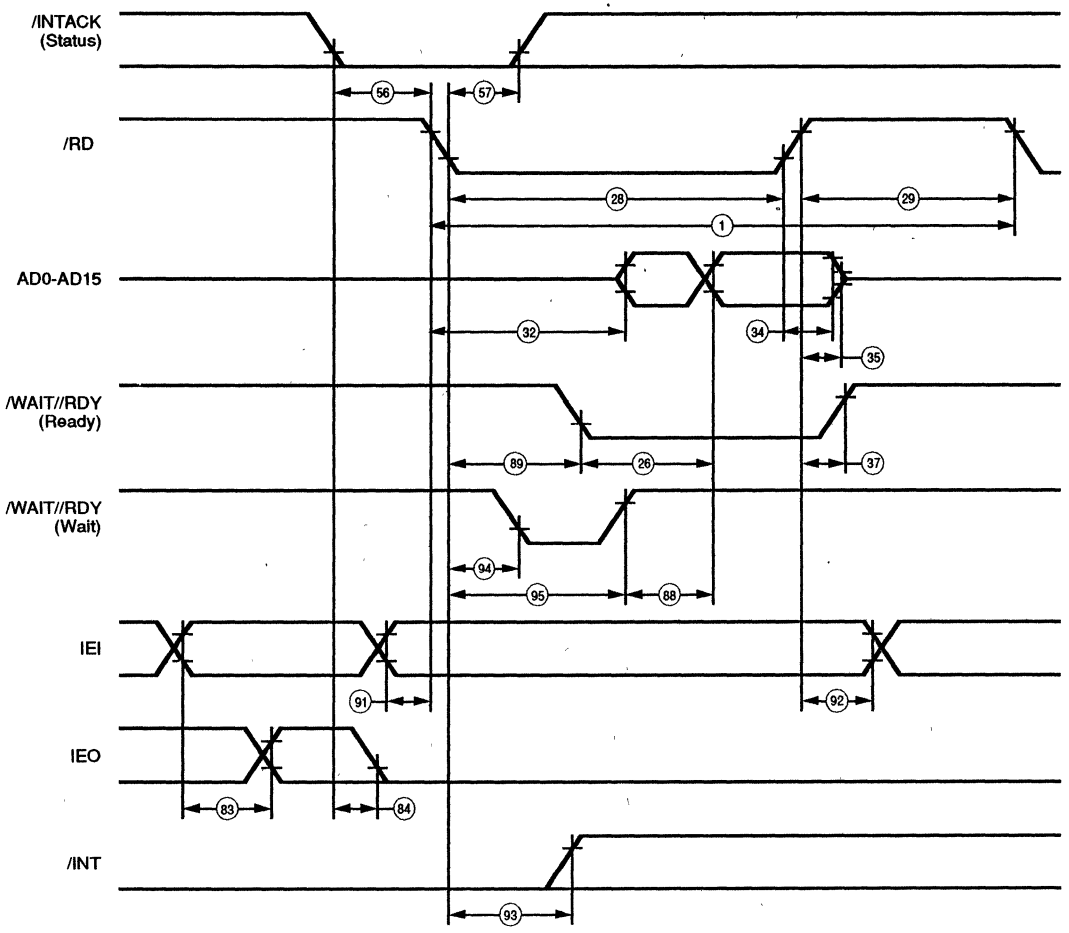


Figure 40. Non-multiplexed /RD Status INTACK Cycle

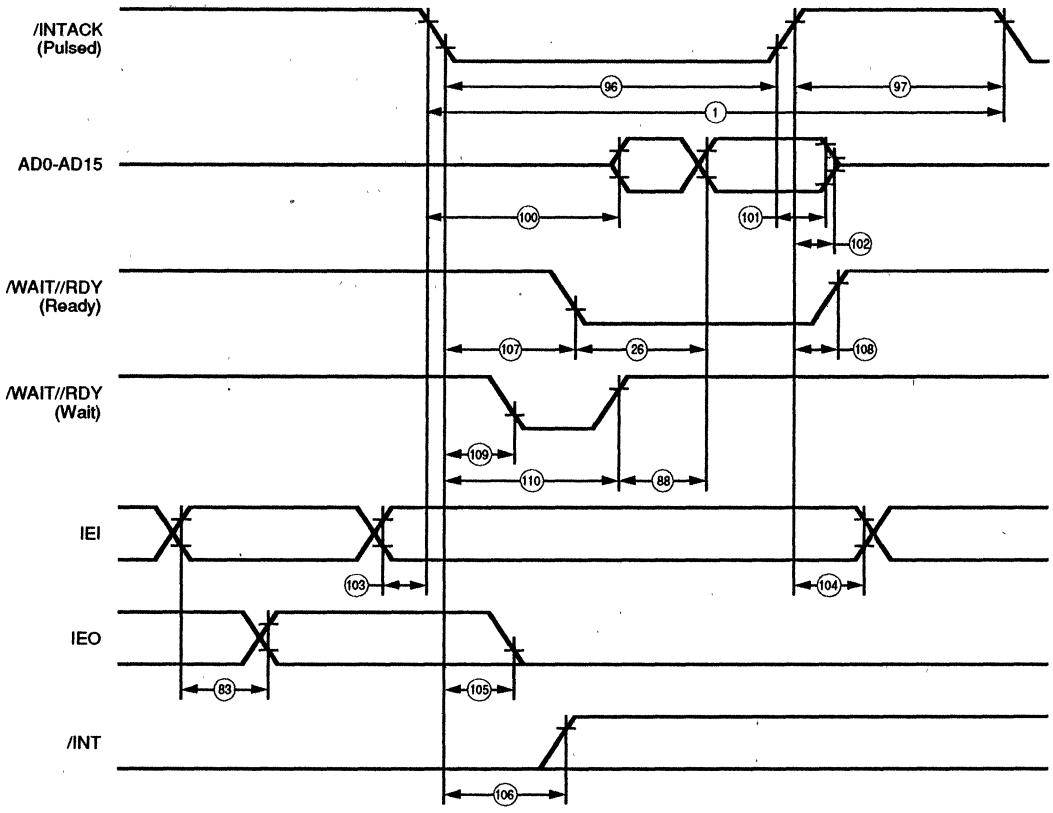


Figure 41. Non-multiplexed Pulsed INTACK Cycle

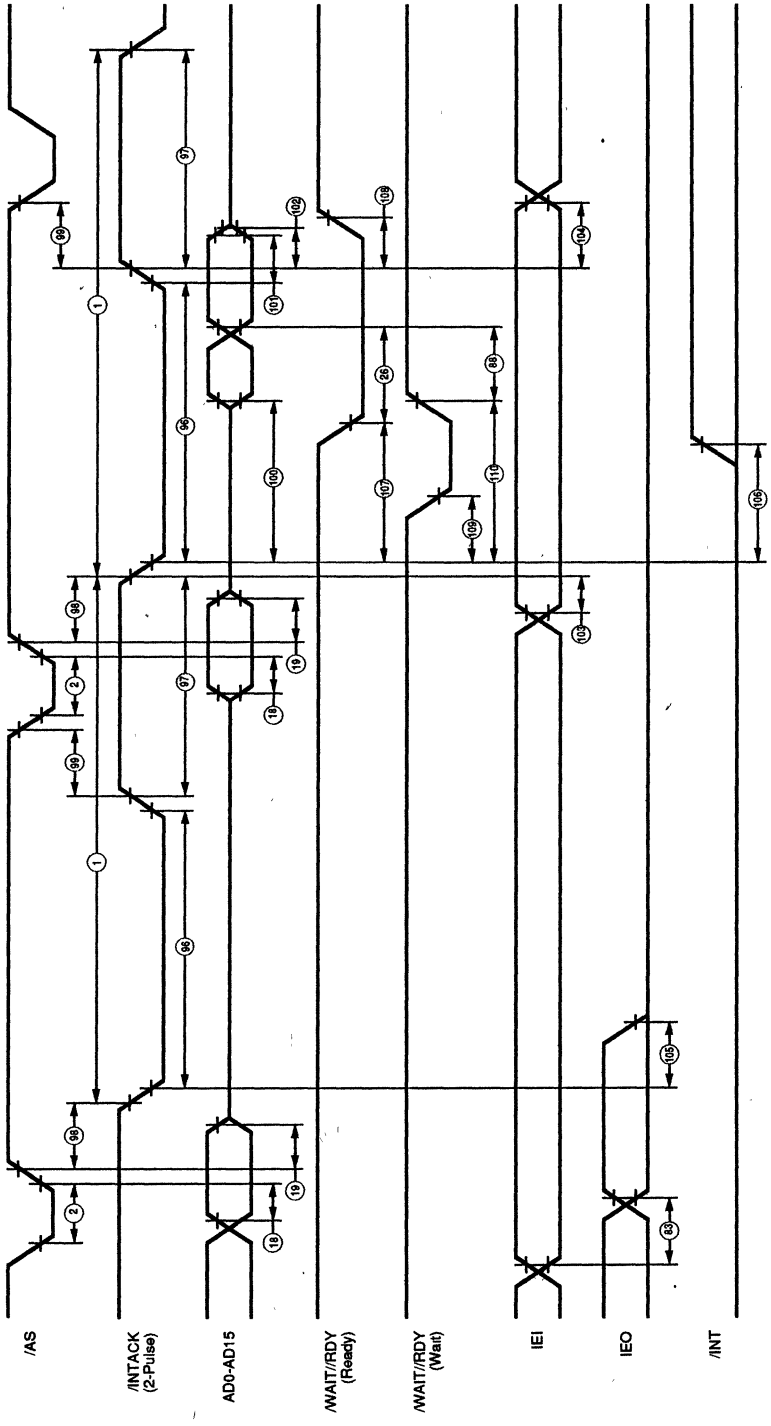


Figure 42. Multiplexed Double-Pulse INTACK Cycle

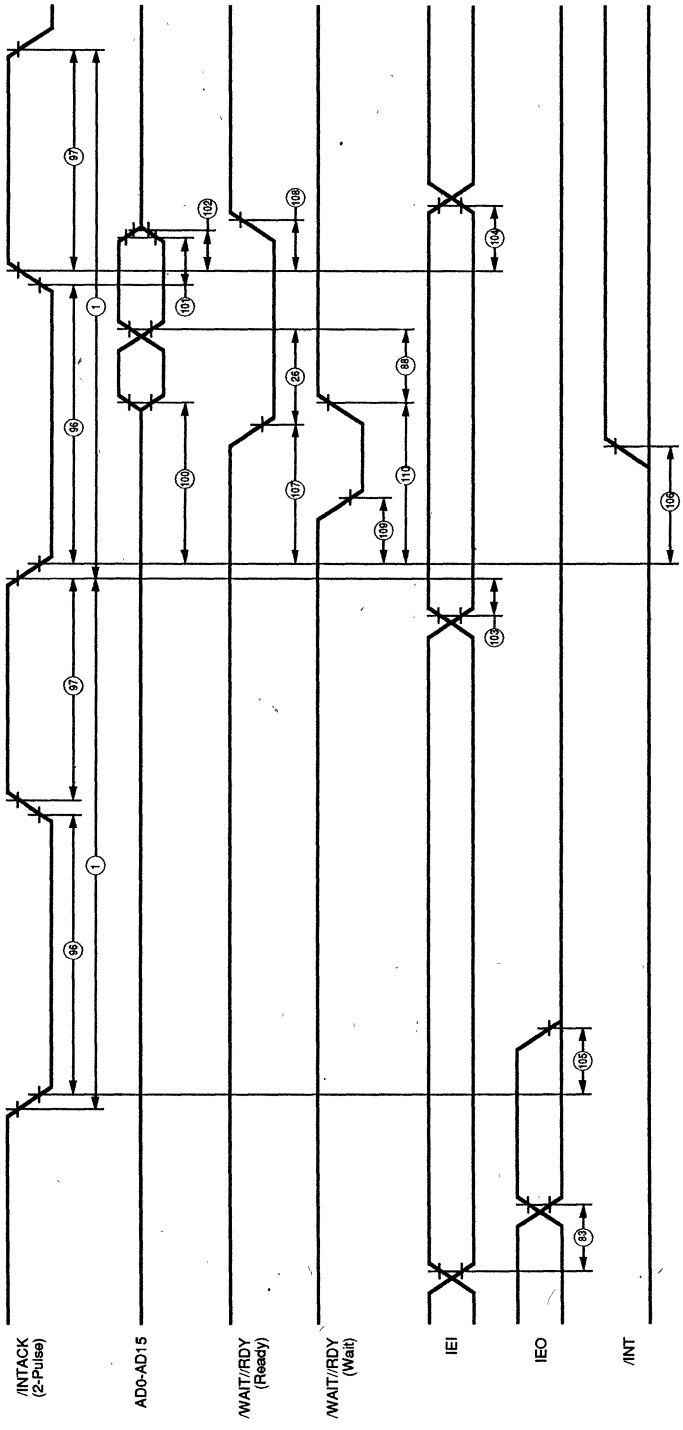


Figure 43. Non-multiplexed Double-Pulsed INTACK Cycle

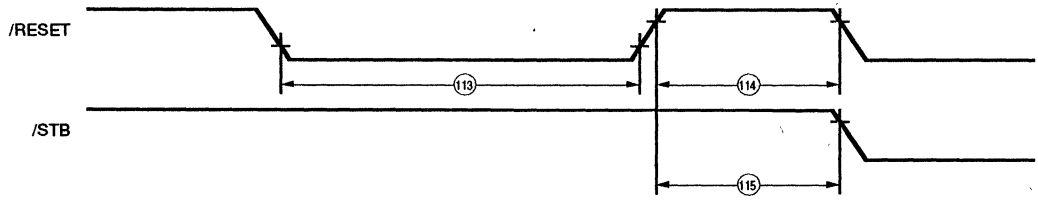


Figure 44. Reset

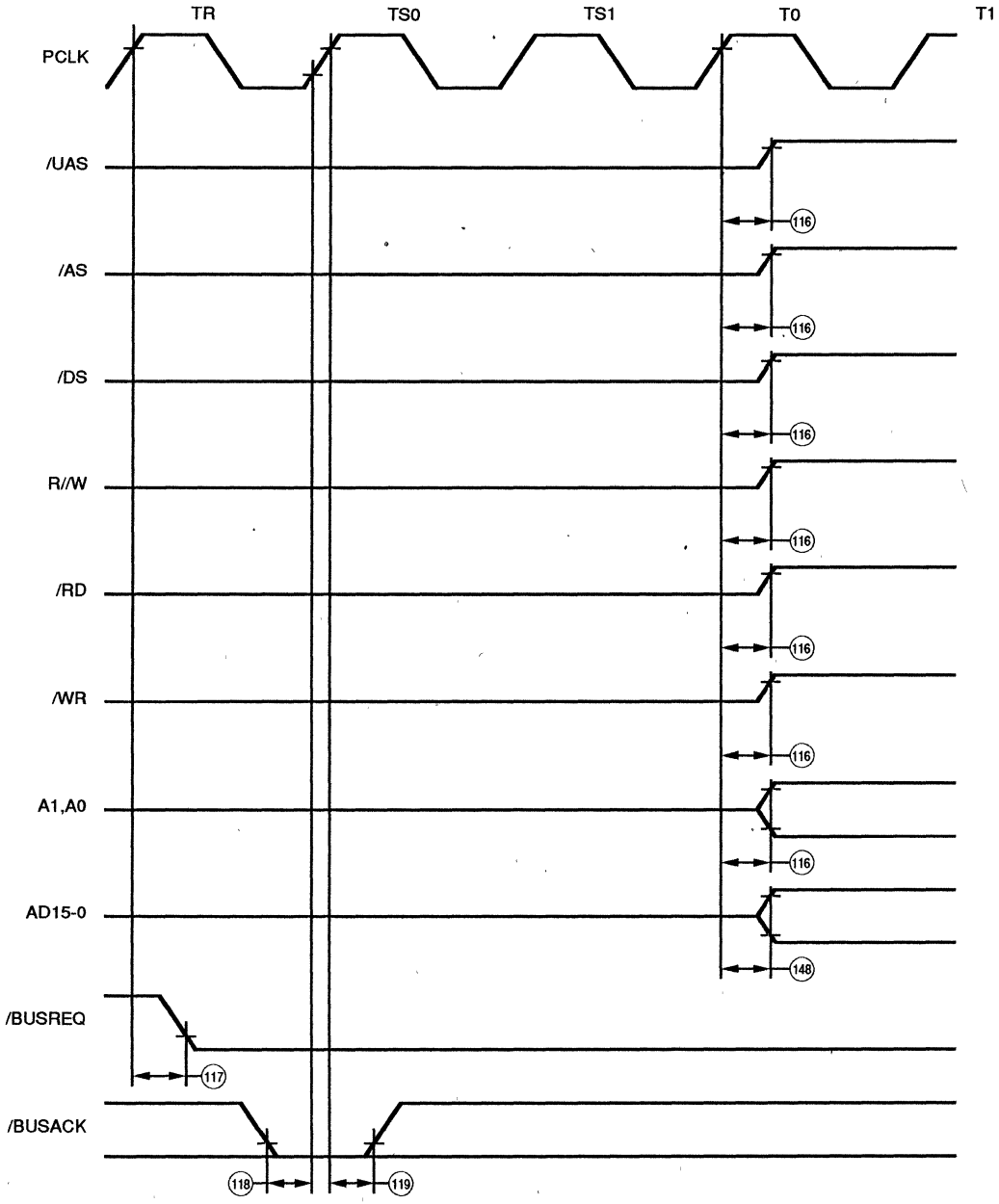


Figure 45. Z16C35 Start-up

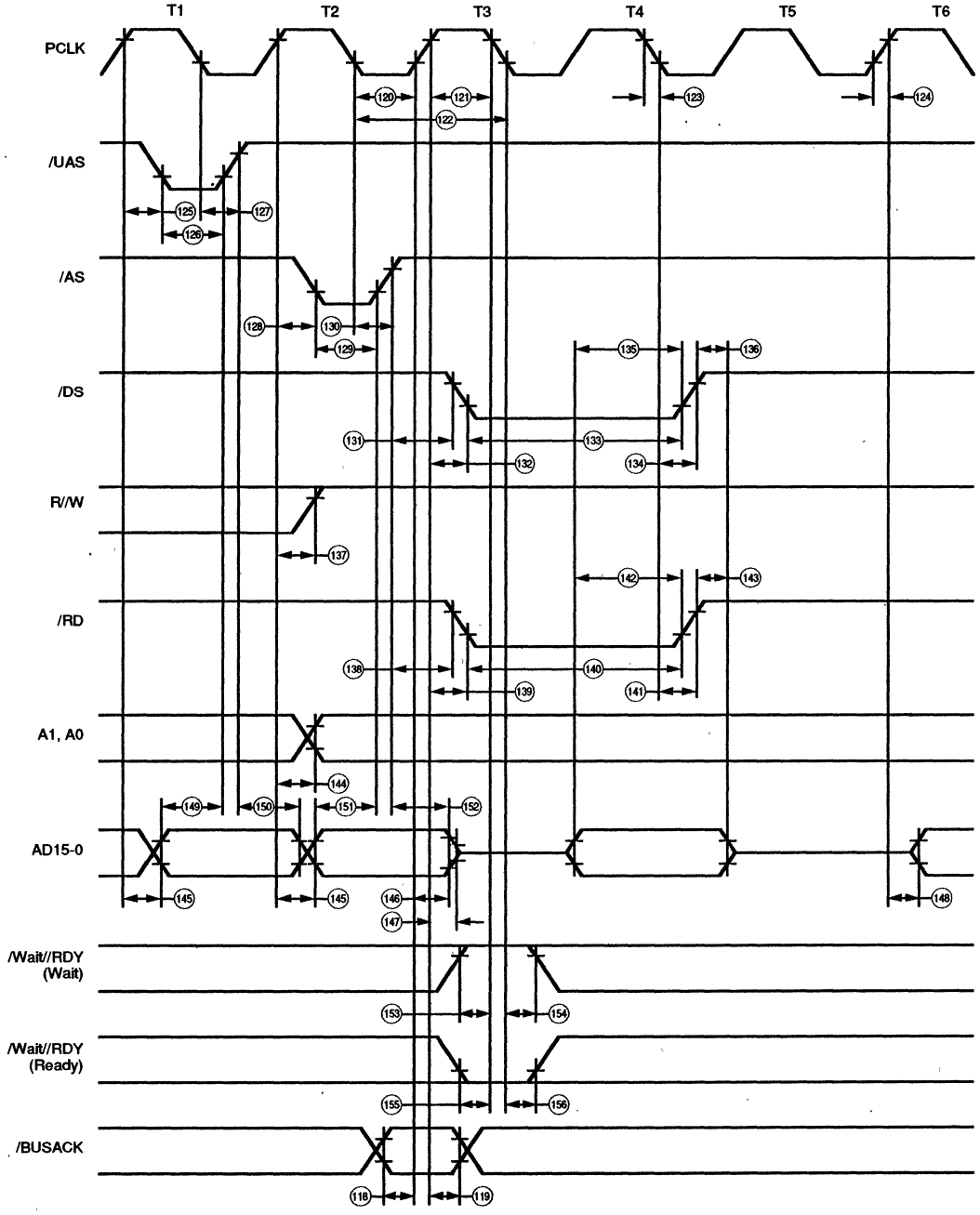


Figure 46. Z16C35 Memory Read

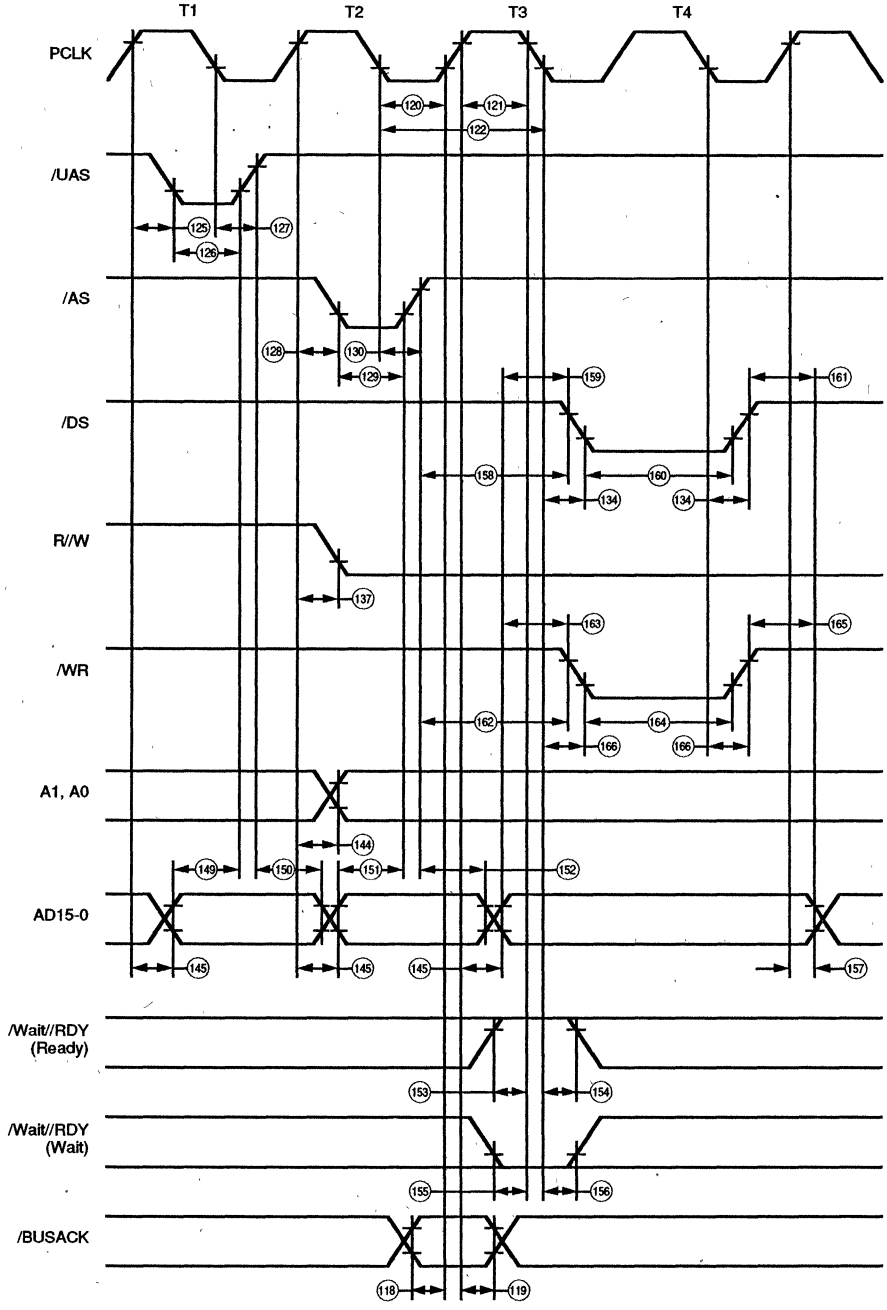


Figure 47. Z16C35 Memory Write

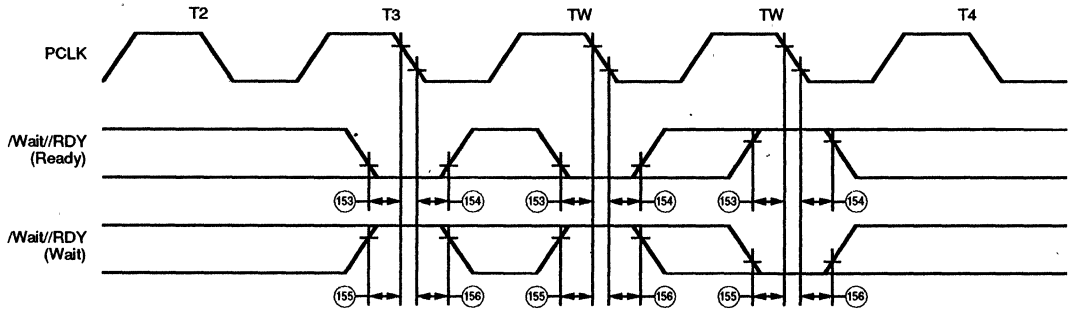


Figure 48. Wait and Ready Timing

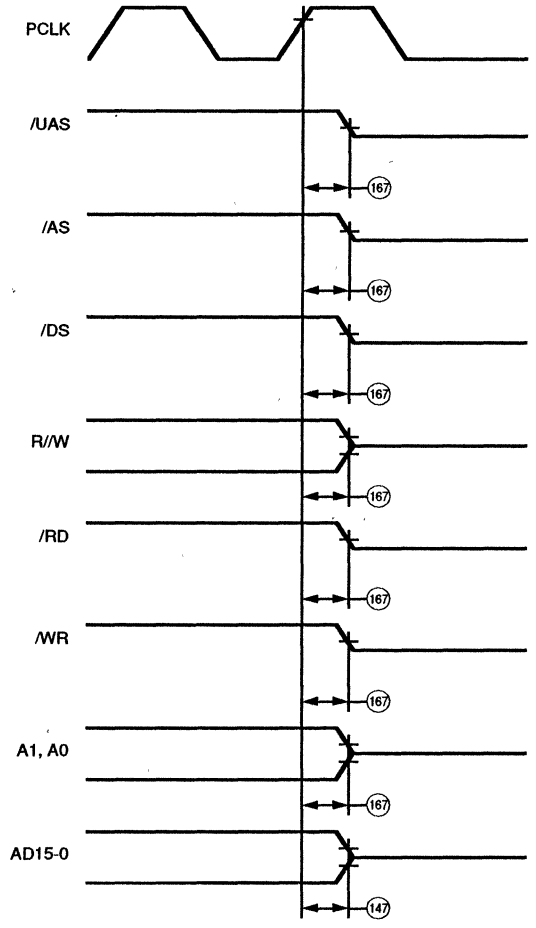


Figure 49. BUS Release

AC CHARACTERISTICS
General Timing

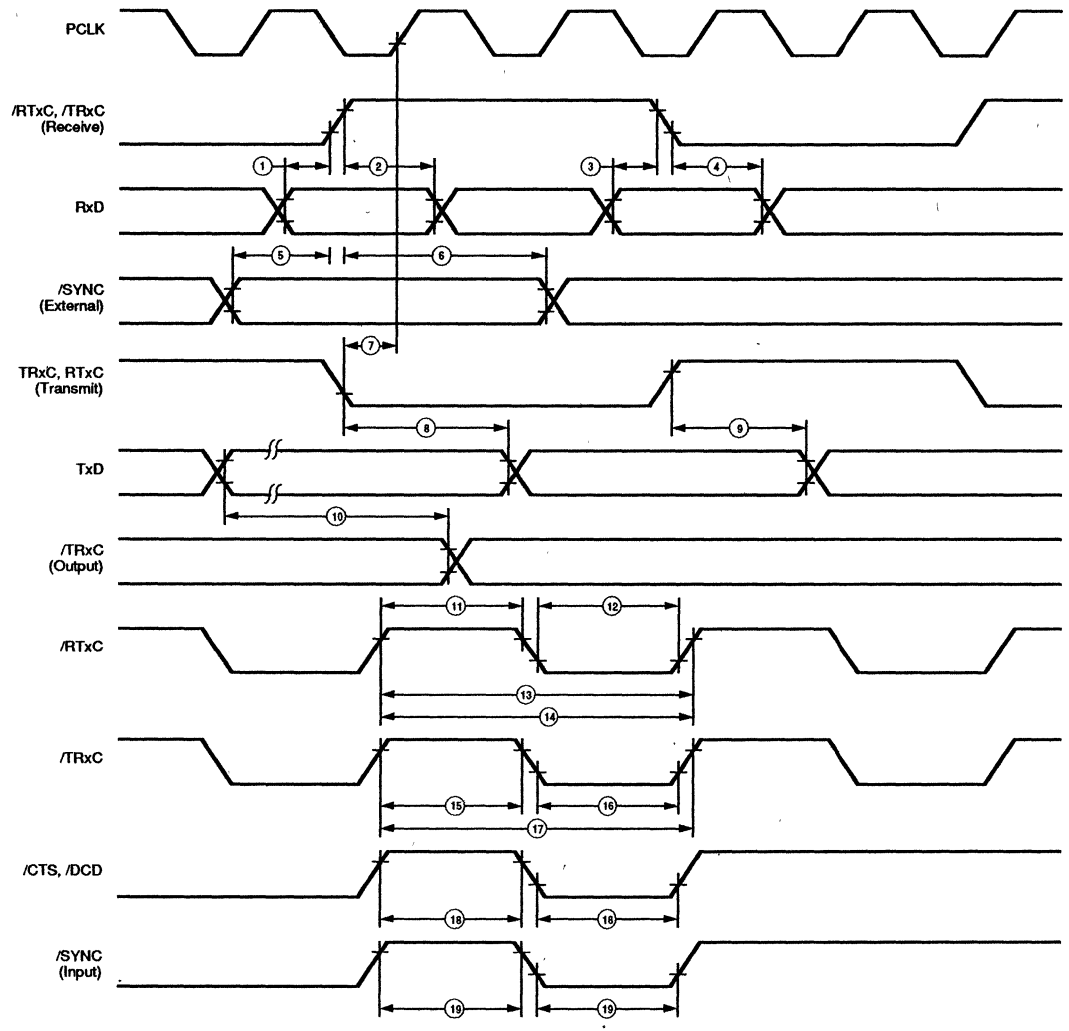


Figure 50. Z16C35 General Timing

AC CHARACTERISTICS
General Timing

No	Symbol	Parameter	10 MHz * Min	10 MHz * Max	16 MHz† Min	16 MHz† Max	Notes
1	TsRXD(RXCr)	RxD to /RxC Rise Setup Time (x1 mode)	0		0		[1]
2	ThRXD(RXCr)	RxD to /RxC Rise Hold Time (x1 mode)	150		60		[1]
3	TsRXD(RXCf)	RxD to /RxC Fall Setup Time (x1 mode)	0		0		[1,5]
4	ThRXD(RXCf)	RxD to /RxC Fall Hold Time (x1 mode)	150		60		[1,5]
5	TsSY(RXC)	/SYNC to /RxC Rise Setup Time	-200		-100		[1]
6	ThSY(RXC)	/SYNC to RxC Rise Hold Time	5TcPc		5TcPc		[1]
7	TsTXC(PC)	/TxC to PCLK Setup Time	0		0		[2,4]
8	TdTXCf(TXD)	/TxC Fall to TxD Delay (x1 mode)		150		85	[2]
9	TdTxCr(TXD)	/TxC Rise to TxD Delay (x1 mode)		150		85	[2,5]
10	TdTXD(TRX)	TxD to /TRXC Delay (Send Clock Echo)		200		80	
11	TwRTXh	/RTxC High Width	150		80		[6]
12	TwRTXl	/RTxC Low Width	150		80		[6]
13	TcRTX	/RTxC Cycle Time (RxD, TxD)	400		244		[6,7]
14	TcRTXX	Crystal Oscillator Period	100	1000	100	1000	[3]
15	TwTRXh	/TRxC High Width	150		80		[6]
16	TwTRXl	/TRxC Low Width	150		80		[6]
17	TcTRX	/TRxC Cycle Time (RxD, TxD)	400		244		[6,7]
18	TwEXT	/DCD or /CTS Pulse Width	200		70		
19	TwSY	/SYNC Pulse Width	200		70		

Notes:
[1] /RxC is /RTxC or /TRxC, whichever is supplying the receive clock.
[2] /TxC is /TRxC or /RTxC, whichever is supplying the transmit clock.
[3] Both /RTxC and /SYNC have 30 pf capacitors to ground connected to them.
[4] Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between /RxC and PCLK or/TxC and PCLK is required.
[5] Parameter applies only to FM encoding/decoding.
[6] Parameter applies only for transmitter and receiver; DPLL and baud rate generator requirements are identical to case PCLK requirements.
[7] The maximum receive or transmit data rate is one-fourth PCLK.

* Units in nanoseconds.
† 16MHz Timing is preliminary

AC CHARACTERISTICS
System Timing

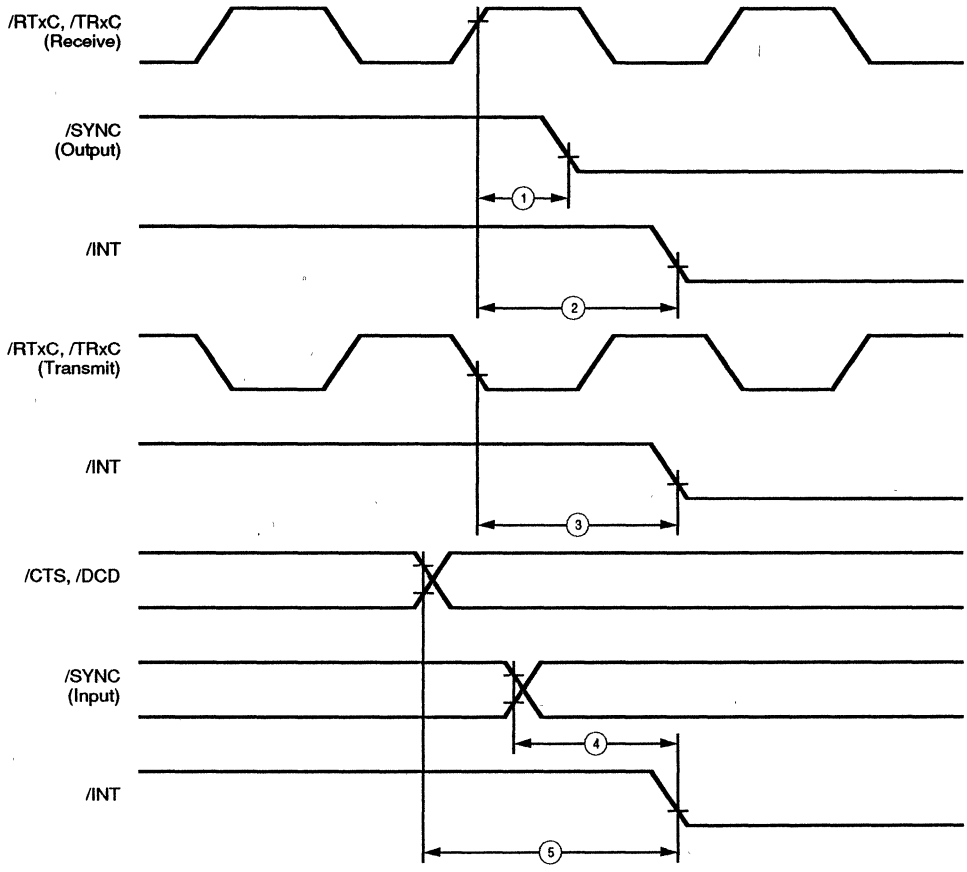


Figure 51. Z16C35 System Timing

AC CHARACTERISTICS
System Timing

No	Symbol	Parameter	10 MHz Min	10 MHz Max	16 MHz† Min	16 MHz† Max	Notes††
1	TdRXC(SY)	/RxC Rise to /SYNC	4	7	4	7	[1]
2	TdRXC(INT)	RxC Rise to /INT Valid Delay	10	16	10	16	[1]
3	TdTXC(INT)	/TxC Fall to /INT Valid Delay	6	10	6	10	
4	TdSY(INT)	/SYNC Transition to /INT Valid Delay	2	6	2	6	
5	TdEXT(INT)	/DCD or /CTS Transition to /INT Valid Delay	2	6	2	6	

Notes:
[1] /RxC is /RTXC or /TRxC, whichever is supplying the receive clock.
[2] /TxC is /TRxC or /RTxC, whichever is supplying the transmit clock.

† 16MHz Timing is preliminary
†† Units equal to TcPc.

Z16C50

DDPLL DUAL DIGITAL PHASE LOCKED LOOP MICROCONTROLLER

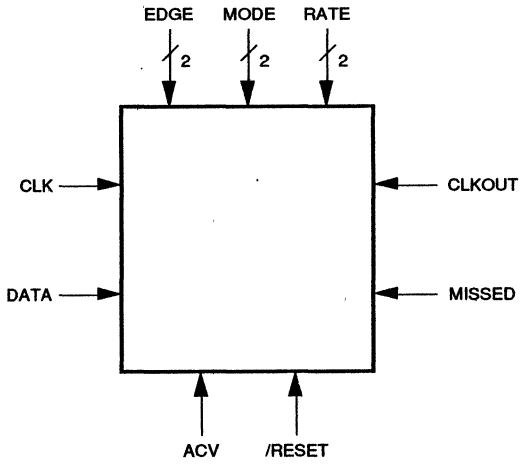
FEATURES

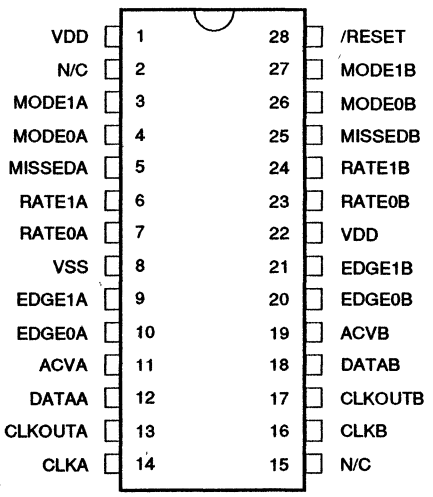
- Two independent Digital Phase Locked Loops in one package.
- 10 MHz and 20 MHz Clock operation
- Selectable clock rate, clock sampling edge, and data decoding.
- Synchronous status output
- Accept Code Violation input
- Implemented in 1.6µ CMOS technology
- 28-pin DIP package

GENERAL DESCRIPTION

The 16C50 DDPLL is a fully static CMOS device that packs two independent Digital Phase Locked Loops, with separate controls for selecting the decoding mode, clock rate, and synchronization edge, in one integrated package (Figure 1). The only common input between the two phase locked loops is /RESET (/ denotes active low signal).

The DDPLL is used in many communication applications requiring detection and extraction of clock from data. It can be used together with Serial Communication Controllers to allow operation at higher data rates. The data rate is programmable at 1/8, 1/16, or 1/32 clock rate. The DDPLL is offered in two speed grades: 10MHz and 20MHz maximum clock speed, which translates to a maximum data rate of 1.25 Mbps and 2.5 Mbps, respectively.





Note: Power connections follow conventional descriptions below:

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

Figure 1. DDPLL Block and Pin Diagrams

PIN DESCRIPTION

The following is a list of DDPLL pins and their descriptions. "A" and "B" at the end of pin names designate the signal connecting to the A- or B-channel of the DDPLL.

ACVA, ACVB. *Accept Code Violation* (input, active HIGH). The ACV signal is used to control the response of the DDPLL to code violations present in the received data stream.

CLKA, CLKB Clock. *Input* (input, active HIGH). The Clock runs at 8-, 16-, or 32-times the received data rate and is used by the DDPLL to generate the CLKOUT signal.

CLKOUTA, CLKOUTB. *Clock Output* (output, active HIGH). CLKOUT is the recovered clock for the receive data stream. The receiver should use this clock to sample and decode the received data.

DATAA, DATAB. *Receive Data* (input, active HIGH). The DATA signal is the received data stream that the DDPLL is attempting to synchronize with. The DDPLL will provide the CLKOUT signal for use by a receiver attempting to decode this data stream.

EDGE0A, EDGE0B, EDGE1A, EDGE1B. *Adjust/Synchronize Edge Controls* (input, active HIGH). These signals select which edge is used by the DDPLL to achieve and maintain synchronization.

MISSEDA, MISSEDB. *Clock Missed* (output, active HIGH). MISSED signal is activated when the DDPLL detects missing edge(s) in the data stream.

MODE0A, MODE0B, MODE1A, MODE1B. *DDPLL Mode Controls* (input, active HIGH). MODE0 and MODE1 are used to control the mode of operation of the DDPLL with respect to the encoded format of the incoming data.

RATE0A, RATE0B, RATE1A, RATE1B. *Clock Rate Selects* (input, active HIGH). RATE inputs are used to select the data rate divisor to generate the DDPLL clock.

/RESET. *Reset* (input, active LOW). This input resets the DDPLL to a known state and must be active for at least two cycles of the slowest CLK signal. This is the only common input to the two Digital Phase Locked Loops.

VDD. +5V supply.

VSS. 0V (GND) supply.

FUNCTIONAL DESCRIPTION

Prior to device operation, the control inputs of the DDPLL must be set to known states corresponding to the desired mode of operation.

Data decoding format is programmed via MODE0 and MODE1 inputs. Table 1 demonstrates the truth table for these inputs. NRZ, NRZI, FM1 (biphase mark), FM0 (biphase space) and Manchester (biphase level) formats are supported. MODE1-MODE0 of LOW-LOW disables the DDPLL, setting the CLKOUT output LOW. In NRZ format, a "1" is represented by a HIGH level and a "0" is represented by a LOW level. In NRZI format, a "1" is represented by no change in level and a "0" is represented by a change in level. A MODE1-MODE0 of LOW-HIGH selects the NRZ or NRZI decoding modes

Table 1. Mode Selection Truth Table

MODE1	MODE0	Selected Mode
0	0	Disable/Sync
0	1	NRZ/NRZI
1	0	Biphase-Mark/Space
1	1	Biphase-Level

In both of these modes, transitions on the input may only occur on bit cell boundaries and the DDPLL provides CLKOUT to match these bit cell boundaries. In FM1 (biphase mark) and FM0 (biphase space) formats, a transition occurs at the beginning of every bit cell. In addition to this, in FM1, a "1" is represented by an additional transition at the center of the bit cell and a "0" is represented by the absence of such transition. In contrast, in FM0, a "0" is represented by an additional transition at the center of the bit cell and a "1" is represented by the absence of such transition. MODE1-MODE0 of HIGH-LOW selects the biphase-mark (FM1) or biphase-space (FM0) modes.

In Manchester (biphase level) mode, a transition occurs at the center of every bit cell. If the bit is "1", the transition is HIGH to LOW, and if the bit is "0", the transition is LOW to HIGH. Additionally, a LOW to HIGH transition occurs at the boundary of a "1" bit. A HIGH-HIGH selects the Manchester (biphase level) mode. Figure 2 demonstrates an example of a serial data stream with its corresponding encoded waveforms in NRZ, NRZI, FM1, FM0, and Manchester modes.

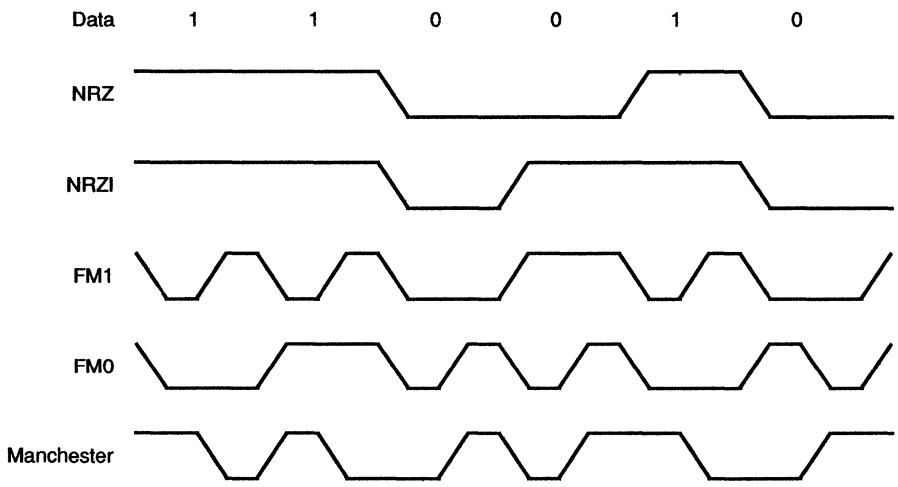


Figure 2. Data Decoding Formats

Clock rate is programmed through RATE1-RATE0 inputs. Clock rate can be set for 8-, 16-, or 32-times the data rate With maximum clock operation of 20MHz in 8X mode, data rates of 2.5Mbps is achieved. Table 2 illustrates clock rate divisor's truth table. Note that RATE1-RATE0 of HIGH-HIGH is illegal. All DDPLL inputs (with the exception of /RESET) are sampled by the rising edge of CLK and all outputs change state in response to the rising edge of the CLK signal The two DPLLs are completely independent except for the /RESET input.

Table 2. Data Rate Divisor Truth Table

RATE1	RATE0	Data Rate Divisor
0	0	32X Clock Mode
0	1	16X Clock Mode
1	0	8X Clock Mode
1	1	Not Allowed

EDGE1 and EDGE0 select the edge(s) in the receive data stream used by the DDPLL to achieve and maintain synchronization Table 3 shows how the rising edge, the falling edge, or both edges of the receive data stream can be used for synchronization. A HIGH on both EDGE inputs inhibits the DDPLL from using either edge for synchronization. As far as the DDPLL is concerned, edges that are not used to achieve or maintain synchronization are not present. They are reported as missing edges when they occur where an edge is expected

Table 3. Clock Edge Selection Truth Table

EDGE1	EDGE0	Selected Edge
0	0	Both Edges
0	1	Rising Edge
1	0	Falling Edge
1	1	Adjust/Sync Inhibit

The response of the DDPLL to code violations present in the received data stream can be controlled using the ACV input. This signal is ignored in the NRZ/NRZI mode, where code violations are not possible In all other modes, however, a HIGH on the ACV allows the DDPLL to recognize an isolated code violation without losing synchronization. Code violations are then used by the receiver for synchronization.

When the DDPLL detects missing edge(s) in the data stream, it activates the MISSED output. If the DDPLL is configured to accept code violations, two consecutive code violations will activate the MISSED output. If the DDPLL is configured not to accept code violations, this output is activated on any missing clock. MISSED will never be activated in the NRZ/NRZI modes of operation, as code violations are not possible in these modes. The DDPLL re-enters the sync-up phase when the MISSED output is activated.

The CLKOUT is the recovered Clock for the receive data stream. The receiver uses this clock to sample and decode the received data.

The only common input between the two phase locked loops in the DDPLL is the /RESET input. This signal must remain active for at least two cycles of the slowest CLK signal and resets the device to a known state: MISSED=LOW and CLKOUT=LOW. Synchronization attempt begins once the /RESET signal is deactivated.

ABSOLUTE MAXIMUM RATINGS

Voltages on all pins
with respect to GND ..... -0.3V to +7.0V
Operating Ambient
Temperature ..... See Ordering Information
Storage Temperature ..... -65°C to +150°C

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 (Figure 3). All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

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

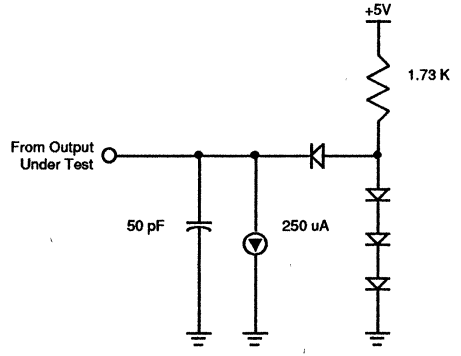


Figure 3. Standard Test Load

DC CHARACTERISTICS

Symbol	Parameter	Min	Max	Units	Conditions
V_{IH}	Input High Voltage	2.0	V_{DD}+0.3	V	
V_{IL}	Input Low Voltage	-0.3	0 8	V	
V_{OL}	Output High Voltage	2.4		V	1OH = -1.6mA
I_{IL}	Input Leakage Current		±10	µA	$0.4V \le V_{IN} \le 2.4V$
I_{CC}	Supply Current		40	mA	V_{DD}=5,V_{IH}=4.8V, V_{IL}=0.2V
C_{IN}	Input Capacitance		10	pf	Unmeasured pins returned to GND
C_{OUT}	Output Capacitance		15	pf	Unmeasured pins returned to GND

Notes:

1. V_{DD}=5V ±10% unless otherwise specified, over specified temperature range.
2. Capacitance values specified at f=1MHz.

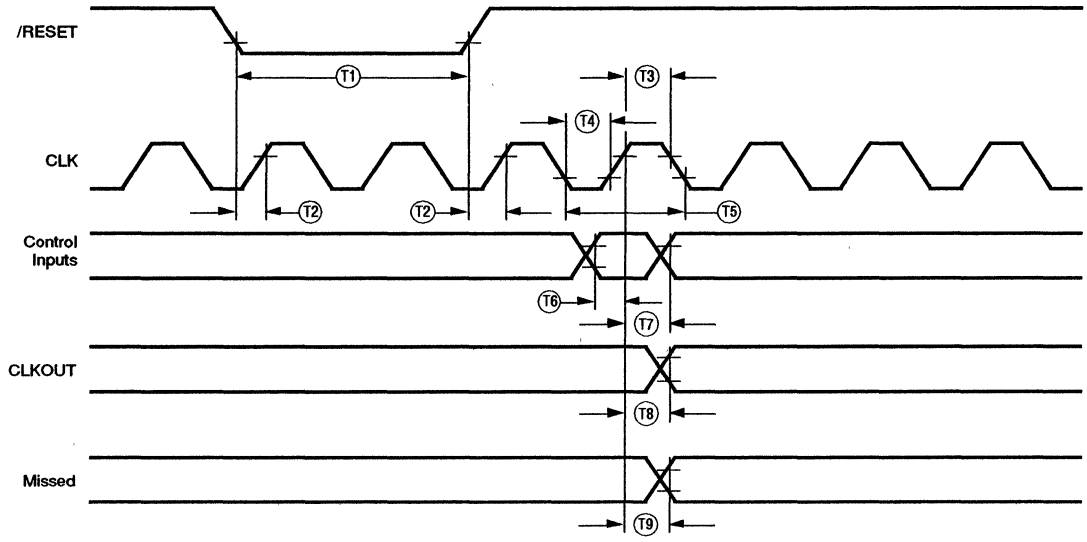


Figure 4. DDPLL Timing Diagram

Figure 2 illustrates the DDPLL timings. "Control Inputs" in Figure 4 refer to MODE1-MODE0, RATE1-RATE0, EDGE1-EDGE0, and ACV inputs.

AC CHARACTERISTICS

Timing	Symbol	Parameter	Z16C5010 Min	Z16C5010 Max	Z16C5020 Min	Z16C5020 Max	Units	Notes
T1	TwRESI	/RESET LOW Width	2TcC		2TcC			
T2	TsRES(CLK)	/RESET to CLK Setup Time	15		15		ns	
T3	TwCLKh	CLK HIGH Time	40		20		ns	
T4	TwCLKl	CLK Low Time	40		20		ns	
T5	TcC	CLK Cycle Time	100		50		ns	
T6	TsIN(CLK)	Input Valid to CLK Setup Time	15		15		ns	1,2,3
T7	ThIN(CLK)	Input Valid to CLK Hold Time	10		10		ns	1,2,3
T8	TdCLK(OUT)	CLK to CLKOUT Delay Time		30		30	ns	
T9	TdCLK(MIS)	CLK to MISSED Delay Time		30		30	ns	

PRODUCT SPECIFICATION

Z5380 SCSI

SMALL COMPUTER SYSTEM INTERFACE

FEATURES

- Compatible 5380 pinout
- Low power CMOS
- Asynchronous interface, supports 1.5 MB/s
- Direct SCSI Bus interface with on-board 48 mA drivers
- Supports Target and Initiator roles
- Arbitration support
- DMA or programmed I/O data transfers
- Supports Normal or Block Mode DMA
- Memory or I/O Mapped CPU interface

GENERAL DESCRIPTION

The Z5380 SCSI (Small Computer System Interface) controller is a 40-pin DIP or 44-pin PLCC CMOS device (Figure 1). It is designed to implement the SCSI protocol as defined by the ANSI X3.131-1986 standard, and is fully compatible with the industry standard 5380. It is capable of operating both as a Target and as an Initiator. Special high-current open-drain outputs enable it to directly interface to, and drive, the SCSI bus. The Z5380 has the necessary interface hook-ups so the system CPU can communicate with it like with any other peripheral device. The CPU can read from, or write to, the SCSI registers which are addressed as standard or memory-mapped I/Os.

The Z5380 increases the system performance by minimizing the CPU intervention in DMA operations which the SCSI controls. The CPU is interrupted by the SCSI when it detects a bus condition that requires attention. It also supports arbitration and reselection. The Z5380 has the proper hand-shake signals to support normal and block mode DMA operations with most DMA controllers available (Figure 2).

Note: 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); /N//S (NORMAL and SYSTEM are both active Low).

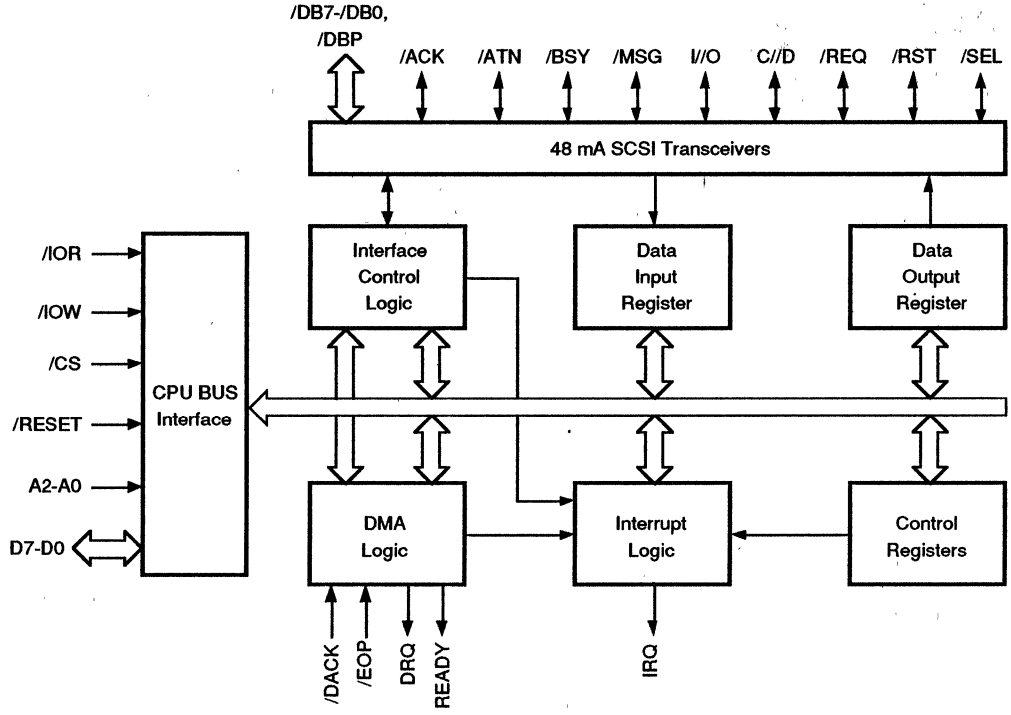
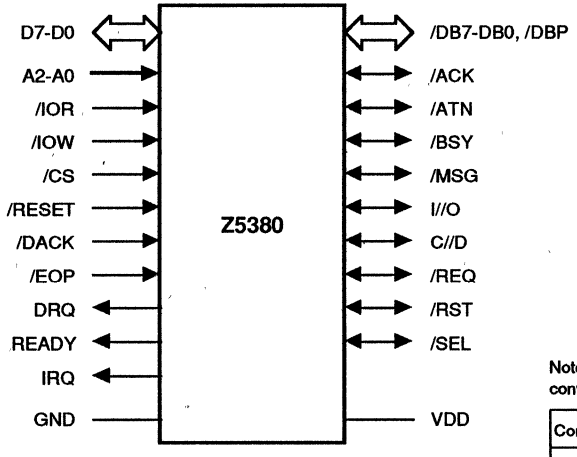


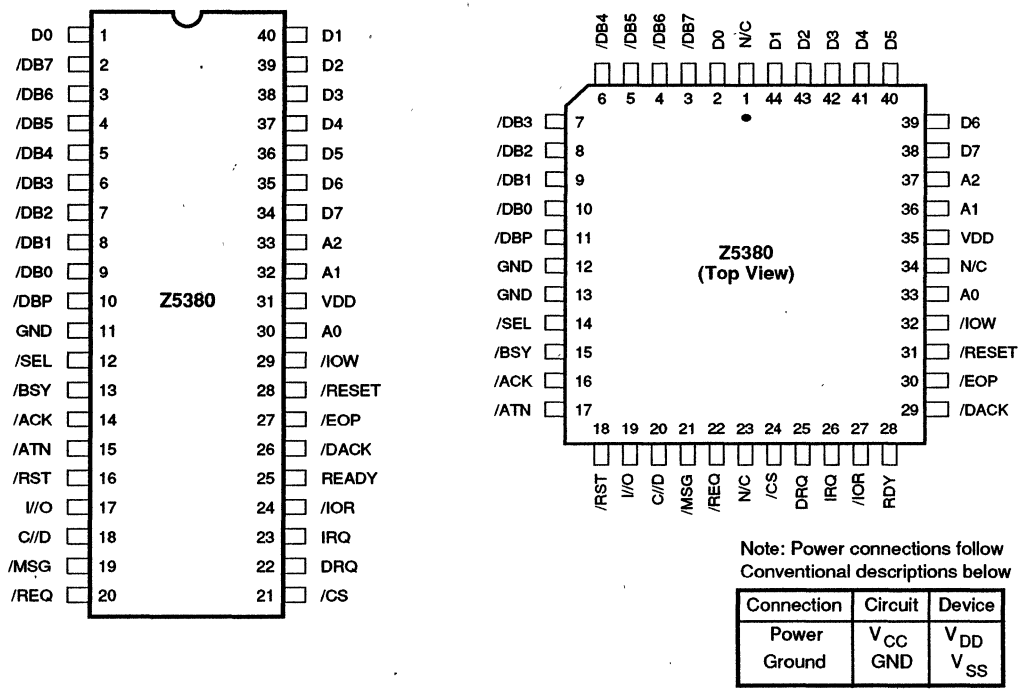
Figure 1. Block Diagram



Note: Power connections follow conventional descriptions below:

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

Figure 2. Logic Symbol



Note: Power connections follow Conventional descriptions below

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

Figure 3. Pin Diagrams

PIN DESCRIPTION

Microprocessor Bus

Figure 3 shows the pins and their respective functions for both the DIP and PLCC.

A2-A0. *Address Lines* (Input). Address lines are used with /CS, /IOR, or /IOW to address all internal registers.

/CS. *Chip Select* (Input, Active Low). This signal, in conjunction with /IOR or /IOW, enables the internal register selected by A2-A0, to be read from or written to.

/DACK. *DMA Acknowledge* (Input, Active Low). /DACK resets DRQ and selects the data register for input or output data transfers. /DACK is used by DMA controller instead of /CS

DRQ. *DMA Request* (Output, Active High) DRQ indicates that the data register is ready to be read or written DRQ is asserted only if DMA mode is set in the Command Register. DRQ is cleared by /DACK.

D7-D0. *Data Lines* (Bidirectional, three-state, Active High). Bidirectional microprocessor data bus lines D0 is the Least Significant Bit of the bus. Data bus lines carry data and commands to and from the SCSI.

/EOP. *End of Process* (Input, Active Low). /EOP is used to terminate a DMA transfer. If asserted during a DMA cycle, the current byte will be transferred, but no additional bytes will be requested.

/IOR. *I/O Read* (Input, Active Low). /IOR is used in conjunction with /CS and A2-A0 to read an internal register. It also selects the Input Data Register when used with /DACK.

/IOW. *I/O Write* (Input, Active Low). /IOW is used in conjunction with /CS and A2-A0 to write an internal register. It also selects the Output Data Register when used with /DACK.

PIN DESCRIPTION (Continued)

IRQ. *Interrupt Request* (Output, Active High). IRQ alerts a microprocessor of an error condition or an event completion

READY. *Ready* (Output, Active High). Ready is used to control the speed of Block Mode DMA transfers. This signal goes active to indicate the chip is ready to send/receive data and remains Low after a transfer until the last byte is sent or until the DMA Mode bit is reset.

/RESET. *Reset* (Input, Active Low) /RESET clears all registers. It has no effect upon the SCSI /RST signal.

SCSI Bus

The following signals are all bidirectional, active Low, open-drain, with 48 mA sink capability. All pins interface directly with the SCSI bus.

/ACK. *Acknowledge* (Bidirectional, Open-drain, Active Low). Driven by an Initiator, /ACK indicates an acknowledgement for a /REQ//ACK data-transfer handshake. In the Target role, /ACK is received as a response to the /REQ signal.

/ATN. *Attention* (Bidirectional, Open-drain, Active Low). Driven by an Initiator, received by the Target, /ATN indicates an Attention condition.

/BSY. *Busy* (Bidirectional, Open-drain, Active Low) This signal indicates that the SCSI bus is being used and can be driven by both the Initiator and the Target device

C//D. *Control/Data* (Bidirectional, Open-drain). Driven by the Target and received by the Initiator, C//D indicates whether Control or Data information is on the Data Bus True indicates Control.

/DB7-/DB0, /DBP. *Data Bus Bits, Data Bus Parity Bit* (Bidirectional, Open-drain). These eight data bits (/DB7-/DB0), plus a parity bit (/DBP) form the data bus. /DB7 is the most significant bit (MSB) and has the highest priority during the Arbitration phase. Data parity is odd. Parity is always generated and optionally checked. Parity is not valid during Arbitration.

I//O. *Input/Output* (Bidirectional, Open-drain). I/O is a signal driven by a Target which controls the direction of data movement on the SCSI bus. True indicates input to the Initiator. This signal is also used to distinguish between Selection and Reselection phases.

/MSG. *Message* (Bidirectional, Open-drain, Active Low). This signal is driven by the Target during the Message phase. This signal is received by the Initiator.

/REQ. *Request* (Bidirectional, Open-drain, Active Low). Driven by the Target and received by the Initiator, this signal indicates a request for a /REQ//ACK data-transfer handshake.

/RST. *SCSI Bus Reset* (Bidirectional, Open-drain, Active Low). This signal indicates a SCSI bus Reset condition.

/SEL. *Select* (Bidirectional, Open-drain, Active Low). This signal is used by an Initiator to select a Target, or by a Target to reselect an Initiator.

Power Signals:

GND. Ground (0V)

VDD. VDD Supply (+5V)

FUNCTIONAL DESCRIPTION

The Z5380 Small Computer System Interface (SCSI) has a set of eight registers that are controlled by the CPU. By reading and writing the appropriate registers, the CPU may initiate any SCSI Bus activity or may sample and assert any signal on the SCSI Bus. This allows the user to implement all or any of the SCSI protocol in software. These registers are read (written) by activating /CS with an address on A2-A0 and then issuing an /IOR (/IOW) pulse This section describes the operation of the internal registers (Table 1).

Table 1. Register Summary

Address				
A2	A1	A0	R/W	Register Name
0	0	0	R	Current SCSI Data
0	0	0	W	Output Data
0	0	1	R/W	Initiator Command
0	1	0	R/W	Mode
0	1	1	R/W	Target Command
1	0	0	R	Current SCSI Bus Status
1	0	0	W	Select Enable
1	0	1	R	Bus and Status
1	0	1	W	Start DMA Send
1	1	0	R	Input Data
1	1	0	W	Start DMA Target Receive
1	1	1	R	Reset Parity/Interrupt
1	1	1	W	Start DMA Initiator Receive

Data Registers

The data registers are used to transfer SCSI commands, data, status, and message bytes between the microprocessor Data Bus and the SCSI Bus The Z5380 does not interpret any information that passes through the data registers. The data registers consist of the transparent Current SCSI Data Register, the Output Data Register, and the Input Data Register.

Current SCSI Data Register. *Address 0* (Read Only). The Current SCSI Data Register (Figure 4) is a read-only register which allows the microprocessor to read the active SCSI Data Bus. This is accomplished by activating /CS with an address on A2-A0 of 000 and issuing an /IOR pulse. If parity checking is enabled, the SCSI Bus parity is checked at the beginning of the read cycle This register is used during a programmed I/O data read or during Arbitration to check for higher priority arbitrating devices. Parity is not guaranteed valid during Arbitration.

Output Data Register. *Address 0* (Write Only). The Output Data Register (Figure 5) is a write-only register that is used to send data to the SCSI Bus. This is accomplished by either using a normal CPU write, or under DMA control, by using /IOW and /DACK. This register also asserts the proper ID bits on the SCSI Bus during the Arbitration and Selection phases.

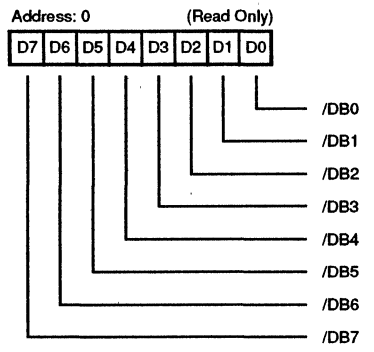


Figure 4. Current SCSI Data Register

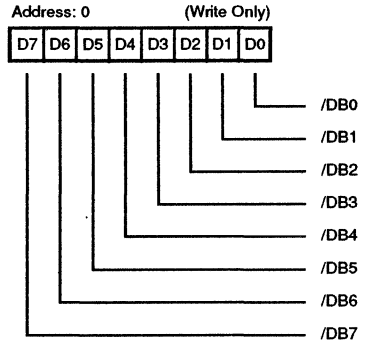


Figure 5. Output Data Register

Initiator Command Register. *Address 1* (Read/Write). The Initiator Command Register (Figures 6 and 7) are read and write registers which assert certain SCSI Bus signals, monitors those signals, and monitors the progress of bus arbitration. Many of these bits are significant only when being used as an Initiator; however, most can be used during Target role operation

FUNCTIONAL DESCRIPTION (Continued)

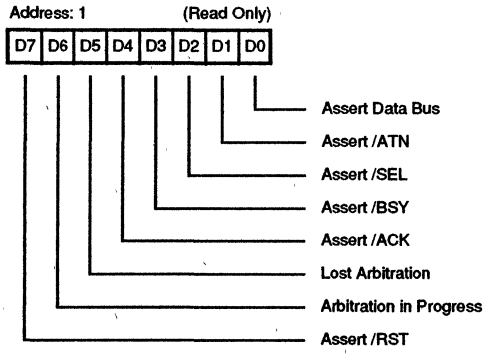


Figure 6. Initiator Command Register

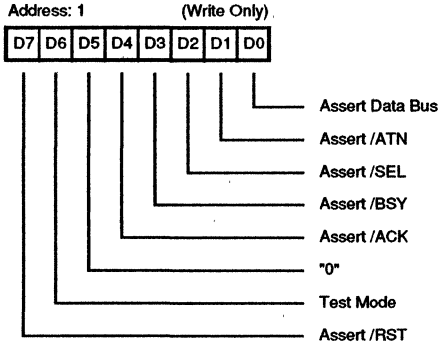


Figure 7. Initiator Command Register

The following describes the operation of all bits in the Initiator Command Register:

Bit 0. *Assert Data Bus.* This bit, when set, allows the contents of the Output Data Register to be enabled as chip outputs on the signals /DB7-DB0. Parity is also generated and asserted on /DBP.

When connected as an Initiator, the outputs are only enabled if the Target Mode bit (Mode Register, bit 6) is 0, the received signal I/O is False, and the phase signals (C/D, I/O, and /MSG) match the contents of the Assert C/D, Assert I/O, and Assert /MSG in the Target Command Register.

This bit should also be set during DMA send operations.

Bit 1. *Assert /ATN.* /ATN may be asserted on the SCSI Bus by setting this bit to a one (1) if the Target Mode bit (Mode Register, bit 6) is False. /ATN is normally asserted by the initiator to request a Message Out bus phase. Note that since Assert /SEL and Assert /ATN are in the same register, a select with /ATN may be implemented with one CPU write. /ATN may be deasserted by resetting this bit to zero. A read of this register simply reflects the status of this bit.

Bit 2. *Assert /SEL.* Writing a one (1) into this bit position asserts /SEL onto the SCSI Bus. /SEL is normally asserted after Arbitration has been successfully completed. /SEL may be disabled by resetting bit 2 to a zero. A read of this register reflects the status of this bit.

Bit 3. *Assert /BSY.* Writing a one (1) into this bit position asserts /BSY onto the SCSI Bus. Conversely, a zero resets the /BSY signal. Asserting /BSY indicates a successful selection or reselection. Resetting this bit creates a Bus-Disconnect condition. Reading this register reflects bit status.

Bit 4. *Assert /ACK.* Bit 4 is used by the bus initiator to assert /ACK on the SCSI Bus. In order to assert /ACK, the Target Mode bit (Mode Register, bit 6) must be False. Writing a zero to this bit deasserts /ACK. Reading this register reflects bit status.

Bit 5. *"0"* (Write Bit). Bit 5 should be written with a zero for proper operation.

Bit 5. *LA* (Lost Arbitration - Read Bit). Bit 5, when active, indicates that the SCSI detected a Bus-Free condition, arbitrated for use of the bus by asserting /BSY and its ID on the Data Bus, and lost Arbitration due to /SEL being asserted by another bus device. This bit is active only when the Arbitrate bit (Mode Register, bit 0) is active.

Bit 6. *Test Mode* (Write Bit). Bit 6 is written during a test environment to disable all output drivers, effectively removing the Z5380 from the circuit. Resetting this bit returns the part to normal operation.

Bit 6. *AIP* (Arbitration in Process - Read Bit). Bit 6 is used to determine if Arbitration is in progress. For this bit to be active, the Arbitrate bit (Mode Register, bit 0) must have been set previously. It indicates that a Bus-Free condition has been detected and that the chip has asserted /BSY and put the contents of the Output Data Register onto the SCSI Bus. AIP will remain active until the Arbitrate bit is reset.

Bit 7. *Assert /RST.* Whenever a one is written to bit 7 of the Initiator Command Register, the /RST signal is asserted on the SCSI Bus. The /RST signal will remain asserted until this bit is reset or until an external /RESET occurs. After this bit

is set (1), IRQ goes active and all internal logic and control registers are reset (except for the interrupt latch and the Assert /RST bit). Writing a zero to bit 7 of the Initiator Command Register deasserts the /RST signal. The status of this bit is monitored by reading the Initiator Command Register.

Mode Register. *Address 2* (Read/Write). The Mode Register controls the operation of the chip. This register determines whether the Z5380 operates as an Initiator or a Target, whether DMA transfers are being used, whether parity is checked, and whether interrupts are generated on various external conditions. This register is read to check the value of these internal control bits (Figure 8).

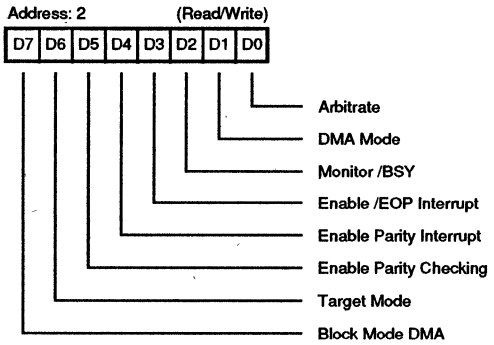


Figure 8. Mode Register

The following describes the operation of all bits in the Initiator Command Register:

Bit 0. *Arbitrate.* The Arbitrate bit is set (1) to start the Arbitration process. Prior to setting this bit, the Output Data Register should contain the proper SCSI device ID value. Only one data bit should be active for SCSI Bus Arbitration. The Z5380 waits for a Bus-Free condition before entering the Arbitration phase. The results of the Arbitration phase is determined by reading the status bits LA and AIP (Initiator Command Register, bits 5 and 6, respectively)

Bit 1. *DMA Mode.* The DMA Mode bit is normally used to enable a DMA transfer and must be set (1) prior to writing Start DMA Send Register, Start DMA Target Register, and Start DMA Initiator Receiver Register. These three registers are used to start DMA transfers. The Target Mode bit (Mode Register, bit 6) must be consistent with writes to Start DMA Target Receive and Start DMA Initiator Receive Registers; i.e., set (1) for a write to Start DMA Target Receive Register and set (0) for Start DMA Initiator Receive Register. The control bit Assert Data Bus (Initiator Command Register, bit 0) must be True (1) for all DMA send operations. In the DMA mode, /REQ and /ACK are automatically controlled.

The DMA Mode bit is not reset upon the receipt of an /EOP signal. Any DMA transfer is stopped by writing a zero into this bit location; however, care must be taken not to cause /CS and /DACK to be active simultaneously.

Bit 2. *Monitor Busy.* The Monitor Busy bit, when True (1), causes an interrupt to be generated for an unexpected loss of /BSY. When the interrupt is generated due to loss of /BSY, the lower six bits of the Initiator Command Register are reset (0) and all signals are removed from the SCSI Bus.

Bit 3. *Enable /EOP interrupt.* The enable /EOP interrupt bit, when set (1), causes an interrupt to occur when the /EOP (End Of Process) signal is received from the DMA controller logic.

Bit 4. *Enable Parity Interrupt.* The Enable Parity Interrupt bit, when set (1), will cause an interrupt (IRQ) to occur if a parity error is detected. A parity interrupt will only be generated if the Enable Parity Checking bit (bit 5) is also enabled (1).

Bit 5. *Enable Parity Checking* The Enable Parity Checking bit determines whether parity errors are ignored or saved in the parity error latch. If this bit is reset (0), parity is ignored. Conversely, if this bit is set (1), parity errors are saved.

Bit 6. *Target Mode.* The Target Mode bit allows the Z5380 to operate as a SCSI Bus Initiator or Target. With this bit reset (0), the Z5380 operates as a SCSI Bus Initiator. Setting Target Mode bit to 1 programs the Z5380 to ope ate as a SCSI Bus Target device. If the signals /ATN and /ACK are to be asserted on the SCSI Bus, the Target Mode bit must be reset (0). If the signals C//D, I//O, /MSG, and /REQ are to be asserted on the SCSI Bus, the Target Mode bit must be set (1).

Bit 7. *Block Mode DMA.* The Block Mode DMA bit controls the characteristics of the DMA DRQ-/DACK handshake. When this bit is reset (0) and the DMA Mode bit is active (1), the DMA handshake uses the normal interlocked handshake, and the rising edge of /DACK indicates the end of each byte being transferred. In Block Mode operation, when the Block Mode DMA bit is set (1) and DMA Mode bit is active (1), the end of /IOR or /IOW signifies the end of each byte transferred and /DACK is allowed to remain active throughout the DMA operation. Ready can then be used to request the next transfer.

FUNCTIONAL DESCRIPTION (Continued)

Target Command Register. *Address 3* (Read/Write). When connected as a target device, the Target Command Register (Figure 9) allows the CPU to control the SCSI Bus Information Transfer phase and/or to assert /REQ by writing this register. The Target Mode bit (Mode Register, bit 6) must be True (1) for bus assertion to occur. The SCSI Bus phases are described in Table 2.

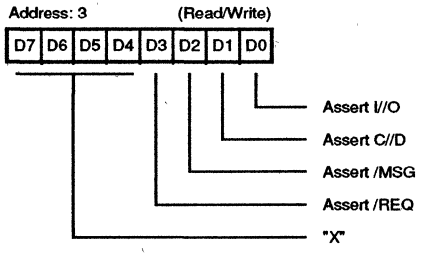


Figure 9. Target Command Register

Table 2. SCSI Information Transfer Phases

Bus Phase	Assert I//O	Assert C//D	Assert /MSG
Data Out	0	0	0
Unspecified	0	0	1
Command	0	1	0
Message Out	0	1	1
Data In	1	0	0
Unspecified	1	0	1
Status	1	1	0
Message In	1	1	1

When connected as an Initiator with DMA Mode bit True, if the phase lines (I//O, C//D, and /MSG) do not match the phase bits in the Target Command Register, a phase mismatch interrupt is generated when /REQ goes active. To send data as an Initiator, the Assert I//O, Assert C//D, and Assert /MSG bits must match the corresponding bits in the Current SCSI Bus Status Register. The Assert /REQ bit (bit 3) has no meaning when operating as an Initiator.

Bits 4, 5, 6, and 7 are not used.

Current SCSI Bus Status Register. *Address 4* (Read Only). The Current SCSI Bus Register is a read-only register which is used to monitor seven SCSI Bus control signals, plus the Data Bus parity bit. For example, an Initiator device can use this register to determine the current bus phase and to poll /REQ for pending data transfers. This register may also be used to determine why a particular interrupt occurred. Figure 10 describes the Current SCSI Bus Status Register.

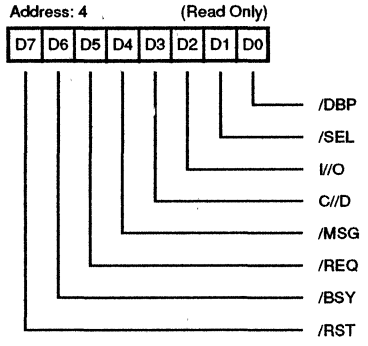


Figure 10. Current SCSI Bus Status Register

Select Enable Register. *Address 4* (Write Only) The Select Enable Register (Figure 11) is a write-only register which is used as a mask to monitor a signal ID during a selection attempt. The simultaneous occurrence of the correct ID bit, /BSY False, and /SEL True causes an interrupt. This interrupt can be disabled by resetting all bits in this register. If the Enable Parity Checking bit (Mode Register, bit 5) is active (1), parity is checked during selection.

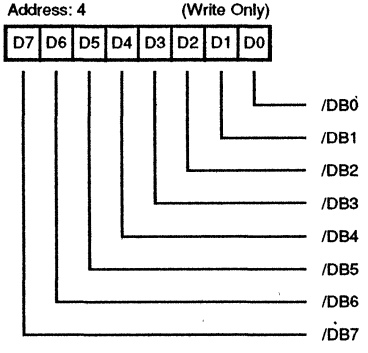


Figure 11. Select Enable Register

Bus and Status Register. *Address 5* (Read Only). The Bus and Status Register (Figure 12) is a read-only register which can be used to monitor the remaining SCSI control signals not found in the Current SCSI Bus Status Registers (/ATN and /ACK), as well as six other status bits. The following describes each bit of the Bus and Status Register individually.

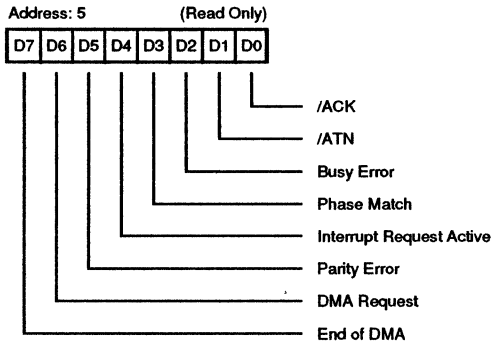


Figure 12. Bus and Status Register

Bit 0. */ACK.* Bit 0 reflects the condition of the SCSI Bus control signal /ACK. This signal is normally monitored by the Target device.

Bit 1. */ATN.* Bit 1 reflects the condition of the SCSI Bus control signal /ATN. This signal is normally monitored by the Target device.

Bit 2. *Busy Error.* The Busy Error bit is active if an unexpected loss of the /BSY signal has occurred. This latch is set whenever the Monitor Busy bit (Mode Register, bit 2) is True and /BSY is False An unexpected loss of /BSY disables any SCSI outputs and resets the DMA Mode bit (Mode Register, bit 1).

Bit 3. *Phase Match.* The SCSI signals /MSG, C//D, and I//O, represent the current information Transfer phase. The Phase Match bit indicates whether the current SCSI Bus phase matches the lower 3 bits of the Target Command Register. Phase Match is continuously updated and is only significant when operating as a Bus Initiator. A phase match is required for data transfers to occur on the SCSI Bus.

Bit 4.*Interrupt Request ACTIVE.* Bit 4 is set if an enabled interrupt condition occurs. It reflects the current state of the IRQ output and can be cleared by reading the Reset Parity/ Interrupt Register

Bit 5. *Parity Error.* Bit 5 is set if a parity error occurs during a data receive or a device selection. The Parity Error bit can only be set (1) if the Enable Parity Check bit (Mode Register, bit 5) is active (1). This bit may be cleared by reading the Reset Parity/Interrupt Register.

Bit 6. *DMA Request.* The DMA Request bit allows the CPU to sample the output pin DRQ. DRQ can be cleared by asserting /DACK or by resetting the DMA Mode bit (bit 1) in the Mode Register. The DRQ signal does not reset when a phase-mismatch interrupt occurs.

Bit 7. *End of DMA Transfer.* The End of DMA Transfer bit is set if /EOP, /DACK, and either /IOR or /IOW are simultaneously active for at least 100ns. Since the /EOP signal can occur during the last byte sent to the Output Data Register, the /REQ and /ACK signals should be monitored to ensure that the last byte has been transferred. This bit is reset when the DMA Mode bit is reset (0) in the Mode Register.

Input Data Register. *Address 6* (Read Only). The input Data Register (Figure 13) is a read-only register that is used to read latched data from the SCSI Bus. Data is latched either during a DMA Target receive operation when /ACK goes active or during a DMA Initiator receive when /REQ goes active The DMA Mode bit (bit 1) must be set before data can be latched in the Input Data Register. This register is read under DMA control using /IOR and /DACK. Parity is optionally checked when the Input Data Register is loaded.

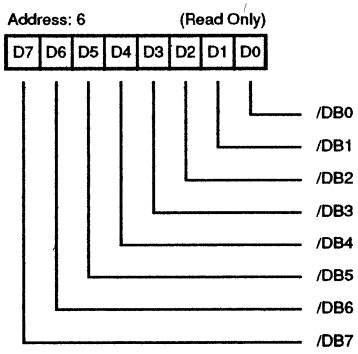


Figure 13. Input Data Register

FUNCTIONAL DESCRIPTION (Continued)

DMA Registers

Three write-only registers are used to initiate all DMA activity They are: Start DMA Send, Start DMA Target Receive, and Start DMA Initiator Receive. Performing a write operation into one of these registers starts the desired type of DMA transfer. Data presented to the Z5380 on signals D7-D0 during the register write is meaningless and has no effect on the operation. Prior to writing these registers, the Block Mode DMA bit (bit 7), the DMA Mode bit (bit 1), and the Target Mode bit (bit 6) in the Mode Register must be appropriately set. The individual registers are briefly described as follows:

Start DMA Send. *Address 5* (Write Only). This register is written to initiate a DMA send, from the DMA to the SCSI Bus, for either Initiator or Target role operations. The DMA Mode bit (Mode Register, bit 1) is set prior to writing this register.

Start DMA Target Receive. *Address 6* (Write Only) This register is written to initiate a DMA receive - from the SCSI Bus to the DMA, for Target operation only. The DMA Mode bit (bit 1) and the Target Mode bit (bit 6) in the Mode Register must both be set (1) prior to writing this register.

Start DMA Initiator Receive. *Address 7* (Write Only). This register is written to initiate a DMA receive - from the SCSI Bus to the DMA, for Initiator operation only. The DMA Mode bit (bit 6) must be False (0) in the Mode Register prior to writing this register.

Reset Parity/Interrupt. *Address 7* (Read Only). Reading this register resets the Parity Error bit (bit 5), the Interrupt Request bit (bit 4), and the Busy Error bit (bit 2) in the Bus and Status Register.

On-Chip SCSI Hardware Support

The Z5380 is easy to use because of its simple architecture. The chip allows direct control and monitoring of the SCSI Bus by providing a latch for each signal. However, portions of the protocol define timings which are much too quick for traditional microprocessors to control. Therefore, hardware support has been provided for DMA transfers, bus arbitration, phase change monitoring, bus disconnection, bus reset, parity generation, parity checking, and device selection/reselection.

Arbitration is accomplished using a bus-free filter to continuously monitor /BSY If /BSY remains inactive for at least 1.2us, the SCSI Bus is considered free and Arbitration may begin Arbitration will begin if the bus is free, /SEL is inactive, and the Arbitrate bit (Mode Register, bit 0) is active. Once arbitration has begun (/BSY asserted), an arbitration delay of 2.2us must elapse before the Data Bus can be examined to determine if Arbitration is enabled This delay is implemented in the controlling software driver.

The Z5380 is a clockwise device. Delays such as bus-free delay, bus-set delay, and bus-settle delay are implemented using gate delays. These delays may differ between devices because of inherent process variations, but are well within the proposed ANSI X3.131 - 1986 specification.

Interrupts

The Z5380 provides an interrupt output (IRQ) to indicate a task completion or an abnormal bus occurrence. The use of interrupts is optional and may be disabled by resetting the appropriate bits in the Mode Register or the Select Enable Register.

When an interrupt occurs, the Bus and Status Register and the Current SCSI Bus Status Register (Figures 12 and 10) must be read to determine which condition created the interrupt IRQ can be reset simply by reading the Reset Parity/Interrupt Register or by an external chip reset /RESET active for 200ns.

Assuming the Z5380 has been properly initialized, an interrupt is generated if the chip is selected or reselected; if an /EOP signal occurs during a DMA transfer; if a SCSI Bus reset occurs; if a parity error occurs during a data transfer; if a bus phase mismatch occurs; or if a SCSI Bus disconnection occurs.

Selection/Reselection Interrupt

The Z5380 generates a select interrupt if /SEL is active (0), its device ID is True and /BSY is False for at least a bus-settle delay If I/O is active, this is considered a reselect interrupt. The correct ID bit is determined by a match in the Select Enable Register. Only a single bit match is required to generate an interrupt. This interrupt may be disabled by writing zeros into all bits of the Select Enable Register.

If parity is supported, parity should be good during the selection phase. Therefore, if the Enable Parity bit (Mode Register, bit 5) is active, the Parity Error bit is checked to ensure that a proper selection has occurred. The Enable Parity Interrupt bit need not be set for this interrupt to be generated.

The proposed SCSI specification also requires that no more than two device ID's be active during the selection process. To ensure this, the Current SCSI Data Register is read.

The proper values for the Bus and Status Register and the Current SCSI Bus Status Register are displayed in Figures 14 and 15, respectively.

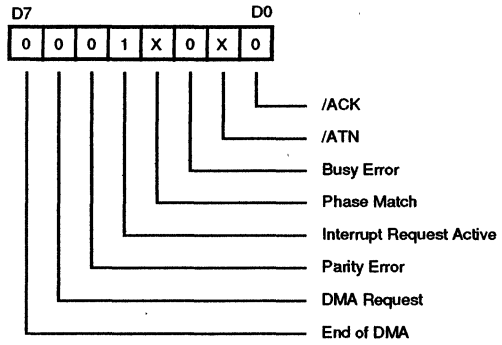


Figure 14. Bus and Status Register

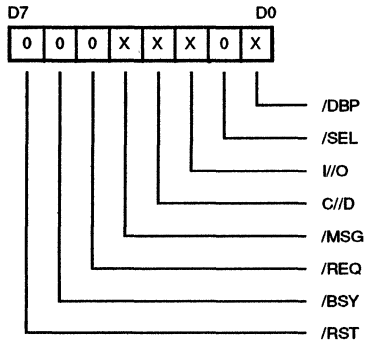


Figure 15. Current SCSI Bus Status Register

End Of Process (EOP) Interrupt

An End Of Process signal (EOP) which occurs during a DMA transfer (DMA Mode True) will set the End of DMA Status bit (bit 7) and will optionally generate an interrupt if Enable EOP Interrupt bit (Mode Register, bit 3) is True. The /EOP pulse will not be recognized (End of DMA bit set) unless /EOP, /DACK, and either /IOR or /IOW are concurrently active for at least 100 ns. DMA transfers can still occur if /EOP was not asserted at the correct time. This interrupt is disabled by resetting the Enable EOP Interrupt bit.

The proper values for the Bus and Status Register and the Current SCSI Bus Status Register for this interrupt are shown in Figures 16 and 17.

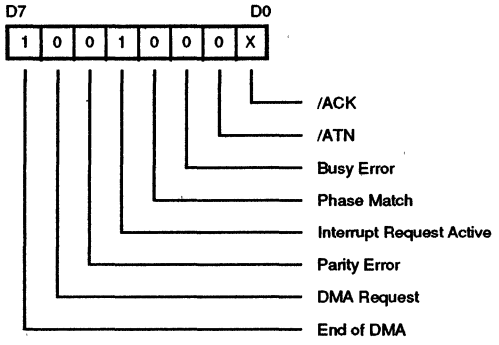


Figure 16. Bus and Status Register

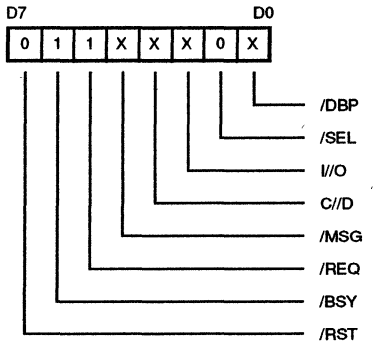


Figure 17. Current SCSI Bus Status Register

The End of DMA bit is used to determine when a block transfer is complete. Receive operations are complete when there is no data left in the chip and no additional handshakes occurring. The only exception to this is receiving data as an Initiator and the Target opts to send additional data for the same phase. In this /REQ goes active and the new data is present in the Input Data Register. Since a phase-mismatch interrupt will not occur, /REQ and /ACK need to be sampled to determine that the Target is attempting to send more data.

For send operations, the End of DMA bit is set when the DMA finishes its transfers, but the SCSI transfer may still be in progress. If connected as a Target, /REQ and /ACK should be sampled until both are False. If connected as an Initiator, a phase change interrupt is used to signal the completion of the previous phase. It is possible for the Target to request additional data for the same phase. In this case, a phase change will not occur and both /REQ and /ACK are sampled to determine when the last byte was transferred.

SCSI Bus Reset Interrupt

The Z5380 generates an interrupt when the /RST signal transitions to True. The device releases all bus signals within a bus-clear delay of this transition. This interrupt also occurs after setting the Assert /RST bit (Initiator Command Register, bit 7). This interrupt cannot be disabled. (Note: /RST is not latched in bit 7 of the Current SCSI Bus Status Register and is not active when this port is read. For this case, the Bus Reset interrupt is determined by default.)

The proper values for the Bus and Status Register and the Current SCSI Bus Status Register are displayed in Figures 18 and 19, respectively.

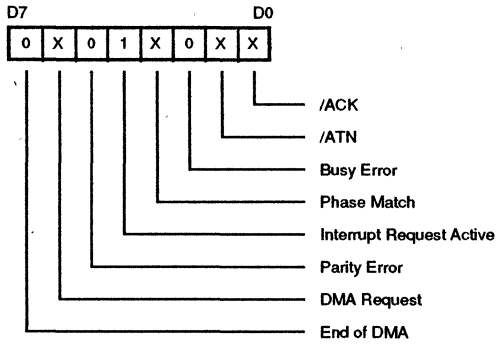


Figure 18. Bus and Status Register

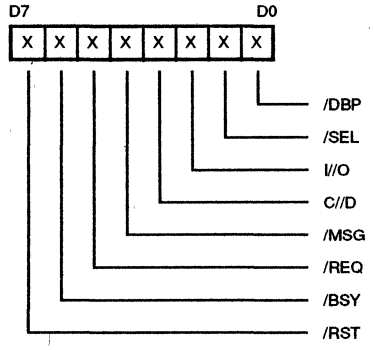


Figure 19. Current SCSI Bus Status Register

Parity Error Interrupt

An Interrupt is generated for a received parity error if the Enable Parity Check (bit 5) and the Enable Parity Interrupt (bit 4) bits are set (1) in the Mode Register. Parity is checked during a read of the Current SCSI Data Register and during a DMA receive operation. A parity error can be detected without generating an interrupt by disabling the Enable Parity Interrupt bit and checking the Parity Error flag (Bus and Status Register, bit 5).

The proper values for the Bus and Status Register and the Current SCSI Bus Status Register are displayed in Figures 20 and 21, respectively.

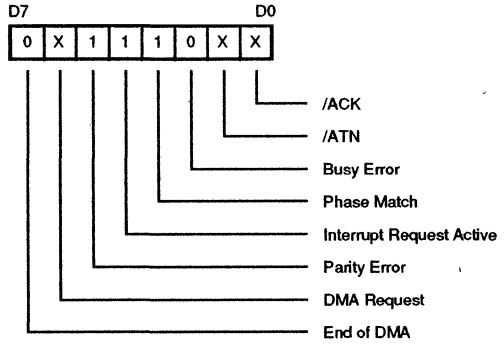


Figure 20. Bus and Status Register

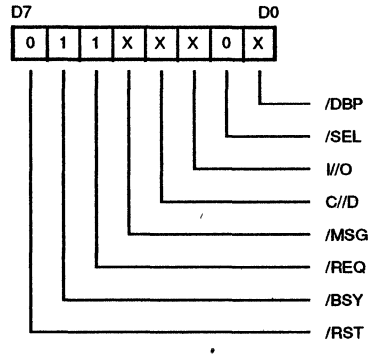


Figure 21. Current SCSI Bus Status Register

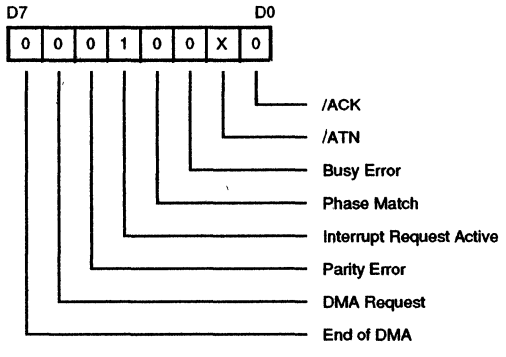


Figure 22. Bus and Status Register

Bus Phase Mismatch Interrupt

The SCSI phase lines are comprised of the signals I//O, C//D, and /MSG. These signals are compared with the corresponding bits in the Target Command Register: Assert I//O (bit 0), Assert C//D (bit 1), and Assert /MSG (bit 2). The comparison occurs continually and is reflected in the Phase Match bit (bit 3) of the Bus and Status Register. If the DMA Mode bit (Mode Register, bit 1) is active and a phase mismatch occurs when /REQ transitions from False to True, an interrupt (IRQ) is generated.

A phase mismatch prevents the recognition of /REQ and removes the chip from the bus during an Initiator send operation (/DB7-/DB0 and /DBP will not be driven even through the Assert Data Bus bit (Initiator Command Register, bit 0) is active). This may be disabled by resetting the DMA Mode bit (Note: It is possible for this interrupt to occur when connected as a Target if another device is driving the phase lines to a different state)

The proper values for the Bus and Status Register and the Current SCSI Bus Status Register are displayed in Figures 22 and 23, respectively.

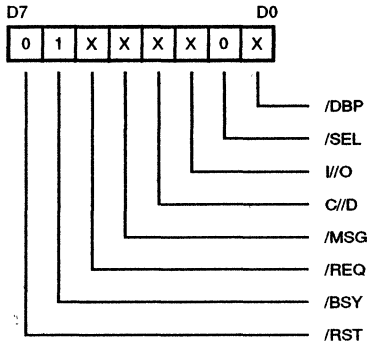


Figure 23. Current SCSI Bus Status Register

Loss of BSY Interrupt

If the Monitor Busy bit (bit 2) in the Mode Register is active, an interrupt is generated if the BSY signal goes False for at least a bus-settle delay. This interrupt is disabled by resetting the Monitor Busy bit. Register values are displayed in Figures 24 and 25.

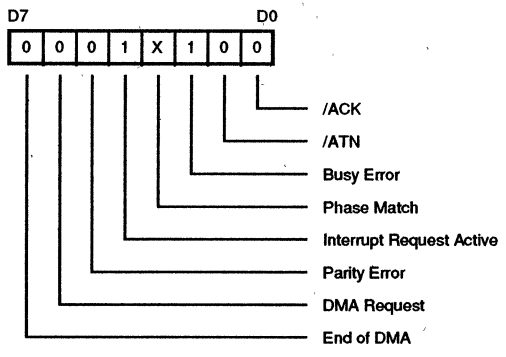


Figure 24. Bus and Status Register

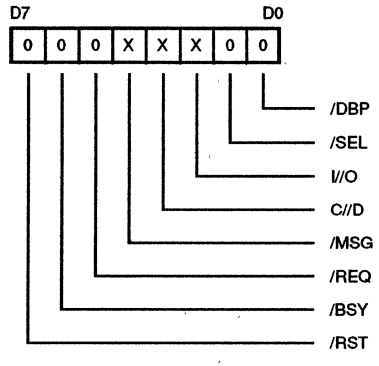


Figure 25. Current SCSI Bus Status Register

Reset Conditions
Three possible reset situations exist with the Z5380, as follows:

Hardware Chip Reset
When the signal /RST is active for at least 200 ns, the Z5380 device is re-initialized and all internal logic and control registers are cleared. This is a chip reset only and does not create a SCSI Bus-Reset condition.

SCSI Bus Reset (/RST) Received
When a SCSI /RST signal is received, an IRQ interrupt is generated and a chip reset is performed. All internal logic and registers are cleared, except for the IRQ interrupt latch and the Assert /RST bit (bit 7) in the Initiator Command Register. (Note: The /RST signal may be sampled by reading the Current SCSI Bus Status Register; however, this signal is not latched and may not be present when this port is read).

SCSI Bus Reset (/RST) Issued
If the CPU sets the Assert /RST bit (bit 7) in the Initiator Command Register, the /RST signal goes active on the SCSI Bus and an internal reset is performed. Again, all internal logic and registers are cleared except for the IRQ interrupt latch and the Assert /RST bit (bit 7) in the Initiator Command Register. The /RST signal will continue to be active until the Assert /RST bit is reset or until a hardware reset occurs.

Data Transfers
Data is transferred between SCSI Bus devices in one of four modes (Reference Figures 26-41):

1. Programmed I/O
2. Normal DMA
3. Block Mode DMA
4. Pseudo DMA

The following sections describe these modes in detail (**Note:** For all data transfer operations, /DACK and /CS should never be active simultaneously).

Programmed I/O Transfers
Programmed I/O is the most primitive form of data transfer The /REQ and /ACK handshake signals are individually monitored and asserted by reading and writing the appropriate register bits. This type of transfer is normally used when transferring small blocks of data such as command blocks or message and status bytes. An Initiator send operation would begin by setting the C//D, I//O, and /MSG bits in the Target Command Register to the correct state so that a phase match exists. In addition to the phase match condition, it is necessary for the Assert Data Bus bit (Initiator Command Register, bit 0) to be True and the received I/O signal to be False for the Z5380 to send data For each transfer, the data is loaded into the Output Data Register. The CPU then waits for the /REQ bit (Current SCSI Bus Status Register, bit 5) to become active. Once /REQ goes active, the Phase Match bit (Bus and Status Register, bit 3) is checked and the Assert /ACK bit (Initiator Command Register, bit 4) is set. The /REQ bit is sampled until it becomes False and the CPU resets the Assert /ACK bit to complete the transfer.

Normal DMA Mode
DMA transfers are normally used for large block transfers. The SCSI chip outputs a DMA request (DRQ) whenever it is ready for a byte transfer. External DMA logic uses this

DRQ signal to generate /DACK and an /IOR or an /IOW pulse to the Z5380. DRQ goes inactive when /DACK is asserted and /DACK goes inactive some time after the minimum read or write pulse width. This process is repeated for every byte. For this mode, /DACK should not be allowed to cycle unless a transfer is taking place.

Block Mode DMA

Some popular DMA Controllers, such as the 9517A, provide a Block Mode DMA transfer. This type of transfer allows the DMA controller to transfer blocks of data without relinquishing the use of the Data Bus to the CPU after each byte is transferred; thus, faster transfer rates are achieved by eliminating the repetitive access and release of the CPU Bus. If the Block Mode DMA bit (Mode Register, bit 7) is active, the Z5380 begins the transfer by asserting DRQ. The DMA controller then asserts /DACK for the remainder of the block transfer. DRQ goes inactive for the duration of the transfer. The Ready output is used to control the transfer rate. Non-Block Mode DMA transfers end when /DACK goes False, whereas Block Mode DMA transfers end when /IOR or /IOW becomes inactive. Since this is the case, DMA transfers may be started sooner in a Block Mode transfer. To obtain optimum performance in Block Mode operation, the DMA logic optionally uses the normal DMA mode interlocking handshake. Ready is still available to throttle the DMA transfer, but DRQ is 30 to 40 ns faster than Ready and is used to start the cycle sooner. The methods described under "Halting a DMA Operation" apply for all DMA operations.

Pseudo DMA Mode

To avoid the tedium of monitoring and asserting the request/acknowledgement handshake signals for programmed I/O transfers, the system can be designed to implement a pseudo DMA mode. This mode is implemented by programming the Z5380 to operate in the DMA mode, but using the CPU to emulate the DMA handshake. DRQ may be detected by polling the DMA Request bit (bit 6) in the Bus and Status Register, by sampling the signal through an external port, or by using it to generate a CPU interrupt. Once DRQ is detected, the CPU can perform a read or write data transfer. This CPU read/write is externally decoded to generate the appropriate /DACK and /IOR or /IOW signals.

Often, external decoding logic is necessary to generate the Z5380 /CS signal. This same logic may be used to generate /DACK at no extra cost and provide an increased performance in programmed I/O transfers.

Halting a DMA Operation

The /EOP signal is not the only way to halt a DMA transfer. A bus phase mismatch or a reset of the DMA Mode bit (Mode Register, bit 1) can also terminate a DMA cycle for the current bus phase.

Using the /EOP Signal

If /EOP is used, it should be asserted for at least 100ns while /DACK and /IOR or /IOW are simultaneously active. Note, however, that if /IOR or /IOW is not active, an interrupt is generated, but the DMA activity continues. The /EOP signal does not reset the DMA Mode bit. Since the /EOP signal can occur during the last byte sent to the Output Data Register, the /REQ and /ACK signals are monitored to ensure that the last byte has transferred.

Bus Phase Mismatch Interrupt

A bus phase mismatch interrupt is used to halt the transfer if operating as an Initiator. Using this method frees the host from maintaining a data length counter and frees the DMA logic from providing the /EOP signal. If performing an Initiator send operation, the Z5380 requires /DACK to cycle before /ACK goes inactive. Since phase changes cannot occur if /ACK is active, either /DACK must be cycled after the last byte is sent or the DMA Mode bit must be reset in order to receive the phase mismatch interrupt.

Resetting the DMA Mode Bit

A DMA operation may be halted at any time simply by resetting the DMA Mode bit. It is recommended that the DMA Mode bit be reset after receiving an /EOP or bus phase-mismatch interrupt. The DMA Mode bit must then be set before writing any of the start DMA registers for subsequent bus phases.

If resetting the DMA Mode bit is used instead of /EOP for Target role operation, then care must be taken to reset this bit at the proper time. If receiving data as a Target device, the DMA Mode bit must be reset once the last DRQ is received and before /DACK is asserted to prevent an additional /REQ from occurring. Resetting this bit causes DRQ to go inactive However, the last byte received remains in the Input Data Register and may be obtained either by performing a normal CPU read or by cycling /DACK and /IOR In most cases, /EOP is easier to use when operating as a Target device.

READ REGISTERS

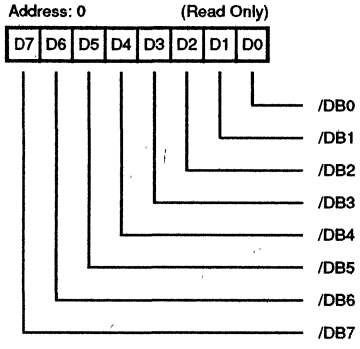


Figure 26. Current SCSI Data Register

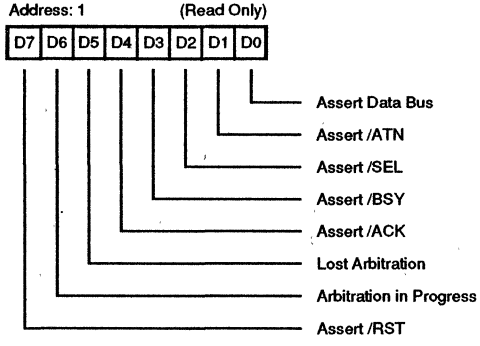


Figure 27. Initiator Command Register

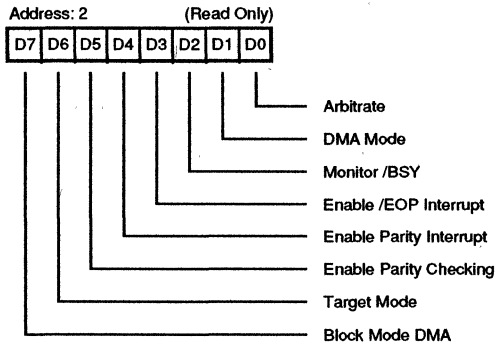


Figure 28. Mode Register

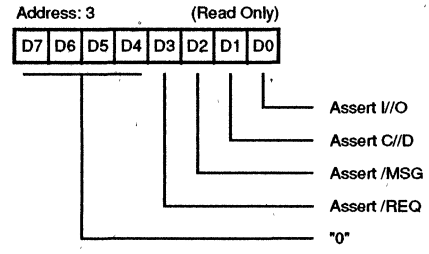


Figure 29. Target Command Register

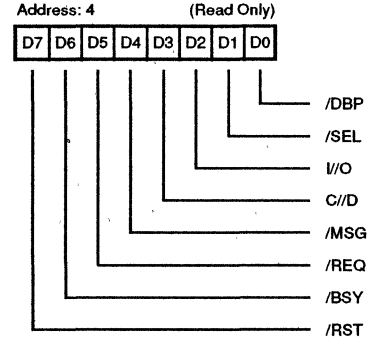


Figure 30. Current SCSI Bus Status Register

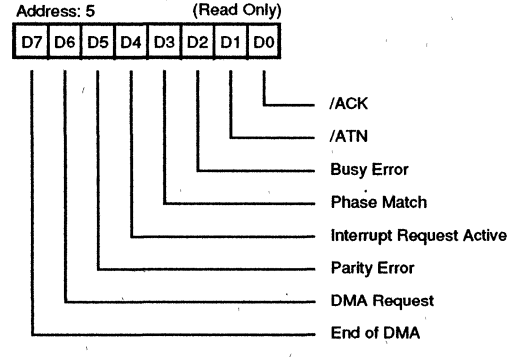


Figure 31. Bus and Status Register

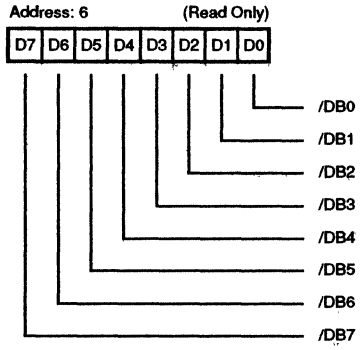


Figure 32. Input Data Register

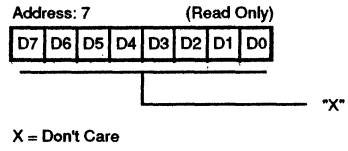


Figure 33. Reset Parity/Interrupt

WRITE REGISTERS

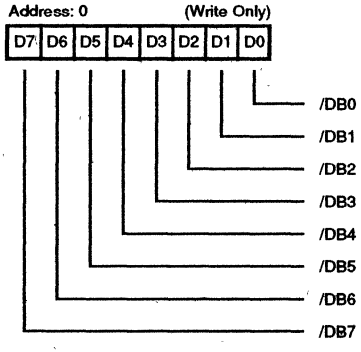


Figure 34. Output Data Register

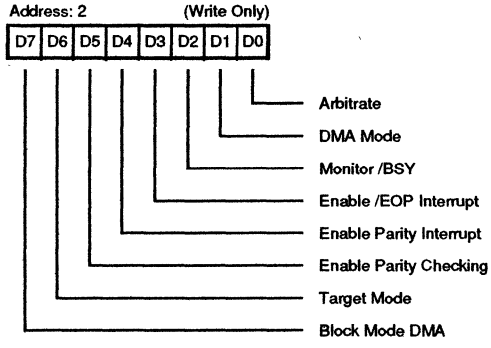


Figure 36. Mode Register

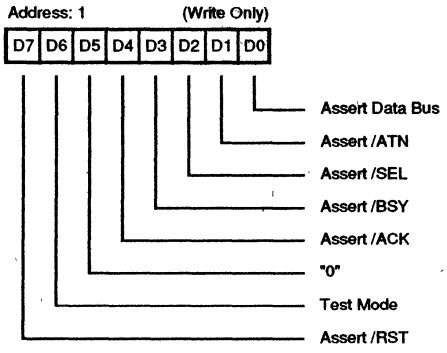


Figure 35. Initiator Command Register

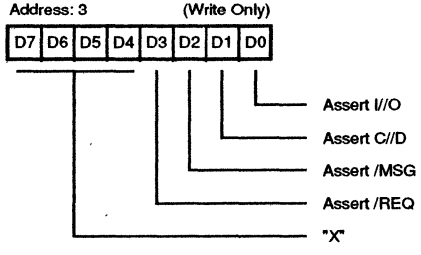


Figure 37. Target Command Register

WRITE REGISTERS (Continued)

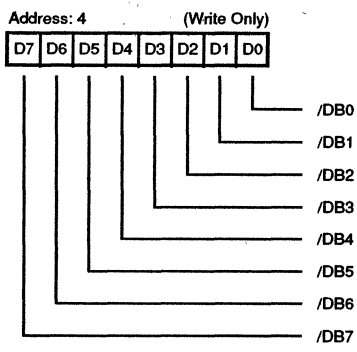


Figure 38. Select Enable Register

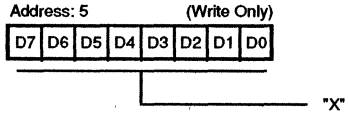


Figure 39. Start DMA Send

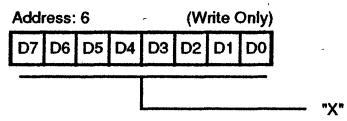


Figure 40. Start DMA Target Receive

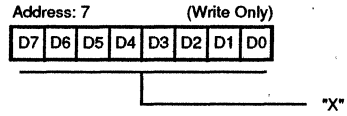


Figure 41. Start DMA Initiator Receive

ABSOLUTE MAXIMUM RATINGS

Voltages on all pins
with respect to GND ..... -0.3V to +7.0V
Operating Ambient Temperature ..... †
Storage Temperature ..... -65°C to +150°C

Note:
† 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 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. Standard conditions are as follows (Figures 42 and 43):

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

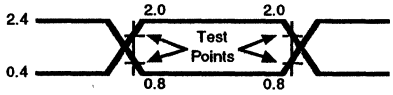


Figure 42. Switching Test Circuit

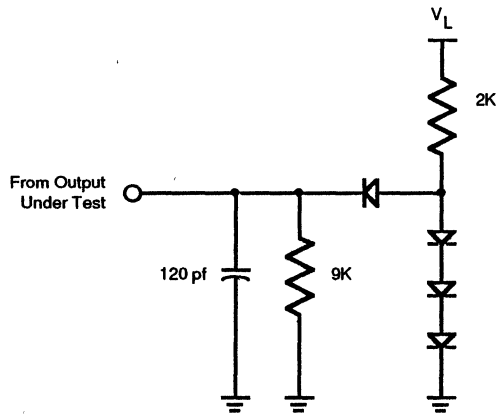


Figure 43. Standard Test Load

DC CHARACTERISTICS

Z5380

Symbol	Parameter	Conditions	Min	Max	Units
V_{DD}	Supply Voltage		4.75	5.25	V
V_{IH}	High-Level Input Voltage		2.0	5.25	V
V_{IL}	Low-Level Input Voltage		-0.3	0.8	V
$I_{IH}1$	High-Level Input Current	V_{IH} = 5 25V			
	SCSI Bus Pins	V_{IL} = 0V		50	µA
$I_{IH}2$	High-Level Input Current	V_{IH} = 5.25V			
	All Other Pins	V_{IL} = 0V		10	µA
$I_{IL}1$	Low-Level Input Current	V_{IH} = 5.25V			
	SCSI Bus Pins	V_{IL} = 0V	-50		µA
$I_{IL}2$	Low-Level Input Current	V_{IH} = 5.25V			
	All Other Pins	V_{IL} = 0V	-10 µA		
V_{OH}	High-Level Output Voltage	I_{OH} = -3mA V_{DD} = 4.75V	2.4		V
$V_{OL}1$	Low-Level Output Voltage SCSI Bus Pins	I_{OL} = 48mA V_{DD} = 4.75V	0.5		V
$V_{OL}2$	Low-Level Output Voltage All Other Pins	I_{OL} = 7mA V_{DD} = 4.75V	0.5		V
I_{DD}	Supply Current	15 mA			
T_A	Operating Free-Air		0	70	C

AC CHARACTERISTICS
CPU Write Cycle Timing Diagram

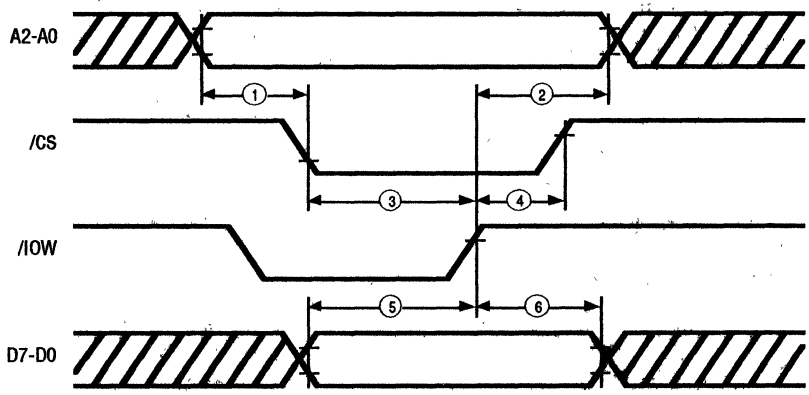


Figure 44. CPU Write Cycle

AC CHARACTERISTICS
CPU Write Cycle Timing Table

No	Description	Min	Max	Units
1	Address Setup to Write Enable[1]	20		ns
2	Address Hold from End Write Enable[1]	20		ns
3	Write Enable Width[1]	70		ns
4	Chip Select Hold from End of /IOW	0		ns
5	Data Setup to end of Write Enable[1]	50		ns
6	Data Hold Time form End of /IOW	30		ns

Note:
[1] Write Enable is the occurrence of /IOW and /CS

AC CHARACTERISTICS
CPU Read Cycle Timing Diagram

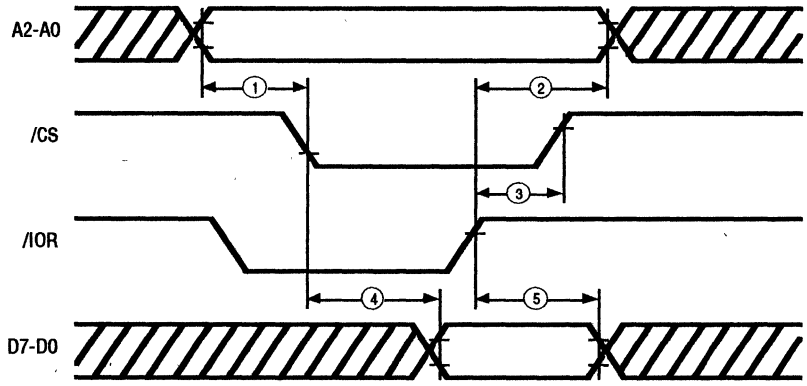


Figure 45. CPU Read Cycle

AC CHARACTERISTICS
CPU Read Cycle Timing Table

No	Description	Min	Max	Units
1	Address Setup to Read Enable[1]	20		ns
2	Address Hold from End Read Enable[1]	20		ns
3	Chip Select Hold from End of /IOR	0		ns
4	Data Access Time from Read Enable[1]	130		ns
5	Data Hold Time from End of Read Enable[1]	20		ns

Note:
[1] Read Enable is the occurrence of /IOR and /CS.

AC CHARACTERISTICS

DMA Write (Non-Block Mode) Target Send Cycle Timing Diagram

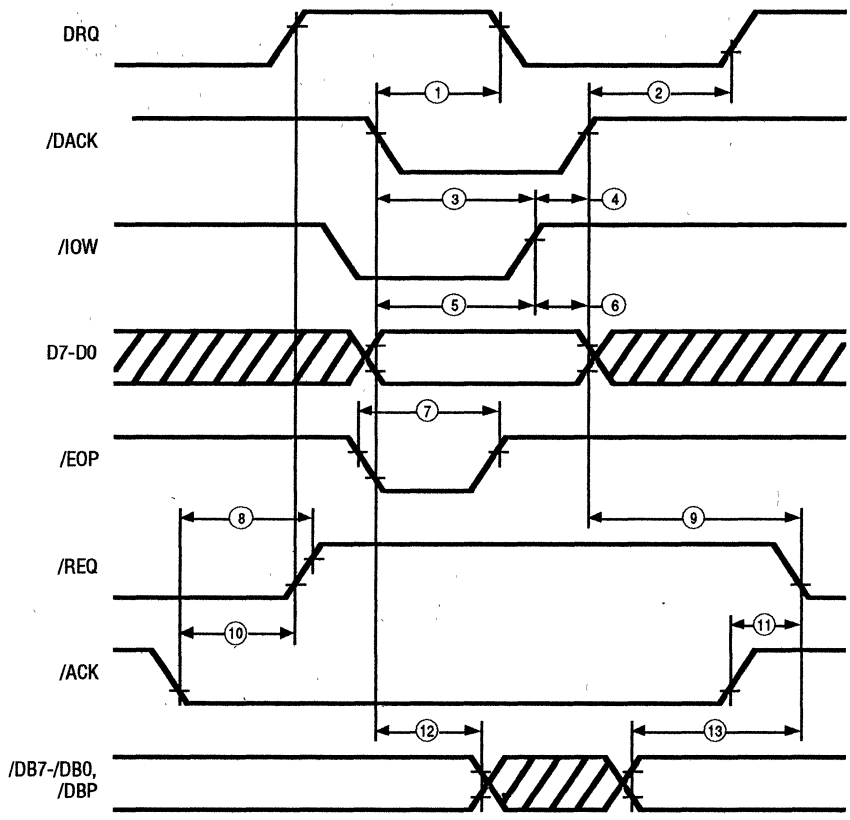


Figure 46. DMA Write (Non-Block Mode) Target Send Cycle

AC CHARACTERISTICS

DMA Write (Non-Block Mode) Target Send Cycle Table

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low	130		ns
2	/DACK High to DRQ High	30		ns
3	Write Enable Width[1]	100		ns
4	/DACK Hold from /IOW High	0		ns
5	Data Setup to End of Write Enable[1]	50		ns
6	Data Hold Time from End of /IOW	40		ns
7	Width of /EOP Pulse[2]	100		ns
8	/ACK Low to /REQ High	25	125	ns
9	/REQ from End of /DACK (/ACK High)	30	150	ns
10	/ACK Low to DRQ High (Target)	15	110	ns
11	/ACK High to /REQ Low (/DACK High)	20	150	ns
12	Data Hold from Write Enable	15		ns
13	Data Setup to /REQ Low (Target)	60		ns

Notes:

[1] Write Enable is the occurrence of /IOW and /DACK.

[2] /EOP, /IOW, and /DACK must be concurrently Low for at least T7 for proper recognition of the /EOP pulse.

AC CHARACTERISTICS

DMA Write (Non-Block Mode) Initiator Send Cycle Timing Diagram

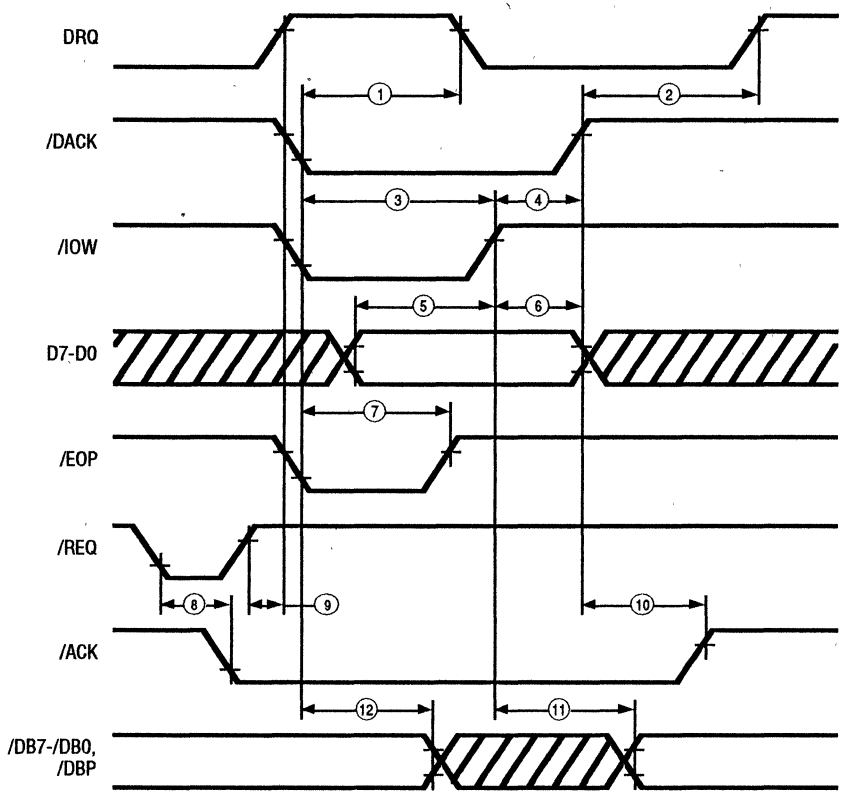


Figure 47. DMA Write (Non-Block Mode) Initiator Send Cycle

AC CHARACTERISTICS

DMA Write (Non-Block Mode) Initiator Send Cycle Table

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low	130		ns
2	/DACK High to DRQ High	30		ns
3	Write Enable Width[1]	100		ns
4	/DACK Hold from End of /IOW	0		ns
5	Data Setup to End of Write Enable[1]	50		ns
6	Data Hold Time from End of /IOW	40		ns
7	Width of /EOP Pulse[2]	100		ns
8	/REQ Low to /ACK Low	20	160	ns
9	/REQ High to DRQ High	20	110	ns
10	/DACK High to /ACK High	25	150	ns
11	/IOW High to Valid SCSI Data	100		ns
12	Data Hold from Write Enable[1]	15		ns

Notes:

[1] Write Enable is the occurrence of /IOW and /DACK.

[2] /EOP, /IOW, and /DACK must be concurrently Low for at least T7 for proper recognition of the /EOP pulse.

AC CHARACTERISTICS

DMA Read (Non-Block Mode) Target Receive Cycle Timing Diagram

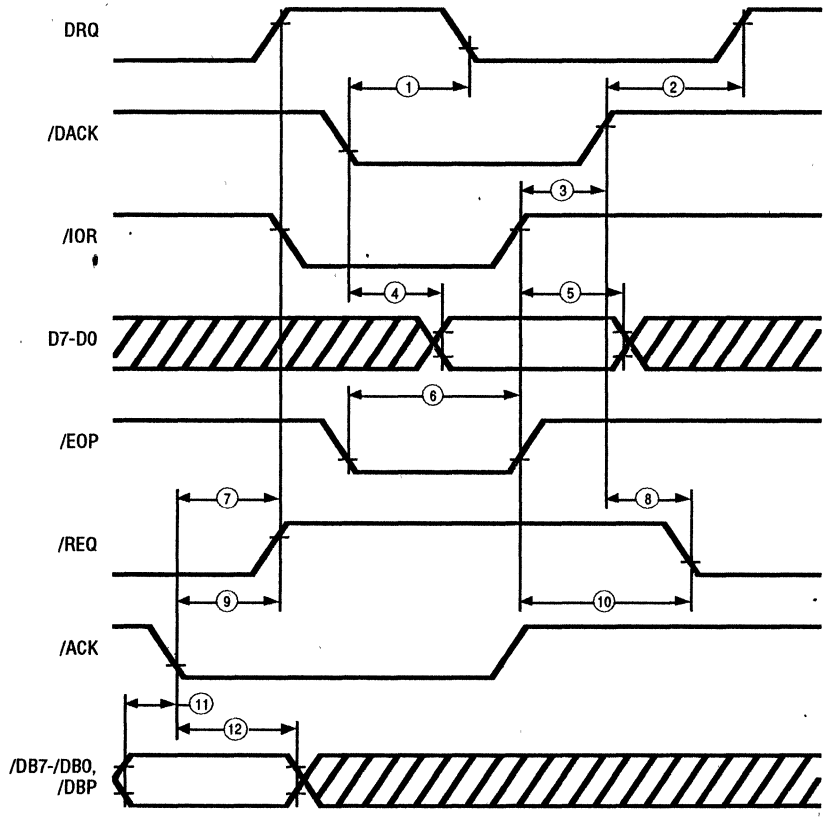


Figure 48. DMA Read (Non-Block Mode) Target Receive Cycle

AC CHARACTERISTICS

DMA Read (Non-Block Mode) Target Receive Cycle Table

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low	130		ns
2	/DACK High to DRQ High	30		ns
3	/DACK Hold Time from End of /IOR	0		ns
4	Data Access Time from Read Enable[1]	115		ns
5	Data Hold Time from End of /IOR	20		ns
6	Width of /EOP Pulse[2]	100		ns
7	/ACK Low to DRQ High	15	110	ns
8	/DACK High to /REQ Low (/ACK High)	30	150	ns
9	/ACK Low to /REQ High	25	125	ns
10	/ACK High to /REQ Low (/DACK High)	20	150	ns
11	Data Setup Time to /ACK	20		ns
12	Data Hold Time from /ACK	50		ns

Notes:

[1] Read Enable is the occurrence of /IOR and /DACK

[2] /EOP, /IOR, and /DACK must be concurrently Low for at least T6 for proper recognition of the /EOP pulse.

AC CHARACTERISTICS

DMA Read (Non-Block Mode) Initiator Receive Cycle Timing Diagram

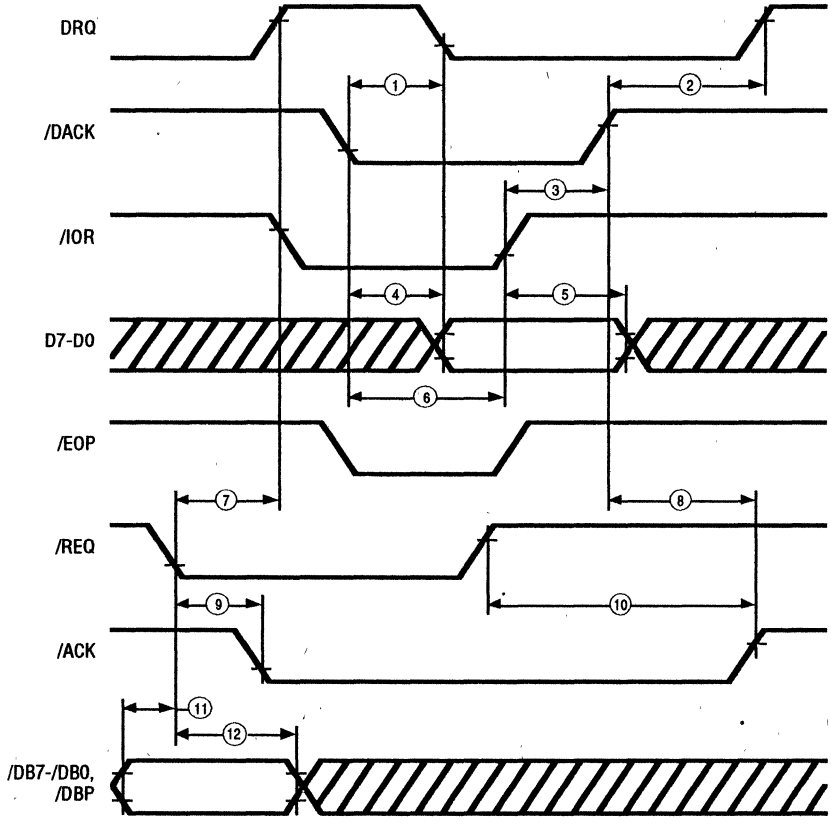


Figure 49. DMA Read (Non-Block Mode) Initiator Receive Cycle

AC CHARACTERISTICS

DMA Read (Non-Block Mode) Initiator Receive Cycle Table

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low	130		ns
2	/DACK High to DRQ High	30		ns
3	/DACK Hold Time from End of /IOR	0		ns
4	Data Access Time from Read Enable[1]	115		ns
5	Data Hold Time from End of /IOR	20		ns
6	Width of /EOP Pulse[2]	100		ns
7	/REQ Low to DRQ High	20		ns
8	/DACK High to /ACK High (/REQ High)	25	160	ns
9	/REQ Low to /ACK Low	20	160	ns
10	/REQ High to /ACK High (/DACK High)	15	140	ns
11	Data Setup Time to /REQ	20		ns
12	Data Hold Time from /REQ	50		ns

Notes:

[1] Read Enable is the occurrence of /IOR and /DACK.

[2] /EOP, /IOR, and /DACK must be concurrently Low for at least T6 for proper recognition of the /EOP pulse.

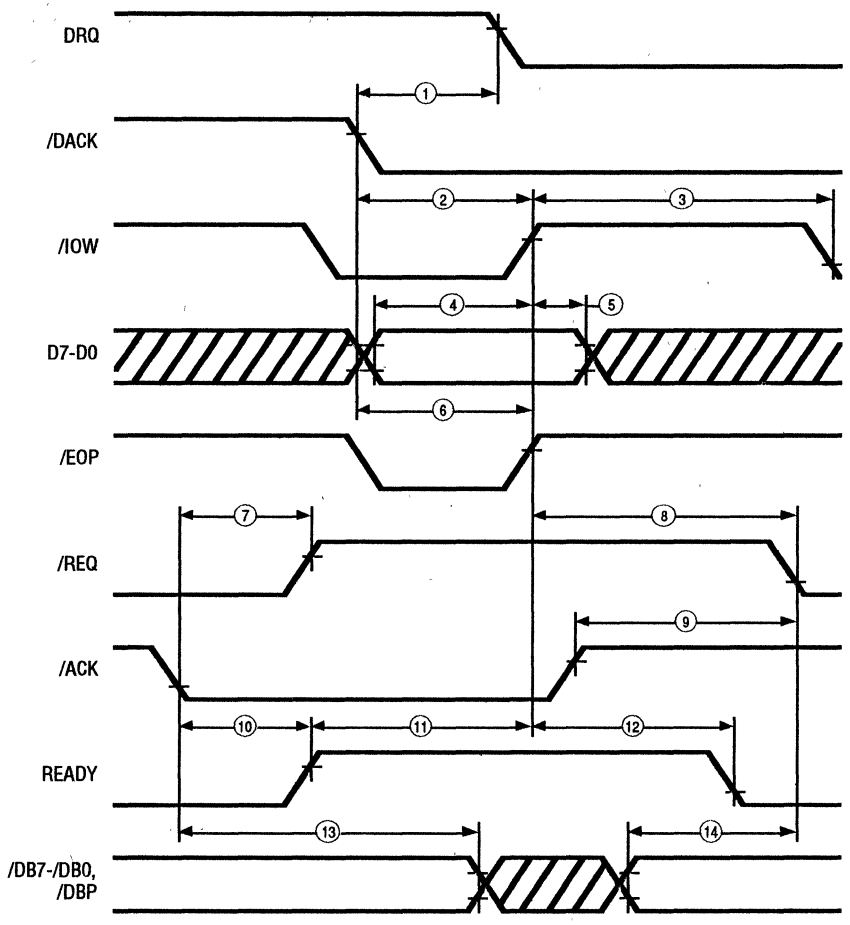


Figure 50. DMA Write (Block Mode) Target Send Cycle

AC CHARACTERISTICS

DMA Write (Block Mode) Target Send Cycle Table

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low	130		ns
2	Write Enable Width[1]	100		ns
3	Write Recovery Time	120		ns
4	Data Setup to End of Write Enable[1]	50		ns
5	Data Hold Time from End of /IOW	40		ns
6	Width of /EOP Pulse[2]	100		ns
7	/ACK Low to /REQ High	25	125	ns
8	/REQ from End of /IOW (/ACK High)	40	180	ns
9	/REQ from End of /ACK (/IOW High)	20	170	ns
10	/ACK Low to READY High	20	140	ns
11	READY High to /IOW High	70		ns
12	/IOW High to READY Low	20	140	ns
13	Data Hold from /ACK Low	40		ns
14	Data Setup to /REQ Low	60		ns

Notes:

[1] Write Enable is the occurrence of /IOW and /DACK.

[2] /EOP, /IOW, and /DACK must be concurrently Low for at least T6 for proper recognition of the /EOP pulse.

AC CHARACTERISTICS

DMA Read (Block Mode) Target Receive Cycle Timing Diagram

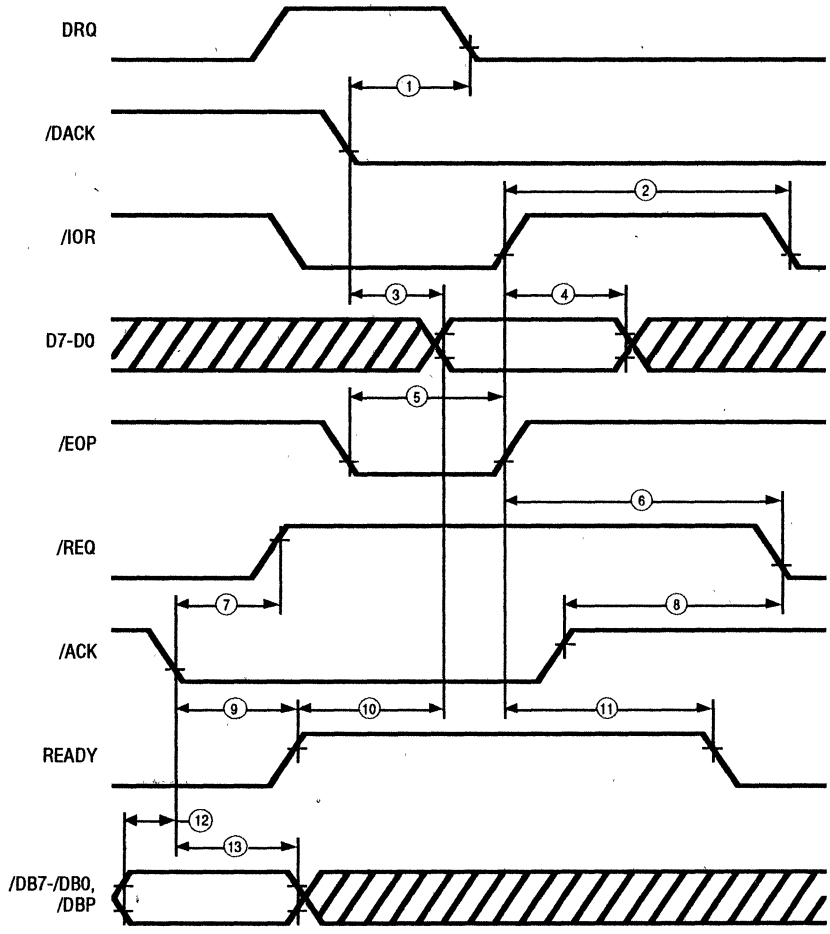


Figure 51. DMA Read (Block Mode) Target Receive Cycle

AC CHARACTERISTICS

DMA Read (Block Mode) Target Receive Cycle Table

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low	130		ns
2	/IOR Recovery Time	120		ns
3	Data Access Time from Read Enable[1]	110		ns
4	Data Hold Time from End of /IOR	20		ns
5	Width of /EOP Pulse[2]	100		ns
6	/IOR High to /REQ Low	30	190	ns
7	/ACK Low to /REQ High	25	125	ns
8	/ACK High to /REQ Low (/IOR High)	20	170	ns
9	/ACK Low to READY High	20	140	ns
10	READY High to Valid Data	50		ns
11	/IOR High to READY Low	20	140	ns
12	Data Setup Time to /ACK	20		ns
13	Data Hold Time from /ACK	50		ns

Notes:

[1] Read Enable is the occurrence of /IOR and /DACK.

[2] /EOP, /IOR, and /DACK must be concurrently Low for at least T5 for proper recognition of the /EOP pulse.

AC CHARACTERISTICS
Arbitration

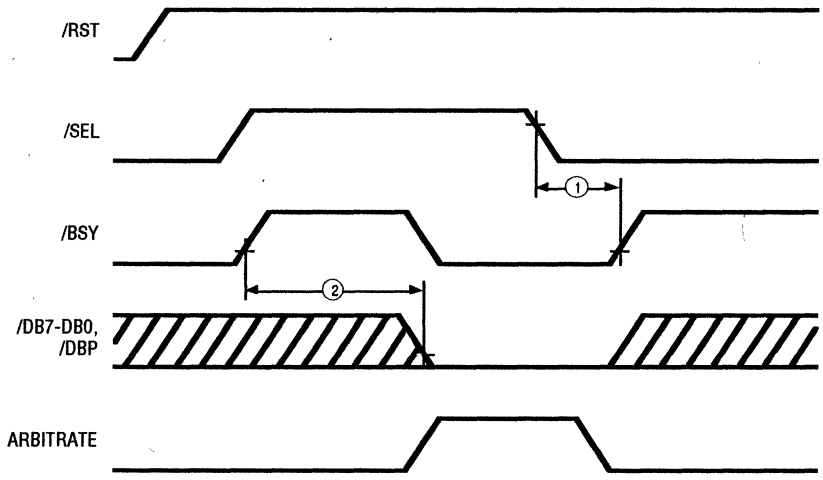


Figure 52. Arbitration

No	Description	Min	Max	Units
1	Bus Clear from /SEL Low		600	ns
2	Arbitrate Start from /BSY High	1200	2200	ns

AC CHARACTERISTICS
Reset

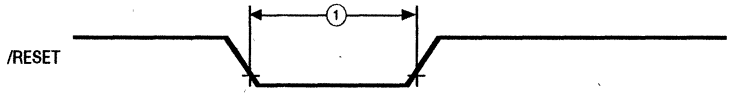


Figure 53. Reset

No	Description	Min	Max	Units
1	Minimum Width of /RESET	200		ns

Z5380 NOTES

1. Edge-triggered /RST Interrupt - If the SCSI Bus is not terminated, the /RST interrupt is continually generated.

2. True End of DMA Interrupt - The Z5380 generates an interrupt when it receives the last byte from the DMA, not when the last byte is transferred to the SCSI Bus.

3. Return to Ready after /EOP Interrupt - When operating in Block Mode DMA, the Z5380 does not return the Ready signal to a Ready condition. This locks up the bus and prevents the CPU from executing.

4. SCSI handshake after /EOP occurs - If an EOP occurs when receiving data, a subsequent request will cause /ACK to be asserted even though no DRQ is issued

5. Reselection Interrupt - During reselection, if the Target Command Register does not reflect the current bus phase (most likely Data Out), the reselection interrupt may get reset.

6. Phase Mismatch Interrupt - A phase mismatch interrupt is not guaranteed after a reselection for the following reasons:

DMA Mode bit must be set in order to receive a phase mismatch interrupt.

DMA Mode bit can not be set unless /BSY is active.

/BSY can not be asserted until after the reselection has occurred.

Once /BSY is asserted, the Target may assert /REQ in less than 500ns.

The phase mismatch interrupt is generated on the active edge of /REQ. If the DMA Mode bit is not set before the /REQ goes active, the phase mismatch interrupt will not occur.

PRELIMINARY PRODUCT SPECIFICATION

Z85230

ESCC™ ENHANCED SERIAL COMMUNICATION CONTROLLER

FEATURES

- Deeper Data FIFOs
  - 4-byte transmit FIFO
  - 8-byte receive FIFO
- Programmable FIFO interrupt levels provide flexible interrupt response
- Pin and Function compatible to CMOS and NMOS Z85C30 SCC
- Many improvements to support SDLC/HDLC transfers:
  - Deactivation of /RTS pin after closing flag
  - Automatic transmission of the opening flag
  - Automatic reset of Tx Underrun/EOM latch
  - Complete CRC reception
  - TxD pin automatically forced high with NRZI encoding when using mark idle.
  - Receive FIFO automatically unlocked for special receive interrupts when using the SDLC status FIFO.
  - Back-to-back frame transmission simplified
- Easier interface to popular CPUs
- Fast speeds:
  - 10.0 MHz for data rates up to 2.5 Mbit/sec.
  - 16 384 MHz for data rates up to 4.096 Mbit/sec.
  - 20.0 MHz for data rates up to 5 0 Mbit/sec.
- Faster interrupt response
- Improved SDLC frame status FIFO
- Low Power CMOS
- New programmable features added with Write Register 7'
- Write registers: WR3, WR4, WR5, and WR10 are now readable
- Read Register 0 latched during access
- Software Interrupt Acknowledge Mode
- DPLL counter output available as jitter-free clock source
- /DTR//REQ pin deactivation time reduced
- Two independent full-duplex channels, each with a crystal oscillator, baud rate generator, and Digital Phase Locked Loop.
- Multi-protocol operation under program control
- Asynchronous mode with five to eight bits, and one, one and one-half, or two stop bits per character; programmable clock factor; break detection and generation; parity, overrun, and framing error detection.
- Synchronous mode with internal or external character synchronization on one or two synchronous characters and CRC generation and checking with programmable CRC preset values.
- Multiplexed Z-Bus version, Z80230, planned in Q2 1991

GENERAL DESCRIPTION

The Zilog Enhanced Serial Communications Controller, Z85230 ESCC, is a pin and software compatible CMOS member of the SCC family introduced by Zilog in 1981. The ESCC is a dual-channel, full-duplex data communications controller capable of supporting a wide range of popular protocols. The ESCC is built from Zilog's industry standard SCC core and is compatible with designs using Zilog's SCC to receive and transmit data. It has many improvements that significantly reduce CPU overhead. The addition of a 4-byte transmit FIFO and an 8-byte receive FIFO significantly reduces the overhead required to provide data to, and get data from, the transmitters and receivers.

GENERAL DESCRIPTION (Continued)

The ESCC also has many features that improve packet handling in SDLC mode. The ESCC will automatically: transmit a flag before the data, reset the Tx Underrun/EOM latch, force the TxD pin high at the appropriate time when using NRZI encoding, deassert the /RTS pin after the closing flag, and better handle ABORTed frames when using the 10x19 status FIFO. The combination of these features along with the deeper data FIFOs significantly simplifies SDLC driver software.

The CPU hardware interface has been simplified by relieving the databus setup time requirement and supporting the software generation of the interrupt acknowledge signal (INTACK). These changes allow an interface with less external logic to many microprocessor families while maintaining compatibility with existing designs. I/O handling of the ESCC is improved over the SCC with faster response of the /INT and /DTR//REQ pins.

The many enhancements added to the ESCC permits a system design that increases overall system performance with better data handling and less interface logic.

Note: 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); /N//S (NORMAL and SYSTEM are both active Low).

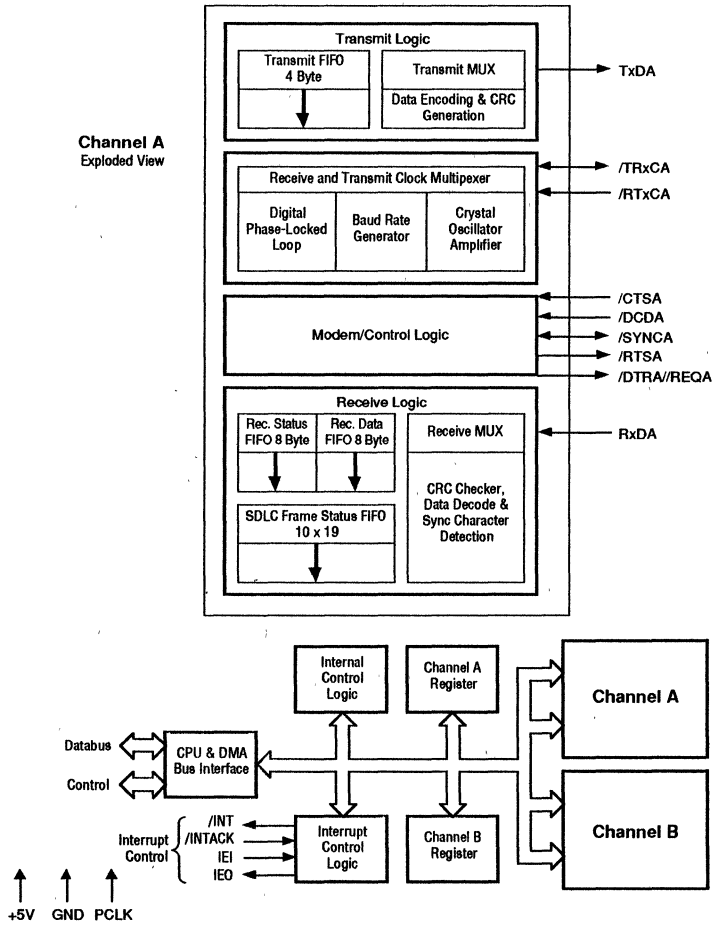


Figure 1. ESCC Block Diagram

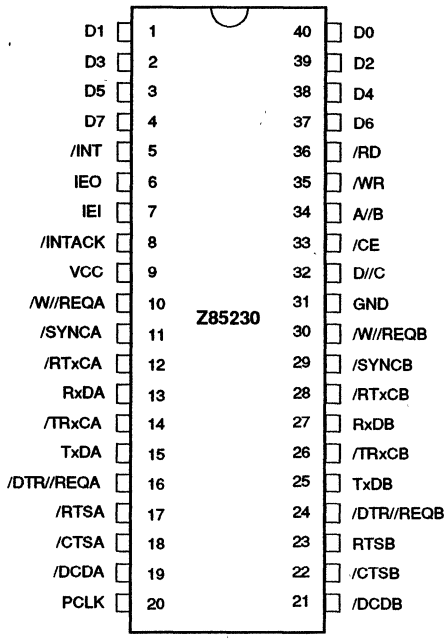


Figure 2. Z85230 DIP Pin Assignments

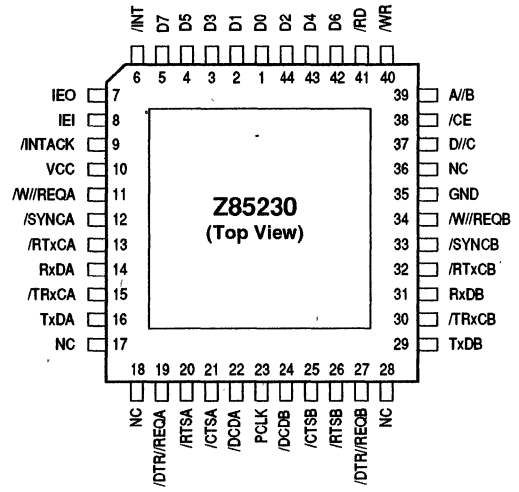
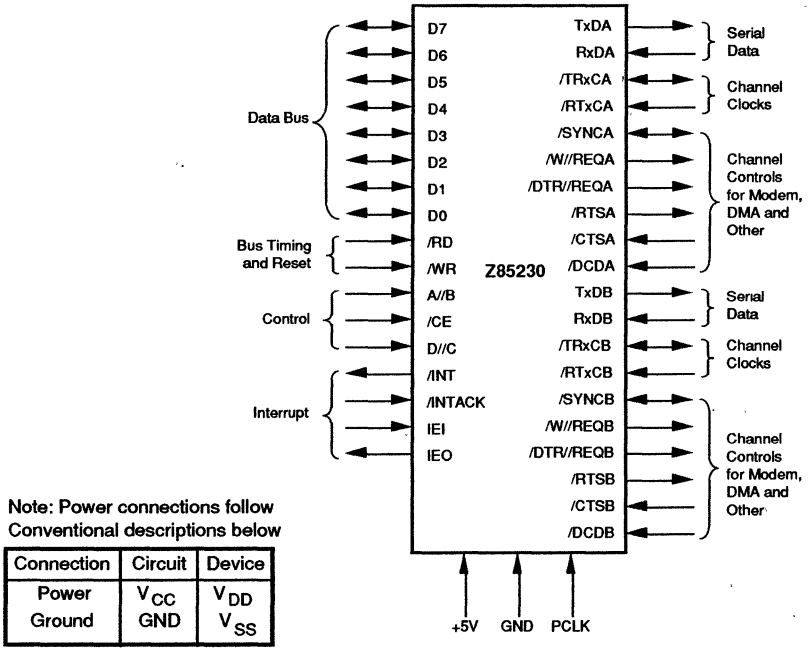


Figure 3. Z85230 PLCC Pin Assignments



Note: Power connections follow Conventional descriptions below

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

Figure 4. Z85230 Pin Functions

PIN DESCRIPTIONS

The following section describes the Z85230 pin functions. Figures 2 and 3 detail the pin assignments for the 40-pin DIP and 44-pin PLCC packages. The Z85230 ESCC is socket compatible with the Zilog Z8530 and Z85C30 as the pin electrical characteristics and pin assignments are the same. Any unused input pins should be pulled up to the +5V supply.

/CTSA, /CTSB. *Clear To Send* (inputs, active Low). These pins function as transmitter enables if they are programmed for Auto Enables (WR3, D5=1). A Low on the inputs enables the respective transmitters. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt trigger buffered to accommodate slow rise-time inputs. The ESCC detects pulses on these inputs and can interrupt the CPU on both logic level transitions.

/DCDA, /DCDB. *Data Carrier Detect* (inputs, active Low). These pins function as receiver enables if they are programmed for Auto Enables (WR3, D5=1); otherwise they are used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accommodate slow rise time signals. The ESCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.

/RTSA, /RTSB. *Request To Send* (outputs, active Low). The /RTS pins can be used as general purpose outputs or with the Auto Enables feature. When used with Auto Enables ON (WR3, D5=1) in asynchronous mode, the /RTS pin goes high after the transmitter is empty. When Auto Enable is OFF, the /RTS pins can be used as general purpose outputs and, they strictly follow the inverse state of the RTS bit (WR5 bit D1).

In SDLC mode, the /RTS pins can be programmed to be deasserted when the closing flag of the message clears the TxD pin if WR7' D2 is set.

/SYNCA, /SYNCB. *Synchronization* (inputs or outputs, active Low). These pins can act either as inputs, outputs, or part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to CTS and DCD. In this mode, transitions on these lines affect the state of the Synchronous/Hunt status bits in Read Register 0 but have no other function.

In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, /SYNC must be driven Low for two receive clock cycles after the last bit in the synchronous character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of /SYNC.

In the Internal Synchronization mode (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which synchronous condition is latched. These outputs are active each time a synchronization pattern is recognized (regardless of character boundaries). In SDLC mode, the pins act as outputs and are valid on receipt of a flag. The /SYNC pins switch from input to output when monosync, bisync, or SDLC is programmed in WR4 and sync modes are enabled.

/DTR//REQA, /DTR//REQB. *Data Terminal Ready/Request* (outputs, active Low). These pins are programmed (WR14, D2) to serve either as general purpose outputs or as DMA Request lines. When programmed for the DTR function (WR14, D2=0), these outputs follow the state programmed into the DTR bit of Write Register 5 (WR5, D7). When programmed for Request mode (WR14, D2=1), these pins serve as DMA Requests for the transmitter.

When used as DMA request lines, the timing for the deactivation Request can be programmed in the added register Write Register 7' (WR7') bit D4. If this bit is set, the /DTR//Request pin will be deactivated with the same timing as the /W//REQ pin. If WR7' D4 is reset, the deactivation timing of /DTR//Req pin will be the same as in the Z85C30.

W//REQA, /W//REQB. *Wait/Request* (outputs, open-drain when programmed for Wait function, driven High or Low when programmed for Ready function). These dual-purpose outputs may be programmed as Request lines for a DMA controller or as Wait lines which synchronize the CPU to the ESCC data rate. The reset state is Wait.

RxDA, RxDB. *Receive Data* (inputs, active High). These input signals receive serial data at standard TTL levels.

/RTxCA, /RTxCB. *Receive/Transmit Clocks* (inputs, active Low). These pins can be programmed to several modes of operation. In each channel, RTxC may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the Digital Phase-Locked Loop. These pins can also be programmed for use with the respective SYNC pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.

TxDA, TxDB. *Transmit Data* (outputs, active High). These output signals transmit serial data at standard TTL levels.

/TRxCA, /TRxCB. *Transmit/Receive Clocks* (inputs or outputs, active Low). These pins can be programmed in several different modes of operation. TRxC may supply the receive clock or the transmit clock in the input mode or

supply the output of the Digital Phase-Locked Loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.

PCLK. *Clock* (input). This is the master ESCC clock used to synchronize internal signals. PCLK is a TTL level signal. PCLK is not required to have any phase relationship with the master system clock.

IEI. *Interrupt Enable In* (input, active High). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt driven device. A high IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.

IEO. *Interrupt Enable Out* (output, active High). IEO is High only if IEI is High and the CPU is not servicing the ESCC interrupt or the ESCC is not requesting an interrupt (Interrupt Acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.

/INT. *Interrupt* (output, open drain, active Low). This signal is activated when the ESCC requests an interrupt. Note that /INT is an open-drain output.

/INTACK. *Interrupt Acknowledge* (input, active Low). This is a strobe which indicates that an interrupt acknowledge cycle is in progress. During this cycle, the ESCC interrupt daisy chain is resolved. The device is capable of returning an interrupt vector that may be encoded with the type of interrupt pending. During the acknowledge cycle, if IEI is high the ESCC places the interrupt vector on the databus when /RD goes active. /INTACK is latched by the rising edge of PCLK.

D7-D0. *Data bus* (bidirectional, tri-state). These lines carry data and commands to and from the ESCC.

/CE. *Chip Enable* (input, active Low). This signal selects the ESCC for a read or write operation.

/RD. *Read* (input, active Low). This signal indicates a read operation and when the ESCC is selected, enables the ESCC's bus drivers. During the Interrupt Acknowledge cycle, /RD gates the interrupt vector onto the bus if the ESCC is the highest priority device requesting an interrupt.

/WR. *Write* (input, active Low). When the ESCC is selected, this signal indicates a write operation. This indicates that the CPU wants to write command bytes or data to the ESCC write registers. The coincidence of /RD and /WR is interpreted as a reset.

A//B. *Channel A/Channel B* (input). This signal selects the channel in which the read or write operation occurs. A High selects channel A and Low selects channel B.

D//C. *Data/Control Select* (input). This signal defines the type of information transferred to or from the ESCC. A High means data is being transferred and a Low indicates a command.

FUNCTIONAL DESCRIPTION

Architecture. The architecture of the ESCC is described from two points of view: as a datacommunications device which transmits and receives data in a wide variety of protocols; and as a microprocessor peripheral in which the ESCC offers valuable features such as vectored interrupts and DMA support.

The ESCC's peripheral and datacommunication are described in the following sections. A block diagram is shown in Figure 1. The details of the communications between the receive and transmit logic to the system bus is shown in Figures 5 and 6. The features and data path for each of the ESCC's A and B channels is identical. See the ESCC Technical Manual for full details on using the ESCC.

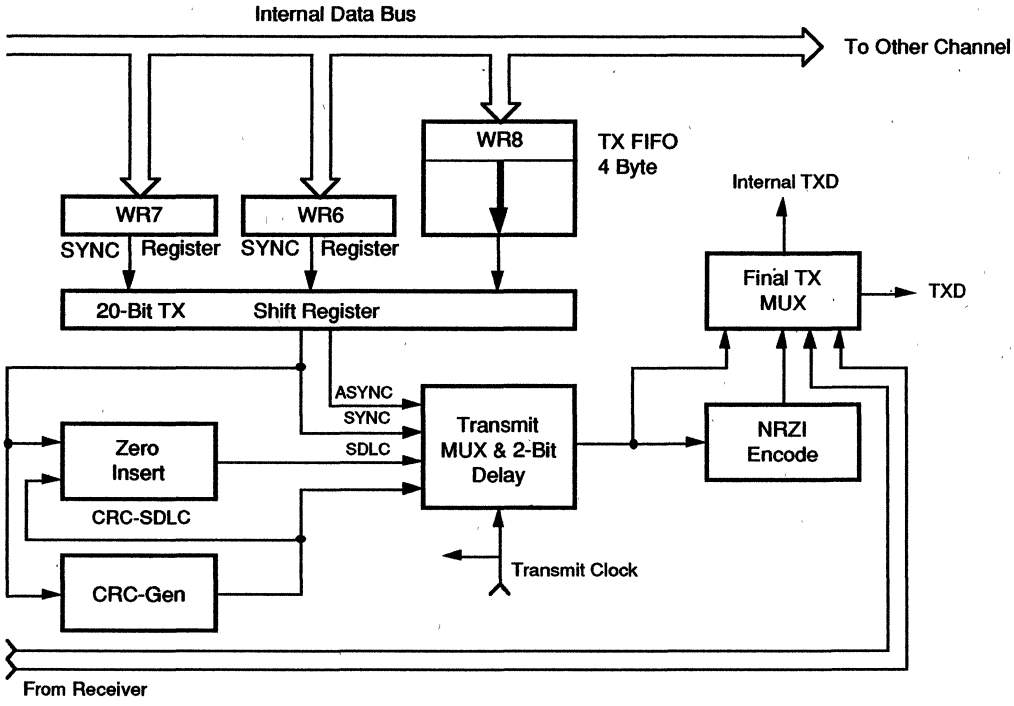


Figure 5. ESCC Transmit Data Path

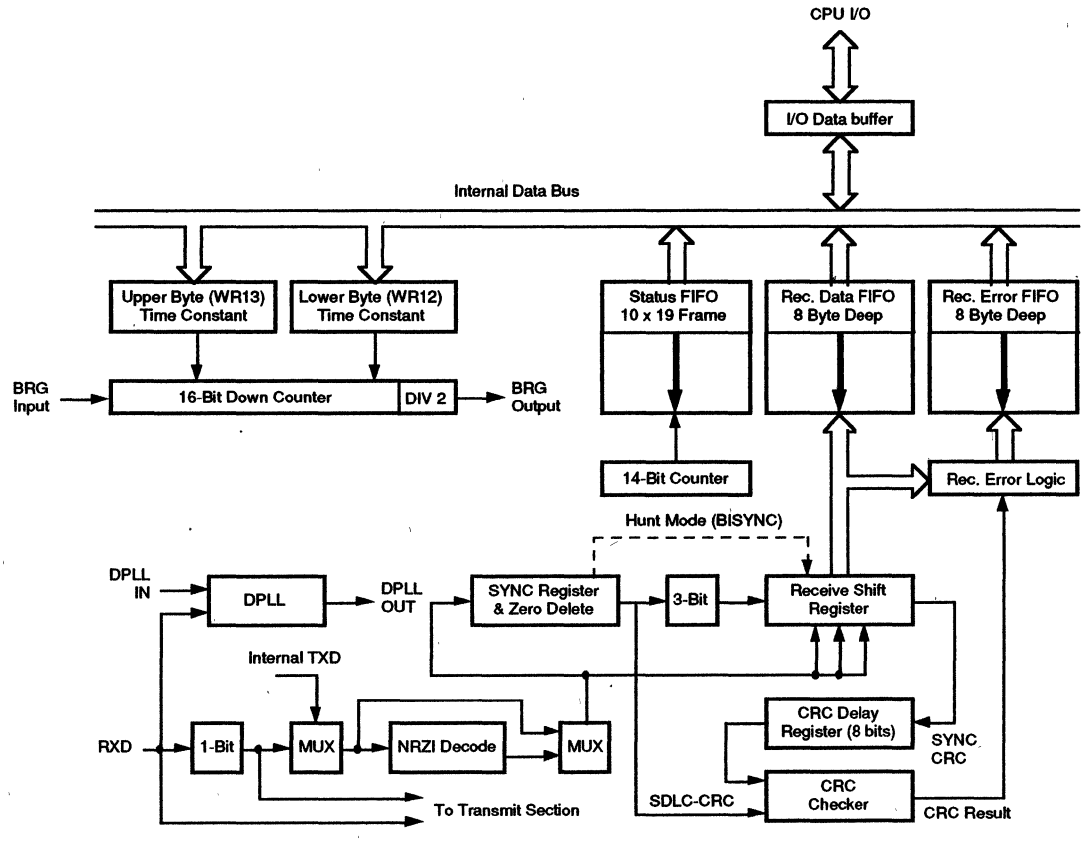


Figure 6. ESCC Receive Data Path

I/O INTERFACE CAPABILITIES

System communication to and from the ESCC is done through the ESCC's register set. There are seventeen write registers and fifteen read registers. Many of the new features on the ESCC are enabled through a new register in the ESCC: Write Register 7 Prime (WR7'). This new register can be accessed if bit D0 of WR15 is set. Table 1 lists all of the ESCC's registers and a brief description of their functions. Throughout this document, the write and read registers are referenced with the following notation: "WR" for Write Register and "RR" for Read Register. For example:

WR4A	Write Register 4 for channel A
RR3	Read Register 3 for either/both channels

Table 1. ESCC Write and Read Registers

Write Register	Functions
WR0	Command Register: Register Pointers, CRC initialization, and resets for various modes.
WR1	Interrupt conditions, Wait/DMA request control.
WR2	Interrupt Vector (accessed through either channel).
WR3	Receive and miscellaneous control parameters.
WR4	Transmit and Receive parameters and modes.
WR5	Transmit parameters and controls.
WR6	Sync character or SDLC address field.
WR7	Sync character or SDLC flag.
WR7'	SDLC enhancements enable (accessed if WR15 D0 is 1).
WR8	Transmit FIFO (4 bytes deep).
WR9	Reset commands and Master INT enable (accessed through either channel).
WR10	Miscellaneous transmit and receive controls.
WR11	Clock mode control.
WR12	Lower byte of BRG time constant.
WR13	Upper byte of BRG time constant.
WR14	Miscellaneous controls and DPLL commands.
WR15	External interrupt control.

Read Register	Functions
RR0	Transmit, Receive and external status.
RR1	Special Receive Condition status bits.
RR2A	Unmodified interrupt vector.
RR2B	Modified interrupt vector.
RR3A	Interrupt Pending bits.
RR4	WR4 status (if WR7' D6=1).
RR5	WR5 status (if WR7' D6=1).
RR6	SDLC Frame LSB Byte Count (if WR15 D2=1).
RR7	SDLC Frame 10x19 FIFO Status and MSB Byte Count (if WR15 D2=1).
RR8	Receive Data FIFO (8 Deep).
RR9	WR3 status (if WR7' D6=1).
RR10	Miscellaneous status bits.
RR11	WR10 status (if WR7' D6=1).
RR12	Lower Byte of BRG time constant.
RR13	Upper byte of BRG time constant.
RR14	WR7' status (if WR7' D6=1).

There are three choices to move data into and out of the ESCC: Polling, interrupt (vectored and non-vectored), and Block Transfer. The Block Transfer mode can be implemented under CPU or DMA control.

Polling. When polling, all interrupts are disabled. Three status registers in the ESCC are automatically updated whenever any function is performed. For example, end-of-frame in SDLC mode sets a bit in one of these status registers. The purpose of polling is for the CPU to periodically read a status register until the register contents indicate the need for data to be transferred. Only one register needs to be read; depending on its contents, the CPU either writes data, reads data, or continues. Two bits in the register indicate the need for data transfer. An alternative is a poll of the Interrupt Pending register to determine the source of an interrupt. The status for both channels resides in one register.

Interrupts. The ESCC's interrupt structure supports vectored and nested interrupts. The fill levels where the transmit and receive FIFOs interrupt the CPU are programmable. This allows the ESCC's requests for data transfers to be tuned to the system interrupt response time. Another enhancement to the ESCC is that the /INT pin will respond faster to interrupting conditions than will the SCC.

Nested interrupts are supported with the interrupt acknowledge feature (/INTACK pin) of the ESCC. This allows the CPU to recognize the occurrence of an interrupt, and re-enable higher priority interrupts. Because an INTACK cycle will release the /INT pin from the active state, a higher priority ESCC interrupt or another higher priority device can interrupt the CPU. When an ESCC responds to an Interrupt Acknowledge signal (INTACK) from the CPU, an interrupt vector may be placed on the data bus. This vector is written in WR2 and may be read in RR2. To speed interrupt response time, the ESCC can modify three bits in this vector to indicate status. If the vector is read in Channel A, status is never included; if it is read in Channel B, status is always included.

Each of the six sources of interrupts in the ESCC (Transmit, Receive, and External/Status interrupts in both channels) has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). Operation of the IE bit is straightforward. If the IE bit is set for a given interrupt source, then that source can request interrupts. The exception is when the MIE (Master Interrupt Enable) bit in WR9 is reset and no interrupts can be requested. The IE bits are write only.

The other two bits are related to the interrupt priority chain (Figure 7). As a microprocessor peripheral, the ESCC may request an interrupt only when no higher priority device is requesting one, e.g., when IEI is High. If the device in question requests an interrupt, it pulls down /INT. The CPU then responds with /INTACK, and the interrupting device places the vector on the data bus.

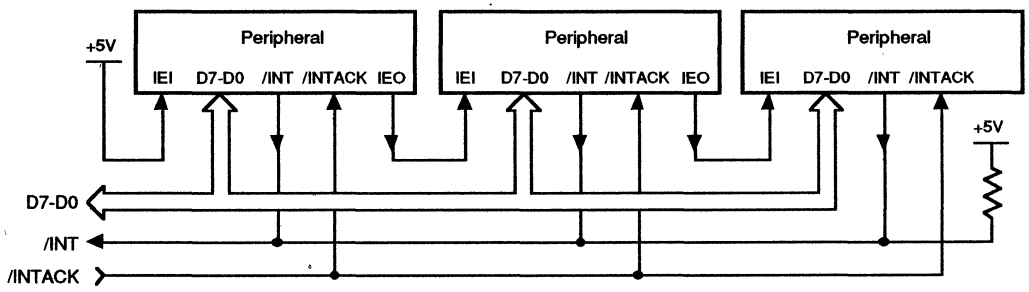


Figure 7. ESCC Interrupt Priority Schedule

I/O INTERFACE CAPABILITIES (Continued)

The ESCC can also execute an interrupt acknowledge cycle through software. In some CPU environments it is difficult to create the /INTACK signal with the necessary timing to acknowledge interrupts and allow the nesting of interrupts. In these cases, the /INTACK signal can be created with a software command to the ESCC. See the New Feature section for more details on this enhancement.

In the ESCC, the Interrupt Pending (IP) bit signals a need for interrupt servicing. When an IP bit is 1 and the IEI input is High, the /INT output is pulled Low, requesting an interrupt. In the ESCC, if the IE bit isn't set by enabling interrupts, then the IP for that source is never set. The IP bits are readable in RR3A.

The IUS bits signal that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority in the ESCC and external to the ESCC 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 ESCC 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 three types of interrupts: Transmit, Receive, and External/Status. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receiver, Transmit, and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled (WR1 D1=1), the occurrence of the interrupt depends on the state of WR7' D5. If this bit is reset, the CPU is interrupted when the top byte of the transmit FIFO becomes empty. If WR7' D5 is set, the CPU is interrupted when the transmit FIFO is completely empty. (This implies that the transmitter must have had a data character written into it so that it can become empty.)

When enabled, the receiver can interrupt the CPU in one of three ways:

- Interrupt on First Receive Character or Special Receive Condition.
- Interrupt on All Receive Characters or Special Receive Conditions.
- Interrupt on Special Receive Conditions Only.

If WR7' bit D3 is set, the Receive character interrupt occurs when there are four bytes available in the receive FIFO. This is most useful in synchronous applications as the data is in consecutive bytes. Interrupt on First Character or Special Condition and Interrupt on Special Condition Only are typically used with the Block Transfer mode. A special Receive Condition is one of the following: receiver overrun, framing error in Asynchronous mode, end-of-frame in SDLC mode and, optionally, a parity error. The Special Receive Condition interrupt is different from an ordinary receive character available interrupt only by the status placed in the vector during the Interrupt Acknowledge cycle. In Interrupt on First Receive Character, an interrupt occurs from Special Receive Conditions any time after the first receive character interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the /CTS, /DCD, and /SYNC pins, however, an External/Status interrupt is also caused by a Transmit Underrun condition; a zero count in the baud rate generator; by the detection of a Break (Asynchronous mode), Abort (SDLC mode) or EOP (SDLC Loop mode) sequence in the data stream. The interrupt caused by the Abort or EOP has a special feature allowing the ESCC to interrupt when the Abort or EOP sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Abort condition by external logic in SDLC mode. In SDLC Loop mode, this feature allows secondary stations to recognize the primary station wishes to regain control of the loop during a poll sequence.

CPU/DMA Block Transfer. The ESCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers. The Block Transfer mode used the /WAIT//REQUEST output in conjunction with the Wait/Request bits in WR1. The /WAIT//REQUEST output can be defined under software control as a WAIT line in the CPU Block Transfer mode or as a REQUEST line in the DMA Block Transfer mode.

To a DMA controller, the ESCC REQUEST output indicates that the ESCC is ready to transfer data to or from memory. To the CPU, the WAIT line indicates that the ESCC is not ready to transfer data, thereby requesting that the CPU extend the I/O cycle. The /DTR//REQUEST line allows full-duplex operation under DMA control. The ESCC can be programmed to deassert the /DTR//REQUEST pin with the same timing as the /WAIT//REQUEST pin if WR7' D4 is set.

ESCC DATA COMMUNICATIONS CAPABILITIES

The ESCC provides two independent full-duplex programmable channels for use in any common asynchronous or synchronous data communication protocols (Figure 8). Each of the datacommunication channels has identical features and capabilities.

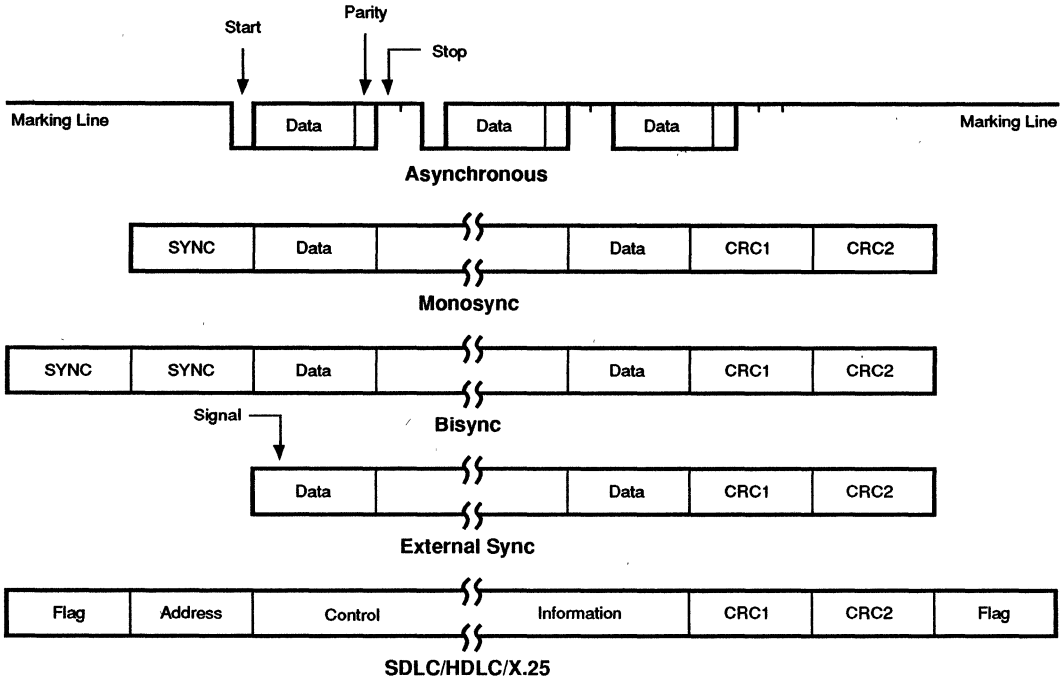


Figure 8. Some ESCC Protocols

The ESCC has significant improvements to its data communications capacity over that of the standard SCC. The addition of the deeper data FIFOs allows for data to be moved in strings instead of on a byte-by-byte basis. The ability to handle data in strings allows for significant improvements in data handling and, consequently, more efficient use of bus bandwidth. The programmability of the INT/DMA level of the FIFOs allows the system designer to determine fill levels as the FIFO's request the system to move data. The deeper data FIFOs are accessible regardless of the protocol used. They do not need to be enabled For more details on these improvements, see the New Feature section of this specification.

Asynchronous Modes. Send and Receive is accomplished independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one-and-a-half, or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break. Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxDA or RxDB pins). If the Low does not persist (e.g., a transient), the character assembly process does not start.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occur. Vectored interrupts allow fast servicing or error conditions using dedicated routines. Furthermore, a built-in checking process avoids the interpretation of a framing error as a new start bit: a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit begins.

ESCC DATA COMMUNICATIONS CAPABILITIES (Continued)

The ESCC does not require symmetric transmit and receive clock signals - a feature allowing use of the wide variety of clock sources. The transmitter and receiver handle data at a rate supplied to the receive and transmit clock inputs. In Asynchronous modes, the SYNC pin may be programmed as an input used for functions such as monitoring a ring indicator.

Synchronous Modes. The ESCC supports both byte-oriented and bit-oriented synchronous communication. Synchronous byte-oriented protocols are handled in several modes. They allow character synchronization with a 6-bit or 8-bit sync character (Monosync), and a 12-bit or 16-bit synchronization pattern (Bisync), or with an external sync signal. Leading sync characters are removed without interrupting the CPU.

Five or 7-bit synchronous characters are detected with 8- or 16-bit patterns in the ESCC by overlapping the larger pattern across multiple incoming synchronous characters as shown in Figure 9.

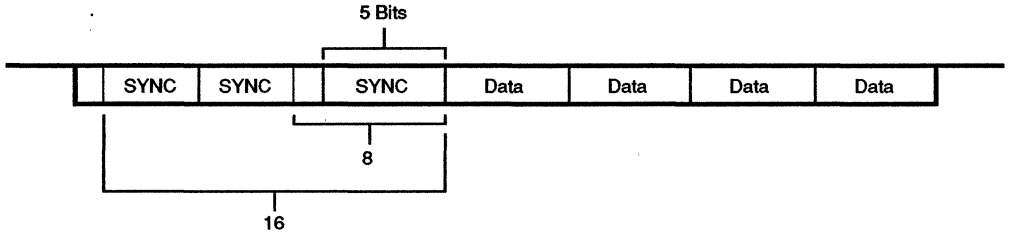


Figure 9. Detecting 5- or 7-Bit Synchronous Characters

CRC checking for Synchronous byte oriented modes is delayed by one character time so that the CPU may disable CRC checking on specific characters. This permits the implementation of protocols such as IBM Bisync.

Both CRC-16 (X16 + X15 + X2 + 1) and CCITT (X16 + X12 + X5 + 1) error checking polynomials are supported. Either polynomial may be selected in all Synchronous modes. Users may preset the CRC generator and checker to all 1's or all 0's. The ESCC also provides a feature that automatically transmits CRC data when no other data is available for transmission. This allows for high speed transmissions under DMA control, with no need for CPU intervention at the end of a message. When there is no data or CRC to send in Synchronous modes, the transmitter inserts 6-,8-, or 16-bit sync characters, regardless of the programmed character length.

SDLC Mode. The ESCC supports Synchronous bit-oriented protocols, such as SDLC and HDLC, by performing automatic flag sending, zero insertion, and CRC generation. A special command is used to abort a frame in transmission. At the end of a message, the ESCC automatically transmits the CRC and trailing flag when the transmitter underruns. The transmitter may also be programmed to send an idle line consisting of continuous flag characters or a steady marking condition.

If a transmit underrun occurs in the middle of a message, an external/status interrupt warns the CPU of this status change so that an abort can be issued. The ESCC may also be programmed to send an abort itself in case of an underrun, relieving the CPU of this task. One to eight bits per character can be sent, allowing reception of a message with no prior information about the character structure in the information field of a frame.

The receiver automatically acquires synchronization on the leading flag of a frame in SDLC or HDLC and provides a synchronization signal on the /SYNC pin (an interrupt can also be programmed). The receiver can be programmed to search for frames addressed by a single byte (or four bits within a byte) of a user-selected address or to a global broadcast address. In this mode, frames not matching either the user-selected or broadcast address are ignored.

The number of address bytes are extended under software control. For receiving data, an interrupt on the first received character, or an interrupt on every character, or on special condition only (end-of-frame) can be selected. The receiver automatically deletes all 0's inserted by the transmitter during character assembly. CRC is also calculated and is automatically checked to validate frame transmission. At the end of transmission, the status of a received frame is available in the status registers. In SDLC mode, the ESCC

must be programmed to use the SDLC CRC polynomial, but the generator and checker may be preset to all 1's or all 0's. The CRC is inverted before transmission and the receiver checks against the bit pattern 0001110100001111.

NRZ, NRZI or FM coding may be used in any 1x mode. The parity options available in Asynchronous modes are available in Synchronous modes.

SDLC Loop Mode. The ESCC supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, there is a primary controller station that manages the message traffic flow on the loop and any number of secondary stations. In SDLC Loop mode, the ESCC performs the functions of a secondary station while an ESCC operating in regular SDLC mode acts as a controller (Figure 10). SDLC loop mode can be selected by setting WR10 bit D1.

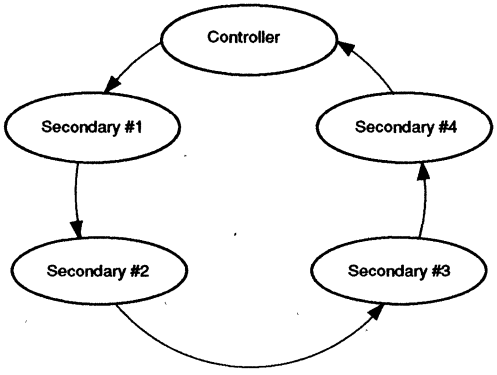


Figure 10. An SDLC Loop

A secondary station in an SDLC Loop is always listening to the messages being sent around the loop and, in fact, passes these messages to the rest of the loop by retransmitting them with a one-bit-time delay. The secondary station places its own message on the loop only at specific times. The controller signals that secondary stations can transmit messages by sending a special character, called an EOP (End Of Poll), around the loop. The EOP character is the bit pattern 11111110. Because of zero insertion during messages, this bit pattern is unique and easily recognized.

When a secondary station has a message to transmit and recognizes an EOP on the line, it changes the last binary 1 of the EOP to a 0 before transmission. This has the effect of turning the EOP into a flag sequence. The secondary station now places its message on the loop and terminates the message with an EOP. Any secondary stations further down the loop with messages to transmit appends their messages to the message of the first secondary station by the same process. Any secondary stations without messages to send merely echo the incoming message and are prohibited from placing messages on the loop (except upon recognizing an EOP). In SDLC Loop mode, NRZ, NRZI, and FM coding may all be used.

SDLC FIFO. The ESCC's ability to receive high speed back-to-back SDLC frames is maximized by a 10- deep by 19-bit wide status FIFO. When enabled (through WR15, bit D2), it provides the DMA the ability to continue to transfer data into memory so that the CPU can examine the message later. For each SDLC frame, a 14-bit byte count and 5 status/error bits are stored. The byte count and status bits are accessed through Read Registers 6 and 7. Read Registers 6 and 7 are only accessible when the SDLC FIFO is enabled The 10x19 status FIFO is separate from the 8-byte receive data FIFO.

Baud Rate Generator. Each channel in the ESCC contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On startup, the flip-flop on the output is set in a High state, the value in the time constant register is loaded into the counter, and the counter starts counting down The output of the baud rate generator toggles upon reaching 0, the value in the time constant register is loaded into the counter, and the process is repeated. The time constant may be changed at any time, but the new value does not take effect until the next load of the counter.

The output of the baud rate generator may be used as either the transmit clock, the receive clock, or both It can also drive the Digital Phase-Locked Loop (see next section)

If the receive clock or transmit clock is not programmed to come from the TRxC pin, the output of the baud rate generator may be echoed out via the TRxC pin.

The following formula relates the time constant to the baud rate where PCLK or RTxC is the baud rate generator input frequency in Hertz. The clock mode is 1, 16, 32, or 64, as selected in Write Register 4, bits D6 and D7. Synchronous operation modes should select 1 and Asynchronous should select 16, 32 or 64.

$$\text{Time Constant} = \frac{\text{PLCK or RTxC Frequency}}{\text{2(Baud Rate) (Clock Mode)}} - 2$$

Digital Phase-Locked Loop. The ESCC contains a Digital Phase-Locked Loop (DPLL) to recover clock information from a data stream with NRZI or FM encoding The DPLL

is driven by a clock that is nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock is then used as the ESCC receive clock, the transmit clock, or both. When the DPLL is selected as the transmit clock source, it will provide a jitter free clock output that is the DPLL input frequency divided by the appropriate divisor for the selected encoding technique.

For NRZI encoding, the DPLL counts the 32x clock to create nominal bit times. As the 32x clock is counted, the DPLL is searching the incoming data stream for edges (either 1 to 0, or 0 to 1). Whenever an edge is detected, the DPLL makes a count adjustment (during the next counting cycle), producing a terminal count closer to the center of the bit cell.

For FM encoding, the DPLL still counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream should occur between counts 15 and 16 and between counts 31 and 0. The DPLL looks for edges only during a time centered on the 15 to 16 counting transition.

The 32x clock for the DPLL can be programmed to come from either the RTxC input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the ESCC via the TRxC pin (if this pin is not being used as an input).

Data Encoding. The ESCC may be programmed to encode and decode the serial data in four different ways (Figure 11). In NRZ encoding, a 1 is represented by a High level and a 0 is represented by a Low level. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level. In FM1 (more properly, bi-phase mark), a transition occurs at the beginning of every bit cell. A 1 is represented by an additional transition at the center of the bit cell and a 0 is represented by no additional transition at the center of the bit cell. In FM0 (bi-phase space), a transition occurs at the beginning of every bit cell. A 0 is represented by an additional transition at the center of the bit cell, and a 1 is represented by no additional transition at the center of the bit cell. In addition to these four methods, the ESCC can be used to decode Manchester (bi-phase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0 to 1, the bit is a 0. If the transition is 1 to 0, the bit is a 1.

Auto Echo and Local Loopback. The ESCC is capable of automatically echoing everything it receives. This feature is useful mainly in Asynchronous modes, but works in Synchronous and SDLC modes as well. Auto Echo mode (TxD is RxD) is used with NRZI or FM encoding with no additional delay because the data stream is not decoded before retransmission. In Auto Echo mode, the /CTS input is ignored as a transmitter enable (although transitions on this input can still cause interrupts if programmed to do so). In this mode, the transmitter is actually bypassed and the programmer is responsible for disabling transmitter interrupts and /WAIT//REQUEST on transmit.

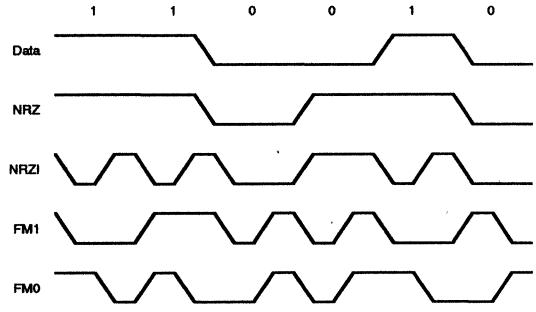


Figure 11. Data Encoding Methods

The ESCC is also capable of local loopback. In this mode, TxD or RxD is just like Auto Echo mode. However, in Local Loopback mode the internal transmit data is tied to the internal receive data and RxD is ignored (except to be echoed out via TxD). The /CTS and /DCD inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local Loopback works in Asynchronous, Synchronous and SDLC modes with NRZ, NRZI or FM coding of the data stream.

NEW FEATURE DESCRIPTION

The following is a detailed description of the enhancements to the Z85230, ESCC from the standard SCC.

4-Byte Deep Transmit FIFO
The ESCC has a 4-byte transmit buffer with programmable interrupt and DMA request levels. It is not necessary to enable the FIFO as it is always available. The user can choose to have the Transmit Buffer Empty (TBE) interrupt and DMA Request on Transmit be generated either when the top byte of transmit FIFO is empty or only when the FIFO is completely empty. A hardware or channel reset will reset the transmit shift register, flush the transmit FIFO, and set WR7' D5=1.

If the transmitter generates the Interrupt or DMA request for data when the top byte of the FIFO is empty (WR7' D5=0), the system can allow for a long response time to the data request without underflowing. The interrupt service routine can write one byte and then test RR0 D2 if more data may be written. The DMA Request in this mode will go inactive after each data write and then go active again until the FIFO is filled. The Transmit Buffer Empty status bit (TBE), RR0 bit D2, is set when the top byte of the FIFO is empty. Note that this in NOT the reset state.

For applications where the frequency of interrupts is important, the transmit interrupt service routine can be optimized by programming the ESCC to generate the TBE interrupt only when the FIFO is completely empty (WR7' D5=1) and then writing four bytes to fill the FIFO. When WR7' D5=1, only one DMA request is generated (filling the bottom of the FIFO). However, this may be preferred for some applications where the possible reassertion of the DMA request is not desired. The Transmit Buffer Empty status bit (TBE), RR0 bit D2, is set when the top byte of the FIFO is empty. (Note that WR7' D5=1 after a hardware or channel reset).

8-Byte Receive FIFO
The ESCC has an 8-byte receive FIFO with programmable interrupt levels. The receive character available interrupt is generated as selected by WR7' bit D3. The Receive Character Available bit, RR0 D0, is set when at least one byte is available in the top of the FIFO (independent of WR7' D3). It is not necessary to enable the 8-byte FIFO as it is always available. A hardware or channel reset resets the receive shift register and flushes the receive FIFO.

A DMA Request on Receive, if enabled, is generated whenever one byte is available in the receive FIFO independent of WR7' D3. If more than one byte is available in the FIFO, the /Wait//Request pin goes inactive and then goes active again until the FIFO is emptied.

By resetting WR7' D3=0, applications which have a long latency to interrupts can generate the request to read data from the FIFO when one byte is available, and then test the Receive Character Available bit to determine if more data is available.

By setting WR7 D3=1, the ESCC can be programmed to interrupt when the receive FIFO is half full (4 bytes available) and, therefore, allowing the frequency of receive interrupt to be reduced. If WR7' D3 is set, the receive character available interrupt is generated when there are 4 bytes available. Therefore, if the interrupt service routine reads 4 bytes during each routine, the frequency of interrupts is reduced.

If WR7' D3=1 and "Receive Interrupt on All Characters and Special Conditions" is enabled, the receive character available interrupt is generated when four characters are available. However, when a character is detected to have a special condition, a special condition interrupt is generated when the character is loaded into the top four bytes of the FIFO. Therefore, the special condition interrupt service routine should read RR1 before reading the data to determine which byte has the special condition.

Write Register 7' (7 prime)
A new register, WR7', has been added to the ESCC to facilitate the programming of six new features. The format of this register is shown in Figure 12.

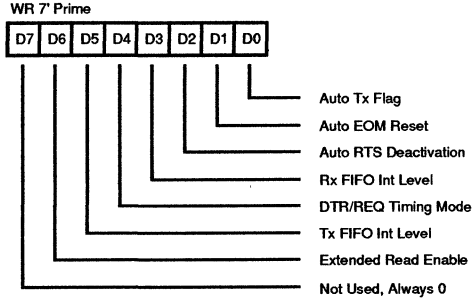


Figure 12. Write Register 7' (7 prime)

WR7' is written to by first setting bit D0 of Write Register 15 (WR15 D0) to one, and then addressing WR7 as normal. All writes to register 7 are to WR7' while WR15 D0 is set. WR15 bit D0 must be reset to 0 to address the sync character register WR7. If bit D6 of WR7' is set, then WR7' can be read by doing a read cycle to RR14. The WR7' features remain enabled until specifically disabled or by a hardware or software reset. Note that bit D5 is set after a reset All other bits are reset to zero following reset.

NEW FEATURE DESCRIPTION (Continued)

For applications which may use either the Zilog Z85C30 or Z85230, these two device types can be identified in software with the following test. Write a 01 hex to Write Register 15. Then read Read Register 15 and if D0 is reset it is a Z85C30 and, if D0 is set it is a Z85230. Note that if the device is Z85C30, a write to WR15 resetting D0 should be done before proceeding. Also, if the device is Z85230, the result in all writes to address seven will be to WR7' until WR15 D0 is reset.

Bit 7. Not used. This bit must always be written zero (0).

Bit 6. Extended Read Enable. Setting this bit enables the ability to read WR3, WR4, WR5, WR7' and WR10. These registers are read by reading RR9 (WR3), RR4, RR5, RR14 (WR7'), and RR11 (WR10), respectively.

Bit 5. Transmit FIFO Interrupt Level. If this bit is set, the transmit buffer empty interrupt is generated when the transmit FIFO is completely empty. If this bit is reset, the transmit buffer empty interrupt is generated when the top byte of the transmit FIFO is empty. This bit is set following a hardware or channel reset.

In DMA Request on Transmit mode, when using either the /W//REQ or /DTR//REQ pins, the request is asserted when the Tx FIFO is completely empty if WR7' D5 is set. The request is asserted when the top byte of the FIFO is empty if D5 is reset.

Bit 4. /DTR//REQ timing. If this bit is set and the /DTR//REQ pin is used for Request mode (WR14 D2=1), the deactivation of the /DTR//REQ pin will be identical to the /W//REQ pin as shown in Figure 13. If this bit is reset, the deactivation time is 4TcPc.

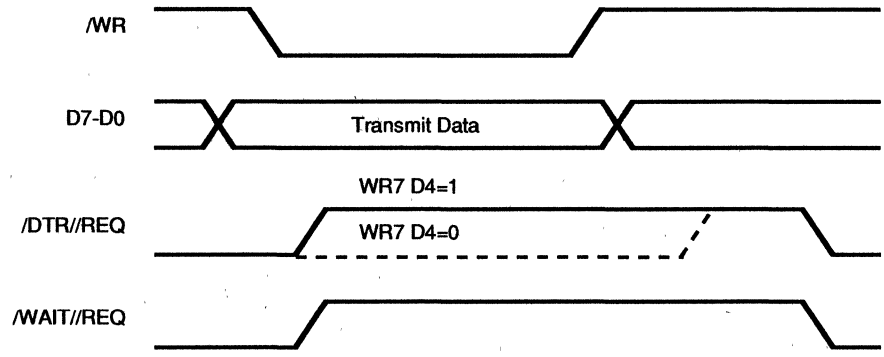


Figure 13. DMA Request on Transmit Deactivation Timing

Bit 3. Receive FIFO Interrupt Level. This bit sets the interrupt level of the receive FIFO. If this bit is set, the receive data available bit is asserted when the receive FIFO is half full (4 bytes available). If the RFF bit is reset, the receive data available interrupt is generated when a byte reaches the top of the FIFO. See the description of the 8 byte receive FIFO for more details.

Bit 2. Automatic /RTS Pin Deassertion. This bit controls the timing of the deassertion of the /RTS pin in SDLC mode. If this bit is set and WR5 D1 is reset during the transmission of a SDLC frame, the deassertion of the /RTS pin is delayed until the last bit of the closing flag clears the TxD pin. The /RTS pin is pulled high after the rising edge of the transmit clock cycle from the last bit of the closing flag. This implies that the ESCC should be programmed for "Flag on Underrun" (WR10 D2=0) for the /RTS pin to deassert at the end of the frame. This feature works independently of the programmed transmitter idle state. In synchronous modes other than SDLC, the /RTS pin will immediately follow the state programmed into WR5 D1. When WR7' D2 is reset, the /RTS follows the state of WR5 D1.

Bit 1. Automatic EOM Reset. If this bit is set, the ESCC automatically resets the Tx Underrun/EOM latch and presets the transmit CRC generator to its programmed preset state (per values set in WR5 D2 and WR10 D7). Therefore, it is not necessary to issue the Reset Tx Underrun/EOM latch command when this feature is enabled.

Bit 0. Automatic Tx SDLC Flag. If this bit is set, the ESCC will automatically transmit an SDLC flag before transmitting data. This removes the requirement to reset the mark idle bit (WR10 D3) before writing data to the transmitter.

Modified Databus Timing
The ESCC's latching of the databus has been modified to simplify the CPU interface. The Z85C30 AC Timing parameter #29, Write Data to /WR falling minimum, has been changed for the Z85230 to: /WR falling to Write Data Valid maximum. See the AC Timing Characteristic section for the specified time at each clock speed. The databus must be valid no later than 20ns after the falling edge of /WR regardless of the system (PCLK) clock rate. The databus hold time, spec #30, remains at 0ns.

Historically, the SCC has latched the databus on the falling edge of /WR. However, as many CPUs do not guarantee that the databus is valid when the /WR pin goes low, Zilog has modified the databus timing to allow a maximum delay from the /WR signal going active low to the latching of the databus.

Complete CRC Reception in SDLC Mode
In SDLC mode, the entire CRC is clocked into the receive FIFO. The ESCC completes clocking in the CRC to allow it to be retransmitted, unaltered, or manipulated in software. In the SCC when the closing flag is recognized, the contents of the receive shift register are immediately transferred to the receive FIFO resulting in the last two bits of the CRC being lost. In the ESCC, it is not necessary to program this feature. When the closing flag is detected, the last two bits of the CRC are clocked into the receive FIFO. In all other synchronous modes, the ESCC does not clock in the last two CRC bits (same as SCC).

TxD Forced High in SDLC with NRZI Encoding When Marking Idle
When the ESCC is programmed for SDLC mode with NRZI data encoding and mark idle (WR10 D6=0, D5=1, D3=1), the TxD pin is automatically forced high when the transmitter goes to the mark idle state. There are several different ways for the transmitter to go into the idle state. In each of the following cases the TxD pin is forced high when the mark idle condition is reached: data, CRC, flag and idle; data, flag and idle; data, abort (on underrun) and idle; data, abort (command) and idle; idle flag and command to idle mark. The force high feature is disabled when the mark idle bit is reset.

This feature is used in combination with the automatic SDLC opening flag transmission feature, WR7' D0=1, to assure that data packets are properly formatted. Therefore, when these features are used together, it is not necessary for the CPU to issue any commands when using the force idle mode in combination with NRZI data encoding. If WR7' D0 is reset, like in the SCC, it is necessary to reset the mark idle bit (WR10 D3) to enable flag transmission before an SDLC packet is transmitted.

Faster Interrupt Response
The interrupt response time of the ESCC has been improved so that the /INT pin (regardless of the interrupt source) is asserted low up to two clock cycles earlier than in the SCC

Improved Transmit Interrupt Handling in Synchronous Modes
The ESCC latches the Transmit Buffer Empty (TBE) interrupt due to the CRC being loaded to the transmit shift register even if the TBE interrupt, due at the last data byte, has not yet been reset. Therefore, the end of a synchronous frame is guaranteed to generate two TBE interrupts even if a reset transmit buffer interrupt command for the data created interrupt is issued after (time "A" in Figure 14) the CRC interrupt had occurred. In this case, two reset TBE commands are required. The TxIP is latched if the EOM latch has been reset before the end of the frame.

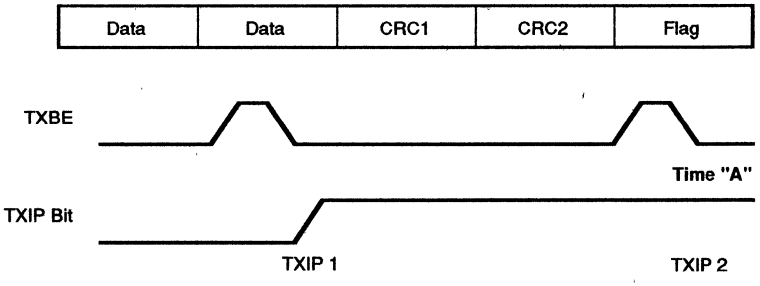


Figure 14. TxIP Latching

DPLL Counter Tx Clock Source

When DPLL output is selected as the transmit clock source, the DPLL counter output is the DPLL source clock divided by the appropriate divisor for the programmed data encoding format. Therefore, in FM mode (FM0 or FM1), the DPLL counter output is the input frequency divided by 16. In NRZI mode, the DPLL counter frequency is the input divided by 32. This feature provides a jitter-free output and replaces the DPLL transmit clock output being available as the transmit clock source. This has no effect on the use of the DPLL as the receive clock source.

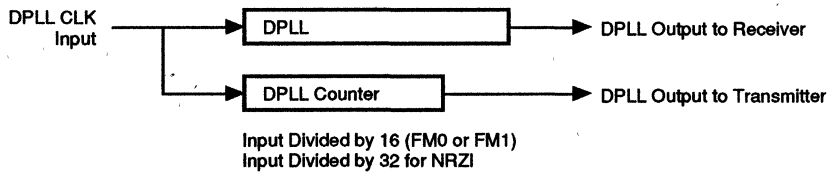


Figure 15. DPLL Outputs

Read Register 0 Status Latched During Read Cycle

The contents of Read Register zero, RR0, are latched during a read to this register. The ESCC prevents the contents of RR0 to change while the Read cycle is active. The SCC allows the status of RR0 to change while reading the register and, therefore, it is necessary to read RR0 twice to detect changes that otherwise may be missed. The contents of RR0 are updated after the rising edge of /RD.

Software Interrupt Acknowledge

The Z85230 interrupt acknowledge cycle can be initiated through software. If Write Register 9 (WR9) bit D5 is set, reading register 2 (RR2) results in an interrupt acknowledge cycle to be executed internally. Like a hardware INTACK cycle, a software acknowledge causes the INT pin to return high, the IEO pin to go low and set the IUS latch for the highest priority interrupt pending.

Similar to when the hardware INTACK signal can be used, a software acknowledge cycle requires that a Reset Highest IUS command be issued in the interrupt service routine. Whenever an interrupt acknowledge cycle is used, hardware or software, a reset highest IUS command is required. If RR2 is read from channel A, the unmodified vector is returned. If RR2 is read from channel B, then the vector is modified to indicate the source of the interrupt. The Vector Includes Status (VIS) and No Vector (NV) bits in WR9 are ignored when bit D5 is set to 1.

When the INTACK and IEI pins are not being used, they should be pulled up to Vcc through a resistor (10k ohm typical).

Fast SDLC Transmit Data Interrupt Response

To more easily facilitate the transmission of back-to-back SDLC frames with a single shared flag between frames, the ESCC allows data for a second frame to be written to the transmit FIFO after the Tx Underrun/EOM interrupt has occurred. This allows application software more time to write the data to the transmitter while allowing the current frame to be properly concluded with CRC and flag. The SCC historically has required that data not be written to the transmitter until a transmit buffer empty interrupt was generated after the CRC has completed transmission. If data is written to the transmit FIFO after the Transmit Underrun/EOM interrupt and before the transmit buffer empty interrupt, the Automatic EOM Reset feature should be enabled (WR7' D1=1). Consequently, the commands "Reset Tx/Underrun EOM" latch and "Reset Tx CRC Generator" should not be used.

SDLC FIFO Frame Status FIFO Enhancement

When used with a DMA controller, the Z85230 SDLC Frame Status FIFO enhancement maximizes the ESCC's ability to receive high speed, back-to-back SDLC messages. It minimizes frame overruns due to CPU latencies in responding to interrupts. Additional logic was added to the industry standard SCC consisting of a 10-deep by 19-bit status FIFO, 14-bit receive byte counter, and control logic as shown in Figure 16. The 10 x 19 bits status FIFO is separate from the 8-byte receive data FIFO.

When the enhancement is enabled, the status in read register 1 (RR1) and byte count for the SDLC frame are stored in the 10 x 19 bit status FIFO. This allows the DMA controller to transfer the next frame into memory while the CPU verifies that the message was properly received.

Summarizing the operation; data is received, assembled, and loaded into the eight byte FIFO before being transferred to memory by the DMA controller. When a flag is received at the end of an SDLC frame, the frame byte count from the 14-bit counter and five status bits are loaded into the status FIFO for verification by the CPU. The CRC checker is automatically reset in preparation for the next frame which can begin immediately. Since the byte count and status are saved for each frame, the message integrity is verified at a later time. Status information for up to 10 frames is stored before a status FIFO overrun can occur.

If a frame is terminated with an ABORT, the byte count is loaded to the status FIFO and the counter reset for the next frame.

FIFO Detail. For a better understanding of details of the FIFO operation, refer to the block diagram in Figure 16.

Enable/Disable. This FIFO is implemented so that it is enabled when WR15, bit D2, is set and the ESCC is in the SDLC/HDLC mode. Otherwise, the status register contents bypass the FIFO and go directly to the bus interface (the FIFO pointer logic is reset either when disabled or via a channel or power-on reset). When the FIFO mode is disabled, the ESCC is completely downward compatible with the NMOS Z8530. The FIFO mode is disabled on power-up (WR15 D2 is set to 0 on reset). The effects of backward compatibility on the register set are that RR4 is an image of RR0, RR5 is an image of RR1, RR6 is an image of RR2 and RR7 is an image of RR3. For the details of the added registers, refer to Figure 18. The status of the FIFO Enable signal is obtained by reading RR15, bit D2. If the FIFO is enabled, the bit will be set to 1; otherwise, it will be reset

Read Operation. When WR15 bit D2 is set and the FIFO is not empty, the next read to status register RR1 or the additional registers RR7 and RR6, are from the FIFO. Reading status register RR1 causes one location of the FIFO to be emptied, so status is read after reading the byte count, otherwise the count is incorrect. Before the FIFO underflows, it is disabled. In this case, the multiplexer is switched to allow status to read directly from the status register and reads from RR7 and RR6 contain bits that are undefined. Bit D6 of RR7 (FIFO Data Available) is used to determine if status data is coming from the FIFO or directly from the status register, since it is set to 1 whenever the FIFO is not empty.

Since not all status bits are stored in the FIFO, the All Sent, Parity, and EOF bits bypass the FIFO. The status bits sent through the FIFO are Residue Bits (3), Overrun, and CRC Error.

The sequence for proper operation of the byte count and FIFO logic is to read the registers in the following order: RR7, RR6, and RR1 (reading RR6 is optional). Additional logic prevents the FIFO from being emptied by multiple reads from RR1. The read from RR7 latches the FIFO empty/full status bit (D6) and steers the status multiplexer to read from the ESCC megacell instead of the status FIFO (since the status FIFO is empty). The read from RR1 allows an entry to be read from the FIFO (if the FIFO was empty, logic was added to prevent a FIFO underflow condition).

Write Operation. When the end of an SDLC frame (EOF) has been received and the FIFO is enabled, the contents of the status and byte-count registers are loaded into the FIFO. The EOF signal is used to increment the FIFO. If the FIFO overflows, the RR7 bit D7 (FIFO Overflow) is set to indicate the overflow. This bit and the FIFO control logic is reset by disabling and re-enabling the FIFO control bit (WR15 bit D2). For details of FIFO control timing during an SDLC frame, refer to Figure 17.

SDLC Status FIFO Anti-Lock Feature. When the Frame Status FIFO is enabled and the ESCC is programmed for "Special Receive Condition Only" (WR1 D4=D3=1), the data FIFO is not locked when a character with End of Frame status is read. When a character with the EOF status is at the top of the FIFO, an interrupt with a vector for receive data is generated. The command "Reset Highest IUS" must be issued at the end of the interrupt service routine regardless if an interrupt acknowledge cycle had been executed (hardware or software). This allows a DMA to complete transfer of the received frame to memory and then interrupt the CPU that a frame has been completed without locking the FIFO. Since in the "Receive Interrupt on Special Condition Only" mode the interrupt vector for receive data is not used, it is used to indicate that the last byte of a frame has been read out the receive FIFO. This eliminates having to read the frame status (CRC and other status is stored in the status FIFO with the frame byte count).

When a character with a special receive condition other than EOF is received (receiver overrun, or parity), a special receive condition interrupt is generated after the character is read from the FIFO and the receive FIFO is locked until the "Error Reset" command is issued.

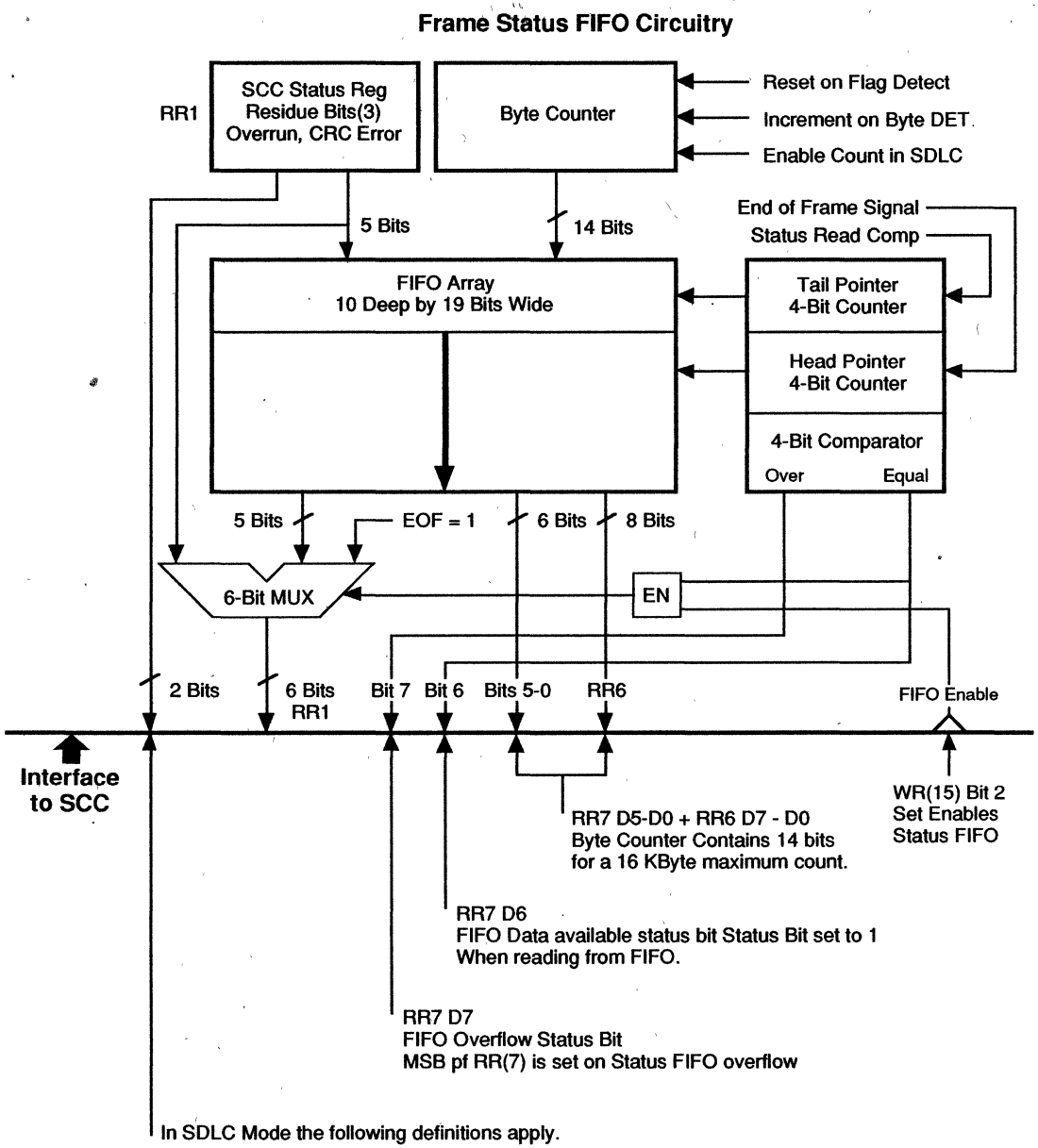


Figure 16. SDLC Frame Status FIFO

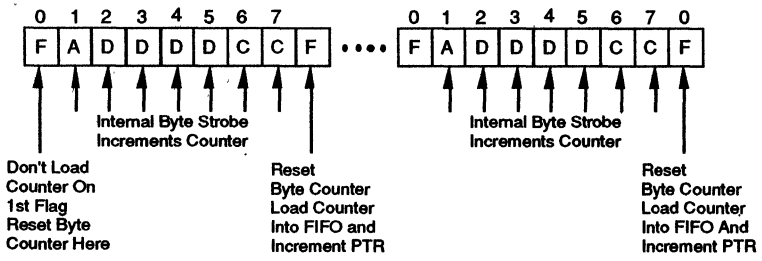


Figure 17. SDLC Byte Counting Detail

PROGRAMMING

The ESCC contains write registers in each channel that are programmed by the system separately to configure the functional uniqueness of the channels.

In the ESCC, the data registers are directly addressed by selecting a High on the D//C pin. With all other registers (with the exception of WR0 and RR0), programming the write registers requires two write operations and reading the read registers requires both a write and a read operation. The first write is to WR0 and contains three bits that point to the selected register. The second write is the actual control word for the selected register, and if the second operation is read, the selected read register is accessed. All of the ESCC registers, including the data registers, may be accessed in this fashion. The pointer bits are automatically cleared after the read or write operation so that WR0 (or RR0) is addressed again.

Initialization. The system program first issues a series of commands to initialize the basic mode of operation. This is followed by other commands to qualify conditions within the selected mode. For example, in the Asynchronous mode, character length, clock rate, number of stop bits, and even or odd parity should be set first. Then the interrupt mode is set, and finally, the receiver and transmitter are enabled.

Write Registers. The ESCC contains 16 write registers (17 counting the transmit buffer) in each channel. These write registers are programmed separately to configure the functional "personality" of the channels. There are two registers (WR2 and WR9) shared by the two channels that are accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits and reset commands. A new register, WR7', was added to the ESCC and may be written to if WR15 D0 is set. Figure 18 shows the format of each write register.

Read Registers. The ESCC contains ten read registers (eleven, counting the receive buffer (RR8) in each channel). Four of these may be read to obtain status information (RR0, RR1, RR10, and RR15). Two registers (RR12 and RR13) are read to learn the baud rate generator time constant. RR2 contains either the unmodified interrupt vector (Channel A) or the vector modified by status information (Channel B). RR3 contains the Interrupt Pending (IP) bits (Channel A only). RR6 and RR7 contain the information in the SDLC Frame Status FIFO, but is only read when WR15 D2 is set. If WR7' D6 is set, Write Registers WR3, WR4, WR5, WR7', and WR10 can be read as RR9, RR4, RR5, and RR14, respectively.

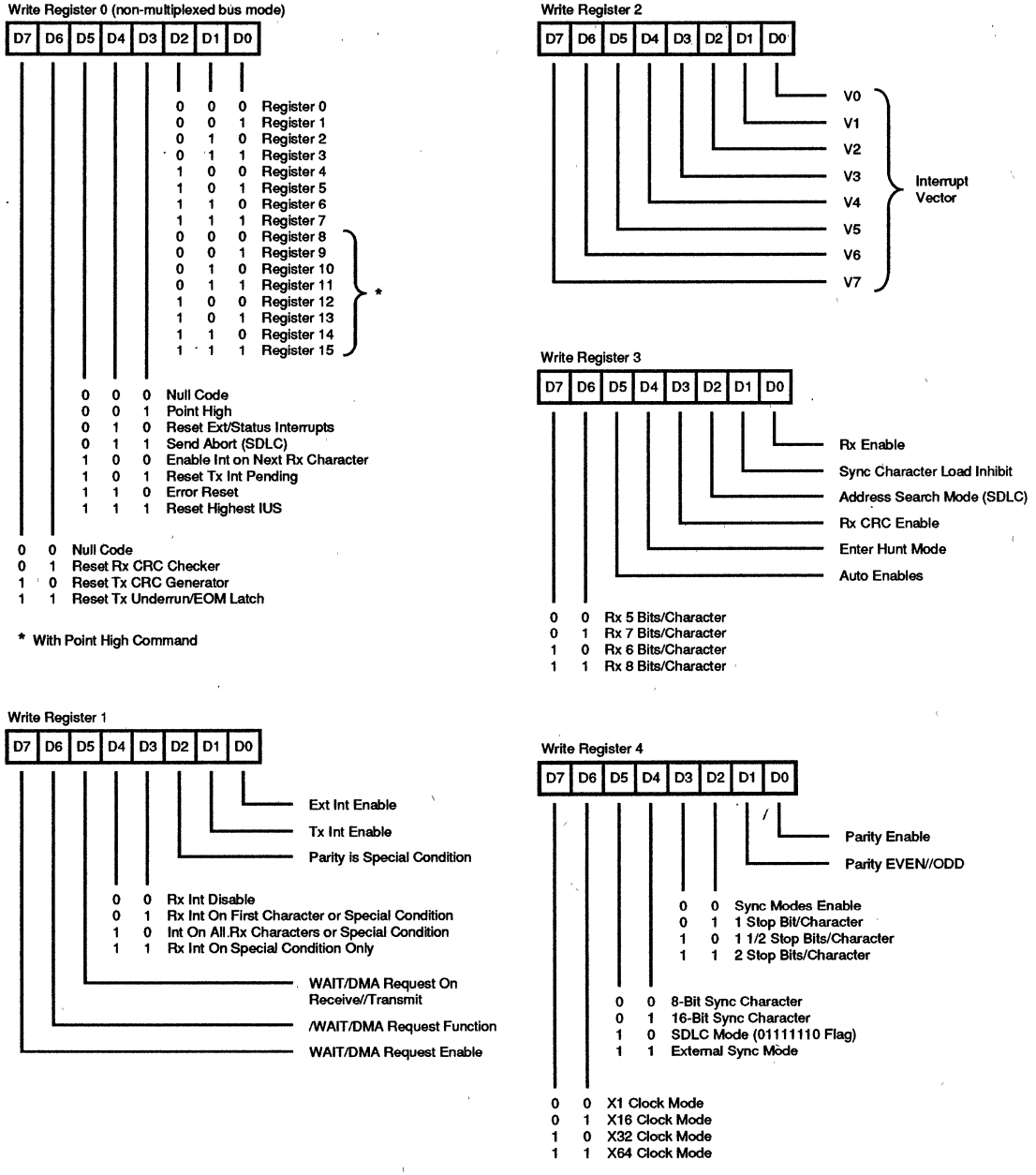


Figure 18. Write Register Bit Functions

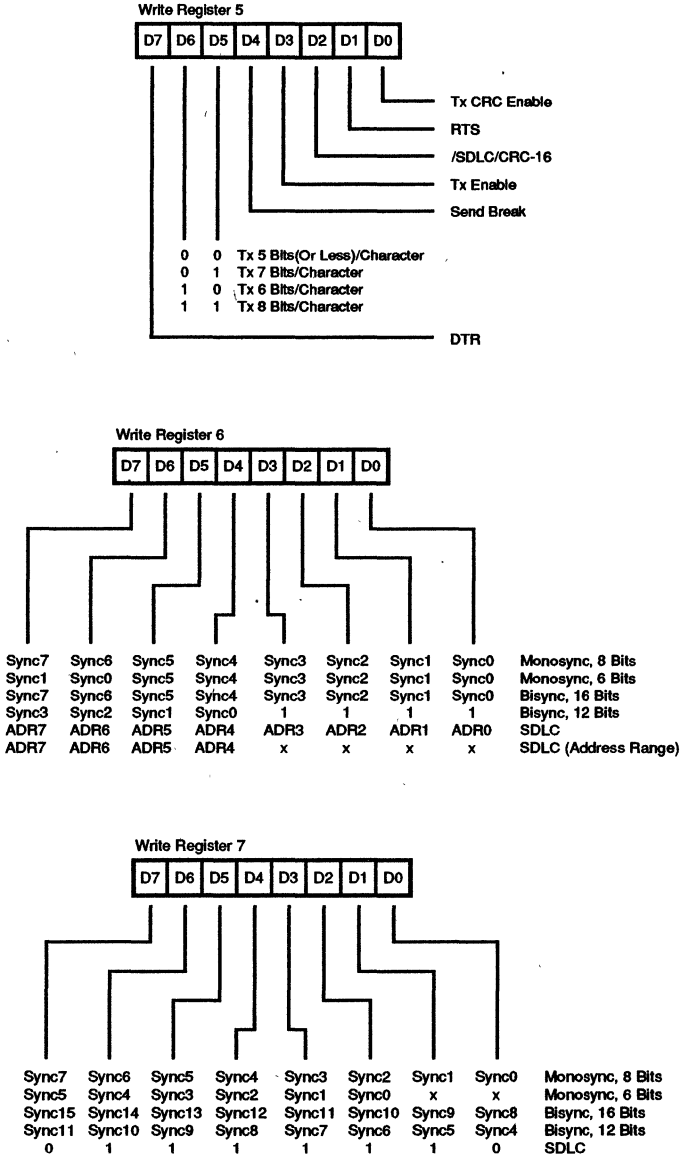


Figure 18. Write Register Bit Functions (Continued)

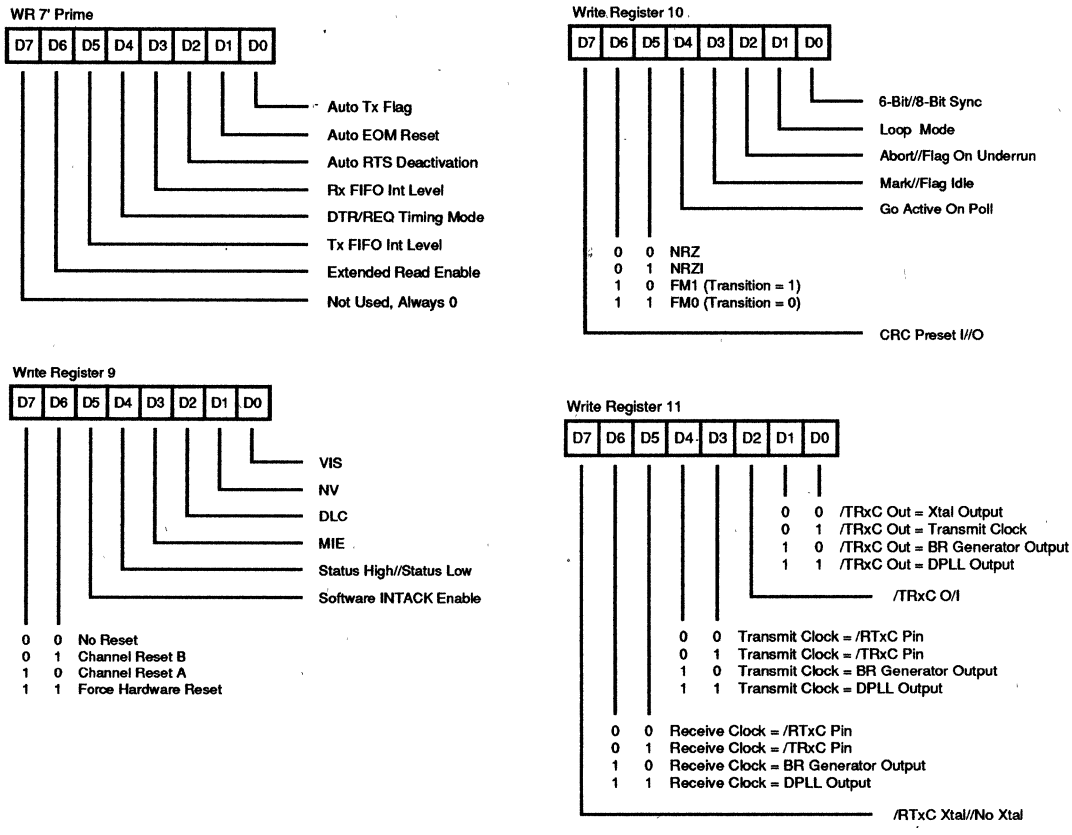


Figure 18. Write Register Bit Functions (Continued)

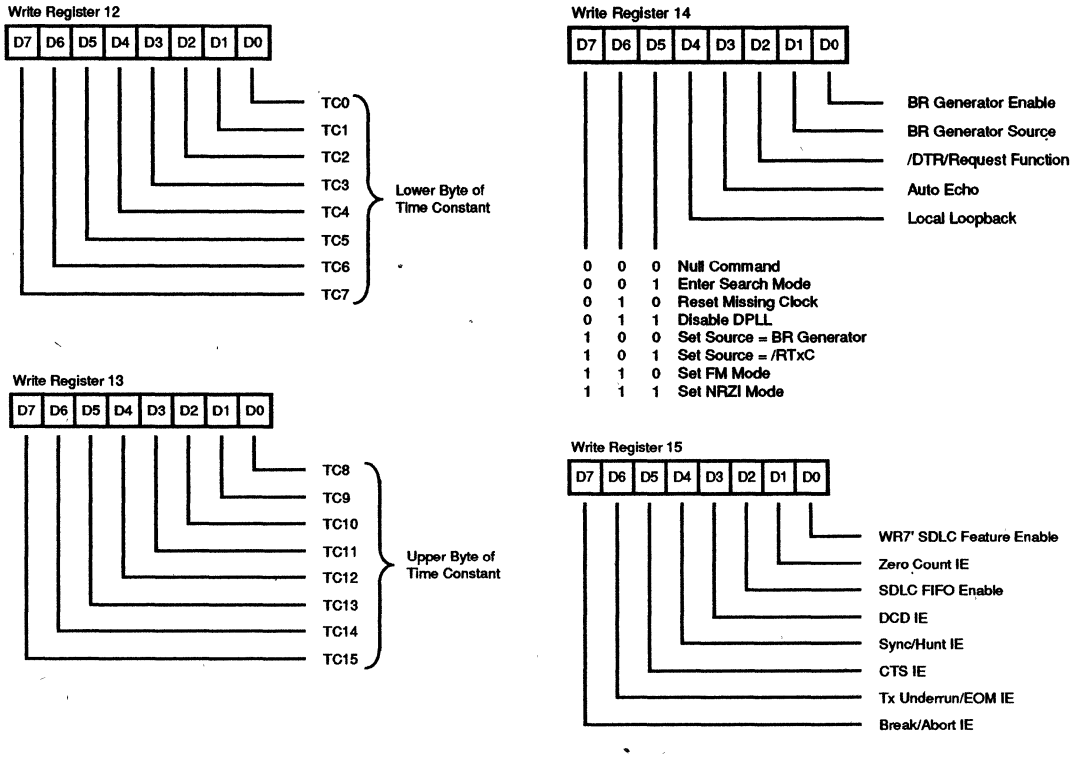


Figure 18. Write Register Bit Functions (Continued)

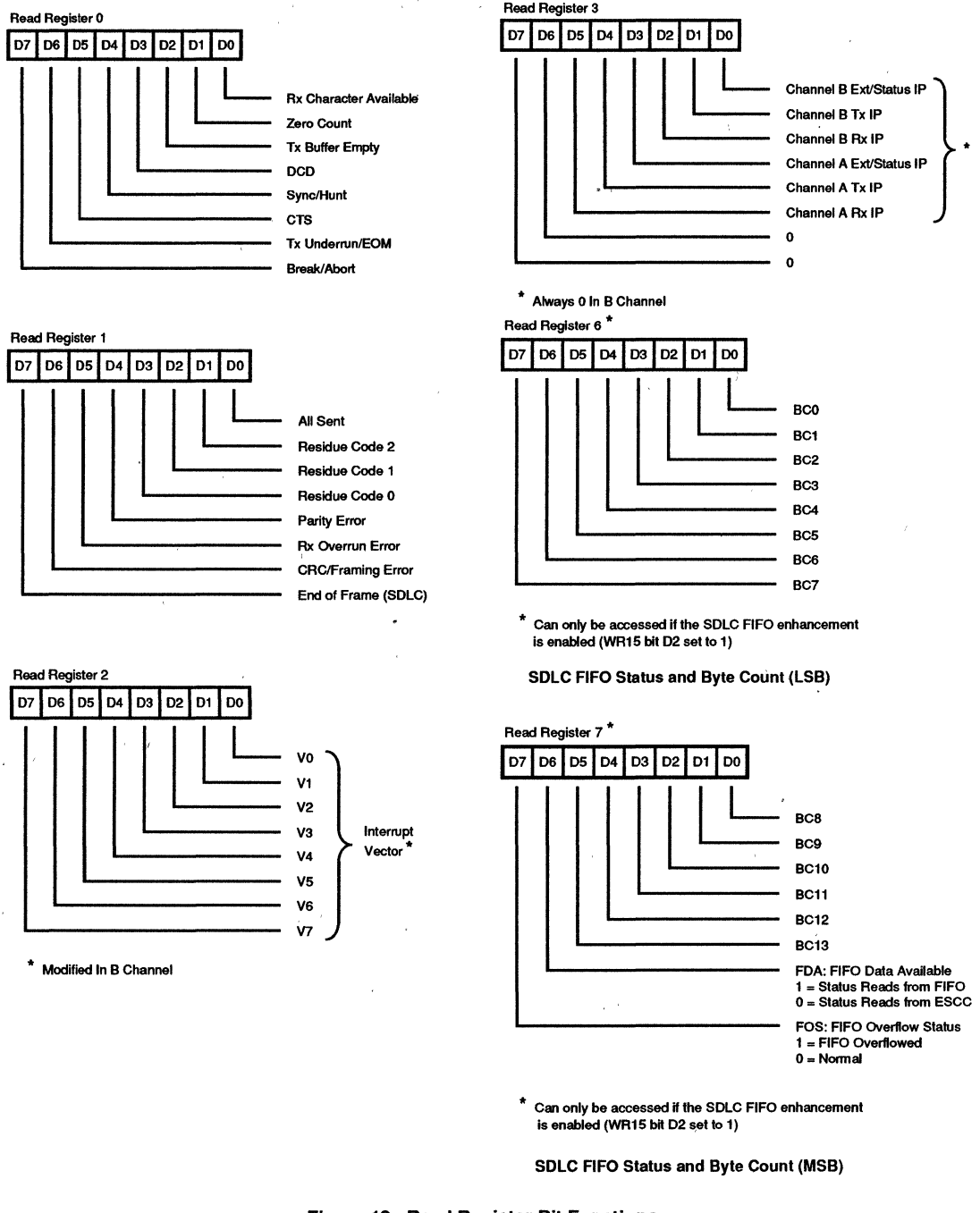


Figure 19. Read Register Bit Functions

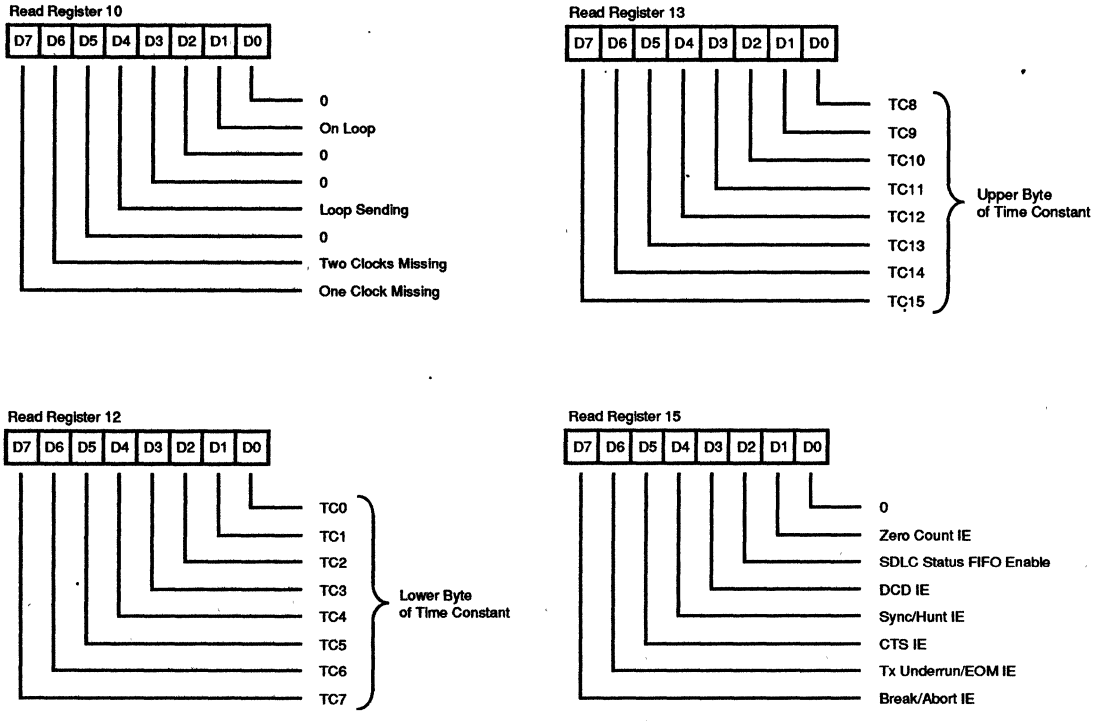


Figure 19. Read Register Bit Functions (Continued)

The ESCC generates internal control signals from the /WR and /RD that are related to PCLK. Since PCLK has no phase relationship with /WR and /RD, the circuitry generating the internal control signals provides time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the ESCC. The recovery time required for proper operation is specified from the falling edge of /WR or /RD in the first transaction involving the ESCC to the falling edge of /WR or /RD in the second transaction involving the ESCC. This time must be at least 4 PCLKs regardless of which register or channel is being accessed.

Read Cycle Timing. Figure 20 illustrates Read cycle timing. Addresses on A//B and D//C and the status on /INTACK must remain stable throughout the cycle. If /CE falls after /RD falls, or if it rises before /RD rises, the effective /RD is shortened.

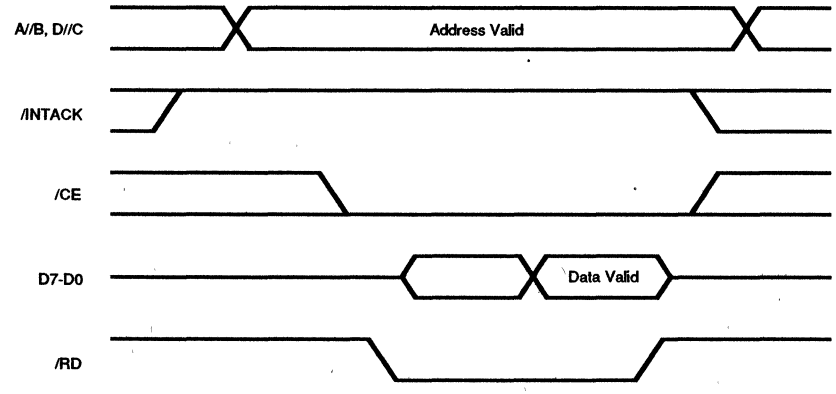


Figure 20. Read Cycle Timing

Write Cycle Timing. Figure 21 illustrates Write cycle timing. Addresses on A//B and D//C and the status on /INTACK must remain stable throughout the cycle. If /CE falls after /WR falls, or if it rises before /WR rises, the effective /WR is shortened. Because many popular CPUs do not guarantee that the databus is valid when /WR is driven low, the databus timing requirements of the ESCC have been modified so that the databus does not have to be valid when the /WR pin goes low. See AC Characteristic #29 for details.

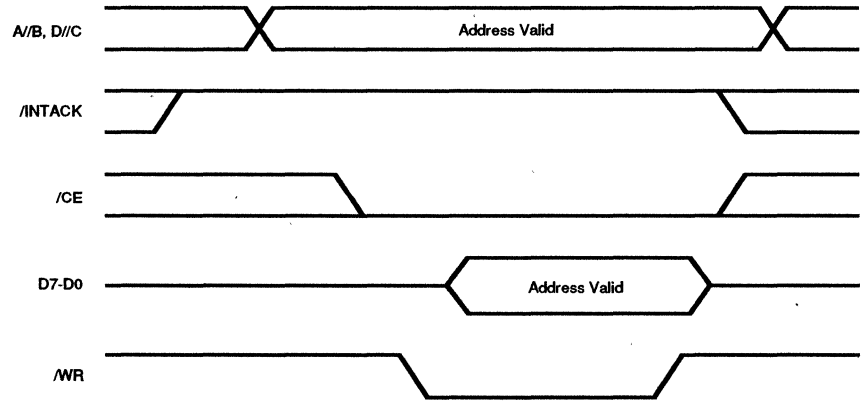


Figure 21. Write Cycle Timing

Interrupt Acknowledge Cycle Timing. Figure 22 illustrates Interrupt Acknowledge cycle timing. Between the time /INTACK goes Low and the falling edge of /RD, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the ESCC and IEI is High when /RD falls, the Acknowledge cycle is intended for the ESCC. In this case, the ESCC may be programmed to respond to /RD Low by placing its interrupt vector on D7-D0. It then sets the appropriate Interrupt-Under-Service latch internally. If the external daisy chain is not used, then AC parameter #38 is required to settle the interrupt priority daisy chain internal to the ESCC. If the external daisy chain is used, the user should follow the equation in AC Characteristics, Note 5, for calculating the required daisy-chain settle time.

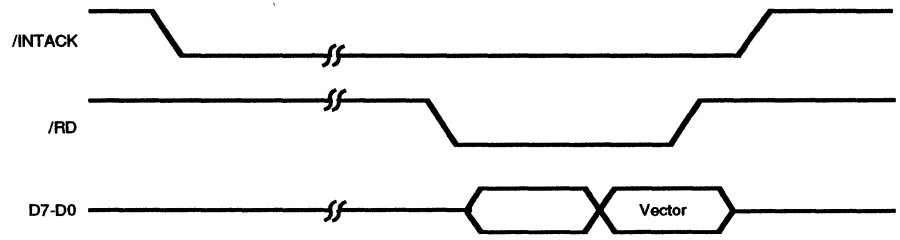


Figure 22. Interrupt Acknowledge Cycle Timing

OTHER ZILOG DATA COMMUNICATIONS PRODUCTS

SIO Family	
Z84C40 SIO	Dual channel multiprotocol USART.
Z84C13 IPC	Z80 CPU with integrated SIO, CTC and WDT.
Z84C15 IPC	Z80 CPU with integrated SIO, CTC, WDT and PIO.

SCC Family	
Z08530 SCC	NMOS SCC Low cost with speeds up to 8 MHz.
Z08030 SCC	NMOS SCC for multiplexed busses.
Z85C30 SCC	CMOS SCC at speeds up to 16 MHz. NMOS compatible.
Z80C30 SCC	CMOS SCC for multiplexed busses.
Z16C35 ISCC	SCC with 4 channel DMA and advanced CPU interface.
Z80181 SAC	Z180 CPU with integrated single channel SCC.

USC Family	
Z16C30 USC	Dual channel high performance multi-protocol data communications up to 10 Megabits/second
Z16C33 MUSC	Single channel USC w/ ISDN Time Slot Assigner.
Z16C31 IUSC	MUSC with high performance dual channel DMA (available Q1/91).
Z16C50 DDPLL	Dual channel DPLL cell from the USC.

ABSOLUTE MAXIMUM RATINGS

V_{CC} Supply Voltage range ..... -0.3V to +7.0V
Voltages on all pins
with respect to GND ..... -3V to V_{CC} +0.3V
Operating Ambient
Temperature ..... See Ordering Information
Storage Temperature ..... -65°C to +150°C

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 sections below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin.

Standard conditions are as follows:

- +4.50 V ≤ V_{CC} ≤ + 5.50 V
- GND = 0 V
- T_A as specified in Ordering Information

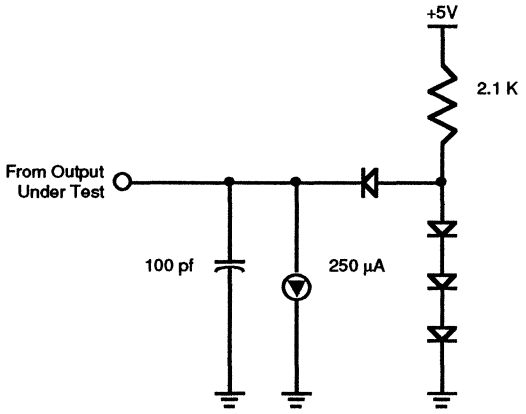


Figure 23. Standard Test Load

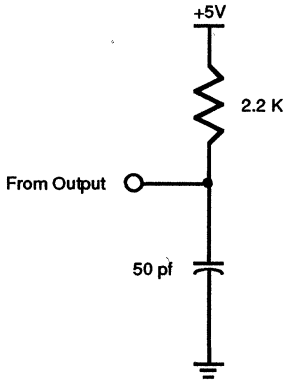


Figure 24. Open-Drain Test Load

CAPACITANCE

Symbol	Parameter	Min	Max	Unit	Test 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.
Unmeasured pins returned to Ground.

MISCELLANEOUS

Gate Count - 6800

DC CHARACTERISTICS
Z85230

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} = -1.6mA
V_{OH2}	Output High Voltage	V_{CC}-0.8			V	I_{OH} = -250µA
V_{OL}	Output Low Voltage			0.4	V	I_{OL} + 2.0mA
I_{IL}	Input Leakage			±10.0	µA	0.4 V_{IN}+2.4V
I_{OL}	Output Leakage			±10.0	µA	0.4 V_{OUT}+2.4V
I_{CC1}	V_{CC} Supply Current		5	12 (10 MHz)	mA	V_{CC}=5V V_{IH}=4.8 V_{IL}=0.2V
			7	15 (16 MHz)	mA	Crystal Oscillators off
			9	20 (20 MHz)	mA	
I_{CCOSC}	Crystal OSC Current		6		mA	Current for each osc. in addition to I_{CC1}

Notes:
[1] V_{CC} = 5V ± 10% unless otherwise specified, over specified temperature range.
[2] Typical I_{CC} was measured with oscillator off.
[3] No I_{CC}(osc) max is specified due to dependency on the external circuit.

AC CHARACTERISTICS

Z85230 Read and Write Timing Diagram

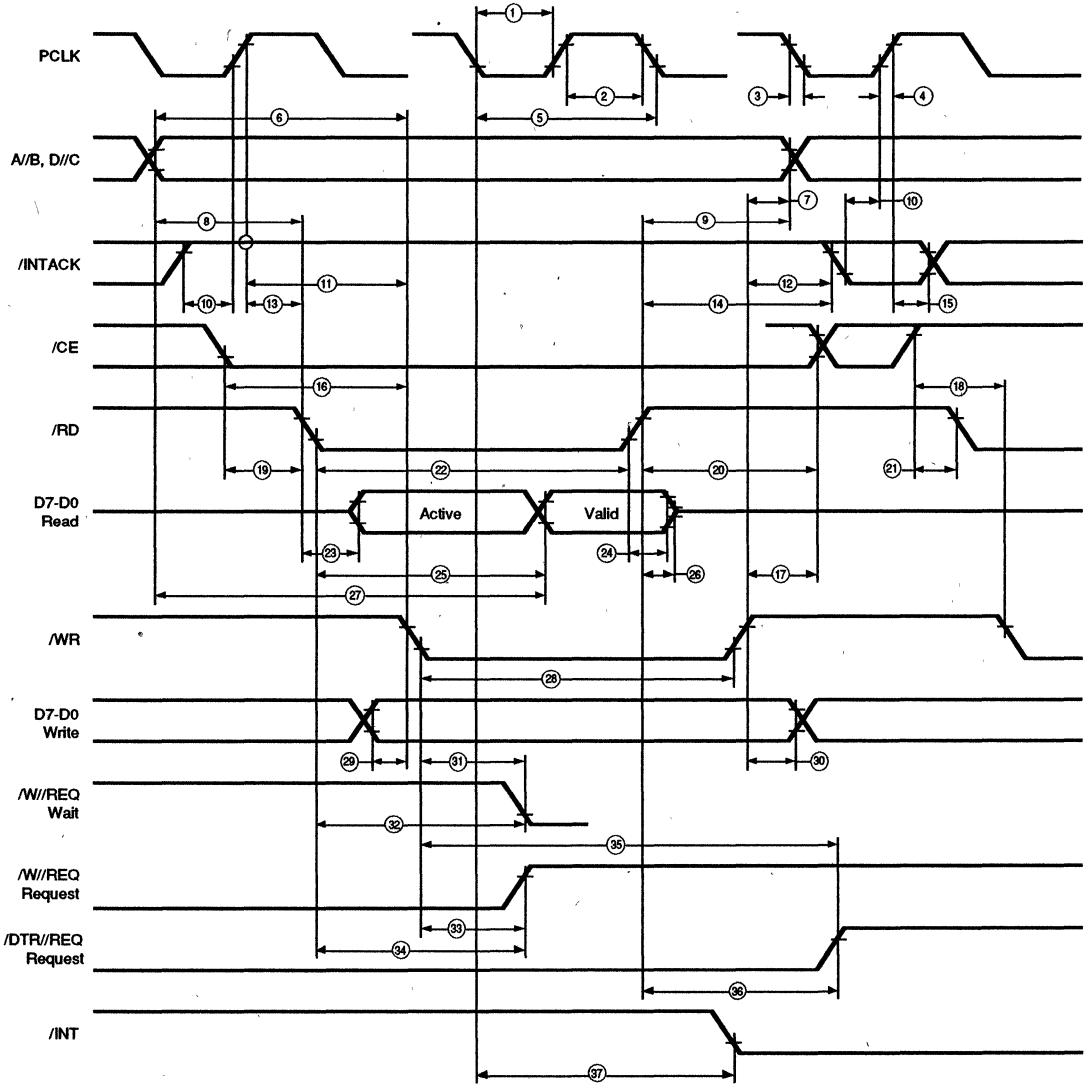


Figure 25. Z85230 Read and Write Timing Diagram

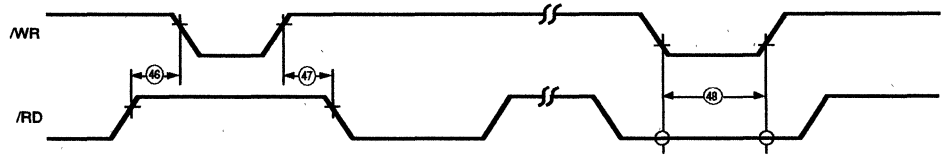


Figure 26. Reset Timing Diagram

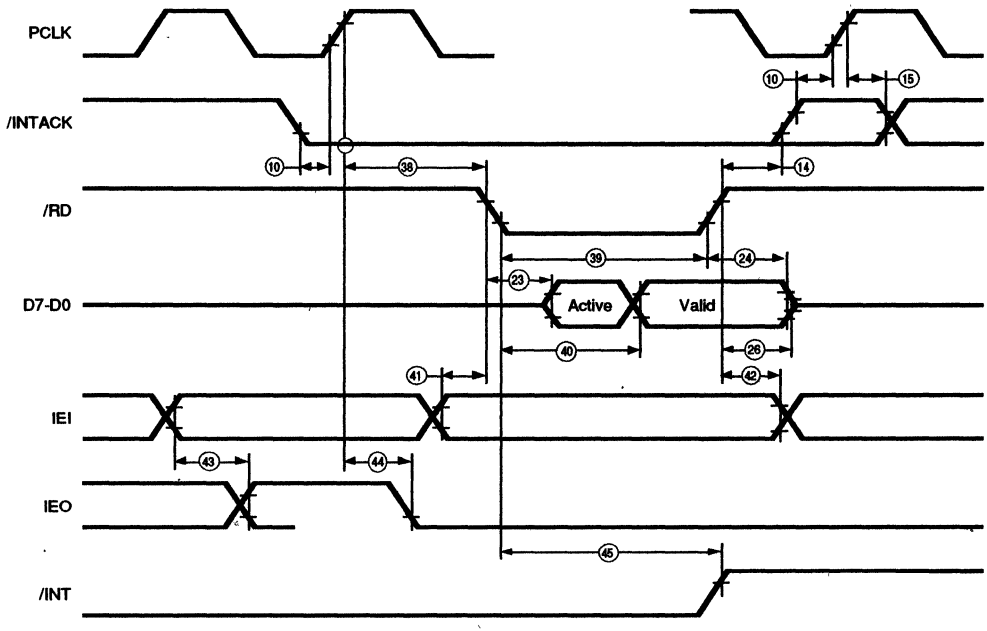


Figure 27. Interrupt Acknowledge Timing Diagram

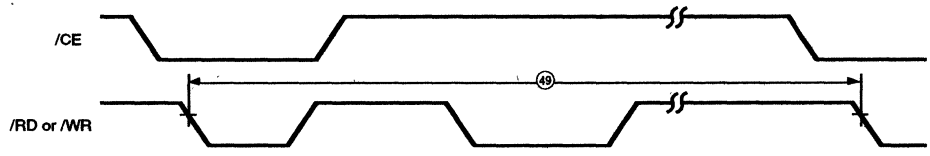


Figure 28. Cycle Timing Diagram

AC CHARACTERISTICS

Z85230 Read and Write Timing Table

No	Symbol	Parameter	10 MHz Min	10 MHz Max	16 MHz Min	16 MHz Max	20 MHz Min	20 MHz Max	Notes
1	TwPCl	PCLK Low Width	40	1000	26	1000	22	1000	
2	TwPCh	PCLK High Width	40	1000	26	1000	22	1000	
3	TfPC	PCLK Fall Time		10		5		5	
4	TrPC	PCLK Rise Time		10		5		5	
5	TcPC	PCLK Cycle Time	100	2000	61	2000	50	2000	
6	TsA(WR)	Address to /WR Fall Setup Time	50		35		30		
7	ThA(WR)	Address to /WR Rise Hold Time	0		0		0		
8	TsA(RD)	Address to /RD Fall Setup Time	50		35		30		
9	ThA(RD)	Address to /RD Rise Hold Time	0		0		0		
10	TsIA(PC)	/INTACK to PCLK Rise Setup Time	20		15		15		

AC CHARACTERISTICS

Z85230 Read and Write Timing Table

No	Symbol	Parameter	10 MHz Min	10 MHz Max	16 MHz Min	16 MHz Max	20 MHz Min	20 MHz Max	Notes
11	TslAi(WR)	/INTACK to /WR Fall Setup Time	130		70		65		[1]
12	ThIA(WR)	/INTACK to /WR Rise Hold Time	0		0		0		
13	TslAi(RD)	/INTACK to /RD Fall Setup Time	130		70		65		[1]
14	ThIA(RD)	/INTACK to /RD Rise Hold Time	0		0		0		
15	ThIA(PC)	/INTACK to PCLK Rise Hold Time	30		15		15		
16	TsCEI(WR)	/CE Low to /WR Fall Setup Time	0		0		0		
17	ThCE(WR)	/CE to /WR Rise Hold Time	0		0		0		
18	TsCEh(WR)	/CE High to /WR Fall Setup Time	50		30		25		
19	TsCEI(RD)	/CE Low to /RD Fall Setup Time	0		0		0		[1]
20	ThCE(RD)	/CE to /RD Rise Hold Time	0		0		0		[1]
21	TsCEh(RD)	/CE High to /RD Fall Setup Time	50		30		25		[1]
22	TwRDI	/RD Low Width	125		70		65		[1]
23	TdRD(DRA)	/RD Fall to Read Data Active Delay	0		0		0		
24	TdRDr(DR)	/RD Rise to Data Not Valid Delay	0		0		0		
25	TdRDI(DR)	/RD Fall to Read Data Valid Delay		120		70		65	
26	TdRD(DRz)	/RD Rise to Read Data Float Delay		35		20		20	
27	TdA(DR)	Addr to Read Data Valid Delay		180		100		90	
28	TwWRI	/WR Low Width	125		75		65		
29	TdWR(DW)	/WR Fall to Write Data Valid Delay		20		20		20	
30	ThDW(WR)	Write Data to /WR Rise Hold Time	0		0		0		
31	TdWR(W)	/WR Fall to Wait Valid Delay		100		50		50	[4]
32	TdRD(W)	/RD Fall to Wait Valid Delay		100		50		50	[4]
33	TdWRf(REQ)	/WR Fall to /W//REQ Not Valid Delay		120		70		65	
34	TdRDf(REQ)	/RD Fall to /W//REQ Not Valid Delay		120		70		65	[6]
35a	TdWRr(REQ)	/WR Fall to /DTR//REQ Not Valid		4TcPc		4TcPc		4TcPc	[6]
35b	TdWRr(REQ)	/WR Fall to /DTR//REQ Not Valid		100		70		65	
36	TdRDr(REQ)	/RD Rise to /DTR//REQ Not Valid Delay		NA		NA		NA	
37	TdPC(INT)	PCLK Fall to /INT Valid Delay		320		175		160	
38	TdIAi(RD)	/INTACK to /RD Fall (Ack) Delay	90		50		45		[5]
39	TwRDA	/RD (Acknowledge) Width	125		75		65		
40	TdRDA(DR)	/RD Fall(Ack) to Read Data Valid Delay	120		70		60		
41	TslEI(RDA)	IEI to /RD Fall (Ack) Setup Time	95		50		45		
42	ThIEI(RDA)	IEI to /RD Rise (Ack) Hold Time	0		0		0		
43	TdIEI(IEO)	IEI to IEO Delay Time		90		45		40	
44	TdPC(IEO)	PCLK Rise to IEO Delay		175		80		70	
45	TdRDA(INT)	/RD Fall to /INT Inactive Delay		320		200		180	[4]
46	TdRD(WRQ)	/RD Rise to /WR Fall Delay for No Reset	15		10		10		
47	TdWRQ(RD)	/WR Rise to /RD Fall Delay for No Reset	15		10		10		
48	TwRES	/WR and /RD Low for Reset	100		75		65		
49	Trc	Valid Access Recovery Time	4TcPc		4TcPc		4TcPc		[3]

Notes:

[1] Parameter does not apply to Interrupt Acknowledge transactions.

[3] Parameter applies only between transactions involving the ESCC.

[4] Open-drain output, measured with open-drain test load.

[5] Parameter is system dependent. For any ESCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain. TslEI(RDA) for the ESCC and TdIEI(IEO) for each device separating them in the daisy chain.

[6] Parameter applies to enhanced Request mode only (WR7' D4=1)

AC CHARACTERISTICS

Z85230 General Timing Diagram

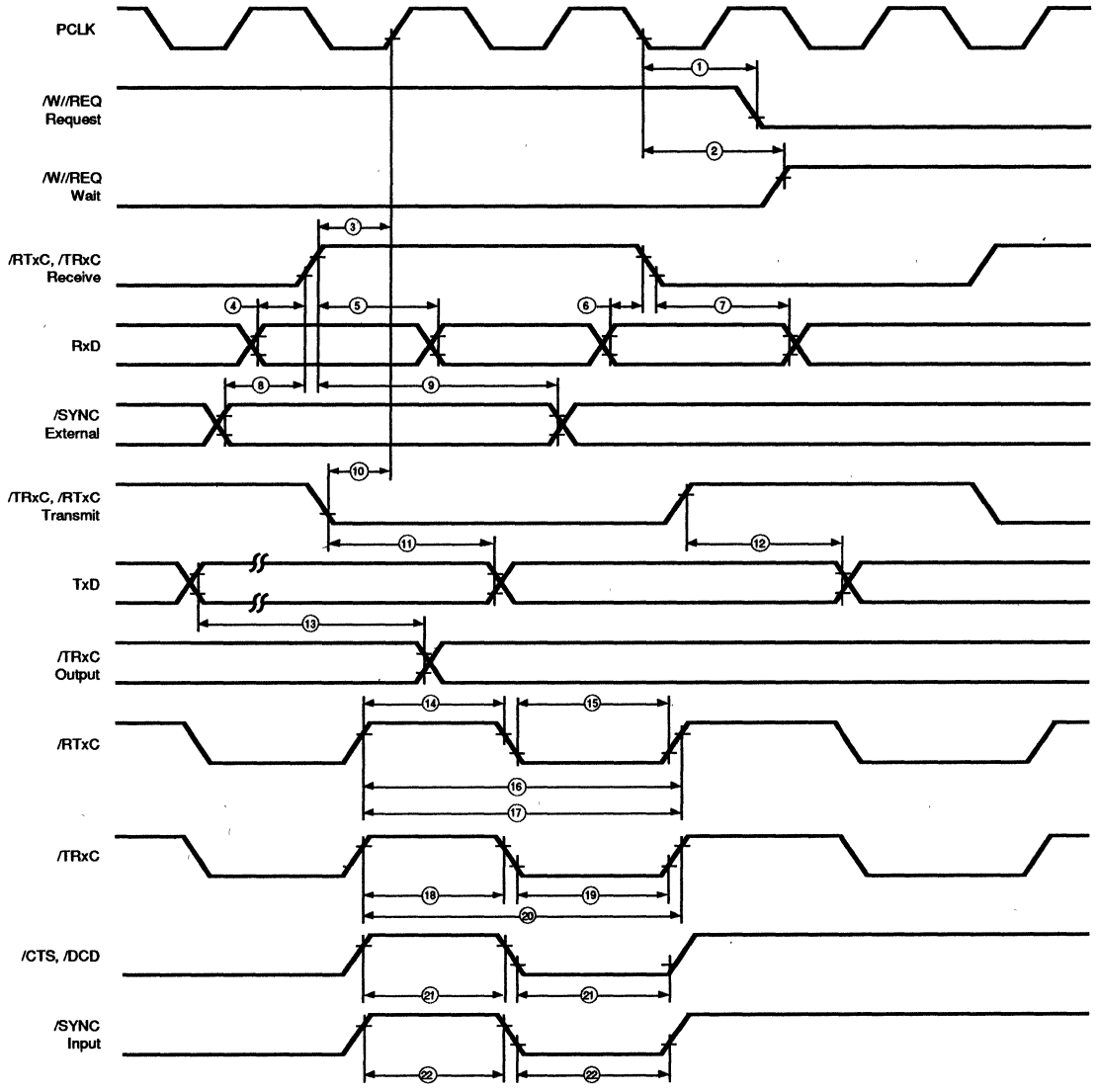


Figure 29. Z85230 General Timing Diagram

AC CHARACTERISTICS

Z85230 General Timing Table (Preliminary)

No	Symbol	Parameter	10 MHz Min	10 MHz Max	16 MHz Min	16 MHz Max	20 MHz Min	20 MHz Max	Notes
1	TdPC(REQ)	/PCLK to W/REQ Valid		200		80		70	
2	TdPC(W)	/PCLK to Wait Inactive		300		180		170	
3	TsRXC(PC)	/RxC to /PCLK Setup Time	NA	NA	NA	NA	NA	NA	[1,4]
4	TsRXD(RXCr)	RxD to /RxC Setup Time	0		0		0		[1]
5	ThRXD(RxCr)	RxD to /RXC Hold Time	125		50		45		[1]
6	TsRXD(RXCf)	RxD to /RXC Setup Time	0		0		0		[1,5]
7	ThRXD(RXCf)	RXD to /RXC Hold Time	125		50		45		[1,5]
8	TsSY(RXC)	/SYNC to /RxC Setup Time	-150		-100		-90		[1]
9	ThSY(RXC)	/SYNC to/RXC Hold Time	5TcPc		5TcPc		5TcPc		[1]
10	TsTXC(PC)	/TxC to /PCLK Setup Time	NA		NA		NA		[2,4]
11	TdTXCf(TXD)	/TxC to TxD Delay		150		80		70	[2]
12	TdTxCr(TXD)	/TxC to TxD Delay		150		80		70	[2,5]
13	TdTXD(TRX)	TxD to TRxC Delay		140		80		70	
14	TwRTXh	RTxC High Width	120		80		70		[6]
15	TwRTXl	TRxC Low Width	120		80		70		[6]
16a	TcRTX	RTxC Cycle Time	400		244		200		[6,7]
16b	TxRX(DPLL)	DPLL Cycle Time Min	50		31		31		[7,8]
17	TcRTXX	Crystal Osc. Period	100	1000	61	1000	61	1000	[3]
18	TwTRXh	TRxC High Width	120		80		70		[6]
19	TwTRXl	TRxC Low Width	120		80		70		[6]
20	TcTRX	TRxC Cycle Time	400		244		200		[6,7]
21	TwEXT	DCD or CTS Pulse Width	120		70		60		
22	TwSY	SYNC Pulse Width	120		70		60		

Notes:

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

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

[3] Both /RTxC and /SYNC have 30 pF capacitors to ground connected to them.

[4] Synchronization of RxC to PCLK is eliminated in divide by four operation.

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

[6] Parameter applies only for transmitter and receiver; DPLL and baud rate generator timing requirements are identical to case PCLK requirements.

[7] The maximum receive or transmit data rate is 1/4 PCLK.

[8] Applies to DPLL clock source only. Maximum data rate of 1/4 PCLK still applies. DPLL clock should have a 50% duty cycle.

AC CHARACTERISTICS

Z85230 System Timing Diagram (Preliminary)

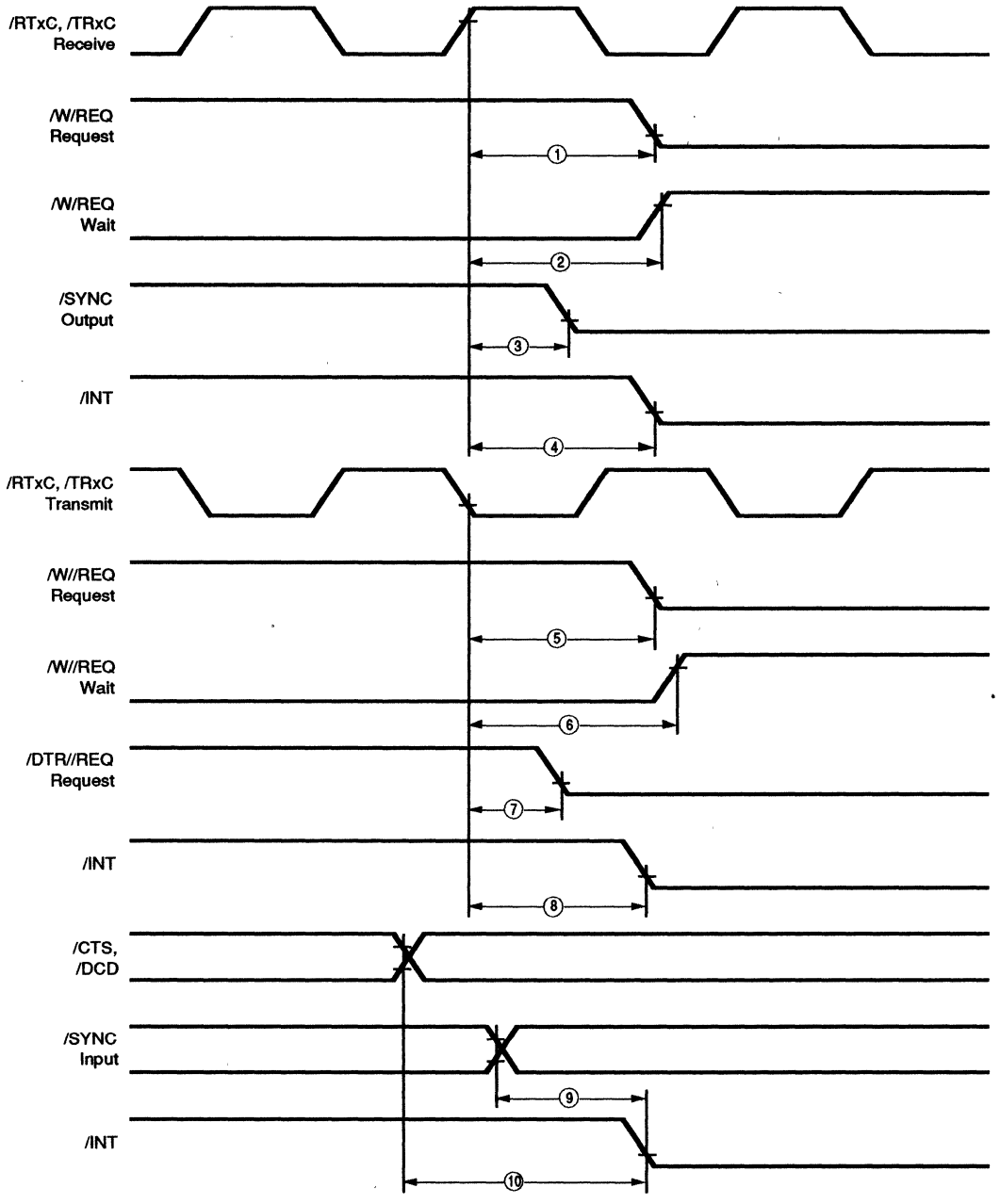


Figure 30. Z85230 System Timing

AC CHARACTERISTICS

Z85230 System Timing Table (Preliminary)

No	Symbol	Parameter	10 MHz Min	10 MHz Max	16 MHz Min	16 MHz Max	20 MHz Min	20 MHz Max	Notes
1	TdRXC(REQ)	/RXC to /W//REQ Valid	8	12	8	12	8	12	[2]
2	TdRXC(W)	/RxC to /Wait Inactive	8	12	8	14	8	14	[1,2]
3	TdRXC(SY)	/RxC to /SYNC Valid	4	7	4	7	4	7	[2]
4	TdRXC(INT)	/RxC to /INT Valid	10	16	10	16	10	16	[1,2]
5	TdTXC(REQ)	/TxC to /W//REQ Valid	5	8	5	8	5	8	[3]
6	TdTXC(W)	/TxC to /Wait Inactive	5	11	5	11	5	11	[1,3]
7	TdTXC(DRQ)	/TxC to /DTR//REQ Valid	4	7	4	7	4	7	[3]
8	TdTXC(INT)	/TxC to /INT Valid	6	10	6	10	6	10	[1,3]
9	TdSY(INT)	/SYNC to /INT Valid	2	6	2	6	2	6	[1]
10	TdEXT(INT)	/DCD or /CTS to /INT Valid	2	6	2	6	2	6	[1]

Notes:

[1] Open drain-output, measured with open-drain test load.

[2] /RxC is /RTxC or /TRxC, whichever is supplying the receive clock.

[3] /TxC is /TRxC or /RTxC, whichever is supplying the transmit clock.

PRODUCT SPECIFICATION

Z80C30/Z85C30

CMOS Z-BUS® SCC™ SERIAL COMMUNICATION CONTROLLER

FEATURES

- Low power CMOS
- Pin compatible to NMOS versions
- Two independent, 0 to 4.1 Mbit/second, full-duplex channels, each with a separate crystal oscillator, baud rate generator, and Digital Phase-Locked Loop (DPLL) for clock recovery.
- Multi-protocol operation under program control; programmable for NRZ, NRZI, or FM data encoding.
- Asynchronous mode with five to eight bits and one, one and one-half, or two stop bits per character, programmable clock factor, break detection and generation; parity, overrun, and framing error detection.
- Synchronous mode with internal or external character synchronization on one or two synchronous characters and CRC generation and checking with CRC-16 or CRC-CCITT preset to either 1s or 0s.
- SDLC/HDLC mode with comprehensive frame-level control, automatic zero insertion and deletion, I-field residue handling, abort generation and detection, CRC generation and checking, and SDLC Loop.
- Software Interrupt Acknowledge feature
- Local Loopback and Auto Echo modes
- Supports T1 digital trunk
- Enhanced DMA support
  10 x 19-bit status FIFO
  14-bit byte counter
- Fast speeds:
  10.0 MHz for data rates up to 2.5 Mbyte/sec.
  16.384 MHz for data rates up to 4.096 Mbyte/sec.

GENERAL DESCRIPTION

The Zilog Serial Communications Controller, Z80C30/Z85C30 SCC, is a pin and software compatible CMOS member of the SCC family introduced by Zilog in 1981. It is a dual channel, multi-protocol data communications peripheral that easily interfaces to CPU's with either multiplexed or non-multiplexed address/data buses. The advanced CMOS process offers lower power consumption, higher performance, and superior noise immunity. The programming flexibility of the internal registers allows the SCC to be configured to satisfy a wide variety of serial communications applications. The many on-chip features such as baud rate generators, digital phase locked loops, and crystal oscillators dramatically reduce the need for external logic. Additional features including a 10 x 19-bit status FIFO and 14-bit byte counter were added to support high speed SDLC transfers using DMA controllers.

The SCC handles asynchronous formats, synchronous byte-oriented protocols such as IBM Bisync, and synchronous bit-oriented protocols such as HDLC and IBM SDLC. This versatile device supports virtually any serial data transfer application (casette, diskette, tape drives, etc.)

The device can generate and check CRC codes in any synchronous mode and can be programmed to check data integrity in various modes. The SCC also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls can be used for general-purpose I/O.

The daisy-chain interrupt hierarchy is also supported as is standard for Zilog peripheral components.

Note: 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); /N//S (NORMAL and SYSTEM are both active Low).

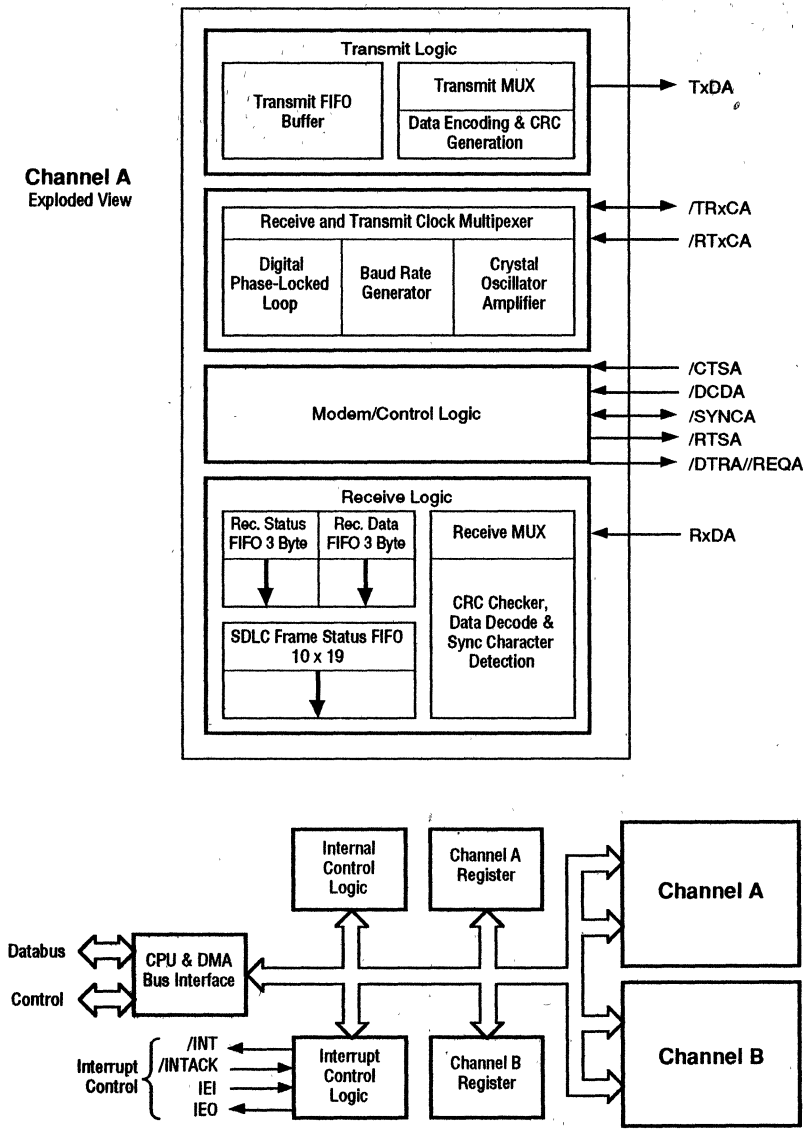


Figure 1. SCC Block Diagram

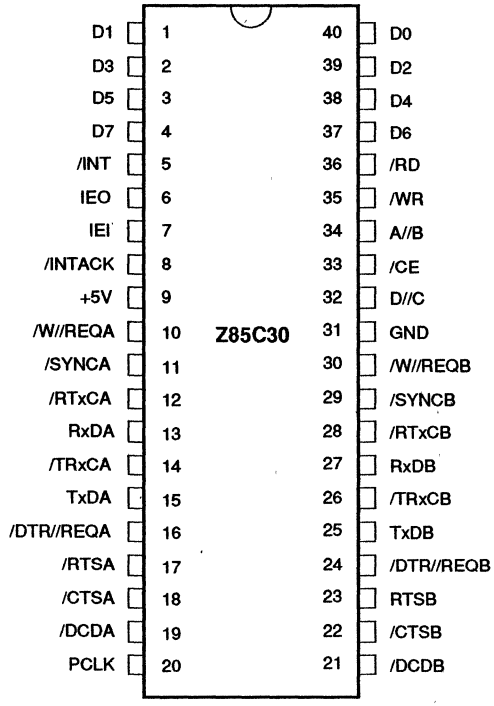


Figure 2. Z85C30 DIP Pin Assignments

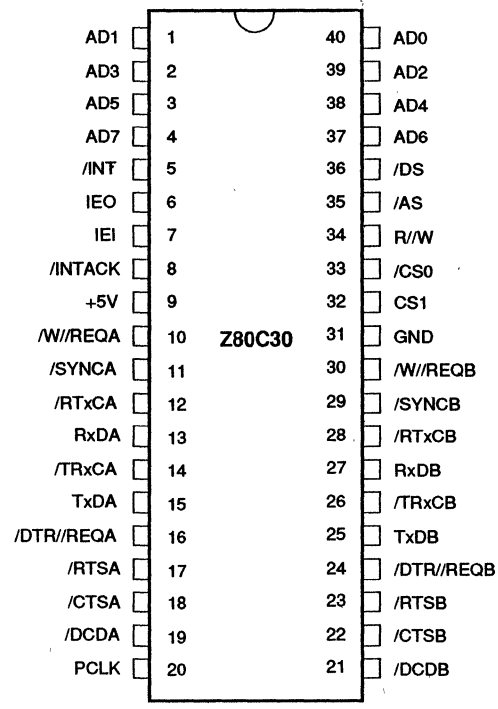


Figure 3. Z80C30 DIP Pin Assignments

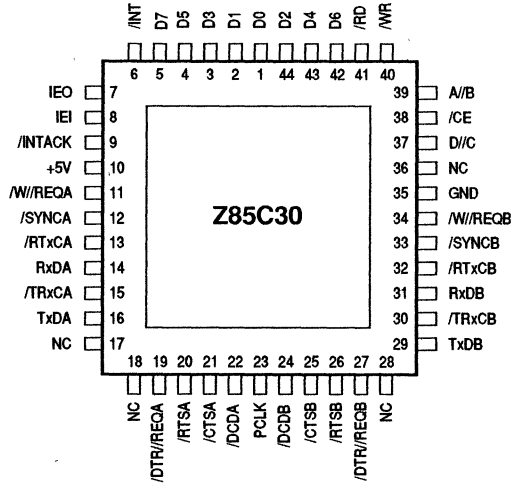


Figure 4. Z85C30 PLCC Pin Assignments

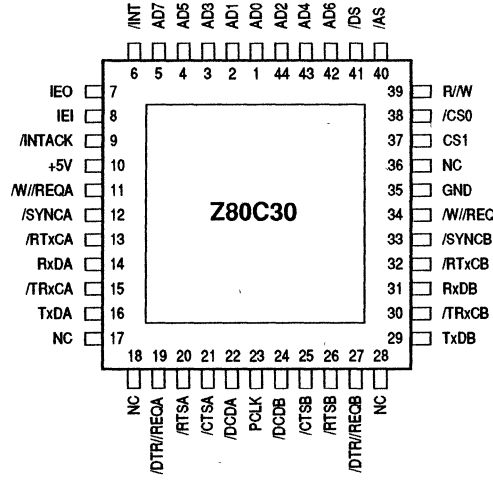


Figure 5. Z80C30 PLCC Pin Assignments

Note Power connections follow conventional descriptions below:

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

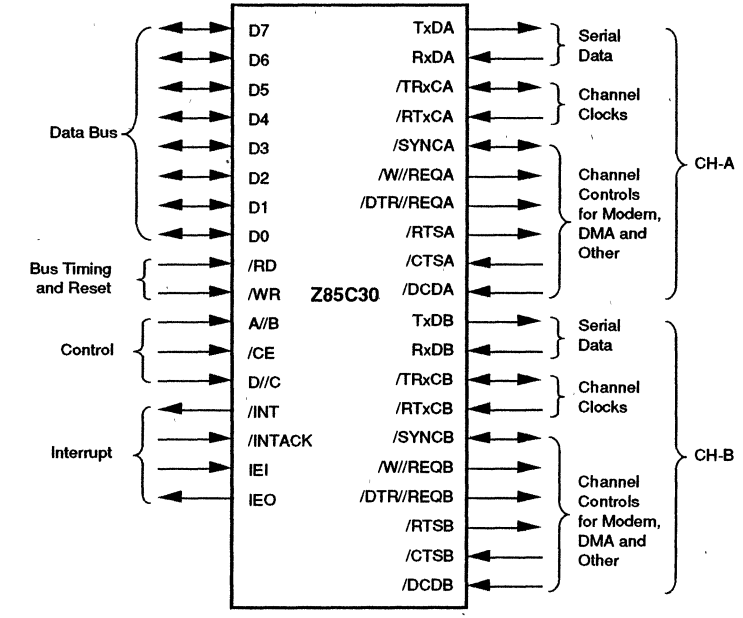


Figure 6. Z85C30 Pin Functions

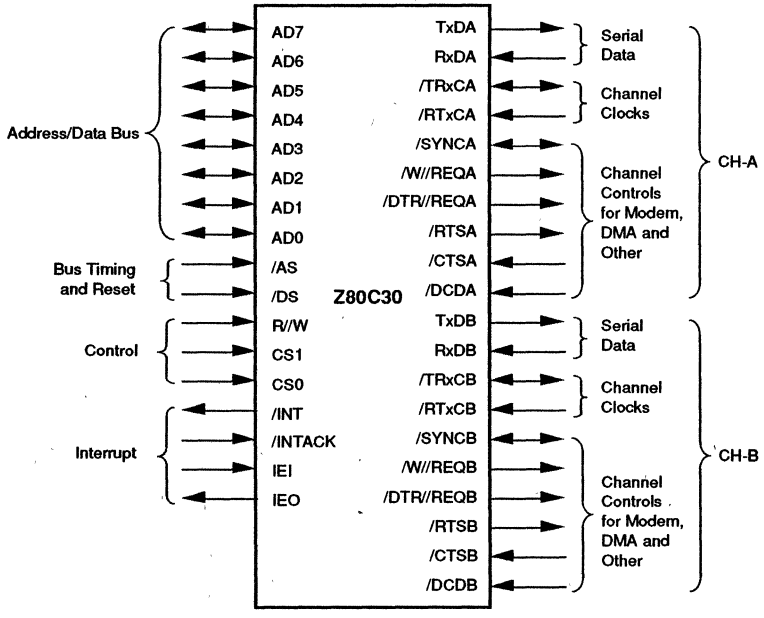


Figure 7. Z80C30 Pin Functions

PIN DESCRIPTION

The following section describes the pin functions common to the Z85C30 and the Z80C30. Figures 2 and 3 detail the respective pin functions and pin assignments.

/CTSA, /CTSB. *Clear To Send* (inputs, active Low). If these pins are programmed for Auto Enables, a Low on the inputs enables the respective transmitters. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The SCC defects pulses on these inputs and can interrupt the CPU on both logic level transitions.

/DCDA, /DCDB. *Data Carrier Detect* (inputs, active Low). These pins function as receiver enables if they are programmed for Auto Enables; otherwise, they are used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accommodate slow rise-time signals. The SCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.

/DTR//REQA, /DTR//REQB. *Data Terminal Ready/Request* (outputs, active Low). These outputs follow the state programmed into the DTR bit. They can also be used as general-purpose outputs or as Request lines for a DMA controller.

IEI. *Interrupt Enable In* (input, active High). IEI is used with IEO to form an interrupt daisy-chain when there is more than one interrupt driven device. A high IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.

IEO. *Interrupt Enable Out* (output, active High). IEO is High only if IEI is High and the CPU is not servicing the SCC interrupt or the SCC is not requesting an interrupt (Interrupt Acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.

/INT. *Interrupt Request* (output, open-drain, active Low). This signal is activated when the SCC requests an interrupt.

/INTACK. *Interrupt Acknowledge* (input, active Low) This signal indicates an active Interrupt Acknowledge cycle. During this cycle, the SCC interrupt daisy chain settles. When /RD or /DS becomes active, the SCC places an interrupt vector on the data bus (if IEI is High). /INTACK is latched by the rising edge of PCLK.

PCLK. *Clock* (input). This is the master SCC clock used to synchronize internal signals. PCLK is a TTL level signal. PCLK is not required to have any phase relationship with the master system clock.

RxDA, RxDB. *Receive Data* (inputs, active High). These signals receive serial data at standard TTL levels.

/RTxCA, /RTxCB. *Receive/Transmit Clocks* (inputs, active Low). These pins can be programmed in several different modes of operation. In each channel, /RTxC may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the Digital Phase-Locked Loop. These pins can also be programmed for use with the respective /SYNC pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in Asynchronous modes.

/RTSA, /RTSB. *Request To Send* (outputs, active Low). When the Request To Send (RTS) bit in Write Register 5 (Figure 11) is set, the /RTS signal goes Low. When the RTS bit is reset in the Asynchronous mode and Auto Enable is on, the signal goes High after the transmitter is empty. In Synchronous mode it strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.

/SYNCA, /SYNCB. *Synchronization* (inputs or outputs, active Low). These pins can act either as inputs, outputs, or part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to /CTS and /DCD In this mode, transitions on these lines affect the state of the Synchronous/Hunt status bits in Read Register 0 (Figure 10) but have no other function.

In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, /SYNC must be driven Low for two receive clock cycles after the last bit in the synchronous character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of /SYNC.

In the Internal Synchronization mode (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which synchronous characters are recognized. This synchronous condition is not latched, so these outputs are active each time a synchronization pattern is recognized (regardless of character boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.

TxDA, TxDB. *Transmit Data* (outputs, active High). These output signals transmit serial data at standard TTL levels.

PIN DESCRIPTION (Continued)

/TRxCA, /TRxCB. *Transmit/Receive Clocks* (inputs or outputs, active Low). These pins can be programmed in several different modes of operation. TRxC may supply the receive clock or the transmit clock in the input mode or supply the output of the Digital Phase-Locked Loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.

/W//REQA, /W//REQB. *Wait/Request* (outputs, open-drain when programmed for a Wait function, driven High or Low when programmed for a Request function). These dual-purpose outputs may be programmed as Request lines for a DMA controller or as Wait lines to synchronize the CPU to the SCC data rate. The reset state is Wait.

Z85C30

A//B. *Channel A/Channel B* (input) This signal selects the channel in which the read or write operation occurs.

/CE. *Chip Enable* (input, active Low). This signal selects the SCC for a read or write operation.

D7-D0. *Data Bus* (bidirectional, 3-state) These lines carry data and command to and from the SCC.

D//C. *Data/Control Select* (input). This signal defines the type of information transferred to or from the SCC. A High means data is transferred; a Low indicates a command.

/RD. *Read* (input, active Low). This signal indicates a read operation and when the SCC is selected, enables the SCC's bus drivers. During the Interrupt Acknowledge cycle, this signal gates the interrupt vector onto the bus if the SCC is the highest priority device requesting an interrupt.

/WR. *Write* (input, active Low). When the SCC is selected, this signal indicates a write operation. The coincidence of /RD and /WR is interpreted as a reset.

Z80C30

AD7-AD0. *Address/Data Bus* (bidirectional, active High, 3-state) These multiplexed lines carry register addresses to the SCC as well as data or control information.

/AS. *Address Strobe* (input, active Low). Addresses on AD7-AD0 are latched by the rising edge of this signal.

/CS0. *Chip Select 0* (input, active Low). This signal is latched concurrently with the addresses on AD7-AD0 and must be active for the intended bus transaction to occur.

/CS1. *Chip Select 1* (input, active High). This second select signal must also be active before the intended bus transaction can occur. CS1 must remain active throughout the transaction.

/DS. *Data strobe* (input, active Low) This signal provides timing for the transfer of data into and out of the SCC. If /AS and /DS coincide, this is interpreted as a reset.

R//W. *Read/Write* (input). This signal specifies whether the operation to be performed is a read or a write.

FUNCTIONAL DESCRIPTION

The architecture of the SCC is described from two points of view: as a datacommunications device which transmits and receives data in a wide variety of protocols; as a microprocessor peripheral in which the SCC offers valuable features such as vectored interrupts and DMA support.

The SCC's peripheral and datacommunication are described in the following sections. A block diagram is shown in Figure 1. The details of the communications between the receive and transmit logic to the system bus is shown in Figures 8 and 9. The features and data path for each of the SCC's A and B channels is identical. See the SCC Technical Manual for full details on using the SCC.

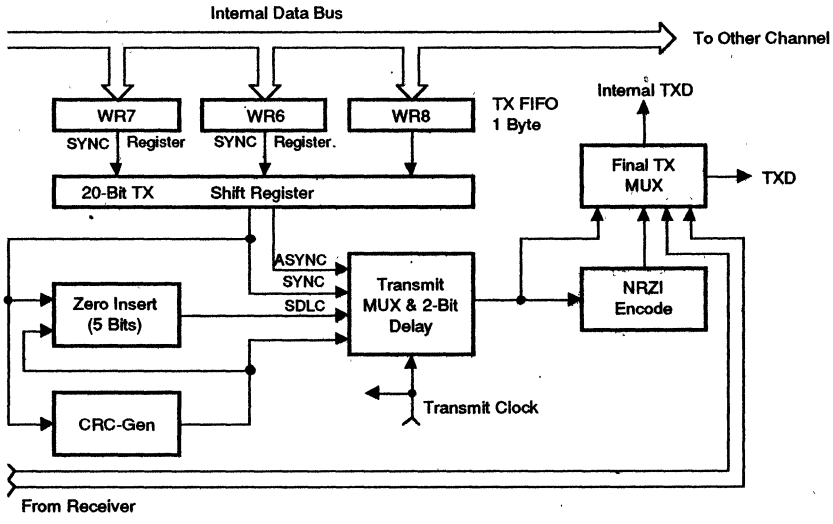


Figure 8. SCC Transmit Data Path

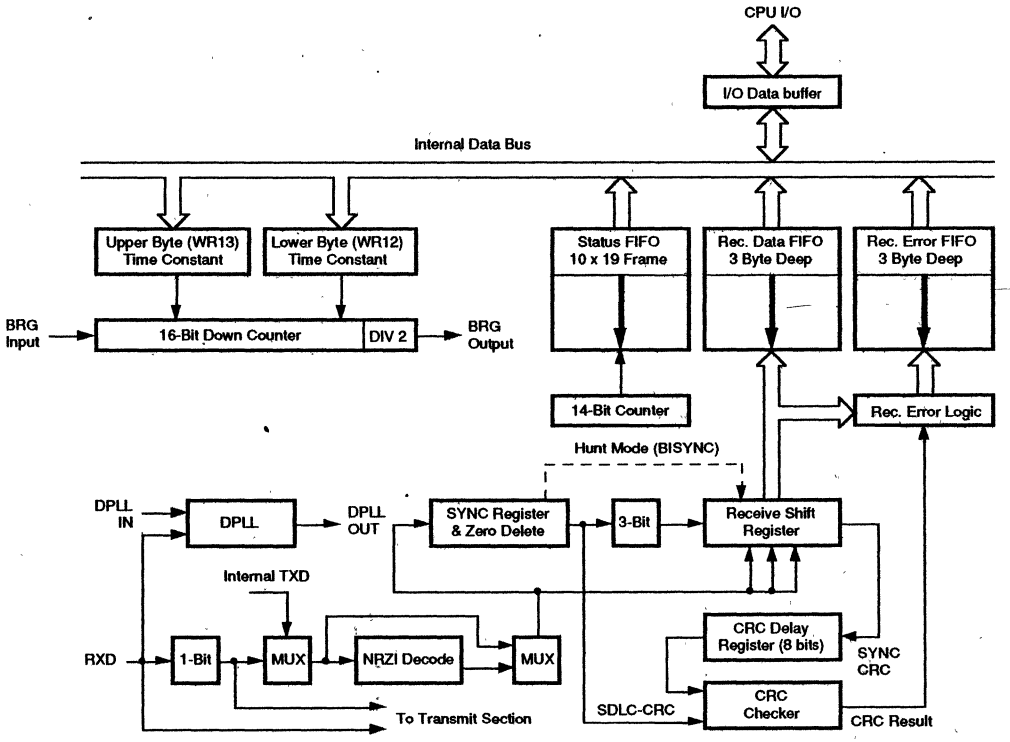


Figure 9. SCC Receive Data Path

I/O Interface Capabilities

System communication to and from the SCC is done through the SCC's register set. There are sixteen write registers and eight read registers. Table 1 lists all of the SCC's registers and a brief description of their functions. Throughout this document, the write and read registers are referenced with the following notation: "WR" for Write Register and "RR" for Read Register. For example:

WR4A	Write Register 4 for channel A
RR3	Read Register 3 for either/both channels

Table 1. SCC Read and Write Registers

	Read Register Functions
RR0	Transmit/Receive buffer status and External status
RR1	Special Receive Condition status
RR2	Modified interrupt vector (Channel B only) Unmodified interrupt vector (Channel A only)
RR3	Interrupt Pending bits (Channel A only)
RR8	Receive Buffer
RR10	Miscellaneous status
RR12	Lower byte of baud rate generator time constant
RR13	Upper byte of baud rate generator time constant
RR15	External/Status interrupt information

	Write Register Functions
WR0	CRC initialize, initialization commands for the various modes, Register Pointers.
WR1	Transmit/Receive interrupt and data transfer mode definition
WR2	Interrupt vector (accessed through either channel)
WR3	Receive parameters and control
WR4	Transmit/Receive miscellaneous parameters and modes
WR5	Transmit parameters and controls
WR6	Sync characters or SDLC address field
WR7	Sync character or SDLC flag
WR8	Transmit buffer
WR9	Master interrupt control and reset (accessed through either channel)
WR10	Miscellaneous transmitter/receiver control bits
WR11	Clock mode control
WR12	Lower byte of baud rate generator time constant
WR13	Upper byte of baud rate generator time constant
WR14	Miscellaneous control bits
WR15	External/Status interrupt control

There are three choices to move data into and out of the SCC: Polling, interrupt (vectored and non-vectored), and Block Transfer. The Block Transfer mode can be implemented under CPU or DMA control.

Polling

When polling, all interrupts are disabled. Three status registers in the SCC are automatically updated whenever any function is performed. For example, End-Of-Frame in SDLC mode sets a bit in one of these status registers. The purpose of polling is for the CPU to periodically read a status register until the register contents indicate the need for data to be transferred. Only one register needs to be read; depending on its contents, the CPU either writes data, reads data, or continues. Two bits in the register indicate the need for data transfer. An alternative is a poll of the Interrupt Pending register to determine the source of an interrupt. The status for both channels resides in one register.

Interrupts

The SCC's interrupt structure supports vectored and nested interrupts. Nested interrupts are supported with the interrupt acknowledge feature (/INTACK pin) of the SCC. This allows the CPU to recognize the occurrence of an interrupt, and re-enable higher priority interrupts. Because an INTACK cycle will release the /INT pin from the active state, a higher priority SCC interrupt or another higher priority device can interrupt the CPU. When an SCC responds to an Interrupt Acknowledge signal (INTACK) from the CPU, an interrupt vector can be placed on the data bus. This vector is written in WR2 and may be read in RR2. To speed interrupt response time, the SCC can modify three bits in this vector to indicate status. If the vector is read in Channel A, status is never included; if it is read in Channel B, status is always included.

Each of the six sources of interrupts in the SCC (Transmit, Receive, and External/Status interrupts in both channels) has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). Operation of the IE bit is straightforward. If the IE bit is set for a given interrupt source, then that source can request interrupts. The exception is when the MIE (Master Interrupt Enable) bit in WR9 is reset and no interrupts can be requested. The IE bits are write only.

The other two bits are related to the interrupt priority chain (Figure 10). As a microprocessor peripheral, the SCC may request an interrupt only when no higher priority device is requesting one, e.g., when IEI is High. If the device in question requests an interrupt, it pulls down /INT. The CPU then responds with /INTACK, and the interrupting device places the vector on the data bus.

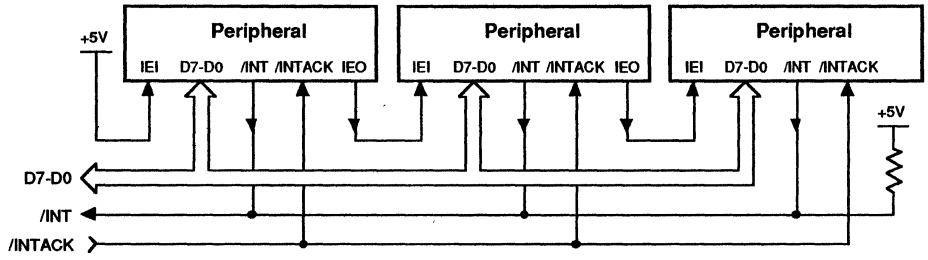


Figure 10. SCC Interrupt Priority Schedule

The SCC can also execute an interrupt acknowledge cycle through software. In some CPU environments it is difficult to create the /INTACK signal with the necessary timing to acknowledge interrupts and allow the nesting of interrupts. In these cases, the /INTACK signal can be created with a software command to the SCC.

In the SCC, the Interrupt Pending (IP) bit signals a need for interrupt servicing. When an IP bit is 1 and the IEI input is High, the /INT output is pulled Low, requesting an interrupt In the SCC, if the IE bit isn't set by enabling interrupts, then the IP for that source is never set. The IP bits are readable in RR3A.

The IUS bits signal that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority in the SCC and external to the SCC 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 SCC 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 three types of interrupts: Transmit, Receive, and External/Status. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receiver, Transmit, and External/Status interrupts prioritized in that order within each channel.

When enabled, the receiver can interrupt the CPU in one of three ways:

1. Interrupt on First Receive Character or Special Receive Condition.

2. Interrupt on All Receive Characters or Special Receive Conditions.

3. Interrupt on Special Receive Conditions Only.

Interrupt on First Character or Special Condition and Interrupt on Special Condition Only are typically used with the Block Transfer mode. A special Receive Condition is one of the following: receiver overrun, framing error in Asynchronous mode, end-of-frame in SDLC mode and, optionally, a parity error. The Special Receive Condition interrupt is different from an ordinary receive character available interrupt only by the status placed in the vector during the Interrupt Acknowledge cycle. In Interrupt on First Receive Character, an interrupt occurs from Special Receive Conditions any time after the first receive character interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the /CTS, /DCD, and /SYNC pins, however, an External/Status interrupt is also caused by a Transmit Underrun condition; a zero count in the baud rate generator; by the detection of a Break (Asynchronous mode), Abort (SDLC mode) or EOP (SDLC Loop mode) sequence in the data stream. The interrupt caused by the Abort or EOP has a special feature allowing the SCC to interrupt when the Abort or EOP sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Abort condition in external logic in SDLC mode. In SDLC Loop mode, this feature allows secondary stations to recognize the primary station wishes to regain control of the loop during a poll sequence.

Software Interrupt Acknowledge

The SCC interrupt acknowledge cycle can be initiated through software. If Write Register 9 (WR9) bit D5 is set, Read Register 2 (RR2) results in an interrupt acknowledge cycle to be executed internally. Like a hardware INTACK cycle, a software acknowledge causes the INT pin to return high, the IEO pin to go low and set the IUS latch for the highest priority interrupt pending.

Similar to when the hardware INTACK signal can be used, a software acknowledge cycle requires that a Reset Highest IUS command be issued in the interrupt service routine. Whenever an interrupt acknowledge cycle is used, hardware or software, a reset highest IUS command is required. If RR2 is read from channel A, the unmodified vector is returned. If RR2 is read from channel B, then the vector is modified to indicate the source of the interrupt. The Vector Includes Status (VIS) and No Vector (NV) bits in WR9 are ignored when bit D5 is set to 1.

When the INTACK and IEI pins are not being used, they should be pulled up to V_{cc} through a resistor (10 kohm typical).

CPU/DMA Block Transfer. The SCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers. The Block Transfer mode used the /WAIT//REQUEST output in conjunction with the Wait/Request bits in WR1. The /WAIT//REQUEST output can be defined under software control as a WAIT line in the CPU Block Transfer mode or as a REQUEST line in the DMA Block Transfer mode.

To a DMA controller, the SCC REQUEST output indicates that the SCC is ready to transfer data to or from memory. To the CPU, the WAIT line indicates that the ESCC is not ready to transfer data, thereby requesting that the CPU extend the I/O cycle. The /DTR//REQUEST line allows full-duplex operation under DMA control.

SCC Data Communications Capabilities

The SCC provides two independent full-duplex programmable channels for use in any common asynchronous or synchronous data communication protocols (Figure 11). Each of the datacommunication channels has identical features and capabilities.

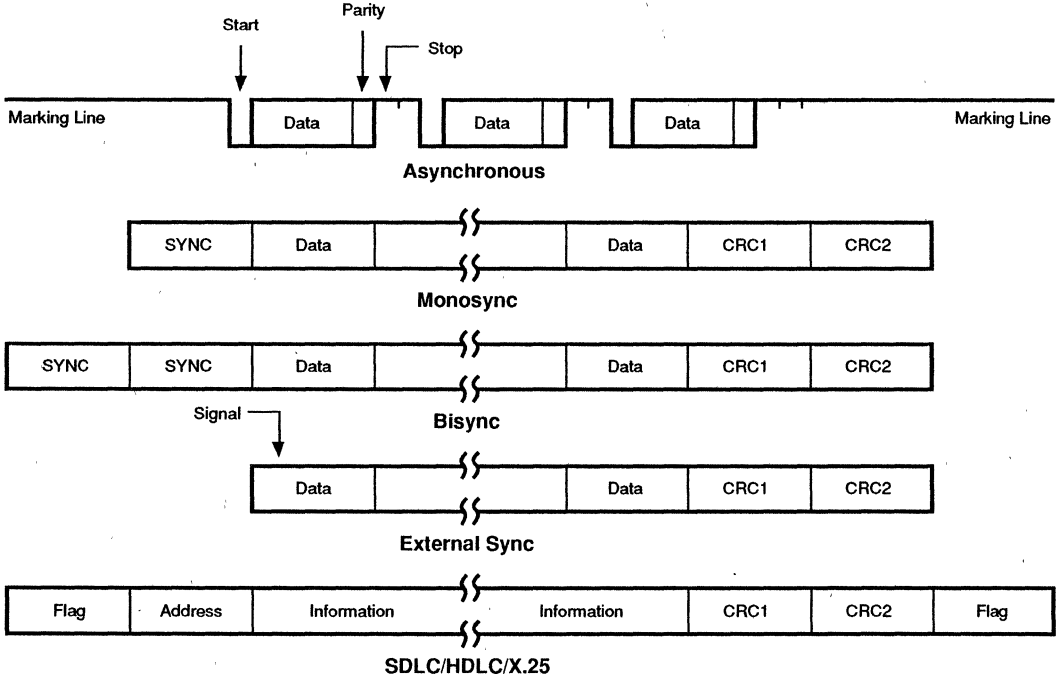


Figure 11. Some SCC Protocols

Asynchronous Modes

Send and Receive is accomplished independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one-and-a-half, or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break. Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxDA or RxDB pins). If the Low does not persist (e.g., a transient), the character assembly process does not start.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occur. Vectored interrupts allow fast servicing or error conditions using dedicated routines. Furthermore, a built-in checking process avoids the interpretation of a framing error as a new start bit: a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit begins.

The SCC does not require symmetric transmit and receive clock signals - a feature allowing use of the wide variety of clock sources. The transmitter and receiver handle data at a rate supplied to the receive and transmit clock inputs. In Asynchronous modes, the SYNC pin may be programmed as an input used for functions such as monitoring a ring indicator.

Synchronous Modes

The SCC supports both byte-oriented and bit-oriented synchronous communication. Synchronous byte-oriented protocols are handled in several modes. They allow character synchronization with a 6-bit or 8-bit sync character (Monosync), and a 12-bit or 16-bit synchronization pattern (Bisync), or with an external sync signal. Leading sync characters are removed without interrupting the CPU.

Five or 7-bit synchronous characters are detected with 8- or 16-bit patterns in the SCC by overlapping the larger pattern across multiple incoming synchronous characters as shown in Figure 12.

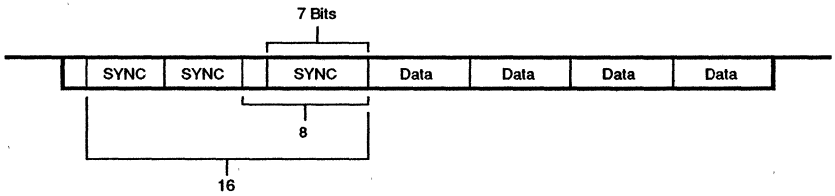


Figure 12. Detecting 5- or 7-Bit Synchronous Characters

CRC checking for Synchronous byte oriented modes is delayed by one character time so that the CPU may disable CRC checking on specific characters. This permits the implementation of protocols such as IBM Bisync.

Both CRC-16 (X16 + X15 + X2 + 1) and CCITT (X16 + X12 + X5 + 1) error checking polynomials are supported. Either polynomial may be selected in all Synchronous modes. Users may preset the CRC generator and checker to all 1's or all 0's. The SCC also provides a feature that automatically transmits CRC data when no other data is available for transmission. This allows for high speed transmissions under DMA control, with no need for CPU intervention at the end of a message. When there is no data or CRC to send in Synchronous modes, the transmitter inserts 6-,8-, or 16-bit sync characters, regardless of the programmed character length.

SDLC Mode

The SCC supports Synchronous bit-oriented protocols, such as SDLC and HDLC, by performing automatic flag sending, zero insertion, and CRC generation. A special command is used to abort a frame in transmission. At the end of a message, the SCC automatically transmits the CRC and trailing flag when the transmitter underruns. The transmitter may also be programmed to send an idle line consisting of continuous flag characters or a steady marking condition.

If a transmit underrun occurs in the middle of a message, an external/status interrupt warns the CPU of this status change so that an abort can be issued. The SCC may also be programmed to send an abort itself in case of an underrun, relieving the CPU of this task. One to eight bits per character can be sent, allowing reception of a message with no prior information about the character structure in the information field of a frame.

The receiver automatically acquires synchronization on the leading flag of a frame in SDLC or HDLC and provides a synchronization signal on the /SYNC pin (an interrupt can also be programmed). The receiver can be programmed to search for frames addressed by a single byte (or four bits within a byte) of a user-selected address or to a global broadcast address. In this mode, frames not matching either the user-selected or broadcast address are ignored.

The number of address bytes are extended under software control. For receiving data, an interrupt on the first received character, or an interrupt on every character, or on special condition only (end-of-frame) can be selected. The receiver automatically deletes all 0's inserted by the transmitter during character assembly. CRC is also calculated and is automatically checked to validate frame transmission. At the end of transmission, the status of a received frame is available in the status registers. In SDLC mode, the SCC must be programmed to use the SDLC CRC polynomial, but the generator and checker may be preset to all 1's or all 0's. The CRC is inverted before transmission and the receiver checks against the bit pattern 0001110100001111.

NRZ, NRZI or FM coding may be used in any 1x mode. The parity options available in Asynchronous modes are available in Synchronous modes.

SDLC Loop Mode. The SCC supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, there is a primary controller station that manages the message traffic flow on the loop and any number of secondary stations. In SDLC Loop mode, the SCC performs the functions of a secondary station while an SCC operating in regular SDLC mode acts as a controller (Figure 13). SDLC loop mode can be selected by setting WR10 bit D1.

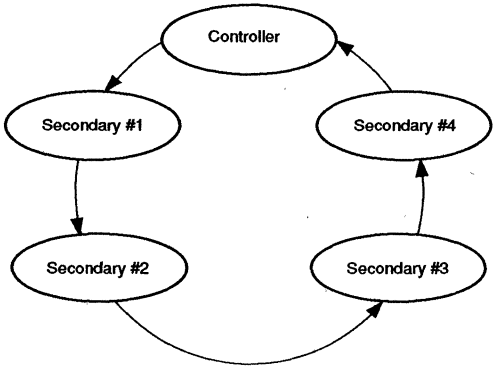


Figure 13. An SDLC Loop

A secondary station in an SDLC Loop is always listening to the messages being sent around the loop and, in fact, passes these messages to the rest of the loop by retransmitting them with a one-bit-time delay. The secondary station places its own message on the loop only at specific times. The controller signals that secondary stations can transmit messages by sending a special character, called an EOP (End Of Poll), around the loop. The EOP character is the bit pattern 11111110. Because of zero insertion during messages, this bit pattern is unique and easily recognized.

When a secondary station has a message to transmit and recognizes an EOP on the line, it changes the last binary 1 of the EOP to a 0 before transmission. This has the effect of turning the EOP into a flag sequence. The secondary station now places its message on the loop and terminates the message with an EOP. Any secondary stations further down the loop with messages to transmit appends their messages to the message of the first secondary station by the same process. Any secondary stations without messages to send merely echo the incoming message and are prohibited from placing messages on the loop (except upon recognizing an EOP). In SDLC Loop mode, NRZ, NRZI, and FM coding may all be used.

The SCC's ability to receive high speed back-to-back SDLC frames is maximized by a 10- deep by 19-bit wide status FIFO. When enabled (through WR15, bit D2), it provides the DMA the ability to continue to transfer data into memory so that the CPU can examine the message later. For each SDLC frame, a 14-bit byte count and 5 status/error bits are stored. The byte count and status bits are accessed through Read Registers 6 and 7. Read Registers 6 and 7 are only accessible when the SDLC FIFO is enabled. The 10x19 status FIFO is separate from the 3-byte receive data FIFO.

Baud Rate Generator

Each channel in the SCC contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On startup, the flip-flop on the output is set in a High state, the value in the time constant register is loaded into the counter, and the counter starts counting down. The output of the baud rate generator toggles upon reaching 0, the value in the time constant register is loaded into the counter, and the process is repeated. The time constant may be changed at any time, but the new value does not take effect until the next load of the counter.

The output of the baud rate generator may be used as either the transmit clock, the receive clock, or both. It can also drive the Digital Phase-Locked Loop (see next section).

If the receive clock or transmit clock is not programmed to come from the TRxC pin, the output of the baud rate generator may be echoed out via the TRxC pin.

The following formula relates the time constant to the baud rate where PCLK or RTxC is the baud rate generator input frequency in Hertz. The clock mode is 1, 16, 32, or 64, as selected in Write Register 4, bits D6 and D7. Synchronous operation modes should select 1 and Asynchronous should select 16, 32 or 64.

$$\text{Time Constant} = \frac{\text{PCLK or RTxC Frequency}}{2(\text{Baud Rate})(\text{Clock Rate})} - 2$$

Digital Phase-Locked Loop

The SCC contains a Digital Phase-Locked Loop (DPLL) to recover clock information from a data stream with NRZI or FM encoding. The DPLL is driven by a clock that is nominally 32 (NRZI) or 16 (FM) times the data rate The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock is then used as the SCC receive clock, the transmit clock, or both. When the DPLL is selected as the transmit clock source, it will provide a jitter free clock output that is the DPLL input frequency divided by the appropriate divisor for the selected encoding technique.

For NRZI encoding, the DPLL counts the 32x clock to create nominal bit times. As the 32x clock is counted, the DPLL is searching the incoming data stream for edges (either 1 to 0, or 0 to 1). Whenever an edge is detected, the DPLL makes a count adjustment (during the next counting cycle), producing a terminal count closer to the center of the bit cell.

For FM encoding, the DPLL still counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream should occur between counts 15 and 16 and between counts 31 and 0 The DPLL looks for edges only during a time centered on the 15 to 16 counting transition.

The 32x clock for the DPLL can be programmed to come from either the RTxC input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the SCC via the TRxC pin (if this pin is not being used as an input).

Data Encoding

The SCC may be programmed to encode and decode the serial data in four different ways (Figure 14). In NRZ encoding, a 1 is represented by a High level and a 0 is represented by a Low level. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level. In FM1 (more properly, bi-phase mark), a transition occurs at the beginning of every bit cell A 1 is represented by an additional transition at the center of the bit cell and a 0 is represented by no additional transition at the center of the bit cell. In FM0 (bi-phase space), a transition occurs at the beginning of every bit cell. A 0 is represented by an additional transition at the center of the bit cell, and a 1 is represented by no additional transition at the center of the bit cell. In addition to these four methods, the SCC can be used to decode Manchester (bi-phase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0 to 1, the bit is a 0. If the transition is 1 to 0, the bit is a 1.

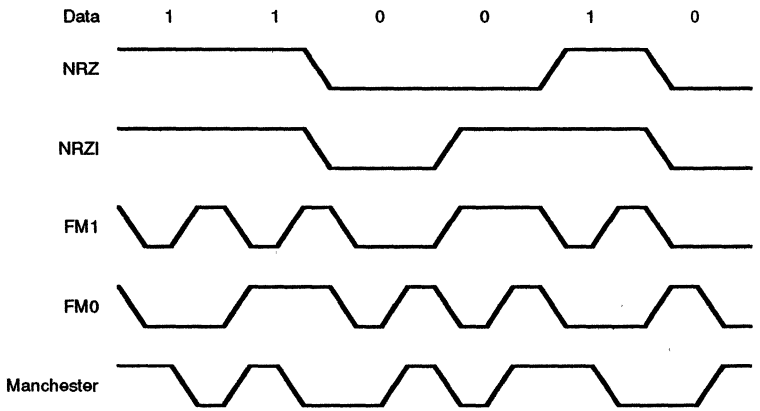


Figure 14. Data Encoding Methods

Auto Echo and Local Loopback
The SCC is capable of automatically echoing everything it receives. This feature is useful mainly in Asynchronous modes, but works in Synchronous and SDLC modes as well. Auto Echo mode (TxD is RxD) is used with NRZI or FM encoding with no additional delay because the data stream is not decoded before retransmission. In Auto Echo mode, the /CTS input is ignored as a transmitter enable (although transitions on this input can still cause interrupts if programmed to do so). In this mode, the transmitter is actually bypassed and the programmer is responsible for disabling transmitter interrupts and /WAIT//REQUEST on transmit.

The SCC is also capable of local loopback. In this mode, TxD or RxD is just like Auto Echo mode. However, in Local Loopback mode the internal transmit data is tied to the internal receive data and RxD is ignored (except to be echoed out via TxD). The /CTS and /DCD inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local Loopback works in Asynchronous, Synchronous and SDLC modes with NRZ, NRZI or FM coding of the data stream.

SDLC FIFO Frame Status FIFO Enhancement
The SCC's ability to receive high speed back-to-back SDLC frames is maximized by a 10- deep by 19-bit wide status FIFO. When enabled (through WR15, bit D2), it provides the DMA the ability to continue to transfer data into memory so that the CPU can examine the message later. For each SDLC frame, a 14-bit byte count and 5 status/error bits are stored. The byte count and status bits are accessed through Read Registers 6 and 7. Read Registers 6 and 7 are only accessible when the SDLC FIFO is enabled. The 10x19 status FIFO is separate from the 3-byte receive data FIFO.

When the enhancement is enabled, the status in read register 1 (RR1) and byte count for the SDLC frame are stored in the 10 x 19 bit status FIFO. This allows the DMA controller to transfer the next frame into memory while the CPU verifies that the message was properly received.

Summarizing the operation; data is received, assembled, and loaded into the eight byte FIFO before being transferred to memory by the DMA controller. When a flag is received at the end of an SDLC frame, the frame byte count from the 14-bit counter and five status bits are loaded into the status FIFO for verification by the CPU. The CRC checker is automatically reset in preparation for the next frame which can begin immediately. Since the byte count and status are saved for each frame, the message integrity is verified at a later time. Status information for up to 10 frames is stored before a status FIFO overrun can occur.

If a frame is terminated with an ABORT, the byte count is loaded to the status FIFO and the counter reset for the next frame.

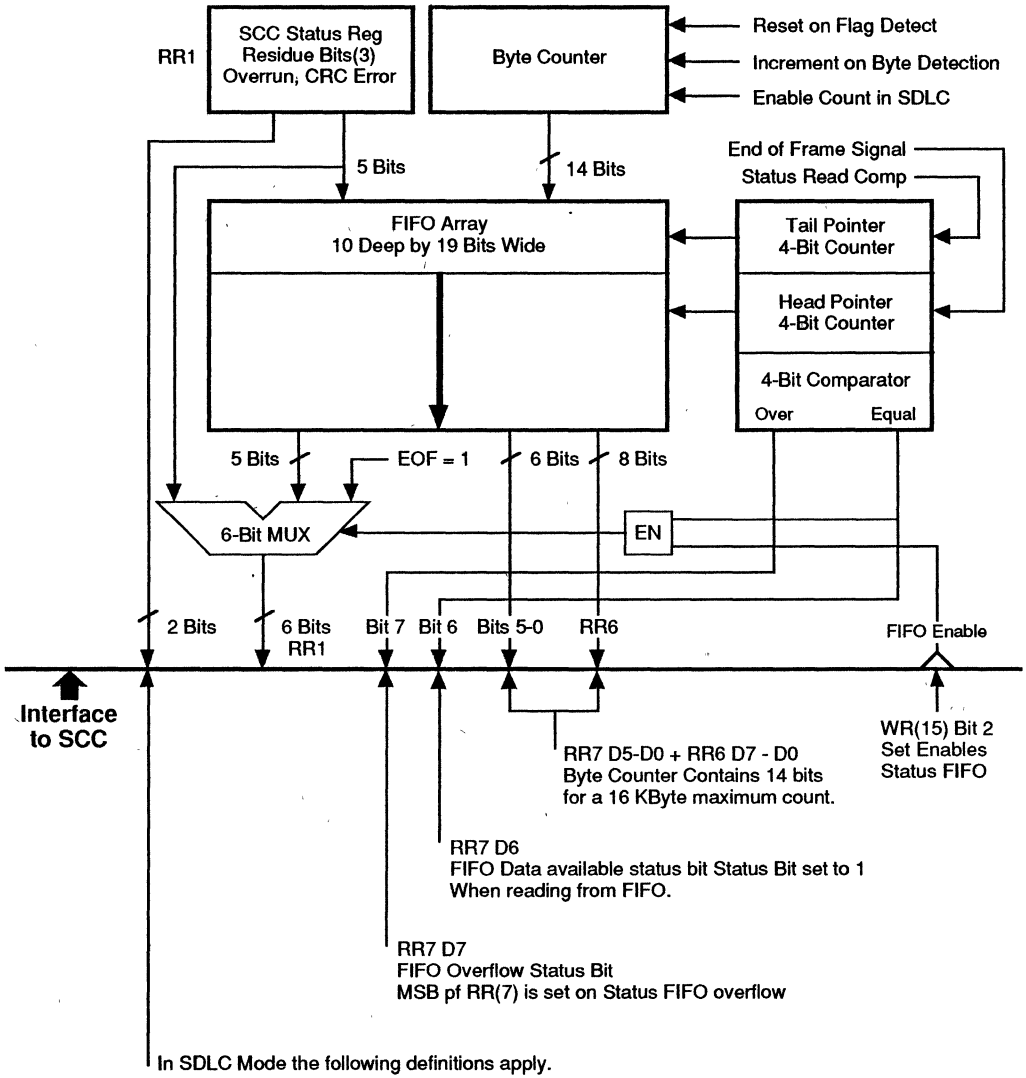
FIFO Detail
For a better understanding of details of the FIFO operation, refer to the block diagram in Figure 15.

Enable/Disable
This FIFO is implemented so that it is enabled when WR15, bit D2, is set and the SCC is in the SDLC/HDLC mode. Otherwise, the status register contents bypass the FIFO and go directly to the bus interface (the FIFO pointer logic is reset either when disabled or via a channel or power-on reset). When the FIFO mode is disabled, the SCC is completely downward compatible with the NMOS Z8530. The FIFO mode is disabled on power-up (WR15 D2 is set to 0 on reset). The effects of backward compatibility on the register set are that RR4 is an image of RR0, RR5 is an image of RR1, RR6 is an image of RR2 and RR7 is an image of RR3. For the details of the added registers, refer to Figure 18. The status of the FIFO Enable signal is obtained by reading RR15, bit D2. If the FIFO is enabled, the bit will be set to 1; otherwise, it will be reset.

Read Operation
When WR15 bit D2 is set and the FIFO is not empty, the next read to status register RR1 or the additional registers RR7 and RR6, are from the FIFO. Reading status register RR1 causes one location of the FIFO to be emptied, so status is read after reading the byte count, otherwise the count is incorrect. Before the FIFO underflows, it is disabled. In this case, the multiplexer is switched to allow status to read directly from the status register and reads from RR7 and RR6 contain bits that are undefined. Bit D6 of RR7 (FIFO Data Available) is used to determine if status data is coming from the FIFO or directly from the status register, since it is set to 1 whenever the FIFO is not empty.

Since not all status bits are stored in the FIFO, the All Sent, Parity, and EOF bits bypass the FIFO. The status bits sent through the FIFO are Residue Bits (3), Overrun, and CRC Error.

Frame Status FIFO Circuitry



In SDLC Mode the following definitions apply.

- All Sent bypasses MUX and equals contents of SCC Status Register.
- Parity Bits bypasses MUX and does the same.
- EOF is set to 1 whenever reading from the FIFO.

Figure 15. SDLC Frame Status FIFO

FUNCTIONAL DESCRIPTION (Continued)

The sequence for proper operation of the byte count and FIFO logic is to read the registers in the following order: RR7, RR6, and RR1 (reading RR6 is optional). Additional logic prevents the FIFO from being emptied by multiple reads from RR1. The read from RR7 latches the FIFO empty/full status bit (D6) and steers the status multiplexer to read from the SCC megacell instead of the status FIFO (since the status FIFO is empty). The read from RR1 allows an entry to be read from the FIFO (if the FIFO was empty, logic was added to prevent a FIFO underflow condition).

Write Operation

When the end of an SDLC frame (EOF) has been received and the FIFO is enabled, the contents of the status and byte-count registers are loaded into the FIFO. The EOF signal is used to increment the FIFO. If the FIFO overflows, the RR7 bit D7 (FIFO Overflow) is set to indicate the overflow. This bit and the FIFO control logic is reset by disabling and re-enabling the FIFO control bit (WR15 bit D2). For details of FIFO control timing during an SDLC frame, refer to Figure 16.

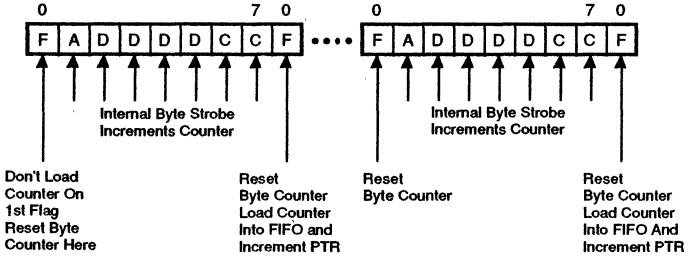


Figure 16. SDLC Byte Counting Detail

PROGRAMMING

The SCC contains write registers in each channel that are programmed by the system separately to configure the functional personality of the channels.

Z85C30

In the SCC, the data registers are directly addressed by selecting a High on the D//C pin. With all other registers (except WR0 and RR0), programming the write registers requires two write operations and reading the read registers requires both a write and a read operation. The first write is to WR0 and contains three bits that point to the selected register. The second write is the actual control word for the selected register, and if the second operation is read, the selected read register is accessed. All of the SCC registers, including the data registers, may be accessed in this fashion. The pointer bits are automatically cleared after the read or write operation so that WR0 (or RR0) is addressed again.

Z80C30

All SCC registers are directly addressable. How the SCC decodes the address placed on the address/data bus at the beginning of a Read or Write cycle is controlled by a command issued in WR0B. In the Shift Right mode the channel select A/B is taken from AD0 and the state of AD5 is ignored. In the Shift Left mode the channel select A/B is taken from AD5 and the state of AD0 is ignored AD7 and AD6 are always ignored as address bits and the register address itself occupies AD4-AD1.

Z85C30/Z80C30 Setup

Initialization. The system program first issues a series of commands to initialize the basic mode of operation. This is followed by other commands to qualify conditions within the selected mode. For example, in the Asynchronous mode, character length, clock rate, number of stop bits, and even or odd parity should be set first. Then the interrupt mode is set, and finally, the receiver and transmitter are enabled.

Write Registers. The SCC contains 15 write registers (16 counting the transmit buffer) in each channel. These write registers are programmed separately to configure the functional "personality" of the channels. There are two registers (WR2 and WR9) shared by the two channels that are accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits and reset commands. Figure 17 shows the format of each write register.

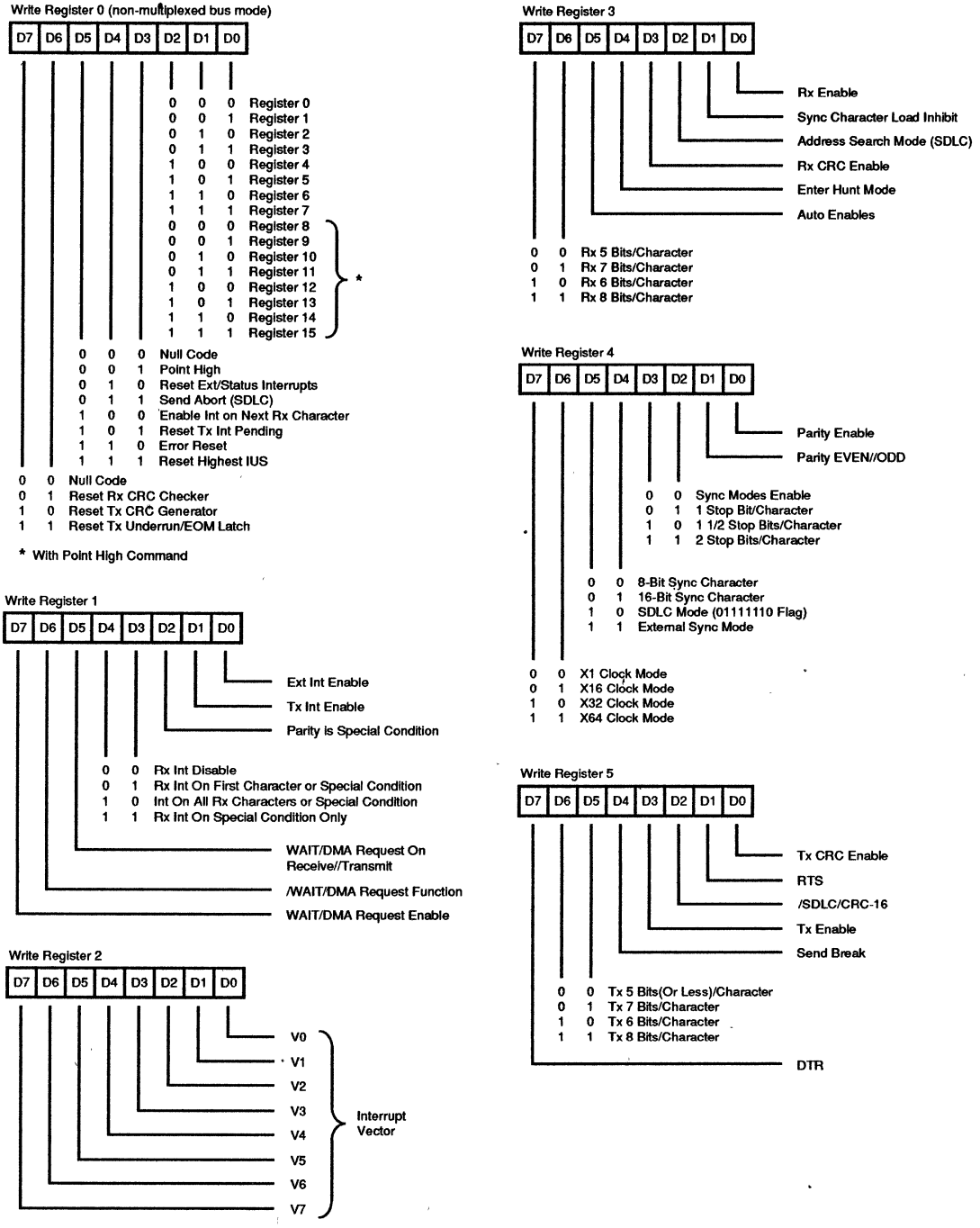


Figure 17. Write Register Bit Functions

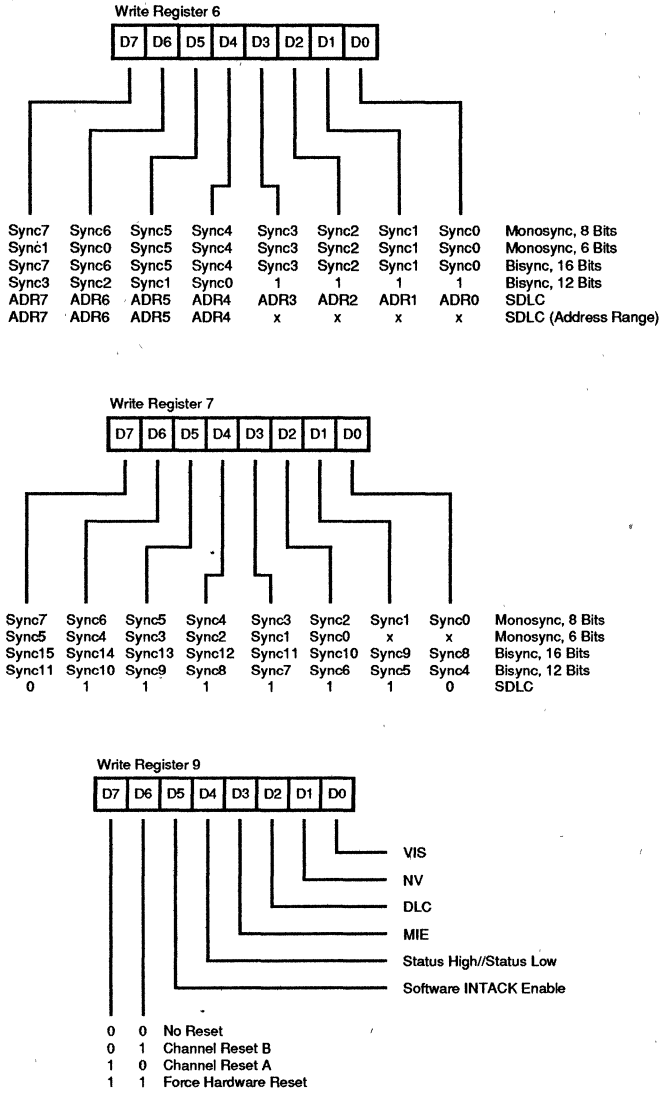


Figure 17. Write Register Bit Functions (Continued)

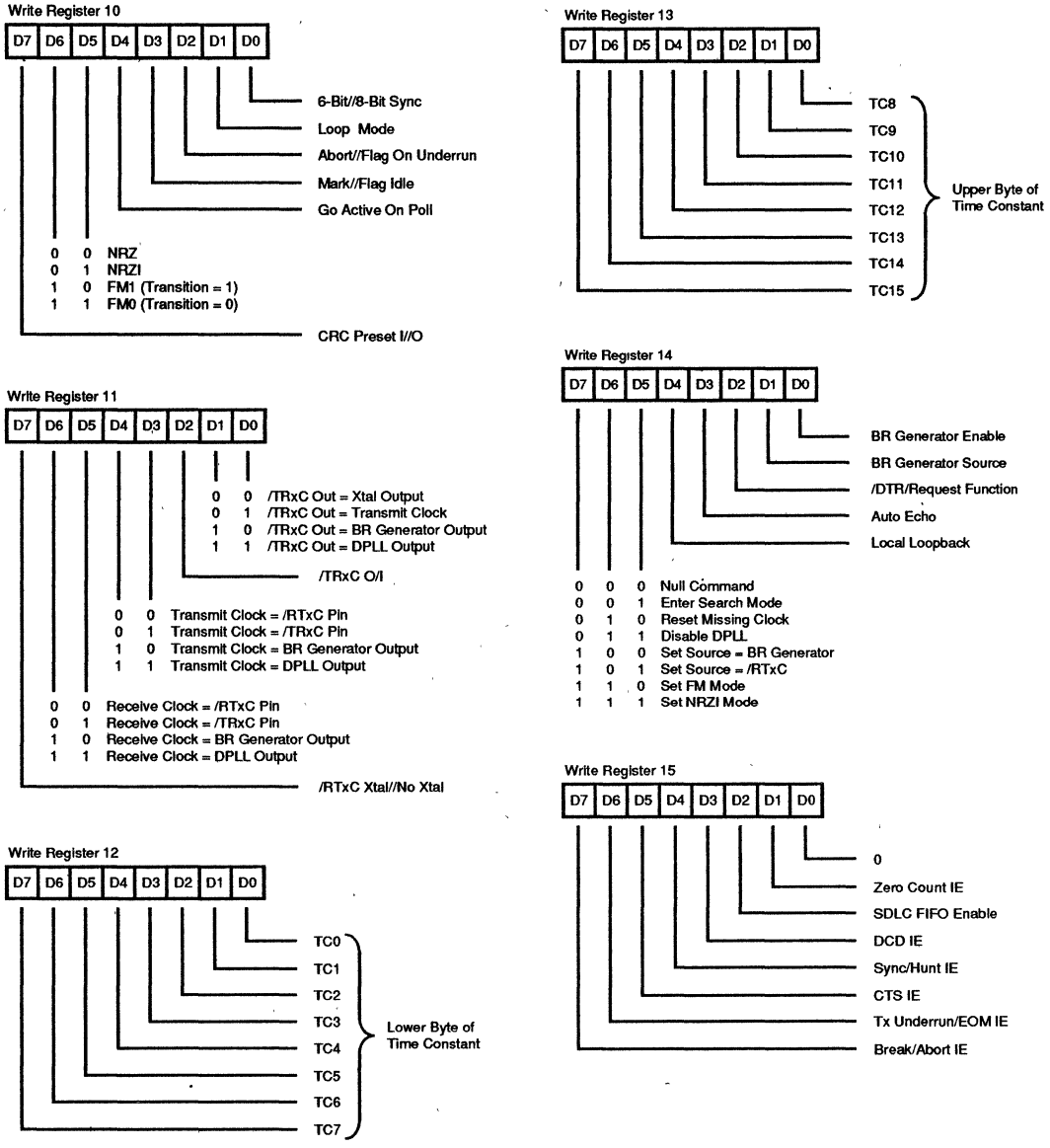


Figure 17. Write Register Bit Functions (Continued)

Read Registers. The SCC contains ten read registers (eleven, counting the receive buffer (RR8) in each channel). Four of these may be read to obtain status information (RR0, RR1, RR10, and RR15). Two registers (RR12 and RR13) are read to learn the baud rate generator time constant. RR2 contains either the unmodified interrupt vector (Channel A) or the vector modified by status information (Channel B). RR3 contains the Interrupt Pending (IP) bits (Channel A only). RR6 and RR7 contain the information in the SDLC Frame Status FIFO, but is only read when WR15 D2 is set (Figure 18).

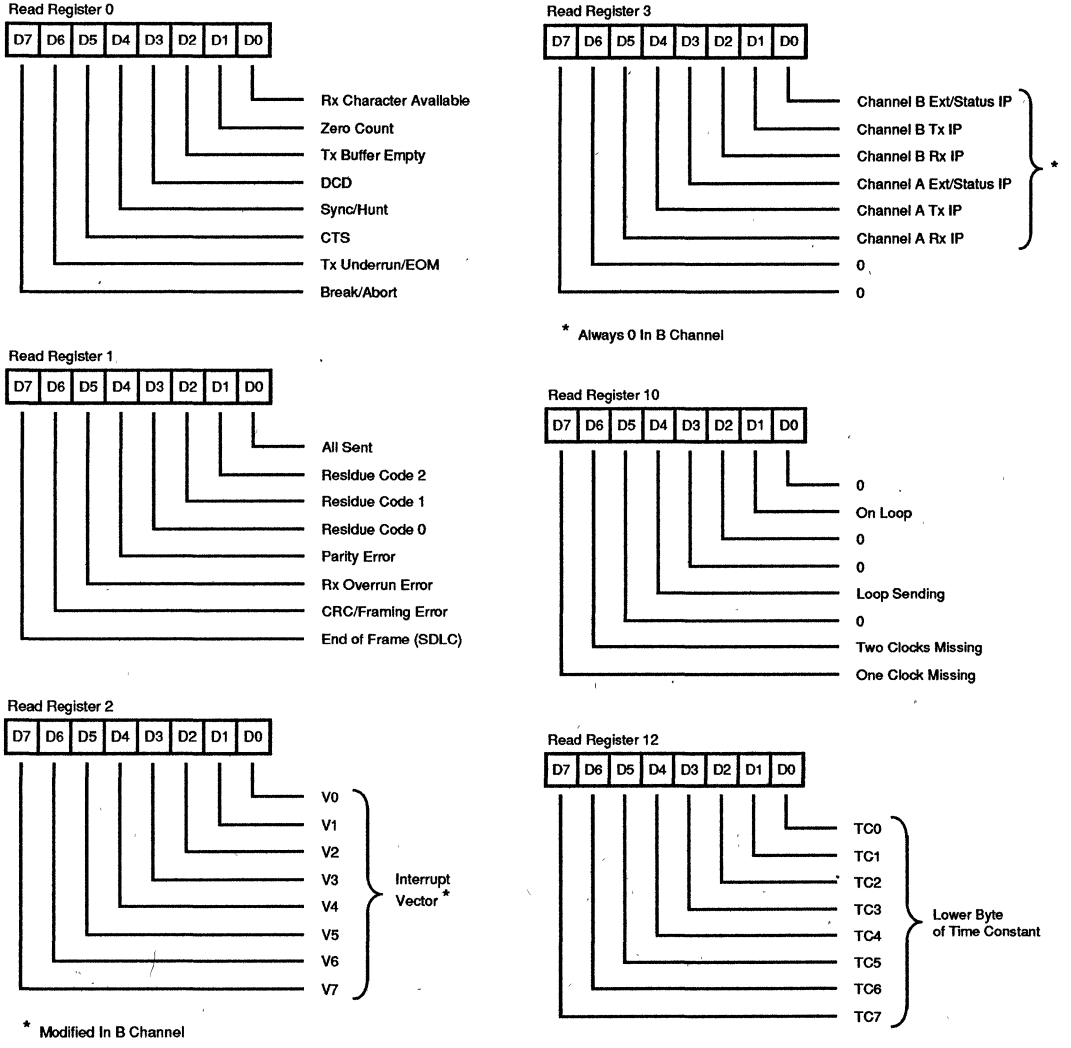


Figure 18. Read Register Bit Functions

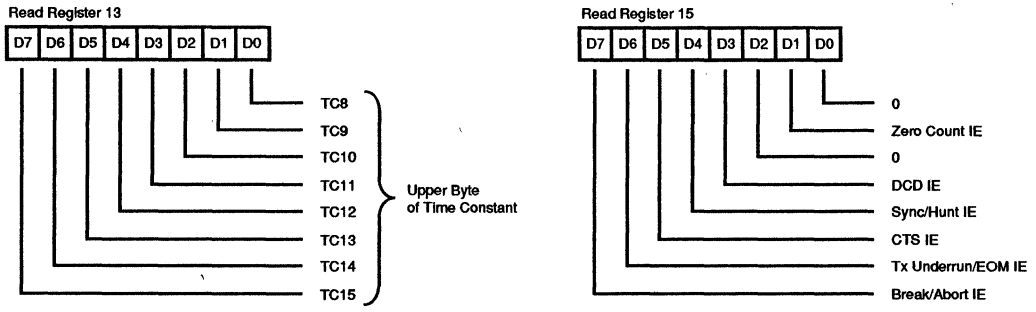


Figure 18. Read Register Bit Functions (Continued)

Z85C30 Timing

The SCC generates internal control signals from the /WR and /RD that are related to PCLK. Since PCLK has no phase relationship with /WR and /RD, the circuitry generating the internal control signals provides time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the SCC The recovery time required for proper operation is specified from the falling edge of /WR or /RD in the first transaction involving the SCC to the falling edge of /WR or /RD in the second transaction involving the SCC This time must be at least 4 PCLKs regardless of which register or channel is being accessed.

Read Cycle Timing

Figure 19 illustrates Read cycle timing. Addresses on A//B and D//C and the status on /INTACK must remain stable throughout the cycle. If /CE falls after /RD falls, or if it rises before /RD rises, the effective /RD is shortened.

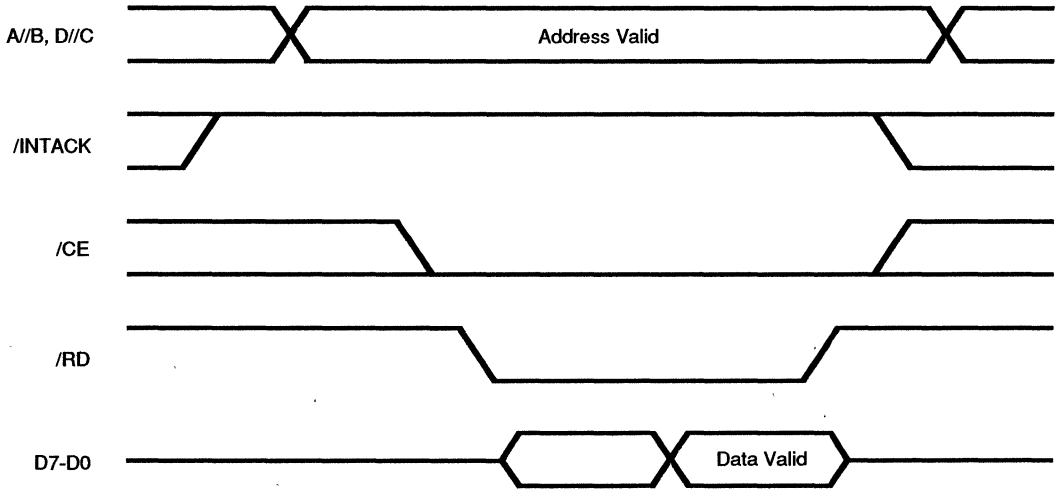


Figure 19. Read Cycle Timing

Write Cycle Timing

Figure 20 illustrates Write cycle timing. Addresses on A//B and D//C and the status on /INTACK must remain stable throughout the cycle. If /CE falls after /WR falls, or if it rises before /WR rises, the effective /WR is shortened. Data must be valid before the falling edge of /WR.

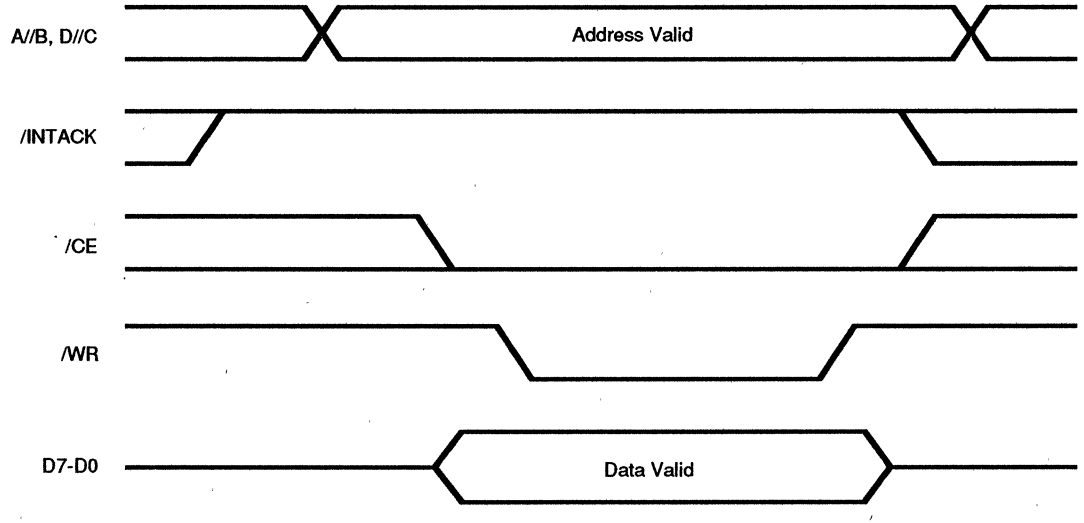


Figure 20. Write Cycle Timing

Interrupt Acknowledge Cycle Timing

Figure 21 illustrates Interrupt Acknowledge cycle timing. Between the time /INTACK goes Low and the falling edge of /RD, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the SCC and IEI is High when /RD falls, the Acknowledge cycle is intended for the SCC. In this case, the SCC may be programmed to respond to /RD Low by placing its interrupt vector on D7-D0. It then sets the appropriate Interrupt-Under-Service latch internally. If the external daisy chain is not used, then AC parameter #38 is required to settle the interrupt priority daisy chain internal to the SCC If the external daisy chain is used, the user should follow the equation in AC Characteristics, Note 5, for calculating the required daisy-chain settle time.

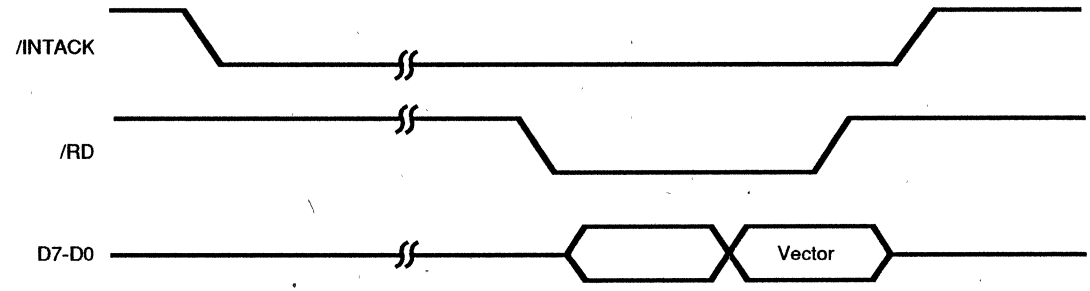


Figure 21. Interrupt Acknowledge Cycle Timing

The SCC generates internal control signals from /AS and /DS that are related to PCLK. Since PCLK has no phase relationship with /AS and /DS, the circuitry generating these internal control signals must provide time for metastable conditions to disappear This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the SCC. The recovery time required for proper operation is specified from the falling edge of /DS in the first transaction involving the SCC to the falling edge of /DS in the second transaction involving the SCC.

Read Cycle Timing

Figure 22 illustrates Read cycle timing. The address on AD7-AD0 and the state of /CS0 and /INTACK are latched by the rising edge of /AS. R//W must be High to indicate a Read cycle CS1 must also be High for the Read cycle to occur The data bus drivers in the SCC are then enabled while /DS is Low.

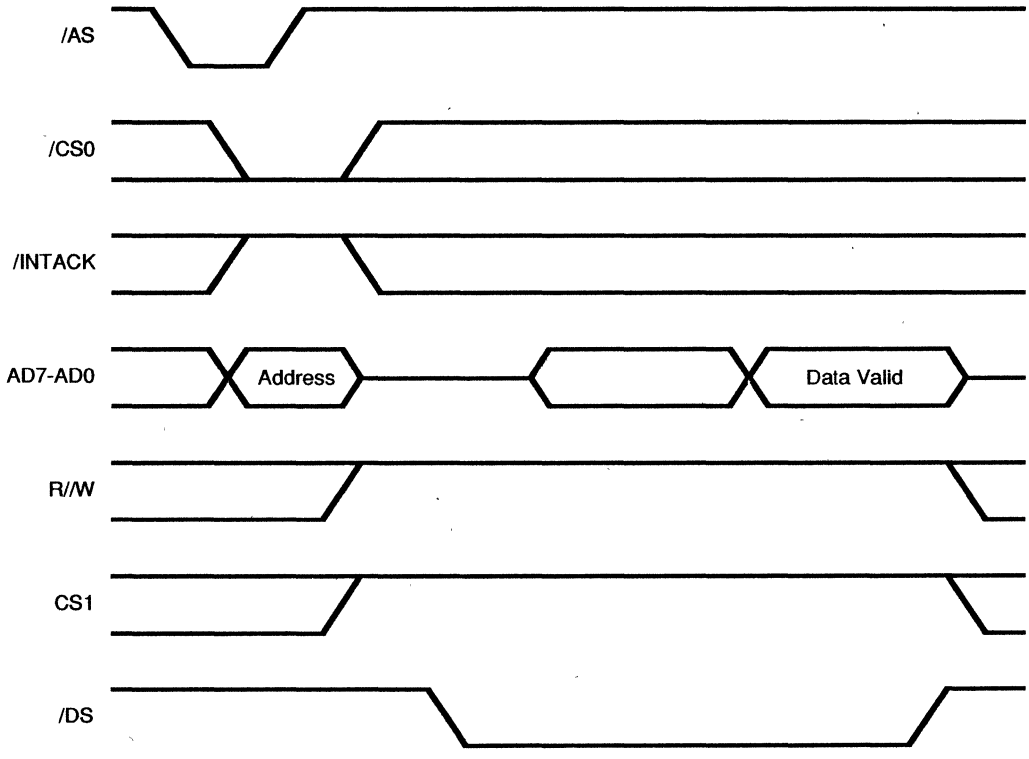


Figure 22. Read Cycle Timing

Write Cycle Timing

Figure 23 illustrates Write cycle timing. The address on AD7-AD0 and the state of /CS0 and /INTACK are latched by the rising edge of /AS. R//W must be Low to indicate a Write cycle. CS1 must be High for the Write cycle to occur. /DS Low strobes the data into the SCC.

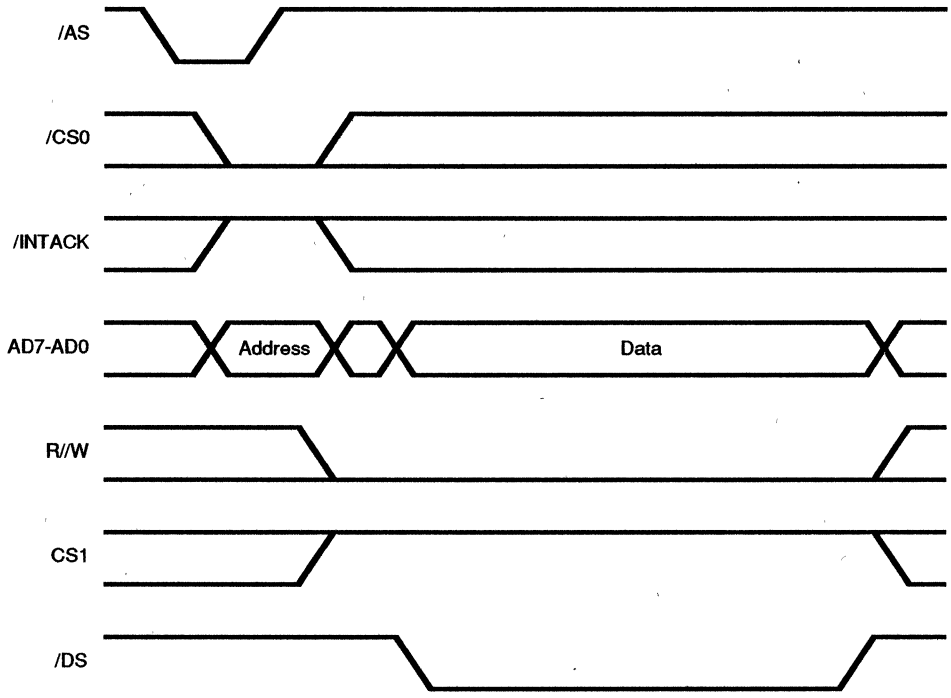


Figure 23. Write Cycle Timing

Interrupt Acknowledge Cycle Timing

Figure 24 illustrates Interrupt Acknowledge cycle timing. The address on AD7-AD0 and the state of /CS0 and /INTACK are latched by the rising edge of /AS. However, if /INTACK is Low, the address and /CS0 are ignored. The state of the R//W and CS1 are also ignored for the duration of the Interrupt Acknowledge cycle. Between the rising edge of /AS and the falling edge of /DS, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the SCC, and IEI is High when /DS falls, the Acknowledge cycle was intended for the SCC. In this case, the SCC is programmed to respond to RD Low by placing its interrupt vector on D7-D0 and then internally set the appropriate Interrupt-Under-Service latch.

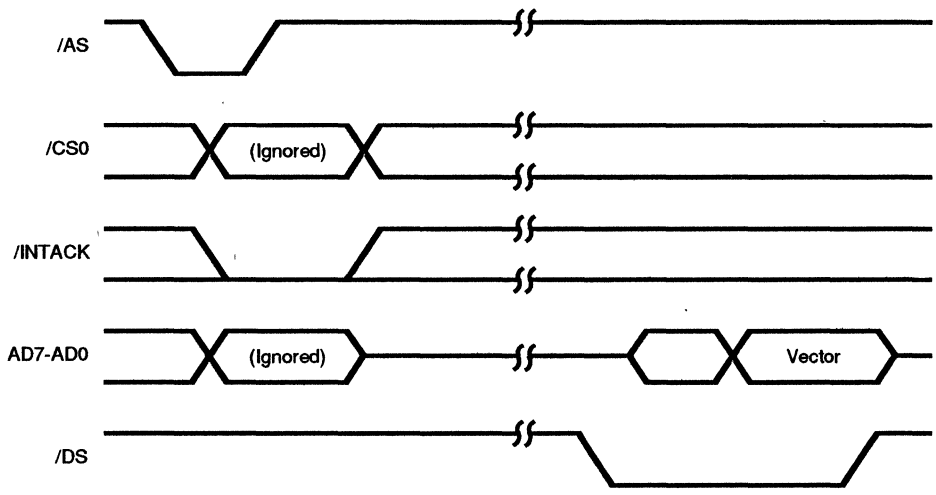


Figure 24. Interrupt Acknowledge Cycle Timing

OTHER ZILOG DATA COMMUNICATIONS PRODUCTS

SIO Family	
Z84C40 SIO	Dual channel multiprotocol USART.
Z84C13 IPC	Z80 CPU with integrated SIO, CTC and WDT.
Z84C15 IPC	Z80 CPU with integrated SIO, CTC, WDT and PIO.

SCC Family	
Z08530 SCC	NMOS SCC Low cost with speeds up to 8 MHz.
Z85130 ESCC	Enhanced SCC with 4-byte Tx and 8-byte Rx FIFOs and many other new features.
Z16C35 ISCC	SCC with 4 channel DMA and advanced CPU interface.
Z80181 SAC	Z180 CPU with integrated single channel SCC

USC Family	
Z16C30 USC	Dual channel high performance multi-protocol data communications up to 10 Megabits/second.
Z16C33 MUSC	Single channel USC w/ ISDN Time Slot Assigner.
Z16C31 IUSC	MUSC with high performance dual channel DMA (available Q1/91).
Z16C50 DDPLL	Dual channel DPLL cell from the USC

ABSOLUTE MAXIMUM RATINGS

V_{CC} Supply Voltage range ..... -0.3V to +7.0V
Voltages on all pins
with respect to GND ..... -3V to V_{CC}+0.3V
Operating Ambient
Temperature ..... See Ordering Information
Storage Temperature ..... -65°C to +150°C

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 sections below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin.

- +4.50 V ≤ V_{CC} ≤ + 5.50 V
- GND = 0 V
- T_A as specified in Ordering Information

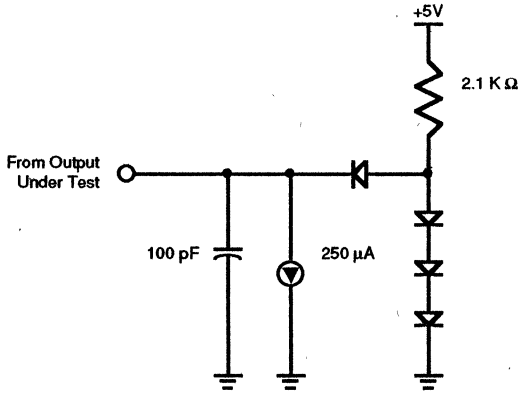


Figure 25. Standard Test Load

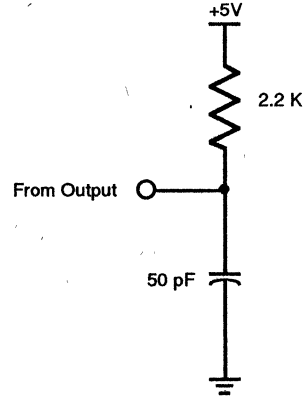


Figure 26. Open-Drain Test Load

CAPACITANCE

Symbol	Parameter	Min	Max	Unit	Test 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	

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

MISCELLANEOUS
Gate Count 6800

DC CHARACTERISTICS
Z80C30/Z85C30

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			±10.0	µA	0 4 V_{IN} + 2.4V
I_{OL}	Output Leakage			±10.0	µA	0.4 V_{OUT} + 2.4V
I_{CC1}	V_{CC} Supply Current [2]		7	12(10 MHz)	mA	V_{CC} = 5V V_{IH} = 4.8 V_{IL} = 0
			9	15(16 MHz)	mA	Crystal Oscillator off
I_{CCOSC}	Crystal OSC Current [3]		4		mA	Current for each OSC in addition to I_{CC1}

Notes:
[1] V_{CC} = 5V ±10% unless otherwise specified, over specified temperature range.
[2] Typical I_{CC} was measured with oscillator off.
[3] No I_{CC}(OSC) max is specified due to dependency on external circuit and freqency of oscillation.

AC CHARACTERISTICS

Z85C30 Read/Write Tming Diagrams

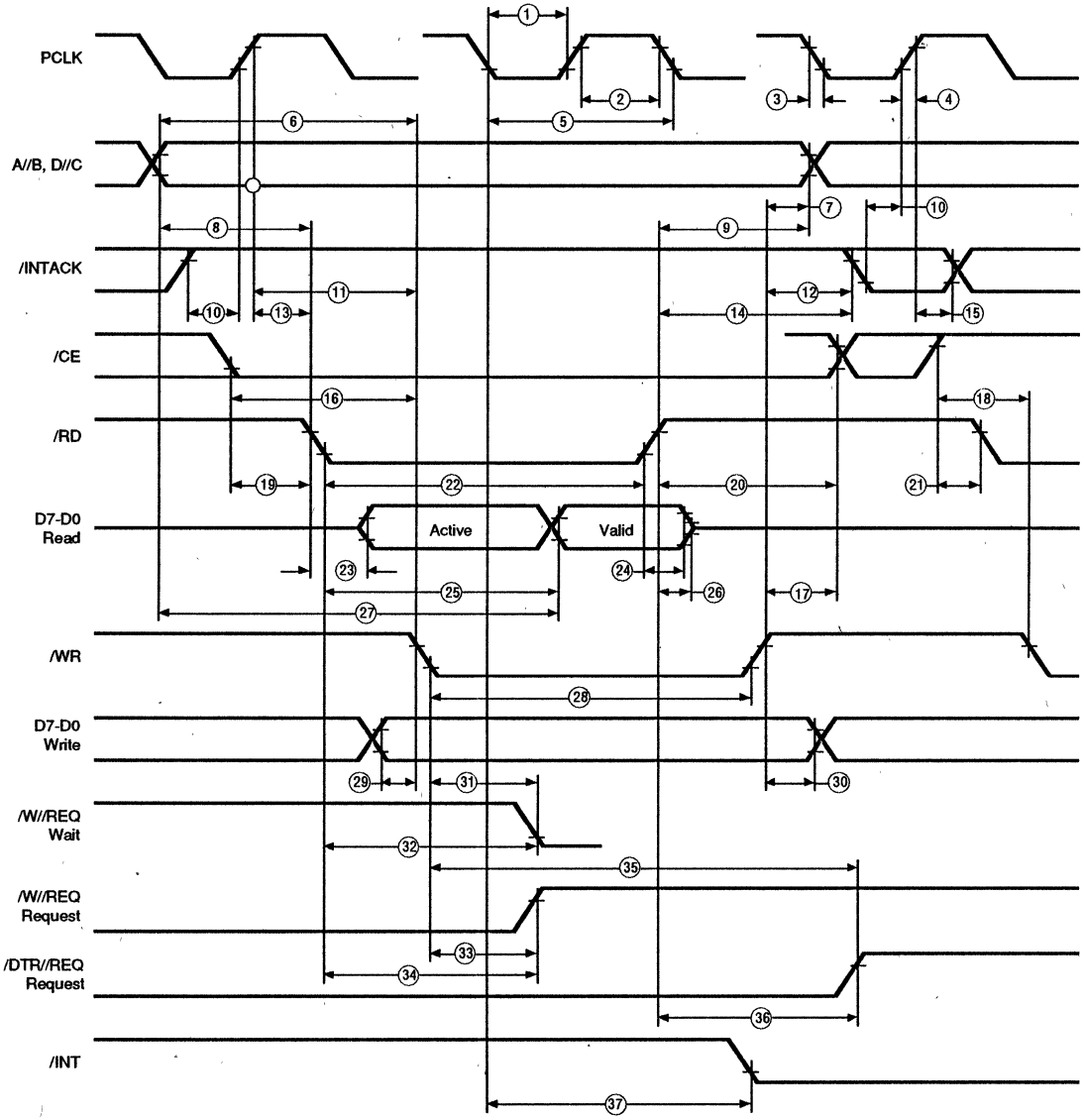


Figure 27. Z85C30 Read/Write Timing Diagram

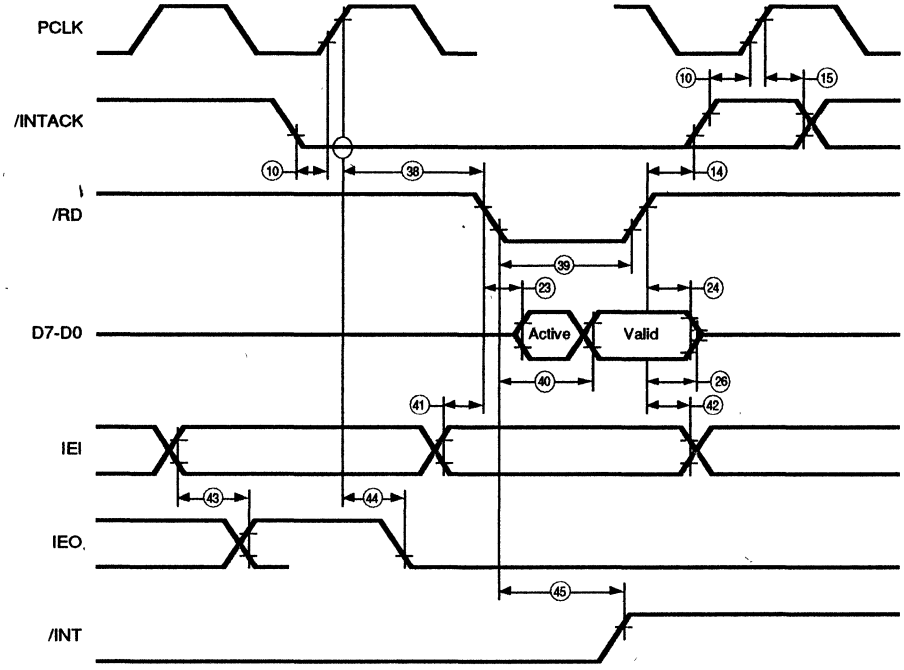


Figure 28. Z85C30 Interrupt Acknowledge Timing Diagram

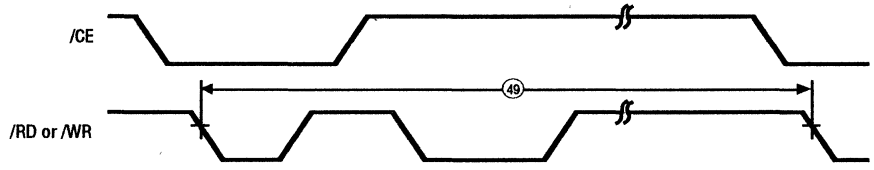


Figure 29. Z85C30 Cycle Timing Diagram

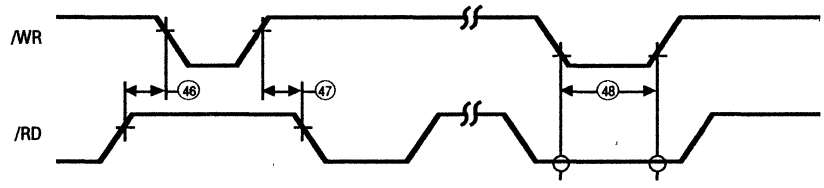


Figure 30. Z85C30 Reset Timing Diagram

AC CHARACTERISTICS

Z85C30 Read/Write Timing Table

No	Symbol	Parameter	8.5 MHz Min	8.5 MHz Max	10 MHz Min	10 MHz Max	16 MHz Min	16 MHz Max	Notes
1	TwPCl	PCLK Low Width	45	1000	40	1000	26	1000	
2	TwPCh	PCLK High Width	45	1000	40	1000	26	1000	
3	TfPC	PCLK Fall Time		10		10		5	
4	TrPC	PCLK Rise Time		10		10		5	
5	TcPC	PCLK Cycle Time	118	2000	100	2000	61	2000	
6	TsA(WR)	Address to /WR Fall Setup Time	66		50		35		
7	ThA(WR)	Address to /WR Rise Hold Time	0		0		0		
8	TsA(RD)	Address to /RD Fall Setup Time	66		50		35		
9	ThA(RD)	Address to /RD Rise Hold Time	0		0		0		
10	TslA(PC)	/INTACK to /PCLK Rise Setup Time	20		20		15		
11	TslAi(WR)	/INTACK to /WR Fall Setup Time	140		130		75		[1]
12	ThlA(WR)	/INTACK to /WR Rise Hold Time	0		0		0		
13	TslAi(RD)	/INTACK to /RD Fall Setup Time	140		130		75		[1]
14	ThlA(RD)	/INTACK to /RD Rise Hold Time	0		0		0		
15	ThlA(PC)	/INTACK to /PCLK Rise Hold Time	38		30		15		
16	TsCEl(WR)	/CE Low to /WR Fall Setup Time	0		0		0		
17	ThCE(WR)	/CE to /WR Rise Hold Time	0		0		0		
18	TsCEh(WR)	/CE High to /WR Fall Setup Time	58		50		30		
19	TsCEl(RD)	/CE Low to /RD Fall Setup Time	0		0		0		[1]
20	ThCE(RD)	/CE to /RD Rise Hold Time	0		0		0		[1]
21	TsCEh(RD)	/CE High to /RD Fall Setup Time	58		50		30		[1]
22	TwRDl	/RD Low Width	145		125		70		[1]
23	TdRD(DRA)	/RD Fall to Read Data Active Delay	0		0		0		
24	TdRDr(DR)	/RD Rise to Read Data Not Valid Delay	0		0		0		
25	TdRDl(DR)	/RD Fall to Read Data Valid Delay		135		120		65	
26	TdRD(DRz)	/RD Rise to Read Data Float Delay		38		35		20	
27	TdA(DR)	Address to Read Data Valid Delay		210		180		100	
28	TwWRl	/WR Low Width	145		125		70		
29	TsDW(WR)	Write Data to /WR Fall Setup Time	10		10		10		
30	ThDW(WR)	/Write Data to /WR Rise Hold Time	0		0		0		
31	TdWR(W)	/WR Fall to Wait Valid Delay		168		160		80	[4]
32	TdRD(W)	/RD Low to Wait Valid Delay		168		160		80	[4]
33	TdWRf(REQ)	/WR Fall to /W//REQ Not Valid Delay		168		160		80	
34	TdRDf(REQ)	/RD Fall to /W//REQ Not Valid Delay		168		160		80	
35	TdWRr(REQ)	/WR Fall to /DTR//REQ Not Valid		4TcPc		4TcPc		4TcPc	
36	TdRDr(REQ)	/RD Rise to /DTR//REQ Not Valid		NA		NA		NA	
37	TdPC(INT)	/PCLK Fall to /INT Valid Delay		500		450		175	
38	TdIAi(RD)	/INTACK to /RD Fall (Ack) Delay	145		125		75		[5]
39	TwRDA	/RD (Acknowledge) Width	145		125		70		
40	TdRDA(DR)	/RD Fall (Ack) to Read Data Valid		135		120		70	

AC CHARACTERISTICS

Z85C30 Read/Write Timing Table (Continued)

No	Symbol	Parameter	8.5 MHz Min	8.5 MHz Max	10 MHz Min	10 MHz Max	16 MHz Min	16 MHz Max	Notes
41	TslEI(RDA)	IEI to /RD Fall (Ack) Setup Time	95		95		50		
42	ThlEI(RDA)	IEI to /RD Rise (Ack) Hold Time	0		0		0		
43	TdIEI(IEO)	IEI to IEO Delay Time		95		90		50	
44	TdPC(IEO)	/PCLK Rise to IEO Delay		195		175		80	
45	TdRDA(INT)	/RD Fall to /INT Inactive Delay		480		320		200	[4]
46	TdRD(WRQ)	/RD Rise to /WR Fall for No Reset	15		15		10		
47	TdWRQ(RD)	/WR Rise to /RD Fall for No Reset	15		15		10		
48	TwRES	/WR & /RD Low for Reset	145		100		75		
49	Trc	Valid Access Recovery Time	4TcPc		4TcPc		4TcPc		[3]

NOTES:

[1] Parameter does not apply to Interrupt Acknowledge transactions.

[3] Parameter applies only between transactions involving the SCC.

[4] Open-drain output, measured with open-drain test load.

[5] Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TslEI(RDA) for the SCC, and TdIEIf(IEO) for each device separating them in the daisy chain.

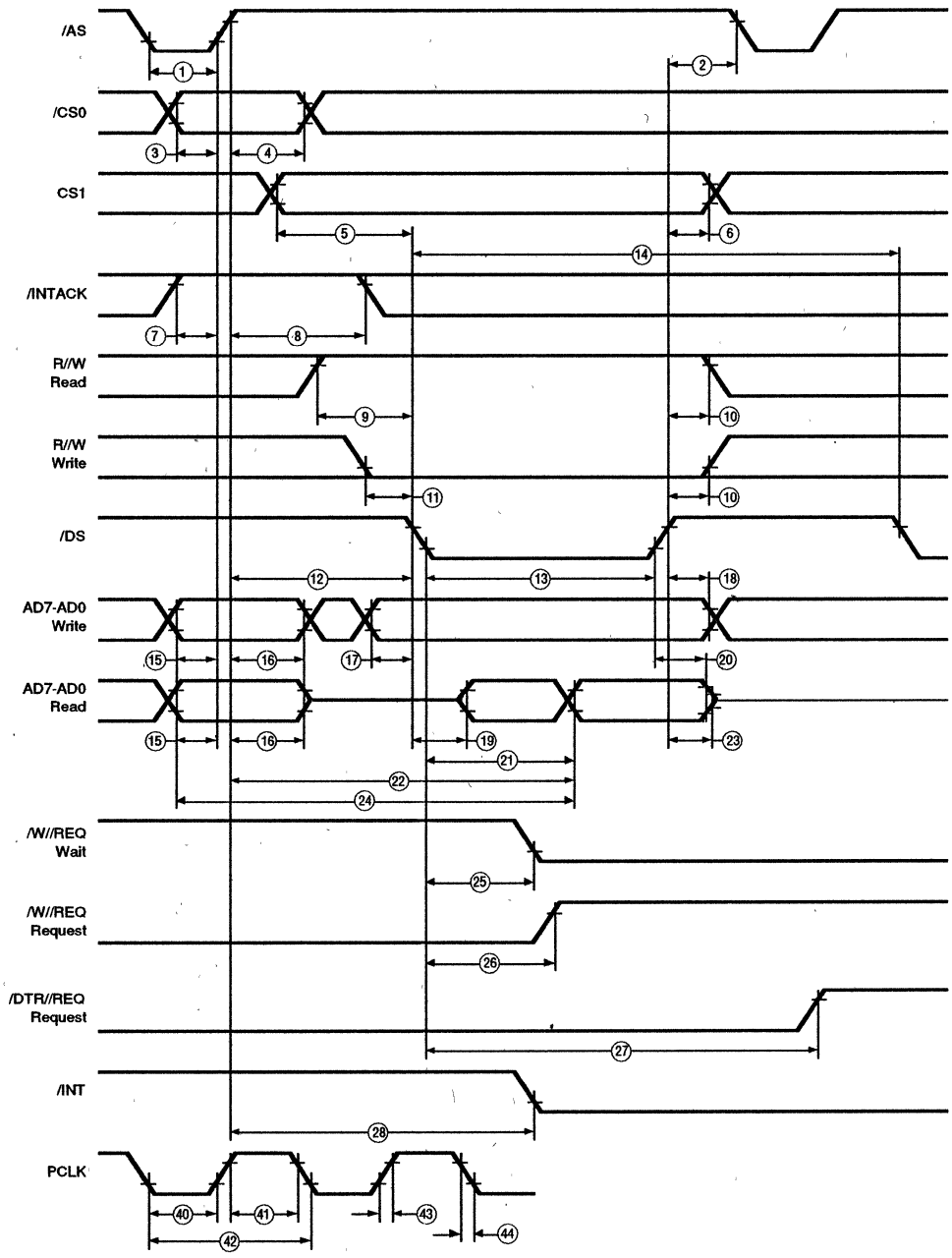


Figure 31. Z80C30 Read/Write Timing Diagram

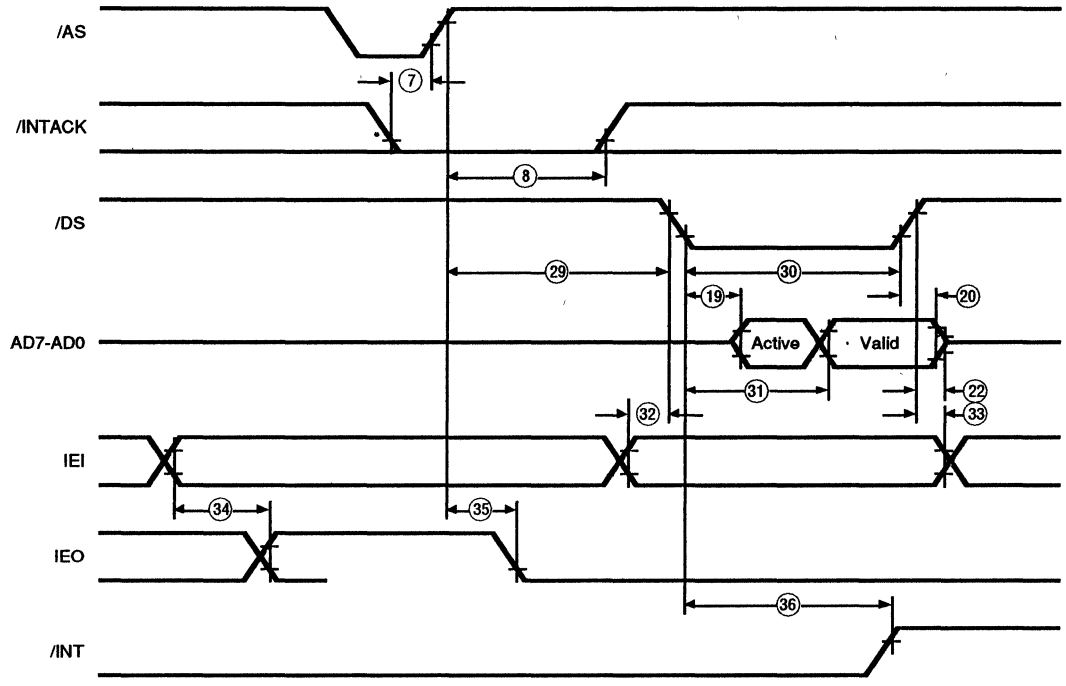


Figure 32. Z80C30 Interrupt Acknowledge Timing Diagram

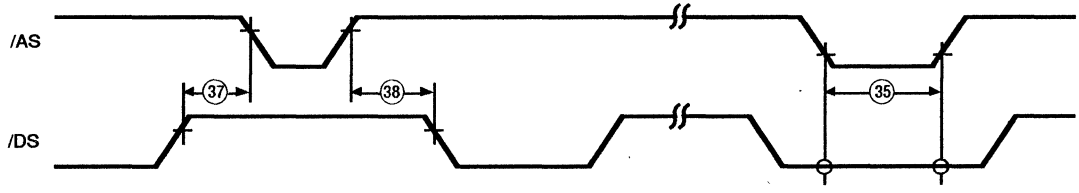


Figure 33. Z80C30 Reset Timing Diagram

AC CHARACTERISTICS

Z80C30 Read/Write Timing Table

No	Symbol	Parameter	8 MHz Min	8 MHz Max	10 MHz Min	10 MHz Max	Notes *
1	TwAS	/AS Low Width	35		30		
2	TdDS(AS)	/DS Rise to /AS Fall Delay	15		10		[1]
3	TsCSO(AS)	/CS0 to /AS Rise Setup Time	0		0		[1]
4	ThCSO(AS)	/CS0 to /AS Rise Hold Time	30		20		[1]
5	TsCS1(DS)	CS1 to /DS Fall Setup Time	65		50		[1]
6	ThCS1(DS)	CS1 to /DS Rise Hold Time	30		20		[1]
7	TsIA(AS)	/INTACK to /AS Rise Setup Time	10		10		
8	ThIA(AS)	/INTACK to /AS Rise Hold Time	150		125		
9	TsRWR(DS)	R//W (Read) to /DS Fall Setup Time	65		50		
10	ThRW(DS)	R//W to /DS Rise Hold Time	0		0		
11	TsRWW(DS)	R//W (Write) to /DS Fall Setup Time	0		0		
12	TdAS(DS)	/AS Rise to /DS Fall Delay	30		20		
13	TwDSI	/DS Low Width	150		125		
14	TrC	Valid Access Recovery Time	4TcPC		4TcPC		[2]
15	TsA(AS)	Address to /AS Rise Setup Time	10		10		[1]
16	ThA(AS)	Address to /AS Rise Hold Time	25		20		[1]
17	TsDW(DS)	Write Data to /DS Fall Setup Time	15		10		
18	ThDW(DS)	Write Data to /DS Rise Hold Time	0		0		
19	TdDS(DA)	/DS Fall to Data Active Delay	0		0		
20	TdDSr(DR)	/DS Rise to Read Data Not Valid Delay	0		0		
21	TdDSf(DR)	/DS Fall to Read Data Valid Delay		140		120	
22	TdAS(DR)	/AS Rise to Read Data Valid Delay		250		190	
23	TdDS(DRz)	/DS Rise to Read Data Float Delay		40		35	[3]
24	TdA(DR)	Address Required Valid to Read Data Valid Delay		260		210	
25	TdDS(W)	/DS Fall to Wait Valid Delay		170		160	[4]
26	TdDSf(REQ)	/DS Fall to /W//REQ Not Valid Delay		170		160	
27	TdDSr(REQ)	/DS Fall to /D1R//REQ Not Valid Delay		4TcPC		4TcPC	
28	TdAS(INT)	/AS Rise to /INT Valid Delay		500		500	[4]
29	TdAS(DSA)	/AS Rise to /DS Fall (Acknowledge) Delay	250		225		[5]
30	TwDSA	/DS (Acknowledge) Low Width	150		125		
31	TdDSA(DR)	/DS Fall (Acknowledge) to Read Data Valid Delay		140		120	
32	TslEI(DSA)	IEI to /DS Fall (Acknowledge) Setup Time	80		80		
33	ThIEI(DSA)	IEI to /DS Rise (Acknowledge) Hold Time	0		0		
34	TdIEI(IEO)	IEI to IEO Delay		90		90	
35	TdAS(IEO)	/AS Rise to IEO Delay		200		175	[6]
36	TdDSA(INT)	/DS Fall (Acknowledge) to /INT Inactive Delay		450		450	[4]
37	TdDS(ASQ)	/DS Rise to /AS Fall Delay for No Reset	15		15		
38	TdASQ(DS)	/AS Rise to /DS Fall Delay for No Reset	20		15		
39	TwRES	/AS and /DS Coincident Low for Reset	150		100		[7]
40	TwPCI	PCLK Low Width	50	1000	40	1000	

AC CHARACTERISTICS

Z80C30 Read/Write Timing Table (Continued)

No	Symbol	Parameter	8 MHz Min	8 MHz Max	10 MHz Min	10 MHz Max	Notes *
41	TwPCh	PCLK High Width	50	1000	40	1000	
42	TcPC	PCLK Cycle Time	125	2000	100	2000	
43	TrPC	PCLK Rise Time		10		10	
44	TfPC	PCLK Fall Time		10		10	

NOTES:

[1] Parameter does not apply to interrupt Acknowledge transactions.

[2] Parameter applies only between transactions involving the SCC.

[3] Float delay is defined as the time required for a ±0.5V change in the output with a maximum DC load and a minimum AC load.

[4] Open-drain output, measured with open-drain test load.

[5] Parameter is system dependent. For any Z-SCC in the daisy chain. TdAS(DSA) must be greater than the sum of TdAS(IEO) for the highest priority device in the daisy chain. TslEl(DSA) for the Z-SCC, and TdlElf(IEO) for each device separating them in the daisy chain.

[6] Parameter applies only to a Z-SCC pulling INT Low at the beginning of the Interrupt Acknowledge transaction.

[7] Internal circuitry allows for the reset provided by the Z8 to be recognized as a reset by the Z-SCC. All timing references assume 2.0V for a logic "1" and 0 8V for a logic "0".

* Units in nanoseconds(ns).

AC CHARACTERISTICS

Z85C30/Z80C30 General Timing Diagram

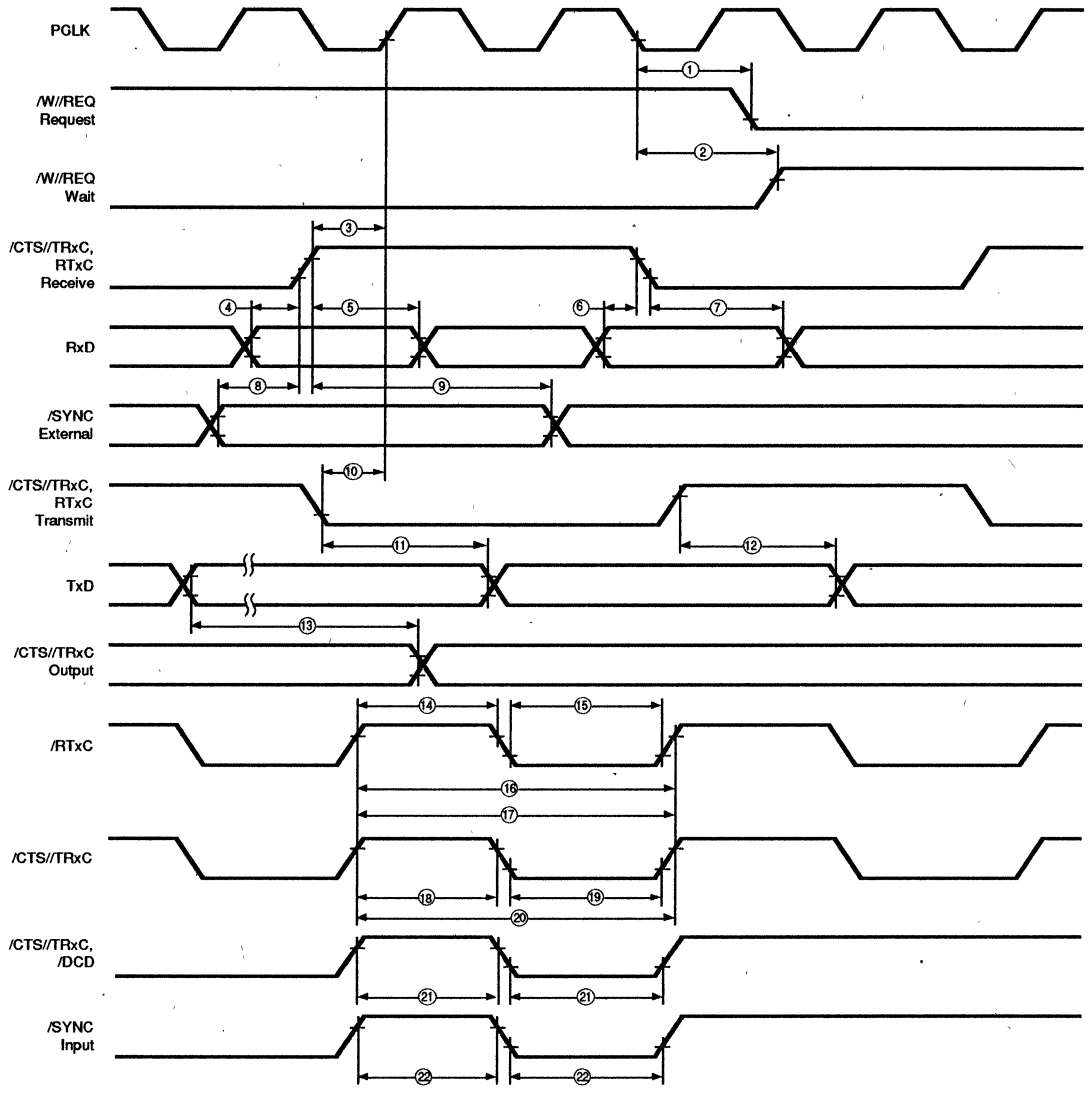


Figure 34. Z85C30/Z80C30 General Timing Diagram

AC CHARACTERISTICS

Z85C30/Z80C30 General Timing Table

No	Symbol	Parameter	8.5 MHz Min	8.5 MHz Max	10 MHz Min	10 MHz Max	16 MHz Min	16 MHz Max	Notes*
1	TdPC(REQ)	/PCLK Low to W/REQ Valid		250		200		110	
2	TsPC(W)	/PCLK Low to Wait Inactive		350		300		180	
3	TsRXC(PC)	/RxC High to /PCLK High Setup Time	NA	NA	NA	NA	NA	NA	[1,4]
4	TsRXD(RXCr)	RxD to /RxC High Setup Time	0		0		0		[1]
5	ThRXD(RxCr)	RxD to /RxC High Hold Time	150		125		60		[1]
6	TsRXD(RXCf)	RxD to /RxC Low Setup Time	0		0		0		[1,5]
7	ThRXD(RXCf)	RxD to /RxC Low Hold Time	150		125		60		[1,5]
8	TsSY(RXC)	SYNC to /RxC High Setup Time	-200		-150		-100		[1]
9	ThSY(RXC)	SYNC to /RxC High Hold Time	5TcPc		5TcPc		5TcPc		[1]
10	TsTXC(PC)	/TxC Low to /PCLK High Setup Time	NA		NA		NA		[2,4]
11	TdTXCf(TXD)	/TxC Low to TxD Delay		190		150		85	[2]
12	TdTxCr(TXD)	/TxC High to TxD Delay		190		150		85	[2,5]
13	TdTXD(TRX)	TxD to TRxC Delay		200		140		80	
14	TwRTXh	RTxC High Width	130		120		80		[6]
15	TwRTXl	TRxC Low Width	130		120		80		[6]
16a	TcRTX	RTxC Cycle Time	472		400		244		[6,7]
16b	TxRX(DPLL)	DPLL Cycle Time Min	59		50		31		[7,8]
17	TcRTXX	Crystal Osc. Period	118	1000	100	1000	100	1000	[3]
18	TwTRXh	TRxC High Width	130		120		80		[6]
19	TwTRXl	TRxC Low Width	130		120		80		[6]
20	TcTRX	TRxC Cycle Time	472		400		244		[6,7]
21	TwEXT	DCD or CTS Pulse Width	200		120		70		
22	TwSY	SYNC Pulse Width	200		120		70		

Notes:

[1] RxC is /RTxC or /TRxC, whichever is supplying the receive clock.
[2] TxC is /TRxC or /RTxC, whichever is supplying the transmit clock.
[3] Both /RTxC and /SYNC have 30 pf capacitors to ground connected to them.
[4] Synchronization of RxC to PCLK is eliminated in divide by four operation.
[5] Parameter applies only to FM encoding/decoding.
[6] Parameter applies only for transmitter and receiver; DPLL and baud rate generator timing requirements are identical to case PCLK requirements.
[7] The maximum receive or transmit data rate is 1/4 PCLK.
[8] Applies to DPLL clock source only. Maximum data rate of 1/4 PCLK still applies. DPLL clock should have a 50% duty cycle.

* Units in nanoseconds (ns).

AC CHARACTERISTICS

Z85C30/Z80C30 System Timing Diagram

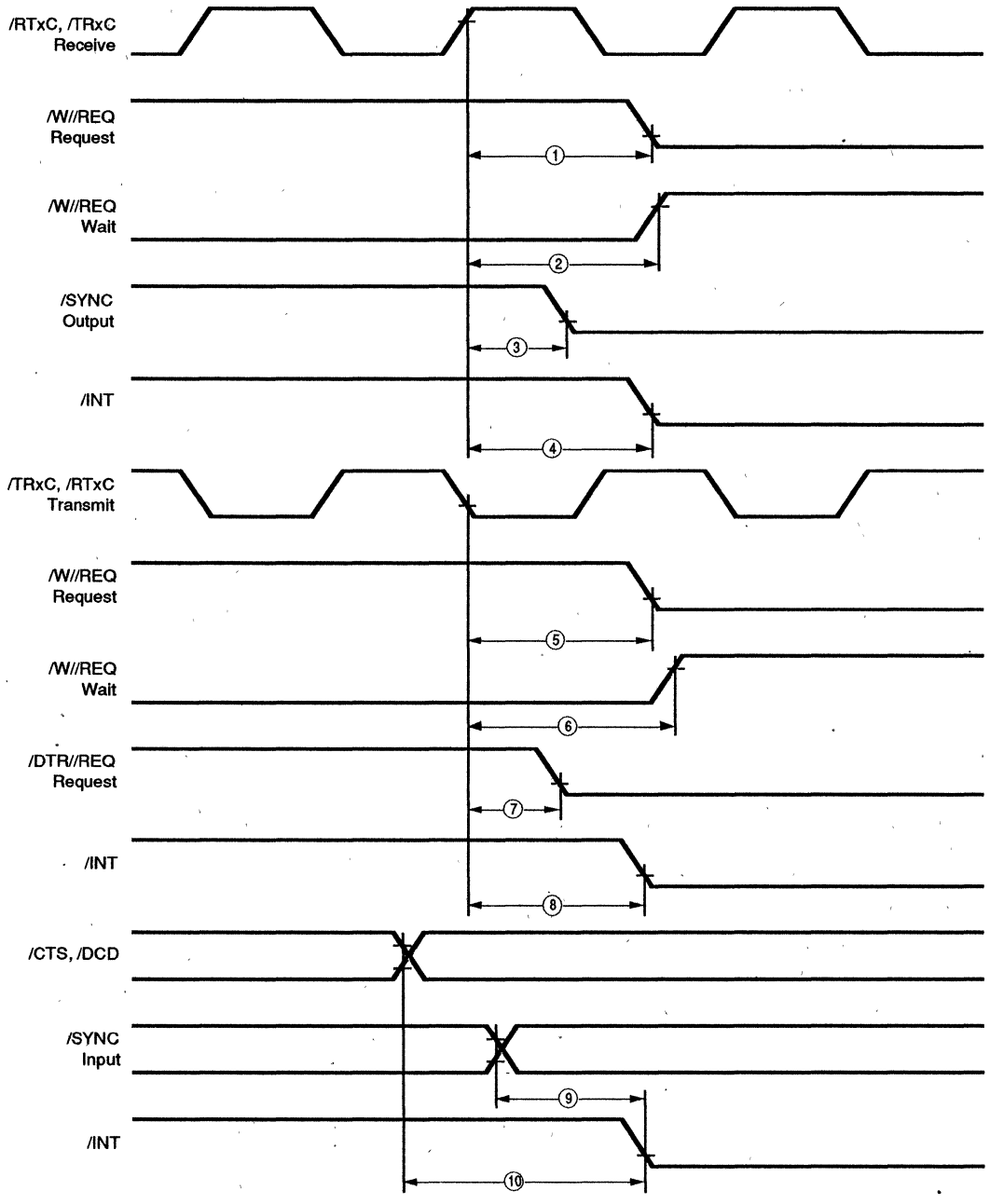


Figure 35. Z85C30/Z80C30 System Timing Diagram

AC CHARACTERISTICS

Z85C30/Z80C30 System Timing Table

No	Symbol	Parameter	8.5 MHz Min	8.5 MHz Max	10 MHz Min	10 MHz Max	16 MHz Min	16 MHz Max	Notes*
1	TdRXC(REQ)	/RxC High to W/REQ Valid	8	12	8	12	8	12	[2]
2	TdRXC(W)	/RxC High to Wait Inactive	8	14	8	14	8	14	[1,2]
3	TdRdXC(SY)	/RxC High to SYNC Valid	4	7	4	7	4	7	[2]
4a	TsRXC(INT), Z85C30	/RxC High to INT Valid	10	16	10	16	10	16	[1,2]
4b	TdRXC(INT), Z80C30	/RxC High to INT Valid	8	12	8	12			[1,2]
			2	3	2	3			[4]
5	TdTXC(REQ)	/TxC Low to W/REQ Valid	5	8	5	8	5	8	[3]
6	TdTXC(W)	/TxC Low to Wait Inactive	5	11	5	11	5	11	[1,3]
7	TdTXC(DRQ)	/Txc Low to DTR/REQ Valid	4	7	4	7	4	7	[3]
8a	TdTXC(INT), Z85C30	/TxC Low to /INT Valid	6	10	6	10	6	10	[1,3]
8b	TdTXC(INT), Z80C30	/TxC Low to /INT Valid	4	6	4	6			[1,3]
			2	3	2	3			[4]
9a	TdSY(INT)	SYNC to INT Valid	2	6	2	6	2	6	[1]
9b	TdSY(INT)	SYNC to INT Valid	2	3	2	3			[1,4]
10a	TdEXT(INT), Z85C30	/DCD or /CTS to /INT Valid	2	6	2	6	2	6	[1]
10b	TdEXT(INT), Z80C30		2	3	2	3			[1,4]

Notes:

[1] Open drain-output, measured with open-drain test load.

[2] /RxC is /RTxC or /TRxC, whichever is supplying the receive clock.

[3] /TxC is /TRxC or /RTxC, whichever is supplying the transmit clock.

[4] Units equal to /AS.

* Units equal to TcPc.

PRODUCT SPECIFICATION

Z8030/Z8530

Z-BUS SCC SERIAL COMMUNICATION CONTROLLER

Features

- Two independent, 0 to 2M bit/second, full-duplex channels, each with a separate crystal oscillator, baud rate generator, and Digital Phase-Locked Loop for clock recovery.
- Multi-protocol operation under program control; programmable for NRZ, NRZI, or FM data encoding.
- Asynchronous mode with five to eight bits and one, one and one-half, or two stop bits per character; programmable clock factor; break detection and generation; parity, overrun, and framing error detection.
- Synchronous mode with internal or external character synchronization on one or two synchronous characters and CRC generation and checking with CRC-16 or CRC-CCITT preset to either 1s or 0s.
- SDLC/HDLC mode with comprehensive frame-level control, automatic zero insertion and deletion, I-field residue handling, abort generation and detection, CRC generation and checking, and SDLC Loop mode operation.
- Local Loopback and Auto Echo modes.
- Supports T1 digital trunk.

General Description

The SCC Serial Communications Controller is a dual-channel, multi-protocol data communications peripheral designed for use with conventional non-multiplexed buses and the Zilog Z-BUS.® The SCC functions as a serial-to-parallel, parallel-to-serial converter/controller. The SCC can be software-configured to satisfy a wide variety of serial communications applications. The device contains a variety of new, sophisticated internal functions including on-chip baud rate generators, Digital PhaseLocked Loops, and crystal oscillators that dramatically reduce the need for external logic.

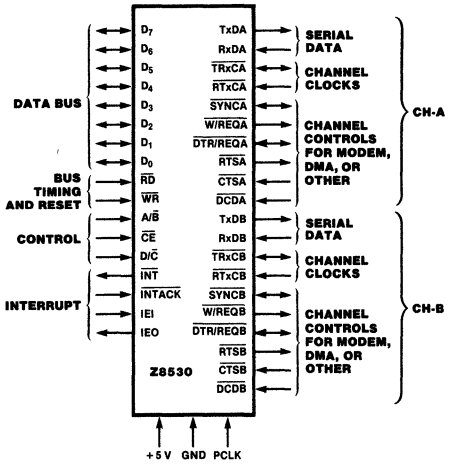


Figure 1a. Pin Functions, Z8530

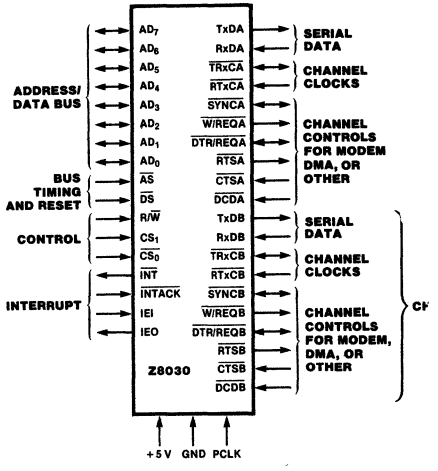


Figure 1b. Pin Functions, Z8030

General Description (Continued)

The SCC handles asynchronous formats, Synchronous byte-oriented protocols such as IBM Bisync, and Synchronous bit-oriented protocols such as HDLC and IBM SDLC. This versatile device supports virtually any serial data transfer application (cassette, diskette, tape drives, etc.).

The device can generate and check CRC codes in any Synchronous mode and can be programmed to check data integrity in various modes. The SCC also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls can be used for general-purpose I/O.

The daisy-chain interrupt hierarchy is also supported—as is standard for Zilog peripheral components.

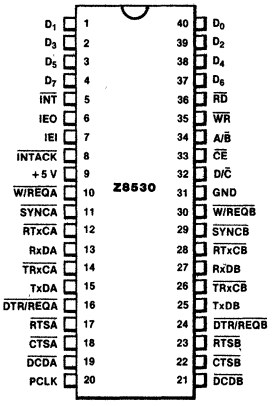


Figure 2a. DIP Pin Assignments, Z8530

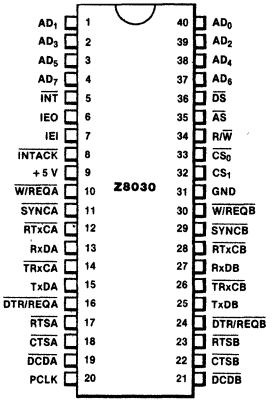


Figure 2b. DIP Pin Assignments, Z8030

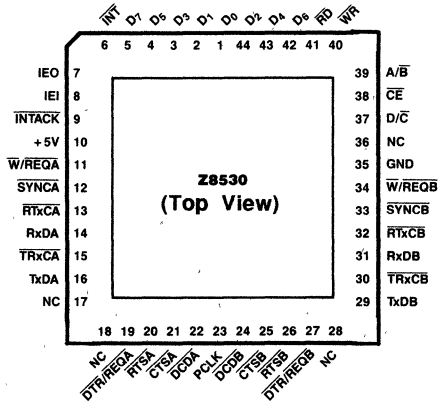


Figure 2c. Chip Carrier Pin Assignments, Z8530

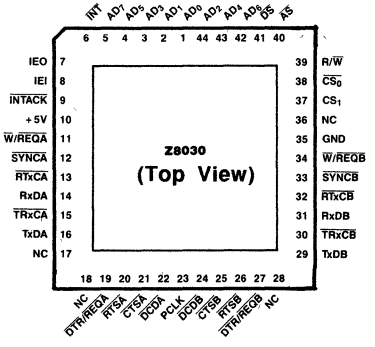


Figure 2d. Chip Carrier Pin Assignments, Z8030

Note: Power connections follow conventional descriptions below:

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

Pin Description

The following section describes the pin functions common to the Z8530 and the Z8030. Figures 1 and 2 detail the respective pin functions and pin assignments.

$\overline{\text{CTSA}}$, $\overline{\text{CTSB}}$. *Clear To Send* (inputs, active Low). If these pins are programmed as Auto Enables, a Low on the inputs enables the respective transmitters. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The SCC detects pulses on these inputs and can interrupt the CPU on both logic level transitions.

$\overline{\text{DCDA}}$, $\overline{\text{DCDB}}$. *Data Carrier Detect* (inputs/outputs, active Low). These pins function as receiver enables if they are programmed for Auto Enables; otherwise they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accomodate slow rise-time signals. The SCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.

$\overline{\text{DTR/REQA}}$, $\overline{\text{DTR/REQB}}$. *Data Terminal Ready/Request* (outputs, active Low). These outputs follow the state programmed into the DTR bit. They can also be used as general-purpose outputs or as Request lines for a DMA controller.

IEI. *Interrupt Enable In* (input, active High). IEI is used with IEO to form an interrupt daisy chain when there is more than one interruptdriven device. A High IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.

IEO. *Interrupt Enable Out* (output, active High). IEO is High only if IEI is High and the CPU is not servicing an SCC interrupt or the SCC is not requesting an interrupt (Interrupt Acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.

$\overline{\text{INT}}$. *Interrupt Request* (output, open-drain, active Low). This signal is activated when the SCC requests an interrupt.

$\overline{\text{INTACK}}$. *Interrupt Acknowledge* (input, active Low). This signal indicates an active Interrupt Acknowledge cycle. During this cycle, the SCC interrupt daisy chain settles. When $\overline{\text{RD}}$ or $\overline{\text{DS}}$ becomes active, the SCC places an interrupt vector on the data bus (if IEI is High). $\overline{\text{INTACK}}$ is latched by the rising edge of PCLK.

PCLK. *Clock* (input). This is the master SCC clock used to synchronize internal signals. PCLK is a TTL level signal. PCLK is not required to have any phase relationship with the master system clock.

RxDA, RxDB. *Receive Data* (inputs, active High). These input signals receive serial data at standard TTL levels.

$\overline{\text{RTxCA}}$ $\overline{\text{RTxCB}}$. *Receive/Transmit Clocks* (inputs, active Low). These pins can be programmed in several different modes of operation. In each channel, $\overline{\text{RTxC}}$ may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the Digital Phase-Locked Loop. These pins can also be programmed for use with the respective $\overline{\text{SYNC}}$ pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in Asynchronous modes.

$\overline{\text{RTSA}}$, $\overline{\text{RTSB}}$. *Request To Send* (outputs, active Low). When the Request To Send (RTS) bit in Write Register 5 (Figure 11) is set, the $\overline{\text{RTS}}$ signal goes Low. When the RTS bit is reset in the Asynchronous mode and Auto Enable is on, the signal goes High after the transmitter is empty. In Synchronous mode or in Asynchronous mode with Auto Enable off, the RTS pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.

$\overline{\text{SYNCA}}$, $\overline{\text{SYNCB}}$. *Synchronization* (inputs or outputs, active Low). These pins can act either as inputs, outputs, or part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to $\overline{\text{CTS}}$ and $\overline{\text{DCD}}$. In this mode, transitions on these lines affect the state of the Synchronous/Hunt status bits in Read Register 0 (Figure 10) but have no other function.

In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, $\overline{\text{SYNC}}$ must be driven Low two receive clock cycles after the last bit in the synchronous character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of $\overline{\text{SYNC}}$.

In the Internal Synchronization mode (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which synchronous characters are recognized. The synchronous condition is not latched, so these outputs are active each time a synchronization pattern is recognized (regardless of character boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.

TxDA, TxDB. *Transmit Data* (outputs, active High). These output signals transmit serial data at standard TTL levels.

$\overline{\text{TRxCA}}$, $\overline{\text{TRxCB}}$. *Transmit/Receive Clocks* (inputs or outputs, active Low). These pins can be programmed in several different modes of operation. TRxC may supply the receive clock or the transmit clock in the input mode or supply the output of the Digital Phase-Locked Loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.

$\overline{\text{W/REQA}}$, $\overline{\text{W/REQB}}$. *Wait/Request* (outputs, open-drain when programmed for a Wait function, driven High or Low when programmed for a Request function). These dual-purpose outputs may be programmed as Request lines for a DMA controller or as Wait lines to synchronize the CPU to the SCC data rate. The reset state is Wait.

Z8530

A/$\overline{\text{B}}$. *Channel A/Channel B Select* (input). This signal selects the channel in which the read or write operation occurs.

$\overline{\text{CE}}$. *Chip Enable* (input, active Low). This signal selects the SCC for a read or write operation.

D_0-D_7 *Data Bus* (bidirectional, 3-state). These lines carry data and commands to and from the SCC.

D/$\overline{\text{C}}$. *Data/Control Select* (input). This signal defines the type of information transferred to or from the SCC. A High means data is transferred; a Low indicates a command.

$\overline{\text{RD}}$. *Read* (input, active Low). This signal indicates a read operation and when the SCC is selected, enables the SCC's bus drivers. During the Interrupt Acknowledge cycle, this signal gates the interrupt vector onto the bus if the SCC is the highest priority device requesting an interrupt.

$\overline{\text{WR}}$. *Write* (input, active Low). When the SCC is selected, this signal indicates a write operation. The coincidence of $\overline{\text{RD}}$ and $\overline{\text{WR}}$ is interpreted as a reset.

Z8030

AD_0-AD_7. *Address/Data Bus* (bidirectional, active High, 3-state). These multiplexed lines carry register addresses to the SCC as well as data or control information.

$\overline{\text{AS}}$. *Address Strobe* (input, active Low). Addresses on AD_0-AD_7 are latched by the rising edge of this signal.

$\overline{\text{CS}}_0$. *Chip Select 0* (input, active Low). This signal is latched concurrently with the addresses on AD_0-AD_7 and must be active for the intended bus transaction to occur.

CS_1. *Chip Select 1* (input, active High). This second select signal must also be active before the intended bus transaction can occur. CS_1 must remain active throughout the transaction.

$\overline{\text{DS}}$. *Data Strobe* (input, active Low). This signal provides timing for the transfer of data into and out of the SCC. If $\overline{\text{AS}}$ and $\overline{\text{DS}}$ coincide, this is interpreted as a reset.

R/$\overline{\text{W}}$. *Read/Write* (input). This signal specifies whether the operation to be performed is a read or a write.

Functional Description

The functional capabilities of the SCC can be 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 SCC offers valuable features such as vectored interrupts, polling, and simple handshake capability.

Data Communications Capabilities. The SCC provides two independent full-duplex channels programmable for use in any common Asynchronous or Synchronous data-communication protocol. Figure 3 and the following description briefly detail these protocols.

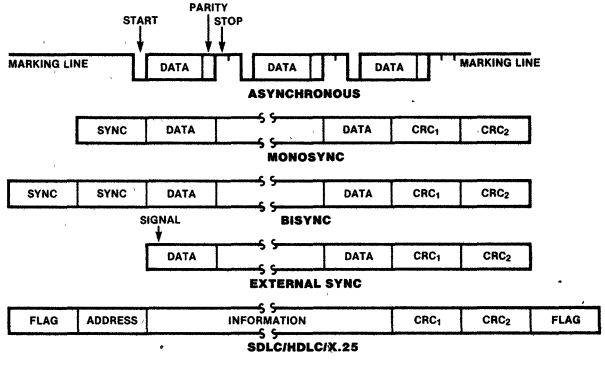


Figure 3. Some SCC Protocols

Functional Description (Continued)

Asynchronous Modes. Transmission and reception can be accomplished independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one-and-a-half, or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break. Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxDA or RxDB in Figure 1). If the Low does not persist (as in the case of a transient), the character assembly process does not start.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occur. Vectored interrupts allow fast servicing or error conditions using dedicated routines. Furthermore, a built-in checking process avoids the interpretation of a framing error as a new start bit: a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit begins.

The SCC does not require symmetric transmit and receive clock signals—a feature allowing use of the wide variety of clock sources. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32, or 1/64 of the clock rate supplied to the receive and transmit clock inputs. In Asynchronous modes, the $\overline{\text{SYNC}}$ pin may be programmed as an input used for functions such as monitoring a ring indicator.

Synchronous Modes. The SCC supports both byte-oriented and bit-oriented synchronous communication. Synchronous byte-oriented protocols can be handled in several modes, allowing character synchronization with a 6-bit or 8-bit synchronous character (Monosync), any 12-bit synchronization pattern (Bisync), or with an external synchronous signal. Leading sync characters can be removed without interrupting the CPU.

Five- or 7-bit synchronous characters are detected with 8- or 16-bit patterns in the SCC by overlapping the larger pattern across multiple incoming synchronous characters as shown in Figure 4.

CRC checking for Synchronous byte-oriented modes is delayed by one character time so that the CPU may disable CRC checking on specific characters. This permits the implementation of protocols such as IBM Bisync.

Both CRC-16 ($X^{16} + X^{15} + X^2 + 1$) and CCITT ($X^{16} + X^{12} + X^5 + 1$) error checking polynomials are supported. Either polynomial may be selected in all Synchronous modes. Users may preset the CRC generator and checker to all 1s or all 0s. The SCC also provides a feature that automatically transmits CRC data when no other data is available for transmission. This allows for high speed transmissions under DMA control, with no need for CPU intervention at the end of a message. When there is no data or CRC to send in Synchronous modes, the transmitter inserts 6-, 8-, or 16-bit synchronous characters, regardless of the programmed character length.

The SCC supports Synchronous bit-oriented protocols, such as SDLC and HDLC, by performing automatic flag sending, zero insertion, and CRC generation. A special command can be used to abort a frame in transmission. At the end of a message, the SCC automatically transmits the CRC and trailing flag when the transmitter underruns. The transmitter may also be programmed to send an idle line consisting of continuous flag characters or a steady marking condition.

If a transmit underrun occurs in the middle of a message, an external/status interrupt warns the CPU of this status change so that an abort may be issued. The SCC may also be programmed to send an abort itself in case of an underrun, relieving the CPU of this task. One to eight bits per character can be sent, allowing reception of a message with no prior information about the character structure in the information field of a frame.

The receiver automatically acquires synchronization on the leading flag of a frame in SDLC or HDLC and provides a synchronization signal on the $\overline{\text{SYNC}}$ pin (an interrupt can also be programmed). The receiver can be programmed to search for frames addressed by a single byte (or four bits within a byte) of a user-selected address or to a global broadcast address. In this mode, frames not matching either the user-selected or broadcast address are ignored. The number of address bytes can be extended under software control. For receiving data, an interrupt on the first received character, or an interrupt on every character, or on special condition only (end-of-frame) can be selected. The receiver automatically deletes all 0s inserted by the transmitter during character assembly. CRC is also calculated and is automatically checked to validate frame transmission. At the end of

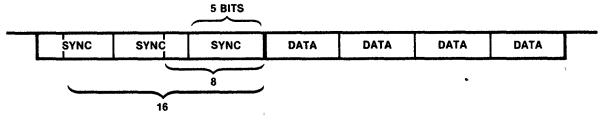


Figure 4. Detecting 5- or 7-Bit Synchronous Characters

Functional Description (Continued)

transmission, the status of a received frame is available in the status registers. In SDLC mode, the SCC must be programmed to use the SDLC CRC polynomial, but the generator and checker may be preset to all 1s or all 0s. The CRC is inverted before transmission and the receiver checks against the bit pattern 0001110100001111.

NRZ, NRZI or FM coding may be used in any 1x mode. The parity options available in Asynchronous modes are available in Synchronous modes.

The SCC can be conveniently used under DMA control to provide high speed reception or transmission. In reception, for example, the SCC can interrupt the CPU when the first character of a message is received. The CPU then enables the DMA to transfer the message to memory. The SCC then issues an end-of-frame interrupt and the CPU can check the status of the received message. Thus, the CPU is freed for other service while the message is being received. The CPU may also enable the DMA first and have the SCC interrupt only on end-of-frame. This procedure allows all data to be transferred via the DMA.

SDLC Loop Mode. The SCC supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, there is a primary controller station that manages the message traffic flow on the loop and any number of secondary stations. In SDLC Loop mode, the SCC performs the functions of a secondary station while an SCC operating in regular SDLC mode can act as a controller (Figure 5).

A secondary station in an SDLC Loop is always listening to the messages being sent around the loop, and in fact must pass these messages to the rest of the loop by retransmitting them with a one-bit-time delay. The secondary station can place its own message on the loop only at specific times. The controller signals that secondary stations may transmit messages by sending a special character, called an EOP (End Of Poll), around the loop. The EOP character is the bit pattern 11111110. Because of zero insertion during messages, this bit pattern is unique and easily recognized.

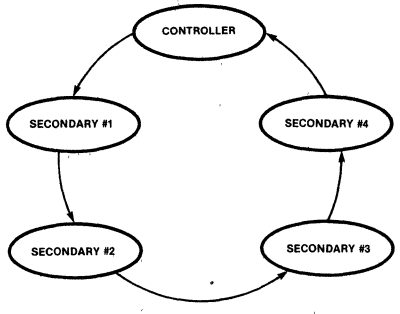


Figure 5. An SDLC Loop

When a secondary station has a message to transmit and recognizes an EOP on the line, it changes the last binary 1 of the EOP to a 0 before transmission. This has the effect of turning the EOP into a flag sequence. The secondary station now places its message on the loop and terminates the message with an EOP. Any secondary stations further down the loop with messages to transmit can then append their messages to the message of the first secondary station by the same process. Any secondary stations without messages to send merely echo the incoming messages and are prohibited from placing messages on the loop (except upon recognizing an EOP).

SDLC Loop mode is a programmable option in the SCC. NRZ, NRZI, and FM coding may all be used in SDLC Loop mode.

Baud Rate Generator. Each channel in the SCC contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On startup, the flip-flop on the output is set in a High state, the value in the time constant register is loaded into the counter, and the counter starts counting down. The output of the baud rate generator toggles upon reaching 0, the value in the time constant register is loaded into the counter, and the process is repeated. The time constant may be changed at any time, but the new value does not take effect until the next load of the counter.

The output of the baud rate generator may be used as either the transmit clock, the receive clock, or both. It can also drive the Digital Phase-Locked Loop (see next section).

If the receive clock or transmit clock is not programmed to come from the TRxC pin, the output of the baud rate generator may be echoed out via the TRxC pin.

The following formula relates the time constant to the baud rate where PCLK or RTxC is the baud rate generator input frequency in Hz. The clock mode is 1, 16, 32, or 64 as selected in Write Register 4, bits D6 and D7. Synchronous operation modes should select 1 and Asynchronous should select 16, 32, or 64.

$$\text{Time Constant} = \frac{\text{PCLK or RTxC Frequency}}{2\,(\text{Baud Rate})\,(\text{Clock Mode})} - 2$$

Digital Phase-Locked Loop. The SCC contains a Digital Phase-Locked-Loop (DPLL) to recover clock information from a data stream with NRZI or FM encoding. The DPLL is driven by a clock that is nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock may then be used as the SCC receive clock, the transmit clock, or both.

For NRZI encoding, the DPLL counts the 32x clock to create nominal bit times. As the 32x clock is counted, the DPLL is searching the

Functional Description (Continued)

incoming data stream for edges (either 1 to 0 or 0 to 1). Whenever an edge is detected, the DPLL makes a count adjustment (during the next counting cycle), producing a terminal count closer to the center of the bit cell.

For FM encoding, the DPLL still counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream should occur between counts 15 and 16 and between counts 31 and 0. The DPLL looks for edges only during a time centered on the 15 to 16 counting transition.

The 32x clock for the DPLL can be programmed to come from either the $\overline{\text{RTxC}}$ input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the SCC via the $\overline{\text{TRxC}}$ pin (if this pin is not being used as an input).

Data Encoding. The SCC may be programmed to encode and decode the serial data in four different ways (Figure 6). In NRZ encoding, a 1 is represented by a High level and a 0 is represented by a Low level. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level. In FM1 (more properly, bi-phase mark), a transition occurs at the beginning of every bit cell. A 1 is represented by an additional transition at the center of the bit cell and a 0 is represented by no additional transition at the center of the bit cell. In FM0 (bi-phase space), a transition occurs at the beginning of every bit cell. A 0 is represented by an additional transition at the center of the bit cell, and a 1 is represented by no additional transition at the center of the bit cell. In addition to these four methods, the SCC can be used to decode Manchester (bi-phase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0 to 1, the bit is a 0. If the transition is 1 to 0, the bit is a 1.

Auto Echo and Local Loopback. The SCC is capable of automatically echoing everything it receives. This feature is useful mainly in Asynchronous modes, but works in Synchronous and SDLC modes as well. In Auto Echo mode, TxD is RxD. Auto Echo mode can be used with NRZI or FM encoding with no additional delay, because the data stream is not decoded before retransmission. In Auto Echo mode, the $\overline{\text{CTS}}$ input is ignored as a transmitter enable (although transitions on this input can still cause interrupts if programmed to do so). In this mode, the transmitter is actually bypassed and the programmer is responsible for disabling transmitter interrupts and $\overline{\text{WAIT/REQUEST}}$ on transmit.

The SCC is also capable of local loopback. In this mode TxD is RxD, just as in Auto Echo mode. However, in Local Loopback mode, the internal transmit data is tied to the internal receive data and RxD is ignored (except to be echoed out via TxD). The $\overline{\text{CTS}}$ and $\overline{\text{DCD}}$ inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local Loopback works in Asynchronous, Synchronous and SDLC modes with NRZ, NRZI or FM coding of the data stream.

I/O Interface Capabilities. The SCC offers the choice of Polling, Interrupt (vectored or nonvectored), and Block Transfer modes to transfer data, status, and control information to and from the CPU. The Block Transfer mode can be implemented under CPU or DMA control.

Polling. All interrupts are disabled. Three status registers in the SCC are automatically updated whenever any function is performed. For example, end-of-frame in SDLC mode sets a bit in one of these status registers. The idea behind polling is for the CPU to periodically read a status register until the register contents indicate the need for data to be transferred. Only one register needs to be

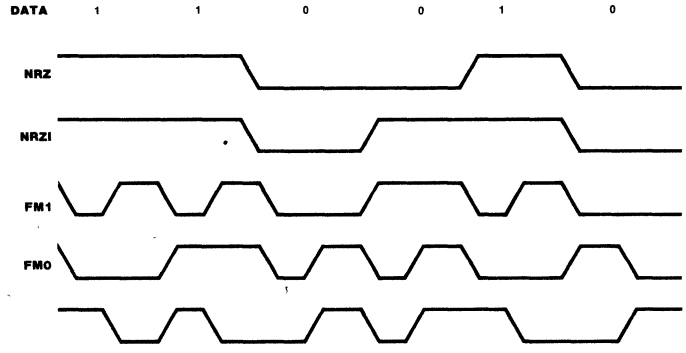


Figure 6. Data Encoding Methods

Functional Description (Continued)

read; depending on its contents, the CPU either writes data, reads data, or continues. Two bits in the register indicate the need for data transfer. An alternative is a poll of the Interrupt Pending register to determine the source of an interrupt. The status for both channels resides in one register.

Interrupts. When an SCC responds to an Interrupt Acknowledge signal ($\overline{\text{INTACK}}$) from the CPU, an interrupt vector may be placed on the data bus. This vector is written in WR2 and may be read in RR2A or RR2B (Figures 10 and 11).

To speed interrupt response time, the SCC can modify three bits in this vector to indicate status. If the vector is read in Channel A, status is never included; if it is read in Channel B, status is always included.

Each of the six sources of interrupts in the SCC (Transmit, Receive, and External/Status interrupts in both channels) has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). Operation of the IE bit is straightforward. If the IE bit is set for a given interrupt source, then that source can request interrupts. The exception is when the MIE (Master Interrupt Enable) bit in WR9 is reset and no interrupts may be requested. The IE bits are write only.

The other two bits are related to the interrupt priority chain (Figure 7). As a microprocessor peripheral, the SCC may request an interrupt only when no higher priority device is requesting one, e.g., when IEI is High. If the device in question requests an interrupt, it pulls down $\overline{\text{INT}}$. The CPU then responds with $\overline{\text{INTACK}}$, and the interrupting device places the vector on the data bus.

In the SCC, the IP bit signals a need for interrupt servicing. When an IP bit is 1 and the IEI input is High, the $\overline{\text{INT}}$ output is pulled Low, requesting an interrupt. In the SCC, if the IE bit is not set by enabling interrupts, then the IP for that source can never be set. The IP bits are readable in RR3A.

The IUS bits signal that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority in the SCC and external to the SCC 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 SCC 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 three types of interrupts: Transmit, Receive, and External/Status. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receiver, Transmit, and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted when the transmit buffer becomes empty. (This implies that the transmitter must have had a data character written into it so that it can become empty.) When enabled, the receiver can interrupt the CPU in one of three ways:

- Interrupt on First Receive Character or Special Receive Condition.
- Interrupt on All Receive Characters or Special Receive Condition.
- Interrupt on Special Receive Condition Only.

Interrupt on First Character or Special Condition and Interrupt on Special Condition Only are typically used with the Block Transfer mode. A Special Receive Condition is one of the following: receiver overrun, framing error in Asynchronous mode, end-of-frame in SDLC mode and, optionally, a parity error. The Special Receive Condition interrupt is different from an ordinary receive character available interrupt only in the status placed in the vector during the Interrupt Acknowledge cycle. In Interrupt on First Receive Character, an interrupt can occur from Special Receive Conditions any time after the first receive character interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the $\overline{\text{CTS}}$, $\overline{\text{DCD}}$, and $\overline{\text{SYNC}}$ pins; however, an

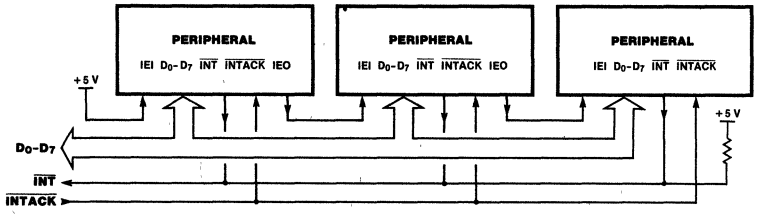


Figure 7. Interrupt Schedule

Functional Description (Continued)

External/Status interrupt is also caused by a Transmit Underrun condition, or a zero count in the baud rate generator, or by the detection of a Break (Asynchronous mode), Abort (SDLC mode) or EOP (SDLC Loop mode) sequence in the data stream. The interrupt caused by the Abort or EOP has a special feature allowing the SCC to interrupt when the Abort or EOP sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Abort condition in external logic in SDLC mode. In SDLC Loop mode, this feature allows secondary stations to recognize the wishes of the primary station to regain control of the loop during a poll sequence.

CPU/DMA Block Transfer. The SCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers. The Block Transfer mode uses the $\overline{\text{WAIT}}$/$\overline{\text{REQUEST}}$ output in conjunction with the Wait/Request bits in WR1. The $\overline{\text{WAIT}}$/$\overline{\text{REQUEST}}$ output can be defined under software control as a $\overline{\text{WAIT}}$ line in the CPU Block Transfer mode or as a $\overline{\text{REQUEST}}$ line in the DMA Block Transfer mode.

To a DMA controller, the SCC $\overline{\text{REQUEST}}$ output indicates that the SCC is ready to transfer data to or from memory. To the CPU, the $\overline{\text{WAIT}}$ line indicates that the SCC is not ready to transfer data, thereby requesting that the CPU extend the I/O cycle. The $\overline{\text{DTR}}$/$\overline{\text{REQUEST}}$ line allows full-duplex operation under DMA control.

Architecture

The SCC internal structure includes two full-duplex channels, two baud rate generators, internal control and interrupt logic, and a bus interface to a nonmultiplexed bus. Associated with each channel are a number of read and write registers for mode control and status information, as well as logic necessary to interface to modems or other external devices (Figure 8).

The logic for both channels provides formats, synchronization, and validation for data transferred to and from the channel interface. The modem control inputs are monitored by the control logic under program control. All of the modem control signals are general-purpose in nature and can optionally be used for functions other than modem control.

The register set for each channel includes ten control (write) registers, two sync-character (write) registers, and four status (read) registers. In addition, each baud rate generator has two (read/write) registers for holding the time constant that determines the baud rate. Finally, associated with the interrupt logic is a write register for the interrupt vector accessible through either channel, a

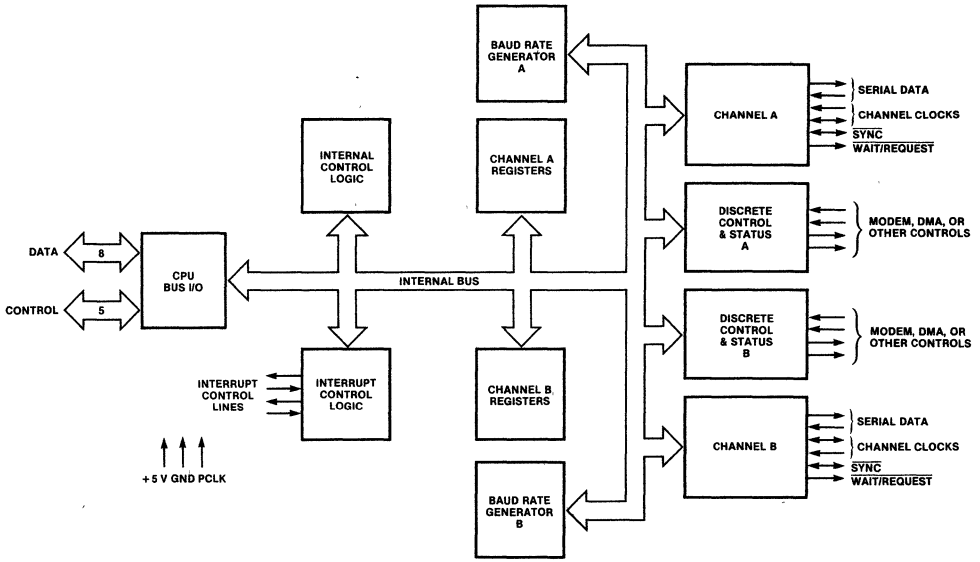


Figure 8. Block Diagram of SCC Architecture

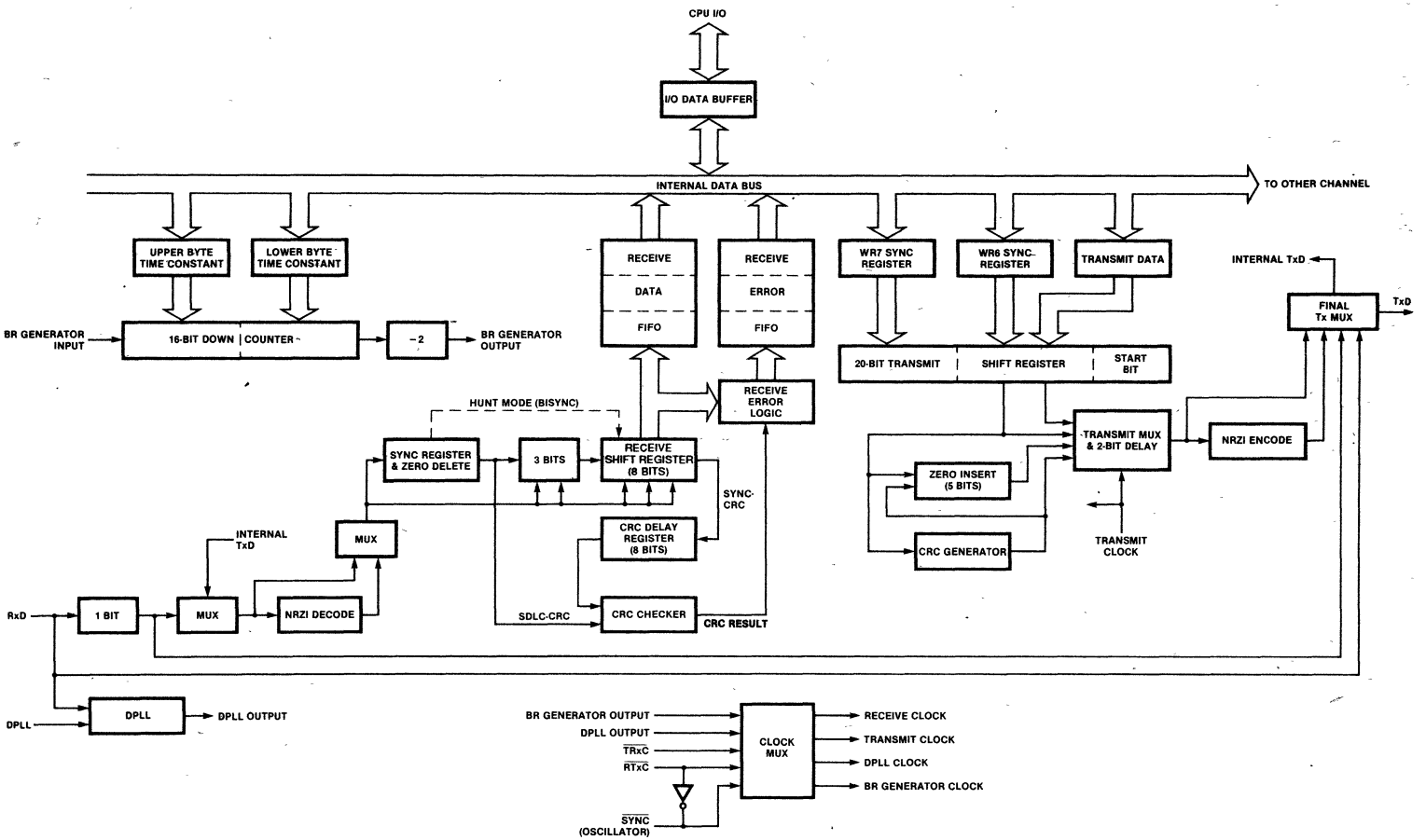


Figure 9. Data Path

Architecture (Continued)

write only Master Interrupt Control register and three read registers: one containing the vector with status infomation (Channel B only), one containing the vector without status (Channel A only), and one containing the Interrupt Pending bits (Channel A only).

The registers for each channel are designated as follows:

WR0–WR15 — Write Registers 0 through 15.

RR0–RR3, RR10, RR12, RR13, RR15 — Read Registers 0 through 3, 10, 12, 13, 15.

Table 1 lists the functions assigned to each read or write register. The SCC contains only one WR2 and WR9, but they can be accessed by either channel. All other registers are paired (one for each channel).

Data Path. The transmit and receive data path illustrated in Figure 9 is identical for both channels. The receiver has three 8-bit buffer registers in an FIFO arrangement, in addition to the 8-bit receive shift register. This scheme creates additional time for the CPU to service an interrupt at the beginning of a block of high speed data. Incoming data is routed through one of several paths (data or CRC) depending on the selected mode (the character length in Asynchronous modes also determines the data path).

The transmitter has an 8-bit Transmit Data buffer register loaded from the internal data bus and a 20-bit Transmit Shift register that can be loaded either from the synchronous character registers or from the Transmit Data register. Depending on the operational mode, outgoing data is routed through one of four main paths before it is transmitted from the Transmit Data output (TxD)

Read Register Functions	
RR0	Transmit/Receive buffer status and External status
RR1	Special Receive Condition status
RR2	Modified interrupt vector (Channel B only) Unmodified interrupt vector (Channel A only)
RR3	Interrupt Pending bits (Channel A only)
RR8	Receive buffer
RR10	Miscellaneous status
RR12	Lower byte of baud rate generator time constant
RR13	Upper byte of baud rate generator time constant
RR15	External/Status interrupt information
Write Register Functions	
WR0	CRC initialize, initialization commands for the various modes, Register Pointers
WR1	Transmit/Receive interrupt and data transfer mode definition
WR2	Interrupt vector (accessed through either channel)
WR3	Receive parameters and control
WR4	Transmit/Receive miscellaneous parameters and modes
WR5	Transmit parameters and controls
WR6	Sync characters or SDLC address field
WR7	Sync character or SDLC flag
WR8	Transmit buffer
WR9	Master interrupt control and reset (accessed through either channel)
WR10	Miscellaneous transmitter/receiver control bits
WR11	Clock mode control
WR12	Lower byte of baud rate generator time constant
WR13	Upper byte of baud rate generator time constant
WR14	Miscellaneous control bits
WR15	External/Status interrupt control

Table 1. Read and Write Register Functions

Programming

The SCC contains write registers in each channel that are programmed by the system separately to configure the functional personality of the channels.

Z8530

In the SCC, register addressing is direct for the data registers only, which are selected by a High on the D/C pin. In all other cases (with the exception of WR0 and RR0), programming the write registers requires two write operations and reading the read registers requires both a write and a read operation. The first write is to WR0 and contains three bits that point to the selected register. The second write is the actual control word for the selected register, and if the second operation is read, the selected read register is accessed. All of the registers in the SCC, including the data registers, may be accessed in this fashion. The pointer bits are automatically cleared after the read or write operation so that WR0 (or RR0) is addressed again.

Z8030

All SCC registers are directly addressable. How the SCC decodes the address placed on the address/data bus at the beginning of a Read or Write cycle is controlled by a command issued in WROB. In the Shift Right mode the channel select A/B is taken from AD_0 and the state of AD_5 is ignored. In the Shift Left mode the channel select A/B is taken from AD_5 and the state of AD_0 is ignored. AD_7 and AD_6 are always ignored as address bits and the register address itself occupies AD_4–AD_1.

Z8530/Z8030

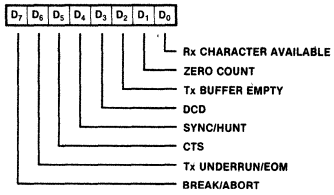
The system program first issues a series of commands to initialize the basic mode of operation. This is followed by other commands to qualify conditions within the selected mode. For example, the Asynchronous mode, character length, clock rate, number of stop bits, even or odd parity might be set first. Then the interrupt mode would be set, and finally, receiver or transmitter enable.

Programming (Continued)

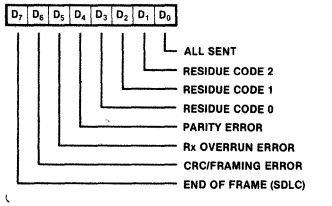
Read Registers. The SCC contains eight read registers (actually nine, counting the receive buffer (RR8) in each channel). Four of these may be read to obtain status information (RR0, RR1, RR10, and RR15). Two registers (RR12 and RR13) may be read to learn the baud rate generator time constant. RR2 contains either the unmodified interrupt vector (Channel A) or the vector modified by status information (Channel B). RR3 contains the Interrupt Pending (IP) bits (Channel A). Figure 10 shows the formats for each read register.

The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring; e.g., when the interrupt vector indicates a Special Receive Condition interrupt, all the appropriate error bits can be read from a single register (RR1).

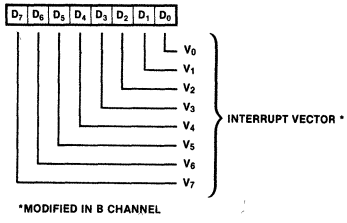
Read Register 0



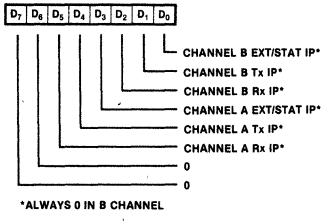
Read Register 1



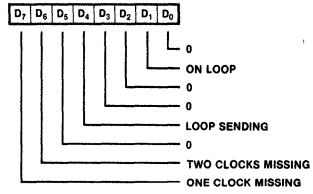
Read Register 2



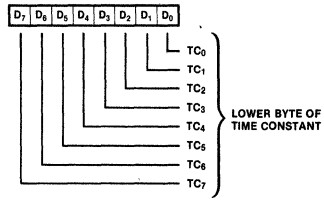
Read Register 3



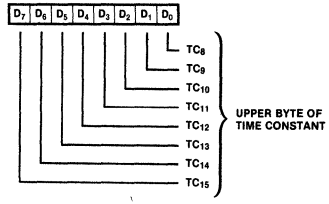
Read Register 10



Read Register 12



Read Register 13



Read Register 15

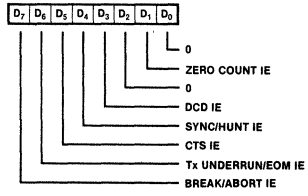
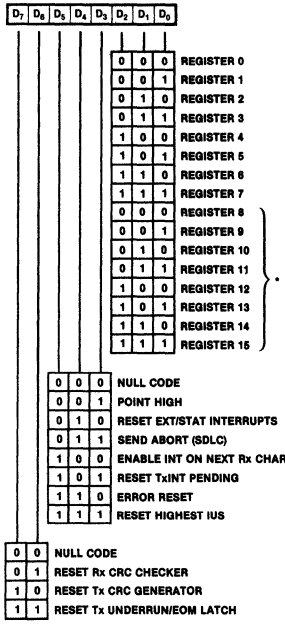


Figure 10. Read Register Bit Functions

Write Registers. The SCC contains 13 write registers (14 counting WR8, the transmit buffer) in each channel. These write registers are programmed separately to configure the functional "personality" of the channels. In addition, there are two registers (WR2 and WR9) shared by the two channels that may be accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits. Figure 11 shows the format of each write register.

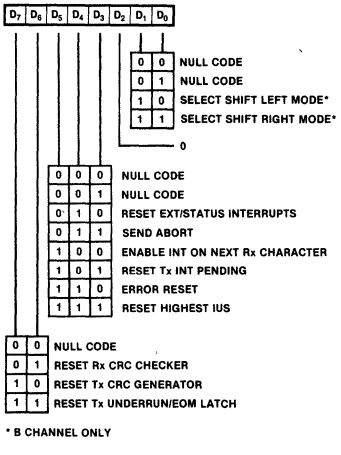
Programming
(Continued)

Write Register 0 (Z8530)



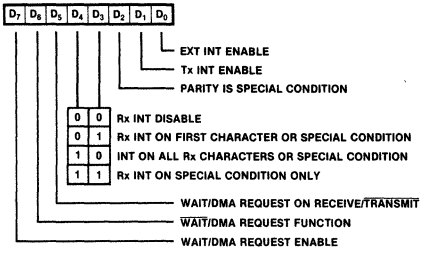
*WITH POINT HIGH COMMAND

Write Register 0 (Z8030)

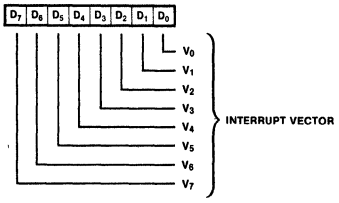


* B CHANNEL ONLY

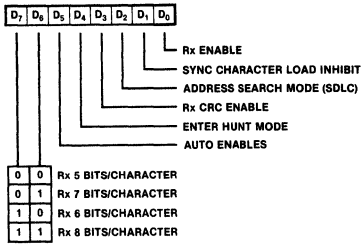
Write Register 1



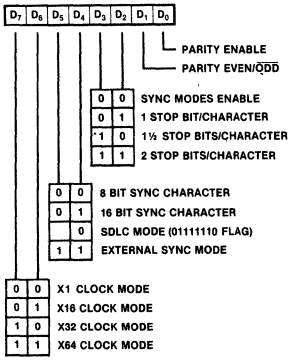
Write Register 2



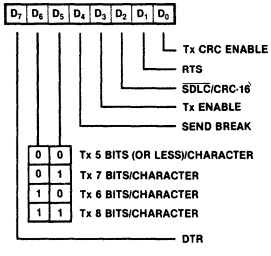
Write Register 3



Write Register 4



Write Register 5



Write Register 6

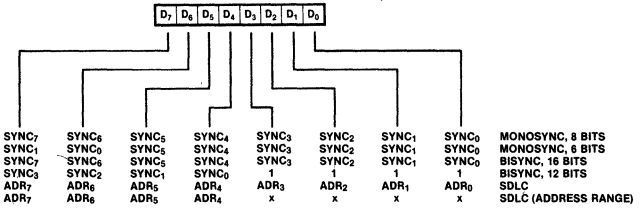
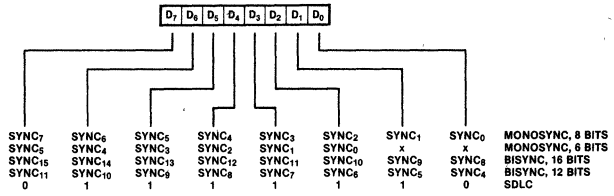
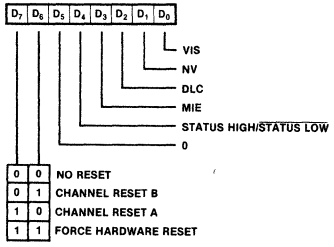


Figure 11. Write Register Bit Functions

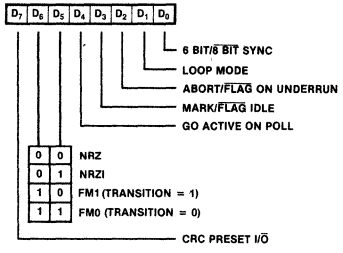
Write Register 7



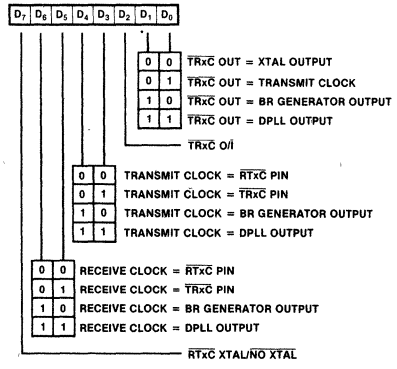
Write Register 9



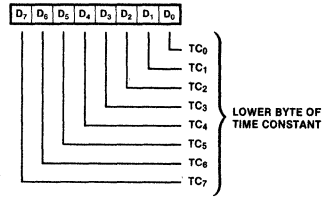
Write Register 10



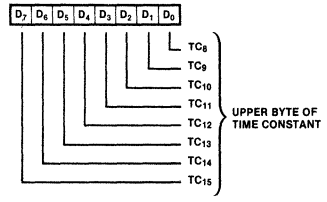
Write Register 11



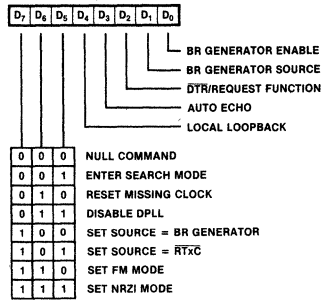
Write Register 12



Write Register 13



Write Register 14



Write Register 15

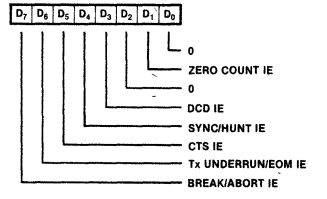


Figure 11. Write Register Bit Functions (Continued)

Z8530 Timing

The SCC generates internal control signals from $\overline{WR}$ and $\overline{RD}$ that are related to PCLK. Since PCLK has no phase relationship with WR and RD, the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the SCC. The recovery time required for proper operation is specified from the falling edge of $\overline{WR}$ or $\overline{RD}$ in the first transaction involving the SCC to the falling edge of $\overline{WR}$ or $\overline{RD}$ in the second transaction involving the SCC. This time must be at least 4 PCLK regardless of which register or channel is being accessed.

Read Cycle Timing. Figure 12 illustrates Read cycle timing. Addresses on A/B and D/C and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{RD}$ falls or if it rises before $\overline{RD}$ rises, the effective $\overline{RD}$ is shortened.

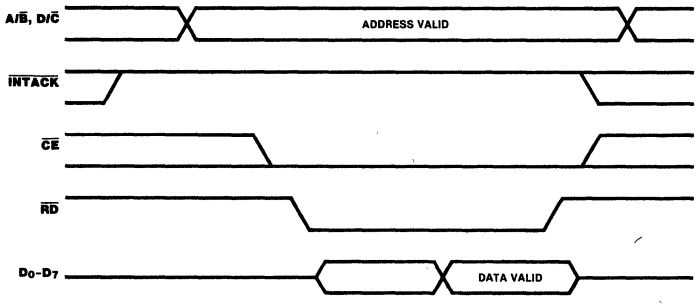


Figure 12. Read Cycle Timing

Write Cycle Timing. Figure 13 illustrates Write cycle timing. Addresses on A/B and D/C and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{WR}$ falls or if it rises before $\overline{WR}$ rises, the effective $\overline{WR}$ is shortened. Data must be valid before the falling edge of $\overline{WR}$.

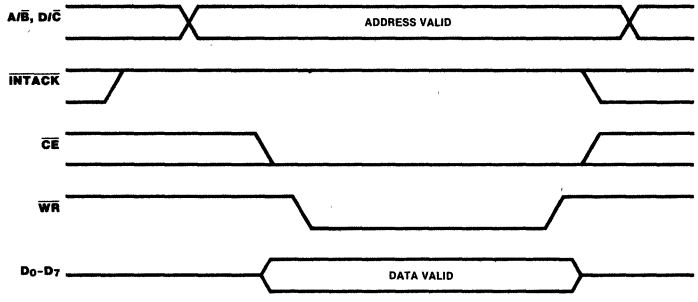


Figure 13. Write Cycle Timing

Interrupt Acknowledge Cycle Timing. Figure 14 illustrates Interrupt Acknowledge cycle timing. Between the time $\overline{INTACK}$ goes Low and the falling edge of $\overline{RD}$, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the SCC and IEI is High when $\overline{RD}$ falls, the Acknowledge cycle is intended for the SCC. In this case, the SCC may be programmed to respond to $\overline{RD}$ Low by placing its interrupt vector on D_0–D_7 and it then sets the appropriate Interrupt-UnderService latch internally.

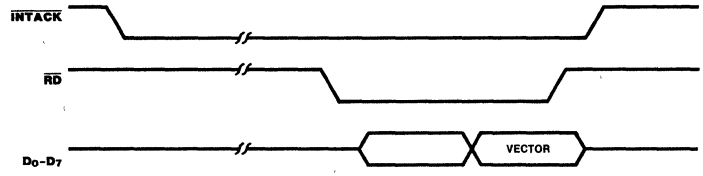


Figure 14. Interrupt Acknowledge Cycle Timing

Z8030 Timing

The SCC generates internal control signals from $\overline{AS}$ and $\overline{DS}$ that are related to PCLK. Since PCLK has no phase relationship with $\overline{AS}$ and $\overline{DS}$, the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the SCC. The recovery time required for proper operation is specified from the falling edge of $\overline{DS}$ in the first transaction involving the SCC to the falling edge of $\overline{DS}$ in the second transaction involving the SCC.

Read Cycle Timing. Figure 15 illustrates Read cycle timing. The address on AD_0–AD_7 and the state of $\overline{CS_0}$ and $\overline{INTACK}$ are latched by the rising edge of $\overline{AS}$. R/$\overline{W}$ must be High to indicate a Read cycle. CS_1 must also be High for the Read cycle to occur. The data bus drivers in the SCC are then enabled while $\overline{DS}$ is Low.

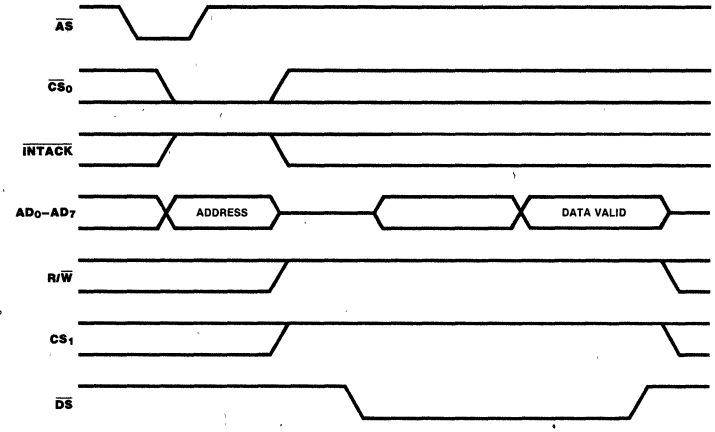


Figure 15. Read Cycle Timing

Write Cycle Timing. Figure 16 illustrates Write cycle timing. The address on AD_0–AD_7 and the state of $\overline{CS_0}$ and $\overline{INTACK}$ are latched by the rising edge of $\overline{AS}$. R/$\overline{W}$ must be Low to indicate a Write cycle. CS_1 must be High for the Write cycle to occur. $\overline{DS}$ Low strobes the data into the SCC.

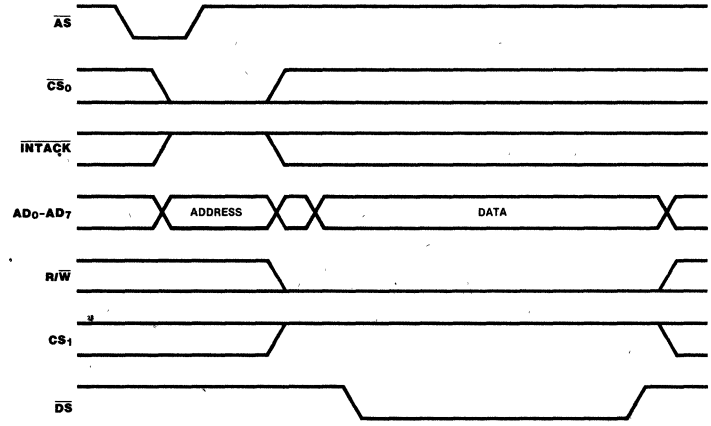


Figure 16. Write Cycle Timing

Interrupt Acknowledge Cycle Timing. Figure 17 illustrates Interrupt Acknowledge cycle timing. The address on AD_0–AD_7 and the state of $\overline{CS_0}$ and $\overline{INTACK}$ are latched by the rising edge of $\overline{AS}$. However, if $\overline{INTACK}$ is Low, the address and $\overline{CS_0}$ are ignored. The state of the $R/\overline{W}$ and CS_1 are also ignored for the duration of the Interrupt Acknowledge cycle. Between the rising edge of $\overline{AS}$ and the falling edge of $\overline{DS}$, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the SCC and IEI is High when $\overline{DS}$ falls, the Acknowledge cycle was intended for the SCC. In this case, the SCC may be programmed to respond to RD Low by placing its interrupt vector on D_0–D_7 and it then internally sets the appropriate Interrupt-Under-Service latch.

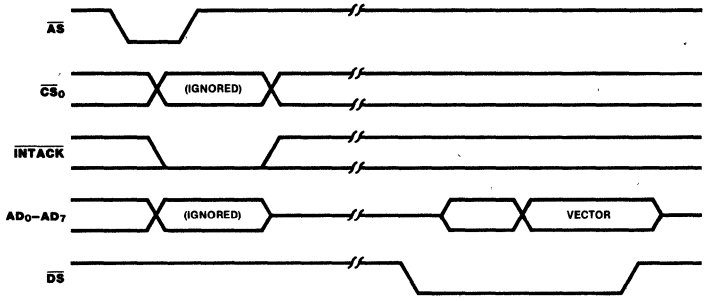


Figure 17. Interrupt Acknowledge Cycle Timing

Absolute Maximum Ratings

Voltages on all pins with respect to GND . . . . . −0.3V to +7.0V

Operating Ambient Temperature . . . . . See Ordering Information

Storage Temperature . . . . . −65°C to +150°C

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.

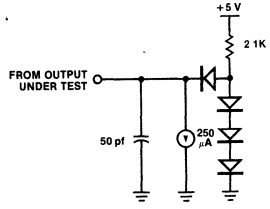
Standard Test Conditions

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.

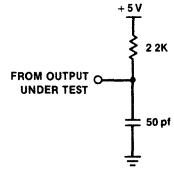
Standard conditions are as follows:

- $+4.75\ V \leq V_{CC} \leq +5.25\ V$
- GND = 0 V
- T_A as specified in Ordering Information

All ac parameters assume a load capacitance of 50 pF max.



Standard Test Load



Open-Drain Test Load

DC Characteristics

Symbol	Parameter	Min	Max	Unit	Condition
V_{IH}	Input High Voltage	2.0	$V_{CC}+0.3$	V	
V_{IL}	Input Low Voltage	-0.3	0.8	V	
V_{OH}	Output High Voltage	2.4		V	$I_{OH} = -250\ \mu A$
V_{OL}	Output Low Voltage		0.4	V	$I_{OL} = +2.0$ mA
I_{IL}	Input Leakage		±10.0	μA	$0.4 \leq V_{IN} \leq +2.4V$
I_{OL}	Output Leakage		±10.0	μA	$0.4 \leq V_{OUT} \leq +2.4V$
I_{CC}	V_{CC} Supply Current		250	mA	

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

Capacitance

Symbol	Parameter	Min	Max	Unit	Test Condition
C_{IN}	Input Capacitance		10	pF	Unmeasured Pins Returned to Ground
C_{OUT}	Output Capacitance		15	pF	
$C_{I/O}$	Bidirectional Capacitance		20	pF	

f = 1 MHz, over specified temperature range.
Unmeasured pins returned to ground.

Miscellaneous

Gate Count	6000	

Z8530 AC CHARACTERISTICS

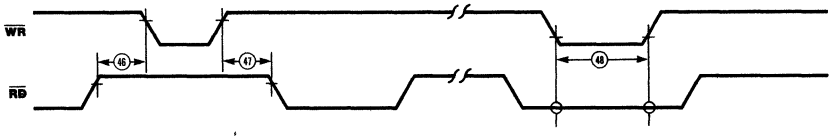
Number	Symbol	Parameter	4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	8 MHz Min	8 MHz Max	Notes †
1	TwPCl	PCLK Low Width	105	2000	70	1000	50	1000	
2	TwPCh	PCLK High Width	105	2000	70	1000	50	1000	
3	TfPC	PCLK Fall Time		20		10		10	
4	TrPC	PCLK Rise Time		20		10		10	
5	TcPC	PCLK Cycle Time	250	4000	165	2000	125	2000	
6	TsA(WR)	Address to $\overline{WR}$ ↓ Setup Time	80		80		70		
7	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		0		0		
8	TsA(RD)	Address to $\overline{RD}$ ↓ Setup Time	80		80		70		
9	ThA(RD)	Address to $\overline{RD}$ ↑ Hold Time	0		0		0		
10	TslA(PC)	$\overline{INTACK}$ to PCLK ↑ Setup Time	10		10		10		
11	TslAi(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time	200		160		145		1
12	ThIA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		0		0		
13	TslAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time	200		160		145		1
14	ThIA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		0		0		
15	ThIA(PC)	$\overline{INTACK}$ to PCLK ↑ Hold Time	100		100		85		
16	TsCEl(WR)	$\overline{CE}$ Low to $\overline{WR}$ ↓ Setup Time	0		0		0		
17	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		0		0		
18	TsCEh(WR)	$\overline{CE}$ High to $\overline{WR}$ ↓ Setup Time	100		70		60		
19	TsCEl(RD)	$\overline{CE}$ Low to $\overline{RD}$ ↓ Setup Time	0		0		0		1
20	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time	0		0		0		1
21	TsCEh(RD)	$\overline{CE}$ High to $\overline{RD}$ ↓ Setup Time	100		70		60		1
22	TwRDl	$\overline{RD}$ Low Width	240		200		150		1
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		0		0		
24	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		0		0		
25	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		250		180		140	
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay		70		45		40	2

NOTES
1. Parameter does not apply to Interrupt Acknowledge transactions
2. Float delay is defined as the time required for a ±0.5V change at the output with a maximum dc load and minimum ac load

†Units in nanoseconds (ns).

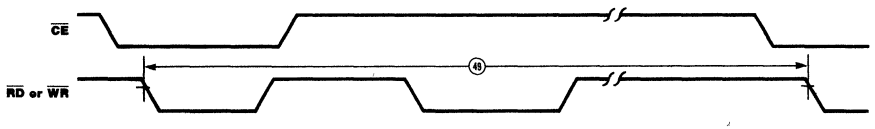
Reset Timing

Z8530



Cycle Timing

Z8530



Z8530 AC CHARACTERISTICS (Continued)

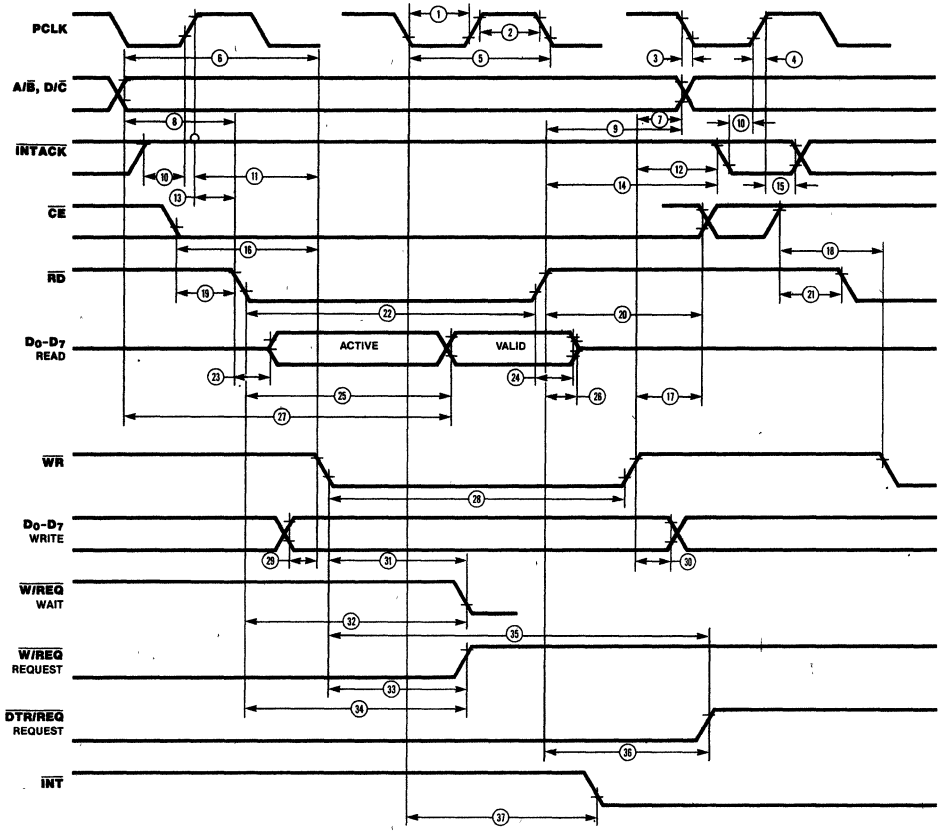
Number	Symbol	Parameter	4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	8 MHz Min	8 MHz Max	Notes †
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		300		280		220	
28	TwWRl	$\overline{WR}$ Low Width	240		200		150		
29	TsDW(WR)	Write Data to $\overline{WR}$ ↓ Setup Time	10		10		10		
30	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time	0		0		0		
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay		240		200		170	4
32	TdRD(W)	$\overline{RD}$ ↓ Wait Valid Delay		240		200		170	4
33	TdWRf(REQ)	$\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		240		200		170	
34	TdRDf(REQ)	$\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay		240		200		170	
35	TdWRr(REQ)	$\overline{WR}$ ↓ $\overline{DTR/REQ}$ Not Valid Delay		4TcPC		4TcPC		4TcPC	
36	TdRDr(REQ)	$\overline{RD}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay		4TcPC		4TcPC		4TcPC	
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay		500		500		500	4
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay	250		200		150		5
39	TwRDA	$\overline{RD}$ (Acknowledge) Width	250		200		150		
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		250		180		140	
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time	120		100		95		
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	0		0		0		
43	TdIEI(IEO)	IEI to IEO Delay Time		120		100		95	
44	TdPC(IEO)	PCLK ↑ to IEO Delay		250		250		200	
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay		500		500		450	4
46	TdRD(WRQ)	$\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset	30		15		15		
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset	30		30		20		
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident Low for Reset	250		200		150		
49	Trc	Valid Access Recovery Time	4TcPC		4TcPC		4TcPC		3

NOTES:

3. Parameter applies only between transactions involving the SCC.
4. Open-drain output, measured with open-drain test load.
5. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEIf(IEO) for each device separating them in the daisy chain.

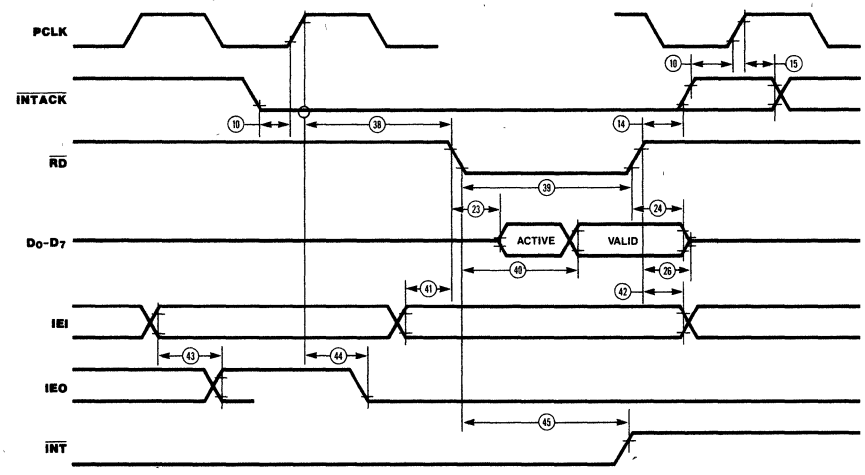
†Units in nanoseconds (ns).

Z8530



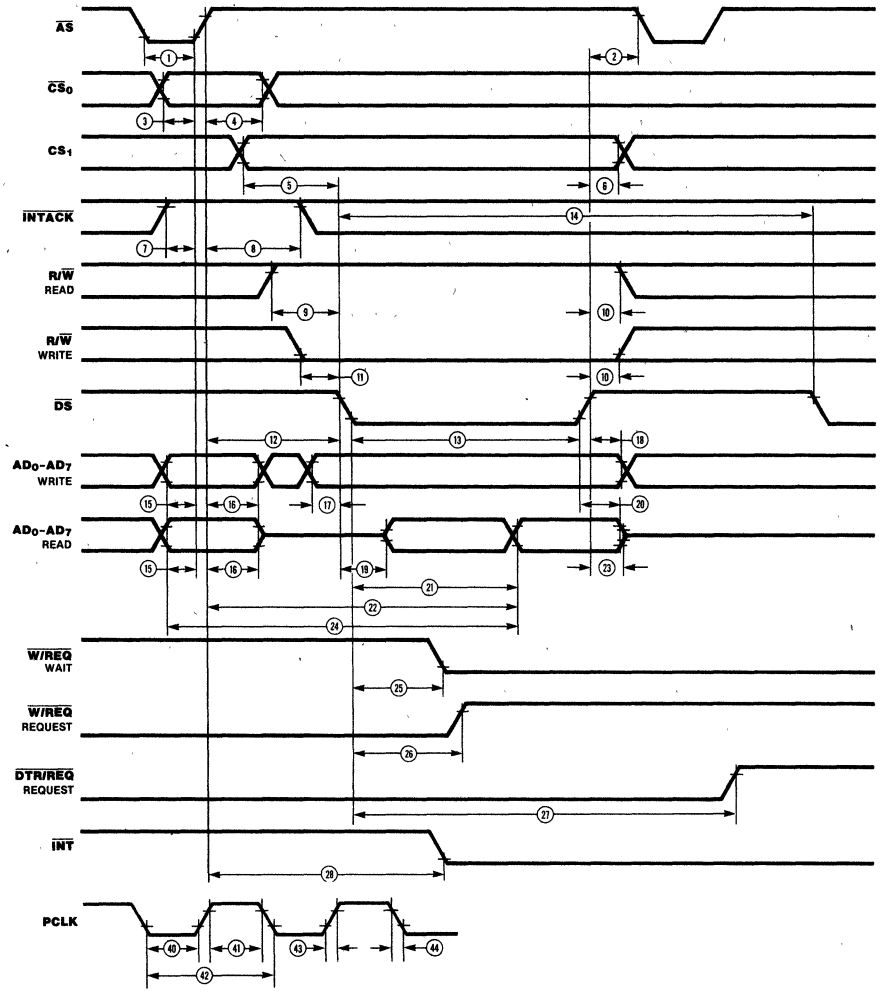
Interrupt Acknowledge Timing

Z8530



Read and Write Timing

Z8030



Z8030 AC CHARACTERISTICS

Number	Symbol	Parameter	4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	8 MHz Min	8 MHz Max	Notes †
1	TwAS	$\overline{AS}$ Low Width	70		50		35		
2	TdDS(AS)	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	50		25		15		
3	TsCSO(AS)	$\overline{CS_0}$ to $\overline{AS}$ ↑ Setup Time	0		0		0		1
4	ThCSO(AS)	$\overline{CS_0}$ to $\overline{AS}$ ↑ Hold Time	60		40		30		1
5	TsCS1(DS)	CS_1 to $\overline{DS}$ ↓ Setup Time	100		80		65		1
6	ThCS1(DS)	CS_1 to $\overline{DS}$ ↑ Hold Time	55		40		30		1
7	TsIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Setup Time	10		10		10		
8	ThIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Hold Time	250		200		150		
9	TsRWR(DS)	R/$\overline{W}$ (Read) to $\overline{DS}$ ↓ Setup Time	100		80		65		
10	ThRW(DS)	R/$\overline{W}$ to $\overline{DS}$ ↑ Hold Time	55		40		35		
11	TsRWW(DS)	R/$\overline{W}$ (Write) to $\overline{DS}$ ↓ Setup Time	0		0		0		
12	TdAS(DS)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ Delay	60		40		30		
13	TwDSl	$\overline{DS}$ Low Width	240		200		150		
14	TrC	Valid Access Recovery Time	4TcPC		4TcPC		4TcPC		2
15	TsA(AS)	Address to $\overline{AS}$ ↑ Setup Time	30		10		10		1
16	ThA(AS)	Address to $\overline{AS}$ ↑ Hold Time	50		30		25		1
17	TsDW(DS)	Write Data to $\overline{DS}$ ↓ Setup Time	30		20		15		
18	ThDW(DS)	Write Data to $\overline{DS}$ ↑ Hold Time	30		20		20		
19	TdDS(DA)	$\overline{DS}$ ↓ to Data Active Delay	0		0		0		
20	TdDSr(DR)	$\overline{DS}$ ↑ to Read Data Not Valid Delay	0		0		0		
21	TdDSf(DR)	$\overline{DS}$ ↓ to Read Data Valid Delay		250		180		140	
22	TdAS(DR)	$\overline{AS}$ ↑ to Read Data Valid Delay		520		300		250	

NOTES:
1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Parameter applies only between transactions involving the SCC

†Units in nanoseconds (ns).

Z8030 AC CHARACTERISTICS (Continued)

Number	Symbol	Parameter	4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	8 MHz Min	8 MHz Max	Notes †
23	TdDS(DRz)	$\overline{DS}$ ↑ to Read Data Float Delay		70		45		40	3
24	TdA(DR)	Address Required Valid to Read Data Valid Delay		570		310		260	
25	TdDS(W)	$\overline{DS}$ ↓ to Wait Valid Delay		240		200		170	4
26	TdDSf(REQ)	$\overline{DS}$ ↓ to W/$\overline{REQ}$ Not Valid Delay		240		200		170	
27	TdDSr(REQ)	$\overline{DS}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay		4TcPC		4TcPC		4TcPC	
28	TdAS(INT)	$\overline{AS}$ ↑ to $\overline{INT}$ Valid Delay		500		500		500	4
29	TdAS(DSA)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ (Acknowledge) Delay	250		250		250		5
30	TwDSA	$\overline{DS}$ (Acknowledge) Low Width	390		200		150		
31	TdDSA(DR)	$\overline{DS}$ ↓ (Acknowledge) to Read Data Valid Delay		250		180		140	
32	TsIEI(DSA)	IEI to $\overline{DS}$ ↓ (Acknowledge) Setup Time	120		100		80		
33	ThIEI(DSA)	IEI to $\overline{DS}$ ↑ (Acknowledge) Hold Time	0		0		0		
34	TdIEI(IEO)	IEI to IEO Delay		120		100		90	
35	TdAS(IEO)	$\overline{AS}$ ↑ to IEO Delay		250		250		200	6
36	TdDSA(INT)	$\overline{DS}$ ↓ (Acknowledge) to $\overline{INT}$ Inactive Delay		500		500		450	4
37	TdDS(ASQ)	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay for No Reset	30		15		15		
38	TdASQ(DS)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ Delay for No Reset	30		30		20		
39	TwRES	$\overline{AS}$ and $\overline{DS}$ Coincident Low for Reset	250		200		150		7
40	TwPCl	PCLK Low Width	105	2000	70	1000	50		
41	TwPCh	PCLK High Width	105	2000	70	1000	50		
42	TcPC	PCLK Cycle Time	250	4000	165	2000	125		
43	TrPC	PCLK Rise Time		20		10		10	
44	TfPC	PCLK Fall Time		20		10		10	

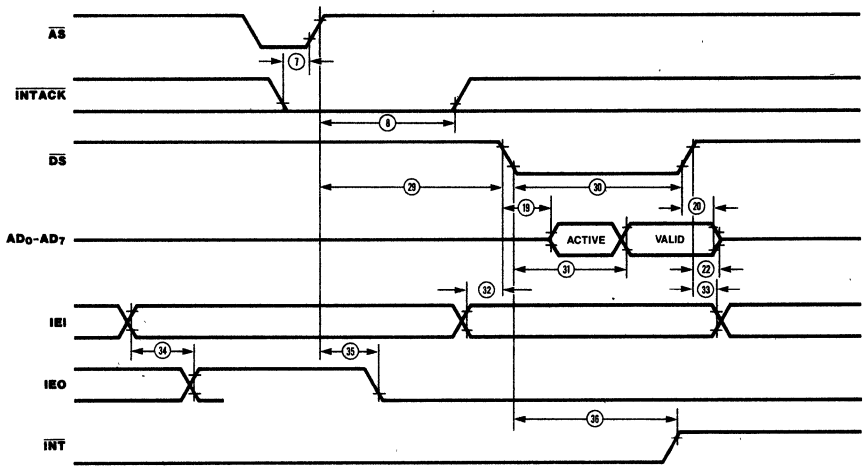
NOTES:

3. Float delay is defined as the time required for a ±0.5V change in the output with a maximum dc load and a minimum ac load.
4. Open-drain output, measured with open-drain test load.
5. Parameter is system dependent. For any Z-SCC in the daisy chain, TdAS(DSA) must be greater than the sum of TdAS(IEO) for the highest priority device in the daisy chain, TsIEI(DSA) for the Z-SCC, and TdIEIf(IEO) for each device separating them in the daisy chain.
6. Parameter applies only to a Z-SCC pulling INT Low at the beginning of the Interrupt Acknowledge transaction.
7. Internal circuitry allows for the reset provided by the Z8 to be recognized as a reset by the Z-SCC. All timing references assume 2.0V for a logic "1" and 0.8V for a logic "0".

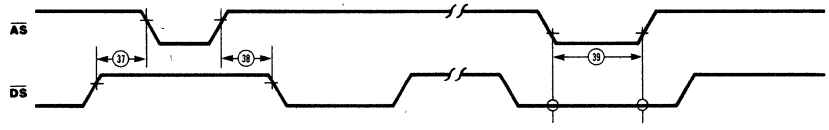
†Units in nanoseconds (ns).

Interrupt Acknowledge Timing

Z8030



Reset Timing



PRELIMINARY PRODUCT SPECIFICATION

Z80181

Z181 SAC
SMART ACCESS CONTROLLER

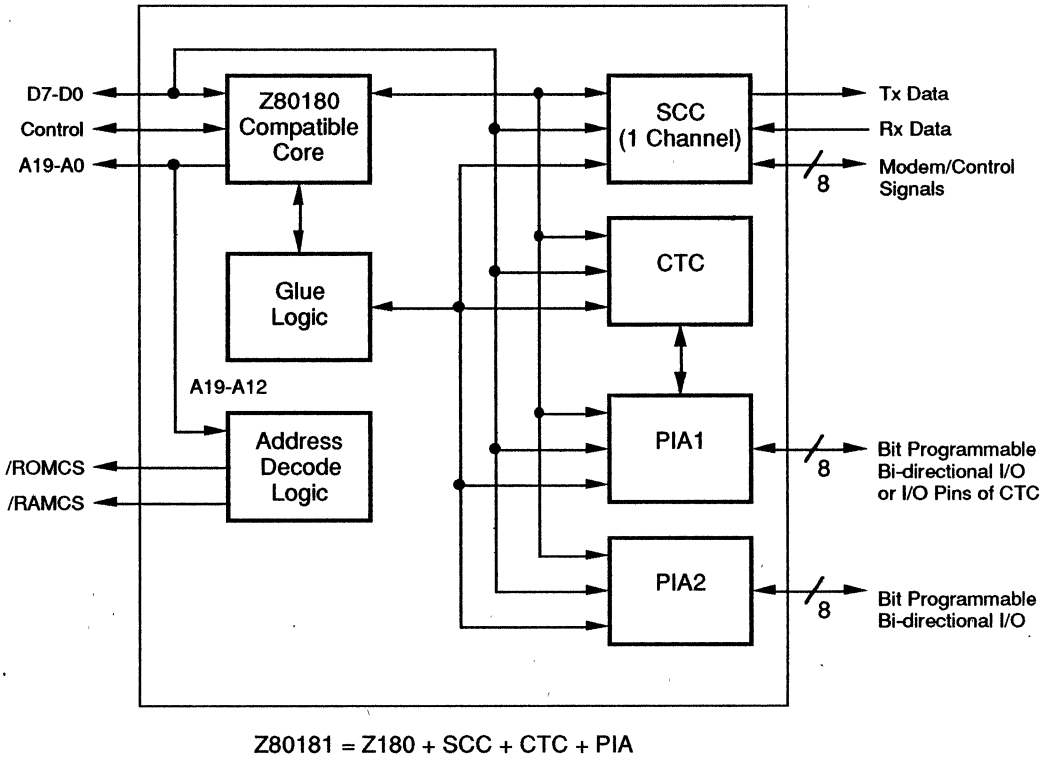
FEATURES

- Z80180 Compatible MPU Core with 1 channel of Z85C30 SCC, Z80 CTC, two 8-bit general purpose parallel ports, and two chip select signals.
- High speed operation (10/12.5 MHz)
- Low power consumption in two operating modes:
  (TBD) mA Typ. (Run mode)
  (TBD) mA Typ. (STOP mode)
- Wide operational voltage range (5V ±10%)
- TTL/CMOS compatible
- Clock Generator
- One channel of Z85C30 Serial Communication Controller (SCC)
- Z180 Compatible MPU core, which has:
  Enhanced Z80 CPU core
  Memory Management Unit (MMU) enables access to 1MB of memory
  Two Asynchronous channels
  Two DMA channels
  Two 16-bit Timers
  Clocked serial I/O Port
- Z84C30 CTC
- Two 8-bit general purpose parallel ports
- Memory configurable RAM and ROM chip select pins
- 100-pin QFP Package

GENERAL DESCRIPTION

The Z80181 SAC Smart Access Controller (hereinafter, referred to as Z181 SAC) is a CMOS 8-bit microprocessor. It is integrated with the Z180 compatible MPU (Z181 MPU), one channel of Z85C30 Serial Communication Controller (SCC), Z80 CTC, two 8-bit general purpose parallel ports, and two chip select signals, all into a single 100-pin QFP (Quad Flat Pack) package. This high-end superintegrated intelligent peripheral controller is targeted for a broad range of intelligent communication control applications, i.e., terminals, printers, modems, and slave communication processors for 8-, 16- and 32-bit MPU based systems. Also included are enhancement/cost reductions of existing hardware using Z80/Z180 with Z8530/Z85C30 applications. Figure 1 shows the block diagram of the Z80181.

Note: 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); /N//S (NORMAL and SYSTEM are both active Low).



Z80181 = Z180 + SCC + CTC + PIA

Figure 1. Z80181 Block Diagram

PIN DEFINITIONS

The pin assignment is shown on Figure 2. Following is the description on each pin.

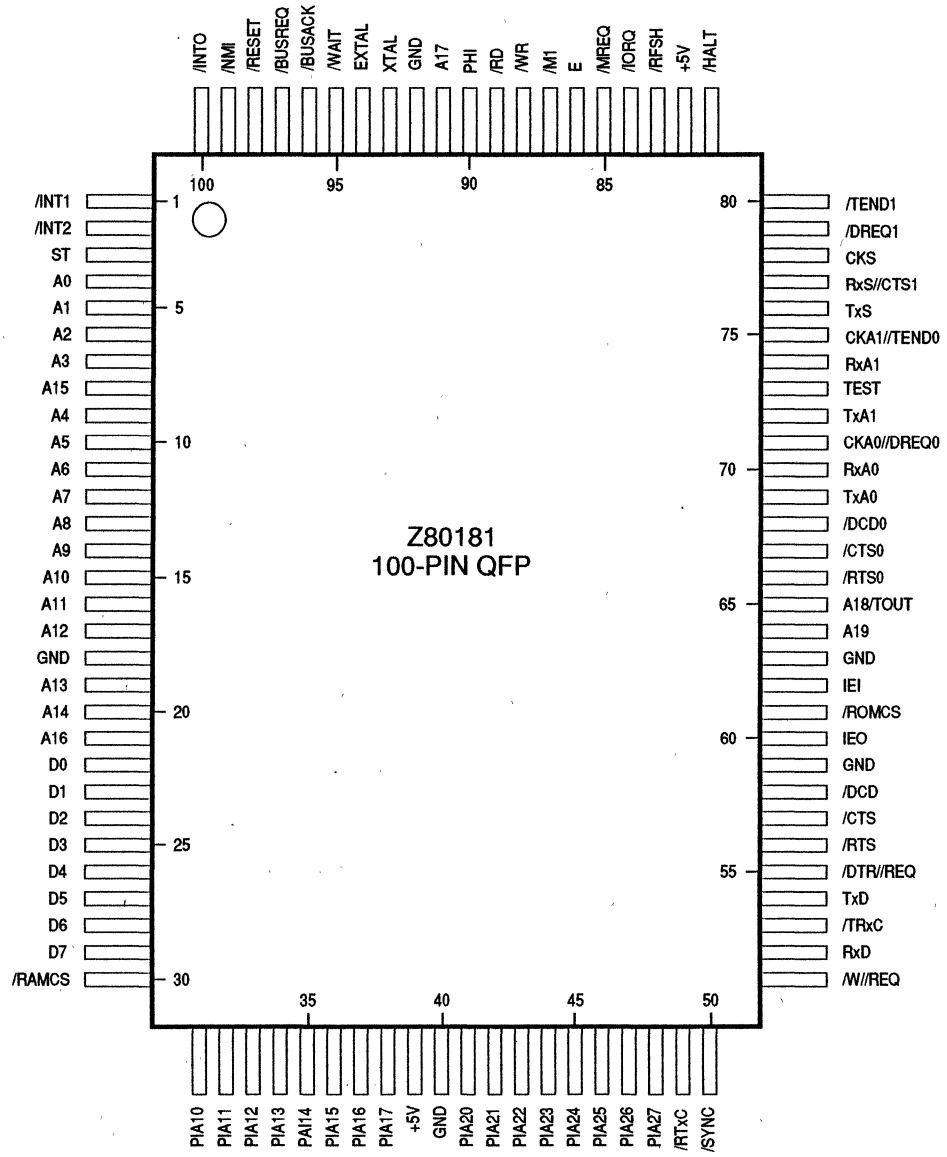


Figure 2. Z80181 Pin-out Assignment

CPU SIGNALS

Pin Name	Pin Number	Input/Output, 3-State	Function
A19 - A0	4-17, 19-21, 64, 65, 91	I/O, Active 1	**Address Bus.** A19 - A0 form a 20-bit address bus which specifies I/O and memory addresses to be accessed During the refresh period, addresses for refreshing are output. The address bus enters a high-impedance state during Reset and external bus acknowledge cycles. The bus is an input when the external bus master is accessing the on-chip peripherals. Address line A18 is multiplexed with the output of PRT Channel 1 (TOUT, selected as address output on Reset).
D0-D7	22-29	I/O, Active 1	**8-bit bidirectional data bus.** When the on-chip CPU is accessing on-chip peripherals, these lines are outputs and hold the data to/from the on-chip peripherals.
/RD	89	I/O, Active 0	**Read signal.** CPU read signal for accepting data from memory or I/O devices. When an external master is accessing the on-chip peripherals, it is an input signal.
/WR	88	I/O, Active 0	**Write Signal.** This signal is active when data to be stored in a specified memory or peripheral device is on the MPU data bus. When an external master is accessing the on-chip peripherals, it is an input signal.
/MREQ	85	I/O, 3-State, Active 0	**Memory request signal.** When an effective address for memory access is on the address bus, /MREQ is active. This signal is analogous to the /ME signal of the Z64180.
/IORQ	84	I/O, 3-State, Active 0	**I/O request signal.** When addresses for I/O are on the lower 8 bits (A7-A0) of the address bus in the I/O operation, "0" is output. In addition, the /IORQ signal is output with the /M1 signal during the interrupt acknowledge cycle to inform peripheral devices that the interrupt response vector is on the data bus. This signal is analogous to the /IOE signal of the Z64180.
/M1	87	Out, 3-State, Active 0	**Machine cycle "1".** /MREQ and /M1 are active together during the operation code fetch cycle /M1 is output for every opcode fetch when a two byte opcode is executed In the maskable interrupt acknowledge cycle, this signal is output together with /IORQ It is also used with /HALT and ST signal to decode the status of the CPU Machine cycle. This signal is analogous to the /LIR signal of the Z64180.
/RFSH	83	Out, 3-state, Active 0	**The Refresh signal.** When the dynamic memory refresh address is on the low order 8-bits of the address bus (A7 - A0), /RFSH is active along with the /MREQ signal This signal is analogous to the /REF signal of the Z64180

CPU SIGNALS (Continued)

Pin Name	Pin Number	Input/Output, 3-State	Function
/INT0	100	Wired-OR I/O, Active 0	**Maskable Interrupt Request 0.** Interrupt is generated by peripheral devices. This signal is accepted if the interrupt enable Flip-Flop (IFF) is set to "1". Internally, the SCC and CTC's interrupt signals are connected to this line, and require an external pull-up resistor.
/INT1, /INT2	1, 2,	In, Active 0	**Maskable Interrupt Request 1 and 2.** This signal is generated by external peripheral devices. The CPU honors these requests at the end of current instruction cycle as long as the /NMI, /BUSREQ and /INT0 signals are inactive. The CPU will acknowledge these interrupt requests with an interrupt acknowledge cycle. Unlike the acknowledgement for /INT0, during this cycle, neither /M1 or /IORQ will become active.
/NMI	99	In, Active 0	**Non-maskable interrupt request signal.** This interrupt request has a higher priority than the maskable interrupt request and does not rely upon the state of the interrupt enable Flip-Flop (IFF).
/HALT	81	Out, 3-State, Active 0	**Halt signal.** This signal is asserted after the CPU has executed either the HALT or SLP instruction, and is waiting for either non-maskable interrupt maskable interrupt before operation can resume. It is also used with the /M1 and ST signals to decode the status of the CPU machine cycle.
/BUSREQ	97	In, Active 0	**BUS request signal.** This signal is used by external devices (such as a DMA controller) to request access to the system bus. This request has higher priority than /NMI and is always recognized at the end of the current machine cycle This signal will stop the CPU from executing further instructions and place the address bus, data bus, /MREQ, /IORQ, /RD and /WR signals into the high impedance state. /BUSREQ is normally wired-OR and a pull-up resistor is externally connected.
/BUSACK	96	Out, Active 0	**Bus Acknowledge signal.** In response to /BUSREQ signal, /BUSACK informs a peripheral device that the address bus, data bus, /MREQ, /IORQ, /RD and /WR signals have been placed in the high impedance state.
/WAIT	95	Wired-OR I/O, Active 0	**Wait signal.** /WAIT informs the CPU that the specified memory or peripheral is not ready for a data transfer. As long as /WAIT signal is active, the MPU is continuously kept in the wait state. Internally, the /WAIT signal from the SCC interface logic is connected to this line, and requres an external pull-up resistor.

PERIPHERAL SIGNALS

Pin Name	Pin Number	Input/Output, 3-State	Function
RXA0, RXA1	70, 74	In, Active 1	**ASCI Receive data 0 and 1.** These signals are the receive data to the ASCI channels.
TXA0, TXA1	69, 72	Out, Active 1	**ASCI Transmit data 0 and 1.** These signals are the receive data to the ASCI channels. Transmit data changes are with respect to the falling edge of the transmit clock
/RTS0	66	Out, Active 0	**Request to send 0.** This is a programmable modem control signal for ASCI channel 0.
/DCD0	68	In, Active 0	**Data Carrier Detect 0.** This is a programmable modem control signal for ASCI channel 0.
/CTS0	67	In, Active 0	**Clear To Send 0.** This is a programmable modem control signal for ASCI channel 0.
/CTS1/RXS	77	In, Active 0	**Clear To Send 0/Clocked Serial Receive Data.** This is a programmable modem control signal for ASCI channel 0. Also, this signal becomes receive data for the CSIO channel under program control. On power-on Reset, this pin is set as RxS.
CKA0//DREQ0	71	I/O, Active 1	**Asynchronous Clock0/DMAC0 request.** This pin is the transmit and receive clock for the Asynchronous channel 0. Also, under program control, this pin is used to request a DMA transfer from DMA channel 0. DMA0 monitors this input to determine when an external device is ready for a read or write operation. On power-on Reset, this pin is initialized as CKA0.
CKA1//TEND0	75	I/O, Active 1	**Asynchronous Clock1/DMAC0 Transfer end.** This pin is the transmit and receive clock for the Asynchronous channel 1. Also, under program control, this pin becomes /TEND0 and is asserted during the last write cycle of the DMA0 operation and is used to indicate the end of the block transfer. On power-on Reset, this pin initializes as CKA1.
/TEND1	80	Out, Active 0	**DMAC1 Transfer end.** This pin is asserted during the last write cycle of the DMA1 operation and is used to indicate the end of the block transfer.
CKS	78	I/O, Active 1	**CSIO clock** This line is the clock for the CSIO channel.
TXS	76	Out, Active 1	**CSI/O Tx Data.** This line carries the transmit data from the CSIO channel.
/DREQ1	79	In, Active 0	**DMAC1 request.** This pin is used to request a DMA transfer from DMA channel 1. DMA1 monitors this input to determine when an external device is ready for a read or write operation.

SCC SIGNALS

Pin Name	Pin Number	Input/Output, 3-State	Function
/W//REQ	51	Active 0	**Wait/Request.** Open-drain when programmed for a Wait function, driven "1" or "0" when programming for a Request function. Used as /WAIT or /REQUEST depending upon SCC programming. When programmed as /WAIT, this signal is asserted to alert the CPU that addressed memory or I/O devices are not ready and that the CPU should wait. When programmed as /REQUEST, this signal is asserted when a peripheral device associated with a DMA port is ready to read/write data. After reset, this pin becomes "/WAIT".
/SYNC	50	I/O, Active 0	**Synchronization.** This pin can act either as input, output, or part of the crystal oscillator circuit. In asynchronous receive mode (crystal oscillator option not selected), this pin is an input similar to /CTS and /DCD. In this mode, transitions on this line affect the state of the Sync/Hunt status bit in Read Register 0 but has no other function.
			In external sync mode with crystal oscillator option not selected, this line also acts as an input. In this mode, /SYNC must be driven "0" two receive clock cycles after the last bit in the synchronous character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of /SYNC.
			In internal sync mode (Monosync and Bisync) with the crystal oscillator option not selected, this line acts as output and is active only during the part of the receive clock cycle in which a synchronous character is recognized (regardless of character boundaries). In SDLC mode, this pin acts as an output and is valid on receipt of a flag
RxD	52	In, Active 1	**Receive Data.** This input signal receives serial data at standard TTL levels.
/RTxC	49	In, Active 0	**Receive/Transmit clock.** This pin can be programmed in several different modes of operation. /RTxC may supply the receive clock, the transmit clock, the clock for the Baud Rate Generator, or the clock for the Digital Phase-Locked Loop. This pin can also be programmed for use with the /SYNC pin as a crystal oscillator The receive clocks can be 1, 16, 32, or 64 times the data transfer rate in Asynchronous mode.

SCC SIGNALS (Continued)

Pin Name	Pin Number	Input/Output, 3-State	Function
/TRxC	53	I/O, Active 0	**Transmit/Receive Clock.** This pin can be programmed in several different modes of operation. /TRxC can supply the receive clock or the transmit clock in the input mode. Also, it can supply the output of the Digital Phase-Locked Loop, the crystal oscillator, the Baud Rate Generator, or the transmit clock in the output mode.
TxD	54	Out, Active 1	**Transmit Data.** This Output signal transmits serial data at standard TTL level.
/DTR//REQ	55	Out, Active 0	**Data Terminal Ready/Request.** This output follows the state programmed into the DTR bit. It can also be used as general purpose output or as Request line for a DMA controller
/RTS	56	Out, Active 0	**Request To Send.** When the RTS bit in Write Register 5 is set, the /RTS signal goes low. When the RTS bit is reset in Asynchronous mode and auto enable is on, the signal goes high after the transmitter is empty. In synchronous mode or in Asynchronous mode, with Auto Enable off, the /RTS pin follows the state of the RTS bit. This pin can be used as a general purpose output.
/CTS	57	In, Active 0	**Clear To Send.** If this pin is programmed as auto enable, a "0" on the input enables the transmitter. If not programmed as Auto Enable, it may be used as a general purpose input. This input is Schmitt-trigger buffered to accommodate inputs with slow rise times. The SCC detects pulses on this input and can interrupt the CPU on both logic level transitions.
/DCD	58	In, Active 0	**Data Carrier Detect.** This pin functions as receiver enable if it is programmed for auto enable. Otherwise, it may be used as a general purpose input. This input is Schmitt-trigger buffered to accommodate slow rise-time inputs The SCC detects pulses on this input and can interrupt the CPU on both logic level transitions.

PIA/CTC SIGNALS

Pin Name	Pin Number	Input/Output, 3-State	Function
PIA17-PIA14	35-38	I/O	**Port 1 Data 7-Port 1 Data 4 or CTC ZC/TO3 - ZC/TO0.** These lines can be configured as inputs or outputs on a bit -by-bit basis. Also, under program control, these bits become Z80 CTC's ZC/TO3 - ZC/TO0, and in either timer or counter mode, pulses are output when the down counter has reached zero. On reset, these signals function as PIA17-14 and are inputs.
PIA13-PIA10	31-34	I/O	**Port 1 Data 3-Port 1 Data 0 or CTC CLK/TRG3-0.** These lines can be configured as inputs or outputs on a bit by bit basis Also, under program control, these bits become Z80 CTC's CLK/TRG3-CLK/TRG0, and correspond to four Counter/Timer Channels. In the counter mode, each active edge causes the downcounter to decrement by one. In timer mode, an active edge starts the timer It is program selectable whether the active edge is rising or falling. On reset, these signals are set to PIA13-10 as inputs.
PIA27-20	41-48	I/O	**Port 2 Data.** These lines are configured as inputs or outputs on a bit-by-bit basis. On reset, they are inputs.

SYSTEM CONTROL SIGNALS

Pin Name	Pin Number	Input/Output, 3-State	Function
ST	3	Out, Active 1	**Status.** This signal is used with the /M1 and /HALT output to decode the status of the CPU machine cycle. Note that the /M1 output is affected by the status of the M1E bit in the OMCR register. The following table shows the status while M1E=1.

ST	/HALT	/M1	Operation
0	1	0	CPU Operation (1st Op-code fetch)
1	1	0	CPU Operation (2nd and 3rd Op-code fetch)
1	1	1	CPU Operation (MC other than Op-code fetch)
0	X	1	DMA operation
0	0	0	HALT mode
1	0	1	SLEEP mode (Incl. System STOP mode)

SYSTEM CONTROL SIGNALS (Continued)

Pin Name	Pin Number	Input/Output, 3-State	Function
IEI	62	In, Active 1	**Interrupt enable input signal.** IEI is used with the IEO to form a priority daisy chain when there is more than one interrupt-driven peripheral.
IEO	60	Out, Active 1	**The interrupt enable output signal.** In the daisy-chain interrupt control, IEO controls the interrupt of external peripherals. IEO is active when IEI is "1" and the CPU is not servicing an interrupt from the on-chip peripherals.
/ROMCS	61	Out, Active 0	**ROM Chip select.** Used to access ROM. Refer to "Functional Description" on chip select signals for further explanation.
/RAMCS	30	Out, Active 0	**RAM Chip Select.** Used to access RAM. Refer to "Functional Description" on chip select signals for further explanation.
/RESET	98	In, Active 0	**Reset signal.** /RESET signal is used for initializing the MPU and other devices in the system. It must be kept in the active state for a period of at least 3 system clock cycles.
EXTAL	94	In, Active 1	**Crystal oscillator connecting terminal.** A parallel resonant crystal is recommended. If an external clock source is used as the input to the Z180 Clock Oscillator unit, supply the clock into this terminal.
XTAL	93	Out	**Crystal oscillator connecting terminal.**
PHI	90	Out, Active 1	**System Clock.** Single-phase clock output from Z181 MPU
E	86	Out, Active 1	**Enable Clock.** Synchronous Machine cycle clock output during a bus transaction.
TEST	73	Out	**Test pin.** Used in the open state.
V_{cc}	39, 82		**Power Supply.** +5 Volts
V_{ss}	18, 40, 59, 63, 92		**Power Supply.** 0 Volts

FUNCTIONAL DESCRIPTION

Functionally, the on-chip Z181 MPU, SCC, and CTC are the same as the discrete devices (Figure 1). Therefore, for a detailed description of each individual unit, refer to the Product Specification/Technical Manual of each discrete product. The following subsections describe each individual functional unit of the SAC.

Z181 MPU

This unit provides all the capabilities and pins of the Zilog Z180 MPU. Figure 3 shows the Z181 MPU block diagram. This allows 100% software compatibility with existing Z180 (and Z80) software. Note that the on-chip I/O address should not be relocated to the I/O address (from 0C0h to 0FFh) to avoid address conflicts. The following is an overview of the major functional units of the Z181.

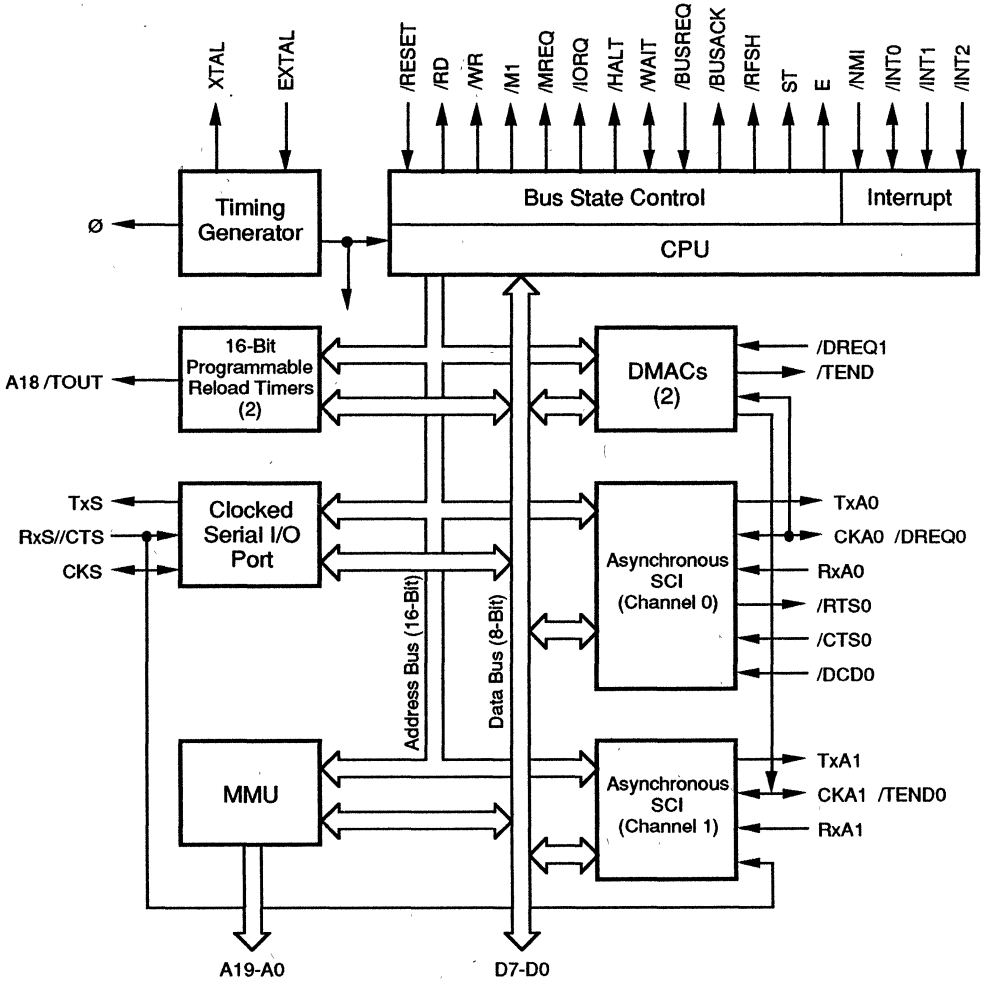


Figure 3. Z181 MPU Block Diagram

FUNCTIONAL DESCRIPTION (Continued)

Z181 CPU

The Z181 CPU has 100% software compatibility with the Z80 CPU. In addition, the Z181 CPU has the following features:

Faster execution speed. The Z181 CPU is "fine tuned" making execution speed, on average, 10% to 20% faster than the Z80 CPU.

Enhanced DRAM Refresh Circuit. Z181 CPU's DRAM refresh circuit does periodic refresh and generates an 8-bit refresh address. It can be disabled or the refresh period adjusted, via software control.

Enhanced Instruction Set. The Z181 CPU has seven additional instructions to those of the Z80 CPU which include the MLT (Multiply) instruction.

HALT and Low Power Modes of Operation. The Z181 CPU has HALT and low power modes of operation, which are ideal for the applications requiring low power consumption like battery operated portable terminals.

System Stop Mode. When the Z181 SAC is in SYSTEM STOP mode, it is only the Z181 MPU which is in STOP mode. The on-chip CTC and SCC continue their normal operation.

Instruction Set. The instruction set of the Z181 CPU is identical to the Z180. For more details about each transaction, please refer to the Data Sheet/Technical Manual for the Z180/Z80 CPU.

Z181 CPU Basic Operation

Z181 CPU's basic operation consists of the following events. These are identical to the Z180 MPU. For more details about each operation, please refer to the Data Sheet/Technical manual for the Z180.

- Operation code fetch cycle.
- Memory Read/Write operation.
- Input/Output operation.
- Bus request/acknowledge operation.
- Maskable interrupt request operation.
- Trap and Non-Maskable interrupt request operation.
- HALT and low power modes of operation.
- Reset Operation.

Memory Management Unit (MMU)

The Memory Management Unit (MMU) allows the user to "map" the memory used by the CPU (64K bytes of logical addressing space) into 1M bytes of physical addressing space. The organization of the MMU allows object code compatibility with the Z80 CPU while offering access to an extended memory space. This is accomplished by using an effective "common area-banked area" scheme.

DMA Controller

The Z181 MPU has two DMA controllers. Each DMA controller provides high-speed data transfers between memory and I/O devices. Transfer operations supported are memory to memory, memory to/from I/O, and I/O to I/O. Transfer modes supported are request, burst, and cycle steal. The DMA can access the full 1M bytes addressing range with a block length up to 64K bytes and can cross over 64K boundaries.

Asynchronous Serial Communication Interface (ASCI)

This unit provides two individual full-duplex UARTs. Each channel includes a programmable baud rate generator and modem control signals The ASCI channels also support a multiprocessor communication format.

Programmable Reload Timer (PRT)

The Z181 MPU has two separate Programmable Reload Timers, each containing a 16-bit counter (timer) and count reload register. The time base for the counters is system clock divided by 20. PRT channel 1 provides an optional output to allow for waveform generation.

Clocked Serial I/O (CSI/O)

The CSI/O channel provides a half-duplex serial transmitter and receiver. This channel can be used for simple high-speed data connection to another CPU or MPU.

Programmable Wait State Generator

To ease interfacing with slow memory and I/O devices, the Z181 MPU unit has a programmable wait state generator. By programming the DMA/WAIT Control Register (DCNTL), up to three wait states are automatically inserted in memory and I/O cycles. This unit also inserts wait states during on-chip DMA transactions

Z85C30 Serial Communication Controller Logic Unit

This logic unit provides the user with a multi-protocol serial I/O channel that is completely compatible with the two channel Z85C30 SCC with the following exceptions

Their basic functions as serial-to-parallel and parallel-to-serial converters can be programmed by the CPU for a broad range of serial communications applications. This logic unit is capable of supporting all common asynchronous and synchronous protocols (Monosync, Bisync, and SDLC/HDLC, byte or bit oriented - Figure 4).

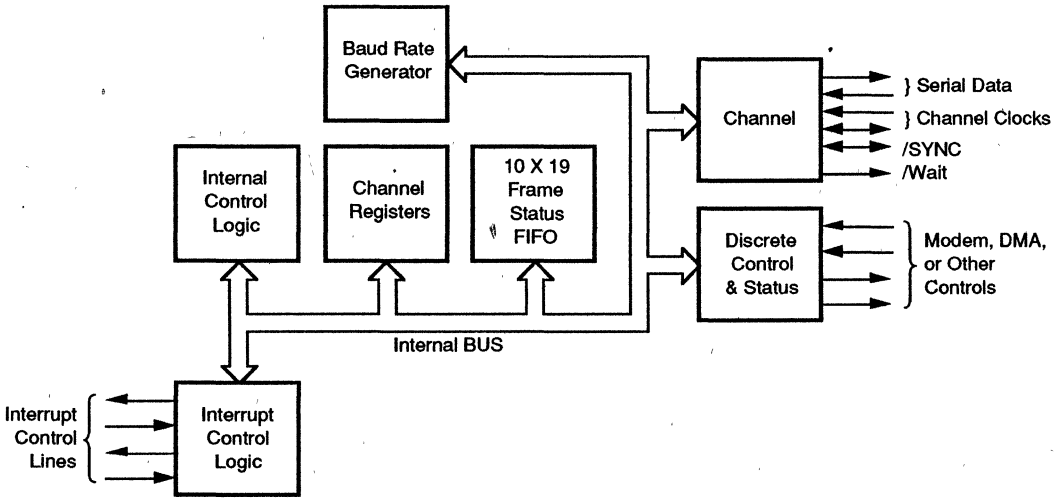


Figure 4. SCC Block Diagram

On the discrete version of the SCC (dual channel version), there are two registers shared between channels A and B, and two registers whose functions are different by channel. These are: WR2, WR9 (shared registers), and RR2 and RR3 (different functionality).

Following are the differences in functionality:

- **RR2** - Returns Unmodified Vector or modified vector depends on the status of "VIS" (Vector Include Status) bit in WR9.
- **RR3** - Returns IP status (Ch.A side).
- **WR9** - Ch.B Software Reset command has no effect.

The PCLK for the SCC is connected to PHI (System clock), the /INT signal is connected to /INT0 signal internally (requires external pull-up resistor) and SCC is reset when /RESET input becomes active Interrupt from the SCC is handled via Mode 2 interrupt. During the interrupt acknowledge cycle, the on-chip SCC interface circuit inserts two wait states automatically

Z84C30 Counter/Timer Logic Unit

This logic unit provides the user with four individual 8-bit Counter/Timer Channels that are compatible with the Z84C30 CTC (Figure 5). The Counter/Timers are programmed by the CPU for a broad range of counting and timing applications. Typical applications include event counting, interrupt and interval counting, and serial baud rate clock generation.

Each of the Counter/Timer Channels, designated Channels 0-3, have an 8-bit prescaler (when used in timer mode) and its own 8-bit counter to provide a wide range of count resolution. Each of the channels have their own Clock/Trigger input to quantify the counting process and an output to indicate zero crossing/timeout conditions These signals are multiplexed with the Parallel Interface Adapter 1 (PIA1). With only one interrupt vector programmed into the logic unit, each channel can generate a unique interrupt vector in response to the interrupt acknowledge cycle.

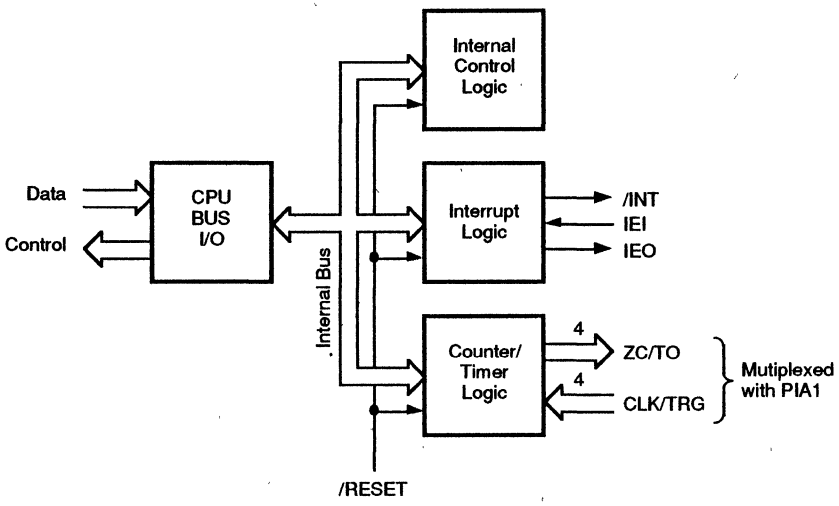


Figure 5. CTC Block Diagram

Parallel Interface Adapter (PIA)

The SAC has two 8-bit Parallel Interface Adapter (PIA) Ports. The ports are referred to as PIA1 and PIA2. Each port has two associated control registers; a Data Register and a register to determine each bit's direction (input or output). PIA1 is multiplexed with the CTC I/O pins. When the CTC I/O feature is selected, the CTC I/O functions override the PIA1 feature Mode Selection is made through the System Configuration Register (Address. EDh, Bit D0). PIA1 has Schmitt-trigger inputs to have a better noise margin. These ports are inputs after reset.

Clock Generator

The SAC uses the Z181 MPU's on-chip clock generator to supply system clock. The required clock is easily generated by connecting a crystal to the external terminals (XTAL, EXTAL). The clock output runs at half the crystal frequency. The system clock inputs of the SCC and the CTC are internally connected to the PHI output of the Z181 MPU.

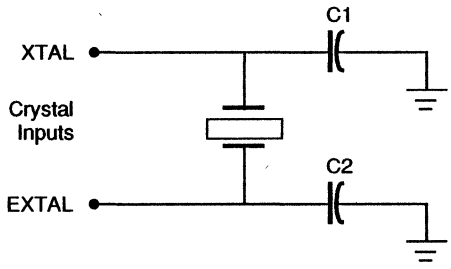


Figure 6. Circuit Configuration For Crystal

Recommended characteristics of the crystal and the values for the capacitor are as follows (the values will change with crystal frequency).

Type of crystal: Fundamental, parallel type crystal (AT cut is recommended).

Frequency tolerance: Application dependent.
CL, Load capacitance: Approximately 22 pf (acceptable range is 20-30 pf)

Rs, equivalent-series resistance: ≤ 30 Ohms
Drive level: 10mW (for ≤ 10 MHz crystal) 5mW (for ≥ 10 MHz crystal)

$C_{IN} = C_{OUT} = 15 \sim 22$ pF.

Chip Select Signals

The SAC has two chip select (/RAMCS, /ROMCS) pins. /ROMCS is the chip select signal for ROM and /RAMCS is the chip select signal for RAM. The boundary value for each chip select signal is 8 bits wide allowing all memory accesses with addresses less than or equal to this boundary value. This causes assertion of the corresponding /CS pin. These features are controlled via the RAM upper boundary address register (I/O address EAh), RAM lower boundary address register (I/O address EBh) and ROM upper boundary address register (I/O address ECh).

These two signals are generated by decoding address lines A19-A12. Note that glitches may be observed on the /RAMCS and /ROMCS signals because the address decoding logic decodes only A19-A12, without any control signals.

Bit D5 of the System Configuration Register allows the option of disabling the /ROMCS signal This feature is used in systems which, for example, have a shadow RAM. However, prior to disabling the /ROMCS signal, the ROMBR and RAMLBR registers must be re-initialized from their default values.

For more details, please refer to "Programming section"

ROM Emulator Mode

To ease development, the SAC has a mode to support "ROM emulator" development systems. In this mode, a read data from on-chip registers (except Z181 MPU on-chip registers) are available (data bus direction set to output) to make data visible from the outside, so that a ROM Emulator/Logic Analyzer can monitor internal transactions. Otherwise, a read from an internal transaction is not available to the outside (data bus direction set to Hi-Z status). Mode selection is made through the D1 bit in the System Configuration Register (I/O Address: EDh).

PROGRAMMING

The following subsections explain and define the parameters for I/O Address assignments, I/O Control Register Addresses and all pertinent Timing parameters.

I/O Address Assignment

The SAC has 78 internal 8-bit registers to control on-chip peripherals and features. Sixty-four registers out of 78 registers are occupied by the Z181 MPU control registers; two for SCC control registers, four for PIA control registers, four for the Counter/Timer, three for RAM/ROM configuration (memory address boundaries) and one for SAC's system control. The SAC's I/O addresses are listed in Table 1. These registers are assigned in the SAC's I/O addressing space and the I/O addresses are fully decoded from A7-A0 and have no image

Table 1. I/O Control Register Address

Address	Register
00h to 3Fh	Z181 MPU Control Registers (Relocatable to 040h-07Fh, or 080h-0BFh)
E0h	PIA1 Data Direction Register (P1DDR)
E1h	PIA1 Data Port (P1DP)
E2h	PIA2 Data Direction Register (P2DDR)
E3h	PIA2 Data Register (P2DP)
E4h	CTC Channel 0 Control Register (CTC0)
E5h	CTC Channel 1 Control Register (CTC1)
E6h	CTC Channel 2 Control Register (CTC2)
E7h	CTC Channel 3 Control Register (CTC3)
E8h	SCC Control Register (SCCCR)
E9h	SCC Data Register (SCCDR)
EAh	RAM Upper Boundary Address Register (RAMUBR)
EBh	RAM Lower Boundary Address Register (RAMLBR)
ECh	ROM Address Boundary Register (ROMBR)
EDh	System Configuration Register (SCR)
EEh	Reserved
EFh	Reserved

Z181 MPU Control Registers

The I/O address for these registers can be relocated in 64 byte boundaries by programming of the I/O Control Register (Address xx111111b).

Do not relocate these registers to address from 0C0h since this will cause an overlap of the Z180 registers and the 16 registers of the Z181 (address 0E0h to 0EFh).

Also, the OMCR register (Address: xx111101b) has to be programmed as 0x0xxxxxb (x: don't care) as a part of the initialization procedure. The M1E bit (Bit D7) of this register must be programmed as 0 or the interrupt daisy chain is corrupted. The /IOC bit (Bit D5) of this register is programmed as 0 so that the timing of the /RD and /IORQ signals are compatible with Z80 peripherals.

For detailed information, refer to the Z180 Technical Manual.

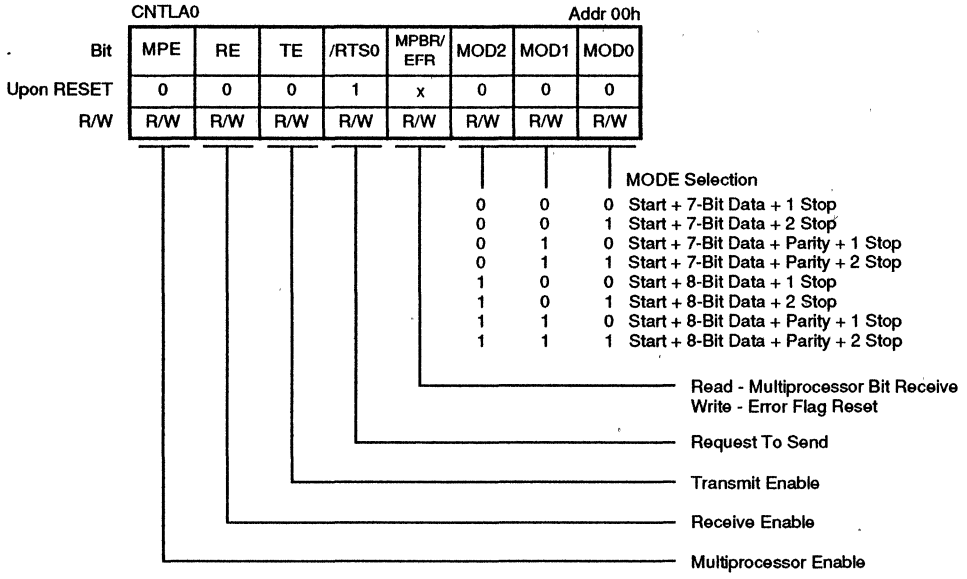


Figure 7. ASCI Control Register A (Ch. 0)

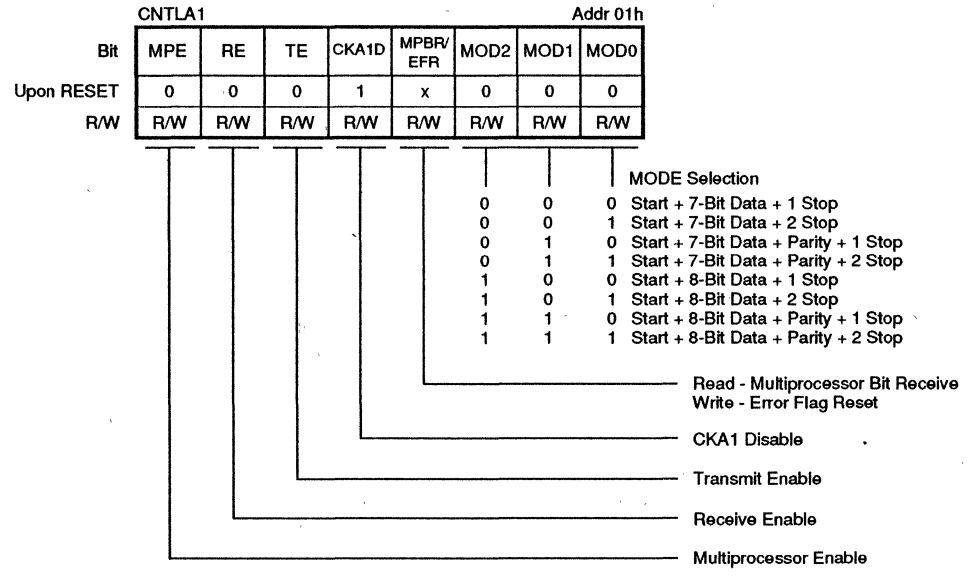
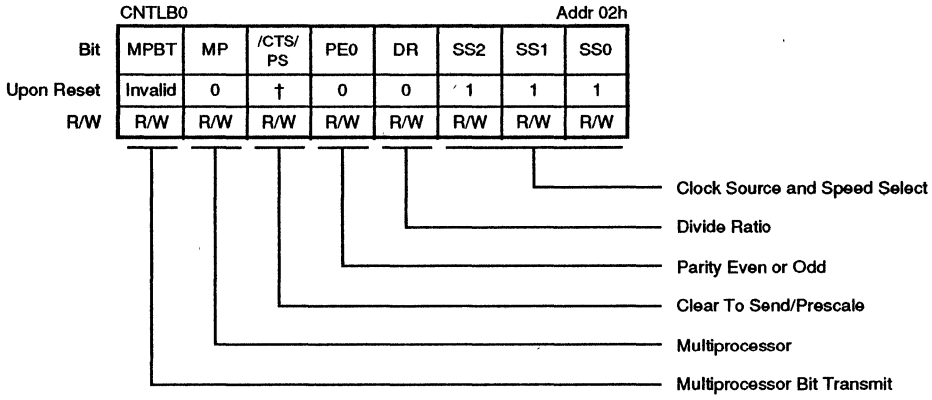


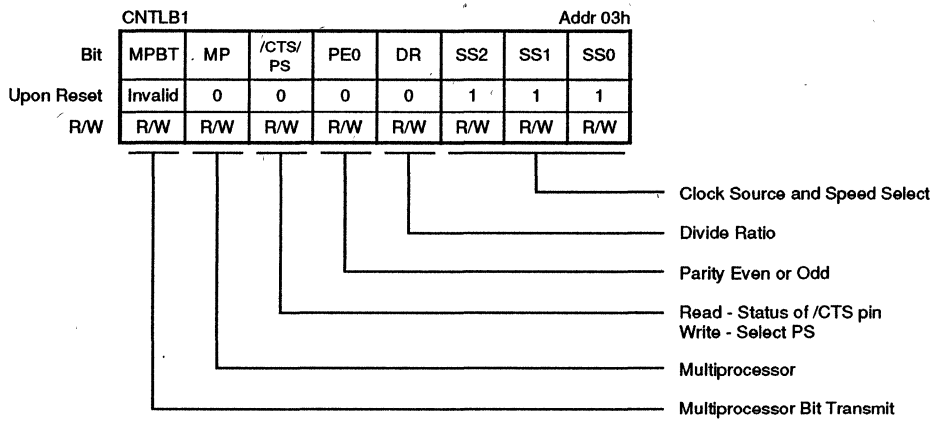
Figure 8. ASCI Control Register A (Ch. 1)



† /CTS - Depending on the condition of /CTS pin.
PS - Cleared to 0.

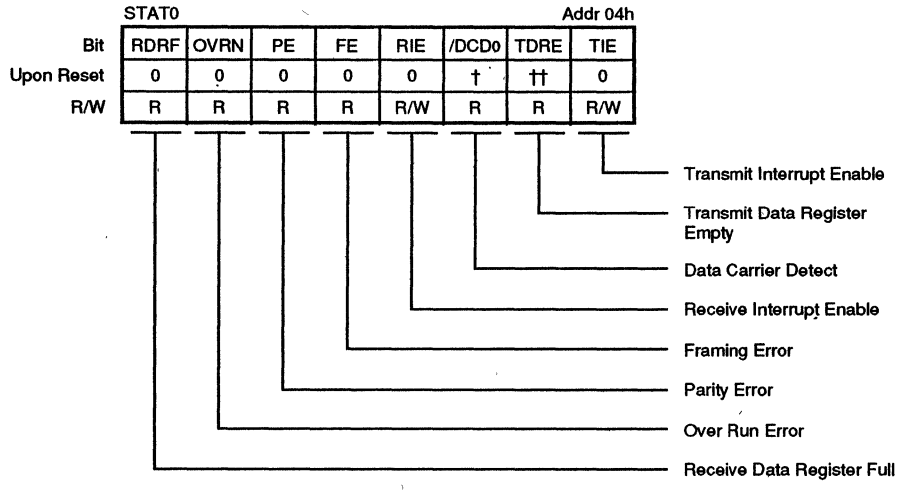
General Divide Ratio	PS = 0 (Divide Ratio = 10)		PS = 1 (Divide Ratio = 30)	
SS, 2, 1, 0	DR = 0 (x16)	DR = 1 (x64)	DR = 0 (x16)	DR = 1 (x64)
000	Ø ÷ 160	Ø ÷ 640	Ø ÷ 480	Ø ÷ 1920
001	Ø ÷ 320	Ø ÷ 1280	Ø ÷ 960	Ø ÷ 3840
010	Ø ÷ 640	Ø ÷ 2580	Ø ÷ 1920	Ø ÷ 7680
011	Ø ÷ 1280	Ø ÷ 5120	Ø ÷ 3840	Ø ÷ 15360
100	Ø ÷ 2560	Ø ÷ 10240	Ø ÷ 7680	Ø ÷ 30720
101	Ø ÷ 5120	Ø ÷ 20480	Ø ÷ 15360	Ø ÷ 61440
110	Ø ÷ 10240	Ø ÷ 40960	Ø ÷ 30720	Ø ÷ 122880
111	External Clock (Frequency < Ø ÷ 40)			

Figure 9. ASCI Control Register B (Ch. 0)



General Divide Ratio	PS = 0 (Divide Ratio = 10)		PS = 1 (Divide Ratio = 30)	
SS, 2, 1, 0	DR = 0 (x16)	DR = 1 (x64)	DR = 0 (x16)	DR = 1 (x64)
000	Ø ÷ 160	Ø ÷ 640	Ø ÷ 480	Ø ÷ 1920
001	Ø ÷ 320	Ø ÷ 1280	Ø ÷ 960	Ø ÷ 3840
010	Ø ÷ 640	Ø ÷ 2580	Ø ÷ 1920	Ø ÷ 7680
011	Ø ÷ 1280	Ø ÷ 5120	Ø ÷ 3840	Ø ÷ 15360
100	Ø ÷ 2560	Ø ÷ 10240	Ø ÷ 7680	Ø ÷ 30720
101	Ø ÷ 5120	Ø ÷ 20480	Ø ÷ 15360	Ø ÷ 61440
110	Ø ÷ 10240	Ø ÷ 40960	Ø ÷ 30720	Ø ÷ 122880
111	External Clock (Frequency < Ø ÷ 40)			

Figure 10. ASCI Control Register B (Ch. 1)



† /DCD0 - Depending on the condition of /DCD0 Pin.

†† /CTS0 Pin	TDRE
L	1
H	0

Figure 11. ASCI Status Register

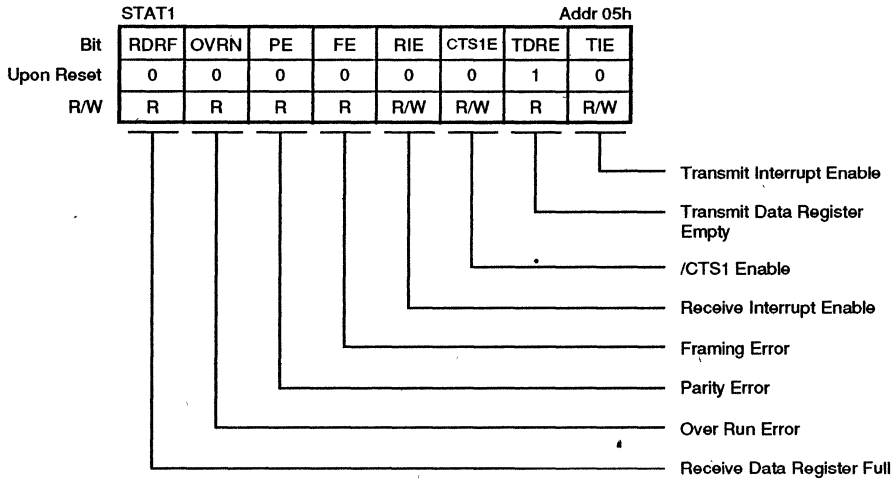


Figure 12. ASCI Status Register (Ch 1)

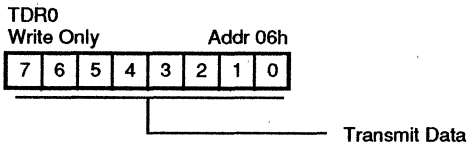


Figure 13. ASCI Transmit Data Register (Ch. 0)

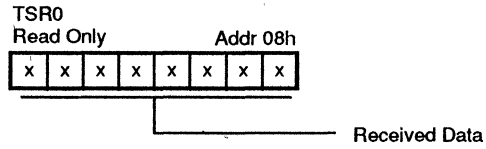


Figure 15. ASCI Receive Data Register (Ch. 0)

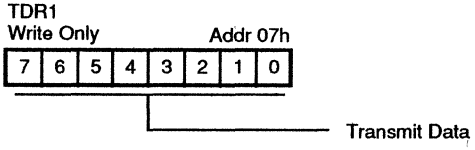


Figure 14. ASCI Transmit Data Register (Ch. 1)

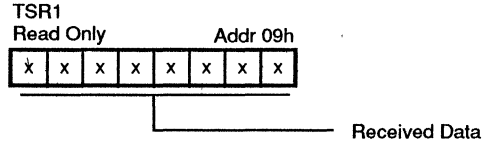
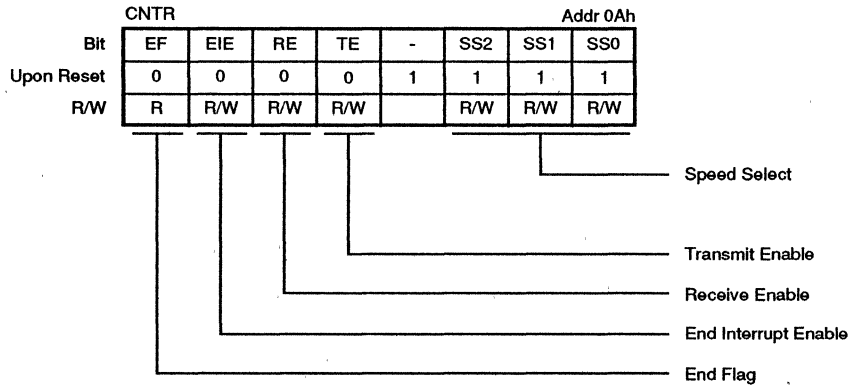


Figure 16. ASCI Receive Data Register (Ch 1)

CSI/O REGISTERS



SS2, 1, 0	Baud Rate
000	Ø ÷ 20
001	Ø ÷ 40
010	Ø ÷ 80
011	Ø ÷ 100

SS2, 1, 0	Baud Rate
100	Ø ÷ 320
101	Ø ÷ 640
110	Ø ÷ 1280
111	External Clock (Frequency < Ø ÷ 20)

Figure 17. CSI/O Control Register

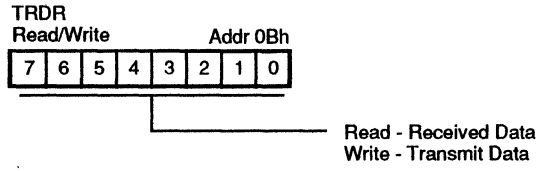


Figure 18. CSI/O Transmit/Receive Data Register

TIMER REGISTERS

Timer Data Registers

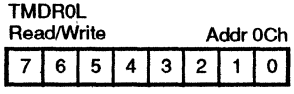


Figure 19. Timer 0 Data Register L

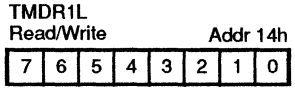


Figure 20. Timer 1 Data Register L

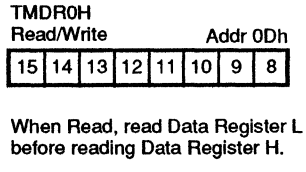


Figure 21. Timer 0 Data Register H

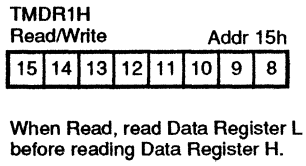


Figure 22. Timer 1 Data Register H

TIMER RELOAD REGISTERS

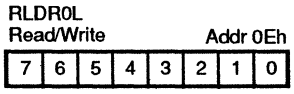


Figure 23. Timer 0 Reload Register L

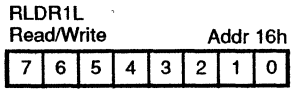


Figure 24. Timer 1 Reload Register L

RLDR0H
Read/Write Addr 0Fh

15	14	13	12	11	10	9	8

Figure 25. Timer 0 Reload Register H

RLDR1H
Read/Write Addr 17h

15	14	13	12	11	10	9	8

Figure 26. Timer 1 Reload Register H

TIMER CONTROL REGISTER

TCR Addr 10h

Bit	TIF1	TIF0	TIE1	TIE0	TOC1	TOC0	TDE1	TDE0
Upon Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Timer Down Count Enable 1,0
Timer Output Control 1,0
Timer Interrupt Enable 1,0
Timer Interrupt Flag 1,0

TOC1,0	A15/TOUT
00	Inhibited
01	Toggle
10	0
11	1

Figure 27. Timer Control Register

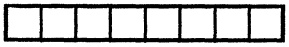
FREE RUNNING COUNTER

FRC
Read Only Addr 18h

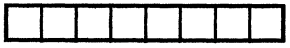
7	6	5	4	3	2	1	0

Figure 28. Free Running Counter

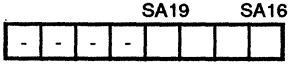
SAR0L
Read/Write Addr 20h



SAR0H
Read/Write Addr 21h



SAR0B
Read/Write Addr 22h

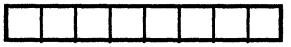


Bits 0-2 (3) are used for SAR0B

A19,	A18,	A17,	A16	DMA Transfer Request
x	x	0	0	/DREQ0 (external)
x	x	0	1	RDR0 (ASCI0)
x	x	1	0	TDR0 (ASCI1)
x	x	1	1	Not Used

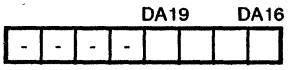
Figure 29. DMA 0 Source Address Registers

DAR0L
Read/Write Addr 23h



DAR0H
Read/Write Addr 24h

DAR0B
Read/Write Addr 25h



Bits 0-2 (3) are used for DAR0B

A19,	A18,	A17,	A16	DMA Transfer Request
x	x	0	0	/DREQ0 (external)
x	x	0	1	RDR0 (ASCI0)
x	x	1	0	TDR0 (ASCI1)
x	x	1	1	Not Used

Figure 30. DMA 0 Destination Address Registers

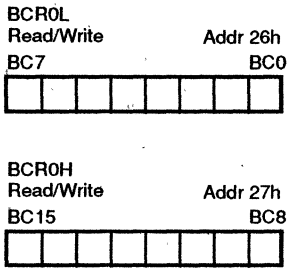


Figure 31. DMA 0 Byte Counter Registers

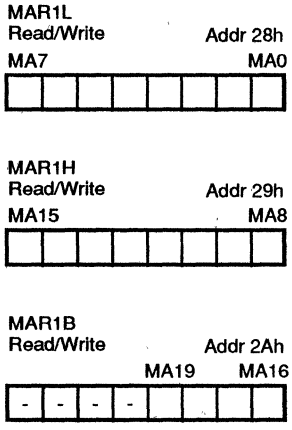


Figure 32. DMA 1 Memory Address Registers

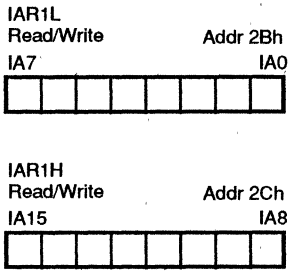


Figure 33. DMA 1 I/O Address Registers

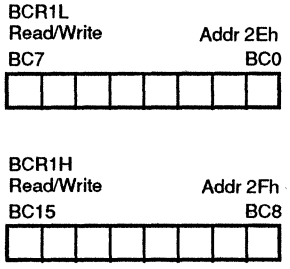


Figure 34. DMA 1 Byte Count Registers

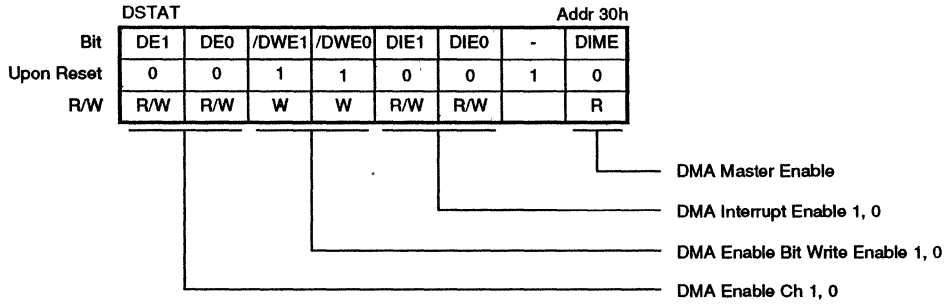
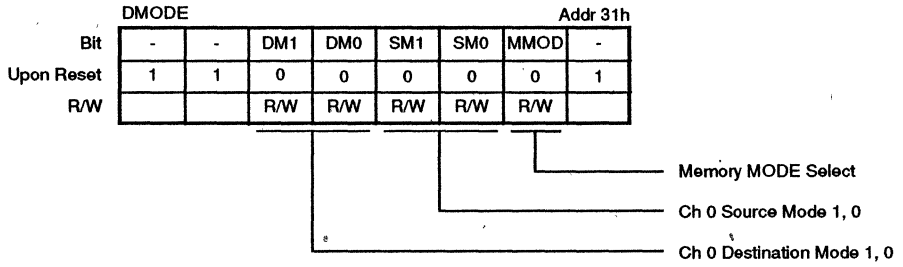


Figure 35. DMA Status Register

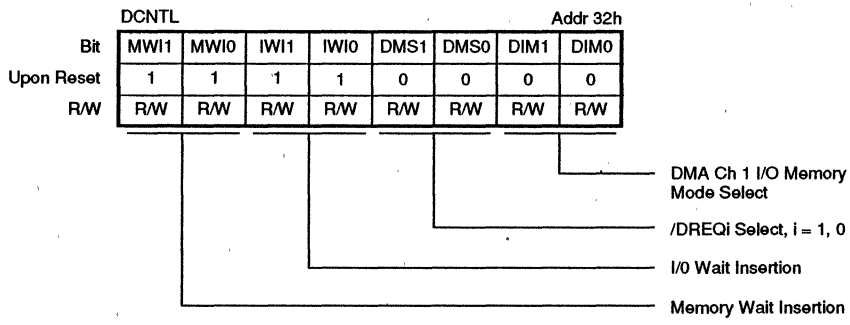


DM1, 0	Destination	Address
00	M	DAR0+1
01	M	DAR0-1
10	M	DAR0 Fixed
11	I/O	DAR0 Fixed

SM1, 0	Source	Address
00	M	SAR0+1
01	M	SAR0-1
10	M	SAR0 Fixed
11	I/O	SAR0 Fixed

MMOD	Mode
0	Cycle Steal Mode
1	Burst Mode

Figure 36. DMA Mode Registers



MWI1, 0	No. of Wait States
00	0
01	1
10	2
11	3

IWI1, 0	No. of Wait States
00	0
01	2
10	3
11	4

DMSi	Sense
1	Edge Sense
0	Level Sense

DM1, 0	Transfer Mode	Address Increment/Decrement	
00	M - I/O	MAR1+1	IAR1 Fixed
01	M - I/O	MAR1-1	IAR1 Fixed
10	I/O - M	IAR1 Fixed	MAR1+1
11	I/O - M	IAR1 Fixed	MAR1-1

Figure 37. DMA/WAIT Control Register

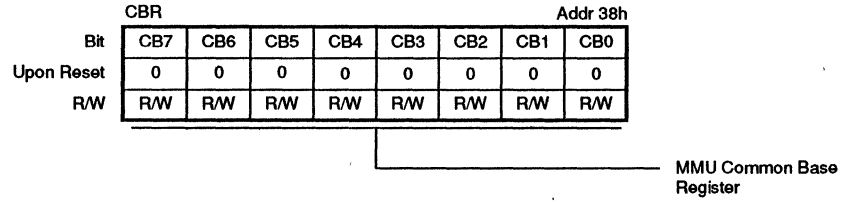


Figure 38. MMU Common Base Register

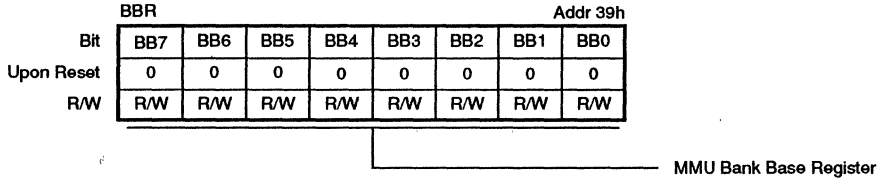


Figure 39. MMU Bank Base Register

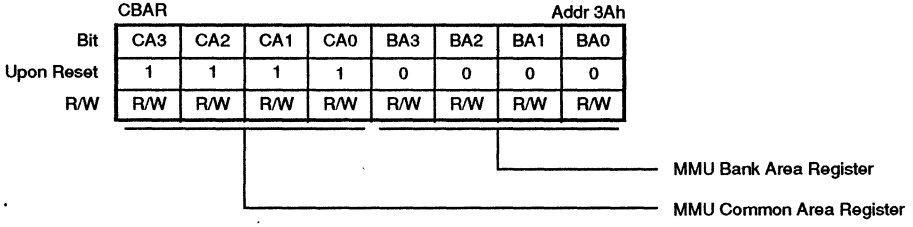


Figure 40. MMU Common/Bank Area Register

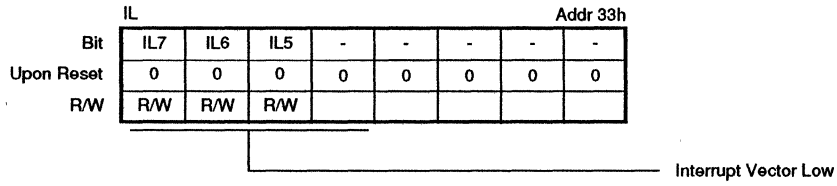


Figure 41. Interrupt Vector Low Register

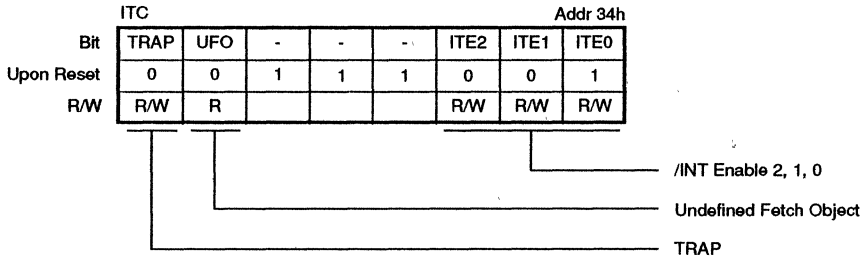
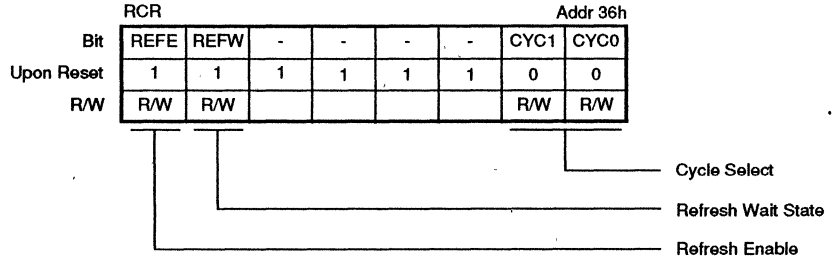
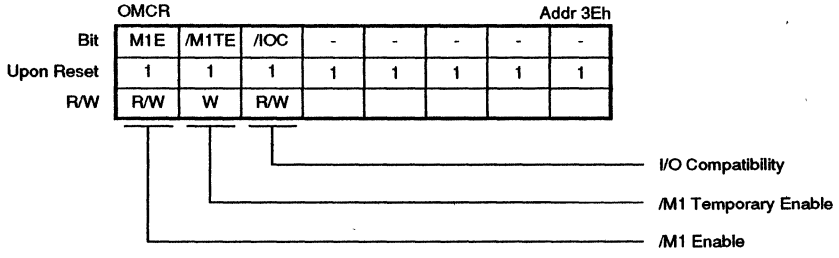


Figure 42. INT/TRAP Control Register



CYC1, 0	Interval of Refresh Cycle
00	10 states
01	20 states
10	40 states
11	80 states

Figure 43. Refresh Control Register



Note: This register has to be programmed as 0x0xxxxxb(x:don't care) as a part of Initialization.

Figure 44. Operation Mode Control Register

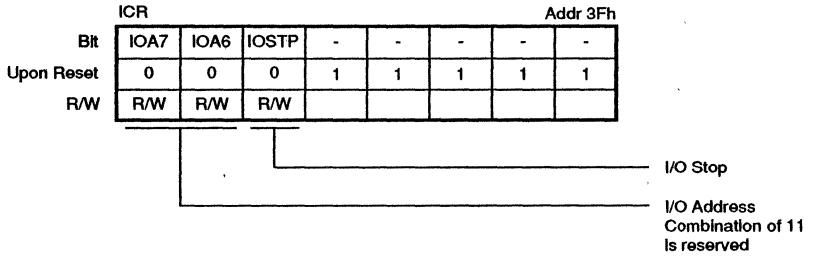


Figure 45. I/O Control Register

CTC CONTROL REGISTERS

Channel Control Word

This word sets the operating modes and parameters as described below. Bit D0 must be a "1" to indicate that this is a Control Word (Figure 46).

For more detailed information, refer to the CTC Technical Manual.

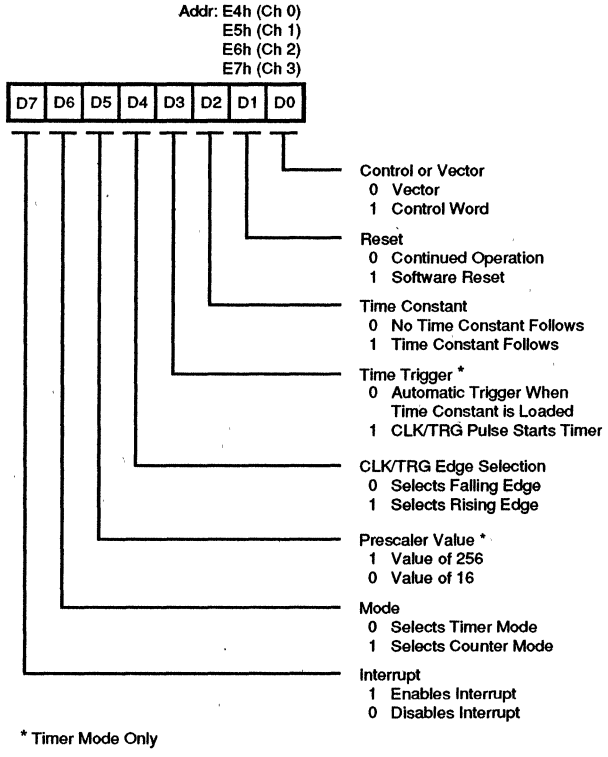


Figure 46. CTC Channel Control Word

This register has the following fields:

Bit D7. *Interrupt Enable.* This bit enables the interrupt logic so that an internal INT is generated at zero count Interrupts are programmed in either mode and may be enabled or disabled at any time.

Bit D6. *Mode Bit.* This bit selects either Timer Mode or Counter Mode.

Bit D5. *Prescaler Factor.* This bit selects the prescaler factor for use in the timer mode. Either divide-by-16 or divide-by-256 is available.

Bit D4. *Clock/Trigger Edge Selector.* This bit selects the active edge of the CLK/TRG input pulses.

Bit D3. *Timer Trigger.* This bit selects the trigger mode for timer operation. Either automatic or external trigger may be selected.

Bit D2. *Time Constant.* This bit indicates that the next word programmed is time constant data for the downcounter.

Bit D1. *Software Reset.* Writing a "1" to this bit indicates a software reset operation, which stops counting activities until another time constant word is written.

CTC CONTROL REGISTERS (Continued)

Time Constant Word

Before a channel can start counting, it must receive a time constant word. The time constant value may be anywhere between 1 and 256, with "0" being accepted as a count of 256 (Figure 47).

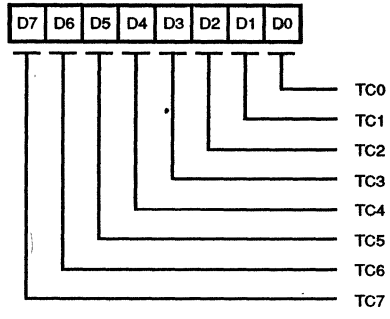


Figure 47. CTC Time Constant Word

Interrupt Vector Word

If one or more of the CTC channels have interrupt enabled, then the Interrupt Vector Word is programmed. Only the five most significant bits of this word are programmed, and bit D0 must be "0". Bits D2-D1 are automatically modified by the CTC channels after responding with an interrupt vector (Figure 48).

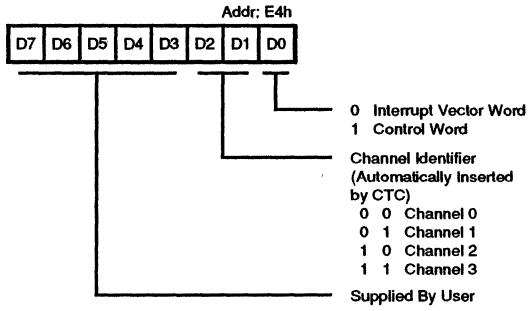


Figure 48. CTC Interrupt Vector Word

SCC REGISTERS

For more detailed information, please refer to the Z8030/ Z8530 SCC Technical Manual.

Note:
The Address for the Control/Status Register is E8h. The Address for the Data Register is E9h.

Read Registers

The SCC contains eight read registers. To read the contents of a register (rather than RR0), the program must first initialize a pointer to WR0 in exactly the same manner as a write operation The next I/O read cycle will place the contents of the selected read registers onto the data bus (Figure 49).

Table 2. SCC Read Registers

Bit	Description
RR0	Transmit and Receive buffer status and external status.
RR1	Special Receive Condition status.
RR2	Interrupt vector (modified if VIS Bit in WR9 is set).
RR3	Interrupt pending bits.
RR6	SDLC FIFO byte counter lower byte (only when enabled).
RR7	SDLC FIFO byte count and status (only when enabled).
RR8	Receive buffer.
RR10	Miscellanous status bits.
RR12	Lower byte of baud rate.
RR13	Upper byte of baud rate generator time constant.
RR15	External Status interrupt information.

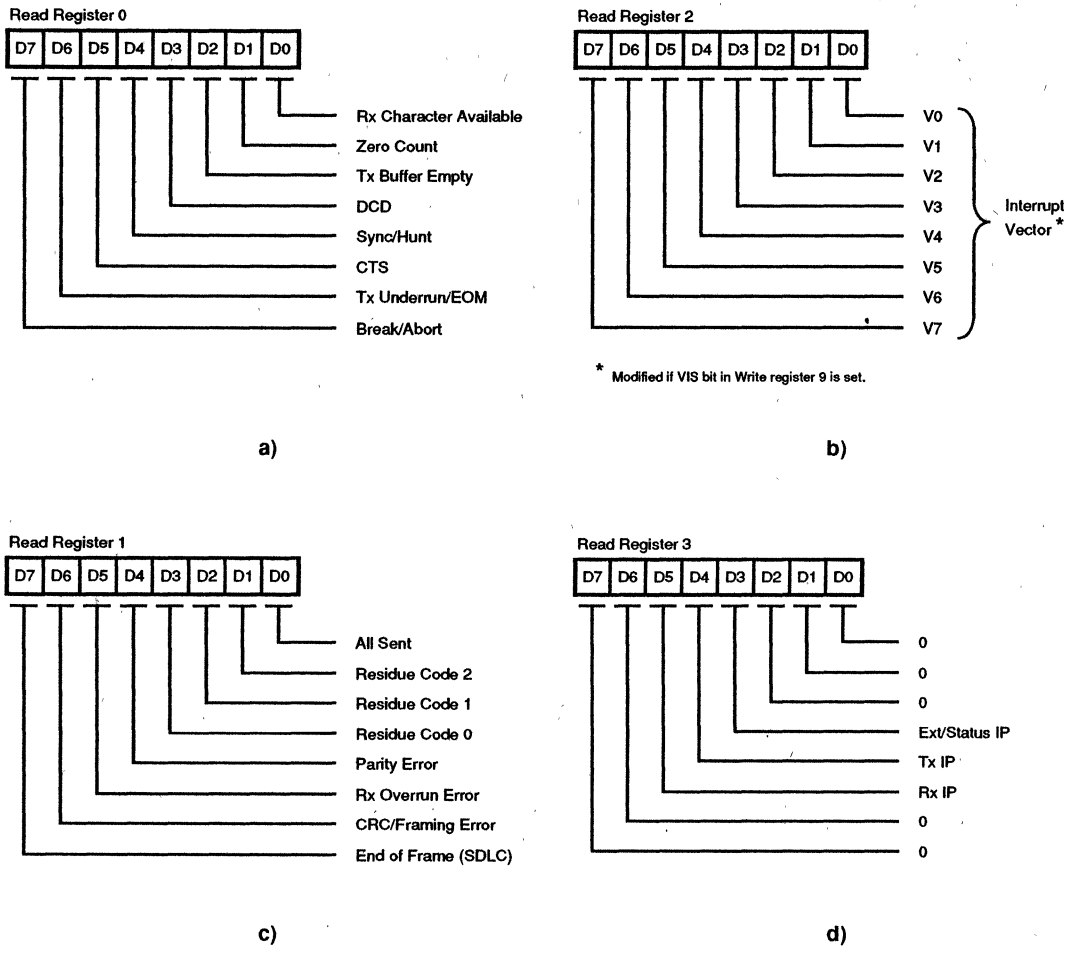
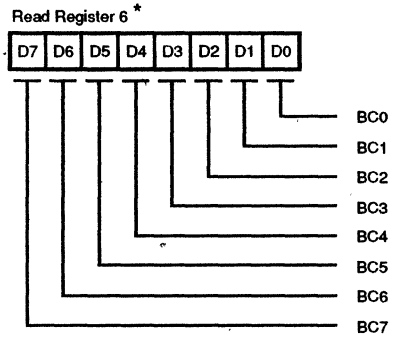
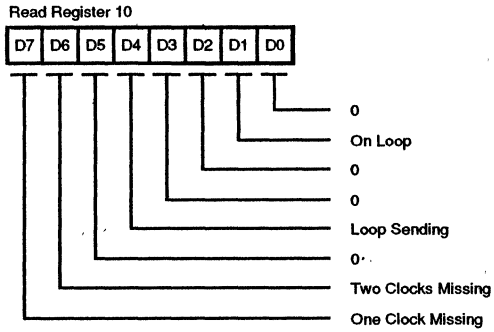


Figure 49. SCC Read Register Bit Functions

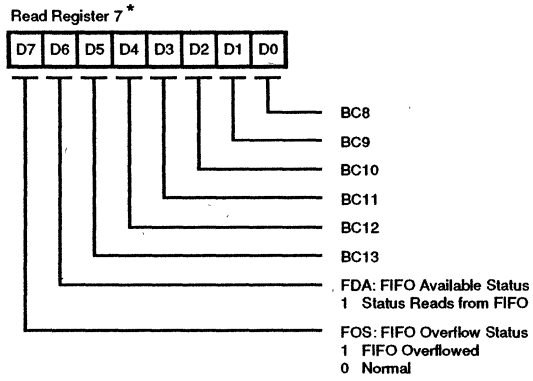


* Can only be accessed if the SDLC FIFO enhancement is enabled (WR15 bit D2 set to 1)

e) SDLC FIFO Status and Byte Count (LSB)

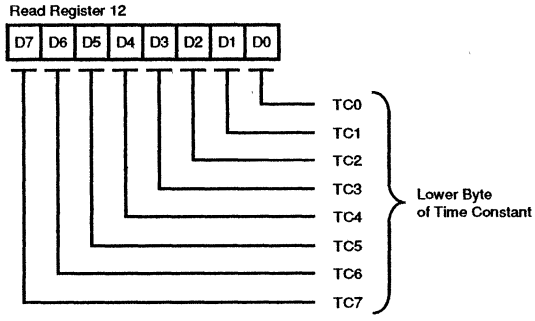


g)



* Can only be accessed if the SDLC FIFO enhancement is enabled (WR15 bit D2 set to 1)

f) SDLC FIFO Status and Byte Count (MSB)



h)

Figure 49. SCC Read Register Bit Functions (Continued)

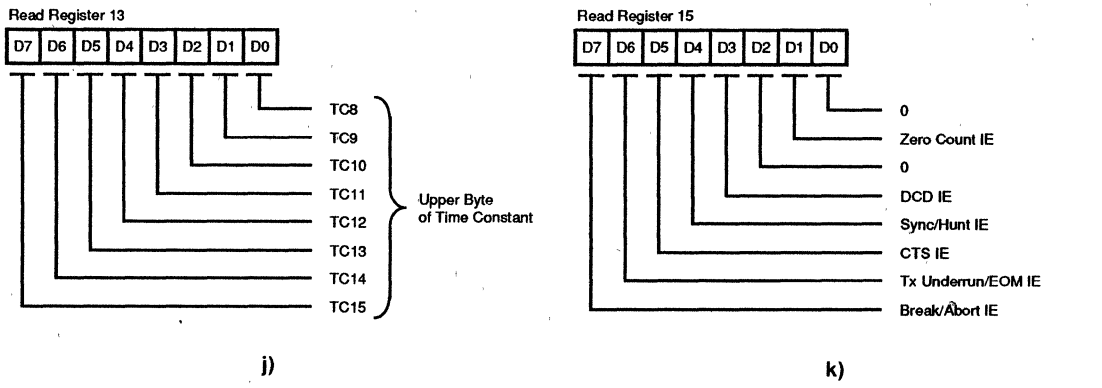


Figure 49. SCC Read Register Bit Functions (Continued)

Write Registers

The SCC contains fifteen write registers that are programmed to configure the operating modes of the channel. With the exception of WR0, programming the write registers is a two step operation. The first operation is a pointer written to WR0 that points to the selected register. The second operation is the actual control word that is written into the register to configure the SCC channel (Figure 50).

Table 2. SCC Write Registers

Bit	Description
WR0	Register Pointers, various initialization commands
WR1	Transmit and Receive interrupt enables, WAIT/DMA commands
WR2	Interrupt Vector
WR3	Receive parameters and control modes
WR4	Transmit and Receive modes and parameters
WR5	Transmit parameters and control modes
WR6	Sync Character or SDLC address
WR7	Sync Character or SDLC flag

Bit	Description
WR8	Transmit buffer
WR9	Master Interrupt control and reset commands
WR10	Miscellaneous transmit and receive control bits
WR11	Clock mode controls for receive and transmit
WR12	Lower byte of baud rate generator
WR13	Upper byte of baud rate generator
WR14	Miscellaneous control bits
WR15	External status interrupt enable control

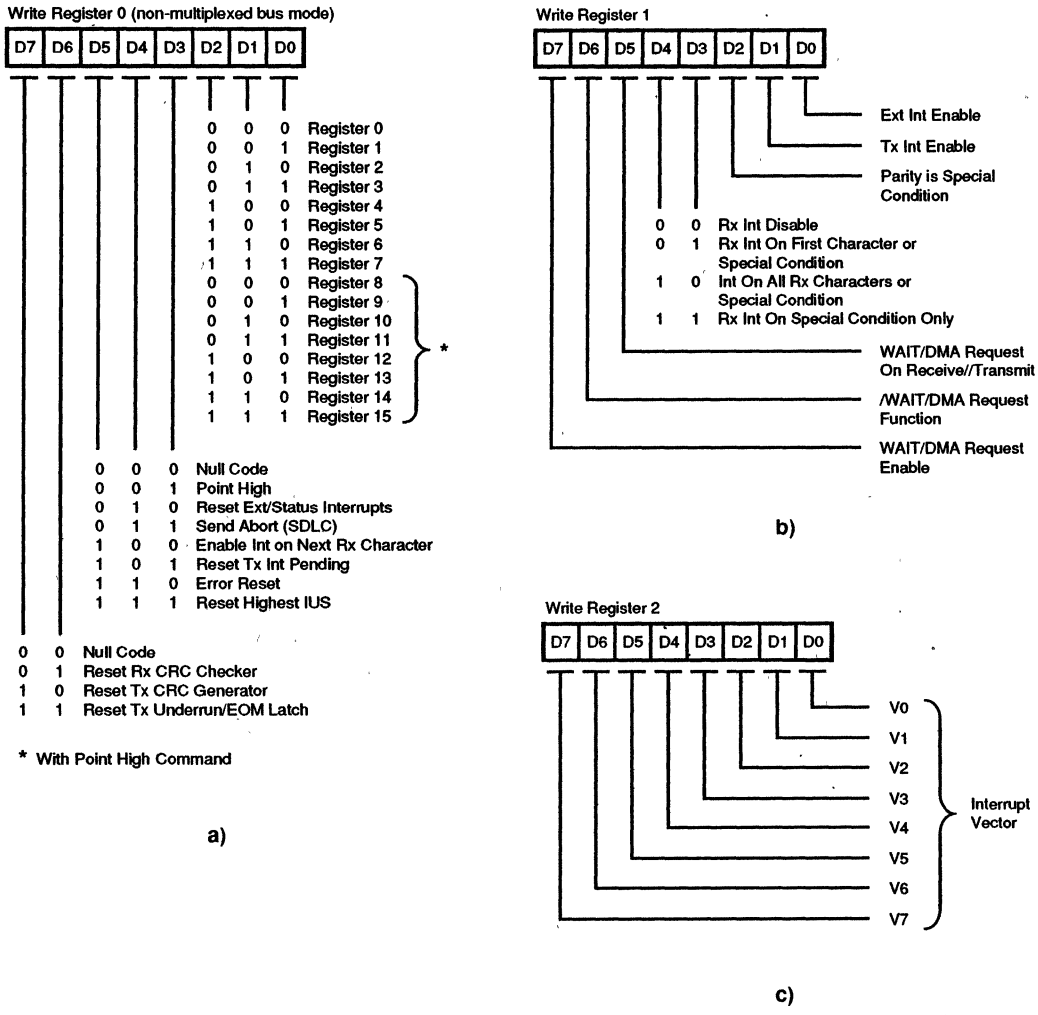
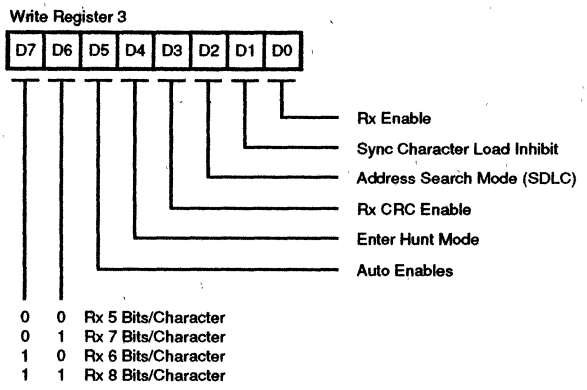
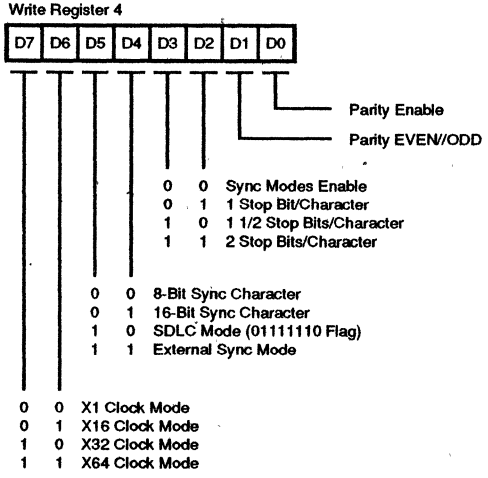


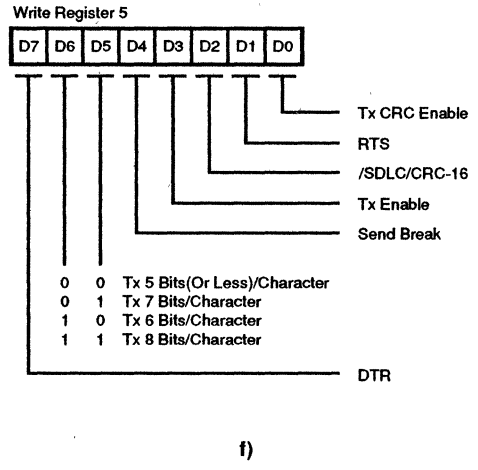
Figure 50. Write Register Bit Functions



d)

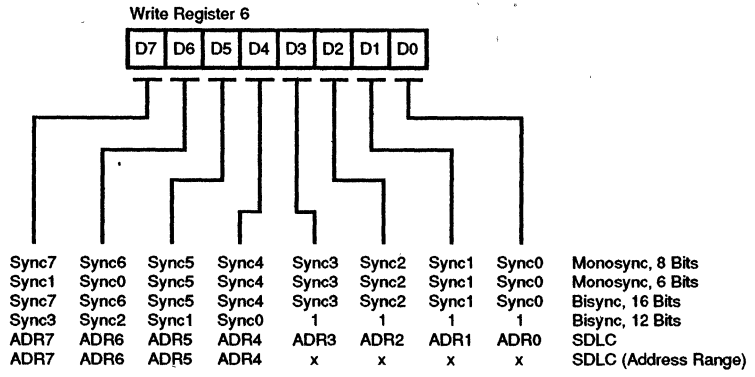


e)

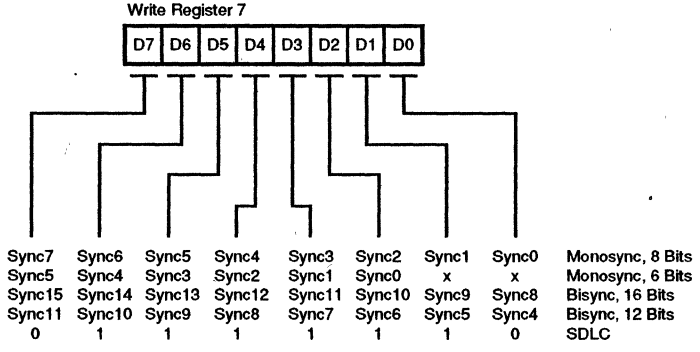


f)

Figure 50. Write Register Bit Functions (Continued)



g)



h)

Figure 50. Write Register Bit Functions (Continued)

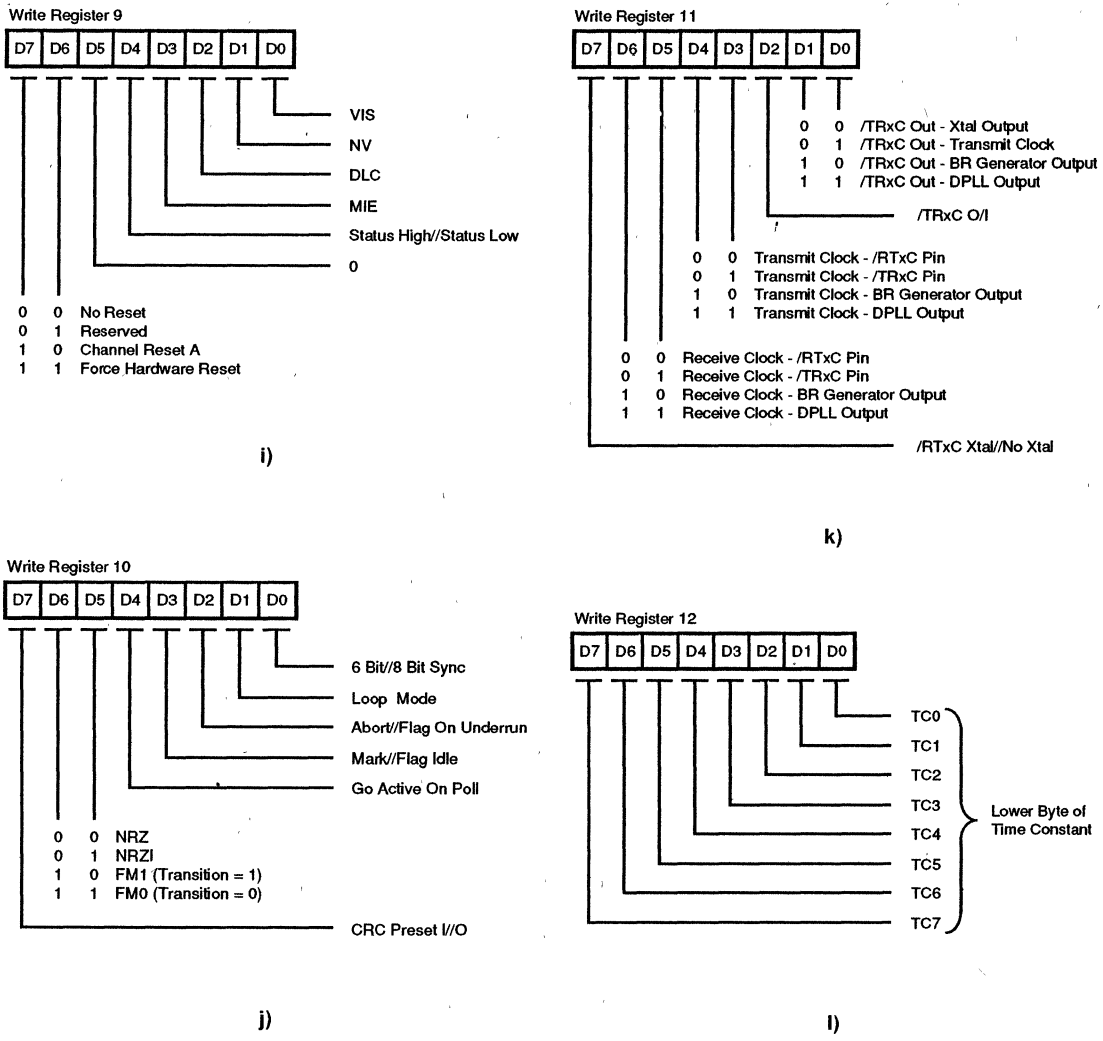
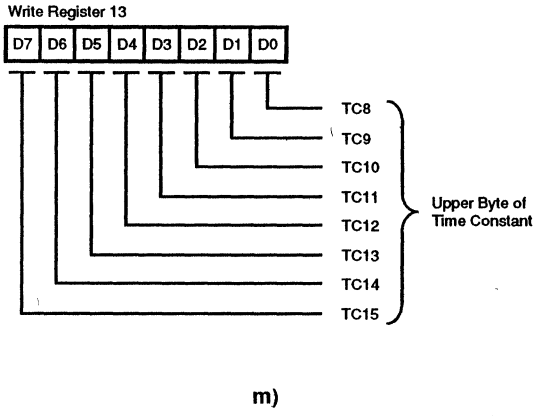
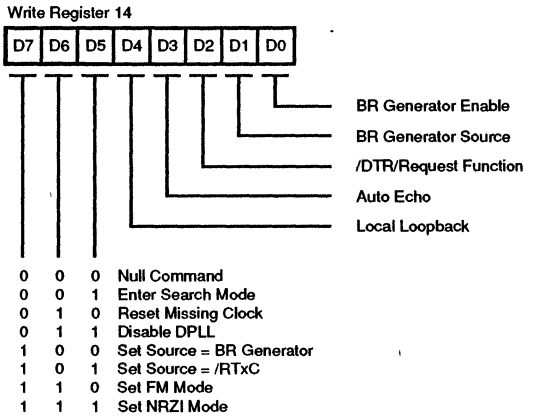


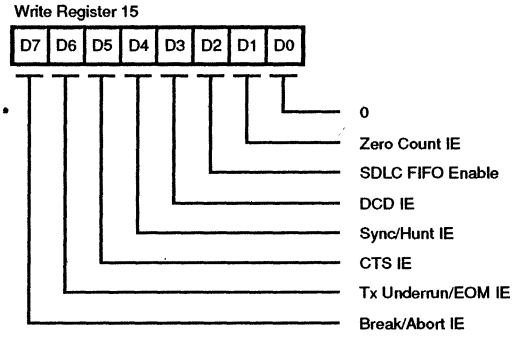
Figure 50. Write Register Bit Functions (Continued)



m)



n)



o)

Figure 50. Write Register Bit Functions (Continued)

PIA CONTROL REGISTERS

PIA1 Data Direction Register (P1DDR, I/O Address E0h), PIA1 Data Port (P1DP, I/O address E1h), PIA2 Data Direction Register (P2DDR, I/O Address E2h) and PIA2 Data Register (P2DP, I/O Address E3h) These four registers are shown in Figures 51-54. Note that if the CTC/PIA bit in the System Configuration Register is set to one, the CTC I/O functions override the PIA1 function, and programming of P1DDR is ignored.

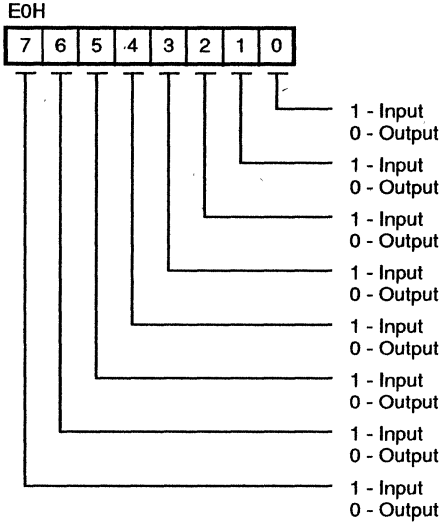


Figure 51. PIA 1 Data Direction Register

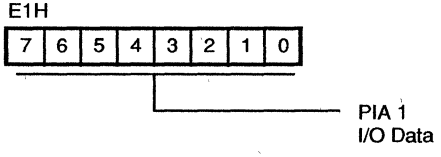


Figure 52. PIA 1 Data Register

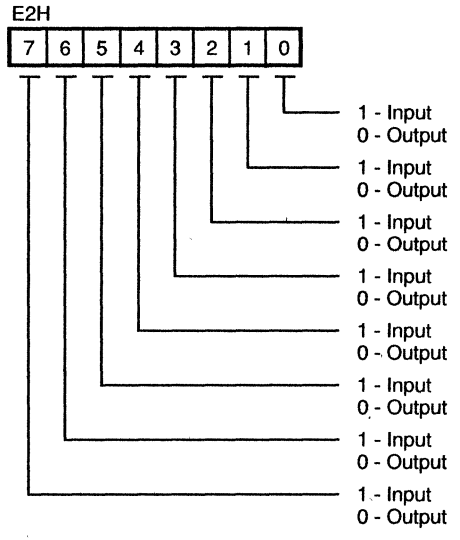


Figure 53. PIA 2 Data Direction Register

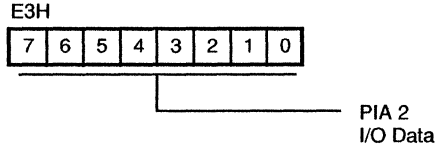


Figure 54. PIA 2 Data Register

The Data Port is the register to/from the 8-bit parallel port. At power on Reset, they are initialized to 1.

The Data Direction Register has eight control bits Individual bits specify each bit's direction. When the bit is set to a "1", the bit becomes an input, otherwise it is an output On reset, these registers are initialized to 1, resulting in all lines being inputs

REGISTERS FOR SYSTEM CONFIGURATION

There are four registers to determine system configuration with the Z181. These registers are: RAM upper boundary address register (RAMUBR, I/O address EAh), RAM lower boundary address register (RAMLBR, I/O address EBh), ROM address boundary register (ROMBR, I/O address ECh) and System Configuration Register (SCR, I/O address EDh).

ROM Address Boundary Register
(ROMBR, I/O Address ECh)
This register specifies the address range for the /ROMCS signal. When accessed memory addresses are less than or equal to the value programmed in this register, the /ROMCS signal is asserted (Figure 55).

The A18 signal from the CPU is obtained before it is multiplexed with "TOUT". This signal can be forced to "1" (inactive state) by setting Bit D5 of the System Configuration Register, to allow the user to overlay the RAM area over the ROM area. At power-up reset, this register contains all 1s so that /ROMCS is asserted for all addresses

RAM Lower Boundary Address Register (RAMLBR, I/O Address EBh) **and RAM Upper Boundary Address Register** (RAMUBR, I/O Address EAh)
These two registers specify the address range for the /RAMCS signal. When accessed memory addresses are less than or equal to the value programmed in the RAMUBR and greater than or equal to the value programmed in the RAMLBR, /RAMCS is asserted. (Figure 13) The A18 signal from the CPU is taken before it is multiplexed with "TOUT". In the case that these register are programmed to overlap, /ROMCS takes priority over /RAMCS (/ROMCS is asserted and /RAMCS is inactive).

Chip Select signals are going active for the address range:

/ROMCS: (ROMBR) ≥ A19-A12 ≥ 0
/RAMCS: (RAMUBR) ≥ A19-A12 > (RAMLBR)

These registers are set to "FFh" at power-on Reset, and the boundary addresses of ROM and RAM are the following:

ROM lower boundary address
(fixed) = 00000h

ROM upper boundary address
(ROMBR register) = 0FFFFFh

RAM lower boundary address
(RAMLBR register) = 0FFFFFh

RAM upper boundary address
(RAMUBR register) = 0FFFFFh

Since /ROMCS takes priority over /RAMCS, the latter will never be asserted until the value in the ROMBR and RAMLBR registers are re-initialized to lower values.

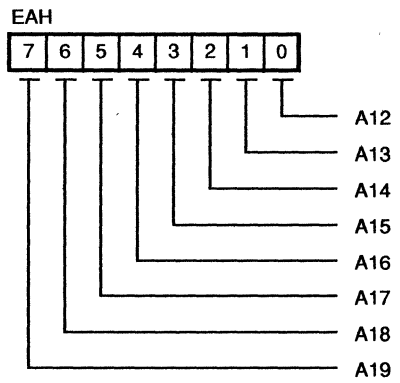


Figure 55. RAM Upper Boundary Register

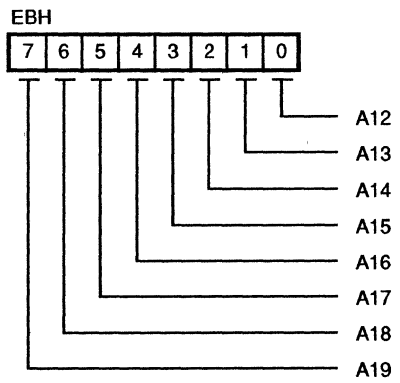


Figure 56. RAM Lower Boundary Register

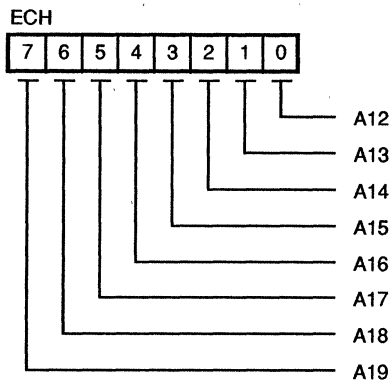


Figure 57. ROM Boundary Register

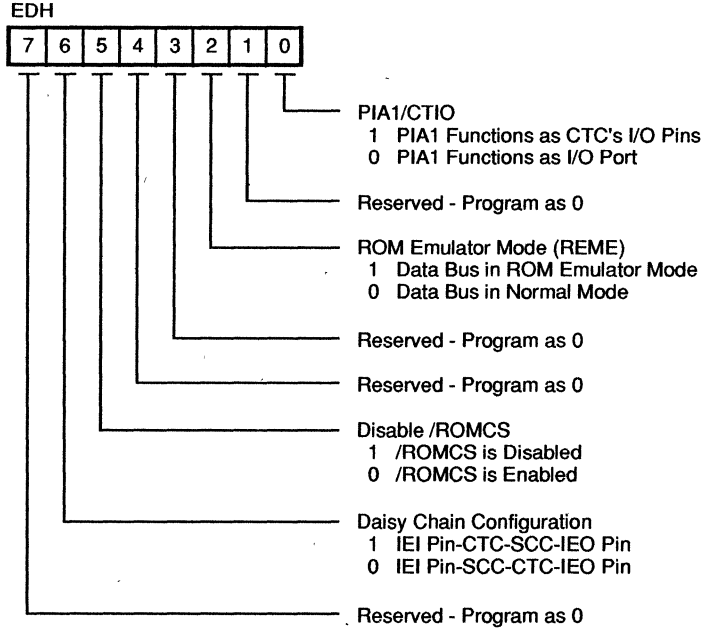


Figure 58. System Configuration Register

REGISTERS FOR SYSTEM CONFIGURATION (Continued)

System Configuration Register (I/O address EDh)
This register is to determine the functionality of PIA1 and the Interrupt Daisy-Chain Configuration (Figure 13). This register has the following control bits:

Bit D7. Reserved and should be programmed as "0".

Bit D6. *Daisy-Chain Configuration.* Determines the arrangement of the interrupt priority daisy chain.

When this bit is set to "1", priority is as follows:

IEI pin - CTC - SCC - IEO pin

When this bit is "0", priority is as follows:

IEI pin - SCC - CTC - IEO pin

This bit's default (after Reset) is 0.

Bit D5. *Disable /ROMCS.* When this bit is set to "1". /ROMCS is forced to a "1" regardless of the status of the address decode logic. This bit's default (after Reset) is 0 and /ROMCS function is enabled.

Bit D4-D3. Reserved and should be programmed as "00".

Bit D2. *ROM Emulator Mode Enable.* When this bit is set to a 1, the Z181 is in "ROM emulator mode". In this mode, bus direction for certain transaction periods are set to the opposite direction to export internal bus transactions outside the Z80181. This allows the use of ROM emulators/ logic analyzers for applications development. This bit's default (after Reset) is 0.

Bit D1. Reserved and shall be programmed as "0".

Bit D0. *CTC/PIA 1.* When this bit is set to "1", PIA1 functions as the CTC's I/O pins. This bit's default (after Reset) is 0.

DATA BUS DIRECTION

Table 3 shows the state of the SAC's data bus for the condition that the SAC is bus master.

Table 3. Data Bus Direction (Z181 Is Bus Master)

I/O And Memory Transactions

	I/O Write To On-Chip Peripherals (SCC/CTC/ PIA1/PIA2)	I/O Read From On-Chip Peripherals (SCC/CTC/ PIA1/PIA2)	I/O Write To Off-Chip Peripheral	I/O Read From Off-Chip Peripheral	Write To Memory	Read From Memory	Refresh	Z80181 Idle Mode
Z80181 Data Bus (REME Bit = 0)	Out	Z	Out	In	Out	In	Z	Z
Z80181 Data Bus (REME Bit = 1)	Out	Out	Out	In	Out	In	Z	Z

Interrupt Acknowledge Transaction

	Intack For On-chip Peripheral (SCC/CTC)	Intack For Off-chip Peripheral	
Z80181 Data Bus (REME Bit = 0)	Z	In	
Z80181 Data Bus (REME Bit = 1)	Out	In	

DATA BUS DIRECTION (Continued)

Table 4 shows the state of the SAC's data bus for the condition that the Z80181 is NOT bus master.

Table 4. Data Bus Direction for External Bus Master (Z0181 Is Not Bus Master)

I/O And Memory Transactions

	I/O Read From On-Chip Peripherals (SCC/CTC/ PIA1/PIA2)	I/O Write To On-Chip Peripherals (SCC/CTC/ PIA1/PIA2)	I/O Read From Off-Chip Peripheral	I/O To Off-Chip Peripheral	Write From Memory	Read Memory	Refresh	Ext. Bus-Master Is Idle
Z80181 Data Bus (REME Bit = 0)	In	Out	Z	Z	Z	In	Z	Z
Z80181 Data Bus (REME Bit = 1)	In	Out	Z	Z	Z	In	Z	Z

Interrupt Acknowledge Transaction

	Intack For On-chip Peripheral (SCC/CTC)	Intack For Off-chip Peripheral
Z80181 Data Bus (REME Bit = 0)	Out	In
Z80181 Data Bus (REME Bit = 1)	Out	In

The word "OUT" means that the Z181 data bus direction is in output mode, "IN" means input mode, and "HI-Z" means high impedance.

"REME" stands for "ROM Emulator Mode" and is the status of D2 bit in the System Configuration Register

ABSOLUTE MAXIMUM RATINGS

Voltage on Vcc with respect to Vss .....-0.3V to +7.0V
Voltages on all inputs
with respect to Vss .....-0.3V to Vcc+0.3V

Operating Ambient

Temperature .....See Ordering Information
Storage Temperature .....-65°C to + 150°C

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 sections below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND (0V). Positive current flows into the referenced pin.

Available operating temperature range is:
E = -40°C to 100°C

Voltage Supply Range:
+4.50V ≤ Vcc ≤ + 5.50V

All AC parameters assume a load capacitance of 100 pf. Add 10 ns delay for each 50 pf increase in load up to a maximum of 150 pf for the data bus and 100 pf for address and control lines. AC timing measurements are referenced to 1.5 volts (except for clock, which is referenced to the 10% and 90% points). Maximum capacitive load for CLK is 125 pf.

The Ordering Information section lists temperature ranges and product numbers. Refer to the Literature List for additional documentation.

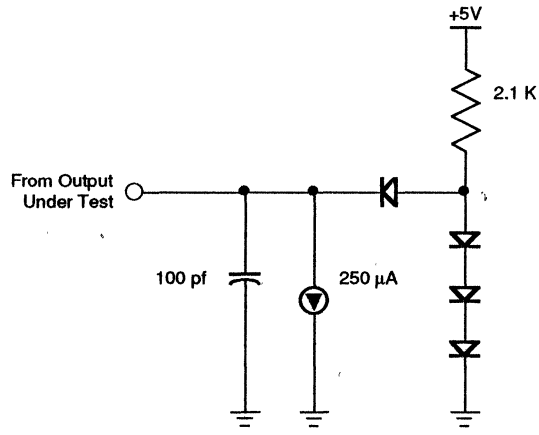


Figure 59. Standard Test Circuit

DC CHARACTERISTICS
Z80181

Symbol	Parameter	Min	Typ	Max	Unit	Condition
V_{IH1}	Input "H" Voltage /RESET, EXTAL, /NMI	V_{CC} -0.6		V_{CC} +0.3	V	
V_{IH2}	Input "H" Voltage Except /RESET, EXTAL, /NMI	2.O		V_{CC} +0.3	V	
V_{IL1}	Input "L" Voltage /RESET, EXTAL, /NMI	-0.3		0.6	V	
V_{IL2}	Input "L" Voltage Except /RESET, EXTAL, /NMI	-0.3		0.8	V	
V_{OH}	Output "H" Voltage All outputs.	2.4 V_{CC} - 1.2			V	I_{OH} = -200 µA I_{OH} = -20 µA
V_{OL}	Output "L" Voltage All outputs.			0.45	V	I_{OL} = 2.2 mA
I_{IL}	Input Leakage Current All Inputs Except XTAL, EXTAL			10	µA	V_{IN} = 0.5 - V_{CC} -0 5
I_{TL}	Tri-State Leakage Current			10	µA	V_{IN} = 0.5 - V_{CC} -0 5
I_{CC}*	Power Dissipation*			100	mA	f = 12.5 MHz
	(Normal Operation)		25	80		f = 10 MHz
	Power Dissipation*			50		f = 12.5 MHz
	(SYSTEM STOP mode)		6.3	40		f = 10 MHz
Cp	Pin Capacitance			12	pf	V_{IN} = 0V, f = 1 MHz T_A = 25°C

* V_{IH} Min = V_{CC} -1.0V, V_{IL} Max = 0.8V (all output terminals are at no load.)
V_{CC} = 5.00V

AC CHARACTERISTICS
Z180 MPU Timing

Figures 60-68 show the timing for the Z181 MPU and the referenced parameters appear in Table A.

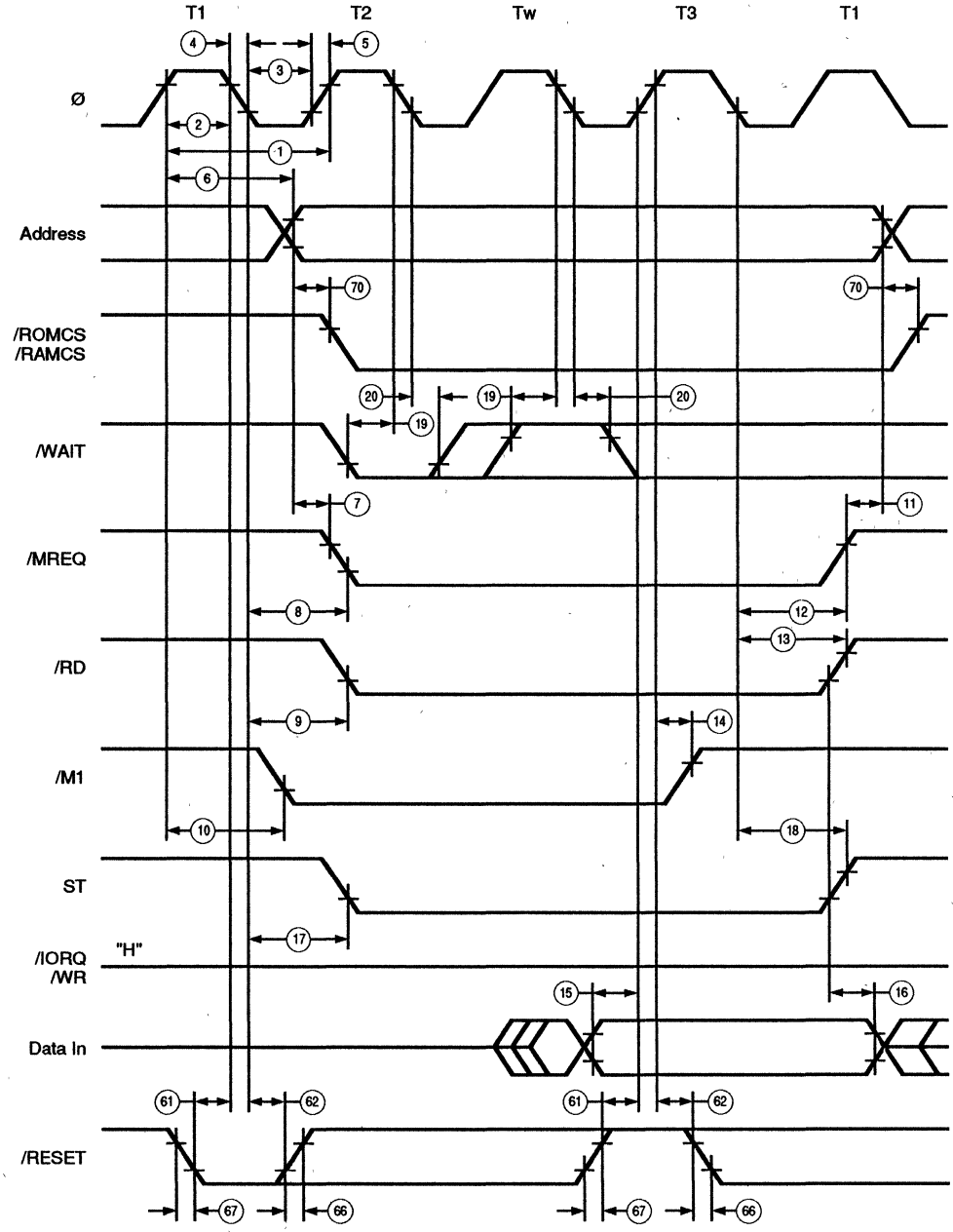
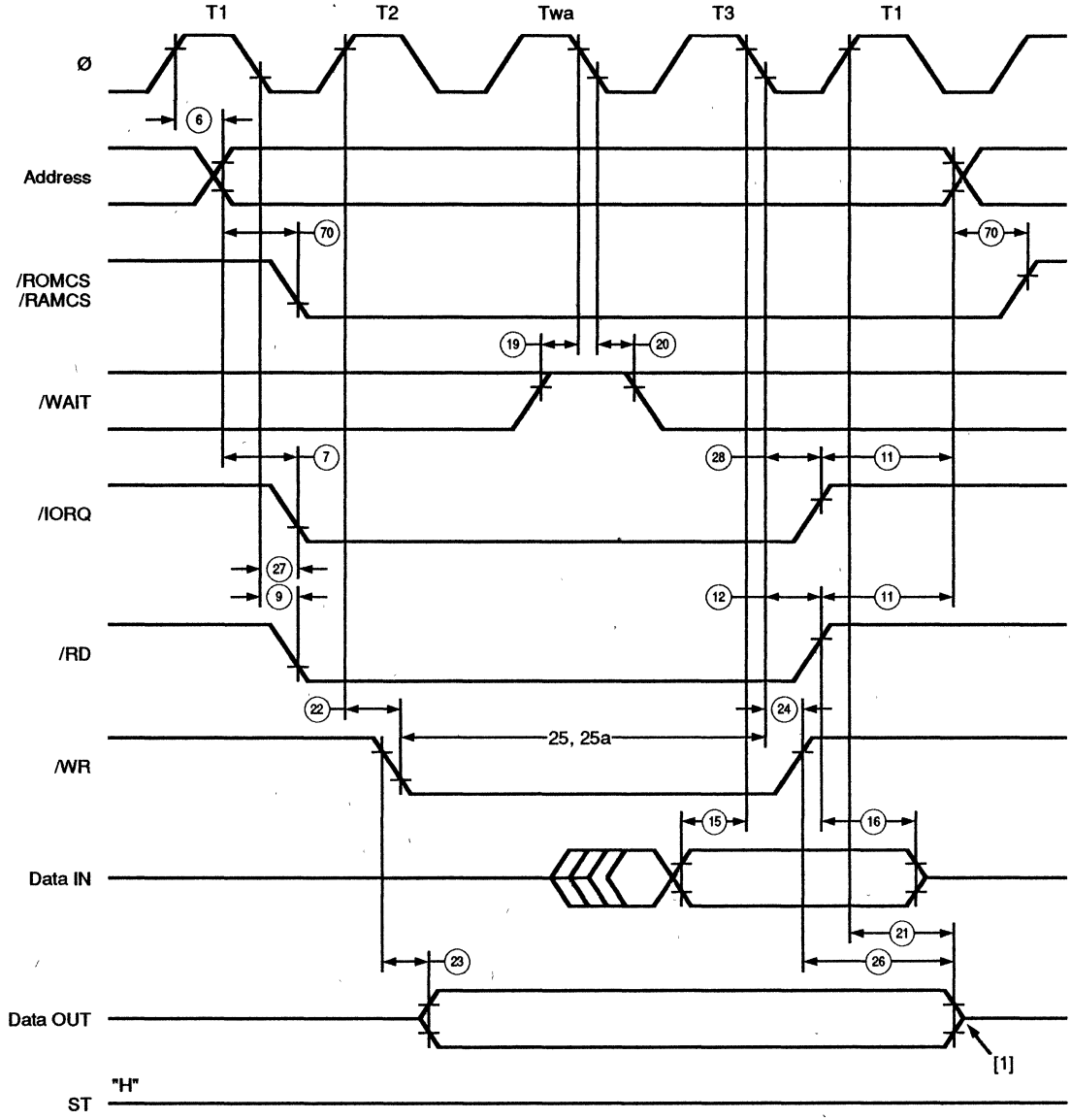


Figure 60a. Op-code Fetch Cycle

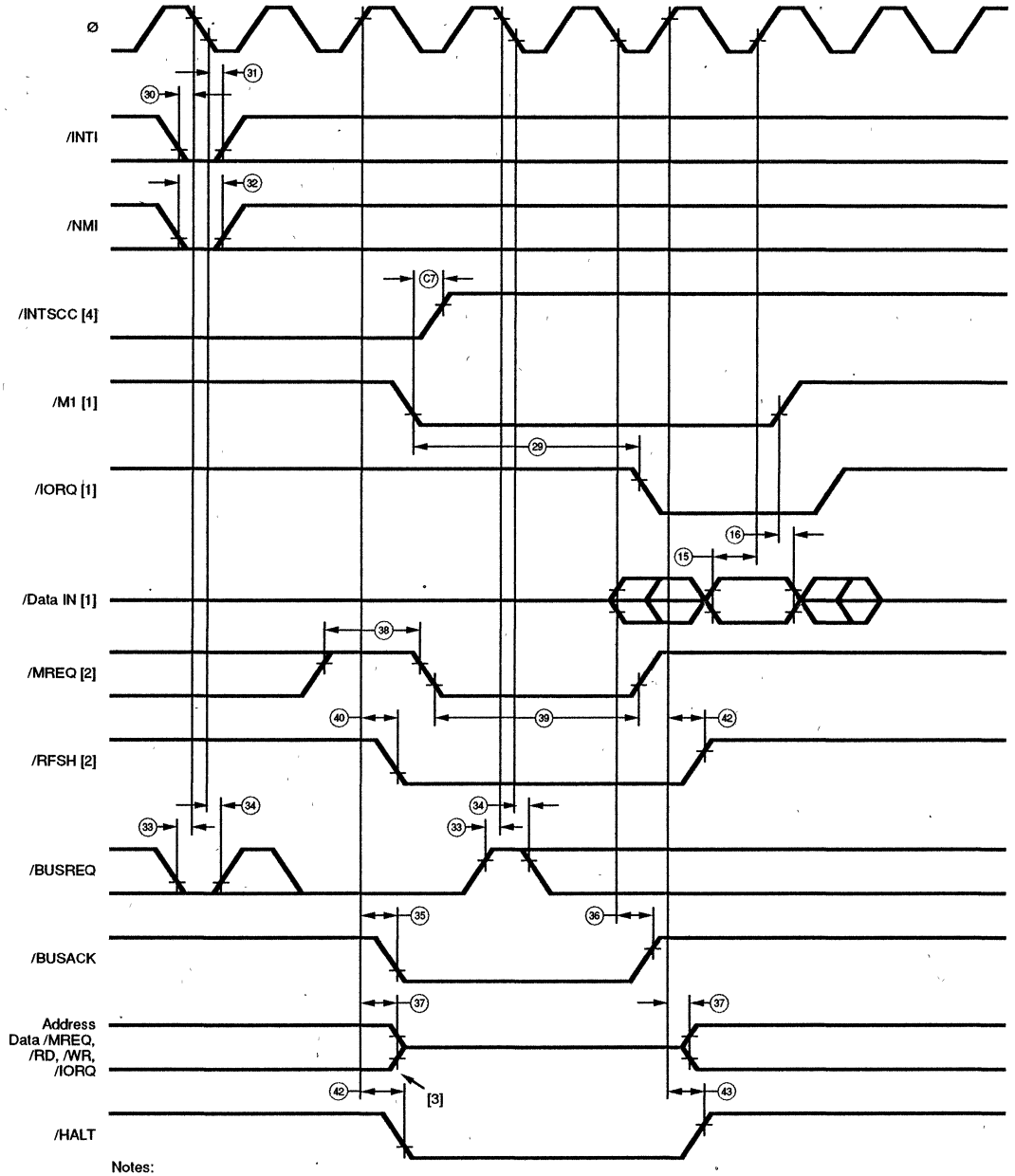
AC CHARACTERISTICS (Continued)

Z180 MPU Timing



[1] Output buffer is off at this point.
[2] Memory Read/Write cycle timing is the same as this figure, except there is no automatic wait status (Twa), and /MREQ is active instead of /IORQ.

Figure 60b. I/O Read/Write, Memory Read/Write Timing



Notes:
[1] During /INT0 acknowledge cycle
[2] During refresh cycle
[3] Output buffer is off at this point
[4] Refer to Table C, parameter 7

Figure 61. CPU Timing
(/INT0 Acknowledge Cycle, Refresh Cycle, BUS RELEASE Mode
HALT Mode, SLEEP Mode, SYSTEM STOP Mode)

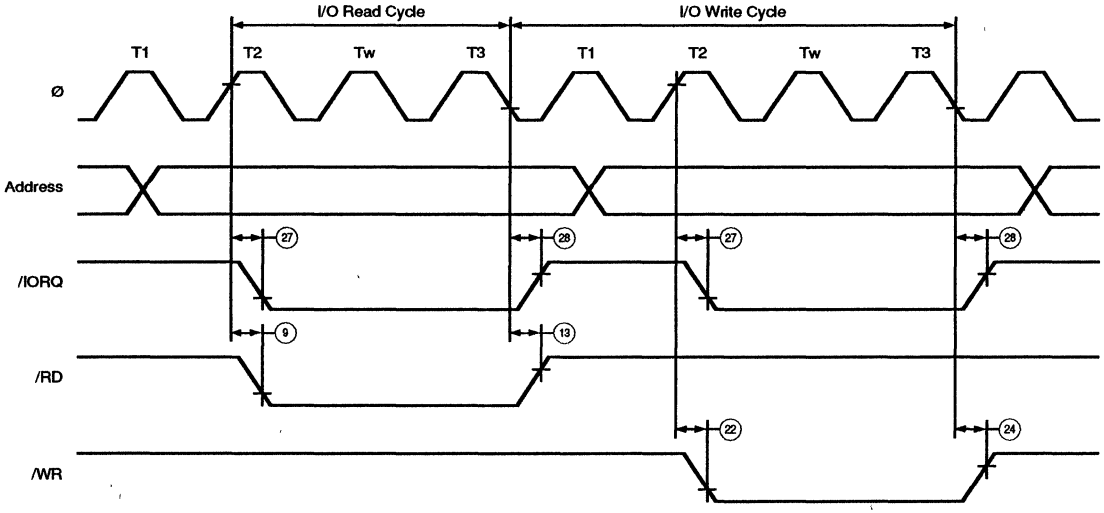
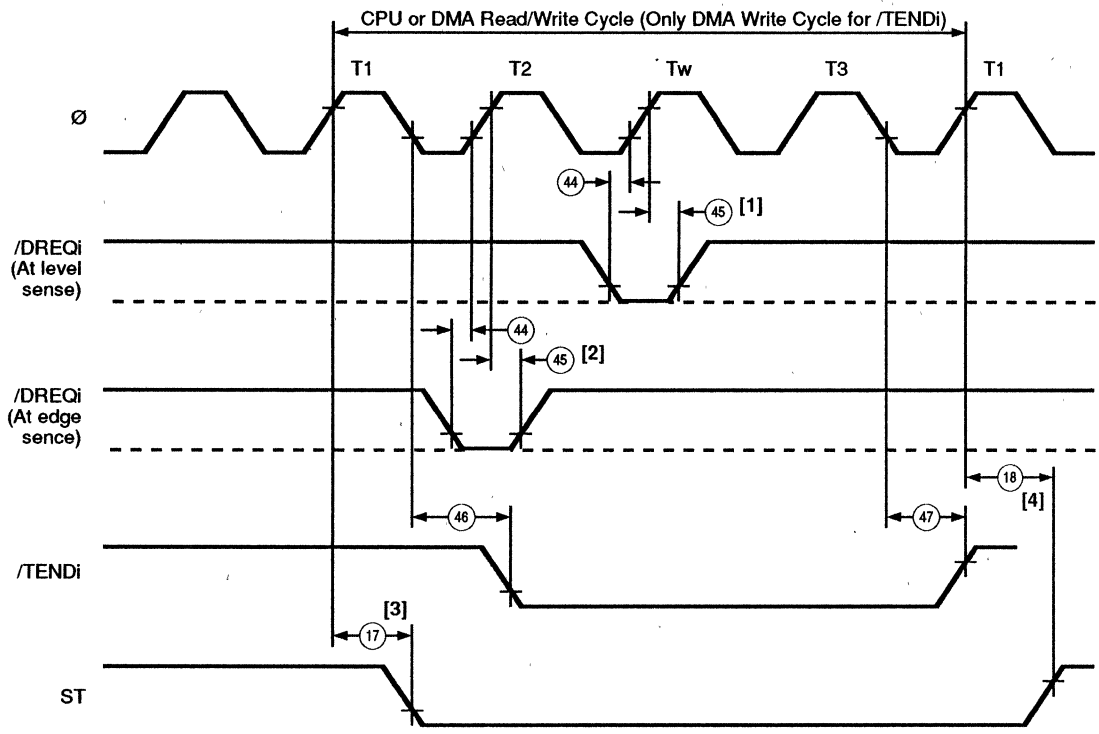
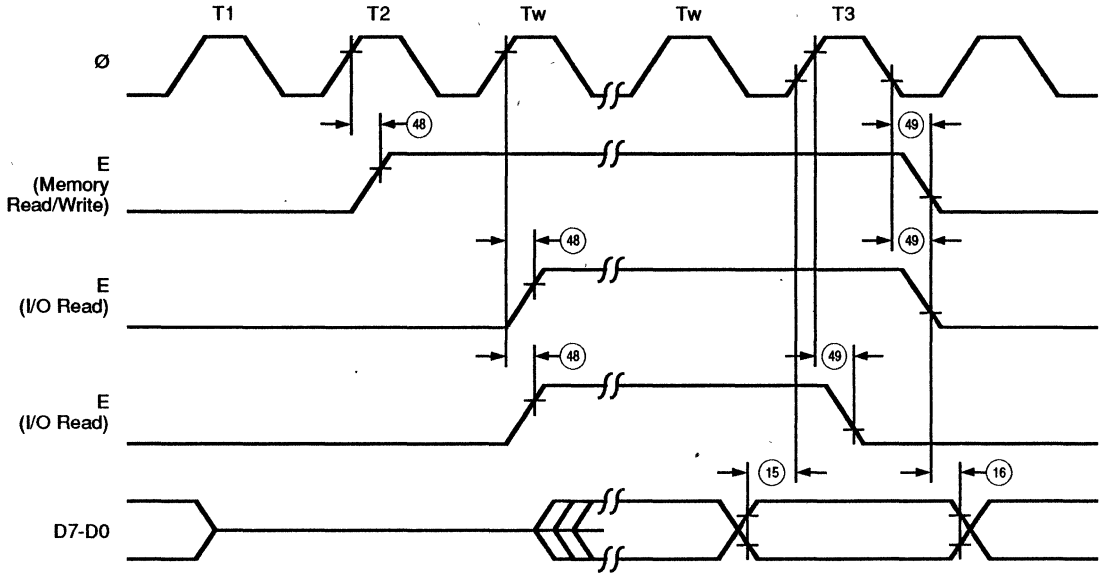


Figure 62. CPU Timing (/IOC = 0)
(I/O Read Cycle, I/O Write Cycle)

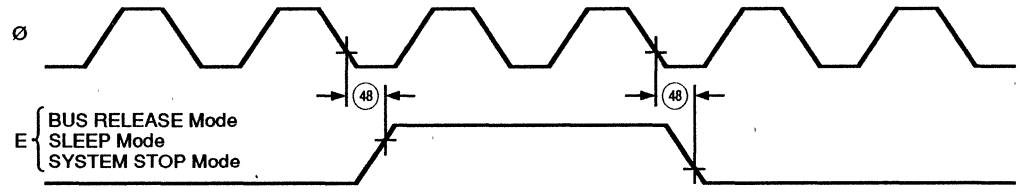


DMA Control Signals
[1] tDRQS and tDRQH are specified for the rising edge of clock followed by T3.
[2] tDRQS and tDRQH are specified for the rising edge of clock.
[3] DMA cycle starts.
[4] CPU cycle starts.

Figure 63. DMA Control Signals



a) E Clock Timing
(Memory Read/Write Cycle, I/O Read/Write Cycle)



b) E Clock Timing
(BUS RELEASE Mode, SLEEP Mode, SYSTEM STOP Mode)

Figure 64. E Clock Timing

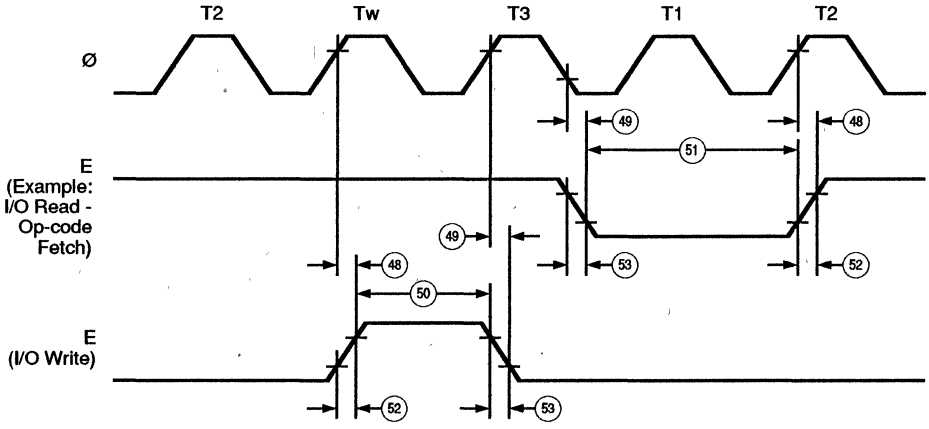


Figure 65. E Clock Timing
(Minimum timing example of PWEL and PWEH)

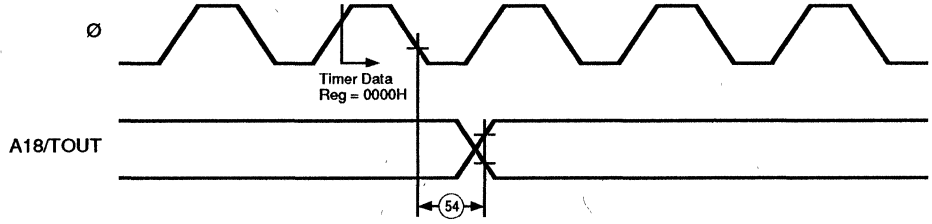


Figure 66. Timer Output Timing

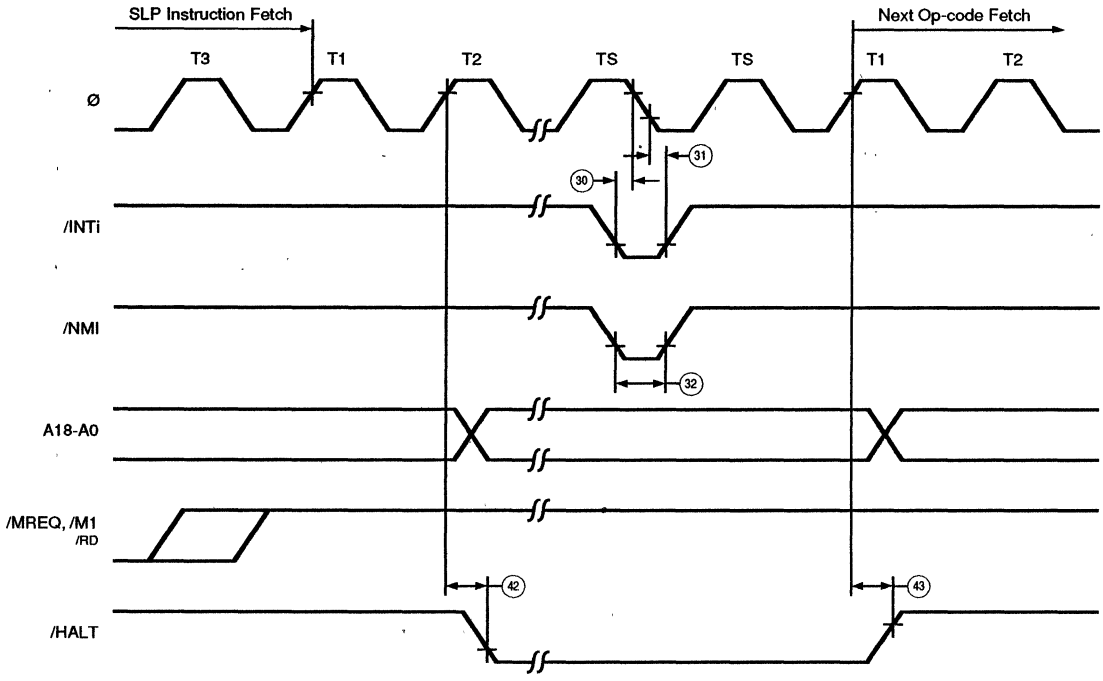


Figure 67. SLP Execution Cycle

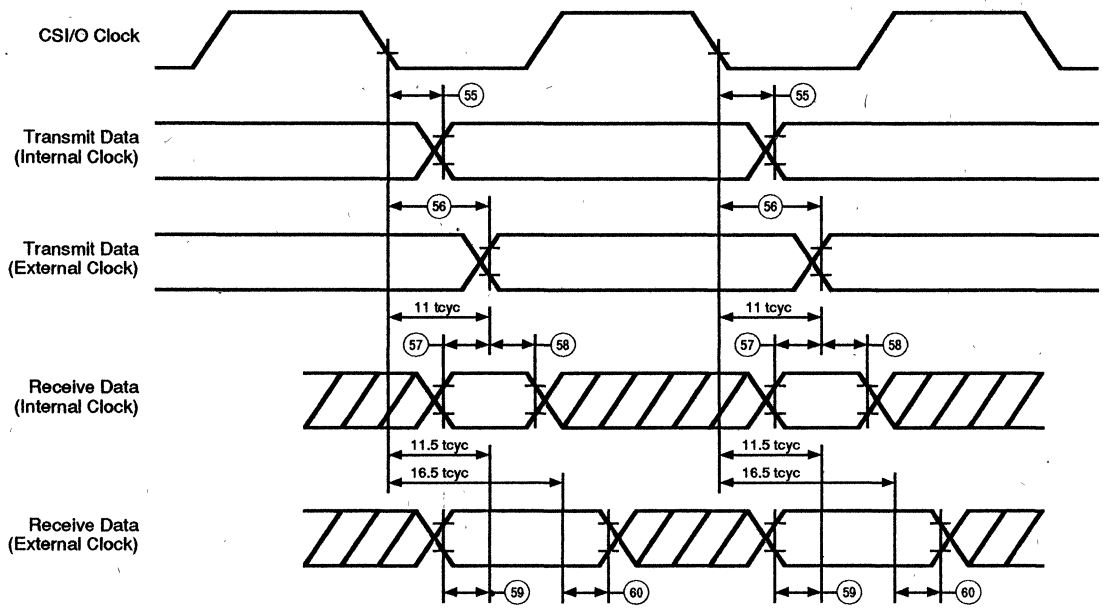


Figure 68. CSI/O Receive/Transmit Timing

Table A. Z180 CPU & 180 Peripherals Timing

No	Symbol	Parameter	Z8018110		Z8018112		Unit	Note
			Min	Max	Min	Max		
1	tcyc	Clock Cycle Time	100	2000	80	2000	ns	
2	tCHW	Clock Pulse Width (High)	40		30		ns	
3	tCLW	Clock Pulse Width (Low)	40		30		ns	
4	tcf	Clock Fall Time		10		10	ns	
5	tcr	Clock Rise Time		10		10	ns	
6	tAD	Address Valid from Clock Rise		70		40	ns	
7	tAS	Address Valid to /MREQ, /IORQ Fall	10		10		ns	
8	tMED1	Clock Fall to /MREQ Fall Delay		50		45	ns	
9	tRDD1	Clock Fall to /RD Fall (/IOC=1)		50		45	ns	
		Clock Rise to /RD Fall (/IOC=0)		55		50	ns	
10	tM1D1	Clock Rise to /M1 Fall Delay		60		50	ns	

AC CHARACTERISTICS (Continued)
Z180 MPU Timing

Table A. Z180 CPU & 180 Peripherals Timing (Continued)

No	Symbol	Parameter	Z8018110 Min	Z8018110 Max	Z8018112 Min	Z8018112 Max	Unit	Note
11	tAH	Address Hold Time (/MREQ, /IORQ, /RD, /WR)	10		10		ns	
12	tMED2	Clock Fall to /MREQ Rise Delay		50		45	ns	
13	tRDD2	Clock Fall to /RD Rise Delay		50		45	ns	
14	tM1D2	Clock Rise to /M1 Rise Delay		60		50	ns	
15	tDRS	Data Read Setup Time	25		20		ns	
16	tDRH	Data Read Hold Time	0		0		ns	
17	tSTD1	Clock Fall to ST Fall		60		50	ns	
18	tSTD2	Clock Fall to ST Rise		60		50	ns	
19	tWS	/WAIT Setup Time to Clock Fall	30		20		ns	
20	tWH	/WAIT Hold time from Clock Fall	30		20		ns	
21	tWDZ	Clock Rise to Data Float Delay		60		60	ns	
22	tWRD1	Clock Rise to /WR Fall Delay		50		30	ns	
23	tWDO	/WR fall to Data Out Delay		10		10	ns	
24	tWRD2	Clock Fall to /WR Rise		50		45	ns	
25	tWRP	/WR Pulse Width (Memory Write Cycles)	110		85		ns	
25a		/WR Pulse Width (I/O Write Cycles)	210		165		ns	
26	tWDH	Write Data Hold Time from /WR Rise	10		10		ns	
27	tIOD1	Clock Fall to /IORQ Fall Delay (/IOC=1)		50		45	ns	
		Clock Rise to /IORQ Fall Delay (/IOC=0)		55		50	ns	
28	tIOD2	Clock Fall /IOQR Rise Delay		50		50	ns	
29	tIOD3	/M1 Fall to /IORQ Fall Delay	200		160		ns	
30	tINTS	/INT Setup Time to Clock Fall	30		20		ns	
31	tINTH	/INT Hold Time from Clock Fall	30		20		ns	
32	tNMIW	/NMI Pulse Width	80		60		ns	
33	tBRS	/BUSREQ Setup Time to Clock Fall	30		20		ns	
34	tBRH	/BUSREQ Hold Time from Clock Fall	30		20		ns	
35	tBAD1	Clock Rise to /BUSACK Fall Delay		60		50	ns	
36	tBAD2	Clock Fall to /BUSACK Rise Delay		60		50	ns	
37	tBZD	Clock Rise to Bus Floating Delay Time		80		60	ns	
38	tMEWH	/MREQ Pulse Width (High)	70		60		ns	
39	tMEWL	/MREQ Pulse Width (LOW)	80		60		ns	
40	tRFD1	Clock Rise to /RFSH Fall Delay		60		40	ns	
41	tRFD2	Clock Rise to /RFSH Rise Delay		60		40	ns	
42	tHAD1	Clock Rise to /HALT Fall Delay		50		30	ns	
43	tHAD2	Clock Rise to /HALT Rise Delay		50		30	ns	
44	tDRQS	/DREQi Setup Time to Clock Rise	30		20		ns	
45	tDRQH	/DREQi Hold Time from Clock Rise	30		20		ns	
46	tTED1	Clock Fall to /TENDi Fall Delay		50		50	ns	

Table A. Z180 CPU & 180 Peripherals Timing (Continued)

No	Symbol	Parameter	Z8018110 Min	Z8018110 Max	Z8018112 Min	Z8018112 Max	Unit	Note
47	tTED2	Clock Fall to /TENDi Rise Delay		50		50	ns	
48	tED1	Clock Rise to E Rise Delay		60		40	ns	
49	tED2	Clock Edge to E Fall Delay		60		40	ns	
50	PWEH	E Pulse Width (High)	55		45		ns	
51	PWEL	E Pulse Width (Low)	110		90		ns	
52	tEr	Enable Rise Time		20		10	ns	
53	tEf	Enable Fall Time		20		10	ns	
54	tTOD	Clock Fall to Timer Output Delay		150		120	ns	
55	tSTDI	CSI/O Tx Data Delay Time (Internal Clock Operation)		150		120	ns	
56	tSTDE	CSI/O Tx Data Delay Time (External Clock Operation)		7.5tcyc+150		7.5tcyc+120	ns	
57	tSRSI	CSI/O Rx Data Setup Time (Internal Clock Operation)	1		1		tcyc	
58	tSRHI	CSI/O Rx Data Hold Time (Internal Clock Operation)	1		1		tcyc	
59	tSRSE	CSI/O Rx Data Setup Time (External Clock Operation)	1		1		tcyc	
60	tSRHE	CSI/O Rx Data Hold Time (External Clock Operation)	1		1		tcyc	
61	tRES	/RESET Setup Time to Clock Fall	80		60		ns	
62	tREH	/RESET Hold Time from Clock Fall	50		45		ns	
63	tOSC	Oscillator Stabilization Time		20		20	ms	
64	tEXr	External Clock Rise Time (EXTAL)		25		20	ns	
65	tEXf	External Clock Fall Time (EXTAL)		25		20	ns	
66	tRr	/RESET Rise Time		50		50	ns	
67	tRf	/RESET Fall Time		50		50	ns	
68	tIr	Input Rise Time (Except EXTAL, /RESET)		100		80	ns	
69	tIf	Input Fall Time (Except EXTAL, /RESET)		100		80	ns	
70	TdCS(A)	Address Valid to /ROMCS, /RAMCS Valid Delay		20		20	ns	

AC CHARACTERISTICS (Continued)
CTC Timing

Figure 69 shows the timing for the on-chip CTC. Parameters referred to in this figure appear in Table B.

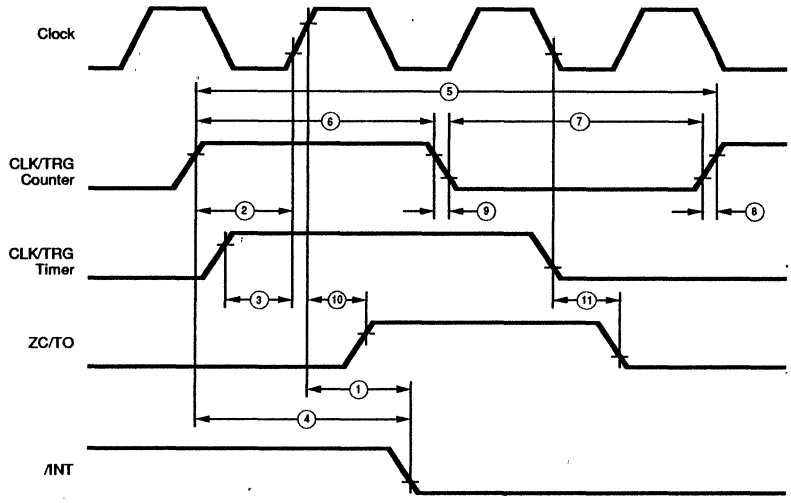


Figure 69. CTC Timing

Table B. CTC Timing Parameters

No	Symbol	Parameter	Z8018110 Min	Z8018110 Max	Z8018112 Min	Z8018112 Max	Unit	Note
1	TdCr(INTf)	Clock Rise to /INT Fall Delay		(TcC+100)		(TcC+80)	ns	[B1]
2	TsCTRr(Cr)c	CLK/TRG Rise to Clock Rise Setup Time for Immediate Count	90		60		ns	[B2]
3	TsCTR(Ct)	CLK/TRG Rise to Clock Rise Setup Time for Enabling of Prescalor On Following Clock Rise	90		60		ns	[B1]
4	TdCTRr(INTf)	CLK/TRG Rise to /INT Fall Delay						
		TsCTR(C) Satisfied		(1)+(3)		(1)+(3)	ns	[B2]
		TsCTR(C) Not Satisfied		TcC+(1)+(3)		TcC+(1)+(3)	ns	[B2]
5	TcCTR	CLK/TRG Cycle Time	(2TcC)	DC	(2TcC)	DC	ns	[B3]
6	TwCTRh	CLK/TRG Width (Low)	90	DC	90	DC	ns	
7	TwCTRl	CLK/TRG Width (High)	90	DC	90	DC	ns	
8	TrCTR	CLK/TRG Rise Time		30		30	ns	
9	TfCTR	CLK/TRG Fall Time		30		30	ns	
10	TdCr(ZCr)	Clock Rise to ZC/TO Rise Delay		80		80	ns	
11	TdCf(ZCf)	Clock Fall to ZC/TO Fall Delay		80		80	ns	

Notes for Table B:
[B1] Timer Mode
[B2] Counter Mode
[B3] Counter Mode Only; When using a cycle time less than 3TcC, parameter #2 must be met.

AC CHARACTERISTICS (Continued)
SCC Timing

Figure 70 shows the AC characteristics for the on-chip SCC. Parameters referred to in this figure appear in Table C.

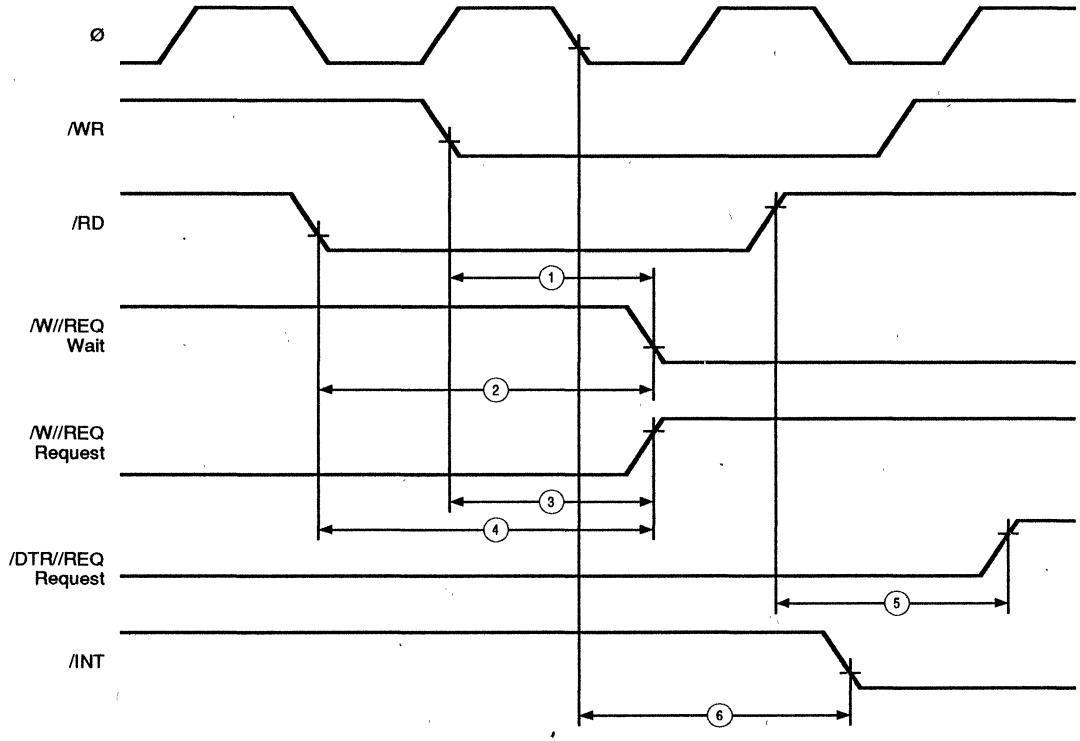


Figure 70. SCC AC Parameters

Table C. SCC Timing Parameters (85C30 AC Characteristics)

No	Symbol	Parameter	Z8018110 Min	Z8018110 Max	Z8018112 Min	Z8018112 Max	Unit	Note
1	TdWR(W)	/WR Fall to Wait Valid Delay		180 + TcC		125 + TcC	ns	[C1]
2	TdWR(W)	/RD Fall to Wait Valid Delay		180		125	ns	[C1]
3	TdWRf(REQ)	/WR Fall to /W//REQ Not Valid Delay		180 + TcC		125 +TcC	ns	
4	TdRDf(REQ)	/RD Fall to /W//REQ Not Valid Delay		180		125	ns	
5	TdWRr(REQ)	/WR Rise to /DTR//REQ Not Valid Delay		5TcC		5TcC	ns	
6	TdPC(INT)	Clock to /INT Valid Delay		500		500	ns	[C1]
7	TdRDA(INT)	/M1 Fall to /INT Inactive Delay		TBS		TBS	ns	[C1]

Note for Table C:
[C1] Open-drain output, measured with Open-drain test load.

AC CHARACTERISTICS (Continued)
SCC General Timing

Figure 71 shows the general timing for the on-chip SCC. Parameters referred to in this figure appear in Table D.

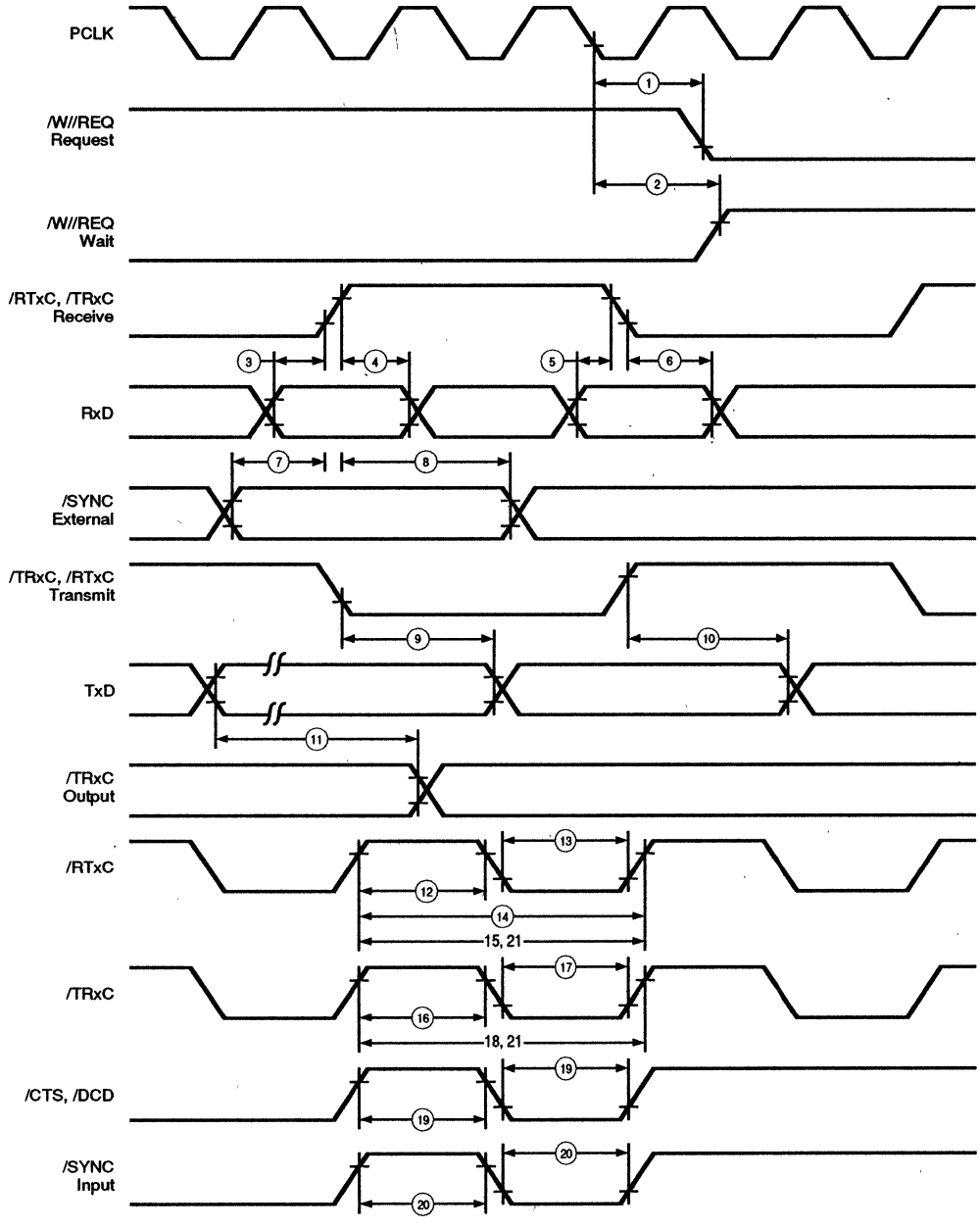


Figure 71. SCC General Timing

AC CHARACTERISTICS (Continued)
SCC General Timing

Table D. SCC General Timing Parameters

No	Symbol	Parameter	Z8018110		Z8018112		Unit	Note
			Min	Max	Min	Max		
1	TdPC(REQ)	Clock Fall to /W//REQ Valid		200		120	ns	
2	TdPC(W)	Clock Fall to Wait Inactive		300		220	ns	
3	TsRXD(RXCr)	RxD to /RxC Rise Setup Time	0		0		ns	[D1]
4	ThRXD(RXCr)	RxD to /RxC Rise Hold Time	125		100		ns	[D1]
5	TsRXD(RXCf)	RxD to /RxC Fall Setup Time	0		0		ns	[D1,4]
6	ThRXD(RXCf)	RxD to /RxC Fall Hold Time	125		100		ns	[D1,4]
7	TsSY(RXC)	/SYNC to /RxC Setup Time	-150		-125		ns	[D1]
8	ThSY(RXC)	/SYNC to /RxC Hold Time	5TcC		5TcC		ns	[D1]
9	TdTXCf(TXD)	/TxC Fall to TxD Delay		150		130	ns	[D2]
10	TdTXCr(TXD)	/TxC Rise to TxD Delay		150		130	ns	[D2,4]
11	TdTXD(TRX)	TxD to /TRxC Delay		140		120	ns	
12	TwRTXh	/RTxC High Width	120		100		ns	[D5]
13	TwRTXl	/RTxC Low Width	120		100		ns	[D5]
14	TcRTX	/RTxC Cycle Time (RxD, TxD)	400		320		ns	[D5,6]
15	TcRTXX	Xtal OSC Period	100	1000	80	1000	ns	[D3]
16	TwTRXh	/TRxC High Width	120		100		ns	[D5]
17	TwTRXl	/TRxC Low Width	120		100		ns	[D5]
18	TcTRX	/TRxC Cycle Time	400		320		ns	[D5,7]
19	TwEXT	/DCD or /CTS Pulse Width	120		100		ns	
20	TwSY	/SYNC Pulse Width	100		70		ns	
21	TxRx(DPLL)	DPLL Cycle Time	50		40		ns	[D6,7]

Notes to Table D:
[D1] /RXC is /RTxC or /TRxC, whichever is supplying the receiver clock.
[D2] /TXC is /TRxC or /RTxC, whichever is supplying the transmitter clock.
[D3] Both /RTxC and /SYNC pin has 30pf Capacitors (to ground).
[D4] Parameter applies only to FM encoding/decoding.
[D5] Parameter applies only to transmitter and receiver; baud rate generator timing requirements are different.
[D6] The maximum receive or transmit data rate is 1/4 TcC.
[D7] Applies to DPLL clock source only. Maximum data rate of 1/4 TcC still applies.

Figure 72 shows the system timing for the on-chip SCC. Parameters referred to in this figure appear in Table E.

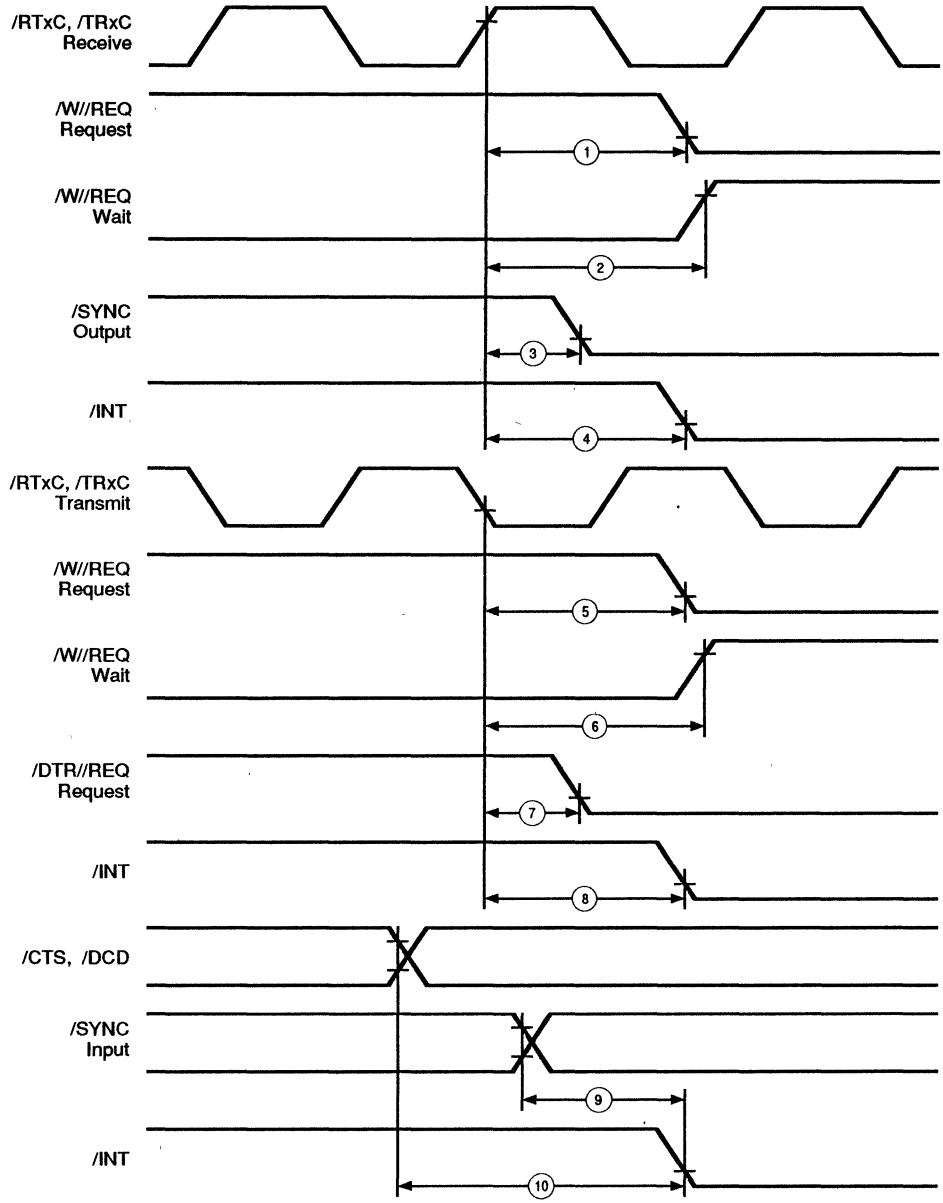


Figure 72. SCC System Timing

AC CHARACTERISTICS (Continued)
SCC System Timing

Table E. SCC System Timing Parameters

No	Symbol	Parameter	Z8018110		Z8018112		Unit	Note
			Min	Max	Min	Max		
1	TdRxC(REQ)	/RxC to /W//REQ Valid	8	12	8	12	TcC	[E2]
2	TdRxC(W)	/RxC to Wait inactive	8	14	8	14	TcC	[E1,2]
3	TdRxC(SY)	/RxC to /SYNC Valid	4	7	4	7	TcC	[E2]
4	TdRxC(INT)	/RxC to /INT Valid	10	16	10	16	TcC	[E1,2]
5	TdTxC(REQ)	/TxC to /W//REQ Valid	5	8	5	8	TcC	[E3]
6	TdTxC(W)	/TxC to Wait inactive	5	11	5	11	TcC	[E1,3]
7	TdRxC(DRQ)	/TxC to /DTR//REQ Valid	4	7	4	7	TcC	[E3]
8	TdTxC(INT)	/TxC to /INT Valid	6	10	6	10	TcC	[E1,3]
9	TdSY(INT)	/SYNC to /INT Valid	2	6	2	6	TcC	[E1]
10	TdEXT(INT)	/DCD or /CTS to /INT Valid	2	6	2	6	TcC	[E1]

Notes for Table E:
[E1] Open-drain output, measured with Open-drain test load.
[E2] /RXC is /RTxC or /TRxC, whichever is supplying the receiver clock.
[E3] /TXC is /TRxC or /RTxC, whichever is supplying the transmitter clock.

AC CHARACTERISTICS (Continued)
PIA General Purpose I/O Port Timing

Figure 73 shows the timing for the PIA ports. Parameters referred to in this figure appear in Table F.

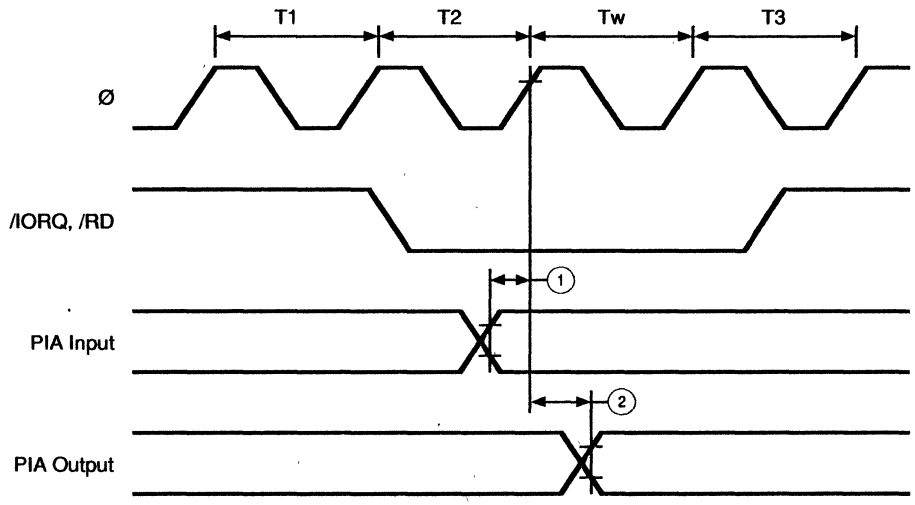


Figure 73. PIA Timing

Table F. PIA General Purpose I/O Timing Parameters

No	Symbol	Parameter	Z8018110		Z8018112		Unit	Note
			Min	Max	Min	Max		
1	TsPIA(C)	PIA Data Setup time to Clock Rise	10		10		ns	
2	TdCr(PIA)	Clock Rise to PIA Data Valid Delay		50		50	ns	

Interrupt Daisy-Chain Timing

Figure 74 shows the interrupt daisy-chain timing Parameters referred to in this figure appear in Table G.

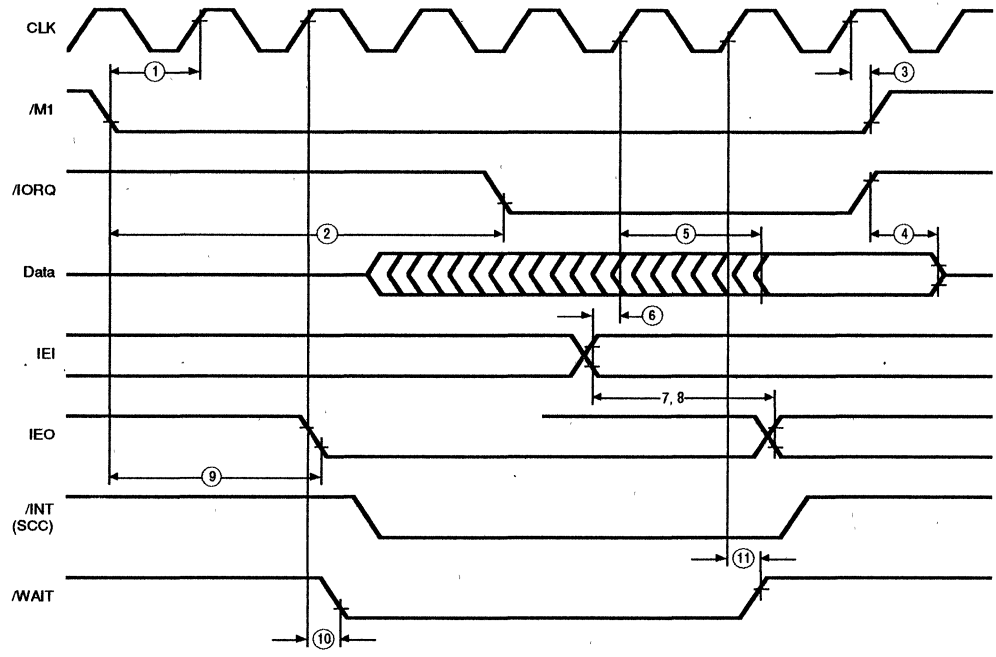


Figure 74. Interrupt Daisy-Chain Timing

Table G. Interrupt Daisy-Chain Timing Parameters

No	Symbol	Parameter	Z8018110		Z8018112	
			Min	Max	Min	Max
1	TsM1(Cr)	/M1 Fall to Clock Rise Setup Time	20		20	
2	TsM1(IO)INTA	/M1 Fall to /IORQ Fall Setup Time (During INTACK Cycle)	2TcC		2TcC	
3	Th	Hold Time	0		0	
4	TdM1r(DOz)	/M1 Rise to Data Out Float Delay	0		0	
5	TdCr(DO)	Clock Rise to Data Out Delay		120		100
6	TsIEI(TW4)	IEI to T_{W4} Rise Setup Time	95		80	
7	TdIEIf(IEOf)	IEI Fall to IEO Fall Delay		20		20
8	TdIEIr(IEOr)	IEO Rise to IEO Rise Delay		140		120
9	TdM1f(IEOf)	/M1 Fall to IEO Fall Delay		140		120
10	TdCWA(f)INTA	Clock Rise to /WAIT Fall Delay		30		25
11	TdCWA(r)INTA	Clock Rise to /WAIT Rise Delay		30		25

AC CHARACTERISTICS (Continued)
Read Write External BUS Master Timing

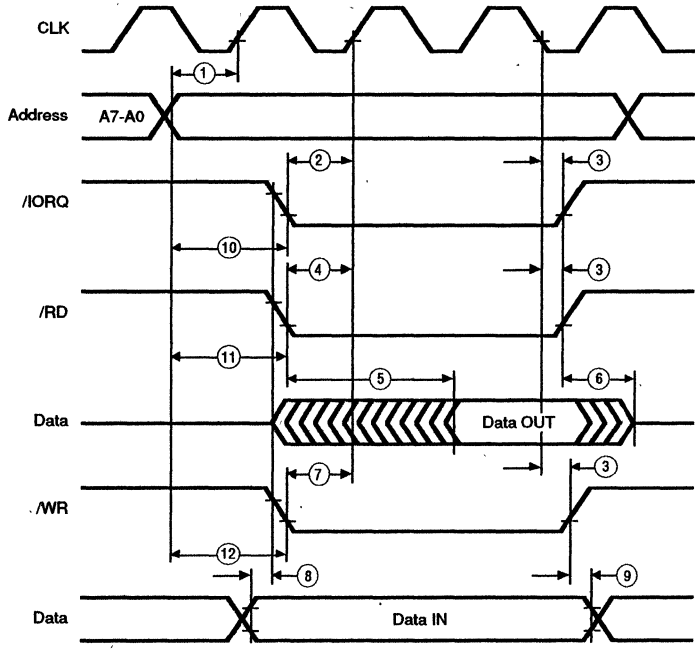


Figure 75. Read/Write External BUS Master Timing

Table H. External Bus Master Interface Timing (Read/Write Cycles)

No	Symbol	Parameter	Z8018110 Min	Z8018110 Max	Z8018112 Min	Z8018112 Max
1	TsA(Cr)	Address to CLK Rise Setup Time	20		20	
2	TsIO(Cr)	/IORQ Fall to CLK Rise Setup Time	20		20	
3	Th	Hold Time	0		0	
4	TsRD(Cr)	/RD Fall to CLK Rise Setup Time	20		20	
5	TdRD(DO)	/RD Fall to Data Out Delay		120		100
6	TdRIr(DOz)	/RD, /IORQ Rise to Read Data Float	0		0	
7	TsWR(Cr)	/WR Fall to CLK Rise Setup Time	20		20	
8	TsDi(WRf)	Data in to /WR Fall Setup Time	0		0	
9	ThWIr(Di)	/IORQ, /WR Rise to Data In Hold Time	0		0	
10	TsA(IORQf)	Address to /IORQ Fall Setup Time	50		40	
11	TsA(RDf)	Address to /RD Fall Setup Time	50		40	
12	TsA(WRf)	Address to /WR Fall Setup Time	50		40	

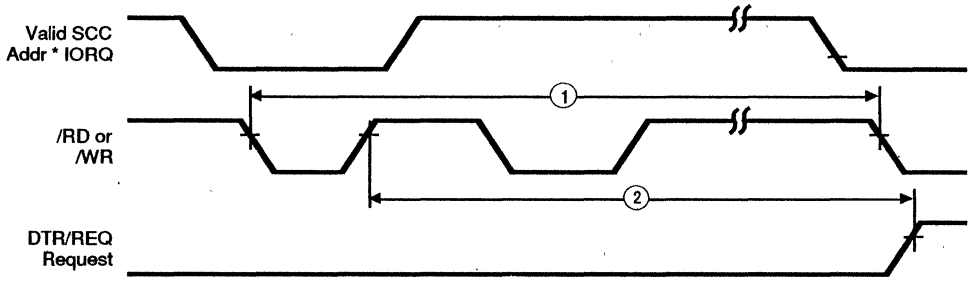


Figure 76. SCC External BUS Master Timing

Table I. External Bus Master Interface Timing (SCC Related Timing)

No	Symbol	Parameter	Z8018110 Min	Z8018110 Max	Z8018112 Min	Z8018112 Max	Unit	Note
1	TrC	Valid Access Recovery Time	4TcC		4TcC		nS	[1]
2	TdRDr(REQ)	/RD Rise to /DTR//REQ Not Valid Delay	4TcC		4TcC		nS	

[1] Applies only between transactions involving the SCC.

When using the interrupt daisy chained device(s) for other than the Z181, these are the following restrictions/notes (without external logic).

The device(s) has to be connected to the higher priority location (Figure 77).

The device(s) IEI-IEO delay has to be less than two clock cycles.

The Z181 on-chip interface logic inserts another three wait states into the interrupt acknowledge cycle to meet the on-chip SCC and the Z80 CTC timing requirements. (Total of 5 wait states; includes the automatic inserted two wait states).

To meet the timing requirements, the Z181's on-chip circuit generates interface signals for the SCC and CTC. Figure 78 has the timing during the interrupt acknowledge cycle, including the internally generated signals.

The following are three separate cases for the daisy-chain settle times:

Case 1 - SCC: The SCC /INTACK signal goes active on the T1 clock fall time. The settle time is from SCC /INTACK active until the SCC /RD signal goes active on the fourth rising wait state clock.

Case 2 - CTC: The settle time for the on-chip /IORQ is between the fall of /M1 until the internal CTC /IORQ goes active on the rise of the fourth wait state (the same time as SCC /RD goes active).

Case 3 - OFF-chip Z80 Peripheral: The settle time for the off-chip Z80 peripheral is from the fall of /M1 until CTC /IORQ goes active. Since the Z181's external /IORQ signal goes active on the clock fall of the first automatically inserted wait state (T_{WA}), the external daisy-chain device has to be connected to the upper chain location. Also, it must settle within two clock cycles.

If any peripheral is connected externally with a lower daisy chain priority than Z181 peripherals, /IORQ has to be delayed by external logic as shown in Figure 79

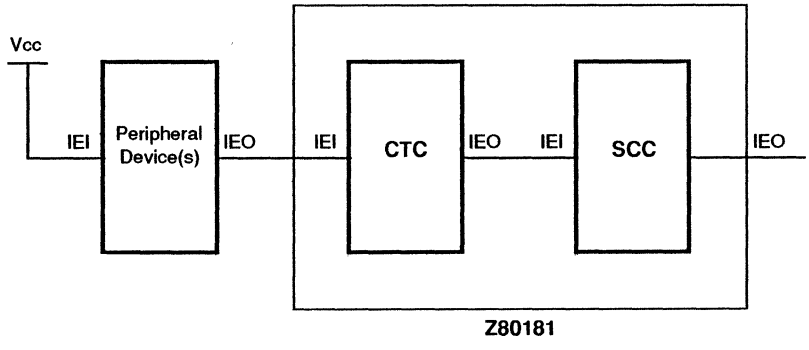


Figure 77. Peripheral Device as Part of the Daisy Chain

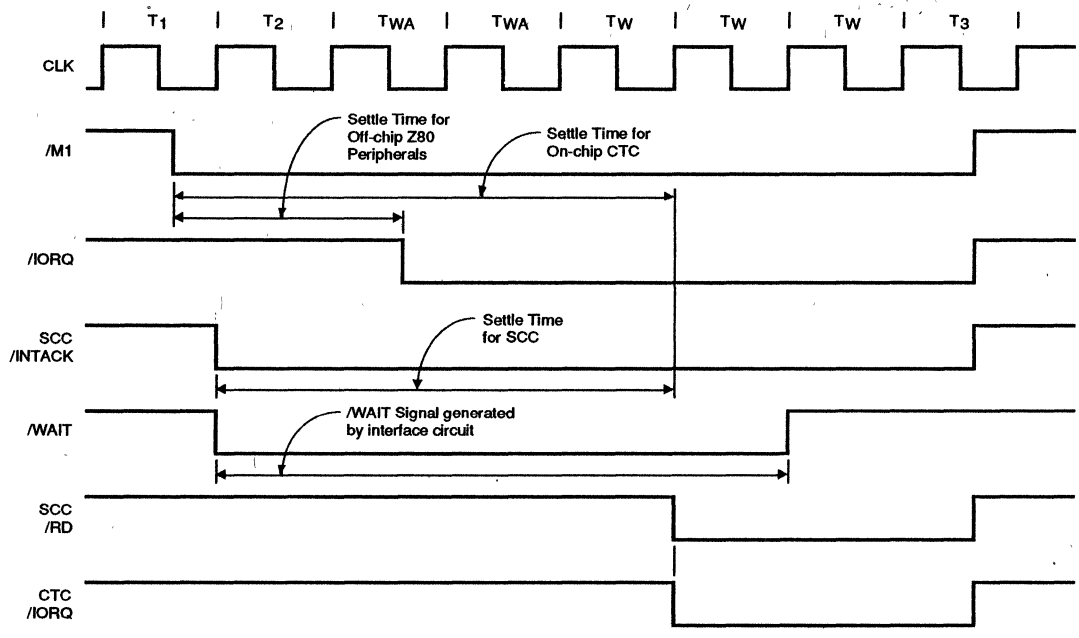


Figure 78. Interrupt Acknowledge Cycle Timing

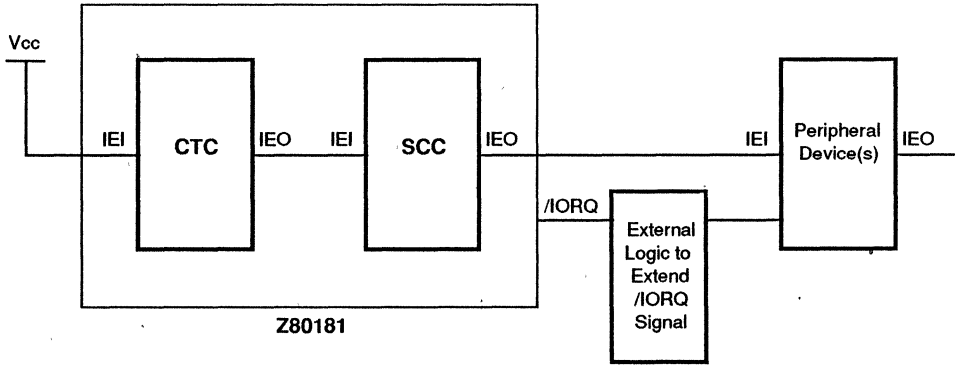


Figure 79. Peripheral Device as Part of the Daisy Chain

ZiLOG

PRELIMINARY PRODUCT SPECIFICATION

Z84013/015
Z84C13/Z84C15
IPC INTELLIGENT PERIPHERAL CONTROLLER

FEATURES

- Z84C00 Z80 CPU with Z84C30 CTC, Z84C4X SIO, CGC, Watch Dog Timer(WDT). In addition, Z84C15 and Z84015 have Z84C20 PIO.
- High speed operation (6/10 MHz).
- Low power consumption in four operation modes:
  (TBD) mA Typ. (Run mode)
  (TBD) mA Typ. (Idle1 mode)
  (TBD) mA Typ. (Idle2 mode)
  (TBD) µA Typ (Stop mode)
- Wide operational voltage range (5V ±10%).
- TTL/CMOS compatible.
- Z84013 features:
  Z84C00 Z80 CPU
  On-chip two channel SIO (Z80 SIO).
  On-chip four channel Counter Timer Controller (Z80 CTC).
  Built-in Clock Generator Controller (CGC).
  Built-in Watch Dog Timer (WDT).
  Noise filter to CLK/TRG inputs of the CTC.
  84-pin PLCC package.
- Z84015 features:
  All Z84013 features, plus on-chip two 8-bit ports (Z80 PIO) and 100-pin QFP package.
- Z84C13/Z84C15 enhancements to Z84013/Z84015:
  Power-on reset.
  Addition of two chip select pins.
  32-bit CRC for Channel A of SIO.
  Wait state generator
  Simplified EV mode selection.
  Schmitt-trigger inputs to transmit and receive clocks of the SIO.
  Crystal divide-by-one-mode.

GENERAL DESCRIPTION

The Intelligent Peripheral Controller(IPC) is a series of highly superintegrated devices with four versions. The Z84C13 and the Z84C15 are upward compatible versions of the Z84013 and the Z84015. The Z84015 is a CMOS 8-bit microprocessor integrated with the CTC, SIO, CGC, WDT and the PIO into a single 100-pin Quad Flat Pack(QFP) package. The Z84013 is the Z84015 without PIO, and is housed in a 84-pin PLCC package. The Z84C13 is the Z84013 with enhancements and the Z84C15 is the Z84015 with enhancements. These high-end superintegrated intelligent peripheral controllers are targeted for a broad range of applications ranging from error correcting modems to enhancement/cost reductions of existing hardware using Z80-based discrete peripherals. Figures 1 and 2 show the difference between the Z84013/015 and the Z84C13/Z84C15.

Hereinafter, use the word IPC on the description covering all versions (Z84C13/Z84C15 and Z84013/Z84015). Use Z84C13/C15 on the description that applies only to the Z84C13 and Z84C15, and use Z84013/015 on the description that applies only to the Z84013 and Z84015.

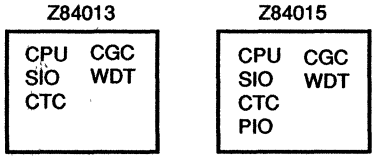


Figure 1. Z84013/015 Version

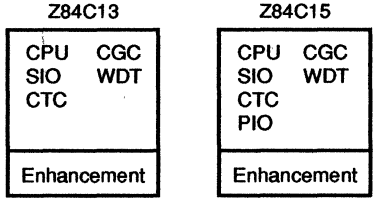
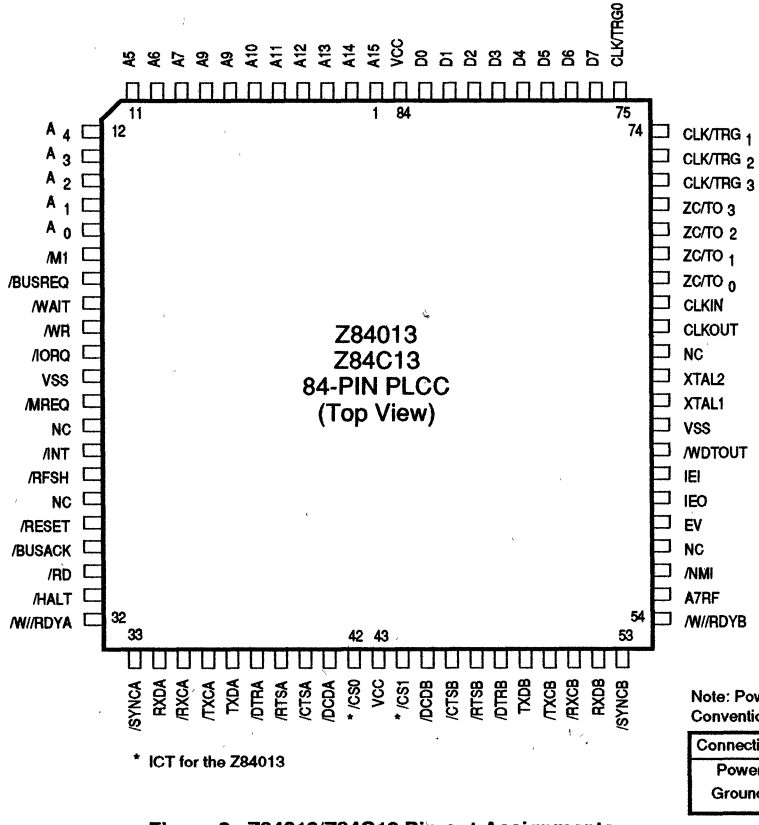


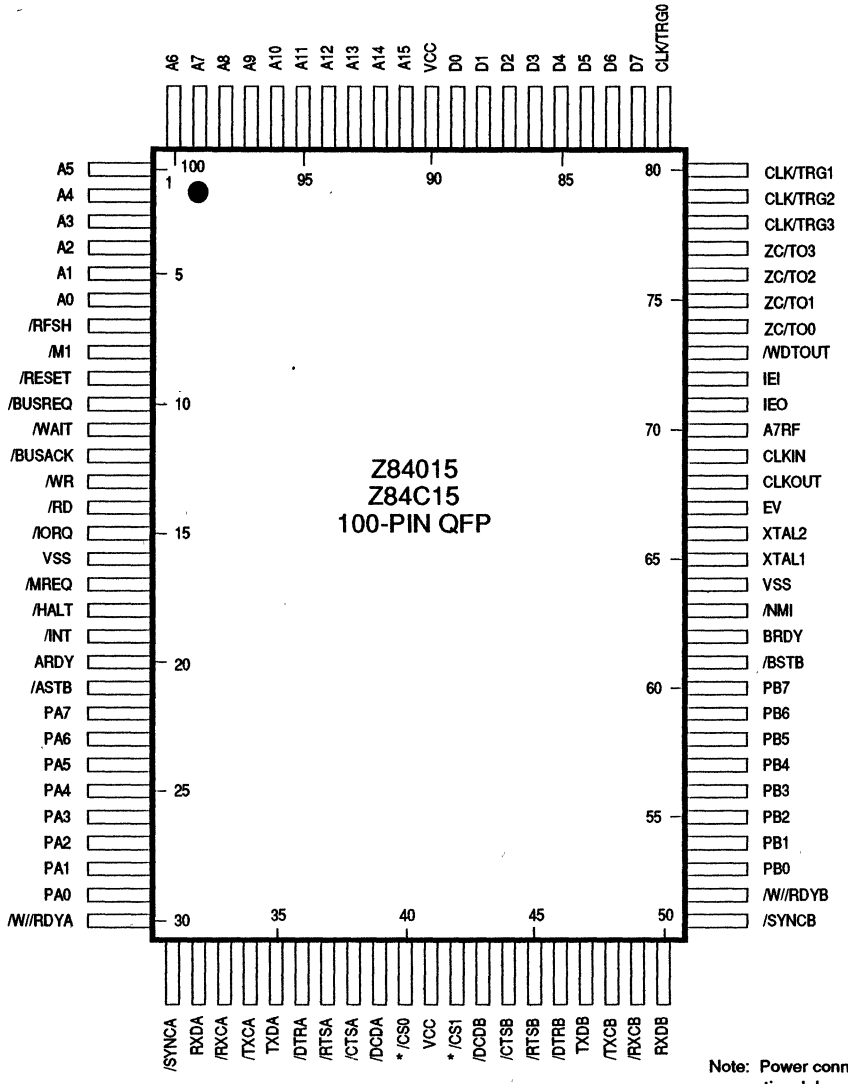
Figure 2. Z84C13/C15 Version



Note: Power connections follow Conventional descriptions below

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

Figure 3. Z84013/Z84C13 Pin-out Assignments



* ICT for the Z84015

Note: Power connections follow conventional descriptions below:

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

Figure 4. Z84015/Z84C15 Pin-out Assignments

PIN DEFINITIONS

The pin assignment for each device is shown in Figures 3 and 4. Following is the description on each pin. For the description and the pin number, if stated as "x13" or "x15", that applies to both Z84C13/Z84013 or Z84C15/Z84015. Otherwise, C13 for Z84C13, C15 for Z84C15, 013 for Z84013 and 015 for Z84015.

CPU SIGNALS

Pin Name	Pin Number	Input/Output, 3-State	Function
A0-A15	1-16(x13), 1-6, 91-100(x15)	I/O	16-bit address bus. Specifies I/O and memory addresses to be accessed. During the refresh period, addresses for refreshing are output. The bus is an input when the external master is accessing the on-chip peripherals.
D0-D7	76-83(x13), 82-89(x15)	I/O	8-bit bidirectional data bus. When the on-chip CPU is accessing on-chip peripherals, these lines are set to output and hold the data to/from on-chip peripherals.
/RD	30(x13), 14(x15)	I/O	Read signal. CPU read signal for accepting data from memory or I/O devices. When an external master is accessing the on-chip peripherals, it is an input signal.
/WR	20(x13), 13(x15)	I/O	Write Signal. This signal is output when data, to be stored in a specified memory or peripheral LSI, is on the MPU data bus. When an external master is accessing the on-chip peripherals, it is an input signal.
/MREQ	23(x13), 17(x15)	I/O, 3-State	Memory request signal. When an effective address for memory access is on the address bus, "0" is output. When an external master is accessing the on-chip peripherals, it is an tri-state signal.
/IORQ	21(x13), 15(x15)	I/O	I/O request signal. When addresses for I/O are on the lower 8 bits (A7-A0) of the address bus in the I/O operation, "0" is output. In addition, the /IORQ signal is output with the /M1 signal at the time of interrupt acknowledge cycle to inform peripheral LSI of the state of the interrupt response vector is when put on the data bus. When an external master is accessing the on-chip peripherals, it is an input signal.
/M1	17(x13), 8(x15)	I/O	Machine cycle "1". /MREQ and "0" are output together in the operation code fetch cycle. /M1 is output for every opcode fetch when a two byte opcode is executed. In the maskable interrupt acknowledge cycle, this signal is output together with /IORQ. It is 3-stated in EV mode.

CPU SIGNALS (Continued)

Pin Name	Pin Number	Input/Output, 3-State	Function
/RFSH	26(x13), 7(x15)	Out, 3-State	The refresh signal. When the dynamic memory refresh address is on the low order byte of the address bus, /RFSH is active along with /MREQ signal. This pin is 3-stated in EV mode.
/INT	25(x13), 19(x15)	Open drain	Maskable interrupt request signal. Interrupt is generated by peripheral LSI This signal is accepted if the interrupt enable Flip-Flop (IFF) is set to "1". The /INT signal of on-chip peripherals is internally wired - OR without pull-up resistors and requires external pull-up. Also, interrupts from on-chip peripherals go out from this pin.
/NMI	56(x13), 63(x15)	In	Non-maskable interrupt request signal This interrupt request has a higher priority than the maskable interrupt request and does not rely upon the state of the interrupt enable Flip-Flop (IFF).
/HALT	31(x13), 81(x15)	Out, 3-State	Halt signal. Indicates that the CPU has executed a HALT instruction. This signal is 3-stated in EV mode.
/BUSREQ	18(x13), 10(x15)	In	BUS request signal. /BUSREQ requests placement of the address bus, data bus, /MREQ, /IORQ, /RD and /WR signals into the high impedance state. /BUSREQ is normally wired-OR and a pull-up resistor is externally connected.
/BUSACK	29(x13), 12(x15)	Out (013/015), Out/3-State (C13/C15)	Bus Acknowledge signal. In response to /BUSREQ signal, /BUSACK informs a peripheral LSI that the address bus, data bus, /MREQ, /IORQ, /RD and /WR signals have been placed in the high impedance state

Note: For the Z84013/015 the /BUSACK signal will not be 3-stated during EV mode. For the Z84C13/C15 the /BUSACK will be 3-stated during EV mode.

/WAIT	19(x13), 11(x15)	In(013/015), I/O(C13/C15)	Wait signal. /WAIT informs the CPU that specified memory or peripheral is not ready for data transfer. As long as /WAIT signal is active, MPU is continuously kept in the wait state.

Note: For the Z84C13/C15, the /WAIT pin becomes an output to bring out on-chip wait state generator during the EV mode.

CPU SIGNALS (Continued)

Pin Name	Pin Number	Input/Output, 3-State	Function
A7RF	55(x13), 70(x15)	Out	1-bit auxiliary address bus. Output is the same as bit-7 (A7) of the address bus. However, during a refresh cycle, this pin outputs the address which is the most significant bit of the 8-bit refresh address signal linked to the low order 7 bits of the address bus.

CTC SIGNALS

Pin Name	Pin Number	Input/Output, 3-State	Function
CLK/TRG0 - CLK/TRG3	72-75(x13), 78-81(x15)	In	External clock/trigger input. These four CLK/TRG pins correspond to four Counter/Timer Channels. In the counter mode, each active edge will cause the downcounter to decrement by one. In timer mode, an active edge will start the timer. It is program selectable whether the active edge is rising or falling.
ZC/TO0 - ZC/TO3	68-71(x13), 74-77(x15)	Out	Zero count/timer out signal. In either timer or counter mode, pulses are output when the down-counter has reached zero.

SIO SIGNALS

Pin Name	Pin Number	Input/Output, 3-State	Function
/W//RDYA, /W//RDYB	32,54(x13), 30,52(x15)	Out	Wait/Ready signal A and Wait/Ready signal B. Used as /WAIT or /READY depending upon SIO programming. When programmed as /WAIT they go active at "0", alerting the CPU that addressed memory or I/O devices are not ready by requesting the CPU to wait. When programmed as /READY, they are active at "0" which determines when a peripheral device associated with a DMA port is for read/write data.
/SYNCA, /SYNCB	33,53(x13), 31,51(x15)	I/O	Synchronous signals.In asynchronous receive mode, they act as /CTS and /CDC In external sync mode, these signals act as inputs. In internal sync mode, they act as outputs.
RxDA, RxDB	34,52(x13), 32,50(x15)	In	Serial receive data signal.

SIO SIGNALS (Continued)

Pin Name	Pin Number	Input/Output, 3-State	Function
/RxCA, /RxCB	35,51(x13), 33,49(x15)	In	Receive clock signal. In the asynchronous mode, the receive clocks can be 1, 16, 32, or 64 times the data transfer rate.
/TxCA, /TxCB	36,50(x13), 34,48(x15)	In	Transmitter clock signal. In the asynchronous mode, the transmitter clocks can be 1, 16, 32, or 64 times the data transfer rate.
TxDA, TxDB	37,49(x13), 35,47(x15)	Out	Serial transmit data signal.
/DTRA, /DTRB	38,48(x13), 36,46(x15)	Out	Data terminal ready signal. When ready, these signals go active to enable the terminal transmitter. When not ready they go inactive to disable the transfer from the terminal.
/RTSA, /RTSB	39,47(x13), 37,45(x15)	Out	Request to send signal. "0" when transmitting serial data. They are active when enabling their receivers to transmit data.
/CTSA, /CTSB	40,46(x13), 38,44(x15)	In	Clear to send signal. When "0", after transmitting these signals the modem is ready to receive serial data. When ready, these signals go active to enable terminal transmitter. When not ready, these signals go inactive to disable transfer from the terminal.
/DCDA, /DCDB	41,45(x13), 39,43(x15)	In	Data carrier detect signal. When "0", serial data can be received. These signals are active to enable receivers to transmit.

SYSTEM CONTROL SIGNALS

Pin Name	Pin Number	Input/Output, 3-State	Function
IEI	60(x13), 72(x15)	In	Interrupt enable input signal. IEI is used with the IEO to form a priority daisy chain when there is more than one interrupt-driven peripheral.
IEO	59(x13), 71(x15)	Out	The interrupt enable output signal. In the daisy chain interrupt control, IEO controls the interrupt of external peripherals. IEO is active when IEI is "1" and the CPU is not servicing an interrupt from the on-chip peripherals.
/CS0 (C13/C15 only)	42(C13), 40(C15)	Out	Chip Select 0. Used to access external memory or I/O devices. This pin has been assigned to "ICT" pin on Z84013/015. This signal is decoded only from A15-A12 without control signals Refer to "Functional Description" on-chip select signals for further explanation.

SYSTEM CONTROL SIGNALS (Continued)

Pin Name	Pin Number	Input/Output, 3-State	Function
/CS1 (C13/C15 only)	40(x13), 42(x15)	Out	Chip Select 1. Used to access external memory or I/O devices. This pin has been assigned to "ICT" pin on Z84013/015. This signal is decoded only from A15-A12 without control signals. Refer to "Functional Description" on-chip select signals for further explanation.
/WDTOUT	61(x13), 73(x15)	Out(013/015), Open Drain(C13/C15)	Watch Dog Timer Output signal. Output pulse width depends on the externally connected pin.
/RESET	28(x13), 9(x15)	Input(013/015), I/O (Open Drain) (C13/C15)	Reset signal. /RESET signal is used for initializing MPU and other devices in the system. Also used to return from the steady state in the STOP or IDLE modes.

Note: For the Z84013/Z84015 the /RESET must be kept in active state for a period of at least three system clock cycles.

Note: For the Z84C13/Z84C15, during the power-up sequence, the /RESET becomes an Open drain output and the Z84C13/C15 will drive this pin to "0" for 25 to 75 msec after the power supply passes through approx. 2.2V and then reverts to input. If it receives the /RESET signal after power-on sequence, it will drive /RESET pin for 16-processor clock cycles depending on the status of Reset Output Disable bit in Misc Control Register If this Reset output is disabled, it must be kept in active state for a period of at least three system clock cycles Note, that if using Z84C13/C15 in a Z84013/015 socket, modification may be required on the reset circuit since this pin is "pure input pin" on the Z84013/015 Also, the /RESET pin doesn't have internal pull-up resistors and therefore requires external pull-ups For more details on the device, please refer to "Functional Description."

XTAL1	63(x13), 65(x15)	In	Crystal oscillator connecting terminal. A parallel resonant crystal is recommended If external clock source is used as an input to the CGC unit, supply clock goes into this terminal. If external clock is supply to CLKIN pin (without CGC unit), this terminal must be connected to "0" or "1".
XTAL2	63(x13), 66(x15)	Out	Crystal oscillator connecting terminal.
CLKIN	67(x13), 69(x15)	In	Single-phase System Clock Input.
CLKOUT	66(x13), 68(x15)	Out	Single-phase clock output from on-chip Clock Generator/Controller.
EV	58(x13), 67(x15)	In	Evaluator signal. When "1" is applied to this pin, IPC is put in Evaluation mode

Note: For the Z84013/015, together with /BUSREQ, the EV signal puts the IPC into the evaluation mode When this signal becomes active, the status of /M1, /HALT and /RFSH change to input. When using Z84013/015 as an evaluator chip, the CPU is electrically disconnected after one machine cycle is executed with the EV signal "1" and the /BUSREQ signal "0". It follows the instructions from the other CPU (of ICE). Upon receiving /BUSREQ; A15-A0, /MREQ, /IORQ, /RD and /WR are changed to input and D7-D0 changes its direction. /BUSACK is NOT 3-stated so it should be disconnected by an externally connected circuit. For details, please refer to "Functional Description" on EV mode.

SYSTEM CONTROL SIGNALS (Continued)

Note: For the Z84C13/C15, to access on-chip resources from the CPU (e.g., ICE CPU), the CPU is electrically disconnected; A15-A0, /MREQ, /IORQ, /RD and /WR are changed to input; D7-D0 changes its direction; /M1, /HALT and /RFSH are put into the high impedance state when the EV pin is set to "1". Also, /BUSACK is 3-stated. For details, please refer to "Functional Description" on EV mode.

Pin Name	Pin Number	Input/Output, 3-State	Function
ICT	42,44(013), 40,42(015), Not with C13/C15	Out	Test pins. Used in the open state.
NC	24,27,57,65(x13), Not with x15		Not connected.
VCC	43,84(x13), 41,90(x15)	Power Supply	+5 Volts
VSS	22, 62(x13), 16,64(x15)	Power Supply	0 Volts

PIO SIGNALS (for the Z84x15 only)

Pin Name	Pin Number	Input/Output, 3-State	Function
/ASTB	21(x15)	In	Port A strobe pulse from a peripheral device. The signal is used as the handshake between Port A and external circuits. The meaning of this signal depends on the mode of operation selected for Port A (see "PIO Basic Timing").
/BSTB	61(x15)	In	Port B strobe pulse from a peripheral device. This signal is used as the handshake between Port B and external circuits. The meaning of this signal is the same as /ASTB, except when Port A is in mode 2 (see "PIO Basic Timing").
ARDY	20(x15)	Out	Register A ready signal Used as the handshake between Port A and external circuits. The meaning of this signal depends on the mode of operation selected for Port A (see "PIO Basic Timing").
BRDY	62(x15)	Out	Register B ready signal Used as the handshake between Port B and external circuits. The meaning of this signal is the same as ARDY except when Port A is in mode 2 (see "PIO Basic Timing")
PA7-PA0	22-29(x15)	I/O, 3-State	Port A data signals. Used for data transfer between Port A and external circuits.
PB7-PB0	53-60(x15)	I/O, 3-State	Port B data signals. Used for transfer between Port B and external circuits.

The following pins have different functions between 013/015 and C13/C15

Pin Name	Pin # X13	Pin # X15	Function
/RESET	28	9	Functionality is different.
/WAIT	19	15	Functionality is different.
EV	58	67	Functionality is different.
/WDTOUT	61	73	Push-pull output on Z84013/015, Open drain on Z84 C13/C15
ICT	40, 42	42, 40	(Test pin) on Z84013/015; /CS0 and /CS1 on Z84C13/15.
TxCA, TxCB, RxCA and RxCB	35, 36, 50, 51	33, 34, 48, 49	On Z84C13/15; these signals have Schmitt-triggered inputs.
/BUSACK	29	12	In EV mode, 3-stated on Z84C13/15; remains active on Z84013/015.

FUNCTIONAL DESCRIPTION

Figure 5(a) shows the functional block diagram of the Z84013/015 and Figure 5(b) shows the functional block diagram of the Z84C13/C15. As described earlier, the only difference between the Z84x13 and the Z84x15 is the PIO not being available on the Z84x13.

Functionally, the on-chip SIO, PIO (not available on Z84x13), CTC, and the Z80 CPU are the same as the discrete devices. Therefore, for detailed description of each individual unit, refer to the Product Specification/Technical Manual of each discrete product.

The following subsections describe each individual functional unit of the IPC.

Z84C00/01 Logic Unit
The CPU provides all the capabilities and pins of the Zilog Z80 CPU. This allows 100% software compatibility with existing Z80 software. In addition, it has the pin called "A7RF" to extend DRAM refresh address to 8-bits. Refer to "Z84C01 Z80 CPU with CGC" Product Specification.

Z84C20 Parallel Input/Output Logic Unit (Z84x15 Only)
This logic unit provides both TTL- and CMOS- compatible interfaces between peripheral devices and a CPU through the use of two 8-bit parallel ports (Figure 6). The CPU configures the logic to interface to a wide range of peripheral devices with no external logic. Typical devices that are compatible with this interface are keyboards, printers, and EPROM/PAL programmers.

The parallel ports (designated Port A and Port B) are byte wide and completely compatible with the Z84C20 PIO. These two ports have several modes of operation; input, output, bi-directional, or bit control mode. Each port has two handshake signals (RDY and /STB) which are used to control data transfers. The RDY (ready) indicates that the port is ready for data transfer while /STB (strobe) is an input to the port that indicates when data transfer has occurred. Each of the ports can be programmed to interrupt the CPU upon the occurrence of specified status conditions, and generate unique interrupt vectors when the CPU responds (for more information on the operation of this portion of the logic, please refer to the Z84C20 PIO Product Specification and Technical Manual).

Z84C30 Counter/Timer Logic Unit
This logic unit provides the user with four individual 8-bit Counter/Timer Channels that are compatible with the Z84C30 CTC (Figure 7). The Counter/Timers can be programmed by the CPU for a broad range of counting and timing applications. Typical applications include event counting, interrupt and interval counting, and serial baud rate clock generation.

Each of the Counter/Timer Channels, designated Channels 0-3, have an 8-bit prescaler (when used in timer mode) and its own 8-bit counter to provide a wide range of count resolution. Each of the channels have their own Clock/Trigger input to quantify the counting process and an output to indicate zero crossing/timeout conditions. With only one interrupt vector programmed into the logic unit, each channel can generate a unique interrupt vector in response to the interrupt acknowledge cycle.

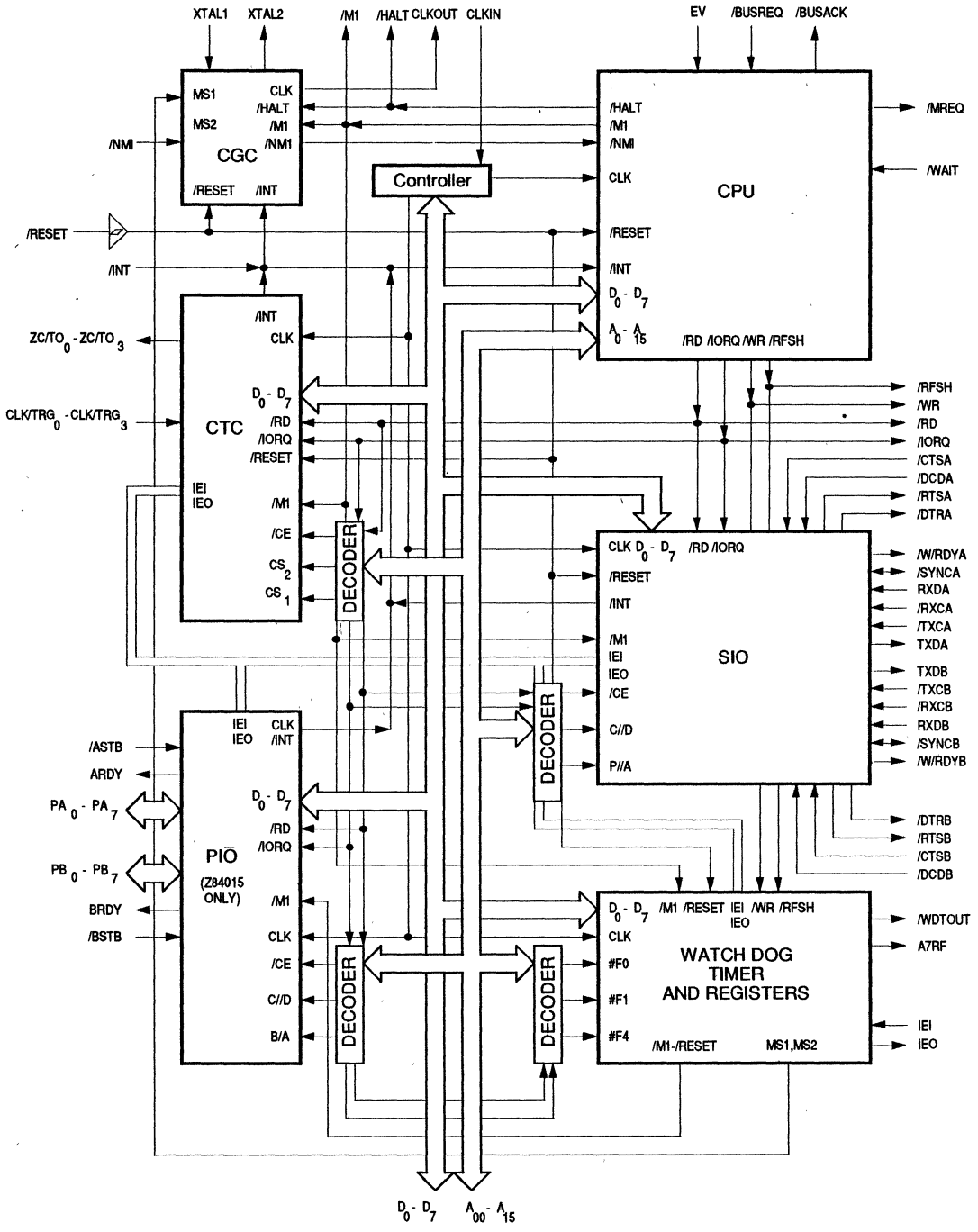


Figure 5(a). Block Diagram for 84013/015 IPC

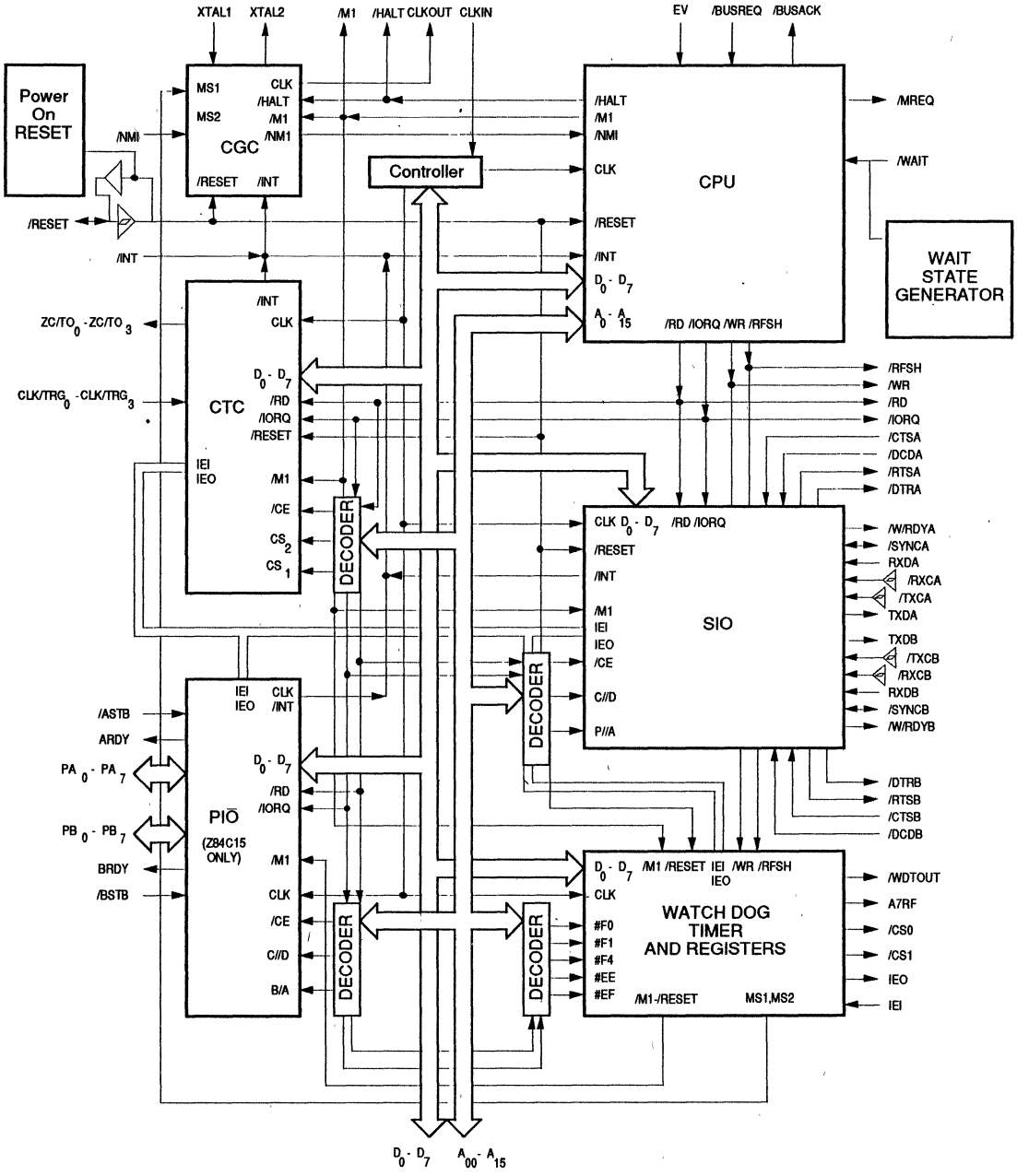


Figure 5(b). Block Diagram for 84C13/C15 IPC

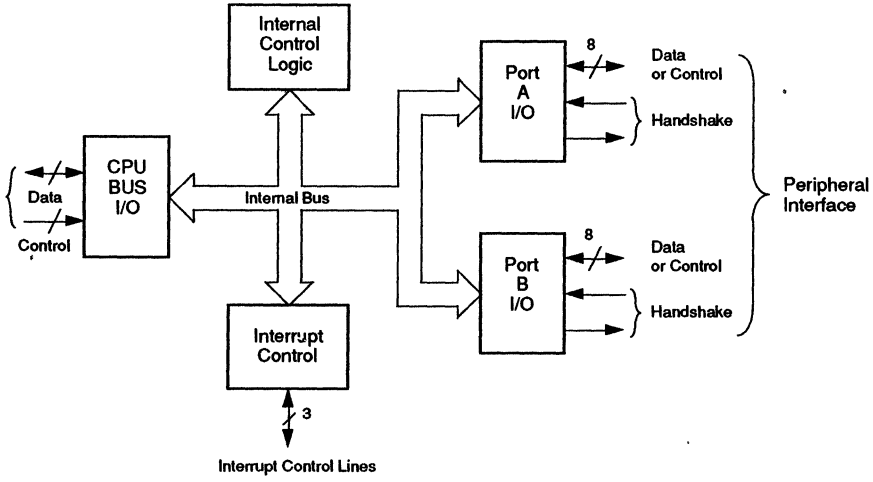


Figure 6. PIO Block Diagram

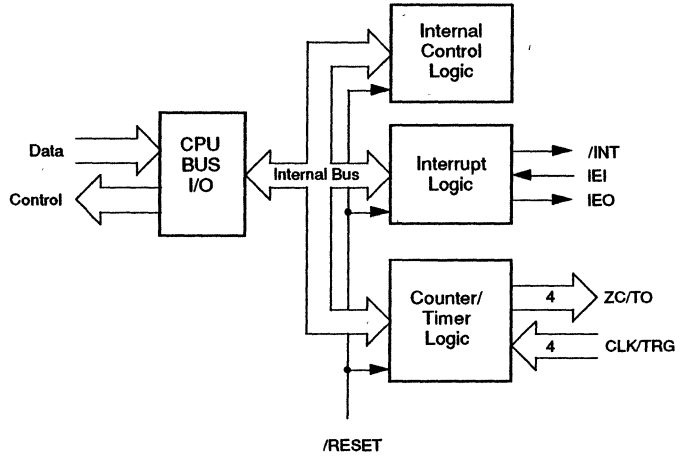


Figure 7. CTC Block Diagram

Z84C4x Serial I/O Logic Unit

This logic unit provides the user with two separate multiprotocol serial I/O channels that are completely compatible with the Z84C4x SIO. Their basic functions as serial-to-parallel and parallel-to-serial converters can be programmed by a CPU for a broad range of serial communications applications. Each channel, designated Channel A and Channel B, is capable of supporting all common asynchronous and synchronous protocols (Monosync, Bisync, and SDLC/HDLC, byte or bit oriented - Figure 8).

Z84C13/C15 Only. As an enhancement to the Z84013/015, the Z84C13/C15 can handle a 32-bit CRC on Channel A and Schmitt-trigger inputs on the /TxC and /RxC pins of both channels.

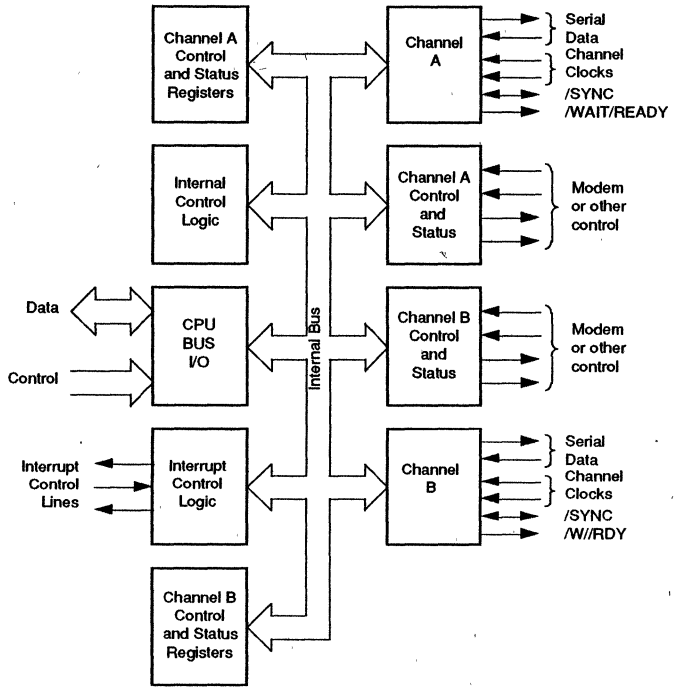


Figure 8. SIO Block Diagram

Watch Dog Timer (WDT) Logic Unit
This logic unit has been superintegrated into the IPC. It detects an operation error, caused by the program runaway, and returns to normal operation. Figure 9, shows the block diagram of the WDT. Upon Power-On Reset, this unit is enabled If WDT is not required, but /WDTOUT is connected to /RESET or any other circuit, it has to be disabled. During the power-down mode of operation (either IDLE1/2 or Stop), the Watch Dog Timer is halted.

WDT Output (/WDTOUT pin). When the WDT is used, the "0" level signal is output from the /WDTOUT pin after a duration of time specified in the WDTP or in the WDTMR. The output pulse width is one of the following, depending on the /WDTOUT pin connection.

- The /WDTOUT is connected to the /RESET pin: The "0" level is pulsed for 5TcC (System clock cycles).
- The /WDTOUT is connected to a pin other than the /RESET pin: The "0" level is kept until the Watch Dog timer is cleared by software, or reset by /RESET pin.

CGC Logic Unit. The IPC has CGC (Clock Generator/Controller) unit. This unit is identical to the one with the Z84C01 and the Z84C50, and supports power-down modes of operation. The output from this unit is on the pin called CLKOUT, and is not connected to the system clock internally. The CLKIN pin is the system clock input. The user can connect CLKOUT to CLKIN to utilize this CGC unit, or supply external clock from CLKIN pin.

The CGC unit allows crystal input (XTAL1, XTAL2) or External Clock input on the XTAL1 pin. It has clock divide-by-two circuits and generates a half-speed clock to the input.

Z84C13/C15. The power-down modes of the IPC vary depending upon whether the system clock is fed from the CGC unit (tie CLKOUT to CLKIN) or the external clock source on the CLKIN pin. They also have divide-by-one Mode. If the clock is supplied by this CGC unit, all of the modes in "halt" state are available. When external clock is provided on the CLKIN pin, XTAL1 is not left open (tied to "0" or "1") to avoid meta-stable conditions to minimize power consumption.

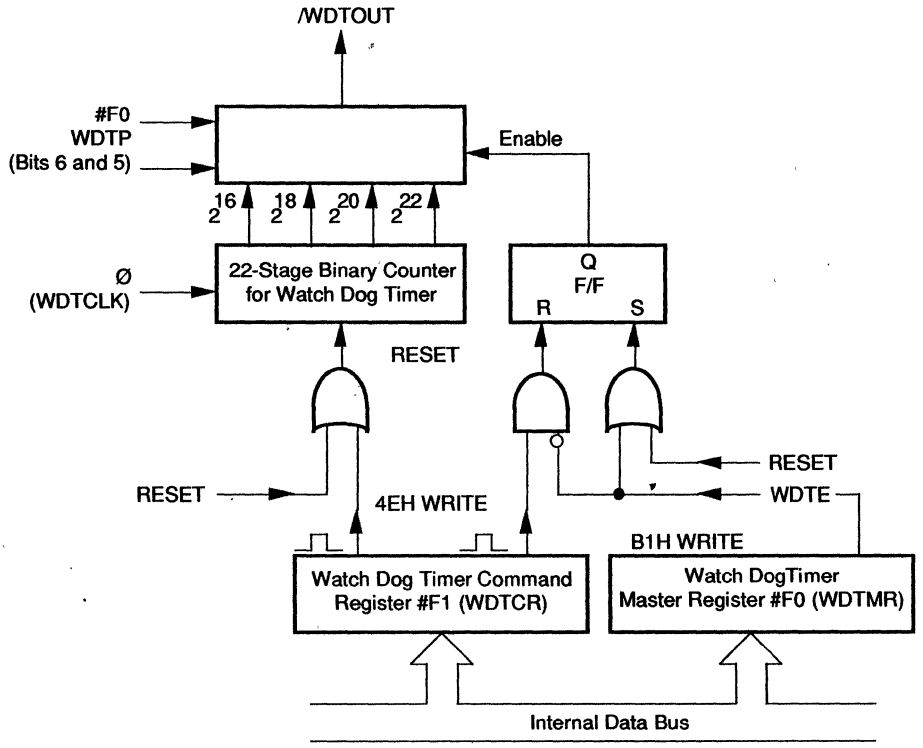


Figure 9. Block Diagram of Watch Dog Timer

Z84013/015 Only. If the system clock is provided on the CLKIN pin, none of the power-down mode (except RUN mode) is supported.

Z84C13/C15 Only. If the system clock is provided on the CLKIN pin, only the IDLE2 mode is applicable. In this mode, if the HALT instruction is executed, internal clock to the CTC is kept on "Continue", but the clock to the other components (CPU, PIO, SIO and Watch Dog Timer) are stopped. The divide-by-two circuit of the CGC unit can be skipped by programming bit D4 of the WDTMR (see "Programming" section). Upon Power-on Reset, it comes up in divide by two mode.

System Clock Generation

The IPC has a built-in oscillator circuit and the required clock can be easily generated by connecting a crystal to the external terminals (XTAL1, XTAL2). Clock output is the same frequency as half the speed of the crystal frequency. Example of oscillator connections are shown in Figure 10.

Z84C13/C15. Clock output is the same, or half, of the external frequency.

Z84X13/X15

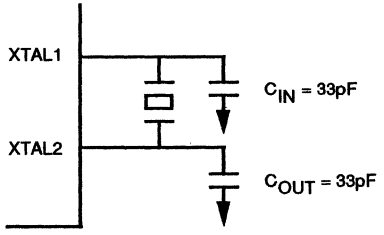


Figure 10. Circuit Configuration For Crystal

Recommended characteristics of the crystal and the values for the capacitor are as follows (the values will change with crystal frequency).

- Type of crystal: Fundamental, parallel type crystal (AT cut is recommended).
- Frequency tolerance: Application dependent.
- CL, Load capacitance: Approximately 22pf (acceptable range is 20-30pf).
- Rs, equivalent-series resistance: ≤ 150 ohms.
- Drive level: 10mW (for ≤ 10MHz crystal); 5mW (for ≥ 10MHz crystal).
- $C_{IN} = C_{OUT} = 33pF$.

Power-On Reset Logic Unit (Z84C13/C15 Only)
The Z84C13/C15 has the enhanced feature of a Power-on Reset Circuit. During the power-up sequence, the open-drain gate of the on-chip power-on Reset circuit drives /RESET pin to "0" for 25 to 75 msec after the power supply passes through approx. 2.2V After the termination of the "Power-on Reset" cycle, the open-drain gate of the on-chip Power-on Reset circuit stops to drive the /RESET pin. It is required to have external pull-up register on the /RESET pin.

If it receives /RESET input from outside after the power-on sequence and while the Reset Output Disable bit in Misc Control Register is cleared to "0", it will drive the /RESET pin for 16-processor clock cycles from the falling edge of the external /RESET input. Otherwise, the /RESET pin must be kept in the active state for a period of at least 3 system clock cycles.

If there are power-on reset circuits outside of this device, drive this pin with OPEN-DRAIN type gates with pull-up resisters because /RESET signal is driven low for the period mentioned above during the Power-on sequence. If the external Power-on Reset circuit has push-pull type drivers and they drive the /RESET pin to "1" during that period, it may cause damage. In particular, when using Z84C13/C15 in the Z84013/015 socket, modification may be required on the external reset circuit.

Wait State Generator Unit (Z84C13/C15 Only)
The Z84C13/C15 has the enhanced feature of a Wait State Generator circuit. It is capable of generating /WAIT signals to the CPU internally. The status of the External /WAIT input line is sampled after the insertion of software wait states, except for the wait state's insertion of Interrupt Daisy Chain Wait (for this cycle, insertion of a wait state is not simple).

The Wait State Control Register can be programmed to generate multiple Wait states during different CPU cycles listed as follows.

Memory Wait and Opcode wait. The Wait State Generator can put 0 to 3 wait states in memory accesses Additionally, one added wait state can be inserted during an /M1 (Opcode fetch) cycle, because /M1 cycle's timing requirement is tighter than memory Read/Write cycles. It generates wait states to the Memory Access in a specified address range, which is programmed in the Memory Wait Boundary Register.

I/O Wait. The Wait State generator can put 0, 2, 4 or 6 wait states in I/O accesses. Regardless of the programming of this field, no I/O wait states are inserted for accesses to on-chip peripherals.

Interrupt Vector Wait. During Interrupt acknowledge cycle, the Wait State Generator can insert one wait state after /IORQ goes active, to extend the time between /IORQ fall to vector fetch by CPU. It allows a slow vector response device.

Interrupt Daisy Chain Wait and RETI sequence extension. During Interrupt acknowledge cycle, the Wait State Generator can insert 0, 2, 4 or 6 wait states between /M1 falling to /IORQ falling edge, to extend the time required to settle daisy chain. This allows a longer daisy chain. Also, this field controls the number of wait states inserted during RETI (Return From Interrupt) cycle. If specified to insert 4 or 6 wait states during Interrupt Acknowledge cycle, Wait State Generator also inserts wait states during RETI fetch sequence. This sequence is generated with two op-code fetch cycles (Op-code is EDh followed by 4Dh). It inserts 2 or 4 wait states, respectively, if op-code followed by EDh is 4Dh. One wait state if the following op-code is not 4Dh.

Chip Select Signals (Z84C13/C15 Only)
The Z84C13/C15 has an enhanced feature of adding two chip select (/CS0, /CS1) pins. Both signals are originally IC test pins (ICT) on the Z84013/015 The boundary value for each Chip Select Signal is 4 bits wide, and compare with A15-A12 of the address. Each Chip Select Signal goes active when:

/CS0: (D3-D0 of CSBR) ≥ A15-A12 ≥ 0
/CS1: (D7-D4 of CSBR) ≥ A15-A12 > (D3-D0 of CSBR)

(Where CSBR is the contents of Chip Select Boundary Register.)

There is also a separate /CS enable bit. /CS0 is enabled on power-up with a boundary value of "F" causing /CS0 to go active for all memory accesses. /CS1 is disabled on

power-up, and boundary address is undefined. These features are controlled via the I/O control registers located at I/O address EEh and EFh. **Note that a glitch may be observed on these pins because address decode logic is decoding only A15-A12, without any control signals.** For more detail, please refer to the "Programming section."

Other functional features (Z84C13/C15 Only)
For more system design flexibility, the Z84C13/C15 has the following unique features. These features are controlled by MCR (Misc. Control Register) which is indirectly accessed via the System Control Register Pointer (SCRP, I/O address EEh), and System Control Data Port (SCDP, I/O address EFh). For more details, please refer to the "Programming" section.

- Clock Divide-by-one option
- Reset Output Disable
- 32-bit CRC Generation/Checking

Clock Divide-by-One Option. This feature is programmed through Bit D4 of MCR. Upon Power-On reset, the Clock from on-chip CGC is passed through a divide-by-two circuit. By setting this bit to one, the divide-by-two circuit is bypassed so the clock on the CLKOUT pin is equal to X'tal input. If the clock is applied to the CLKIN pin from external clock source, the status of this bit is ignored. Upon Power-on Reset, it is cleared to 0. For details, please refer to "Programming" section.

Reset Output Disable. This feature is programmed by Bit D3 of MCR. If this bit is cleared to "0", The /RESET pin becomes "Open-drain output" and is driven to "0" for 16-clock cycles from the falling edge of /RESET input. This feature is for the cases where /RESET is used to get out from the "HALT" state If this bit is set to one, the on-chip reset circuit will not drive /RESET pin.

32-bit CRC Generation/Checking. This feature is programmed by Bit D2 of MCR By setting this bit to one, Channel A of SIO is set to use the 32-bit CRC generator/checker instead of the original 16-bit CRC generator/checker in synchronous communication modes. The polynomial to be used in this mode is the one for the protocols such as V.42, and is (X32 + X26 + X23+ X22 + X16 + X12 + X11 + X10 + X8 + X7 + X5 + X4 + X2 + X + 1). Upon Power-on Reset, this bit is cleared to 0.

Evaluation Mode
The IPC has a built evaluation (or development) mode feature which allows the users to utilize standard Z80 development systems conveniently. This mode virtually replaces the on-chip Z80 CPU with the external CPU. In this mode, the on-chip CPU is electrically disconnected from internal bus and all 3-state signals (A15-0, D7-0, /MREQ, /IORQ, /RD, /WR, /HALT, /M1 and /RFSH; for C13/C15, /BUSREQ as well) are tri-stated, or changed to input. This allows the development system CPU to take over and use the internal I/O registers of the IPC exactly as if the CPU was on-chip.

Z84013/015 Only. When this signal is active,the /M1, /HALT and /RFSH pins are put in the high-impedance state. In using the Z84013/015 as an evaluator chip, the CPU is electrically disconnected (put in high-impedance state) after one machine cycle is executed with the EV signal being "1" and the /BUSREQ signal being "0". Then, on-chip resources can be accessed from the outside. /BUSACK is disconnected by an externally connected circuit.

Z84C13/C15 Only. If the EV pin is tied to Vcc on Power-up, the Z84C13/C15 enters into an evaluation mode. In this mode, the internal CPU is immediately disconnected from the internal bus and all 3-state signals mentioned above are tri-stated, or changed to input. Note that the /WAIT pin became the OUTPUT pin in EV mode, and the Wait State Generator generates wait states only as programmed. If the target application board has a separate wait state generator, modification of the target may be required. /BUSACK is 3-stated in this mode.

The Z84C13/C15 behaves similarly to the situation where in regular operation, the /BUSREQ signal is asserted by an external master causing all 3-state signals to be tri-stated by the Z84C13/C15 during T1 of the following machine cycle. The /BUSREQ approach was not used for the evaluation mode to avoid significant external circuitry to work around the time period before the external CPU uses the bus for Z84C13/C15 accesses.

PROGRAMMING

I/O address assignment
The IPC 's on-chip peripherals' I/O addresses are listed in Table 1. They are fully decoded from A7-A0 and have no image.The registers with Z84C13/C15 located at I/O Address EEh and EFh are the registers to control enhanced features to Z84013/015, and not assigned on Z84C013/015.

Table 1. I/O Control Register Address

Address	Device	Channel	Register
10h	CTC	Ch 0	Control Register
11h	CTC	Ch 1	Control Register
12h	CTC	Ch 2	Control Register
13h	CTC	Ch 3	Control Register
18h	SIO	Ch. A	Data Register
19h	SIO	Ch. A	Control Register
1Ah	SIO	Ch. B	Data Register
1Bh	SIO	Ch B	Control Register
1Ch	PIO	Port A	Data Register (Not with Z84x13)
1Dh	PIO	Port A	Command Register (Not with Z84x13)
1Eh	PIO	Port B	Data Register (Not with Z84x13)
1Fh	PIO	Port B	Command Register (Not with Z84x13)
F0h	Watch-Dog Timer		Master Register (WDTMR)
F1h	Watch-Dog Timer		Control Register (WDTCR)
F4h	Interrupt Priority Register		
EEh			System Control Register Pointer (SCRP) (Not with Z84013/015)
EFh			System Control Data Port (SCDP) (Not with Z84013/015)
	Through SCRP and SCDP		Control Register 00 - Wait State Control register (WCR) Control Register 01 - Memory Wait state Boundary Register (MWBR)
			Control Register 02 - Chip Select Boundary Register (CSBR) Control Register 03 - Misc. Control Register (MCR)

PIO REGISTERS

For more detailed information, please refer to the PIO Technical Manual. These registers are not in the Z84x13.

Interrupt Vector Word

The PIO logic unit is designed to work with the Z80 CPU in interrupt Mode 2. The interrupt word must be programmed if interrupts are used. Bit D0 must be a zero (Figure 11).

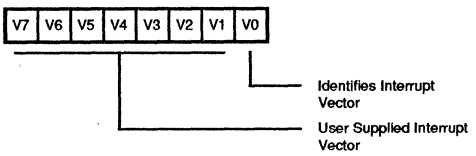


Figure 11. PIO Interrupt Vector Word

Mode Control Word

Selects the port operating mode. This word is required and is written at any time (Figure 12).

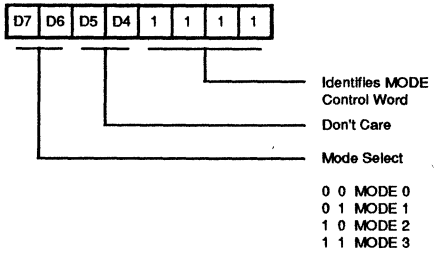


Figure 12. PIO Mode Control Word

I/O Register Control Word

When Mode 3 is selected, the Mode Control Word is followed by the I/O Register Control Word This word configures the I/O register, which defines which port lines are inputs or outputs. A "1" indicates input while a "0" indicates output. This word is required when in Mode 3 (Figure 13).

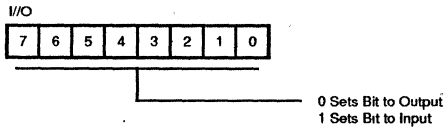
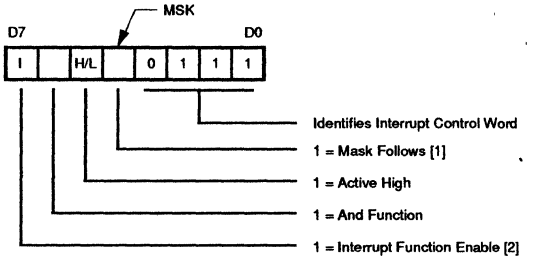


Figure 13. I/O Register Control Word

Interrupt Control Word

In Mode 3 operation, handshake signals are not used Interrupts are generated as a logic function of the input signal levels. The Interrupt Control Word sets the logic conditions and the logic levels required for generating an interrupt. Two logic conditions or functions are available: AND (if all input bits change to the active level, an interrupt is triggered), OR (if any one of the input bits change to the active logic level, an interrupt is triggered). The user can program which input bits are to be considered as part of this logic function. Bit D6 sets the logic function, bit D5 sets the logic level, and bit D4 specifies a mask control word to follow (Figure 14).



Note:
[1] Regardless of the operating mode, setting Bit D4 = 1 causes any pending interrupts to be cleared.
[2] The port interrupt is not enabled until the interrupt function enable is followed by an active /M1.

Figure 14. Interrupt Control Word

Mask Control Word

This word sets the mask control register, thus allowing any unused bits to be masked off. If any bits are to be masked, then bit D4 of the interrupt Control Word is set. When bit D4 of the interrupt Control Word is set, then the next word programmed is the Mask Control Word. To mask an input bit, the corresponding Mask Control Word bit is a "1" (Figure 15).

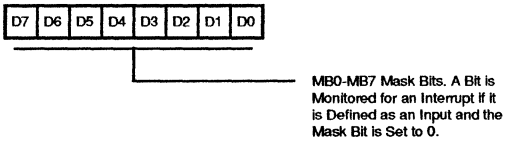


Figure 15. Mask Control Word

Interrupt Disable Word

This word can be used to enable or disable a port's interrupts without changing the rest of the port's interrupt conditions (Figure 16).

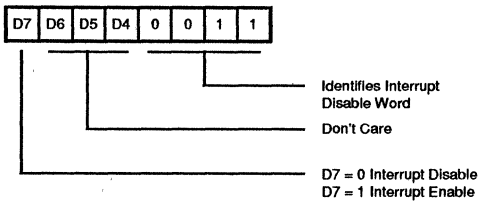


Figure 16. Interrupt Disable Word

CTC CONTROL REGISTERS

For more detailed information, refer to the CTC Technical Manual.

Channel Control Word

This word sets the operating modes and parameters as described below. Bit D0 is a "1" to indicate that this is a Control Word (Figure 17).

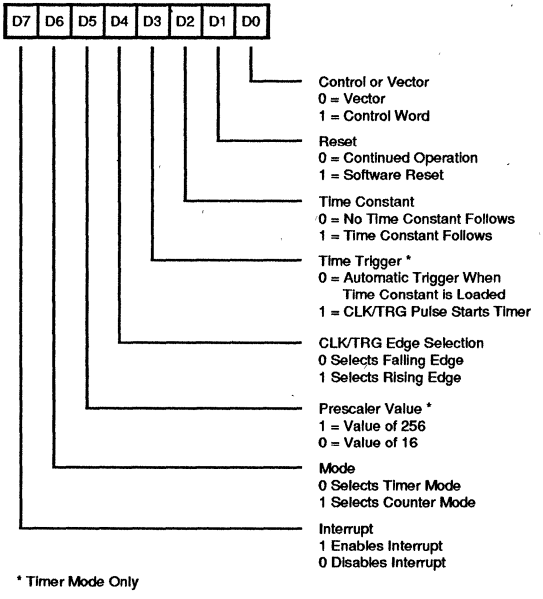


Figure 17. CTC Channel Control Word

Bit D7. *Interrupt Enable.* This bit enables the interrupt logic so that an internal INT can be generated at zero count. Interrupts are programmed in either mode and may be enabled or disabled at any time.

Bit D6. *Mode Bit.* This bit selects either Timer Mode or Counter Mode.

Bit D5. *Prescalor Factor.* This bit selects the prescalor factor for use in the timer mode Either divide-by-16 or divide-by-256 is available.

Bit D4. *Clock/Trigger Edge Selector.* This bit selects the active edge of the CLK/TRG input pulses.

Bit D3. *Timer Trigger.* This bit selects the trigger mode for timer operation. Either automatic or external trigger may be selected.

Bit D2. *Time Constant.* This bit indicates that the next word programmed is time constant data for the downcounter.

Bit D1. *Software Reset.* Writing 1 to this bit indicates a software reset operation, which stops counting activities until another time constant word is written.

Time Constant Word

Before a channel starts counting, it must receive a time constant word The time constant value is anywhere between 1 and 256, with "0" being accepted as a count of 256 (Figure 18).

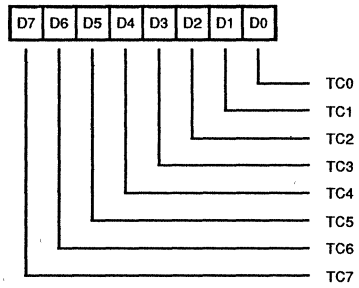


Figure 18. CTC Time Constant Word

Interrupt Vector Word

If one or more of the CTC channels have interrupt enabled, then the Interrupt Vector Word must be programmed Only the five most significant bits of this word are programmed, and bit D0 must be "0". Bits D2-D1 are automatically modified by the CTC channels when it responds with an interrupt vector (Figure 19).

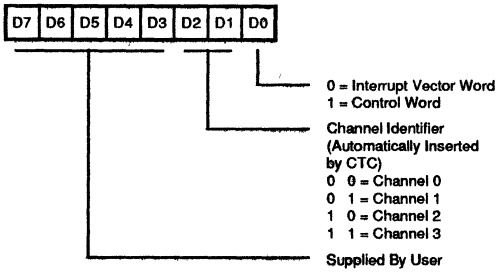
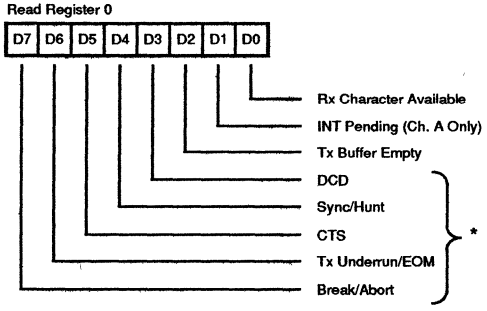


Figure 19. CTC Interrupt Vector Word

SIO REGISTERS

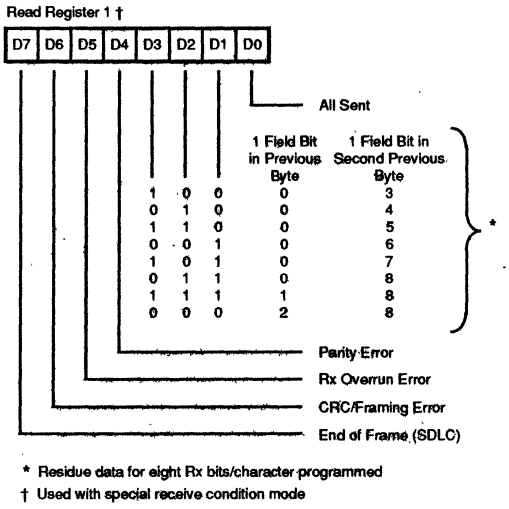
For more detailed information, refer to the SIO Technical Manual.

Read Registers. The SIO channel B contains three read registers while channel A contains only two that are read to obtain status information. To read the contents of a register (rather than RR0), the program must first write a pointer to WR0 in exactly the same manner as a write operation. The next I/O read cycle will place the contents of the selected read registers onto the data bus (Figure 20a, b, c).



* Used With "External/Status Interrupt" Modes

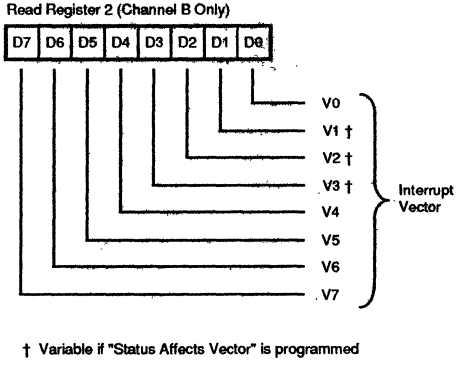
Figure 20a. SIO Read Register 0



* Residue data for eight Rx bits/character programmed

† Used with special receive condition mode

Figure 20b. SIO Read Register 1



† Variable if "Status Affects Vector" is programmed

Figure 20c. SIO Read Register 2

Write Registers. The SIO Channel B contains eight write registers while Channel A contains only seven that are programmed to configure the operating mode characteristics of each channel. With the exception of WR0, programming the write registers is a two step operation. The first operation is a pointer written to WR0 which points to the selected register. The second operation is the actual control word that is written into the register to configure the SIO channel (Figure 21)

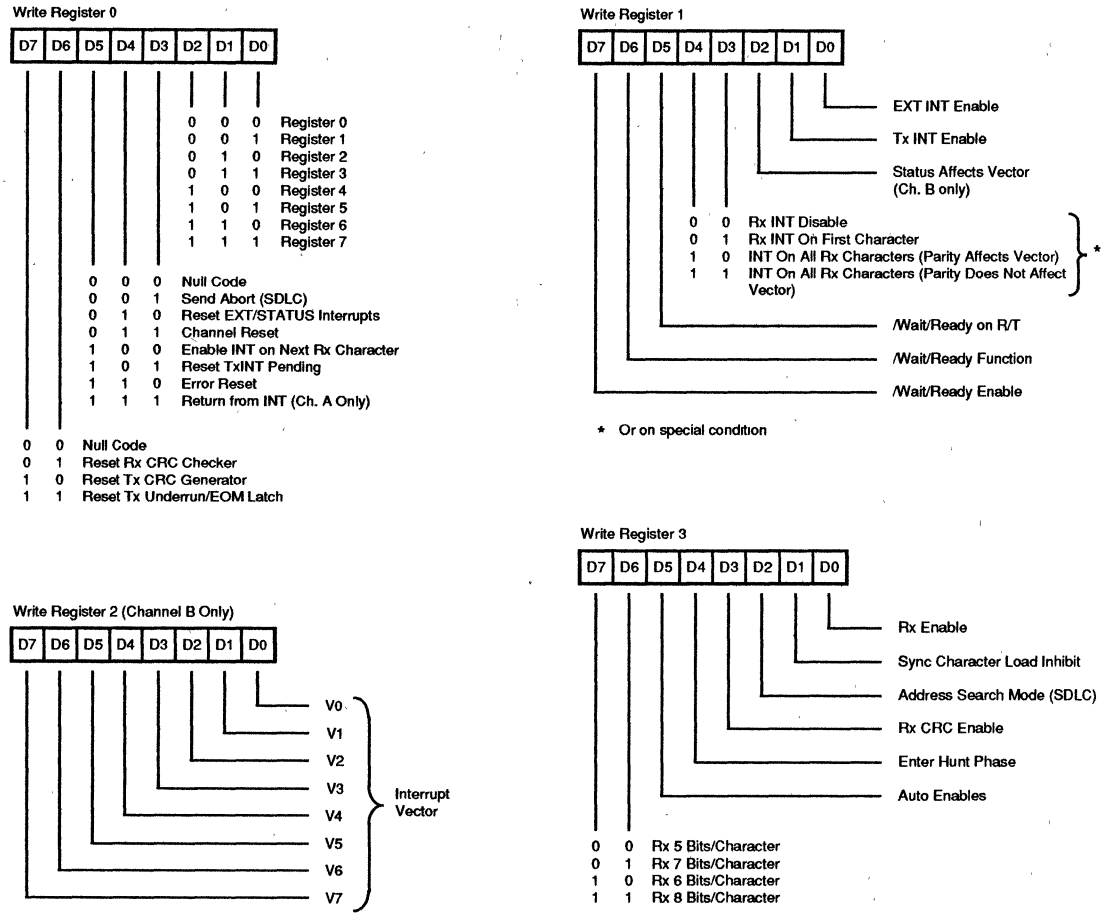


Figure 21. SIO Write Registers

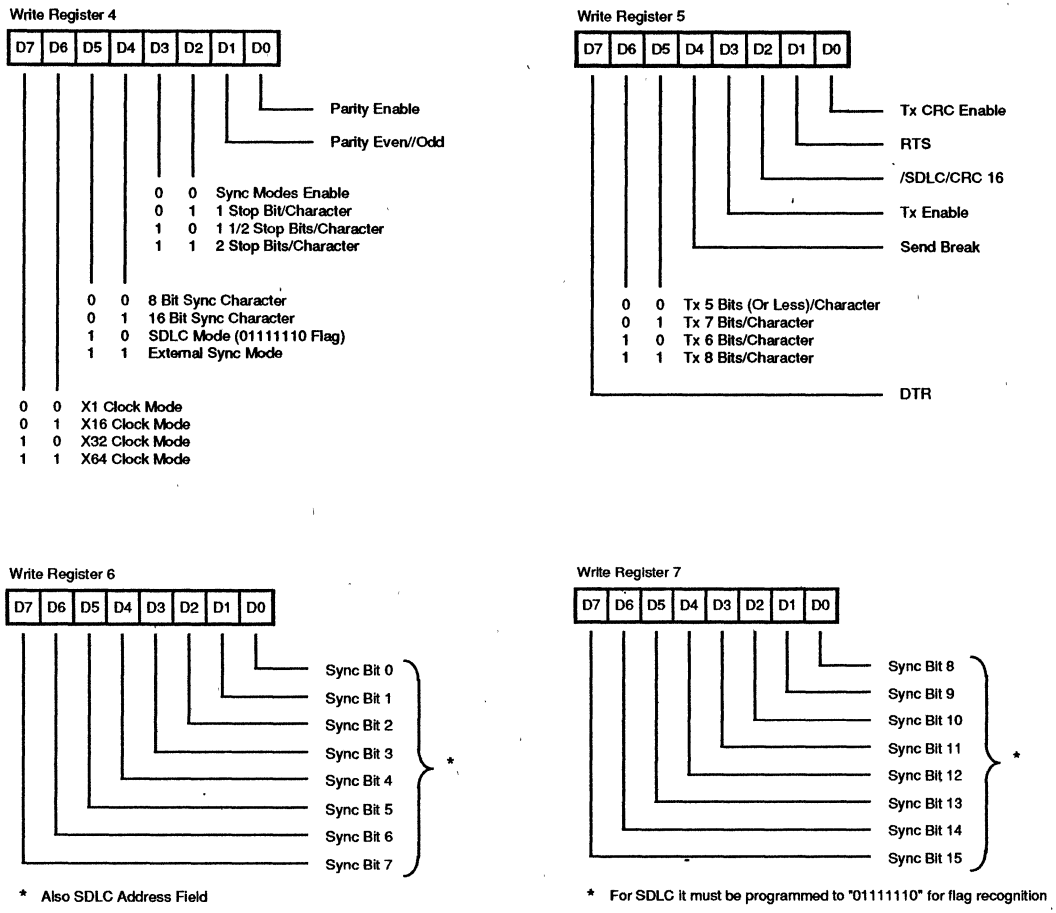


Figure 21. SIO Write Registers (Continued)

WATCH DOG CONTROL REGISTERS

There are two registers to control Watch Dog Timer operations. These are Watch Dog Timer Master Register (WDTMR; I/O Address F0h) and the WDT Command Register (WDTCR; I/O Address F1h). Watch Dog Timer Logic has a "double key" structure to prevent the WDT disabling error, which may lead to the WDT operation to stop due to program runaway. Programming the WDT follows this procedure. Also, these registers program the power-down mode of operation The "Second key" is needed when turning off the Watch Dog Timer.

Enabling the WDT. The WDT is enabled by setting the WDT Enable Bit (D7:WDTE) to "1" and the WDT Periodic field (D5,D6:WDTP) to the desired time period. These command bits are in the Watch Dog Timer Master Register (WDTMR; I/O Address F0h).

Disabling the WDT. The WDT is disabled by clearing WDT Enable bit (WDTE) in the WDTMR to "0" followed by writing "B1h" to the WDT Command Register (WDTCR; I/O Address F1h).

Clearing the WDT. The WDT can be cleared by writing "4Eh" into the WDTCR.

Watch Dog Timer Master Register (WDTMR;I/O address F0h). This register controls the activities of the Watch Dog Timer and selects power-down mode of operation (Figure 22).

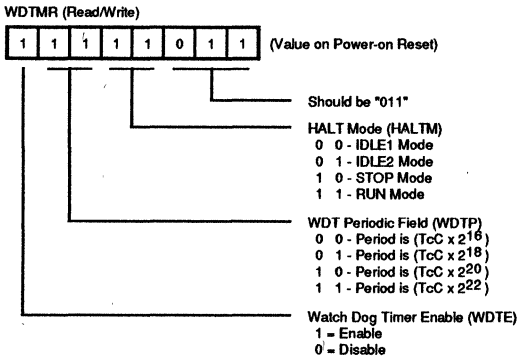


Figure 22. Watch Dog Timer Master Register

Bit D7. *Watch Dog Timer Enable* (WDTE). This bit controls the activities of Watch Dog Timer. The WDT can be enabled by setting this bit to "1". To disable WDT, write "0" to this bit followed by writing "B1h" in the WDT Command Register. Watch Dog Timer Logic has a "double key" structure to prevent the WDT disabling error, which may lead to the WDT operation to stop, due to program runaway. Upon Power-on reset, this bit is set to "1" and the WDT is enabled.

Bit D6-D5. *WDT Periodic field* (WDTP). This two bit field determines the desired time period. Upon Power-on reset, this field sets to "11".

00 - Period is (TcC * 2^{16})
01 - Period is (TcC * 2^{18})
10 - Period is (TcC * 2^{20})
11 - Period is (TcC * 2^{22})

Bit D4-D3. *HALT mode* (HALTM). This two bit field specifies one of four power-down modes. To change this field, write "DBh" to the WDT command register, followed by a write to this register. For detailed descriptions of this field, please refer to the section "Mode of operations." Upon Power-on Reset, this field is set to 11, which specifies "RUN mode."

00 - IDLE 1 Mode
01 - IDLE 2 Mode
10 - STOP Mode
11 - RUN Mode

Bit D2-D0. *Reserved.* These three bits are reserved and should always be programmed as "011". A read to these bit returns "011".

Watch Dog Timer Command Register (WDTCR; I/O address F1h). In conjunction with the WDTMR, this register works as a "Second key" for the Watch Dog Timer. This register is write only (Figure 23).

Write B1h after clearing WDTE to "0" - Disable WDT.
Write 4Eh - Clear WDT.
Write DBh followed by a write to HALTM - Change Power-down mode.

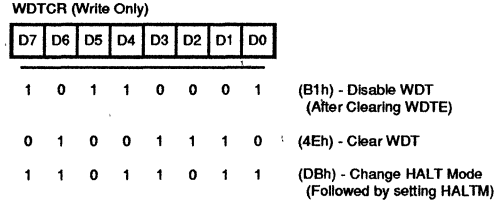


Figure 23. Watch Dog Timer Command Register

INTERRUPT PRIORITY REGISTER
(INTPR; I/O address F4h)

This register (write only) is provided to determine the interrupt priority for the CTC, SIO and the PIO (Figure 24).

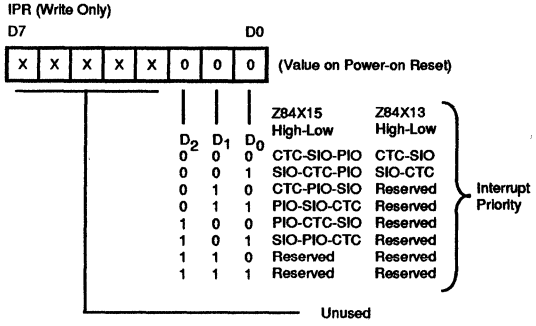


Figure 24. Interrupt Priority Register

Bit D7-D3. Unused

Bit D2-D0. This field specifies the order of the interrupt daisy chain. Upon Power-on Reset, this field is set to "000".

	Z84C15 High - Low	Z84C13 High - Low
000	CTC-SIO-PIO	CTC-SIO
001	SIO-CTC-PIO	SIO-CTC
010	CTC-PIO-SIO	Reserved
011	PIO-SIO-CTC	Reserved
100	PIO-CTC-SIO	Reserved
101	SIO-PIO-CTC	Reserved
110	Reserved	Reserved
111	Reserved	Reserved

REGISTERS FOR SYSTEM CONFIGURATION

(The following registers are not available on Z84013/015.) There are four indirectly accessible registers to determine System configuration with the Z84C13/C15. These indirectly accessible registers are: Wait State Control Register (WCR, Control Register 00h), Memory Wait Boundary Register (MWBR, Control Register 01h), Chip Select Boundary Register (CSBR, Control Register 02h) and Misc. Control Register (MCR, Control Register 03h). To access these registers, Z84C13/C15 writes "register number to be accessed" to the System Control Register Pointer (SCRP, I/O address EEh), and then accesses the target register through the System Control Data Port (SCDP, I/O address EFh). The pointer which writes into SCRP is kept until modified.

System Control Register Pointer (SCRP, I/O address EEh)
This register stores the pointer to access System Control Registers (WCR, MWBR, CSBR and MCR). This register is Read/Write and it holds the pointer value until modified. Upon Power-on Reset, all bits are cleared to zero. The pointer value, other than 00h to 03h is reserved and is not written. Upon Power-on Reset, this register is set to "00h" (Figure 25).

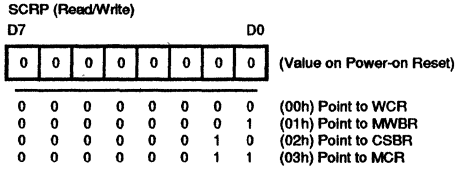


Figure 25. System Control Register Pointer

System Control Data Port (SCDP, I/O address EFh)
This register is to access WCR, MWBR, CSBR and MCR (Figure 26).

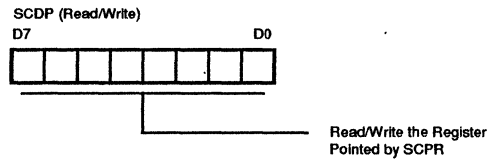


Figure 26. System Control Data Port

Wait State Control Register (WCR, Control Register 00h)
This register can be accessed through SCDP with the pointer value 00h in SCRP (Figure 27). To maintain compatibility with the Z84013/015, the Z84C13/C15 inserts the maximum number of wait states (set all bits of this register to one) for fifteen /M1 cycles after Power-on Reset. It automatically clears the contents of this register (move to no-wait state insertion) on the trailing edge of the 16th /M1 signal unless software has programmed a value. If automatic wait state insertion is needed, the wait state is programmed within this time period. A read to WCR during this period will return FFh, unless programmed.

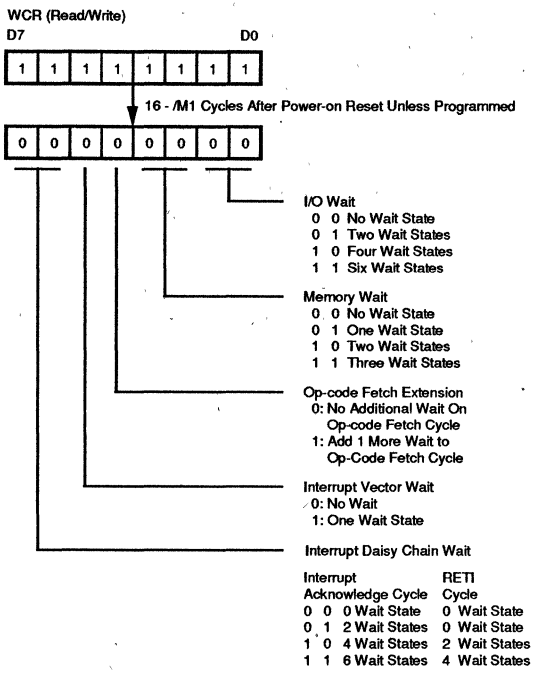


Figure 27. Wait State Control Register

This register has the following fields:

Bit 7-6. *Interrupt Daisy Chain Wait.* This 2-bit field specifies the number of wait states to be inserted during an Interrupt Daisy Chain settle period of the Interrupt Acknowledge cycle, which is/IORQ falls after the settling period from /M1 going active "0". Also, this field controls the number of wait states inserted during the RETI (Return From Interrupt) cycle. If specified to insert 4 or 6 wait states during Interrupt Acknowledge cycle, the Wait state generator also inserts wait states during RETI fetch sequence. This sequence is formed with two op-code fetch cycles (Op-code is EDh followed by 4Dh). It inserts 1 wait state if op-code followed by EDh is NOT 4Dh, and inserts 2 or 4 wait states, respectively, if the following op-code is 4Dh.

Interrupt Acknowledge	RETI cycle
00 - No Wait states	No Wait states
01 - 2 Wait states	No Wait states
10 - 4 Wait states	2 Wait states
11 - 6 Wait states	4 Wait states

For fifteen /M1 cycles from Power-on Reset, bits 7-6 are set to "11" . They clear to "00" on the trailing edge of the 16th /M1 signal unless programmed.

Bit 5. *Interrupt Vector Wait.* While this bit is set to one, the wait state generator inserts one wait state after the /IORQ signal goes active during the Interrupt acknowledge cycle. This gives more time for the vector read cycle. While this bit is cleared to zero, no wait state is inserted (standard timing). For fifteen /M1 cycles from Power-on Reset, this bit is set to "1", then cleared to "0" on the trailing edge of the 16th /M1 signal, unless programmed.

Bit 4. *Opcode Fetch Extension.* If this bit is set to "1", one additional wait state is inserted during the Op-code fetch cycle in addition to the number of wait states programmed in the Memory Wait field. For fifteen /M1 cycles from Power-on Reset, this bit is set to "1", then cleared to "0" on the trailing edge of the 16th /M1 signal, unless programmed.

Bit 3-2. *Memory Wait States.* This 2-bit field specifies the number of wait states to be inserted during memory Read/ Write transactions.

00 - No Wait states
01 - 1 Wait states
10 - 2 Wait states
11 - 3 Wait states

For fifteen /M1 cycles from Power-on Reset, these bits are set to "11", then cleared to "00" on the trailing edge of the16th /M1 signal, unless programmed.

Bit 1-0. *I/O Wait states.* This 2-bit field specifies the number of wait states to be inserted during I/O transactions.

00 - No Wait states
01 - 2 Wait states
10 - 4 Wait states
11 - 6 Wait states

For fifteen /M1 cycles from Power-on Reset, these bits are set to "11", then cleared to "00" on the trailing edge of the16th /M1 signal, unless programmed. For the accesses to the on-chip I/O registers, no Wait states are inserted regardless of the programming of this field.

Memory Wait Boundary Register (MWBR, Control Register 01h)

This register specifies the address range to insert memory wait states. When accessed memory addresses are within this range, the Memory Wait State generator inserts Memory Wait States specified in the Memory Wait field of WCR (Figure 28).

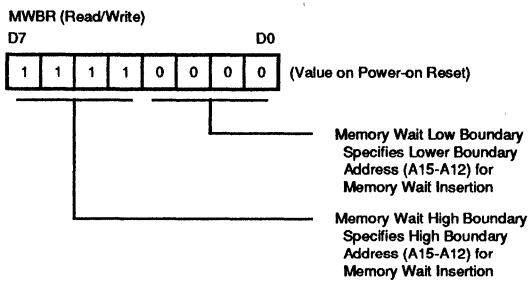


Figure 28. Memory Wait Boundary Register

Bit D7-D4. *Memory Wait High Boundary.* This field specifies A15-A12 of the upper address boundary for Memory Wait.

Bit D3-D0. *Memory Wait Low Boundary.* This field specifies A15-12 of the lower address boundary for Memory Wait.

Memory Wait states are inserted for the address range:

(D7-D4 of MWBR) ≥ A15-A12 ≥ (D3-D0 of MWBR)

This register is set to "F0h" on Power-on Reset, which specifies the address range for Memory Wait as "0000h to FFFFh".

Chip Select Boundary Register (CSBR, Control Register 02h)

This register specifies the address range for each chip select signal. When accessed memory addresses are within this range, chip select signals are active (Figure 29).

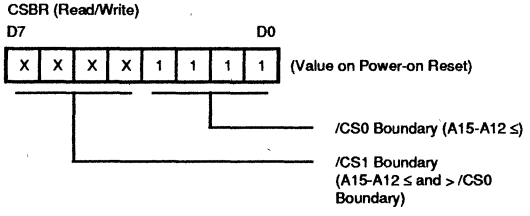


Figure 29. Chip Select Boundary Register

D7-D4. */CS1 Boundary Address.* These bits specify the boundary address range for /CS1. The bit values are ignored on power-up as the /CS1 enable bit is off. The /CS1 is asserted if the address lines A15-12 have an address value greater than the programmed value for /CS0, and less than or equal to the programmed value in these bits.

D3-D0. */CS0 Boundary Address.* These bits specify the boundary address range for /CS0. /CS0 is asserted if the address lines A15-12 have an address value less than or equal to the programmed boundary value. The /CS0 enable bit in the MCR must be set to 1. Upon Power-up reset, these bits come up as all 1's so that /CS0 is asserted for all addresses.

Chip Select signals are active for the address range:

/CS0: (D3-D0 of CSBR) ≥ A15-A12 ≥ 0
/CS1: (D7-D4 of CSBR) ≥ A15-A12 > (D3-D0 of CSBR)

This register is set to "xxxx1111b" on Power-on Reset, which specifies the address range of /CS0 for "0000h to FFFFh" (all Memory location) and /CS1 "undefined."

Misc Control Register (MCR, Control Register 03h)

This register specifies miscellaneous options on this device (Figure 30).

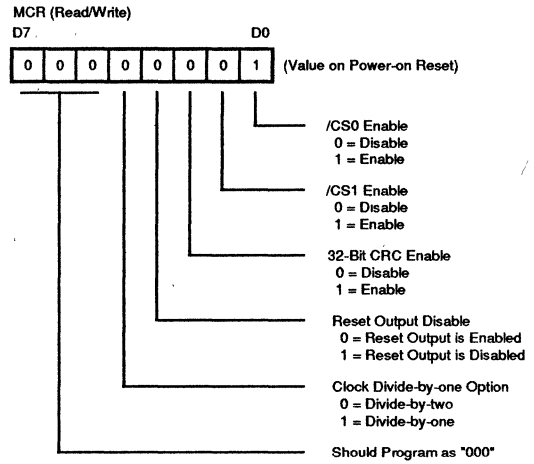


Figure 30. Misc Control Register

Bit D7-D5. *Reserved.* These three bits are reserved and are always programmed as "000".

Bit D4. *Clock Divide-by-one option.* "0"-Disable, "1"-enable. On-chip CGC unit has divide-by-two circuit. By setting this bit to one, this circuit is bypassed and CLKOUT is equal to X'tal oscillator frequency (or external clock input on the XTAL1 pin). This bit has no effect when the on-chip CGC unit is not in use and the external system clock is fed from CLKIN pin. Upon Power-on Reset, this bit is cleared to 0 and the clock is divided by two.

Bit D3. *Reset Output Disable.* "0"-Reset output is enabled, "1"-Reset output is disabled. This bit controls the /RESET signal and is driven out when reset input is used to take the Z84C13/C15 out of the "Halt" state. The reset pulse is driven out for 16-clock cycles from the falling edge of /RESET input, unless this bit is set. Upon Power-on reset, this bit is cleared to 0.

Bit D2. *32-Bit CRC enable.* "0"-Normal mode (16-bit CRC) "1"-32-bit CRC generation/Checking is enabled on SIO Channel A. This bit determines if the 32-bit CRC feature is enabled on Channel A of the SIO. If this bit is 0, the SIO is in a normal mode of operation. If this bit is set to 1, a normal CRC generator/checker is replaced with a 32-bit CRC generator/checker. Upon Power-on Reset, this bit is clear to "0".

Bit D1. */CS1 Enable. "0"-Disable, "1"-Enable.* This bit enables /CS1 output. While this bit is "0", /CS1 is forced to "1". While this bit is "1", /CS1 carries the address range specified in the CSBR. Upon Power-on Reset, this bit is cleared to "0".

Bit D0. */CS0 Enable. "0"-Disable, "1"-Enable.* This bit enables /CS0 output. While this bit is "0", /CS1 pin is forced to "1". While this bit is "1", the /CS0 carries address range specified in the CSBR. Upon Power-on Reset, this bit is set to "1".

Operation modes

There are four kinds of operation modes available for the IPC in connection with clock generation; RUN Mode, IDLE1/2 Modes and STOP Mode.

The Operation mode is effective when the HALT instruction is executed. Restart of the MPU from the stopped state under IDLE1/2 Mode or STOP mode is affected by inputting either /RESET or interrupt (/NMI or /INT). The mode selection of these power-down modes is made by programming the HALTM field (Bit D4-3) of WDTMR.

Setting Halt Mode

Duplicate control is provided to prevent the stopping of the WDT operation caused by the halt mode setting an error due to program runaway. As described in the programming section, changing the Halt Mode field of WDTMR is in two steps. First, write "DBh" to WDTCR followed by a write to the WDTMR with the value in HALTM. Table 2 has descriptions of each mode, and Table 3 has device status in the Halt state.

Table 2. Power-down Modes

(When using on-chip CGC unit; CLKOUT and CLKIN are tied together)

Operation Mode	WDTMR Bit D4	Bit D3	Description at HALT State
RUN Mode	1	1	The IPC continues the operation and continuously supplies a clock to the outside.
IDLE1 Mode	0	0	The internal oscillator's operation is continued. Clock output (CLKOUT) as well as internal clock to the CPU, PIO, SIO, CTC and the Watch Dog Timer is stopped at "0" level of T4 state in the halt instruction operation code fetch cycle.
IDLE2 Mode	0	1	The internal oscillator and the CTC's operation continues and supplies clock to the outside on the CLKOUT pin continuously. But the internal clock to the CPU, PIO, SIO and the Watch Dog Timer is stopped at "0" level of T4 state in the halt instruction operation code fetch cycle.
STOP Mode	1	0	All operations of the internal oscillator, clock (CLK) output, internal clock to the CPU, PIO, CTC, SIO and the Watch Dog Timer are stopped at "0" level of T4 state in the halt instruction operation code fetch cycle.

Table 3. Device status in Halt state
(When using on-chip CGC unit; CLKOUT and CLKIN are tied together)

Mode	CGC	CPU	CTC	PIO	SIO	WDT	CLKOUT
IDLE1	O	X	X	X	X	X	X
IDLE2	O	X	O	X	X	X	O
STOP	X	X	X	X	X	X	X
RUN	O	O	O	O	O	O	O

O: Operating
X: Stop

All of the operating modes listed here are valid with crystal input (Crystal connected between XTAL1/2 or external clock input on XTAL1). For the external clock on the CLKIN pin, only the IDLE2 and RUN modes are applicable.

TIMING

Basic Timing

The basic timing is explained here with emphasis placed on the halt function relative to the clock generator The following items are identical to those for the Z84C00. Refer to the data sheet for the Z84C00.

- Operation code fetch cycle
- Memory Read/Write operation
- Input/Output operation
- Bus request/acknowledge operation
- Maskable interrupt request operation
- Non-Maskable interrupt request operation
- Reset Operation

Operation When HALT Instruction is Executed. When the CPU fetches a halt instruction in the operation code fetch cycle, /HALT goes active (Low) in synch with the falling edge of T4 state before the peripheral LSI and CPU stops the operation. After this, the system clock generation differs depending upon the operation mode (RUN Mode, IDLE1/2 Mode or STOP Mode). If the internal system clock is running, the CPU continues to execute NOP instruction even in the halt state.

RUN Mode (HALTM = 11). Shown in Figure 31 is the basic timing when the halt instruction is executed in RUN Mode

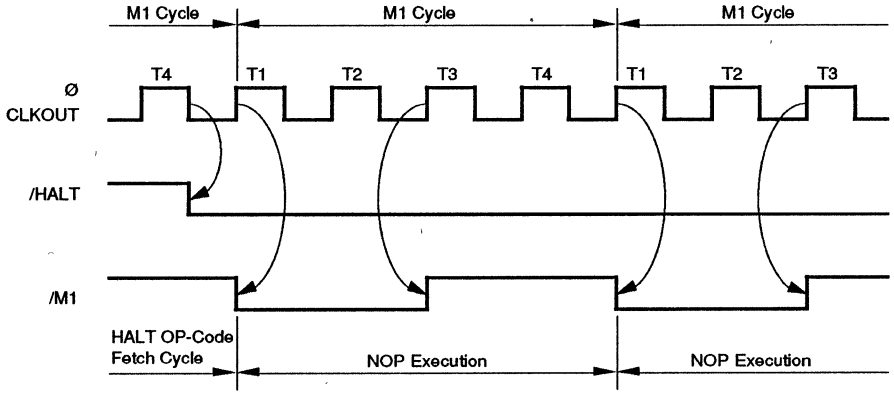


Figure 31. Timing of RUN Mode
(at Halt Instruction Command Execution)

In RUN Mode, output from the CGC unit (CLKOUT) is not stopped and the internal system clock (Ø) continues even after the halt instruction is executed. Therefore, until the halt state is released by the interrupt signal (/NMI or /INT) or /RESET signal, MPU continues to execute HALT instructions (internally executing NOP instructions).

IDLE1 Mode (HALTM=00). Shown in Figure 32 is the basic timing when the halt instruction is executed in IDLE1 Mode.

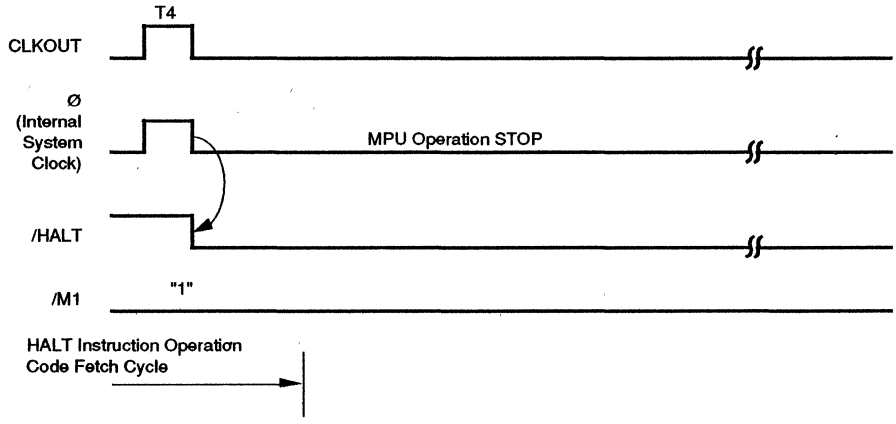


Figure 32. IDLE1 Mode Timing
(At Halt Instruction Execution)

In IDLE1 Mode, the internal oscillator continues to operate, but clock output (CLKOUT) is stopped at T4 Low state of HALT instruction execution. Then all components in the MPU stop their operation. This mode is not supported when the CGC unit is inactive and the external clock is fed from CLKIN pin; CLKOUT should be connected to CLKIN.

IDLE2 Mode (HALTM=01). Shown in Figure 33 is the basic timing when the halt instruction is executed in IDLE2 Mode.

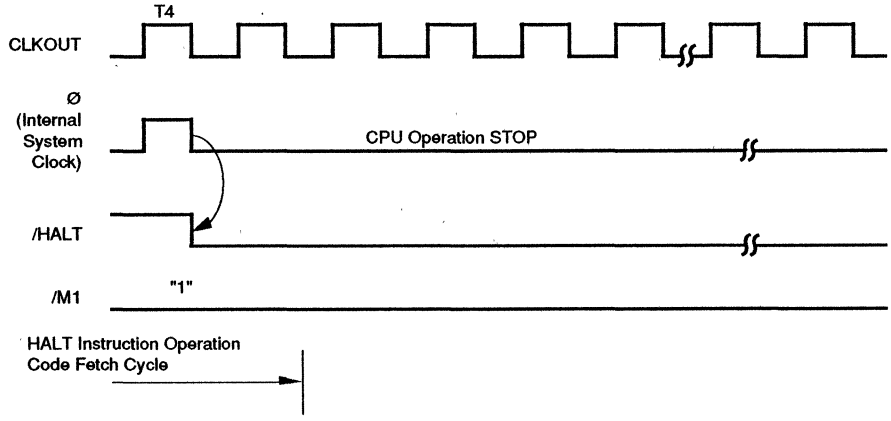


Figure 33. IDLE2 Mode Timing
(At Halt Instruction Execution)

In IDLE2 Mode, the internal oscillator and clock output (CLKOUT) continue to operate. The internal system clock, fed from CLKIN to the components other than CTC is stopped at the T4 Low state of HALT instruction execution.

STOP Mode (HALTM=10). Shown in Figure 34 is the basic timing when the halt instruction is executed in STOP Mode

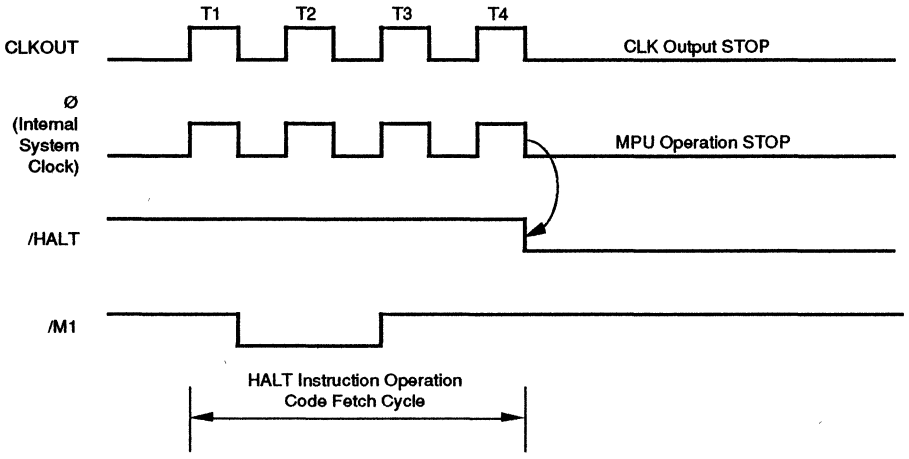


Figure 34. STOP Mode Timing
(At Halt Instruction Execution)

In STOP Mode, the on-chip CGC unit is stopped at T4 Low state of HALT instruction execution. Therefore, clock output (CLKOUT), operation of Watch Dog Timer, CPU, PIO, CTC, SIO are stopped.

Release from Halt State. The halt state of the CPU is released when "0" is input to the /RESET signal and the MPU is reset or an interrupt request is accepted An interrupt request signal is sampled at the leading edge of the last clock cycle (T4 state) of NOP instruction. In case of the maskable interrupt, interrupt will be accepted by an active /INT signal ("0" level). Also, the interrupt enable flip-flop is set to "1". The accepted interrupt process is started from the next cycle.

Further, when the internal system clock is stopped (IDLE1/2 Mode, STOP Mode), it is necessary first to restart the internal system clock. The internal system clock is re-started when /RESET or interrupt signal (/NMI or /INT) is asserted.

RUN Mode (HALTM=11). The halt release operation is enabled by interrupt request in RUN Mode (Figure 35).

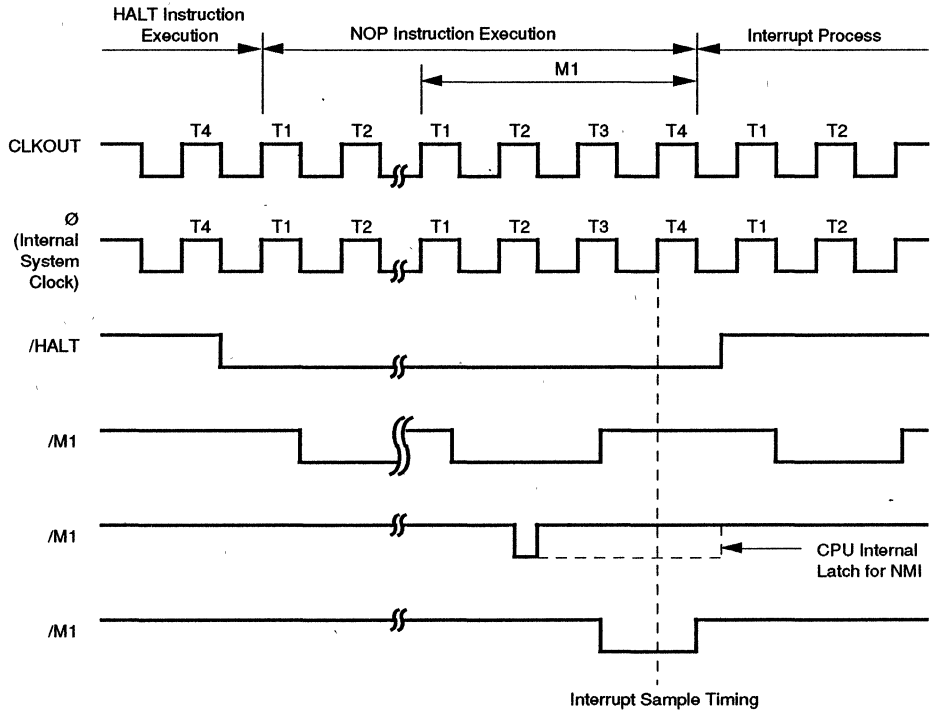


Figure 35. Halt Release Operation Timing By Interrupt Request Signal in RUN Mode

In RUN Mode the internal system clock is not stopped. If the interrupt signal is recognized on the rising clock edge of T4 of the continued NOP instruction, CPU will execute the interrupt process from the next cycle.

The halt release resets CPU in RUN Mode (Figure 36). After reset, CPU will execute an instruction starting from address 0000H. However, in order to reset the CPU it is necessary to keep /RESET signal at "0" for at least 3 system clock cycles. (For Z84C13/C15: 3 clock cycles if Reset output is disabled.) In addition, if /RESET signal becomes "1", after the dummy cycle for at least two T states, CPU executes an instruction from address 0000H.

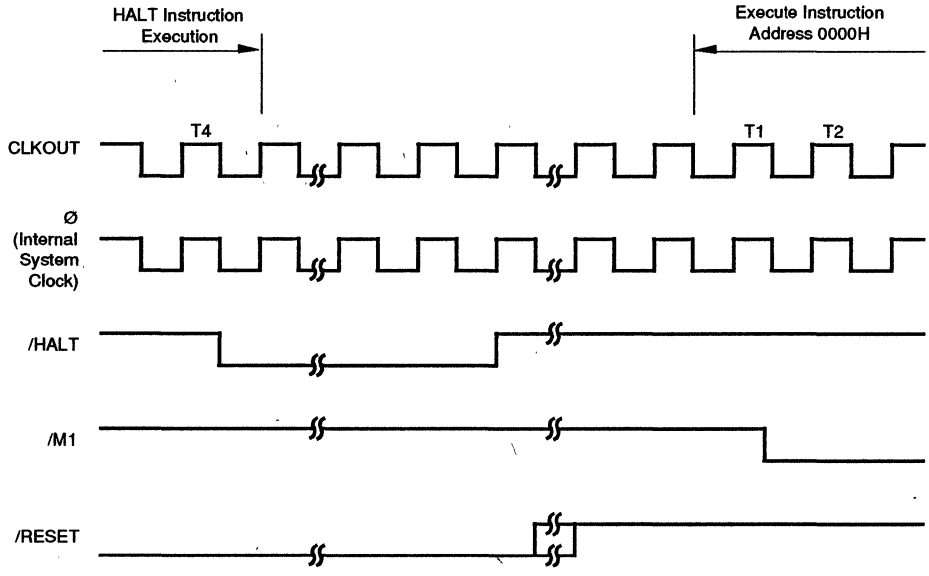
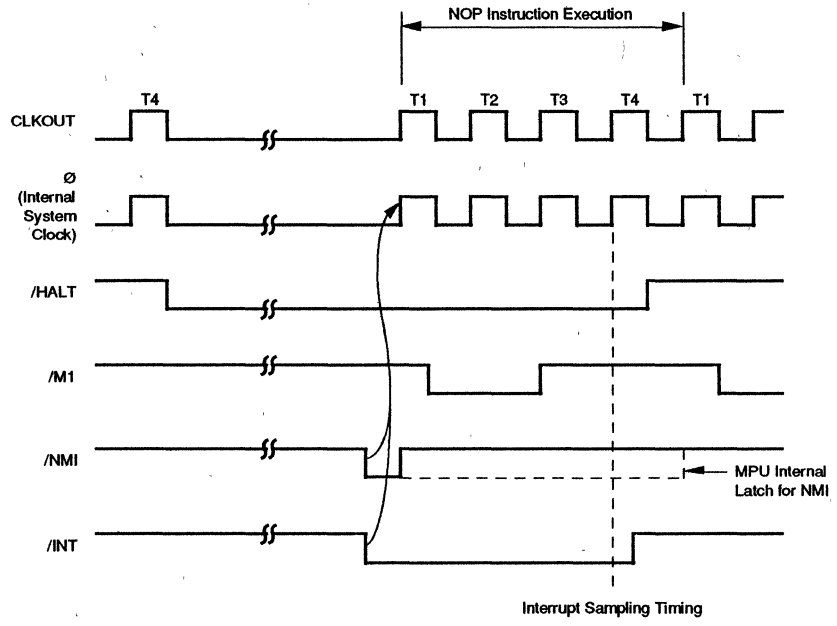
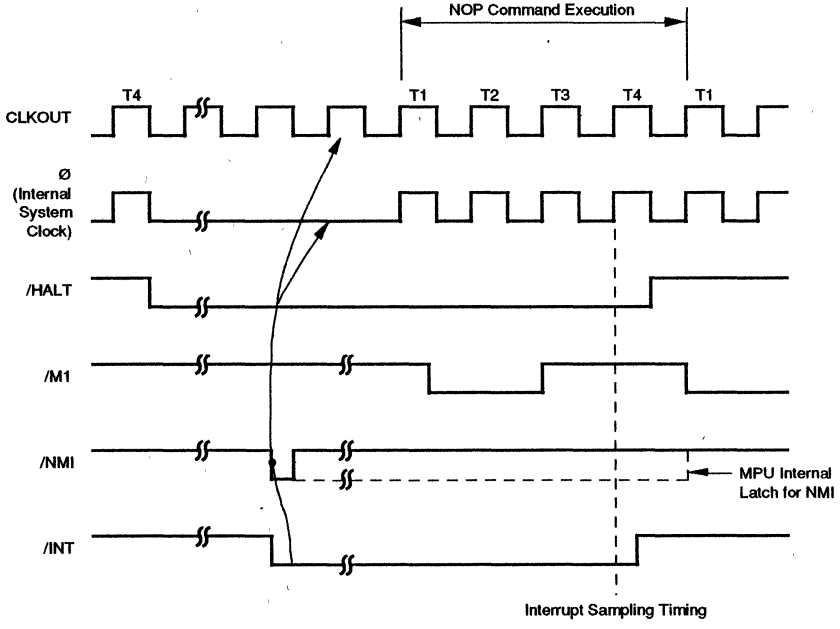


Figure 36. Halt Release Operation Timing By Reset in RUN Mode

IDLE1 Mode (HALTM=00), IDLE2 Mode (HALTM=01). The shown in Figure 37 (a) and in IDLE2 Mode in Figure 37 (b). halt release operation by interrupt signal in IDLE1 Mode is



(a) IDLE1 Mode



(b) IDLE2 Mode

Figure 37. Halt Release Operation Timing By Interrupt Request Signal in IDLE1/2 Mode

When receiving /NMI or /INT signals, the stopped internal system clock starts to feed. In IDLE1 Mode, the IPC starts clock output on CLKOUT at the same time.

The operation stop of CPU in IDLE2 mode is taking place at "0" level during T4 state in the halt instruction op-code fetch cycle. Therefore, after being restarted by the interrupt signal, CPU executes one NOP instruction and samples an interrupt signal at the rise of T4 state during the execution of this NOP instruction, and executes the interrupt process from next cycle.

If no interrupt signal is accepted during the execution of the first NOP instruction after the internal system clock is restarted, CPU is not released from the halt state. It is placed in IDLE1/2 Mode again at "0" level during T4 state of the NOP instruction, stopping the internal system clock. If /INT signal is not at "0" level at the rise of T4 state, no interrupt request is accepted.

The halt release operation resets the IPC in IDLE1 Mode (Figure 38a) and in IDLE2 Mode (Figure 38b).

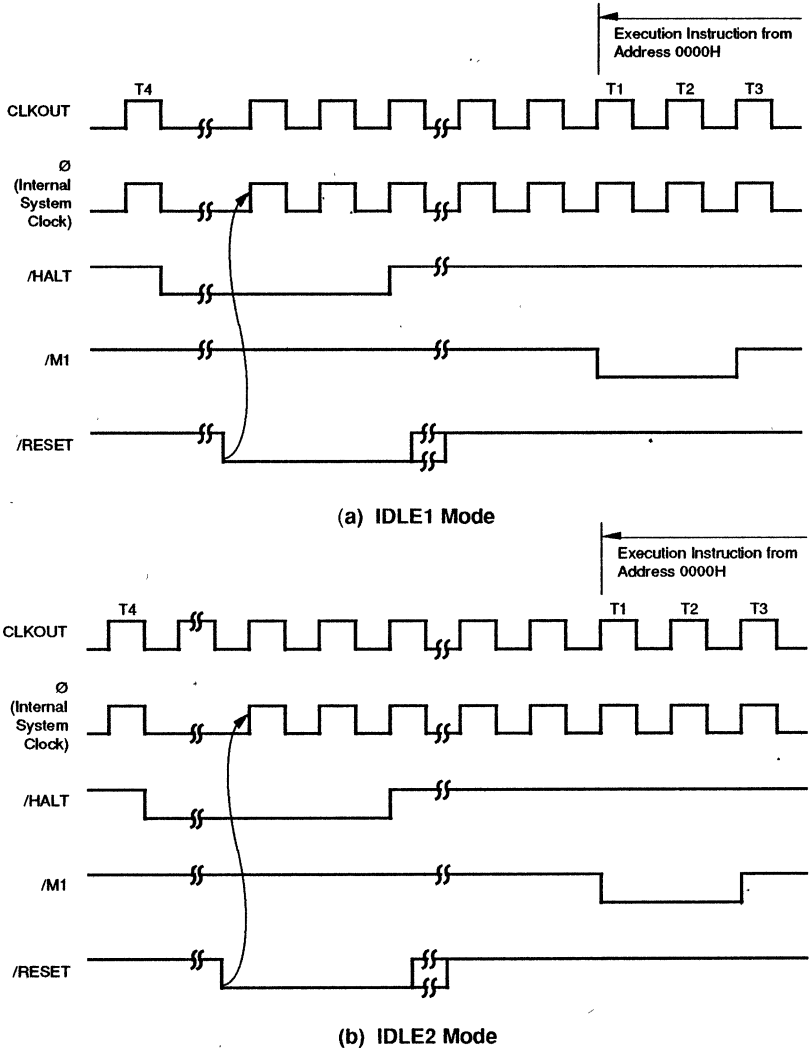


Figure 38. Halt Release Operation Timing By Reset in IDLE1/2 Mode

When /RESET signal at "0" level is input into the IPC, the internal system clock is restarted and the IPC will exocute an instruction stored in address 0000H.

At time of /RESET signal input, it is necessary to take the same care as that in resetting the IPC in RUN Mode.

Halt release in STOP Mode (HALTM=10) by interrupt. The halt release operation by interrupt signal in STOP Mode is shown in Figure 39.

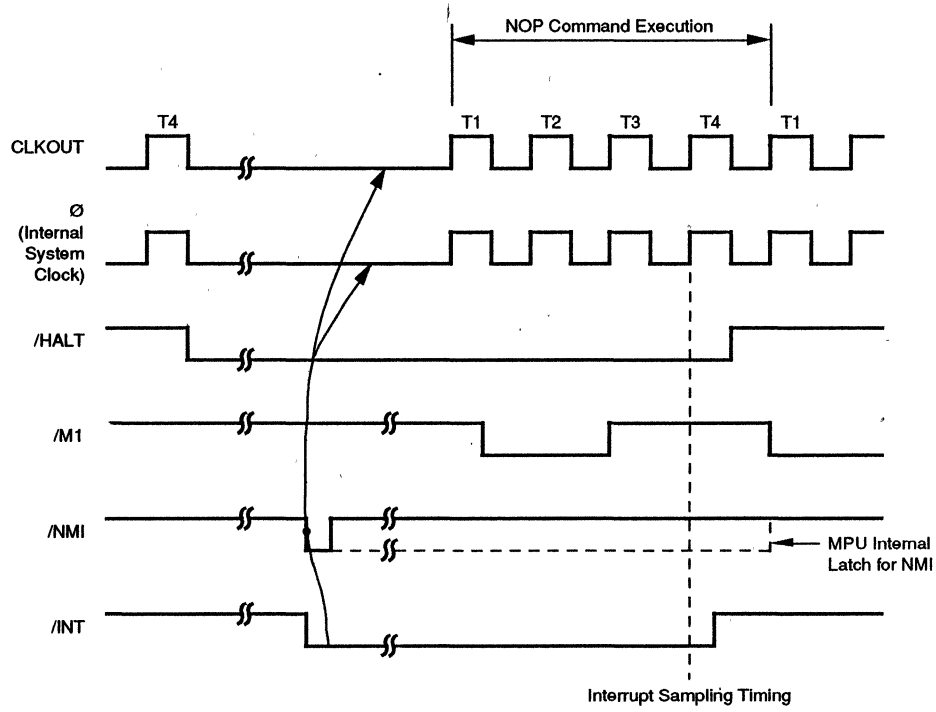


Figure 39. Halt Release Operation Timing By Interrupt Request Signal in STOP Mode

When the IPC receives an interrupt signal, the internal oscillator is restarted. To obtain stabilized oscillation, CLKOUT (and the internal system clock) are started after a start-up time of $(2^{14}+2.5)$ TcC (TcC: Clock Cycle) by the internal counter.

CPU executes one NOP instruction after the internal system clock is restarted. At the same time, it samples an interrupt signal at the rise of T4 state during the execution of this NOP instruction. If the interrupt signal is accepted, CPU executes the interrupt process operation from the next cycle.

During interrupt signal input, it is necessary to take the same care as the interrupt signal input in IDLE1/2 Mode.

Halt release in STOP Mode (HALTM=10) by /RESET. When /RESET at "0" level is input into the IPC, the internal oscillator is restarted. However, the internal clock counter for warm-up does not operate. Therefore, the operation is not carried out properly due to unstable clock oscillation. It is necessary to hold /RESET at "0" level for sufficient time. The halt release operation by the IPC resetting in STOP Mode is shown in Figure 40.

Z84C13/C15 Only. The /RESET pulse is stretched to a minimum of 16 cycles and driven out of the Z84C13/C15 on the /RESET pin if Reset output is enabled (bit D3 of MCR is cleared to "0"). Setting bit D3 disables the driving out of /RESET. The values in the control registers (WDTMR, SCRP, WCR, MWBR, CSBR and MCR) are initialized to the default value on /RESET.

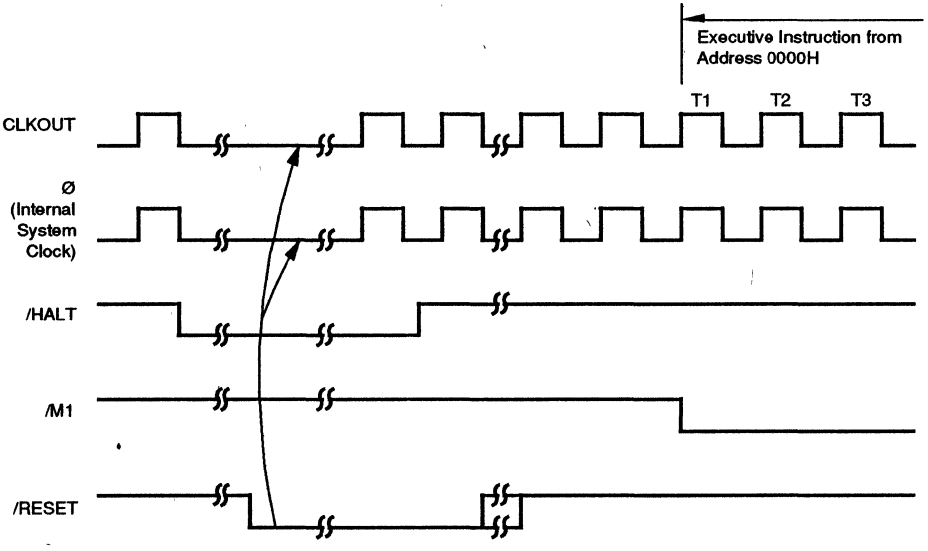


Figure 40. Halt Release Operation Timing By Reset in STOP Mode

Start-up Time at Time of Restart (STOP Mode). When the MPU is released from the halt state by accepting an interrupt request, it executes an interrupt service routine Therefore, when an interrupt request is accepted, it starts generating clock on the CLKOUT pin, after a start-up time, by the internal counter [(2^{14}+2 5) TcC (TcC·Clock Cycle)]. This obtains a stabilized oscillation for operation.

Further, in case of restart by the /RESET signal, the internal counter does not operate.

Evaluation operation. Each of the CPU signals (A15-0, D7-0, /MREQ, /IORQ, /RD, /WR, /HALT, /M1, /RFSH) can be 3-stated by activating the EV pin. The Z84C13/C15 enhances the counter part by eliminating the requirement of /BUSREQ to go active

Instruction set. The instruction set of the IPC is the same for the Z84C00. For details, refer to the data sheet of the Z84C00 Technical Manual.

AC TIMING

The following section describes the timing of the IPC The numbers appearing in the figures refer to the parameters on Table A - F.

CPU Timing
Parameters referenced in Figure 41 through Figure 48 appear in Table A.

The IPC's CPU executes instructions by proceeding through the following specific sequence of operations.

Memory read or write
I/O device read or write
Interrupt acknowledge

The basic clock period is referred to as a Time or Cycle and three or more T cycles make up a machine cycle (e.g , M1, M2 or M3). Machine cycles are extended either by the CPU automatically inserting one or more Wait states or by the insertion of one or more Wait states by the user.

Instruction Op-code Fetch. The CPU places the contents of the Program Counter (PC) on the address bus at the start of the cycle (Figure 41). Approximately one-half clock cycle later, /MREQ goes active. When active, /RD indicates that the memory data can be enabled onto the CPU data bus.

The CPU samples the /WAIT input with the falling edge of clock state T2. During clock states T3 and T4 of an M1 cycle, dynamic RAM refresh can occur while the CPU starts decoding and executing the instruction.

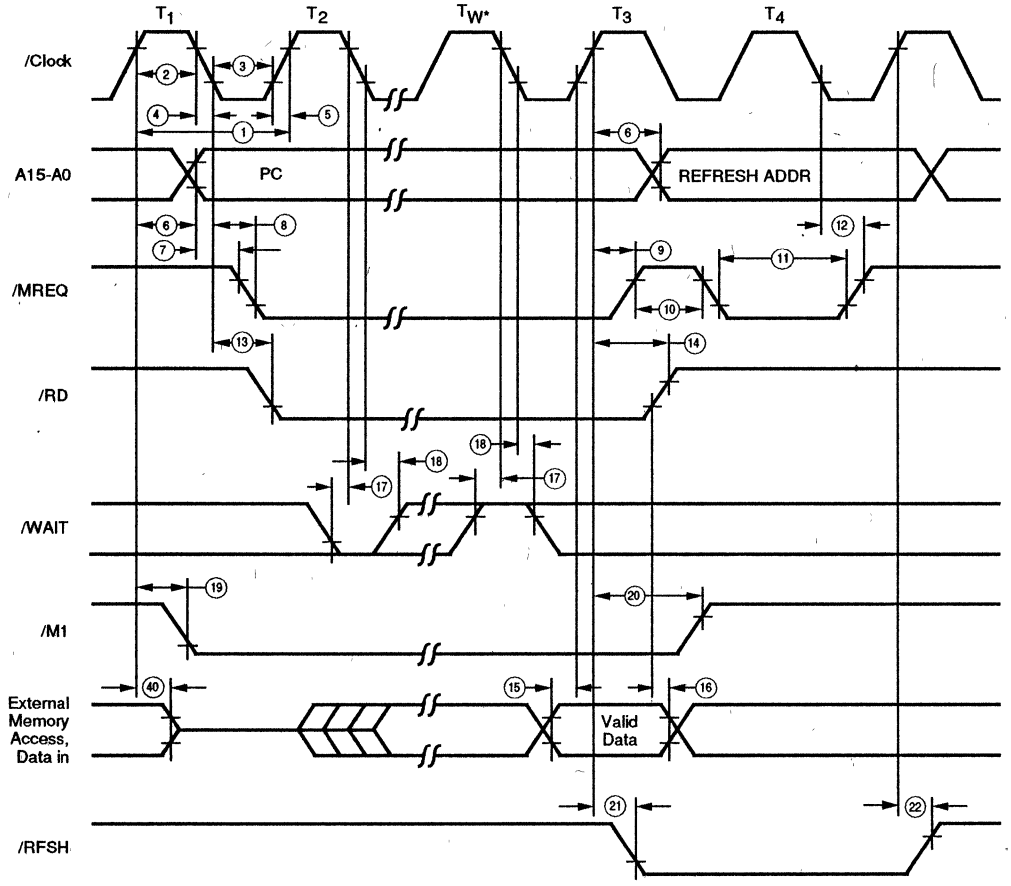


Figure 41. Instruction Op-code Fetch
(See Table A)

Memory Read or Write Cycles. Figure 42 shows the timing of memory read or write cycles other than an Op-code fetch (/M1) cycle. The /MREQ and /RD signals function like the Op-code fetch cycle.

In a memory write cycle, /MREQ also becomes active when the Address Bus is stable. The /WR line is active when the Data Bus is stable, so that it can be used directly as an R/W pulse to most semiconductor memories.

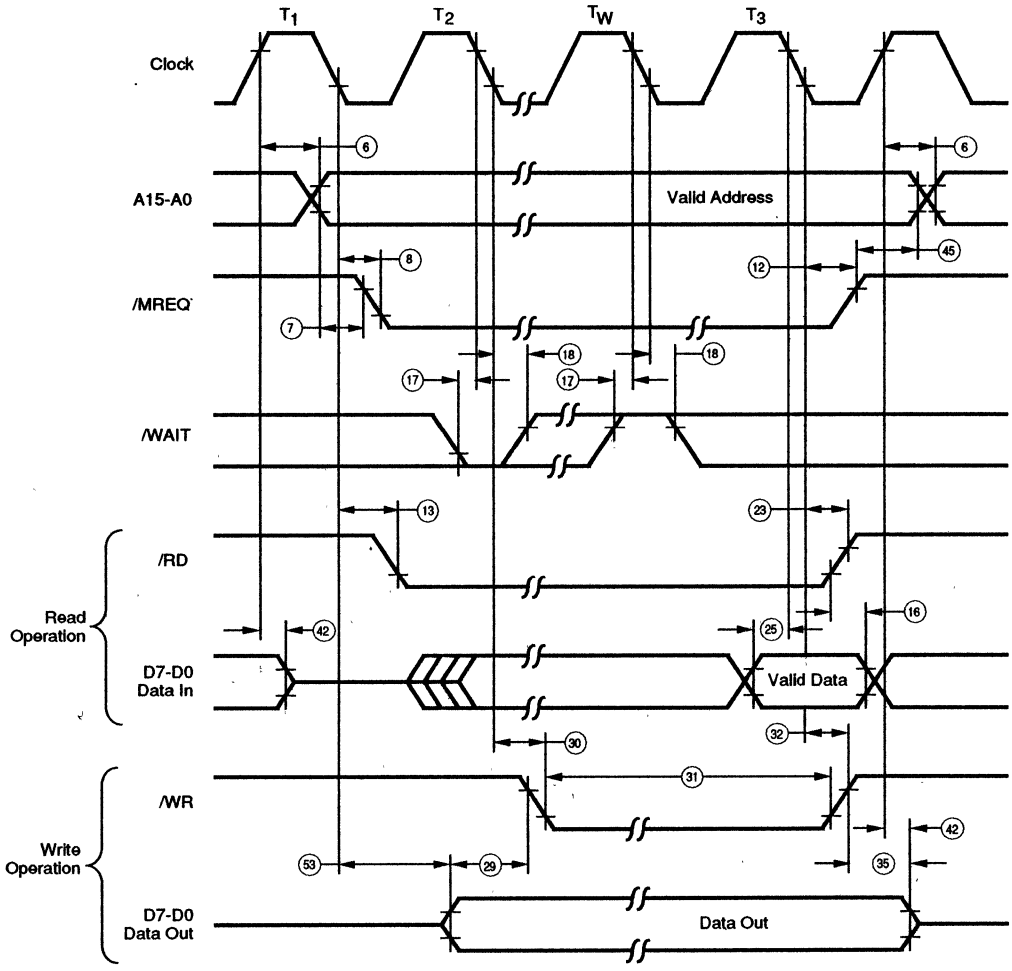
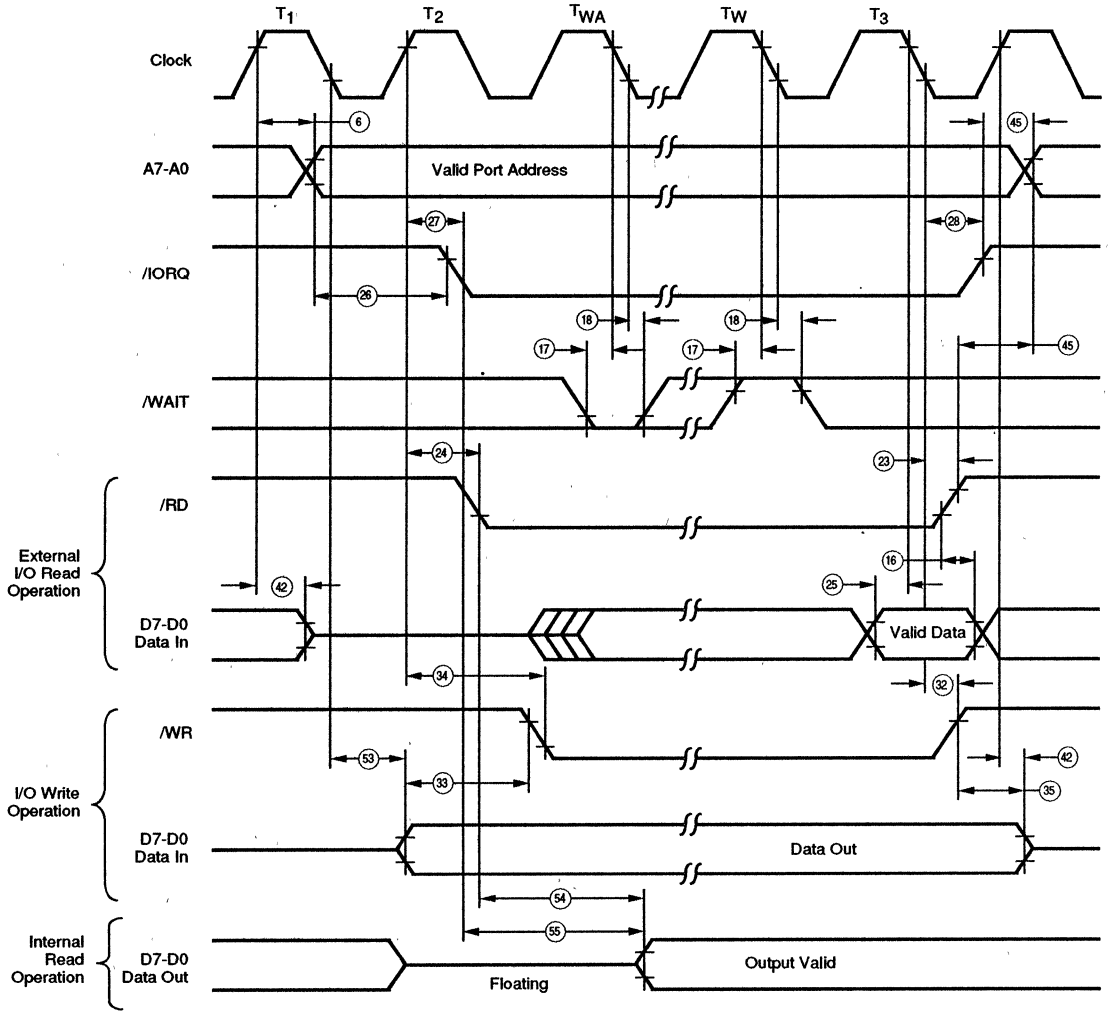


Figure 42. Memory Read or Write Cycle
(See Table A)

Input or Output Cycles. Figure 43 shows the timing for an I/O read or I/O write operation During I/O operations, the CPU automatically inserts a single Wait state (T_{WA}). This extra Wait state allows sufficient time for an I/O port to decode the address from the port address lines.

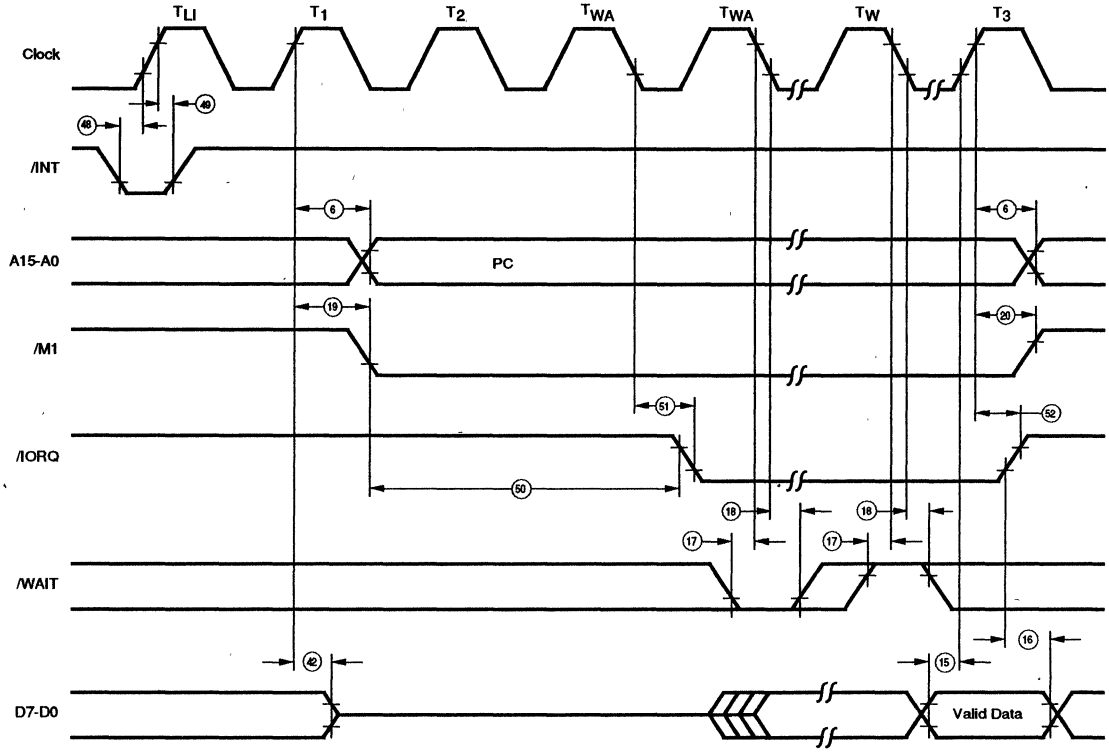
When the CPU is accessing the on-chip I/O registers (PIO, CTC, SIO and system control registers), the data from/to these registers also appears on the data bus, or data bus is output during I/O cycle.



Note: T_{WA} = One wait cycle automatically inserted by CPU

Figure 43. Input or Output Cycle
(See Table A)

Interrupt Request/Acknowledge Cycle. The CPU samples the interrupt signal with the rising edge of the last clock cycle at the end of any instruction (Figure 44). When an interrupt is accepted, a special /M1 cycle is generated. During this /M1 cycle, /IORQ becomes active (instead of /MREQ) to indicate that the interrupting device can place an 8-bit vector on the data bus. The CPU automatically adds two Wait states to this cycle.

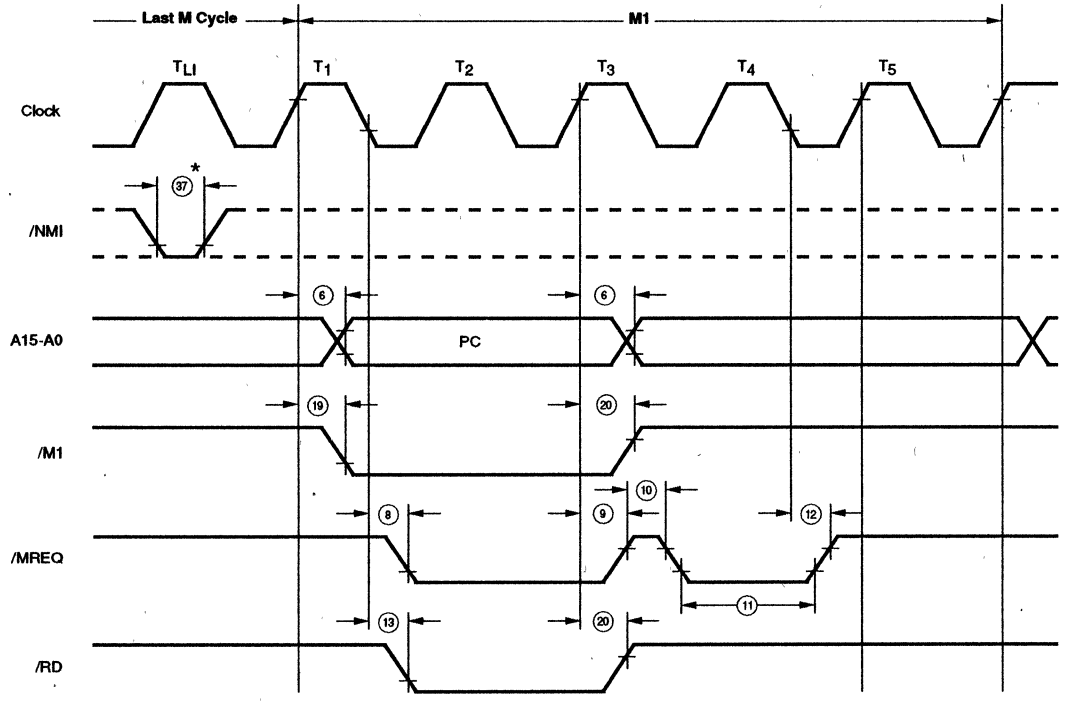


NOTE: 1) T_{LI} = Last state of any instruction cycle

2) T_{WA} = Wait cycle automatically inserted by CPU

Figure 44. Interrupt Request/Acknowledge Cycle
(See Table A)

Non-Maskable Interrupt Request Cycle. /NMI is sampled at the same time as the maskable interrupt input /INT, but has higher priority and cannot be disabled under software control. The subsequent timing is similar to that of a normal memory read operation except that data put on the bus by the memory is ignored. The CPU instead executes a restart (RST) operation and jumps to the /NMI service routine located at the address 0066H (Figure 45).



* Although /NMI is an asynchronous input, to guarantee its being recognized on the following machine cycle, /NMI's falling edge must occur no later than the rising edge of the clock cycle preceding the last state of any instruction cycle (T_{LI}).

Figure 45. Non-Maskable Interrupt Request Operation
(See Table A)

Bus Request/Acknowledge Cycle. The CPU samples /BUSREQ with the rising edge of the last clock period of any machine cycle (Figure 46) If /BUSREQ is active, the CPU sets its address, data, and /MREQ to Inputs, and /IORQ, /RD and /WR lines set to an input for on-chip peripheral access from an external bus master with the rising edge of the next clock pulse. At that time, any external device can take control of these lines, usually to transfer data between memory and I/O devices.

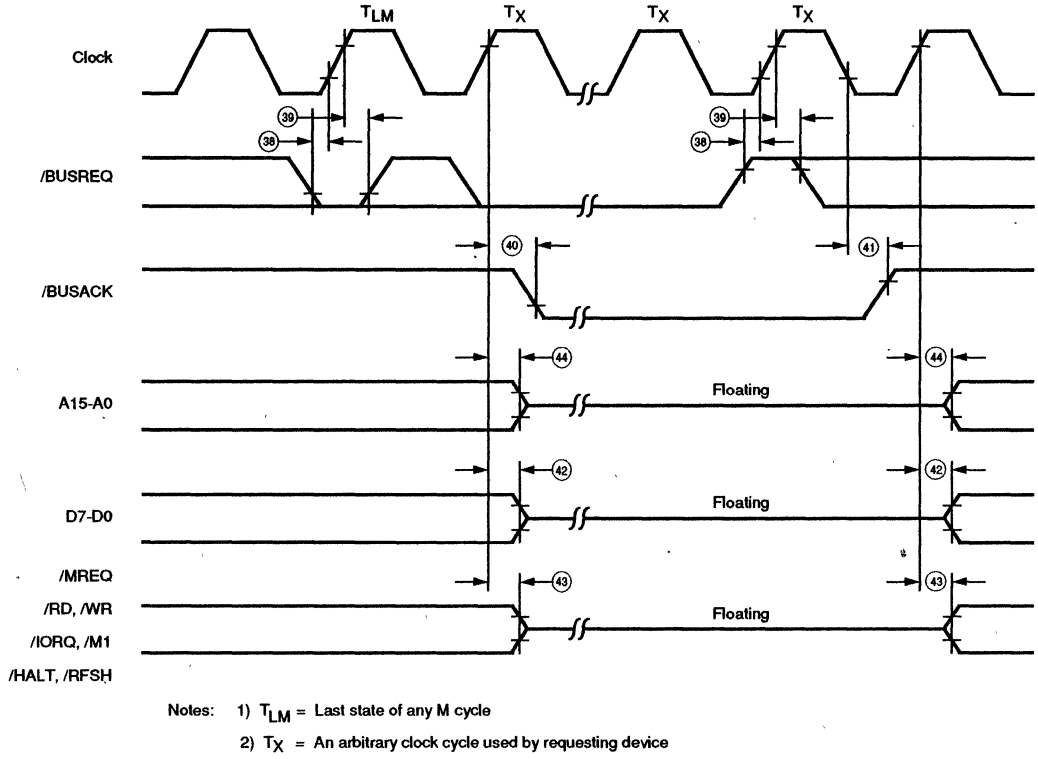
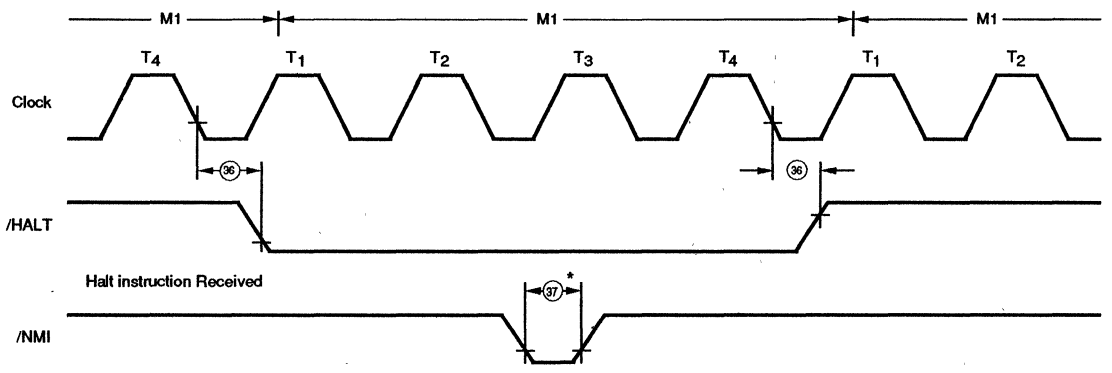


Figure 46. BUS Request/Acknowledge Cycle
(See Table A)

Halt acknowledge cycle. Figure 47 shows the timing for Halt acknowledge cycle.



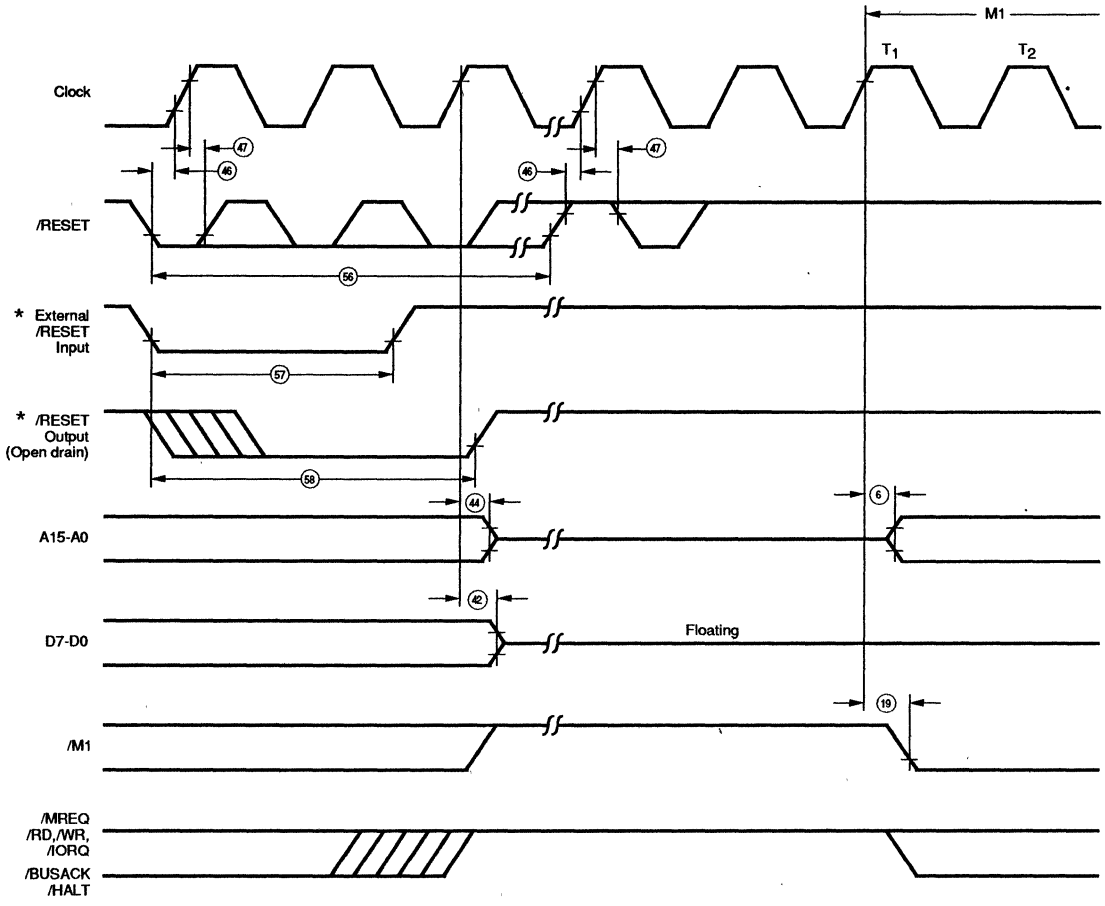
* Although /NMI is an asynchronous input, to guarantee its being recognized on the following machine cycle, /NMI's falling edge must occur no later than the rising edge of the clock preceding the last state of any instruction cycle (T_{LI}).

Figure 47. Halt Acknowledge
(See Table A)

Reset Cycle. /RESET must be active for at least three clock cycles for the CPU to properly accept it. As long as /RESET remains active, the address and data buses float, and the control outputs are inactive.

Once /RESET goes inactive, two internal T cycles are consumed before the CPU resumes normal processing operation. /RESET clears the PC register, so the first op-code fetch location is 0000H (Figure 48).

Z84C13/C15 Only. If Reset output is disabled, /RESET must be active for at least three clock cycles for the CPU to properly accept it. Otherwise, /RESET must be active for at least two clock cycles and the on-chip reset circuit extends /RESET signal to at least a minimum of 16-clock cycles.



* 84C13/15 Only Reset Output is Enabled

Figure 48. Reset Cycle
(See Table A)

CGC TIMING

Figure 49 to Figure 52 shows the timing related CGC and Power-On Reset circuit.

Parameters referenced in Figure 49 thru Figure 52 appear in Table B.

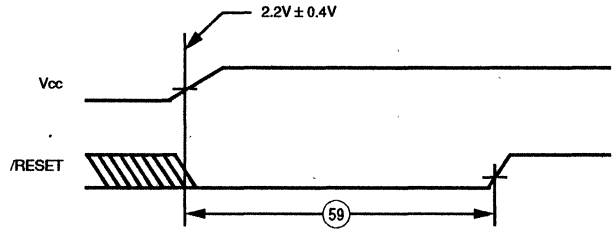


Figure 49. Reset on Power-up (Applies only for Z84C13/C15)
(See Table B)

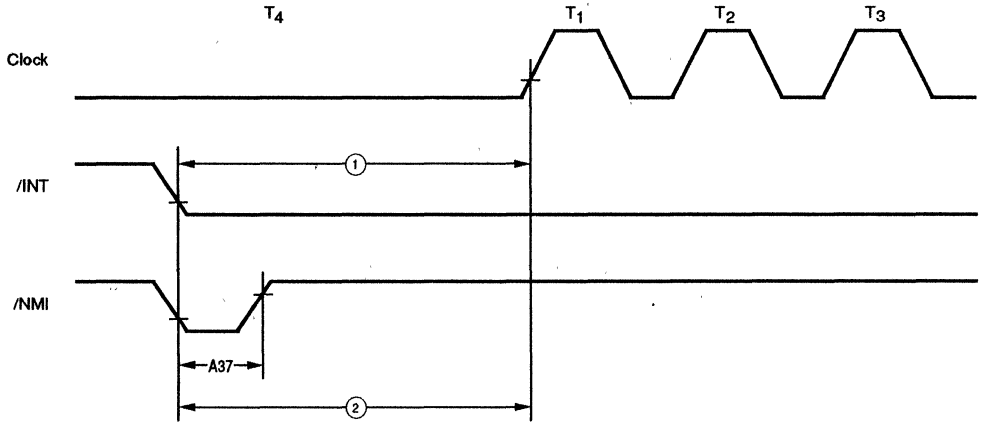
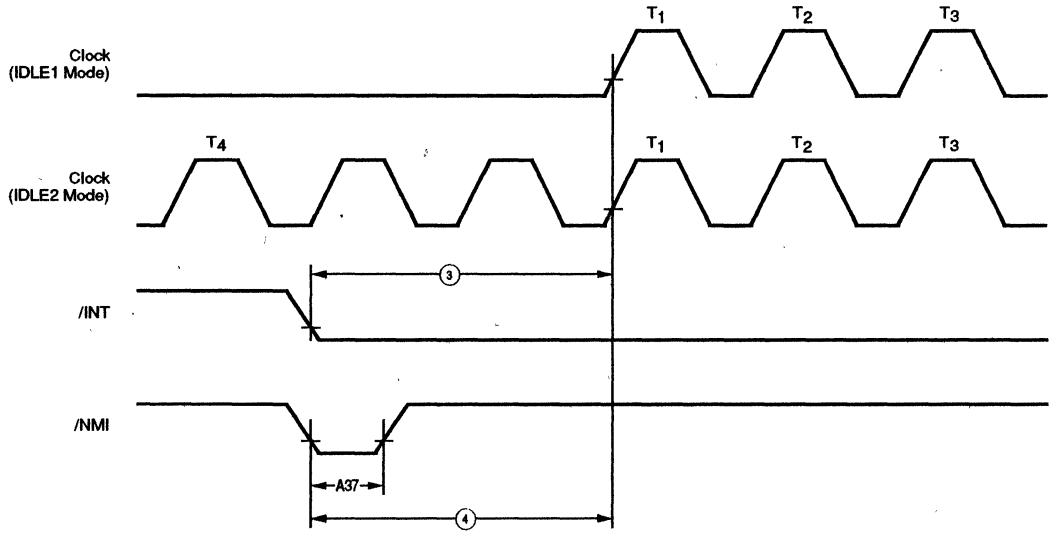
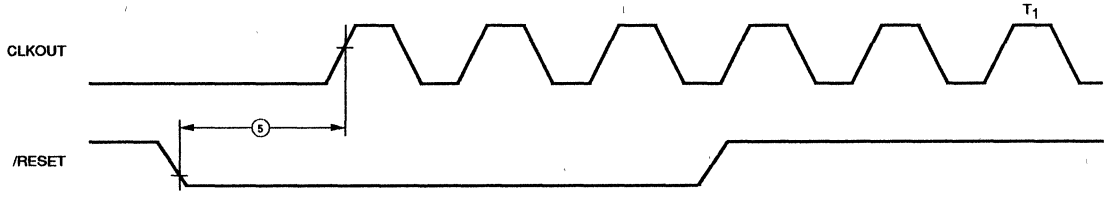


Figure 50. Clock Restart Timing by /INT, /NMI (STOP Mode)
(See Table B)

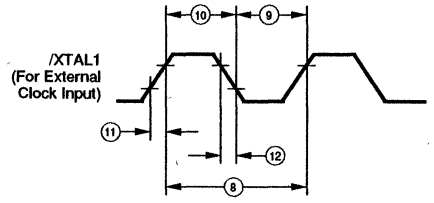


(a) Clock Restart Timing by /INT, /NMI (IDLE1/2 Mode)

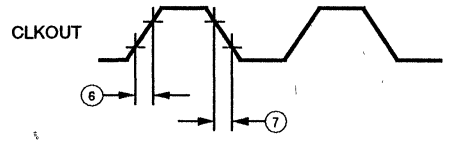


(b) Clock Restart Timing by /RESET (IDLE1/2 Mode)

Figure 51. Clock Restart Timing (IDLE1/2 Mode)
(See Table B)



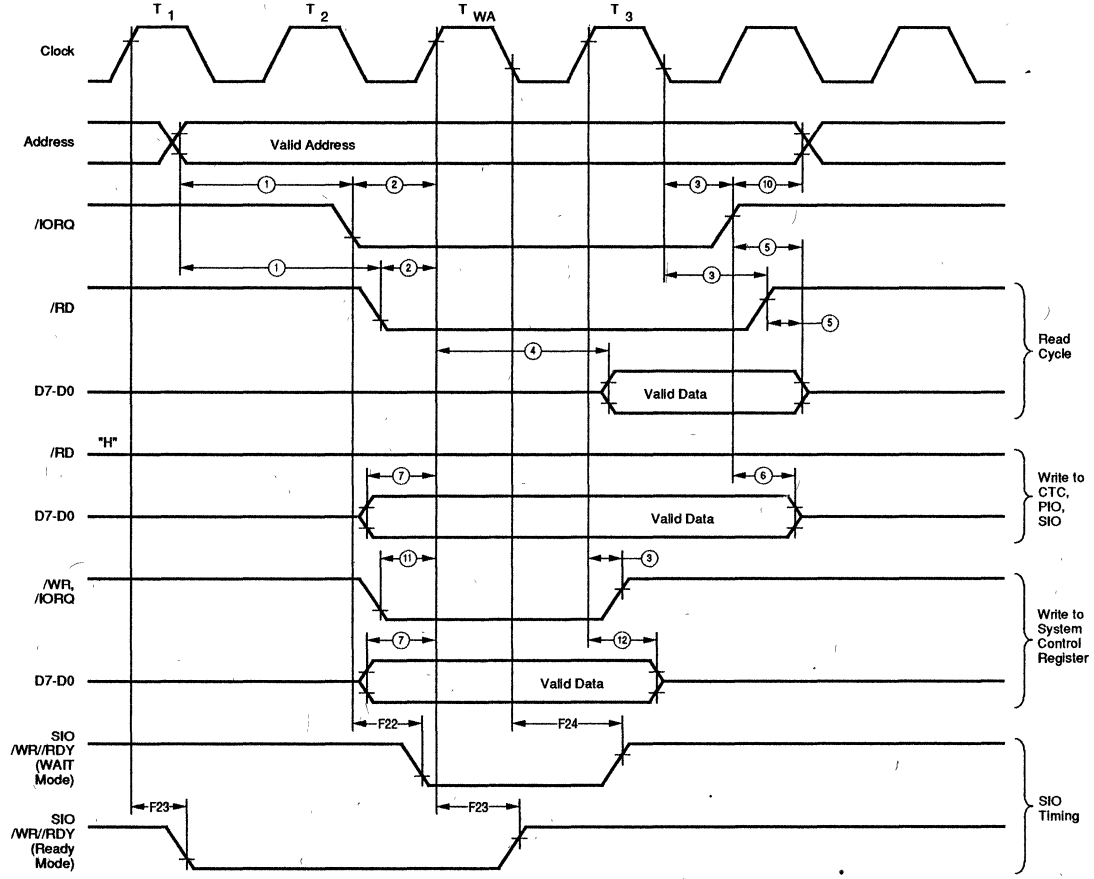
(a) XTAL1 Timing for External Clock Input



(b) CLKOUT Timing

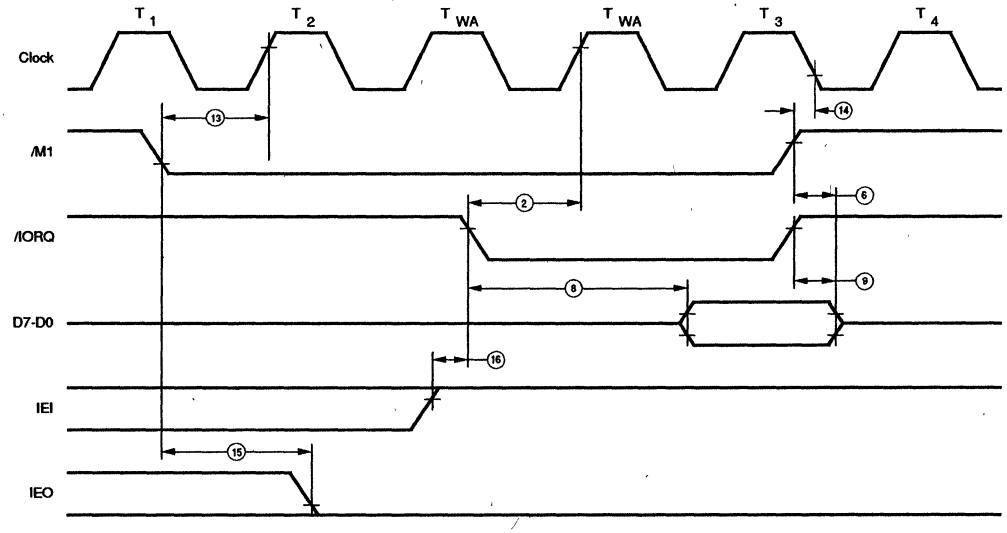
Figure 52. Clock Timing
(See Table B)

On-chip peripheral access from External Bus master. The timing for the on-chip I/O device access from the external bus master is shown in Figure 53. This timing also applies to the timing during EV mode of operation.

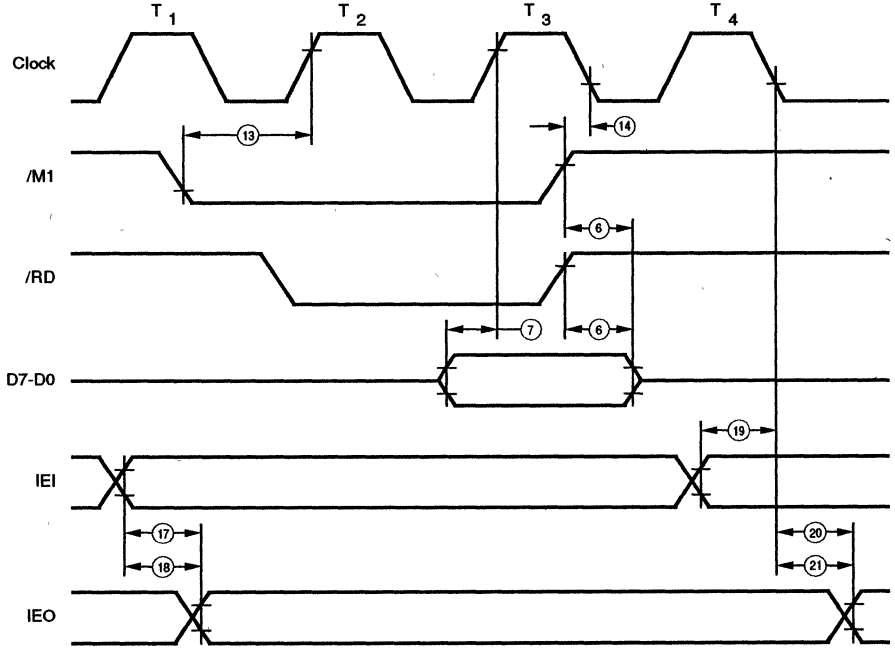


(a) On-chip peripheral I/O access from External Bus master
(See Tables C and F)

Figure 53. On-chip Peripheral Timing from External Bus master



(b) Interrupt Acknowledge Cycle Timing for On-chip peripheral from External Bus master
(See Table C)



(c) Op-code fetch Cycle Timing for On-chip peripheral from External Bus master
(See Table C)

Figure 53. On-chip Peripheral Timing from External Bus master (Continued)

PIO timing
(Not applicable on Z84x13) Figure 54 shows the timing for on-chip PIO.

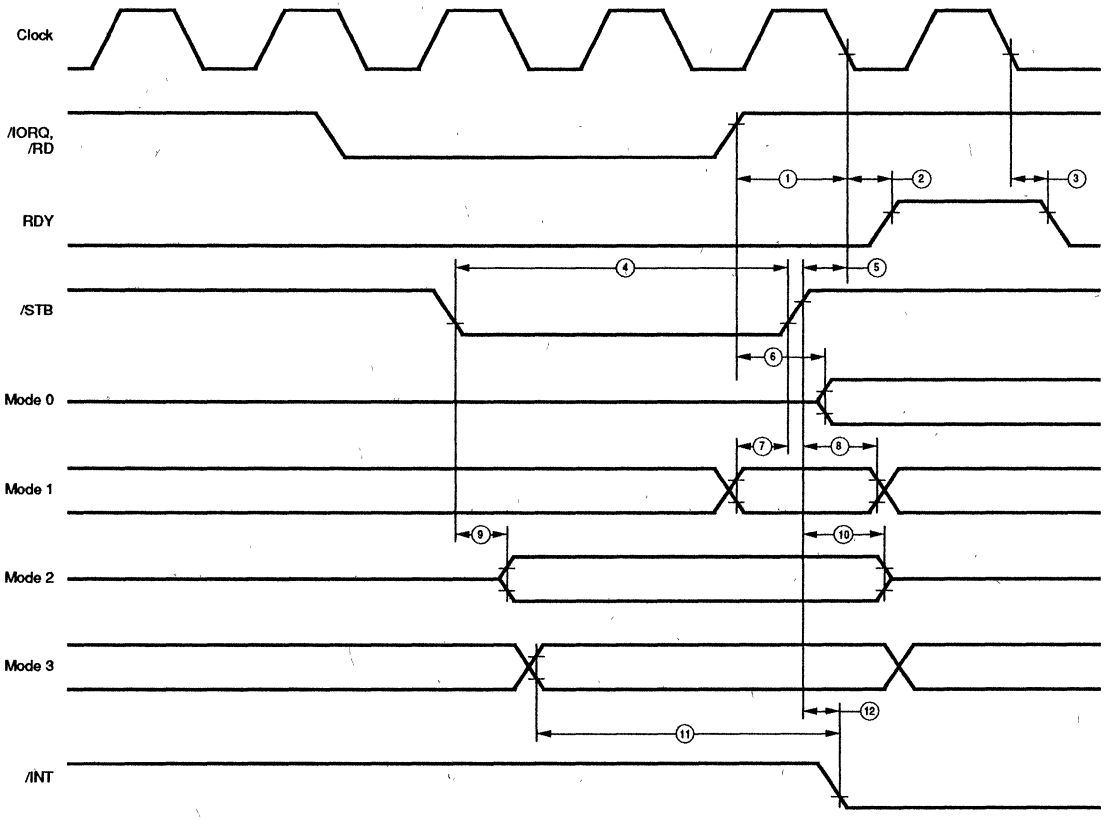


Figure 54. PIO Timing
(See Table D)

CTC Timing

Figure 55 shows the timing for on-chip CTC.

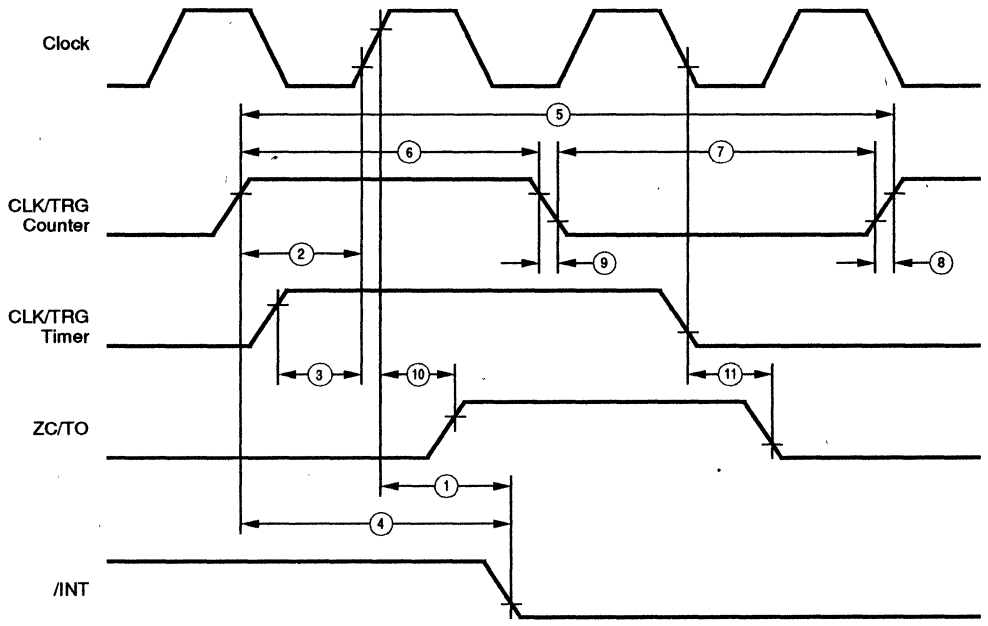


Figure 55. Counter/Timer Timing
(See Table E)

SIO Timing

Figure 56 shows the timing for on-chip SIO.

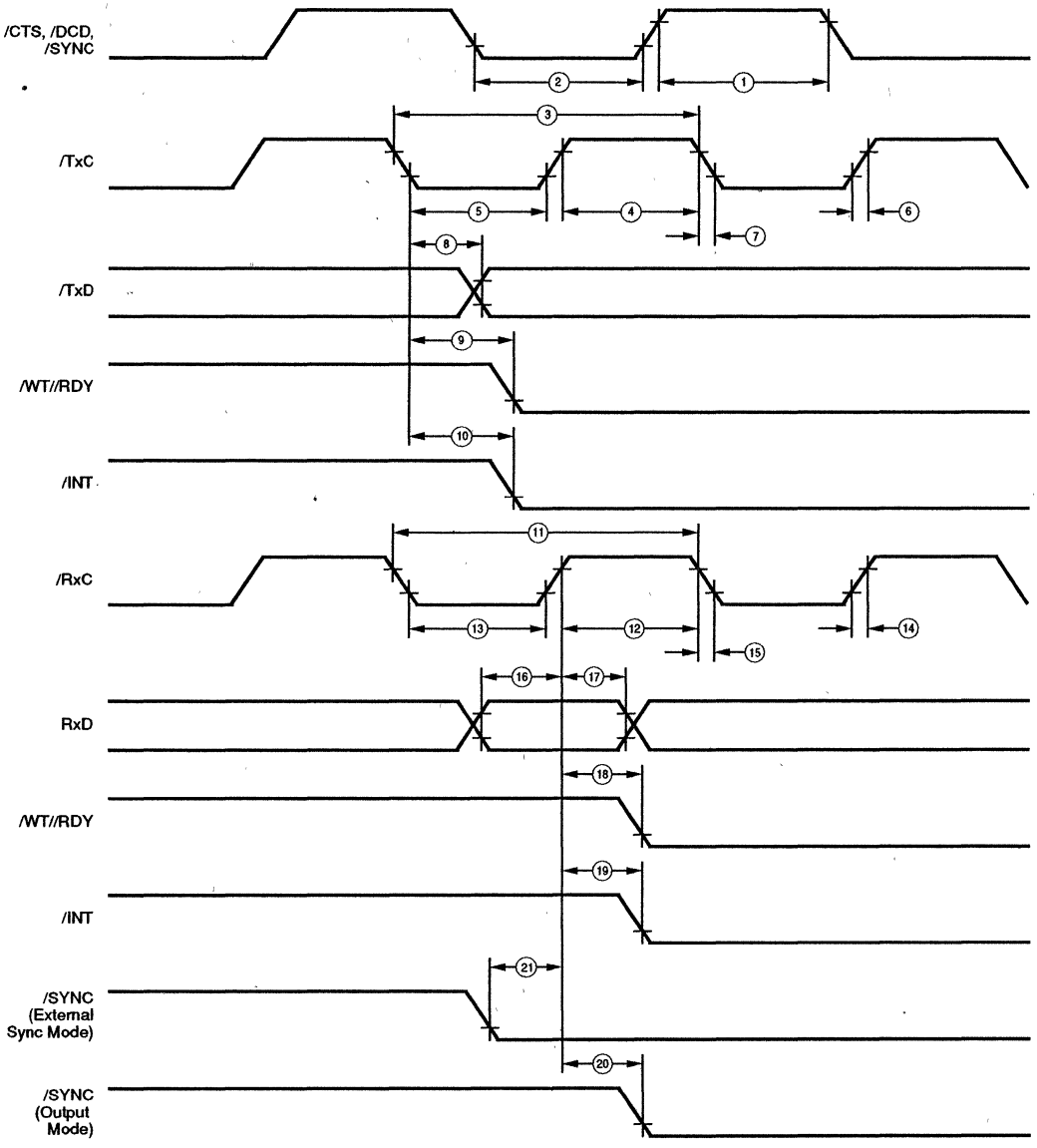


Figure 56. SIO Timing
(See Table F)

Watch-Dog Timer Timing
Figure 57 shows the timing for Watch-dog Timer.

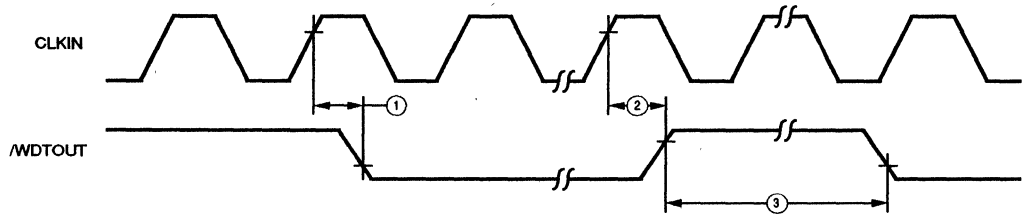


Figure 57. Watch-dog Timer Timing (See Table H)

PRECAUTIONS

(1) To release the HALT state by /RESET signal in STOP Mode, hold the /RESET signal at "0" until the output from the internal oscillator stabilizes.

Z84013/015 Only. To reset MPU, it is necessary to hold /RESET signal input at "0" level for at least three clocks.

Z84C13/C15 Only. If Reset output is disabled, /RESET must be active for at least three clock cycles for the CPU to properly accept it. Otherwise, the on-chip reset circuit extends /RESET signal to at least a minimum of 16-clock cycles.

(2) Releasing the MPU from the HALT state by the interrupt signal in IDLE1/2 Mode and STOP Mode, depends upon the HALT state and the internal system clock. They will stop unless an interrupt signal is accepted during the execution of NOP instruction, even when the internal system clock is restarted by the interrupt signal input. In particular, care must be taken when /INT is used.

Other precautions are identical to those for the Z84C00. Refer to the data sheet for the Z84C00.

ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

Voltage on Vcc with respect to Vss .....-0.3V to +7.0V
Voltages on all inputs
with respect to Vss .....-0.3V to Vcc+0.3V
Operating Ambient
Temperature .....See Ordering Information
Storage Temperature .....-65 °C to + 150 °C

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 sections below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND (0V). Positive current flows into the referenced pin.

Available operating temperature range is:
E = -40°C to 100°C
Voltage Supply Range:
+4.50V ≤ Vcc ≤ + 5.50V

All AC parameters assume a load capacitance of 100 pf. Add 10 ns delay for each 50 pf increase in load up to a maximum of 150 pf for the data bus and 100 pf for address and control lines. AC timing measurements are referenced to 1.5 volts (except for clock, which is referenced to the 10% and 90% points). Maximum capacitive load for CLK is 125 pf.

The Ordering Information section lists temperature ranges and product numbers. Package drawings are in the Package Information section. Refer to the Literature List for additional documentation.

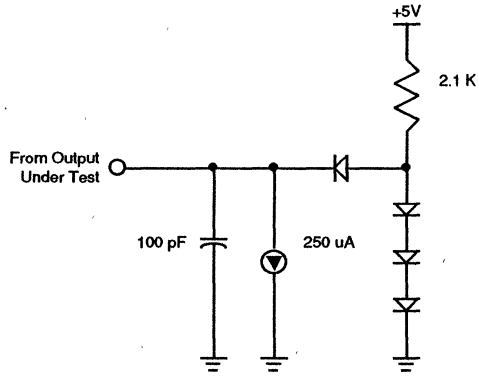


Figure 58. Standard Test Load

CAPACITANCE

Guaranteed by design and characterization

Symbol	Parameter	Min	Max	Unit
C_{CLOCK}	Clock Capacitance		35	pF
C_{IN}	Input Capacitance		5	pF
C_{OUT}	Output Capacitance		15	pF

DC CHARACTERISTICS

V_{CC}=5.0V ± 10%, unless otherwise specified

Symbol	Parameter	Min	Max	Unit	Condition
V_{OLC}	Clock Output High Voltage	V_{CC}-0.6		V	-2.0mA
V_{OHC}	Clock Output Low Voltage		0.4	V	+2.0mA
V_{IHC}	Clock Input High Voltage	V_{CC}-0.6		V	
V_{ILC}	Clock Input Low Voltage		0.4	V	
V_{IH}	Input High Voltage	2.2	V_{CC}	V	
V_{IL}	Input Low Voltage	-0.3	0.8	V	
V_{OL}	Output Low Voltage		0.4 [5]	V	I_{LO}=2.0mA
V_{OH1}	Output High Voltage	2.4		V	I_{OH}=-1.6mA
V_{OH2}	Output High Voltage	V_{CC}-0.8 [5]		V	I_{OH}=-250µA
I_{CC1}	Power Supply Current				V_{CC}=5V
	XTALIN =10MHz		50	mA	V_{IH}=V_{CC}-0.2V
	XTALIN = 6MHz		30	mA	V_{IL}=0 2V
I_{CC2}	Power Supply Current (STOP Mode)		50	µA	V_{CC}=5V
I_{CC3}	Power Supply Current (IDLE1 Mode)				V_{CC}=5V
	XTALIN =10MHz		6	mA	V_{IH}=V_{CC}-0 2V
	XTALIN = 6MHz		4	mA	V_{IL}=0.2V
I_{CC4}	Power Supply Current (IDLE2 Mode)				V_{CC}=5V
	XTALIN =10MHz		TBD [1]	mA	V_{IH}=V_{CC}-0.2V
	XTALIN = 6MHz		TBD [1]	mA	V_{IL}=0.2V
I_{LI}	Input Leakage Current	-10	10 [4]	µA	V_{IN}=0 4V to V_{CC}
$I_{L(SY)}$	SYNC pin Leakage Current	-40	10	µA	V_{OUT}=0.4V to V_{CC}
I_{LO}	3-state Output Leakage Current in Float	-10	10 [2]	µA	V_{OUT}=0 4V to V_{CC}
I_{OHD}	Darlington Drive Current				V_{OH}=1.5V
	(Port B and CTC ZC/TO)	-1.5		mA	REXT = 390 Ohms

Notes:

[1] Measurements made with outputs floating.

[2] A15-A0, D7-D0, /MREQ, /IORQ, /RD and /WR

[3] I_{CC2} Standby Current is guaranteed when the /HALT pin is low in STOP mode.

[4] All Pins except XTALI, where I_{LI}=±25µA.

[5] A15-A0, D7-D0, /MREQ, /IORQ, /RD, /WR, /HALT, /M1 and /BUSACK.

AC CHARACTERISTICS

Table A. CPU Timing (See Figure 41 to 48)

No	Symbol	Parameter	Z84x1306 Z84x1506 Min	Max	Z84x1310 Z84x1510 Min	Max	Unit	Note
1	TcC	Clock Cycle time	162**	DC	100**	DC	nS	[A1]
2	TwCh	Clock pulse width (high)	65	DC	40	DC	nS	[A1]
3	TwCl	Clock pulse width (low)	65	DC	40	DC	nS	[A1]
4	TfC	Clock Fall time		20		10	nS	[A1]
5	TrC	Clock Rise time		20		10	nS	[A1]
6	TdCr(A)	Address valid from Clock Rise		90		65	nS	
7	TdA(MREQf)	Address valid to /MREQ Fall	35**		0**		nS	
8	TdCf(MREQf)	Clock Fall to /MREQ Fall delay		70		55	nS	
9	TdCr(MREQr)	Clock Rise to /MREQ Rise delay		70		55	nS	
10	TwMREQh	/MREQ pulse width (High)	65**		30**		nS	[A2]
11	TwMREQl	/MREQ pulse width (low)	132**		75**		nS	[A2]
12	TdCf(MERQr)	Clock Fall to /MREQ Rise delay		70		55	nS	
13	TdCf(RDf)	Clock Fall to /RD Fall delay		80		65	nS	
14	TdCr(RDr)	Clock Rise to /RD Rise delay		70		55	nS	
15	TsD(Cr)	Data setup time to clock Rise	30		25		nS	
16	ThD(RDr)	Data hold time after /RD Rise	0		0		nS	
17	TsWAIT(Cf)	/WAIT setup time to Clock Fall	60		20		nS	
18	ThWAIT(Cf)	/WAIT hold time after Clock Fall	10		10		nS	
19	TdCr(M1f)	Clock Rise to /M1 Fall delay		80		65	nS	
20	TdCr(M1r)	Clock Rise to /M1 Rise delay		80		65	nS	
21	TdCr(RFSHf)	Clock Rise to /RFSH Fall delay		110		80	nS	
22	TdCr(RFSHr)	Clock Rise to /RFSH Rise delay		100		80	nS	
23	TdCf(RDr)	Clock Fall to /RD Rise delay		70		55	nS	
24	TdCr(RDf)	Clock Rise to /RD Fall delay		70		55	nS	
25	TsD(Cf)	Data setup to Clock Fall during M2, M3, M4 or M5 cycles	40		25		nS	
26	TdA(IORQf)	Address stable prior to /IORQ Fall	107**		50**		nS	
27	TdCr(IORQf)	Clock Rise to /IORQ Fall delay		65		50	nS	
28	TdCf(IORQr)	Clock Fall to /IORQ Rise delay		70		55	nS	
29	TdD(WRf)Mw	Data stable prior to /WR Fall	22**		40**		nS	
30	TdCf(WRf)	Clock Fall to /WR Fall delay		70		55	nS	
31	TwWR	/WR pulse width	132**		75**		nS	
32	TdCf(WRr)	Clock Fall to /WR Rise delay		70		55	nS	
33	TdD(WRf)IO	Data stable prior to /WR Fall	-55**		-10**		nS	
34	TdCr(WRf)	Clock Rise to /WR Fall delay		60		50	nS	
35	TdWRr(D)	Data stable from /WR Fall	30**		10**		nS	
36	TdCf(HALT)	Clock Fall to /HALT "0" or "1"		260		90	nS	
37	TwNMI	/MNI pulse width	60		60		nS	
38	TsBUSREQ(Cr)	/BUSREQ setup time to Clock Rise	50		30		nS	
39	ThBUSREQ(Cr)	/BUSREQ hold time after Clock Rise	10		10		nS	
40	TdCr(BUSACKf)	Clock Rise to /BASACK Fall delay			90			75 nS

Table A. CPU Timing (Continued)

No	Symbol	Parameter	Z84x1306 Z84x1506 Min	Z84x1306 Z84x1506 Max	Z84x1310 Z84x1510 Min	Z84x1310 Z84x1510 Max	Unit	Note
41	TdCf(BUSACKr)	Clock Fall to /BASACK Rise delay		90		75	nS	
42	TdCr(Dz)	Clock Rise to Data Float delay		80		65	nS	
43	TdCr(CTz)	Clock Rise to Control Outputs Float Delay (/MREQ, /IORQ, /RD and /WR)		70		65	nS	
44	TdCr(Az)	Clock Rise to Address Float delay		80		75	nS	
45	TdCTr(A)	Address Hold time from /MREQ, /IORQ, /RD or /WR	35**		20**		nS	
46	TsRESET(Cr)	/RESET to Clock Rise setup time	60		40		nS	
47	ThRESET(Cr)	/RESET to Clock Rise Hold time	10		10		nS	
48	TsINTf(Cr)	/INT Fall to Clock Rise Setup Time	70		50		nS	
49	ThINTr(Cr)	/INT Rise to Clock Rise Hold Time	10		10		nS	
50	TdM1f(IORQf)	/M1 Fall to /IORQ Fall delay	359**		220**		nS	
51	TdCf(IORQf)	Clock Fall to /IORQ Fall delay		70		55	nS	
52	TdCf(IORQr)	Clock Rise to /IORQ Rise delay		70		55	nS	
53	TdCf(D)	Clock Fall to Data Valid delay		130		110	nS	
54	TRDf(D)	/RD Fall to Output Data Valid		TBD		60	nS	
55	TMI(D)	/IORQ Fall to output data valid		TBD		70	nS	
56	TwRESET	/RESET Pulse Width 013/015, or C13/C15 with RESET output disabled	3TcC		3TcC		nS	[A3]
57	TwRESEToe	/RESET Pulse Width C13/C15 Only, RESET output Enabled	2TcC		2TcC		nS	[A3]
58	TwRESETdo	/RESET drive duration C13/C15 Only; RESET output Enabled	16TcC		16TcC		nS	[A3]
59	TwRESETpor	/RESET drive duration on Power-on sequence C13/C15 only	25	75	25	75	mS	[A3]

Notes:

** For clock period other than the minimum shown, calculate parameters using the formula on Table H.

[A1] These parameters apply to the external Clock input on CLKIN pin For the cases where external Clock is fed from XTAL1, please refer to Table B.
[A2] For loading >= 50pF, decrease width by 10nS for each additional 50pF.
[A3] Apply to Z84C13/C15 only

Table H. Footnote to Table A.

No	Symbol	Parameter	Z84x1306 Z84x1506	Z84x1310 Z84x1510
1	TcC	TwCh + TwCl + TrC + TfC		
7	TdA(MREQf)	TwCh + TfC	-50	-50
10	TwMREQh	TwCh + TfC	-20	-20
11	TwMREQl	TcC	-30	-25
26	TdA(IORQf)	TcC	-55	-50
29	TdD(WRf)	TcC	-140	-60
31	TwWR	TcC	-30	-25
33	TdD(WRf)	TwCl + TrC	-140	-60
35	TdWRr(D)	TwCl + TrC	-55	-40
45	TdCTr(A)	TwCl + TrC	-50	-30
50	TdM1f(IORQf)	2TcC + TwCh + TfC	-50	-30

Table B. CGC Timing (See Figure 49 to 52)

No	Symbol	Parameter	Z84x1306 Z84x1506 Min	Z84x1306 Z84x1506 Max	Z84x1310 Z84x1510 Min	Z84x1310 Z84x1510 Max	Unit	Note
1	TRST(INT)S (STOP Mode)	Clock Restart Time by /INT	(Typ)2^{14}+2.5TcC		(Typ)2^{14}+2.5TcC		nS	
2	TRST(MNI)S (STOP Mode)	Clock Restart Time by /NMI	(Typ)2^{14}+2.5TcC		(Typ)2^{14}+2.5TcC		nS	
3	TRST(INT)I	Clock Restart Time by /INT (IDLE Mode)	2.5TcT		2.5TcT		nS	
4	TRST(MNI)I	Clock Restart Time by /NMI (IDLE Mode)	2.5TcT		2.5TcT		nS	
5	TRST(RESET)I	Clock Restart Time by /RESET (IDLE Mode)	1TcC		1TcC		nS	
6	TfCLKOUT	CLKOUT Rise Time		15		10	nS	
7	TrCLKOUT	CLKOUT Fall time		15		10	nS	
8	TcX1	XTAL1 Cycle Time (for External Clock input on XTAL1)						
		Divide-by-two mode	81		50		nS	
		Divide-by-one mode	162		100		nS	[B2]
9	TwlX1	XTAL1 Low pulse width (for External Clock input on XTAL1)						
		Divide-by-two mode	35		20		nS	
		Divide-by-one mode (C13/15 only)	65		40		nS	[B2]
10	TwhX1	XTAL1 High Pulse Width (for External Clock input on XTAL1)						
		Divide-by-two mode	35		20		nS	
		Divide-by-one mode (C13/15 only)	65		40		nS	
11	TrX1	XTAL1 Rise Time (for External Clock input on XTAL1)		25		25	nS	[B1]
12	TfX1	XTAL1 Fall Time (for External Clock input on XTAL1)		25		25	nS	[B1]

Notes:
[B1] If parameters 8 and 9 are not met, adjust parameter 11 and 12 to satisfy parameter 8 and 9.
[B2] Does not apply to Z84013/015.

Table C. Timing for on-chip peripheral access from external bus master and daisy chain timing (See Figure 53)

No	Symbol	Parameter	Z84x1306 Z84x1506 Min	Z84x1306 Z84x1506 Max	Z84x1310 Z84x1510 Min	Z84x1310 Z84x1510 Max	Unit	Note
1	TsA(Rlf)	Address Setup Time to /RD, /IORQ Fall	50		40		nS	
2	TsRI(Cr)	/RD, /IORQ Rise to Clock Rise Setup	60		50		nS	
3	Th	Hold time for Specified Setup	15		15		nS	
4	TdCr(DO)	Clock Rise to Data out delay		100		80	nS	
5	TdRIr(DOz)	/RD, /IORQ Rise to Data Out Float Delay		75		60	nS	
6	ThRDr(D)	/M1, /RD, /IORQ Rise to Data Hold	15	40	15	30	nS	[C1]
7	TsD(Cr)	Data In to Clock Rise Setup Time	30		25		nS	
8	TdIOf(DOI)	/IORQ Fall to Data Out Delay (INTACK cycle)		95		95	nS	
9	ThIOr(D)	/IORQ Rise to Data Hold	15		15		nS	
10	ThIOr(A)	/IORQ Rise to Address Hold	15		15		nS	
11	ThWIf(Cr)	/IORQ, /WR setup time to Clock Rise New parameter	20		20		nS	[C2]
12	ThWRr(Cr)	Clock Rise to /IORQ, /WR Rise hold time New parameter	0		0		nS	[C2]
13	TsM1f(Cr)	/M1 Fall to Clock Rise Setup Time	40		40		nS	
14	TsM1r(Cf)	/M1 Rise to Clock Rise Setup Time (/M1 cycle)	-15		-15		nS	
15	TdM1f(IEOf)	/M1 Fall to IEO Fall delay (Interrupt Immediately Preceding /M1 Fall)		140		80	nS	
16	TsIEI(IOf)	IEI to /IORQ Fall Setup Time (INTACK cycle)		160		100	nS	[C3]
17	TdIEIf(IEOf)	IEI Fall to IEO Fall delay	*		*		nS	[C3]
18	TdIEIr(IEOr)	IEI Rise to IEO Rise Delay (After ED decode)		120		70	nS	[C3]
19	TsIEI(Cr)	IEI to Clock Fall Setup (For 4D Decode)		290		150	nS	[C3]
20	TdCf(IEOr)	Clock Fall to IEO Rise Delay	50		40		nS	
21	TdCf(IEOf)	Clock Fall to IEO Rise Delay		90		75	nS	

Notes:
[C1] For I/O write to PIO, CTC and SIO.
[C2] For I/O Write to system control registers.
[C3] For daisy chain timing, please refer to the note on page 63.

Table D. PIO Timing (Z84x15 only) (See Figure 54)

No	Symbol	Parameter	Z84x1506 Min	Z84x1506 Max	Z84x1510 Min	Z84x1510 Max	Unit	Note
1	TsIOr(Cr)	/IORQ Rise to Clock Fall Setup Time (To Activate RDY on Next Clock Cycle)	100		100		nS	
2	TdCf(RDYr)	Clock Fall to RDY Rise delay		100		115	nS	[D2]
3	TdCf(RDYf)	Clock Fall to RDY Fall delay		100		115	nS	[D2]
4	TwSTB	/STB Pulse Width	100		80		nS	[D1]
5	TsSTBr(Cr)	/STB Rise to Clock Fall Setup Time (To Activate RDY on Next Clock Cycle)	100		100		nS	[D2]
6	TdIOr(PD)	/IORQ Rise to Port Data Stable Delay (Mode 0)		140		120	nS	[D2]
7	TsPD(STBr)	Port Data to /STB Rise Setup Time (Mode 1)	140		75		nS	
8	ThPD(STBr)	Port Data to /STB Rise Hold Time (Mode 1)	15		15		nS	
9	TdSTBf(PD)	/STB Fall to Port Data stable (Mode 2)		150		120	nS	[D2]
10	TdSTBr(PDz)	/STB Rise to Port Data Float Delay (Mode 2)		140		120	nS	
11	TdPD(INTf)	Port Data Match to /INT Fall Delay (Mode 3)		250		200	nS	
12	TdSTBr(INTf)	/STB Rise to /INT Fall Delay		290		220	nS	

Notes:
[D1] For Mode 2: TwSTB >TsPD(STB).
[D2] Increase these values by 2nS for 10pF increase in loading up to 100pF Max.

Table E. CTC Timing (Figure 55)

No	Symbol	Parameter	Z84x1306 Z84x1506 Min	Z84x1306 Z84x1506 Max	Z84x1310 Z84x1510 Min	Z84x1310 Z84x1510 Max	Unit	Note
1	TdCr(INTf)	Clock Rise to /INT Fall delay		(TcC+100)		(TcC+80)		[E1]
2	TsCTRr(Cr)c	CLK/TRG Rise to Clock Rise Setup time for Immediate Count	90		90		nS	[E2]
3	TsCTR(Ct)	CLK/TRG Rise to Clock Rise Setup time for enabling of Prescalor on following Clock Rise	90		90		nS	[E1]
4	TdCTRr(INTf)	CLK/TRG Rise to /INT Fall Delay						
		TsCTR(C) satisfied		(36)+(38)		(36)+(38)	nS	[E2]
		TsCTR(C) not satisfied		(1)+(36)+(38)		(1)+(36)+(38)	nS	[E2]
5	TcCTR	CLK/TRG cycle time	(2TcC)	DC	(2TcC)	DC	nS	[E3]
6	TwCTRh	CLK/TRG Width (Low)	90	DC	90	DC	nS	
7	TwCTRl	CLK/TRG Width (High)	90	DC	90	DC	nS	
8	TrCTR	CLK/TRG Rise Time		30		30	nS	
9	TfCTR	CLK/TRG Fall Time		30		30	nS	
10	TdCr(ZCr)	Clock Rise to ZC/TO Rise Delay		80		80	nS	
11	TdCf(ZCf)	Clock Fall to ZC/TO Fall Delay		80		80	nS	

Notes:
[E1] Timer Mode.
[E2] Counter Mode.
[E3] Counter Mode Only; when using a cycle time less than 3TcC, parameter #2 must be met.

Table F. SIO Timing (See Figures 53(a) and 56)

No	Symbol	Parameter	Z84x1306 Z84x1506 Min	Z84x1306 Z84x1506 Max	Z84x1310 Z84x1510 Min	Z84x1310 Z84x1510 Max	Units	Note
1	TwPh	Pulse Width (High)	150		120		nS	
2	TwPl	Pulse Width (Low)	150		120		nS	
3	TcTxC	/TxC Cycle Time	250		200		nS	[F1]
4	TwTxCl	/TxC Width (High)	85		80		nS	
5	TwTxCh	/TxC Width (Low)	85		80		nS	
6	TrTxC	/TxC Rise Time		60		60	nS	
7	TfTxC	/TxC Fall Time		60		60	nS	
8	TdTxCf(TxD)	/TxC Fall to TxD Delay		160		120	nS	
9	TdTxCf(W/RRf)	/TxC Fall to /W//RDY Fall Delay (Ready Mode)	5	9	5	9	TcC	
10	TdTxCf(INTf)	/TxC Fall to /INT Fall Delay	5	9	5	9	TcC	
11	TcRxC	/RxC Cycle Time	250		200		nS	[F1]
12	TwRxCh	/RxC Width (High)	85		80		nS	
13	TwRxCl	/RxC Width (Low)	85		80		nS	
14	TrRxC	/RxC Rise Time		60		60	nS	
15	TfRxC	/RxC Fall Time		60		60	nS	
16	TsRxD(RxCr)	RxD to /RxC Rise Setup Time (X1 mode)	0		0		nS	
17	ThRxCr(RxD)	/RxC Rise to RxD Hold Time (X1 Mode)	80		60		nS	
18	TdRxCr(W/RRf)	/RxC Rise to /W//RDY Fall Delay (Ready Mode)	10	13	10	13	TcC	
19	TdRxCr(INTf)	/RxC Rise to /INT Fall Delay	10	13	10	13	TcC	
20	TdRxCr(SYNCf)	/RxC Rise to /SYNC Fall Delay (Output Modes)	4	7	4	7	TcC	
21	TsSYNCf(RxCr)	/SYNC Fall to /RxC Rise Setup (External Sync Modes)	-100		-100		nS	[F2]
22	TdIOf(W/Rf)	/IORQ Fall or valid address to /W//RDY Delay (Wait Mode)		130		110	nS	[F2]
23	TdCr(W/Rf)	Clock Rise to /W//RDY Delay (Ready Mode)		85		85	nS	[F2]
24	TdCf(W/Rz)	Clock Fall to /W//RDY Float Delay (Wait Mode)		90		80	nS	[F2]

Notes:
[F1] In All Modes, the System Clock rate must be at least five times the maximum data rate.
[F2] Parameters 22 to 24 are on Figure 53(a).

Table G. Watch Dog Timer Timing (See Figure 57)

No	Symbol	Parameter	Min	Max	Min	Max	nS
1	TdC(WDTf)	Clock Rise to /WDTOUT Fall Delay		160		160	nS
2	TwPI	Clock Rise to /WDTOUT Rise Delay		165		165	nS
3	TcWDT	/WDTOUT Cycle Time					
		WDTP = 00	(Typ)2^{16}TcC		(Typ)2^{16}TcC		nS
		WDTP = 01	(Typ)2^{18}TcC		(Typ)2^{18}TcC		nS
		WDTP = 10	(Typ)2^{20}TcC		(Typ)2^{20}TcC		nS
		WDTP = 11	(Typ)2^{22}TcC		(Typ)2^{22}TcC		nS

Notes:
* In All Modes, the System Clock rate must be at least five times the maximum data rate.
RESET must be active a minimum of one complete clock cycle.
[1] Units equal to System Clock Periods.
[2] Units in nanoseconds (nS).

Additional information for note [C3]
Parameter #15, 16, 17 and 18 of Table C. These parameters are daisy chain timing and calculated values, and vary depending on the inside daisy chain configuration, which is specified in the Interrupt Priority Register. Inside the IPC, the daisy chain can be figured as follows:

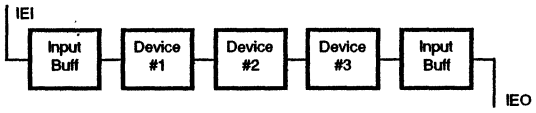


Figure 59. Internal Daisy Chain Configuration

No	Parameter	6MHz Min	6MHz Max	10MHz Min	10MHz Max
15	TdM1(IEO)		160nS		100nS
16	TsIEI(IO) (PIO at #3)	230nS		140nS	
	(CTC at #3)	280nS		160nS	
	(SIO at #3)	290nS		160nS	
17	TdIEI(IEOf)		120nS		70nS
18	TdIEI(IEOr)		290nS		150nS

To calculate IPC daisy chain timing, it can be treated as if there are Z80 PIO, CTC and SIO with Input buffer and look ahead circuit on the chain Following are the calculation formulas:

Parameter #15, /M1 falling to IEO delay
TsM1(IEO) = Max[TdM1(IO)#1,,TdM1(IO)#2, TdM1(IO)#3] + (look-ahead gate Delay)

Parameter #16, IEI to /IORQ falling setup time
TsIEI(IO) = TdIEI(IEO)#1 + TdIEI(IEO)#2 + TsIEI(IO)#3 + (Input Buffer delay)

Parameter #17, IEI falling to IEO falling delay
TdIEI(IEOf) = Max[TdIEI(IEOf)PIO, TdIEI(IEOf)CTC, TdIEI(IEOf)SIO] + (Input Buffer delay) + (look-ahead gate Delay)

Parameter #18, IEI rising to IEO rising delay (After ED decode)
TdIEI(IEOr) = TdIEI(IEOr)PIO+ TdIEI(IEOr)CTC + TdIEI(IEOr)SIO + (Input Buffer delay) + (look-ahead gate Delay)

* Where TdIEI(IEO) is worse number between TdIEI(IEOr) and TdIEI(IEOf)

Numbers to calculate these parameters for the above formulas are on the next page.

	6MHz Min	Max	10MHz Min	Max
Input Buffer delay	10nS		10nS	
Look ahead gate delay	10nS		10nS	

6MHz	PIO part Min	Max	CTC part Min	Max	SIO part Min	Max
TdM1(IEO)		90nS		130nS		150nS
TsIEI(IO)		90nS		100nS		70nS
TdIEI(IEOf)		100nS		90nS		50nS
TdIEI(IEOr)		130nS		90nS		50nS

10MHz	PIO part Min	Max	CTC part Min	Max	SIO part Min	Max
TdM1(IEO)		60nS		60nS		90nS
TsIEI(IO)		50nS		70nS		50nS
TdIEI(IEOf)		50nS		50nS		30nS
TdIEI(IEOr)		50nS		50nS		30nS

If using an interrupt from only a portion of the IPC, these numbers are smaller than the values shown above. For more details about the "Z80 Daisy Chain Structure," please refer to the Application Note "Z80 Family Interrupt Structure" included in the Z80 Data book.

PRODUCT SPECIFICATION

Z8440/1/2/4, Z84C40/1/2/3/4

SERIAL INPUT/OUTPUT CONTROLLER

FEATURES

- Two independent full-duplex channels, with separate control and status lines for modems or other devices.
- **Data rate in the x1 clock mode of 0 to 1.6M bits/second with a 8.0 MHz clock.**
- **NMOS version for high cost performance solutions, CMOS version for the designs requires low power consumption.**
- **NMOS Z0844x04 - 4 MHz Z0844x06 - 6.17 MHz (Where x is the designator for the bonding option; 0, 1, 2 or 4)**
- **CMOS Z84C4x04 - DC 4 MHz Z84C4x06 - DC to 6.7 MHz Z84C4x08 - DC to 8 MHz (Where x is the designator for the bonding option; 0, 1, 2 or 3, 4)**
- **6 MHz version supports 6.144 MHz CPU clock operation.**
- Asynchronous protocols: everything necessary for complete messages in 5, 6, 7, or 8 bits/character. Includes variable stop bits and several clock-rate multipliers; break generation and detection; parity; overrun and framing error detection.
- Synchronous protocols: everything necessary for complete bit- or byte-oriented messages in 5, 6, 7, or 8 bits/character, including IBM Bisync, SDLC, HDLC, CCITT-X.25 and others. Automatic CRC generation/checking, sync character and zero insertion/deletion, abort generation/detection, and flag insertion.
- Receiver data registers quadruply buffered, transmitter registers doubly buffered.
- Highly sophisticated and flexible daisy-chain interrupt vectoring for interrupts without external logic.

GENERAL DESCRIPTION

The Z80 SIO (here in after referred to as the Z80 SIO or, SIO). Serial Input/Output Controller is a dual-channel data communication interface with extraordinary versatility and capability. Its basic functions as a serial-to-parallel, parallel-to-serial converter/controller can be programmed by a CPU for a broad range of serial communication applications.

The device supports all common asynchronous and synchronous protocols, byte- or bit-oriented, and performs all of the functions traditionally done by UARTs, USARTs, and synchronous communication controllers combined, plus additional functions traditionally performed by the CPU. Moreover, it does this on two fully-independent channels, with an exceptionally sophisticated interrupt structure that allows very fast transfers.

Full interfacing is provided for CPU or DMA control. In addition to data communication, the circuit can handle virtually all types of serial I/O with fast, or slow, peripheral devices. While designed primarily as a member of the Z80 family, its versatility makes it well suited to many other CPUs.

The Z80 SIO uses a single +5V power supply and the standard Z80 family single-phase clock. The SIO/0, SIO/1, and SIO/2 are packaged in a 40-pin DIP, the SIO/4 is packaged in a 44-pin PCC and the SIO/3 is packaged in a 44-pin QFP. Note that SIO/3 is only available in CMOS and in QFP package.

PIN DESCRIPTION

Figures 1 through 6 illustrate the three 40-pin configurations (bonding options) available in the Z80C SIO (hereafter referred to as SIO or Z80 SIO). The constraints of a 40-pin package make it impossible to bring out the Receive Clock ($\overline{RxC}$), Transmit Clock ($\overline{TxC}$), Data Terminal Ready ($\overline{DTR}$) and Sync ($\overline{SYNC}$) signals for both channels. Therefore, either Channel B lacks a signal or two signals are bonded together:

- Z80 SIO/2 lacks $\overline{SYNCB}$
- Z80 SIO/1 lacks $\overline{DTRB}$
- Z80 SIO/0 has all four signals, but $\overline{TxCB}$ and $\overline{RxCB}$ are bonded together

The 44-pin package, the Z80 SIO/4 for PLCC package, and Z80 SIO/3 for QFP, has all options (Figure 7a and 7b).

The first bonding option above (SIO/2) is the preferred version for most applications. The pin descriptions are as follows:

B/$\overline{A}$. *Channel A or B Select* (input, High selects Channel B). This input defines which channel is accessed during a data

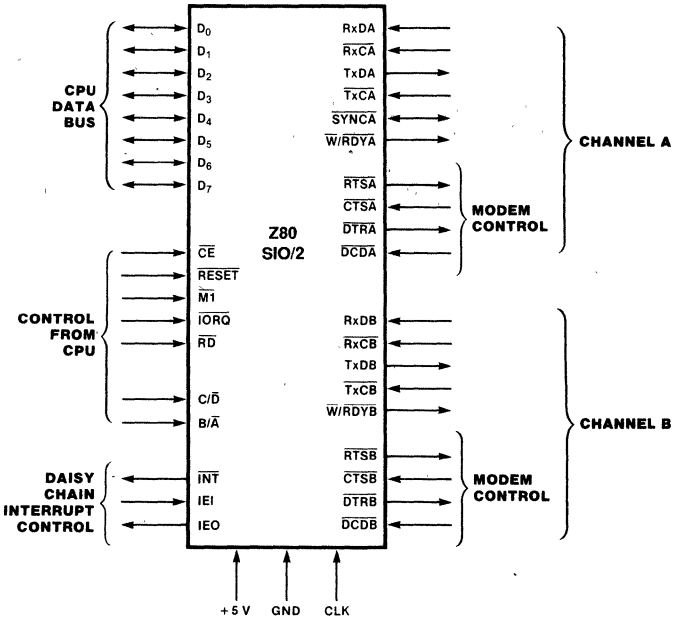


Figure 1. Pin Functions

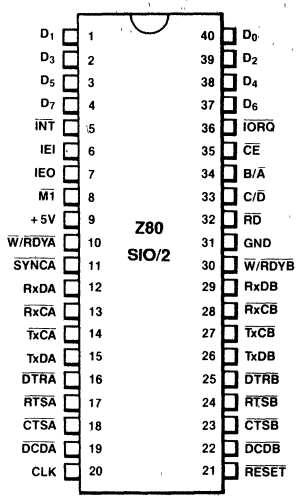


Figure 2. 40-pin Dual-In-Line Package (DIP), Pin Assignments

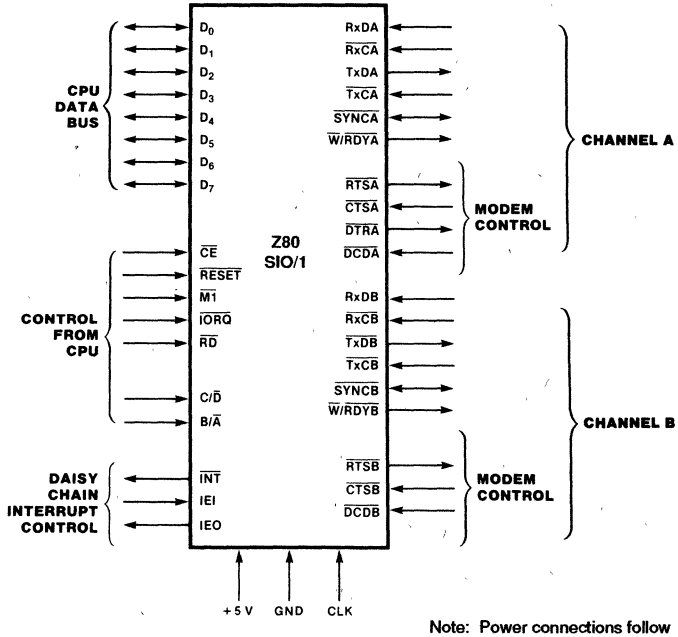


Figure 3. Pin Functions

Note: Power connections follow conventional descriptions below:

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

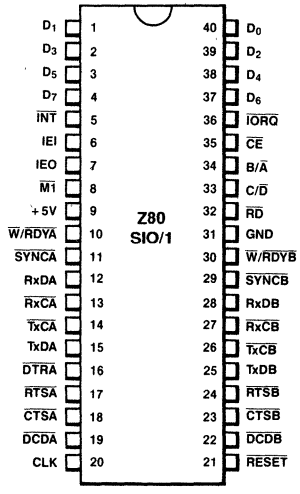


Figure 4. 40-pin Dual-In-Line Package (DIP), Pin Assignments

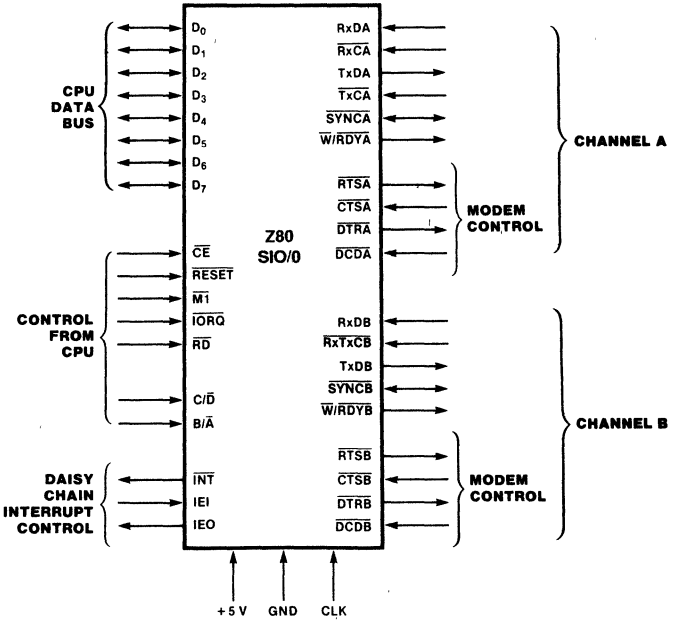


Figure 5. Pin Functions

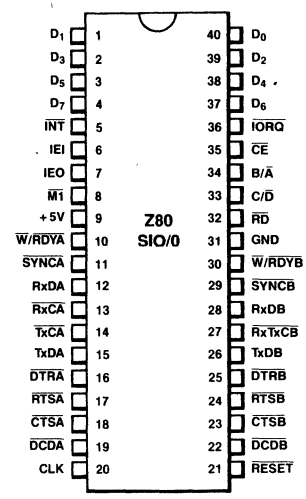


Figure 6. 40-pin Dual-In-Line Package (DIP), Pin Assignments

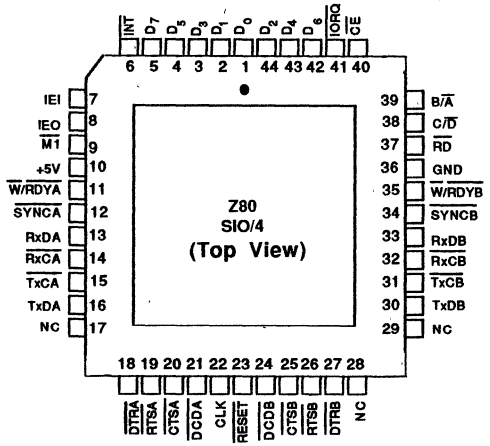


Figure 7a. 44-pin Chip Carrier, Pin Assignments

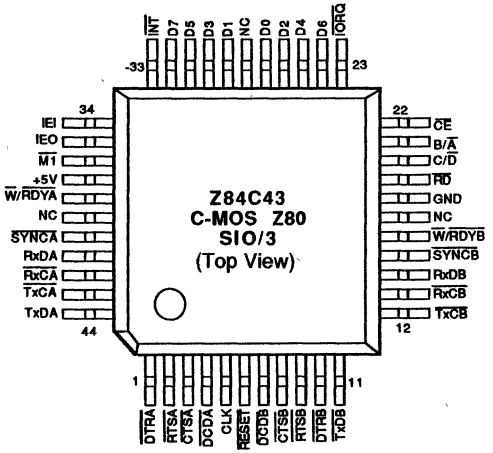


Figure 7b. 44-pin Quad Flat Pack Pin Assignments

transfer between the CPU and the SIO. Address bit A_0 from the CPU is often used for the selection function.

C/$\overline{D}$. *Control or Data Select* (input, High selects Control). This input defines the type of information transfer performed between the CPU and the SIO. A High at this input during a CPU write to the SIO causes the information on the data bus to be interpreted as a command for the channel selected by B/$\overline{A}$. A Low at C/$\overline{D}$ means that the information on the data bus is data. Address bit A_1 is often used for this function.

$\overline{CE}$. *Chip Enable* (Input, active Low). A Low level at this input enables the SIO to accept command or data input from the CPU during a write cycle, or to transmit data to the CPU during a read cycle.

CLK. *System Clock* (input). The SIO uses the standard Z80 System Clock to synchronize internal signals. This is single-phase clock.

$\overline{CTSA}$, $\overline{CTSB}$. *Clear To Send* (inputs, active Low). When programmed as Auto Enables, a Low on these inputs enables the respective transmitter. If not programmed as Auto Enables, these inputs may be programmed as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow-risetime signals. The SIO detects pulses on these inputs and interrupts the CPU on both logic level transitions. The Schmitt-trigger buffering does not guarantee a specified noise-level margin.

D_0-D_7. *System Data Bus* (bidirectional, 3-state). The system data bus transfers data and commands between the CPU and the Z80 SIO. D_0 is the least significant bit.

$\overline{DCDA}$, $\overline{DCDB}$. *Data Carrier Detect* (inputs, active Low). These pins function as receiver enables if the SIO is programmed for Auto Enables; otherwise they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accommodate slow-risetime signals. The SIO detects pulses on these pins and interrupts the CPU on both logic level transitions. Schmitt-trigger buffering does not guarantee a specific noise-level margin.

$\overline{DTRA}$, $\overline{DTRB}$. *Data Terminal Ready* (outputs, active Low). These outputs follow the state programmed into the Z80 SIO. They can also be programmed as general-purpose outputs.

In the Z80 SIO/1 bonding option, $\overline{DTRB}$ is omitted.

IEI. *Interrupt Enable In* (input, active High). This signal is used with IEO to form a priority daisy chain when there is more than one interrupt-driven device. A High on this line indicates that no other device of higher priority is being serviced by a CPU interrupt service routine.

IEO. *Interrupt Enable Out* (output, active High). IEO is High only if IEI is High and the CPU is not servicing an interrupt from this SIO. Thus, this signal blocks lower priority devices from interrupting while a higher priority device is being serviced by its CPU interrupt service routine.

$\overline{INT}$. *Interrupt Request* (output, open drain, active Low). When the SIO is requesting an interrupt, it pulls $\overline{INT}$ Low.

$\overline{IORQ}$. *Input/Output Request* (input from CPU, active Low). $\overline{IORQ}$ is used in conjunction with B/$\overline{A}$, C/$\overline{D}$, $\overline{CE}$, and $\overline{RD}$ to transfer commands and data between the CPU and the SIO. When $\overline{CE}$, $\overline{RD}$, and $\overline{IORQ}$ are all active, the channel selected by B/$\overline{A}$ transfers data to the CPU (a read operation). When $\overline{CE}$ and $\overline{IORQ}$ are active, but $\overline{RD}$ is inactive, the channel selected by B/$\overline{A}$ is written to by the CPU with either data or control information as specified by C/$\overline{D}$. As mentioned previously, if $\overline{IORQ}$ and $\overline{M1}$ are active simultaneously, the CPU is acknowledging an interrupt and the SIO automatically places its interrupt vector on the CPU data bus if it is the highest priority device requesting an interrupt.

$\overline{M1}$. *Machine Cycle One*(input from Z80 CPU, active Low). When $\overline{M1}$ is active and $\overline{RD}$ is also active, the Z80 CPU is fetching an instruction from memory; when $\overline{M1}$ is active while $\overline{IORQ}$ is active, the SIO accepts $\overline{M1}$ and $\overline{IORQ}$ as an interrupt acknowledge if the SIO is the highest priority device that has interrupted the Z80 CPU.

$\overline{RxCA}$, $\overline{RxCB}$. *Receiver Clocks* (inputs). Receive data is sampled on the rising edge of $\overline{RxC}$. The Receive Clocks may be 1, 16, 32, or 64 times the data rate in asynchronous modes. These clocks may be driven by the Z80 CTC Counter Timer Circuit for programmable baud rate generation. Both inputs are Schmitt-trigger buffered; no noise level margin is specified.

In the Z80 SIO/0 bonding option, $\overline{RxCB}$ is bonded together with $\overline{TxCB}$.

$\overline{RD}$. *Read Cycle Status* (input from CPU, active Low). If $\overline{RD}$ is active, a memory or I/O read operation is in progress. $\overline{RD}$ is used with B/$\overline{A}$, $\overline{CE}$, and $\overline{IORQ}$ to transfer data from the SIO to the CPU.

RxDA, RxDB. *Receive Data* (inputs, active High). Serial data at TTL levels.

$\overline{RESET}$. *Reset* (input, active Low). A Low $\overline{RESET}$ disables both receivers and transmitters, forces TxDA and TxDB marking, forces the modem controls High, and disables all interrupts. The control registers must be rewritten after the SIO is reset and before data is transmitted or received.

$\overline{RTSA}$, $\overline{RTSB}$. *Request To Send* (outputs, active Low). When the RTS bit in Write Register 5 (Figure 14) is set, the $\overline{RTS}$ output goes Low. When the RTS bit is reset in the Asynchronous mode, the output goes High after the transmitter is empty. In Synchronous modes, the $\overline{RTS}$ pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.

$\overline{SYNCA}$, $\overline{SYNCB}$. *Synchronization* (bidirectional, active Low). These pins can act either as inputs or outputs. In the asynchronous receive mode, they are inputs similar to $\overline{CTS}$ and $\overline{DCD}$. In this mode, the transitions on these lines affect the state of the Sync/Hunt status bits in Read Register 0 (Figure 13), but have no other function. In the External Sync mode, these lines also act as inputs. When external synchronization is achieved, $\overline{SYNC}$ must be driven Low on the second rising edge of $\overline{RxC}$ after that rising edge of $\overline{RxC}$ on which the last bit of the sync character was received. In other words, after the sync pattern is detected, the external logic must wait for two full Receive Clock cycles to activate the $\overline{SYNC}$ input. Once $\overline{SYNC}$ is forced Low, it should be kept Low until the CPU informs the external synchronization detect logic that synchronization has been lost or a new message is about to start. Character assembly begins on the rising edge of $\overline{RxC}$ that immediately precedes the falling edge of $\overline{SYNC}$ in the External Sync mode.

In the internal synchronization mode (Monosync and Bisync), these pins act as outputs that are active during the part of the receive clock ($\overline{RxC}$) cycle in which sync characters are recognized. The sync condition is not latched, so these outputs are active each time a sync pattern

is recognized, regardless of character boundaries.

In the Z80 SIO/2 bonding option, $\overline{\text{SYNCB}}$ is omitted.

$\overline{\text{TxCA}}$, $\overline{\text{TxCB}}$. *Transmitter Clocks* (inputs). In asynchronous modes, the Transmitter Clocks may be 1, 16, 32, or 64 times the data rate, however, the clock multiplier must be the same for the transmitter and the receiver. The Transmit Clock inputs are Schmitt-trigger buffered for relaxed rise- and fall-time requirements; no noise level margin is specified. Transmitter Clocks may be driven by the Z80 CTC Counter Timer Circuit for programmable baud rate generation.

In the Z80 SIO/0 bonding option, $\overline{\text{TxCB}}$ is bonded together with $\overline{\text{RxCB}}$.

TxDA, TxDB. *Transmit Data* (outputs, active High). Serial data at TTL levels. TxD changes from the falling edge of $\overline{\text{TxC}}$.

$\overline{\text{W/RDYA}}$, $\overline{\text{W/RDYB}}$. *Wait/Ready* (outputs, open drain when programmed for Wait function; driven High and Low when programmed for Ready function). These dual-purpose outputs may be programmed as Ready lines for a DMA controller or as Wait lines that synchronize the CPU to the SIO data rate. The reset state is open drain.

FUNCTIONAL DESCRIPTION

The functional capabilities of the Z80 SIO can be described from two different points of view: as a data communications device, it transmits and receives serial data in a wide variety of data-communication protocols; as a Z80 family peripheral, it interacts with the Z80 CPU and other peripheral circuits, sharing the data, address and control buses, as well as being a part of the Z80 interrupt structure. As a peripheral to other microprocessors, the SIO offers valuable features such as non-vectored interrupts, polling, and simple handshake capability. Figure 8 is a block diagram.

Figure 9 illustrates the conventional devices that the SIO replaces.

The first part of the following discussion covers SIO data-communication capabilities; the second part describes interactions between the CPU and the SIO.

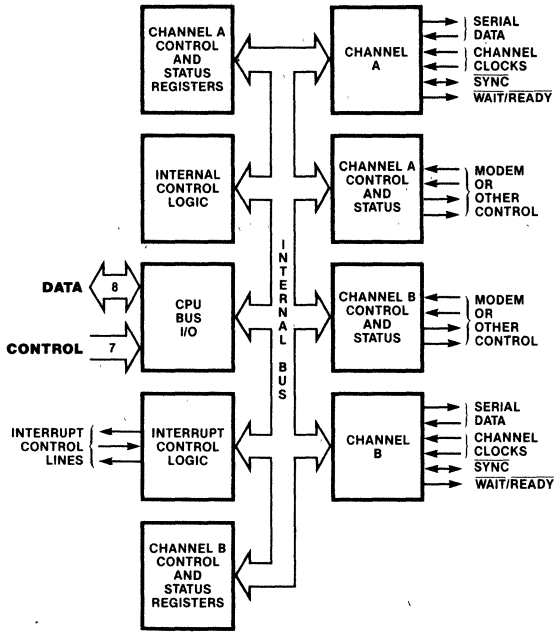


Figure 8. Block Diagram

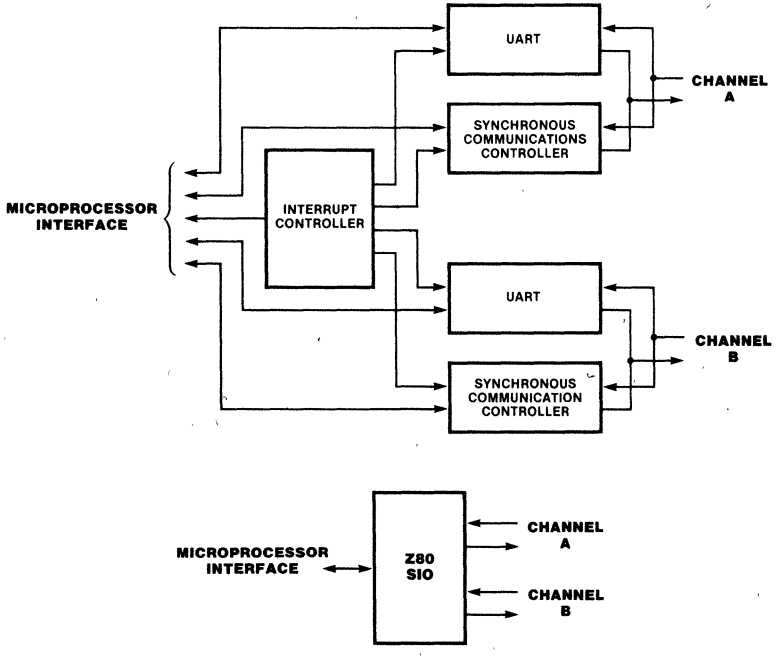


Figure 9. Conventional Devices Replaced by the Z80 SIO

DATA COMMUNICATION CAPABILITIES

The SIO provides two independent full-duplex channels that can be programmed for use in any common asynchronous, or synchronous data-communication protocol. Figure 10a illustrates some of these protocols. The following is a short description of them. A more detailed explanation of these modes can be found in the *Z80 SIO Technical Manual* (03-3033-01).

Asynchronous Modes. Transmission and reception can be done independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one-and-a-half, or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and end of a received break. Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxDA or RxDB in Figure 5). If the Low does not persist, as in the case of a transient, the character assembly process is not started.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occurred. Vectored interrupts allow fast servicing of error conditions using dedicated routines. Furthermore, a built-in checking process avoids interpreting a framing error as a new start bit: a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit is begun.

The SIO does not require symmetric transmit and receive clock signals, a feature that allows it to be used with a Z80 CTC or many other clock sources. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32, or 1/64 of the clock rate supplied to the receive and transmit clock inputs.

In asynchronous modes, the $\overline{\text{SYNC}}$ pin may be programmed as an input that can be used for functions such as monitoring a ring indicator.

Synchronous Modes. The SIO supports both byte-oriented and bit-oriented synchronous communication.

Synchronous byte-oriented protocols can be handled in several modes that allow character synchronization with an 8-bit sync character (Monosync), any 16-bit sync pattern (Bisync), or with an external sync signal. Leading sync characters can be removed without interrupting the CPU.

Five-, six-, or seven-bit sync characters are detected with 8- or 16-bit patterns in the SIO by overlapping the larger pattern across multiple incoming sync characters, as shown in Figure 10b.

CRC checking for synchronous byte-oriented modes is delayed by one character time so the CPU may disable CRC checking on specific characters. This permits implementation of protocols such as IBM Bisync.

Figure 10a. Some Z80 SIO Protocols

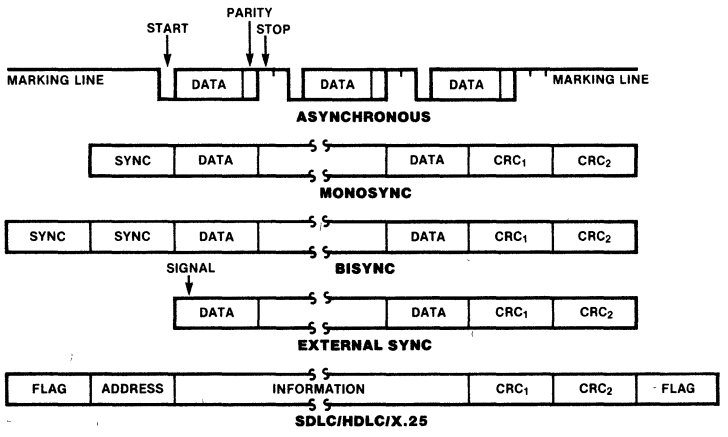


Figure 10b. Six-Bit Sync Character Recognition

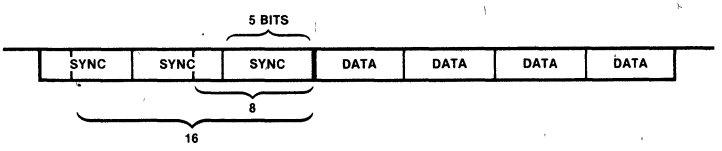


Figure 10. Data Communication

Both CRC-16 ($X^{16} + X^{15} + X^2 + 1$) and CCITT ($X^{16} + X^{12} + X^5 + 1$) error checking polynomials are supported. In all non-SDLC modes, the CRC generator is initialized to 0s; in SDLC modes, it is initialized to 1s. The SIO can be used for interfacing to peripherals such as hard-sectored floppy disks, but it cannot generate or check CRC for IBM-compatible soft-sectored disks. The SIO also provides a feature that automatically transmits CRC data when no other data is available for transmission. This allows very high-speed transmissions under DMA control with no need for CPU intervention at the end of a message. When there is no data or CRC to send in synchronous modes, the transmitter inserts 8- or 16-bit sync characters regardless of the programmed character length.

The SIO supports synchronous bit-oriented protocols such as SDLC and HDLC by performing automatic flag sending, zero insertion, and CRC generation. A special command can be used to abort a frame in transmission. At the end of a message the SIO automatically transmits the CRC and trailing flag when the transmit buffer becomes empty. If a transmit underrun occurs in the middle of a message, an external/status interrupt warns the CPU of this status change so that an abort may be issued. One to eight bits per character can be sent, which allows reception of a message with no prior information about the character structure in the information field of a frame.

The receiver automatically synchronizes on the leading flag of a frame in SDLC or HDLC, and provides a synchronization signal on the $\overline{SYNC}$ pin; an interrupt can also be programmed. The receiver can be programmed to search for frames addressed by a single byte to only a specified user-selected address or to a global broadcast address. In this mode, frames that do not match either the user-selected or broadcast address are ignored. The number of address bytes can be extended under software control. For transmitting data, an interrupt on the first received character or on every character can be selected. The receiver automatically deletes all zeroes inserted by the transmitter during character assembly. It also calculates and automatically checks the CRC to validate frame transmission. At the end of transmission, the status of a received frame is available in the status registers.

The SIO can be conveniently used under DMA control to provide high-speed reception or transmission. In reception, for example, the SIO can interrupt the CPU when the first character of a message is received. The CPU then enables the DMA to transfer the message to memory. The SIO then issues an end-of-frame interrupt and the CPU can check the status of the received message. Thus, the CPU is freed for other service while the message is being received.

I/O INTERFACE CAPABILITIES

The SIO offers the choice of polling, vectored or non-vectored interrupts and block-transfer modes to transfer data, status, and control information to, and from, the CPU. The block-transfer mode can also be implemented under DMA control.

Polling. Two status registers are updated at appropriate times for each function being performed (for example, CRC error-status valid at the end of a message). When the CPU is operated in a polling fashion, one of the SIO's two status registers is used to indicate whether the SIO has some data or needs some data. Depending on the contents of this register, the CPU will either write data, read data, or just go on. Two bits in the register indicate that a data transfer is needed. In addition, error and other conditions are indicated. The second status register (special receive conditions) does not have to be read in a polling sequence, until a character has been received. All interrupt modes are disabled when operating the device in a polled environment.

Interrupts. The SIO has an elaborate interrupt scheme to provide fast interrupt service in real-time applications. A control register and a status register in Channel B contain the interrupt vector. When programmed to do so, the SIO can modify three bits of the interrupt vector in the status register so that it points directly to one of eight interrupt service routines in memory, thereby servicing conditions in both channels and eliminating most of the needs for a status-analysis routine.

Transmit interrupts, receive interrupts, and external/status interrupts are the main sources of interrupts. Each interrupt source is enabled under program control, with Channel A having a higher priority than Channel B, and with receive, transmit, and external/status interrupts prioritized in that order within each channel. When the transmit interrupt is enabled, the CPU is interrupted by the transmit buffer becoming empty. (This implies that the transmitter must have had a data character written into it so it can become empty.) The receiver can interrupt the CPU in one of two ways:

- Interrupt on first received character
- Interrupt on all received characters

Interrupt-on-first-received-character is typically used with the block-transfer mode. Interrupt-on-all-received-characters has the option of modifying the interrupt vector in the event of a parity error. Both of these interrupt modes will also interrupt under special receive conditions on a character or message basis (end-of-frame interrupt in SDLC, for example). This means that the special-receive condition can cause an interrupt only if the interrupt-on-first-received-character or interrupt-on-all-received-characters mode is selected. In interrupt-on-first-received-character, an interrupt can occur from special-receive conditions (except parity error) after the first-received-character interrupt (example: receive-overrun interrupt).

The main function of the external/status interrupt is to monitor the signal transitions of the Clear To Send ($\overline{CTS}$), Data Carrier Detect ($\overline{DCD}$), and Synchronization ($\overline{SYNC}$) pins (Figures 1 through 7). In addition, an external/status

interrupt is also caused by a CRC-sending condition, or by the detection of a break sequence (asynchronous mode) or abort sequence (SDLC mode) in the data stream. The interrupt caused by the break/abort sequence allows the SIO to interrupt when the break/abort sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the break/abort condition in external logic.

In a Z80 CPU environment (Figure 11), SIO interrupt vectoring is "automatic": the SIO passes its internally-modifiable 8-bit interrupt vector to the CPU, which adds an additional 8 bits from its interrupt-vector (I) register to form the memory address of the interrupt-routine table. This table contains the address of the beginning of the interrupt routine itself. The process entails an indirect transfer of CPU control to the interrupt routine, so that the next instruction executed after an interrupt acknowledge by the CPU is the first instruction of the interrupt routine itself.

CPU/DMA Block Transfer. The SIO's block-transfer mode accommodates both CPU block transfers and DMA controllers (Z80 DMA or other designs). The block-transfer mode uses the Wait/Ready output signal, which is selected with three bits in an internal control register. The Wait/Ready output signal can be programmed as a $\overline{WAIT}$ line in the CPU block-transfer mode or as a $\overline{READY}$ line in the DMA block-transfer mode.

To a DMA controller, the SIO $\overline{READY}$ output indicates that the SIO is ready to transfer data to, or from, memory. To the CPU, the $\overline{WAIT}$ output indicates that the SIO is not ready to transfer data, thereby requesting the CPU to extend the I/O cycle.

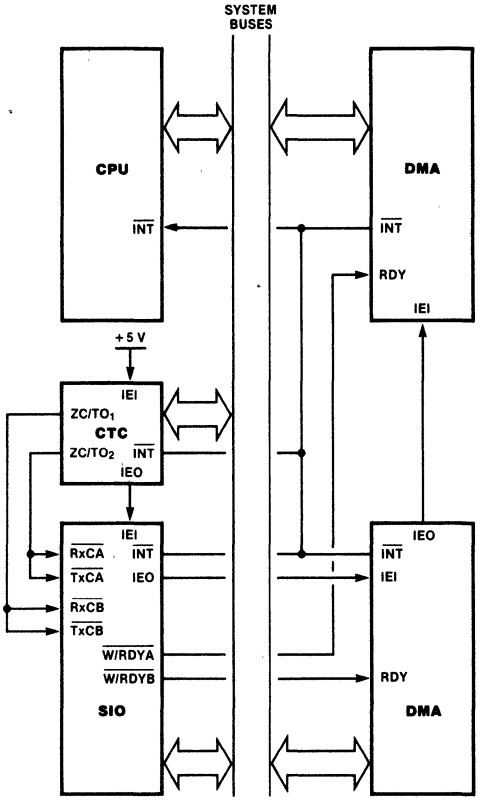


Figure 11. Typical Z80 Environment

INTERNAL STRUCTURE

The internal structure of the device includes a Z80 CPU interface, internal control and interrupt logic, and two full-duplex channels. Each channel contains its own set of control and status (write and read) registers, and control and status logic that provides the interface to modems or other external devices

The registers for each channel are designated as follows:

WR0-WR7 — Write Registers 0 through 7
RR0-RR2 — Read Registers 0 through 2

The register group includes five 8-bit control registers, two sync-character registers and two status registers. The interrupt vector is written into an additional 8-bit register (Write Register 2) in Channel B that may be read through another 8-bit register (Read Register 2) in Channel B. The bit assignment and functional grouping of each register is configured to simplify and organize the programming process. Table 1 lists the functions assigned to each read or write register.

The logic for both channels provides formats, synchronization, and validation for data transferred to and from the channel interface. The modem control inputs, Clear To Send ($\overline{CTS}$) and Data Carrier Detect ($\overline{DCD}$), are

Table 1. Register Functions

Read Register Functions	
RR0	Transmit/Receive buffer status, interrupt status and external status
RR1	Special Receive Condition status
RR2	Modified interrupt vector (Channel B only)
Write Register Functions	
WR0	Register pointers, CRC initialize, and initialization commands for the various modes.
WR1	Transmit/Receive interrupt and data transfer mode definition.
WR2	Interrupt vector (Channel B only)
WR3	Receive parameters and control
WR4	Transmit/Receive miscellaneous parameters and modes
WR5	Transmit parameters and controls
WR6	Sync character or SDLC address field
WR7	Sync character or SDLC flag

monitored by the external control and status logic under program control. All external control-and-status-logic signals are general-purpose in nature and can be used for functions other than modem control.

Data Path. The transmit and receive data path illustrated for Channel A in Figure 12 is identical for both channels. The receiver has three 8-bit buffer registers in a FIFO arrangement, in addition to the 8-bit receive shift register. This scheme creates additional time for the CPU to service an interrupt at the beginning of a block of high-speed data. Incoming data is routed through one of several paths (data or CRC) depending on the selected mode and—in asynchronous modes—the character length.

The transmitter has an 8-bit transmit data buffer register that is loaded from the internal data bus, and a 20-bit transmit shift register that can be loaded from the sync-character buffers or from the transmit data register. Depending on the operational mode, outgoing data is routed through one of four main paths before it is transmitted from the Transmit Data output (TxD).

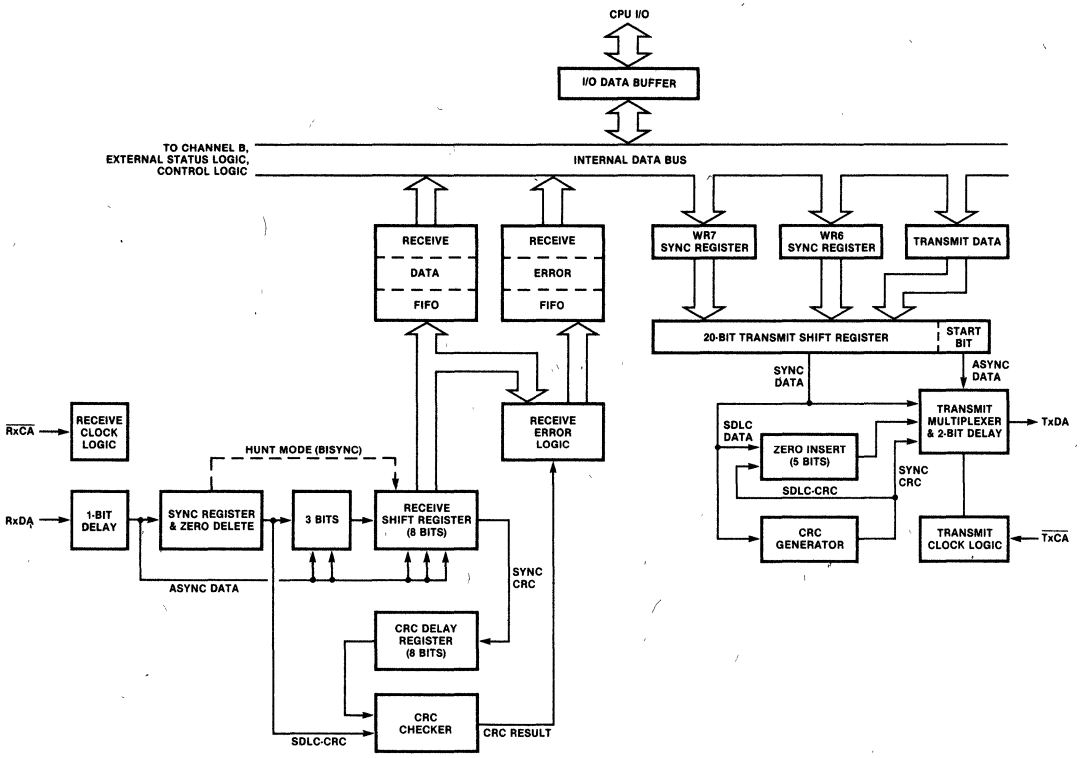


Figure 12. Transmit and Receive Data Path (Channel A)

PROGRAMMING

The system program first issues a series of commands that initialize the basic mode of operation and then issues other commands that qualify conditions within the selected mode. For example, the asynchronous mode, character length, clock rate, number of stop bits, even or odd parity might be set first; then the interrupt mode; and finally, receiver or transmitter enable.

Both channels contain registers that must be programmed via the system program prior to operation. The channel-select input (B/$\overline{A}$) and the control/data (C/$\overline{D}$) are the command-structure addressing controls, and are normally controlled by the CPU address bus. Figures 15 and 16 illustrate the timing relationships for programming the write registers and transferring data and status

Read Registers. The SIO contains three read registers for Channel B and two read registers for Channel A (RR0-RR2 in Figure 13) that can be read to obtain the status information; RR2 contains the internally-modifiable interrupt vector and is only in the Channel B register set. The status information includes error conditions, interrupt vector, and standard communications-interface signals.

To read the contents of a selected read register other than RR0, the system program must first write the pointer byte to WR0 in exactly the same way as a write register operation. Then, by executing a read instruction, the contents of the addressed read register can be read by the CPU.

The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring. For example, when the interrupt vector indicates that a Special Receive Condition interrupt has occurred, all the appropriate error bits can be read from a single register (RR1).

Write Registers. The SIO contains eight write registers for Channel B and seven write registers for Channel A (WR0-WR7 in Figure 14) that are programmed separately to configure the functional personality of the channels; WR2 contains the interrupt vector for both channels and is only in the Channel B register set. With the exception of WR0, programming the write registers requires two bytes. The first byte is to WR0 and contains three bits (D_0-D_2) that point to the selected register, the second byte is the actual control word that is written into the register to configure the SIO.

WR0 is a special case in that all of the basic commands can be written to it with a single byte. Reset (internal or external) initializes the pointer bits D_0-D_2 to point to WR0. This implies that a channel reset must not be combined with the pointing to any register.

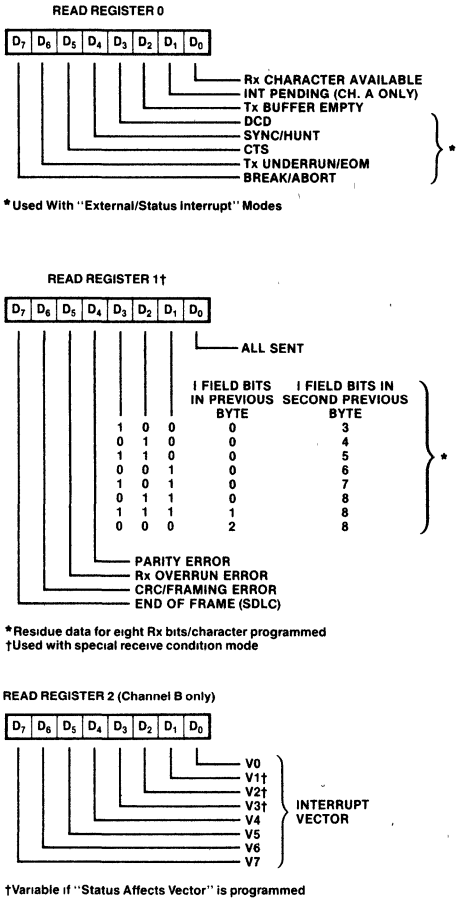


Figure 13. Read Register Bit Functions

WRITE REGISTER 0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

D_2 D_1 D_0
- 0 0 0 REGISTER 0
- 0 0 1 REGISTER 1
- 0 1 0 REGISTER 2
- 0 1 1 REGISTER 3
- 1 0 0 REGISTER 4
- 1 0 1 REGISTER 5
- 1 1 0 REGISTER 6
- 1 1 1 REGISTER 7

D_5 D_4 D_3
- 0 0 0 NULL CODE
- 0 0 1 SEND ABORT (SDLC)
- 0 1 0 RESET EXT/STATUS INTERRUPTS
- 0 1 1 CHANNEL RESET
- 1 0 0 ENABLE INT ON NEXT Rx CHARACTER
- 1 0 1 RESET TxINT PENDING
- 1 1 0 ERROR RESET
- 1 1 1 RETURN FROM INT (CH-A ONLY)

D_7 D_6
- 0 0 NULL CODE
- 0 1 RESET Rx CRC CHECKER
- 1 0 RESET Tx CRC GENERATOR
- 1 1 RESET Tx UNDERRUN/EOM LATCH

WRITE REGISTER 1

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

D_0
- EXT INT ENABLE
- Tx INT ENABLE
- STATUS AFFECTS VECTOR (CH. B ONLY)

D_4 D_3
- 0 0 Rx INT DISABLE
- 0 1 Rx INT ON FIRST CHARACTER
- 1 0 INT ON ALL Rx CHARACTERS (PARITY AFFECTS VECTOR) *
- 1 1 INT ON ALL Rx CHARACTERS (PARITY DOES NOT AFFECT VECTOR) *

D_2
- $\overline{\text{WAIT}}$/READY ON R/T

D_1
- $\overline{\text{WAIT}}$/READY FUNCTION

D_0
- $\overline{\text{WAIT}}$/READY ENABLE

*Or on special condition

WRITE REGISTER 2 (Channel B only)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

INTERRUPT VECTOR
- V0
- V1
- V2
- V3
- V4
- V5
- V6
- V7

WRITE REGISTER 3

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

D_0
- Rx ENABLE

D_1
- SYNC CHARACTER LOAD INHIBIT

D_2
- ADDRESS SEARCH MODE (SDLC)

D_3
- Rx CRC ENABLE

D_4
- ENTER HUNT PHASE

D_5
- AUTO ENABLES

D_7 D_6
- 0 0 Rx 5 BITS/CHARACTER
- 0 1 Rx 7 BITS/CHARACTER
- 1 0 Rx 6 BITS/CHARACTER
- 1 1 Rx 8 BITS/CHARACTER

WRITE REGISTER 4

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

D_0
- PARITY ENABLE

D_1
- PARITY EVEN/$\overline{\text{ODD}}$

D_3 D_2
- 0 0 SYNC MODES ENABLE
- 0 1 1 STOP BIT/CHARACTER
- 1 0 1½ STOP BITS/CHARACTER
- 1 1 2 STOP BITS/CHARACTER

D_5 D_4
- 0 0 8 BIT SYNC CHARACTER
- 0 1 16 BIT SYNC CHARACTER
- 1 0 SDLC MODE (01111110 FLAG)
- 1 1 EXTERNAL SYNC MODE

D_7 D_6
- 0 0 X1 CLOCK MODE
- 0 1 X16 CLOCK MODE
- 1 0 X32 CLOCK MODE
- 1 1 X64 CLOCK MODE

WRITE REGISTER 5

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

D_0
- Tx CRC ENABLE

D_1
- RTS

D_2
- SDLC/CRC-16

D_3
- Tx ENABLE

D_4
- SEND BREAK

D_6 D_5
- 0 0 Tx 5 BITS (OR LESS)/CHARACTER
- 0 1 Tx 7 BITS/CHARACTER
- 1 0 Tx 6 BITS/CHARACTER
- 1 1 Tx 8 BITS/CHARACTER

D_7
- DTR

WRITE REGISTER 6

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

- SYNC BIT 0
- SYNC BIT 1
- SYNC BIT 2
- SYNC BIT 3
- SYNC BIT 4
- SYNC BIT 5
- SYNC BIT 6
- SYNC BIT 7

*

*Also SDLC address field

WRITE REGISTER 7

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

- SYNC BIT 8
- SYNC BIT 9
- SYNC BIT 10
- SYNC BIT 11
- SYNC BIT 12
- SYNC BIT 13
- SYNC BIT 14
- SYNC BIT 15

*

*For SDLC it must be programmed to "01111110" for flag recognition

Figure 14. Write Register Bit Functions

TIMING

The SIO must have the same clock as the CPU (same phase and frequency relationship, not necessarily the same driver).

Read Cycle. The timing signals generated by a Z80 CPU input instruction to read a data or status byte from the SIO are illustrated in Figure 15.

Write Cycle. Figure 16 illustrates the timing and data signals generated by a Z80 CPU output instruction to write a data or control byte into the SIO.

Interrupt-Acknowledge Cycle. After receiving an interrupt-request signal from an SIO ($\overline{INT}$ pulled Low), the Z80 CPU sends an interrupt-acknowledge sequence, $\overline{M1}$ Low and $\overline{IORQ}$ Low, a few cycles later (Figure 17).

The SIO contains an internal daisy-chained interrupt structure for prioritizing nested interrupts for the various functions of its two channels, and this structure can be used within an external user-defined daisy chain that prioritizes several peripheral circuits.

The IEI of the highest-priority device is terminated High. A device that has an interrupt pending or under service forces its IEO Low. For devices with no interrupt pending or under service, IEO = IEI.

To insure stable conditions in the daisy chain, all interrupt status signals are prevented from changing while $\overline{M1}$ is Low. When $\overline{IORQ}$ is Low, the highest priority interrupt requestor (the one with IEI High) places its interrupt vector on the data bus and sets its internal interrupt-under-service latch.

Return From Interrupt Cycle. Figure 18 illustrates the return from interrupt cycle. Normally, the Z80 CPU issues a Return From Interrupt (RETI) instruction at the end of an interrupt service routine. RETI is a 2-byte opcode (ED-4D) that resets the interrupt-under-service latch in the SIO to terminate the interrupt that has just been processed. This is accomplished by manipulating the daisy chain in the following way.

The normal daisy-chain operation can be used to detect a pending interrupt; however, it cannot distinguish between an interrupt under service and a pending unacknowledged interrupt of a higher priority. Whenever ED is decoded, the daisy chain is modified by forcing High the IEO of any interrupt that has not yet been acknowledged. Thus the daisy chain identifies the device presently under service as the only one with an IEI High and an IEO Low. If the next opcode byte is 4D, the interrupt-under-service latch is reset.

The ripple time of the interrupt daisy chain (both the High-to-Low and the Low-to-High transitions) limits the number of devices that can be placed in the daisy chain. Ripple time can be improved with carry-look-ahead, or by extending the interrupt-acknowledge cycle. For further information about techniques for increasing the number of daisy-chained devices, refer to the *Z8400 Z80 CPU Product Specification* (00-2001-04).

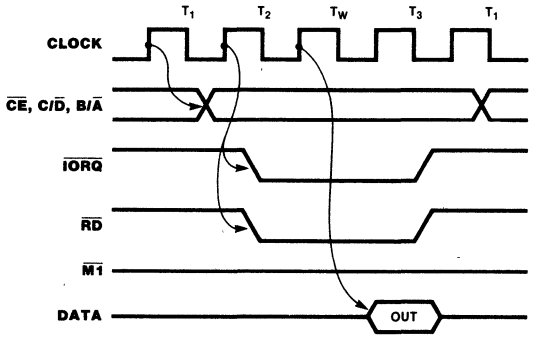


Figure 15. Read Cycle

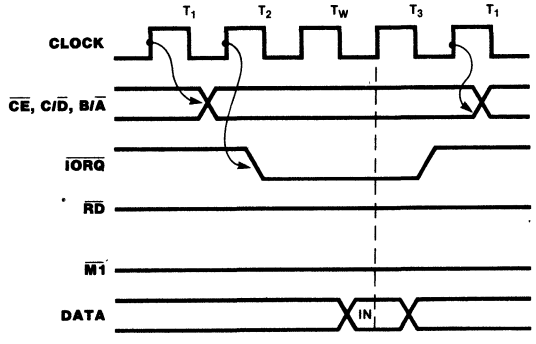


Figure 16. Write Cycle

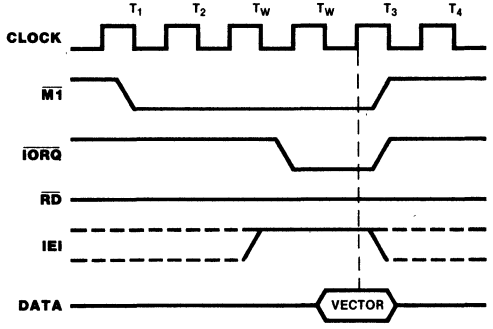


Figure 17. Interrupt Acknowledge Cycle

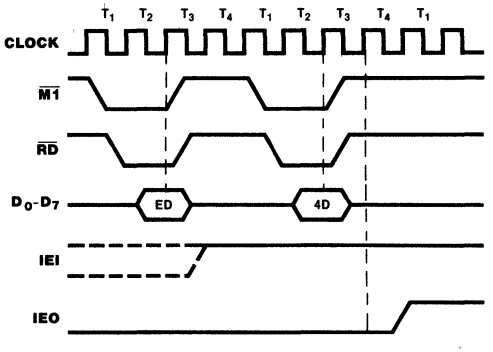


Figure 18. Return from Interrupt Cycle

ABSOLUTE MAXIMUM RATINGS

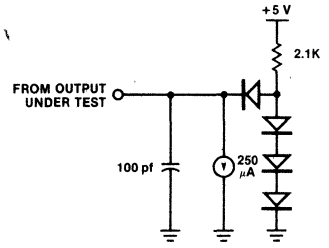
Voltages in V_{CC} with respect to V_{SS} . . . . – 0.3V to + 0.7V
Voltages on all inputs with respect to V_{SS} . . . . . – 0.3V to V_{CC} + 0.3V
Storage Temperature . . . . . – 65 °C to + 150 °C

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 these 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 characteristics below apply for the following test conditions, unless otherwise noted. All voltages are referenced to GND (0V). Positive current flows into the referenced pin Available operating temperature range is.

- **S = 0°C to +70°C, V_{CC} Range**
  NMOS: +4.75V < V_{CC} < +5.25V
  CMOS: +4.50V < V_{CC} < +5.50V
- **E = -40°C to 100°C, =4.50V < V_{CC} < +5.50V**



DC CHARACTERISTICS

Z84C40 CMOS Z80 SIO, Z84C40/41/42/43/44 DC CHARACTERISTICS

V_{CC}=5.0V ± 10%, unless otherwise specified

Symbol	Parameter	Min	Max	Typ	Unit	Condition
V_{ILC}	Clock Input Low Voltage	-0.3	+0.45		V	
V_{IHC}	Clock Input High Voltage	V_{CC}-0.6	V_{CC}+0.3		V	
V_{IL}	Input High Voltage	2.2	V_{CC}		V	
V_{IH}	Input Low Voltage	-0.3	0.8		V	
V_{OL}	Output Low Voltage		0.4		V	I_{LO}=2.0mA
V_{OH1}	Output High Voltage	2.4			V	I_{OH}=-1.6mA
V_{OH2}	Output High Voltage	V_{CC}-0.8			V	I_{OH}=-250µA
I_{LI}	Input Leakage Current	-10	10		µA	V_{IN}=0.4V to V_{CC}
I_{LO}	3-state Output Leakage Current in Float	-10	10		µA	V_{OUT}=0.4V to V_{CC}
I_L(SY)	SYNC Pin Leakage Current	-40	10		µA	
I_{CC1}	Power Supply Current - 4MHz		10 [1]	7	mA	V_{CC}=5V
	- 6MHz		10 [1]	7	mA	CLK=4,6,8,10MHz
	- 8MHz		12 [1]	8	mA	V_{IH}=V_{CC}-0.2V
	- 10MHz		15 [1]	TBD	mA	V_{IL}=0.2V
I_{CC2}	Standby Supply Current		10		µA	V_{CC}=5V CLK=(0) V_{IH}=V_{CC}-0.2V V_{IL}=0.2V

Note:
[1] Measurements made with outputs floating.

CAPACITANCE

Symbol	Parameter	Min	Max	Unit
C	Clock Capacitance		7	pf
C_{IN}	Input Capacitance		5	pf
C_{OUT}	Output Capacitance		10	pf

Over specified temperature range; f = 1 MHz.
Unmeasured pins returned to ground.

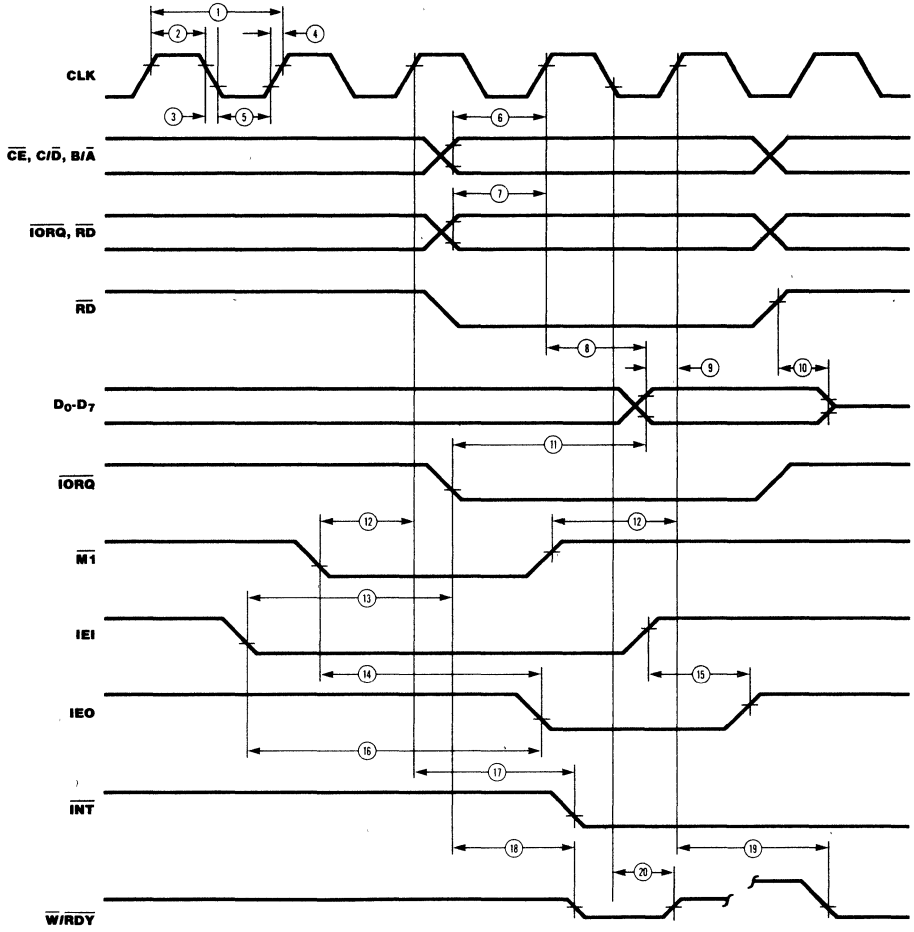
AC CHARACTERISTICS*

Z84C40/41/42/43/44 AC CHARACTERISTICS

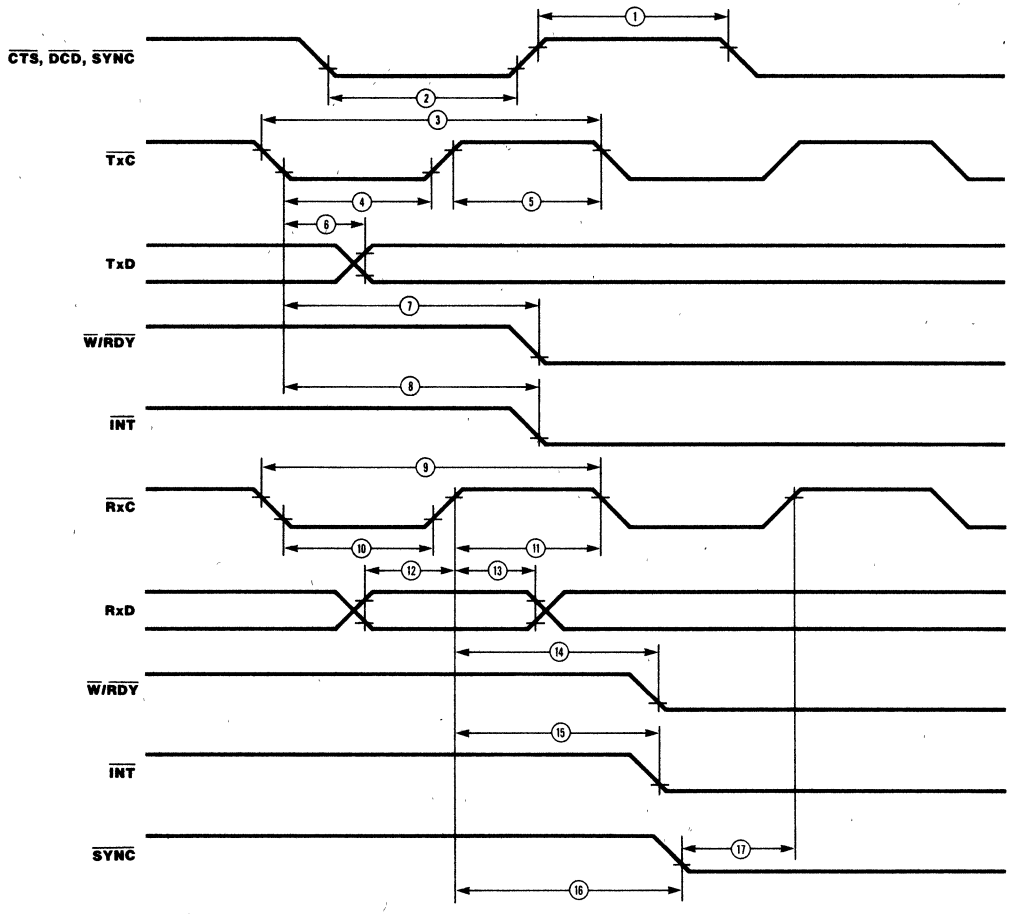
No	Symbol	Parameter	Z84C4X04 Min	Z84C4X04 Max	Z84C4X06 Min	Z84C4X06 Max	Z84C4X08 Min	Z84C4X08 Max	Z84C4X10 Min	Z84C4X10 Max	Note
1	TcC	Clock Cycle Time	250	DC	162	DC	125	DC	100	DC	
2	TwCh	Clock Pulse Width (High)	105	DC	65	DC	55	DC	42	DC	
3	TfC	Clock Fall Time		30		20		10		10	
4	TrC	Clock Rise Time		30		20		10		10	
5	TwCl	Clock Pulse Width (Low)	105	DC	65	DC	55	DC	42	DC	
6	TsAD	/CE,B//A,C//D to Clock Rise Setup Time	145		60		40		35		
7	TsCS(C)	/IORQ, /RD to Clock Rise	115		60		40		35		
8	TdC(DO)	Clock Rise to Data Out Delay		220		150		100		85	
9	TsDI(C)	Data In to Clock Rise Setup Time (Write or /M1 Cycle)	50		30		20		20		
10	TdRD(DOz)	/RD Rise to Data Out Float Delay		110		90		75		65	
11	TdIO(DOI)	/IORQ Fall to Data Out Delay (/INTACK Cycle)		160		120		90		80	
12	TsM1(C)	/M1 to Clock Rise Setup Time	90		75		55		40		
13	TsIEI(IO)	IEI to /IORQ Fall Setup Time (/INTACK Cycle)	140		120		80		60		
14	TdM1(IEO)	M1 Fall to IEO Fall Delay (Interrupt Before /M1)		190		160		130		100	
15	TdIEI(IEOr)	IEI Rise to IEO Rise Delay (After ED Decode)		100		70		60		50	
16	TdIEI(IEOf)	/M1 Fall to IEO Fall Delay		100		70		60		50	
17	TdC(INT)	Clock Rise to /INT Fall Delay		200		150		120		100	
18	TdIO(W/RWf)	/IORQ or /CE Fall to /W//RDY Delay (Wait Mode)		210		175		130		110	
19	TdC(W/RR)	Clock Rise to /W//RDY Delay (Ready Mode)	120		100		90		85		
20	TdC(W/RWz)	Clock Fall to /W//RDY Float Delay (Wait Mode) When Setup is Specified		130		110		90		80	
21	Th	Any Unspecified Hold	0		0		0		0		

Note:
* Units in nanoseconds (nS).

AC CHARACTERISTICS TIMING (Z84C4X CMOS Z80 SIO)



AC CHARACTERISTICS TIMING (Z84C4X CMOS Z80 SIO; Continued)



AC CHARACTERISTICS (Z84C4X CMOS Z80 SIO; Continued)

Z84C40/41/42/43/44 AC CHARACTERISTICS

No	Symbol	Parameter	Z84C4X04 Min	Z84C4X04 Max	Z84C4X06 Min	Z84C4X06 Max	Z84C4X08 Min	Z84C4X08 Max	Z84C4X10 Min	Z84C4X10 Max	Note
1	TwPh	Pulse Width (High)	200		200		150		150		[2]
2	TwPl	Pulse Width (Low)	200		200		150		150		[2]
3	TcTxC	/TxC Cycle Time	400		330		250		200		[2]
4	TwTxCl	/TxC Width (Low)	180		100		85		80		[2]
5	TwTxCh	/TxC Width (High)	180		100		85		80		[2]
6	TdTxC(TxD)	/TxC Fall to TxD Delay		300		220		160		120	[2]
7	TdTxC(W/RRf)	/TxC Fall to /W//RDY Fall Delay (Ready Mode)	5	9	5	9	5	9	5	9	[1]
8	TdTxC(INT)	/TxC Fall to /INT Fall Delay	5	9	5	9	5	9	5	9	[1]
9	TcRxC	/RxC Cycle Time	400		330		250		200		[2]
10	TwRxCl	/RxC Width (Low)	180		100		85		80		[2]
11	TwRxCh	/RxC Width (High)	180		100		85		80		[2]
12	TsRxD(RxC)	RxD to /RxC Setup Time (X1 Mode)	0		0		0		0		[2]
13	ThRxD(RxC)	/RxC Rise to RxD Hold Time (X1 Mode)	140		100		80		60		[2]
14	TdRxC(W/RRf)	/RxC Rise to /W//RDY Fall Delay (Ready Mode)	10	13	10	13	10	13	10	13	[1]
15	TdRxC(INT)	/RxC Rise to /INT Fall Delay	10	13	10	13	10	13	10	13	[1]
16	TdRxC(SYNC)	/RxC Rise to /SYNC Fall Delay (Output Modes)	4	7	4	7	4	7	4	7	[1]
17	TsSYNC(RxC)	/SYNC Fall to /RxC Rise Setup (External Sync Modes)	-100		-100		-100		-100		[2]

* In All Modes, the System Clock rate must be at least five times the maximum data rate. /RESET must be active a minimum of one complete clock cycle.

Notes:
[1] Units equal to System Clock Periods.
[2] Units in nanoseconds (nS).

DC CHARACTERISTICS (Z844X / NMOS Z80 SIO)

Symbol	Parameter	Min	Max	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	−0.3	+0.45	V	
V_{IHC}	Clock Input High Voltage	V_{CC} −0.6	V_{CC} +0.3	V	
V_{IL}	Input Low Voltage	−0.3	+0.8	V	
V_{IH}	Input High Voltage	+2.0	V_{CC}	V	
V_{OL}	Output Low Voltage		+0.4	V	
V_{OH1}	Output High Voltage	+2.4		V	I_{OL} = 2.0 mA
V_{OH2}	Output High Voltage			V	I_{OH} = −250 μA
I_{LI}	Input Leakage Current		±10	μA	V_{IN} = 0 4 to V_{CC}
I_{LO}	3-State Output Leakage Current in Float		±10	μA	V_{OUT} = 0.4 to V_{CC}
$I_{L(SY)}$	$\overline{SYNC}$ Pin Leakage Current		+10/−40	μA	$0 < V_{IN} < V_{CC}$
ICC_1	Power Supply Current		100	mA	

Over specified temperature and voltage range

CAPACITANCE

Symbol	Parameter	Min	Max	Unit
C	Clock Capacitance		40	pf
C_{IN}	Input Capacitance		5	pf
C_{OUT}	Output Capacitance		15	pf

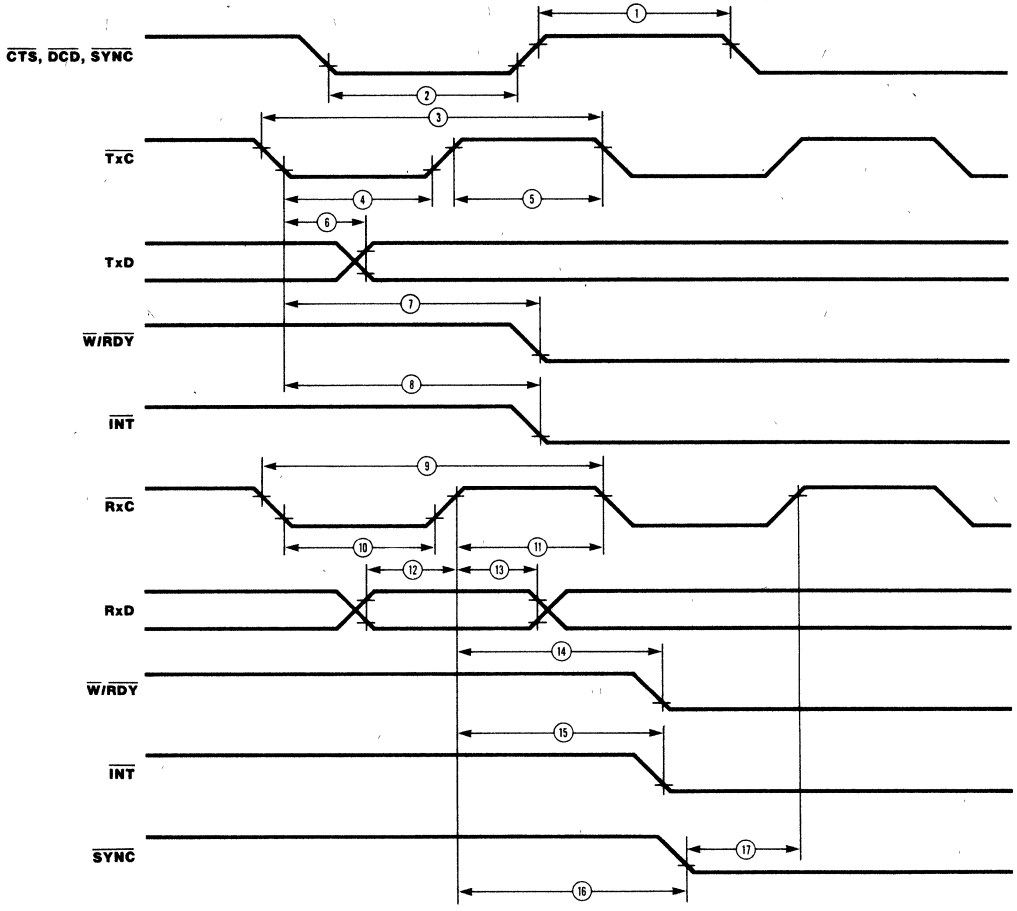
Over specified temperature range; f = 1 MHz
Unmeasured pins returned to ground.

AC CHARACTERISTICS* (Z844X / NMOS Z80 SIO)

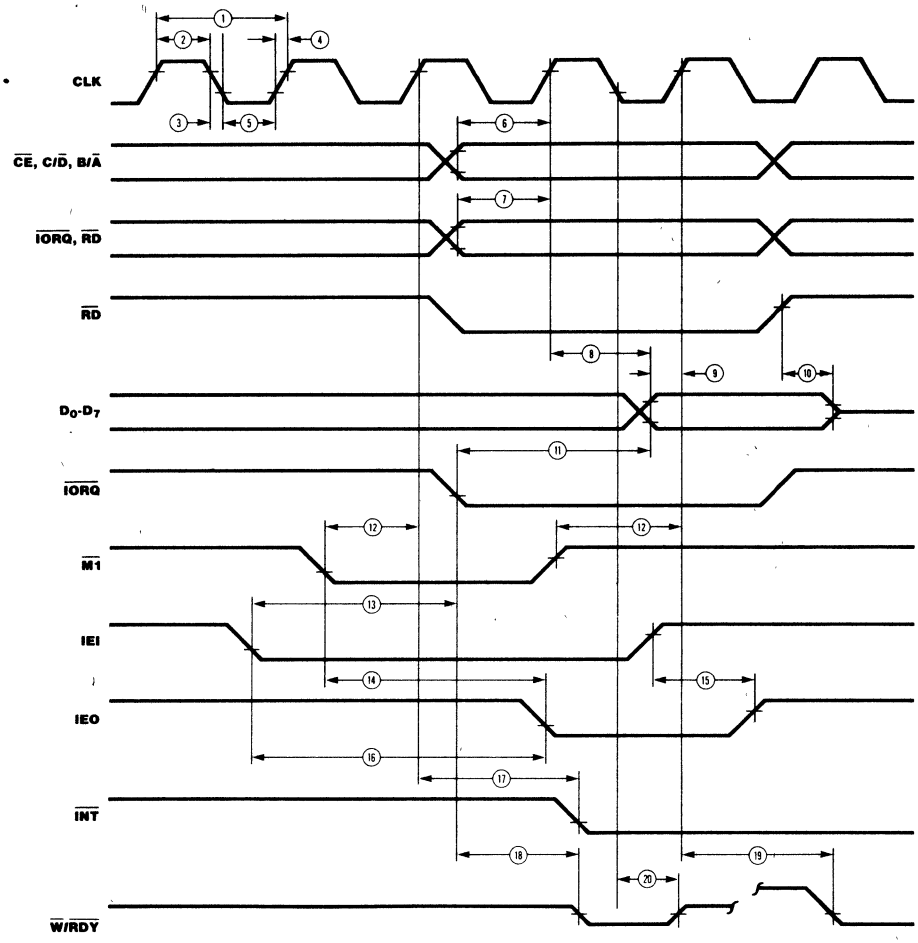
Number	Symbol	Parameter	Z0844X04 Min	Z0844X04 Max	Z0844X06 Min	Z0844X06 Max
1	TcC	Clock Cycle Time	250	4000	162	4000
2	TwCh	Clock Width (High)	105	2000	70	2000
3	TfC	Clock Fall Time		30		15
4	TrC	Clock Rise Time		30		15
5	TwCl	Clock Width (Low)	105	2000	70	2000
6	TsAD(C)	$\overline{CE}$, C/$\overline{D}$, B/$\overline{A}$ to Clock ↑ Setup Time	145		60	
7	TsCS(C)	$\overline{IORQ}$, $\overline{RD}$ to Clock ↑ Setup Time	115		60	
8	TdC(DO)	Clock ↑ to Data Out Delay		220		150
9	TsDI(C)	Data In to Clock ↑ Setup (Write or $\overline{M1}$ Cycle)	50		30	
10	TdRD(DOz)	$\overline{RD}$ ↑ to Data Out Float Delay		110		90
11	TdIO(DOI)	$\overline{IORQ}$ ↓ to Data Out Delay (INTACK Cycle)		160		120
12	TsM1(C)	$\overline{M1}$ to Clock ↑ Setup Time	90		75	
13	TsIEI(IO)	IEI to $\overline{IORQ}$ ↓ Setup Time (INTACK Cycle)	140		120	
14	TdM1(IEO)	$\overline{M1}$ ↓ to IEO ↓ Delay (interrupt before $\overline{M1}$)		190		160
15	TdIEI(IEOr)	IEI ↑ to IEO ↑ Delay (after ED decode)		100		70
16	TdIEI(IEOf)	IEI ↓ to IEO ↓ Delay		100		70
17	TdC(INT)	Clock ↑ to $\overline{INT}$ ↓ Delay		200		150
18	TdIO(W/RWf)	$\overline{IORQ}$ ↓ or $\overline{CE}$ ↓ to $\overline{W}$/$\overline{RDY}$ ↓ Delay (Wait Mode)		210		175
19	TdC(W/RRf)	Clock ↑ to $\overline{W}$/$\overline{RDY}$ ↓ Delay (Ready Mode)		120		100
20	TdC(W/RWz)	Clock ↓ to $\overline{W}$/$\overline{RDY}$ Float Delay (Wait Mode)		130		110
21	Th	Any unspecified Hold when Setup is specified	0		0	

*Units in nanoseconds (ns).

AC CHARACTERISTICS TIMING (Z844X / NMOS Z80 SIO; Continued)



AC CHARACTERISTICS TIMING (Z844X / NMOS Z80 SIO)



AC CHARACTERISTICS (Z844X / NMOS Z80 SIO; Continued)

No.	Symbol	Parameter	Z0844X04 Min	Z0844X04 Max	Z0844X06 Min	Z0844X06 Max	Notes*
1	TwPh	Pulse Width (High)	200		200		2
2	TwPl	Pulse Width (Low)	200		200		2
3	TcTxC	$\overline{TxC}$ Cycle Time	400	∞	330	∞	2
4	TwTxCl	$\overline{TxC}$ Width (Low)	180	∞	100	∞	2
5	TwTxCh	$\overline{TxC}$ Width (High)	180	∞	100	∞	2
6	TdTxC(TxD)	$\overline{TxC}$ ↓ to TxD Delay		300		220	2
7	TdTxC(W/RRf)	$\overline{TxC}$ ↓ to $\overline{W/RDY}$ ↓ Delay (Ready Mode)	5	9	5	9	1
8	TdTxC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ ↓ Delay	5	9	5	9	1
9	TcRxC	$\overline{RxC}$ Cycle Time	400	∞	330	∞	2
10	TwRxCl	$\overline{RxC}$ Width (Low)	180	∞	100	∞	2
11	TwRxCh	$\overline{RxC}$ Width (High)	180	∞	100	∞	2
12	TsRxD(RxC)	RxD to $\overline{RxC}$↑ Setup Time (x1 Mode)	0		0		2
13	ThRxD(RxC)	$\overline{RxC}$ ↑ RxD Hold Time (x1 Mode)	140		100		2
14	TdRxC(W/RRf)	$\overline{RxC}$ ↑ to $\overline{W/RDY}$ ↓ Delay (Ready Mode)	10	13	10	13	1
15	TdRxC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ ↓ Delay	10	13	10	13	1
16	TdRxC(SYNC)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ ↓ Delay (Output Modes)	4	7	4	7	1
17	TsSYNC(RxC)	$\overline{SYNC}$ ↓ to $\overline{RxC}$ ↑ Setup (External Sync Modes)	-100		-100		2

*In all modes, the System Clock rate must be at least five times the maximum data rate. $\overline{RESET}$ must be active a minimum of one complete clock cycle.

1. Units equal to System Clock Periods.

2. Units in nanoseconds (ns).

ZiLOG

PRELIMINARY PRODUCT SPECIFICATION

Z85C80

SCSCI SERIAL COMMUNICATIONS AND SMALL COMPUTER INTERFACE

FEATURES

- Low power CMOS
- Two independent, 0 to 2.5M bit/second, full-duplex channels, each with a separate crystal oscillator, baud rate generator, and Digital Phase-Locked Loop for clock recovery.
- Multi-protocol operation under program control; programmable for NRZ, NRZI, or FM data encoding.
- Asynchronous mode with five to eight bits and one, one and one-half, or two stop bits per character, programmable clock factor, break detection and generation; parity, overrun, and framing error detection.
- Synchronous mode with internal or external character synchronization on one or two synchronous characters and CRC generation and checking with CRC-16 or CRC-CCITT preset to either 1s or 0s.
- SDLC/HDLC mode with comprehensive frame-level control, automatic zero insertion and deletion, I-field residue handling, abort generation and detection, CRC generation and checking, and SDLC Loop mode operation.
- Local Loopback and Auto Echo modes.
- Supports T1 digital trunk.
- Enhanced DMA support
  10 X 19-bit status FIFO
  14-bit byte counter
- Arbitration Support
- DMA or Programmed I/O Data Transfers
- Supports Normal or Block Mode DMA
- Memory or I/O Mapped CPU Interface.
- Asynchronous Interface, Supports 3 MB/s
- Direct SCSI Bus Interface with On-Board 48 mA Drivers
- Supports Target and Initiator Roles

GENERAL DESCRIPTION

The Z85C80 CMOS SCSCI is an industry standard 85C30 dual channel Serial Communication Controller (SCC) and an industry standard 53C80 Small Computer System Interface (SCSI) integrated into one monolithic Integrated Circuit. The internal SCC and SCSI share the 8-bit data bus (D7 through D0) and read and write inputs (/RD and /WR).

Note: 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); /N//S (NORMAL and SYSTEM are both active Low).

The Z85C80 is offered in a 68-pin PLCC package. With a few exceptions, all of the internal SCC and SCSI signals are connected to the outside pins.

GENERAL DESCRIPTION (Continued)

The exceptions are:

- IEI input to SCC is internally connected to VDD.
- IEO output from SCC is not internally connected (N/C).
- READY output from SCSI is not internally connected (N/C).
- /SYNCB output from the SCC is not internally connected (N/C).
- /TRXCA and /CTSA inputs to the SCC are internally connected.
- /TRXCB and /CTSB inputs to the SCC are internally connected.

The internal SCC is a dual channel, multi-protocol data communications peripheral that easily interfaces to CPUs with non-multiplexed address/data buses. The programming flexibility of the internal registers allows the SCC to be configured to satisfy a wide variety of serial communications applications. The many on-chip features such as baud rate generators, digital phase locked loops, and crystal oscillators dramatically reduce the need for external logic. Additional features, including a 10x19-bit status FIFO and 14-bit byte counter, were added to support high speed SDLC transfers using DMA controllers.

The SCC handles asynchronous formats, synchronous byte-oriented protocols such as IBM Bisync, and synchronous bit-oriented protocols such as HDLC and IBM SDLC. The internal SCC can generate and check CRC codes in any synchronous mode and can be programmed to check data integrity in various modes. It also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls are used for general-purpose I/O. The daisy-chain interrupt hierarchy is also supported and is standard for Zilog peripheral components.

The internal SCSI is designed to implement the SCSI protocol as defined by the ANSI X3.131-1986 standard, and is fully compatible with the industry standard 53C80. It is capable of operating both as a target and as an initiator. Special high current open-drain outputs enable it to directly interface to, and drive, the SCSI bus. The internal SCSI has the necessary interface hook-ups so the system CPU can communicate with it like with any other peripheral device. The CPU can read from, or write to, the SCSI registers which are addressed as standard or memory-mapped I/Os.

The Internal SCSI increases the system performance by minimizing the CPU intervention in DMA operations which the SCSI controls. The CPU is interrupted by the SCSI when it detects a bus condition that requires attention. It also supports arbitration and reselection. The internal SCSI has the proper hand-shake signals to support normal and block mode DMA operations with most DMA controllers available.

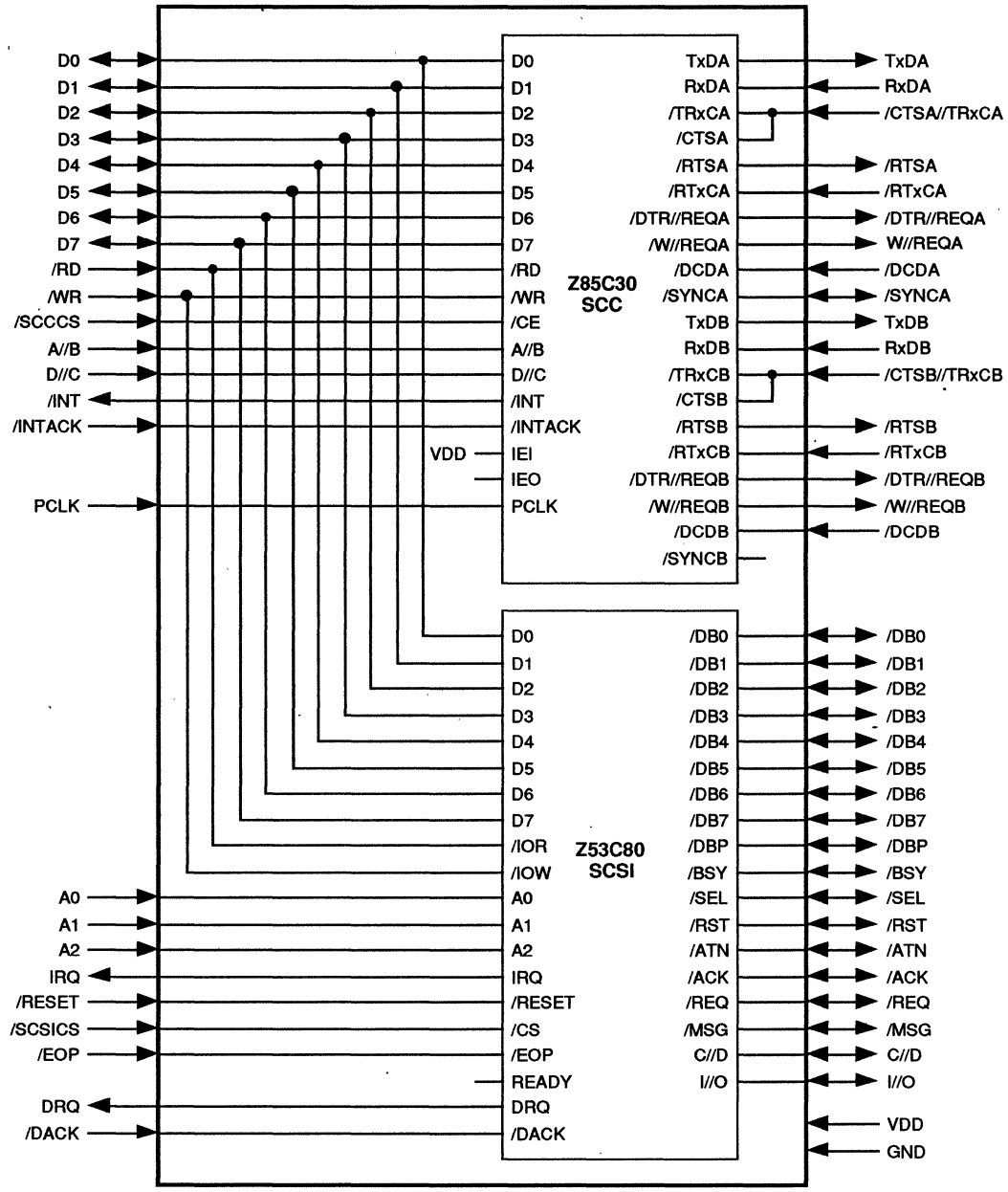
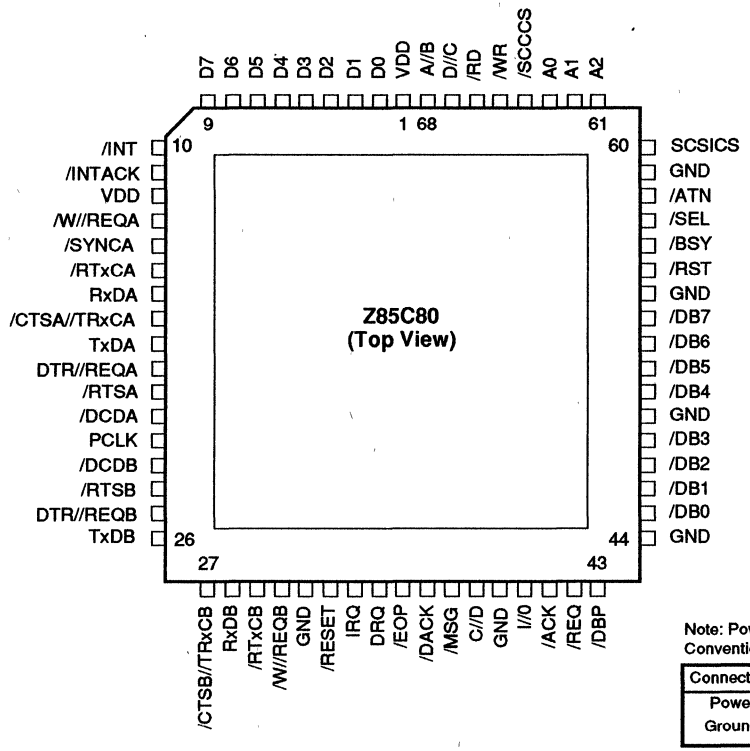


Figure 1. Z85C80 SCSCI Block Diagram



Note: Power connections follow Conventional descriptions below

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

Figure 2. 68-Pin PLCC Pin Diagram

PIN DESCRIPTION

Signal	Pin	Type	Description
A0	63	I	**SCSI Address Line Bit 0** (SCSI). Address lines are used with /SCSICS, /RD, or /WR to address all internal registers.
A1	62	I	**SCSI Address Line Bit 1** (SCSI). Address lines are used with /SCSICS, /RD, or /WR to address all internal registers.
A2	61	I	**SCSI Address Line Bit 2** (SCSI). Address lines are used with /SCSICS, /RD, or /WR to address all internal registers.
A//B	68	I	**Channel A/Channel B** (SCC) This signal selects the SCC channel in which the read or write operation occurs.
/ACK	41	I/O	**Acknowledge** (open-drain, active low, SCSI). Driven by an Initiator, /ACK indicates an acknowledgement for a /REQ//ACK data-transfer handshake. In the Target role, /ACK is received as a response to the /REQ signal.
/ATN	58	I/O	**Attention** (open-drain, active low, SCSI). Driven by an Initiator, received by the Target. /ATN indicates an Attention condition

PIN DESCRIPTION (Continued)

Signal	Pin	Type	Description
/BSY	56	I/O	**Busy** (open-drain, active low, SCSI). This signal indicates that the SCSI bus is being used and can be driven by both the Initiator and the Target device.
C//D	38	I/O	**Control/Data** (open-drain, SCSI). Driven by the Target and received by the Initiator. C//D indicates whether Control or Data information is on the Data Bus. True indicates control.
/CTSA//TRXCA	17	I	**Clear To Send for channel A; Transmit/Receive Clock for channel A** (active low, SCC). This pin is internally connected to SCC's A Channel /CTS and /TRXC. Receive clock or the transmit clock is supplied via this pin to the SCC's A channel. When programmed as Auto Enables, a low on this pin enables the A-channel transmitter.
/CTSB//TRXCB	27	I	**Clear To Send for channel B/Transmit/Receive Clock for channel B** (active low, SCC). This pin is internally connected to SCC's B-channel /CTS and /TRXC. Receive clock or the transmit clock is supplied via this pin to the SCC's B channel. When programmed as Auto Enables, a low on this pin enables the B-channel transmitter.
D0	2	I/O	**Data bus bit 0** (tri-state, active high, SCC and SCSI). This is the Least Significant Bit of the bus. Data bus lines carry data and commands to and from the SCSCI.
D1	3	I/O	**Data bus bit 1** (tri-state, active high, SCC and SCSI). Data bus lines carry data and commands to and from the SCSCI.
D2	4	I/O	**Data bus bit 2** (tri-state, active high, SCC and SCSI). Data bus lines carry data and commands to and from the SCSCI.
D3	5	I/O	**Data bus bit 3** (tri-state, active high, SCC and SCSCI). Data bus lines carry data and commands to and from the SCSCI.
D4	6	I/O	**Data bus bit 4** (tri-state, active high, SCC and SCSI). Data bus lines carry data and commands to and from the SCSCI.
D5	7	I/O	**Data bus bit 5** (tri-state, active high, SCC and SCSI). Data bus lines carry data and commands to and from the SCSCI.
D6	8	I/O	**Data bus bit 6** (tri-state, active high, SCC and SCSI). Data bus lines carry data and commands to and from the SCSCI.
D7	9	I/O	**Data bus bit 7** (tri-state, active high, SCC and SCSI). This is the most significant bit of the bus. Data bus lines carry data and commands to and from the SCSCI.
/DACK	36	I	**DMA Acknowledge** (active low, SCSI). /DACK resets DRQ and selects the data register for input or output data transfers. /DACK is used by DMA controller instead of /SCSICS.
/DBO	45	I/O	**SCSI Data Bus bit 0** (open-drain, active low, SCSI). Least significant bit in the SCSI data bus.
/DB1	46	I/O	**SCSI Data Bus bit 1** (open-drain, active low, SCSI).
/DB2	47	I/O	**SCSI Data Bus bit 2** (open-drain, active low, SCSI).

PIN DESCRIPTION (Continued)

Signal	Pin	Type	Description
/DB3	48	I/O	**SCSI Data Bus bit 3** (open-drain, active low, SCSI).
/DB4	50	I/O	**SCSI Data Bus bit 4** (open-drain, active low, SCSI).
/DB5	51	I/O	**SCSI Data Bus bit 5** (open-drain, active low, SCSI).
/DB6	52	I/O	**SCSI Data Bus bit 6** (open-drain, active low, SCSI).
/DB7	53	I/O	**SCSI Data Bus bit 7** (open-drain, active low, SCSI). This is the most significant bit in the SCSI data bus.
/DBP	43	I/O	**SCSI Data Bus Parity bit** (open-drain, active low, SCSI). Data parity is odd. Parity is always generated and optionally checked. Parity is not valid during Arbitration.
D//C	67	I	**Data/Control Select** (SCC). This signal defines the type of information transferred to and from the SCC.
/DCDA	21	I	**Data Carrier Detect for A channel** (active low, SCC). This pin functions as receive enable if it is programmed for Auto Enable; otherwise, it may be used as general-purpose input pin The SCC detects pulses on this pin and can interrupt the CPU on both logic level transactions.
/DCDB	23	I	**Data Carrier Detect for B channel** (active low, SCC). This pin functions as receive enable if it is programmed for Auto Enable; otherwise, it may be used as general-purpose input pin. The SCC detects pulses on this pin and can interrupt the CPU on both logic level transactions.
DRQ	34	0	**DMA Request** (active high, SCSI). DRQ indicates that the data register is ready to be read or written. DRQ is asserted only if DMA mode is set in the Command Register. DRQ is cleared by /DACK.
/DTR//REQA	19	0	**Data Terminal Ready/Request for channel A** (active low, SCC). This output follows the state programmed into the DTR bit. It can also be used as general-purpose output or as Request line for a DMA controller.
/DTR//REQB	25	0	**Data Terminal Ready/Request for channel B** (active low, SCC). This output follows the state programmed into the DTR bit. It can also be used as general-purpose output or as Request line for a DMA controller.
/EOP	35	I	**End of process** (active low, SCSI). EOP is used to terminate a DMA transfer. If asserted during a DMA cycle, the current byte will be transferred, but no additional bytes will be requested.
GND	31, 39,44, 45,54, 59	S	**Ground supply** (SCC and SCSI).
/INT	10	0	**SCC Interrupt Request** (open-drain, active low, SCC). This signal is activated when the SCC requests an interrupt.

PIN DESCRIPTION (Continued)

Signal	Pin	Type	Description
/INTACK	11	I	**Interrupt acknowledge** (active low, SCC). This signal indicates an active Interrupt Acknowledge cycle. /INTACK is latched by the rising edge of PCLK.
I/O	40	I/O	**Input/Output** (open-drain, SCSI). I/O is a signal driven by a Target which controls the direction of data movement on the SCSI bus. TRUE indicates input to the Initiator. This signal is also used to distinguish between Selection and Reselection phases.
IRQ	33	0	**SCSI Interrupt Request** (active high, SCSI). This signal alerts a microprocessor of an error condition or an event completion
/MSG	37	I/O	**Message** (open-drain, SCSI). This signal is driven by the Target during the Message phase. This signal is received by the Initiator.
PCLK	22	I	**Clock** (SCC). This is the master SCC clock used to synchronize internal signals. PCLK is not required to have any phase relationship with the master system clock.
/RD	66	I	**Read** (active low, SCC and SCSI). When the SCC is selected, it enables the SCC's bus drivers. When the SCSI is selected, it is used in conjunction with /SCSICS and A2-A0 to read an internal register. It also selects the Input Data Register in SCSI when used with /DACK.
/REQ	42	I/O	**Request** (open-drain, active low, SCSI). Driven by a Target and received by the Initiator, this signal indicates a request for a /REQ/ /ACK data-transfer handshake.
/RESET	32	I	**SCSI Reset** (active low, SCSI). This signal clears all registers in the SCSI. It has no effect upon the SCSI /RST signal.
/RST	55	I/O	**SCSI bus Reset** (open-drain, active low, SCSI). This signal indicates a SCSI bus Reset condition.
/RTSA	20	0	**Request To Send for channel A** (active low, SCC). When the RTS bit in Write Register 5 is set, the /RTS signal goes low. When the RTS bit is reset in the Asynchronous mode and Auto Enable is on, the signal goes high after the transmitter is empty. In Synchronous mode or in Asynchronous mode with Auto Enable off, the /RTS pin strictly follows the state of the RTS bit. This pin can be used as a general-purpose output.
/RTSB	24	0	**Request To Send for channel B** (active low, SCC). When the RTS bit in Write Register 5 is set, the /RTS signal goes low. When the RTS bit is reset in the Asynchronous mode and Auto Enable is on, the signal goes high after the transmitter is empty. In Synchronous mode or in Asynchronous mode with Auto Enable off, the /RTS pin strictly follows the state of the RTS bit. This pin can be used as a general-purpose output.

PIN DESCRIPTION (Continued)

Signal	Pin	Type	Description
/RTXCA	15	I	**Receive/Transmit Clock for channel A** (active low, SCC). This pin can be programmed in several modes of operation. It may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the digital phase-locked loop. This pin can also be programmed for use with the /SYNCA pin as a crystal oscillator. The receive clock can be 1, 16, 32, or 64 times the data rate in Asynchronous modes.
/RTXCB	29	I	**Receive/Transmit Clock for channel B** (active low, SCC). This pin can be programmed in several modes of operation. It may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the digital phase-locked loop. The receive clock can be 1, 16, 32, or 64 times the data rate in Asynchronous modes.
RXDA	16	I	**Receive Data for channel A** (active high, SCC). This input signal receives serial data.
RXDB	28	I	**Receive Data for channel B** (active high, SCC). This input signal receives serial data.
/SCCCS	64	I	**SCC Chip Select** (active low, SCSI). This signal selects SCC for a read or write operation.
/SCSICS	60	I	**SCSI Chip Select** (active low, SCSI). This signal, in conjunction with /RD or /WR, enables the internal register selected by A2-A0, to be read from or written to.
/SEL	57	I/O	**Select** (open-drain, active low, SCSI). This signal is used by an Initiator to select a Target, or by a Target to reselect an Initiator.
/SYNCA	14	I/O	**Synchronization for channel A** (active low, SCC). This pin can act as input, output, or part of the crystal oscillator circuit.
TXDA	18	O	**Transmit Data for channel A** (active high, SCC). This output signal transmits serial data at standard TTL levels.
TXDB	26	O	**Transmit Data for channel B** (active high, SCC). This output signal transmits serial data at standard TTL levels.
VDD	1,12	S	**VDD supply** (SCC and SCSI).
/WR	65	I	**Write** (active low, SCC and SCSI). When the SCC is selected, this signal indicates a write operation. The coincidence of /RD and /WR is interpreted as a reset. When the SCSI is selected, it is used in conjunction with /SCSICS and A2-A0 to write an internal register. It also selects the Output Data Register in SCSI, when used with /DACK.
/W//REQA	13	O	**Wait/Request for channel A** (open-drain when programmed for a Wait function, driven high or low when programmed for a Request function, SCC). This dual purpose output may be programmed as request line for a DMA controller or as a Wait line to synchronize the CPU to the SCC data rate. The reset state is Wait.

PIN DESCRIPTION (Continued)

Signal	Pin	Type	Description
/W//REQB	30	O	**Wait/Request for channel B** (open-drain when programmed for a Wait function, driven high or low when programmed for a Request function, SCC). This dual purpose output may be programmed as request line for a DMA controller or as a Wait line to synchronize the CPU to the SCC data rate. The reset state is Wait.

FUNCTIONAL DESCRIPTION

The Z85C80 consists of an industry standard Z85C30 Serial Communication Controller (SCC) and an industry standard Z53C80 Small Computer System Interface (SCSI), sharing the data bus and read and write signals. With the exception of the following special configurations, the internal SCC and SCSI can be used as standard devices.

SCC Configuration

- IEI (Interrupt Enable In) is hardwired to VDD. Thus no external interrupt daisy-chain can be used.
- IEO (Interrupt Enable Out) is not bonded out. Since no daisy-chain interrupt is used, this pin is left unbonded.
- /TRXC and /CTS are connected together in each of the two channels to form /CTS//TRXC. In this configuration, the pin in each channel is sued as receive or transmit clock input.
- /SYNCB (channel B Synchronization) is not bonded.

SCSI Configuration

- Data lines of the SCSI are shared with the SCC's data bus (D7 through D0 on both devices). Care must be taken not to cause bus contention by inappropriately selecting the two internal devices using their respective /CS.
- /IOR of SCSI connected to /RD of SCC to generate Z85C80's /RD pin.
- /IOW of SCSI is connected to /WR of SCC to generate Z85C80's /WR pin.
- READY (Ready) is not bonded out. READY is normally used to control the speed of Block Mode DMA transfers. It goes active to indicate the SCSI is ready to send/receive data.

SCC Functional Description

The functional capabilities of the SCC 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 SCC offers valuable features such as vectored interrupts, polling, and simple handshake capability.

Data Communications Capabilities. The SCC provides two independent full-duplex channels programmable for use in any common Asynchronous or Synchronous data communication protocol. Figure 3 and the following description briefly detail these protocols.

Asynchronous Modes. Transmission and reception can be accomplished independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one and one half, or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxDA or RxDB in Figure 1). If the Low does not persist (as in the case of a transient), the character assembly process does not start.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occur. Vectored interrupts allow fast servicing or error conditions using dedicated routines. Furthermore, a built-in checking process avoids the interpretation of a framing error as a new start bit: a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit begins.

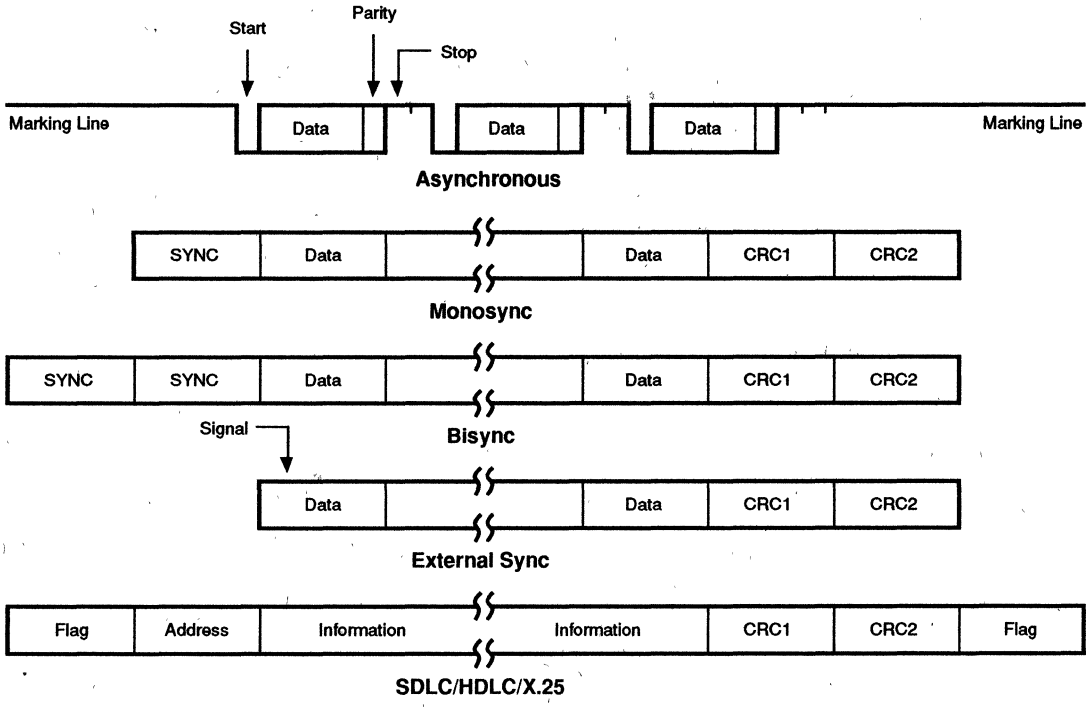


Figure 3. Some SCC Protocols

The SCC does not require symmetric transmit and receive clock signals - a feature allowing the use of a wide variety of clock sources. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32, or 1/64 of the clock rate supplied to the receive and transmit clock inputs. In Asynchronous modes, the /SYNC pin may be programmed as an input used for functions such as monitoring a ring indicator.

Synchronous Modes. The SCC supports both byte-oriented and bit-oriented synchronous communication. Synchronous byte-oriented protocols can be handled in several modes, allowing character synchronization with a 6-bit or 8-bit synchronous character (Monosync), any 12-bit synchronization pattern (Bisync), or with an external synchronous signal. Leading sync characters can be removed without interrupting the CPU.

Five- or 7-bit synchronous characters are detected with 8- or 16-bit patterns in the SCC by overlapping the larger pattern across multiple incoming synchronous characters as shown in Figure 4.

CRC checking for Synchronous byte-oriented modes is delayed by one character time so that the CPU may disable CRC checking on specific characters. This permits the implementation of protocols such as IBM Bisync.

Both CRC-16 ($X^{16} + X^{15} + X^{12} + 1$) and CCITT ($X^{16} + X^{15} + X^{12} + 1$) error checking polynomials are supported. Either polynomial may be selected in all Synchronous modes. Users may preset the CRC generator and checker to all 1s or all 0s. The SCC also provides a feature that automatically transmits CRC data when no other data is available for transmission. This allows for high speed transmission under DMA control, with no need for CPU intervention at the end of a message. When there is no data or CRC to send in Synchronous modes, the transmitter inserts 6-, 8-, or 16-bit synchronous characters, regardless of the programmed character length.

The SCC supports Synchronous bit-oriented protocols, such as SDLC and HDLC, by performing automatic flag sending, zero insertion, and CRC generation. A special command can be used to abort a frame in transmission. At

the end of a message, the SCC automatically transmits the CRC and trailing flag when the transmitter underruns. The transmitter may also be programmed to send an idle line consisting of continuous flag characters or a steady marking condition.

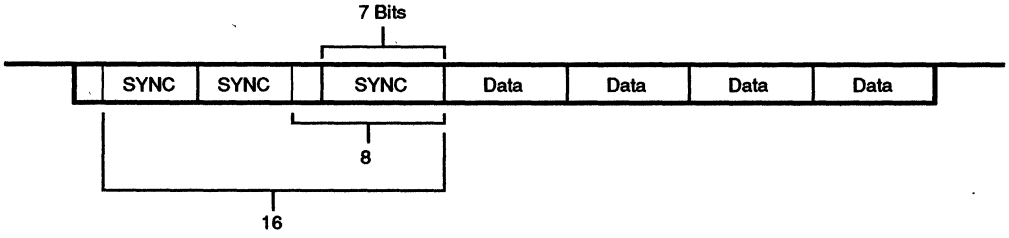


Figure 4. Detecting 5- or 7-Bit Synchronous Characters

If a transmit underrun occurs in the middle of a message, an external/status interrupt warns the CPU of this status change so that an abort may be issued. The SCC may also be programmed to send an abort itself in case of an underrun, relieving the CPU of this task. One to eight bits per character can be sent, allowing reception of a message with no prior information about the character structure in the information field of a frame.

The receiver automatically acquires synchronization on the leading flag of a frame in SDLC or HDLC and provides a synchronization signal on the /SYNC pin (an interrupt can also be programmed). The receiver can be programmed to search for frames addressed by a single byte (or four bits within a byte) of a user-selected address or to a global broadcast address. In this mode, frames not matching either the user-selected or broadcast address are ignored. The number of address bytes can be extended under software control. For receiving data, an interrupt on the first received character, or an interrupt on every character, or on special condition only (end-of-frame) can be selected. The receiver automatically deletes all 0s inserted by the transmitter during character assembly. CRC is also calculated and is automatically checked to validate frame transmission. At the end of transmission, the status of a received frame is available in the status registers. In SDLC mode, the SCC must be programmed to use the SDLC CRC polynomial, but the generator and checker may be preset to all 1s or all 0s. The CRC is inverted before transmission and the receiver checks against the bit pattern 0001110100001111.

NRZ, NRZI or FM coding may be used in any 1x mode. The parity options available in Asynchronous modes are available in Synchronous modes.

The SCC can be conveniently used under DMA control to provide high speed reception or transmission. In reception, for example, the SCC can interrupt the CPU when the first character of a message is received. The CPU then enables the DMA to transfer the message to memory. The SCC then issues an end-of-frame interrupt and the CPU can check the status of the received message. Thus, the CPU is freed for other service while the message is being received. The CPU may also enable the DMA first and have the SCC interrupt only on end-of-frame. This procedure allows all data to be transferred via the DMA.

SDLC Loop Mode. The SCC supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, there is a primary controller station that manages the message traffic flow on the loop and any number of secondary stations. In SDLC Loop mode, the SCC performs the functions of a secondary station while an SCC operating in regular SDLC mode can act as a controller (Figure 5).

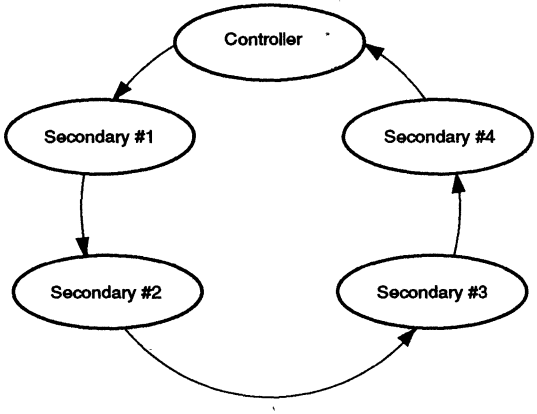


Figure 5. An SDLC Loop

FUNCTIONAL DESCRIPTION (Continued)

A secondary station in an SDLC Loop is always listening to the messages being sent around the loop, and in fact must pass these messages to the rest of the loop by re-transmitting them with a one-bit-time delay. The secondary station can place its own message on the loop only at specific times. The controller signals that secondary stations may transmit messages by sending a special character, called an EOP (End of Poll), around the loop. The EOP character is the bit pattern 11111110. Because of zero insertion during messages, this bit pattern is unique and easily recognized.

When a secondary station has a message to transmit and recognizes an EOP on the line, it changes the last binary 1 of the EOP to a 0 before transmission. This has the effect of turning the EOP into a flag sequence. The secondary station now places its message on the loop and terminates the message with an EOP. Any secondary stations further down the loop with messages to transmit can then append their messages to the message of the first secondary station by the same process. Any secondary stations without messages to send merely echo the incoming messages and are prohibited from placing messages on the loop (except upon recognizing an EOP).

SDLC Loop mode is a programmable option in the SCC. NRZ, NRZI, and FM coding may all be used in SDLC Loop mode.

Baud Rate Generator. Each channel in the SCC contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On startup, the flip-flop on the output is set in a High state, the value in the timne constant register is loaded into the counter, and the counter starts counting down. The output of the baud rate generator toggles upon reaching 0, the value in the time constant register is loaded into the counter, and the process is repeated. The time constant may be changed at any time, but the new value does not take effect until the next load of the counter.

The output of the baud rate generator may be used as either the transmit clock, the receive clock, or both. It can also drive the Digital Phase-Locked Loop (see next section).

The following formula relates the time constant to the baud rate where PCLK or /RTxC is the baud rate generator input frequency in Hz. The clock mode is 1, 16, 32, or 64 as selected in Write Register 4, bits D6 and D7. Synchronous operation modes should select 1 and Asynchronous should select 16, 32, or 64.

$$\text{Time Constant} = \frac{\text{PLCK or RTxC Frequency}}{2\ (\text{Baud Rate})\ (\text{Clock Mode})} - 2$$

Digital Phase-Locked Loop. The SCC contains a Digital Phase-Locked Loop (DPLL) to recover clock information from a data stream with NRZI or FM encoding. The DPLL is driven by a clock that is nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock may then be used as the SCC receive clock, the transmit clock, or both.

For NRZI encoding, the DPLL counts the 32x clock to create nominal bit times. As the 32x clock is counted, the DPLL is searching the incoming data stream for edges (either 1 to 0 or 0 to 1). Whenever an edge is detected, the DPLL makes a count adjustment (during the next counting cycle), producing a terminal count closer to the center of the bit cell.

For FM encoding, the DPLL still counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream should occur between counts 15 and 16 and between counts 31 and 0. The DPLL looks for edges only during a time centered on the 15 or 16 counting transition.

The 32x clock for the DPLL can be programmed to come from either the /RTxC input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the SCC via the /TRxC pin (if this pin is not being used as an input).

Data Encoding. The SCC may be programmed to encode and decode the serial data in four different ways (Figure 6). In NRZ encoding, a 1 is represented by a High level and a 0 is represented by a Low level. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level. In FM1 (more properly, bi-phase mark), a transition occurs at the beginning of every bit cell. A 1 is represented by an additional transition at the center of the bit cell and a 0 is represented by no additional transition at the center of the bit cell. In FM0 (bi-phase space), a transition occurs at the beginning of every bit cell. A 0 is represented by an additional transition at the center of the bit cell, and a 1 is represented by no additional transition at the center of the bit cell. In addition to these four methods, the SCC can be used to decode Manchester (bi-phase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0 to 1, the bit is a 0. If the transition is 1 to 0, the bit is a 1.

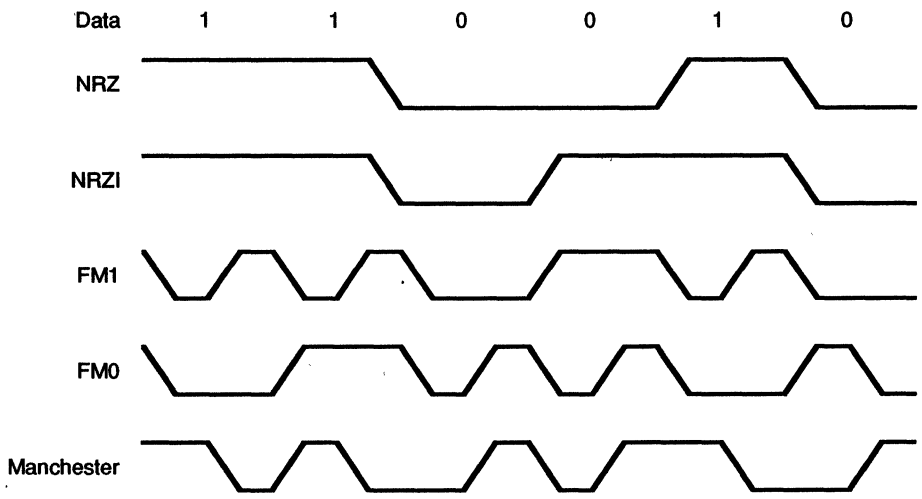


Figure 6. Data Encoding Methods

Auto Echo and Local Loopback. The SCC is capable of automatically echoing everything it receives. This feature is useful mainly in Asynchronous modes, but works in Synchronous and SDLC modes as well. In Auto Echo mode, TxD is RxD. Auto Echo mode can be used with NRZI or FM encoding with no additional delay, because the dat stream is not decoded before re-transmission. In Auto Echo mode, the /CTS input is ignored as a transmitter enable (although transitions on this input can still cause interrupts if programmed to do so). In this mode, the transmitter is actually bypassed and the programmer is responsible for disabling transmitter interrupts and /W//REQ on transmit.

The SCC is also capable of local loopback. In this mode TxD is RxD, just as in Auto Echo mode. However, in Local Loopback mode, the internal transmit data is tied to the internal receive data and RxD is ignored (except to be echoed out via TxD). The /CTS and /DCD inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local Loopback works in Asynchronous, Synchronous and SDLC modes with NRZ, NRZI, or FM coding of the data stream.

I/O Interface Capabilities. The SCC offers the choice of Polling, Interrupt (vectored or nonvectored), and Block Transfer modes to transfer data, status, and control information to and from the CPU. The Block Transfer mode can be implemented under CPU or DMA control.

Polling. All interrupts are disabled. Three status registers in the SCC are automatically updated whenever any function is performed. For example, end-of-frame in SDLC mode sets a bit in one of these status registers. The idea behind polling is for the CPU to periodically read a status register until the register contents indicate the need for data to be transferred. Only one register needs to be read; depending on its contents, the CPU either writes data, reads data, or continues. Two bits in the register indicate the need for data transfer. An alternative is a poll of the Interrupt Pending register to determine the source of an interrupt. The status for both channels resides in one register.

Interrupts

When an SCC responds to an Interrupt Acknowledge signal (/INTACK) from the CPU, an interrupt vector may be placed on the data bus. This vector is written in WR2 and may be read in RR2A or RR2B - Read Register 2, Channel A, or Channel B (Figures 9 and 10).

To speed interrupt response time, the SCC can modify three bits in this vector to indicate status. If the vector is read in Channel A, status is never included; if it is read in Channel B, status is always included.

Each of the six sources of interrupts in the SCC (Transmit, Receive, and External/Status interrupts in both channels)

has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). Operation of the IE bit is straightforward. If the IE bit is set for a given interrupt source, then that source can request interrupts. The exception is when the MIE (Master Interrupt Enable) bit in WR9 is reset and no interrupts may be requested. The IE bits are write only.

In the SCC, the IP bit signals a need for interrupt servicing. When an IP bit is 1, the /INT output is pulled Low, requesting an interrupt. In the SCC, if the IE bit is not set by enabling interrupts, then the IP for that source can never be set. The IP bits are readable in RR3A.

The IUS bits signal that an interrupt request is being serviced. There are three types of interrupts: Transmit, Receive, and External/Status. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receiver, Transmit, and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted when the transmit buffer becomes empty. (This implies that the transmitter must have had a data character written into it so that it can become empty.) When enabled, the receiver can interrupt the CPU in one of three ways.

1. Interrupt on First Receive Character or Special Receive Condition.

2. Interrupt on All Receive Characters or Special Receive Condition.

3. Interrupt on Special Receive Condition Only.

Interrupt on First Character or Special Condition and Interrupt on Special Condition Only are typically used with the Block Transfer mode. A Special Receive Condition is one of the following: receiver overrun, framing error in Asynchronous mode, end-of-frame in SDLC mode and, optionally, a parity error. The Special Receive Condition interrupt is different from an ordinary receive character available interrupt only in the status placed in the vector during the Interrupt Acknowledge cycle. In Interrupt on First Receive Character, an interrupt can occur from Special Receive Conditions any time after the first receive character interrupt.

The main function of the External/Status interrupt is to monitor the signal transactions of the /CTC//TRXC, /DCD, and /SYNC pins; however, an External/Status interrupt is also caused by a Transmit Underrun condition, or a zero count in the baud rate generator, or by the detection, or a zero count in the baud rate generator, or by the detection of a Break (Asynchronous mode), Abort (SDLC mode) or EOP (SDLC Loop mode) sequence in the data stream. The interrupt caused by the Abort or EOP has a special feature allowing the SCC to interrupt when the Abort condition in external logic in SDLC mode. In SDLC Loop mode, this feature allows secondary stations to recognize the wishes of the primary station to regain control of the loop during a poll sequence.

CPU/DMA Block Transfer. The SCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers. The Block Transfer mode uses the /W//REQ output in conjunction with the Wait/Request bits in WR1. The /W//REQ output can be defined under software control as a /W line in the CPU Block Transfer mode or as a /REQ line in the DMA Block Transfer mode.

To a DMA controller, the SCC /REQ output indicates that the SCC is ready to transfer data to or from memory. To the CPU, the /W line indicates that the SCC is not ready to transfer data, thereby requesting that the CPU extend the I/O cycle. The /DTR//REQ line allows full-duplex operation under DMA control.

ARCHITECTURE

The SCC internal structure includes two full-duplex channels, two baud rate generators, internal control and interrupt logic, and a bus interface to a nonmultiplexed bus. Associated with each channel are a number of read and write registers for mode control and status information, as well as logic necessary to interface to modems or other external devices (Figure 7).

The logic for both channels provides formats, synchronization, and validation for data transferred to and from the channel interface. The modem control inputs are monitored by the control logic under program control. All of the modem control signals are general-purpose in nature and can optionally be used for functions other than modem control.

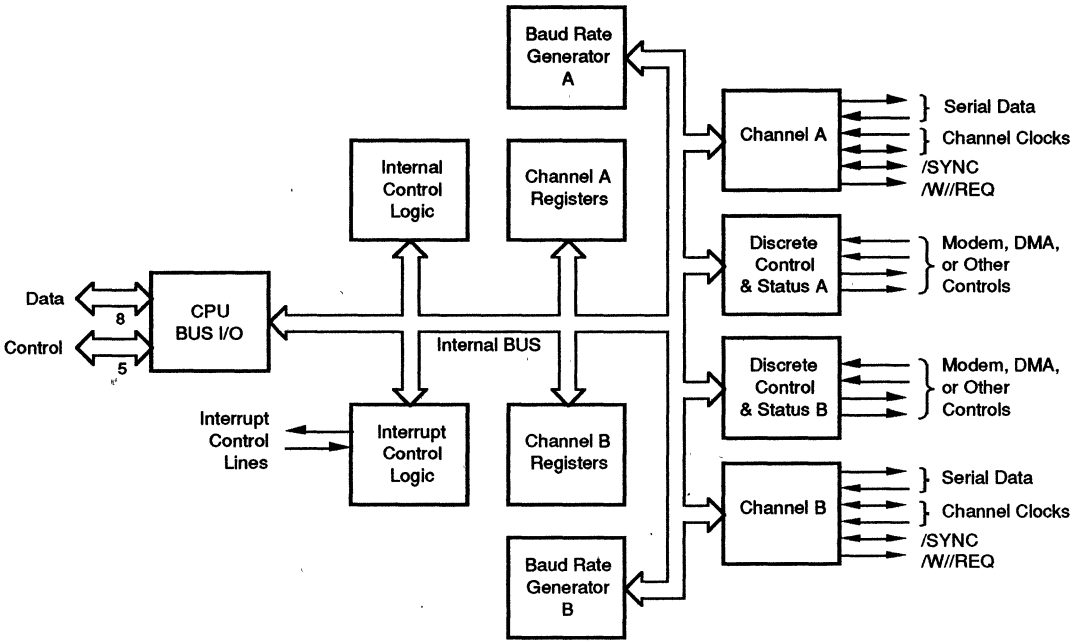


Figure 7. Block Diagram of SCC Architecture

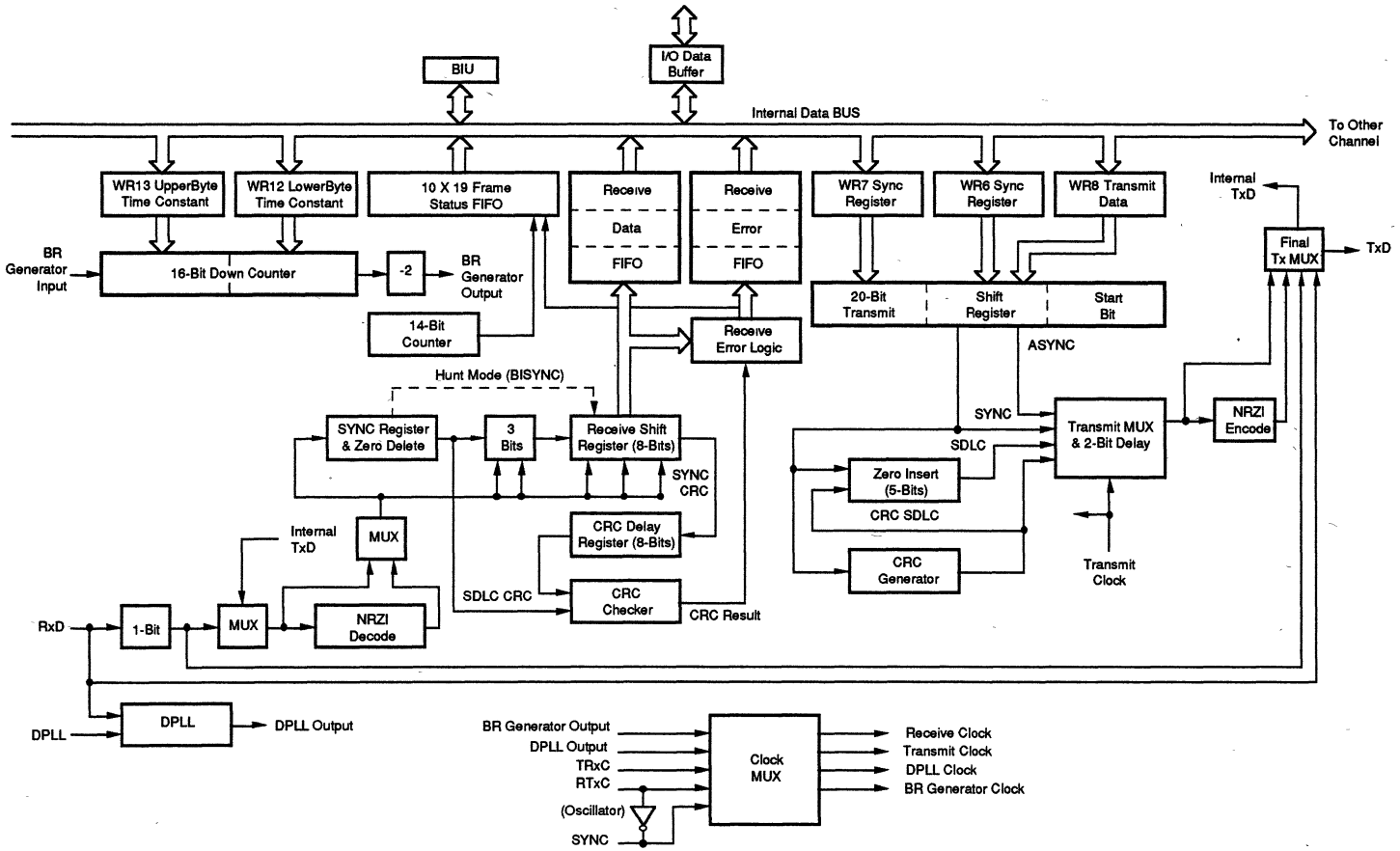


Figure 8. Data Path

ARCHITECTURE (Continued)

The register set for each channel includes ten control (write) registers, two sync-character (write) registers, and four status (read) registers. In addition, each baud rate generator has two (read/write) registers for holding the time constant that determines the baud rate. Finally, associated with the interrupt logic is a write register for the interrupt vector accessible through either channel, a write only Master Interrupt Control register and three read registers: one containing the vector with status information (Channel B only), one containing the vector without status (Channel A only), and one containing the Interrupt Pending bits (Channel A only).

The registers for each channel are designated as follows:

- WR0-WR15 - Write Registers 0 through 15.
- RR0-RR3, RR10, RR12, RR13, RR15 - Read Registers 0 through 3, 10, 12, 13, 15.

Table 1 lists the functions assigned to each read or write register. The SCC contains only one WR2 and WR9, but they can be accessed by either channel. All other registers are paired (one for each channel).

Data Path. The transmit and receive data path illustrated in Figure 8 is identical for both channels. The receiver has three 8-bit buffer registers in a FIFO arrangement, in addition to the 8-bit receive shift register. This scheme creates additional time for the CPU to service an interrupt at the beginning of a block of high speed data. Incoming data is routed through one of several paths (data or CRC) depending on the selected mode (the character length in Asynchronous modes also determines the data path).

The transmitter has an 8-bit Transmit Data buffer register loaded from the internal data bus and a 20-bit Transmit Shift register that can be loaded either from the synchronous character registers or from the Transmit Data register. Depending on the operational mode, outputting data is routed through one of four main paths before it is transmitted from the Transmit Data output (TxD).

Table 1. Read and Write Register Functions

Read Register Functions	
RR0	Transmit/Receive buffer status and External status.
RR1	Special Receive Condition status
RR2	Modified interrupt vector (Channel B only). Unmodified interrupt vector (Channel A only).
RR3	Interrupt Pending bits (Channel A only).
RR8	Receive buffer
RR10	Miscellaneous status
RR12	Lower byte of baud rate generator time constant.
RR13	Upper byte of baud rate generator time constant.
RR15	External/Status interrupt information.

Write Register Functions	
WR0	CRC initialize, initialization commands for the various modes, Register Pointers.
WR1	Transmit/Receive interrupt and data transfer mode definition.
WR2	Interrupt vector (accessed through either channel).
WR3	Receive parameters and control.
WR4	Transmit/Receive miscellaneous parameters and modes.
WR5	Transmit parameters and controls.
WR6	Sync characters or SDLC address field.
WR7	Sync character of SDLC flag.
WR8	Transmit buffer.
WR9	Master interrupt control and reset (accessed through either channel).
WR10	Miscellaneous transmitter/receiver control bits
WR11	Clock mode control.
WR12	Lower byte of baud rate generator time constant.
WR13	Upper byte of baud rate generator time constant.
WR14	Miscellaneous control bits.
WR15	External/Status interrupt control.

PROGRAMMING

The SCC contains write registers in each channel that are programmed by the system separately to configure the functional characteristics of the channels.

In the SCC, register addressing is direct for the data registers only, which are selected by a High on the D//C pin. In all other cases (with the exception of WR0 and RR0), programming the write registers requires two write operations and reading the read registers requires both a write and a read operation. The first write is to WR0 and contains three bits that point to the selected register. The second write is the actual control word for the selected register, and if the second operation is read, the selected read register is accessed.

All of the registers in the SCC, including the data registers, may be accessed in this fashion. The pointer bits are automatically cleared after the read or write operation so that WR0 (or RR0) is addressed again.

The system program first issues a series of commands to initialize the basic mode of operation. This is followed by other commands to qualify conditions within the selected mode. For example, the Asynchronous mode, character length, clock rate, number of stop bits, even or odd parity might be set first. Then the interrupt mode would be set, and finally, receiver or transmitter enable.

Read Registers. The SCC contains eight read registers (actually nine, counting the receive buffer (RR8) in each channel). Four of these may be read to obtain status information (RR0, RR1, RR10, and RR15). Two registers (RR12 and RR13) may be read to learn the baud rate generator time constant. RR2 contains either the unmodified interrupt vector (Channel A) or the vector modified by status information (Channel B). RR3 contains the Interrupt Pending (IP) bits (Channel A). Figure 9 shows the formats for each read register.

The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring; e.g., when the interrupt vector indicates a Special Receive Condition interrupt, all the appropriate error bits can be read from a single register (RR1).

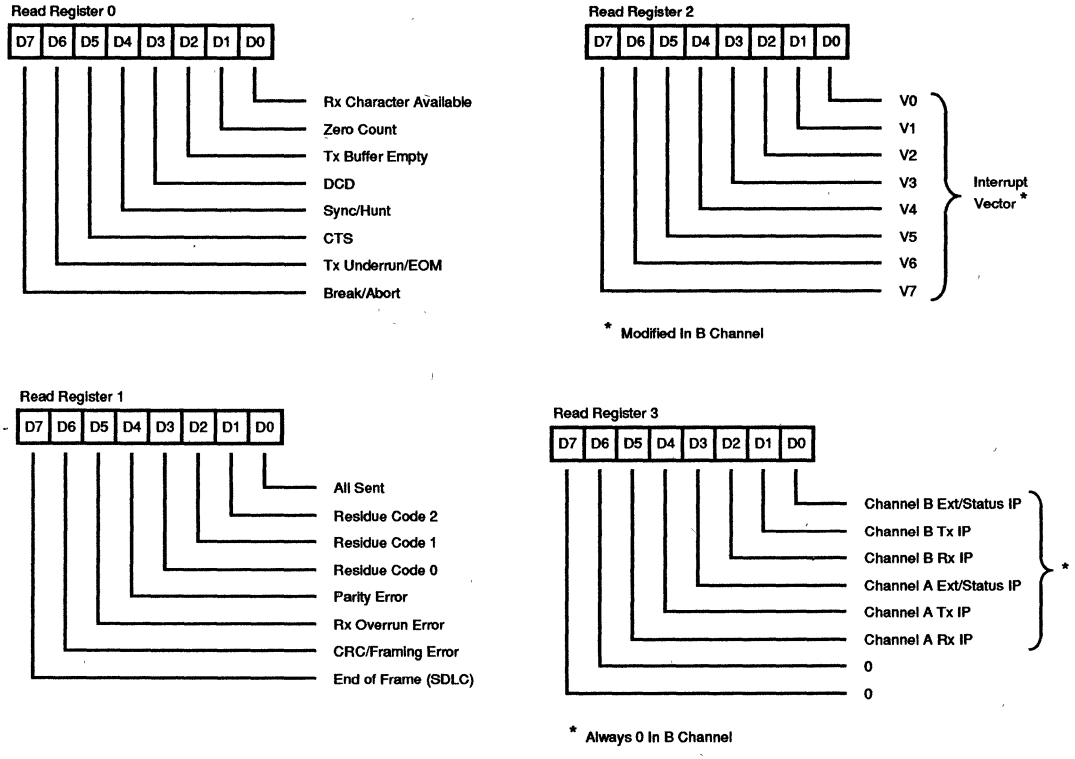


Figure 9. Read Register Bit Functions

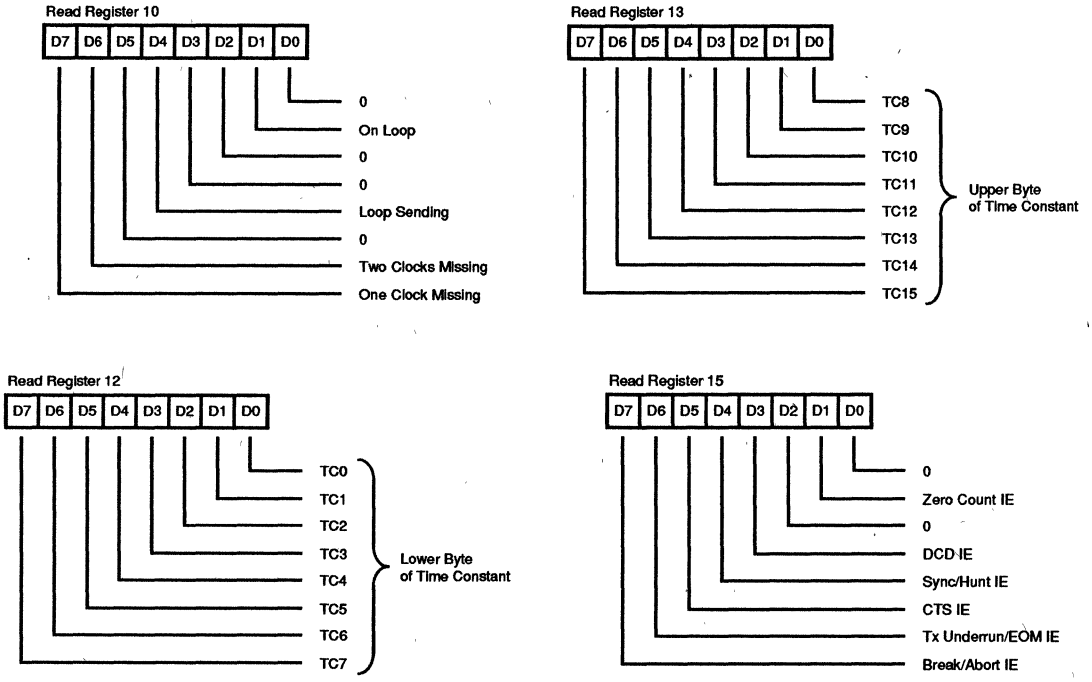


Figure 9. Read Register Bit Functions (Continued)

Write Registers. The SCC contains 13 write registers (14 counting WR8, the transmit buffer) in each channel. These write registers are programmed separately to configure the functional "personality" of the channels. In addition, there are two registers (WR2 and WR9) shared by the two channels that may be accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits. Figure 10 shows the format of each write register.

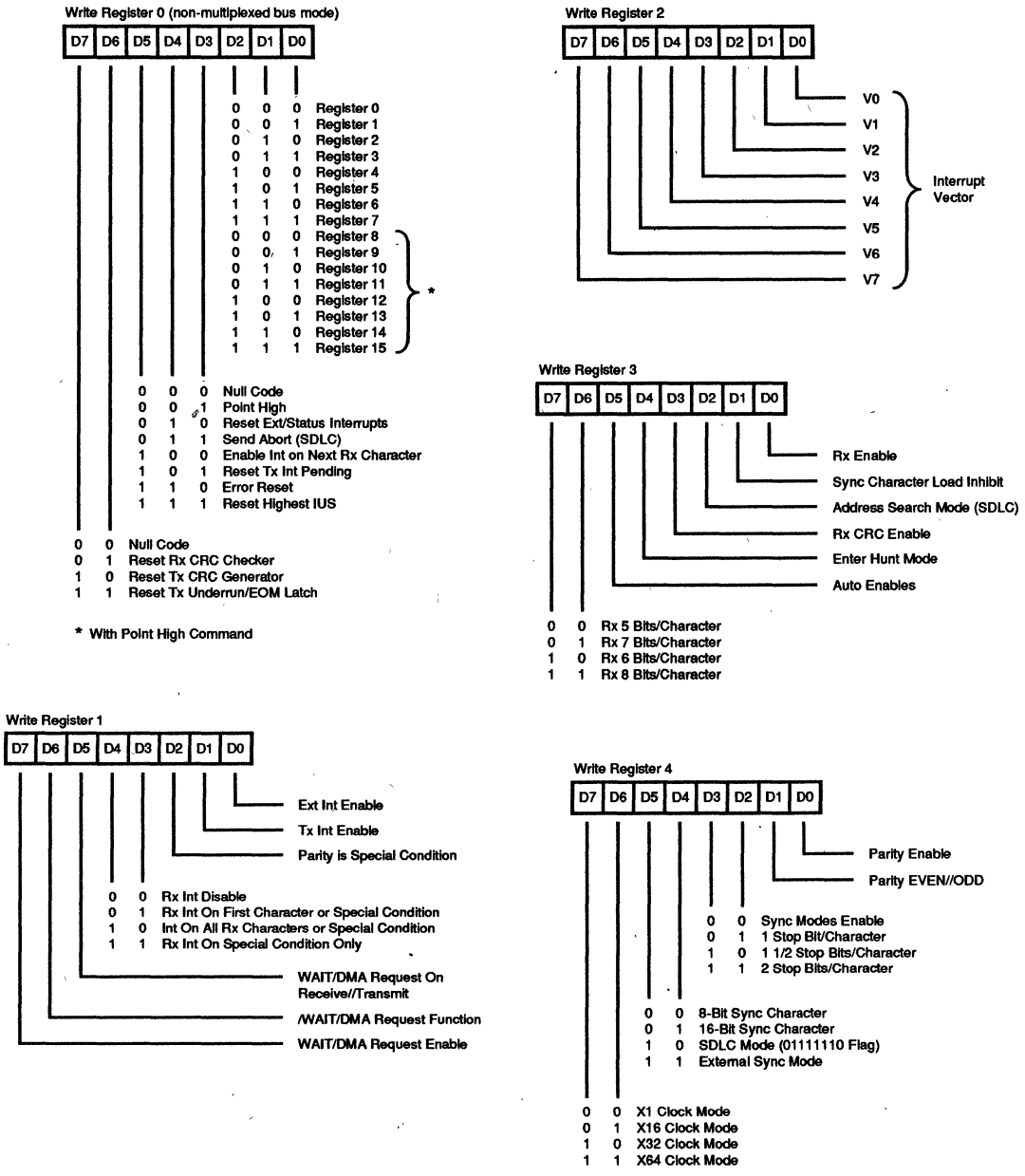


Figure 10. Write Register Bit Functions

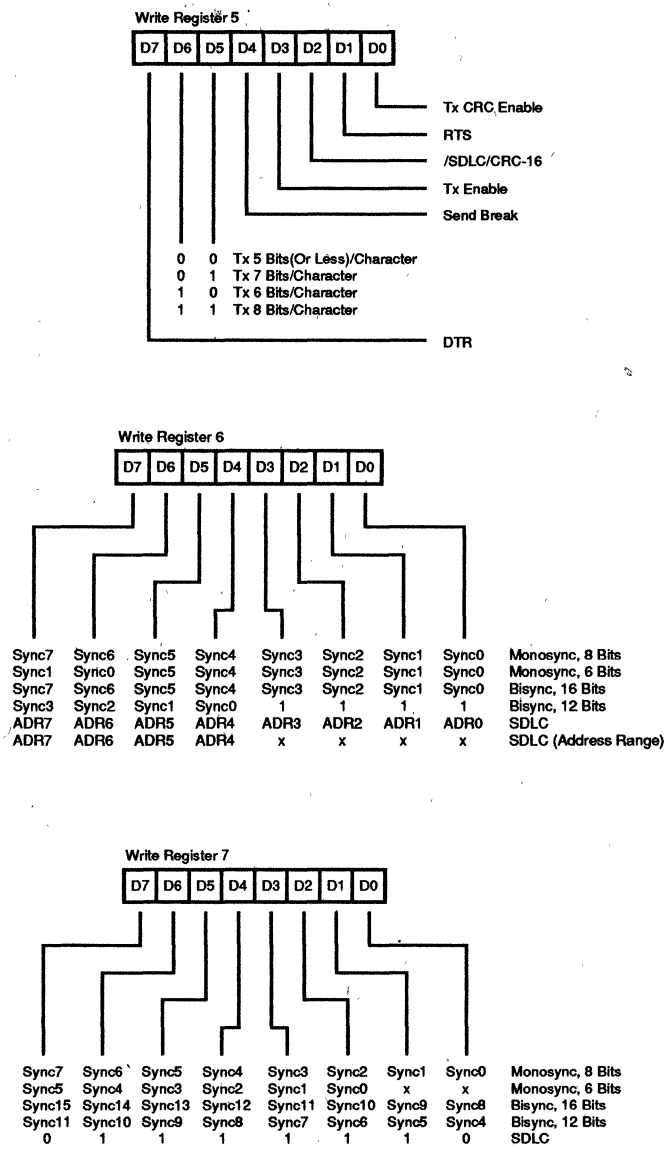


Figure 10. Write Register Bit Functions (Continued)

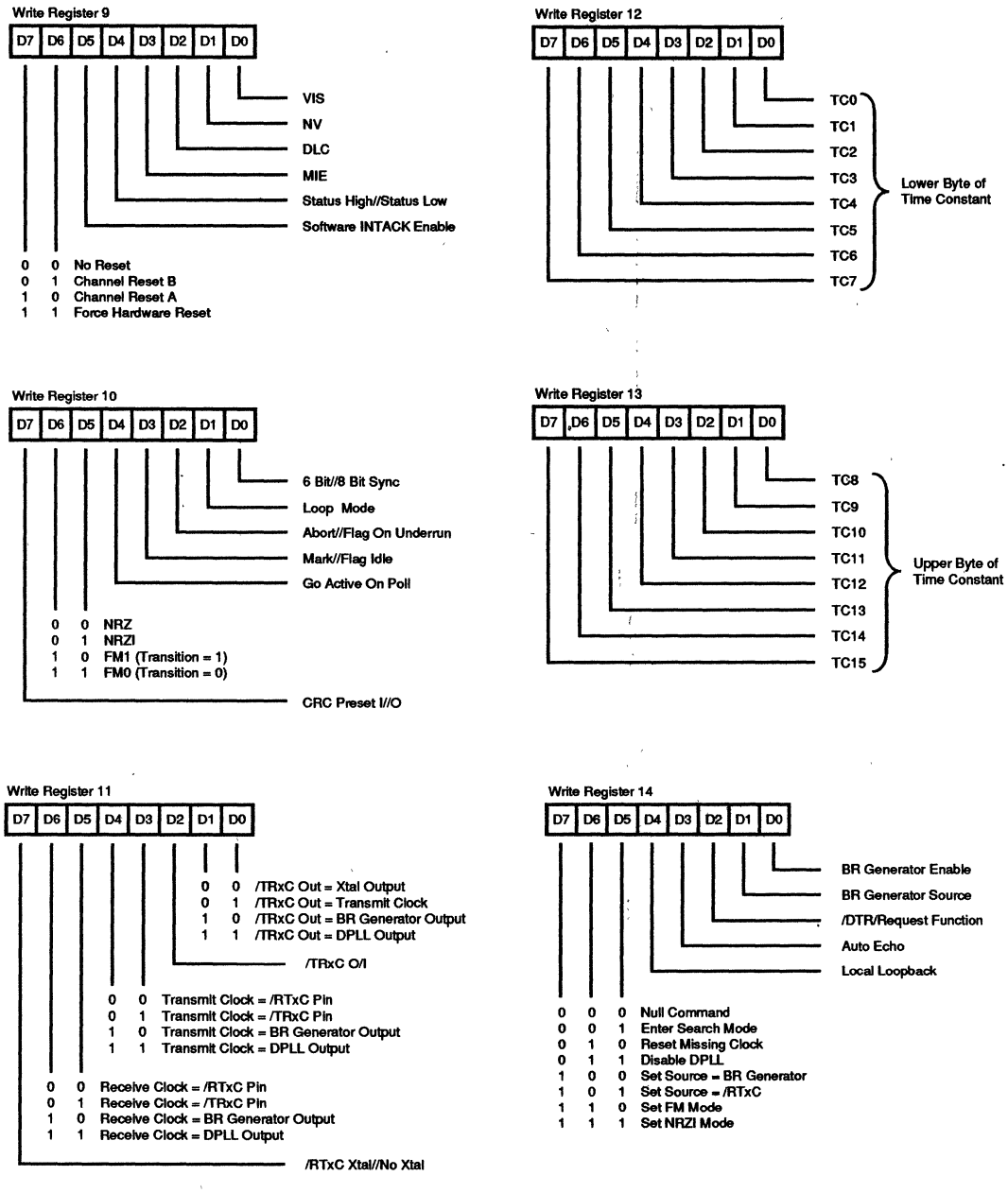


Figure 10. Write Register Bit Functions (Continued)

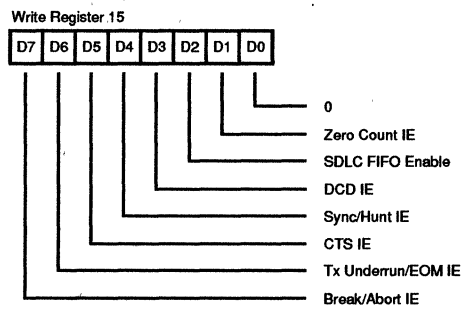


Figure 10. Write Register Bit Functions (Continued)

TIMING

The SCC generates internal control signals from /WR and /RD that are related to PCLK. Since PCLK has no phase relationship with /WR and /RD, the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the SCC. The recovery time applies only between bus transactions involving the SCC. The recovery time required for proper operation is specified from the falling edge of /WR or /RD in the first transaction involving the SCC to the falling edge of /WR or /RD in the second transaction involving the SCC. This time must be at least 4 PCLK regardless of which register or channel is being accessed.

Read Cycle Timing. Figure 11 illustrates Read cycle timing. Addresses on A//B and D//C and the status on /INTACK must remain stable throughout the cycle. If /SCCCS falls after /RD falls or if it rises before /RD rises, the effective /RD is shortened.

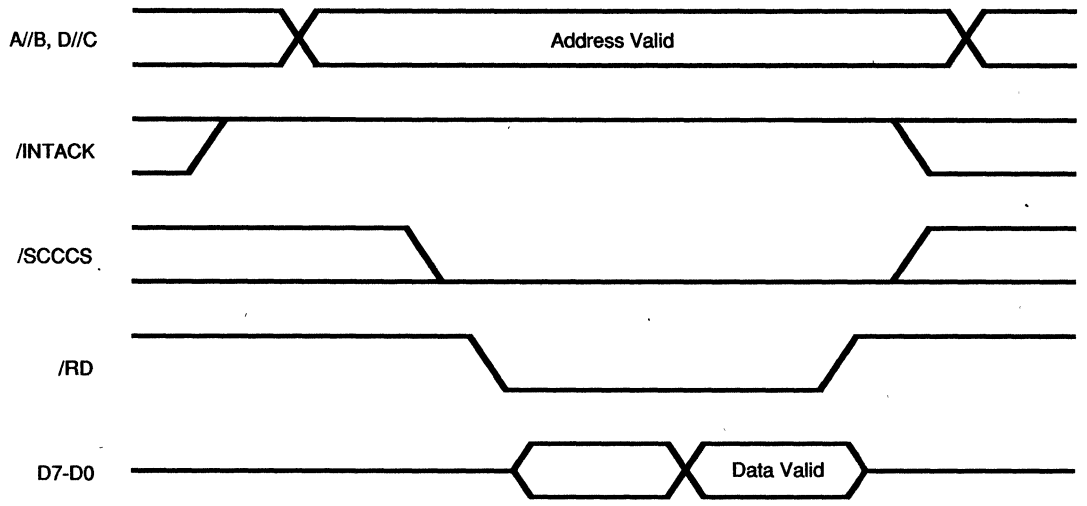


Figure 11. Read Cycle Timing

Write Cycle Timing. Figure 12 illustrates Write cycle timing. Addresses on A//B and D//C and the status on /INTACK must remain stable throughout the cycle. If /SCCCS falls after /WR falls or if it rises before /WR rises, the effective /WR is shortened. Data must be valid before the falling edge of /WR.

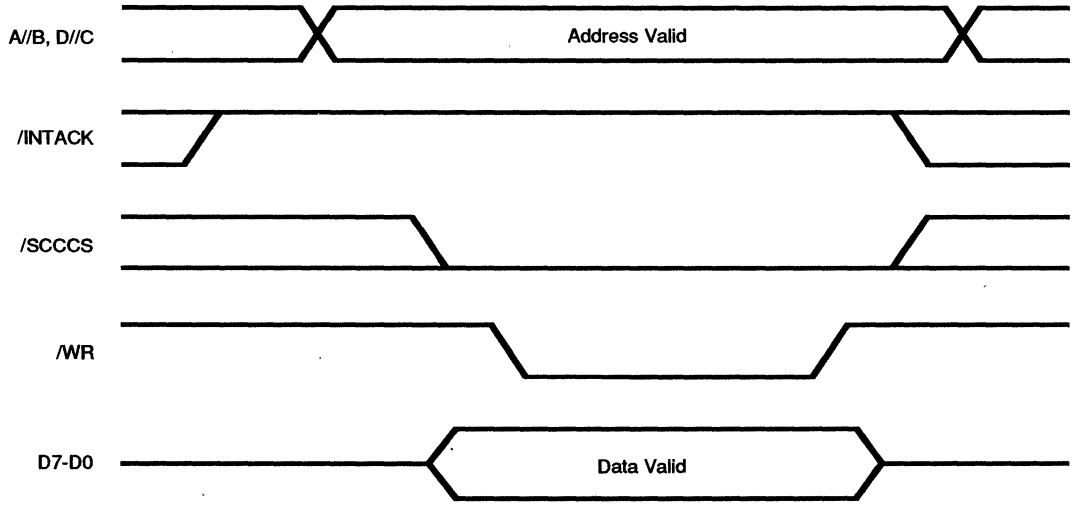


Figure 12. Write Cycle Timing

TIMING (Continued)

Interrupt Acknowledge Cycle Timing. Figure 13 illustrates Interrupt Acknowledge cycle timing.

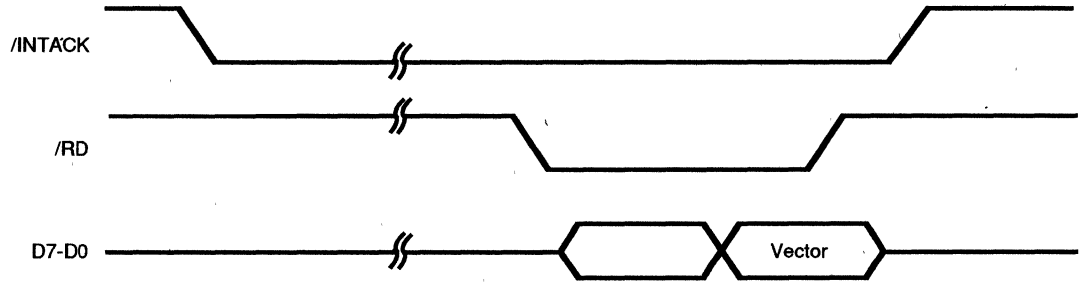


Figure 13. Interrupt Acknowledge Cycle Timing

FIFO

The following text explains the functional operations of the FIFO.

FIFO Enhancements. When used with a DMA controller, the Z85C30 FIFO enhancement maximizes the SCC's ability to receive high speed back-to-back SDLC messages while minimizing frame overruns due to CPU latencies in responding to interrupts.

Additional logic was added to the industry standard NMOS SCC consisting of a 10 deep by 19 bit status FIFO, 14-bit receive byte counter, and control logic as shown in Figure 14. The 10 x 19 bit status FIFO is separate from the existing three byte receive data FIFO.

When the enhancement is enabled, the status in read register 1 (RR1) and byte count for the SDLC frame will be stored in the 10 x 19 bit status FIFO. This allows the DMA controller to transfer the next frame into memory while the CPU verifies the message was properly received

Summarizing the operation, data is received, assembled, loaded into the three byte receive FIFO before being transferred to memory by the DMA controller. When a flag is received at the end of an SDLC frame, the frame byte count from the 14-bit counter and five status bits are loaded into the status FIFO for verification by the CPU. The CRC checker is automatically reset in preparation for the next frame which can begin immediately. Since the byte count and status are saved for each frame, the message integrity can be verified at a later time. Status information for up to 10 frames can be stored before a status FIFO overrun could occur.

FIFO Detail. For a better understanding of details of the FIFO operation, refer to the block diagram contained in Figure 14.

Enable/Disable. This FIFO is implemented so that it is enabled when WR15 bit 2 is set and the SCC is in the SDLC/HDLC mode, otherwise the status register contents bypass the FIFO and go directly to the bus interface (the FIFO pointer logic is reset either when disabled or via a channel or power-on reset). When the FIFO mode is disabled, the SCC is completely downward-compatible with the NMOS 8530. The FIFO mode is disabled on power-up (WR15 bit 2 is set to 0 on reset). The effects of backward compatibility on the register set are that RR4 is an image of RR0, RR5 is an image of RR1, RR6 is an image of RR2 and RR7 is an image of RR3 For the details of the added registers, refer to Figure 16. The status of the FIFO Enable signal can be obtained by reading RR15 bit 2. If the FIFO is enabled, the bit will be set to 1; otherwise, it will be reset.

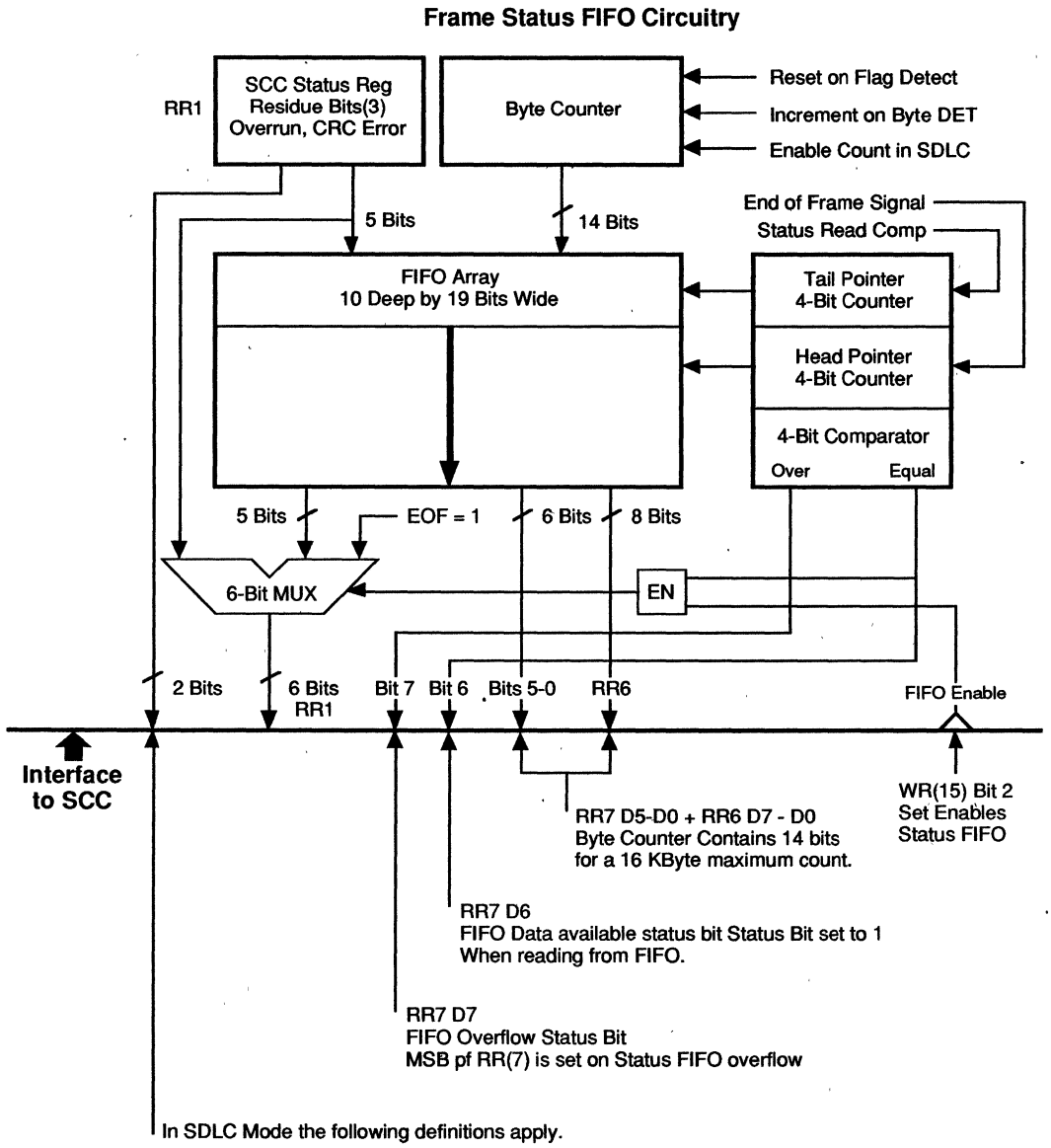


Figure 14. SCC Status Register Modifications

Read Operation. When WR15 bit 2 is set and the FIFO is not empty, the next read to any of status register RR1 or the additional registers RR7 and RR6 will actually be from the FIFO. Reading status register RR1 causes one location of the FIFO to be emptied, so status should be read after reading the byte count, otherwise the count will be incorrect. Before the FIFO underflows, it is disabled. In this case, the multiplexer is switched to allow status to read directly from the status register, and reads from RR7 and RR6 will contain bits that are undefined. Bit 6 of RR7 (FIFO Data Available) can be used to determine if status data is coming from the FIFO or directly from the status register, since it is set to 1 whenever the FIFO is not empty.

Since not all status bits must be stored in the FIFO, the All Sent, Parity, and EOF bits will bypass the FIFO. The status bits sent through the FIFO will be Residue Bits (3), Overrun, and CRC Error.

The sequence for proper operation of the byte count and FIFO logic is to read the registers in the following order: RR7, RR6, and RR1 (reading RR6 is optional). Additional logic prevents the FIFO from being emptied by multiple reads from RR1. The read from RR7 latches the FIFO empty/full status bit (bit 6) and steers the status multiplexer to read from the SCC megacell instead of the status FIFO (since the status FIFO is empty). The read from RR1 allows an entry to be read from the FIFO (if the FIFO was empty, logic is added to prevent a FIFO underflow condition).

Write Operation. When the end of an SDLC frame (EOF) has been received and the FIFO is enabled, the contents of the status and byte-count registers are loaded into the FIFO. The EOF signal is used to increment the FIFO. If the FIFO overflows, the MSB of RR7 (FIFO Overflow) is set to indicate the overflow. This bit and the FIFO control logic is reset by disabling and re-enabling the FIFO control bit (WR15 bit 2). For details of FIFO control timing during an SDLC frame, refer to Figure 15.

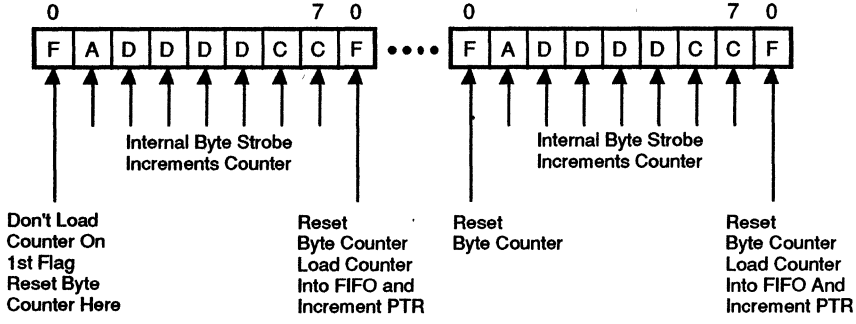


Figure 15. SDLC Byte Counting Detail

Byte Counter Detail. The 14-bit byte counter allows for packets up to 16K bytes to be received. For a better understanding of its operation refer to Figures 14 and 15.

Enable. The byte counter is enabled in the SDLC/HDLC mode.

Reset. The byte counter is reset whenever an ADLC flag character is received. The reset is timed so that the contents of the byte counter are successfully written into the FIFO.

Increment. The byte counter is incremented by writes to the data FIFO. The counter represents the number of bytes received by the SCC, rather than the number of bytes transferred from the SCC. (These counts may differ by up to the number of bytes in the receive data FIFO contained in the SCC).

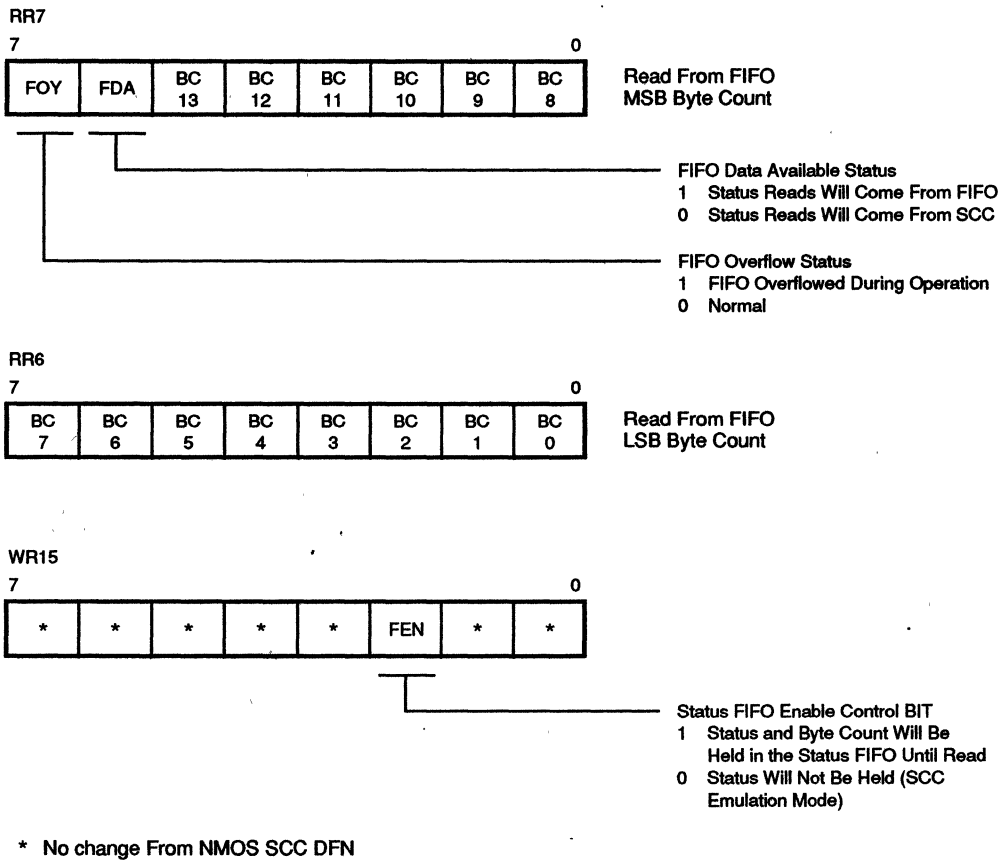


Figure 16. SCC Additional Registers

SOFTWARE INTERRUPT ACKNOWLEDGE

The SCC can do an interrupt acknowledge cycle through software. In some CPU environments it is difficult to create the /INTACK signal with the necessary timing to acknowledge interrupts and allow the nesting of interrupts. In these cases, it would be desirable to create this signal in software.

If bit 5 of Write Register 9 (WR9) is set, reading register 2 (RR2) will result in an interrupt acknowledge cycle to be executed internally. Like a hardware INTACK cycle, a software acknowledge will cause the /INT pin to return high.

Similarly to when the /INTACK signal is used, when a software acknowledge cycle is sued, a Reset Highest IUS command must be issued in the interrupt service routine. If the RR2 is read from channel B, the modified vector will be returned. If the RR2 is read from channel A, then the vector will be returned unmodified. The Vector Includes Status (VIS) and no vector (NV) bits (WR9) and are ignored when bit 5 is set to 1.

When the /INTACK is not being used, it should be pulled up to VDD through a resistor (10K ohm typical).

SCSI FUNCTIONAL DESCRIPTION

General. The Small Computer System interface (SCSI) device has a set of eight registers that are controlled by the CPU. By reading and writing the appropriate registers, the CPU may initiate any SCSI Bus activity or may sample and assert any signal on the SCSI Bus. This allows the user to implement all or any of the SCSI protocol in software. These registers are read (written) by activating /SCSICS with an address on A2-A0 and then issuing a /RD (/WR) pulse. This section describes the operation of the internal registers (Table 2).

Table 2. Register Summary

Address				
A2	A1	A0	R/W	Register Name
0	0	0	R	Current SCSI Data
0	0	0	W	Output Data
0	0	1	R/W	Initiator Command
0	1	0	R/W	Mode
0	1	1	R/W	Target Command
1	0	0	R	Current SCSI Bus Status
1	0	0	W	Select Enable
1	0	1	R	Bus and Status
1	0	1	W	Start DMA Send
1	1	0	R	Input Data
1	1	0	W	Start DMA Target Receive
1	1	1	R	Reset Parity/Interrupt
1	1	1	W	Start DMA Initiator Receive

Data Registers. The data registers are used to transfer SCSI commands, data, status, and message bytes between the microprocessor Data Bus and the SCSI Bus. The SCSI does not interpret any information that passes through the data registers. The data registers consist of the transparent Current SCSI Data Register, the Output Data Register, and the Input Data Register.

Current SCSI Data Register. *Address 0* (Read Only). The Current SCSI Data Register (Figure 17) is a read-only register which allows the microprocessor to read the active SCSI Data Bus. This is accomplished by activating /SCSICS with an address on A2-A0 and issuing a /RD pulse. If parity checking is enabled, the SCSI Bus parity is checked at the beginning of the read cycle. This register is used during a programmed I/O data read or during Arbitration to check for higher priority arbitrating devices. Parity is not guaranteed valid during Arbitration.

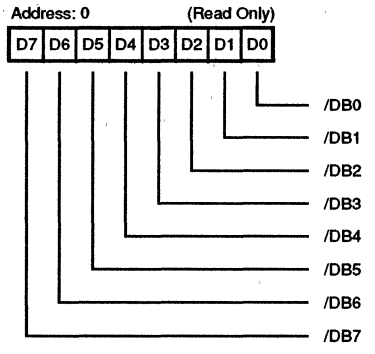


Figure 17. Current SCSI Data Register

Output Data Register. *Address 0* (Write Only). The Output Data Register (Figure 18) is a write-only register that is used to send data to the SCSI Bus. This is accomplished by either using a normal CPU write, or under DMA control, by using /WR and /DACK. This register also asserts the proper ID bits on the SCSI Bus during the Arbitration and Selection phases.

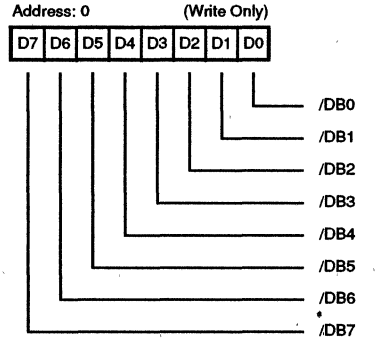


Figure 18. Output Data Register

Input Data Register. *Address 6* (Read Only). The input Data Register (Figure 19) is a read-only register that is used to read latched data from the SCSI Bus. Data is latched either during a DMA Target receive operation

when /ACK goes active or during a DMA Initiator receive when /REQ goes active. The DMA Mode bit (Mode Register bit 1) must be set before data can be latched in the Input Data Register. This register is read under DMA control using /RD and /DACK. Parity is optionally checked when the Input Data Register is loaded.

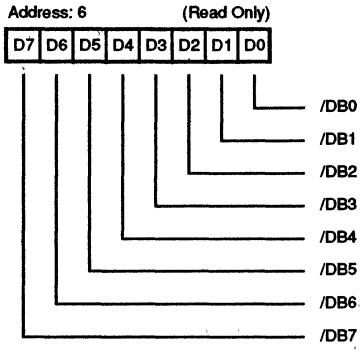


Figure 19. Input Data Register

Initiator Command Register. *Address 1* (read/write). The Initiator Command Register (Figures 20 and 21) are read and write registers which assert certain SCSI Bus signals, monitors those signals, and monitors the progress of bus arbitration. Many of these bits are significant only when being used as an Initiator; however, most can be used during Target role operation.

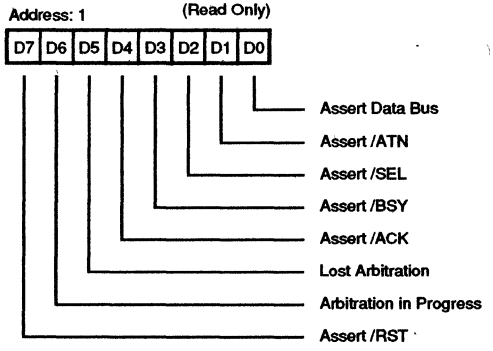


Figure 20. Initiator Command Register (Register Read)

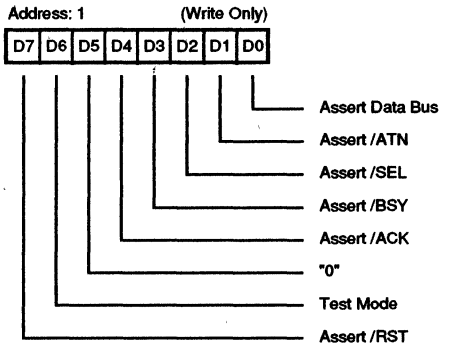


Figure 21. Initiator Command Register (Register Write)

The following describes the operation of all bits in the Initiator Command Register.

Bit 0. *Assert Data Bus.* The ASSERT DATA BUS bit, when set, allows the contents of the Output Data Register to be enabled as chip outputs on the signals /DB7-/DB0. Parity is also generated and asserted on /DBP.

When connected as an Initiator, the outputs are only enabled if the TARGETMODE bit (Mode Register, bit 6) is FALSE, the received signal I/O is FALSE, and the phase signals C/D, I/O, and /MSG match the contents of the ASSERT C/D, ASSERT I/O and ASSERT /MSG in the Target Command Register.

This bit should also be set during DMA send operations.

Bit 1. *ASSERT/ATN/ATN.* Bit 1 may be asserted on the SCSI Bus by setting this bit to a one (1) if the TRAGETMODE bit (Mode Register, bit 6) is FALSE /ATN is normally asserted by the initiator to request a Message Out bus phase. Note that since ASSERT/SEL and ASSERT/ATN are in the same register, a select with /ATN may be implemented with one CPU write /ATN may be deasserted by resetting this bit to zero. A read on this register simply reflects the status of this bit.

Bit 2. *ASSERT/SEL.* Writing a one (1) into this bit position asserts /SEL onto the SCSI Bus. /SEL is normally asserted after Arbitration has been successfully completed /SEL may be disabled by resetting bit 2 to a zero. A read of this register reflects the status of this bit.

Bit 3. *ASSERT/BSY.* Writing a one (1) into this bit position asserts /BSY onto the SCSI Bus. Conversly, a zero resets the /BSY signal. Asserting /BSY indicates a successful selection or reselection. Resetting this bit creates a Bus-Disconnect condition. Reading this register reflects bit status.

Bit 4. *ASSERT/ACK.* Bit 4 is used by the bus initiator to assert /ACK on the SCSI Bus. In order to assert /ACK, the TARGETMODE bit (Mode Register, bit 6) must be FALSE. Writing a zero to this bit deasserts /ACK. Reading this register reflects bit status.

Bit 5. *"0"* (Write Bit). Bit 5 should be written with a zero for proper operation.

Bit 5. *LA* (Lost Arbitration - Read Bit). Bit 5, when active, indicates that the SCSI detected a Bus-Free condition, arbitrated for use of the bus by asserting /BSY and its ID on the Data Bus, and lost Arbitration due to /SEL being asserted by another bus device. This bit is active only when the ARBITRATE bit (Mode Register, bit 0) is active.

Bit 6. *TEST MODE* (Write Bit). Bit 6 is written during a test environment to disable all output drivers, effectively removing the Z53C80 from the circuit. Resetting this bit returns the part to normal operation.

Bit 6. *AIP* (Arbitration in Process - Read Bit). Bit 6 is used to determine if Arbitration is in progress. For this bit to be active, the ARBITRATE bit (Mode Register, bit 0) must have been set previously. It indicates that a Bus-Free condition has been detected and that the chip has asserted /BSY and put the contents of the Output Data Register onto the SCSI Bus. AIP will remain active until the ARBITRATE bit is reset.

Bit 7. *ASSERT/RST.* Whenever a one is written to bit 7 of the Initiator Command Register, the /RST signal is asserted on the SCSI Bus. The /RST signal will remain asserted until this bit is reset or until an external /RESET occurs. After this bit is set (1), IRQ goes active and all internal logic and control registers are reset (except for the interrupt latch and the ASSERT/RST bit). Writing a zero to bit 7 of the Initiator Command Register deasserts the /RST signal. The status of this bit is monitored by reading the Initiator Command Register.

Mode Register. *Address 2* (Read/Write). The Mode Register controls the operation of the chip. This register determines whether the SCSI operates as an Initiator or a Target, whether DMA transfers are being used, whether parity is checked, and whether interrupts are generated on various external conditions. This register is read to check the value of these internal control bits (Figure 22).

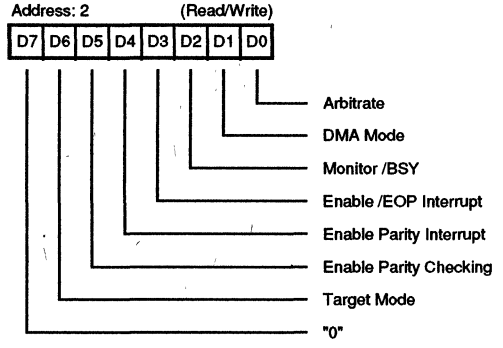


Figure 22. Mode Register

Bit 0. *ARBITRATE.* The ARBITRATE bit is set (1) to start the Arbitration process. Prior to setting this bit, the Output Data Register should contain the proper SCSI device ID value. Only one data bit should be active for SCSI Bus Arbitration. The SCSI waits for a Bus-Free condition before entering the Arbitration phase. The results of the Arbitration phase is determined by reading the status bits LA and AIP (Initiator Command Register, bits 5 and 6, respectively).

Bit 1. *DMA MODE.* The DMA MODE bit is normally used to enable a DMA transfer and must be set (1) prior to writing Start DMA Send Register, Start DMA Target Receive Register, and Start DMA Initiator Receiver Register. These three registers are used to start DMA transfers. The TARGETMODE bit (Mode Register, bit 6) must be consistent with writes to Start DMA Target Receive and Start DMA Initiator Receive Registers [i.e., set (1) for a write to start DMA Target Receive Register and set (0) for a write to Start DMA Initiator Receive Register]. The control bit ASSERT DATA BUS (Initiator Command Register, bit 0) must be TRUE (1) for all DMA send operations. In the DMA mode, /REQ and /ACK are automatically controlled.

The DMA MODE bit is not reset upon the receipt of an /EOP signal. Any DMA transfer is stopped by writing a zero into this bit location; however, care must be taken not to cause /SCSICS and /DACK to be active simultaneously.

Bit 2. *MONITOR BUSY.* The MONITOR BUSY bit, when TRUE (1), causes an interrupt to be generated for an

unexpected loss of /BSY. When the interrupt is generated due to loss of /BSY, the lower six bits of the Initiator Command Register are reset (0) and all signals are removed from the SCSI Bus.

Bit 3. *ENABLE/EOP interrupt.* The enable /EOP interrupt, when set (1), causes an interrupt to occur when the /EOP (End of Process) signal is received from the DMA controller logic.

Bit 4. *ENABLE PARITY INTERRUPT.* Tthe ENABLE PARITY INTERRUPT bit, when set (1), will cause an interrupt (IRQ) to occur if a parity error is detected. A parity interrupt will only be generated if the ENABLE PARITY CHECKING bit (bit 5) is also enabled (1).

Bit 5. *ENABLE PARITY CHECKING.* The ENABLE PARITY CHECKING bit determines whether parity errors are ignored or saved in the parity error latch. If this bit is reset (0), parity is ignored. Conversely, if this bit is set (1), parity errors are saved.

Bit 6. *TARGETMODE.* The TARGETMODE bit allows the SCSI to operate as either a SCSI Bus Initiator, bit reset (0), or as a SCSI Bus Target device, bit set (1). If the signals /ATN and /ACK are to be asserted on the SCSI Bus, the TARGETMODE bit must be reset (0). If the signals C//D, I//O, /MSG, and /REQ are to be asserted on the SCSI Bus, the TARGETMODE bit must be set (1).

Bit 7. *"0".* Bit 7 should be written with a zero for proper operation.

Target Command Register. *Address 3* (Read/Write). When connected as a target device, the Target Command Register (Figure 23) allows the CPU to control the SCSI Bus Information Transfer phase and/or to assert /REQ by writing this register. The TARGETMODE bit (Mode Register, bit 6) must be TRUE (1) for bus assertion to occur. The SCSI Bus phases are described in Table 3.

Table 3. SCSI Information Transfer Phase

Bus Phase	ASSERT I//O	ASSERT C//D	ASSERT /MS
Data Out	0	0	0
Unspecified	0	0	1
Command	0	1	0
Message Out	0	1	1
Data In	1	0	0
Unspecified	1	0	1
Status	1	1	0
Message In	1	1	1

When connected as an Initiator with DMA MODE TRUE, if the phase lines I//O, C//D, and /MSG do not match the phase bits in the Target Command Register, a phase mismatch interrupt is generated when /REQ goes active. To send data as an Initiator, the ASSERT I//O, ASSERT C//D, and ASSERT /MSG bits must match the corresponding bits in the Current SCSI Bus Status Register. The ASSERT /REQ bit (bit 3) has no meaning when operating as an Initiator.

Bits 4, 5, and 6 are not used.

Bit 7. *LAST BYTE SENT* (Read Only). The END OF DMA TRANSFER bit (Bus and Status Register, bit 7) only indicates when the last byte was received from the DMA controller. The LAST BYTE SENT bit can be used to flag that the last byte of the DMA send operation has been transferred on the SCSI Data Bus.

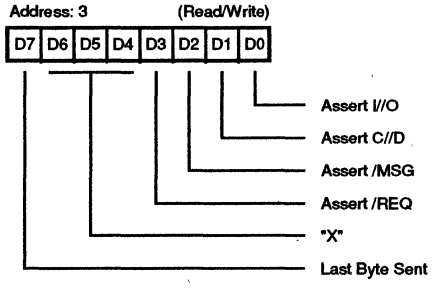


Figure 23. Target Command Register

Current SCSI Bus Status Register. *Address 4* (Read Only). The Current SCSI Bus Register is a read-only register which is used to monitor seven SCSI Bus control signals, plus the Data Bus parity bit. For example, an Initiator device can use this register to determine the current bus phase and to poll /REQ for pending data transfers. This register may also be used to determine why a particular interrupt occurred. Figure 24 describes the Current SCSI Bus Status Register.

Select Enable Register. *Address 4* (Write Only). The Select Enable Register (Figure 25) is a write-only register which is used as a mask to monitor a signal ID during a selection attempt. The simultaneous occurrence of the correct ID bit, /BSY FALSE, and /SEL TRUE will cause an interrupt. This interrupt can be disabled by resetting all bits in this register. If the ENABLE PARITY CHECKING bit (Mode Register, bit 5) is active (1), parity is checked during selection.

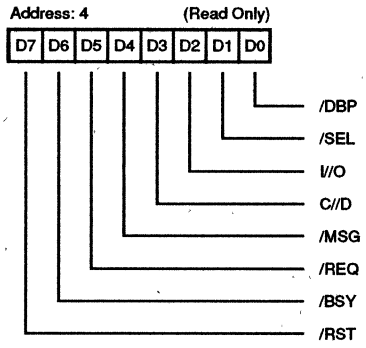


Figure 24. Current SCSI Bus Status Register

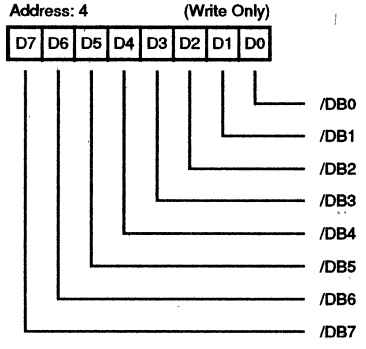


Figure 25. Select Enable Register

Bus and Status Register. *Address 5* (Read Only). The Bus and Status Register (Figure 26) is a read-only register which can be used to monitor the remaining SCSI control signals not found in the Current SCSI Bus Status Registers (/ATN and /ACK), as well as six other status bits. The following describes each bit of the Bus Status Register individually.

Bit 0. */ACK.* Bit 0 reflects the condition of the SCSI Bus control signal /ACK. This signal is normally monitored by the Target device.

Bit 1. */ATN.* Bit 1 reflects the condition of the SCSI Bus control signal /ATN. This signal is normally monitored by the Target device.

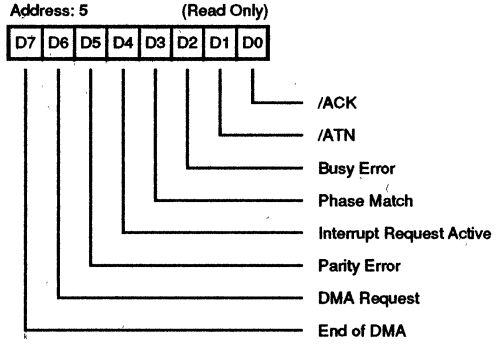


Figure 26. Bus and Status Register

Bit 2. *BUSY ERROR.* The BUSY ERROR bit is active if an unexpected loss of the /BSY signal has occurred. This latch is set whenever the MONITOR BUSY bit (Mode Register, bit 2) is TRUE and /BSY is FALSE. An unexpected loss of /BSY disables any SCSI outputs and resets the DMA MODE bit (Mode Register, bit 1).

Bit 3. *PHASE MATCH.* The SCSI signals /MSG, C//D, and I//O, represent the current information Transfer phase. The PHASE MATCH bit indicates whether the current SCSI Bus phase matches the lower 3 bits of the Target Command Register. PHASE MATCH is continuously updated and is only significant when operating as a Bus Initiator. A phase match is required for data transfers to occur on the SCSI Bus.

Bit 4. *INTERRUPT REQUEST ACTIVE.* Bit 4 is set if an enabled interrupt condition occurs. It reflects the current state of the IRQ output and can be cleared by reading the Reset Parity/Interrupt Register.

Bit 5. *PARITY ERROR.* Bit 5 is set if a parity error occurs during a data receive or a device selection. The PARITY ERROR bit can only be set (1) if the ENABLE PARITY CHECK bit (Mode Register, bit 5) is active (1). This bit may be cleared by reading the Reset Parity/Interrupt Register.

Bit 6. *DMA REQUEST.* The DMA REQUEST bit allows the CPU to sample the output pin DRQ. DRQ can be cleared by asserting /DACK or by resetting the DMA MODE bit (bit 1) in the Mode Register. The DRQ signal does not reset when a phase-mismatch interrupt occurs.

Bit 7. *END OF DMA TRANSFER.* The END OF DMA TRANSFER bit is set if /EOP, /DACK, and either /RD or /WR are simultaneously active for at least 100ns. Since the /EOP signal can occur during the last byte sent to the Output Data Register, the /REQ and /ACK signals should be monitored to ensure that the last byte has been transferred. This bit is reset when the DMA MODE bit is reset (0) in the Mode Register.

DMA Registers. Three write-only registers are used to initiate all DMA activity. They are: Start DMA Send, Start DMA Target Receive, and Start DMA Initiator Receive. Performing a write operation into one of these registers starts the desired type of DM transfer. Data presented to the SCSI on signals D7-D0 during the register write is meaningless and has no effect on the operation. Prior to writing these registers, the DMA MODE bit (bit 1), and the TARGETMODE bit (bit 6) in the Mode Register must be appropriately set. The individual registers are briefly described as follows:

Start DMA Send. *Address 5* (Write Only). This register is written to initiate a DMA send, from the DMA to the SCSI Bus, for either Initiator or Target role operations. The DMA MODE bit (Mode Register, bit 1) is set prior to writing this register.

Start DMA Target Receive. *Address 6* (Write Only). This register is written to initiate a DMA receive - from the SCSI Bus to the DMA, for Target operation only. The DMA MODE bit (bit 1) and the TARGETMODE bit (bit 6) in the Mode Register must both be set (1) prior to writing this register.

Start DMA Initiator Receive. *Address 7* (Write Only). This register is written to initiate a DMA receive - from the SCSI Bus to the DMA, for Initiator operation only. The DMA MODE bit (bit 6) must be FALSE (0) in the Mode Register prior to writing this register.

Reset Parity/Interrupt. *Address 7* (Read Only). Reading this register resets the PARITY ERROR bit (bit 5), the INTERRUPT REQUEST bit (bit 4), and the BUSY ERROR bit (bit 2) in the Bus and Status Register.

On-Chip SCSI Hardware Support. The SCSI is easy to use because of its simple architecture. The chip allows direct control and monitoring of the SCSI Bus by providing a latch for each signal. However, portions of the protocol define timings which are much too quick for traditional microprocessors to control. Therefore, hardware support has been provided for DMA transfers, bus arbitration, phase change monitoring, bus disconnection, bus reset, parity generation, parity checking, and device selection/reselection.

Arbitration is accomplished using a Bus-Free filter to continuously monitor /BSY. If /BSY remains inactive for at least 400ns, the SCSI is considered free and Arbitration may begin. Arbitration will begin if the bus is free, /SEL is inactive, and the ARBITRATE bit (Mode Register, bit 0) is active. Once arbitration has begun (/BSY asserted), an arbitration delay of 2.2 µs must elapse before the Data Bus can be examined to determine if Arbitration is enabled. This delay is implemented in the controlling software driver.

The Z53C80 is a clockwise device. Delays such as bus-free delay, bus-set delay, and bus-settle delay are implemented using gate delays. These delays may differ between devices because of inherent process variations, but are well within the proposed ANSI X3.131 - 1986 specification.

INTERRUPTS. The Z53C80 provides an interrupt output (IRQ) to indicate a task completion or an abnormal bus occurrence. The use of interrupts is optional and may be disabled by resetting the appropriate bits in the Mode Register or the Select Enable Register.

When an interrupt occurs, the Bus and Status Register and the Current SCSI Bus Status Register (Figures 26 and 24) must be read to determine which condition created the interrupt. IRQ can be reset simply by reading the Reset Parity/Interrupt Register or by an external chip reset /RESET active for 100ns.

Assuming the Z53C80 has been properly initialized, an interrupt will be generated if the chip is selected or reselected, if an /EOP signal occurs, if a parity error occurs during a data transfer, if a bus phase mismatch occurs, or if a SCSI Bus disconnection occurs.

Selection Reselection. The Z53C80 generates a select interrupt if SEL is active (0), its device ID is TRUE and /BSY is FALSE for at least a bus-settle delay. If I//O is active, this is considered a reselect interrupt The correct ID bit is determined by a match in the Select Enable Register. Only a single bit match is required to generate an interrupt. This interrupt may be disabled by writing zeros into all bits of the Select Enable Register.

If parity is supported, parity should be good during the selection phase. Therefore, if the ENABLE PARITY bit (Mode Register, bit 5) is active, the PARITY ERROR bit is checked to ensure that a proper selection has occurred. The ENABLE PARITY INTERRUPT bit need not be set for this interrupt to be generated.

The proposed SCSI specification also requires that no more than two device ID's be active during the selection process. To ensure this, the Current SCSI Data Register is read.

The proper values for the Bus and Status Register and the Current SCSI Bus Status Register are displayed in Figures 27 and 28, respectively.

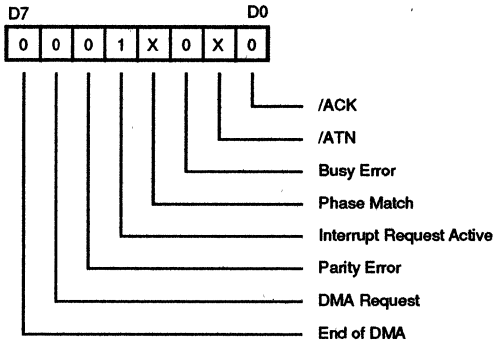


Figure 27. Bus and Status Register

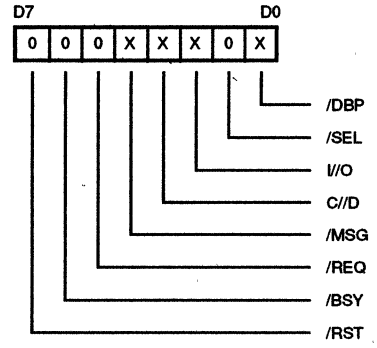


Figure 28. Current SCSI Bus Status Register

End of Process (EOP) Interrupt. An End Of Process signal (EOP) which occurs during a DMA transfer (DMA MODE TRUE) will set the END OF DMA Status bit (Bus and Status Register bit 7) and will optionally generate an interrupt if ENABLE EOP INTERRUPT bit (Mode Register, bit 3) is TRUE. The /EOP pulse will not be recognized (END OF DMA bit set) unless /EOP, /DACK, and either /RD or /WR are concurrently active for at least 50 ns. DMA transfers can still occur if /EOP was not asserted at the correct time. This interrupt is disabled by resetting the ENABLE EOP INTERRUPT bit.

The proper values for the Bus and Status Register and the Current SCSI Bus Status Register for this interrupt are shown in Figures 29 and 30.

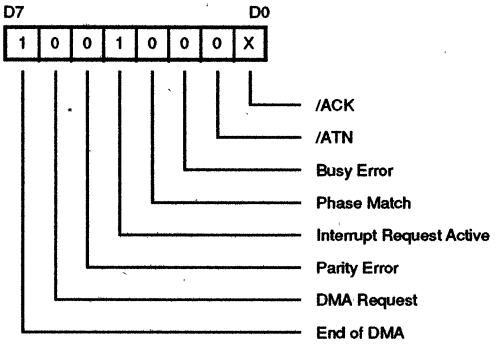


Figure 29. Bus and Status Register

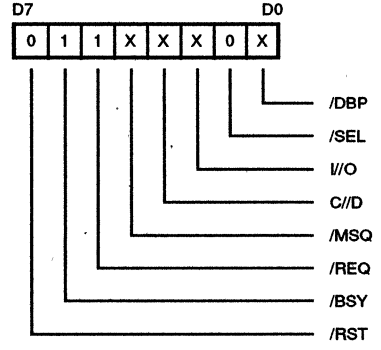


Figure 30. Current SCSI Bus Status Register

The END OF DMA bit is used to determine when a block transfer is complete. Receive operations are complete when there is no data left in the chip and no additional handshakes occurring. The only exception to this is receiving data as an Initiator and the Target opts to send additional data for the same phase. In this /REQ goes active and the new data is present in the Input Data Register. Since a phase-mismatch interrupt will not occur, /REQ and /ACK need to be sampled to determine that the Target is attempting to send more data.

For send operations, the END OF DMA bit is set when the DMA finishes its transfers, but the SCSI transfer may still be in progress. If connected as a Target, /REQ and /ACK should be sampled until both are FALSE. If connected as an Initiator, a phase change interrupt is used to signal the completion of the previous phase. It is possible for the Target to request additional data for the same phase. In this case, a phase change will not occur and both /REQ and /ACK are sampled to determine when the last byte was transferred.

SCSI Bus Reset. The SCSI generates an interrupt when the /RST signal transitions to TRUE. The device releases all bus signals within a bus-clear delay of this transition. This interrupt also occurs after setting the ASSERT /RST bit (Initiator Command Register, bit 7). This interrupt cannot be disabled. (Note: /RST is not latched in bit 7 of the Current SCSI Bus Status Register and is not active when this port is read. For this case, the Bus Reset interrupt is determined by default.)

The proper values for the Bus and Status Register and the Current SCSI Bus Status Register are displayed in Figures 31 and 32, respectively.

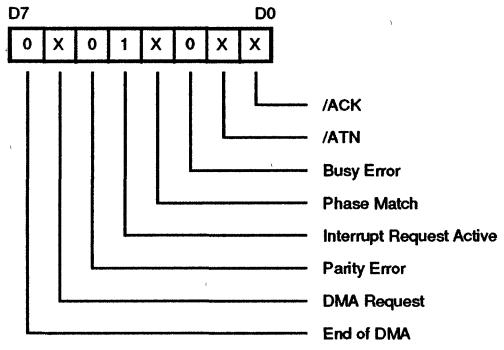


Figure 31. Bus and Status Register

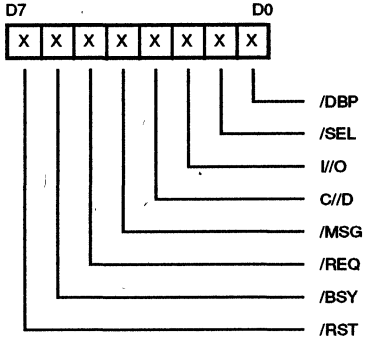


Figure 32. Current SCSI Bus Status Register

Parity Error. An interrupt is generated for a received parity error it the ENABLE PARITY CHECK (bit 5) and the ENABLE PARITY INTERRUPT (bit 4) bits are set (1) in the Mode Register. Parity is checked during a read of the Current SCSI Data Register and during a DMA receive operation. A parity error can be detected without generating an interrupt by disabling the ENABLE PARITY INTERRUPT bit and checking the PARITY ERROR flag (Bus and Status Register, bit 5).

The proper values for the Bus and Status Register and the Current SCSI Bus Status Register are displayed in Figures 33 and 34, respectively.

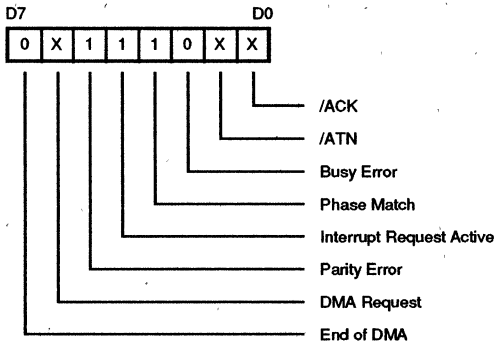


Figure 33. Bus and Status Register

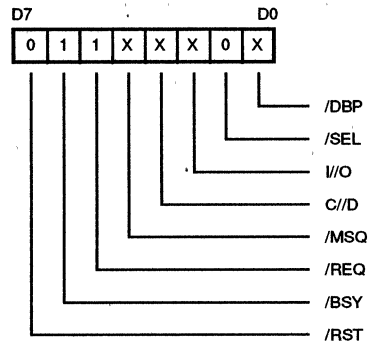


Figure 34. Current SCSI Bus Status Register

Bus Phase Mismatch. The SCSI phase lines have the signals I/O, C/D, and /MSG. These signals are compared with the corresponding bits in the Target Command Register: ASSERT I/O (bit 0), ASSERT C/D (bit 1), and ASSERT /MSG (bit 2). The comparison occurs continually and is reflected in the PHASE MATCH bit (bit 3) of the Bus and Status Register. If the DMA MODE bit (Mode Register, bit 1) is active and a phase mismatch occurs when /REQ transitions from FALSE to TRUE, an interrupt (IRQ) is generated.

A phase mismatch prevents the recognition of /REQ and removes the chip from the bus during an Initiator send operation (/DB7-/DB0 and /DBP will not be driven even through the ASSERT DATA BUS bit (Initiator Command Register, bit 0) is active). This may be disabled by resetting the DMA MODE bit (Note: It is possible for this interrupt to occur when connected as a Target if another device is driving the phase lines to a different state).

The proper values for the Bus and Status Register and the Current SCSI Bus Status Register are displayed in Figures 35 and 36, respectively

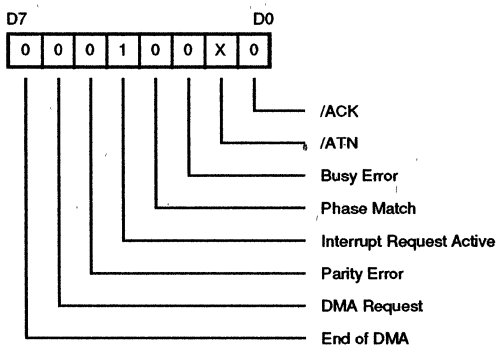


Figure 35. Bus and Status Register

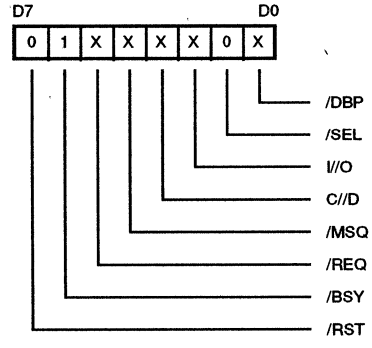


Figure 36. Current SCSI Bus Status Register

Loss of BSY. If the MONITOR BUSY bit (bit 2) in the Mode Register is active, an interrupt is generated if the BSY signal goes FALSE for at least a bus-settle delay. This interrupt is disabled by resetting the MONITOR BUSY bit. Register values are displayed in Figures 37 and 38.

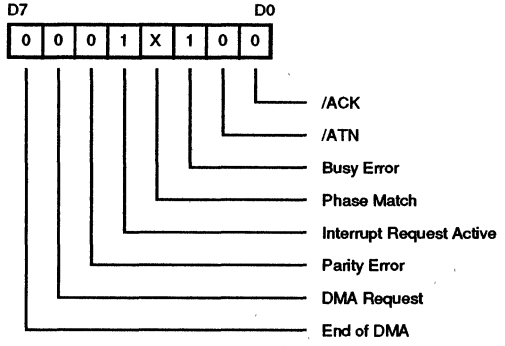


Figure 37. Bus and Status Register

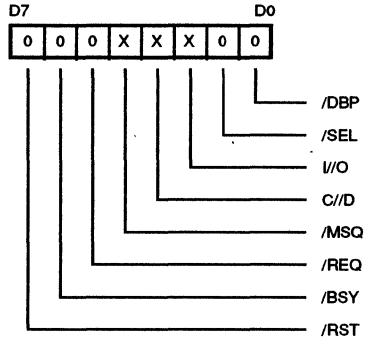


Figure 38. Current SCSI Bus Status Register

Reset Conditions. Three possible reset situations exist with the Z85C80, as follows:

Hardware Chip Reset. When the signal RST is active for at least 100 ns, the Z53C80 device is re-initialized and all internal logic and control registers are cleared. This is a chip reset only and does not create a SCSI Bus-Reset condition.

SCSI Bus Reset (/RST) Received. When a SCSI /RST signal is received, an IRQ interrupt is generated and a chip reset is performed. All internal logic and registers are cleared, except for the IRQ interrupt latch and the ASSERT /RST bit (bit 7) in the Initiator Command Register. (Note: The /RST signal may be sampled by reading the Current SCSI Bus Status Register, however, this signal is not latched and may not be present when this port is read).

SCSI Bus Reset (/RST) Issued. If the CPU sets the ASSERT/RST bit (bit 7) in the Initiator Command Register, the /RST signal goes active on the SCSI Bus and an internal reset is performed. Again, all internal logic and registers are cleared except for the IRQ interrupt latch and the ASSERT/RST bit (bit 7) in the Initiator Command Register. The /RST signal will continue to be active until the ASSERT/RST bit is reset or until a hardware reset occurs.

Data Transfers. Data is transferred between SCSI Bus devices in one of four modes. 1) Programmed I/O, 2) Normal DMA, 3) Block Mode DMA, or 4) Pseudo DMA. The following sections describe these modes in detail (Note: for all data transfer operations /DACK and /SCSICS should never be active simultaneously).

Programmed I/O Transfers. Programmed I/O is the most primitive form of data transfer The /REQ and /ACK handshake signals are individually monitored and asserted by reading and writing the appropriate register bits. This type of transfer is normally used when transferring small blocks of data such as command blocks or message and status bytes. An Initiator send operation would begin by setting the C//D, I//O, and /MSG bits in the Target Command Register to the correct state so that a phase match exists. In addition to the phase match condition, it is necessary for the ASSERT DATA BUS bit (Initiator Command Register, bit 0) to be TRUE and the received I/O signal to be FALSE for the Z53C80 to send data. For each transfer, the data is loaded into the Output Data Register. The CPU then waits for the /REQ bit (Current SCSI Bus Status Register, bit 5) to become active. Once /REQ goes active, the PHASE MATCH bit (Initiator Command Register, bit 4) is set. The /REQ bit is sampled until it becomes FALSE and the CPU resets the ASSERT /ACK bit to complete the transfer.

Normal DMA Mode. DMA transfers are normally used for large block transfers. The SCSI chip outputs a DMA request (DRQ) whenever it is ready for a byte transfer. External DMA logic uses this DRQ signal to generate /DACK and a /RD or a /WR pulse to the Z53C80. DRQ goes inactive when /DACK is asserted and /DACK goes inactive some time after the minimum read or write pulse width. This process is repeated for every byte. For this mode, /DACK should not be allowed to cycle unless a transfer is taking place.

Pseudo DMA Mode. To avoid the tedium of monitoring and asserting the request/acknowledgement handshake signals for programmed I/O transfers, the system may be designed to implement a pseudo DMA mode. This mode is implemented by programming the Z53C80 to operate in the DMA mode, but using the CPU to emulate the DMA handshake. DRQ may be detected by polling the DMA REQUEST bit (bit 6) in the Bus and Status Register, by sampling the signal through an external port, or by using it to generate a CPU interrupt. Once DRQ is detected, the CPU can perform a read or write data transfer. This CPU read/write is externally decoded to generate the appropriate /DACK and /RD or /WR signals.

Often, external decoding logic is necessary to generate the /SCSICS signal. This same logic may be used to generate /DACK at no extra cost and provide an increased performance in programmed I/O transfers.

Halting a DMA Operation. The EOP signal is not the only way to halt a DMA transfer. A bus phase mismatch or a reset of the DMA MODE bit (Mode Register, bit 1) can also terminate a DMA cycle for the current bus phase.

Using the /EOP Signal. If /EOP is used, it should be asserted for at least 50 ns while /DACK and /RD or /WR are simultaneously active. Note, however, that if /RD or /WR is not active, an interrupt is generated, but the DMA activity continues. The /EOP signal does not reset the DMA MODE bit. Since the /EOP signal can occur during the last byte sent to the Output Data Register, the /REQ and /ACK signals are monitored to ensure that the last byte has transferred.

Bus Phase Mismatch Interrupt. A bus phase mismatch interrupt is used to halt the transfer if operating as an Initiator. Using this method frees the host from maintaining a data length counter and frees the DMA logic from providing the /EOP signal. If performing an Initiator send operation, the Z53C80 requires /DACK to cycle before /ACK goes inactive. Since phase changes cannot occur if /ACK is active, either /DACK must be cycled after the last byte is sent or the DMA MODE bit must be reset in order to receive the phase mismatch interrupt.

Resetting the DMA MODE Bit. A DMA operation may be halted at any time simply by resetting the DMA MODE bit. It is recommended that the DMA MODE bit be reset after receiving an /EOP or bus phase-mismatch interrupt. The DMA MODE bit must then be set before writing any of the start DMA registers for subsequent bus phases.

If resetting the DMA MODE bit is used instead of /EOP for Target role operation, then care must be taken to reset this bit at the proper time. If receiving data as a Target device, the DMA MODE bit must be reset once the last DRQ is received and before /DACK is asserted to prevent an additional /REQ from occurring. Resetting this bit causes DRQ to go inactive. However, the last byte received remains in the Input Data Register and may be obtained either by performing a normal CPU read or by cycling /DACK and /RD. In most cases, /EOP is easier to use when operating as a Target device.

READ REGISTERS

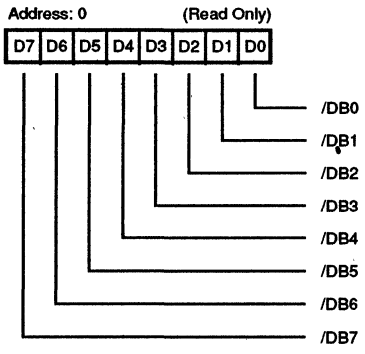


Figure 39. Current SCSI Data Register

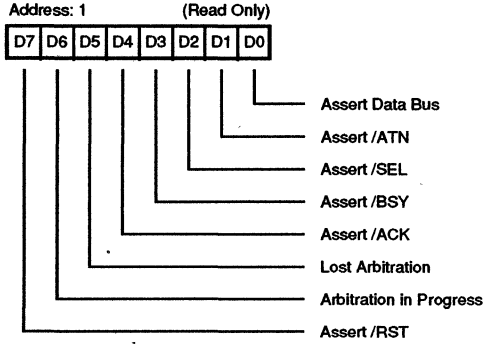


Figure 40. Initiator Command Register

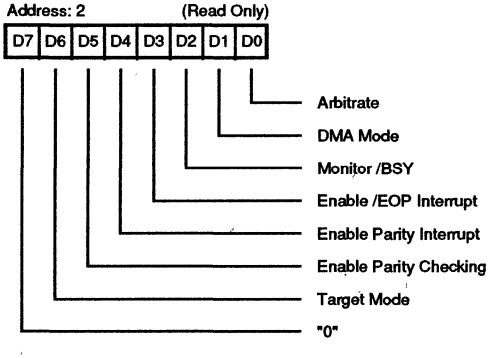


Figure 41. Mode Register

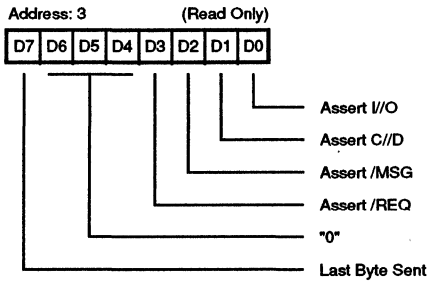


Figure 42. Target Command Register

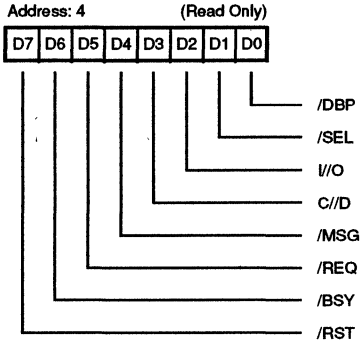


Figure 43. Current SCSI Bus Status Register

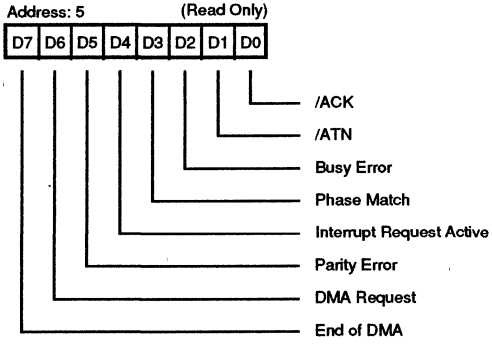


Figure 44. Bus and Status Register

READ REGISTERS (Continued)

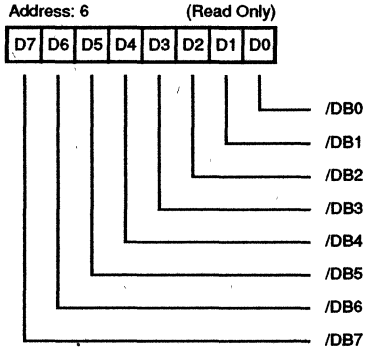


Figure 45. Input Data Register

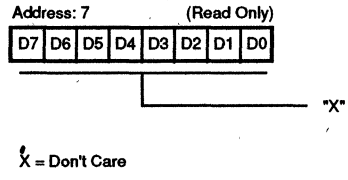


Figure 46. Reset Parity/Interrupt

WRITE REGISTERS

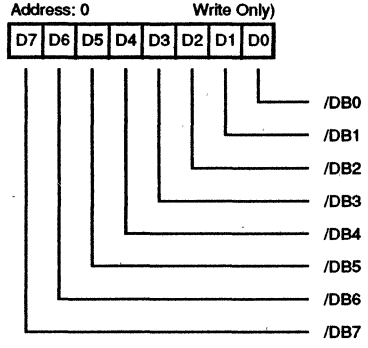


Figure 47. Output Data Register

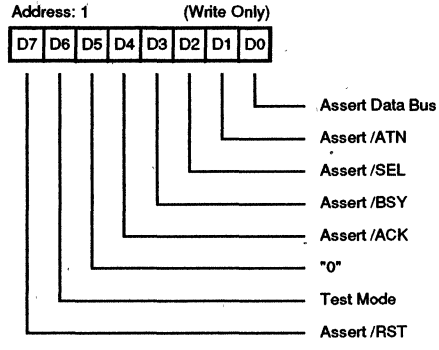


Figure 48. Initiator Command Register

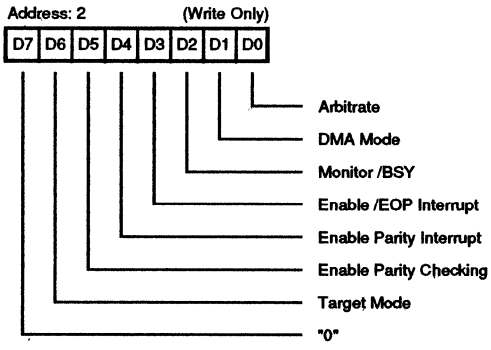


Figure 49. Mode Register

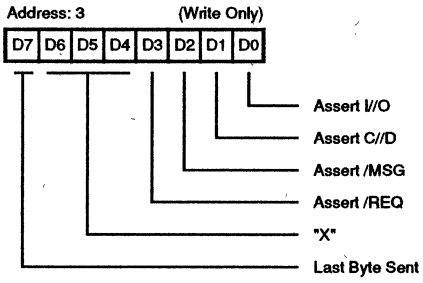


Figure 50. Target Command Register

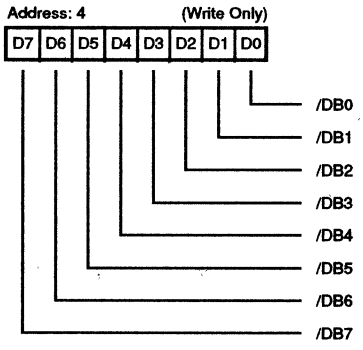


Figure 51. Select Enable Register

Note: X = Don't care

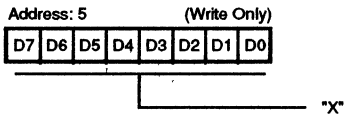


Figure 52. Start DMA Send

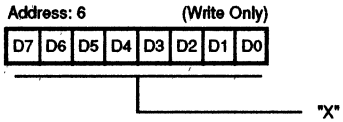


Figure 53. Start DMA Target Receive

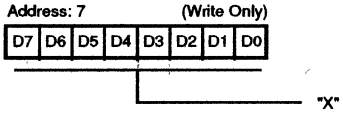


Figure 54. Start DMA Initiator Receive

ABSOLUTE MAXIMUM RATINGS

Voltages on all pins with respect to GND .....-0.3V to +7.0V
Operating Ambient Temperature ....See Ordering Information
Storage Temperature.....-65°C to +150°C

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to this 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. Standard conditions are as follows:

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

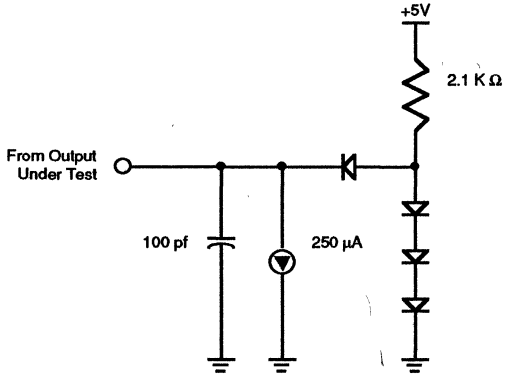


Figure 55. Standard Test Load

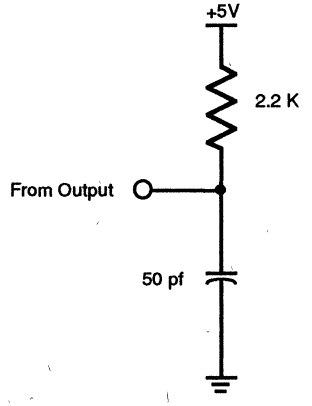


Figure 56. Open-Drain Test Load

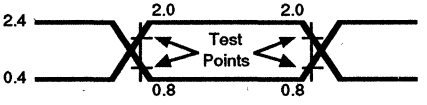


Figure 57. Switching Test Waveform

DC CHARACTERISTICS

Symbol	Parameter	Condition	Min	Max	Units
V_{DD}	Supply Voltage		4.5	5.5	V
V_{IH}	High-Level Input Voltage		2.0	5.5	V
V_{IL}	Low-Level Input Voltage		-0.3	0.8	V
$I_{IH}1$	High-Level Input Current	V_{IH} = 5.5V			
	SCSI Bus Pins	V_{IL} = 0V		50	µA
$I_{IH}2$	High-Level Input Current	V_{IH} = 5.5V			
	All Other Pins	V_{IL} = 0V		10	µA
$I_{IL}1$	Low-Level Input Current	V_{IH} = 5.5V			
	SCSI Bus Pins	V_{IL} = 0V		-50	µA
$I_{IL}2$	Low-Level Input Current	V_{IH} = 5.5V			
	All Other Pins	V_{IL} = 0V		-10	µA
$V_{OH}1$	High-Level Output Voltage	I_{OH} = -3mA	2.4		
$V_{OH}2$	High-Level Output Voltage	I_{OH} = -250 µA	V_{DD} -0.8	V	
$V_{OL}1$	Low-Level Output Voltage				
	SCSI Bus Pins	I_{OL} = 48 mA		0.5	V
$V_{OL}2$	Low-Level Output Voltage				
	All Other Pins	I_{OL} = 7 mA		0.5	V
I_{DD}	Supply Current			40	mA
C_{IN}	Input Capacitance			10	pf
C_{OUT}	Output Capacitance			15	pf
$C_{I/O}$	Bidirectional Capacitance			20	pf
T_A	Operating Free-Air Temperature		0	70	°C

AC CHARACTERISTICS
General Timing

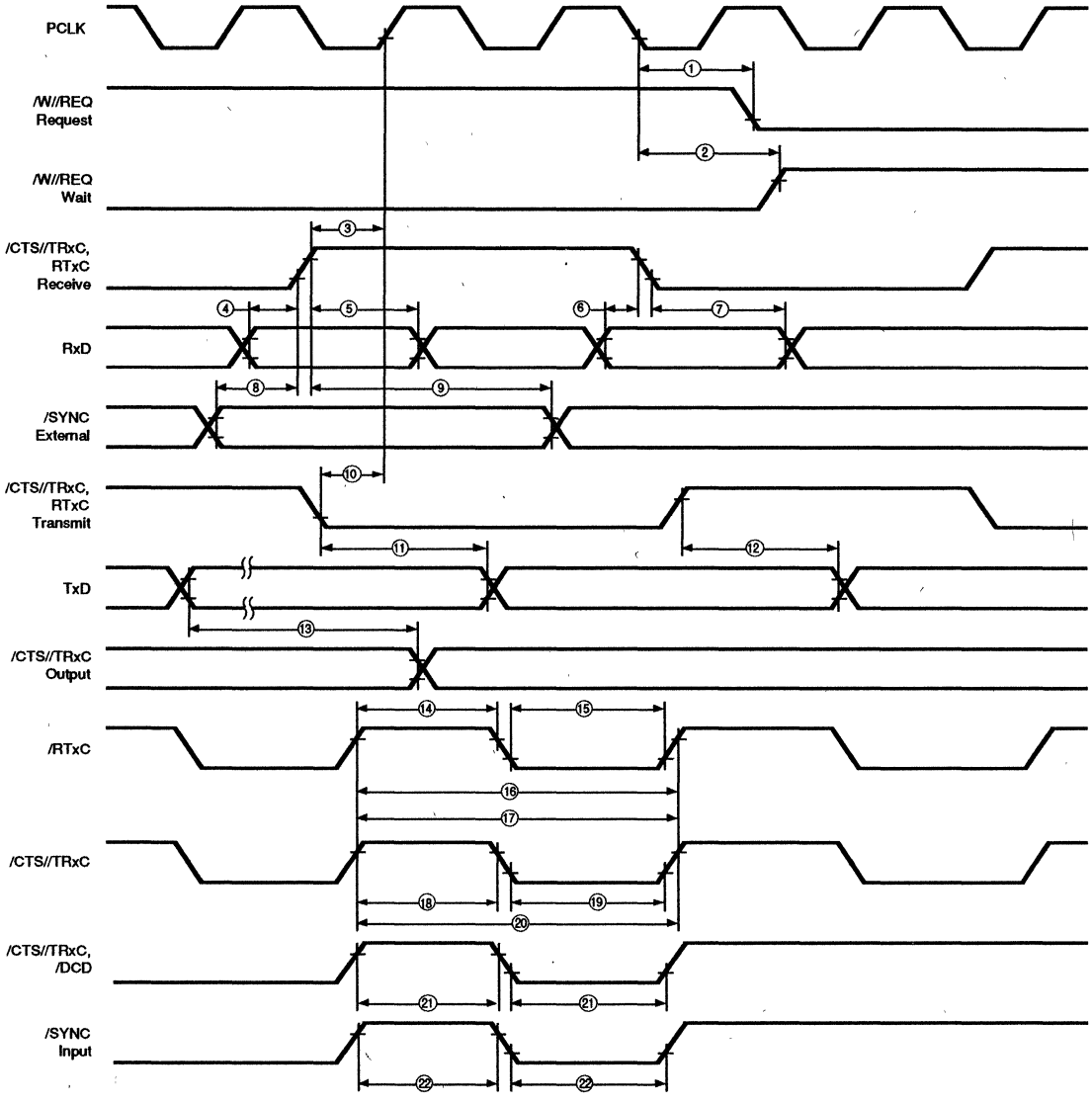


Figure 58. General Timing

AC CHARACTERISTICS (Continued)
Z85C80 General Timing

No	Symbol	Parameter	Min	Max	Notes †
1	TdPC(REQ)	PCLK FALL to /W//REQ Valid Delay		200	
2	TdPC(W)	PCLK FALL to Wait Inactive Delay		300	
3	TsRXC(PC)	/RxC Rise to PCLK Rise Setup Time	N/A	N/A	[1,4]
4	TsRXD(RXCr)	RxD to /RxC Rise Setup Time	0		[1]
5	ThRXD(RXCr)	RxD to /RxC Rise Hold Time	125		[1]
6	TsRXD(RXCf)	RxD to/RxC FALL Setup Time	0		[1,5]
7	ThRXD(RXCf)	RxD to /RxC FALL Hold Time	125		[1,5]
8	TsSY(RXC)	/SYNC to /RxC Rise Setup Time	-150		[1]
9	ThSY(RXC)	/SYNC to /RxC Rise Hold Time	5TcPc		[1]
10	TsTXC(PC)	/TxC FALL to PCLK Rise Setup Time	N/A		[2,4]
11	TdTXC(TXD)	/TxC FALL to TxD Delay		150	[2]
12	TdTxCr(TXD)	/TxC Rise to TxD Delay		150	[2,5]
13	TdTXD(TRX)	TxD to /TRxC Delay		140	
14	TwRTXh	/RTxC High Width	120		[6]
15	TwRTXl	/RTxC Low Width	120		[6]
16a	TcRTX	/RTxC Cycle Time	400		[6,7]
16b	TxRX(DPLL)	DPLL Cycle Time	50		[7,8]
17	TcRTXX	Crystal Oscillator Period	100	1000	[3]
18	TwTRXh	/TRxC High Width	120		[6]
19	TwTRXl	/TRxC Low Width	120		[6]
20	TcTRX	/TRxC Cycle Time	400		[6,7]
21	TwEXT	/DCD or /CTS Pulse Width	120		
22	TwSY	/SYNC Pulse Width	120		

Notes:
[1] /RxC is /RTxC or TRxC, whichever is supplying the receive clock.
[2] /TxC is /TRxC or RTxC, whichever is supplying the transmit clock.
[3] Both /RTxC and /SYNC have 300 pf capacitors to ground connected to them.
[4] Synchronization of /RxC to PCLK is eliminated in divide by four operation.
[5] Parameter applies only to FM encoding/decoding.
[6] Parameter applies only for transmitter and receiver; DPLL and baud rate timing requirements are identical to case PCLK requirements
[7] The maximum receive or transmit data is 1/4 PCLK
[8] Applies to DPLL clock source only Maximum data rate of 1/4 PCLK still applies. DPLL clock should have a 50% duty cycle.

† Units in nanoseconds (ns)

AC CHARACTERISTICS

Z85C80 System Timing

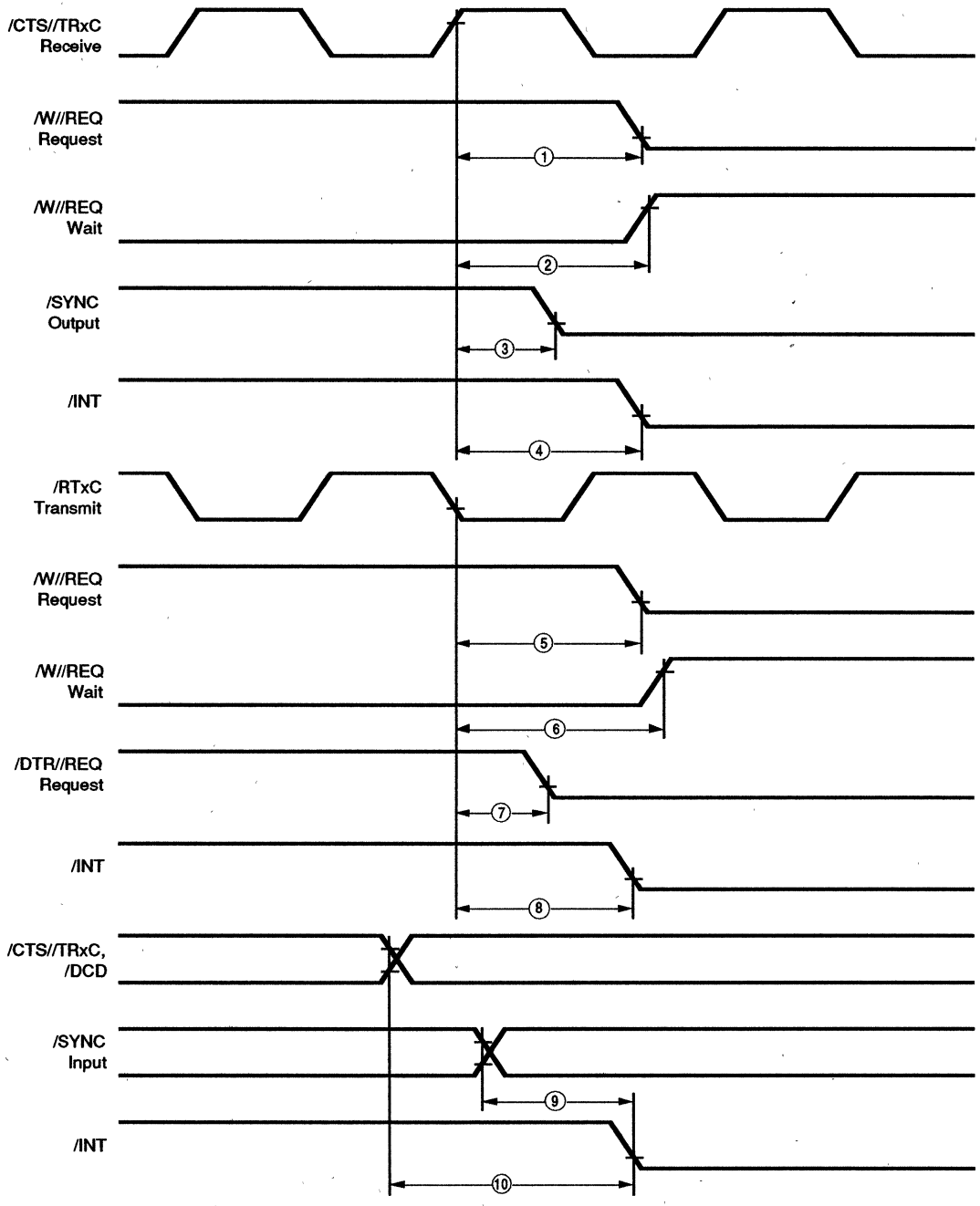


Figure 59. System Timing

AC CHARACTERISTICS (Continued)
Z85C80 System Timing

No	Symbol	Parameter	Min	Max	Notes †
1	TdRXC(REQ)	/RxC Rise to /W//REQ Valid	8	12	[2]
2	TdRXCW)	/RxC Rise to Wait Inactive	8	14	[1,2]
3	TdRXC(SY)	/RxC Rise to /SYNC Valid	4	7	[2]
4	TdRXC(INT)	/RxC Rise to /INT Valid Delay	10	16	[1,2]
5	TdTXC(REQ)	/TxC Fall to /W//REQ	5	8	[3]
6	TdTXC(W)	/TxC Fall to Wait Inactive	5	11	[1,3]
7	TdTXC(DRQ)	/TxC Fall to /DTR//REQ Valid	4	7	[3]
8	TdTXC(INT)	/TxC Fall to /INT Valid	6	10	[1,3]
9	TdSY(INT)	/SYNC to /INT Valid	2	6	[1]
10	TdEXT(INT)	/DCD or /CTS//TRxC to /INT Valid	2	6	[1]

Notes:
[1] Open-drain output measured with open-drain test load.
[2] /RxC is /RTxC or /CTS//TRxC, whichever is supplying the receive clock.
[3] /TxC is /CTS//TRxC or RTxC, whichever is supplying the transmit clock.

† Units equal to TcPC

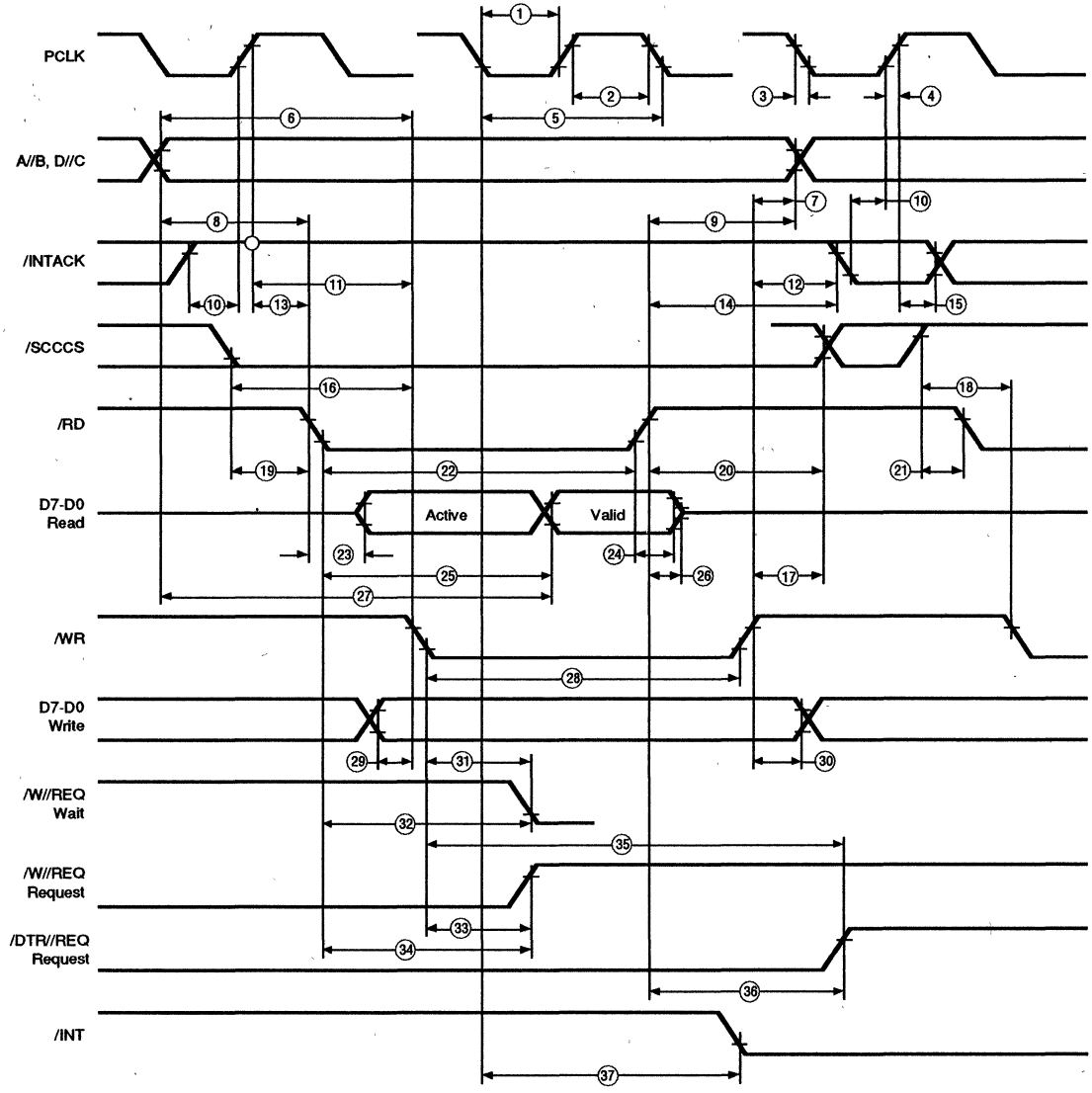


Figure 60. Read/Write Timing

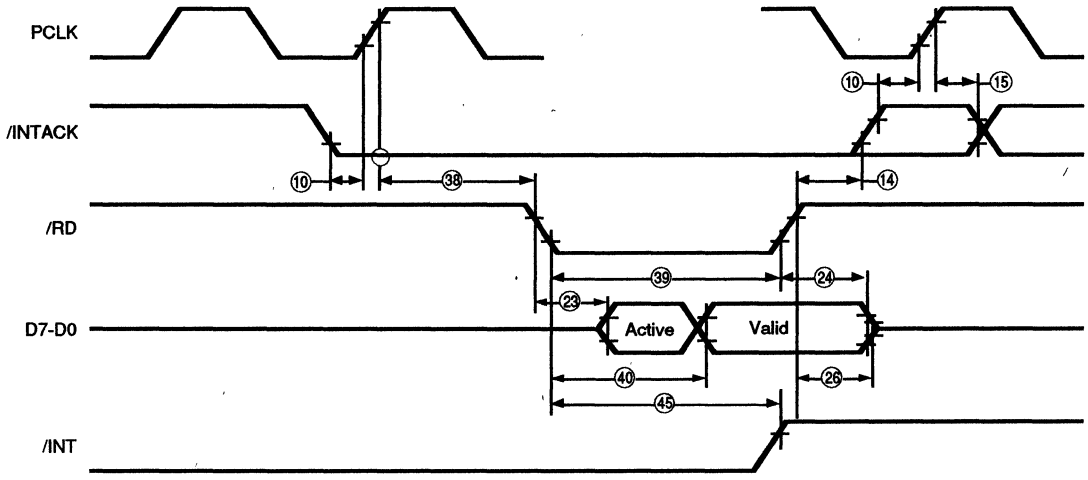


Figure 61. Interrupt Acknowledge Timing

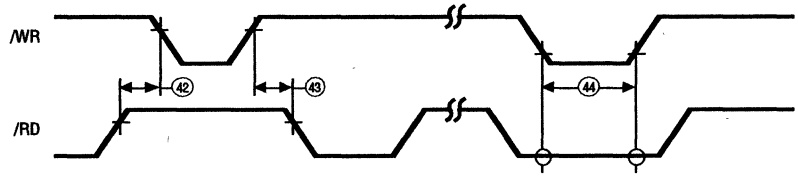


Figure 62. Reset Timing

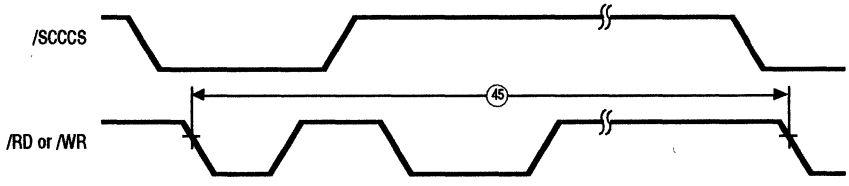


Figure 63. Cycle Timing

AC CHARACTERISTICS
Additional Timing

No	Symbol	Parameter	Min	Max	Notes †
1	TwPCl	PCLK Low Width	40	1000	
2	TwPCh	PCLK High Width	40	1000	
3	TfPC	PCLK Fall Time		10	
4	TrPC	PCLK Rise Time		10	
5	TcPC	PCLK Cycle Time	100	2000	
6	TsA(WR)	Address to /WR Fall Setup Time	50		
7	ThA(WR)	Address to /WR Rise Hold Time	0		
8	TsA(RD	Address to /RD Fall Setup Time	50		
9	ThA(RD)	Address to /RD Rise Hold Time	0		
10	TslA(PC)	/INTACK to PCLK Rise Setup Time	20		
11	TslAi(WR)	/INTACK to /WR Fall Setup Time	130		[1]
12	ThIA(WR)	/INTACK to /WR Rise Hold Time	0		
13	TslAi(RD)	/INTACK to /WR Fall Setup Time	130		[1]
14	ThIA(RD)	/INTACK to /RD Rise Hold Time	0		
15	ThIA(PC)	/INTACK to PCLK Rise Hold Time	30		
16	TsCEl(WR)	/SCCCS Low to /WR Fall Setup Time	0		
17	ThCE(WR)	/SCCCS to /WR Rise Hold Time	0		
18	TsCEh(WR)	/SCCCS High to /WR Fall Setup Time	50		
19	TsCEl(RD)	/SCCCS Low to /RD Fall Setup Time	0		[1]
20	ThCE(RD)	/SCCCS to /RD Rise Hold Time	0		[1]
21	TsCEh(RD)	/SCCCS High to /RD Fall Setup Time	50		[1]
22	TwRDl	/RD Low Width	125		[1]
23	TdRD(DRA)	/RD Fall to Read Data Active Delay	0		
24	TdRDr(DR)	/RD Rise to Read Data Not Valid Delay	0		
25	TdRDl(DR)	/RD Fall to Read Data Valid Delay		120	
26	TdRD(DRz)	/RD Rise to Read Data Float Delay		35	
27	TdA(DR)	Address to Read Data Valid Delay		180	
28	TwWRl	/WR Low Width	125		
29	TsDW(WR)	Write Data to /WR Fall Setup Time	10		
30	ThDW(WR)	Write Data to /WR Rise Hold Time	0		
31	TdWR(W)	/WR Fall to Wait Valid Delay		160	[2]
32	TdRD(W)	/RD Fall to Wait Valid Delay		160	[2]
33	TdWRf(REQ)	/WR Fall to /W//REQ Not Valid Delay		160	
34	TdRDf(REQ)	/RD Fall to /W//REQ Not Valid Delay		160	
35	TdWRr(REQ)	/WR Fall /DTR//REQ Not Valid Delay		4TcPC	
36	TdRDr(REQ)	/RD Rise to /DTR//REQ Not Valid Delay		N/A	
37	TdPC(INT)	PCLK Fall to /INT Valid Delay		450	
38	TdIAiRD)	/INTACK to /RD Fall (Acknowledge) Delay	125		[3]
39	TwRDA	/RD (Acknowledge) Width	125		[3]
40	TdRDA(DR)	/RD Fall (Acknowledge) to Read Data Valid Delay	120		
41	TdRDA(INT)	/RD Fall to /INT Inactive Delay		320	[2]
42	TdRD(WRQ)	/RD Rise to /WR Fall Delay for No Reset	15		

AC CHARACTERISTICS (Continued)
Additional Timing

No	Symbol	Parameter	Min	Max	Notes †
43	TdWRQ(RD)	/WR Rise to /RD Fall Delay for No Reset	15		
44	TwRES	/WR and /RD Coincident Low for Reset	100		
45	Trc	Valid Access Recovery Time	4TcPC		[1]

Notes:
[1] Parameter does not apply to Interrupt Acknowledge transactions.
[2] Open-drain output, measured with open-drain test load.
[3] Parameter is system dependent.

† Units in nanoseconds (ns)

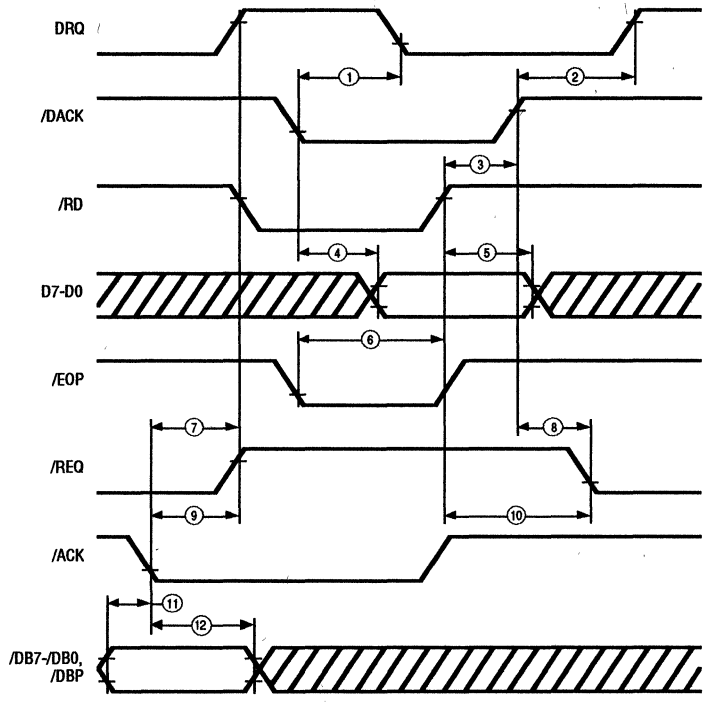


Figure 64. DMA Read Target Receive Cycle

AC CHARACTERISTICS
DMA Read Target Receive Cycle

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low		60	ns
2	/DACK High to DRQ High	30		ns
3	/DACK Hold Time from End of /RD	0		ns
4	Data Access Time from Read Enable*		70	ns
5	Data Hold Time from End of /RD	10		ns
6	Width of /EOP Pulse [1]	50		ns
7	/ACK Low to DRQ High		70	ns
8	/DACK High to /REQ Low (/ACK High)		90	ns
9	/ACK Low to /REQ High		80	ns
10	/ACK High to /REQ Low (/DACK High)		100	ns
11	Data Setup Time to /ACK	20		ns
12	Data Hold Time from /ACK	30		ns

Notes:
[1] /EOP, /RD, and /DACK must be concurrently Low for at least T6 for proper recognition of the /EOP pulse.
* Read Enable is the occurrence of /RD and /DACK.

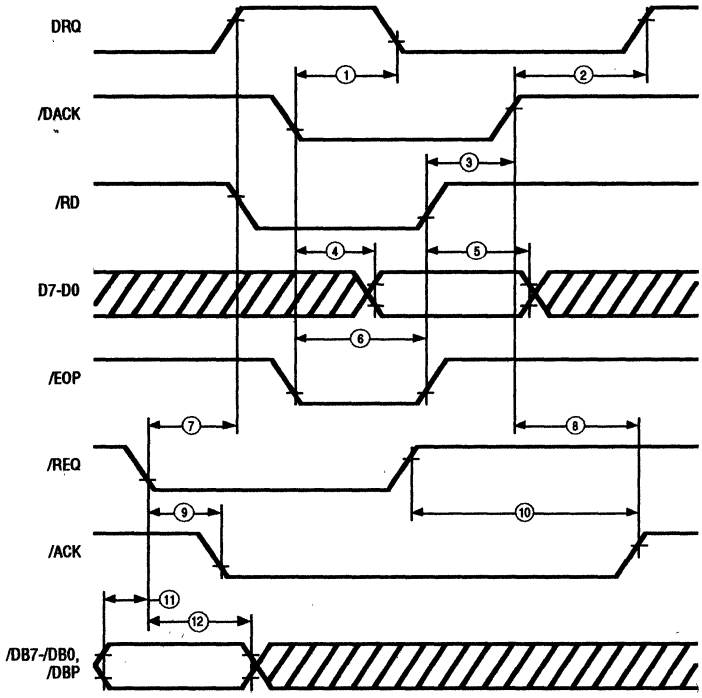


Figure 65. DMA Read Initiator Receive Cycle

AC CHARACTERISTICS

DMA Read Initiator Receive Cycle

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low		60	ns
2	/DACK High to DRQ High	30		ns
3	/DACK Hold Time from End of /RD	0		ns
4	Data Access Time from Read Enable*		70	ns
5	Data Hold Time from End of /RD	10		ns
6	Width of /EOP Pulse [1]	50		ns
7	/REQ Low to DRQ High		70	ns
8	/DACK High to /ACK High (/REQ High)		90	ns
9	/REQ Low to /ACK Low		70	ns
10	/REQ High to /ACK High (/DACK High)		80	ns
11	Data Setup Time to /REQ	20		ns
12	Data Hold Time from /REQ	50		ns

Notes:

[1] /EOP, /RD, and /DACK must be concurrently Low for at least T6 for proper recognition of the /EOP pulse.

* Read Enable is the occurrence of /RD and /DACK.

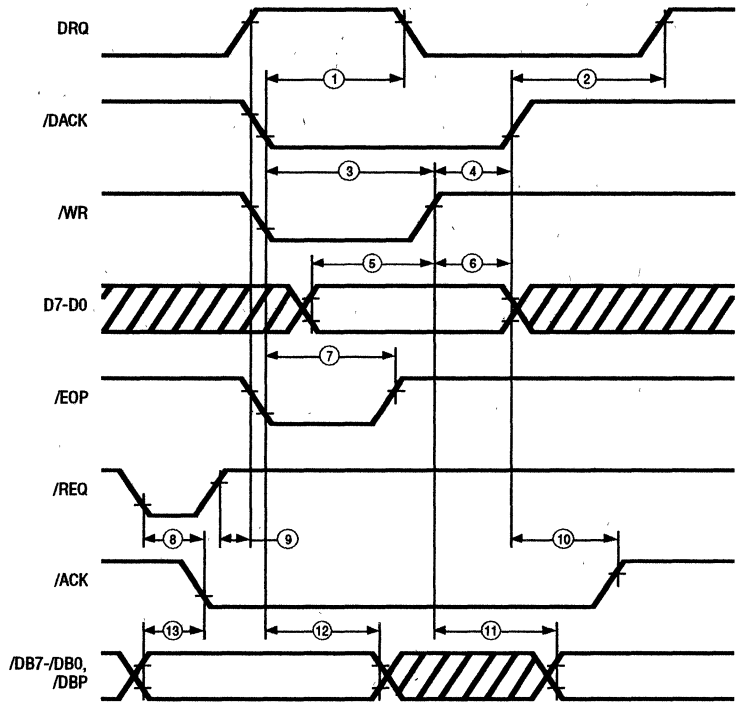


Figure 66. DMA Write Initiator Send Cycle

AC CHARACTERISTICS

DMA Write Initiator Send Cycle

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low		60	ns
2	/DACK High to DRQ High	30		ns
3	Write Enable Width*	50		ns
4	/DACK Hold from End of /WR	0		ns
5	Data Setup to End of Write Enable*	50		ns
6	Data Hold Time from End of /WR	25		ns
7	Width of /EOP Pulse [1]	50		ns
8	/REQ Low to /ACK Low		70	ns
9	/REQ High to DRQ High		70	ns
10	/DACK High to /ACK High		90	ns
11	/WR High to Valid SCSI Data		50	ns
12	Data Hold from Write Enable*	15		ns
13	Data Setup to /ACK Low	55		ns

Notes:

[1] /EOP, /WR, and /DACK must be concurrently Low for at least T7 for proper recognition of the /EOP pulse.

* Write Enable is the occurrence of /WR and /DACK.

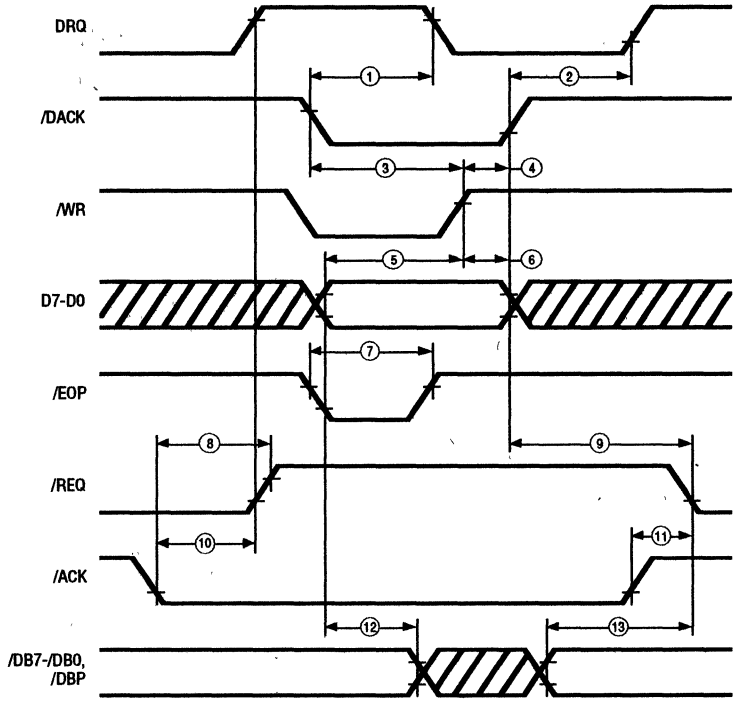


Figure 67. DMA Write Target Send Cycle

AC CHARACTERISTICS
DMA Write Target Send Cycle

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low		60	ns
2	/DACK High to DRQ High	30		ns
3	Write Enable Width*	50		ns
4	/DACK Hold from /WR High	0		ns
5	Data Setup to End of Write Enable*	50		ns
6	Data Hold Time from End of /WR	25		ns
7	Width of /EOP Pulse [1]	50		ns
8	/ACK Low to /REQ High		80	ns
9	/REQ from End of /DACK (/ACK High)		90	ns
10	/ACK Low to DRQ High (Target)		70	ns
11	/ACK High to /REQ Low (/DACK High)		100	ns
12	Data Hold from Write Enable	15		ns
13	Data Setup to /REQ Low (Target)	55		ns

Notes:
[1] /EOP, /WR, and /DACK must be concurrently Low for at least T7 for proper recognition of the /EOP pulse.
* Write Enable is the occurrence of /IOW and /DACK

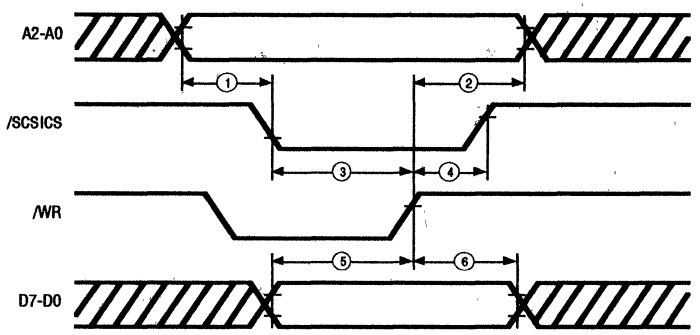


Figure 68. CPU Write Cycle

AC CHARACTERISTICS
CPU Write Cycle

No	Description	Min	Max	Units
1	Address Setup to Write Enable*	10		ns
2	Address Hold from End Write Enable*	10		ns
3	Write Enable Width*	40		ns
4	Chip Select Hold from End of /IOW	0		ns
5	Data Setup to end of Write Enable*	20		ns
6	Data Hold Time form End of /IOW	20		ns

Note:
* Write Enable is the occurrence of /WR and /SCSICS

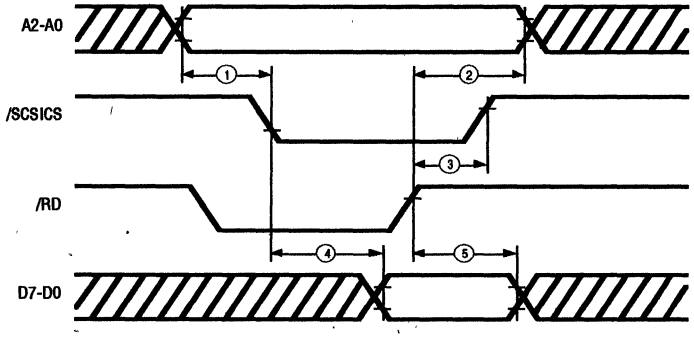


Figure 69. CPU Read Cycle

AC CHARACTERISTICS
CPU Read Cycle

No	Description	Min	Max	Units
1	Address Setup to Read Enable*	10		ns
2	Address Hold from End Read Enable*	10		ns
3	Chip Select Hold from End of /RD	0		ns
4	Data Access Time from Read Enable*		70	ns
5	Data Hold Time from End of Read Enable*	10		ns

Note:
* Read Enable is the occurrence of /RD and /SCSICS

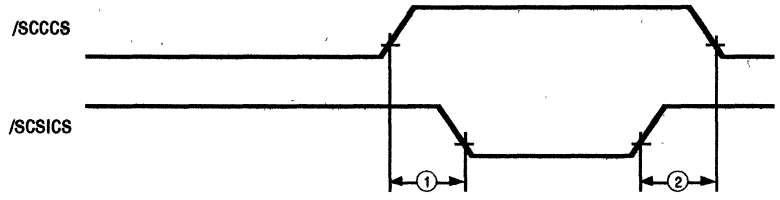


Figure 70. Selection

AC CHARACTERISTICS
Selection

No	Description	Min	Max	Units
1	/SCCCS to /SCSICS	35		ns
2	/SCSICS to /SCCCS	35		ns

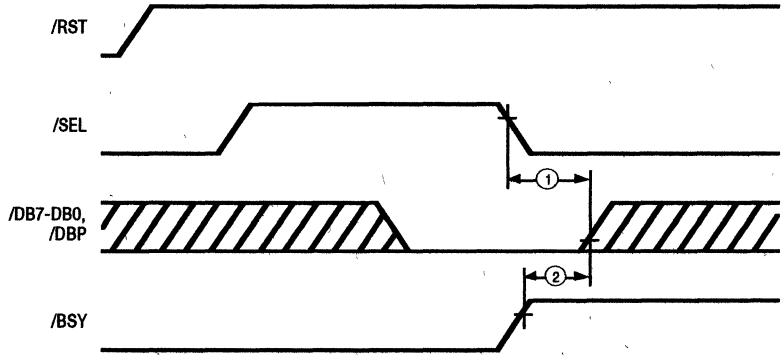


Figure 71. Arbitration

AC CHARACTERISTICS
Arbitration

No	Description	Min	Max	Units
1	Bus Clear from /SEL Low		600	ns
2	Bus Clear from /BSY High		1100	ns

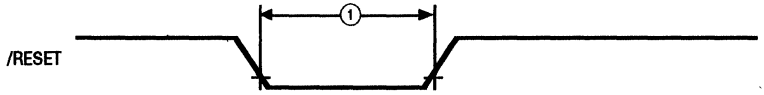


Figure 72. Reset

AC CHARACTERISTICS
Reset

No	Description	Min	Max	Units
1	Minimum Width of /RESET	100		ns

ZiLOG

APPLICATION NOTE

DESIGN A SERIAL BOARD TO HANDLE MULTIPLE PROTOCOLS

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 protocol 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. Practically, there could 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 (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 via this bus. Another microcontroller with its own ROM can be used in place of the Z8002 chip and ROM.

Note: 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); /N//S (NORMAL and SYSTEM are both active Low).

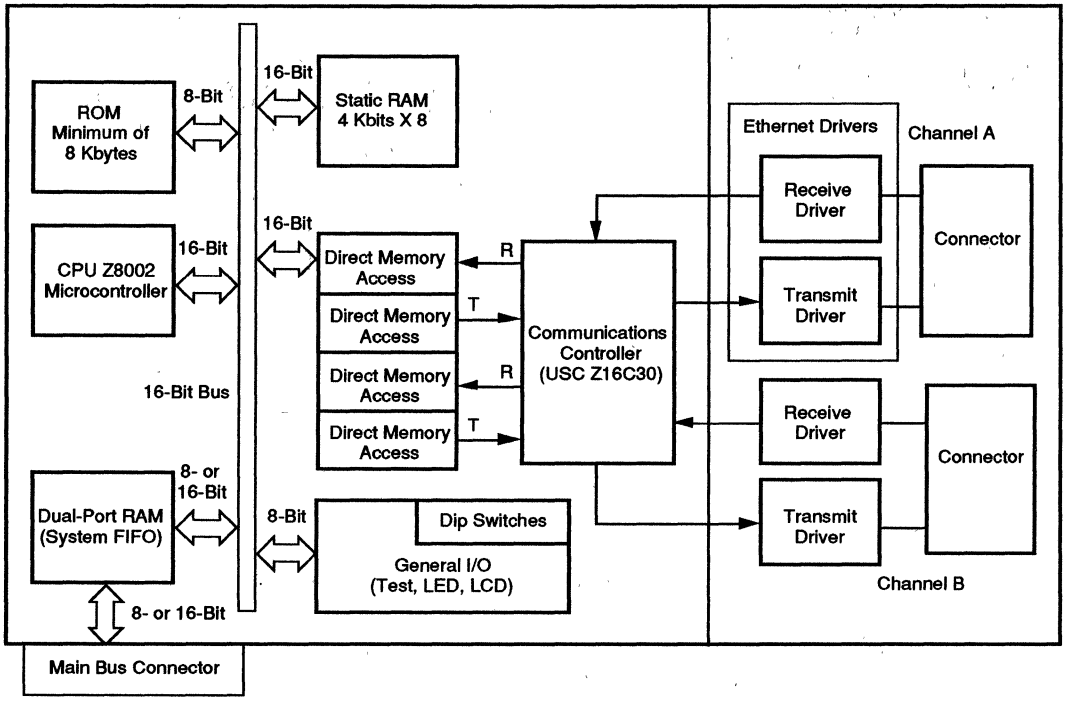


Figure 1. Dual-Channel Communications Board for an IBM 80386 PC consists of four main sections: The communications controller (a Z16C30 Universal Serial Controller), an interface to the PC's bus, buffer memory, and two communications channels. In this application, Channel A drives an Ethernet interface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

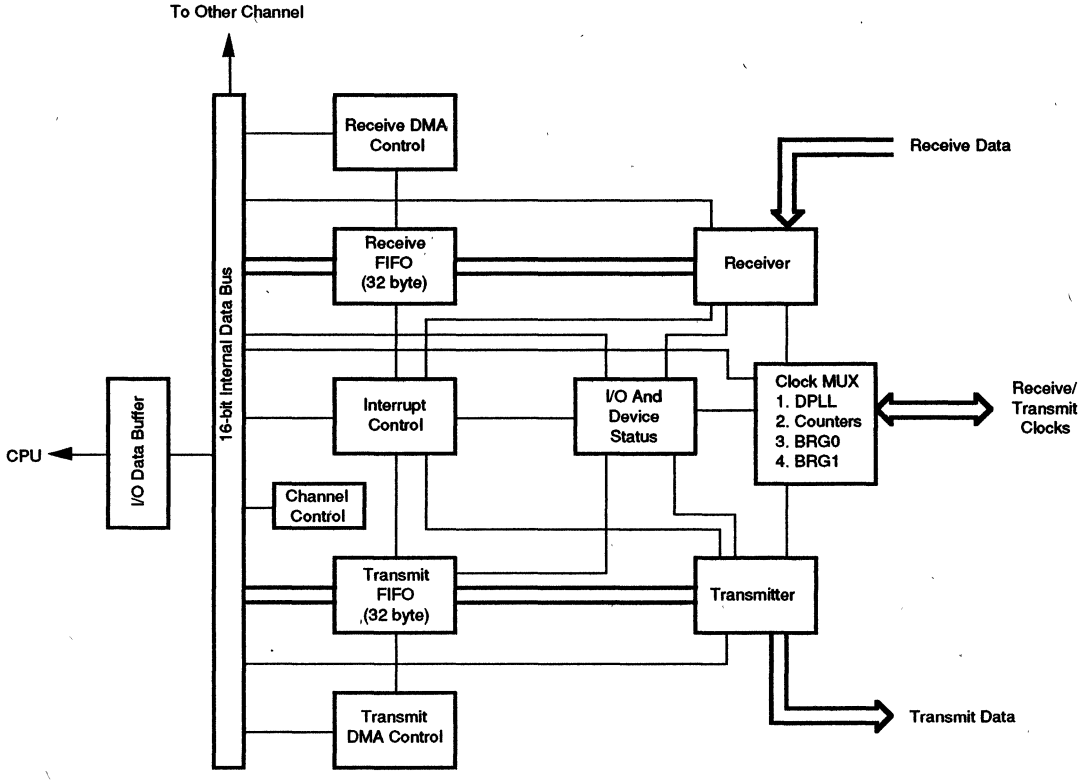


Figure 2. At The Heart of the dual-channel communications board is the Z16C30 Universal Serial Controller chip, which supports 10 different protocols and eight encoding formats. In the upper portion of the diagram, serial data from the receiver is placed in a 32-byte FIFO buffer prior to being placed 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.

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 via 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 to keeping data throughput from being derated.

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 via syscom and an IEEE 802.3 Ethernet protocol (via channel A) using a synchronous serial port. At the same time, the host is multitasking by sending data through the syscom (via 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 reference 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 via 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, biphase-level (BIPHASE-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.

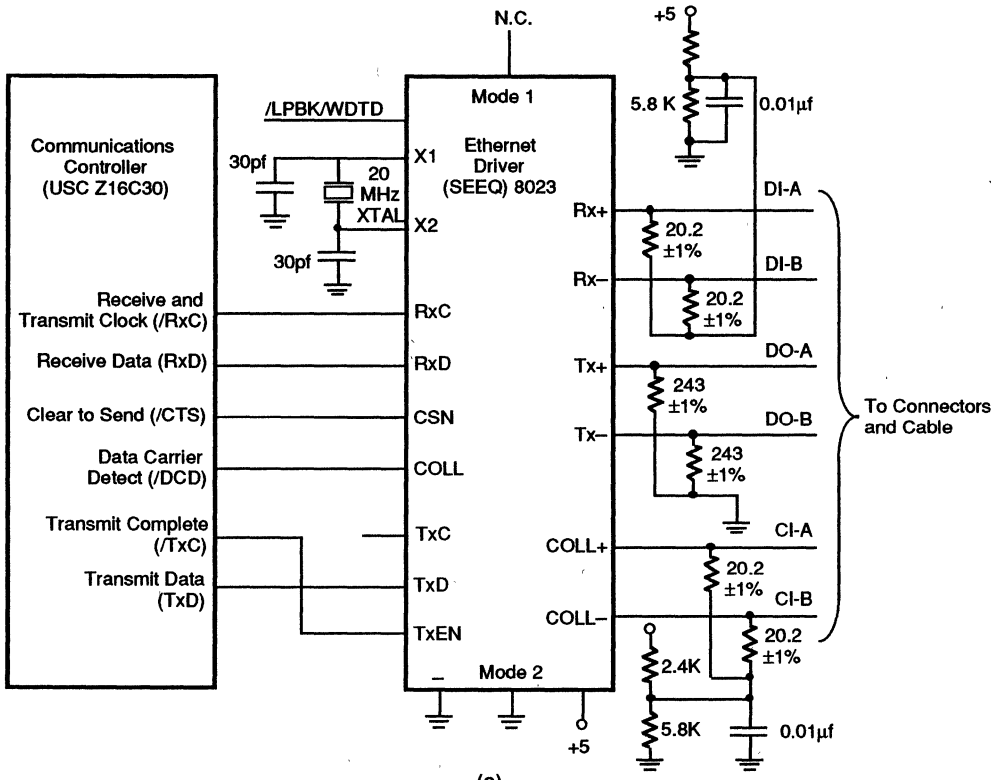


Figure 3. The Ethernet interface consists of an ethernet driver, which connects to the communications controller through clock and data lines (a). All that's required to change to another communications protocol is replacing the receive and transmit drivers, connectors, and cable. Transfer of Ethernet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

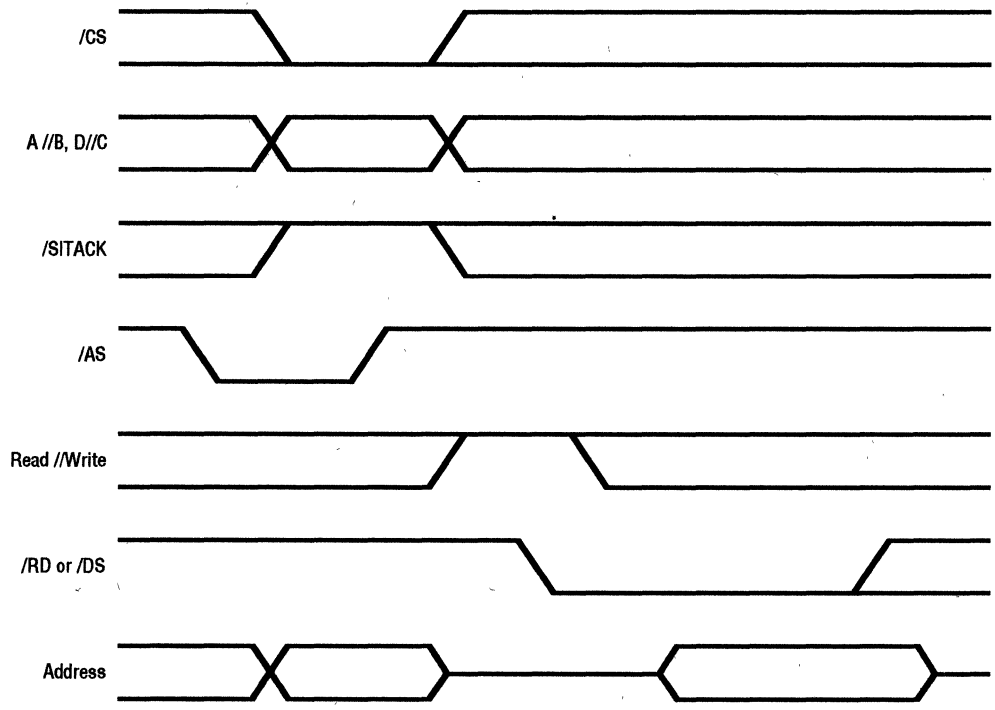


Figure 4. The USC/Ethernet Interface handles reading and writing of data through the two serial channels. Here, after the controller acknowledges a network request, raw IEEE 802.3 data is read from an Ethernet serial link.

Biphase 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

DPLL. 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.

Now, the data is 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 D15-11. 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 D15-11.

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 underrun, 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 D7-0). 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.

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 interrupt acknowledgement 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 postamble data decoded by the USC, to produce the terminate transmission signal, the host acknowledges the termination and frees the syscom CPU. Practically, 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 via 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 AD7-0 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 AD7-0 carry addresses to the USC and 8-bit data to and from the USC (AD15-8 is not used). Fly-by data transfers are 8-bits in this case.

Non-Multiplexed Case. AD15-0 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 via fly-by DMA transfers using /TXACK or /RXACK.

The non-multiplexed 8-bit case enables AD7-0 to carry 8-bit data to and from the USC with AD15-8 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 via 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 D15-11. 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 D15-11.

Now, let's look at what channel B was doing during channel A's flowthrough 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 via 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.

Now, the DMA controller 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.

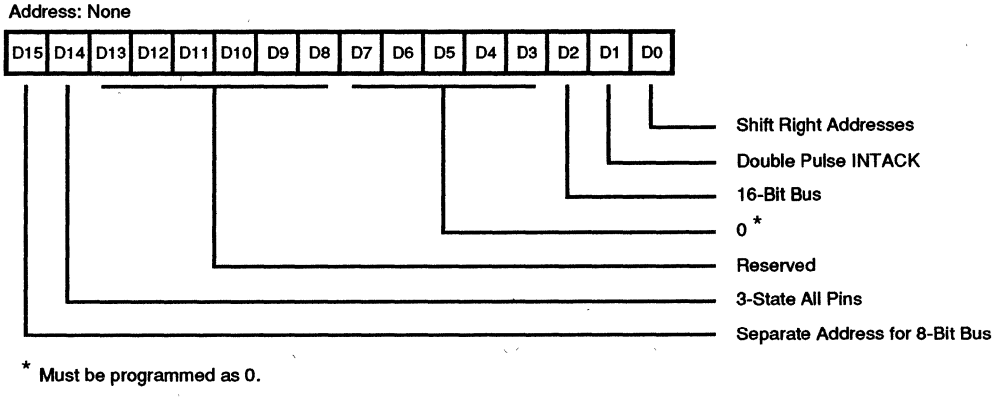


Figure 5. Bus Configuration Register (BCR)

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 via 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, via 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.

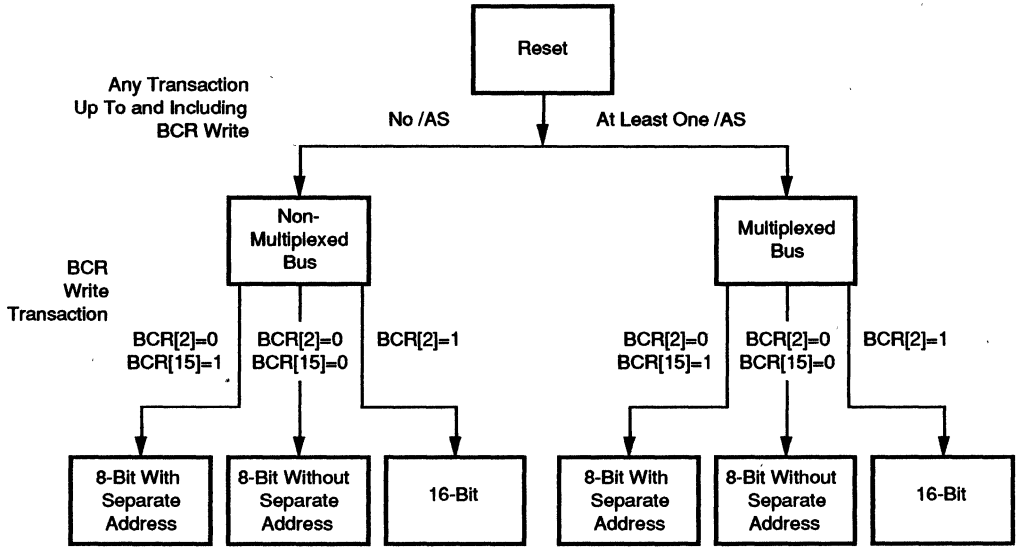
Non-Multiplexed Bus
In the non-multiplexed bus case, the registers in each channel are accessed indirectly using the address pointer in the Channel Command/Address Register (CCAR) of each channel. The address of the desired register is first written to the CCAR and then the selected register is accessed; 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-AØ			Address A4-AØ		
00000	CCAR	Channel Command/Address Register	10010	RCSR	Receive Command/Status Register
00001	CMR	Channel Mode Register	10011	RICR	Receive Interrupt Control Register
00010	CCSR	Channel Command/Status Register	10100	RSR	Receive Sync Register
00011	CCR	Channel Control Register	10101	RCLR	Receive Count Limit Register
00110	TMDR	Test Mode Data Register	10110	RCCR	Recieve Character Count Register
00111	TMCR	Test Mode Control Register	10111	TC0R	Time Constant 0 Register
01000	CMCR	Clock Mode Control Register	1X000	TDR	Transmit Data Register (Write Only)
01001	HCR	Hardware Configuration Register	11001	TMR	Transmit Mode Register
01010	IVR	Interrupt Vector Register	11010	TCSR	Transmit Command/Status Register
01011	IOCR	I/O Control Register	11011	TICR	Transmit Interrupt Control Register
01100	ICR	Interrupt Control Register	11100	TSR	Transmit Sync Register
01101	DCCR	Daisy-Chain Control Register	11101	TCLR	Transmit Count Limit Register
01110	MISR	Misc Interrupt Status Register	11110	TCCR	Transmit Character Count Register
01111	SICR	Status Interrupt Control Register	11111	TC1R	Time Constant 1 Register
1X000	RDR	Receive Data Register (Read Only)			
10001	RMR	Receive Mode Register	XXXXX	BCR	Bus Configuration Register

APPLICATION NOTE

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 for not only 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.

Note: 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); /N//S (NORMAL and SYSTEM are both active Low).

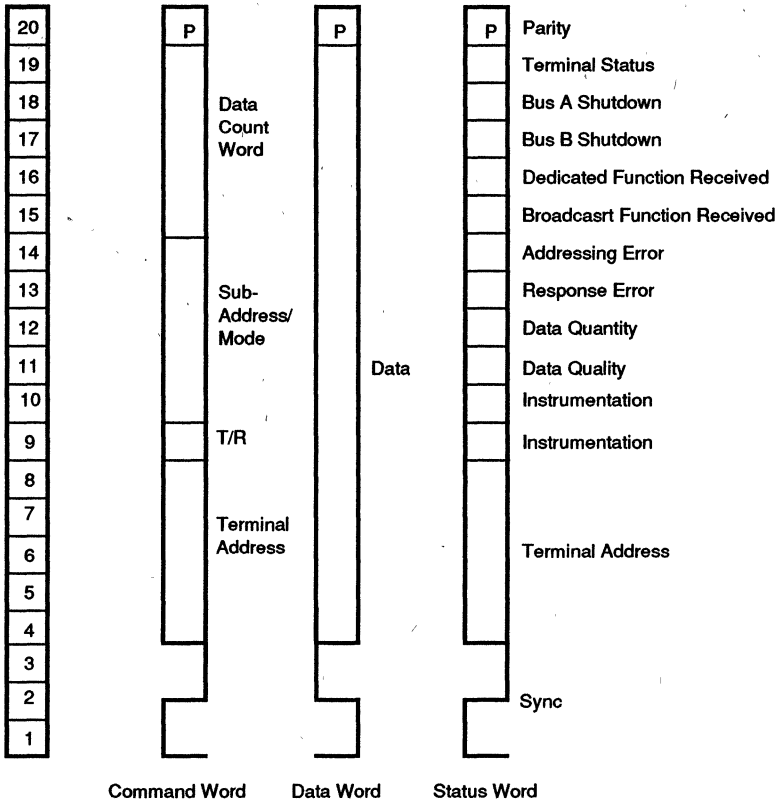


Figure 1. Message Formats

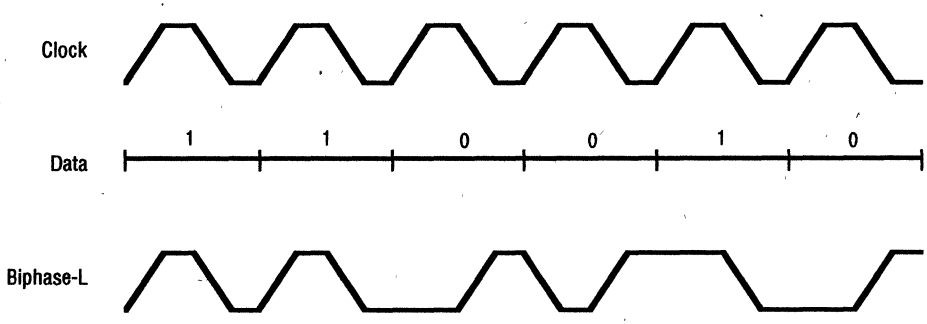


Figure 2. Manchester II (Biphase-Level) Data Encoding

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 byproduct 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.

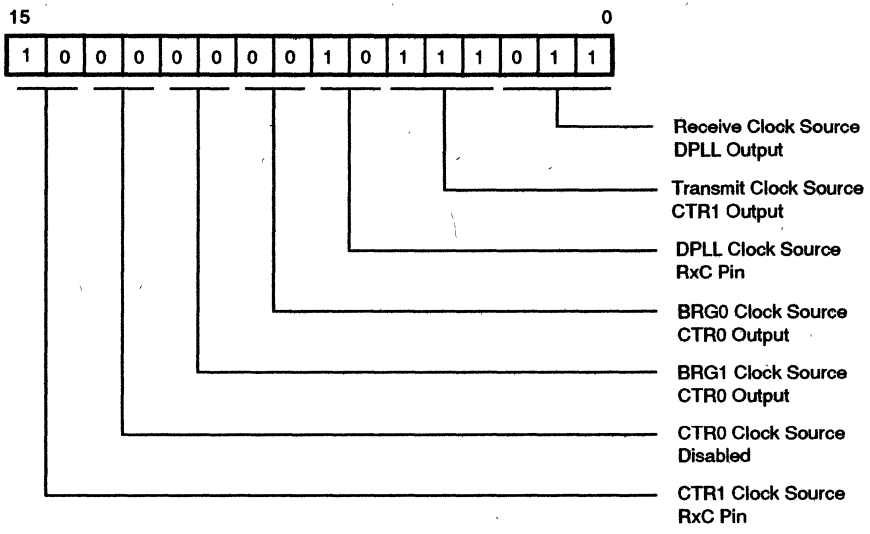


Figure 3. Clock Mode Control Register

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.

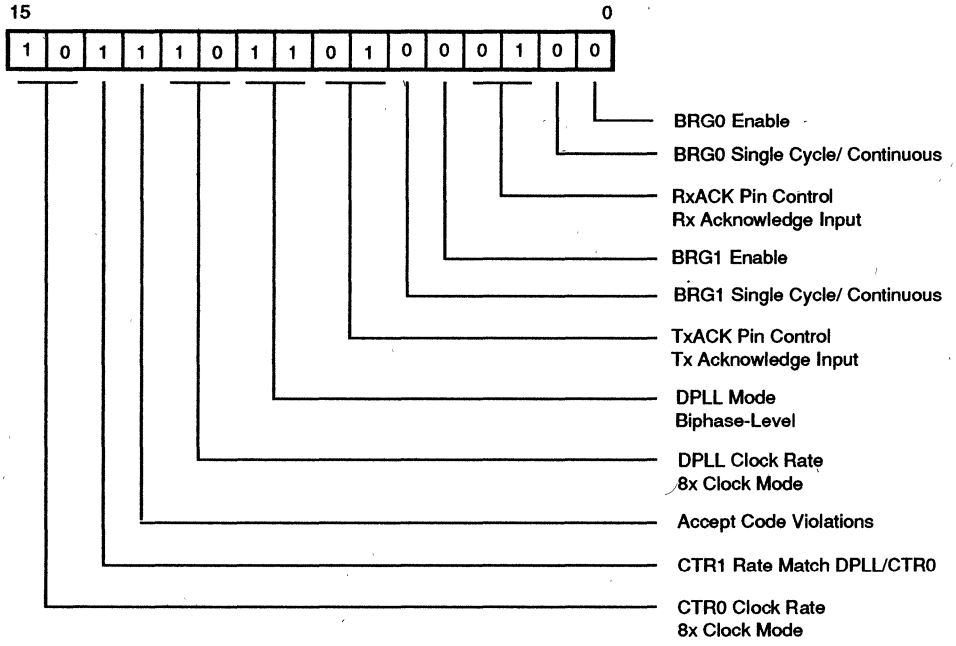


Figure 4. Hardware Configuration Register

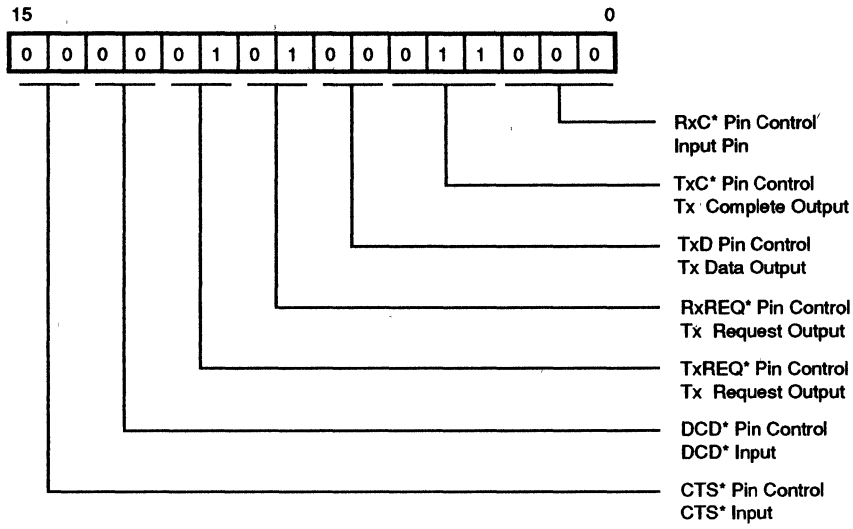


Figure 5. I/O Control Register

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 8.000MHz 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 Biphase-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.

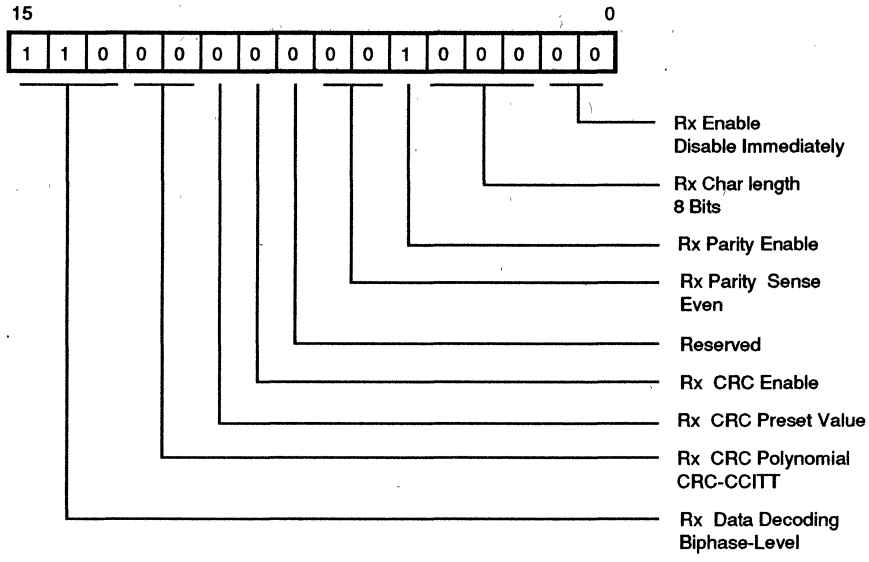


Figure 6. Receive Mode Register

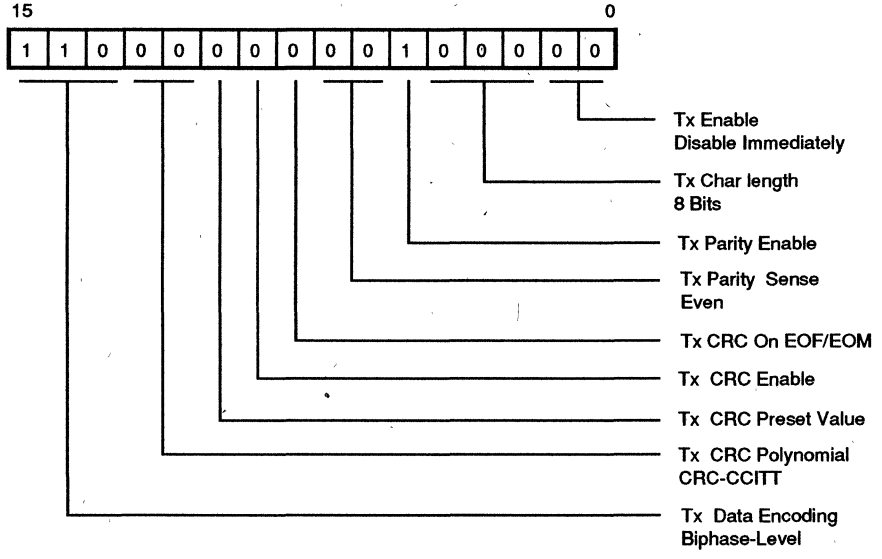


Figure 7. Transmit Mode Register

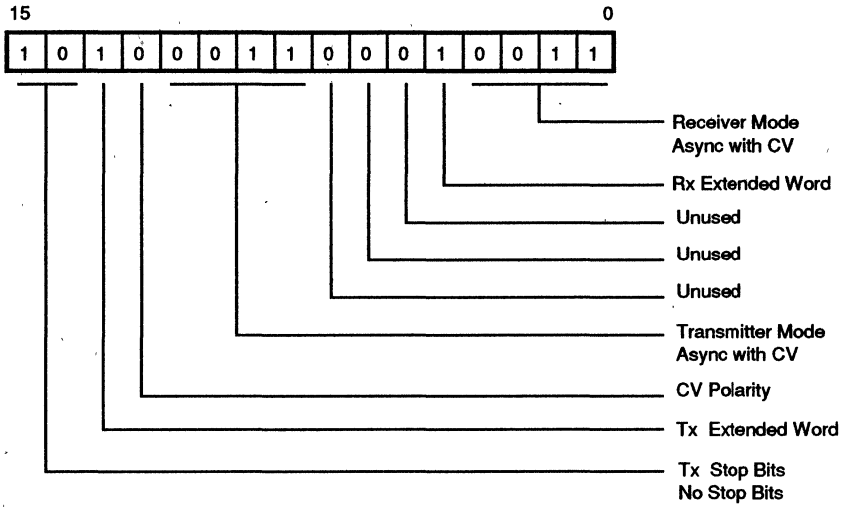


Figure 8. Channel Mode Register

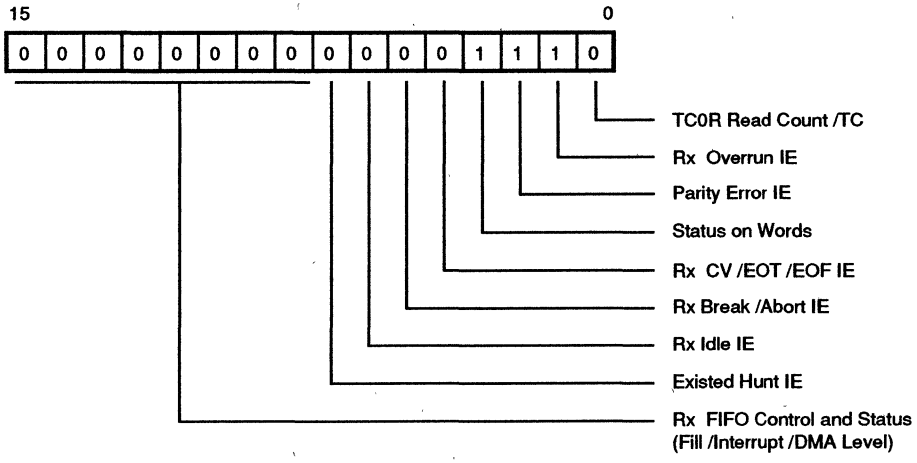


Figure 9. Receive Interrupt Control Register

At this point the interrupt and DMA details are programmed. Data transfer via 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 Biphase-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).

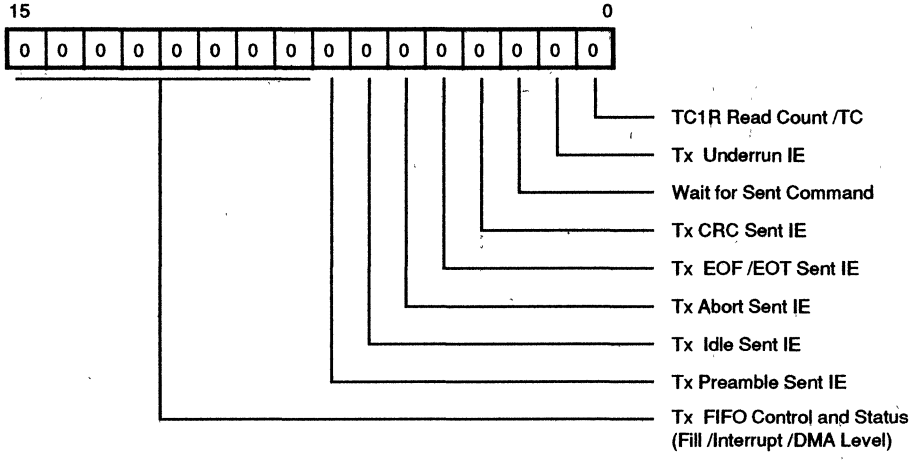


Figure 10. Transmit Interrupt Control Register

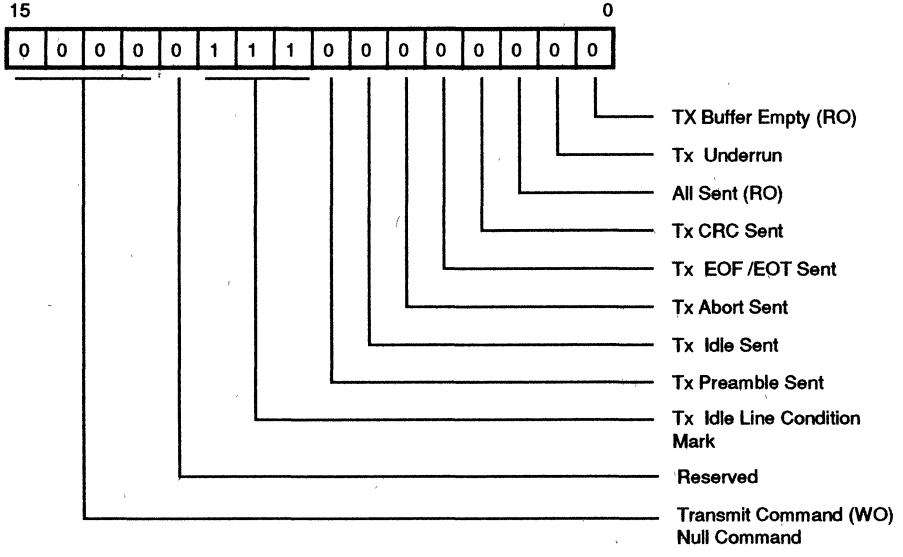


Figure 11. Transmit Command/Status Register

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.

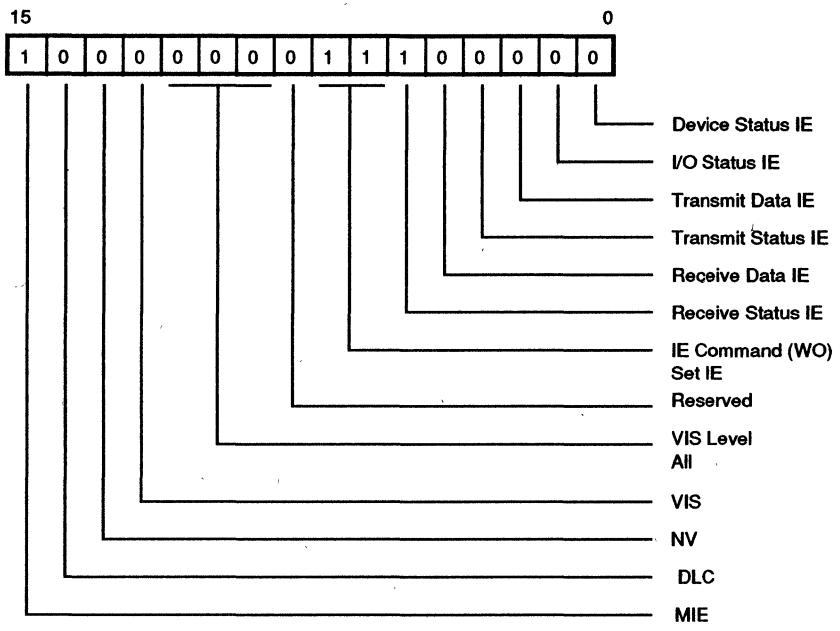


Figure 12. Interrupt Control Register

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

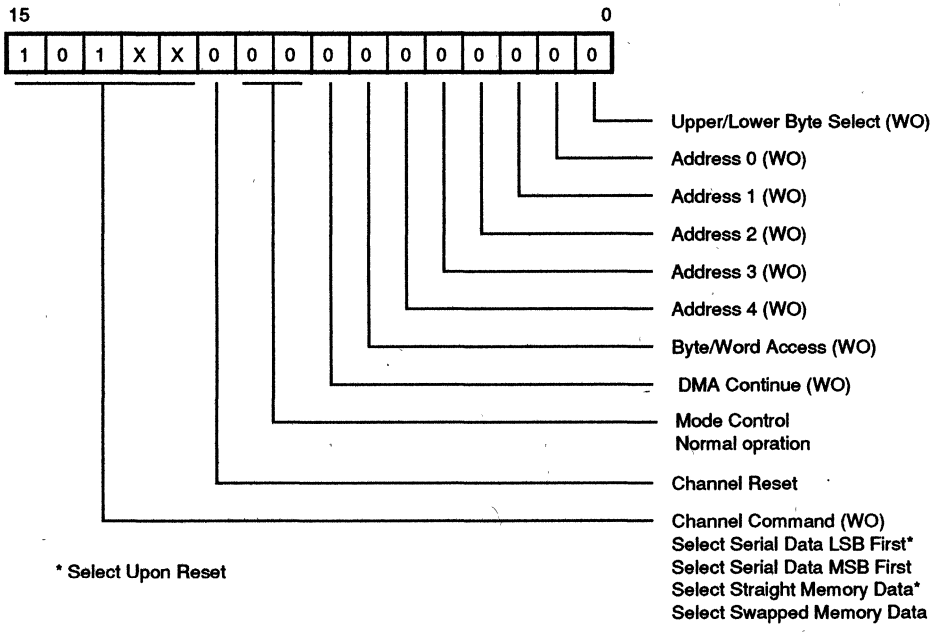


Figure 13. Channel Command/Address Register

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 byproduct 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.

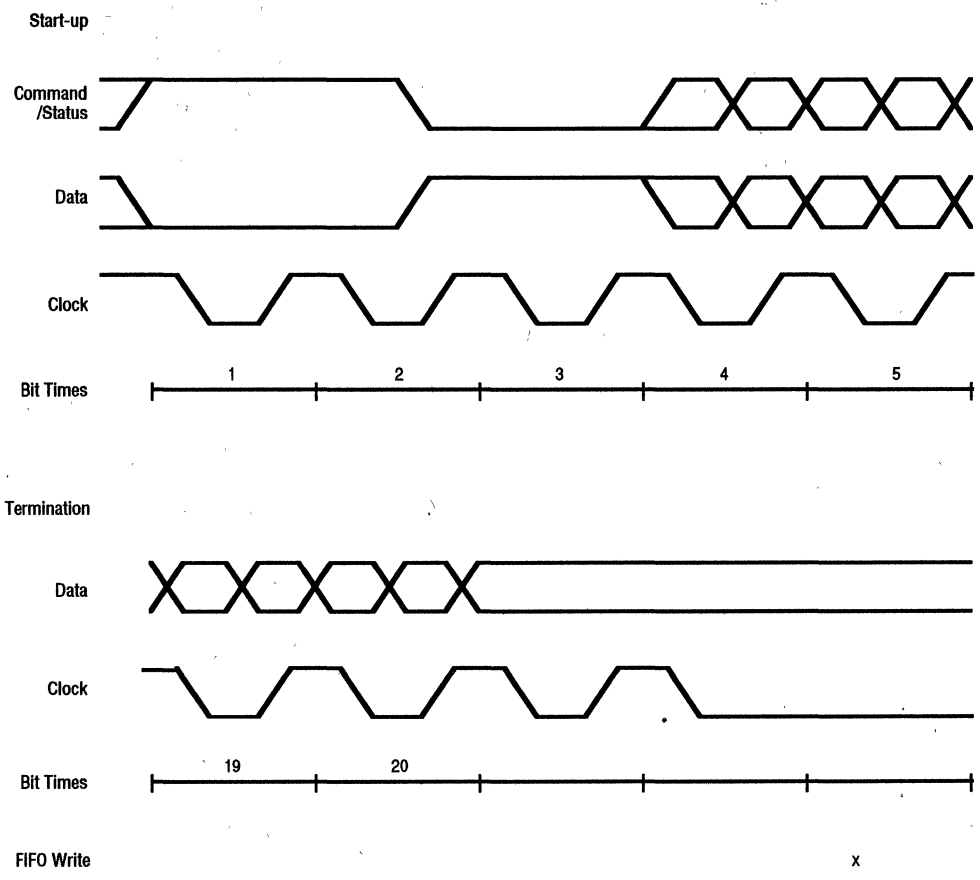


Figure 14. Receiver Clocking Requirements

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 Biphase-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 17).

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 that it sees when it is in search mode (waiting to sync up) is a valid clock edge. In Biphase-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.

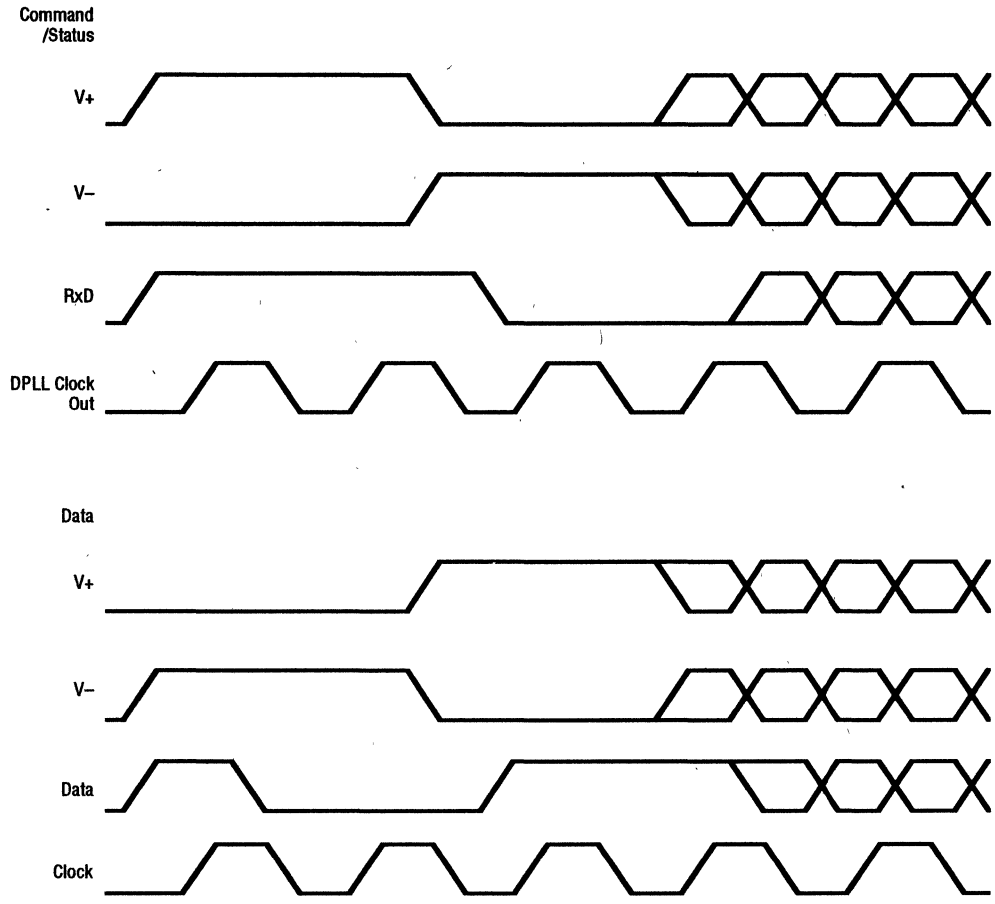
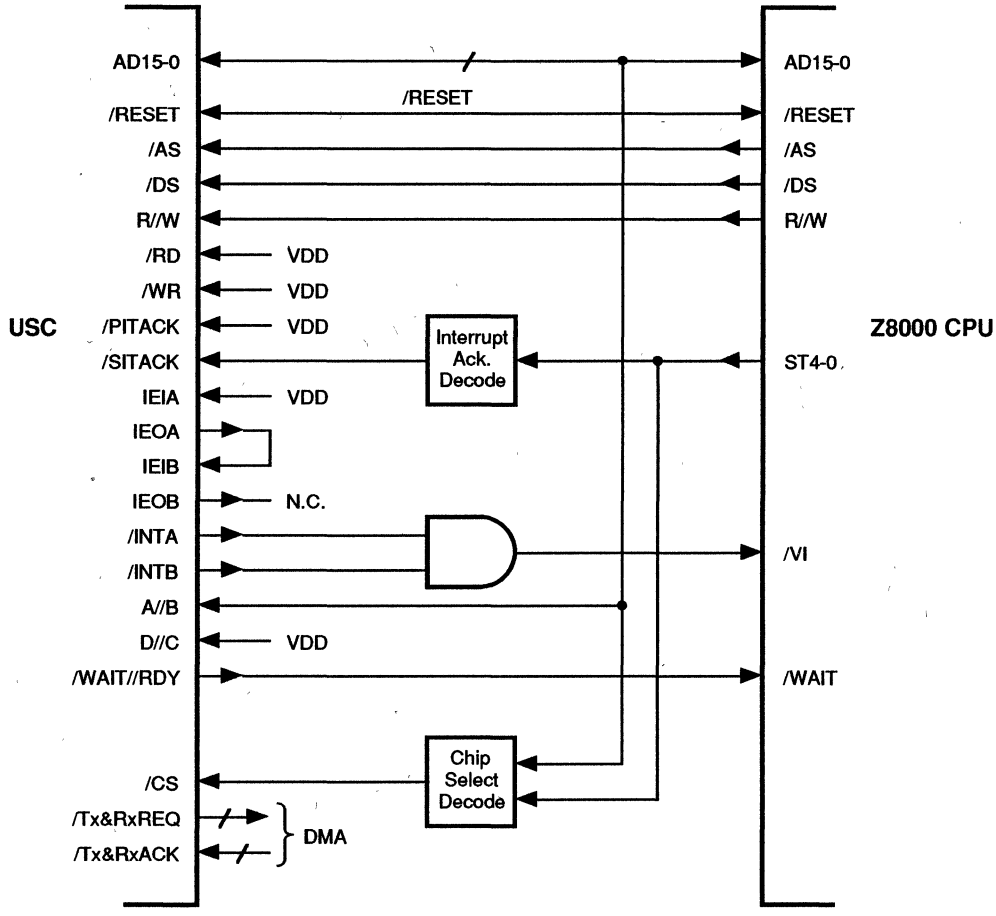


Figure 15. DPLL Idle Line Start-Up Requirements

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* is shown in Figure 16.



* CMOS Z8000 family CPU

Figure 16. USC to Z8000 CPU Connection (Example)

APPLICATION NOTE

DATACOMMUNICATIONS IUSC/MUSC TIME SLOT ASSIGNER

Use the IUSC/MUSC for ISDN and Fractional T1 in High Speed, Time-Multiplexed Datacommunications

INTRODUCTION

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 Time Slot Assigner logic (TTSA and RTSA).

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.

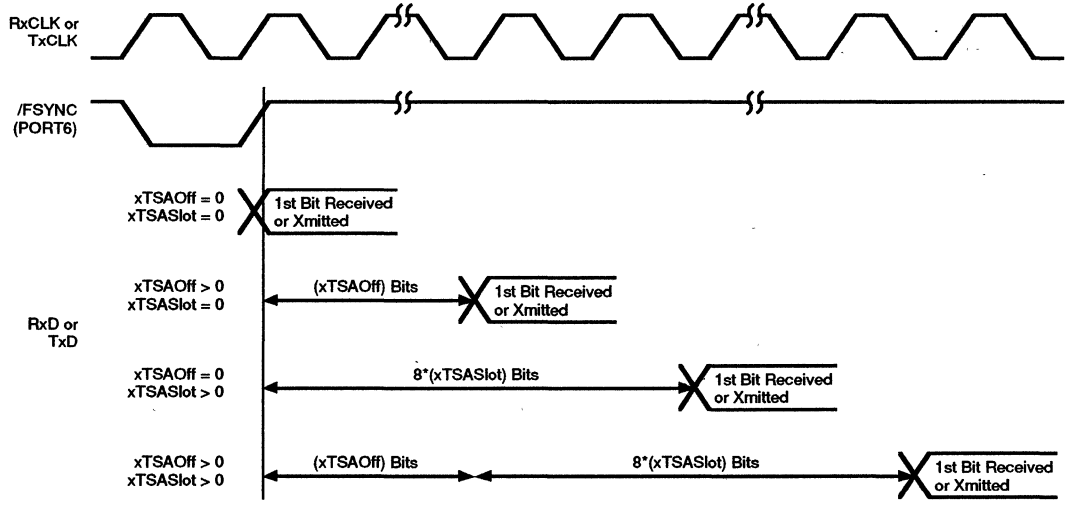


Figure 1. Start of Received or Transmitted Data in a TSA Application

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 8(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.

Note: 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); /N//S (NORMAL and SYSTEM are both active Low).

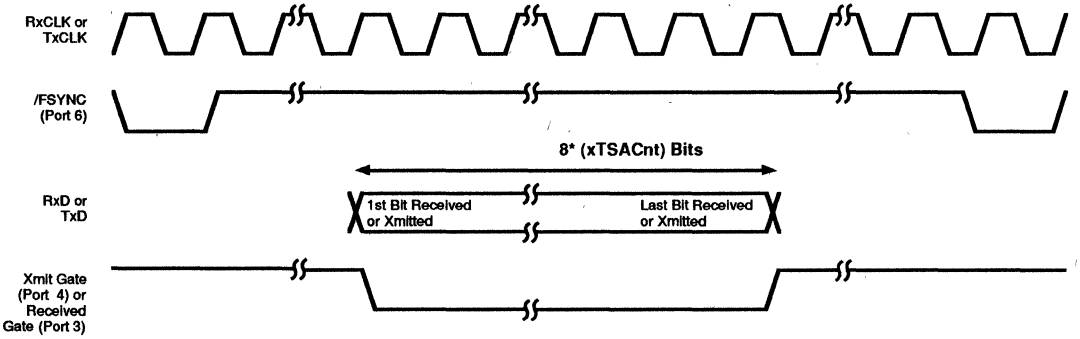


Figure 2. Length of Received or Transmitted Data in a TSA Application

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 8(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 8(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 low-active 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 (PCR9-8 and/or PCR7-6, respectively) as 01.

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 less-significant byte (bits 7-0) of both the RICR and TICR contains fixed data, but any of five different internal registers can be selected as the more-significant byte of each register (Figure 3). At the first level of data structure, the contents of RICR15-8 are selected by means of four of the commands that are written to the RCMD field of the Receive Command/Status Register (RCSR15-12); the contents of TICR15-8 are selected by four of the commands written to the TCMD field of the Transmit Command/Status Register (TCSR15-12). The encoding of both sets of commands is the same:

xCMD	Contents of xICR15-8
0100	xTSA data
0101	Current xFIFO Level
0110	xFIFO Level for Interrupt
0111	xFIFO Level for DMA Request

(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 xICR15-8, if the 0100 command has been written to xSCR15-12 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 LSBit of the MSByte of data written:

xICR8 value	Destination of xICR15-9
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 LSBit of the MSByte equal to zero.

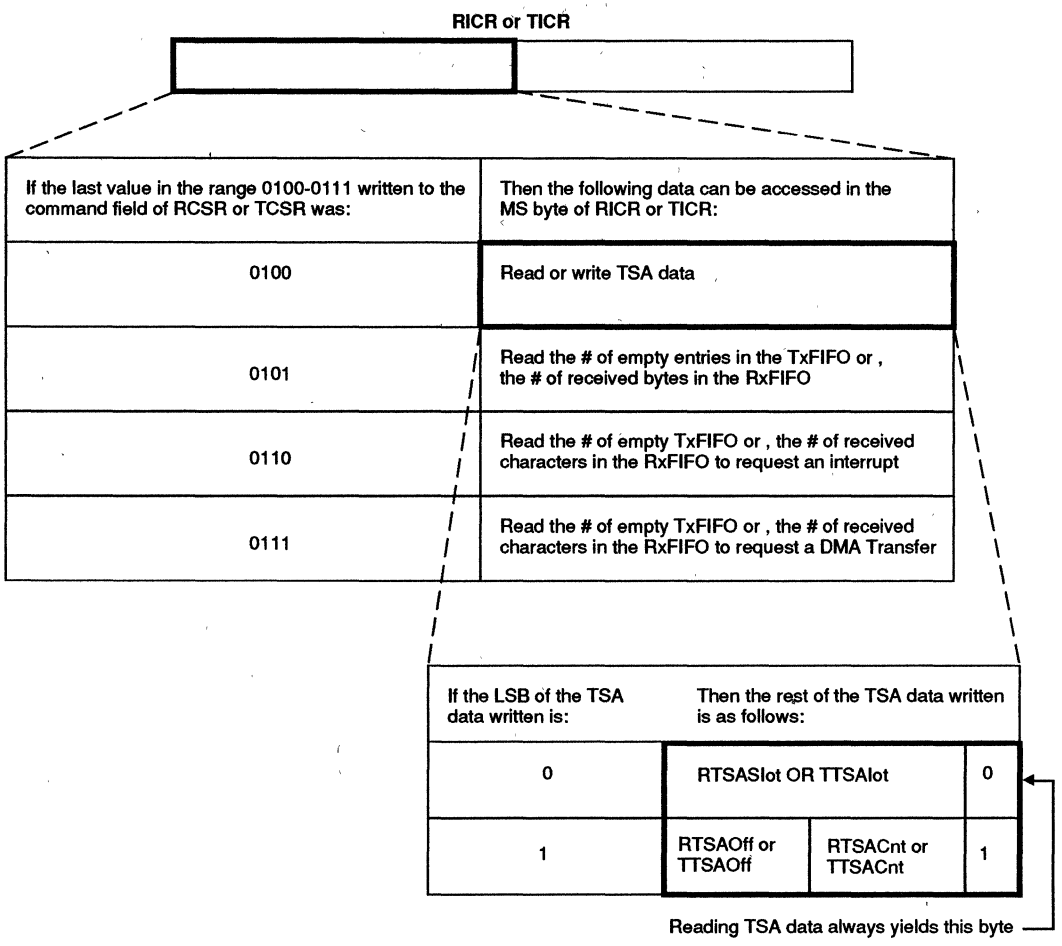


Figure 3. Structure of RICR and TICR

SUMMARY

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

ZiLOG

APPLICATION NOTE

ISCC INTERFACE TO THE 68000 AND 8086

INTRODUCTION

The ISCC uses its flexible bus to interface with a variety of microprocessors and microcontrollers; included are the 68000 and 8086.

The Z16C35 ISCC is a Superintegration form of the 85C30/80C30 Serial Communications Controller (SCC). Super integration includes four DMA channels, one for each receiver and transmitter and a flexible Bus Interface Unit (BIU). The BIU supports a wide variety of buses including the bus types of the 680x0 and the 8086 families of microprocessors.

This Application Note presents the details of BIU operation for both slave peripheral and DMA modes. Included are application examples of interconnecting an ISCC to a 68000 and a 8086 (These examples are currently under test).

ISCC BUS INTERFACE UNIT (BIU)

The following subsections describe and illustrate the functions and parameters of the ISCC Bus Interface Unit

Overview

The ISCC contains a flexible bus interface that is directly compatible with a variety of microprocessors and microcontrollers. The bus interface unit adds to the chip by allowing ease of connection to several standard bus configurations; among others are the 68000 and the 8086 family microprocessors. This compatibility is achieved by initializing the ISCC after a reset to the desired bus configuration.

The device also configures to work with a variety of other 8- or 16-bit bus systems and is used with address/data multiplexed or non-multiplexed buses. In addition, the wait/ready handshake, the interrupt acknowledge, and the bus high byte/low byte selection are all programmable. Separate read/write, data strobe, write, read, and address strobe signals are available for direct system interface with a minimum of external logic.

Modes Description

There are basically two bus modes of operation: multiplexed and non-multiplexed. In the multiplexed bus mode, the ISCC internal registers are directly accessible as separate registers with their own unique hardware addresses. By contrast, in the non-multiplexed mode, all registers access through an internal pointer which first loads with the register address. Loading of the pointer is done as a data write. In either case, there are some external addressing signals.

Chip Enable (CE) allows external selection through the decode of upper order address bits like accessing separate chips. A separate input (not part of the AD15-0 bus connection) selects between the internal SCC and DMA sections of the chip. This input is A0/SCC/DMA and provides direct transfers to the appropriate chip subsystem; either multiplexed or non-multiplexed bus mode.

A second separate input (not part of the AD15-0 bus connection) provides for a selection between the internal SCC; both channels A and B (Table A-1). This input is A1/A/B and provides direct transfers to the appropriate SCC channel when A0/SCC/DMA selects the SCC; either multiplexed or non-multiplexed bus mode. Note that these two signals, A1/A/B and A0/SCC/DMA, are inputs when the ISCC is a slave peripheral; they become outputs when the ISCC is a bus master during DMA operations.

Table A-1. Accessing the ISCC Registers

A0/SCC/DMA	A1/A/B	ACCESS
1	1	SCC Channel A
1	0	SCC Channel B
0	x	DMA

The following discussions assume knowledge of the SCC Serial Communications Controller opertions and refer to internal register designations. For a detailed explanation, refer to the SCC Technical Manual.

Non-multiplexed Bus Operation

When the ISCC initializes for non-multiplexed operation, Write Register 0 (WR0) takes on the form of WR0 in the Z8530, Write Register Bit Functions (Figure A-1). Register addressing for the SCC section is (except for WR0 and RR0) accomplished as follows. Programming the write registers requires two write operations. Reading the read registers requires both a write and a read operation.

The first write is to WR0 which contains three bits that point to the selected register (note the point high command). The second write is the actual control word for the selected register. If the second operation is a read, the selected register is accessed. When in the non-multiplexed mode, all registers in the SCC section of the ISCC, including the data registers, access this way.

The pointer register automatically clears after the second read or write operation so WR0 (or RR0) addresses again. There is no direct access to the data registers. They are addressed through the pointer (this is in contrast to the Z8530 which allows direct addressing of the data registers through the C/D pin).

When the ISCC starts for non-multiplexed operation, register addressing for the DMA section is (except for CSAR) accomplished as follows. It is completely independent of the SCC section register addressing. Programming the write registers requires two write operations and reading the read registers requires both a write and a read operation. The first write is to the Command Status Address Register (CSAR) which contains five bits that point to the selected register (CSAR bits 4 - 0). The second write is the actual control word for the selected register. If the second operation is a read, the selected register is accessed.The pointer bits automatically clear after the second read or write operation so CSAR addresses again. When in the non-multiplexed mode, all registers in the DMA section of the ISCC are accessed.

Multiplexed Bus Operation

When the ISCC initializes for multiplexed bus operation, all registers in the SCC section are directly addressable with the register address occupying AD5 through AD1 or AD4 through AD0 (Shift Left/Shift Right modes).

The Shift Left/Shift Right modes for the address decoding of the internal registers (multiplexed bus) are separately programmable for the SCC and DMA sections For the SCC section, the programming and operation is the same as the SCC; programming occurs through Write Register 0 (WR0), bits 1 and 0 , and Write Register Bit Functions (Figure A-2). The programming of the Shift Left/Shift Right modes for the DMA section occurs in the BCR, bit 0. In this case, the shift function is similar to the SCC section; with Left Shift, the internal register addresses decode from bits AD5 through AD1. In Right Shift, the internal register addresses decode from bits AD4 through AD0.

During multiplexed bus mode selection, Write Register 0 (WR0) becomes WR0 in the Z8030, Write Register Bit Functions (Figure A-2).

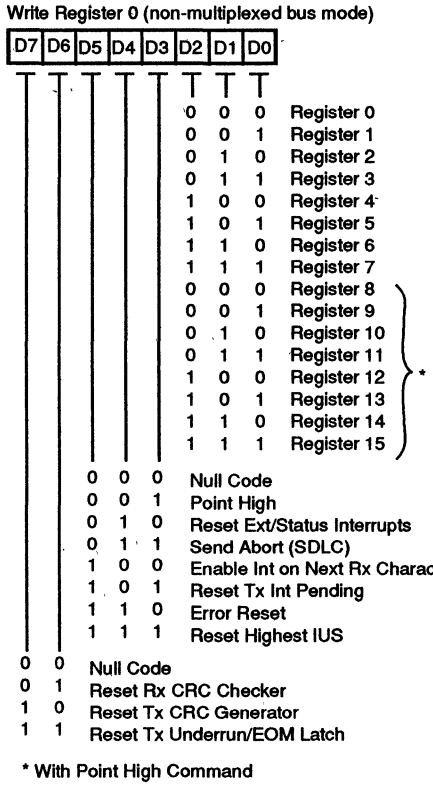


Figure A-1. Write Register 0 Bit Functions (Non-Multiplexed Bus Mode)

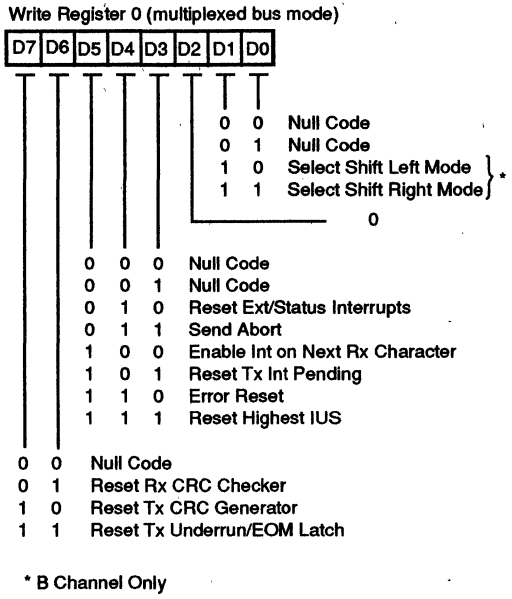


Figure A-2. Write Register 0 Bit Functions (Multiplexed Bus Mode)

BUS DATA TRANSFERS

All data transfers to and from the ISCC are done in bytes regardless of whether data occupies the lower or upper byte of the 16 bit bus. Bus transfers as a slave peripheral are done differently from bus transfers when the ISCC is the bus master during DMA transactions. The ISCC is fundamentally an 8-bit peripheral but supports 16-bit buses in the DMA mode. Slave peripheral and DMA transactions appear in the next sections.

Data Bus Transfers as a Slave Peripheral

When accessed as a peripheral device (when the ISCC is not a bus master performing DMA transfers), only 8 bits transfer. During ISCC register read, the byte data present on the lower 8 bits of the bus is replicated on the upper 8 bits of the bus. Data is accepted by the ISCC only on the lower 8 bits of the bus.

ISCC DMA Bus Transfers

During DMA transfers, when the ISCC is bus master, only byte data transfers occur. However, data transfers to or from the ISCC on the upper 8 bits of the bus or on the lower 8 bits of the bus. Moreover, odd or even byte transfers activate on the lower or upper 8 bits of the bus. This is programmable and explained next.

During DMA transfers to memory from the ISCC, only byte data transfers occur. Data appears on the lower 8 bits and replicates on the upper 8 bits of the bus. Thus, the data is written to an odd or even byte of the system memory by address decoding and strobe generation.

During DMA transfers to the ISCC from memory, byte data only transfers. Normally, data appears only on the lower 8

bits of the bus. However, the byte swapping feature determines which byte of the bus data is accepted. The byte swapping feature activates by programming the Byte Swap Enable bit to a 1 in the BCR. The odd/even byte transfer selection occurs by programming the Byte Swap Select bit in the BCR. If Byte Swap Select is a 1, then even address bytes (transfers where the DMA address has A0 = 0) are accepted on the lower 8 bits of the bus. Odd address bytes (transfers where the DMA address has A0 = 1) are accepted on the upper 8 bits of the bus. If Byte Swap Select is a 0, then even address bytes (transfers where the DMA address has A0 = 0) are accepted on the upper 8 bits of the bus. Odd address bytes (transfers where the DMA address has A0 =1) are accepted on the lower 8 bits of the bus.

Bus Interface Handshaking

The ISCC supports data transfers by either a data strobe (DS) combined with a read/write (R/W) status line, or separate read (RD) and write (WR) strobes. These transactions activate via chip enable (CE).

ISCC programming generates interrupts upon the occurrence of certain internal events. The ISCC internally prioritizes its own interrupts, therefore, the ISCC presents one interrupt to the processor even though lower priority internal interrupts may be pending. Interrupts are individually enabled or disabled. Refer to the sections on the SCC core.

Interrupt Acknowledge (INTACK) is an input to the ISCC showing that an interrupt acknowledge cycle is progressing. INTACK is programmed to accept a status acknowledge, a single pulse acknowledge, or a double pulse acknowledge. This programming activates in the BCR. The double pulse acknowledge is compatible with 8X86 family microprocessors and the status acknowledge is compatible with 68000 family microprocessors.

During an interrupt acknowledge cycle, the SCC and DMA interrupt priority daisy chain internally resolves. Thus, the highest priority internal interrupt is presented to the CPU. The ISCC can return an interrupt vector that encodes with the type of interrupt pending enabled during this acknowledge cycle. The ISCC may request an interrupt but not return an interrupt vector [note that the no vector bit(s) in the SCC section (WR9 bit 1) and in the DMA section (ICR bit 5) individually control whether or not an interrupt vector returns by these cores]. The interrupt vector can program to include a status field showing the internal ISCC source of the interrupt. During the interrupt acknowledge cycle, the ISCC returns the interrupt vector when INTACK, RD or DS go active and IEI is high (if the ISCC is not programmed for the no vector option).

During the programmed pulsed acknowledge type (whether single or double), INTACK is the strobe for the interrupt vector. Thus when INTACK goes active, the ISCC drives the bus and presents the interrupt vector to the CPU. When the status acknowledge type programs, the ISCC drives the bus with the interrupt vector when RD or DS are active.

WAITRDY programs to function either as a WAIT signal or a READY signal using the BCR write. When programmed as a wait signal, it supports the READY function of 8X86 family microprocessors. When programmed as a ready signal, it supports the DTACK function of 680x0 family microprocessors.

The WAIT/RDY signal functions as an output when the ISCC is not a bus master. In this case, 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.

When the ISCC is the bus master (DMA section has taken control of the bus), the WAIT/RDY signal functions as a WAIT or RDY input. Slow memories and peripheral devices use WAIT to extend the data strobe (/DS) during bus transfers. Similarly, memories and peripheral devices use RDY to indicate valid output or that it is ready to latch input data.

CONFIGURING THE BUS

The bus configuration programming is done in two separate steps (actually it is one operation), to enable the write to the Bus Configuration Register (BCR). The first operation that accesses the ISCC after a device reset must be a write to the BCR since this is the only time that the BCR is accessible. Before and during the write, various external signals are sampled to program bus configuration parameters. During this write, the AØ/SCC//DMA pin must be Low.

Address strobe programs multiplexed/non-multiplexed selection. In a non-multiplexed bus environment, address strobe (as an input) is not used but tied high through a suitable pull-up resistor. Thus, no address strobe is present before the BCR write. Then, when write to the BCR takes place, the non-multiplexed mode is programmed because there is no address strobe before this first write to the device. Note that address strobe becomes an output during DMA operations so it is not tied directly to Vcc.

During the write operation to the BCR, the A1/A/B input is sampled to select the function of the WAIT/RDY pin (Table A-2). When the BCR Write is to the SCC Channel A (A1/A//B High during the BCR write), the WAIT/RDY signal functions as a wait. When the BCR Write is to Channel B (A1/A//B Low during the BCR write), the WAIT/RDY signal functions as a ready.

Table A-2. Signals Sampled During the BCR Write

A1/A//B	WAIT/RDY Function
1	WAIT (8086 RDY compatible)
0	READY (68000 DTACK compatible)

This programming affects the function of the WAIT/RDY signal both as an input, when the ISCC is bus master during DMA operations, and as an output when the ISCC is a bus slave.

With this programming, the ISCC is immediately configured to function successfully on this first and subsequent bus transactions. The remaining bus configuration options are programmed by the value written to the BCR.

Bit 0 of the BCR controls the Shift Left/Shift Right address decoding modes for the DMA section. In this case, the shift function is similar to the SCC section. During Left Shift, the internal register addresses decode from bits AD5 through AD1. During Right Shift, the internal register addresses are decode from bits AD4 through AD0. This function is only applicable in the multiplexed bus mode.

Bits 1 and 2 of the BCR control the interrupt acknowledge type as shown in the Table A-3.

Table A-3. BCR Control of Interrupt Acknowledge

BCR bit 2	BCR bit 1	Interrupt Acknowledge
0	0	Status Acknowledge
0	1	Pulsed Acknowledge (single)
0	1	Reserved (action not defined)
1	1	Double Pulsed Acknowledge

The Status Acknowledge remains active throughout the interrupt cycle and is directly compatible with the 680x0 family interrupt handshaking. The Status Acknowledge signal latches with the rising edge of AS for multiplexed bus operation. It latches by the falling edge of the strobe (RD or DS) for non-multiplexed bus operation. The Pulsed Acknowledges are timed to be active during a specified period in the interrupt cycle. The Double Pulsed Acknowledge is directly compatible with the 8x86 family interrupt handshaking. Refer to the timing diagrams in the ISCC Product Specification for details on the Acknowledge signal operation.

Reserve bits 3, 4, and 5 of the BCR program as zeros. Bits 6 and 7 of the BCR control the byte swap feature (Table A-4). Byte swap is applicable only in DMA transfers when the ISCC is the bus master and only affects ISCC data acceptance (transfers from memory to the ISCC).

Table A-4. Byte Swap Control

Enable (BCR bit 7)	DMA Data Read by the ISCC
0	lower 8 bits of bus only
1	upper or lower 8 bits of bus

Swap Select*	A0	DMA Data read by the ISCC
0	0	upper 8 bits of bus
0	1	lower 8 bits of bus
1	0	lower 8 bits of bus
1	1	upper 8 bits of bus

* BCR bit 6

APPLICATIONS EXAMPLES

The following application examples explain and illustrate the methods of interfacing the ISCC to a Motorola 68000 and an Intel 8086.

68000 Interface to the ISCC

Figure A-3 shows a connection of the ISCC to a 68000 microprocessor. The 68000 data bus connects directly, or through bus transceivers, to the ISCC address/data bus. R/W and RESET also directly connect In this example, the ISCC is on the lower half of the bus; DS of the ISCC connects to LDS of the 68000. The processor address lines decode to produce a chip enable for the ISCC. In addition, processor addresses A1 and A2 connect to A0/SCC/DMA and A1/A/B, respectively, through a tri-state driver.

The driver is normally ON (enabled) but turns OFF by BGACK to grant the bus to ISCC for DMA transfers. This is done since the A0/SCC/DMA and A1/A/B pins become outputs during DMA transfers and should not drive the system address bus. RD and WR tie high through independent pull-ups. They are not used in this application but become active outputs during DMA transfers and are not tied directly to Vcc.

Although not shown in Table A-5, the A0/SCC/DMA and A1/A/B pins may be decoded during DMA transfers to identify the active DMA channel.

Table A-5. DMA A/B Channel Decode

A1/A/B	A0/SCC/DMA	DMA Channel
1	1	Receiver Channel A
1	0	Transmitter Channel A
0	1	Receiver Channel B
0	0	Transmitter Channel B

External logic can use this information to abort a DMA in progress.

For normal slave device bus interaction, a DTACK is generated. WAIT/RDY is programed for ready operation and INTACK programs for the status type. WAIT/RDY generates a DTACK for normal data transfers and interrupt responses. Additional logic may be required when other interrupt sources are present.

During DMA transfers, the ISCC becomes bus master. Becoming bus master is done through the BUSREQ output and BUSACK input signals of the ISCC. They connect to an external bus arbitration circuit. This circuit performs bus arbitration for multiple bus master requests and generates bus grant acknowledge (BGACK) which controls certain bus drive signal sources.

When the ISCC becomes the bus master, a 32-bit address generation by the DMA section is output on the ISCC address/data bus. The lower 16 bits of this address store in an external latch by AS (Address Strobe). Also, the upper 16 bits of this address store in an external latch by UAS (Upper Address Strobe). With BGACK low (active) and with the processor address lines tri-stated, the latch outputs drive the system address bus.

AS is pulled high by an external resistor. This pull-up insures an inactive AS (at a logic high level) when the ISCC is not driving this signal. Therefore, on power up or after a RESET, AS is inactive and programs the non-multiplexed bus mode on BCR write.

In this application, the outputs of the address latches are connected to the address bus so that A1 through A23 of the ISCC drives the system address bus (the ISCC provides a total of 32 address lines). A0 from the address latch is diverted to logic which generates UDS and LDS bus signals from the ISCC data strobe (DS). UDS is generated when A0 is low and LDS is generated when A0 is high. The lower and upper data strobes are applied to the system bus through tri-state drivers which are enabled only when BGACK is active. Bus direction is now controlled by the ISCC R/W signal which is now an output.

For initialization, the BCR write (the first write to the ISCC after RESET) is done with A2 = 0 (A1/A/B ISCC input at logic low). This selects the ready option of the WAIT/RDY signal to conform to the 68000 bus style. The AS signal programming of the non-multiplexed bus has already been discussed. The BCR is written with C0h to enable byte swapping. It also selects the sense of byte swapping with respect to A0 appropriate to this bus style and selects the STATUS type of interrupt acknowledge.

8086 Interface with the ISCC

Figure A-4 shows the connection of the ISCC to an 8086 microprocessor and companion clock state generator. In this application, the ISCC connects for multiplexed address access to the internal ISCC registers. AD15 through AD0 of the 8086 connect directly, or through a bus transceiver, to the corresponding AD15 through AD0 address/data ISCC bus pins. RD and WR are directly compatible and tie together to form the read and write bus signals.

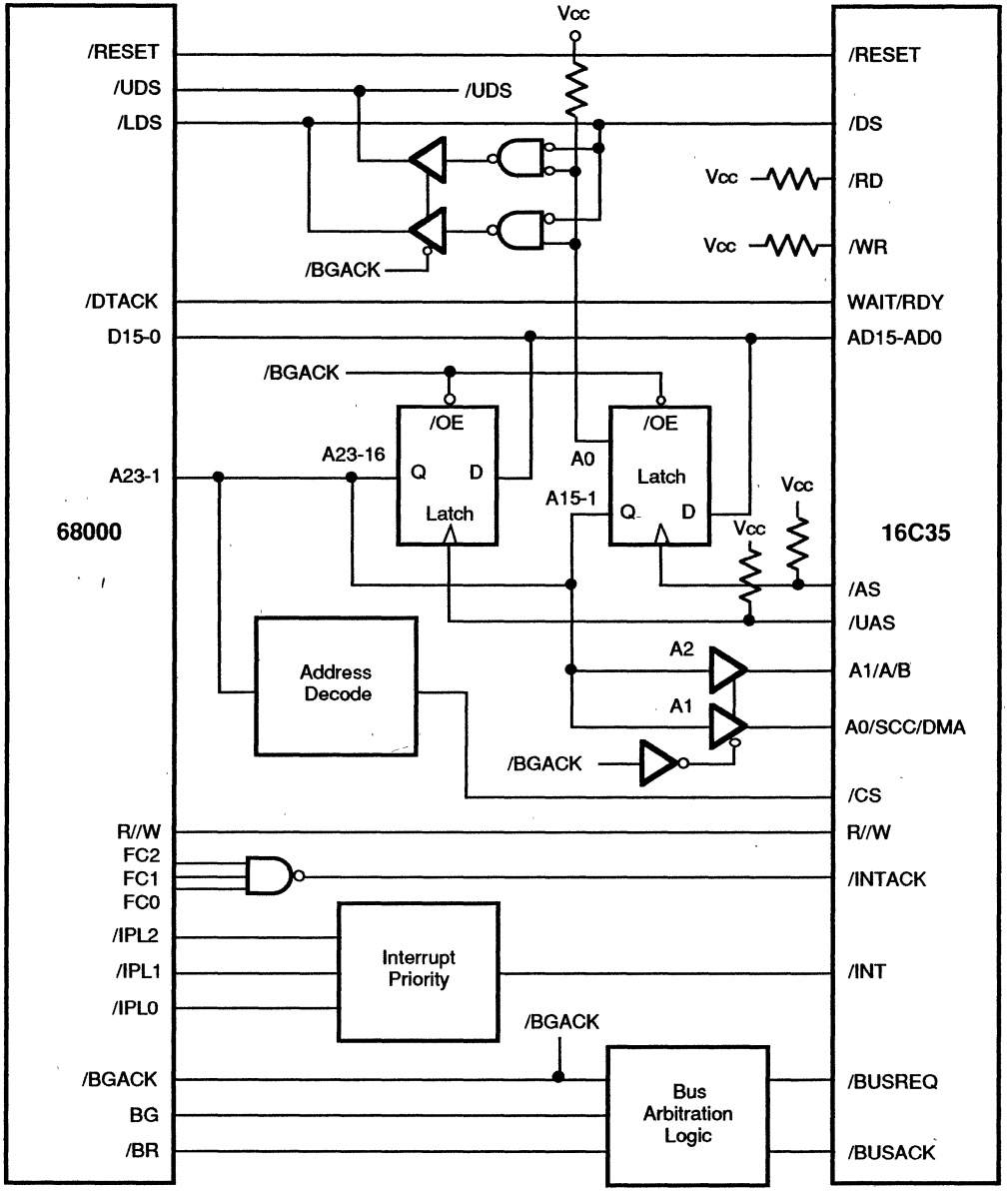
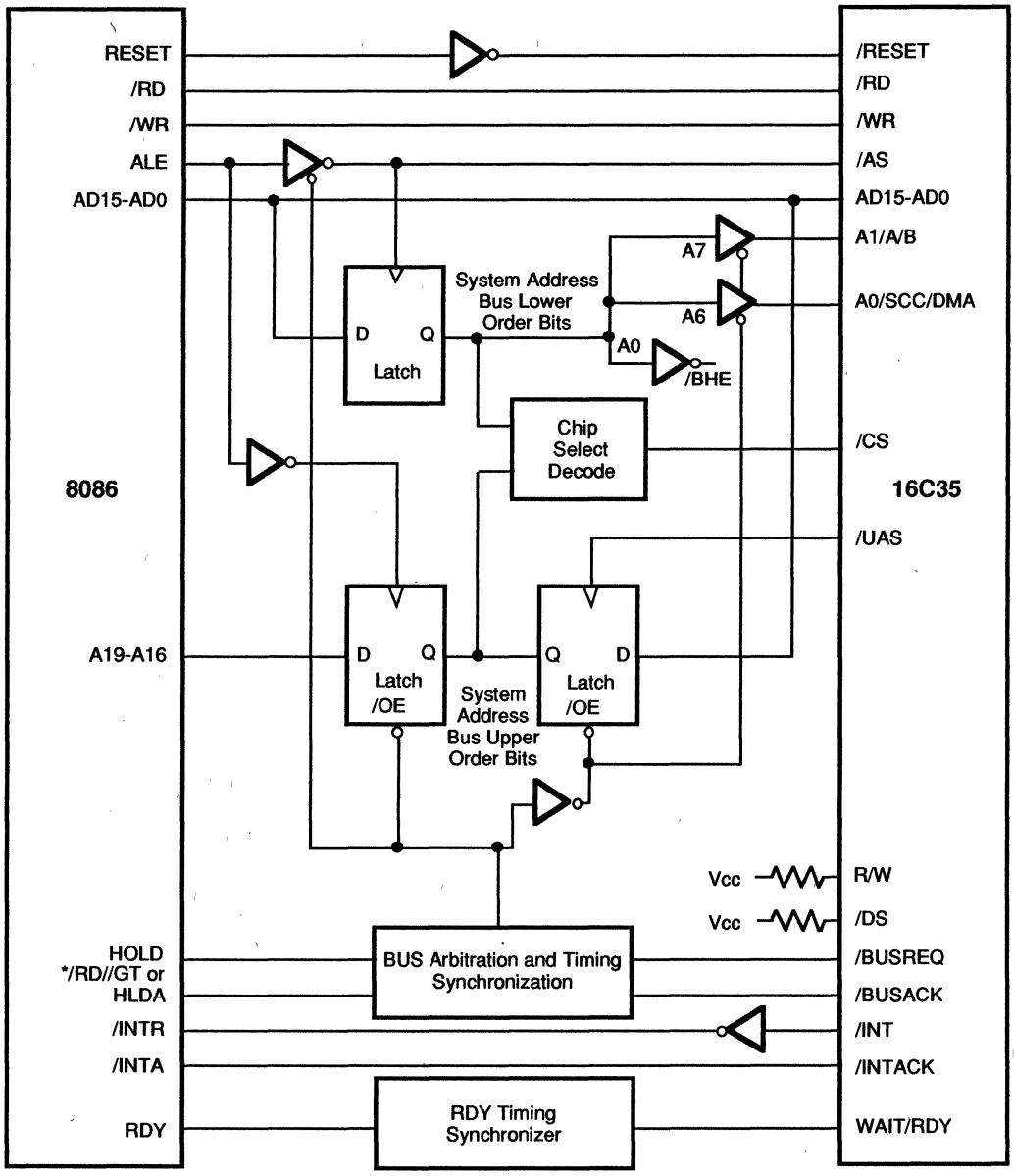


Figure A-3. ISCC Interface to a 68000 Microprocessor



* maximum mode

Figure A-4. ISCC Interface to an Intel 8086 Microprocessor

When the ISCC becomes a bus master during DMA operations, RD and WR of the 8086 are tri-stated which allows the corresponding ISCC signals to control the bus transactions. The sense of RESET reverses, so the ISCC RESET signal inverts from the reset applied to the 8086 from the clock state generator.

RD/WR and DS of the ISCC are inactive in this application and tie high. They tie high through independent pull-ups since these signals become active when the ISCC is bus master during DMA transactions.

Assuming other devices in the system, the ISCC chip enable input (CE) activates from a decode of the address. In this example, the ISCC internally decodes addresses A1 through A5 and uses A6 and A7, externally. Thus, the address decode circuitry decodes address lines A0 and A8 and above. The decode of A0 for chip enable places the ISCC as an 8-bit peripheral on the lower byte of the bus. A0 and the upper level address lines (including A6 and A7) demultiplex from the 8086 address/data bus through a latch strobed by ALE.

The demultiplexed addresses A6 and A7 connect to A0/SCC/DMA and A1/A/B, respectively, of the ISCC to control selection of the DMA and SCC channels A and B. This connects through the tri-state drivers. They enable when the 8086 is the bus master and disable when the ISCC is bus master. This prevents the ISCC from improperly driving the system address bus since A0/SCC/DMA and A1/A/B become active outputs when the ISCC is the bus master.

The address map for the ISCC appears in Table A-6 for this application.

Table A-6. ISCC Address Map

A0	A1-A5	A6	A7	Registers Addressed
1	x	x	x	ISCC not enabled
0	-	0	x	DMA Registers per A1 - A5
0	-	1	1	SCC Core Channel A Registers
0	-	1	0	SCC Core Channel B Registers

Since A0 specifies the lower byte of the bus and includes the chip enable decode, the internal ISCC register addresses decode without A0. Thus, Table 6 implies that the Left Shift address decode selection is made for both the SCC and DMA sections of the ISCC. The left shift selection is the default selection after reset. Left/Right Shift selection programming is discussed later.

The ALE signal of the 8086 applies to AS of the ISCC through an inverting tri-state buffer. The buffer disables when the ISCC becomes a bus master during DMA transactions. This prevents conflicts since ALE remains active even when the 8086 is in the HOLD mode during DMA transfers. Now, the ISCC AS is an active output. The address strobe for the demultiplexing latch of addresses A0 through A15 connects on the ISCC side of the ALE tri-state buffer. This allows the latch to serve two functions; to hold either the 8086 or the ISCC address when it is bus master.

After reset, ALE is active and the tri-state buffer enabled. This supplies address strobes to the ISCC. The presence of one of these address strobes, before writing to the BCR, programs the ISCC to the multiplexed bus mode of operation. The ISCC chip enable (CE) can be inactive and still recognize an address strobe (AS) before the BCR write (Figure 4 shows open latches when the input strobe is low).

When the ISCC is bus master during DMA transactions, BHE generates from A0. This is done from the output of the lower order address latch through an inverting tri-state driver. This driver enables only when the ISCC is the bus master. Whole word transfers are not done by the ISCC DMA, thus, BHE generated for the ISCC is always the inverse of A0.

The upper bus system address lines demultiplex from the 8086 and the ISCC in separate latches. Like the 68000 example, high order address lines from the ISCC latch via UAS (upper address strobe). The separate latches drive the same upper order address lines. A16 from the ISCC connects to the corresponding A16 address bus line as derived from the 8086. The output of the two latches alternately enable depending upon bus mastership.

The diagram shows INT fron the ISCC connected to the 8086 INTR input via an inverter since these signals are of opposite sense. In actual practice, the ISCC interrupt request is first processed by an interrupt priority circuit. INTA (Interrupt Acknowledge) of the 8086 connects directly to the INTACK input of the ISCC. Conforming to the 8086 style of interrupt acknowledge, the ISCC is programed to the Double Pulse Interrupt Acknowledge type. When this selection occurs, the ISCC responds to two interrupt acknowledge pulses. The first pulse is recognized but no action follows. The second pulse causes the ISCC to go active on the data bus and return the interrupt vector to the CPU. This action also takes place with the Single Pulse Interrupt Acknowledge type selection, except that the bus goes active with the first and only interrupt acknowledge pulse.

To start, the BCR write (first write to the ISCC after RESET) is done with A7 = 1 (A1/A/B ISCC input at logic high) This selects the wait option of the WAIT/RDY signal to conform to the 8086 bus style. The AS signal programming of the multiplexed bus was covered earlier. The BCR is written

with 86h to enable byte swapping, select the sense of the byte swapping with respect to A0 (appropriate to this bus style), and select the Double Pulse type of interrupt acknowledge.

When the ISCC begins DMA transfers, it communicates requests for the bus through BUSREQ and BUSACK. The 8086 receives and grants bus requests through HOLD and HLDA in the minimum mode and through RQ/GT in the maximum mode. Depending upon the system requirements, there could be more than one potential bus master. Therefore, there is a requirement for a bus arbitration circuit.

The minimum mode connection is relatively straightforward. The maximum mode configuration requires a translation of the ISCC BUSREQ and BUSACK signals into/from the 8086 RQ/GT timed pulse style of handshake. Refer to the information on the 8086 for detailed application information.

The ISCC WAIT/RDY output is compatible with the 8086 clock generator RDY input except that one edge of the signal must be synchronous with the 8086 clock. The synchronization occurs through external circuitry. Refer to the information on the 8086 for detailed application information.

ZiLOG

APPLICATION NOTE

THE Z180 INTERFACED WITH THE SCC AT 10 MHz

INTRODUCTION

Build a simple system to prove and test the Z180 MPU interfacing the SCC at 10 MHz.

This Application Note describes the design of a system using a Z80180 MPU (Microprocessor Unit) and a Z85C30 SCC (Serial Communications Controller), both running at 10 MHz. Hereinafter, all references are to the Z180 and SCC.

The system board is a vehicle for demonstration and evaluation of the 10 MHz interface and includes the following parts:

- Z8018010VSC Z180 MPU 10 MHz, PLCC package
- Z85C3010VSC C-MOS Z8530 SCC Serial Communication Controller, 10 MHz, PLCC package
- 27C256 EP-ROM
- 55257 Static RAM

The Z180 is a Z80 compatible High Integration device with various peripherals on board. Using this device as an alternative to the Z80 CPU, reduces the number of parts and board space. However, processing speed and reliability increase.

The serial communication devices on the Z180 are: two asynchronous channels and one clocked serial channel. This means handling synchronous serial communications protocols requires an off-chip "multi-protocol serial communication controller." The SCC is the ideal device to meet the requirements.

Zilog's SCC is the multi-protocol (@ 10 MHz) universal serial communication controller which covers most serial communication applications including Monosync, Bisync and SDLC at 2.5M bits/sec speeds. Further, the wide acceptance of this device by the market ensures it is an "industrial standard" serial communication controller. Also, the Z180 has special numbers for system clock frequencies of 6.144 - and 9.216 MHz which generate exact baud rates for on-chip asynchronous serial communication channels. This is due to the SCC's on-chip, 16-bit wide baud rate generator for asynchronous ASCI communications.

The following 10 MHz interface explanation defines how the interrupt structure works. Also included is a discussion of the hardware and software considerations involved in running the system's communication board. This Application Note assumes the reader has a strong working knowledge of the Z180 and SCC; this is not a tutorial for each device.

Note: 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); /N//S (NORMAL and SYSTEM are both active Low).

INTERFACES

The following subsections explain the interfaces between the:

- Z180 and Memory
- Z180 and I/O
- Z180 and SCC

Basic goals of this system design are:

- System clock up to 10 MHz
- Using the Z8018010VSC (Z180 10 MHz PLCC package) to take advantage of 1M byte addressing space and compactness (DIP versions' addressing range is half; 512K bytes)
- Using Z85C3010VSC (CMOS SCC 10MHz PLCC package)
- Minimum parts count
- Worst case design
- Using EPLD for glue wherever possible
- Expandability

The design method for EPLD is using TTLs (74HCT) and then translating them into EPLD logic. This design uses TTLs and EPLDs. With these goals in mind, the discussion begins with the Z180-to-memory interface.

Z180 to Memory Interface

The memory access cycle timing of the Z180 is similar to the Z80 CPU memory access cycle timing. The three classifications are:

- Op-code fetch cycle (Figure 1)
- Memory read cycle (Figure 2)
- Memory write cycle (Figure 3)

Table 1 shows the Z180's basic timing elements for the op-code's fetch/memory read/write cycle.

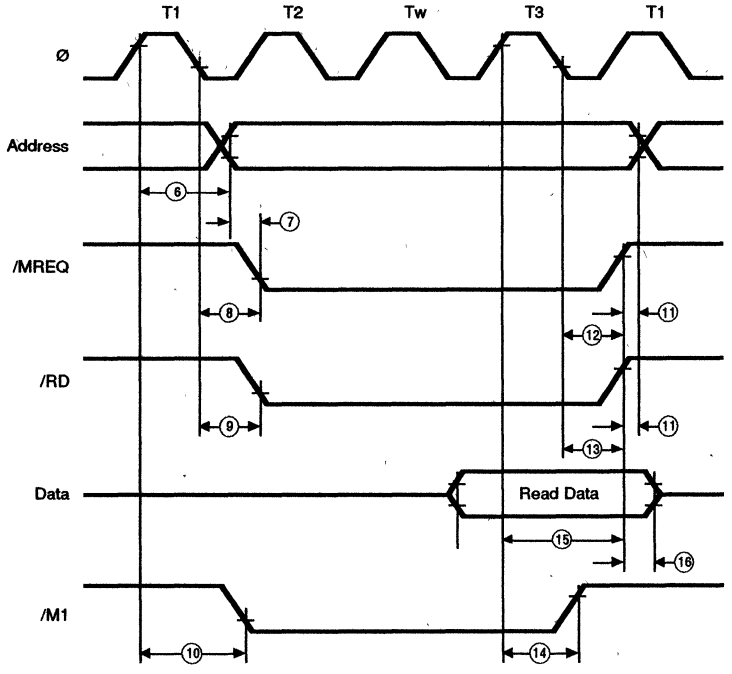


Figure 1. Z180 Op-code Fetch Cycle Timing (One Wait State)

Table 1. Z8018010 Timing Parameters for Op-code Fetch Cycle (Worst Case: Z180 10 MHz)

No	Symbol	Parameter	Min	Max	Units
1	tcyc	Clock Cycle Period	100		ns
2	tCHW	Clock Cycle High Width	40		ns
3	tCLW	Clock Cycle Low Width	40		ns
4	tcf	Clock Fall Time		10	ns
6	tAD	Clock High to Address Valid		70	ns
8	tMED1	Clock Low to /MREQ Low		50	ns
9	tRDD1	Clock Low to /RD Low		50	ns
11	tAH	Address Hold Time	10		ns
12	tMED2	Clock Low to /MREQ High		50	ns
15	tDRS	Data to Clock Setup	25		ns
16	tDRH	Data Read Hold Time	0		ns
22	tWRD1	Clock High to /WR Low		50	ns
23	tWDD	Clock Low to Write Data Delay		60	ns
24	tWDS	Write Data Setup to /WR Low	15		ns
25	tWRD2	Clock Low to /WR High		50	ns
26	tWRP	/WR Pulse Width		110	ns
27	tWDH	/WR High to Data Hold Time	10		ns

Note:
Parameter numbers in this table are in the Z180 technical manual.

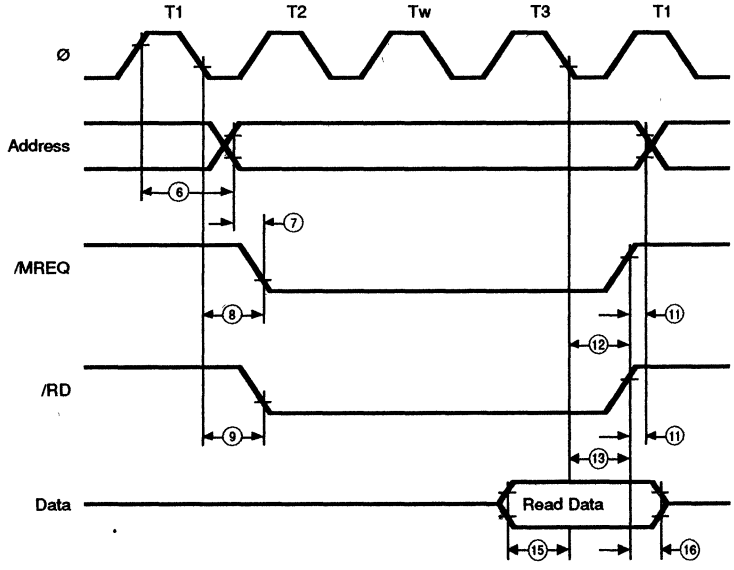


Figure 2. Z180 Memory Read Cycle Timing (One Wait State)

EP-ROM Interface

During an Op-code fetch cycle, data sampling of the bus is on the rising PHI clock edge of T3 and on the falling edge of T3 during a memory read cycle. Op-code fetch cycle data sample timing is half a clock cycle earlier. Table 2 shows how a memory read cycles' timing requirements are easier than an op-code fetch cycle by half a PHI cycle time. If the timing requirements for an Op-code fetch cycle meet specifications, the design satisfies the timing requirements for a memory read cycle.

Table 2 has some equations for an op-code fetch, memory read/write cycle.

Table 2. Parameter Equations (10 MHz) Op-code Fetch/Memory Read/Write Cycle

Parameters	Z180 Equation	Value	Units
Address Valid to Data Valid (Op-code Fetch)	2(1+w)tcyc-tAD-tDRS	105+100w min	ns
Address Valid to Data Valid (Memory Read	2(1+w)tcyc+tCHW+tcf-tAD-tDRS	155+100w min	ns
/MREQ Active to Data Valid (Op-code Fetch)	(1+w)tcyc+tCLW-tMED1-tDRS	55+100w min	ns
/MREQ Active to Data Valid (Memory Read)	(2+w)tcyc-tMED1-tDRS	105+100w min	ns
/RD Active to Data Valid (Op-code Fetch)	(1+w)tcyc+tCLW-tRRD1-tDRS	55+100w min	ns
/RD Active to Data Valid (Memory Read)	(2+w)tcyc-tRRD1-tDRS	105+100w min	ns
Memory Write Cycle /WR Pulse Width	tWRP+w*tcyc	110+100w min	ns

Note:
* w is the number of wait states.

The propagation delay for the decoded address and gates in the previous calculation is zero. Hence, on the real design, subtracting another 20-30 ns to pay for propagation delays, is possible. The 27C256 provides the EP-ROM for this board. Typical timing parameters for the 27C256 are in Table 3.

Table 3. EP-ROM (27C256) Key Timing Parameters (Values May Vary Depending On Mfg.)

	Access Time		
Parameter	170ns Max	200ns Max	250ns Max
Addr Access Time	170	200	250
/E to Data Valid	170	200	250
/OE to Data Valid	75	75	100

Note:
Table 3 shows "Access Time" as applying /E to data valid. "/OE active to data valid" is shorter than "address access time". Hence, the interface logic for the EP-ROM is: Realize a 170ns or faster EPROM access time by adding one wait state (using the on-chip wait state generator of the Z180). A 200ns requirement uses two wait states for memory access.

SRAM Interface

Table 4 has timing parameters for 256K bit SRAM for this design.

Table 4. 256K SRAM Key Timing parameters (Values May Vary Depending On Mfg.)

	Access Time		
Parameter	85nS Min	100nS Min	150nS Min
Read Cycle:			
/E to Data Valid	85	100	150
/G to Data Valid	45	40	60
Write Cycle:			
Write Cycle Time	85	100	150
Addr Valid to End of Write	75	80	100
Chip Select to End of Write	75	80	100
Data Select to End of Write	40	40	60
Write Pulse Width	60	60	90
Addr Setup Time	0	0	0

SRAM Read Cycle. An SRAM read cycle shares the same considerations as an EPROM interface.

Like EPROM, SRAMs' "access time" applies /G to data valid, and "/E active to data valid" is shorter than "access time." This design allows the use of a 150ns access time SRAM by adding one wait state (using the on-chip wait state generator of the Z180). The circuit is common to the EPROM memory read cycle.

No wait states are necessary if there is a 85ns, or faster, access time by using SRAMs. Since the Z180 has on-chip MMU with 85ns or faster SRAM just copy the contents of EPROM (application program starts at logical address 0000h) into SRAM after power on. Set up the MMU to SRAM area to override the EPROM area and stop inserting wait states. With this scheme, you can get the highest performance with moderate cost.

SRAM Write Cycle. During a Z180 memory write cycle, the Z180 write data is stable before the falling edge of /WR (Z180 parameter #24; 15ns min at 10 MHz). It is stable throughout the write cycle (Z180 parameter #27; 10nS min at 10 MHz). Further, the address is fixed before the falling edge of /WR. As long as the /WR pulse width meets the SRAM's spec, there is no problem (reference Table 2).

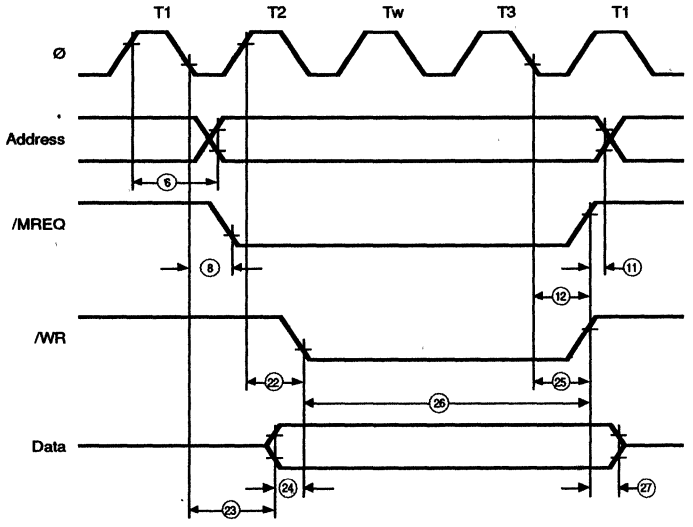


Figure 3. Z180 Memory Write Cycle Timing (One Wait State)

Memory Interface Logic

The memory devices (EPROM and SRAM) for this design are 256K bit (32K byte). There are two possible memory interface designs:

Connect Address Decode output to /E input. Put the signal generated by /RD and /MREQ ANDed together to /OE of EPROM and SRAM. Put the signal generated by /WR and /MREQ ANDed together to the /WE pin of SRAM (Figure 4a).

Connect the signal Address ANDed together with inactive /IORQ to the /E input. Connect /RD to /OE of EPROM and SRAM, and /WR to /WE pin of SRAM (Figure 4b).

Using the second method, there could be a narrow glitch on the signal to the /E-pin during I/O cycles and the Interrupt acknowledge cycle. During I/O cycles, /IORQ and /RD or /WR go active at almost the same time. Since the delay times of these signals are similar there is no "overlapping time" between /CE generated by the address (/IORQ inactive), and /WR or /RD active. During the Interrupt Acknowledge cycle, /WR and /RD signals are inactive.

To keep the design simple and flexible, use the second method (Figure 4b). To expand memory, decode the address A15 NANDed with /USRRAM//USRROM and /IORQ to produce /CSRAM or /CSROM. These are chip select inputs to chips 55257 or 27C256, respectively. This either disables or enables on-board ROM or RAM depending upon selection control.

The circuit on Figure 4b gives the physical memory address as shown on Figure 5.

If there are no Z80 peripherals and /M1 is enabled (M1E bit in Z180 OMCR register set to 1), active wait states occur only during op-code fetch cycles (Figure 6). If the M1E bit is cleared to 0, /M1E is active only during the Interrupt

Acknowledge cycle and Return from Interrupt cycle. This case depends on the propagation delay of the address decoder which uses 135ns or faster EPROM assess time (assume there is 20ns propagation delay). Figure 6 shows the example of this implementation.

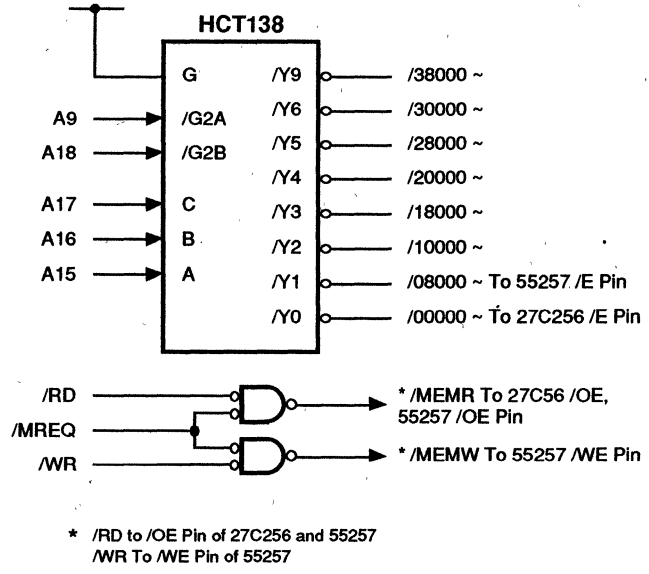


Figure 4A. Memory Interface Logic

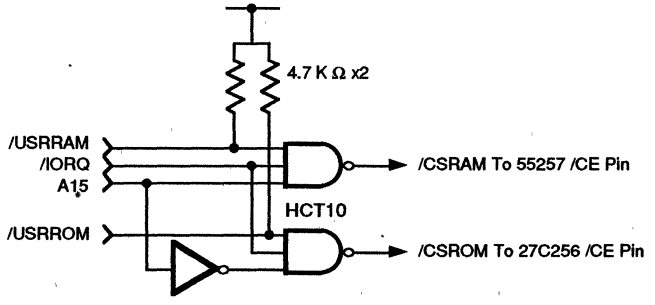


Figure 4B. Memory Interface Logic

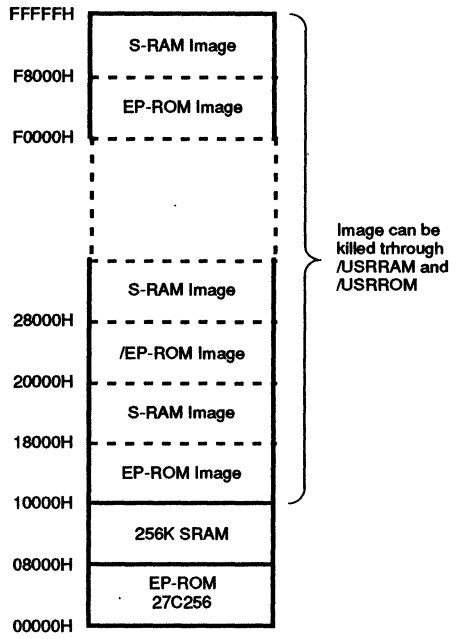


Figure 5. Physical Memory Address Map

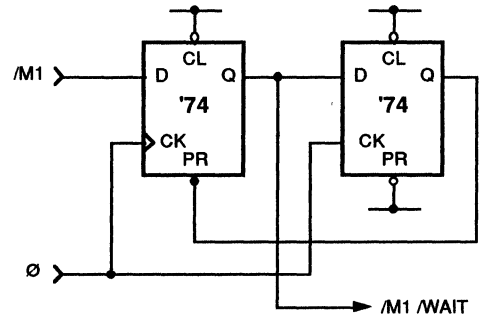


Figure 6. Wait State Generator Logic

(Extends Op-code Fetch Cycle Only; Not Working in Z Mode of Operation)

Z180 TO I/O INTERFACE

The Z180 I/O read/write cycle is similar to the Z80 CPU if you clear the /IOC bit in the OMCR register to 0 (Figures 7 and 8). Table 5 shows the Z180 key parameters for an I/O cycle.

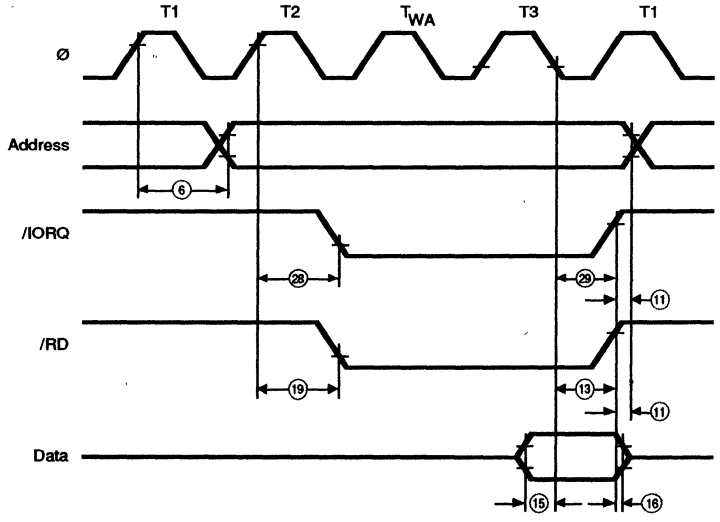


Figure 7. Z180 I/O Read Cycle Timing (/IOC = 0)

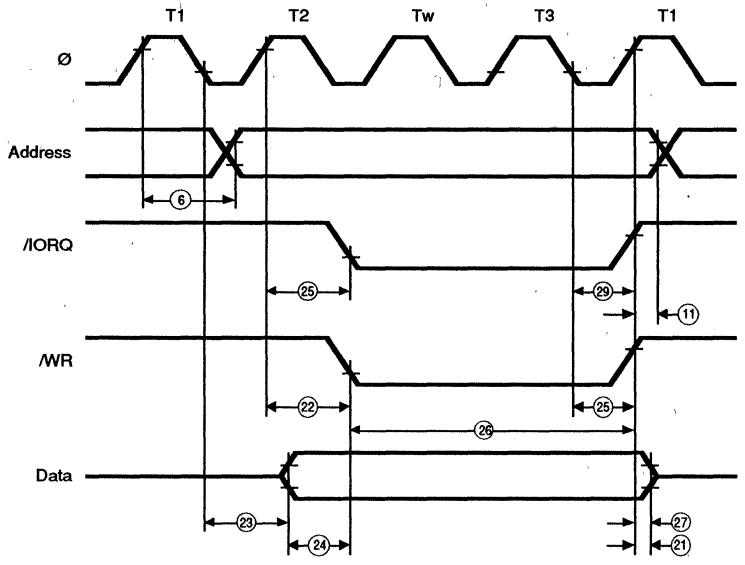


Figure 8. Z180 I/O Write Cycle Timing

Table 5. Z8018010 Timing Parameters for I/O Cycle (Worst Case)

No	Symbol	Parameter	Min	Max	Units
1	tcyc	Clock Cycle Period	100		ns
2	tCHW	Clock Cycle High Width	40		ns
3	tCLW	Clock Cycle Low Width	40		ns
4	tcf	Clock Fall Time		10	ns
6	tAD	Clock High to Address Valid		70	ns
9	tRDD1	Clock High to /RD Low IOC=0		55	ns
11	tAH	Address Hold Time	10		ns
13	tRDD2	Clock Low to /RD High		50	ns
15	tDRS	Data to Clock Setup	25		ns
16	tDRH	Data Read Hold Time	0		ns
21	tWDZ	Clock High to Data Float Delay		60	ns
22	tWRD1	Clock High to /WR Low		50	ns
23	tWDD	Clock Low to Write Data Delay		60	ns
24	tWDS	Write Data Setup to /WR Low	15		ns
25	tWRD2	Clock Low to /WR High		50	ns
26a	tWRP	/WR Pulse Width (I/O Write)	210		ns
27	tWDH	/WR High to Data Hold Time	10		ns
28	tIOD1	Clock High to /IORQ Low IOC=0		55	ns
29	tIOD2	Clock Low to /IORQ High		50	ns

Note:
Parameter numbers in this table are the numbers in the Z180 technical manual.

If you are familiar with the Z80 CPU design, the same interfacing logic applies to the Z180 and I/O interface (see Figure 9a). This circuit generates /IORD (Read) or IORD (Write) for peripherals from inputs /IORQ, /RD, and /WR. The address decodes the Chip Select signal. Note, if you have Z80 peripherals, the decoder logic decodes only from addresses (does not have /IORQ). The Z180 signals /IORQ, /RD, and /WR are active at about the same time (Param #9, 22, 28). However, most of the Z80 peripherals require /CE to /RD or /WR setup time.

Since the Z180 occupies 64 bytes of I/O addressing space for system control and on-chip peripherals, there are no overlapping I/O addresses for off-chip peripherals. In this design, leave the area as default or assign on-chip registers at I/O address 0000h to 003Fh.

Figure 9 shows a simple address decoder (the required interface signals, other than address decode outputs, are discussed later).

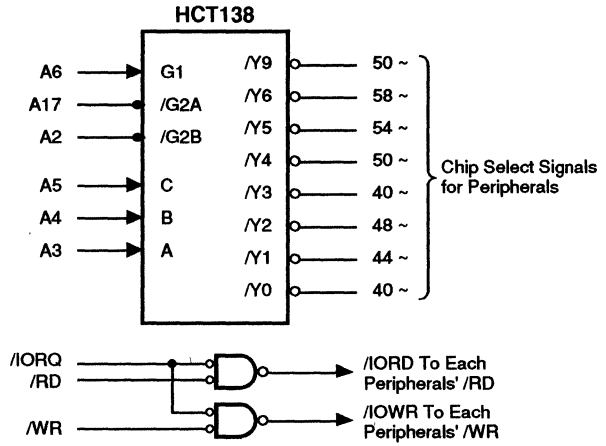


Figure 9A. I/O Interface Logic (Example)

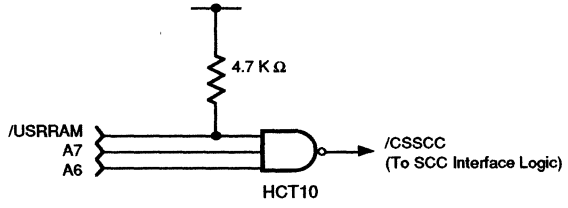


Figure 9B. I/O Address Decoder for this Board

When expanding this board to enable other peripherals, the decoded address A6/A7 is NANDed with USRIO to produce the Chip Enable (CSSCC) output signal (HC10). The SCC registers are assigned from address xxC0h to xxC3h; with image, they occupy xxC0h to xxFFh. To add wait states during I/O transactions, use the Z180 on-chip wait state generator instead of external hardware logic.

If there is a Z80 PIO on board in a Z-mode of operation (that is, clear /M1E in OMCR register to zero) and after enabling a Z80 PIO interrupt, zero is written to M1TE in the OMCR register. Without a zero, there is no interrupt from the Z80 PIO. The Z80 PIO requires /M1 to activate an interrupt circuit after enabling interrupt by software.

Z180 TO SCC INTERFACE

The following subsections discuss the various parameters between the Z180/SCC interface: CPU hardware, I/O operation (read/write), SCC interrupts, Z80 interrupt daisy-chain operation, SCC interrupt daisy-chain operation, I/O cycles.

CPU Hardware Interfacing

The hardware interface has three basic groups of signals: Data bus, system control, and interrupt control. For more detailed signal information, refer to Zilog's Technical Manuals, and Product Specifications for each device.

Data Bus Signals
D7-D0. *Data bus* (Bidirectional, 3-state). This bus transfers data between the Z180 and SCC.

System Control Signals
A//B, C//D. *Register select signals* (Input). These lines select the registers.

/CE. *Chip enable* (Input, active low). /CE selects the proper peripheral for programming. /CE is gated with /IORQ or /MREQ to prevent false chip selects during other machine cycles.

/RD+. *Read* (input, active low). /RD activates the chip-read circuitry and gates data from the chip onto the data bus.

/WR+. *Write* (Input, active low). /WR strobes data from the data bus into the peripheral.

Chip reset occurs when /RD and /WR are active simultaneously.

Interrupt Control
/INTACK. *Interrupt Acknowledge* (input, active low). This signal shows an Interrupt Acknowledge cycle which combines with /RD to gate the interrupt vector onto the data bus.

/INT. *Interrupt request* (output, open-drain, active low).

IEI. *Interrupt Enable In* (input, active high).

IEO. *Interrupt Enable* Out (Output, active high).

These lines control the interrupt daisy chain for the peripheral interrupt response.

SCC I/O Operation

The SCC generates internal control signals from /RD or /WR. Since PCLK has no required phase relationship to /RD or /WR, the circuitry generating these signals provides time for meta stable conditions to disappear.

The SCC starts the different operating modes by programming the internal registers. Accessing these internal registers occurs during I/O Read and Write cycles, described below.

Read Cycle Timing
Figure 10 illustrates the SCC Read cycle timing. All register addresses and /INTACK are stable throughout the cycle. The timing specification of SCC requires that the /CE signal (and address) be stable when /RD is active.

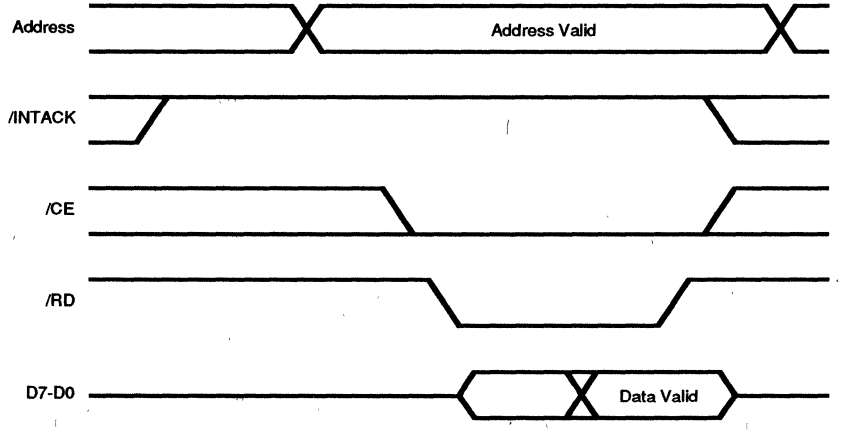


Figure 10. SCC Read Cycle Timing

Write Cycle Timing

Figure 11 illustrates the SCC Write cycle timing. All register addresses and /INTACK are stable throughout the cycle. The timing specification of the SCC requires that the /CE signal (and address) be stable when /RD is active. Data is available to the SCC before the falling edge of /WR and remains active until /WR goes inactive.

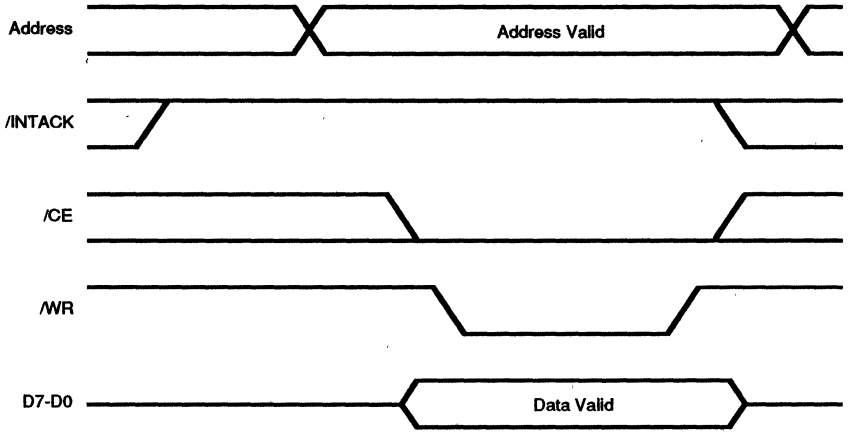


Figure 11. SCC Write Cycle Timing

SCC Interrupt Operation

Understanding SCC interrupt operations requires a basic knowledge of the Interrupt Pending (IP) and Interrupt Under Service (IUS) bits in relation to the daisy chain. The Z180 and SCC design allow no additional interrupt requests during an Interrupt Acknowledge cycle. This permits the interrupt daisy chain to settle, ensuring proper response of the interrupt device.

The IP bit sets in the SCC for CPU intervention requirements (that is, buffer empty, character available, error detection, or status changes). The interrupt acknowledge cycle does not reset the IP bit. The IP bit clears by a software command to the SCC, or when the action that generated the interrupt ends, for example, reading a receive character for receive interrupt. Others are, writing data to the transmitter data register, issuing Reset Tx interrupt pending command for Tx buffer empty interrupt, etc.). After servicing the interrupt, other interrupts can occur.

The IUS bit means the CPU is servicing an interrupt. The IUS bit sets during an Interrupt Acknowledge cycle if the IP bit sets and the IEI line is High. If the IEI line is low, the IUS bit is not set. This keeps the device from placing its vector onto the data bus.

The IUS bit clears in the Z80 peripherals by decoding the RETI instruction. A software command also clears the IUS bit in the Z80 peripherals. Only software commands clear the IUS bit in the SCC.

Z80 Interrupt Daisy-Chain Operation

In the Z80 peripherals, both IP and IUS bits control the IEO line and the lower portion of the daisy chain. When a peripheral's IP bit sets, the IEO line goes low This is true regardless of the state of the IEI line. Additionally, if the peripheral's IUS bit clears and its IEI line is High, the /INT line goes low.

The Z80 peripherals sample for both /M1 and /IORQ active (and /RD inactive) to identify an Interrupt Acknowledge cycle. When /M1 goes active and /RD is inactive, the peripheral detects an Interrupt Acknowledge cycle and allows its interrupt daisy chain to settle. When the /IORQ line goes active with /M1 active, the highest priority interrupting peripheral places its interrupt vector onto the data bus. The IUS bit also sets to show that the peripheral is now under service. As long as the IUS bit sets, the IEO line remains low. This inhibits any lower priority devices from requesting an interrupt.

When the Z180 CPU executes the RETI instruction, the peripherals check the data bus and the highest priority device under service resets its IUS bit.

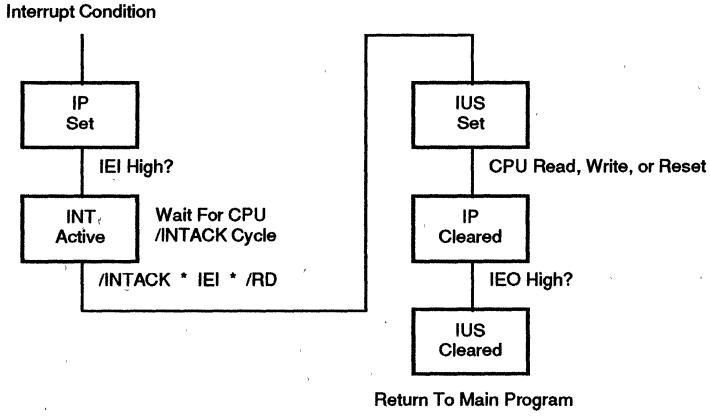
SCC Interrupt Daisy-Chain Operation

In the SCC, the IUS bit normally controls the state of the IEO line. The IP bit affects the daisy chain only during an Interrupt Acknowledge cycle. Since the IP bit is normally not part of the SCC interrupt daisy chain, there is no need to decode the RETI instruction To allow for control over the daisy chain, the SCC has a Disable Lower Chain (DLC) software command that pulls IEO low. This selectively deactivates parts of the daisy chain regardless of the interrupt status. Table 6 shows the truth table for the SCC interrupt daisy chain control signals during certain cycles. Table 7 shows the interrupt state diagram for the SCC.

Table 6. SCC Daisy Chain Signal Truth Table

During Idle State			During INTACK Cycle				
IEI	IP	IUS	IEO	IEI	IP	IUS	IEO
0	X	X	0	0	X	X	0
1	X	0	1	1	1	X	0
1	X	1	0	1	X	1	0
1	0	0	1				

Table 7. SCC Interrupt Status Diagram



The SCC uses /INTACK (Interrupt Acknowledge) for recognition of an interrupt acknowledge cycle. This pin, used with /RD, allows the SCC to gate its interrupt vector onto the data bus. An active /RD signal during an interrupt acknowledge cycle performs two functions. First, it allows the highest priority device requesting an interrupt to place its vector on the data bus. Secondly, it sets the IUS bit in the highest priority device to show the device is now under service.

INPUT/OUTPUT CYCLES

Although the SCC is a universal design, certain timing parameters differ from the Z180 timing. The following subsections discuss the I/O interface for the Z180 MPU and SCC.

Z180 MPU to SCC Interface

Table 8 shows key parameters of the 10 MHz SCC for I/O read/write cycles.

Table 8. 10 MHz SCC Timing Parameters for I/O Read/Write Cycle (Worst Case)

No	Symbol	Parameter	Min	Max	Units
6	TsA(WR)	Address to /WR Low Setup	50		ns
7	ThA(WR)	Address to /WR High Hold	0		ns
8	TsA(RD)	Address to /RD Low Setup	50		ns
9	ThA(RD)	Address to /RD High Hold	0		ns
16	TsCEI(WR)	/CE Low to /WR Low Setup	0		ns
17	ThCE(WR)	/CE to /WR High Hold	0		ns
19	TsCEI(RD)	/CE Low to /RD Low Setup	0		ns
20	ThCE(RD)	/CE to /RD High Hold	0		ns
22	TwRDI	/RD Low Width	125		ns
25	TdRDf(DR)	/RD Low to Read Data Valid		120	ns
27	TdA(DR)	Address to Read Data Valid		180	ns
28	TwWRI	/WR Low Width	125		ns
29	TsDW(WR)	Write Data to /WR Low Setup	10		ns
30	TdWR(W)	Write Data to /WR High Hold	0		ns

SCC I/O Read/Write Cycle

Assume that the Z180 MPU's /IOC bit in the OMCR (Operation Mode Control Register) clears to 0 (this condition is a Z80 compatible timing mode for /IORQ and /RD). The following are several design points to consider (also see Table 3).

I/O Read Cycle
Parameters 8 and 9 mean that Address is stable 50ns before the falling edge of /RD and until /RD goes inactive.

Parameters 19 and 20 mean that /CE is stable at the falling edge of /RD and until /RD goes inactive.

Parameter 22 means the /RD pulse width is wider than 125ns.

Parameters 25 and 27 mean that Read data is available on the data bus 120ns later than the falling edge of /RD and 180ns from a stable Address.

I/O Write Cycle
Parameters 6 and 7 mean that Address is stable 50ns before the falling edge of /WR and is stable until /WR goes inactive.

Parameters 16 and 17 mean that /CE is stable at the falling edge of /WR and is stable until /W goes inactive.

Parameter 28 means /WR pulse width is wider than 125ns.

Parameters 28 and 29 mean that Write data is on the data bus 10ns before the falling edge of /WR It is stable until the rising edge of /WR.

Tables 9 and 10 show the worst case SCC parameters calculating Z180 parameters at 10 MHz.

Table 9. Parameter Equations Worst Case (Without Delay Signals - No Wait State)

SCC Parameters	Z180 Equation	Value	Units
TsA(RD)	tcyc-tAD+tRDD1	30 min	ns
TdA(DR)	3tcyc+tCHW+tcf-tAD-tDRS	245 min	ns
TdRDf(DR)	2tcyc+tCHW+tcf-tRDD1-tDRS	160 min	ns
TwRDI	2tcyc+tCHW+tcf-tDRS+tRDD2	185 min	ns
TsA(WR)	tcyc-tAD+tWRD1	30 min	ns
TsDW(WR)	tWDS	15 min	ns
TwWRI	tWRP	210 min	ns

Table 10. Parameter Equations

Z180 Parameters	SCC Equation	Value	Units
tDRS	Address 3tcyc+tCHW-tAD-TdA(DR) RD 2tcyc+tCHW-tRDD1-TdRD(DR)	241 min 184 min	ns ns

I/O Read Cycle

These tables show that a delay of the falling edge of /RD satisfies the SCC TsA(RD) timing requirement of 50ns min. The Z180 calculated value is 30ns min for the worst case. Also, Z180 timing specification tAH (Address Hold time) is 10ns min. The SCC timing parameters ThA(RD) {Address to /RD High Hold} and ThCE(RD) {/CE to /RD High Hold} are minimum at 0ns. The rising edge of /RD is early to guarantee these parameters when considering address decoders and gate propagation delays.

I/O Write Cycle

Delay the falling edge of /WR to satisfy the SCC TsA(/WR) timing requirement of 50ns min. The Z180 calculates 30ns min worst case. Further, the Z180 timing specifications tAH (Address Hold time) and tWDH (/WR high to data hold time) are both 10ns min. The SCC timing parameters ThA(WR) {Address to /WR High Hold}, ThCE(WR) {/CE to /WR High Hold} and TdWR(W) {Write data to /WR High hold} are a minimum of 0 ns. The rising edge of /WR is early to guarantee these parameter requirements.

This circuit depicts logic for the I/O interface and the Interrupt Acknowledge Interface for 10 MHz clock of operation. Figure 12 is the I/O read/write timing chart (discussions of timing considerations on the Interrupt Acknowledge cycle and the circuit using EPLD occur later).

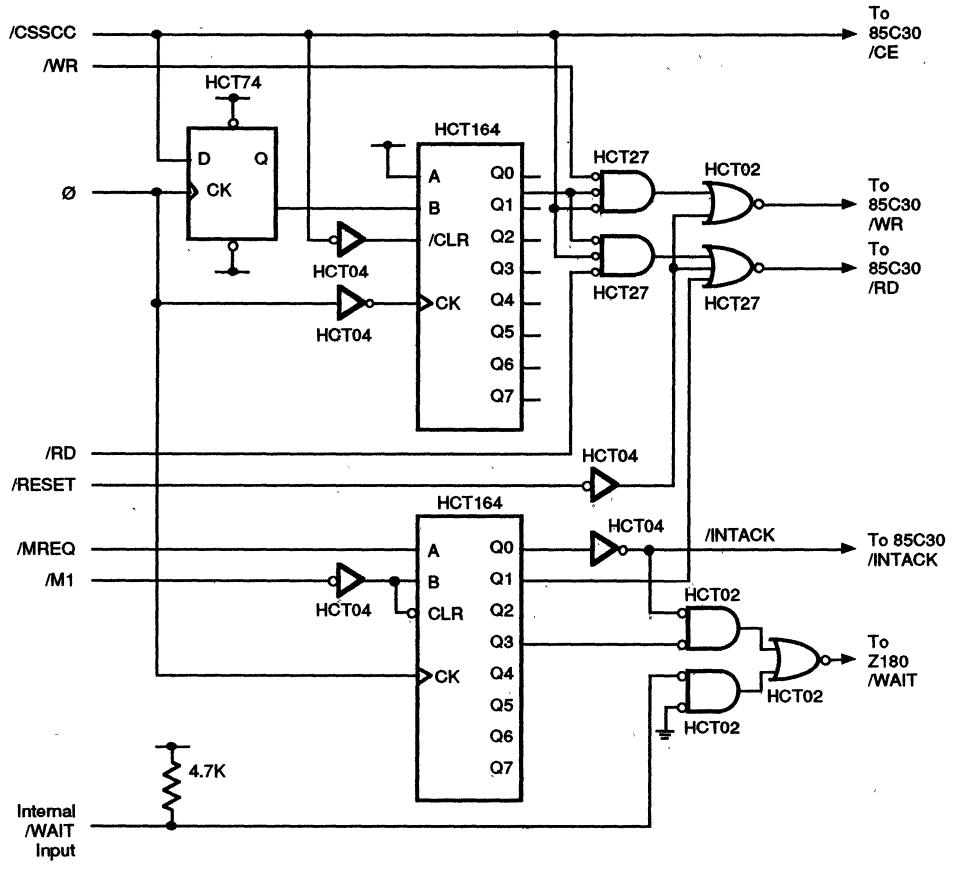


Figure 12. SCC I/O Read/Write Cycle Timing

If you are running your system slower than 8 MHz, remove the HCT74, D-Flip/Flop in front of HCT164. Connect the inverted CSSCC to the HCT164 B input. This is a required Flip/Flop because the Z180 timing specification on tIOD1 (Clock High to /IORQ Low, IOC=0) is maximum at 55ns. This is longer than half the PHI clock cycle. Sample it using the rising edge of clock, otherwise, HCT164 does not generate the same signals.

The RESET signal feeds the SCC /RD and /WR through HCT27 and HCT02 to supply the hardware reset signal. To reduce the gate count, drop these gates and make the SCC reset by its software command. The SCC software reset - 0C0h to Write Register 9, "Hardware Reset command" has the same effect as hardware reset by "Hardware."

Interrupt Acknowledge Cycle Timing

The primary timing differences between the Z180 and SCC occur in the Interrupt Acknowledge cycle. The SCC timing parameters that are significant during Interrupt Acknowledge cycles are in Table 11. The Z180 timing parameters are in Table 12. The reference numbers in Tables 11 and 12 refer to Figure 14.

Table 11. 10MHz SCC Timing Parameters for Interrupt Acknowledge Cycle

No	Symbol	Parameter	Min	Max	Units
13	TsIAi(RD)	/INTACK Low to /RD Low Setup	130		ns
14	ThIA(RD)	/INTACK High to /RD High Hold	0		ns
15	ThIA(PC)	/INTACK to PCLK High Hold	30		ns
38	TwRDA	/INTACK Low to /RD Low Delay (Acknowledge)	125		ns
39	TwRDA	/RD (Acknowledge) Width	125		ns
40	TdRDA(DR)	/RD Low (Acknowledge) to Read Data Valid Delay		120	ns
41	TsIEI(RDA)	IEI to /RD Low (Acknowledge) Setup Time	95		ns
42	ThIEI(RDA)	IEI to /RD High (Acknowledge) Hold Time	0		ns
43	TdIEI(IEO)	IEI to IEO Delay		175	ns

Table 12. Z180 Timing Parameters Interrupt Acknowledge Cycles (Worst Case Z180)

No	Symbol	Parameter	Min	Max	Units
10	tM1D1	Clock High to /M1 Low		60	ns
14	tM1D2	Clock High to /M1 High		60	ns
15	tDRS	Data to Clock Setup	25		ns
16	tDRH	Data Read Hold Time	0		ns
28	tIOD1	Clock LOW to /IORQ Low		50	ns
29	tIOD2	Clock LOW to /IORQ High		50	ns
30	tIOD3	/M1 Low to /IORQ Low Delay	200		ns

Note:
Parameter numbers in this table are the numbers in the Z180 technical manual.

During an Interrupt Acknowledge cycle, the SCC requires both /INTACK and /RD to be active at certain times. Since the Z180 does not issue either /INTACK or /RD, external logic generates these signals.

The Z180 is in a Wait condition until the vector is valid. If there are other peripherals added to the interrupt priority daisy chain, more Wait states may be necessary to give it time to settle. Allow enough time between /INTACK active and /RD active for the entire daisy chain to settle

There is no need of decoding the RETI instruction used by the Z80 peripherals since the SCC daisy chain does not use IP, except during Interrupt Acknowledge. The SCC and other Z8500 peripherals have commands that reset the individual IUS flag.

External Interface for Interrupt Acknowledge Cycle: The bottom half of Figure 13 is the interface logic for the Interrupt Acknowledge cycle.

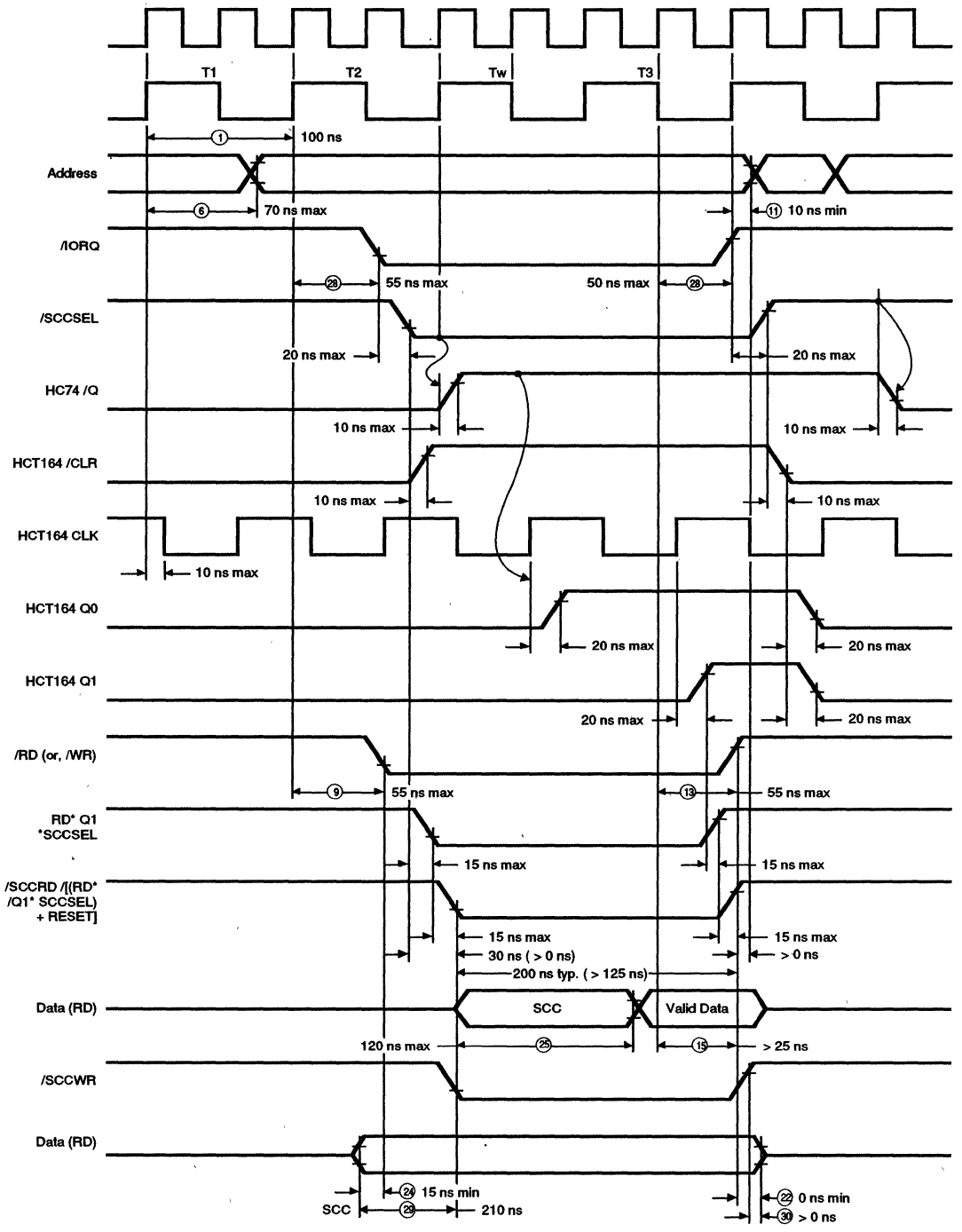


Figure 13. Z180 to SCC Interface Logic (Example)

The primary chip in this logic is the Shift register (HCT164), which generates /INTACK, /SCCRD and /WAIT. During I/O and normal memory access cycles, the Shift Register (HCT164) remains cleared because the /M1 signal is inactive during the op-code fetch cycle. Since the Shift Register output is Low, control of /SCCRD and /WAIT is by other system logic and gated through the NOR gate (HCT27). During I/O and normal memory access cycles, /SCCRD and /SCCWR are generated from the system /RD and /WR signals, respectively. The generation is by the logic at the top of Figure 14.

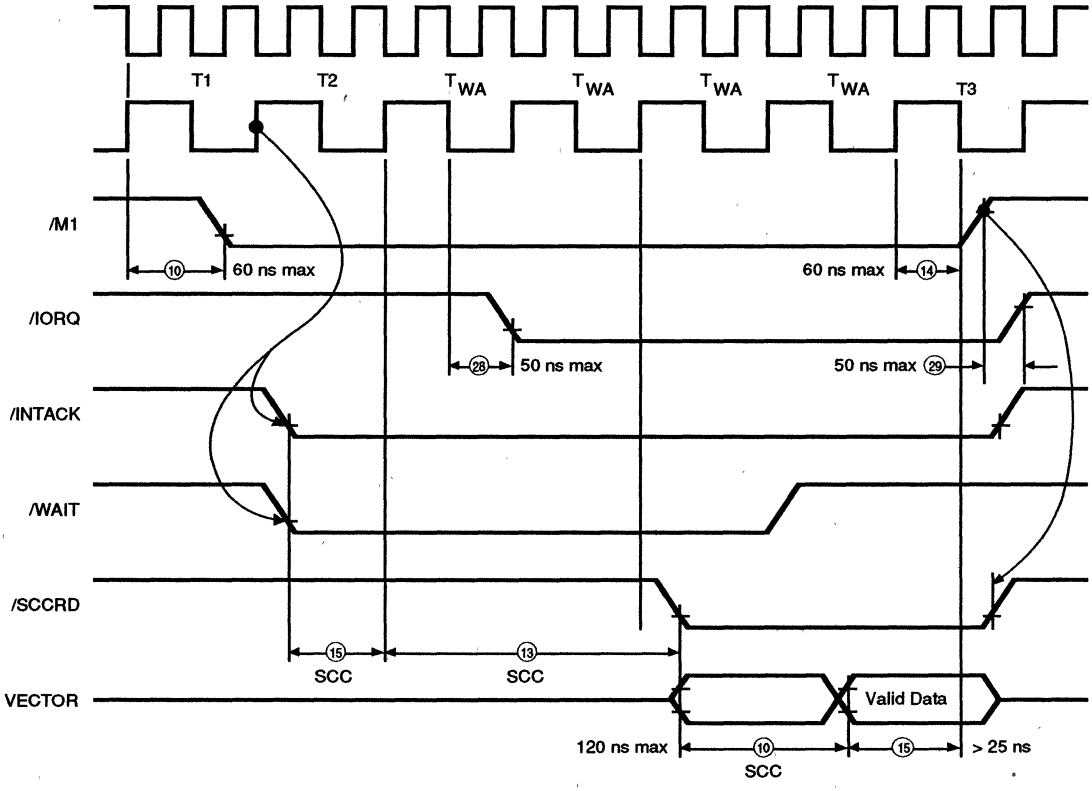


Figure 14. SCC Interrupt Acknowledge Cycle Timing

Normally, an Interrupt Acknowledge cycle appears from the Z180 during /M1 and /IORQ active (which is detected on the third rising edge of PHI after T1). To get an early sign of an Interrupt Acknowledge cycle, the Shift register decodes an active /M1. This is during the presence of an inactive /MREQ on the rising edge of T2.

During an Interrupt Acknowledge cycle, the /INTACK signal is generated on the rising edge of T2. Since it is the presence of /INTACK and an active SCCRD that gates the interrupt vector onto the data bus, the logic also generates /SCCRD at the proper time. The timing parameter of concern here is TdIAi(RD) [/INTACK to /RD (Acknowledge) Low delay]. This time delay allows the interrupt daisy chain to settle so the device requesting the interrupt places its interrupt vector onto the data bus.

The Shift Register allows enough time delay from the generation of /INTACK before it generates /SCCRD. During this delay, it places the Z180 into a Wait state until the valid interrupt vector is placed onto the data bus. If the time between these two signals is not enough for daisy chain settling, more time is added by taking /SCCRD and /WAIT from a later position on the Shift Register. If there is a requirement for more wait states, the time is calculated by PHI cycles.

USING EPLD

Figure 15a and Figure 15b show the logic using either EPLD or the circuit of this system. The EPLD is ALTERA 610 which is a 24-Pin EPLD. The method to convert random gate logic to EPLD is to disassemble MSIs' logic into SSI level, and then simplify the logic.

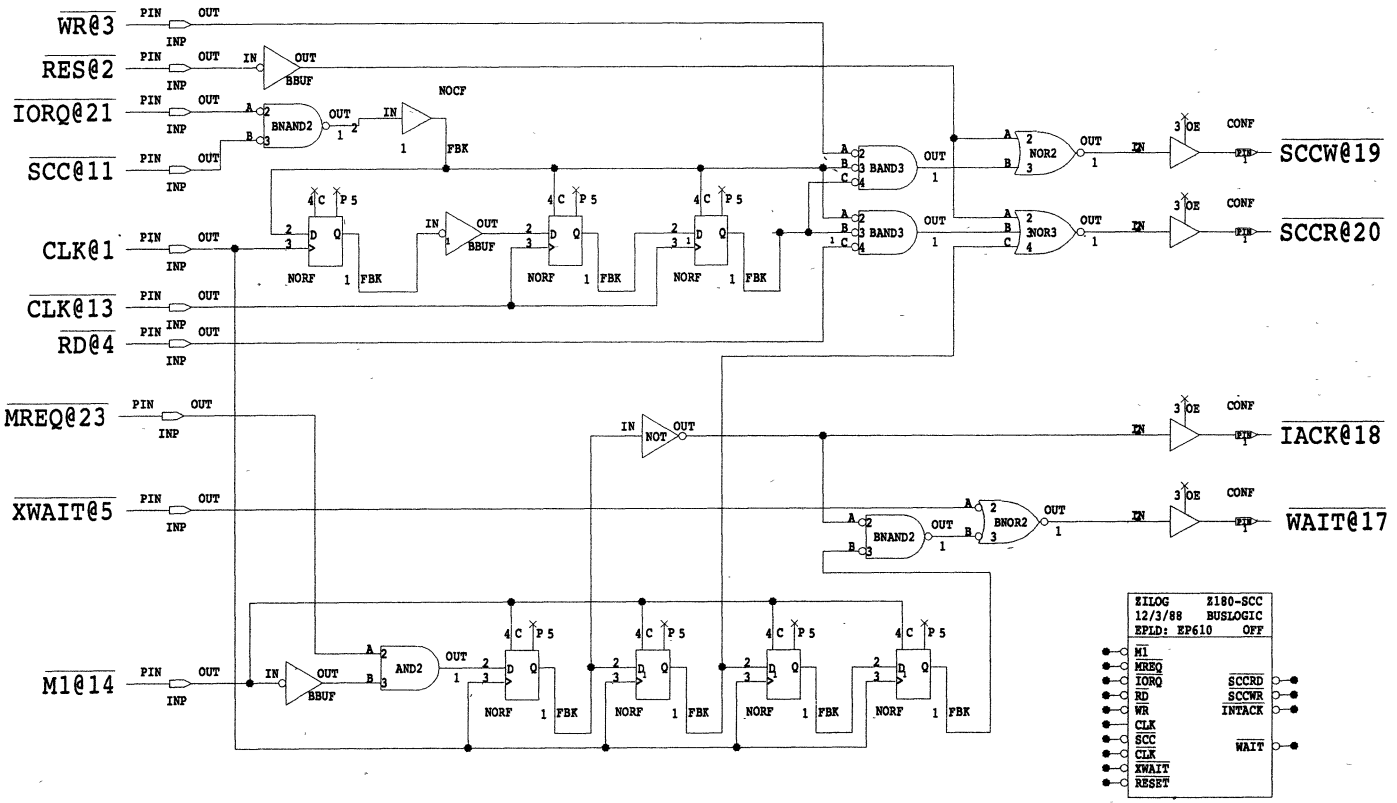


Figure 15a. EPLD Circuit Implementation

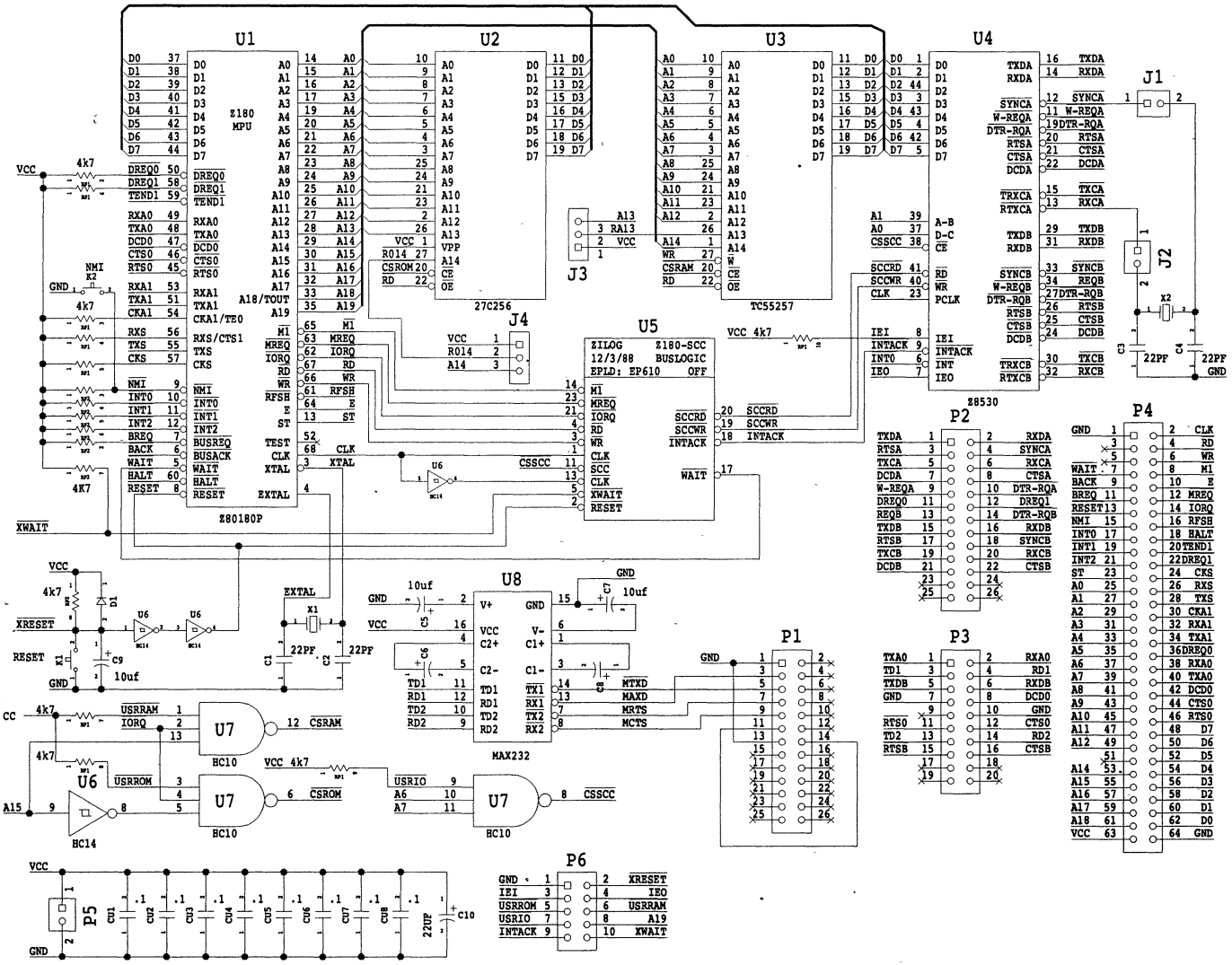


Figure 15b. EPLD Circuit Implementation

System Checkout

After completion of the board (PC board or wire wrapped board, etc.), the following methods verify that the board is working.

Software Considerations

Based on the previous discussion, it is necessary to program the Z180 internal registers, as follows, before system checkout:

- Z80 mode of operation - Clear /M1E bit in OMCR register to zero (to provide expansion for Z80 peripherals).
- Z80 compatible mode - Clear IOC bit in OMCR register to zero.
- Put one wait state in memory cycle, and no wait state for I/O cycle DMCR register bits 7 and 6 to "1" and bits 5 and 4 to "0".

SCC Read Cycle Proof

Read cycle checking is first because it is the simplest operation. The SCC Read cycle is checked by reading the bits in RR0. First, the SCC is hardware reset by simultaneously pulling /RD and /WR LOW (The circuit above includes the circuit for this). Then, reading out the Read Register 0 returns:

D7-D0 = 01xxx100b
Bit D2, D6:1
Bit D7, D1, D0:0
Bit D5: Reflects /CTS pin
Bit D4: Reflects /SYNC
Bit D3: Reflects /DCD pin

SCC Write Cycle Proof

Write cycle checking involves writing to a register and reading back the results to the registers which return the written value. The Time Constant registers (WR12 and WR13) and External/Status Interrupt Enable register (WR15) are on the SCC.

Interrupt Acknowledge Cycle

Checking an Interrupt Acknowledge (/INTACK) cycle consists of several steps. First, the SCC makes an Interrupt Request (/INT) to the Z180. When the processor is ready to service the interrupt, it shows an Interrupt Acknowledge (/INTACK) cycle. The SCC then puts an 8-bit vector on the bus and the Z180 uses that vector to get the correct service routine. The following test checks the simplest case.

First, load the Interrupt Vector Register (WR2) with a vector, disable the Vector Interrupt Status (VIS) and enable interrupts (IE=1, MIE=1 IEI=1). Disabling VIS guarantees only one vector on the bus. The address of the service routine corresponding to the 8-bit vector number loads the Z180 vector table, and the Z180 is under Interrupt Mode 2.

Because the user cannot set the SCC Interrupt Pending Bit (IP), setting an interrupt sequence is difficult. An interrupt is generated indirectly via the CTS pin by enabling the following explanation.

Enable interrupt by /CTS (WR15, 20h), External/Status Interrupt Enable (WR1, 01h), and Master Interrupt Enable (WR9, 08h). Any change on the /CTS pin begins the interrupt sequence. The interrupt is re-enabled by Reset External/Status Interrupt (WR0, 10h) and Reset Highest IUS (WR0, 38h).

A sample program of an SCC Interrupt Test is shown in Table 13. The following programs in Tables 13, 14, and 15 assume that the 180 is correctly initialized. Table 13 uses the Assembler for the Z80 CPU.

Table 13. SCC Test Program - Interrupt for 180/SCC Application Board (Under Mode2 Interrupt)

```
;*          B register returns status info:
;*          Bit D0 : current /cts stat
;*          D1set : /cts int received
;*

.z800

                                    ;Read in Z180 register names and
*include 180macro.lib               ;macro for Z180 new instructions

;SCC Registers

scc_ad:         equ     0C3h        ;addr of scc ch a - data
scc_ac:         equ     0C2h        ;addr of scc ch a - control
scc_bd:         equ     0C1h        ;addr of scc ch b - data
scc_bc:         equ     0C0h        ;addr of scc ch b - control

scc_a:          equ     000h        ;set 0ffh to test ch a
                                    ;clear 00h to test ch b.

        if      scc_a
scc_cont:       equ     scc_ac
        else
scc_cont:       equ     scc_bc
        endif

                org     09000h      ;top of user ram area

inttest:        ld      sp,top_of_sp    ;init sp
                ld      a,high sccvect and 0ffh ;init i reg
                ld      i,a
                im      2           ;set interrupt mode 2
                call    initscc     ;initialize scc
                ld      b,0         ;clear status
                ei                  ;enable interrupt

wait_loop:      bit     1,b         ;check int status
                jr      z,wait_loop ;if not, loop again

wait_here:      jr      $           ;interrupt has been received
                                    ;ou can set breakpoint here!

;subroutine to initialize scc registers
;initialization table format is
;register number, then followed by the data to be written
;and the register number is 0ffh, then return

initscc:        ld      hl,scctab   ;initializ  scc
init0:          ld      a,(hl)      ;get register number
                cp      0ffh        ;reached at the end of table?
                ret     z           ;yes, return.
                out     (scc_cont),a ;write it
```

```
                inc     hl              ;point to next data
                ld      a,(hl)          ;get the data to be written
                out     (scc_cont),a    ;write it
                inc     hl              ;point to next data
                jr      init0           ;then loop

;external/status interrupt service routine

ext_stat:   ld      a,10h
                out     (scc_cont),a    ;reset ext/stat int
                in      a,(scc_cont)    ;read stat
                and     00100000b       ;mask off bits other than /cts
                rra                     ;shift into D0 loc
                rra
                rra
                rra
                rra
                set     1,a             ;set interrupt flag
                ld      b,a             ;save it
                ld      a,38h
                out     (scc_cont),a    ;reset highest ius
                ei                      ;enable int
                ret                     ;return from int

;initialization data table for scc
;table format - register number, then value for the register
;and ends with 0ffh - since scc doesn't have
;register 0ffh...

scctab:         db      09h             ;select WR9
        if      scc_a
                db      10000000b       ;ch a reset
        else
                db      01000000b       ;ch b reset
        endif
                db      0eh             ;select WR15
                db      20h             ;only enable /cts int

                db      01h             ;select WR1
                db      00000001b       ;enable ext/stat int

                db      10h             ;reset ext/stat int
                db      10h             ;twice

                db      09h             ;select WR9
                db      08h             ;mie, vect not incl. stat

                db      0ffh            ;end of table

;interrupt vector table
                org     inttest + 100h
sccvect:        dw      ext_stat

                .block  100h            ;reserve area for stack
top_of_sp:
                end
```

Table 14 shows a "macro" to enable the Z180 to use the Z80 Assembler, as well as register definitions.

There is one good test to ensure proper function. Generate a data transfer between the Z180/SCC using the Z180 on-chip DMA. The SCC self loop-back test transfers data using the Z180 DMA at the highest transmission rate (Table 14).

Table 14. Program Example - Z180 CPU Macro Instructions

```
;*          File name - 180macro.lib
;*  Macro library for Z180 new instructions for asm800
;*
;
;Z180 System Control Registers

;ASCI Registers
cntla0:     equ     00h         ; ASCI Cont Reg A Ch0
cntla1:     equ     01h         ; ASCI Cont Reg A Ch1
cntlb0:     equ     02h         ; ASCI Cont Reg B Ch0
cntlb1:     equ     03h         ; ASCI Cont Reg B Ch1
stat0:      equ     04h         ; ASCI Stat Reg Ch0
stat1:      equ     05h         ; ASCI Stat Reg Ch1
tdr0:       equ     06h         ; ASCI Tx Data Reg Ch0
tdr1:       equ     07h         ; ASCI Tx Data Reg Ch1
rdr0:       equ     08h         ; ASCI Rx Data Reg Ch0
rdr1:       equ     09h         ; ASCI Rx Data Reg Ch1

;CSI/O Registers
cntr:       equ     0ah         ; CSI/O Cont Reg
trdr:       equ     0bh         ; CSI/O Tx/Rx Data Reg

;Timer Registers
tmdr0l:     equ     0ch         ; Timer Data Reg Ch0-low
tmdr0h:     equ     0dh         ; Timer Data Reg Ch0-high
rldr0l:     equ     0eh         ; Timer Reload Reg Ch0-low
rldr0h:     equ     0fh         ; Timer Reload Reg Ch0-high
tcr:        equ     10h         ; Timer Cont Reg
tmdr1l:     equ     14h         ; Timer Data reg Ch1-low
tmdr1h:     equ     15h         ; Timer Data Reg Ch1-high
rldr1l:     equ     16h         ; Timer Reload Reg Ch1-low
rldr1h:     equ     17h         ; Timer Reload Reg Ch1-high
frc:        equ     18h         ; Free Running Counter

;DMA Registers
sar0l:      equ     20h         ; DMA Source Addr Reg Ch0-low
sar0h:      equ     21h         ; DMA Source Addr Reg Ch0-high
sar0b:      equ     22h         ; DMA Source Addr Reg Ch0-b
dar0l:      equ     23h         ; DMA Dist Addr Reg Ch0-low
dar0h:      equ     24h         ; DMA Dist Addr Reg Ch0-high
dar0b:      equ     25h         ; DMA Dist Addr Reg Ch0-B
bcr0l:      equ     26h         ; DMA Byte Count Reg Ch0-low
bcr0h:      equ     27h         ; DMA Byte Count Reg Ch0-high
mar1l:      equ     28h         ; DMA Memory Addr Reg Ch1-low
mar1h:      equ     29h         ; DMA Memory Addr Reg Ch1-high
mar1b:      equ     2ah         ; DMA Memory Addr Reg Ch1-b
iar1l:      equ     2bh         ; DMA I/O Addr Reg Ch1-low
iar1h:      equ     2ch         ; DMA I/O Addr Reg Ch1-high
```

```
bcr1l:      equ     2eh         ; DMA Byte Count Reg Ch1-low
bcr1h:      equ     2fh         ; DMA Byte Count Reg Ch1-high
dstat:      equ     30h         ; DMA Stat Reg
dmode:      equ     31h         ; DMA Mode Reg
dcntl:      equ     32h         ; DMA/WAIT Control Reg

;System Control Registers
il:         equ     33h         ; INT Vector Low Reg
itc:        equ     34h         ; INT/TRAP Cont Reg
rcr:        equ     36h         ; Refresh Cont Reg
cbr         equ     38h         ; MMU Common Base Reg
bbr:        equ     39h         ; MMU Bank Base Reg
cbar:       equ     3ah         ; MMU Common/Bank Area Reg
omcr:       equ     3eh         ; Operation Mode Control Reg
icr:        equ     3fh         ; I/O Control Reg


?b          equ     0
?c          equ     1
?d          equ     2
?e          equ     3
?h          equ     4
?l          equ     5
?a          equ     7

??bc        equ     0
??de        equ     1
??hl        equ     2
??sp        equ     3

slp         macro
            db      11101101B
            db      01110110B
            endm

mlt         macro   ?r
            db      11101101B
            db      01001100B+(??&?r AND 3) SHL 4
            endm

in0         macro   ?r, ?p
            db      11101101B
            db      00000000B+(?&?r AND 7) SHL 3
            db      ?p
            endm

out0        macro   ?p, ?r
            db      11101101B
            db      00000001B+(?&?r AND 7) SHL 3
            db      ?p
            endm

otim        macro
            db      11101101B
            db      10000011B
            endm
```

```
otimr   macro
        db      11101101B
        db      10010011B
        endm

otdm    macro
        db      11101101B
        db      10001011B
        endm

otdmr   macro
        db      11101101B
        db      10011011B
        endm

tstio   macro   ?p
        db      11101101B
        db      01110100B
        db      ?p
        endm

tst     macro   ?r
        db      11101101B
        ifidn   <?r>,<(hl)>
                db      00110100B
        else
        ifdef   ?&?r
                db      00000100B+(?&?r AND 7) SHL 3
        else
                db      01100100B
                db      ?r
        endif
        endif
        endm
        .list
end
```

Table 15 lists a program example for the Z180/SCC DMA transfer test.

Table 15. Test Program - Z180/SCC DMA Transfer

```
;
;*      Test program for 180 DMA/SCC
;*
;*     Test 180's DMA function with SCC
;*
;*     180 dma - dma0 for scc rx data
;*               dma1 for scc tx data
;*  async, X1 mode, 1 stop, speed = pclk/4
;*               self loop-back
;*     Connect W/REQ to   DREQ0 of 180
;*              DTR/REQ to DREQ1 of 180
;*
;*      B register returns status info:
;*       Bit D0 set : Tx DMA end
;*            D1 set : Rx DMA end
;*            D2 set : Data doesn't match
;*


.z800

                                                    ;Read in Z180 register names and
*include 180macro.lib                               ;macro for Z180 new instructions

;SCC Registers

scc_ad:         equ     0C3h        ;addr of scc ch a - data
scc_ac:         equ     0C2h        ;addr of scc ch a - control
scc_bd:         equ     0C1h        ;addr of scc ch b - data
scc_bc:         equ     0C0h        ;addr of scc ch b - control

scc_a:          equ     00h         ;if test ch. a, set this to 0ffh
                                    ;for ch.b, set this to 00h
        if      scc_a
scc_cont:       equ     scc_ac
scc_data:       equ     scc_ad
        else
scc_cont:       equ     scc_bc
scc_data:       equ     scc_bd
        endif

length:         equ     1000h       ;transfer length

                org     09000h      ;top of user ram area

sccdma:         ld      sp,tx_buff  ;init sp
                ld      a,(high z180vect) and 0ffh  ;init i reg
                ld      i,a
                ld      a,00h       ;init il
                out0    (il),a
```

```
                im      2               ;Set interrupt mode 2
                call    fill_mem        ;initialize tx/rx buffer area
                call    initscc         ;initialize scc
                call    initdma
                ld      b,0             ;init status

                ld      a,00h           ;load 1st data to be sent
                out     (scc_data),a

                ld      a,11001100b     ;enable dmac and int from DMA0
                out0    (dstat),a

                ld      a,05h           ;select WR5
                out     (scc_cont),a
                ld      a,01101000b     ;start tx
                out     (scc_cont),a

                ei                      ;wait here for completion

loop:           bit     1,b             ;rx dma end?
                jr      z,loop          ;not, then loop again

                push    bc              ;save bc reg
                ld      bc,length       ;compare tx data with rx data
                ld      de,tx_buff
                ld      hl,rx_buff
chkloop:    ld  a,(de)
                cpi
                jr      nz,bad_data
                jp      v,good
                inc     de
                jr      chkloop
bad_data:       pop     bc              ;restore bc
                set     2,b             ;set error flag
                jr      enddma
good:           pop     bc              ;restore bc

enddma:         jr      $               ;tx/rx completed
                                        ;you can put breakpoint here


fill_mem:   l   d       hl,temp         ; prepare data to be sent
                ld      bc,length       ; set length
                ld      de,tx_buff
                ld      (hl),00h
fill_loop:      ldi
                jp      nv,fill_00
                dec     hl
                inc     (hl)
                jr      fill_loop

fill_00:        ld      bc,length       ; clear rx buffer area to zero
                ld      de,rx_buff
                ld      (hl),00h
fill_00l:       ldi
                ret     nv
```

```
                dec     hl
                jr      fill_00l

initscc:        ld      hl,scctab       ; initializa scc
init0:          ld      a,(hl)
                cp      0ffh
                ret     z
                out     (scc_cont),a
                inc     hl
                ld      a,(hl)
                out     (scc_cont),a
                inc     hl
                jr      init0

;intialize z180's scc
;

initdma:        ld      hl,addrtab      ;initialize DMA
                ld      c,sar0l
                ld      b,dstat - sar0l
                otimr
                ld      a,00001100b     ;dmac0 - i/o to mem++
                out0    (dmode),a
                ld      a,01001000b     ;1 mem wait, no i/o wait,
                out0    (dcntl),a       ;EDGE trigger, mem ++ to i/o
                                        ;should be EDGE for Tx DMA
                                        ;NOT level
                                        ;- because of DTR/REQ timing
                ret

txend:          ld      a,00010100b     ;isr for dma1 int-complete tx
                out0    (dstat),a       ;disable dma1
                set     0,b             ;set status
                ei
                ret

rxend:          ld      a,00100000b     ;isr for dma0 int
                out0    (dstat),a       ;disable dma0
                set     1,b             ;set status
                ei
                ret

;initialization data table for scc
;table format - register number, then value for the register
;and ends with 0ffh - since scc doesn't have
;register 0ffh...

scctab:         db      09h             ;select WR9
        if scc_a
                db      10000000b       ;reset ch a
        else
                db      01000000b       ;Reset Ch B
        endif
                db      04h             ;select WR4
                db      00000100b       ;async,x1,1stop,parity off
```

```
db      01h             ;select WR1
db      01100000b       ;REQ on Rx

db      02h             ;select WR2
db      00h             ;00h as vector base

db      03h             ;select WR3
db      11000000b       ;Rx 8bit/char

db      05h             ;select WR5
db      01100000b       ;tx 8bit/char

db      06h             ;select WR6
db      00h             ;

db      07h             ;selecl WR7
db      00h             ;

db      09h             ;select WR9
db      00000001b       ;stat low, vis

db      0ah             ;select WR10
db      00000000b       ;set as default

db      0bh             ;select WR11
db      01010110b       ;
;       0                   No xtal
;        1010               TxC,RxC from BRG
;           110             TRxC = BRG output

db      0ch             ;select WR12
db      00h             ;BR TC Low

db      0dh             ;select WR12
db      00h             ;BR TC high

db      0eh             ;select WR14
db      00010110b       ;
;       000                 nothing about DPLL
;         1                 Local loopback
;         0                 No local echo
;          1                DTR/REQ is req
;            1              BRG source = PCLK
;             0             Not enabling BRG yet

db      0eh             ;select WR14
db      00010111b       ;
;       000                 nothing about DPLL
;         1                 Local loopback
;         0                 No local echo
;          1                DTR/REQ is REQ
;            1              BRG source = PCLK
;             1             Enable BRG

db      03h             ;select WR3
db      11000001b       ;rx enable
```

```
                db          01h             ;select WR1
                db          11100000b       ;enable DMA

                db          0fh             ;select WR15
                db          00000000b       ;don't use any of ext/stat int

                db          10h             ;reset ext/stat twice
                db          10h

                db          01h             ,select WR1
                db          11100000b       ;no int

                db          09h             ;selsct WR9
                db          00001001b       ;enable int

                db          0ffh            ;end of table

;source/dist addr table for Z180's dma

addrtab:        db          scc_data        ;dmac0 source
                db          00h
                db          00h

                dw          rx_buff         ,dmac0 dist
                db          00h

                dw          length          ,byte count

                dw          tx_buff+1       ;mar
                db          00h

                db          scc_data        ;iar
                db          00h

                db          00h             ;dummy!

                dw          length-1        ;byte count

;interrupt vector table
                org         sccdma + 200h
z180vect:       .block      2               ;180 int1 vect  00000
                .block      2               ;180 int2 vect  00010
                .block      2               ;180 prt0 vect  00100
                .block      2               ;180 prt1 vect  00110
                dw          rxend           ;180 dmac0 vect 01000
                dw          txend           ;180 dmac1 vect 01010
                 block      2               ,180 csi/o vect 01100
                .block      2               ;180 asci0 vect 01110
                 block      2               ;180 asci1 vect 10000

                org         sccdma + 1000h
tx_buff:        .block      length
rx_buff:        .block      length
temp:            block      1

                end
```

First, this program (Table 15) initializes the SCC by:

Async, X1 mode, 8-bit 1 stop, Non-parity.
Tx and Rx clock from BRG, and BRG set to PCLK/4.Self Loopback

Then, it initializes 4K bytes of memory with a repeating pattern beginning with 00h and increases by one to FFh (uses this as Tx buffer area). Also, it begins another 4K bytes of memory as a Rx buffer with all zeros. After starting, DMA initialization follows:

DMAC0: For Rx data transfer: I/O to Mem, Source address-fixed,Destination address-increasing. Edge sense mode: Interrupt on end of transfer.

DMAC1: For Tx data transfer: Mem to I/O, Source address-increasing, Destination address - fixed. Edge sense mode: Interrupt on end of transfer.

Now, start sending with DMA.

On completion of the transfer, the Z180 DMAC1 generates an interrupt. Then, wait for the interrupt from DMAC0 which shows an end of receive. Now, compare received data with sent data. If the transfer was successful (source data matched with destination), 00h is left in the accumulator. If not successful, 0FFh is left in the accumulator.

This program example specifies a way to initialize the SCC and the Z180 DMA.

CONCLUSION

This Application Note describes only one example of implementation, but gives you an idea of how to design the system using the Z180 and SCC.

For further design assistance, a completed board together with the Debug/Monitor program and the listed sample program are available. If interested, please contact your local Zilog sales office.

ZiLOG

APPLICATION NOTE

USING SCC WITH Z8000 IN SDLC PROTOCOL

This application note describes the use of the Z8030 Serial Communications Controller (Z-SCC) with the Z8000™ CPU to implement a communications controller in a Synchronous Data Link Control (SDLC) mode of operation. In this application, the Z8002 CPU acts as a controller for the Z-SCC. This application note also applies to the non-multiplexed Z8530.

One channel of the Z-SCC communicates with the remote station in Half Duplex mode at 9600 bits/second. To test this application, two Z8000 Development Modules are used. Both are loaded with the same software routines for initialization and for transmitting and receiving messages. The main program of one module requests the transmit routine to send a message of the length indicated by the 'COUNT' parameter. The other system receives the incoming data stream, storing the message in its resident memory.

DATA TRANSFER MODES

The Z-SCC system interface supports the following data transfer modes:

- **Polled Mode.** The CPU periodically polls the Z-SCC status registers to determine if a received character is available, if a character is needed for transmission, and if any errors have been detected.
- **Interrupt Mode.** The Z-SCC interrupts the CPU when certain previously defined conditions are met.
- **Block/DMA Mode.** Using the Wait/Request ($\overline{W}/\overline{REQ}$) signal, the Z-SCC introduces extra wait cycles in order to synchronize the data transfer between a controller or DMA and the Z-SCC.

The example given here uses the block mode of data transfer in its transmit and receive routines.

SDLC PROTOCOL

Data communications today require a communications protocol that can transfer data quickly and reliably. One such protocol, Synchronous Data Link Control (SDLC), is the link control used by the IBM Systems Network Architecture (SNA) communications package. SDLC is a subset of the International Standards Organization (ISO) link control called High-Level Data Link Control (HDLC), which is used for international data communications.

SDLC is a bit-oriented protocol (BOP). It differs from byte-control protocols (BCPs), such as Bisync, in that it uses only a few bit patterns for control functions instead of several special character sequences. The attributes of the SDLC protocol are position dependent rather than character dependent, so the data link control is determined by the position of the byte as well as by the bit pattern.

A character in SDLC is sent as an octet, a group of eight bits. Several octets combine to form a message frame, in which each octet belongs to a particular field. Each message contains: opening flag, address, control, information, Frame Check Sequence (FCS), and closing flag (figure 1).

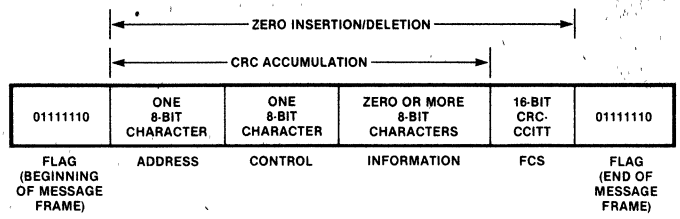


Figure 1. Fields of the SDLC Transmission Frame

Both flag fields contain a unique binary pattern, 01111110, which indicates the beginning or the end of the message frame. This pattern simplifies the hardware interface in receiving devices so that multiple devices connected to a common link do not conflict with one another. The receiving devices respond only after a valid flag character has been detected. Once communication is established with a particular device, the other devices ignore the message until the next flag character is detected.

The address field contains one or more octets, which are used to select a particular station on the data link. An address of eight 1s is a global address code that selects all the devices on the data link. When a primary station sends a frame, the address field is used to select one of several secondary stations. When a secondary station sends a message to the primary station, the address field contains the secondary station address, i.e., the source of the message.

The control field follows the address field and contains information about the type of frame being sent. The control field consists of one octet that is always present.

The information field contains any actual transferred data. This field may be empty or it may contain an unlimited number of octets. However, because of the limitations of the error-checking algorithm used in the frame-check sequence, however, the maximum recommended block size is approximately 4096 octets.

The frame check sequence field follows the information or control field. The FCS is a 16-bit Cyclic Redundancy Check (CRC) of the bits in the address, control, and information fields. The FCS is based on the CRC-CCITT code, which uses the polynomial $(x^{16} + x^{12} + x^5 + 1)$. The Z8030 Z-SCC contains the circuitry necessary to generate and check the FCS field.

Zero insertion and deletion is a feature of SDLC that allows any data pattern to be sent. Zero insertion occurs when five consecutive 1s in the data pattern are transmitted. After the fifth 1, a 0 is inserted before the next bit is sent. The extra 0 does not affect the data in any way and is deleted by the receiver, thus restoring the original data pattern.

Zero insertion and deletion insures that the data stream will not contain a flag character or abort sequence. Six 1s preceded and followed by 0s indicate a flag sequence character. Seven to fourteen 1s signify an abort; 15 or more 1s indicate an idle (inactive) line. Under these three conditions, zero insertion and deletion are inhibited. Figure 2 illustrates the various line conditions.

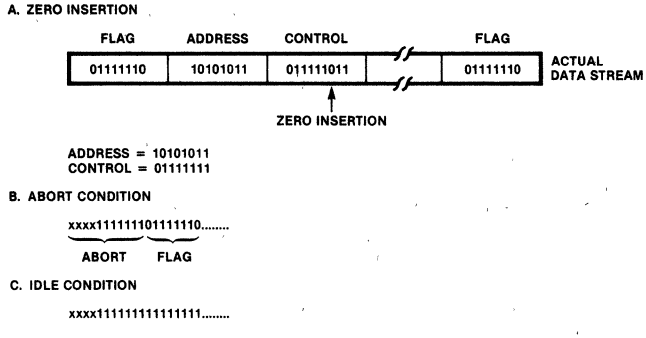


Figure 2. Bit Patterns for Various Line Conditions

The SDLC protocol differs from other synchronous protocols with respect to frame timing. In Bisync mode, for example, a host computer might temporarily interrupt transmission by sending sync characters instead of data. This suspended condition continues as long as the receiver does not time out. With SDLC, however, it is invalid to send flags in the middle of a frame to idle the line. Such action causes an error condition and disrupts orderly operation. Thus, the transmitting device must send a complete frame without interruption. If a message cannot be transmitted completely, the primary station sends an abort sequence and restarts the message transmission at a later time.

SYSTEM INTERFACE

The Z8002 Development Module consists of a Z8002 CPU, 16k words of dynamic RAM, 2k words of EPROM monitor, a Z80A SIO providing dual serial ports, a Z801 CTC peripheral device providing four counter/timer channels, two Z80A PIO devices providing 32 programmable I/O lines, and wire wrap area for prototyping. The block diagram is depicted in Figure 3. Each of the peripherals in the development module is connected in a prioritized daisy chain configuration. The Z-SCC is included in this configuration by tying its IEI line to the IEO line of another device, thus making it one step lower in interrupt priority compared to the other device.

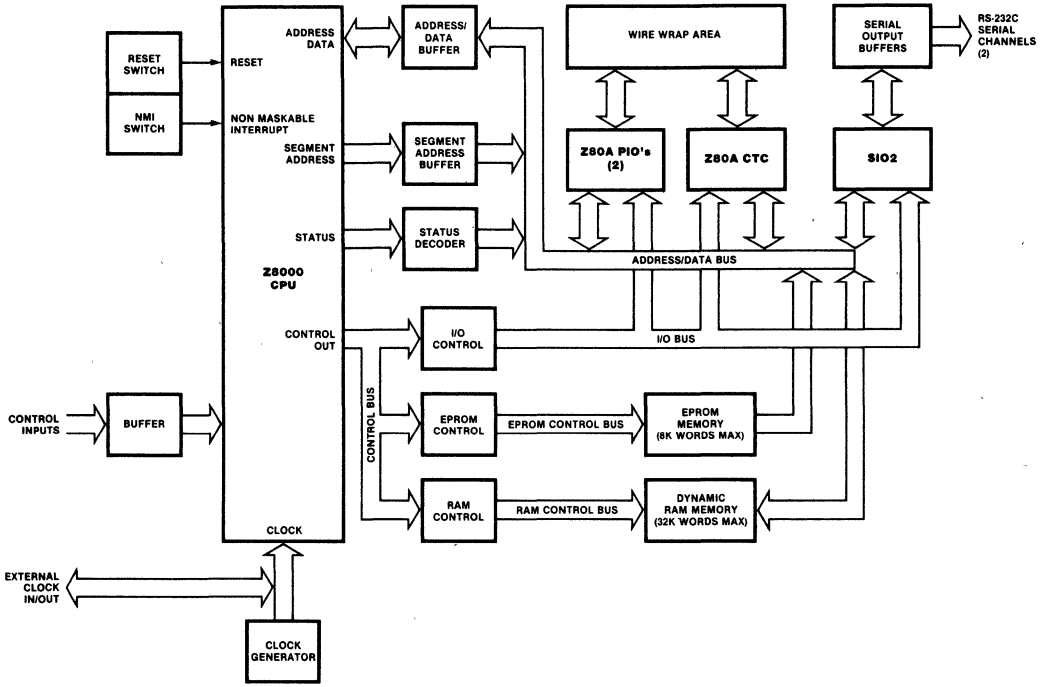


Figure 3. Block Diagram of Z8000 DM

Two Z8000 Development Modules containing Z-SCCs are connected as shown in Figure 4 and Figure 5. The Transmit Data pin of one is connected to the Receive Data pin of the other and vice versa. The Z8002 is used as a host CPU for loading the modules' memories with software routines.

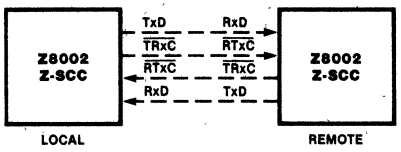


Figure 4. Block Diagram of Two Z8000 CPUs

The Z8002 CPU can address either of the two bytes contained in 16-bit words. The CPU uses an even address (16 bits) to access the most significant byte of a word and an odd address for the least significant byte of a word.

When the Z8002 CPU uses the lower half of the Address/Data bus (AD_0-AD_7 the least significant byte) for byte read and write transactions during I/O operations, these transactions are performed between the CPU and I/O ports located at odd I/O addresses. Since the Z-SCC is attached to the CPU on the lower half of the A/D bus, its registers must appear to the CPU at odd I/O addresses. To achieve this, the Z-SCC can be programmed to select its internal registers using lines AD1-AD5. This is done either automatically with the Force Hardware Reset command in WR9 or by sending a Select Shift Left Mode command to WR0B in channel B of the Z-SCC. For this application, the Z-SCC registers are located at I/O port address 'FExx'. The Chip Select signal ($\overline{CS0}$) is derived by decoding I/O address 'FE' hex from lines AD_8-AD_{15} of the controller.

To select the read/write registers automatically, the Z-SCC decodes lines AD_1-AD_5 in Shift Left mode. The register map for the Z-SCC is depicted in Table 1.

Table 1. Register Map

Address (hex)	Write Register	Read Register
FE01	WR0B	RR0B
FE03	WR1B	RR1B
FE05	WR2	RR2B
FE07	WR3B	RR3B
FE09	WR4B	
FE0B	WR5B	
FE0D	WR6B	
FE0F	WR7B	
FE11	B DATA	B DATA
FE13	WR9	
FE15	WR10B	RR10B
FE17	WR11B	
FE19	WR12B	RR12B
FE1B	WR13B	RR13B
FE1D	WR14B	
FE1F	WR15B	RR15B
FE21	WR0A	RR0A
FE23	WR1A	RR1A
FE25	WR2	RR2A
FE27	WR3A	RR3A
FE29	WR4A	
FE2B	WR5A	
FE2D	WR6A	
FE2F	WR7A	
FE31	A DATA	A DATA
FE33	WR9	
FE35	WR10A	RR10A
FE37	WR11A	
FE39	WR12A	RR12A
FE3B	WR13A	RR13A
FE3D	WR14A	
FE3F	WR15A	RR15A

INITIALIZATION

The Z-SCC can be initialized for use in different modes by setting various bits in its write registers. First, a hardware reset must be

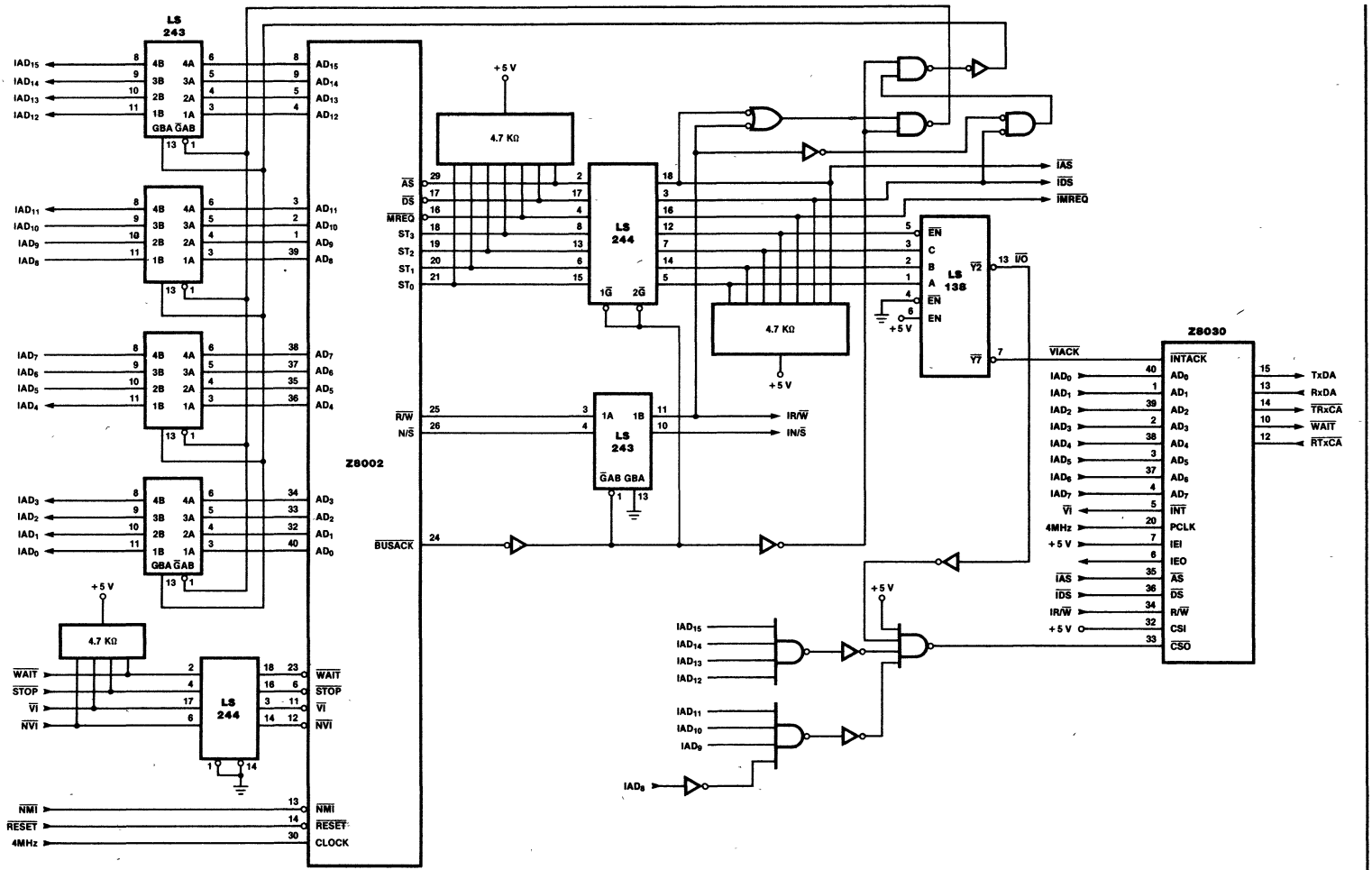


Figure 5. Z8002 With SCC

performed by setting bits 7 and 6 of WR9 to one; the rest of the bits are disabled by writing a logic zero.

SDLC protocol is established by selecting a SDLC mode, sync mode enable, and a x1 clock in WR4. A data rate of 9600 baud, NRZ encoding, and a character length of eight bits are among the other options that are selected in this example (Table 2).

Note that WR9 is accessed twice, first to perform a hardware reset and again at the end of the initialization sequence to enable interrupts. The programming sequence depicted in Table 2 establishes the necessary parameters for the receiver and transmitter so that they are ready to perform communication tasks when enabled.

Table 2. Programming Sequence for Initialization

Register	Value (hex)	Effect
WR9	C0	Hardware reset
WR4	20	x1 clock, SDLC mode, sync mode enable
WR10	80	NRZ, CRC preset to one
WR6	AB	Any station address e.g. "AB"
WR7	7E	SDLC flag (01111110) = "7E"
WR2	20	Interrupt vector "20"
WR11	16	Tx clock from BRG output, $\overline{TRxC}$ pin = BRG out
WR12	CE	Lower byte of time constant = "CE" for 9600 baud
WR13	0	Upper byte = 0
WR14	03	BRG source bit = 1 for PCLK as input, BRG enable
WR15	00	External Interrupt Disable
WR5	60	Transmit 8 bits/character SDLC CRC
WR3	C1	Rx 8 bits/character, Rx enable (Automatic Hunt mode)
WR1	08	RxInt on 1st char & sp. cond., ext int. disable
WR9	09	MIE, VIS, status Low

The Z8002 CPU must be operated in System mode to execute privileged I/O instructions. So the Flag and Control Word (FCW) should be loaded with system normal (S/$\overline{N}$), and the Vectored Interrupt Enable (VIE) bits set. The Program Status Area Pointer (PSAP) is loaded with the address %4400 using the Load Control instruction (LDCTL). If the Z8000 Development Module is intended to be used, the PSAP need not be loaded by the programmer because the development module's monitor loads it automatically after the NMI button is pressed.

Since VIS and Status Low are selected in WR9, the vectors listed in Table 3 will be returned during the Interrupt Acknowledge cycle. Of the four interrupts listed, only two, Ch A Receive Character Available and Ch A Special Receive Condition, are used in the example given here.

Table 3. Interrupt Vectors

Vector (hex)	PS Address* (hex)	Interrupt
28	446E	Ch A Transmit Buffer Empty
2A	4472	Ch A External Status Change
2C	4476	Ch A Receive Char. Available
2E	447A	Ch A Special Receive Condition

*Assuming that PSAP has been set to 4400 hex, "PS Address" refers to the location in the Program Status Area where the service routine address is stored for that particular interrupt.

TRANSMIT OPERATION

To transmit a block of data, the main program calls up the transmit data routine. With this routine, each message block to be transmitted is stored in memory, beginning with location 'TBUF'. The number of characters contained in each block is determined by the value assigned to the 'COUNT' parameter in the main module.

To prepare for transmission, the routine enables the transmitter and selects the Wait On Transmit function; it then enables the wait function. The Wait On Transmit function indicates to the CPU whether or not the Z-SCC is ready to accept data from the CPU. If the CPU attempts to send data to the Z-SCC when the transmit buffer is full, the Z-SCC asserts its $\overline{Wait}$ line and keeps it Low until the buffer is empty. In response, the CPU extends its I/O cycles until the $\overline{Wait}$ line goes inactive, indicating that the Z-SCC is ready to receive data.

The CRC generator is reset and the Transmit CRC bit is enabled before the first character is sent, thus including all the characters sent to the Z-SCC in the CRC calculation.

The Z-SCC's transmit underrun/EOM latch must be reset sometime after the first character is transmitted by writing a Reset Tx Underrun/EOM command to WR0. When this latch is reset, the Z-SCC automatically appends the CRC characters to the end of the message in the case of an underrun condition.

Finally, a three-character delay is introduced at the end of the transmission, which allows the Z-SCC sufficient time to transmit the last data byte and two CRC characters before disabling the transmitter.

RECEIVE OPERATION

Once the Z-SCC is initialized, it can be prepared to receive the message. First, the receiver is enabled, placing the Z-SCC in Hunt mode and thus setting the Sync/Hunt bit in status register RR0 to 1. In Hunt mode, the receiver searches the incoming data stream for flag characters. Ordinarily, the receiver transfers all the data received between flags to the receive data FIFO. If the receiver is in Hunt mode, however, no data transfer takes place until an opening flag is received. If an abort sequence is received, the receiver automatically re-enters Hunt mode. The Hunt status of the receiver is reported by the Sync/Hunt bit in RR0.

The second byte of an SDLC frame is assumed by the Z-SCC to be the address of the secondary stations for which the frame is intended. The Z-SCC provides several options for handling this address. If the Address Search Mode bit D2 in WR3 is set to zero, the address recognition logic is disabled and all the received data bytes are transferred to the receive data FIFO. In this mode, software must perform any address recognition. If the Address Search Mode bit is set to one, only those frames with addresses that match the address programmed in WR6 or the global address (all 1s) will be transferred to the receive data FIFO. If the Sync Character Load Inhibit bit (D1) in WR3 is set to zero, the address comparison is made across all eight bits of WR6. The comparison can be modified so that only the four most significant bits of WR6 need match the received address. This alteration is made by setting the Sync Character Load Inhibit bit to one. In this mode, the address field is still eight bits wide and is transferred to the FIFO in the same manner as the data. In this application, the address search is performed.

When the address match is accomplished, the receiver leaves the Hunt mode and establishes the Receive Interrupt on First Character mode. Upon detection of the receive interrupt, the CPU generates an Interrupt Acknowledge Cycle. The Z-SCC returns the programmed vector %2C. This vector points to the location %4472 in the Program Status Area which contains the receive interrupt service routine address.

The receive data routine is called from within the receive interrupt service routine. While expecting a block of data, the Wait On Receive function is enabled. Receive read buffer RR8 is read and the characters are stored in memory location RBUF. The Z-SCC in SDLC mode automatically enables the CRC checker for all data between opening and closing flags and ignores the Receive CRC Enable bit (D3) in WR3. The result of the CRC calculation for the entire frame in RR1 becomes valid only when the End Of Frame bit is set in RR1. The processor does not use the CRC bytes, because the last two bits of the CRC are never transferred to the receive data FIFO and are not recoverable.

When the Z-SCC recognizes the closing flag, the contents of the Receive Shift register are transferred to the receive data FIFO, the Residue Code (not applicable in this application) is latched, the CRC error bit is latched in the status FIFO, and the End Of Frame bit is set in the receive status FIFO. When the End Of Frame bit reaches the top of the FIFO, a special receive condition interrupt occurs. The special receive condition register RR1 is read to determine the result of the CRC calculation. If the CRC error bit is zero, the frame received is assumed to be correct; if the bit is 1, an error in the transmission is indicated.

Before leaving the interrupt service routine, Reset Highest IUS (Interrupt Under Service), Enable Interrupt on Next Receive Character, and Enter Hunt Mode commands are issued to the Z-SCC.

If receive overrun error is made, a special condition interrupt occurs. The Z-SCC presents vector %2E to the CPU, and the service routine located at address %447A is executed. Register RR1 is read to determine which error occurred. Appropriate action to correct the error should be taken by the user at this point. Error Reset and Reset Highest IUS commands are given to the Z-SCC before returning to the main program so that the other lower-priority interrupts can occur.

In addition to searching the data stream for flags, the receiver also scans for seven consecutive 1s, which indicates an abort condition. This condition is reported in the Break/Abort bit (D7) in RR0. This is one of many possible external status conditions. As a result transitions of this bit can be programmed to cause an external status interrupt. The abort condition is terminated when a zero is received, either by itself or as the leading zero of a flag. The receiver leaves Hunt mode only when a flag is found.

SOFTWARE

Software routines are presented in the following pages. These routines can be modified to include various other options (e.g., SDLC Loop, Digital Phase Locked Loop etc.). By modifying the WR10 register, different encoding methods (e.g., NRZI, FM0, FM1) other than NRZ can be used.

Appendix

Software Routines

```
plzasm    1.3
LOC     OBJ CODE     STMT SOURCE STATEMENT

                       1
                       2
                       3          SDLC MODULE
                         $LISTON $TTY
                         CONSTANT
                         WR0A     :=  %FE21              !BASE ADDRESS FOR WR0 CHANNEL A!
                         RR0A     :=  %FE21              !BASE ADDRESS FOR RR0 CHANNEL A!
                         RBUF     :=  %5400              !BUFFER AREA FOR RECEIVE CHARACTER!
                         PSAREA   :=  %4400              !START ADDRESS FOR PROGRAM STAT AREA!
                         COUNT    :=   12                !NO. OF CHAR. FOR TRANSMIT ROUTINE!
0000                     GLOBAL MAIN PROCEDURE
                         ENTRY

0000 7601                          LDA      R1,PSAREA
0002 4400
0004 7D1D                          LDCTL    PSAPOFF,R1     !LOAD PSAP!
0006 2100                          LD       R0,#%5000
0008 5000
000A 3310                          LD       R1(#%1C),R0    !FCW VALUE(%5000) AT %441C FOR VECTORED!
000C 001C                                                  !INTERRUPTS!
000E 7600                          LDA      R0,REC
0010 00D6'
0012 3320                          LD       R1(#%76),R0    !EXT. STATUS SERVICE ADDR. AT %4476 IN!
0014 0076                                                  !PSA!
0016 7600                          LDA      R0,SPCOND
0018 00FA'
001A 3310                          LD       R1(#%7A),R0    !SP.COND.SERVICE ADDR AT %447A IN PSA!
001C 007A
001E 5F00                          CALL     INIT
0020 0034'
0022 5F00                          CALL     TRANSMIT
0024 008C'
0026 E8FF                          JR       $

0028 AB                  TBUF:     BVAL     %AB            !STATION ADDRESS!
0029 48                            BVAL     'H'
002A 45                            BVAL     'E'
002B 4C                            BVAL     'L'
002C 4C                            BVAL     'L'
002D 4F                            BVAL     'O'
002E 20                            BVAL     ' '
002F 54                            BVAL     'T'
0030 48                            BVAL     'H'
0031 45                            BVAL     'E'
0032 52                            BVAL     'R'
0033 45                            BVAL     'E'

0034                               END      MAIN
```

```
                        !****************** INITIALIZATION ROUTINE FOR Z-SCC ***********************!

0034                            GLOBAL  INIT PROCEDURE
                                ENTRY
0034 2100                       LD      R0,#15          !NO.OF PORTS TO WRITE TO!
0036 000F
0038 7602                       LDA     R2,SCCTAB       !ADDRESS OF DATA FOR PORTS!
003A 004E'
003C 2101                   ALOOP:  LD      R1,#WR0A
003E FE21
0040 0029                       ADDB    RL1,@R2
0042 A920                       INC     R2
0044 3A22                       OUTIB   @R1,@R2,R0      !POINT TO WR0A,WR1A ETC THRO LOOP!
0046 0018
0048 8D04                       TEST    R0              !END OF LOOP?!
004A EEF8                       JR      NZ,ALOOP        !NO,KEEP LOOPING!
004C 9E08                       RET
004E 12                     SCCTAB: BVAL    2*9
004F C0                         BVAL    %C0             !WR9=HARDWARE RESET!
0050 08                         BVAL    2*4
0051 20                         BVAL    %20             !WR4=X1 CLK,SDLC,SYNC MODE!
0052 14                         BVAL    2*10
0053 80                         BVAL    %80             !WR10=CRC PRESET ONE,NRZ,FLAG ON IDLE,!
                                                            !FLAG ON UNDERRUN!
0054 0C                         BVAL    2*6
0055 AB                         BVAL    %AB             !WR6= ANY ADDRESS FOR SDLC STATION!
0056 0E                         BVAL    2*7
0057 7E                         BVAL    %7E             !WR7=SDLC FLAG CHAR!
0058 04                         BVAL    2*2
0059 20                         BVAL    %20             !WR2=INT VECTOR %20!
005A 16                         BVAL    2*11
005B 16                         BVAL    %16             !WR11=Tx CLOCK & TRxC OUT=BRG OUT!
005C 18                         BVAL    2*12
005D CE                         BVAL    %CE             !WR12= LOWER TC=CE!
005E 1A                         BVAL    2*13
005F 00                         BVAL    0               !WR13= UPPER TC=0!
0060 1C                         BVAL    2*14
0061 03                         BVAL    %03             !WR14=BRG ON,BRG SRC=PCLK!
0062 1E                         BVAL    2*15
0063 00                         BVAL    %00             !WR15=EXT INT. DISABLE!
0064 0A                         BVAL    2*5
0065 60                         BVAL    %60             !WR5=Tx 8 BITS/CHAR, SDLC CRC!
0066 06                         BVAL    2*3
0067 C5                         BVAL    %C5             !WR3=ADDR SRCH,REC ENABLE!
0068 02                         BVAL    2*1
0069 08                         BVAL    %08             !WR1=RX INT ON 1ST & SP COND,!
                                                            !EXT INT DISABLE!
006A 12                         BVAL    2*9
006B 09                         BVAL    %09             !WR9= MIE,VIS,STATUS LOW!
006C                        END     INIT


                        !****************** RECEIVE ROUTINE ************************************!

                        !                   RECEIVE A BLOCK OF MESSAGE                          !

006C                            GLOBAL  RECEIVE PROCEDURE
                                ENTRY
006C C828                       LDB     RL0,#%28        !WAIT ON RECV.!
006E 3A86                       OUTB    WR0A+2,RL0
0070 FE23
0072 6008                       LDB     RL0,%A8
0074 00A8
0076 3A86                       OUTB    WR0A+2,RL0      !ENABLE WAIT FNC. SP. COND. INT!
0078 FE23
007A 2101                       LD      R1,#RR0A+16
007C FE31
007E 2102                       LD      R2,#COUNT+2     !COUNT+2 CHARACTERS TO READ!
0080 000E
0082 2103                       LD      R3,#RBUF        !RECEIVE BUFFER IN MEMORY!
0084 5400
0086 3A18                       INDRB   @R3,@R1,R2      !READ THE ENTIRE MESSAGE!
0088 0230
008A 9E08                       RET
008C                        END RECEIVE
```

```
                        |*****************  TRANSMIT ROUTINE ************************************|
                        |                   SEND A BLOCK OF EIGHT  DATA CHARACTERS               |
                        |                   THE BLOCK STARTS AT LOCATION TBUF                    |

008C                            GLOBAL  TRANSMIT PROCEDURE
                                ENTRY
008C 2102                               LD      R2,#TBUF        !PTR TO START OF BUFFER!
008E 0028'
0090 C868                               LDB     RL0,#%68
0092 3A86                               OUTB    WR0A+10,RL0     !ENABLE TRANSMITTER!
0094 FE2B
0096 C800                               LDB     RL0,#%00        !WAIT ON TRANSMIT!
0098 3A86                               OUTB    WR0A+2,RL0
009A FE23
009C C888                               LDB     RL0,#%88
009E 3A86                               OUTB    WR0A+2,RL0      !WAIT ENABLE!
00A0 FE23
00A2 C880                               LDB     RL0,#%80
00A4 3A86                               OUTB    WR0A,RL0        !RESET TxCRC GENERATOR!
00A6 FE21
00A8 2101                               LD      R1,#WR0A+16     !WR8A SELECTED!
00AA FE31
00AC 2100                               LD      R0,#1
00AE 0001
00B0 C869                               LDB     RL0,#%69        !SDLC CRC!
00B2 3A86                               OUTB    WR0A+10,RL0     !WR5A=TxCRC ENABLE!
00B4 FE2B
00B6 3A22                               OTIRB   @R1,@R2,R0      !SEND ADDRESS!
00B8 0010
00BA C8C0                               LDB     RL0,#%C0
00BC 3A86                               OUTB    WR0A,RL0        !RESET TxUND/EOM LATCH!
00BE FE21
00C0 2100                               LD      R0,#COUNT-1
00C2 000B
00C4 3A22                               OTIRB   @R1,@R2,R0      !SEND MESSAGE!
00C6 0010
00C8 2100                               LD      R0,#926         !CREATE DELAY BEFORE DISABLING!
00CA 039E
00CC F081                       DEL:    DJNZ    R0,DEL          !TRANSMITTER SO THAT CRC CAN BE!
00CE C800                               LDB     RL0,#0          !SENT!
00D0 3A86                               OUTB    WR0A+10,RL0     !DISABLE TRANSMITTER!
00D2 FE2B
00D4 9E08                               RET
00D6                            END TRANSMIT


                        |************** RECEIVE INT. SERVICE ROUTINE **************************|

00D6                              GLOBAL REC PROCEDURE
                                  ENTRY
00D6 93F3                               PUSH    @R15,R3
00D8 93F2                               PUSH    @R15,R2
00DA 93F1                               PUSH    @R15,R1
00DC 93F0                               PUSH    @R15,R0
00DE 3A94                               INB     RL1,RR0A        !READ STATUS REG RR0A!
00E0 FE21
00E2 A690                               BITB    RL1,#0          !TEST IF Rx CHAR SET!
00E4 E602                               JR      Z,RESET         !YES CALL RECEIVE ROUTINE!
00E6 5F00                               CALL    RECEIVE
00E8 006C'
00EA C838                       RESET:  LDB     RL0,#%38
00EC 3A86                               OUTB    WR0A,RL0        !RESET HIGHEST IUS!
00EE FE21
00F0 97F0                               POP     R0,@R15
00F2 97F1                               POP     R1,@R15
00F4 97F2                               POP     R2,@R15
00F6 97F3                               POP     R3,@R15
00F8 7B00                               IRET
00FA                            END REC
```

```
                    !************ SPECIAL CONDITION INTERRUPT SERVICE ROUTINE ***************!

00FA                        GLOBAL SPCOND PROCEDURE
                            ENTRY

00FA 93F0                           PUSH    @R15,R0
00FC 3A84                           INB     RL0,RR0A+2      !READ ERRORS!
00FE FE23
0100 A687                           BITB    RL0,#7          !END OF FRAME ?!
                            !PROCESS OVERRUN,FRAMING ERRORS IF ANY!
0102 E603                           JR      Z,RESE
0104 C820                           LDB     RL0,#%20
0106 3A86                           OUTB    WR0A,RL0        ! YES,ENABLE INT ON NEXT REC CHAR!
0108 FE21
010A C830                   RESE:   LDB     RL0,#%30
010C 3A86                           OUTB    WR0A,RL0        !ERROR RESET!
010E FE21
0110 C808                           LDB     RL0,#%08
0112 3A86                           OUTB    WR0A+2,RL0      !WAIT DISABLE,RxINT ON 1ST OR SP COND.!
0114 FE23
0116 C838                           LDB     RL0,#%38
0118 3A86                           OUTB    WR0A,RL0        !RESET HIGHEST IUS!
011A FE21
011C 97F0                           POP     R0,@R15
011E 7B00                           IRET

0120                        END SPCOND

                            END SDLC
```

ZiLOG

APPLICATION NOTE

SCC IN BINARY SYNCHRONOUS COMMUNICATIONS

October 1982

Zilog's Z8030 Z-SCC Serial Communications Controller is one of a family of components that are Z-BUS™ compatible with the Z8000™ CPU. Combined with a Z8000 CPU (or other existing 8- or 16-bit CPUs with nonmultiplexed buses when using the Z8530 SCC), the Z-SCC forms an integrated data communications controller that is more cost effective and more compact than systems incorporating UARTs, baud rate generators, and phase-locked loops as separate entities.

The approach examined here implements a communications controller in a Binary Synchronous mode of operation, with a Z8002 CPU acting as controller for the Z-SCC.

One channel of the Z-SCC is used to communicate with the remote station in Half Duplex mode at 9600 bits/second. To test this application, two Z8000 Development Modules are used. Both are loaded with the same software routines for initialization and for transmitting and receiving messages. The main program of one module requests the transmit routine to send a message of the length indicated in the 'COUNT' parameter. The other system receives the incoming data stream, storing the message in its resident memory.

DATA TRANSFER MODES

The Z-SCC system interface supports the following data transfer modes:

- **Polled Mode.** The CPU periodically polls the Z-SCC status registers to determine the availability of a received character, if a character is needed for transmission, and if any errors have been detected.
- **Interrupt Mode.** The Z-SCC interrupts the CPU when certain previously defined conditions are met.
- **Block/DMA Mode.** Using the Wait/Request ($\overline{W}/\overline{REQ}$) signal, the Z-SCC introduces extra wait cycles to synchronize data transfer between a CPU or DMA controller and the Z-SCC.

The example given here uses the block mode of data transfer in its transmit and receive routines.

SYNCHRONOUS MODES

Three variations of character-oriented synchronous communications are supported by the Z-SCC: Monosync, Bisync, and External Sync (Figure 1). In Monosync mode, a single sync character is transmitted, which is then compared to an identical sync character in the receiver. When the receiver recognizes this sync character, synchronization is complete; the receiver then transfers subsequent characters into the receiver FIFO in the Z-SCC.

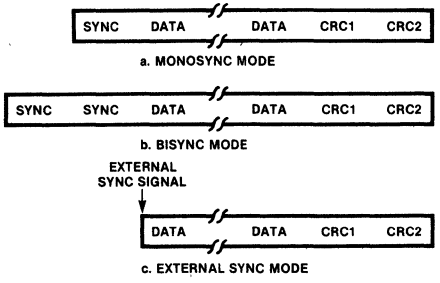


Figure 1. Synchronous Modes of Communication

Bisync mode uses a 16-bit or 12-bit sync character in the same way to obtain synchronization. External Sync mode uses an external signal to mark the beginning of the data field; i.e., an external input pin (SYNC) indicates the start of the information field.

In all synchronous modes, two Cyclic Redundancy Check (CRC) bytes can be concatenated to the message to detect data transmission errors. The CRC bytes inserted in the transmitted message are compared to the CRC bytes computed to the receiver. Any differences found are held in the receive error FIFO.

SYSTEM INTERFACE

The Z8002 Development Module consists of a Z8002 CPU, 16K words of dynamic RAM, 2K words of EPROM

Two Z8000 Development Modules containing Z-SCCs are connected as shown in Figure 3 and Figure 4. The Transmit Data pin of one is connected to the Receive Data pin of the other and vice versa. The Z8002 is used as a host CPU for loading the modules' memories with software routines.

The Z8000 CPU can address either of the two bytes contained in 16-bit words. The CPU uses an even address (16 bits) to access the most-significant byte of a word and an odd address for the least-significant byte of a word.

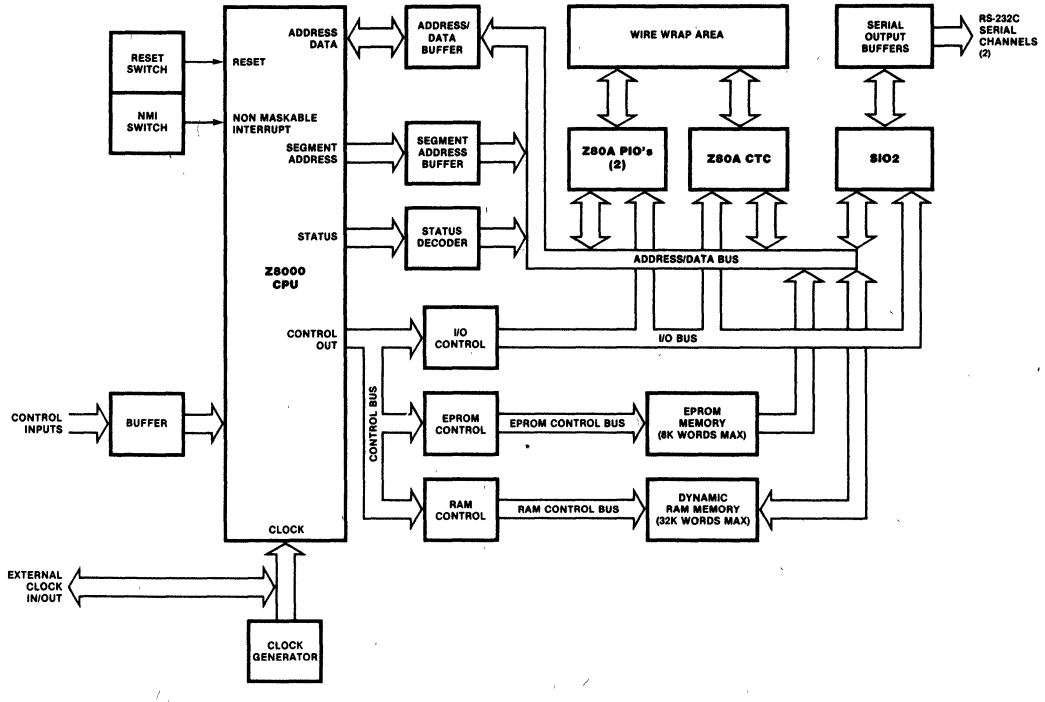


Figure 2. Block Diagram of Z8000 DM

monitor, a Z80A SIO providing dual serial ports, a Z80A CTC peripheral device providing four counter/timer channels, two Z80A PIO devices providing 32 programmable I/O lines, and wire wrap area for prototyping. The block diagram is depicted in Figure 2. Each of the peripherals in the development module is connected in a prioritized daisy-chain configuration. The Z-SCC is included in this configuration by tying its IEI line to the IEO line of another device, thus making it one step lower in interrupt priority compared to the other device.

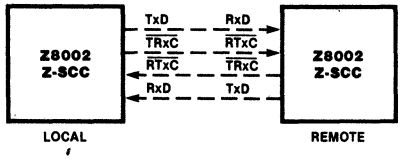


Figure 3. Block Diagram of Two Z8000 Development Modules

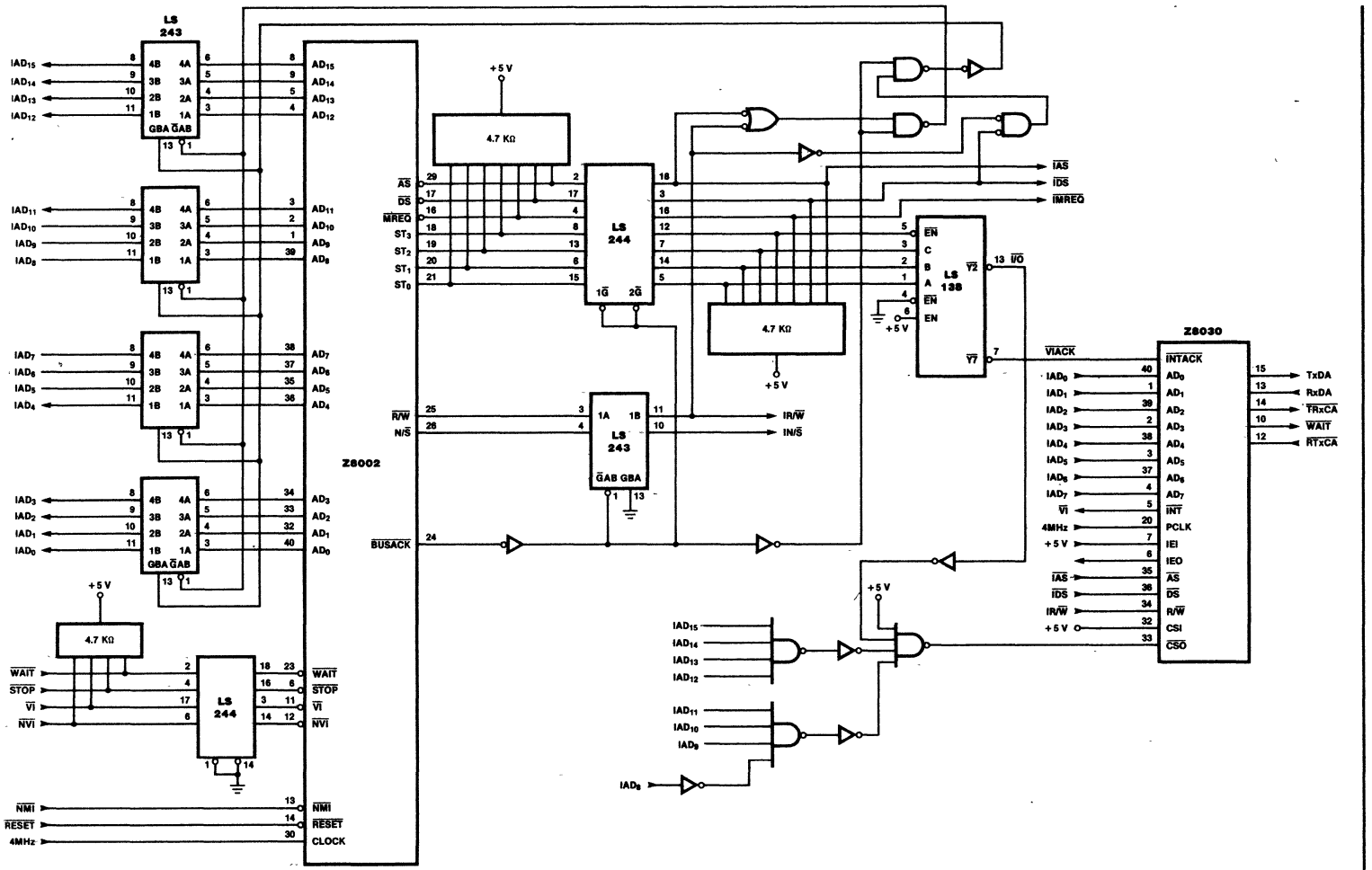


Figure 4. Z8002 with SCC

When the Z8002 CPU uses the lower half of the Address/Data bus (AD_0-AD_7 the least significant byte) for byte read and write transactions during I/O operations, these transactions are performed between the CPU and I/O ports located at odd I/O addresses. Since the Z-SCC is attached to the CPU on the lower half of the A/D bus, its registers must appear to the CPU at odd I/O addresses. To achieve this, the Z-SCC can be programmed to select its internal registers using lines AD_1-AD_5. This is done either automatically with the Force Hardware Reset command in WR9 or by sending a Select Shift Left Mode command to WR0B in channel B of the Z-SCC. For this application, the Z-SCC registers are located at I/O port address 'FExx'. The Chip Select signal ($\overline{CS0}$) is derived by decoding I/O address 'FE' hex from lines AD_8-AD_{15} of the controller. The Read/Write registers are automatically selected by the Z-SCC when internally decoding lines AD_1-AD_5 in Shift Left mode. To select the Read/Write registers automatically, the Z-SCC decodes lines AD_1-AD_5 in Shift Left mode. The register map for the Z-SCC is depicted in Table 1.

Table 1. Register Map

Address (hex)	Write Register	Read Register
FE01	WR0B	RR0B
FE03	WR1B	RR1B
FE05	WR2	RR2B
FE07	WR3B	RR3B
FE09	WR4B	
FE0B	WR5B	
FE0D	WR6B	
FE0F	WR7B	
FE11	B DATA	B DATA
FE13	WR9	
FE15	WR10B	RR10B
FE17	WR11B	
FE19	WR12B	RR12B
FE1B	WR13B	RR13B
FE1D	WR14B	
FE1F	WR15B	RR15B
FE21	WR0A	RR0A
FE23	WR1A	RR1A
FE25	WR2	RR2A
FE27	WR3A	RR3A
FE29	WR4A	
FE2B	WR5A	
FE2D	WR6A	
FE2F	WR7A	
FE31	A DATA	A DATA
FE33	WR9	
FE35	WR10A	RR10A
FE37	WR11A	
FE39	WR12A	RR12A
FE3B	WR13A	RR13A
FE3D	WR14A	
FE3F	WR15A	RR15A

INITIALIZATION

The Z-SCC can be initialized for use in different modes by setting various bits in its Write registers. First, a hardware reset must be performed by setting bits 7 and 6 of WR9 to one; the rest of the bits are disabled by writing a logic zero.

Bisync mode is established by selecting a 16-bit sync character, Sync Mode Enable, and a X1 clock in WR4. A data rate of 9600 baud, NRZ encoding, and a data character length of eight bits are among the other options that are selected in this example (Table 2).

Note that WR9 is accessed twice, first to perform a hardware reset and again at the end of the initialization sequence to enable the interrupts. The programming sequence depicted in Table 2 establishes the necessary parameters for the receiver and the transmitter so that, when enabled, they are ready to perform communication tasks. To avoid internal race and false interrupt conditions, it is important to initialize the registers in the sequence depicted in this application note.

The Z8002 CPU must be operated in System mode in order to execute privileged I/O instructions, so the Flag Control Word (FCW) should be loaded with System/Normal ($S/\overline{N}$), and the Vectored Interrupt Enable (VIE) bits set. The Program Status Area Pointer (PSAP) is loaded with address %4400 using the Load Control instruction (LDCTL). If the Z8000 Development Module is intended to be used, the PSAP need not be loaded by the programmer as the development modules monitor loads it automatically after the NMI button is pressed.

Table 2. Programming Sequence for Initialization

Register	Value (hex)	Effect
WR9	C0	Hardware reset
WR4	10	x1 clock, 16-bit sync, sync mode enable
WR10	0	NRZ, CRC preset to zero
WR6	AB	Any sync character "AB"
WR7	CD	Any sync character "CD"
WR2	20	Interrupt vector "20"
WR11	16	Tx clock from BRG output, TRxC pin = BRG out
WR12	CE	Lower byte of time constant = "CE" for 9600 baud
WR13	0	Upper byte = 0
WR14	03	BRG source bit = 1 for PCLK as input, BRG enable
WR15	00	External interrupt disable
WR5	64	Tx 8 bits/character, CRC-16
WR3	C1	Rx 8 bits/character, Rx enable (Automatic Hunt mode)
WR1	08	RxInt on 1st char & sp. cond., ext. int. disable)
WR9	09	MIE, VIS, Status Low

Since VIS and Status Low are selected in WR9, the vectors listed in Table 3 will be returned during the Interrupt Acknowledge cycle. Of the four interrupts listed, only two, Ch A Receive Character Available and Ch A Special Receive Condition, are used in the example given here.

Table 3. Interrupt Vectors

Vector (hex)	PS Address* (hex)	Interrupt
28	446E	Ch A Transmit Buffer Empty
2A	4472	Ch A External Status Change
2C	4476	Ch A Receive Char. Available
2E	447A	Ch A Special Receive Condition

* "PS Address" refers to the location in the Program Status Area where the service routine address is stored for that particular interrupt, assuming that PSAP has been set to 4400 hex.

TRANSMIT OPERATION

To transmit a block of data, the main program calls up the transmit data routine. With this routine, each message block to be transmitted is stored in memory, beginning with location 'TBUF'. The number of characters contained in each block is determined by the value assigned to the 'COUNT' parameter in the main module.

To prepare for transmission, the routine enables the transmitter and selects the Wait On Transmit function; it then enables the wait function. The Wait On Transmit function indicates to the CPU whether or not the Z-SCC is ready to accept data from the CPU. If the CPU attempts to send data to the Z-SCC when the transmit buffer is full, the Z-SCC asserts its Wait line and keeps it Low until the buffer is empty. In response, the CPU extends its I/O cycles until the Wait line goes inactive, indicating that the Z-SCC is ready to receive data.

The CRC generator is reset and the Transmit CRC bit is enabled before the first character is sent, thus including all the characters sent to the Z-SCC in the CRC calculation, until the Transmit CRC bit is disabled. CRC generation can be disabled for a particular character by resetting the TxCRC bit within the transmit routine. In this application, however, the Transmit CRC bit is not disabled, so that all characters sent to the Z-SCC are included in the CRC calculation.

The Z-SCC's transmit underrun/EOM latch must be reset sometime after the first character is transmitted by writing a Reset Tx Underrun/EOM command to WR0. When this latch is reset, the Z-SCC automatically appends the CRC characters to the end of the message in the case of an underrun condition.

Finally, a five-character delay is introduced at the end of the transmission, which allows the Z-SCC sufficient time to transmit the last data byte, two CRC characters, and two sync characters before disabling the transmitter.

RECEIVE OPERATION

Once the Z-SCC is initialized, it can be prepared to receive data. First, the receiver is enabled, placing the Z-SCC in Hunt mode and thus

setting the Sync/Hunt bit in status register RR0 to 1. In Hunt mode, the receiver is idle except that it searches the incoming data stream for a sync character match. When a match is discovered between the incoming data stream and the sync characters stored in WR6 and WR7, the receiver exits the Hunt mode, resetting the Sync/Hunt bit in status register RR0 and establishing the Receive Interrupt On First Character mode. Upon detection of the receive interrupt, the CPU generates an Interrupt Acknowledge cycle. The Z-SCC sends to the CPU vector %2C, which points to the location in the Program Status Area from which the receive interrupt service routine is accessed.

The receive data routine is called from within the receive interrupt service routine. While expecting a block of data, the Wait On Receive function is enabled. Receive data buffer RR8 is read, and the characters are stored in memory locations starting at RBUF. The Start of Text (%02) character is discarded. After the End of Transmission character (%04) is received, the two CRC bytes are read. The result of the CRC check becomes valid two characters later, at which time, RR1 is read and the CRC error bit is checked. If the bit is zero, the message received can be assumed correct; if the bit is 1, an error in the transmission is indicated.

Before leaving the interrupt service routine, Reset Highest IUS (Interrupt Under Service), Enable Interrupt on Next Recieve Character, and Enter Hunt Mode commands are issued to the Z-SCC.

If a receive overrun error is made, a special condition interrupt occurs. The Z-SCC presents the vector %2E to the CPU, and the service routine located at address %447A is executed. The Special Receive Condition register RR1 is read to determine which error occurred. Appropriate action to correct the error should be taken by the user at this point. Error Reset and Reset Highest IUS commands are given to the Z-SCC before returning to the main program so that the other lower priority interrupts can occur.

SOFTWARE

Software routines are presented in the following pages. These routines can be modified to include various versions of Bisync protocol, such as Transparent and Nontransparent modes. Encoding methods other than NRZ (e.g., NRZI, FM0, FM1) can also be used by modifying WR10.

Appendix

Software Routines

```
plzasm    1.3
LOC    OBJ CODE     STMT SOURCE STATEMENT

                      1          BISYNC MODULE
                      $LISTON $TTY
                      CONSTANT
                      WR0A     :=  %FE21              !BASE ADDRESS FOR WR0 CHANNEL A!
                      RR0A     :=  %FE21              !BASE ADDRESS FOR RR0 CHANNEL A!
                      RBUF     :=  %5400              !BUFFER AREA FOR RECEIVE CHARACTER!
                      PSAREA   :=  %4400              !START ADDRESS FOR PROGRAM STAT AREA!
                      COUNT    :=   12                !NO. OF CHAR. FOR TRANSMIT ROUTINE!
0000                  GLOBAL MAIN PROCEDURE
                      ENTRY
0000 7601                        LDA     R1,PSAREA
0002 4400
0004 7D1D                        LDCTL   PSAPOFF,R1     !LOAD PSAP!
0006 2100                        LD      R0,#%5000
0008 5000
000A 3310                        LD      R1(#%1C),R0    !FCW VALUE(%5000) AT %441C FOR VECTORED!
000C 001C                                               !INTERRUPTS!
000E 7600                        LDA     R0,REC
0010 00F4'
0012 3310                        LD      R1(#%76),R0    !EXT. STATUS SERVICE ADDR. AT %4476 IN!
0014 0076                                               !PSA!
0016 7600                        LDA     R0,SPCOND
0018 011E'
001A 3310                        LD      R1(#%7A),R0    !SP.COND.SERVICE ADDR AT %447A IN PSA!
001C 007A
001E 5F00                        CALL    INIT
0020 0034'
0022 5F00                        CALL    TRANSMIT
0024 00A6'
0026 E8FF                        JR      $
0028 02                TBUF:     BVAL    %02            !START OF TEXT!
0029 31                          BVAL    '1'            !BVAL MEANS BYTE VALUE. MESSAGE CHAR.!
002A 32                          BVAL    '2'
002B 33                          BVAL    '3'
002C 34                          BVAL    '4'
002D 35                          BVAL    '5'
002E 36                          BVAL    '6'
002F 37                          BVAL    '7'
0030 38                          BVAL    '8'
0031 39                          BVAL    '9'
0032 30                          BVAL    '0'
0033 31                          BVAL    '1'
0034                             END     MAIN
```

```
                !****************** INITIALIZATION ROUTINE FOR Z-SCC ***********************!

0034                    GLOBAL  INIT PROCEDURE
                        ENTRY
0034 2100               LD      R0,#15          !NO.OF PORTS TO WRITE TO!
0036 000F
0038 7602               LDA     R2,SCCTAB       !ADDRESS OF DATA FOR PORTS!
003A 004E'
003C 2101       ALOOP:  LD      R1,#WR0A
003E FE21
0040 0029               ADDB    RL1,@R2
0042 A920               INC     R2
0044 3A22               OUTIB   @R1,@R2,R0      !POINT TO WR0A,WR1A ETC THRO LOOP!
0046 0018
0048 8D04               TEST    R0              !END OF LOOP?!
004A EEF8               JR      NZ,ALOOP        !NO,KEEP LOOPING!
004C 9E08               RET
004E 12         SCCTAB: BVAL    2*9
004F C0                 BVAL    %C0             !WR9=HARDWARE RESET!
0050 08                 BVAL    2*4
0051 10                 BVAL    %10             !WR4=X1 CLK,16 BIT SYNC MODE!
0052 14                 BVAL    2*10
0053 00                 BVAL    0               !WR10=CRC PRESET ZERO,NRZ,16 BIT SYNC!
0054 0C                 BVAL    2*6
0055 AB                 BVAL    %AB             !WR6=ANY SYNC CHAR %AB!
0056 0E                 BVAL    2*7
0057 CD                 BVAL    %CD             !WR7=ANY SYNC CHARR %CD!
0058 04                 BVAL    2*2
0059 20                 BVAL    %20             !WR2=INT VECTOR %20!
005A 16                 BVAL    2*11
005B 16                 BVAL    %16             !WR11=TxCLOCK & TRxC OUT=BRG OUT!
005C 18                 BVAL    2*12
005D CE                 BVAL    %CE             !WR12= LOWER TC=%CE!
005E 1A                 BVAL    2*13
005F 00                 BVAL    0               !WR13= UPPER TC=0!
0060 1C                 BVAL    2*14
0061 03                 BVAL    %03             !WR14=BRG ON, ITS SRC=PCLK!
0062 1E                 BVAL    2*15
0063 00                 BVAL    %00             !WR15=NO EXT INT EN.!
0064 0A                 BVAL    2*5
0065 64                 BVAL    %64             !WR5= TX 8 BITS/CHAR, CRC-16!
0066 06                 BVAL    2*3
0067 C1                 BVAL    %C1             !WR3=RX 8 BITS/CHAR, REC ENABLE!
0068 02                 BVAL    2*1
0069 08                 BVAL    %08             !WR1=RxINT ON 1ST OR SP COND!
                                                !    EXT INT DISABLE!
006A 12                 BVAL    2*9
006B 09                 BVAL    %09             !WR9= MIE,VIS,STATUS LOW!
006C            END INIT


                !****************** RECEIVE ROUTINE ************************************!

                !                   RECEIVE A BLOCK OF MESSAGE                          !
                !                   THE LAST CHARACTER SHOULD BE EOT(%04)               !

006C                    GLOBAL  RECEIVE PROCEDURE
                        ENTRY
006C C828               LDB     RL0,#%28        !WAIT ON RECV.!
006E 3A86               OUTB    WR0A+2,RL0
0070 FE23
0072 6008               LDB     RL0,%A8
0074 00A8
0076 3A86               OUTB    WR0A+2,RL0      !ENABLE WAIT 1ST CHAR,SP.COND. INT!
0078 FE23
007A 2101               LD      R1,#RR0A+16
007C FE31
007E 3C18               INB     RL0,@R1         !READ STX CHARACTER!
0080 C8C9               LDB     RL0,#%C9
0082 3A86               OUTB    WR0A+6,RL0      !Rx CRC ENABLE!
0084 FE27
0086 2103               LD      R3,#RBUF
0088 5400
008A 3C18       READ:   INB     RL0,@R1         !READ MESSAGE!
008C 2E38               LDB     @R3,RL0         !STORE CHARACTER IN RBUF!
008E AB30               DEC     R3,#1
0090 0A08               CPB     RL0,#%04        !IS IT END OF TRANSMISSION ?!
0092 0404
0094 EEFA               JR      NZ,READ
0096 3C18               INB     RL0,@R1         !READ PAD1!
0098 3C18               INB     RL0,@R1         !READ PAD2!
009A 3A84               INB     RL0,RR0A+2      !READ CRC STATUS!
009C FE23
                !       PROCESS CRC ERROR IF ANY, AND GIVE ERROR RESET COMMAND IN WR0A !
009E C800               LDB     RL0,#0
00A0 3A86               OUTB    WR0A+6,RL0      !DISABLE RECEIVER!
00A2 FE27
00A4 9E08               RET
00A6            END RECEIVE
```

```
                        |*****************  TRANSMIT ROUTINE ************************************|
                        |                   SEND A BLOCK OF DATA CHARACTERS                     |
                        |                   THE BLOCK STARTS AT LOCATION TBUF                   |

00A6                            GLOBAL  TRANSMIT PROCEDURE
                                ENTRY
00A6 2102                               LD      R2,#TBUF        |PTR TO START OF BUFFER|
00A8 0028'
00AA C86C                               LDB     RL0,#%6C
00AC 3A86                               OUTB    WR0A+10,RL0     |ENABLE TRANSMITTER|
00AE FE2B
00B0 C800                               LDB     RL0,#%00        |WAIT ON TRANSMIT|
00B2 3A86                               OUTB    WR0A+2,RL0
00B4 FE23
00B6 C888                               LDB     RL0,#%88
00B8 3A86                               OUTB    WR0A+2,RL0      |WAIT ENABLE,INT ON 1ST & SP COND|
00BA FE23
00BC C880                               LDB     RL0,#%80
00BE 3A86                               OUTB    WR0A,RL0        |RESET TxCRC GENERATOR|
00C0 FE21
00C2 2101                               LD      R1,#WR0A+16     |WR8A SELECTED|
00C4 FE31
00C6 C86D                               LDB     RL0,#%6D
00C8 3A86                               OUTB    WR0A+10,RL0     |Tx CRC ENABLE|
00CA FE2B
00CC 2100                               LD      R0,#1
00CE 0001
00D0 3A22                               OTIRB   @R1,@R2,R0      |SEND START OF TEXT|
00D2 0010
00D4 C8C0                               LDB     RL0,#%C0
00D6 3A86                               OUTB    WR0A,RL0        |RESET TxUND/EOM LATCH|
00D8 FE21
00DA 2100                               LD      R0,#COUNT-1
00DC 000B
00DE 3A22                               OTIRB   @R1,@R2,R0      |SEND MESSAGE|
00E0 0010
00E2 C804                               LDB     RL0,#%04
00E4 3E18                               OUTB    @R1,RL0         |SEND END OF TRANSMISSION CHARACTER|
00E6 2100                               LD      R0,#1670        |CREATE DELAY BEFORE DISABLING|
00E8 0686
00EA F081                       DEL:    DJNZ    R0,DEL
00EC C800                               LDB     RL0,#0
00EE 3A86                               OUTB    WR0A+10,RL0     |DISABLE TRANSMITTER|
00F0 FE2B
00F2 9E08                               RET
00F4                            END TRANSMIT


                        |************** RECEIVE INT. SERVICE ROUTINE ************************|

00F4                              GLOBAL REC PROCEDURE
                                  ENTRY
00F4 93F0                               PUSH    @R15,R0
00F6 3A84                               INB     RL0,RR0A        |READ STATUS FROM RR0A|
00F8 FE21
00FA A684                               BITB    RL0,#4          |TEST IF SYNC HUNT RESET|
00FC EE02                               JR      NZ,RESET        |YES CALL RECEIVE ROUTINE|
00FE 5F00                               CALL    RECEIVE
0100 006C'
0102 C808                       RESET:  LDB     RL0,#%08
0104 3A86                               OUTB    WR0A+2,RL0      |WAIT DISABLE|
0106 FE23
0108 C8D1                               LDB     RL0,#%D1
010A 3A86                               OUTB    WR0A+6,RL0      |ENTER HUNT MODE|
010C FE27
010E C820                               LDB     RL0,#%20
0110 3A86                               OUTB    WR0A,RL0        |ENABLE INT ON NEXT CHAR|
0112 FE21
0114 C838                               LDB     RL0,#%38
0116 3A86                               OUTB    WR0A,RL0        |RESET HIGHEST IUS|
0118 FE21
011A 97F0                               POP     R0,@R15
011C 7B00                               IRET
011E                            END REC
```

```
                !************ SPECIAL CONDITION INTERRUPT SERVICE ROUTINE ***************!

011E                    GLOBAL SPCOND PROCEDURE
                        ENTRY

011E 93F0                       PUSH    @R15,R0
0120 3A84                       INB     RL0,RR0A+2      !READ ERRORS!
0122 FE23
                        !PROCESS ERRORS!
0124 C830                       LDB     RL0,#%30
0126 3A86                       OUTB    WR0A,RL0        !ERROR RESET!
0128 FE21
012A C808                       LDB     RL0,#%08
012C 3A86                       OUTB    WR0A+2,RL0      !WAIT DISABLE,RxINT ON 1ST OR SP COND.!
012E FE23
0130 C8D1                       LDB     RL0,#%D1
0132 3A86                       OUTB    WR0A+6,RL0      !HUNT MODE,REC. ENABLE!
0134 FE27
0136 C838                       LDB     RL0,#%38
0138 3A86                       OUTB    WR0A,RL0        !RESET HIGHEST IUS!
013A FE21
013C 97F0                       POP     R0,@R15
013E 7B00                       IRET

0140                    END SPCOND

                        END BISYNC

   0 errors
Assembly complete
```

ZiLOG

APPLICATION NOTE

ON-CHIP OSCILLATOR DESIGN

DESIGN AND BUILD RELIABLE, COST-EFFECTIVE, ON-CHIP OSCILLATOR CIRCUITS THAT ARE TROUBLE FREE. PUTTING OSCILLATOR THEORY INTO A PRACTICAL DESIGN MAKES FOR A MORE DEPENDABLE CHIP.

INTRODUCTION

This Application Note (App Note) is written for designers using Zilog Integrated Circuits with on-chip oscillators; circuits in which the amplifier portion of a feedback oscillator is contained on the IC. This App Note covers common theory of oscillators, and requirements of the circuitry (both internal and external to the IC) which comes from the theory for crystal and ceramic resonator based circuits.

Purpose and Benefits

The purposes and benefits of this App Note include:

1. Providing designers with greater understanding of how oscillators work and how to design them to avoid problems.

2. To eliminate field failures and other complications resulting from an unawareness of critical on-chip oscillator design constraints and requirements.

Problem Background

Inadequate understanding of the theory and practice of oscillator circuit design, especially concerning oscillator startup, has resulted in an unreliable design and subsequent field problems (See on page 10 for reference materials and acknowledgements).

OSCILLATOR THEORY OF OPERATION

The circuit under discussion is called the Pierce Oscillator (Figures 1, 2). The configuration used is in all Zilog on-chip oscillators. Advantages of this circuit are low power consumption, low cost, large output signal, low power level in the crystal, stability with respect to V_{cc} and temperature, and low impedances (not disturbed by stray effects). One drawback is the need for high gain in the amplifier to compensate for feedback path losses.

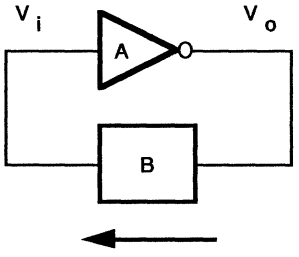


Figure 1. Basic Circuit and Loop Gain

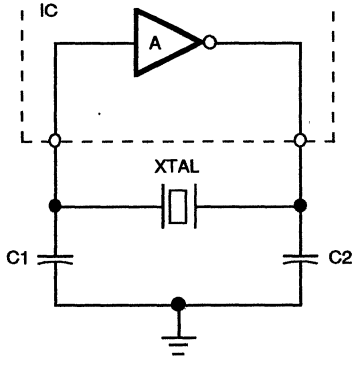


Figure 2. Zilog Pierce Oscillator

Pierce Oscillator (Feedback Type)

The basic circuit and loop gain is shown in Figure 1. The concept is straightforward; gain of the amplifier is A = Vo/Vi. The gain of the passive feedback element is B = Vi/Vo. Combining these equations gives the equality AB = 1. Therefore, the total gain around the loop is unity. Also, since the gain factors A and B are complex numbers, they have phase characteristics. It is clear that the total phase shift around the loop is forced to zero (i.e., 360 degrees), since V_{IN} must be in phase with itself. In this circuit, the amplifier ideally provides 180 degrees of phase shift (since it is an inverter). Hence, the feedback element is forced to provide the other 180 degrees of phase shift.

Additionally, these gain and phase characteristics of both the amplifier and the feedback element vary with frequency. Thus, the above relationships must apply at the frequency of interest. Also, in this circuit the amplifier is an active element and the feedback element is passive. Thus, by definition, the gain of the amplifier at frequency must be greater than unity, if the loop gain is to be unity.

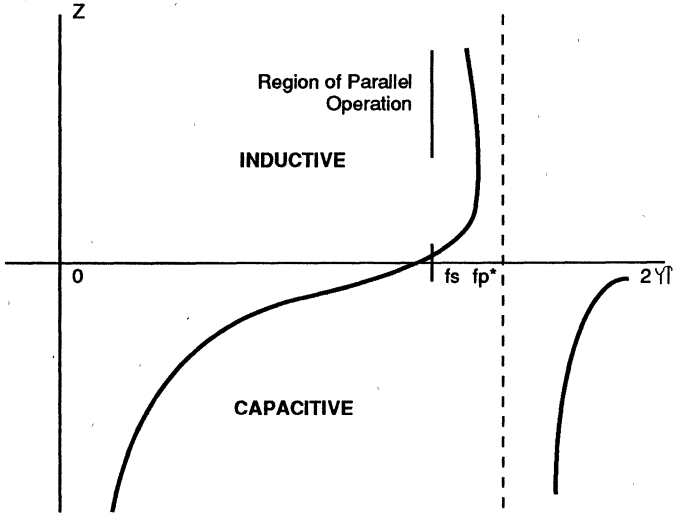
The described oscillator amplifies its own noise at startup until it settles at the frequency which satisfies the gain/phase requirement AB = 1. This means loop gain equals one, and loop phase equals zero (360 degrees). To do this, the loop gain at points around the frequency of oscillation must be greater than one. This achieves an average loop gain of one at the operating frequency.

The amplifier portion of the oscillator provides gain > 1 plus 180 degrees of phase shift. The feedback element provides the additional 180 degrees of phase shift without attenuating the loop gain to < 1. To do this the feedback element is inductive, i.e., it must have a positive reactance at the frequency of operation. The feedback elements discussed are quartz crystals and ceramic resonators.

Quartz Crystals

A quartz crystal is a piezoelectric device; one which transforms electrical energy to mechanical energy and vice versa. The transformation occurs at the resonant frequency of the crystal. This happens when the applied AC electric field is sympathetic in frequency with the mechanical resonance of the slice of quartz. Since this characteristic can be made very accurate, quartz crystals are normally used where frequency stability is critical. Typical frequency tolerance is .005 to 0.3%.

The advantage of a quartz crystal in this application is its wide range of positive reactance values (i.e., it looks inductive) over a narrow range of frequencies (Figure 3).



* fs - fp is very small (approximately 300 parts per million)

Figure 3. Series vs. Parallel Resonance

However, there are several ranges of frequencies where the reactance is positive; these are the fundamental (desired frequency of operation), and the third and fifth mechanical overtones (approximately 3 and 5 times the fundamental frequency). Since the desired frequency range in this application is always the fundamental, the overtones must be suppressed. This is done by reducing the loop gain at these frequencies. Usually, the amplifier's gain roll off, in combination with the crystal parasitics and load capacitors, is sufficient to reduce gain and prevent oscillation at the overtone frequencies.

The following parameters are for an equivalent circuit of a quartz crystal (Figure 4):

L - motional inductance (typ 120 mH @ 4 MHz)

C - motional capacitance (typ .01 pf @ 4 MHz)

R - motional resistance (typ 36 ohm @ 4 MHz)

Cs - shunt capacitance resulting from the sum of the capacitor formed by the electrodes (with the quartz as a dielectric) and the parasitics of the contact wires and holder (typ 3 pf @ 4 MHz).

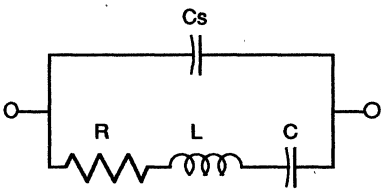
The series resonant frequency is given by:

Fs = 1/(2π x sqrt of LC),
where Xc and Xl are equal.

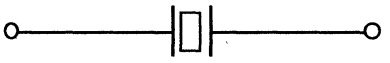
Thus, they cancel each other and the crystal is then R shunted by Cs with zero phase shift.

The parallel resonant frequency is given by:

Fp = 1/[2π x sqrt of L (C Ct/C+Ct)],
where: $Ct = C_L + C_s$



Quartz Equivalent Circuit



Symbolic Representation

Figure 4. Quartz Oscillator

Series vs. Parallel Resonance. There is very little difference between series and parallel resonance frequencies (Figure 3). A series resonant crystal (operating at zero phase shift) is desired for non-inverting amplifiers. A parallel resonant crystal (operating at or near 180 degrees of phase shift) is desired for inverting amps. Figure 3 shows that the difference between these two operating modes is small. Actually, all crystals have operating points in both serial and parallel modes. A series resonant circuit will NOT have load caps C1 and C2. A data sheet for a crystal designed for series operation does not have a load cap spec. A parallel resonant crystal data sheet specifies a load cap value which is the series combination of C1 and C2. For this App Note discussion, since all the circuits of interest are inverting amplifier based, only the parallel mode of operation is considered.

OSCILLATOR THEORY OF OPERATION

Ceramic Resonators

Ceramic resonators are similar to quartz crystals, but are used where frequency stability is less critical and low cost is desired. They operate on the same basic principle as quartz crystals as they are piezoelectric devices and have a similar equivalent circuit. The frequency tolerance is wider (0.3 to 3%), but the ceramic costs less than quartz.

Figure 5 shows reactance vs. frequency and Figure 6 shows the equivalent circuit.

Typical values of parameters are L = .092 mH, C = 4.6 pf, R = 7 ohms and Cs = 40 pf, all at 8 MHz. Generally, ceramic resonators tend to start up faster but have looser frequency tolerance than quartz. This means that external circuit parameters are more critical with resonators.

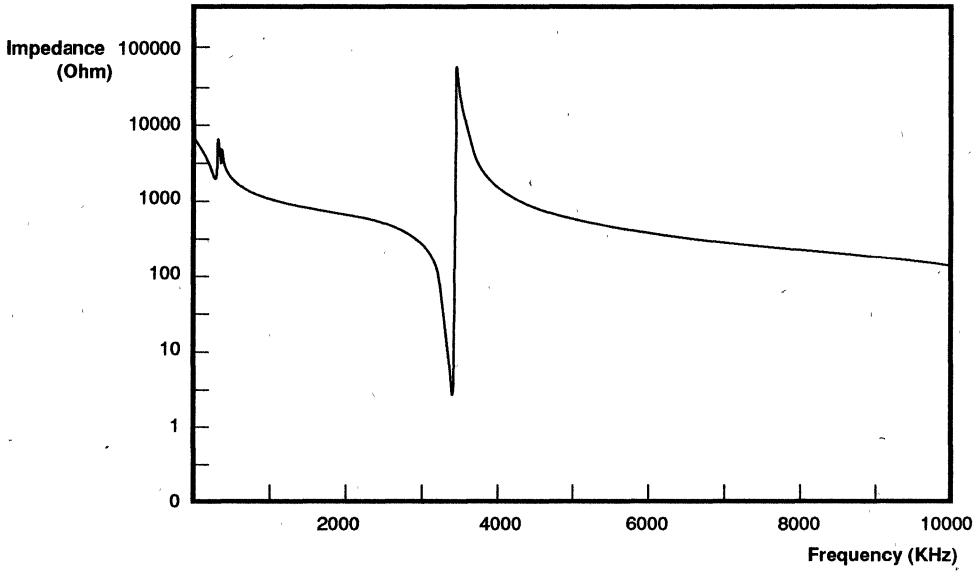


Figure 5. Ceramic Resonator Reactance

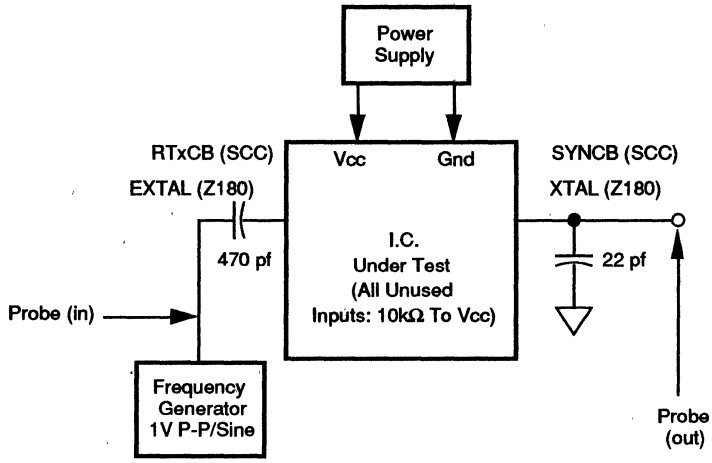


Figure 6. Gain Measurement

Load Capacitors

The effects/purposes of the load caps are:

Cap C2 combined with the amp output resistance provides a small phase shift. It also provides some attenuation of overtones.

Cap C1 combined with the crystal resistance provides additional phase shift.

These two phase shifts place the crystal in the parallel resonant region of Figure 3.

Crystal manufacturers specify a load capacitance number. This number is the load seen by the crystal which is the series combination of C1 and C2, including all parasitics (PCB and holder). This load is specified for crystals meant to be used in a parallel resonant configuration. The effect on startup time; if C1 and C2 increase, startup time increases to the point at which the oscillator will not start. Hence, for fast and reliable startup, over manufacture of large quantities, the load caps should be sized as low as possible without resulting in overtone operation.

Amplifier Characteristics

The following text discusses open loop gain vs. frequency, open loop phase vs. frequency, and internal bias.

Open Loop Gain vs. Frequency over Iot, VCC, Processs Split, and Temp. Closed loop gain must be adequate to start the oscillator and keep it running at the desired frequency. This means that the amplifier open loop gain must be equal to one plus the gain required to overcome the losses in the feedback path, across the frequency band and up to the frequency of operation. This is over full process, Iot, V_{cc}, and temperature ranges. Therefore, measuring the open loop gain is not sufficient; the losses in the feedback path (crystal and load caps) must be factored in.

Open Loop Phase vs. Frequency. Amplifier phase shift at and near the frequency of interest must be 180 degrees plus some, minus zero. The parallel configuration allows for some phase delay in the amplifier. The crystal adjusts to this by moving slightly down the reactance curve (Figure 3).

Internal Bias. Internal to the IC, there is a resistor placed from output to input of the amplifier. The purpose of this feedback is to bias the amplifier in its linear region and to provide the startup transition. Typical values are 1M to 20M ohms.

PRACTICE: CIRCUIT ELEMENT AND LAY OUT CONSIDERATIONS

The discussion now applies prior theory to the practical application.

Amplifier and Feedback Resistor

The elements of the circuit, internal to the IC, include the amplifier, feedback resistor, and output resistance. The amplifier is modeled as a transconductance amplifier with a gain specified as I_{OUT}/V_{IN} (amps per volt).

Transconductance/Gain. The loop gain AB = gm x Z1, where gm is amplifier transconductance (gain) in amps/volt and Z1 is the load seen by the output. AB must be greater than unity at and about the frequency of operation to sustain oscillation.

Gain Measurement Circuit. The gain of the amplifier can be measured using the circuits of Figures 6 & 7. This may be necessary to verify adequate gain at the frequency of interest and in determining design margin.

Gain Requirement vs. Temperature, Frequency and Supply Voltage. The gain to start and sustain oscillation (Figure 8) must comply with:

$gm > 4\pi^2 f^2 Rq\ C_{IN} C_{OUT} t \times M$

where: M is a quartz form factor $= (1 + C_{OUT}/C_{IN} + C_{OUT}/C_{OUT})^2$

Output Impedance. The output impedance limits power to the XTAL and provides small phase shift with load cap C2.

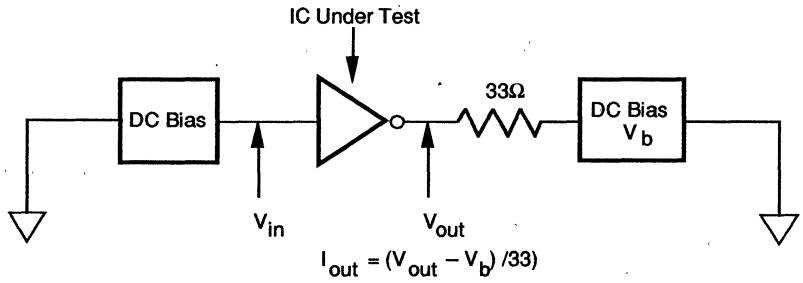
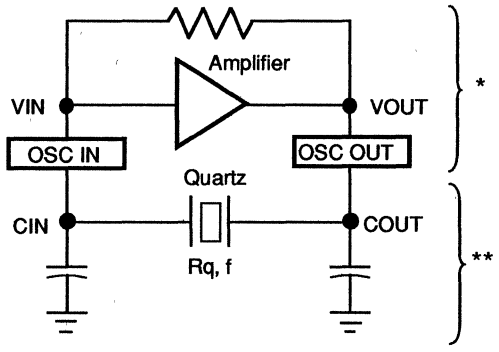


Figure 7. Transconductance (gm) Measurement



* Inside chip, feedback resistor biases the amplifier in the high gm region.

** External components typically: CIN = COUT = 30 to 50 pf (add 10 pf pin cap).

Figure 8. Quartz Oscillator Configuration

Load Capacitors

In the selection of load caps it is understood that parasitics are always included.

Upper Limits. If the load caps are too large, the oscillator will not start because the loop gain is too low at the operating frequency. This is due to the impedance of the load capacitors. Larger load caps produce a longer startup.

Lower Limits. If the load caps are too small, either the oscillator will not start (due to inadequate phase shift around the loop), or it will run at a 3rd, 5th, or 7th overtone frequency (due to inadequate suppression of higher overtones).

Capacitor Type and Tolerance. Ceramic caps of ±10% tolerance should be adequate for most applications.

Ceramic vs. Quartz. Manufacturers of ceramic resonators generally specify larger load cap values than quartz crystals. Quartz C is typically 15 to 30 pf and ceramic typically 100pf.

Summary. For reliable and fast startup, capacitors should be as small as possible without resulting in overtone operation. The selection of these capacitors is critical and all of the factors covered in this note should be considered.

Feedback Element

The following text describes the specific parameters of a typical crystal:

Drive Level. There is no problem at frequencies greater than 1 MHz and V_{cc} = 5V since high frequency AT cut crystals are designed for relatively high drive levels (5-10 mw max).

A typical calculation for the approximate power dissipated in a crystal is:

$$P = 2R (\pi \times f \times C \times V_{cc})^2$$

Where. R = crystal resistance of 40 ohms, C = C1 + Co = 20 pf. The calculation gives a power dissipation of 2 mW at 16 MHz.

Series Resistance. Lower series resistance gives better performance but costs more. Higher R results in more power dissipation and longer startup, but can be compensated by reduced C1 and C2. This value ranges from 200 ohms at 1 MHz down to 15 ohms at 20 MHz.

Frequency. The frequency of oscillation in parallel resonant circuits is mostly determined by the crystal (99.5%). The external components have a negligible effect (0.5%) on frequency. The external components (C1,C2) and layout are chosen primarily for good startup and reliability reasons.

Frequency Tolerance (initial temperature and aging). Initial tolerance is typically ±.01%. Temperature tolerance is typically ±.005% over the temp range (-30 to +100 degrees C). Aging tolerance is also given, typically ±.005%.

Holder. Typical holder part numbers are HC6, 18, 25, 33, 44.

Shunt Capacitance. (Cs) typically <7 pf.

Mode. Typically the mode (fundamental, 3rd or 5th overtone) is specified as well as the loading configuration (series vs. parallel).

The ceramic resonator equivalent circuit is the same as shown in Figure 4. The values differ from those specified in the theory section. Note that the ratio of L/C is much lower than with quartz crystals. This gives a lower Q which allows a faster startup and looser frequency tolerance (typically ±0.9% over time and temperature) than quartz.

Layout

The following text explains trace layout as it affects the various stray capacitance parameters (Figure 9).

Traces and Placement. Traces connecting crystal,caps, and the IC oscillator pins should be as short and wide as possible (this helps reduce parasitic inductance and resistance). Therefore, the components (caps and crystal) should be placed as close to the oscillator pins of the IC as possible.

Grounding/Guarding. The traces from the oscillator pins of the IC should be guarded from all other traces (clock, V_{cc}, address/data lines) to reduce crosstalk. This is usually accomplished by keeping other traces away from the oscillator circuit and by placing a ground ring around the traces/components (Figure 9).

Measurement and Observation

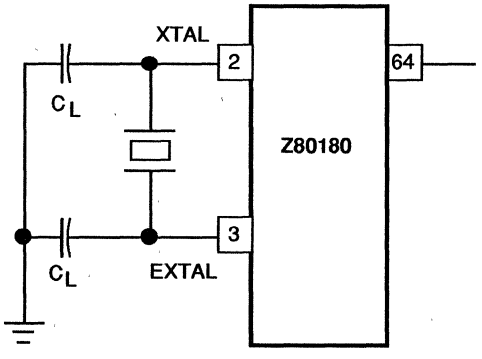
Connection of a scope to either of the circuit nodes is likely to affect operation because the scope adds 3-30 pf of capacitance and 1M-10M ohms of resistance to the circuit.

Indications of an Unreliable Design

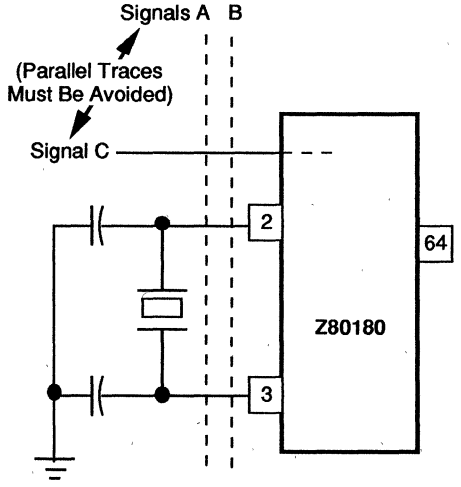
There are two major indicators which are used in working designs to determine their reliability over full lot and temperature variations. They are:

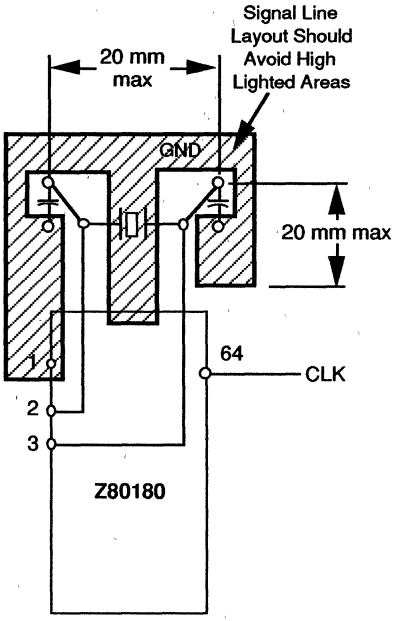
Start Up Time. If start up time is excessive, or varies widely from unit to unit, there is probably a gain problem. C1/C2 needs to be reduced; the amplifier gain is not adequate at frequency, or crystal Rs is too large.

Output Level. The signal at the amplifier output should swing from ground to V_{CC}. This indicates there is adequate gain in the amplifier. As the oscillator starts up, the signal amplitude grows until clipping occurs, at which point, the loop gain is effectively reduced to unity and constant oscillation is achieved. A signal of less than 2.5 Vp-p is an indication that low gain may be a problem. Either C1/C2 should be made smaller or a low R crystal should be used.



Clock Generator Circuit





Board Design Example
(Top View)

- To prevent induced noice, the crystal and load capacitors should be physically located as close to the LSI as possible.
- Signal lines should not run parallel to the clock oscillator inputs. In particullar, the clock input circuitry and the system clock output (pin 64) should be separated as much as possible.
- V_{CC} power lines should be separated from the clock oscillator input circuitry.
- Resistivity between XTAL or EXTAL and the other pin should be greater than 10 MΩ

Figure 9. Circuit Board Design Rules

SUMMARY

Understanding the Theory of Operation of oscillators, combined with practical applications, should give designers enough information to design reliable oscillator circuits. Proper selection of crystals and load capacitors, along with good layout practices, results in a cost effective, trouble free design. Reference the following text for Zilog products with on-chip oscillators and their general/specific requirements.

ZILOG PRODUCT USING ON-CHIP OSCILLATORS

Zilog products that have on-chip oscillators:

Z8® Family: All
Z80®: C01, C11, C13, C15, C50, C90, 180, 181, 280
Z8000®: 8581
Communications Products: SCC™, ISCC™, ESCC™

ZILOG CHIP PARAMETERS

The following are some recommendations on values/parameters of components for use with Zilog on-chip oscillators. These are only recommendations; no guarantees are made by performance of components outside of Zilog ICs. Finally, the values/parameters chosen depend on the application. This App Note is meant as a guideline to making these decisions. Selection of optimal components is always a function of desired cost/performance tradeoffs.

Note: All load capacitance specs include stray capacitance.

Z8 Family

General Requirements:

Crystal Cut: AT cut, parallel resonant, fundamental mode.
Crystal Co: < 7 pf for all frequencies.
Crystal Rs: < 100 ohms for all frequencies.
Load Capacitance: 10 to 22 pf, 15 pf typical.

Specific Requirements:

8604: xtal or ceramic, f = 1 - 8 MHz.
8600/10: f = 8 MHz.
8601/03/11/13: f = 12.5 MHz.
8602: xtal or ceramic, f = 4 MHz.
8680/81/82/84/91: f = 8, 12, 16, MHz.
8671: f = 8 MHz.
8612: f = 12, 16 MHz.
86C08/E08: f = 8, 12 MHz.
86C09/19: xtal/resonator, f = 8 MHz, C = 47 pf max.
86C00/10/20/30: f = 8, 12, 16 MHz.
86C11/21/91/40/90: f = 12, 16, 20 MHz.
86C27/97: f = 4, 8 MHz.
86C12: f = 12, 16 MHz.
Super8 (all): f = 1 - 20 MHz.

Z8000 Family (8581 only)

General Requirements:

Crystal cut: AT cut, parallel resonant, fundamental mode.
Crystal Co: < 7 pf for all frequencies.
Crystal Rs: < 150 ohms for all frequencies.
Load capacitance: 10 to 33 pf.

Z80 Family

General Requirements:

Crystal cut: AT cut, parallel resonant, fundamental mode.
Crystal Co: < 7 pf for all frequencies.
Crystal Rs: < 60 ohms for all frequencies.
Load capacitance: 10 to 22 pf.

Specific Requirements:

84C01: C1 = 22 pf, C2 = 33 pf (typ); f = DC to 10 MHz.
84C90: DC to 8 MHz.
84C50: same as 84C01.
84C11/13/15: C1 = C2 = 20 -33 pf; f = 6 -10 MHz
80180: f = 12, 16, 20 MHz (Fxtal = 2 x sys. clock).
80280: f = 20 MHz (Fxtal = 2 x Fsysclk).
80181: TBD.

ZILOG CHIP PARAMETERS (Continued)

Communications Family

General Requirements:

Crystal cut: AT cut, parallel resonant, fundamental mode.
Crystal Co: < 7 pf for all frequencies.
Crystal Rs: < 130 ± 15% ohms for all frequencies.
Load capacitance: 20 to 33 pf.
Frequency: cannot exceed PCLK.

Specific Requirements:

8530/85C30/SCC: f = 1 - 6 MHz (10 MHz SCC), 1 - 8.5 MHz (8 MHz SCC).
85130/ESCC (16/20 MHz), f = 1 - 16.384 MHz.
16C35/ISCC: f = 1 -10 MHz.

REFERENCES MATERIALS AND ACKNOWLEDGEMENTS

Intel Corp., Application Note AP-155, "Oscillators for Micro Controllers", order #230659-001, by Tom Williamson, Dec. 1986.

Motorola 68HC11 Reference Manual.

National Semiconductor Corp., App Notes 326 and 400.

Zilog, Inc., Steve German; Figures 4 and 8.

Zilog, Inc., Application Note, "Design Considerations Using Quartz Crystals with Zilog Components" - Oct. 1988.

Data Sheets; CTS Corp. Knights Div., Crystal Oscillators.

ZiLOG

APPLICATION NOTE

INTERFACING THE Z8500 PERIPHERALS TO THE 68000

INTRODUCTION

This application note discusses interfacing Zilog's Z8500 family of peripherals to the 68000 microprocessor. The Z8500 peripheral family includes the Z8536 Counter/Timer and Parallel I/O Unit (CIO), the Z8038 FIFO Input/Output Interface Unit (FIO), and the Z8530 Serial Communications Controller (SCC). This document discusses the Z8500/68000 interfaces and presents hardware examples and verification techniques. One of the three hardware examples given in this application note shows how to implement the Z8500/68000 interface using a single-chip programmable logic array (PAL).

This application note about interfacing supplements the following documents, which discuss the individual components of the interface.

- Z8036 Z-CIO/Z8536 CIO Technical Manual (document number 00-2091-01)
- Z8038 Z-FIO Technical Manual (document number 00-2051-01)
- Z8030/Z8530 SCC Technical Manual (document number 00-2057-01)
- Motorola 16-Bit Microprocessor User's Manual 3rd ed. Englewood Cliffs, N.J., Prentice-Hall, Inc. 1979.
- Monolithic Memories Bipolar LSI 1982 Databook

This application note is divided into four sections. The first section gives a general description of the Z8500 family and discusses pin functions, interrupt structures, and the programming of operating modes. The second section discusses the Z8500 interface itself. It shows how the different Z8500 control signals are generated from the 68000 signals and summarizes the critical timings for the three types of bus cycle. The third section shows three examples of implementing the 68000-to-Zilog-peripheral interface. The fourth section suggests methods of verifying the interface design by checking the three different types of bus cycle: Read, Write, and Interrupt Acknowledge.

GENERAL Z8500 FAMILY DESCRIPTION

The Z8500 family is made up of programmable peripherals that can interface easily to the bus of any nonmultiplexed CPU microprocessor, such as the 68000. The three members of this family, the CIO, SCC, and FIO, can solve many design problems. The peripherals' operating modes can be programmed simply by writing to their internal registers.

Programming the Operating Modes

The CPU can access two types of register: Control and Data. Depending on the peripheral, registers are selected with either the A_0, A_1, $A/\overline{B}$, or $D/\overline{C}$ function pins.

Peripheral operating modes are initialized by programming internal registers. Since these registers are not directly addressable by the CPU, a two-step procedure using the Control register is required: first, the address of the internal register is written to the Control register, then the data is written to the Control register. A state machine determines whether an address or data is being written to the Control register. Reading an internal register follows a similar two-step

procedure: first, the address is written, then the data is read.

The Data registers that are most frequently accessed, for example, the SCC's transmit and receive buffer, can be addressed directly by the CPU with a single read or write operation. This reduces overhead in data transfers between the peripheral and CPU.

GENERATING Z8500 CONTROL SIGNALS

This section shows how to generate the Z8500 control signals. To simplify the discussion, the section is divided into two parts. The first part takes each individual Z8500 signal and shows how it is generated from the 68000 signals. The second part discusses the Z8500 timing that must be met when generating the control signals.

Z8500 Signal Generation

The right-hand side of Table 1 lists the Z8500 signals that must be generated. Each of these signals is discussed in a separate paragraph.

A_0, A_1, A/$\overline{B}$, D/$\overline{C}$. These pins are used to select the peripheral's Control and Data registers that program the different operating modes. They can be connected to the 68000 A_1 and A_2 Address bus lines.

$\overline{CE}$. Each peripheral has an active Low Chip Enable that can be derived by ANDing the selected address decode and the 68000's Address Strobe ($\overline{AS}$). The active Low $\overline{AS}$ guarantees that the 68000 addresses are valid.

D_0-D_7. The Z8500 Data bus can be directly connected to the lowest byte (D_0-D_7) of the 68000 Data bus.

IEI and IEO. The peripherals use these pins to decide the interrupt priority. The highest priority device should have its IEI tied High. Its IEO should be connected to the IEI pin of the next highest priority device. This pattern continues with the next highest priority peripheral, until the peripherals are all connected, as shown in Figure 1.

$\overline{INT}$. The interrupt request pins for each peripheral in the daisy chain can be wire-ORed and connected to the 68000's ILP_n pins. The 68000 has seven interrupt levels that can be encoded into the ILP_0, ILP_1, and ILP_2 pins. Multiple 68000 interrupt levels can be implemented by using a multiplexer like the 74LS148.

Table 1. Z8500 and 68000 Pin Functions

68000 Signals		Z8500 Signals	
Mnemonic	**Function**	**Mnemonic**	**Function**
A_1-A_{23}	Address bus	A_0,A_1,A/$\overline{B}$,D/$\overline{C}$*	Register select
$\overline{AS}$	Address Strobe	$\overline{CE}$	Chip Enable
CLK	68000 clock (8 MHz)	D_0-D_7	Data bus
D_0-D_{15}	Data bus	IEI,IEO	Interrupt daisy chain
$\overline{DTACK}$	Data Transfer Acknowledge		control
FC_0-FC_2	Processor status	$\overline{INT}$	Interrupt Request
ILP_0-ILP_2	Interrupt request	$\overline{INTACK}$	Interrupt Acknowledge
R/$\overline{W}$	Read/Write	PCLK	Peripheral Clock
$\overline{VMA}$	Valid Memory Address	$\overline{RD}$	Read strobe
$\overline{VPA}$	Valid Peripheral Address	$\overline{WR}$	Write strobe

* The register select pins on each peripheral have different names.

$\overline{\text{INTACK}}$. The INTACK pin signals the peripheral that an Interrupt Acknowledge cycle is occurring. The following equation describes how $\overline{\text{INTACK}}$ is generated:

$$\overline{\text{INTACK}} = \overline{(FC_0)\cdot(FC_1)\cdot(FC_2)\cdot(AS)}$$

The 68000 FC_0-FC_2 are status pins that indicate an Interrupt Acknowledge when they are all High. They should be ANDed with inverted $\overline{\text{AS}}$ to guarantee their validity. The $\overline{\text{INTACK}}$ signal must be synchronized with PCLK to guarantee set-up and hold times. This can be accomplished by changing the state of $\overline{\text{INTACK}}$ on the falling edge of PCLK. If the $\overline{\text{INTACK}}$ pin is not used, it must be tied High.

PCLK. The SCC and CIO require a clock for internal synchronization. The clock can be generated by dividing down the 68000 CLK.

$\overline{\text{RD}}$. The Read strobe goes active Low under three conditions: hardware reset, normal Read cycle, and an Interrupt Acknowledge cycle. The following equation describes how $\overline{\text{RD}}$ is generated:

$$\overline{\text{RD}} = \overline{[(R/\overline{W})\cdot(AS) + \text{RESET}]}$$

The Read strobe timing must meet both the Read timing and Interrupt Acknowledge timing discussed in the following section. In addition to enabling the Data bus drivers, the falling edge of $\overline{\text{RD}}$ sets the Interrupt Under Service (IUS) bits during an Interrupt Acknowledge cycle.

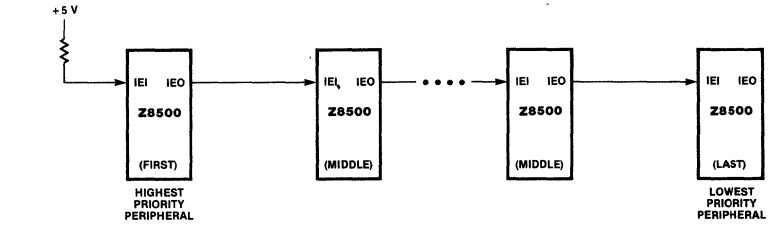
$\overline{\text{WR}}$. This signal strobes data into the peripheral. A data-to-write setup time requires that data be valid before $\overline{\text{WR}}$ goes active Low. The equation for generating the $\overline{\text{WR}}$ strobe is made up of two components: an active reset and a normal Write cycle, as shown in the following equation:

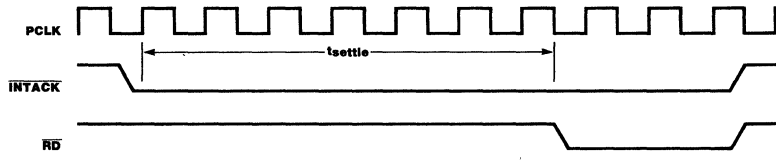
$$\overline{\text{WR}} = \overline{[(\overline{R/\overline{W}})\cdot(AS) + \text{RESET}]}$$

Forcing $\overline{\text{RD}}$ and $\overline{\text{WR}}$ simultaneously Low resets the peripherals.

Z8500 Timing Cycles

This section discusses the timing parameters that must be met when generating the control signals. The Z8500 family uses the control signals to communicate with the CPU via three types of bus cycle: Read, Write, and Interrupt Acknowledge.





t_{settle} (NS)

PERIPHERAL (4MHz)	FIRST	MIDDLE	LAST
CIO	350	150	100
FIO	350	150	100
SCC	250	120	120

Figure 1. Peripheral Interrupt Daisy Chain

The discussion that follows pertains to the 4 MHz peripherals, but the 6 MHz devices have similar timing considerations.

Although the peripherals have a standard CPU interface, some of their particular timing requirements vary. The worst-case parameters are shown below; the timing can be optimized if only one or two of the Z8500 family devices are used.

Read Cycle

The Read cycle transfers data from the peripheral to the CPU. It begins by selecting the peripheral and appropriate register (Data or Control). The data is gated onto the bus with the $\overline{RD}$ line. A setup time of 80 ns from the time the register select inputs ($A/\overline{B}$, $C/\overline{D}$, A_0, A_1) are stable to the falling edge of $\overline{RD}$ guarantees that the proper register is accessed. The access time specification is usually measured from the falling edge of $\overline{RD}$ to valid data and varies between peripherals. The SCC specifies an additional register select to valid data time. The Read cycle timing is shown in Figure 2.

Write Cycle

The Write cycle transfers data from the CPU to the peripheral. It begins by selecting the peripheral and addressing the desired register. A setup time of 80 ns from register select stable to the falling edge of $\overline{WR}$ is required. The data must be valid prior to the falling edge of $\overline{WR}$. The $\overline{WR}$ pulse width is specified at 400 ns. Write cycle timing is shown in Figure 2.

Interrupt Acknowledge Cycle

The Z8500 peripheral interrupt structure offers the designer many options. In the simplest case, the Z8500 peripherals can be polled with interrupts disabled. If using interrupts, the timing shown in Figure 2 should be observed. (Detailed discussions of the interrupt processing can be found in the Zilog Data Book, document number 00-2034-02.) An interrupt sequence begins with an $\overline{INT}$ going active because of an interrupt condition. The CPU acknowledges the interrupt with an $\overline{INTACK}$ signal.

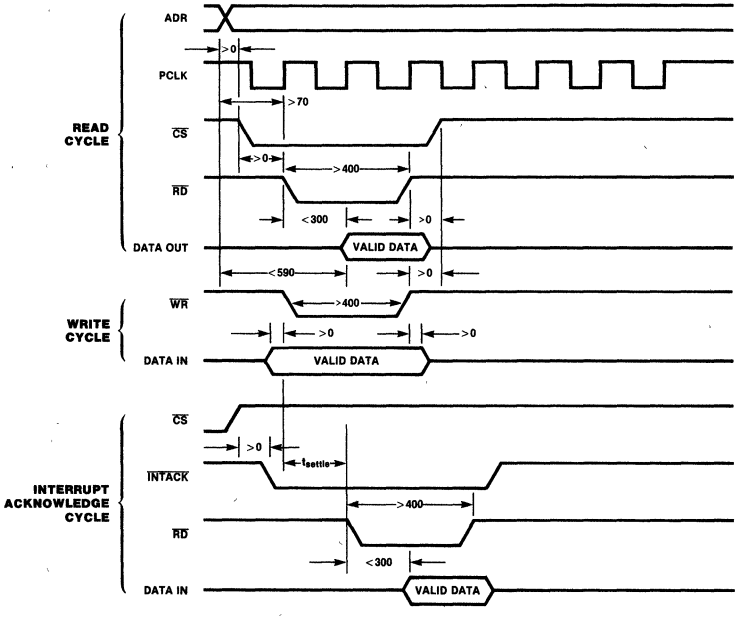


Figure 2. Z8500 Interface Timing (4 MHz)

A daisy-chain settle time (dependent upon the number of devices in the chain) ensures that the interrupts are prioritized. The falling edge of $\overline{RD}$ causes the IUS bit to be set and enables a vector to go out on the bus.

The table given in Figure 1 can be used to calculate the amount of settling time required by a daisy chain. Even if there is only one peripheral in the chain, a minimum settling time is still required because of the internal daisy chain. The first column specifies the amount of settling time for only one peripheral. If there are two peripherals, the time is computed by adding together the times shown in the first and the last columns. For each additional peripheral in the chain, the time specified in the middle column is added.

Recovery Time

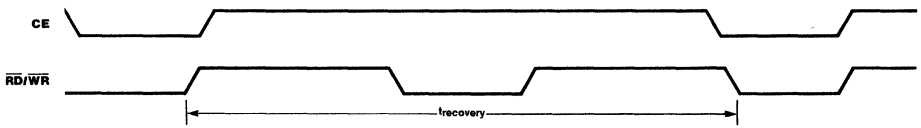
The read/write recovery time specifies a minimum amount of time between Read or Write cycles to the same peripheral. The recovery time differs among peripherals and is summarized in Figure 3. In most cases, this parameter is met because of the time required for instruction fetches. The recovery time specification does not have to be met if $\overline{CE}$ is deselected when Read or Write occurs.

68000 INTERFACE EXAMPLES

This section shows three examples, presented in increasing order of complexity, for interfacing Zilog's 4 MHz Z8500 peripherals to an 8 MHz 68000. Faster CPUs or peripherals can be used by modifying some of the timing. These examples suggest possible ways of implementing the interface but may require some modifications to operate properly. They were chosen because they give the user a variety of interface design ideas. The first example uses a minimum amount of TTL logic to implement the interface because the Valid Peripheral Address ($\overline{VPA}$) cycle meets the Z8500 timing requirements. In this mode the 68000 accepts only nonvectored interrupts. The second example uses the Data Transfer Acknowledge ($\overline{DTACK}$) pin. This interface allows faster operation and makes use of the Z8500's 8-bit vectored interrupts. The third example also uses a $\overline{DTACK}$ cycle and is similar to the second, except the external logic is integrated into a single chip, the PAL20X10 programmable array logic.

EXAMPLE 1: A TTL Interface Using a VPA Cycle

The 68000 has a special input pin, Valid Peripheral Address ($\overline{VPA}$), that can be activated by the Z8500 chip select logic at the beginning of the cycle to indicate to the 68000 that a peripheral is being accessed. This generates a special Read/Write cycle that meets the peripheral timing requirements. This cycle allows the Z8500 control signals to be generated easily. The 68000 responds to interrupts using an autovector and the Z8500 can be programmed not to return a vector.



Peripheral (4 MHz)	Recovery Time
CIO	Greater than 3 PCLK cycles or 1000ns
FIO	Greater than 1000ns
SCC	Greater than 4 PCLK cycles

NOTE. The diagram shows that the recovery time is measured between consecutive reads and writes only if the peripheral is selected

Figure 3. Recovery Time

Figure 4 shows how the hardware can be implemented. PCLK is generated by dividing down the 68000 CLK. $\overline{RD}$, $\overline{WR}$, and $\overline{INTACK}$ are simply ANDed 68000 signals. The worst-case daisy-chain settle time is 450 ns. Connecting $\overline{INT}$ to IPL_0 generates a level 1 interrupt. The internal registers are accessed by A_0, A_1, $D/\overline{C}$, and $A/\overline{B}$, which can be the 68000 lowest order addresses. The timing is shown in Figure 5.

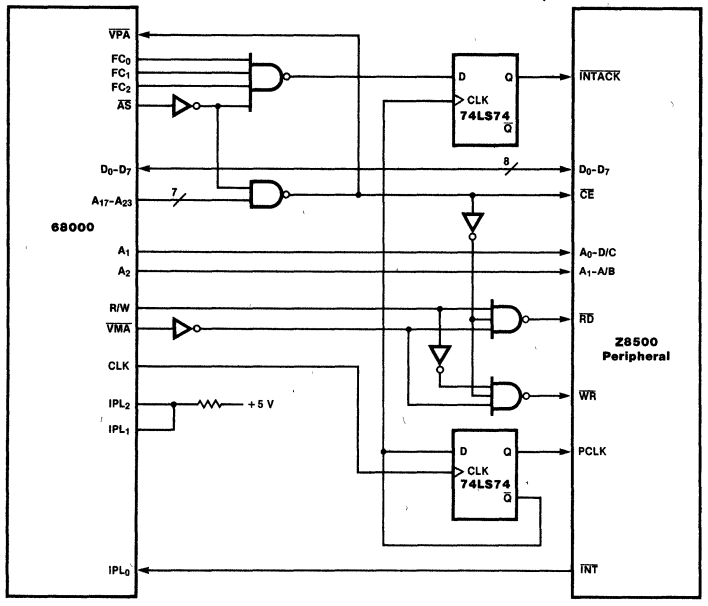


Figure 4. Interface Using the $\overline{VPA}$ Cycle

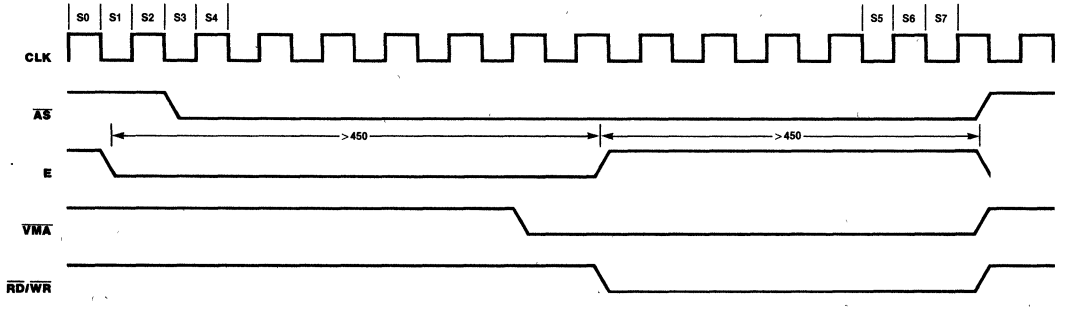


Figure 5. $\overline{VPA}$ Cycle Timing

Functional Description

$\overline{VPA}$ is pulled Low at the beginning of the cycle and the CPU automatically inserts Wait states until E is synchronized.

VPA = [(AS)•(CE)]

RD = [(CE)•(VMA)•(R/$\overline{W}$)]

WR = [(CE)•(VMA)•($\overline{R/W}$)]

INTACK = [(FC0)•(FC1)•(FC2)•(AS)]

EXAMPLE 2: A TTL Interface Using DTACK Cycles

Using the 68000 Data Transfer Acknowledge ($\overline{DTACK}$) cycle is a second way of interfacing to the Z8500 peripherals. The 68000 inserts Wait states until the $\overline{DTACK}$ input is strobed Low to complete the transfer. In addition to generating the control signals, the interface logic must also generate $\overline{DTACK}$.

The timing shown in Figure 6 can be generated by the hardware shown in Figure 7. The 8-bit Shift register (74LS164) is used to generate the proper timing. At the beginning of each cycle, Q_A (Figure 7) is set High for one PCLK cycle and then reset. This pulse is shifted through the Q_A-Q_H outputs and is used to generate $\overline{RD}$, $\overline{WR}$, and $\overline{DTACK}$ signals. Some of the extra Wait states can be eliminated by tapping the Shift register sooner (e.g., Q_C).

EXAMPLE 3: Single-Chip Pal Interface

This example illustrates how to interface the 4 MHz Z8500 peripherals to the 8 MHz 68000 using a PAL20X10 device to generate all the required control signals. The PAL reduces the required interface logic to a single chip, thus minimizing board space. This interface offers flexibility because the internal logic can be reprogrammed without changing the pin functions. The PAL uses 68000 signals to generate Read, Write, and Interrupt Acknowledge cycles. In addition to generating the Z8500 control signals, the PAL also generates a $\overline{DTACK}$ to inform the 68000 of a completed data transfer cycle. This allows the 68000 to use the peripheral's vectored interrupts.

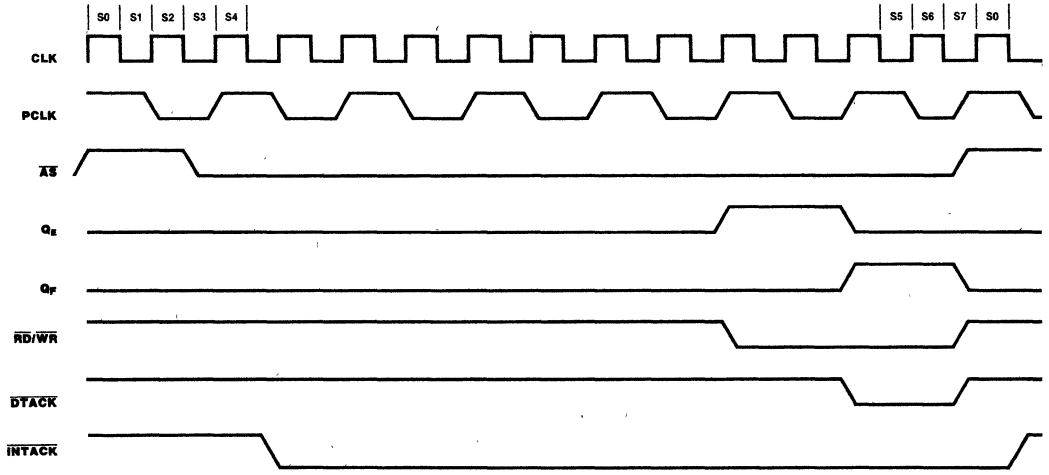


Figure 6. Timing for $\overline{DTACK}$ Interface

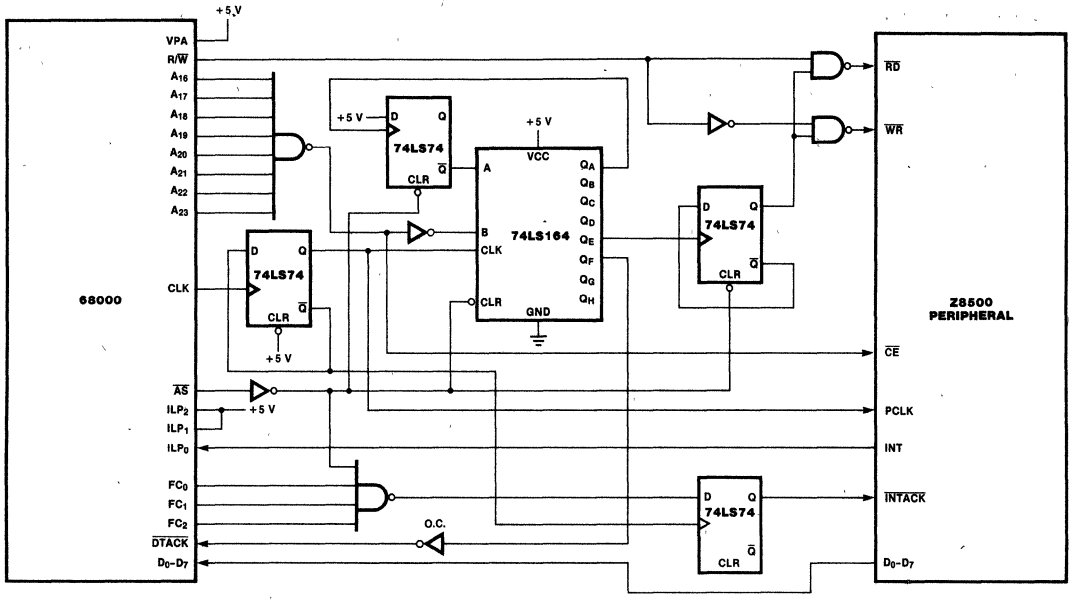


Figure 7. Hardware Diagram for $\overline{\text{DTACK}}$ Interface

Functional Description

Figure 8 shows the PAL's pin functions. The PAL generates five control signals, of which four ($\overline{WR}$, $\overline{RD}$, C_0, and $\overline{INTACK}$) go to the Z8500 and one ($\overline{DTACK}$) goes to the 68000. The remaining signals are used internally to generate these outputs.

Timing diagrams for the Read, Write, and Interrupt Acknowledge cycles are shown in Figure 9.

The PAL uses a 4-bit downcounter to generate the proper placement of the control signals where C_0 is the least-significant bit and C_3 is the

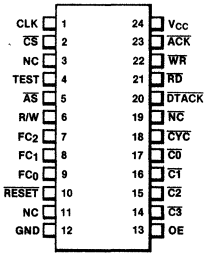


Figure 8. PAL Pinout

most-significant bit. All of the PAL is clocked with the rising edge of the 68000's CLK. The counter toggles between counts 14 and 15 and starts counting down when $\overline{AS}$ goes active. The counter goes back to toggling when $\overline{AS}$ goes inactive. CYC goes active Low at the same time the counter starts counting down. The equations in Figure 10 can be entered into a development board to program the PAL.

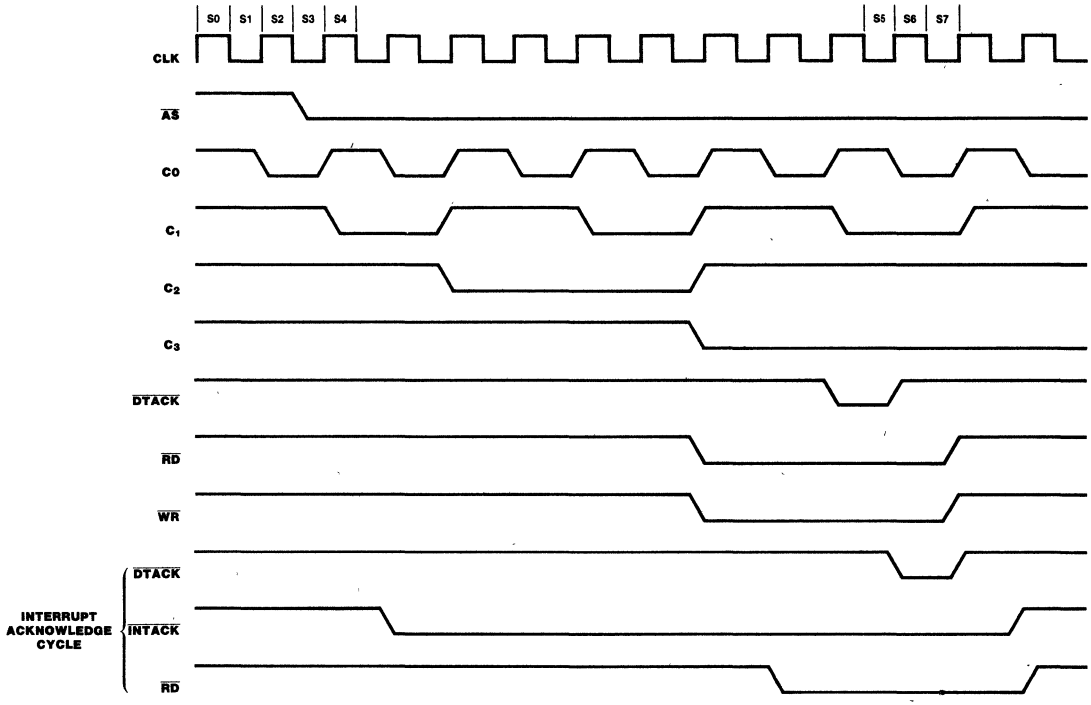


Figure 9. PAL Interface Timing

```
PAL20X10                                              PAL DESIGN SPECIFICATION
P7089 (10)
MC68000 TO ZILOG PERIPHERAL INTERFACE
MMI, SUNNYVALE, CA
CLK /CS  NC TEST /AS RW
FC2 FC1 FC0 /RESET NC GND
/OE /C3 /C2 /C1 /C0 /CYC
NC /DTK /RD /WR /ACK VCC

C0  :=     /C0*/TEST                              ; COUNT/HOLD (LSB)

C1  :=     /RESET*AS*C1                           ; HOLD
    :+:    /RESET*AS*C0                           ; DECREMENT

C2  :=     /RESET*AS*C2                           ; HOLD
    :+:    /RESET*AS*C0*C1                        ; DECREMENT

C3  :=     /RESET*AS*C3                           ; HOLD
    :+:    /RESET*AS*C0*C1*C2                     ; DECREMENT

DTK :=     /RESET*/ACK*CYC*C3*/C2*/C1* C0*CS      ; DTACK FOR RD/WR CYCLE
     +     /RESET* ACK*CYC*C3*/C2* C1*/C0         ; DTACK FOR INTERRUPT
                                                  ; OPERATION

CYC :=     /RESET*AS*/CYC*C0                      ; NEW CYCLE STARTED
     +     /RESET*AS* CYC                         ; PROCESSING OF CYCLE
    :+:    /RESET*CYC*DTK                         ; END OF CYCLE

RD  :=     /RESET*CYC*/ACK*RW* C3*/C2*CS          ; NORMAL READ OPERATION
     +     /RESET*CYC*/ACK*RW*/C3*C2*C1*C0*CS     ; NORMAL READ OPERATION
    :+:    /RESET*CYC* ACK*RW* C3                 ; READ DURING OPERATION
     +      RESET

WR  :=     /RESET*CYC*/ACK*/RW* C3*/C2*CS         ; WRITE
     +     /RESET*CYC*/ACK*/RW*/C3* C2*C1*C0*CS   ; WRITE
    :+:     RESET

ACK :=     /RESET*FC0*FC1*FC2*AS* CYC*/C0         ; INTERRUPT ACKNOWLEDGE
     +     /RESET*FC0*FC1*FC2*CYC                 ; INTERRUPT ACKNOWLEDGE
```

Figure 10. PAL Equations

Hardware Diagram

The hardware diagram of the PAL interface is shown in Figure 11. The 68000 signals CLK, $\overline{CS}$, $\overline{AS}$, R/$\overline{W}$, FC_0, FC_1, and FC_2 are used to generate the Z8500 control signals. The control signals are synchronous with the rising edge of the 68000's CLK. TEST and $\overline{OE}$ must be grounded. $\overline{CS}$ is used to enable $\overline{DTACK}$, $\overline{RD}$, and $\overline{WR}$ as shown in the equations. The Z8500 $\overline{INT}$ is connected to $\overline{ILP}_0$, which generates a 68000 level 1 interrupt. The peripherals are memory-mapped into the highest 64K byte block of memory, where A_{17}-A_{23} equals "FF_H". Addresses A_4-A_6 are used to select the peripheral; A_1-A_3 select the internal registers. Table 2 shows the peripheral's memory map.

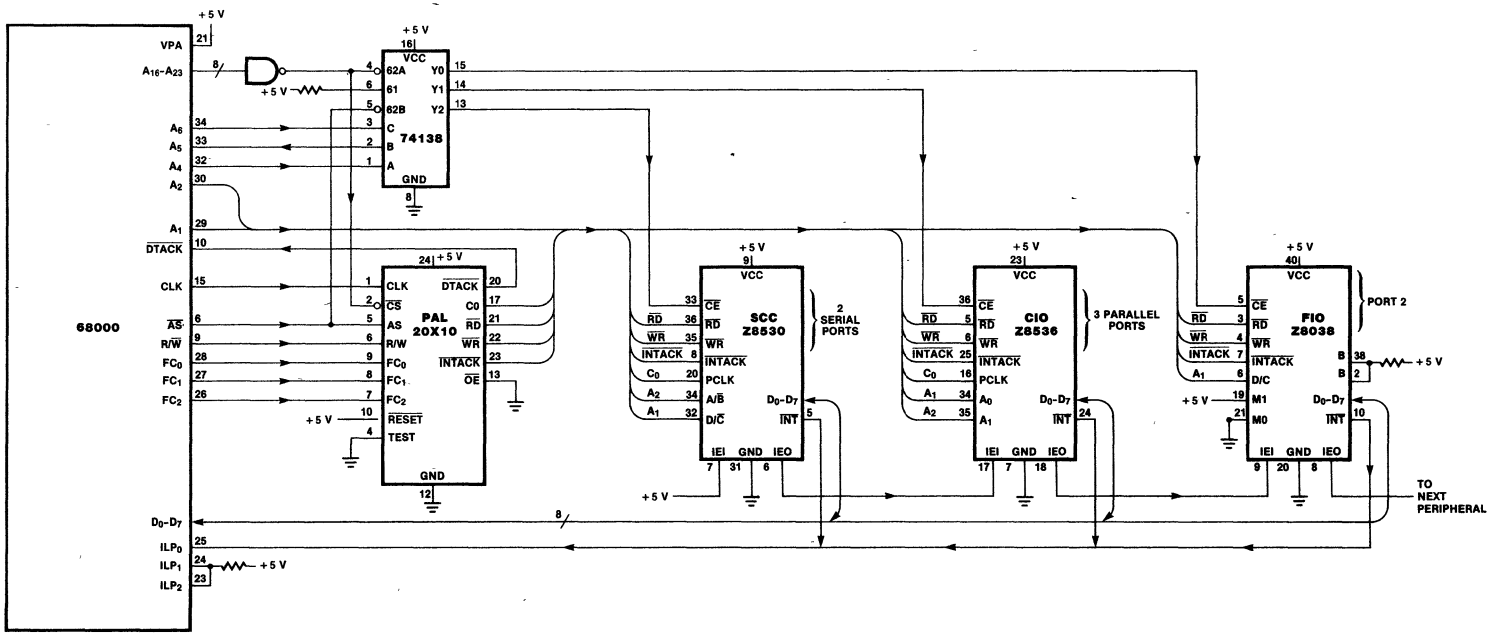


Figure 11. PAL Hardware Diagram

Table 2. Peripheral Memory Map

Peripheral	Register	Hex Address
SCC (Z8530)		
	Channel B Control	FF0020
	Channel B Data	FF0022
	Channel A Control	FF0024
	Channel B Data	FF0026
CIO (Z8536)		
	Port C's Data Register	FF0010
	Port B's Data Register	FF0012
	Port A's Data Register	FF0014
	Control Register	FF0016
FIO (Z8038)		
	Data Registers	FF0000
	Control Registers	FF0002

INTERFACE VERIFICATION TECHNIQUES

This section suggests possible ways of verifying the Read, Write, and Interrupt Acknowledge cycles.

Read Cycle Verification

The Read cycle should be checked first because it is the simplest operation. The Z8500 should be hardware reset by simultaneously pulling $\overline{RD}$ and $\overline{WR}$ Low. When the peripheral is in the reset state, the Control register containing the reset bit can be read without writing the pointer. Reading back the FIO or CIO Control register should yield a 01_H.

The SCC's Read cycle can be verified by reading the bits in RR0. Bits D_2 and D_6 are set to 1 and bits D_0, D_1, and D_7 are 0. Bits D_3-D_5 reflect the input pins DCD, SYNC, and CTS, respectively.

Write Cycle Verification

The Write cycle can be checked by writing to a register and reading back the results. Both the CIO and FIO must have their reset bits cleared by writing 00_H to their Control registers and reading back the result. The SCC can be checked by writing and reading to an arbitrary read/write register, for example, the Time Constant register (WR12 or WR13).

Interrupt Acknowledge Cycle Verification

Verifying an Interrupt Acknowledge ($\overline{INTACK}$) cycle consists of several steps. First, the peripheral makes an Interrupt Request ($\overline{INT}$) to the CPU. When the processor is ready to service the interrupt, it initiates an Interrupt Acknowledge ($\overline{INTACK}$) cycle. The peripheral then puts an 8-bit vector on the bus, and the 68000 uses that vector to get to the correct service routine. This test checks the simplest case.

First, load the Interrupt Vector register with a vector, disable the Vector Includes Status (VIS), and enable interrupts (IE = 1, MIE = 1, IEI = 1). Disabling VIS guarantees that only one vector is put on the bus. The address of the service routine corresponding to the 8-bit vector number must be loaded into the 68000's vector table.

Initiating an interrupt sequence in the FIO and CIO can be accomplished by setting one of the interrupt pending (IP) bits and seeing if the 68000 jumps to the service routine (setting a breakpoint at the beginning of the service routine is an easy way to check if this has happened).

Initiating an interrupt sequence in the SCC is not quite as simple because the IP bits are not as accessible to the user. An interrupt can be generated indirectly via the CTS pin by enabling the following: CTS IE (WR15 20), EXT INT EN (WR1 01), and MIE (WR9 08). Any transition on the CTS pin can initiate the interrupt sequence. The interrupt can be re-enabled by RESET EXT/STATUS INT (WR0 10) and RESET HIGHEST IUS (WR0 38).

CONCLUSION

Zilog's Z8500 family of nonmultiplexed Address/Data bus peripherals can interface easily with the 68000 and provide all the support required in a high-performance microprocessor system. The many features offered by the SCC, FIO, and CIO solve many system design problems by making interfacing to the external world easy. These intelligent peripherals also greatly enhance the system performance by relieving the CPU of many burdensome overhead tasks. Additionally, the powerful interrupt structure allows the 68000 to use vectors and reduce interrupt response time.

ZiLOG

APPLICATION NOTE

INTERFACING Z80 CPUs TO THE Z8500 PERIPHERAL FAMILY

INTRODUCTION

The Z8500 Family consists of universal peripherals that can interface to a variety of microprocessor systems that use a non-multiplexed address and data bus. Though similar to Z80 peripherals, the Z8500 peripherals differ in the way they respond to I/O and Interrupt Acknowledge cycles. In addition, the advanced features of the Z8500 peripherals enhance system performance and reduce processor overhead.

To design an effective interface, the user needs an understanding of how the Z80 Family interrupt structure works, and how the Z8500 peripherals interact with this structure. This application note provides basic information on the interrupt structures, as well as a discussion of the hardware and software considerations involved in interfacing the Z8500 peripherals to the Z80 CPUs. Discussions center around each of the following situations:

- Z80A 4 MHz CPU to Z8500 4 MHz peripherals
- Z80B 6 MHz CPU to Z8500A 6 MHz peripherals
- Z80H 8 MHz CPU to Z8500 4 MHz peripherals
- Z80H 8 MHz CPU to Z8500A 6 MHz peripherals

This application note assumes the reader has a strong working knowledge of the Z8500 peripherals; it is not intended as a tutorial.

CPU HARDWARE INTERFACING

The hardware interface consists of three basic groups of signals: data bus, system control, and interrupt control, described below. For more detailed signal information, refer to Zilog's Data Book, Universal Peripherals.

Data Bus Signals

D_7-D_0 Data Bus (bidirectional, 3-state). This bus transfers data between the CPU and the peripherals.

System Control Signals

A_n-A_0 Address Select Lines (optional). These lines select the port and/or control registers.

$\overline{CE}$ Chip Enable (input, active Low). $\overline{CE}$ is used to select the proper peripheral for programming. $\overline{CE}$ should be gated with $\overline{IORQ}$ or $\overline{MREQ}$ to prevent spurious chip selects during other machine cycles.

$\overline{RD}$* Read (input, active Low). $\overline{RD}$ activates the chip-read circuitry and gates data from the chip onto the data bus.

$\overline{WR}$* Write (input, active Low). $\overline{WR}$ strobes data from the data bus into the peripheral.

*Chip reset occurs when $\overline{RD}$ and $\overline{WR}$ are active simultaneously.

Interrupt Control

$\overline{INTACK}$ Interrupt Acknowledge (input, active Low). This signal indicates an Interrupt Acknowledge cycle and is used with $\overline{RD}$ to gate the interrupt vector onto the data bus.

$\overline{INT}$ Interrupt Request (output, open-drain, active Low).

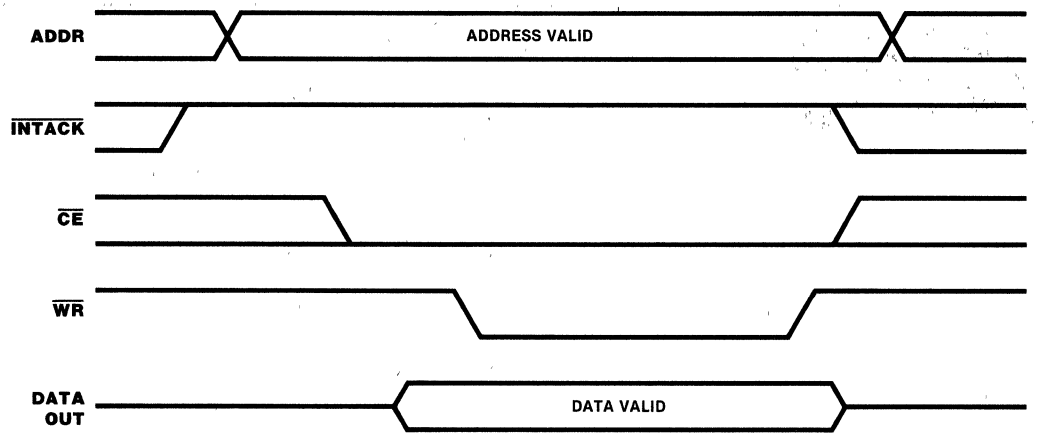


Figure 2. Z8500 Peripheral I/O Write Cycle Timing

The IUS bit indicates that an interrupt is currently being serviced by the CPU. The IUS bit is set during an Interrupt Acknowledge cycle if the IP bit is set and the IEI line is High. If the IEI line is Low, the IUS bit is not set, and the device is inhibited from placing its vector onto the data bus. In the Z80 peripherals, the IUS bit is normally cleared by decoding the RETI instruction, but can also be cleared by a software command (SIO). In the Z8500 peripherals, the IUS bit is cleared only by software commands.

Z80 Interrupt Daisy-Chain Operation

In the Z80 peripherals, both the IP and IUS bits control the IEO line and the lower portion of the daisy chain.

When a peripheral's IP bit is set, its IEO line is forced Low. This is true regardless of the state of the IEI line. Additionally, if the peripheral's IUS bit is clear and its IEI line High, the $\overline{INT}$ line is also forced Low.

The Z80 peripherals sample for both $\overline{M1}$ and $\overline{IORQ}$ active, and $\overline{RD}$ inactive to identify an Interrupt Acknowledge cycle. When $\overline{M1}$ goes active and $\overline{RD}$ is inactive, the peripheral detects an Interrupt Acknowledge cycle and allows its interrupt daisy chain to settle. When the $\overline{IORQ}$ line goes active with $\overline{M1}$ active, the highest priority interrupting peripheral places its interrupt vector onto the data bus. The IUS bit is also set to indicate that the peripheral is currently under service. As long as the IUS bit is set, the IEO line is forced Low. This inhibits any lower priority devices from requesting an interrupt.

When the Z80 CPU executes the RETI instruction, the peripherals monitor the data bus and the highest priority device under service resets its IUS bit.

Z8500 Interrupt Daisy-Chain Operation

In the Z8500 peripherals, the IUS bit normally controls the state of the IEO line. The IP bit affects the daisy chain only during an Interrupt Acknowledge cycle. Since the IP bit is normally not part of the Z8500 peripheral interrupt daisy chain, there is no need to decode the RETI instruction. To allow for control over the daisy chain, Z8500 peripherals have a Disable Lower Chain (DLC) software command that pulls IEO Low. This can be used to selectively deactivate parts of the daisy chain regardless of the interrupt status. Table 1 shows the truth tables for the Z8500 interrupt daisy-chain control signals during certain cycles. Table 2 shows the interrupt state diagram for the Z8500 peripherals.

Table 1. Z8500 Daisy-Chain Control Signals

Truth Table for Daisy Chain Signals During Idle State				Truth Table for Daisy Chain Signals During $\overline{INTACK}$ Cycle			
IEI	IP	IUS	IEO	IEI	IP	IUS	IEO
0	X	X	0	0	X	X	0
1	X	0	1	1	1	X	0
1	X	1	0	1	X	1	0
				1	0	0	1

IEI Interrupt Enable In (input, active High).

IEO Interrupt Enable Out (output, active High).

These lines control the interrupt daisy chain for the peripheral interrupt response.

Z8500 I/O OPERATION

The Z8500 peripherals generate internal control signals from $\overline{RD}$ and $\overline{WR}$. Since PCLK has no required phase relationship to $\overline{RD}$ or $\overline{WR}$, the circuitry generating these signals provides time for metastable conditions to disappear.

The Z8500 peripherals are initialized for different operating modes by programming the internal registers. These internal registers are accessed during I/O Read and Write cycles, which are described below.

Read Cycle Timing

Figure 1 illustrates the Z8500 Read cycle timing. All register addresses and $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ goes active after $\overline{RD}$ goes active, or if $\overline{CE}$ goes inactive before $\overline{RD}$ goes inactive, then the effective Read cycle is shortened.

Write Cycle Timing

Figure 2 illustrates the Z8500 Write cycle timing. All register addresses and $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ goes active after $\overline{WR}$ goes active, or if $\overline{CE}$ goes inactive before $\overline{WR}$ goes inactive, then the effective Write cycle is shortened. Data must be available to the peripheral prior to the falling edge of $\overline{WR}$.

PERIPHERAL INTERRUPT OPERATION

Understanding peripheral interrupt operation requires a basic knowledge of the Interrupt Pending (IP) and Interrupt Under Service (IUS) bits in relation to the daisy chain. Both Z80 and Z8500 peripherals are designed in such a way that no additional interrupts can be requested during an Interrupt Acknowledge cycle. This allows the interrupt daisy chain to settle, and ensures proper response of the interrupting device.

The IP bit is set in the peripheral when CPU intervention is required (such conditions as buffer empty, character available, error detection, or status changes). The Interrupt Acknowledge cycle does not necessarily reset the IP bit. This bit is cleared by a software command to the peripheral, or when the action that generated the interrupt is completed (i.e., reading a character, writing data, resetting errors, or changing the status). When the interrupt has been serviced, other interrupts can occur.

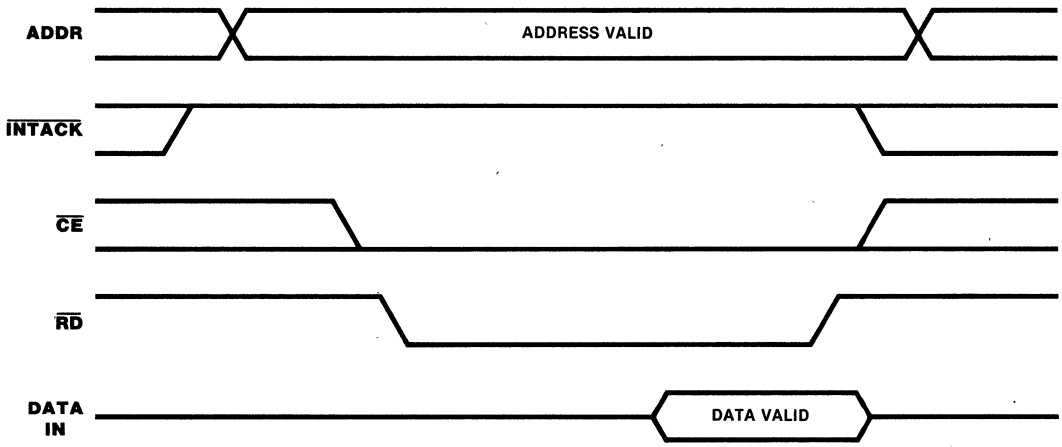
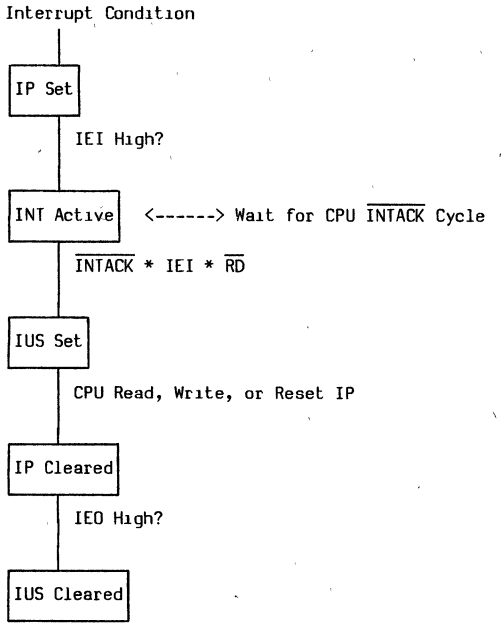


Figure 1. Z8500 Peripheral I/O Read Cycle Timing

Table 2. Z8500 Interrupt State Diagram



The Z8500 peripherals use INTACK (Interrupt Acknowledge) for recognition of an Interrupt Acknowledge cycle. This pin, used in conjunction with RD, allows the Z8500 peripheral to gate its interrupt vector onto the data bus. An active RD signal during an Interrupt Acknowledge cycle performs two functions. First, it allows the highest priority device requesting an interrupt to place its interrupt vector on the data bus. Secondly, it sets the IUS bit in the highest priority device to indicate that the device is currently under service.

INPUT/OUTPUT CYCLES

Although Z8500 peripherals are designed to be as universal as possible, certain timing parameters differ from the standard Z80 timing. The following sections discuss the I/O interface for each of the Z80 CPUs and the Z8500 peripherals. Figure 5 depicts logic for the Z80A CPU to Z8500 peripherals (and Z80B CPU to Z8500A peripherals) I/O interface as well as the Interrupt Acknowledge interface. Figures 4 and 7 depict some of the logic used to interface the Z80H CPU to the Z8500 and Z8500A peripherals for the I/O and Interrupt Acknowledge interfaces. The logic required for adding additional Wait states into the timing flow is not discussed in the folowing sections.

Z80A CPU to Z8500 Peripherals

No additional Wait states are necessary during the I/O cycles, although additional Wait states can be inserted to compensate for timing delays that are inherent in a system. Although the Z80A timing parameters indicate a negative value for data valid prior to WR, this is a worse than "worst case" value. This parameter is based upon the longest (worst case) delay for data available from the falling edge of the CPU clock minus the shortest (best case) delay for CPU clock High to WR Low. The negative value resulting from these two parameters does not occur because the worst case of one parameter and the best case of the other do not occur within the same device. This indicates that the value for data available prior to WR will always be greater than zero.

All setup and pulse width times for the Z8500 peripherals are met by the standard Z80A timing. In determining the interface necessary, the CE signal to the Z8500 peripherals is assumed to be the decoded address qualified with the IORQ signal.

Figure 3a shows the minimum Z80A CPU to Z8500 peripheral interface timing for I/O cycles. If additional Wait states are needed, the same number of Wait states can be inserted for both I/O Read and Write cycles to simplify interface logic. There are several ways to place the Z80A CPU into a Wait condition (such as counters or shift registers to count system clock pulses), depending upon whether or not the user wants to place Wait states in all I/O cycles, or only during Z8500 I/O cycles. Tables 3 and 4 list the Z8500 peripheral and the Z80A CPU timing parameters (respectively) of concern during the I/O cycles. Tables 5 and 6 list the equations used in determining if these parameters are satisfied. In generating these equations and the values obtained from them, the required number of Wait states was taken into account. The reference numbers in Tables 3 and 4 refer to the timing diagram in Figure 3a.

Table 3. Z8500 Timing Parameters I/O Cycles

	Worst Case		Min	Max	Units
6.	TsA(WR)	Address to $\overline{WR}$ Low Setup	80		ns
1.	TsA(RD)	Address to $\overline{RD}$ Low Setup	80		ns
2.	TdA(DR)	Address to Read Data Valid		590	ns
	TsCE1(WR)	$\overline{CE}$ Low to $\overline{WR}$ Low Setup	0		ns
	TsCE1(RD)	$\overline{CE}$ Low to $\overline{RD}$ Low Setup	0		ns
4.	TwRD1	$\overline{RD}$ Low Width	390		ns
8.	TwWR1	$\overline{WR}$ Low Width	390		ns
3.	TdRDf(DR)	$\overline{RD}$ Low to Read Data Valid		255	ns
7.	TsDW(WR)	Write Data to $\overline{WR}$ Low Setup	0		ns

Table 4. Z80A Timing Parameters I/O Cycles

	Worst Case		Min	Max	Units
	TcC	Clock Cycle Period	250		ns
	TwCh	Clock Cycle High Width	110		ns
	TfC	Clock Cycle Fall Time		30	ns
	TdCr(A)	Clock High to Address Valid		110	ns
	TdCr(RDf)	Clock High to $\overline{RD}$ Low		85	ns
	TdCr(IORQf)	Clock High to $\overline{IORQ}$ Low		75	ns
	TdCr(WRf)	Clock High to $\overline{WR}$ Low		65	ns
5.	TsD(Cf)	Data to Clock Low Setup	50		ns

Table 5. Parameter Equations

Z8500 Parameter	Z80A Equation	Value	Units
TsA(RD)	TcC-TdCr(A)	140 min	ns
TdA(DR)	3TcC+TwCh-TdCr(A)-TsD(Cf)	800 min	ns
TdRDf(DR)	2TcC+TwCh-TsD(Cf)	460 min	ns
TwRD1	2TcC+TwCh+TfC-TdCr(RDf)	525 min	ns
TsA(WR)	TcC-TdCr(A)	140 min	ns
TsDW(WR)		> 0 min	ns
TwWR1	2TcC+TwCh+TfC-TdCr(WRf)	560 min	ns

Table 6. Parameter Equations

Z80A Parameter	Z8500 Equation	Value	Units
TsD(Cf)	Address		
	3TcC+TwCh-TdCr(A)-TdA(DR)	160 min	ns
	$\overline{RD}$		
	2TcC+TwCh-TdCr(RDf)-TdRD(DR)	135 min	ns

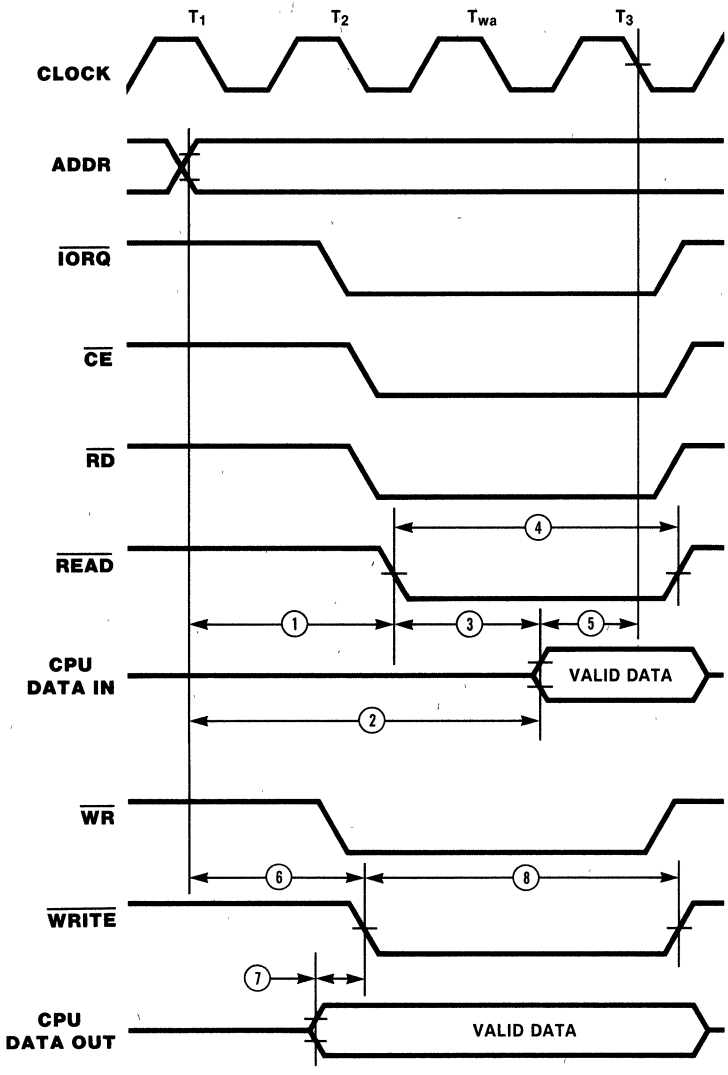


Figure 3a. Z80A CPU to Z8500 Peripheral Minimum I/O Cycle Timing

Z80B CPU to Z8500A Peripherals

No additional Wait states are necessary during I/O cycles, although Wait states can be inserted to compensate for any system delays. Although the Z80B timing parameters indicate a negative value for data valid prior to $\overline{WR}$, this is a worse than "worst case" value. This parameter is based upon the longest (worst case) delay for data available from the falling edge of the CPU clock minus the shortest (best case) delay for CPU clock High to $\overline{WR}$ Low. The negative value resulting from these two parameters does not occur because the worst case of one parameter and the best case of the other do not occur within the same device. This indicates that the value for data available prior to $\overline{WR}$ will always be greater than zero.

All setup and pulse width times for the Z8500A peripherals are met by the standard Z80B timing. In determining the interface necessary, the $\overline{CE}$ signal to the Z8500A peripherals is assumed to be the decoded address qualified with the $\overline{IORQ}$ signal.

Figure 3b shows the minimum Z80B CPU to Z8500A peripheral interface timing for I/O cycles. If additional Wait states are needed, the same number of Wait states can be inserted for both I/O Read and I/O Write cycles in order to simplify interface logic. There are several ways to place the Z80B CPU into a Wait condition (such as counters or shift registers to count system clock pulses), depending upon whether or not the user wants to place Wait states in all I/O cycles, or only during Z8500A I/O cycles. Tables 7 and 8 list the Z8500A peripheral and the Z80B CPU timing parameters (respectively) of concern during the I/O cycles. Tables 9 and 10 list the equations used in determining if these parameters are satisfied. In generating these equations and the values obtained from them, the required number of Wait states was taken into account. The reference numbers in Tables 7 and 8 refer to the timing diagram of Figure 3b.

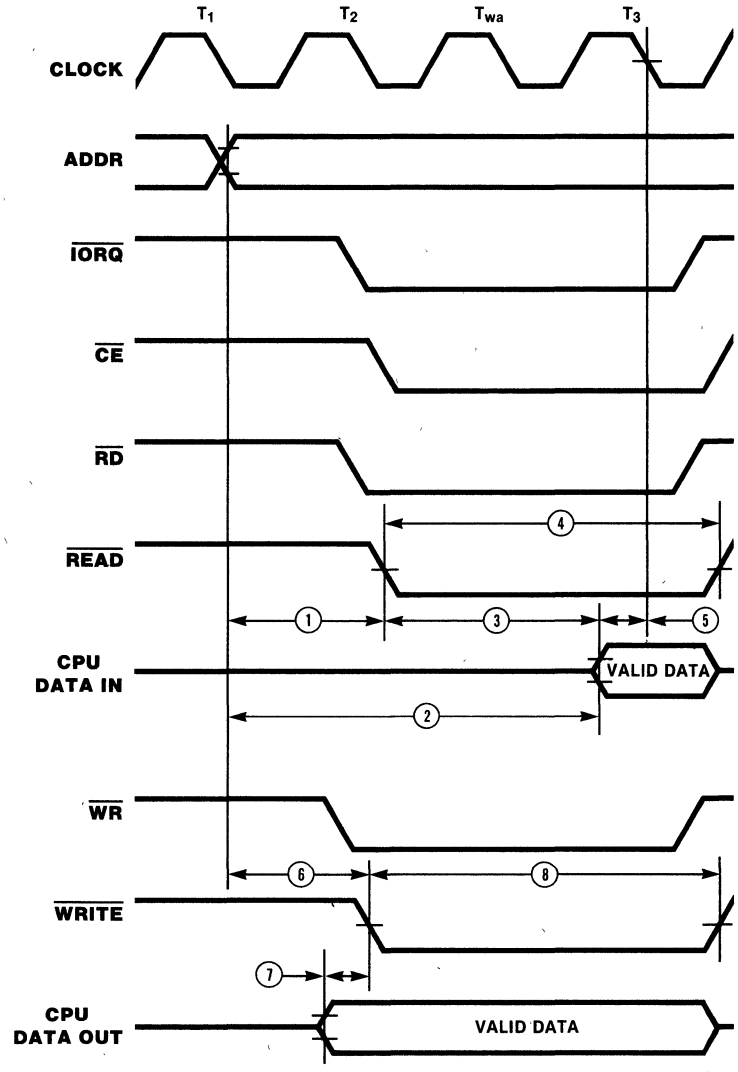


Figure 3b. Z80B CPU to Z8500A Peripheral Minimum I/O Cycle Timing

Table 7. Z8500A Timing Parameters I/O Cycles

	Worst Case		Min	Max	Units
6.	TsA(WR)	Address to $\overline{WR}$ Low Setup	80		ns
1.	TsA(RD)	Address to $\overline{RD}$ Low Setup	80		ns
2.	TdA(DR)	Address to Read Data Valid		420	ns
	TsCE1(WR)	$\overline{CE}$ Low to $\overline{WR}$ Low Setup	0		ns
	TsCE1(RD)	$\overline{CE}$ Low to $\overline{RD}$ Low Setup	0		ns
4.	TwRD1	$\overline{RD}$ Low Width	250		ns
8.	TwWR1	$\overline{WR}$ Low Width	250		ns
3.	TdRDf(DR)	$\overline{RD}$ Low to Read Data Valid		180	ns
7.	TsDW(WR)	Write Data to $\overline{WR}$ Low Setup	0		ns

Table 8. Z80B Timing Parameters I/O Cycles

	Worst Case		Min	Max	Units
	TcC	Clock Cycle Period	165		ns
	TwCh	Clock Cycle High Width	65		ns
	TfC	Clock Cycle Fall Time		20	ns
	TdCr(A)	Clock High to Address Valid		90	ns
	TdCr(RDf)	Clock High to $\overline{RD}$ Low		70	ns
	TdCr(IORQf)	Clock High to $\overline{IORQ}$ Low		65	ns
	TdCr(WRf)	Clock High to $\overline{WR}$ Low		60	ns
5.	TsD(Cf)	Data to Clock Low Setup	40		ns

Table 9. Parameter Equations

Z8500A Parameter	Z80B Equation	Value	Units
TsA(RD)	TcC-TdCr(A)	>75 min	ns
TdA(DR)	3TcC+TwCh-TdCr(A)-TsD(Cf)	430 min	ns
TdRDf(DR)	2TcC+TwCh-TsD(Cf)	345 min	ns
TwRD1	2TcC+TwCh+TfC-TdCr(RDf)	325 min	ns
TsA(WR)	TcC-TdCr(A)	75 min	ns
TsDW(WR)		> 0 min	ns
TwWR1	2TcC+TwCh+TfC-TdCr(WRf)	352 min	ns

Table 10. Parameter Equations

Z80B Parameter	Z8500A Equation	Value	Units
TsD(Cf)	Address 3TcC+TwCh-TdCr(A)-TdA(DR) $\overline{RD}$ 2TcC+TwCh-TdCr(RDf)-TdRD(DR)	50 min 75 min	ns ns

Z80H CPU to Z8500 Peripherals

During an I/O Read cycle, there are three Z8500 parameters that must be satisfied. Depending upon the loading characteristics of the $\overline{RD}$ signal, the designer may need to delay the leading (falling) edge of $\overline{RD}$ to satisfy the Z8500 timing parameter TsA(RD) (Address Valid to $\overline{RD}$ Setup). Since Z80H timing parameters indicate that the $\overline{RD}$ signal may go Low after the falling edge of T_2, it is recommended that the rising edge of the system clock be used to delay $\overline{RD}$ (if necessary). The CPU must also be placed into a Wait condition long enough to satisfy TdA(DR) (Address Valid to Read Data Valid Delay) and TdRDf(DR) ($\overline{RD}$ Low to Read Data Valid Delay).

During an I/O Write cycle, there are three other Z8500 parameters that must be satisfied. Depending upon the loading characteristics of the $\overline{WR}$ signal and the data bus, the designer may need to delay the leading (falling) edge of $\overline{WR}$ to satisfy the Z8500 timing parameters TsA(WR) (Address Valid to $\overline{WR}$ Setup) and TsDW(WR) (Data Valid Prior to $\overline{WR}$ setup). Since Z80H timing parameters indicate that the $\overline{WR}$ signal may go Low after the falling edge of T_2, it is recommended that the rising edge of the system clock be used to delay $\overline{WR}$ (if necessary).. This delay will ensure that both parameters are satisfied. The CPU must also be placed into a Wait condition long enough to satisfy TwWRl ($\overline{WR}$ Low Pulse Width). Assuming that the $\overline{WR}$ signal is delayed, only two additional Wait states are needed during an I/O Write cycle when interfacing the Z80H CPU to the Z8500 peripherals.

To simplify the I/O interface, the designer can use the same number of Wait states for both I/O Read and I/O Write cycles. Figure 3c shows the minimum Z80H CPU to Z8500 peripheral interface timing for the I/O cycles (assuming that the same number of Wait states are used for both cycles and that both $\overline{RD}$ and $\overline{WR}$ need to be delayed). Figure 4 shows two circuits that can be used to delay the leading (falling) edge of either the $\overline{RD}$ or the $\overline{WR}$ signals. There are several ways to place the Z80A CPU into a Wait condition (such as counters or shift registers to count system clock pulses), depending upon whether or not the user wants to place Wait states in all I/O cycles, or only during Z8500 I/O cycles. Tables 4 and 11 list the Z8500 peripheral and the Z80H CPU timing parameters (respectively) of concern during the I/O cycles. Tables 14 and 15 list the equations used in determining if these parameters are satisfied. In generating these equations and the values obtained from them, the required number of Wait states was taken into account. The reference numbers in Tables 4 and 11 refer to the timing diagram of Figure 3c.

Table 11. Z80H Timing Parameter I/O Cycles

		Equation	Min	Max	Units
	TcC	Clock Cycle Period	125		ns
	TwCh	Clock Cycle High Width	55		ns
	TfC	Clock Cycle Fall Time		10	ns
	TdCr(A)	Clock High to Address Valid		80	ns
	TdCr(RDf)	Clock High to $\overline{RD}$ Low		60	ns
	TdCr(IORQf)	Clock High to $\overline{IORQ}$ Low		55	ns
	TdCr(WRf)	Clock High to $\overline{WR}$ Low		55	ns
5.	TsD(Cf)	Data to Clock Low Setup	30		ns

Table 12. Parameter Equations

Z8500 Parameter	Z80H Equation	Value	Units
TsA(RD)	2TcC-TdCr(A)	170 min	ns
TdA(DR)	6TcC+TwCh-TdCr(A)-TsD(Cf)	695 min	ns
TdRDf(DR)	4TcC+TwCh-TsD(Cf)	523 min	ns
TwRDl	4TcC+TwCh+TfC-TdCr(RDf)	503 min	ns
TsA(WR)	$\overline{WR}$ - delayed		
	2TcC-TdCr(A)	170 min	ns
TsDW(WR)		> 0 min	ns
TwWRl	4TcC+TwCh+TfC	563 min	ns

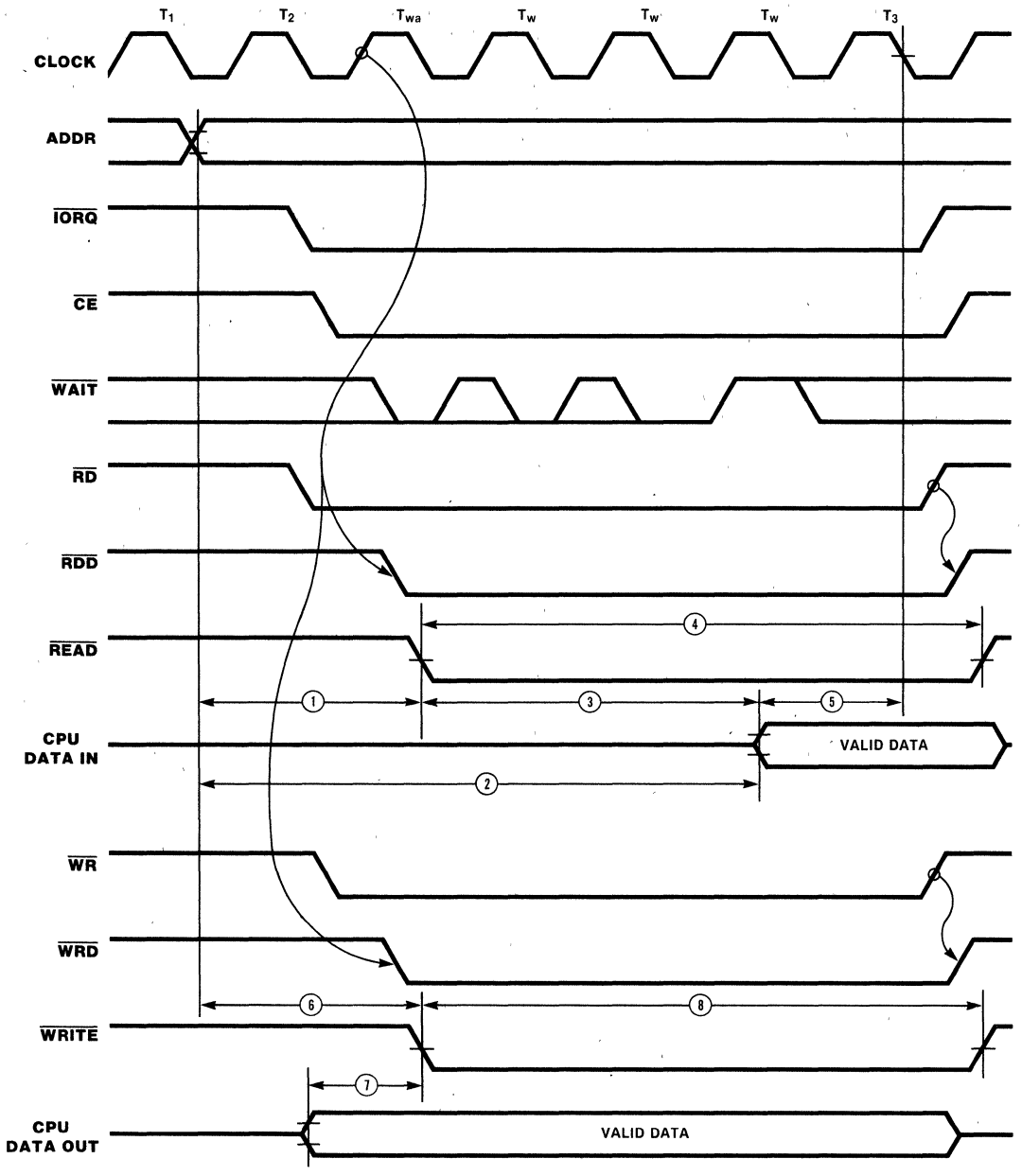


Figure 3c. Z80H CPU to Z8500 Peripheral Minimum I/O Cycle Timing

Z80H CPU to Z8500A Peripherals

During an I/O Read cycle, there are three Z8500A parameters that must be satisfied. Depending upon the loading characteristics of the $\overline{RD}$ signal, the designer may need to delay the leading (falling) edge of $\overline{RD}$ to satisfy the Z8500A timing parameter TsA(RD) (Address Valid to $\overline{RD}$ Setup). Since Z80H timing parameters indicate that the $\overline{RD}$ signal may go Low after the falling edge of T_2, it is recommended that the rising edge of the system clock be used to delay $\overline{RD}$ (if necessary). The CPU must also be placed into a Wait condition long enough to satisfy TdA(DR) (Address Valid to Read Data Valid Delay) and TdRDf(DR) ($\overline{RD}$ Low to Read Data Valid Delay). Assuming that the $\overline{RD}$ signal is delayed, then only one additional Wait state is needed during an I/O Read cycle when interfacing the Z80H CPU to the Z8500A peripherals.

During an I/O Write cycle, there are three other Z8500A parameters that have to be satisfied. Depending upon the loading characteristics of the $\overline{WR}$ signal and the data bus, the designer may need to delay the leading (falling) edge of $\overline{WR}$ to satisfy the Z8500A timing parameters TsA(WR) (Address Valid to $\overline{WR}$ Setup) and TsDW(WR) (Data Valid Prior to $\overline{WR}$ Setup). Since Z80H timing parameters indicate that the $\overline{WR}$ signal may go Low after the falling edge of T_2, it is recommended that the rising edge of the system clock be used to delay $\overline{WR}$ (if necessary). This delay will ensure that both parameters are satisfied. The CPU must also be placed into a Wait condition long enough to satisfy TwWR1 ($\overline{WR}$ Low Pulse Width). Assuming that the $\overline{WR}$ signal is delayed, then only one additional Wait state is needed during an I/O Write cycle when interfacing the Z80H CPU to the Z8500A peripherals.

Figure 3d shows the minimum Z80H CPU to Z8500A peripheral interface timing for the I/O cycles (assuming that the same number of Wait states are used for both cycles and that both $\overline{RD}$ and $\overline{WR}$ need to be delayed). Figure 4 shows two circuits that may be used to delay the leading (falling) edge of either the $\overline{RD}$ or the $\overline{WR}$ signals. There are several methods used to place the Z80A CPU into a Wait condition (such as counters or shift registers to count system clock pulses), depending upon whether or not the user wants to place Wait states in all I/O cycles, or only during Z8500A I/O cycles. Tables 7 and 11 list the Z8500A peripheral and the Z80H CPU timing parameters (respectively) of concern during the I/O cycles. Tables 14 and 15 list the equations used in determining if these parameters are satisfied. In generating these equations and the values obtained from them, the required number of Wait states was taken into account. The reference numbers in Tables 4 and 11 refer to the timing diagram of Figure 3d.

Table 13. Parameter Equations

Z80H Parameter	Z8500 Equation	Value	Units
TsD(Cf)	Address		
	6TcC+TwCh-TdCr(A)-TdA(DR)	135 min	ns
	$\overline{RD}$ - delayed		
	4TcC+TwCh+TfC-TdRD(DR)	300 min	ns

Table 14. Parameter Equations

Z8500A Parameter	Z80H Equation	Value	Units
TsA(RD)	2TcC-TdCr(A)	170 min	ns
TdA(DR)	6TcC+TwCh-TdCr(A)-TsD(Cf)	695 min	ns
TdRDf(DR)	4TcC+TwCh-TsD(Cf)	525 min	ns
TwRD1	4TcC+TwCh+TfC-TdCr(RDf)	503 min	ns
TsA(WR)	$\overline{WR}$ - delayed		
	2TcC-TdCr(A)	170 min	ns
TsDW(WR)		> 0 min	ns
TwWR1	2TcC+TwCh+TfC	313 min	ns

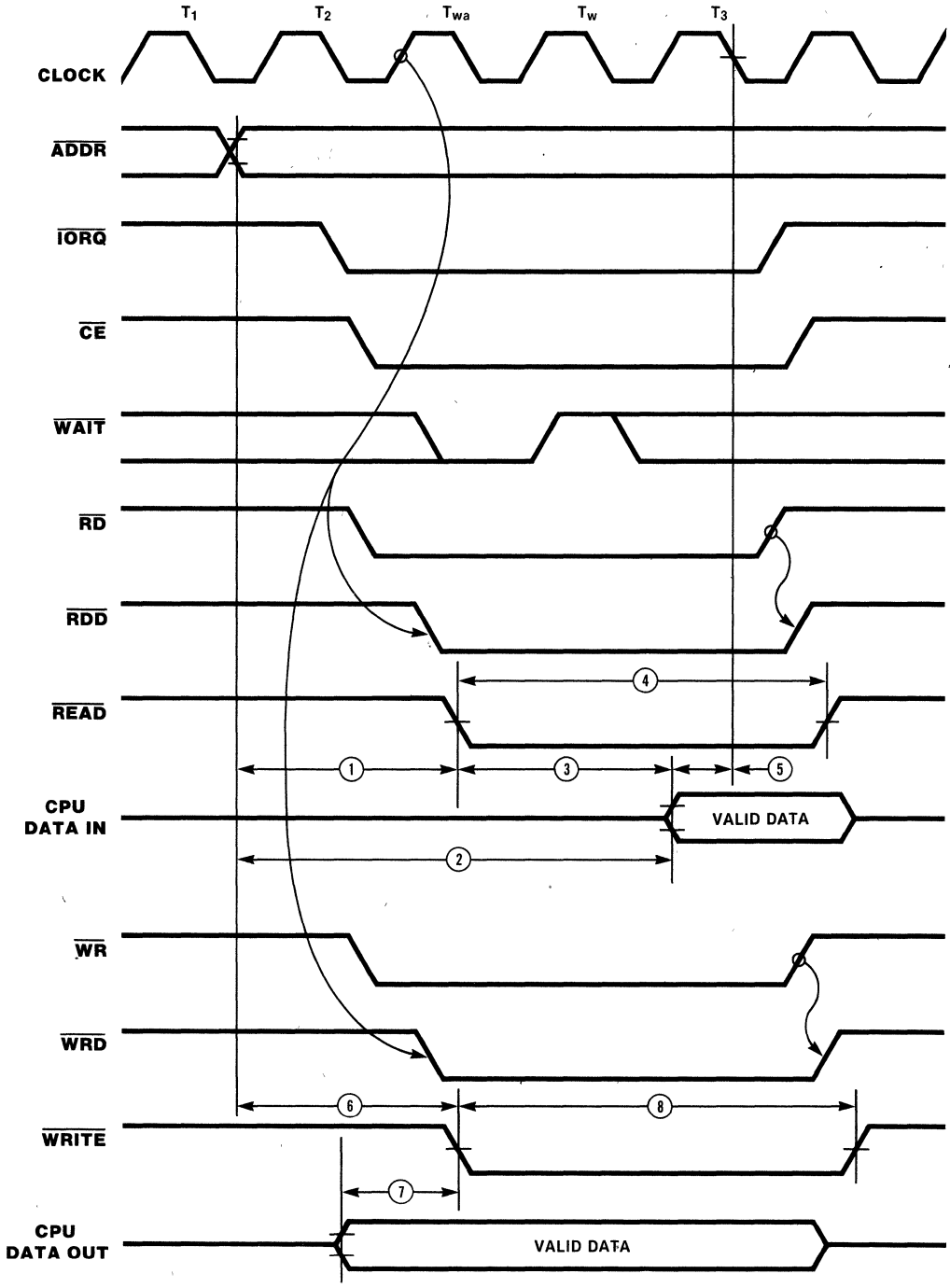


Figure 3d. Z80H CPU to Z8500A Peripheral Minimum I/O Cycle Timing

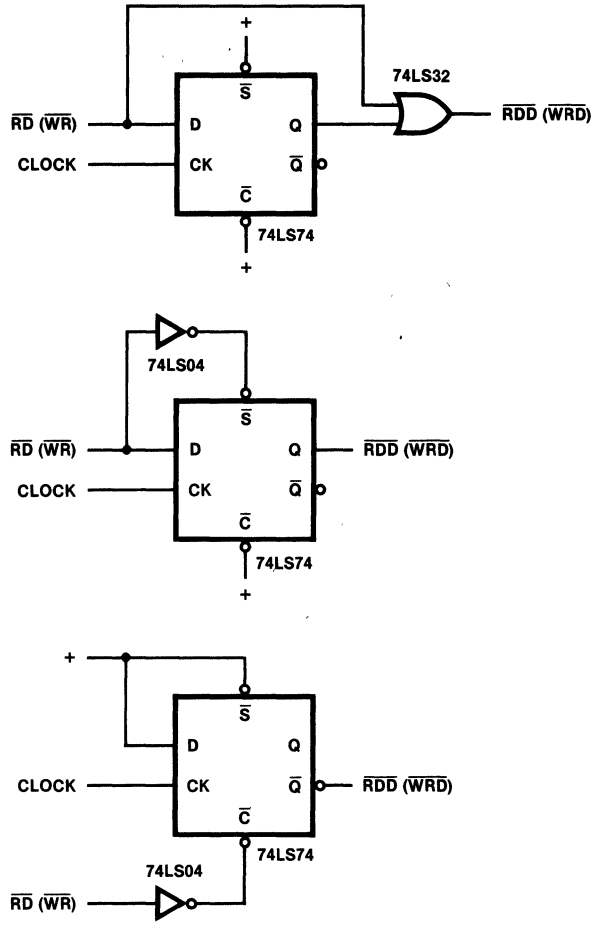


Figure 4. Delaying $\overline{RD}$ or $\overline{WR}$

Table 15. Parameter Equations

Z80H Parameter	Z8500A Equation	Value	Units
TsD(Cf)	Address 4TcC+TwCh-TdCr(A)-TdA(DR) $\overline{RD}$ - delayed 2TcC+TwCh-TdRD(DR)	55 min 125 min	ns ns

INTERRUPT ACKNOWLEDGE CYCLES

The primary timing differences between the Z80 CPUs and Z8500 peripherals occur in the Interrupt Acknowledge cycle. The Z8500 timing parameters that are significant during Interrupt Acknowledge cycles are listed in Table 16, while the Z80 parameters are listed in Table 17. The reference numbers in Tables 16 and 17 refer to Figures 6, 8a, and 8b.

If the CPU and the peripherals are running at different speeds (as with the Z80H interface), the $\overline{INTACK}$ signal must be synchronized to the peripheral clock. Synchronization is discussed in detail under Interrupt Acknowledge for Z80H CPU to Z8500/8500A Peripherals.

During an Interrupt Acknowledge cycle, Z8500 peripherals require both $\overline{INTACK}$ and $\overline{RD}$ to be active at certain times. Since the Z80 CPUs do not issue either $\overline{INTACK}$ or $\overline{RD}$, external logic must generate these signals.

Generating these two signals is easily accomplished, but the Z80 CPU must be placed into a Wait condition until the peripheral interrupt vector is valid. If more peripherals are added to the daisy chain, additional Wait states may be necessary to give the daisy chain time to settle. Sufficient time between $\overline{INTACK}$ active and $\overline{RD}$ active should be allowed for the entire daisy chain to settle.

Since the Z8500 peripheral daisy chain does not use the IP flag except during interrupt acknowledge, there is no need for decoding the RETI instruction used by the Z80 peripherals. In each of the Z8500 peripherals, there are commands that reset the individual IUS flags.

EXTERNAL INTERFACE LOGIC

The following sections discuss external interface logic required during Interrupt Acknowledge cycles for each interface type.

CPU/Peripheral Same Speed

Figure 5 shows the logic used to interface the Z80A CPU to the Z8500 peripherals and the Z80B CPU to Z8500A peripherals during an Interrupt Acknowledge cycle. The primary component in this logic is the Shift register (74LS164), which generates $\overline{INTACK}$, $\overline{READ}$, and $\overline{WAIT}$.

Table 16. Z8500 Timing Parameters Interrupt Acknowledge Cycles

	Worst Case		4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	Units
1.	TsIA(PC)	$\overline{INTACK}$ Low to PCLK High Setup	100		100		ns
	ThIA(PC)	$\overline{INTACK}$ Low to PCLK High Hold	100		100		ns
2.	TdIAi(RD)	$\overline{INTACK}$ Low to RD (Acknowledge) Low	350		250		ns
5.	TwRDA	$\overline{RD}$ (Acknowledge) Width	350		250		ns
3.	TdRDA(DR)	$\overline{RD}$ (Acknowledge) to Data Valid		250		180	ns
	TsIEI(RDA)	IEI to $\overline{RD}$ (Acknowledge) Setup	120		100		ns
	ThIEI(RDA)	IEI to $\overline{RD}$ (Acknowledge) Hold	100		70		ns
	TdIEI(IE)	IEI to IEO Delay		150		100	ns

Table 17. Z80 CPU Timing Parameters Interrupt Acknowledge Cycles

	Worst Case		4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	8 MHz Min	8 MHz Max	Units
	TdC(M1f)	Clock High to $\overline{M1}$ Low Delay		100		80		70	ns
	TdM1f(IORQf)	$\overline{M1}$ Low to $\overline{IORQ}$ Low Delay	575*		345*		275*		ns
4.	TsD(Cr)	Data to Clock High Setup	35		30		25		ns

*Z80A: 2TcC + TwCh + TfC - 65
Z80B: 2TcC + TwCh + TfC - 50
Z80H: 2TcC + TwCh + TfC - 45

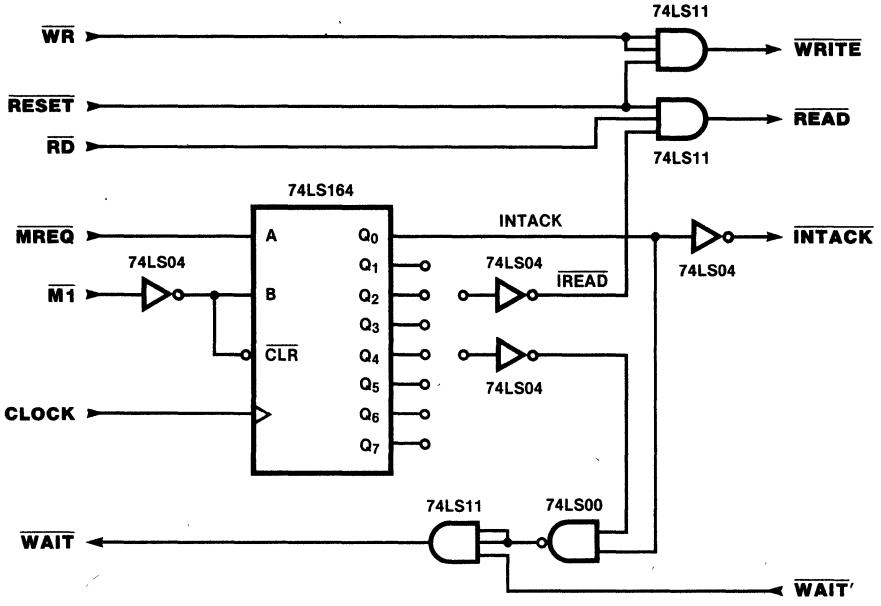


Figure 5. Z80A/Z80B CPU to Z8500/Z8500A Peripheral Interrupt Acknowledge Interface Logic

During I/O and normal memory access cycles, the Shift register remains cleared because the $\overline{MT}$ signal is inactive. During opcode fetch cycles, also, the Shift register remains cleared, because only 0s can be clocked through the register. Since Shift register outputs are Low, $\overline{READ}$, $\overline{WRITE}$, and $\overline{WAIT}$ are controlled by other system logic and gated through the AND gates (74LS11). During I/O and normal memory access cycles, $\overline{READ}$ and $\overline{WRITE}$ are active as a result of the system $\overline{RD}$ and $\overline{WR}$ signals (respectively) becoming active. If system logic requires that the CPU be placed into a Wait condition, the $\overline{WAIT}$' signal controls the CPU. Should it be necessary to reset the system, $\overline{RESET}$ causes the interface logic to generate both $\overline{READ}$ and $\overline{WRITE}$ (the Z8500 peripheral Reset condition).

Normally an Interrupt Acknowledge cycle is indicated by the Z80 CPU when $\overline{MT}$ and $\overline{IORQ}$ are both active (which can be detected on the third rising clock edge after T_1). To obtain an early indication of an Interrupt Acknowledge cycle, the Shift register decodes an active $\overline{MT}$ in the presence of an inactive $\overline{MREQ}$ on the rising edge of T_2.

During an Interrupt Acknowledge cycle, the $\overline{INTACK}$ signal is generated on the rising edge of T_2. Since it is the presence of $\overline{INTACK}$ and an active $\overline{READ}$ that gates the interrupt vector onto the data bus, the logic must also generate $\overline{READ}$ at the proper time. The timing parameter of concern here is TdIAi(RD) [$\overline{INTACK}$ to $\overline{RD}$ (Acknowledge) Low Delay]. This time delay allows the interrupt daisy chain to settle so that the device requesting the interrupt can place its interrupt vector onto the data bus. The Shift register allows a sufficient time delay from the generation of $\overline{INTACK}$ before it generates $\overline{READ}$. During this delay, it places the CPU into a Wait state until the valid interrupt vector can be placed onto the data bus. If the time between these two signals is insufficient for daisy chain settling, more time can be added by taking $\overline{READ}$ and $\overline{WAIT}$ from a later position on the Shift register.

Figure 6 illustrates Interrupt Acknowledge cycle timing resulting from the Z80A CPU to Z8500 peripheral and the Z80B CPU to Z8500A peripheral interface. This timing comes from the logic illustrated in Figure 5, which can be used for both interfaces. Should more Wait states be required, the additional time can be calculated in terms of system clocks, since the CPU clock and PCLK are the same.

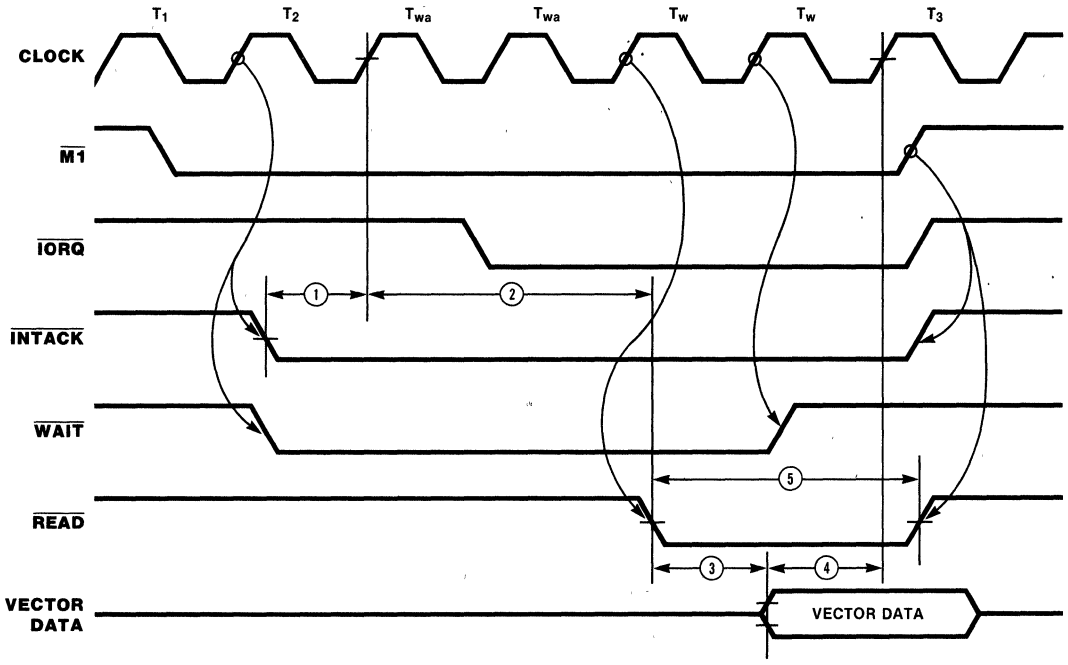


Figure 6. Z80A/Z80B CPU to Z8500/Z8500A Peripheral Interrupt Acknowledge Interface Timing

Z80H CPU to Z8500/Z8500A Peripherals

Figure 7 depicts logic that can be used in interfacing the Z80H CPU to the Z8500/Z8500A peripherals. This logic is the same as that shown in Figure 5, except that a synchronizing flip-flop is used to recognize an Interrupt Acknowledge cycle. Since Z8500 peripherals do not rely upon PCLK except during Interrupt Acknowledge cycles, synchronization need occur only at that time. Since the CPU and the peripherals are running at different speeds, $\overline{INTACK}$ and $\overline{RD}$ must be synchronized to the Z8500 peripherals clock.

During I/O and normal memory access cycles, the synchronizing flip-flop and the Shift register remain cleared because the $\overline{M1}$ signal is inactive. During opcode fetch cycles, the flip-flop and the Shift register again remain cleared, but this time because the $\overline{MREQ}$ signal is active. The synchronizing flip-flop allows an Interrupt Acknowledge cycle to be recognized on the rising edge of T_2 when $\overline{M1}$ is active and $\overline{MREQ}$ is inactive, generating the INTA signal. When INTA is active, the Shift register can clock and generate $\overline{INTACK}$ to the peripheral and $\overline{WAIT}$ to the CPU. The Shift register delays the generation of $\overline{READ}$ to the peripheral until the daisy chain settles. The $\overline{WAIT}$ signal is removed when sufficient time has been allowed for the interrupt vector data to be valid.

Figure 8a illustrates Interrupt Acknowledge cycle timing for the Z80H CPU to Z8500 peripheral interface. Figure 8b illustrates Interrupt Acknowledge cycle timing for the Z80H CPU to Z8500A peripheral interface. These timings result from the logic in Figure 7. Should more Wait states be required, the needed time should be calculated in terms of PCLKs, not CPU clocks.

Z80 CPU to Z80 and Z8500 Peripherals

In a Z80 system, a combination of Z80 peripherals and Z8500 peripherals can be used compatibly. While there is no restriction on the placement of the Z8500 peripherals in the daisy chain, it is recommended that they be placed early in the chain to minimize propagation delays during RETI cycles.

During an Interrupt Acknowledge cycle, the IEO line from the Z8500 peripherals changes to reflect the interrupt status. Time should be allowed for this change to ripple through the remainder of the daisy chain before activating $\overline{IORQ}$ to the Z80 peripherals, or $\overline{READ}$ to the Z8500 peripherals.

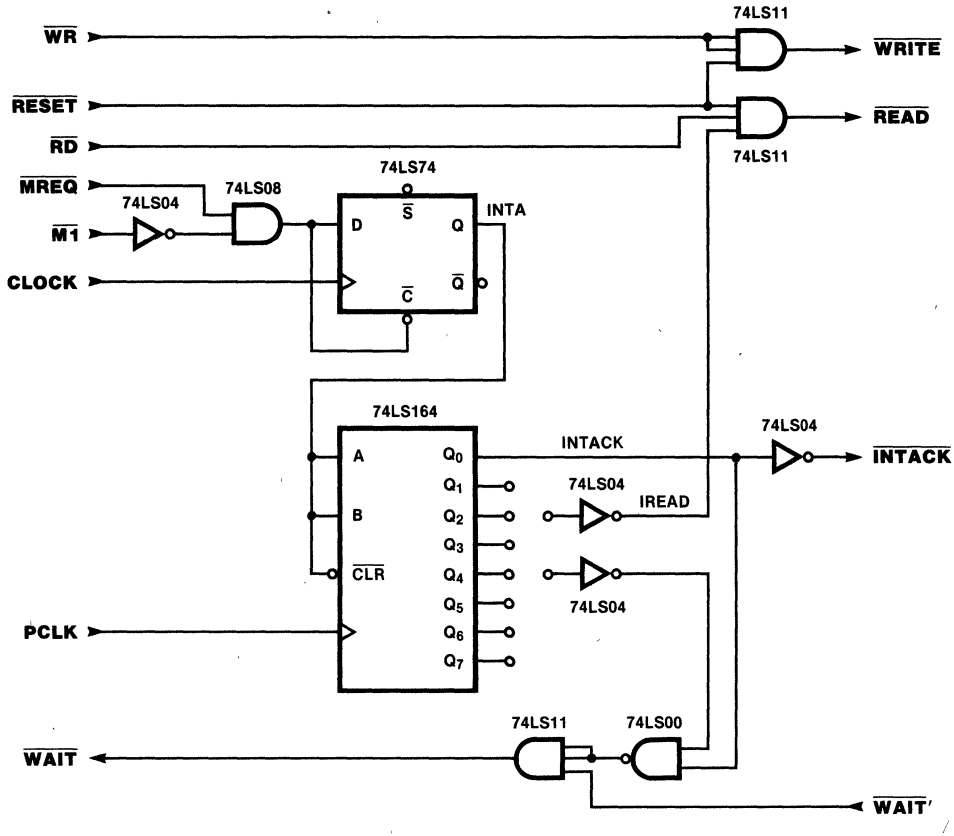


Figure 7. Z80H to Z8500/Z8500A Peripheral Interrupt Acknowledge Interface Logic

During the RETI cycles, the IEO line from the Z8500 peripherals does not change state as in the Z80 peripherals. As long as the peripherals are at the top of the daisy chain, propagation delays are minimized.

The logic necessary to create the control signals for both Z80 and Z8500 peripherals is shown in Figure 9. This logic delays the generation of $\overline{IORQ}'$ to the Z80 peripherals by the same amount of time necessary to generate $\overline{READ}$ for the Z8500 peripherals. Timing for this logic during an Interrupt Acknowledge cycle is depicted in Figure 10.

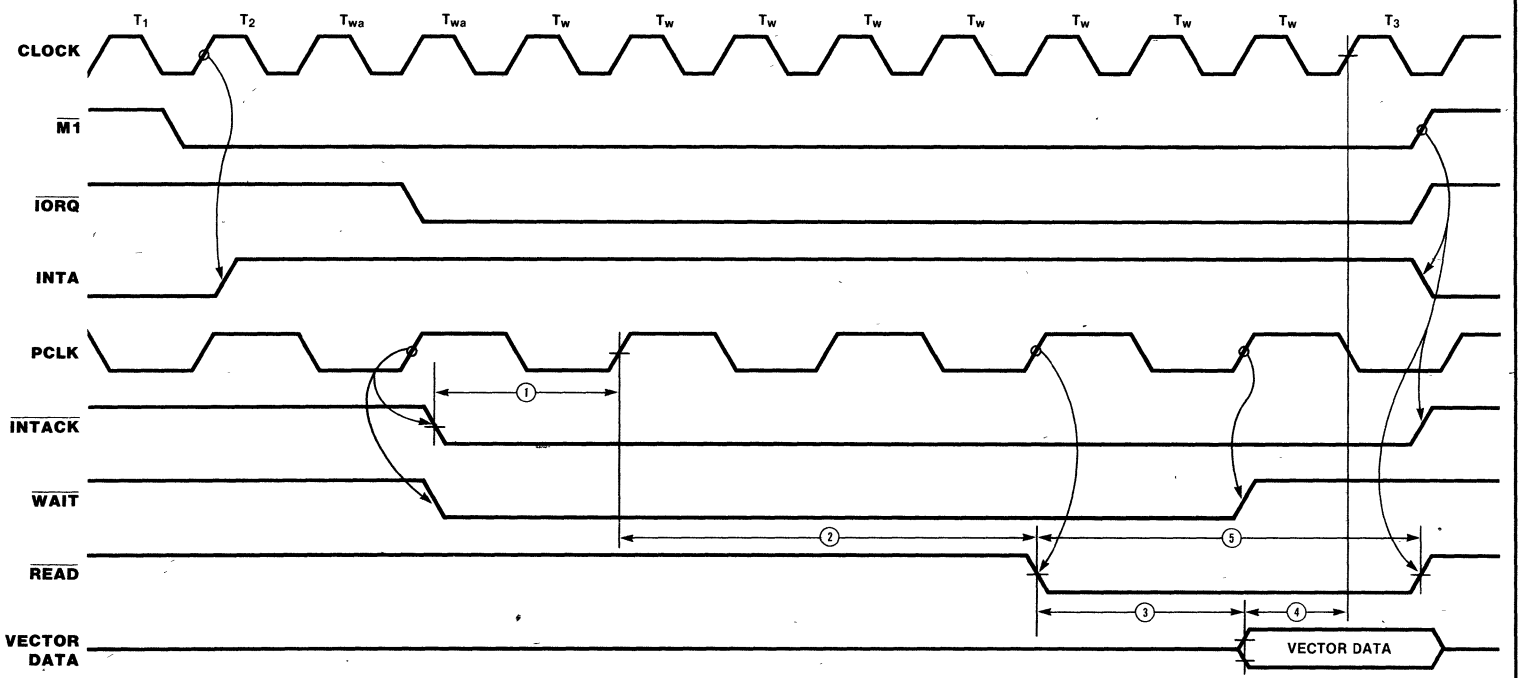


Figure 8a. Z80H CPU to Z8500 Peripheral Interrupt Acknowledge Interface Timing

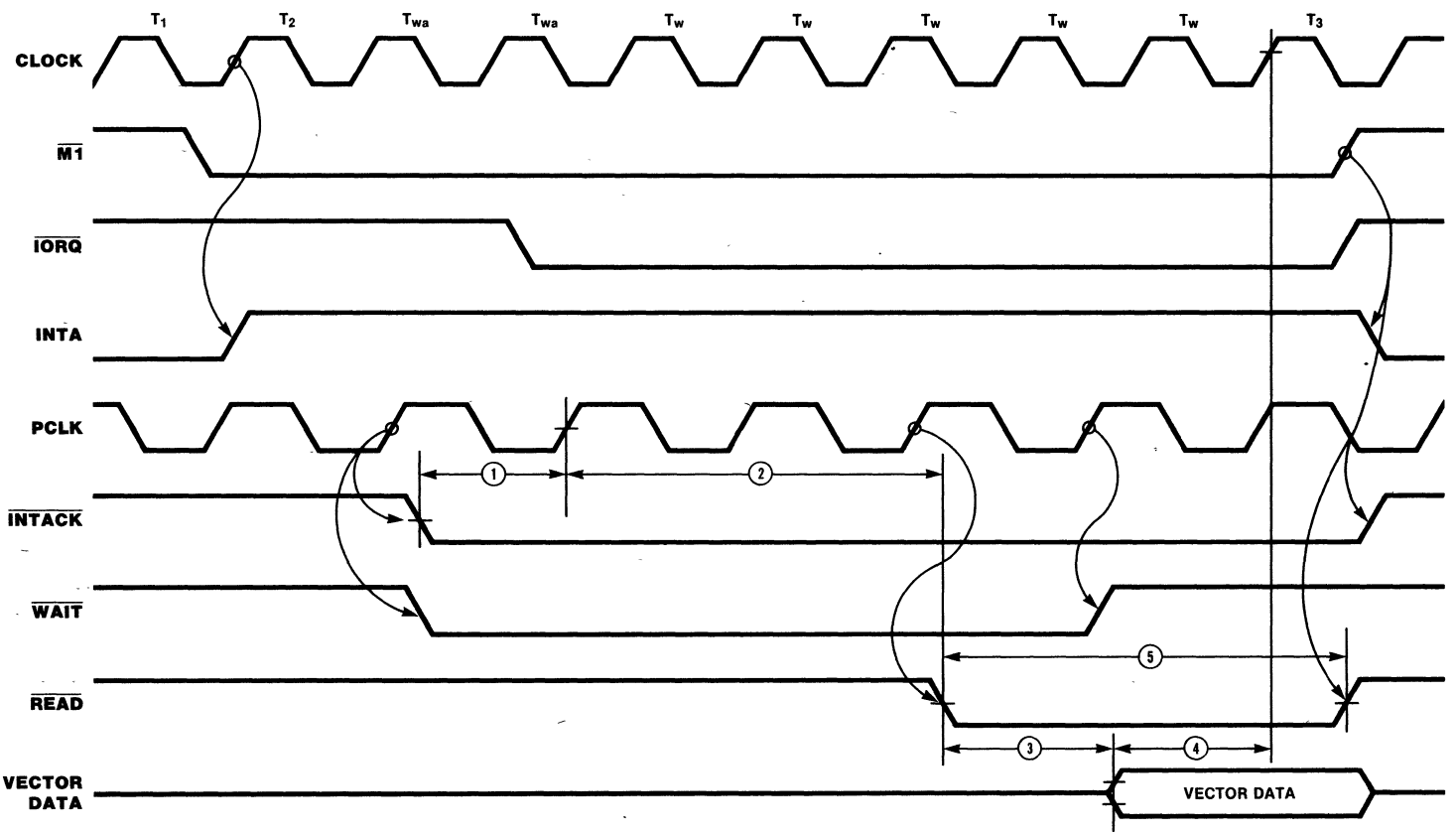


Figure 8b. Z80H CPU to Z8500A Peripheral Interrupt Acknowledge Interface Timing

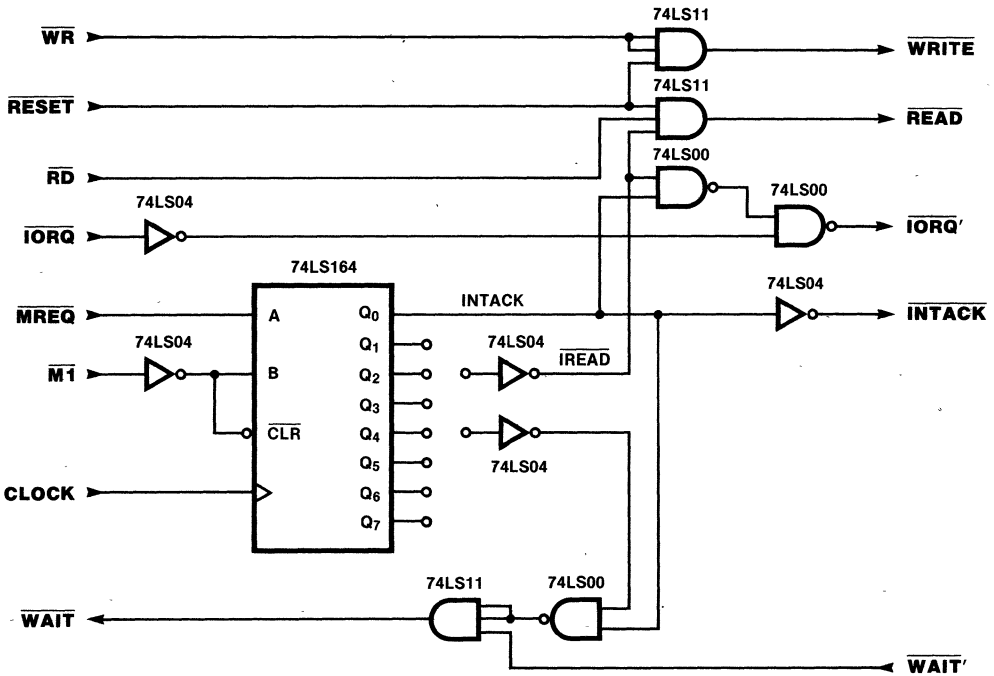


Figure 9. Z80 and Z8500 Peripheral Interrupt Acknowledge Interface Logic

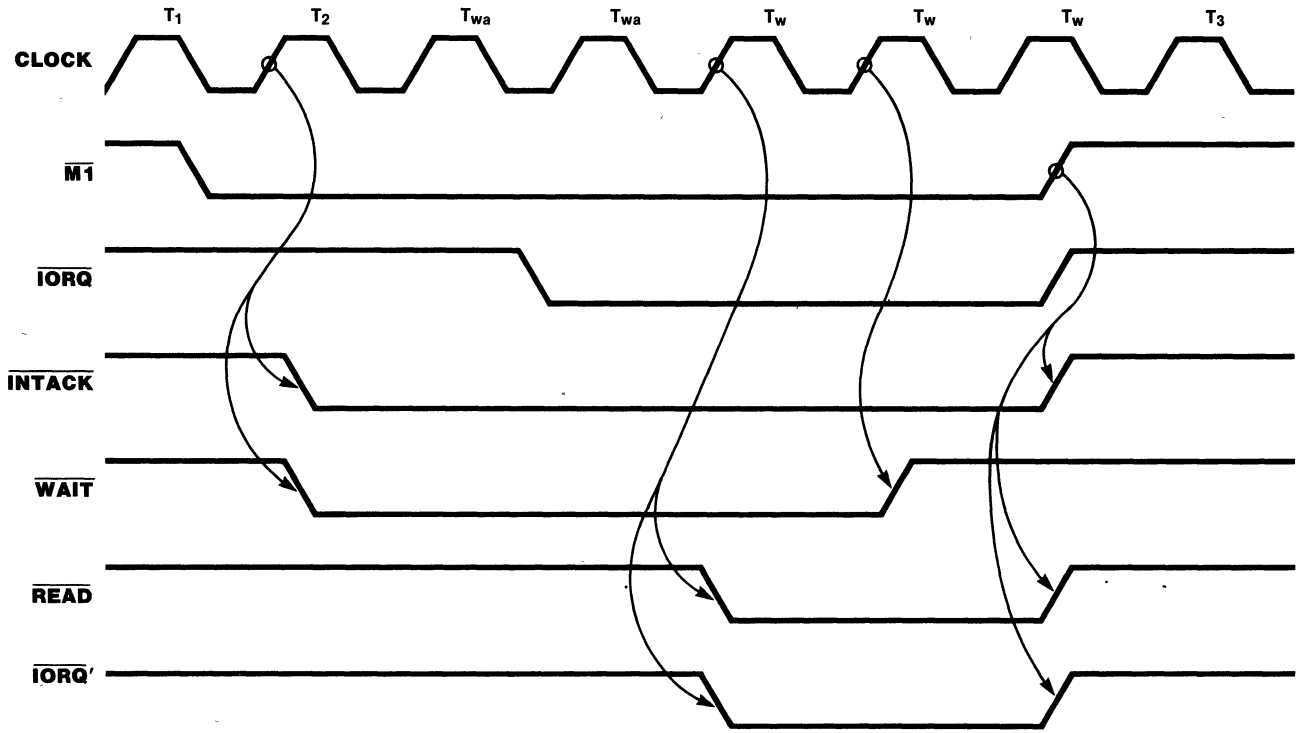


Figure 10. Z80 and Z8500 Peripheral Interrupt Acknowledge Interface Timing

SOFTWARE CONSIDERATIONS -- POLLED OPERATION

There are several options available for servicing interrupts on the Z8500 peripherals. Since the vector or IP registers can be read at any time, software can be used to emulate the Z80 interrupt response. The interrupt vector read reflects the interrupt status condition even if the device is programmed to return a vector that does not reflect the status change (SAV or VIS is not set). The code below is a simple software routine that emulates the Z80 vector response operation.

Z80 Vector Interrupt Response, Emulation by Software

```
;This code emulates the Z80 vector interrupt
;operation by reading the device interrupt
;vector and forming an address from a vector
;table.  It then executes an indirect jump to
;the interrupt service routine.

INDX:    LD      A,CIVREG     ;CURRENT INT. VECT. REG.
         OUT     (CTRL),A     ;WRITE REG. PTR.
         IN      A,(CTRL)     ;READ VECT. REG.
         INC     A            ;VALID VECTOR?
         RET     Z            ;NO INT - RETURN
         AND     00001110B    ;MASK OTHER BITS
         LD      E,A
         LD      D,0          ;FORM INDEX VALUE
         LD      HL,VECTAB
         ADD     HL,DE        ;ADD VECT. TABLE ADDR.
         LD      A,(HL)       ;GET LOW BYTE
         INC     HL
         LD      H,(HL)       ;GET HIGH BYTE
         LD      L,A          ;FORM ROUTINE ADDR.
         JP      (HL)         ;JUMP TO IT

VECTAB:  DEFW    INT1
         DEFW    INT2
         DEFW    INT3
         DEFW    INT4
         DEFW    INT5
         DEFW    INT6
         DEFW    INT7
         DEFW    INT8
```

A SIMPLE Z80-Z8500 SYSTEM

The Z8500 devices interface easily to the Z80 CPU, thus providing a system of considerable flexibility. Figure 11 illustrates a simple system using the Z80A CPU and the Z8536 Counter/Timer and Parallel I/O Unit (CIO) in a mode 1 or non-interrupt environment. Since interrupt vectors are not used, the $\overline{\text{INTACK}}$ line is tied High and no additional logic is needed. Because the CIO can be used in a polled interrupt environment, the $\overline{\text{INT}}$ pin is connected to the CPU. The Z80 should not be set for mode 2 interrupts since the CIO will never place a vector onto the data bus. Instead, the CPU should be placed into mode 1 interrupt mode and a global interrupt service routine can poll the CIO to determine what caused the interrupt to occur. In this system, the software emulation procedure described above is effective.

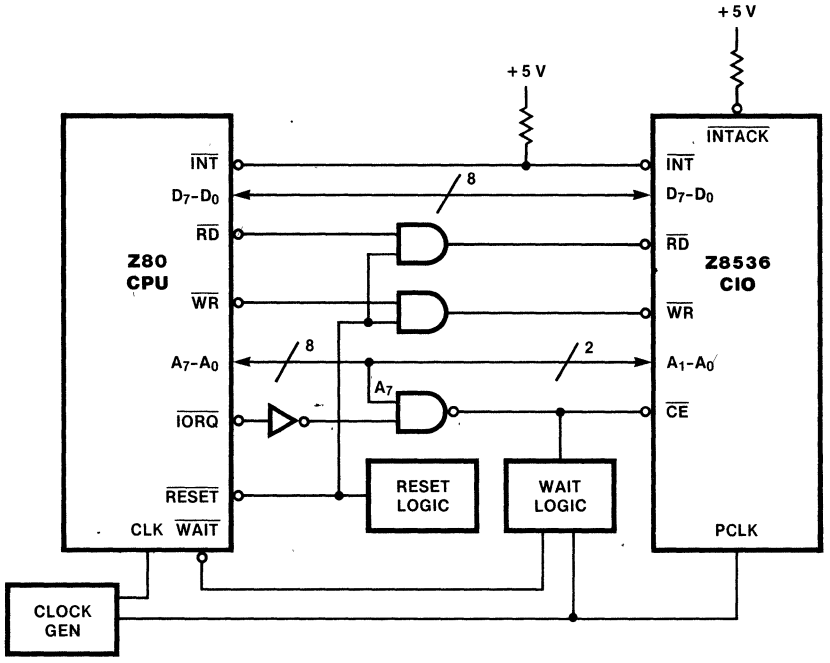


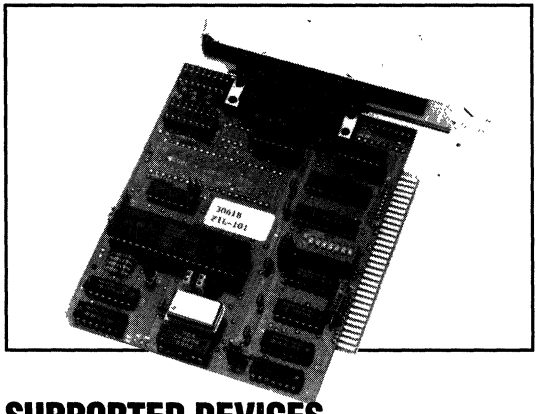
Figure 11. Z80 to Z8500 Simple System Mode 1 Interrupt or Non-Interrupt Structure

Additional Information - Zilog Publications

1. Z80 CPU Technical Manual (03-0029-01)
2. Z80 DMA Technical Manual (00-2013-A0)
3. Z80 PIO Technical Manual (03-0008-01)
4. Z80 CTC Technical Manual (03-0036-02)
5. Z80 SIO Technical Manual (03-3033-01)
6. Z8000 CPU AC Characteristics (00-2293-01)
7. Z80 Family Interrupt Structure Tutorial (611-1809-0003)
8. Z8530 SCC Technical Manual (00-2057-01)
9. Z8536 CIO Technical Manual (00-2091-01)
10. Z8038 FIO Technical Manual (00-2051-01)

Z8000® Datacommunications

Z85C3000ZCO
PRODUCT SPECIFICATION



SUPPORTED DEVICES
Z8530, Z85C30, Z85130

DESCRIPTION
The kit contains an assembled PC/XT/AT circuit board with one high speed serial port, selectively driven by RS-232C or RS-422 line drivers. The kit also contains software and documentation to support software and hardware development for Zilog's SCC and ESCC™ devices.

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

SPECIFICATIONS
Power Requirements
+5 Vdc @ .5 A

Dimensions
Width: 4 in. (10.16 cm)
Length: 5 in. (12.70 cm)

Serial Interface
A DB25 port selectively driven by RS-232C or RS-422 at selectable baud rates.

KIT CONTENTS
Z85130 Evaluation Board
CMOS Z85130 ESCC and Z85C30 SCC
4.9152 MHz Crystal
RS-232C and RS-422 line drivers
DB25 connector

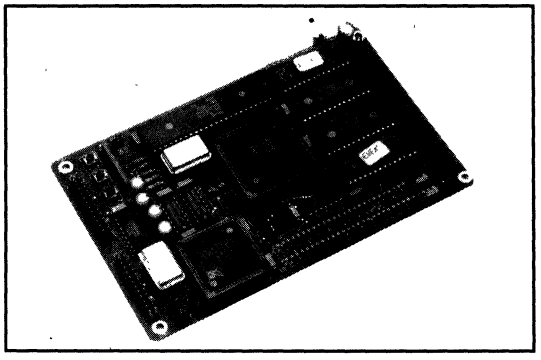
Software (IBM-PC Platform)
Source and executable codes to run SCC or ESCC™ Controller in SDLC/HDLC and ASYNC modes using DMA, Interrupt and polling methods. All codes are written in C and compiled using the Microsoft®* Quick C compiler.

Documentation
Z85C30 and Z85130 Product Specifications
Z85C30 and Z85130 Technical Manuals
Z85C3000ZCO Kit User Guide
SEALEVEL User's Manual

ORDERING INFORMATION
Part No: Z85C3000ZCO

* Microsoft is a registered trademark of Microsoft Corporation.

Z16C0100ZCO
PRODUCT SPECIFICATION



SUPPORTED DEVICES
Z16C01, Z16C20, Z16C30

DESCRIPTION
The kit contains an assembled circuit board, software and documentation to support software, and hardware development for the Z16C01 CPU, Z16C20 General Logic Unit and Z16C30 Universal Serial Controller. The supplied cross C compiler, assembler and link/loader package allows full C and assembly language programming support. A board resident debug monitor program and its PC based counterpart allow object code to be down-loaded and subsequently debugged.

SPECIFICATIONS
Power Requirements
+5 Vdc @ .5 A

Dimensions
Single Euro Card Format
Width: 3.94 in. (10 cm)
Length: 6.30 in. (16 cm)

Serial Interface
RS-232C @ 9600 baud

KIT CONTENTS
Z16C01/20/30 Evaluation Board
CMOS Z16C01 CPU
CMOS Z16C20 GLU
CMOS Z16C30 USC
20 MHz Crystal USC Clock
20 MHz System Oscillator
Two (64K)/8K x 8 EPROMs
(programmed with Debug Monitor)
Two 32K/(8K) x 8 STATIC RAM
RS-232C PC Interface
Z16C01 Expansion Bus Connector

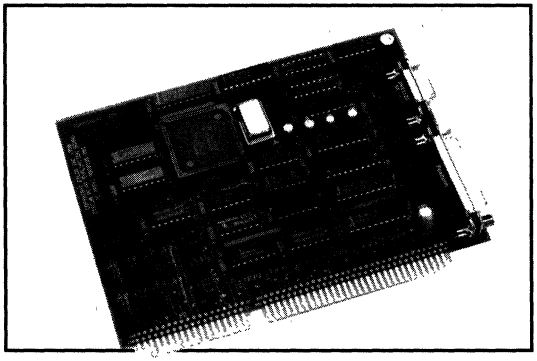
Cables
25-Pin RS-232C Cable

Software (IBM-PC Platform)
Z8000 Cross C Compiler
Z8/Z80/Z8000 Cross Assembler
MOBJ Link/Loader
Resident Debug Monitor Source Code
Host Package Source Code
Z16C20 Example Software

Documentation
Z8000 CPU Technical Manual
Z8000 CPU Programmer's Pocket Guide
Z16C20 GLU Product Specification
Z16C30 USC™ Controller Technical Manual
Z16C0100ZCO Kit User Manual
CC8K C Compiler User Guide
Z8000 Cross Assembler User Guide
MOBJ Link/Loader User Guide

ORDERING INFORMATION
Part No: Z16C0100ZCO

Z16C3001ZCO
PRODUCT SPECIFICATION



SUPPORTED DEVICES
Z16C30, Z16C33

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 and MUSC devices.

The board illustrates the use of Zilog's USC and MUSC devices 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.

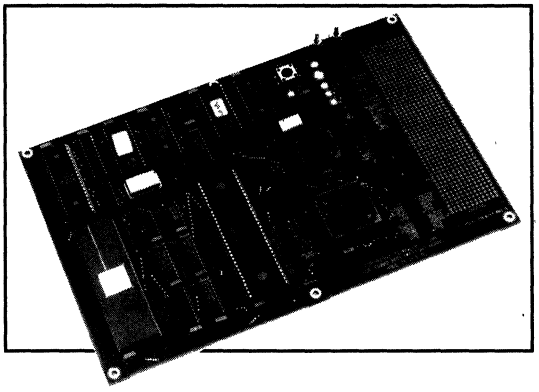
KIT CONTENTS
Z16C30/Z16C33 Evaluation Board
CMOS Z16C30 USC and Z16C33 MUSC
20 MHz Crystal
RS-232C and RS-422 line drivers
DB9 and DB25 Interfaces

Software (IBM-PC Platform)
Source and executable codes to run the USC or MUSC in SDLC/HDLC or ASYNC mode. All codes are written in C and compiled using the Microsoft C 5.1 compiler.

Documentation
Z16C30 and Z16C33 Product Specifications
Z16C30/Z16C33 Technical Manual
Z16C3001ZCO Kit User Guide

ORDERING INFORMATION
Part No: Z16C3001ZCO

Z16C3000ZCO
PRODUCT SPECIFICATION



SUPPORTED DEVICES
Z16C30, Z85C30

DESCRIPTION
The kit contains an assembled circuit board, software and documentation to support software and hardware development for the Zilog Z16C30 Universal Serial Controller and Z85C30 Serial Communication Controller devices in a Motorola 68000 environment.

A board resident debug monitor program allows object code to be down-loaded and subsequently debugged.

SPECIFICATIONS
Power Requirements
+5 Vdc @ .75 A

Dimensions
Width: 6.5 in. (16.5 cm)
Length: 9.8 in. (24.9 cm)

Serial Interface
RS-232C @ 9600 baud

KIT CONTENTS
USC / 68000 Evaluation Board
CMOS Z16C30 USC
CMOS Z85C30 SCC
MC68000 CPU
MC68450 DMAC
4 MHz Crystal SCC CLock
20 MHz System Oscillator
Two (64K)/8K x 8 EPROMs
programmed with Debug Monitor)
Two 8K x 8 STATIC RAM
RS-232C PC Interface
Prototype Wire Wrap Area

Cables
25-Pin RS-232C Cable

Software (IBM-PC Platform)
Resident Debug Monitor Source Code
Z16C30 Example Software

Documentation
Z16C30 USC Product Specification
Z16C30 USC Technical Manual
Z85C30 SCC Product Specification
Z8530 SCC Technical Manual
Z16C3000ZCO Kit User Manual
Z16C3000ZCO Kit Note to User

ORDERING INFORMATION
Part No: Z16C3000ZCO

Military Qualified Datacom Products

Zilog P/N	Description	Speed	Package	883C	SMD P/N	JAN P/N
Z0803004CMB	Z-BUS SCC	4 MHz	40-Pin DIP	Qualed	5962-8551802QA	N/A
Z0803004LMB	Z-BUS SCC	4 MHz	44-Pin LCC	Qualed	5962-8551802YA	N/A
Z0803006CMB	Z-BUS SCC	6 MHz	40-Pin DIP	Qualed	5962-8551801QA	N/A
Z0803006LMB	Z-BUS SCC	6 MHz	44-Pin LCC	Qualed	5962-8551801YA	N/A
Z0853004CMB	Z8530 SCC	4 MHz	40-Pin DIP	Qualed	5962-8752702QA	N/A
Z0853004LMB	Z8530 SCC	4 MHz	44-Pin LCC	Qualed	5962-8752702YA	N/A
Z0853006CMB	Z8530 SCC	6 MHz	40-Pin DIP	Qualed	5962-8752701QA	N/A
Z0853006LMB	Z8530 SCC	6 MHz	44-Pin LCC	Qualed	5962-8752701YA	N/A
Z85C3006CMB	Z85C30 CMOS SCC	6 MHz	40-Pin DIP	Qualed	5962-8868901QA	M38510/48601BQA
Z85C3006LMB	Z85C30 CMOS SCC	6 MHz	44-Pin LCC	Qualed	5962-8868901YA	N/A
Z85C3008CMB	Z85C30 CMOS SCC	8 MHz	40-Pin DIP	Qualed	5962-8868902QA	M38510/48602BQA
Z85C3008LMB	Z85C30 CMOS SCC	8 MHz	44-Pin LCC	Qualed	5962-8868902YA	N/A
Z85C3010CMB	Z85C30 CMOS SCC	10 MHz	40-Pin DIP	Qualed	Q1 '91	N/A
Z85C3010LMB	Z85C30 CMOS SCC	10 MHz	44-Pin LCC	Qualed	Q1 '91	N/A
Z16C3010GMB	CMOS USC	10 MHz	68-Pin PGA	Qualed	Q1 '91	N/A
Z8523010CMB	CMOS ESCC	10 MHz	40-Pin DIP	Q1 '91	Q2 '91	N/A
Z8523010LMB	CMOS ESCC	10 MHz	44-Pin LCC	Q1 '91	Q2 '91	N/A
Z8523016CMB	CMOS ESCC	16.0 MHz	40-Pin DIP	Q1 '91	Q2 '91	N/A
Z8523016LMB	CMOS ESCC	16.0 MHz	44-Pin LCC	Q1 '91	Q2 '91	N/A

ZiLOG QUALITY AND RELIABILITY

Zilog's Quality and Reliability Program

Introduction

Zilog has an excellent reputation for the quality and reliability of its products.

Zilog's Quality and Reliability Program is based on careful study of the principles laid down by such pioneers as W.E. Deming and J.M. Juran and, perhaps even more important, observation of the practical implementation of those principles in Japanese, European and American manufacturing facilities.

The Zilog program begins with employee involvement. Whether the judgement of our performance is based on perfection in incoming inspection, trouble free service in the field or timely and accurate customer service, we recognize that our employees ultimately control these factors. Hence, our Quality Program is broadly shared throughout the organization.

1. Harmony Between Design and Process

High product quality and reliability in VLSI products is possible only if there is structural harmony between product design and the manufacturing process. Great care is taken to assure that the statistical process control limits observed within the manufacturing plants properly guardband the design technology used to configure the circuit and layout in Zilog's automated design methodology.

Through use of a technique which we call Process Templating, the technology file in the automated design system is periodically updated to assure that product design parameters fall within the statistical control limits with which the process is actually operated. In simple terms, the Process Template is the profile displayed by the process evaluation parameters which are automatically recorded from the test patterns on wafers as they proceed through the production line. These parameters are translated into the design technology file attributes such that every product design bears a key and lock relationship to the process.

2. Training

Product Design and Processing are people dependent. Zilog training emphasizes the fundamentals involved in design for quality and reliability.

Customer Service, an important aspect of Zilog's quality performance as a vendor, also depends upon our people clearly understanding their jobs, and our obligations to our customers. This too is part of the training curriculum administered by Zilog.

3. Order Acknowledgement Policy

One definition of vendor quality performance is that the vendor "does what he promises or acknowledges." Reliability and quality warranties can be met only if Zilog and the customer are in agreement on product and delivery specifications. Zilog makes an extra effort to assure that the customer is fully informed by providing documents with its purchase order acknowledgements that clearly state what Zilog understands the specifications to be.

4. Test Guardbanding

No physical attribute is absolute. Customers' test methods may differ from Zilog's due to variations in test equipment, temperature or specification interpretation. To assure that every Zilog product performs to full customer expectations, Zilog uses a "waterfall" methodology in its testing. The first electrical tests made on the circuit, at the wafer probe operations, are guardbanded to the final test specifications. The final test specifications, in turn, are guardbanded to the quality control outgoing sample The quality control outgoing sample is guardbanded to the customer procurement or data sheet specifications. This technique of "waterfall" guardbanding assures that circuits which may be marginal to the customer's expectations are eliminated in the manufacturing process long before they get to the shipping container.

5. Probe at Temperature

Semiconductor devices tend to exhibit their most limited performance at the highest operating temperature. Therefore, it is Zilog's policy that all chips are tested at high temperature the very first time they are electrically screened, at the wafer probe station. The circuits are tested again at their upper operating temperature limit in the 100% final test operation.

6. Process Characterization

Before release to production, every process is thoroughly characterized by an exhaustive series of pilot production runs and tests which identify the statistical, electrical, and mechanical limits of which that particular process regime is capable. This documentation, which fills a large loose leaf binder for each process, is maintained as the historical record or "footprint" for that particular regime

Process recharacterization is done any time there is a major process or manufacturing site change, and the resulting documentation is then added to the characterization history. Once the process is fully characterized, the frequent test site evaluation and process template data demonstrates that the process remains in specification.

7. Product Characterization

Every Zilog product design is evaluated over extremes of operating temperature, supply voltage and clock frequencies, prior to release to production. This information permits the proper guardbanding of the test program waterfall and identification of many marginal "corners" in design tolerances.

A product characterization report, which summarizes the more important tolerances identified in the process of this exhaustive product design evaluation, is available to Zilog's customers.

8. Process Qualification

Zilog also qualifies every process prior to production by an exhaustive stress sequence performed on test chips and on representative products. Once a process regime is qualified, a process requalification is performed any time there is a major process change, or whenever the process template statistical quality limits are significantly exceeded or adjusted.

9. Product Qualification

In addition to characterization, every new Zilog product design is fully qualified by a comprehensive series of life, electrical, and environmental tests before release to production. Again, a qualification report is available to our customers which summarizes certain key life and environmental data taken in the course of these evaluations. Whenever possible, industry standard environmental and life tests are employed.

10. PPM Measurement, Direct and Indirect

It is frequently said that if you want to improve something, you need to put a measure on it. Therefore, Zilog measures its outgoing quality "parts per million" by the maintenance of careful records on the statistical sampling of production lots prepared for shipment. This information is then translated by our statisticians to a statement of our parts per million (or parts per billion) outgoing quality performance.

Of course, it is one thing for Zilog to think it is doing a good job in outgoing product quality and it is another for a customer to agree. Therefore, we ask certain key customers to provide us with their incoming inspection data which helps us calibrate our outgoing performance in terms of the actual results in the field. The fact that Zilog has been awarded "ship to stock" status by many customers testifies to our success in this area.

11. FIT Measurement Direct and Indirect

Just as Zilog records its outgoing quality in terms of parts per million, it also measures its outgoing product reliability in terms of "FITS" or failures per billion device hours, using the results of weekly operating life test measurements on the circuits, performed in accordance with the standard specifications.

12. Field Quality Engineers

It is frequently said that, "the customer is always right." If the customer has an application quality or reliability problem while using a Zilog product, whether it is Zilog's responsibility or not, we believe that we have a responsibility to resolve it. Therefore, Zilog maintains a force of skilled Applications Engineers who are also trained as field quality engineers and are available on immediate call to consult at the customer's locations on any problems they may be experiencing with Zilog product performance.

13. Product Analysis

As noted earlier, we feel that a customer problem is a Zilog problem. Accordingly, Product Analysis facilities, staffed by experienced professionals, exist at each Zilog site to provide rapid evaluation of in-process and in-field rejects to determine the cause and provide corrective action through a feedback loop into the production, design, and applications process. Zilog is pleased to share product analysis reports on specific products with the customer upon request.

14. Test Site Step-Stress

The process evaluation test sites on the wafer are packaged and subjected to step-stress testing. Any drift in parameters under severe conditions of stress outside the norm is taken as an indication of possible process contamination or variation.

15. Statistical Process Control

Zilog employs Statistical Process Control at all critical process steps. Deviations from norms must be evaluated by a Q/R review board.

16. Perfection Plus Program

Zilog employees actively participate in meetings in which methods which will enable a department to do its job more perfectly are proposed, reviewed, and adopted. Employees who have made suggestions proudly wear the Zilog Perfection Plus pin.

17. Zilog Vendor of the Year Award

Zilog is proud of the many quality and performance awards it has received from its customers. In turn, Zilog makes an annual award to the vendor who has done the best overall job for Zilog.

Zilog's Quality and Reliability Summary

Zilog is proud of its Quality and Reliability programs and is pleased to share this data with its customers. For further information, contact Zilog's Director of R/QA.

Z8®/SUPER8™ MICROCONTROLLER FAMILY

Handbook	Part No	Unit Cost
Z8 Design Handbook (includes the following documents)	DC-8275-03	5.00

Z8 NMOS MCU Microcontroller
Z8600 Z8 MCU 2K 28-Pin Product Specification
Z8601/03/11/13 Z8 MCU 2K/4K Prod. Specification and Protopak
Z8671 MCU with Basic/Debug Interpreter
Z8681/82 Z8 MCU ROMless Product Specification
Z8691 Z8 MCU ROMless Product Specification
Super8 MCU ROMless Product Specification

Z8 CMOS MCU Microcontroller
Z86C08 MCU 2K 18-Pin Product Specification
Z86C00/C10/C20 MCU 4K/8K 28-Pin OTP™Product Specification
Z86C11/ MCU 4K Product Specification
Z86C21/Z86E21/C12 8K/OTP Product Specification
Z86C91 MCU ROMless Product Specification

Z8 Application Notes and Technical Articles
Memory Space and Register Organization
A Programmer's Guide to the Z8 MCU
Z8 Subroutine Library
A Comparison of MCU Units
Z86xx Interrupt Request Registers
Z8 Family Framing

Z8 MCU Technical Manual

Super8 MCU MicrocontrollerTechnical Manual
Z8800/01 MCU ROMless
Z8820 MCU 8K
Z8822 MCU 8K Protopak

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

Z8 Product Specifications, Technical Manuals and Users Guides	Part No	Unit Cost
Z8671 Single Chip Basic Interpreter Basic Debug Software Reference Manual	DC-3149-03	3.00
Z8 Universal Object File Utilities User's Guide	DC-8236-04	3.00
asm S8 Super 8/Z8 Cross Assembler User's Guide	DC-8267-05	3.00
Z86C21/E21 CMOS Z8 8K ROM Preliminary Product Specification	DC-2512-01	N/C
Z86C30 CMOS Z8 8-Bit MCU Microcontroller Preliminary Product Specification	DC-2509-01	N/C
Z86C40/90 ROM/ROMless CMOS Z8 8-Bit Microcontroller Preliminary Product Specification	DC-2510-01	N/C
Z86C08 CMOS Z8 8-Bit Microcontroller Preliminary Product Specification	DC-2527-02	N/C
Z86E08 CMOS Z8 8-Bit Microcontroller Preliminary Product Specification	DC-2542-01	N/C
Z86C09/C19 CMOS Z8 8-Bit Microcontroller Product Specification	DC-2506-01	N/C
Z8602 NMOS Z8 8-Bit MCU Microcomputer Keyboard Controller Preliminary Product Specification	DC-2525-01	N/C
Z8604 NMOS Z8 8-Bit Microcontroller Preliminary Product Specification	DC-2524-02	N/C

Z8 Application Notes and Technical Articles	Part No	Unit Cost
The Z8 MCU In Telephone Answering Systems Applications Note	DC-2514-01	N/C
Z8602 Controls A 101/102 PC/Keyboard Application Note	DC-2521-01	N/C
The Z8 MCU Dual Analog Comparator Application Note	DC-2516-01	N/C
Z86C09/19 Low Cost Z8 MCU Emulator Application Note	DC-2537-01	N/C
Z8 Applications for I/O Port Expansions Application Note	DC-2539-01	N/C
Z86E21 Z8 Low Cost Thermal Printer Application Note	DC-2541-01	N/C

(Additional Application Notes are contained in the above Design Handbook)

Z80®/Z180™/Z280™ MICROPROCESSOR FAMILY

Data Book	Part No	Unit Cost
Z80 Family Data Book (includes the following documents)	DC-2480-01	5.00

Z80 NMOS/CMOS

Z84C00 NMOS/CMOS Z80 CPU Prelim. Product Specification
Z84C01 Z80 CPU w/CGC Product Specification
Z84C10 NMOS/CMOS Z80 DMA Product Specification
Z84C20 NMOS/CMOS Z80 PIO Product Specification
Z84C30 NMOS/CMOS Z80 CTC Product Specification
Z8440/1/2/4 NMOS Z80 SIO Product Specification
Z84C40/1/2/3/4 CMOS Z80 SIO Product Specification
Z84C50 RAM 80 Preliminary Product Specification
Z8470 Z80™DART Product Specification
Z84C80 CMOS Z80 GLU Product Specification
Z84C90 CMOS Z80 KIO™Product Specification
Z80180 Z180 MPU Product Specification
Z280 MPU Preliminary Product Specification

Z80 Application Notes and Technical Articles

Z80 Family Interrupt Structure
Using the Z80 SIO in Async Communications
Using the Z80 SIO with SDLC
Binary Synchronous Comm Using the Z80 SIO
Serial Communication with the Z80A DART
Timing in Interrupt-Based System with Z80 CTC
Interfacing Z80 CPUs to the Z8500 Peripheral Family
Serial Clock Generation using the Z8536CI
A Z80-Based System Using the DMA with the SIO
Zilog Quality and Reliability Report
Package Information
Ordering Information
Literature List
Package Information

Addendum	Part No	Unit Cost
Z80 Family Data Book Addendum	DC-2518-01	N/C

Z80/Z180/Z280 Product Specifications, Technical Manuals and Users Guides	Part No	Unit Cost
Z80 CPU Central Processing Unit Technical Manual	DC-0029-03	3.00
Z80 Family Programmer's Reference Guide	DC-0012-04	3.00
Z80 DMA Direct Memory Access Technical Manual	DC-2013-A0	3.00
Z80 PIO Parallel Input/Output Technical Manual	DC-0008-02	3.00
Z80 CTC Counter/Timer Circuit Technical Manual	DC-0036-02	3.00
Z80 SIO Serial I/O Technical Manual	DC-3033-01	3.00
Z80181 Z180 MPU Microprocessor Unit Technical Manual	DC-8276-02	3.00
Z280 MPU Microprocessor Unit Technical Manual	DC-8224-03	3.00
Z80181 Z181 SAC™Smart Access Controller Preliminary Product Specification	DC-2519-02	N/C
Z84013/15, Z84C13/C15 CMOS IPC™Intelligent Peripheral Controller Preliminary Product Specification	DC-2507-02	N/C
Z84011/C11 PIO Parallel I/O Controller Product Specification	DC-2526-02	N/C
Z84C00 20 MHz Z80 CPU Central Processing Unit Preliminary Product Specification	DC-2523-01	N/C
Z84C50 Z80 RAM 80 Z80 CPU/2K SRAM Preliminary Product Specification	DC-2498-01	N/C

Z80/Z180/Z280 Application Notes	Part No	Unit Cost
Z180/SCC™Serial Communications Controller Interface at 10 MHz Application Note	DC-2520-01	N/C

(Additional Application Notes are contained in the above Databook)

Z8000®/80,000 MICROPROCESSOR FAMILY

Data Book	Part No	Unit Cost
Z8000 Family Data Book (includes the following documents)	DC-2488-01	5.00

Z8000/80,000 NMOS/CMOS Microprocessors

Z160 CPU Product Specification
Z5380 CMOS SCSI Product Specification
Z7220A HPGD Product Specification
Z765A FDC Product Specification
Z8001®/Z8002® CPU Product Specification
Z8010 MMU Product Specification
Z8016 Z-DTC™Product Specification
Z16C20 CMOS Z-BUS® GLU Preliminary Product Specification
Z80C30/Z85C30 CMOS SCC™Product Specification
Z8030/8530 SCC Product Specification
Z8036/Z8536 CIO Product Specification
Z8038/8538 FIO FIFO Product Specification
Z8060/8560 FIFO Product Specification
Z8068/Z9518 Z-DCP Product Specification
Z8516/Z9516 DMA (DTC) Product Specification
Z8581 Clock Generator Controller Product Specification

Application Notes and Technical Articles

Interfacing Z80® CPUs to Z8500 Peripheral Family
Interfacing the Z8500 Peripherals to the 68000
Design Considerations Using Quartz Crystals with Zilog's Components
Using Z8581 Clock Stretches in Z80® CPU Applications
Interfacing Z-BUS Peripherals to the V20/V30/8086/8088
Interfacing the Z-BUS Peripherals Articles Reprint
Using SCC with Z8000 in SDLC Protocol
SCC in Binary Synchronous Communications
Z8000 Development Support
Zilog Quality and Reliability Report
Literature Guide
Ordering Information
Package Information

Z8000 Product Specifications, Technical Manuals and Users Guides	Part No	Unit Cost
Z8000 CPU Central Processing Unit Technical Manual	DC-2010-06	3.00
Z8010 MMU Memory Mapping Unit Technical Manual	DC-2015-A0	3.00
Z8030/Z8530 SCC Serial Communications Controller Technical Manual	DC-2057-06	3.00
Z8036 Z-CIO/Z8536 CIO Counter/Timer and Parallel Input/Output Technical Manual	DC-2091-02	3.00
Z8036 Z8000 Z-CIO Counter/Timer and Parallel Input/Output Product Specification	DC-2014-02	N/C
Z8536 CIO Counter/Timer and Parallel Input/Output Product Specification	DC-2021-03	N/C
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
Z5380 SCSI Small Computer System Interface Preliminary Product Specification	DC-2477-01	N/C
Z80C30/Z85C30 CMOS SCC Serial Communications Controller Product Specification	DC-2442-04	N/C
Z85C80 SCSCI™Serial Communications and Small Computer Interface Preliminary Product Specification	DC-2534-02	N/C
Z16C01/2/3 CPU Central Processing Unit Preliminary Product Specification	DC-2504-02	N/C
Z16C20 CMOS ZBUS GLU General Logic Unit Preliminary Product Specification	DC-2505-02	N/C
Z16C30 CMOS USC™Universal Serial Controller Preliminary Product Specification	DC-2492-02	N/C
Z16C30/Z16C33 CMOS USC/MUSC™Universal Serial Controller Technical Manual	DC-8285-01	3.00
Z16C30/Z16C33 CMOS USC/MUSC Universal Serial Controller Addendum	DC-8285-01A	N/C
Z16C31 IUSC Integrated Universal Serial Controller Advanced Information Specification	DC-2544-01	N/C
Z16C33 CMOS MUSC Mono-Universal Serial Controller Preliminary Product Specification	DC-2517-02	N/C
Z16C35 CMOS ISCC™Integrated Serial Communications Controller Product Specification	DC-2515-03	N/C
Z16C35 ISCC Integrated Serial Communications Controller Technical Manual	DC-8286-01	3.00
Z16C35 ISCC Integrated Serial Communications Controller Addendum	DC-8286-01A	N/C
Z16C50 DDPLL™Dual Digital Phase Locked Loop Preliminary Product Specification	DC-2540-01	N/C
Z85130 ESCC™Enhanced Serial Communications Controller Preliminary Product Specification	DC-2543-01	N/C
Z16C30 Using the USC in Military Applications Application Note	DC-2536-01	N/C
Z16C35 ISCC Interface to Intel and Motorola Microprocessors Application Note	DC-2522-01	N/C

COMPONENTS MILITARY LITERATURE

Military Products Binder	Part No	Unit Cost
Zilog Military Products Binder (includes the following documents)	DC-5498-01	8.00

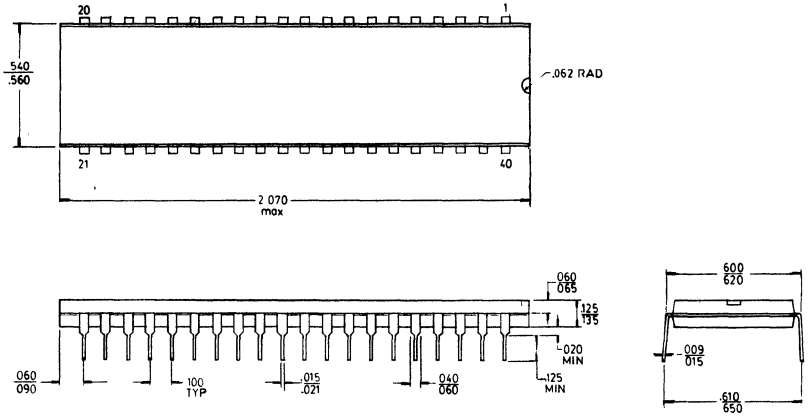
Military Specifications	Part No	Unit Cost
Z8681 ROMless Microcomputer Military Product Specification	DC-2392-02	N/C
Z8001/8002 Military Z8000 CPU Central Processing Unit Military Product Specification	DC-2342-03	N/C
Z8581 Military CGC Clock Generator and Controller Military Product Specification	DC-2346-01	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
Z16C01/2 CPU Central Processing Unit Military Product Specification	DC-2532-01	N/C
Z80180 Z180 MPU Microprocessor Unit Military Product Specification	DC-2532-01	N/C

Note: Military Product Specifications may be ordered individually at no charge.

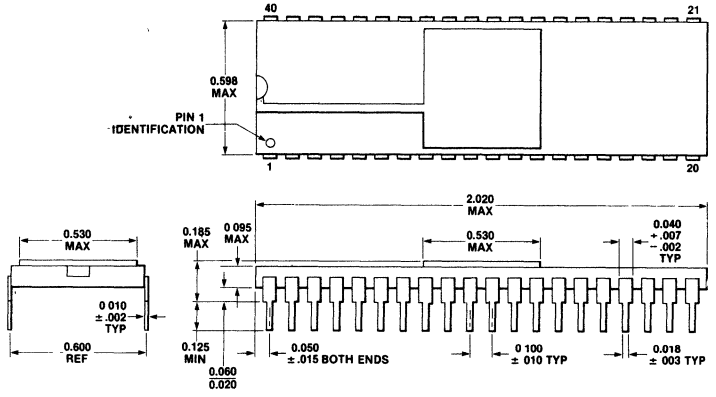
GENERAL LITERATURE

Catalogs, Handbooks and Users Guides	Part No	Unit Cost
Superintegration Shortform Catalog 1991	DC-5472-06	N/C
Quality and Reliability Report	DC-2475-06	N/C
Superintegration Products Guide	DC-5499-03	N/C
The Handling and Storage of Surface Mount Device's User Guide	DC-5500-02	N/C
Support Products Summary	DC-2545-02	N/C

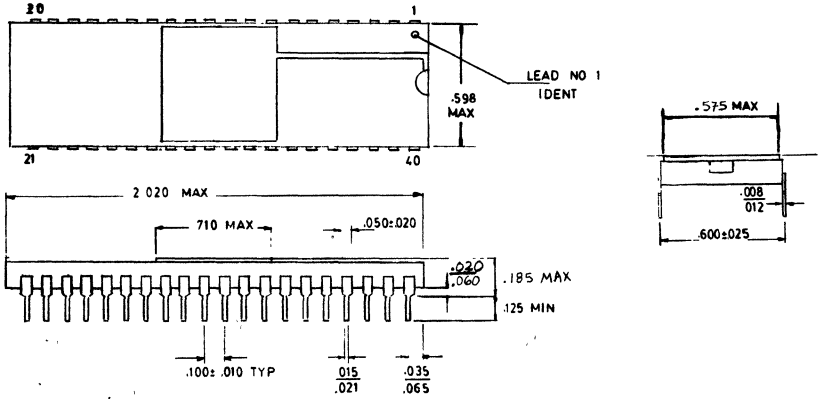
PACKAGE INFORMATION



40-Pin Plastic Dual In-Line Package (PDIP)

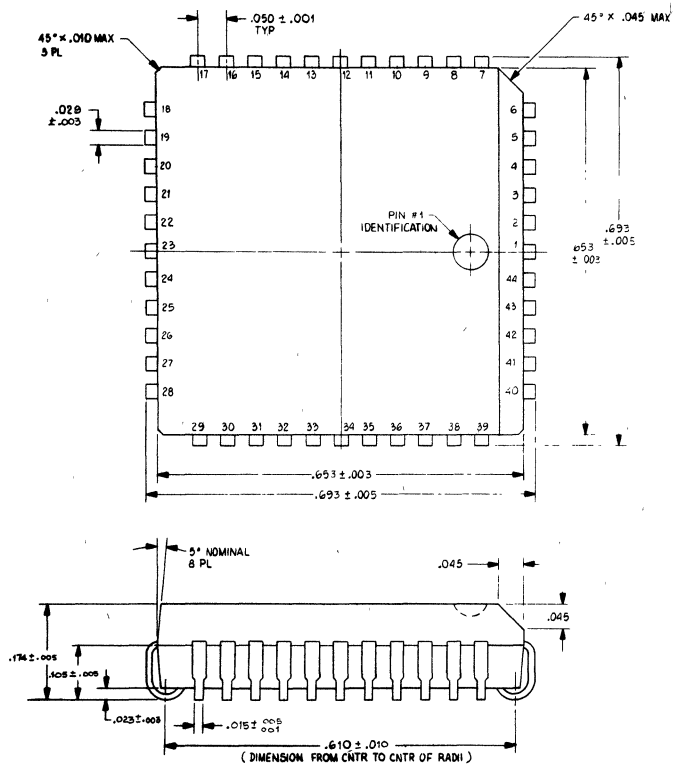


40-Pin Ceramic Package

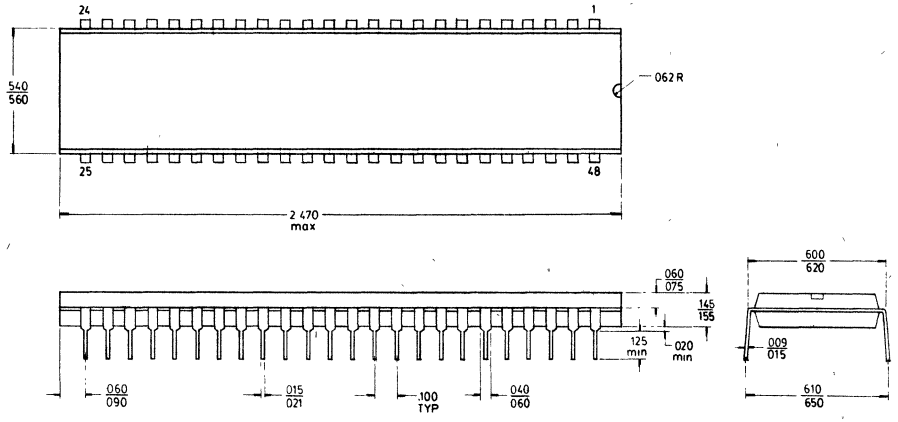


40-Pin Ceramic Dual In-Line Package (DIP)

PACKAGE INFORMATION (Continued)

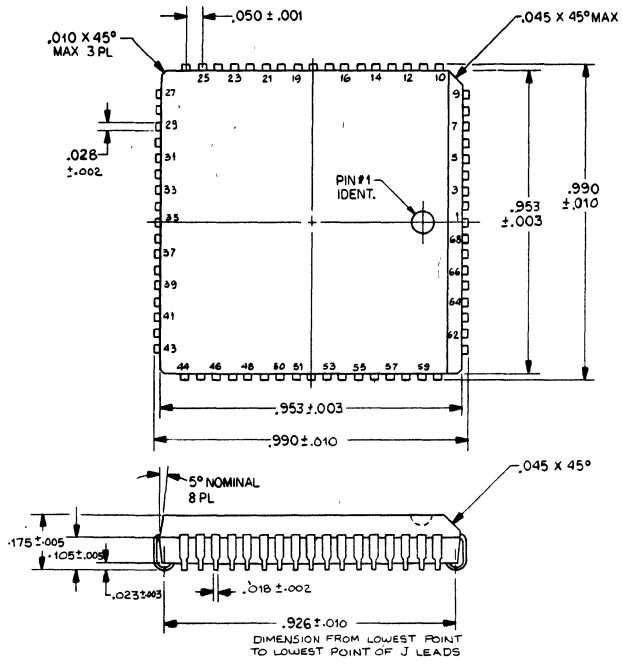


44-Pin Plastic Chip Carrier

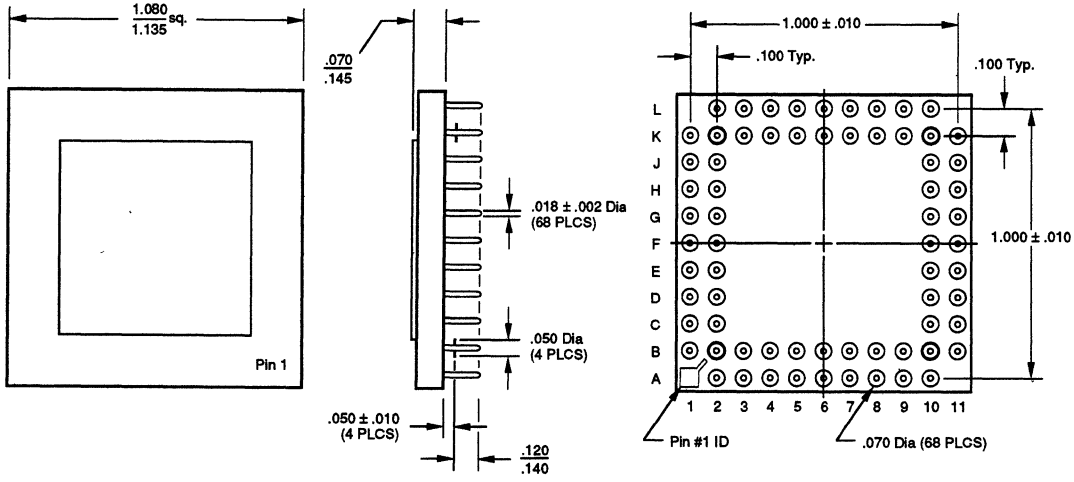


48-Pin Plastic Dual-In Line Package (PDIP)

PACKAGE INFORMATION (Continued)

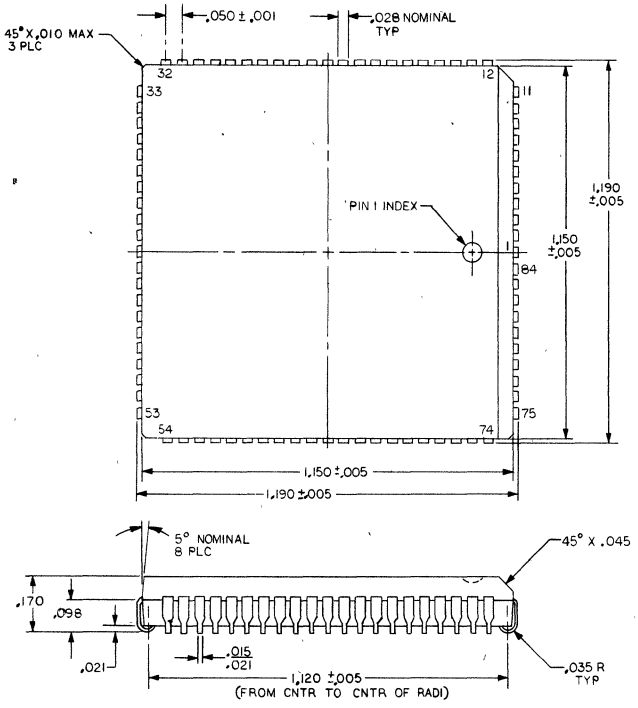


68-Pin Plastic Chip Carrier (PLCC)

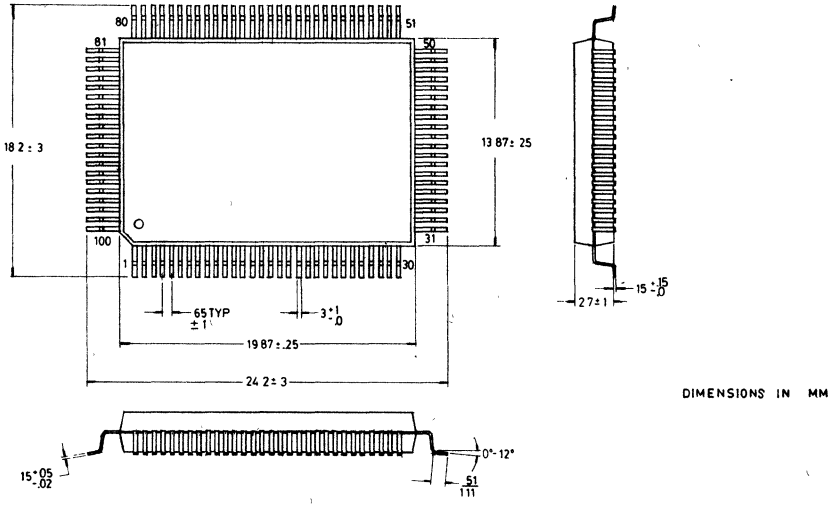


68-Pin Pin Grid Array (PGA)

PACKAGE INFORMATION (Continued)



84-Pin Plastic Chip Carrier (PLCC)



DIMENSIONS IN MM

100-Pin Quad Flat Pack (QFP)

ORDERING INFORMATION

Z16C30

68-pin PLCC	68-pin PGA
Z16C3010VSC	Z16C3010GEE
Z16C3010VEC	

Z16C31

68-pin PLCC

Z16C3120VSC

Z16C33

68-pin PLCC

Z16C3310VSC

Z16C35

68-pin PLCC

Z16C3510VSC
Z16C3516VSC

Z16C50

28-pin P-DIP

Z16C5010PSC
Z16C5020PSC

Z5380 1.5MB/Sec

40-pin P-DIP	44-pin PLCC
Z0538010PSC	Z0538010VSC

Z85130

40-pin P-DIP	44-pin PLCC
Z8513010PSC	Z8513010VSC
Z8513016PSC	Z8513016VSC

Z85230

40-pin P-DIP	44-pin PLCC
Z8523010PSC	Z8523010VSC
Z8523016PSC	Z8523016VSC

Z80C30

40-pin DIP	44-pin PLCC
Z80C3008PSC	Z80C3008VSC
Z80C3010PSC	Z80C3010VSC

Notes:

For military grade devices and the package types other than listed above, please contact your local Zilog sales office.

Please check the availability before placing order.

Z85C30

40-pin DIP	44-pin PLCC
Z85C3008PSC	Z85C3008VSC
Z85C3008PEC	Z85C3008VEC
Z85C3008CEE	Z85C3010VSC
Z85C3010PSC	Z85C3010VEC
Z85C3010PEC	Z85C3016VSC
Z85C3010CEE	
Z85C3016PSC	

Z8030

40-pin DIP	44-pin PLCC
Z0803006PSC	Z0803006VSC
Z0803006DSE	Z0803008VSC
Z0803008PSC	
Z0803008DSE	

Z8530

40-pin DIP	44-pin PLCC
Z0853004PSC	Z0853006VSC
Z0853006PSC	Z0853008VSC
Z0853006PEC	
Z0853006DSE	
Z0853008PSC	
Z0853008DSE	
Z0853008DEA	

Z85C80

68-pin PLCC

Z85C8010VSC

Z80181

100-pin QFP

Z8018110FEC
Z8018112FEC

Z84013/C13 & Z84015/C15

84-pin PLCC	100-pin QFP
Z8401306VEC	Z8401506FEC
Z8401310VEC	Z8401510FEC
Z84C1306VEC	Z84C1506FEC
Z84C1310VEC	Z84C1510FEC

ORDERING INFORMATION (Continued)

Z8440/1/2/4 & Z84C40/1/2/3/4 Z80 SIO NMOS/CMOS

SIO/0

NMOS 40-pin DIP	CMOS 40-pin DIP
Z0844004DSE	Z84C4004DEE*
Z0844004PSC	Z84C4004PEC*
Z0844006DSE	Z84C4006DEE
Z0844006PSC	Z84C4006PEC
	Z84C4008DEE
	Z84C4008PEC
	Z84C4010DEE
	Z84C4010PEC

SIO/1

NMOS 40-pin DIP	CMOS 40-pin DIP
Z0844104DSE	Z84C4104DEE*
Z0844104PSC	Z84C4104PEC*
Z0844106DSE	Z84C4106PEC
Z0844106PSC	Z84C4108PEC
	Z84C4110PEC

SIO/2

NMOS 40-pin DIP	CMOS 40-pin DIP
Z08442004DSE	Z84C4204DEE*
Z0844204PSC	Z84C4204PEC*
Z0844206DSE	Z84C4206DEE
Z0844206PSC	Z84C4206PEC
	Z84C4208DEE
	Z84C4208PEC
	Z84C4210DEE
	Z84C4210PEC

SIO/3

CMOS 44-pin QFP
Z84C4306FEC
Z84C4308FEC
Z84C4310FEC

SIO/4

NMOS 44-pin PLCC	CMOS 44-pin PLCC
Z084440VSC	Z84C4404VEC*
Z0844406VSC	Z84C4406VEC
	Z84C4408VEC
	Z84C4410VEC

Notes:

* 4 MHz CMOS SIO will be phased out and updraded to 6 MHz in 1991. Please contact Zilog for availability

For military grade devices and the package types other than listed above, please contact your local Zilog sales office.

Please check the availability before placing order.

ORDERING INFORMATION (Continued)

CODES

Package

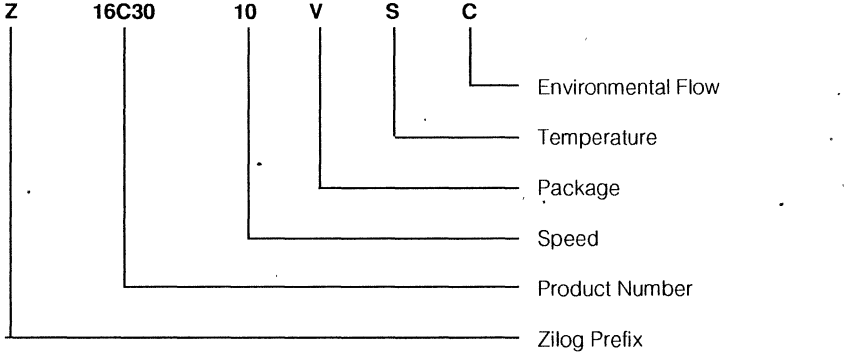
Preferred
P = Plastic DIP
V = Plastic Leaded Chip Carrier

Longer Lead Time
D = Cerdip
C = Ceramic
F = Plastic Quad Flat Pack
G = Ceramic PGA (Pin Grid Array)
L = Ceramic LCC

Temperature
S = 0°C TO +70°C (Standard)
E = -40°C TO +85°C (Extended)
M = -55°C TO +125°C (Military)

Environmental
C = Plastic Standard
E = Hermetic Standard
A = Hermetic Stressed
B = 833 Class B Military
D = Plastic Stressed
J = Jan 38510 Military

Example: Z16C3010VSC is a USC, 10 MHz, Plastic PLCC, 0°C to +70°C, Plastic Standard Flow.



ZILOG DOMESTIC SALES OFFICES AND TECHNICAL CENTERS

CALIFORNIA
Agoura 818-707-2160
Campbell 408-370-8120
Tustin 714-838-7800

COLORADO
Boulder 303-494-2905

FLORIDA
Largo 813-585-2533

GEORGIA
Norcross 404-448-9370

ILLINOIS
Schaumburg 708-517-8080

NEW HAMPSHIRE
Nashua 603-888-8590

NEW JERSEY
Clark 201-382-5700

NORTH CAROLINA
Raleigh 919-790-7706

OHIO
Independence 216-447-1480

PENNSYLVANIA
Ambler 215-653-0230

TEXAS
Dallas 214-987-9987

WASHINGTON
Seattle 206-523-3591

INTERNATIONAL SALES OFFICES

CANADA
Toronto 416-673-0634

GERMANY
Munich 49-89-672-045
Sömmerda 37-626-23906

JAPAN
Tokyo 81-3-587-0528

HONG KONG
Kowloon 852-7238979

KOREA
Seoul 82-2-552-5401

SINGAPORE
Singapore 65-2357155

TAIWAN
Taipei 886-2-741-3125

UNITED KINGDOM
Maidenhead 44-628-392-00

Zilog, Inc. 210 E. Hacienda Ave., Campbell, CA 95008-6600, Tel: (408) 370-8000, FAX: 408-370-8056/8027

DC-2503-02