ZiLOG

Serial Communication Controllers

Includes Specifications and Application Notes for the following parts:

Z8030/Z8530
Z80C30/Z85C30
Z80230
Z85230
Z85233
Z85C80
Z16C35/Z85C35

Product Specifications Databook

ZiLOG

Serial Communication Controllers

Includes Specifications for the following parts:

- Z8030/Z8530
- Z80C30/Z85C30
- Z80230
- Z85230
- Z85233
- Z85C80
- Z16C35/Z85C35
- Application Notes

Databook

DC 8316-01

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

SCC DATABOOK

TABLE OF CONTENTS

Introduction

INTRODUCTION

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

Zilog's Serial Communication Controllers products are suitable for a broad range of applications, from general-purpose use through high-end LANs. Whichever device you choose, you'll find a comprehensive feature set, along with easy-to-use evaluation boards and software tools to speed your design time to production.

Z8030/Z8530 NMOS Serial Communication Controller
The SCC Serial Communication Controller is a dual-channel, multi-protocol datacommunications 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, and may be software configured to satisfy a wide variety of serial communication applications. On-chip baud rate generators, digital phase-locked loops, and crystal oscillators reduce the need for external logic and make the SCC ideal for both 8- and 16-bit datacom applications.

Z80C30/Z85C30 CMOS Serial Communication Controller
The Z80C30/Z85C30 CMOS SCC Serial Communication Controller is a CMOS version of the industry standard NMOS SCC. It is a dual channel, multi-protocol datacommunications peripheral that easily interfaces to CPUs with either multiplexed or non-multiplexed address/data buses. The advanced CMOS process offers lower power consumption, higher performance and superior noise immunity.

Z80230 CMOS Z-Bus® ESCC™ Enhanced Serial Communication Controller
The Zilog Enhanced Serial Communication Controller is a pin and software compatible CMOS member of the SCC family designed for use with Zilog Z-Bus. The ESCC is a dual-channel, full-duplex datacommunications 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. There are many improvements in the ESCC that significantly reduce CPU overhead, such as the addition of a 4-byte transmit FIFO and an 8-byte receive FIFO.

Z85230 CMOS ESCC™ Enhanced Serial Communication Controller
The Zilog Enhanced Serial Communication Controller is a pin and software compatible CMOS member of the SCC family designed for use with non-multiplexed buses. The ESCC is a dual-channel, full-duplex datacommunications controller capable of supporting a wide range of popular protocols. Other improvements for the Z85230/Z80230 are SDLC link layer improvements, multiprotocol support, easy modem interface, and deeper FIFOs that reduce system supervision and improve SCC system throughout.

Z85233 CMOS EMSCC™ Enhanced Mono Serial Communication Controllers
The Z85233 Enhanced Mono Serial Communication Controller is a software compatible CMOS member of the SCC family introduced by Zilog in 1981. The Z85233 EMSCC is a single channel version (Channel A) of Zilog's Z85230 ESCC. Based on Zilog's unique Superintegration™ Technology, the EMSCC is compatible with designs using Zilog SCC and ESCC™ to receive and transmit data.

Z85C80 CMOS SCSCI Serial Computer System Interface Controller
The Serial Communication and Small Computer Interface (SCSCI) combines Zilog's 10 MHz SCC with a 3 Mbps SCSI. Eliminating bus interface while reducing board area and complexity, it is a popular choice for disk drives, multimedia peripherals, and any computer system having a SCSI interface defined by the IEEE X3.131-1986 standard.

Z16C35 CMOS ISCC Integrated Serial Communication Controller
The Z16C35 Integrated Serial Communication Controller (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 Communication Controller (SCC) cell. The results include a higher average data throughput rate, high-speed, elimination of the need for a discrete SCC and DMA controller in a design, and significant overall space and cost savings.

Z8030/Z8530

Z-BUS SCC SERIAL COMMUNICATION CONTROLLER

FEATURES

- Z8530 Optimized for Non-Multiplexed Bus Microprocessors.
- Z8030 Optimized for Multiplexed Bus Microprocessors.
- Two Independent, 0 to 2 Mbit/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 Loop-Back and Auto Echo Modes.
- Supports T1 Digital Trunk.
- Speeds 4, 6, and 8 MHz.

GENERAL DESCRIPTION

The SCC Serial Communications Controller is a dual-channel, multi-protocol datacommunications 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 phase-locked loops, and crystal oscillators that dramatically reduce the need for external logic.

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.

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

Power connections follow conventional descriptions below:

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

GENERAL DESCRIPTION (Continued)

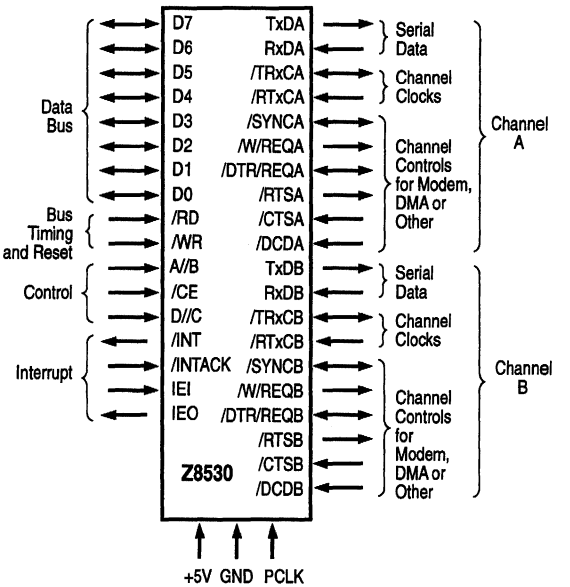


Figure 1a. Z8530 Pin Functions

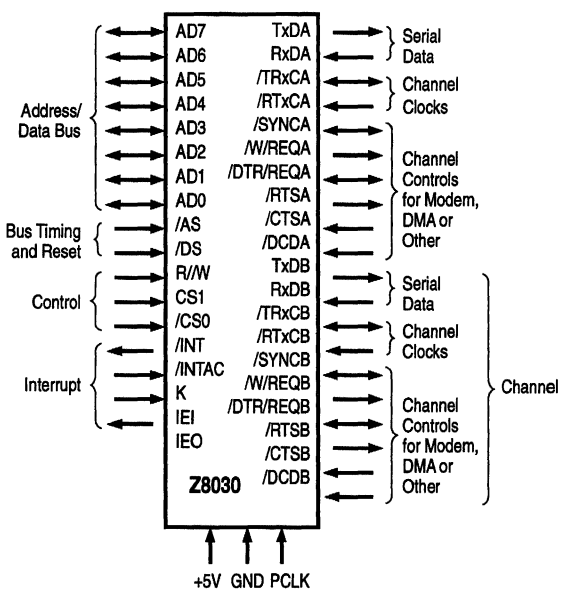


Figure 1b. Z8030 Pin Functions

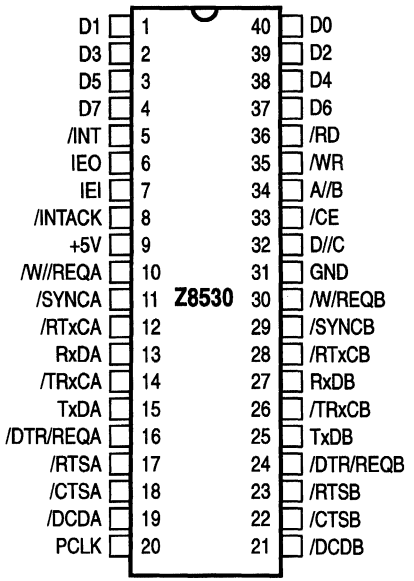


Figure 2a. Z8530 40-Pin DIP Pin Assignments

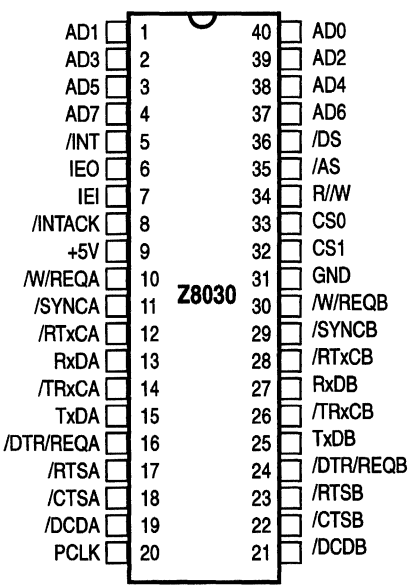


Figure 2b. Z8030 40-Pin DIP Pin Assignments

GENERAL DESCRIPTION (Continued)

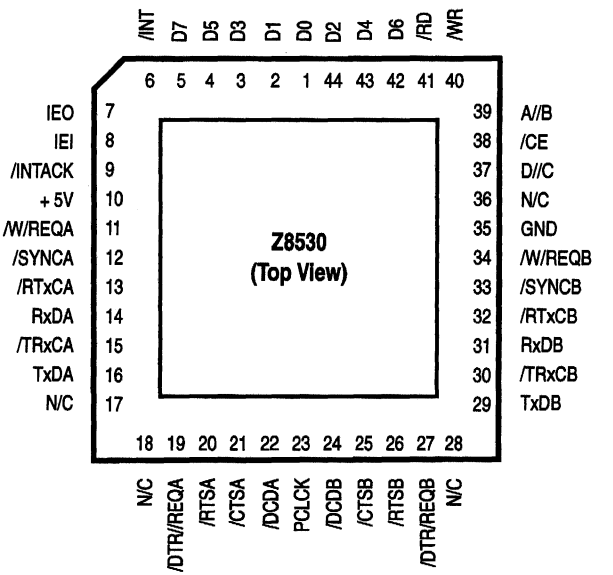


Figure 2c. Z8530 44-Pin PLCC Pin Assignments

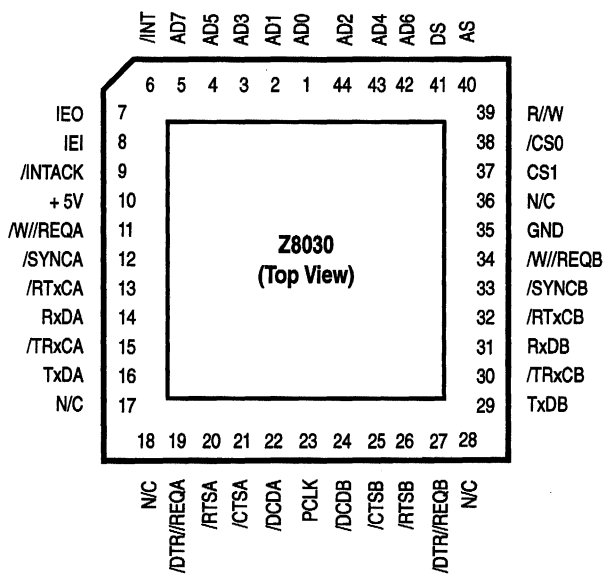


Figure 2d. Z8030 44-Pin PLCC Pin Assignments

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.

/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 detects pulses on these inputs and can interrupt the CPU on both logic level transitions.

/DCDA, /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 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 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.

/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 input 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 is set (Figure 11), 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 (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 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 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.

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.

Z8530

A//B *Channel A/Channel B Select* (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* (bi-directional, tri-state). These lines carry data and commands 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.

Z8030

AD7–AD0 *Address/Data Bus* (bi-directional, active High, tri-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 functional capabilities of the SCC can be described from two different points of view: as a datacommunications device, it transmits and receives data in a wide variety of datacommunications protocols; as a microprocessor peripheral, the SCC offers valuable features such as vectored interrupts, polling, and simple handshake capability.

Datacommunications Capabilities. The SCC provides two independent full-duplex channels, programmable for use in any common asynchronous or synchronous data-communication protocol. The following description and Figure 3 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-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.

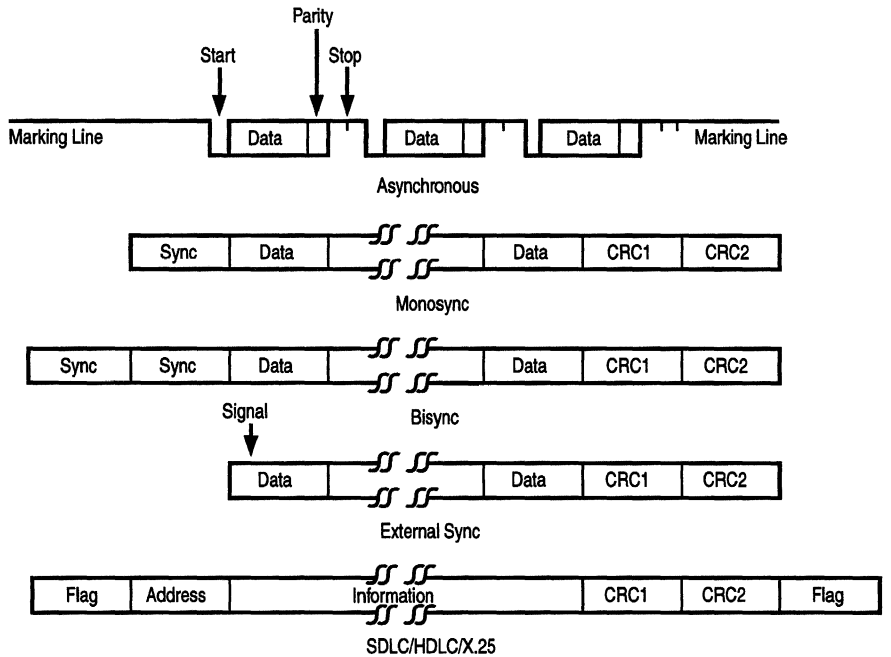


Figure 3. Some SCC Protocols

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 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 synchronous 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 (X16 + XI5 + 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 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.

FUNCTIONAL DESCRIPTION

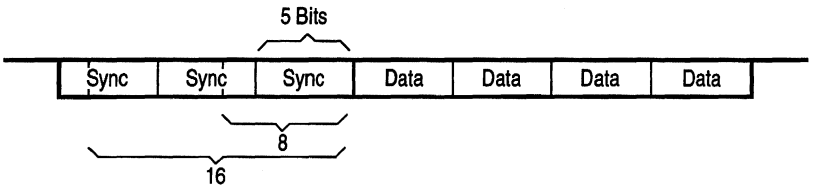


Figure 4. Detecting 5- or 7-Bit Synchronous Characters

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 /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 an 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 through 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.

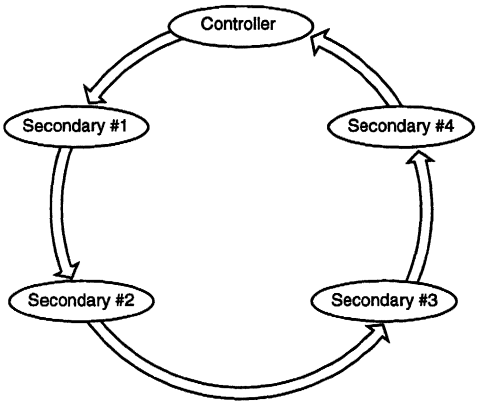


Figure 5. An SDLC Loop

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 append messages to the first secondary station message 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 start-up, 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 through 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 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 through the /TRxC pin (if this pin is not being used as an input).

FUNCTIONAL DESCRIPTION (Continued)

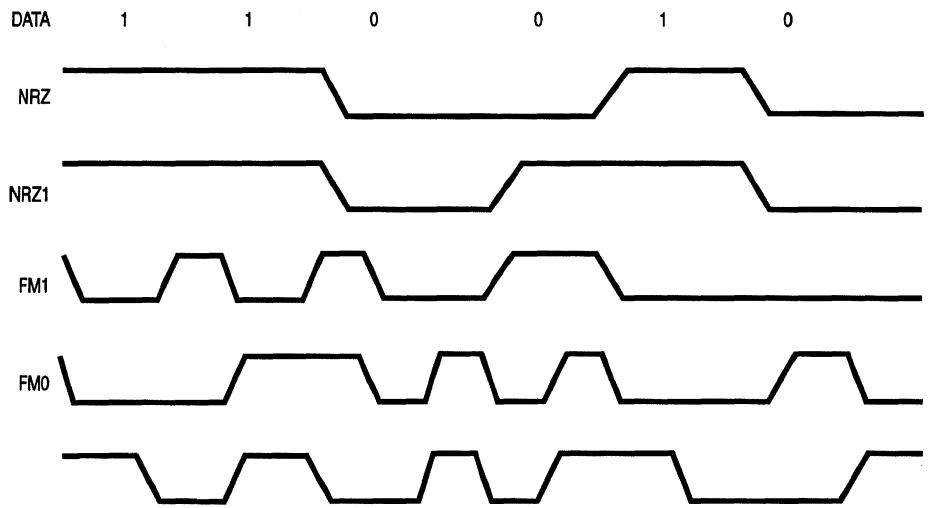


Figure 6. Data Encoding Methods

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 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 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 through 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 non-vectored), 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.

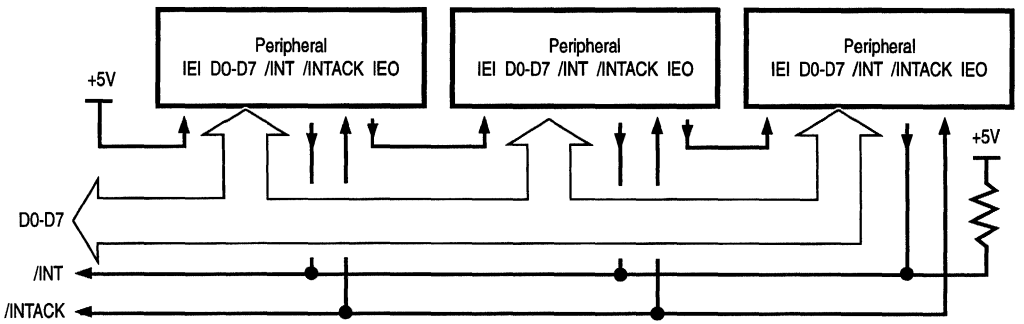


Figure 7. Interrupt Schedule

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 (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 /INT. The CPU then responds with /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 /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.

FUNCTIONAL DESCRIPTION (Continued)

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 /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 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 /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 SCC 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.

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 non-multiplexed 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 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).

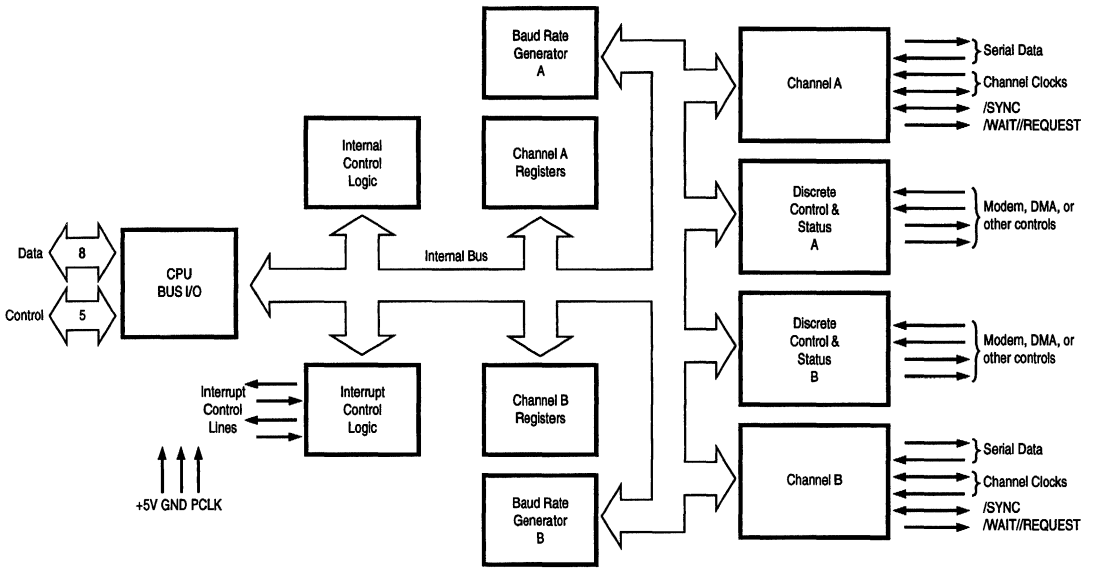


Figure 8. Block Diagram of SCC Architecture

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 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, outgoing data is routed through one of four main paths before it is transmitted from the Transmit Data output (TxD).

ARCHITECTURE (Continued)

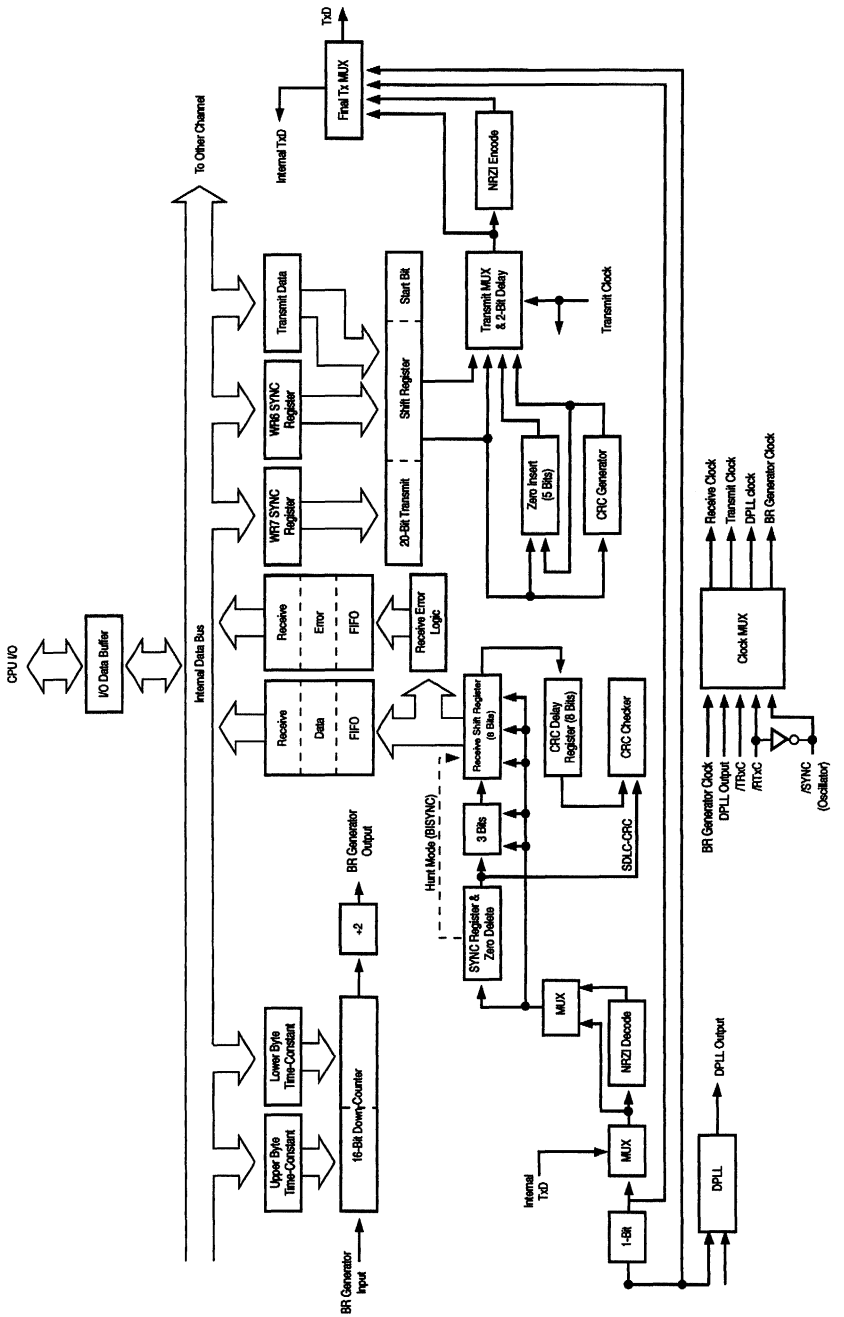


Figure 9. Data Path

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 Command 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 Control
WR6	Sync Characters or SDLC Address
WR7	Sync Character or SDLC Flag
WR8	Transmit Buffer
WR9	Master Interrupt Control and Reset (Accessed Through Either Channel)
WR10	Miscellaneous, Transmitter/Receiver Control Bit
WR11	Clock Mode Control
WR12	Lower Byte of Baud Rate Generator Time Constant
WR13	Upper Byte of Baud Rate Generator Time
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 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 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.

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

Read Registers. The SCC contains eight read registers (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).

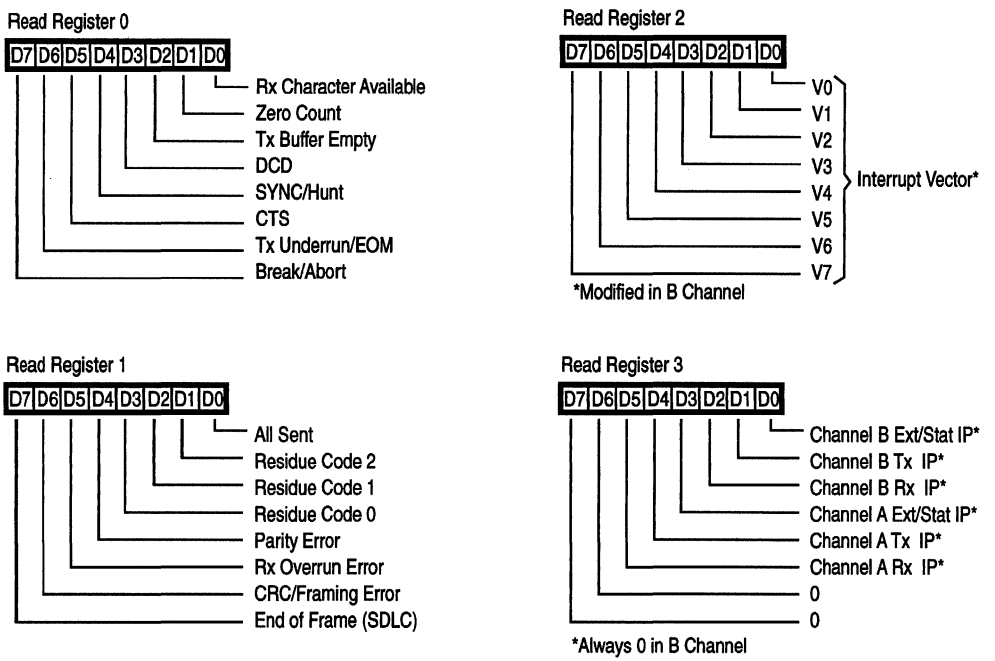


Figure 10.1. Read Register Bit Functions

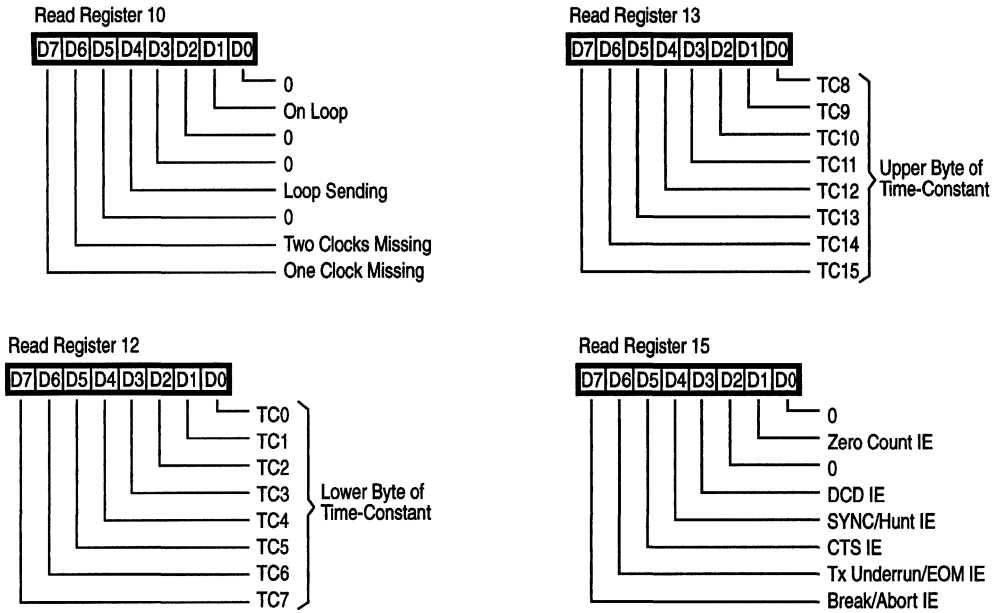


Figure 10.2. Read Register Bit Functions

PROGRAMMING (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 11 shows the format of each write register.

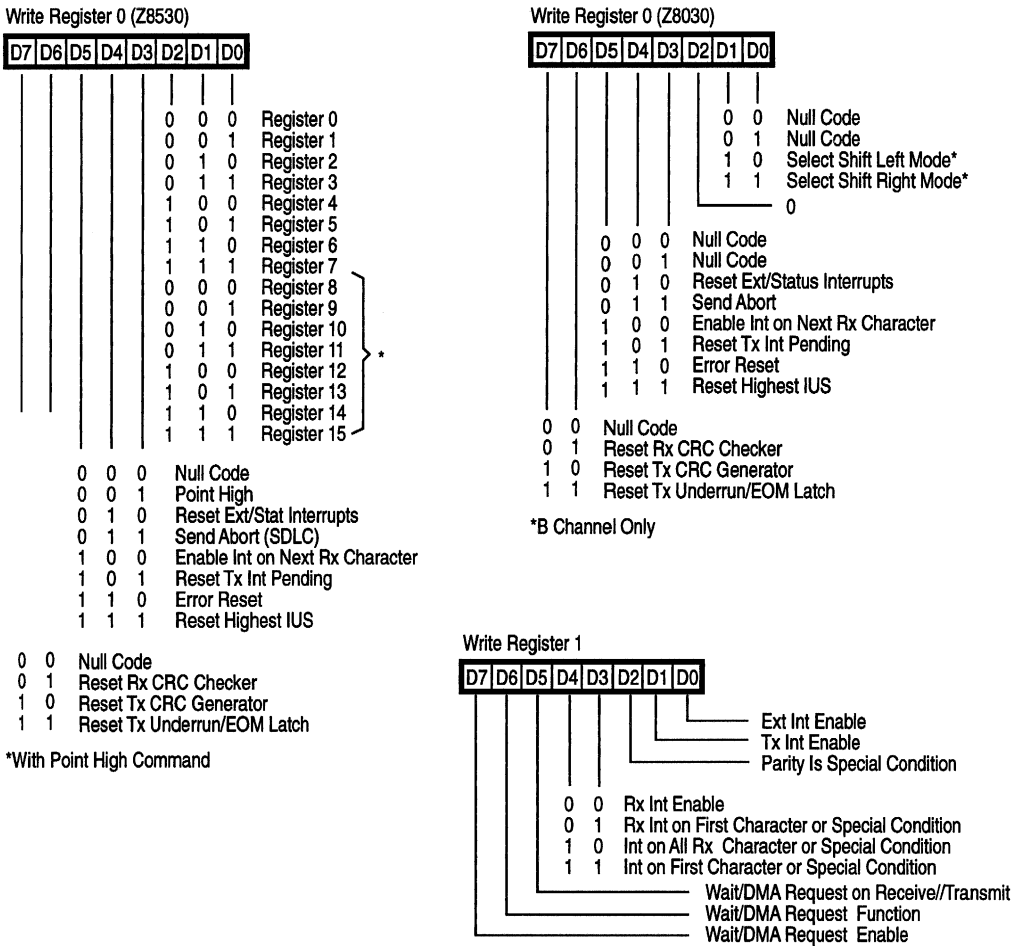


Figure 11.1. Write Register Bit Functions

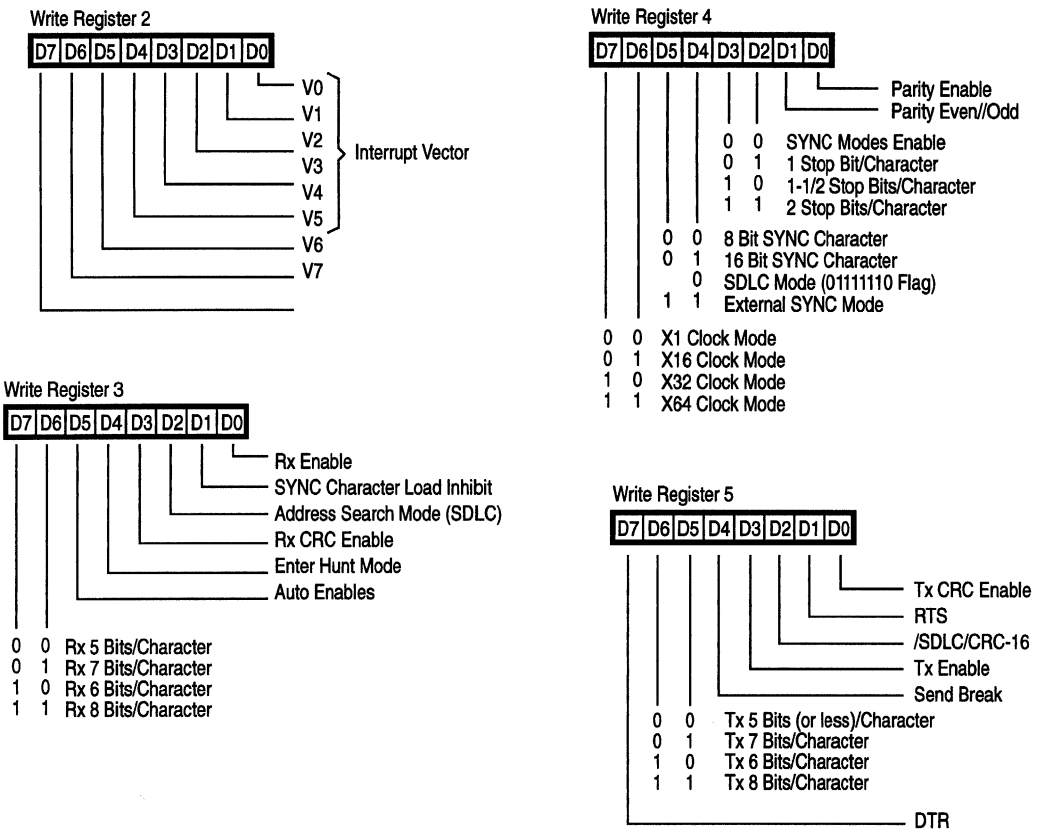


Figure 11.2. Write Register Bit Functions (Continued)

PROGRAMMING (Continued)

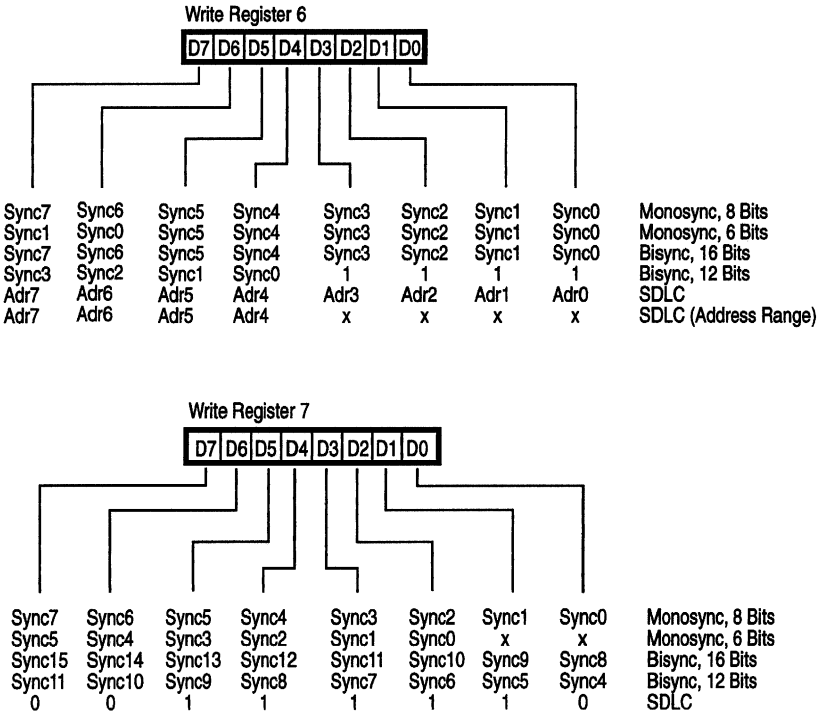


Figure 11.3. Write Register Bit Functions (Continued)

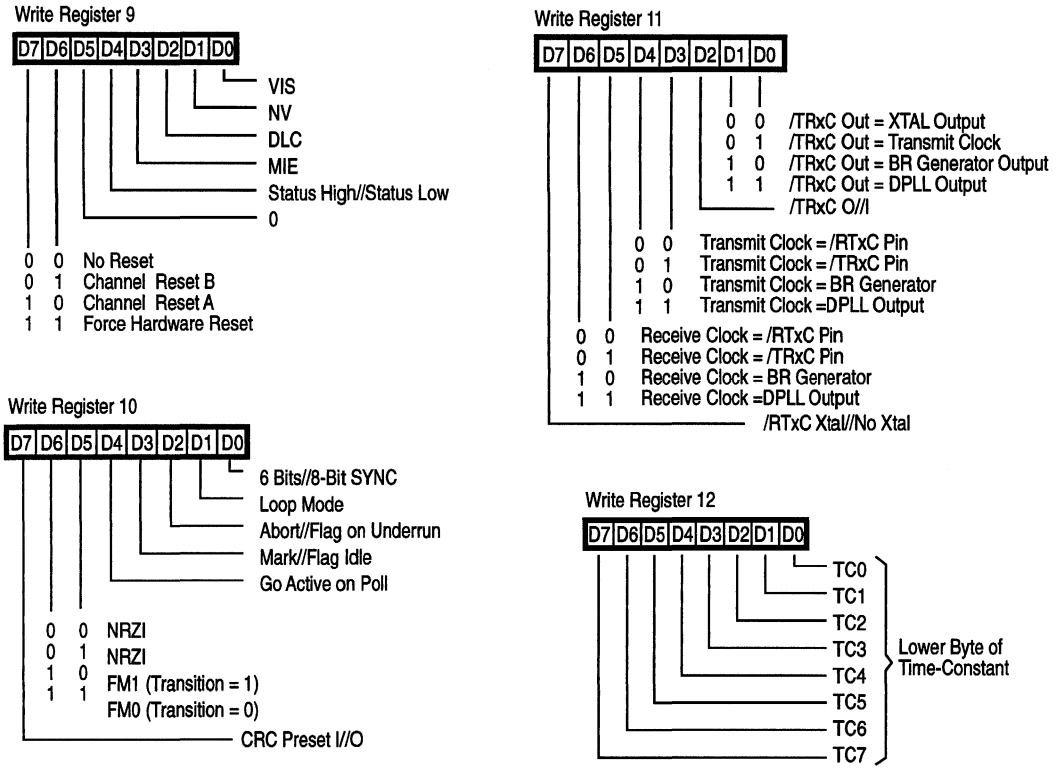


Figure 11.4. Write Register Bit Functions (Continued)

PROGRAMMING (Continued)

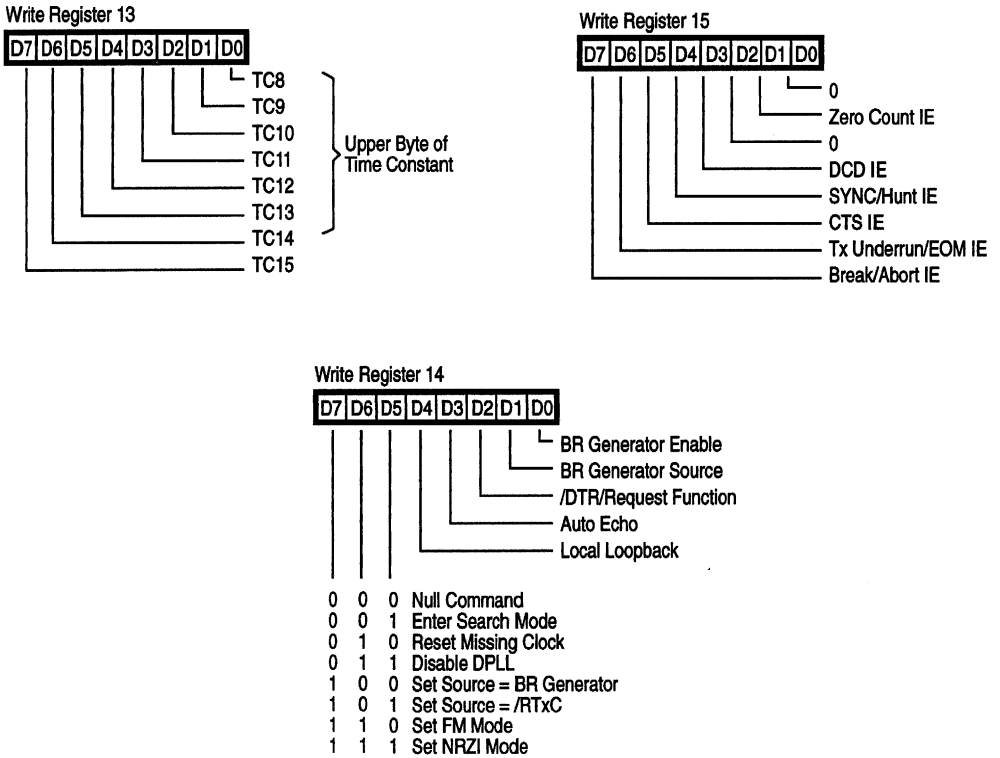


Figure 11.5. Write Register Bit Functions (Continued)

Z8530 Timing

The SCC generates internal control signals from /WR and /TFD 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 /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 12 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.

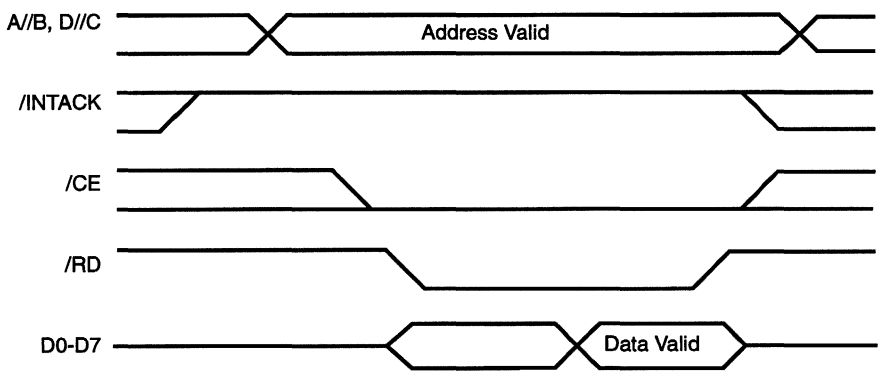


Figure 12. Read Cycle Timing

PROGRAMMING (Continued)

Write Cycle Timing. Figure 13 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.

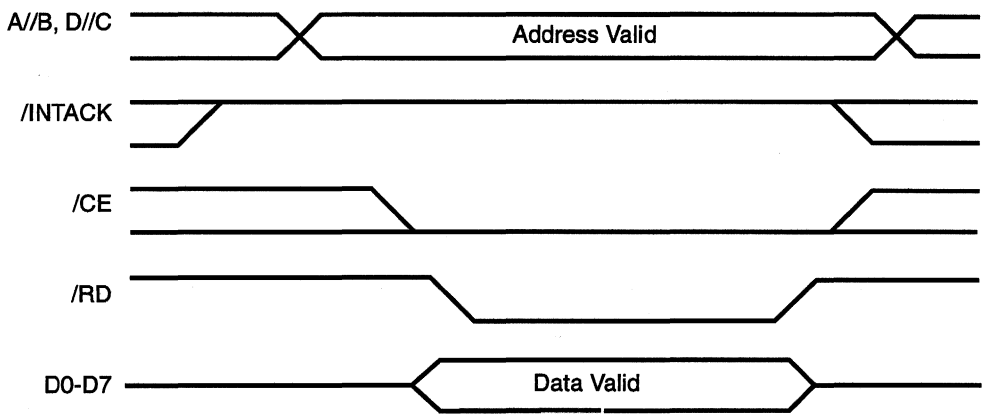


Figure 13. Write Cycle Timing

Interrupt Acknowledge Cycle Timing. Figure 14 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.

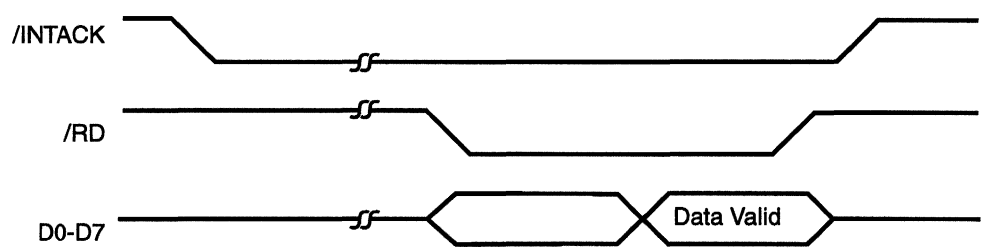


Figure 14. Interrupt Acknowledge Cycle Timing

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 15 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.

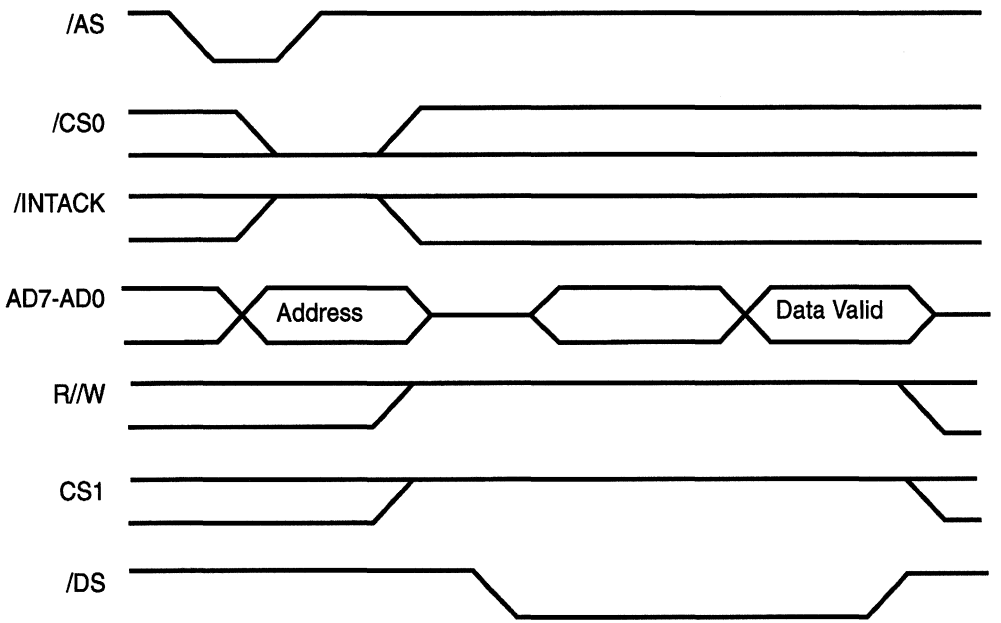


Figure 15. Read Cycle Timing

PROGRAMMING (Continued)

Write Cycle Timing. Figure 16 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.

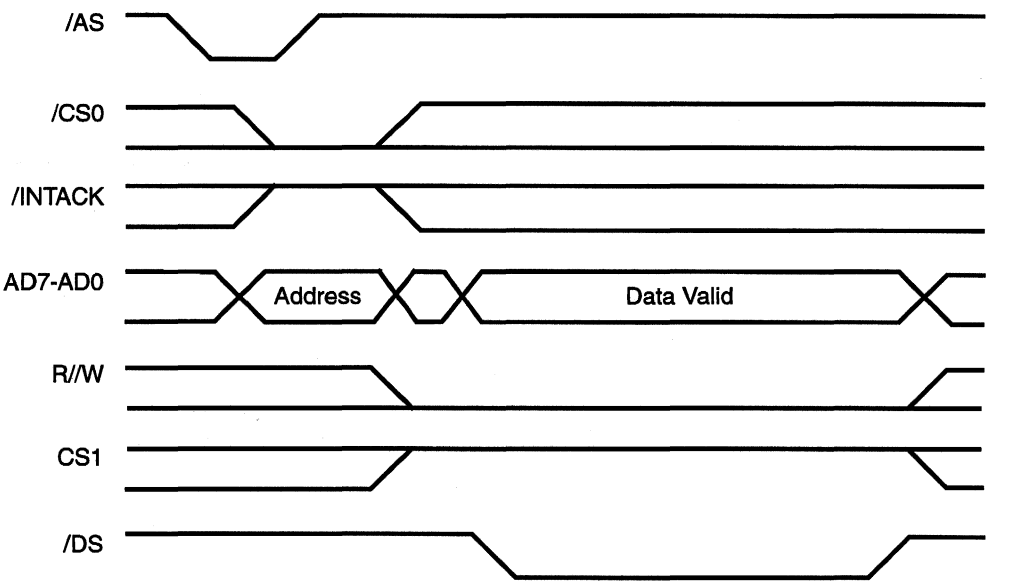


Figure 16. Write Cycle Timing

Interrupt Acknowledge Cycle Timing. Figure 17 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 may be programmed to respond to RD Low by placing its interrupt vector on D7-D0, it then internally sets the appropriate Interrupt Under Service latch.

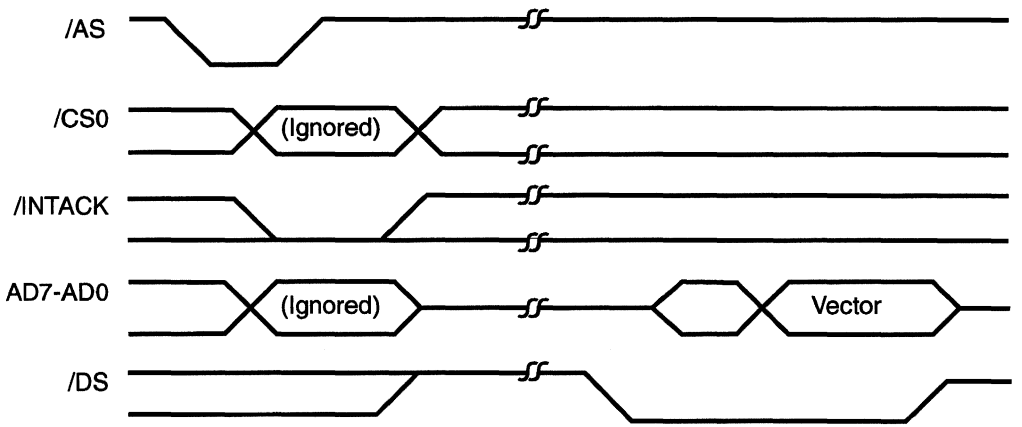


Figure 17. Interrupt Acknowledge Cycle Timing

ABSOLUTE MAXIMUM RATINGS

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

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

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

STANDARD TEST CONDITIONS

The DC characteristics and capacitance section below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin.

Standard conditions are as follows:

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

All AC parameters assume a load capacitance of 50 pF max.

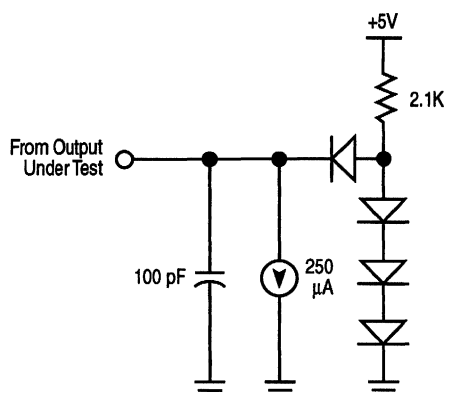


Figure 18. Standard Test Load

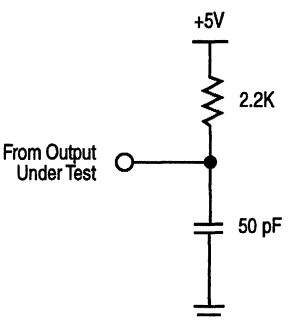


Figure 19. Open-Drain Test Load

CAPACITANCE

Symbol	Parameter	Min	Max	Unit	Condition
C_{IN}	Input Capacitance		10	pF	
C_{OUT}	Output Capacitance		15	pF	
$C_{I/O}$	Bi-directional Capacitance		20	pF	
	Gate Count		6000		

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

DC CHARACTERISTICS (Z8530/8030)

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 µA
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = +2.0 mA
I_{IL}	Input Leakage		±10.0	µA	$0.4 \le V_{IN} \le +2.4V$
I_{OL}	Output Leakage		±10.0	µA	$0.4 \le V_{OUT} \le +2.4V$
I_{CC}	V_{CC} Supply Current		250	mA	

Notes:
V_{CC} = 5V ±5% unless otherwise specified.

Z8530 AC CHARACTERISTICS

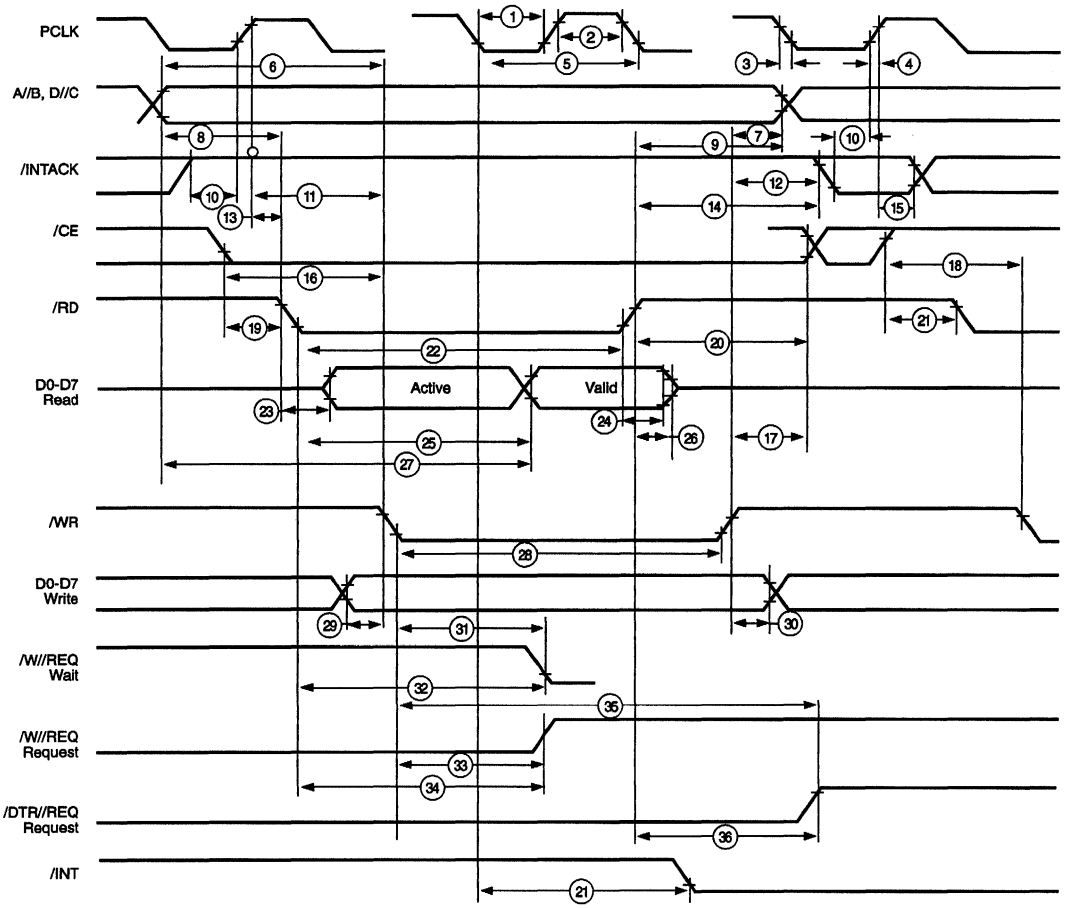


Figure 20. Read and Write Timing

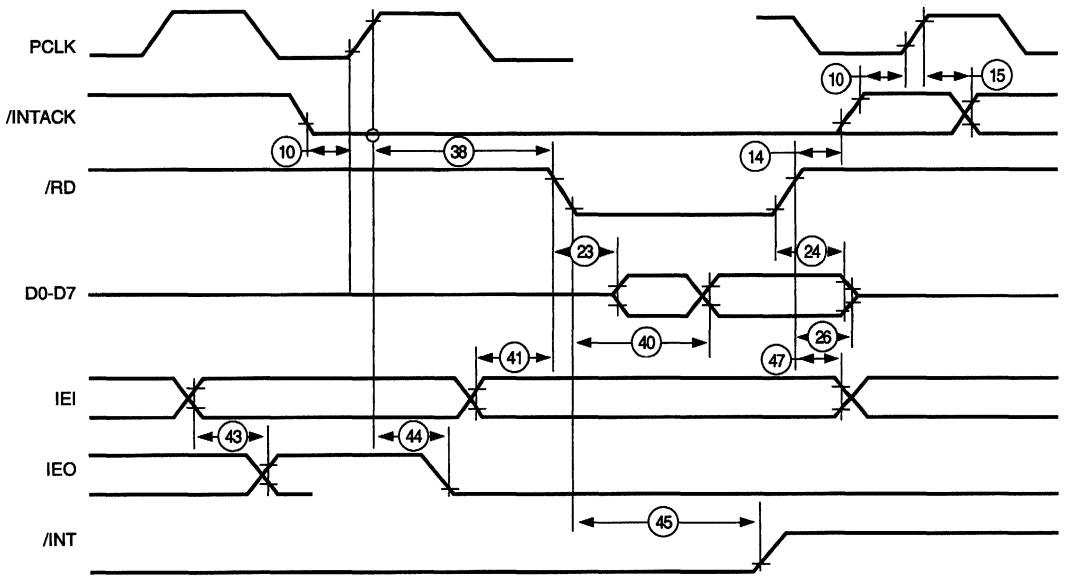


Figure 21. Interrupt Acknowledge Timing

Z8530 AC CHARACTERISTICS (Continued)

No.	Symbol	Parameter	4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	8 MHz Min	8 MHz Max	Notes [6]
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 /WR Fall Setup Time	80		80		70		
7	ThA(WR)	Address to /WR Rise Hold Time	0		0		0		
8	TsA(RD)	Address to /RD Fall Setup Time	80		80		70		
9	ThA(RD)	Address to /RD Rise Hold Time	0		0		0		
10	TslA(PC)	/INTACK to PCLK Rise Setup Time	10		10		10		
11	TslAi(WR)	/INTACK to /WR Fall Setup Time	200		160		145		[1]
12	ThlA(WR)	/INTACK to /WR Rise Hold Time	0		0		0		
13	TslAi(RD)	/INTACK to /RD Fall Setup Time	200		160		145		[1]
14	ThlA(RD)	/INTACK to /RD Rise Hold Time	0		0		0		
15	ThlA(PC)	/INTACK to PCLK Rise Hold Time	100		100		85		
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	100		70		60		
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	100		70		60		[1]
22	TwRDl	/RD Low Width	240		200		150		
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	TdRDf(DR)	/RD Fall to Read Data Valid Delay		250		180		140	
26	TdRD(DRz)	/RD Rise 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.

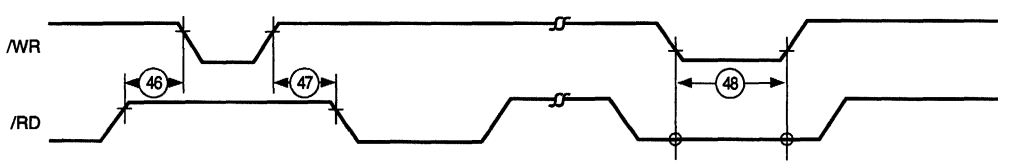


Figure 22. Reset Timing

No.	Symbol	Parameter	4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	8 MHz Min	8 MHz Max	Notes [6]
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		300		280		220	
28	TwWRI	/WR Low Width	240		200		150		
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		240		200		170	[4]
32	TdRD(W)	/RD Fall Wait Valid Delay		240		20		170	[4]
33	TdWRf(REQ)	/WR Fall to /W//REQ Not Valid Delay		240		200		170	
34	TdRDf(REQ)	/RD Fall to /W//REQ Not Valid Delay		240		200		170	
35	TdWRr(REQ)	/WR Fall /DTR//REQ Not Valid Delay		4TcPC		4TcPC		4TcPC	
36	TdRDr(REQ)	/RD Rise to /DTR//REQ Not Valid Delay		4TcPC		4TcPC		4TcPC	
37	TdPC(INT)	PCLK Fall to /INT Valid Delay		500		500		500	[4]
38	TdIAi(RD)	/INTACK to /RD Fall (Acknowledge) Delay	250		200		150		[5]
39	TwRDA	/RD (Acknowledge) Width	250		200		150		
40	TdRDA(DR)	/RD Fall (Acknowledge) to Read Data Valid Delay			250		180	140	
41	TsIEI(RDA)	IEI to FD Fall (Acknowledge) Setup Time	120		100		95		
42	ThIEI(RDA)	IEI to /RD Rise (Acknowledge) Hold Time	0		0		0		
43	TdIEI(IEO)	IEI to IEO Delay Time		120		100		95	
44	TdPC(IEO)	PCLK Rise to IEO Delay		250		250		200	
45	TdRDA(INT)	/RD Fall to /INT Inactive Delay		500		500		450	[4]
46	TdRD(WRQ)	/RD Rise to /WR Fall Delay for No Reset	30		15		15		
47	TdWRQ(RD)	/WR Rise to /RD Fall Delay for No Reset	30		30		20		
48	TwRES	/WR and /RD Coincide 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.
6. Units in nanoseconds (ns), unless otherwise noted.

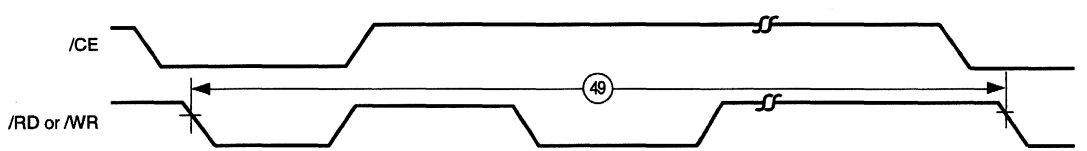


Figure 23. Cycle Timing

AC CHARACTERISTICS

Z8530 General Timing Diagram

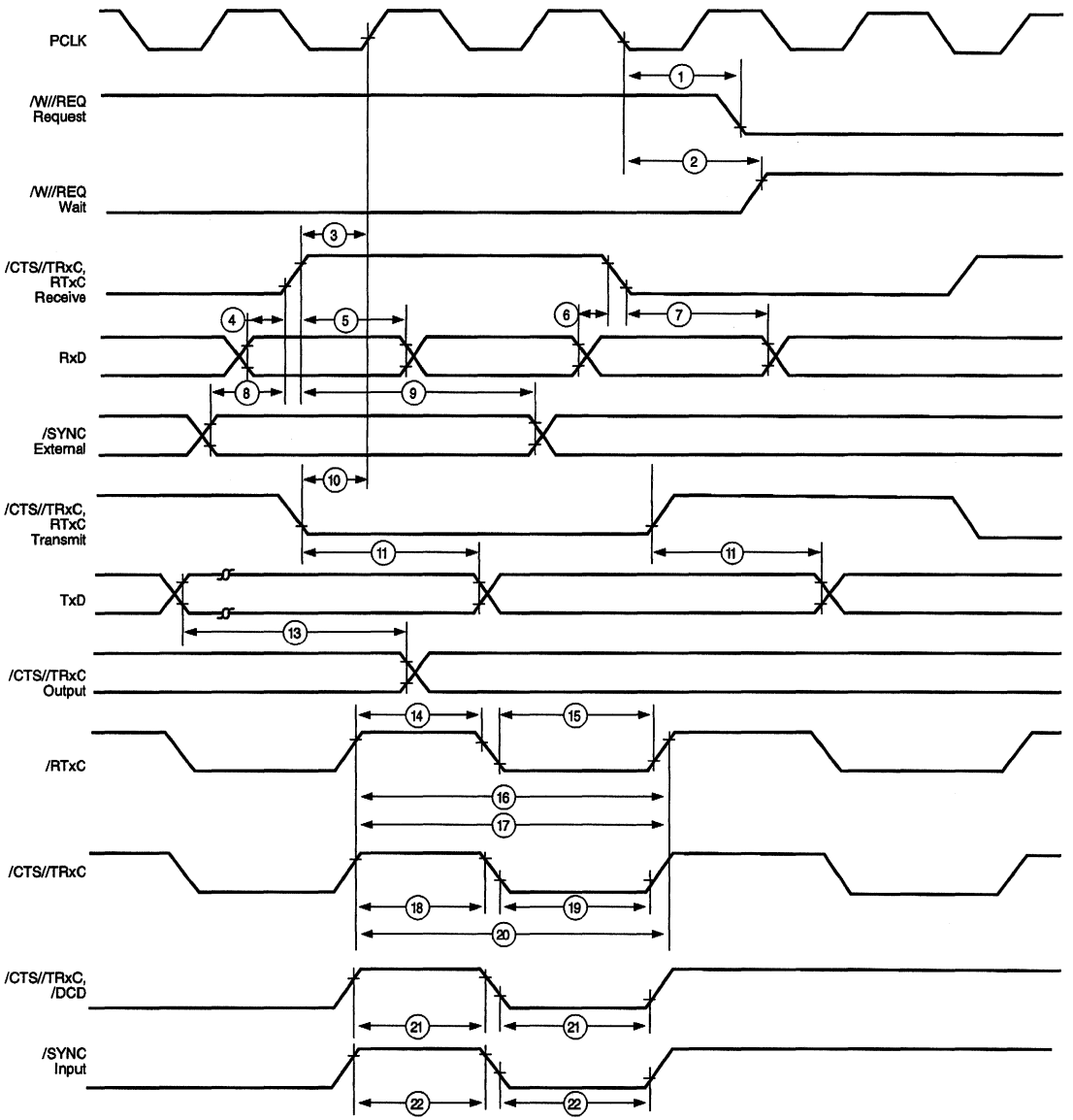


Figure 24. 28530 General Timing Diagram

AC CHARACTERISTICS

Z8530 General Timing Table

No.	Symbol	Parameter	4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	8 MHz Min	8 MHz Max	Notes [6]
1	TdPC(REQ)	/PCLK Low to /W//REQ Valid		250		250		250	
2	TsPC(W)	/PCLK Low to Wait Inactive		350		350		350	
3	TsRXC(PC)	/PxC High to /PCLK High Setup Time /PCLK ÷ 4 case only)	80	TwPCL	70	TwPCL	60	TwPCL	[1,4]
4	TsRXD(RxCr)	RxD to /RxC High Setup Time (X1 Mode)	0		0		0		[1]
5	ThRXD(RxCr)	RxD to /RxC High Hold Time (X1 Mode)	150		150		150		[1]
6	TsRXD(RxCf)	RxD to /RxC Low Setup Time (X1 Mode)	0		0		0		[1,5]
7	ThRXD(RxCf)	RxD to /RxC Low Hold Time (X1 Mode)	150		150		150		[1,5]
8	TsSY(RXC)	SYNC to /RxC High Setup Time	–200		–200		–200		[1]
9	TsSY(RXC)	SYNC to /RxC High Hold Time	3TcPc+400		3TcPc+ 400		3TcPc+400		[1]
10	TsTXC(PC)	/TxC Low to /PCLK High Setup Time	0		0		0		[2,4]
11	TdTXCf(TXD)	/TxC Low to TxD Delay (X1 Mode)		300		230		200	[2]
12	TsTxCcr(TXD)	/TxC High to TxD Delay (X1 Mode)	300		230		200		[12,5]
13	TdTXD(TRX)	TxD to TRxC Delay (Send Clock Echo)		200		200		200	
14	TwRTXh	/RTxC High Width	180		180		150		[6]
15	TwRTXl	/RTxC Low Width	180		180		150		[6]
16	TcRTX	/RTxC Cycle Time (RxD, TxD)	1000		640		500		[7]
17	TcRTXX	Crystal Osc. Period	250	1000	165	1000	125	1000	[3]
18	TwTRXh	/TRxC High Width	180		180		150		[6]
19	TwTRXl	/TRxC Low Width	180		180		150		[6]
20	TcTRX	/TRxC Cycle Time	1000		640		500		[6,7]
21	TwEXT	/DCD or /CTS Pulse Width	200		200		200		
22	TwSY	/SYNC Pulse Width	200		200		20		

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 /SNC 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 timing requirements are identical to case PCLK requirements.

[7] The maximum receive or transmit data ratio is one-fourth PCLK.

[8] Units in nanoseconds (ns), unless otherwise noted.

AC CHARACTERISTICS

Z8530 System Timing Diagram

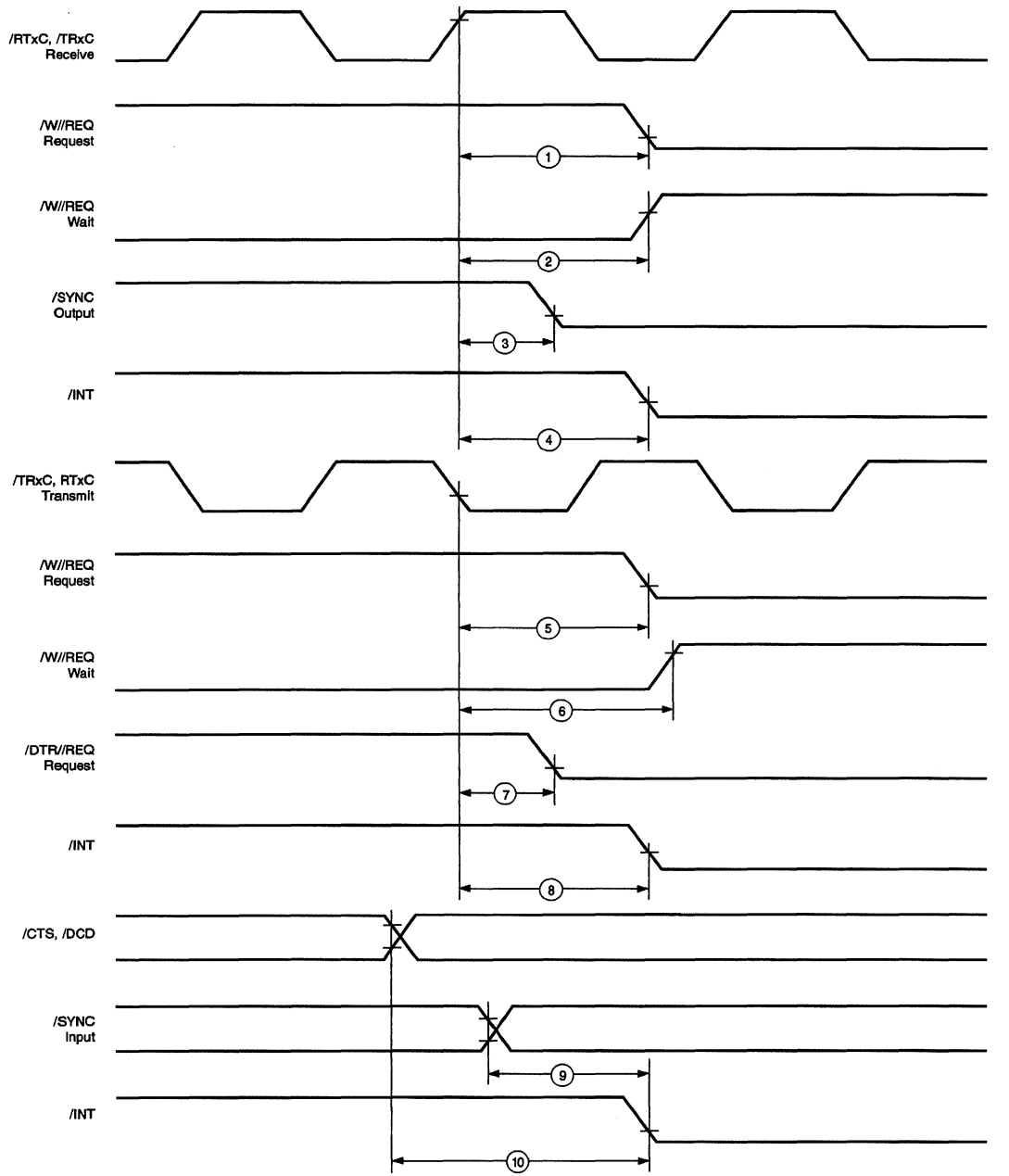


Figure 25. System Timing Diagram

AC CHARACTERISTICS

Z8530 System Timing Table

No.	Symbol	Parameter	4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	8 MHz Min	8 MHz Max	Notes [5]
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]
4	TsRXC(INT)	/RxC High to INT Valid	10	16	10	16	10	16	[1,2]
			2	3	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]
8	TdTXC(INT)	/TxC Low to INT Valid	6	10	6	10	6	10	[1,3]
			2	3	2	3	2	3	[4]
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.

[4] Units equal to /AS.

[5] Units equal to TcPc.

Z8030 AC CHARACTERISTICS
Read and Write Timing

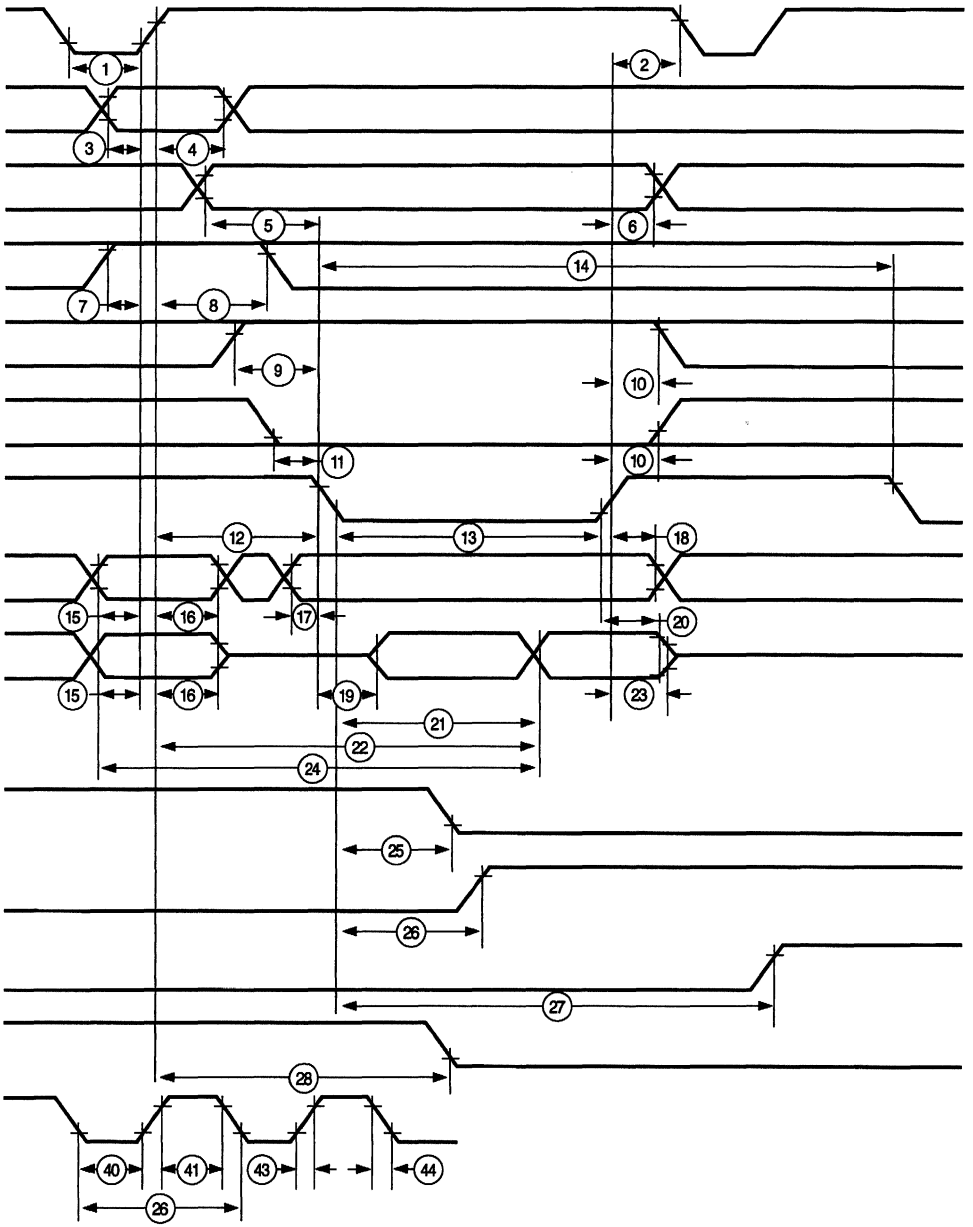


Figure 26. Read and Write Timing

Z8030 AC CHARACTERISTICS
Interrupt Acknowledge Timing

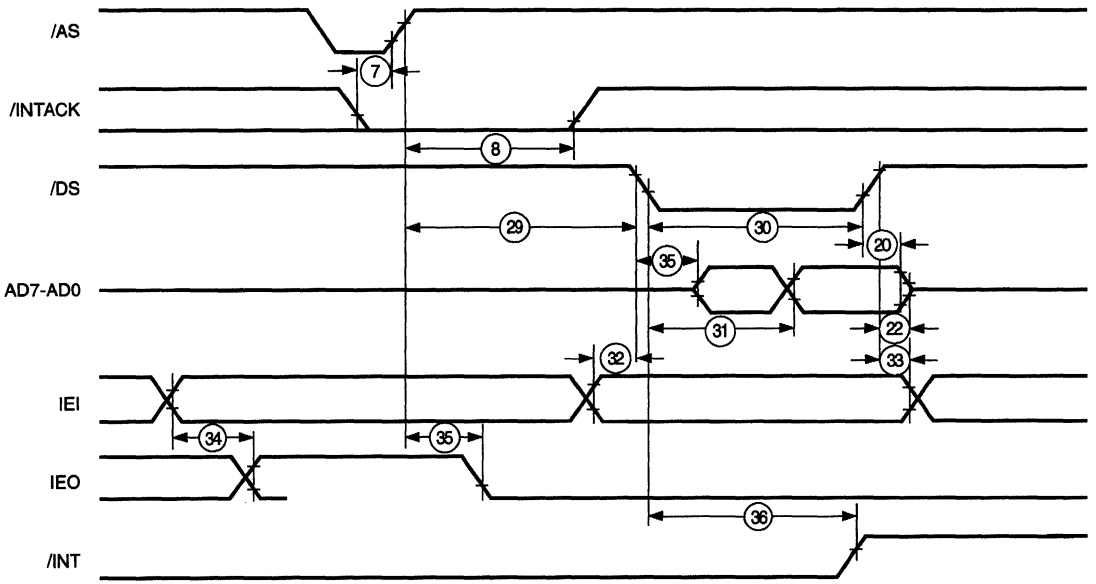


Figure 27. Interrupt Acknowledge Timing

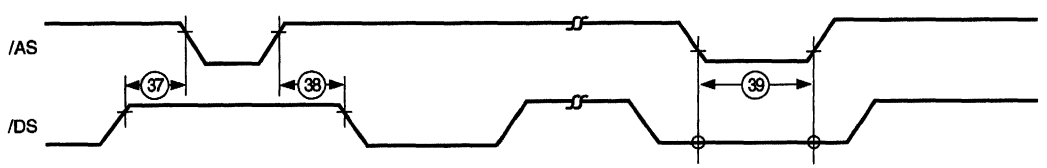


Figure 28. Rest Timing

Z8030 AC CHARACTERISTICS (Continued)

No.	Symbol	Parameter	4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	8 MHz Min	8 MHz Max	Notes [3]
1	TwAS	/AS Low Width	70		50		35		
2	TdDS(AS)	/DS to /AS Fall Delay	50		25		15		
3	TsCSC)(AS)	/CS0 to /AS Rise Setup Time	0		0		0		[1]
4	ThCSO(AS)	/CS0 to /AS Rise Hold lime	60		40		30		[1]
5	TsCS1(DS)	CS1 to /DS Fall Setup Time	100		80		65		[1]
6	ThCS1(DS)	CS1 to /DS Rise Hold Time	55		40		30		[1]
7	TsIA(AS)	/INTACK to /AS Rise Setup Time	10		10		10		
8.	ThIA(AS)	/INTACK to /AS Rise Hold Time	250		200		150		
9	TsSRWR(DS)	R//W (Read) to /DS Fall Setup Time	100		80		65		
10	ThRW(DS)	R//W to /DS Rise Hold Time	55		40		35		
11	TsRWW(DS)	R//W (Write) to /DS Fall Setup Time	0		0		0		
12	TdAS(DS)	/AS Rise to /DS Fall Delay	60		40		30		
13	TwDSI	/DS Low Width	240		200		150		
14	Trc	Valid Access Recovery Time	4TcPC		4TcPC		4TcPC		[2]
15	TsA(AS)	Address to /AS Rise Setup Time	30		10		10		[1]
16	ThA(AS)	Address to /AS Rise Hold Time	50		30		25		[1]
17	TsDW(DS)	Write Data to /DS Fall Setup Time	30		20		15		
18	ThDW(DS)	Write Date to /DS Rise Hold Time	30		20		20		
19	TdDS(DA)	/DS Fall to Data Active Delay	0		0		0		
20	TdDSr(DR)	/DS Rise to Read Data Not Valid Delay	0		0		0		
21	TdDSf(DR)	/DS Fall to Read Data Valid Delay		250		180		140	
22	TdAS(DR)	/AS Rise 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.
3. Units in nanoseconds (ns), unless otherwise noted.

No.	Symbol	Parameter	4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	8 MHz Min	8 MHz Max	Notes [8]
23	TdDS(DRz)	/DS Rise 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)	/DS Fall to Wait Valid Delay		240		200		170	[4]
26	TdDSf(REQ)	/DS Fall to /W//REQ Not Valid Delay		240		200		170	
27	TdDSr(REQ)	/DS Fall to /DTR//REQ Not Valid Delay		4TcPC		4TcPC		4TcPC	
28	TdAS(INT)	/AS Rise to /INT Valid Delay		500		500		500	[4]
29	TdAS(DSA)	/AS Rise to /DS Fall (Acknowledge) Delay	250		250		250		[5]
30	TwDSA	/DS (Acknowledge) Low Width	390		200		150		
31	TdDSA(DR)	/DS Fall (Acknowledge) to Read Data Valid Delay		250		180		140	
32	TslEI(DSA)	IEI to /DS Fall (Acknowledge) Setup Time	120		100		80		
33	ThIEI(DSA)	IEI to /DS Fall (Acknowledge) Hold Time	0		0		0		
34	TdIEI(IEO)	IEI to IEO Delay		120		100		90	
35	TdAS(IEO)	/AS Rise to IEO Delay		250		250		200	[6]
36	TdDSA(INT)	/DS Fall (Acknowledge to /INT Inactive Delay		500		500		450	[4]
37	TdDS(ASO)	/DS Rise to /AS Fall Delay for No Reset	30		15		15		
38	TdASQ(DS)	/AS Rise to /DS Fall Delay for No Reset	30		30		20		
39	TwRES	/AS and /DS Coincide 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, TslEI(DSA) for the Z-SCC, and TdIElf(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 ZB 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."
8. Units in nanoseconds (ns), unless otherwise noted.

AC CHARACTERISTICS

Z8030 General Timing Diagram

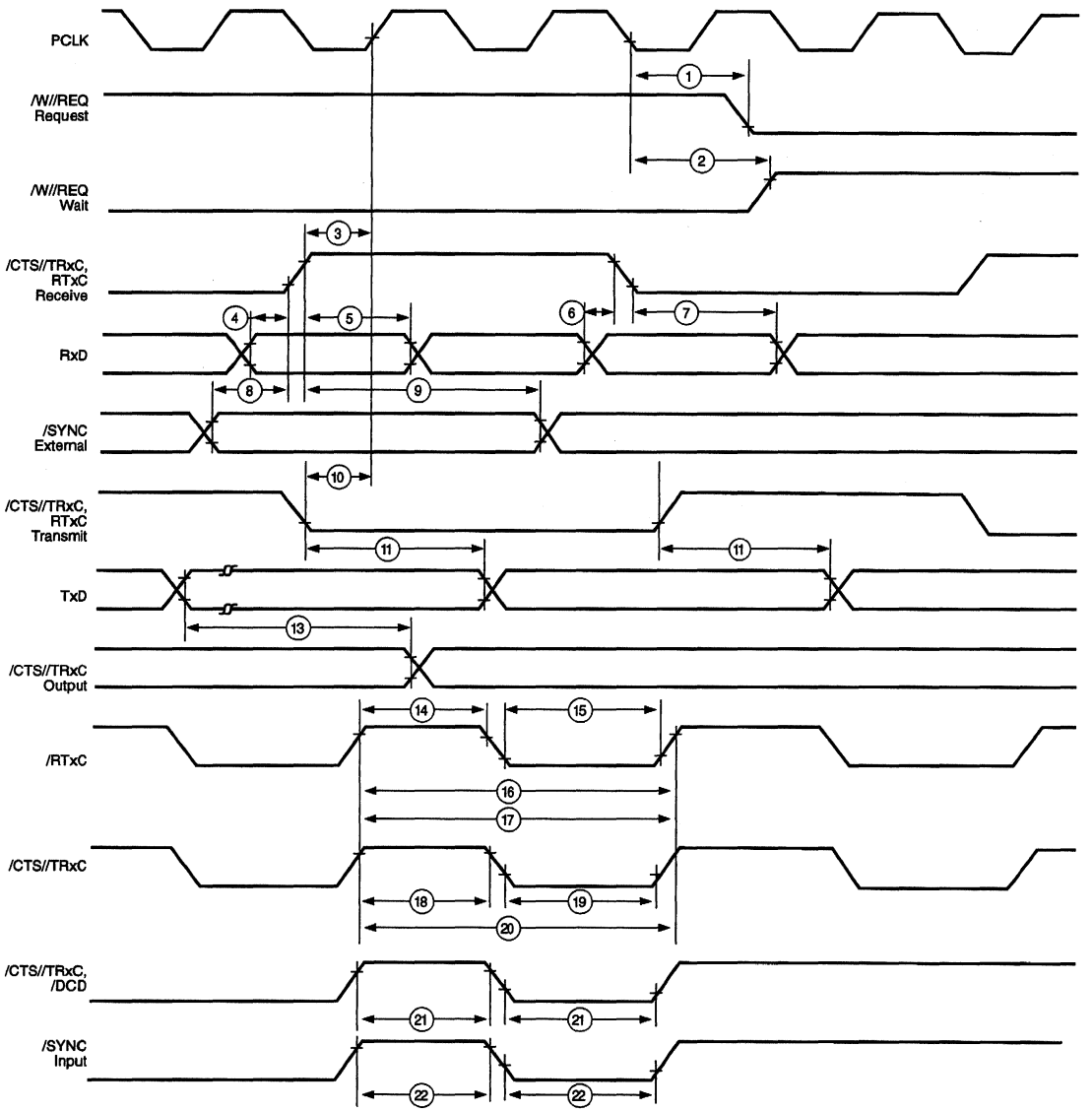


Figure 29. Z8030 General Timing Diagram

AC CHARACTERISTICS

Z8030 General Timing Diagram

No.	Symbol	Parameter	4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	8 MHz Min	8 MHz Max	Notes [8]
1	TdPC(REQ)	/PCLK Low to /W//REQ Valid		250		250		250	
2	TsPC(W)	/PCLK Low to Wait Inactive		350		350		350	
3	TsRXC(PC)	/RxC High to /PCLK High Setup Time (PCLK ÷ 4 case only)	80	TwPCL	70	TvvPCL	60	TvvPCL	[1,4]
4	TsRXD(RXCr)	RxD to /RxC High Setup Time (X1 Mode)	0		0		0		[1]
5	ThRXD(RxCr)	RxD to /RxC High Hold Time (X1 Mode)	150		150		150		[1]
6	TsRXD(RXCf)	RxD to /RxC Low Setup Time (X1 mode)	0		0		0		[1,5]
7	ThRXD(RXCf)	RxD to /RxC Low Hold Time (X1 Mode)	150		150		150		[1,5]
8	TsSY(RXC)	SYNC to /RxC High Setup Time	–200		–200		–200		[1]
9	ThSY(RxC)	/SYNC to /RxC High Hold Time	3TcPc+400		3TcPc+400		3TcPc+400		[1]
10	TsT XC(PC)	/TxC Low to /PCLK High Setup Time	0		0		0		[2,4]
11	TdTXCf(TXD)	/TxC Low to TxD Delay (X1 Mode)		300		230		200	[2]
12	TdTxCr(TXD)	/TxC High to TxD Delay (X1 Mode)		300		230		200	[2,5]
13	TdTXD(TRX)	TxD to TRxC Delay (Send Clock Echo)		200		200		200	
14	TwRTXh	/RTxC High Width	180		180		150		[6]
15	TwRT Xl	/RTxC Low Width	180		180		150		[6]
16	TcRTX	/RTxC Cycle Time (RxD,TxD)	1000		640		500		[6,7]
17	TcRTXX	Crystal Osc. Period	250	1000	165	1000	125	1000	[3]
18	TvvTRXh	/TRxC High Width	180		180		150		[6]
19	TWTRXl	/TRxC Low Width	180		180		150		[6]
20	TcTRX	/TRXC Cycle Time	1000		640		500		[6,7]
21	TwEXT	/DCD or /CTS Pulse Width	200		200		200		
22	TwSY	/SYNC Pulse Width	200		200		200		

Notes:

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

[2] rTxC is fTRxC 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 timing requirements are identical to case PCLK requirements.

[7] The maximum receive or transmit data rate is one-fourth PCLK.

[8] Units in nanoseconds (ns), unless otherwise noted.

AC CHARACTERISTICS

Z8030 System Timing Diagram

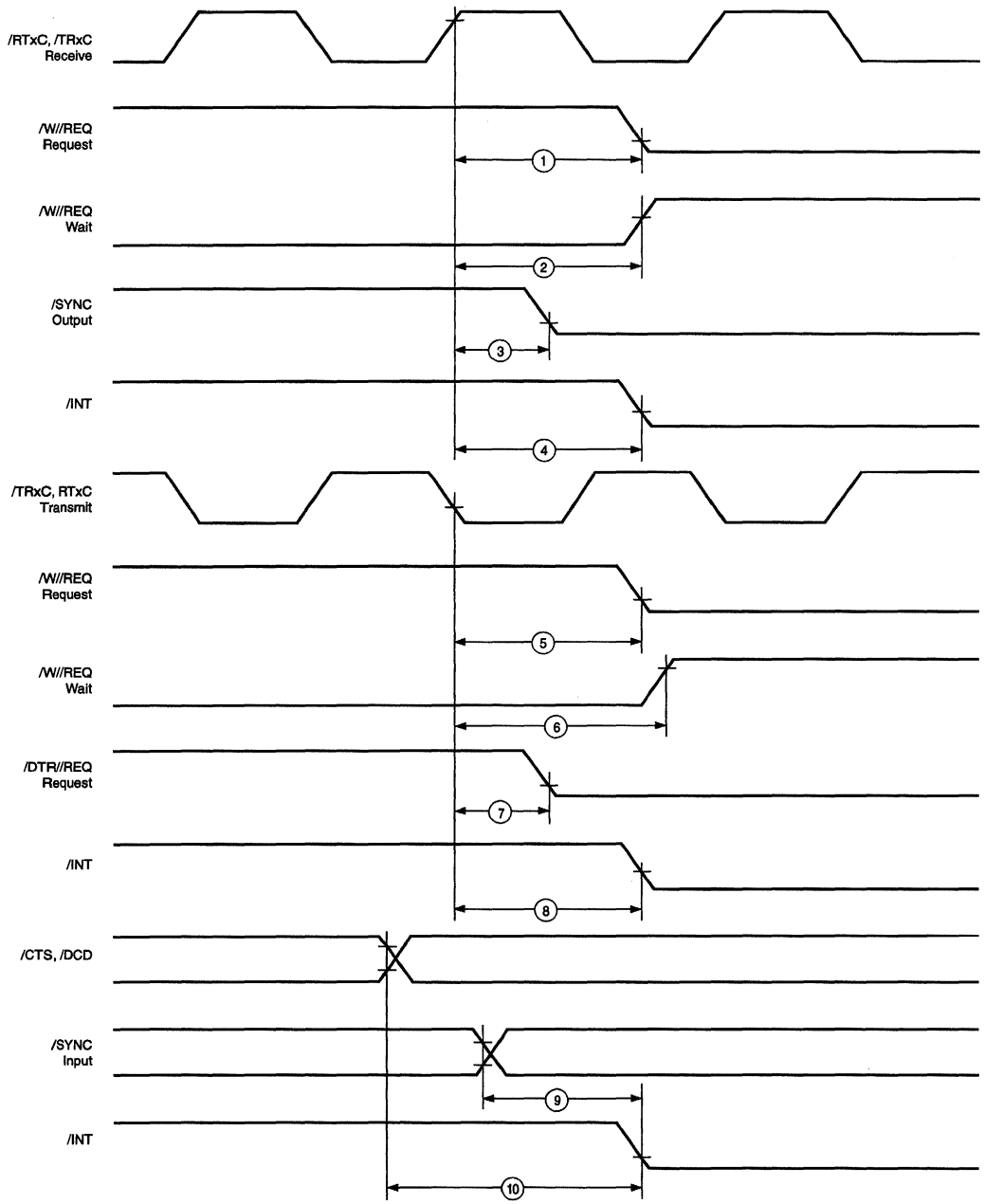


Figure 30. 28530 System Timing Diagram

AC CHARACTERISTICS

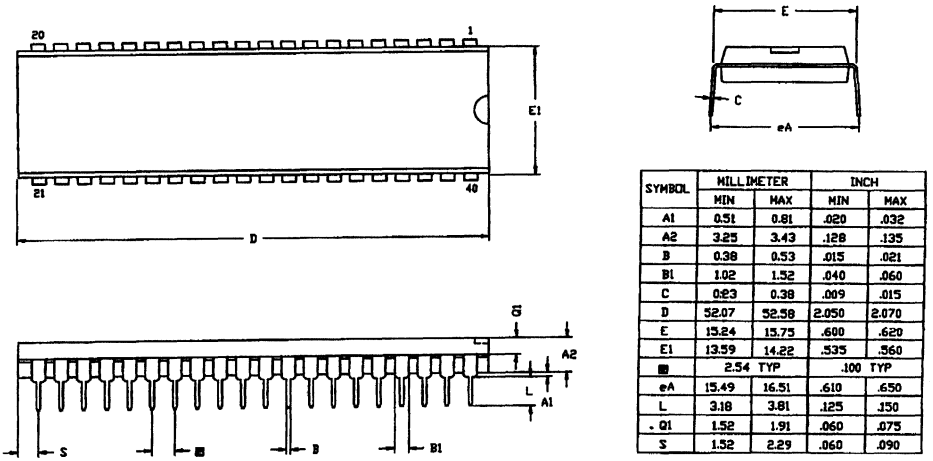
Z8030 System Timing Diagram

No.	Symbol	Parameter	4 MHz Min	4 MHz Max	6 MHz Min	6 MHz Max	8 MHz Min	8 MHz Max	Notes [5]
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]
4	TdRXC(INT)	/RxC High to INT Valid	8	12	8	12	8	12	[1,2]
			2	3	2	3	2	3	[4]
5	TdTXC(REQ)	/TxC Low to /W//REQ Valid	5	8	5	8	5	8	[3]
6	TdT XC(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]
8	TdTXC(INT)	/TxC Low to /INT Valid	4	6	4	6	4	6	[1,3]
			2	3	2	3	2	3	[4]
9	TdSY(INT)	SYNC to INT Valid	2	3	2	3	2	3	[1,4]
10	TdEXT(INT)		2	3	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.
[5] Units in nanoseconds (ns), unless otherwise noted.

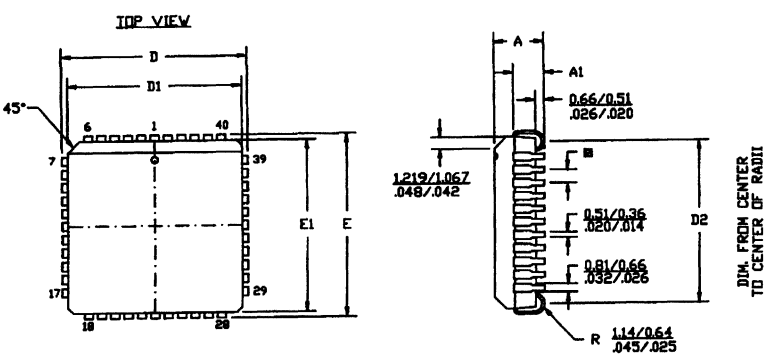
PACKAGE INFORMATION



SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A1	0.51	0.81	.020	.032
A2	3.25	3.43	.128	.135
B	0.38	0.53	.015	.021
B1	1.02	1.52	.040	.060
C	0.23	0.38	.009	.015
D	52.07	52.58	2.050	2.070
E	15.24	15.75	.600	.620
E1	13.59	14.22	.535	.560
e	2.54 TYP		.100 TYP	
eA	15.49	16.51	.610	.650
L	3.18	3.81	.125	.150
Q1	1.52	1.91	.060	.075
S	1.52	2.29	.060	.090

CONTROLLING DIMENSIONS : INCH

40-Pin DIP Package Diagram



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

SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	4.27	4.57	.168	.180
A1	2.67	2.92	.105	.115
D/E	17.40	17.65	.685	.695
D1/E1	16.51	16.66	.650	.656
D2	15.24	16.00	.600	.630
e	1.27 TYP		.050 TYP	

44-Pin PLCC Package Diagram

ORDERING INFORMATION

Z8030

4 MHz	6 MHz	8 MHz
40-Pin DIP	**40-Pin DIP**	**40-Pin DIP**
Z803004PSC	Z803006PSC	Z803008PSC
4 MHz	**6 MHz**	**8 MHz**
44-Pin PLCC	**44-Pin PLCC**	**44-Pin PLCC**
Z803004VSC	Z803006VSC	Z803008VSC

Z8530

4 MHz	6 MHz	8 MHz
40-Pin DIP	**40-Pin DIP**	**40-Pin DIP**
Z853004PSC	Z853006PSC	Z853008PSC
4 MHz	**6 MHz**	**8 MHz**
44-Pin PLCC	**44-Pin PLCC**	**44-Pin PLCC**
Z853004VSC	Z853006VSC	Z853008VSC

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

Speed
04 = 4 MHz
06 = 6 MHz
08 = 8 MHz

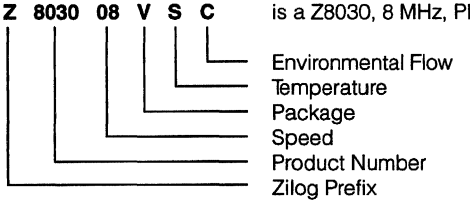
Package
V = Plastic Chip Carrier

Temperature
S = 0°C to +70°C

Environmental
C = Plastic Standard

Example:

Z 8030 08 V S C is a Z8030, 8 MHz, PLCC, 0°C to +70°C, Plastic Standard Flow



Z80C30/Z85C30

CMOS SCC
SERIAL COMMUNICATION CONTROLLER

FEATURES

- Z85C30 Optimized for Non-Multiplexed Bus Microprocessors. Z80C30 Optimized for Multiplexed Bus Microprocessors.
- 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 (not with NMOS)
- Local Loopback and Auto Echo Modes
- Supports T1 Digital Trunk
- Enhanced DMA Support (not with NMOS)
  10 x 19-Bit Status FIFO
  14-Bit Byte Counter
- Speeds:
  Z85C30 - 8.5, 10, 16.384 MHz
  Z80C30 - 8, 10 MHz

Other Features for Z85C30 only:

- New programmable WR7' (write register 7 prime) to enable new features.
- Improvements to support SDLC mode of synchronous communication:
  - Improve functionality to ease sending back-to-back frames.
  - Automatic SDLC opening Flag transmission*
  - Automatic Tx Underrun/EOM Latch reset in SDLC mode*
  - Automatic /RTS deactivation*
  - TxD pin forced "H" in SDLC NRZI mode after closing flag*
  - Complete CRC reception*
  - Improved response to Abort sequence in status FIFO
  - Automatic Tx CRC generator preset/reset
  - Extended read for write registers*
  - Write data set-up timing improvement
- Improved AC timing
  - Three to 3.6 PCLK access recovery time.
  - Programmable /DTR//REQ timing*
  - Write data to falling edge of /WR set-up time requirement is now eliminated.
  - Reduced /INT timing
- Other features include:
  - Extended read function to read back the written value to the write registers. *
  - Latching RR0 during read
  - RR0, bit D7 and RR10, bit D6 now has reset default value.

Some of the features listed above are available by default, and some of them (features with "*") are disabled on default to maintain compatibility with the existing SCC design, and "program to enable" through WR7".

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 (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.

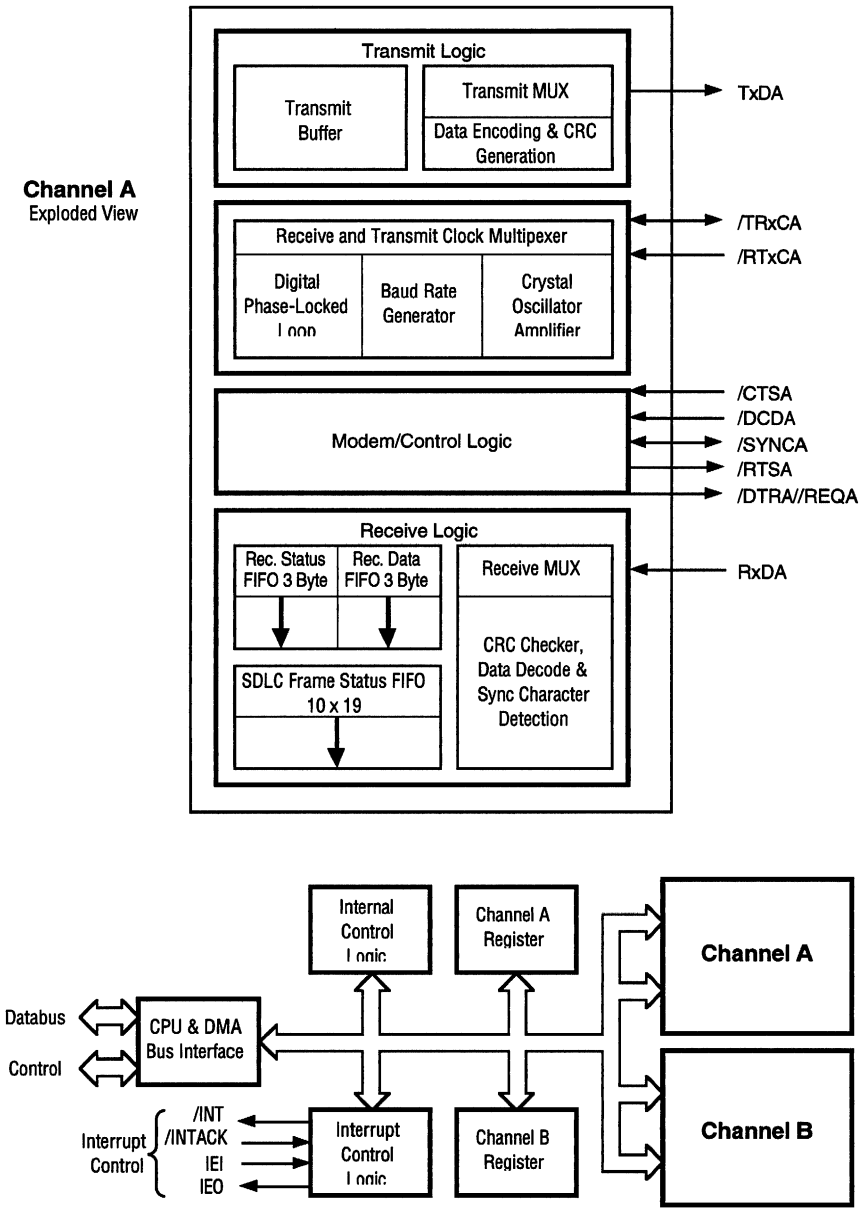


Figure 1. SCC Block Diagram

GENERAL DESCRIPTION (Continued)

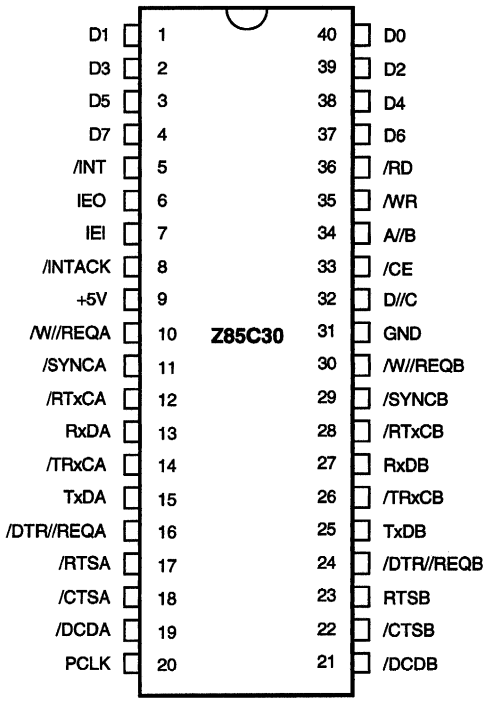


Figure 2. Z85C30 DIP Pin Assignments

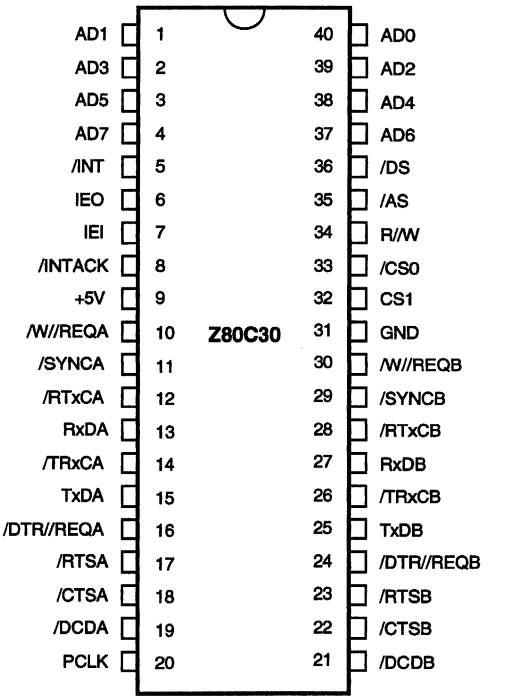


Figure 3. Z80C30 DIP Pin Assignments

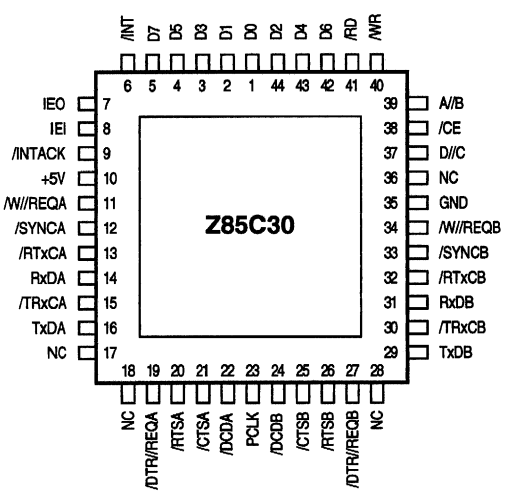


Figure 4. Z85C30 PLCC Pin Assignments

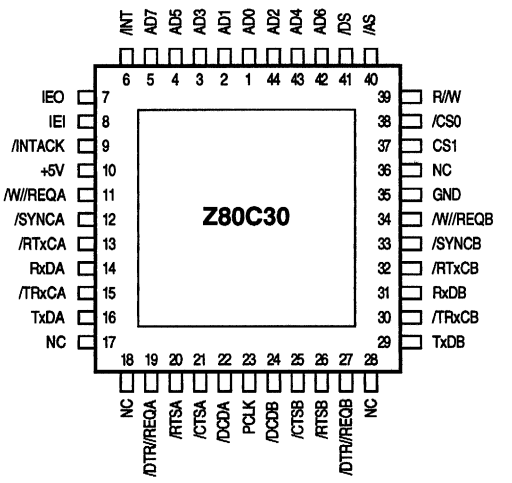


Figure 5. Z80C30 PLCC Pin Assignments

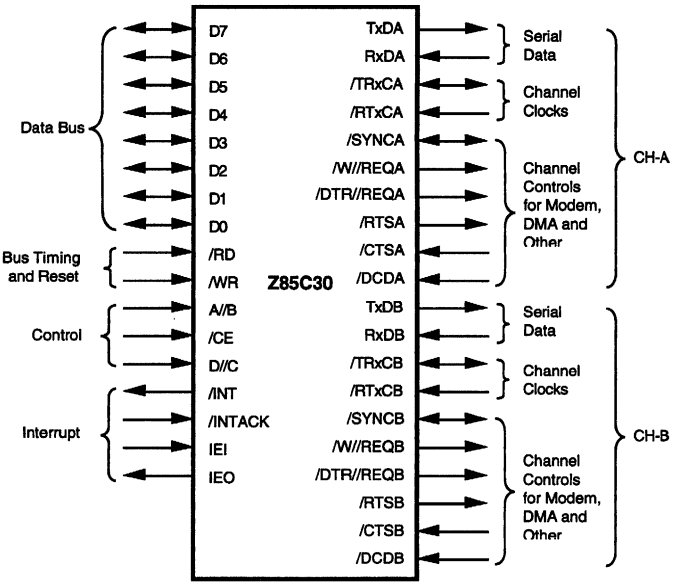


Figure 6. Z85C30 Pin Functions

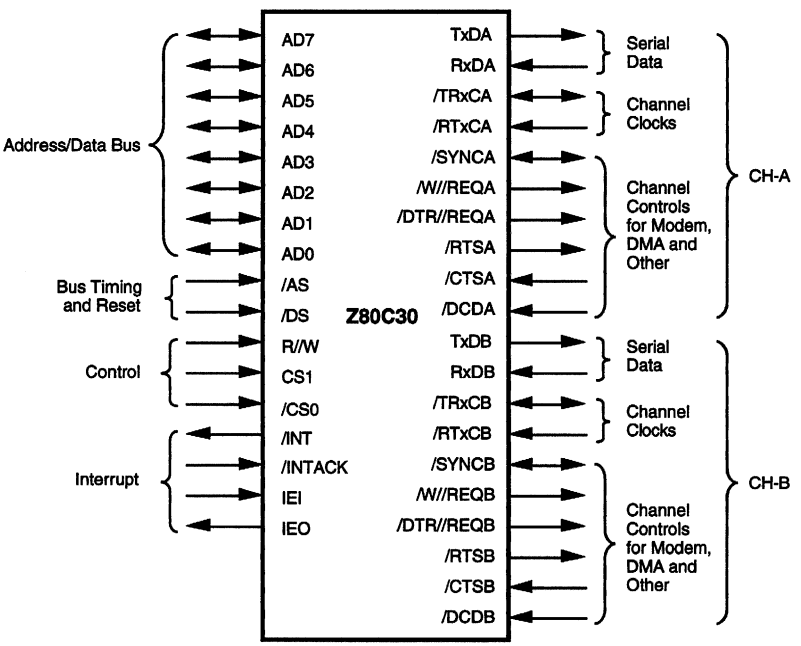


Figure 7. Z80C30 Pin Functions

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 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 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.

/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, tri-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, Tri-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.

2

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.

FUNCTIONAL DESCRIPTION (Continued)

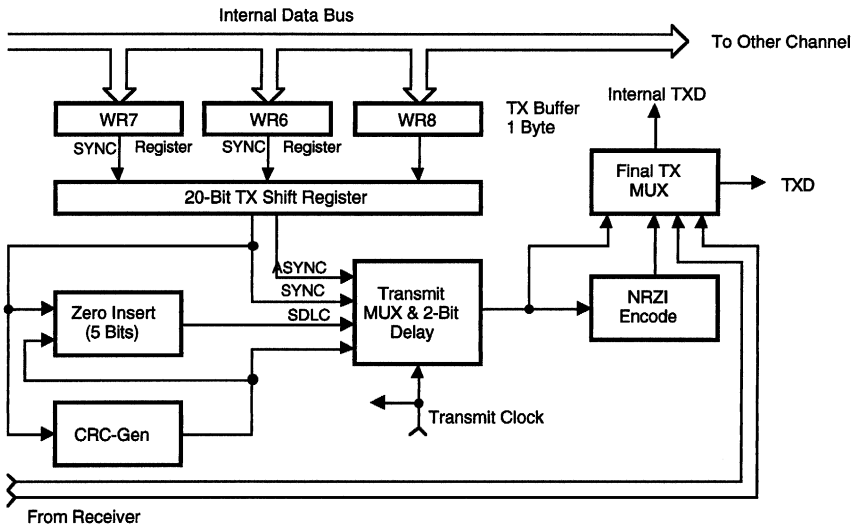


Figure 8. SCC Transmit Data Path

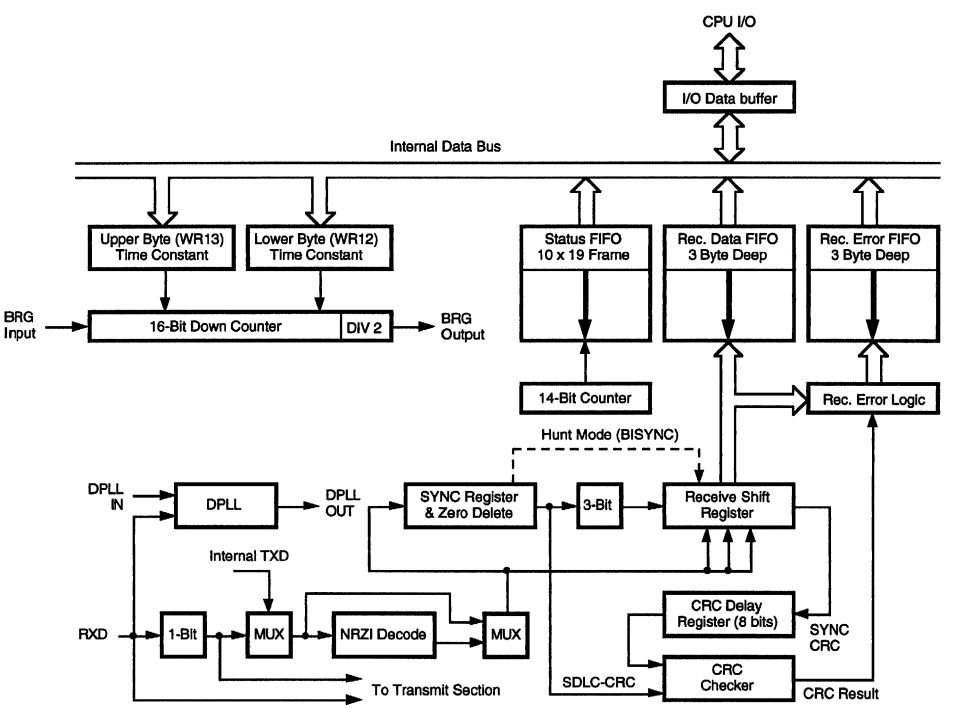


Figure 9. SCC Receive Data Path

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
*WR7'	Extended Feature and FIFO Control (WR7 Prime)
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

* 85C30 Only

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.

FUNCTIONAL DESCRIPTION (Continued)

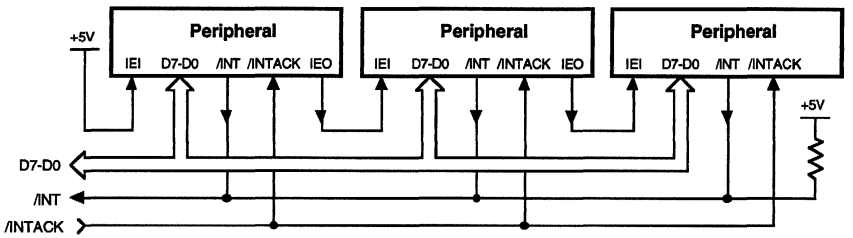


Figure 10. SCC Interrupt Priority Schedule

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

On the CMOS version of the SCC, 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.

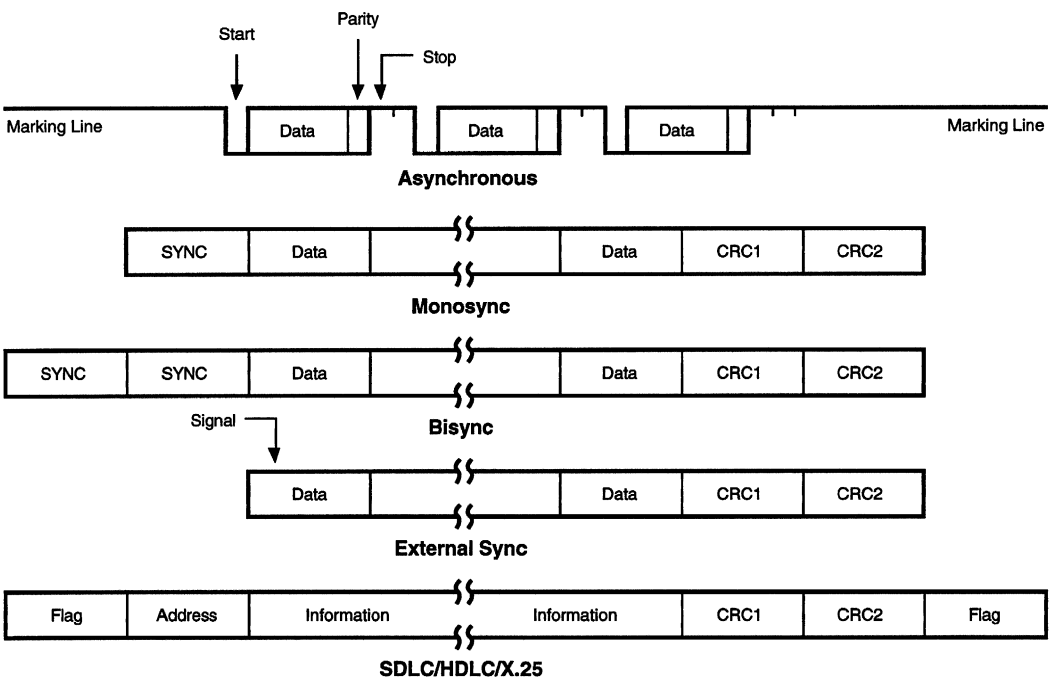


Figure 11. Some SCC Protocols

FUNCTIONAL DESCRIPTION (Continued)

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.

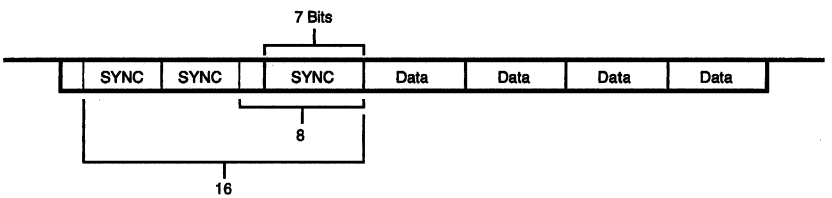


Figure 12. Detecting 5- or 7-Bit Synchronous Characters

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.

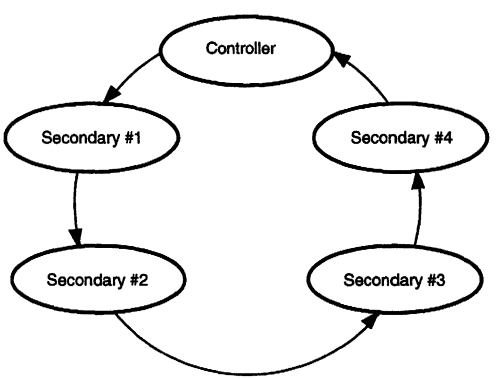


Figure 13. An SDLC Loop

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 10 x 19 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 start-up, 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.

FUNCTIONAL DESCRIPTION (Continued)

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 through 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 through 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.

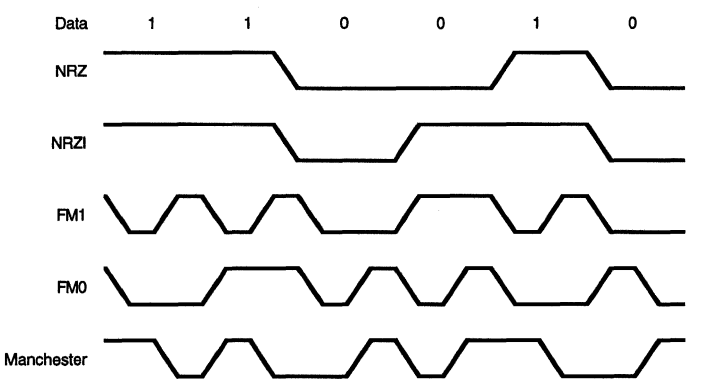


Figure 14. Data Encoding Methods

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 through 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 through 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.

FUNCTIONAL DESCRIPTION (Continued)

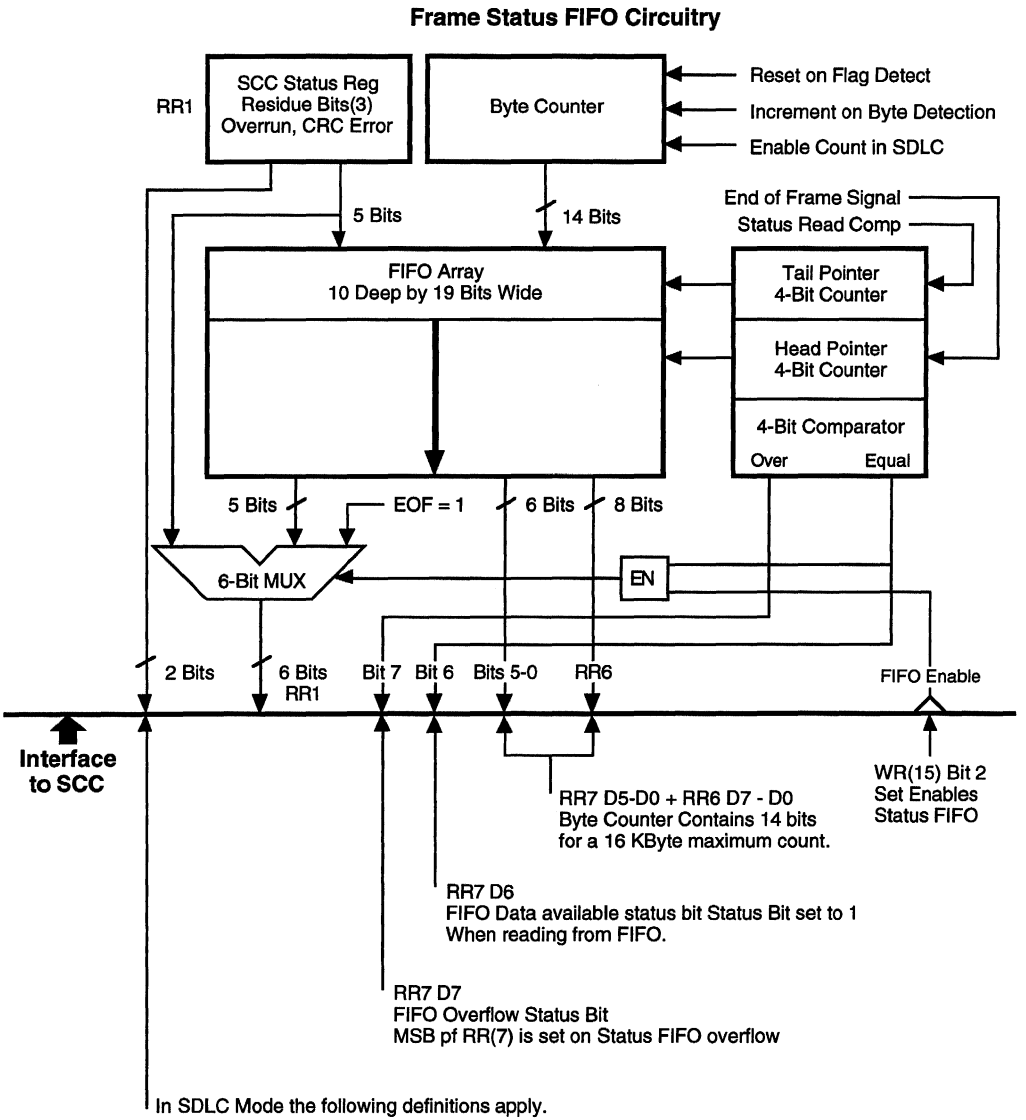


Figure 15. SDLC Frame Status FIFO

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.

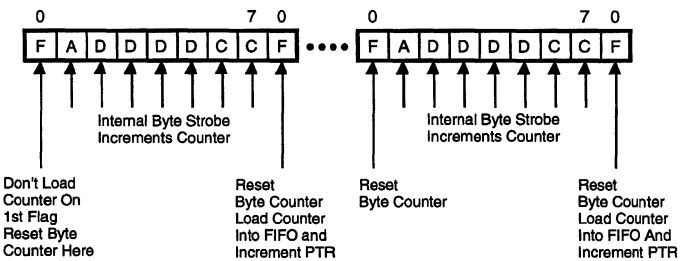


Figure 16. SDLC Byte Counting Detail

2

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 for the 80C30, while there are 16 for the 85C30 (one more additional write register if 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.

PROGRAMMING (Continued)

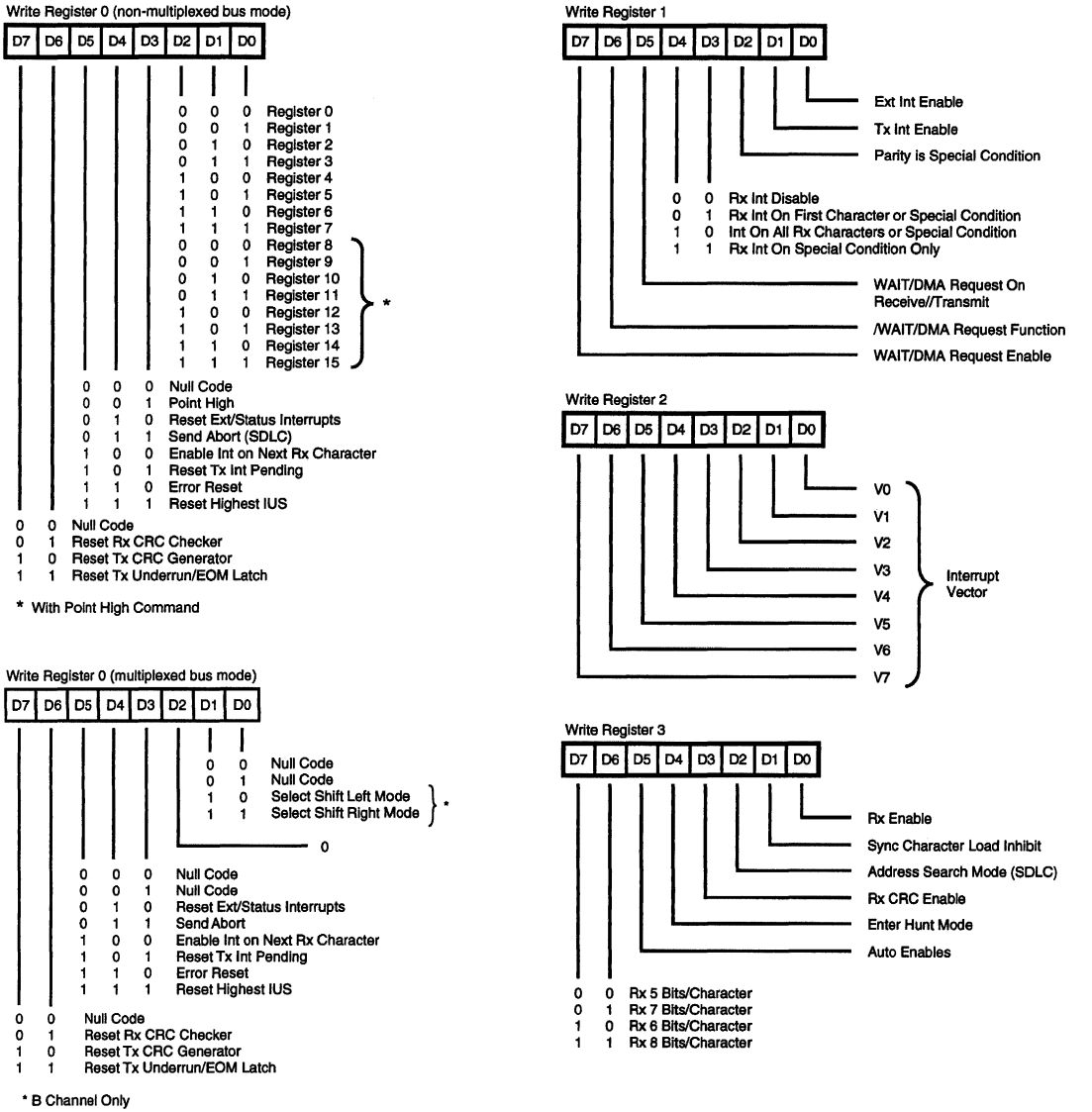


Figure 17. Write Register Bit Functions

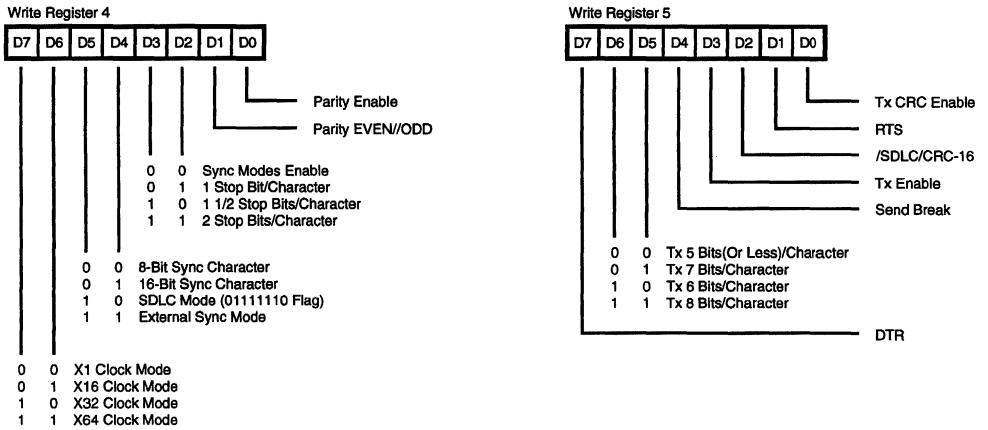
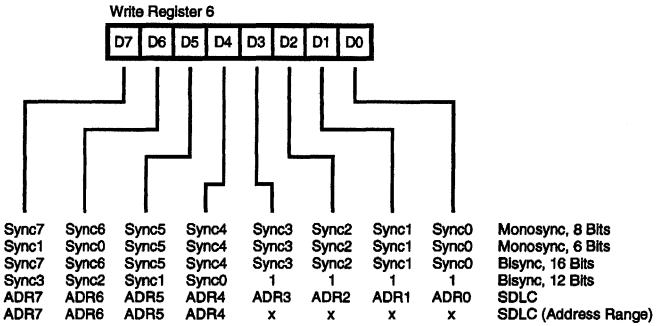
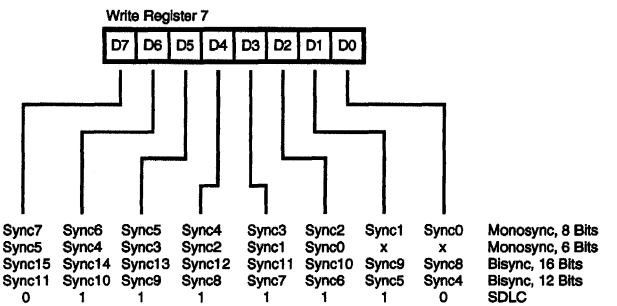


Figure 17. Write Register Bit Functions (Continued)

2

PROGRAMMING (Continued)





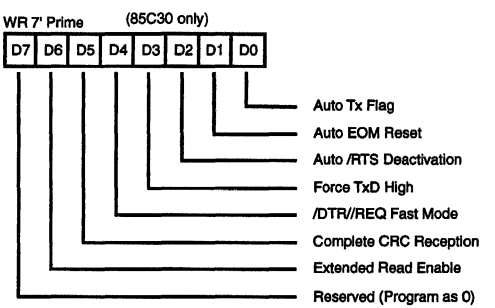


Figure 17. Write Register Bit Functions (Continued)

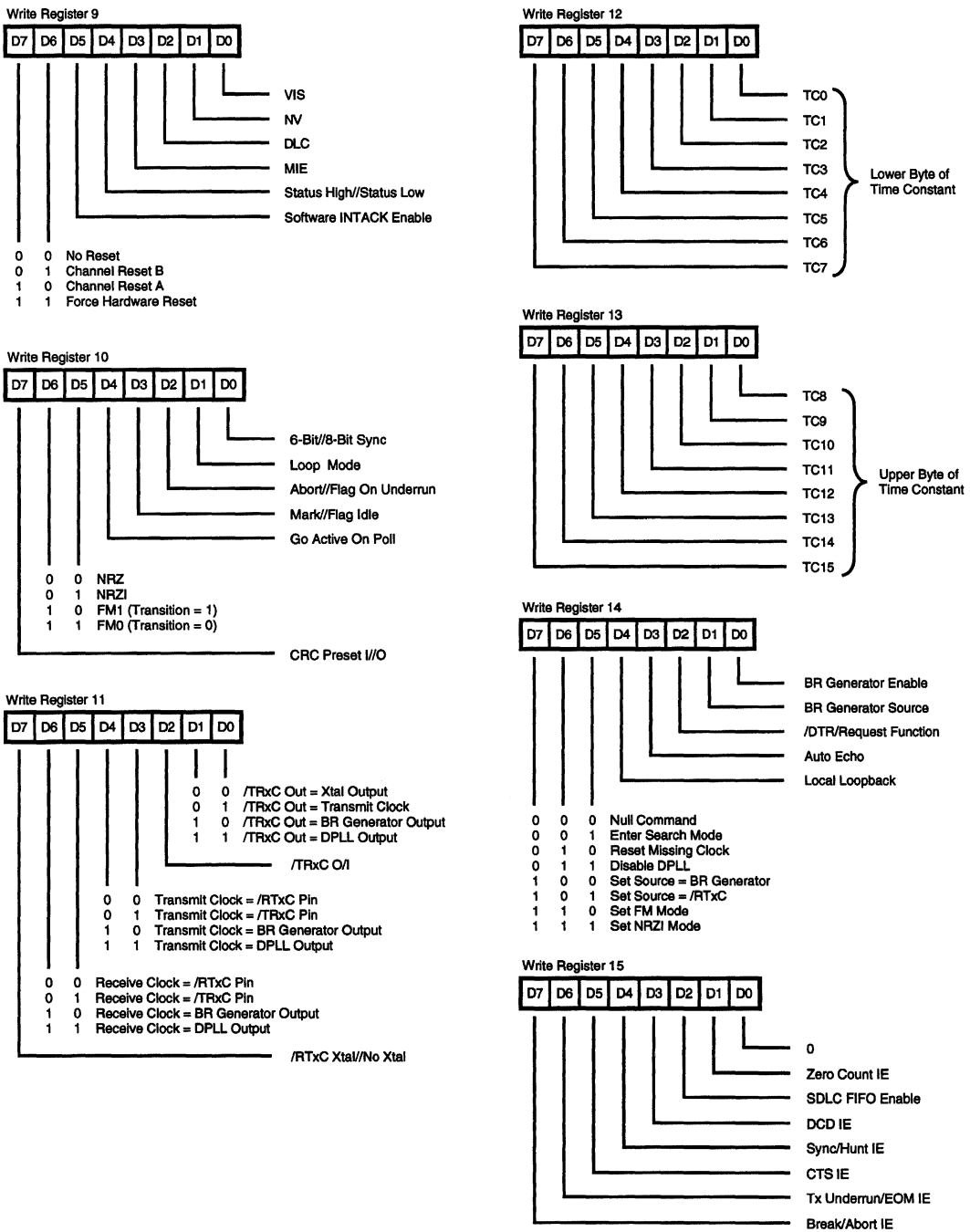


Figure 17. Write Register Bit Functions (Continued)

PROGRAMMING (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 -, Figure18). RR6 and RR7 contain the information in the SDLC Frame Status FIFO, but is only read when WR15 D2 is set (Figure 15).

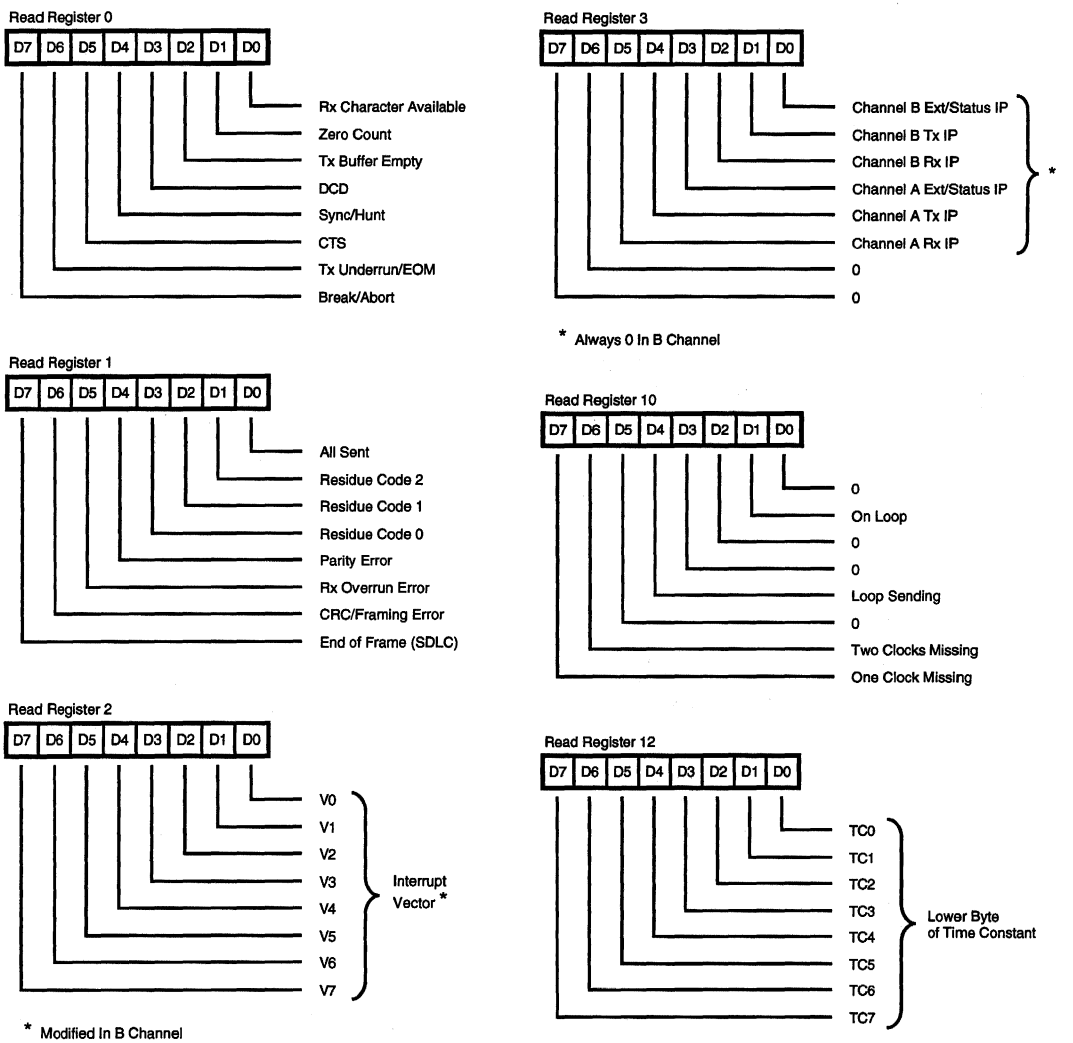


Figure 18. Read Register Bit Functions

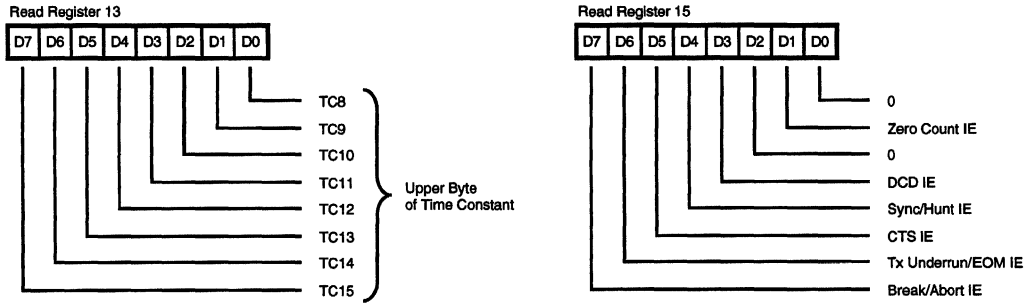


Figure 18. Read Register Bit Functions (Continued)

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 3 PCLKs regardless of which register or channel is being accessed.

2

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.

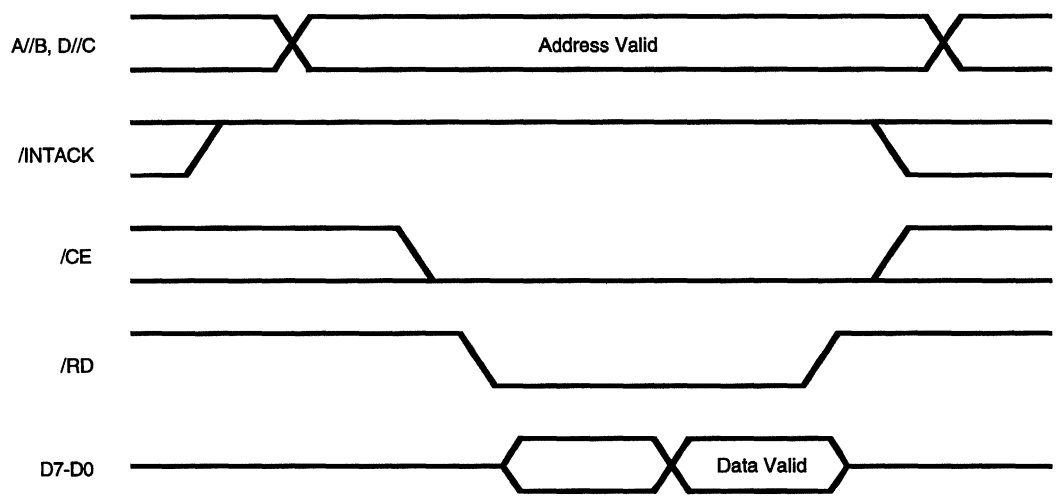


Figure 19. Read Cycle Timing

PROGRAMMING (Continued)

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.

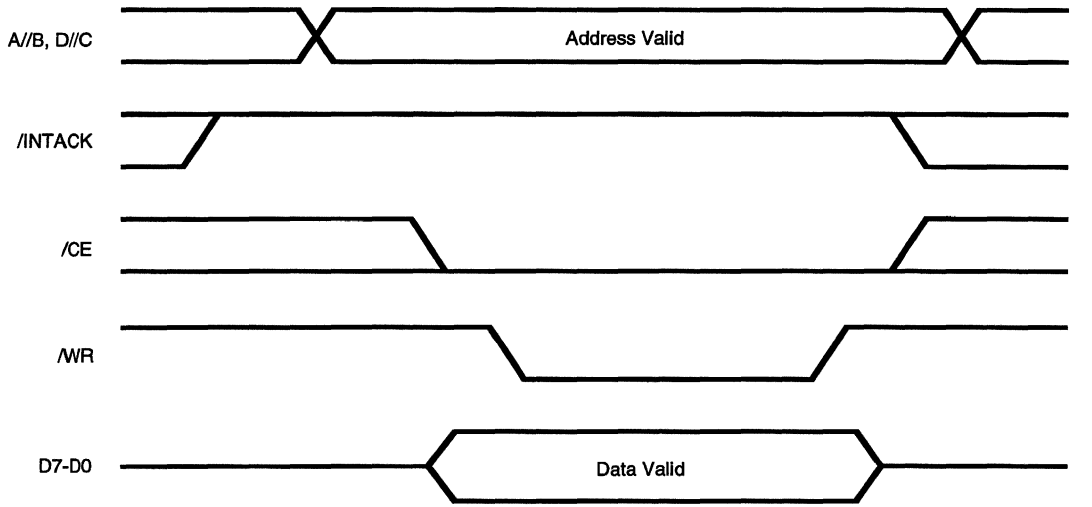


Figure 20. Write Cycle Timing

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 of the Read/Write Timing Table for calculating the required daisy-chain settle time.

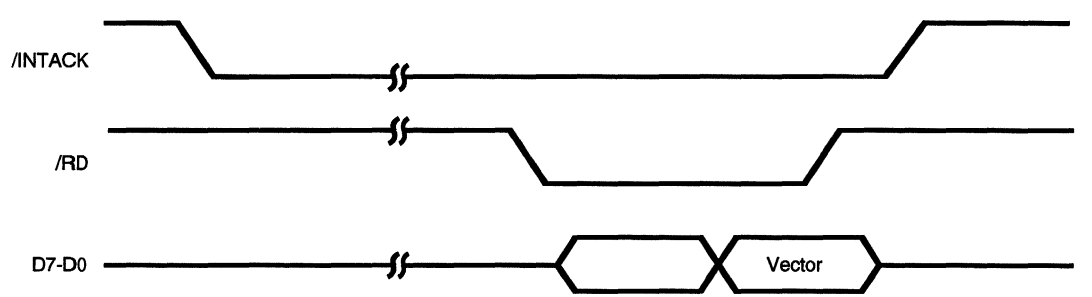


Figure 21. Interrupt Acknowledge Cycle Timing

Z80C30 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.

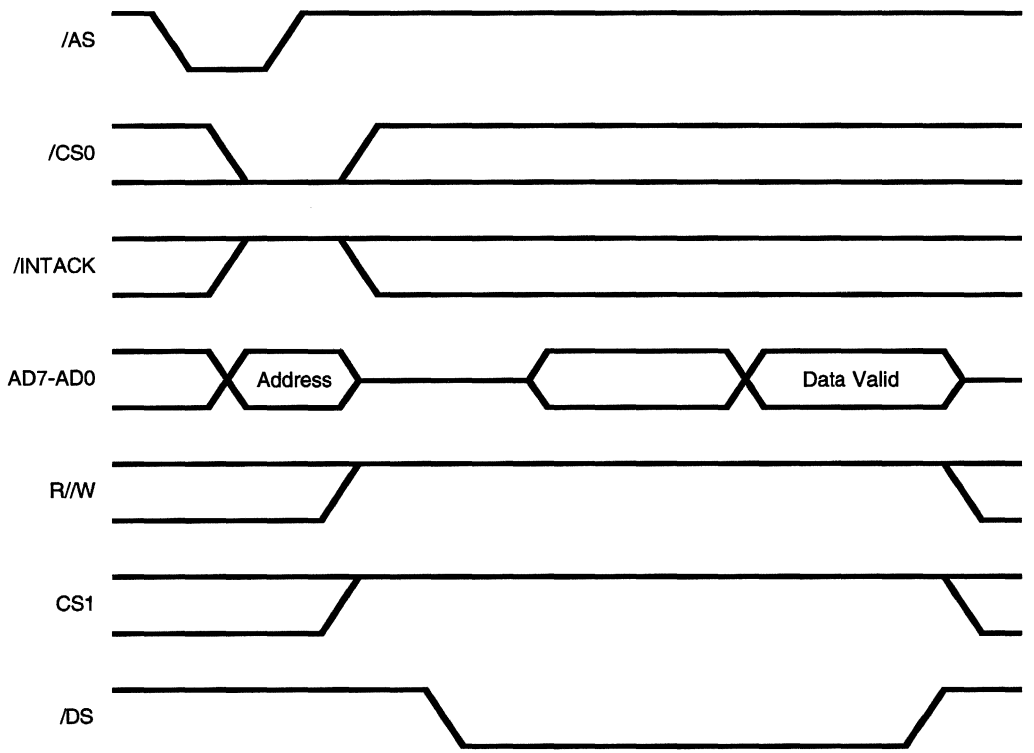


Figure 22. Read Cycle Timing

PROGRAMMING (Continued)

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.

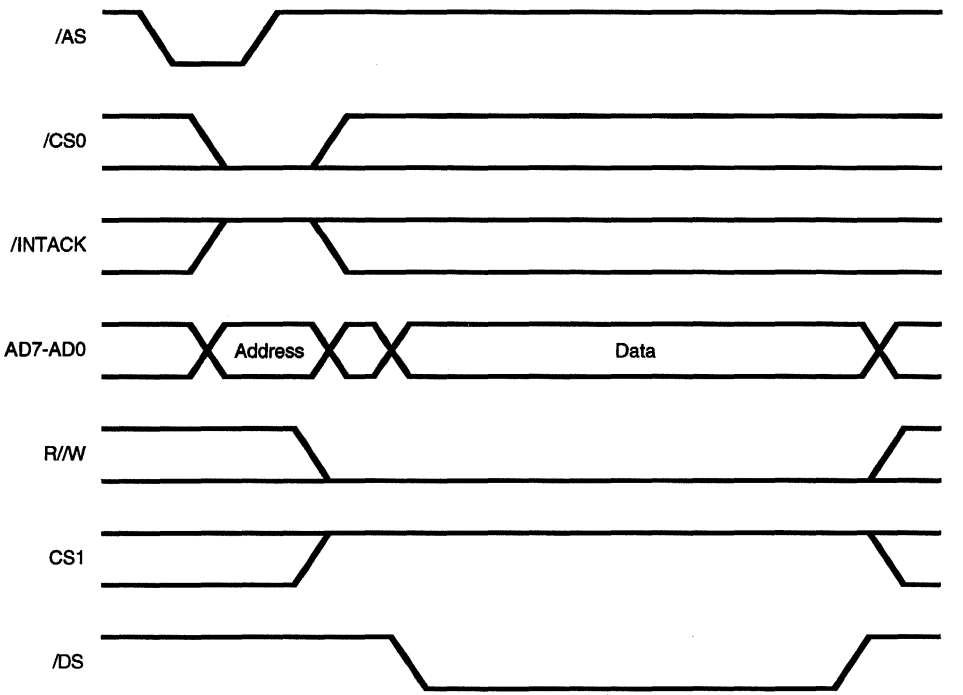


Figure 23. Write Cycle Timing

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.

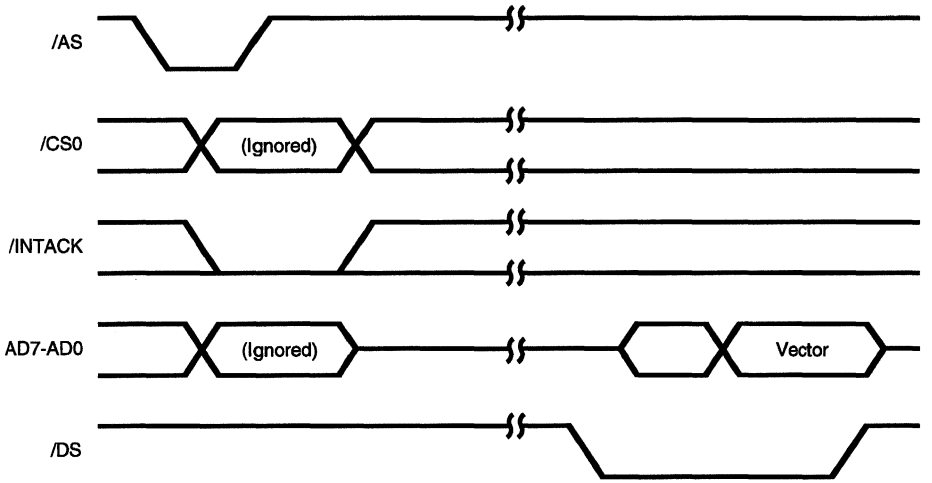


Figure 24. Interrupt Acknowledge Cycle Timing

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
T_A 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

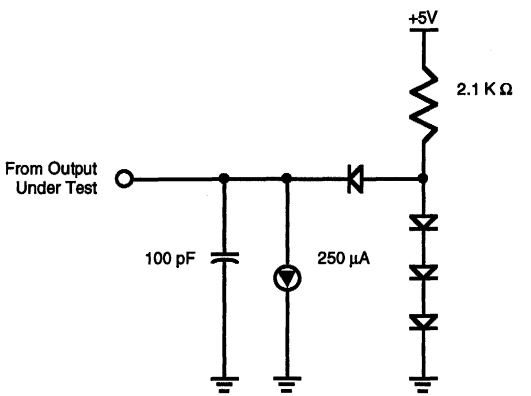


Figure 25. Standard Test Load

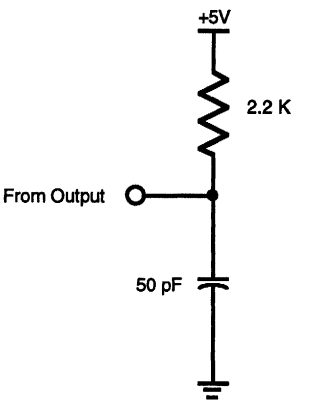


Figure 26. Open-Drain Test Load

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(16.384 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 frequency of oscillation.

AC CHARACTERISTICS

Z85C30 Read/Write Timing Diagrams

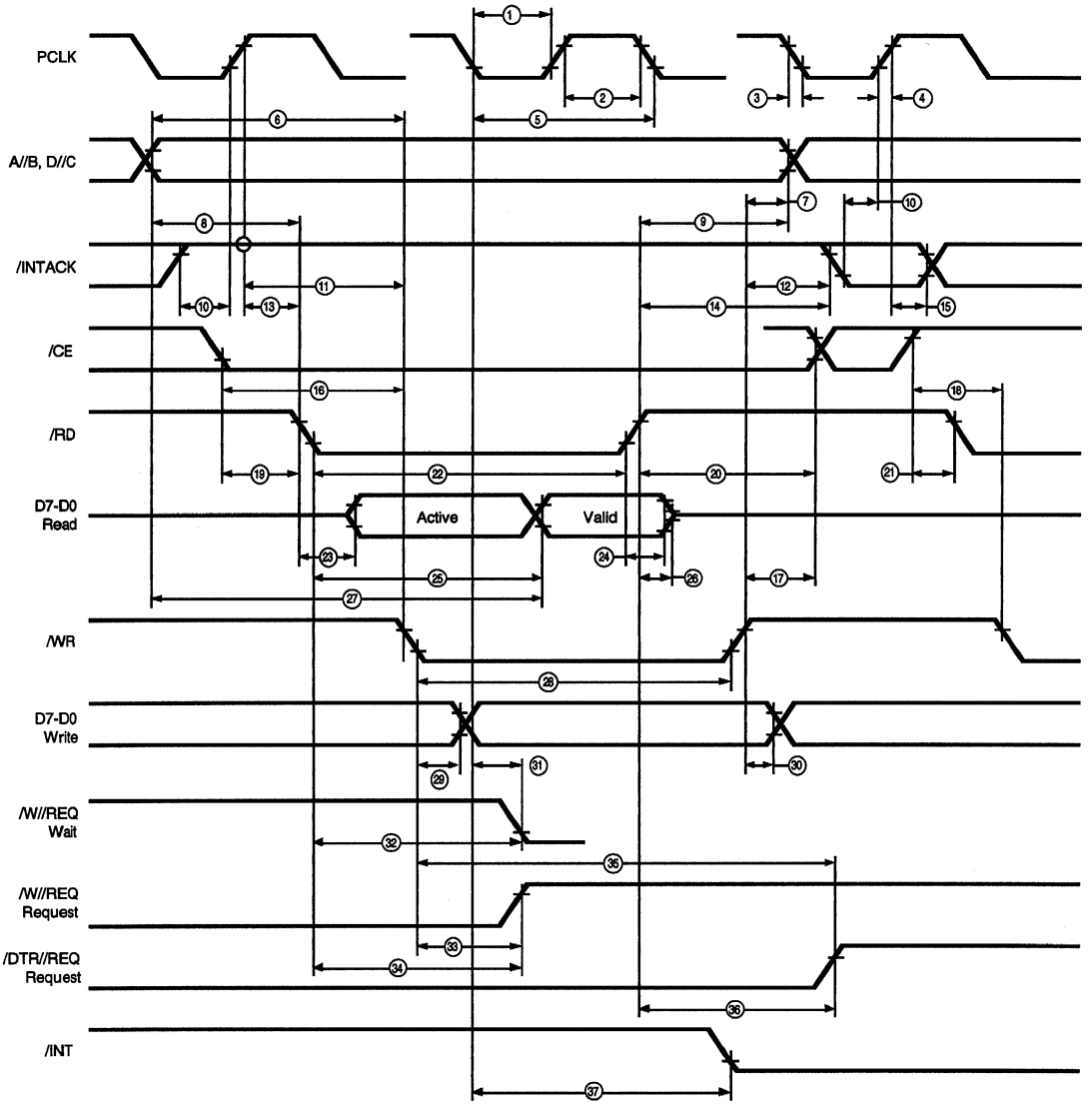


Figure 27. Z85C30 Read/Write Timing Diagram

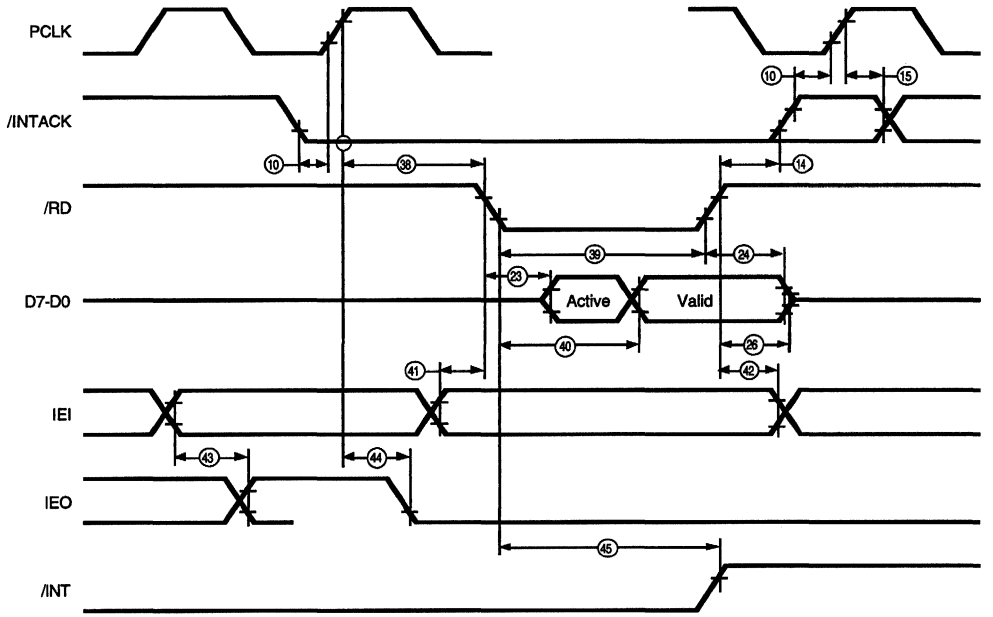


Figure 28. Z85C30 Interrupt Acknowledge Timing Diagram

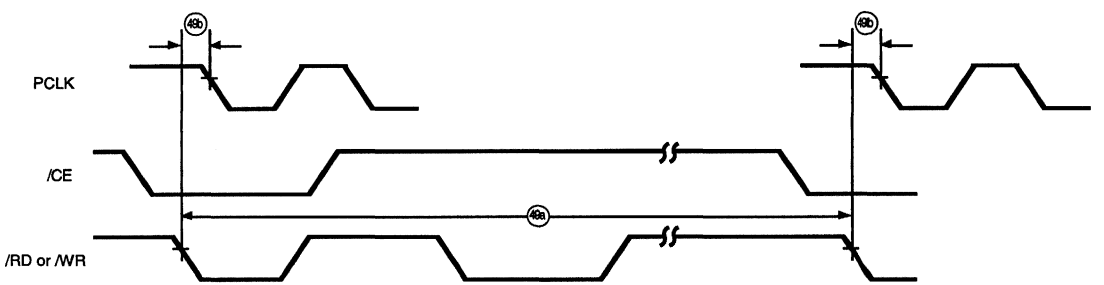


Figure 29. Z85C30 Cycle Timing Diagram

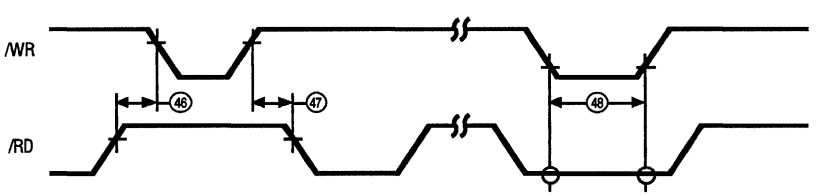


Figure 30. Z85C30 Reset Timing Diagram

AC CHARACTERISTICS

Z85C30 Read/Write Timing Table

			8.5 MHz		10 MHz		16 MHz		
No	**Symbol**	**Parameter**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Notes**
1	TwPCl	PCLK Low Width	45	2000	40	2000	26	2000	
2	TwPCh	PCLK High Width	45	2000	40	2000	26	2000	
3	TfPC	PCLK Fall Time		10		10		5	
4	TrPC	PCLK Rise Time		10		10		5	
5	TcPC	PCLK Cycle Time	118	4000	100	4000	61	4000	
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		120		70		[1]
12	ThlA(WR)	/INTACK to /WR Rise Hold Time	0		0		0		
13	TslAi(RD)	/INTACK to /RD Fall Setup Time	140		120		70		[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 Data Not Valid Delay	0		0		0		
25	TdRDl(DR)	/RD Fall to Read Data Valid Delay		135		120		70	
26	TdRD(DRz)	/RD Rise to Read Data Float Delay		38		35		30	
27	TdA(DR)	Addr to Read Data Valid Delay		210		160		100	
28	TwWRl	/WR Low Width	145		125		75		
29	TdWR(DW)	/WR Fall to Write Data Valid Delay		35		35		20	
30	ThDW(WR)	Write Data to /WR Rise Hold Time	0		0		0		
31	TdWR(W)	/WR Fall to Wait Valid Delay		168		100		50	[4]
32	TdRD(W)	/RD Fall to Wait Valid Delay		168		100		50	[4]
33	TdWRf(REQ)	/WR Fall to /W//REQ Not Valid Delay		168		120		70	
34	TdRDf(REQ)	/RD Fall to /W//REQ Not Valid Delay		168		120		70	[6]

No	Symbol	Parameter	8.5 MHz Min	8.5 MHz Max	10 MHz Min	10 MHz Max	16 MHz Min	16 MHz Max	Notes
35a	TdWRr(REQ)	/WR Fall to /DTR//REQ Not Valid		4TcPc		4TcPc		4TcPc	
35b	TdWRr(REQ)	/WR Fall to /DTR//REQ Not Valid		168		100		70	[6]
36	TdRDr(REQ)	/RD Rise to /DTR//REQ Not Valid Delay		NA		NA		NA	
37	TdPC(INT)	PCLK Fall to /INT Valid Delay		500		320		175	
38	TdIAi(RD)	/INTACK to /RD Fall (Ack) Delay	145		90		50		[5]
39	TwRDA	/RD (Acknowledge) Width	145		125		75		
40	TdRDA(DR)	/RD Fall(Ack) to Read Data Valid Delay	135		120		70		
41	TsIEI(RDA)	IEI to /RD Fall (Ack) Setup Time	95		80		50		
42	ThIEI(RDA)	IEI to /RD Rise (Ack) Hold Time	0		0		0		
43	TdIEI(IEO)	IEI to IEO Delay Time		95		80		45	
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 Delay for No Reset	15		15		10		
47	TdWRQ(RD)	/WR Rise to /RD Fall Delay for No Reset	15		15		10		
48	TwRES	/WR and /RD Low for Reset	145		100		75		
49a	Trc	Valid Access Recovery Time	3.5TcPc	3.5TcPc	3.5TcPc				[3]
49b	Trci	/RD or /WR Fall to PC Fall Setup Time	0		0		0		[7]

Notes:

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

[3] Parameter applies only between transactions involving the Z85C30 SL1480, if WR/RD falling edge is synchronized to PCLK falling edge, then TrC=3TcPc.

[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 TdIEI(IEO) for each device separating them in the daisy chain.

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

[7] This specification is only applicable when Valid Access Recovery Time is less than 3.5 PCLK.

2

AC CHARACTERISTICS

Z85C30 General Timing Diagram

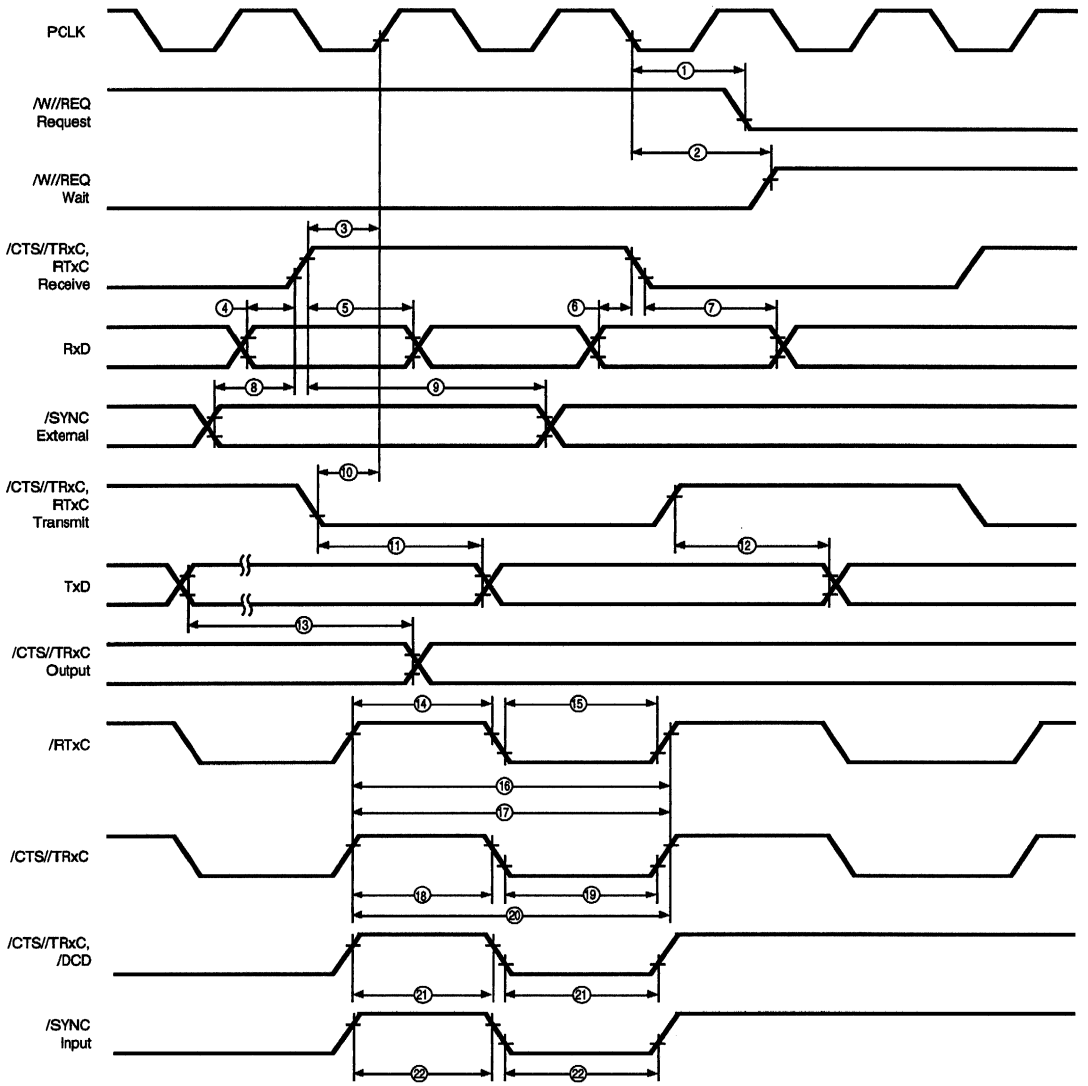


Figure 31. Z85C30 General Timing Diagram

AC CHARACTERISTICS

Z85C30 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 to W/REQ Valid		250		150		80	
2	TdPC(W)	/PCLK to Wait Inactive		350		250		180	
3	TsRXC(PC)	/RxC to /PCLK Setup Time	N/A		N/A		N/A		[1,4]
4	TsRXD(RXCr)	RxD to /RxC Setup Time		0		0		0	[1]
5	ThRXD(RxCr)	RxD to /RXC Hold Time	150		125		50		[1]
6	TsRXD(RXCf)	RxD to /RXC Setup Time	0		0		0		[1,5]
7	ThRXD(RXCf)	RXD to /RXC Hold Time	150		125		50		[1,5]
8	TsSY(RXC)	/SYNC to /RxC Setup Time	–200		–150		–100		[1]
9	ThSY(RXC)	/SYNC to/RXC Hold Time	5TcPc		5TcPc		5TcPc		[1]
10	TsTXC(PC)	/TxC to /PCLK Setup Time	N/A		N/A		N/A		[2,8]
11	TdTXCf(TXD)	/TxC to TxD Delay		200		150		80	[2]
12	TdTxCr(TXD)	/TxC to TxD Delay		200		150		80	[2,5]
13	TdTXD(TRX)	TxD to TRxC Delay		200		140		80	
14a	TwRTXh	RTxC High Width	150		120		80		[6]
14b	TwRTXh(E)	/RTxC High Width	50		40		15.6		[9]
15a	TwRTXl	TRxC Low Width	150		120		80		[6]
15b	TwRTXl(E)	/RTxC Low Width	50		40		15.6		[9]
16a	TcRTX	RTxC Cycle Time	488		400		244		[6,7]
16b	TcRTX(E)	/RTxC Cycle Time	125		100		31.25		[9]
17	TcRTXX	Crystal Osc. Period	125	1000	100	1000	62	1000	[3]
18	TwTRXh	/TRxC High Width	150		120		180		[6]
19	TwTRXl	/TRxC Low Width	150		120		80		[6]
20	TcTRX	/TRxC Cycle Time	488		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] External PCLK to /RTxC or /RTxC synchronization requirement eliminated for PCLK divide-by-four operation.

/TRxC and /RTxC rise and fall times are identical to PCLK. Reference timing specs TfPC and TrPC.

Tx and Rx input clock slew rates should be kept to a maximum of 30 nsec. All parameters related to input CLK edges should be referenced at the point at which the transition begins or ends, whichever is worst case.

[9] ENHANCED FEATURE - /RTxC used as input to internal DPLL only.

AC CHARACTERISTICS

Z85C30 System Timing Diagram

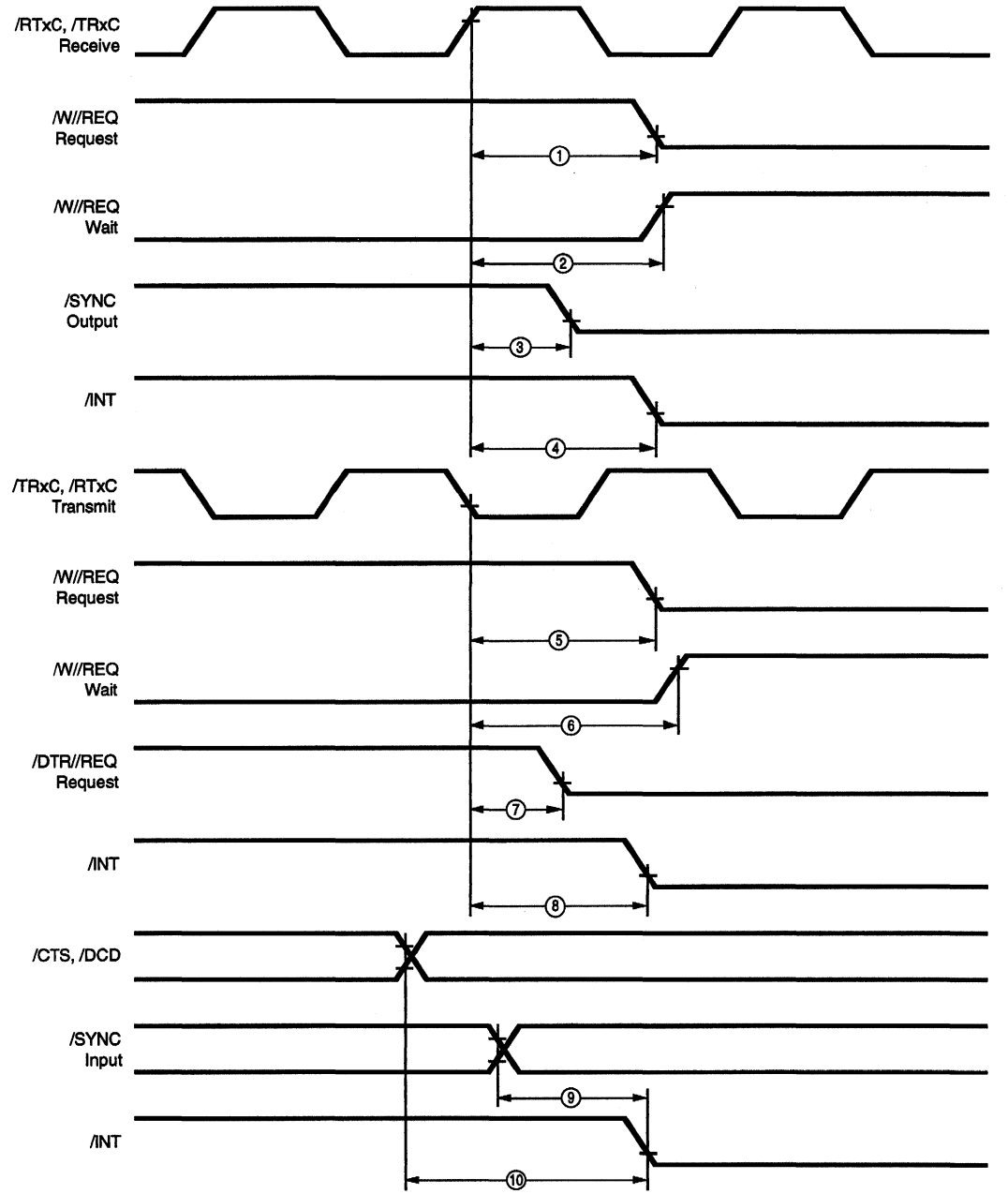


Figure 32. Z85C30 System Timing Diagram

AC CHARACTERISTICS

Z85C30 System Timing Table

No	Symbol	Parameter	8.5 MHz Min	8.5 MHz Max	10 MHz Min	10 MHz Max	16.384 MHz Min	16.384 MHz Max	Notes*
1	TdRXC(REQ)	/RxC High to /W//REQ Valid	8	12	8	12	8	12	[2,5]
2	TdRXC(W)	/RxC High to Wait Inactive	8	14	8	14	8	14	[1,2,5]
3	TdRdXC(SY)	/RxC High to /SYNC Valid	4	7	4	7	4	7	[2,5]
4	TsRXC(INT)	/RxC High to INT Valid	10	16	10	16	10	16	[1,2,5]
5	TdTXC(REQ)	/TxC Low to /W//REQ Valid	5	8	5	8	5	8	[3,5]
6	TdTXC(W)	/TxC Low to Wait Inactive	5	11	5	11	5	11	[1,3,5]
7	TdTXC(DRQ)	/TxC Low to /DTR//REQ Valid	4	7	4	7	4	7	[3,5]
8	TdTXC(INT)	/TxC Low to /INT Valid	6	10	6	10	6	10	[1,3,5]
9a	TdSY(INT)	SYNC to INT Valid	2	6	2	6	2	6	[1,5]
9b	TdSY(INT)	SYNC to INT Valid	2	3	2	3	2	3	[1,4,5]
10	TdEXT(INT)	/DCD or /CTS to /INT Valid	2	6	2	6	2	6	[1,5]

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.
[5] Units equal to TcPc.

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	2000	40	2000	26	2000	
2	TwPCh	PCLK High Width	45	2000	40	2000	26	2000	
3	TfPC	PCLK Fall Time		10		10		5	
4	TrPC	PCLK Rise Time		10		10		5	
5	TcPC	PCLK Cycle Time	118	4000	100	4000	61	4000	
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		

AC CHARACTERISTICS

Z80C30 Read and Write Timing Diagrams

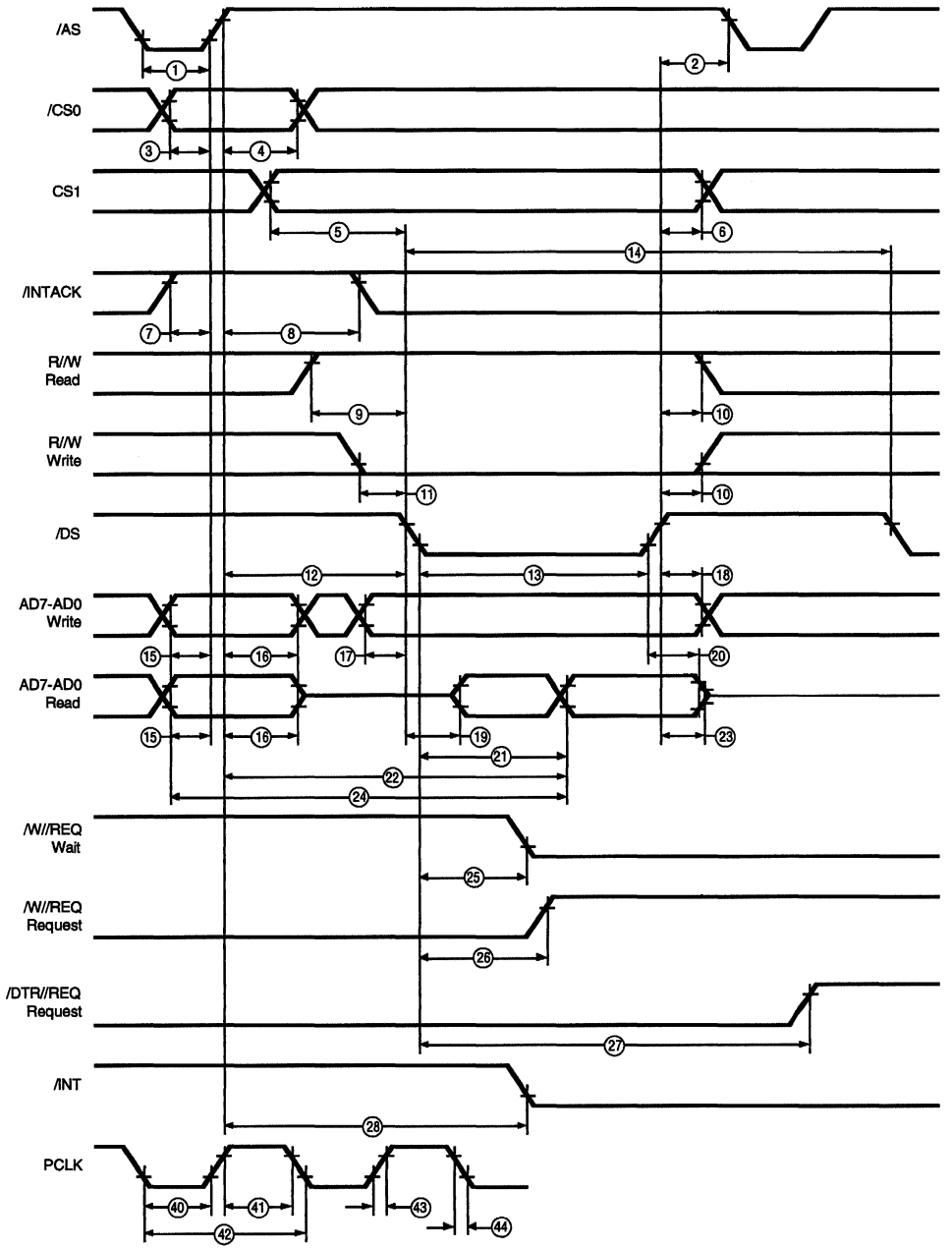


Figure 33. Z80C30 Read/Write Timing Diagram

AC CHARACTERISTICS

Z80C30 Read and Write Timing Diagrams

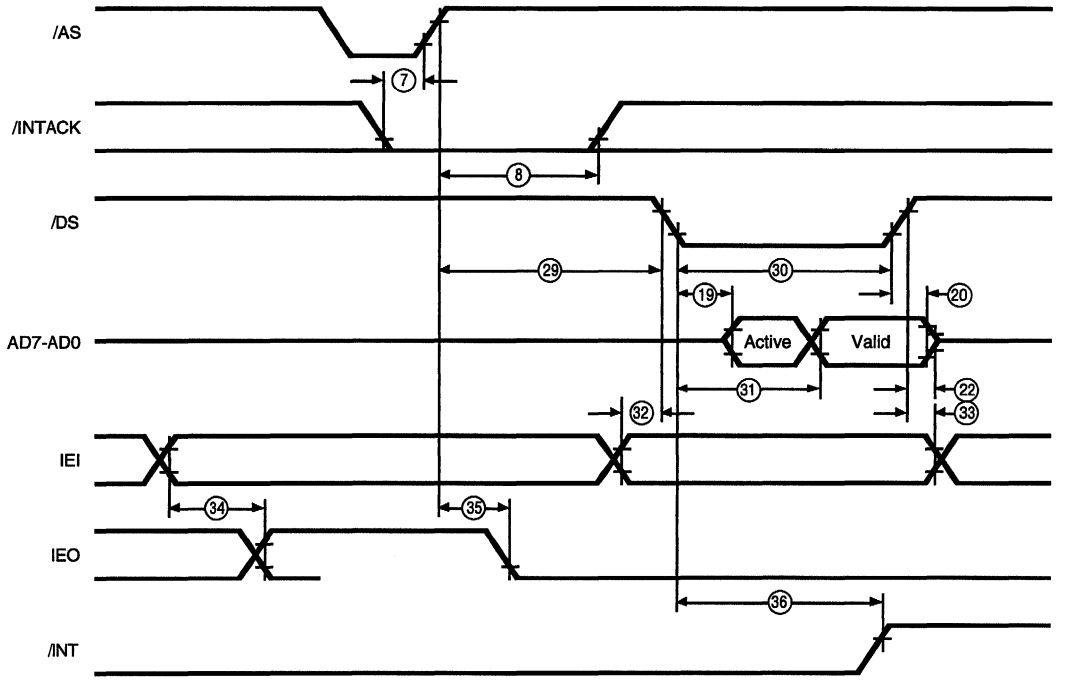


Figure 34. Z80C30 Interrupt Acknowledge Timing Diagram

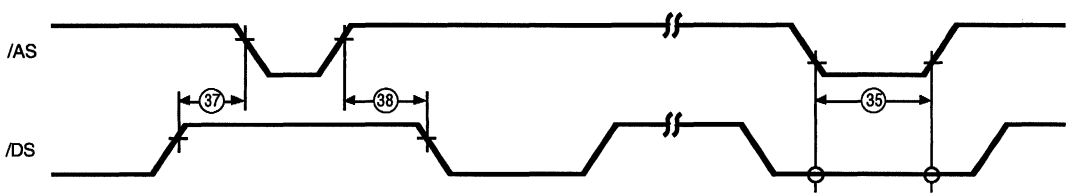


Figure 35. Z80C30 Reset Timing Diagram

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 /DTR//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	TsIEI(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	

2

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. 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) otherwise noted.

AC CHARACTERISTICS

Z80C30 General Timing Diagram

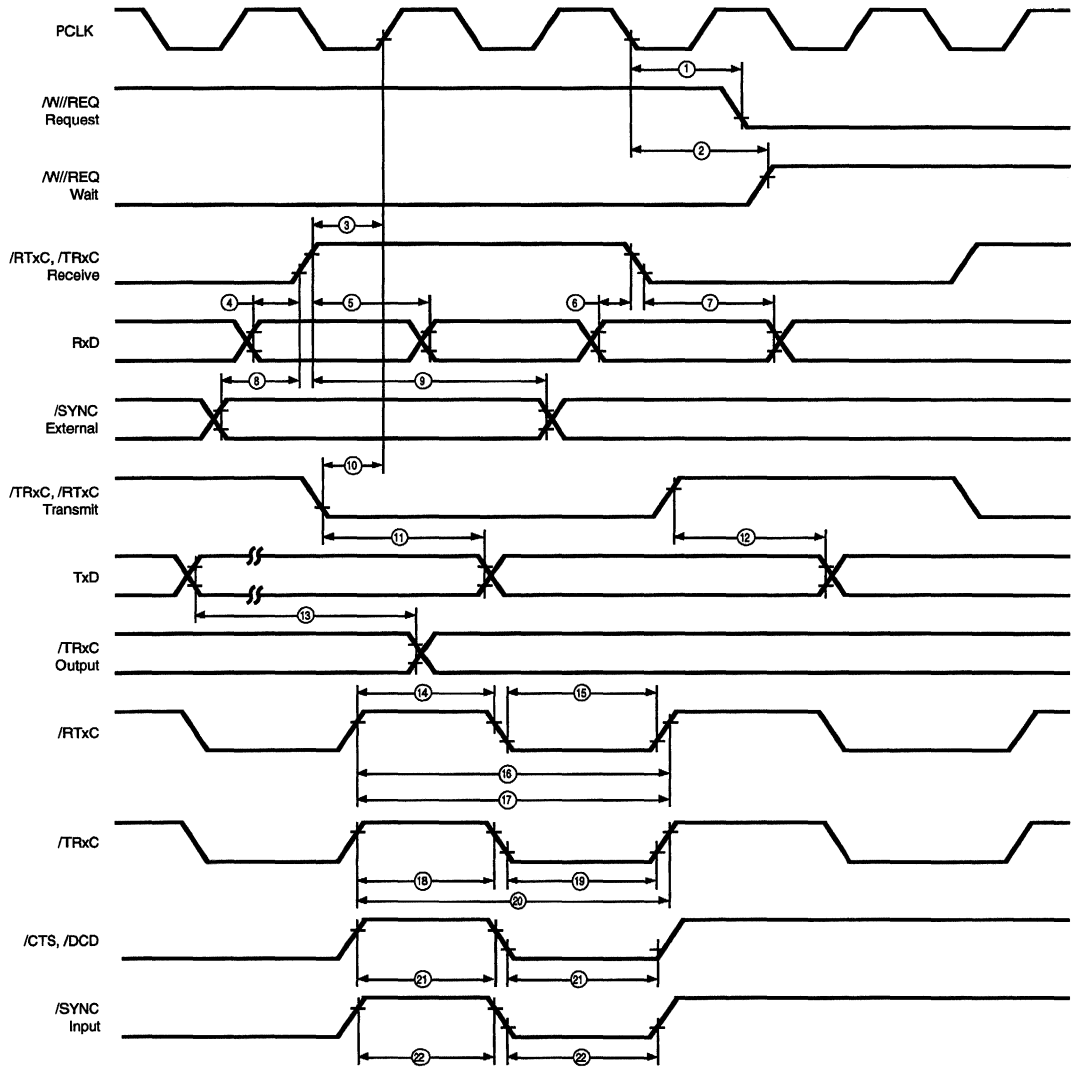


Figure 36. Z80C30 General Timing Diagram

AC CHARACTERISTICS

Z80C30 General Timing Table

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

AC CHARACTERISTICS

Z80C30 System Timing Diagram

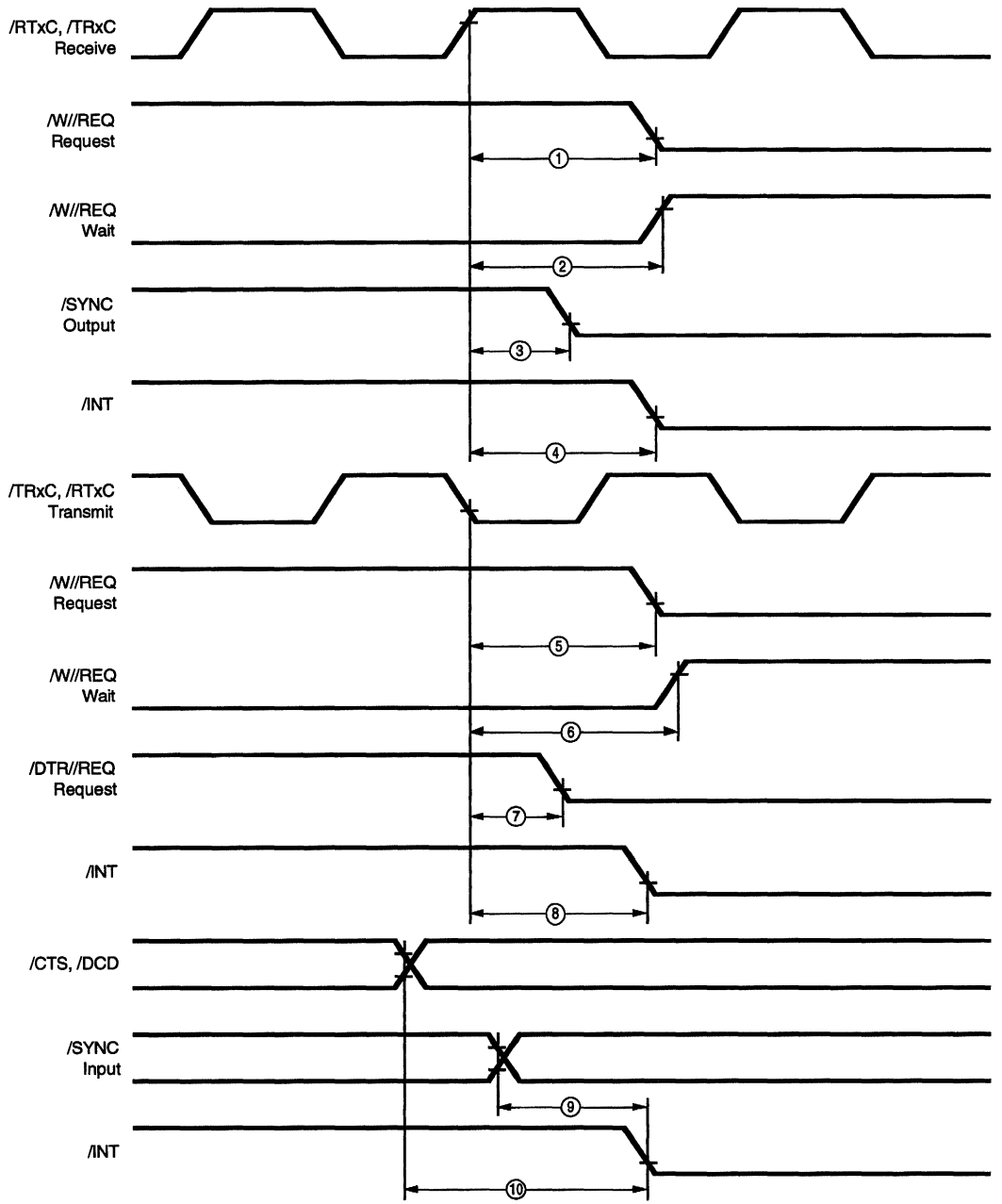


Figure 37. Z80C30 System Timing Diagram

AC CHARACTERISTICS

Z80C30 System Timing Table

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

Notes:

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

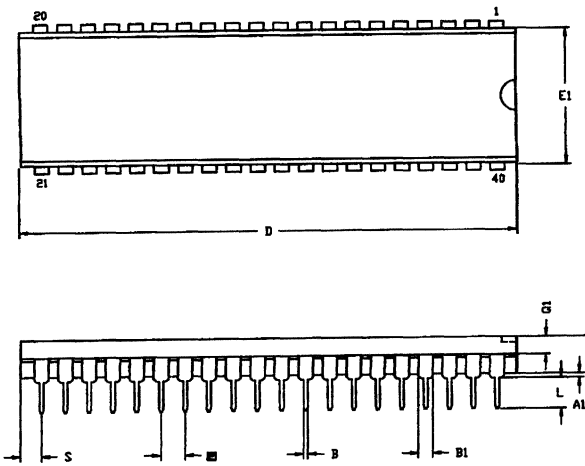
[2] /RxC is /RTxC or /TRxC, whichever is supplying the receive clock.

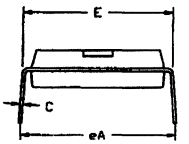
[3] /TxC is /TRxC or /RTxC, whichever is supplying the transmit clock.

[4] Units equal to /AS.

[5] Units equal to TcPc.

PACKAGE INFORMATION

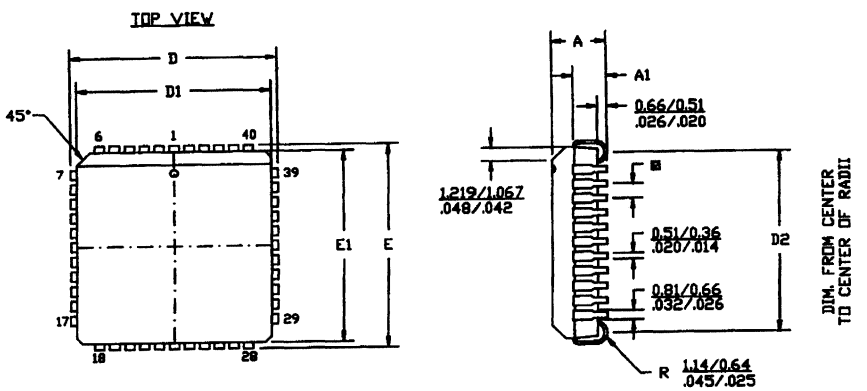




SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A1	0.51	0.81	.020	.032
A2	3.25	3.43	.128	.135
B	0.38	0.53	.015	.021
B1	1.02	1.52	.040	.060
C	0.23	0.38	.009	.015
D	52.07	52.58	2.050	2.070
E	15.24	15.75	.600	.620
E1	13.59	14.22	.535	.560
e	2.54 TYP		.100 TYP	
eA	15.49	16.51	.610	.650
L	3.18	3.81	.125	.150
Q1	1.52	1.91	.060	.075
S	1.52	2.29	.060	.090

CONTROLLING DIMENSIONS : INCH

40-Pin DIP Package Diagram



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

SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	4.27	4.57	.168	.180
A1	2.67	2.92	.105	.115
D/E	17.40	17.65	.685	.695
D1/E1	16.51	16.66	.650	.656
D2	15.24	16.00	.600	.630
e	1.27 TYP		.050 TYP	

44-Pin PLCC Package Diagram

ORDERING INFORMATION

Z80C30/Z85C30

8 MHz	10 MHz	16 MHz
Z80C3008PSC	Z80C3010PSC	Z85C3016PSC
Z80C3008VSC	Z80C3010VSC	Z85C3016VSC
Z85C3008PSC/PEC	Z85C3010PSC/PEC	
Z85C3008VSC/VEC	Z85C3010VSC/VEC	

Package
P = Plastic DIP
V = Plastic Leaded Chip Carrier
D = Ceramic DIP

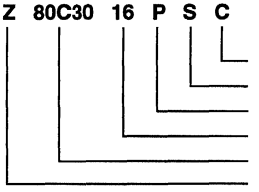
Temperature
E = –40°C to +100°C
S = 0° to +70°C

Speeds
8 = 8 MHz
10 = 10 MHz
16 = 16 MHz

Environmental
C = Plastic Standard

Example:

Z 80C30 16 P S C is a Z80C30, 16 MHz, PLCC, 0°C to +70°C, Plastic Standard Flow



Z80230

Z-BUS™ 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 Z80C30 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.
- 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.

GENERAL DESCRIPTION

The Zilog Enhanced Serial Communications Controller, Z80230 Z-Bus™ ESCC, is a pin and software compatible CMOS member of the SCC family (The SCC was introduced by Zilog in 1981.). The ESCC is a dual-channel, full-duplex datacommunications 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 10 x 19 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 (Figure 1).

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

Power connections follow conventional descriptions below:

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

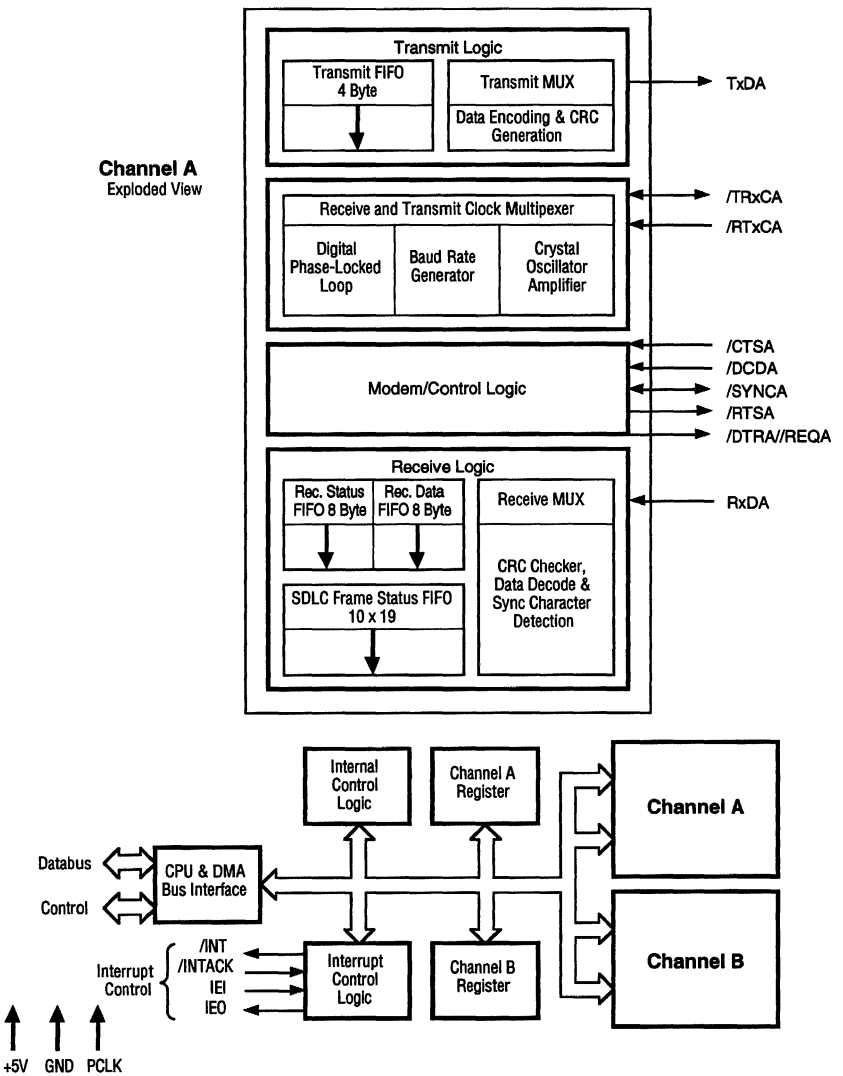


Figure 1. ESCC Block Diagram

PIN FUNCTIONS

The following section describes the Z80230 pin functions. Figures 2 and 3 detail the pin assignments for the 40-pin DIP and 44-pin PLCC packages. The Z80230 ESCC is socket compatible with the Zilog Z8030 and Z80230 as the pin electrical characteristics and pin assignments are the same. Any unused input pins should be pulled up to the +5V supply.

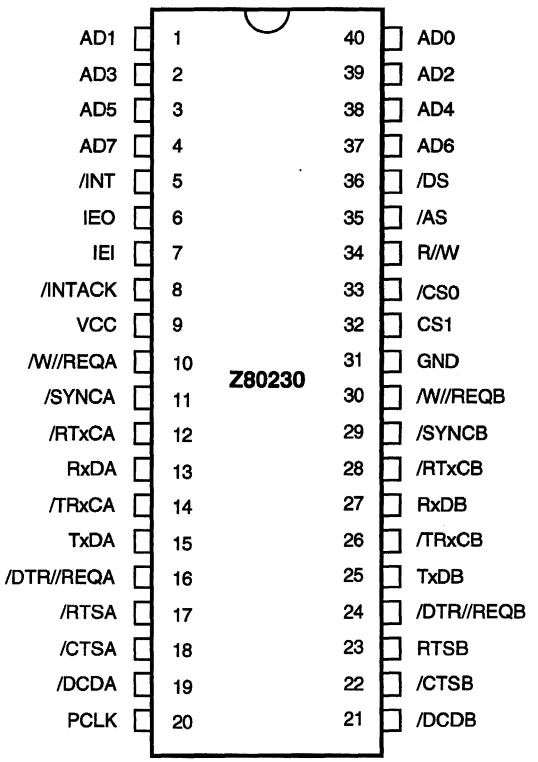


Figure 2. Z80230 DIP Pin Assignments

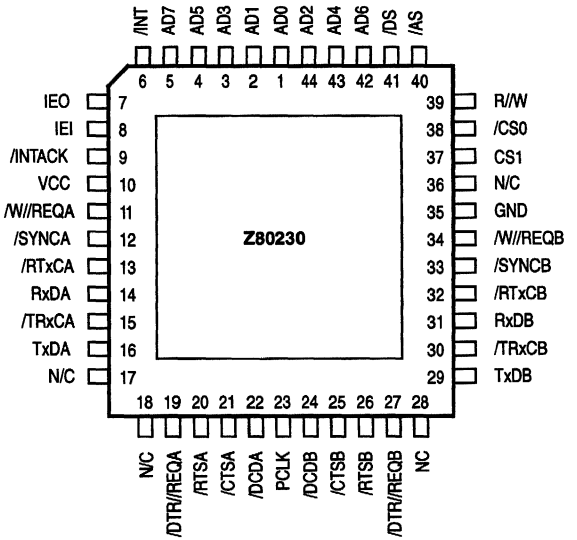


Figure 3. Z80230 PLCC Pin Assignments

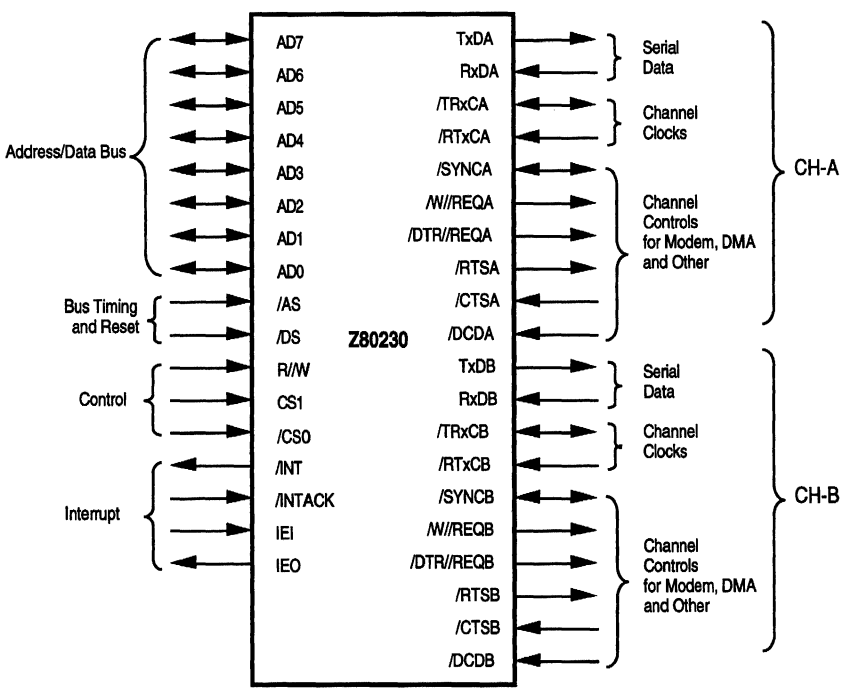


Figure 4. Z80230 Pin Functions

3

PIN DESCRIPTIONS

/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, RxD *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.

Z80230

AD7-AD0 Address/Data Bus (bidirectional, active High, tri-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

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 functions 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 are 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.

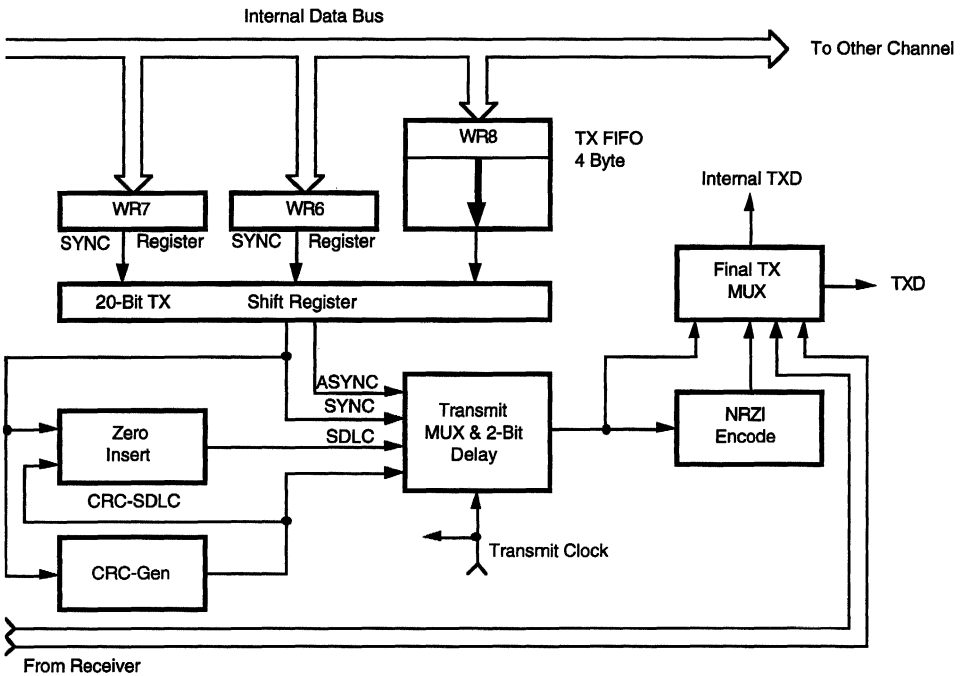


Figure 5. ESCC Transmit Data Path

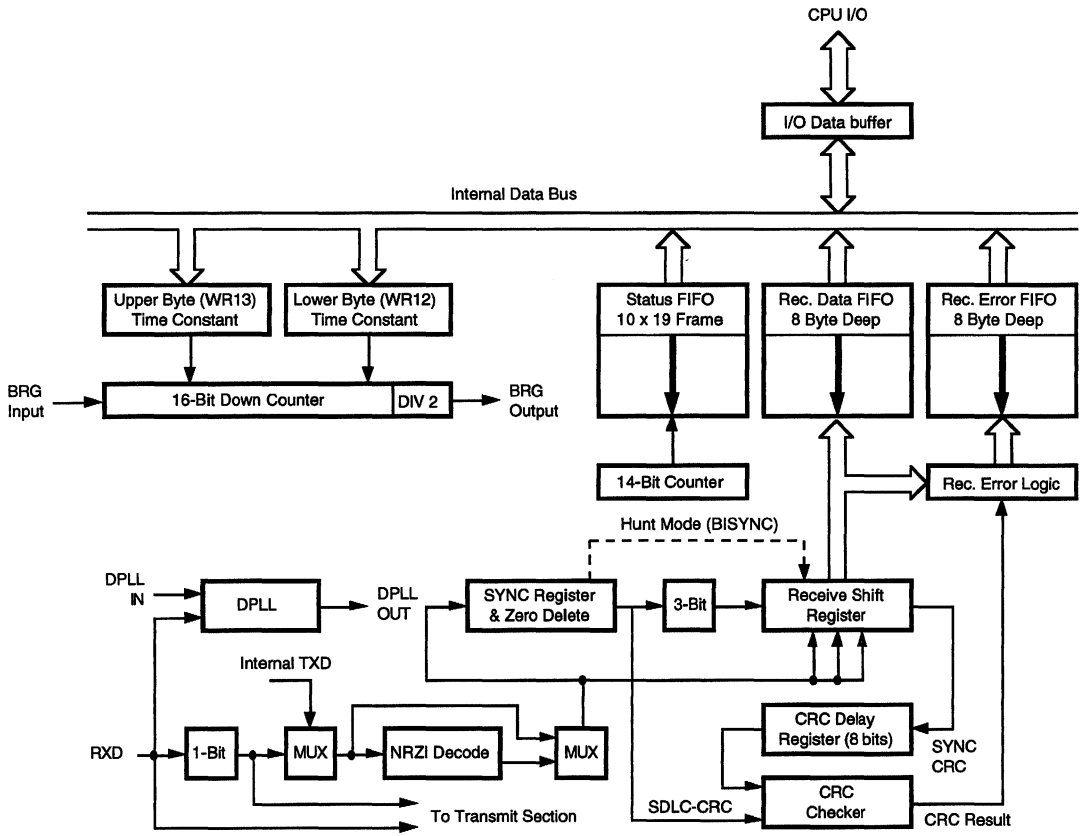


Figure 6. ESCC Receive Data Path

I/O INTERFACE CAPABILITIES

System communication to and from the ESCC is done through the ESCC's register set. There are 17 write registers and 15 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: Select Shift Left/Right Mode, 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 10 x 19 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.

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.

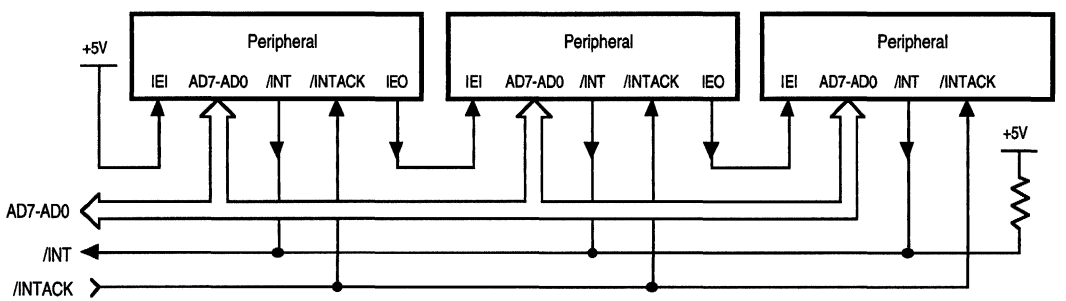


Figure 7. ESCC Interrupt Priority Schedule

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 Z80230 Enhancements 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:

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.

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 uses 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 datacommunication protocols (Figure 8). Each of the datacommunication channels has identical features and capabilities.

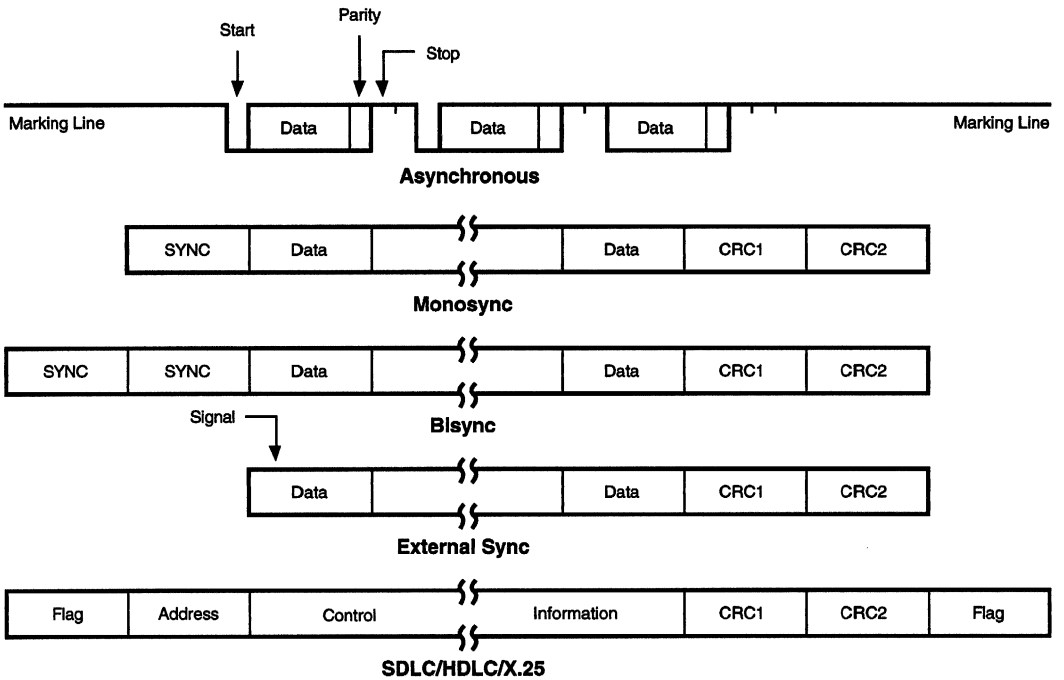


Figure 8. Some ESCC Protocols

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 FIFOs 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 Z80230 Enhancements 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.

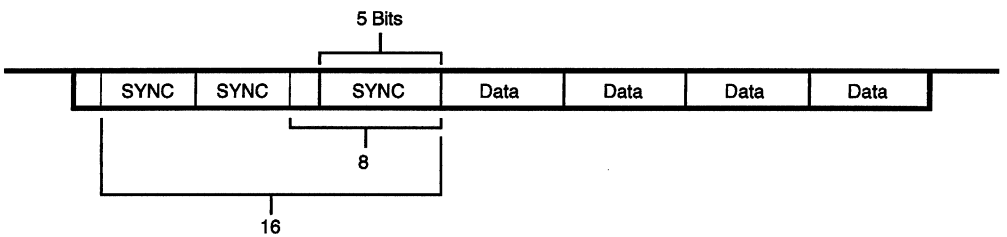


Figure 9. Detecting 5- or 7-Bit Synchronous Characters

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 1s or all 0s. 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 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 ESCC 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.

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.

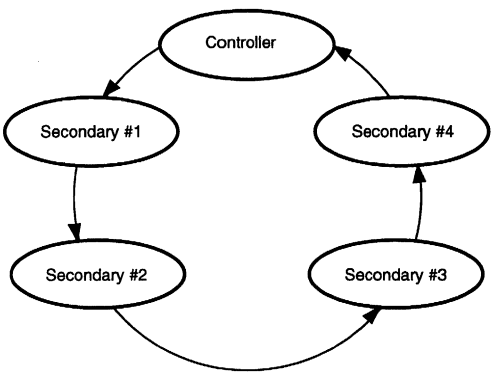


Figure 10. An SDLC Loop

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-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 6 and 7 are only accessible when the SDLC FIFO is enabled. The 10 x 19 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 start-up, 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.

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

ESCC DATA COMMUNICATIONS CAPABILITIES (Continued)

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.

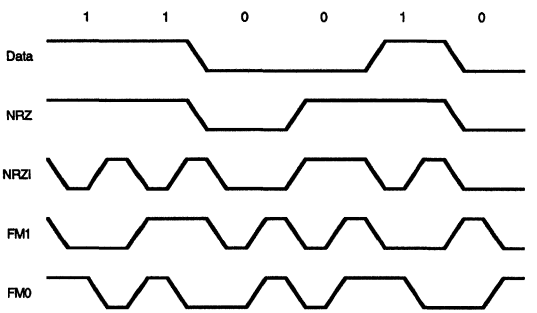


Figure 11. Data Encoding Methods

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.

Z80230 ENHANCEMENTS

The following is a detailed description of the enhancements to the Z80230, 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 IS 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.

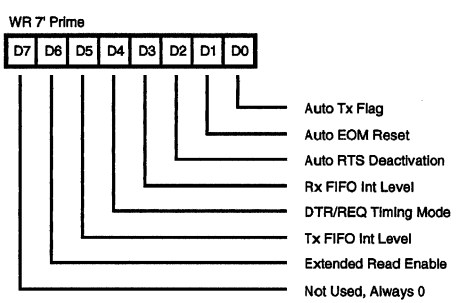


Figure 12. Write Register 7' (7 prime)

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.

Z85230 ENHANCEMENTS (Continued)

For applications which may use either the Zilog Z80C30 or Z80230, these two device types can be identified in software with the following test. Write a 01 hex to Write Register 15. Then Read Register 15 and if D0 is reset it is a Z80C30 and, if D0 is set it is a Z80230. Note that if the device is Z80C30, a write to WR15 resetting D0 should be done before proceeding. Also, if the device is Z80230, 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.

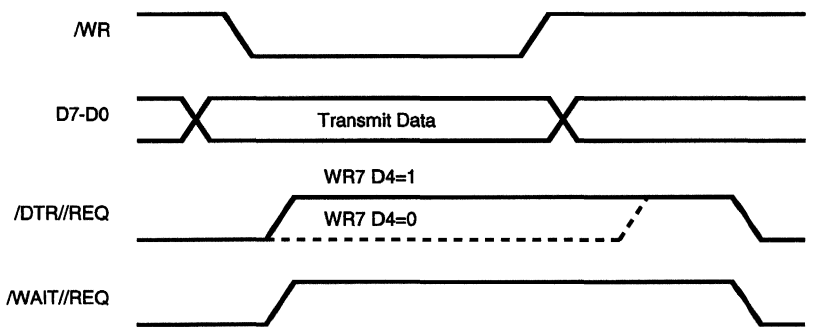


Figure 13. DMA Request on Transmit Deactivation Timing

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.

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.

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.

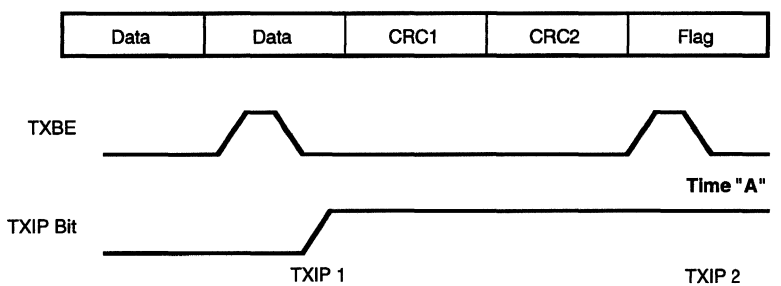


Figure 14. TxIP Latching

NEW FEATURE DESCRIPTION (Continued)

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).

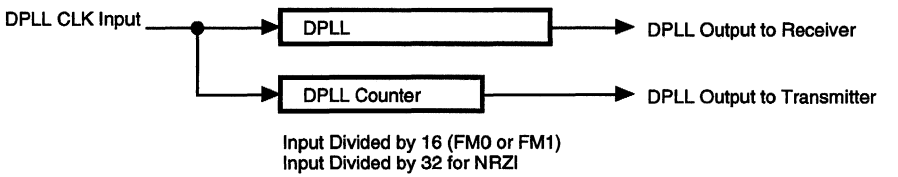


Figure 15. DPLL Outputs

Read Register 0 Status Latched During Read Cycle

The contents of Read Register 0, 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 Z80230 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 V_{cc} through a resistor (2.2 kOhm 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 Z80230 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-bit deep by 19-bit wide 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 8-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 and status will be 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 Z8030. 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 details on 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. 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 (Figure 16). 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. Reading the frame status (CRC, byte count and other status stored in the status FIFO) to determine EOF is not required.

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.

NEW FEATURE DESCRIPTION (Continued)

Frame Status FIFO Circuitry

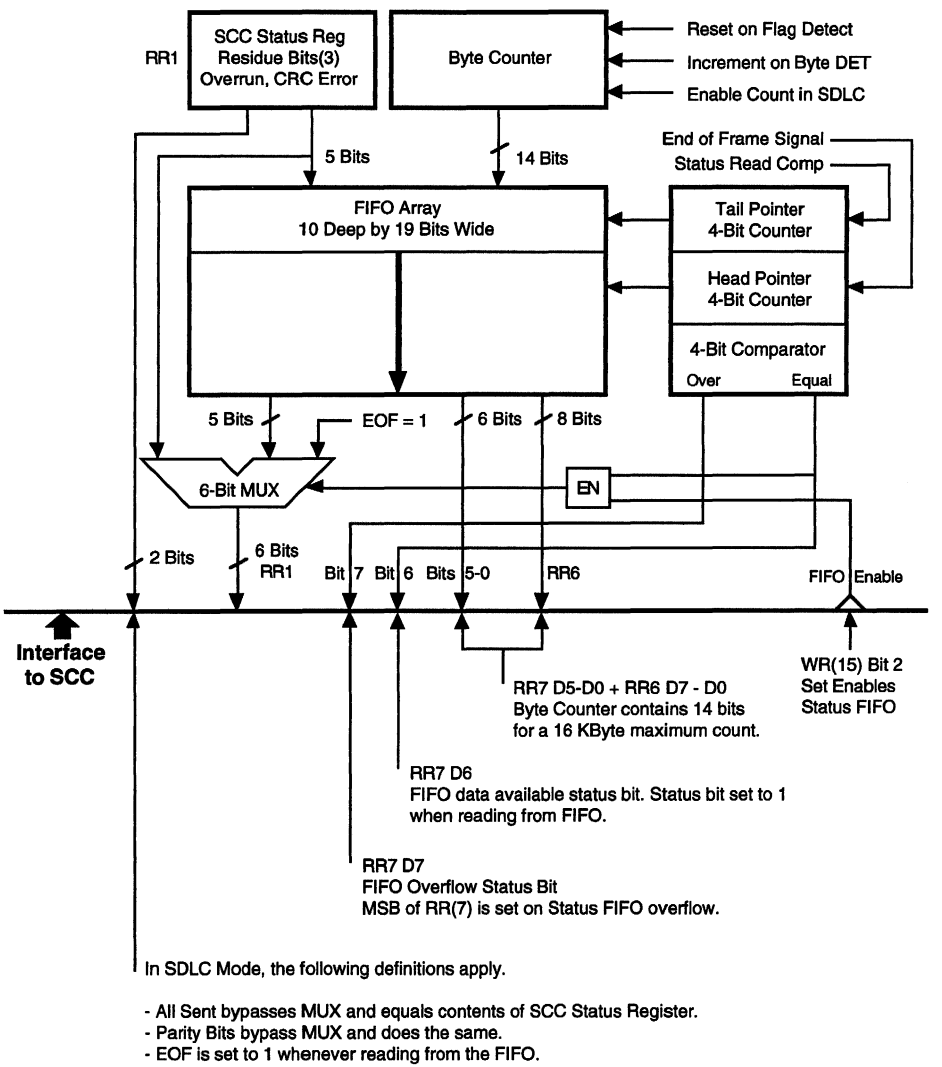


Figure 16. SDLC Frame Status FIFO

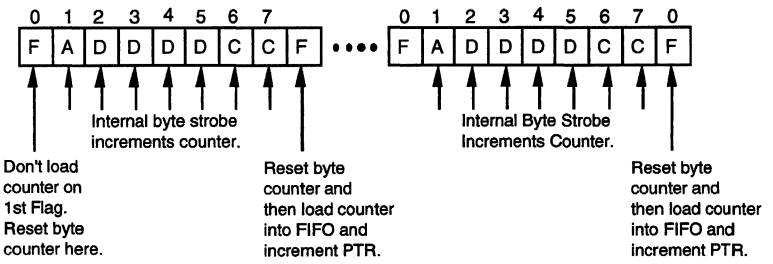


Figure 17. SDLC Byte Counting Detail

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 19 shows the format of each Read register.

3

CONTROL REGISTERS

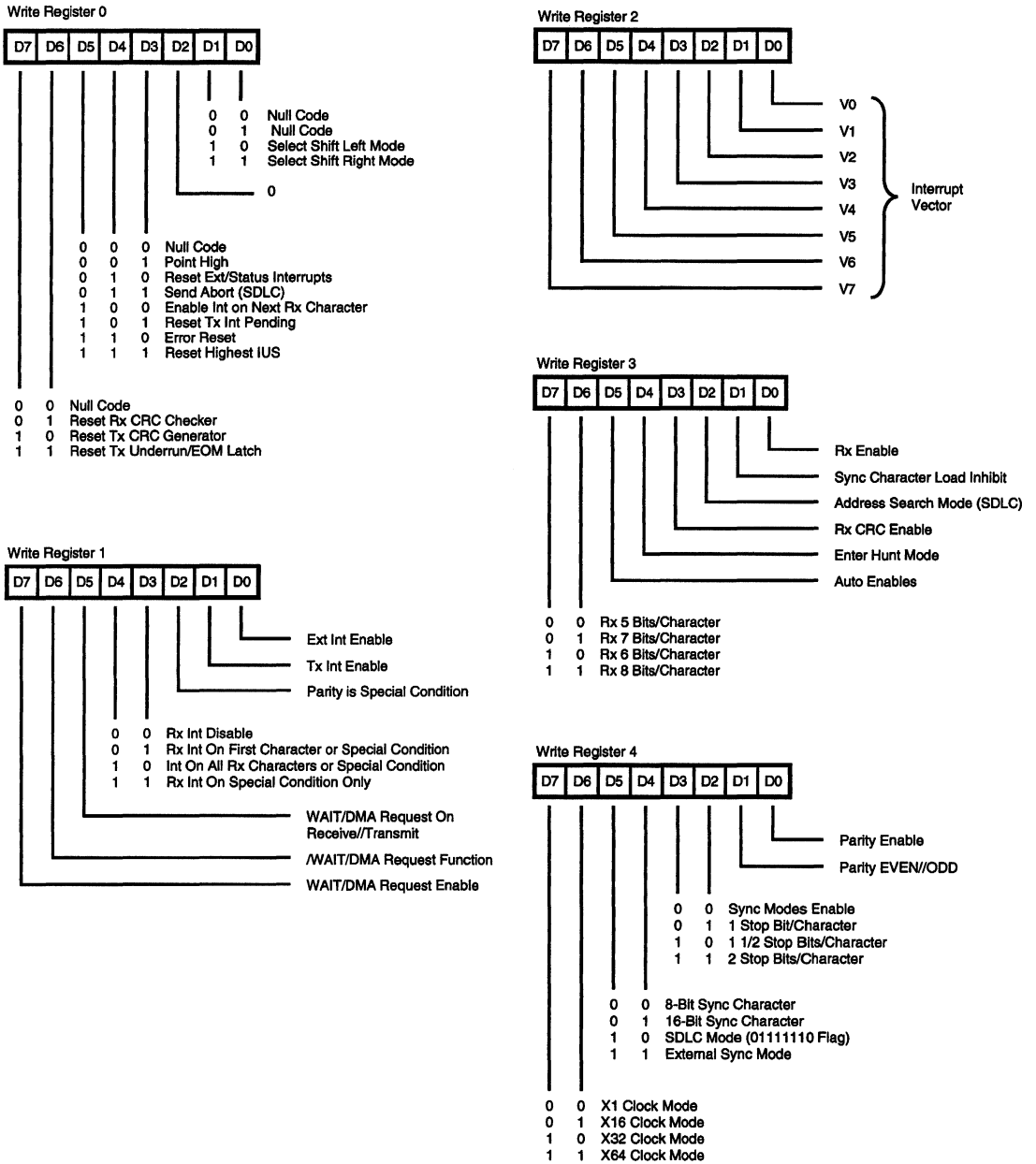
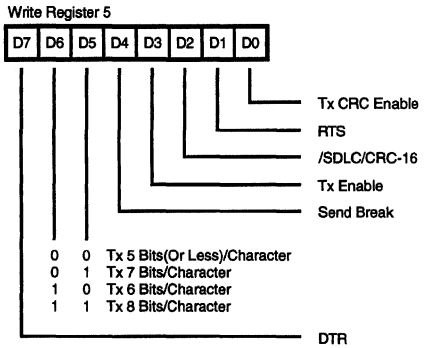
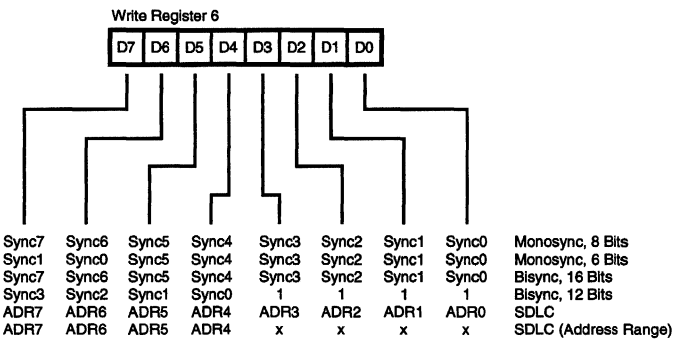


Figure 18. Write Register Bit Functions





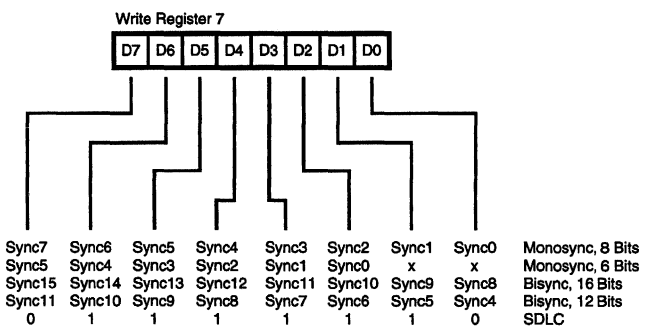


Figure 18. Write Register Bit Functions (Continued)

CONTROL REGISTERS (Continued)

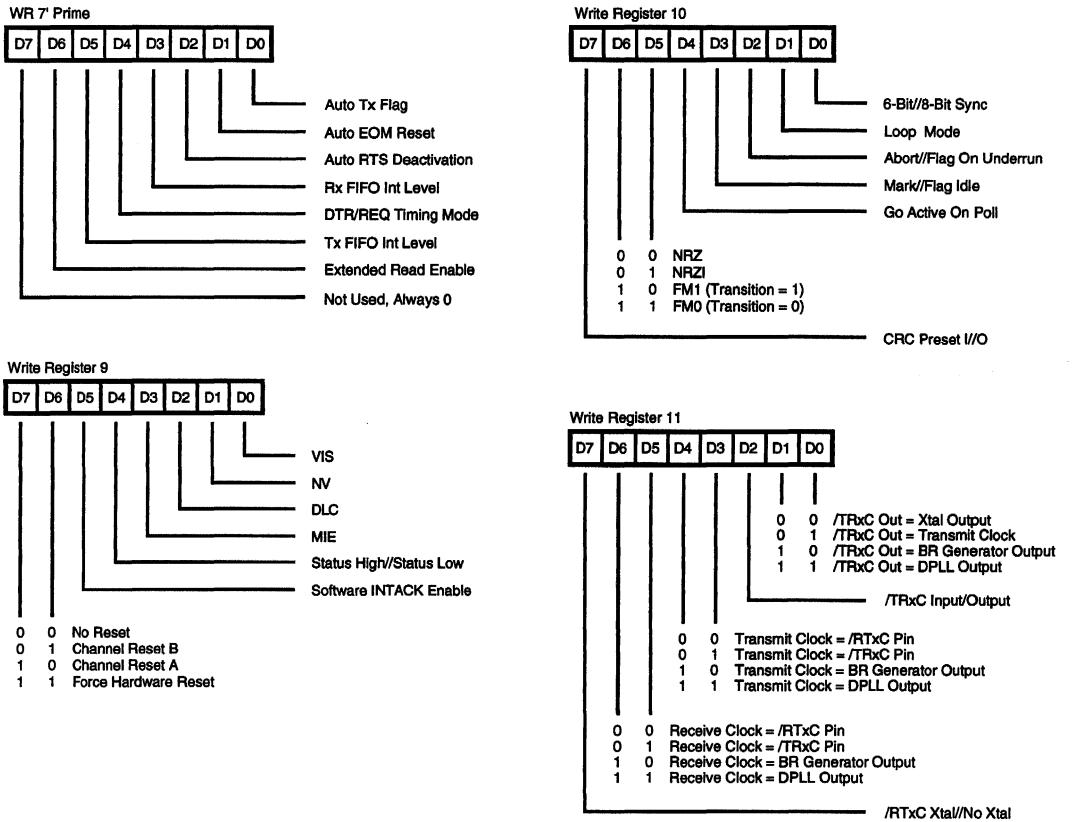


Figure 18. Write Register Bit Functions (Continued)

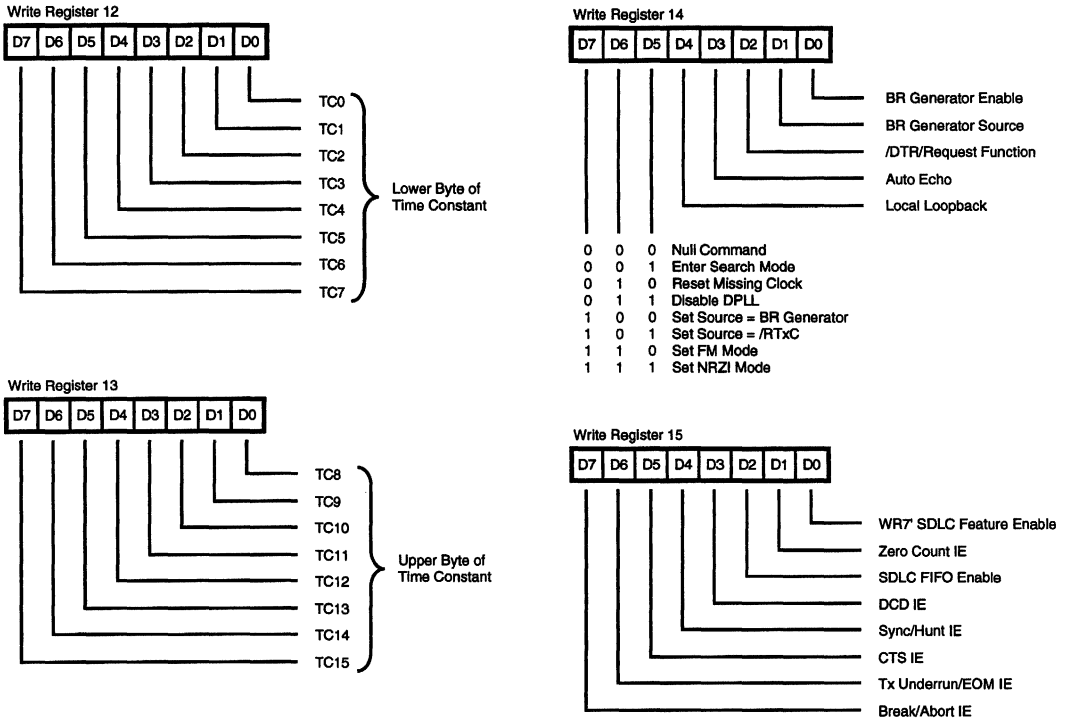
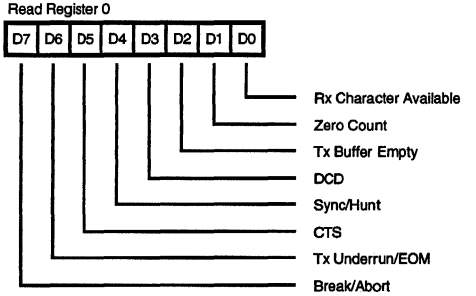


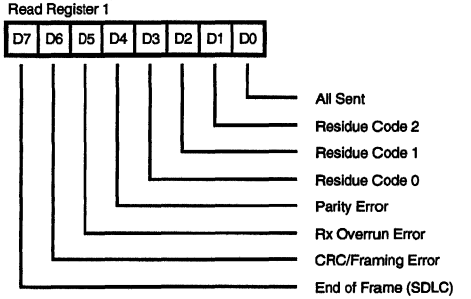
Figure 18. Write Register Bit Functions (Continued)

CONTROL REGISTERS (Continued)

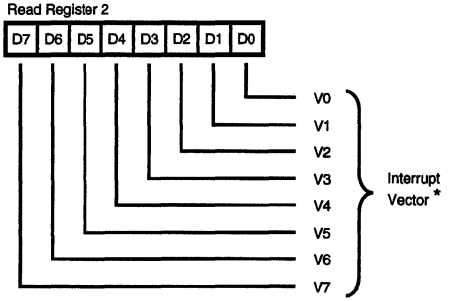
Read Register 0



Read Register 1

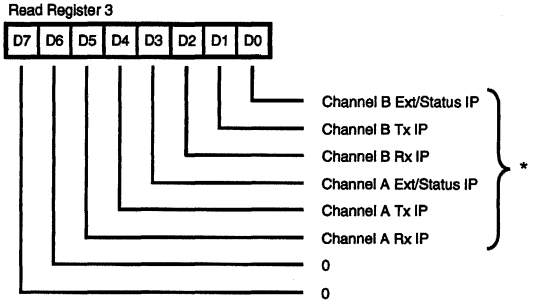


Read Register 2



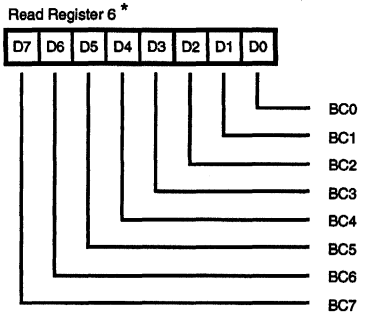
* Modified In B Channel

Read Register 3



* Always 0 In B Channel

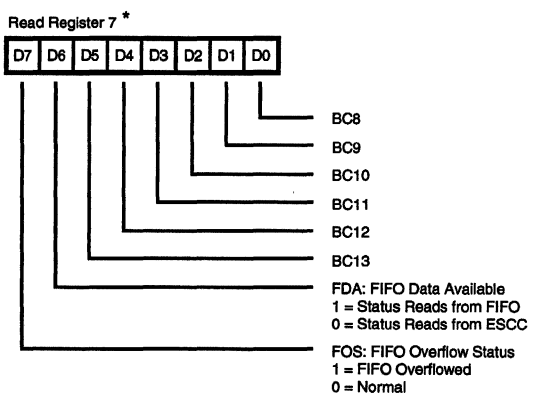
Read Register 6 *



* Can only be accessed if the SDLC FIFO enhancement is enabled (WR15 bit D2 set to 1)

SDLC FIFO Status and Byte Count (LSB)

Read Register 7 *



* Can only be accessed if the SDLC FIFO enhancement is enabled (WR15 bit D2 set to 1)

SDLC FIFO Status and Byte Count (MSB)

Figure 19. Read Register Bit Functions

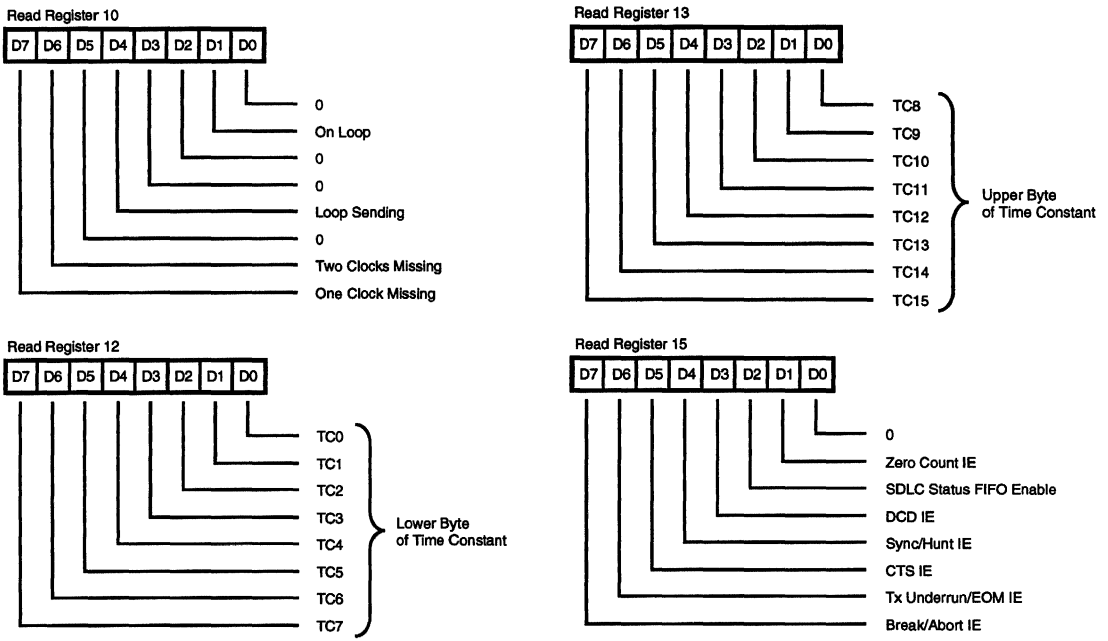


Figure 19. Read Register Bit Functions (Continued)

3

Z80230 INTERFACE TIMING

The Z-Bus compatible ESCC is suited for system applications with multiplexed address/data buses similar to the Z8®, Z8000®, Z280®.

Two control signals, /AS and /DS, are used by the Z80230 to time bus transactions. In addition, four other control signals (/CS0, CS1, R/W, and /INTACK) are used to control the type of bus transaction that occurs. A bus transaction is initiated by /AS; the rising edge latches the register address on the Address/Data bus and the state of /INTACK and /CSO.

In addition to timing bus transactions. /AS is used by the interrupt section to set the Interrupt Pending (IP) bits. Because of this, /AS must be kept cycling for the interrupt section to function properly.

The Z80230 generates internal control signals in response to a register access. Since /AS and /DS have no phase relationship with PCLK, the circuitry generating these internal control signals provide time for metastable condition to disappear. This results in a recovery time related to PCLK.

This recovery time applies only to transactions involving the Z80230, and any intervening transactions are ignored. This recovery time is four PCLK cycles, measure from the falling edge of /DS of one access to the ESCC, to the falling edge of /DS for a subsequent access.

Z80230 Read Cycle Timing

The Read cycle timing for the Z80230 is shown in Figure 20. The register address on AD7-AD0, as well as the state of /CS0 and /INTACK, are latched by the rising edge of /AS. R/W must be HIGH before /DS falls to indicate a Read cycle. The Z80230 data bus drivers are enabled while CS1 is HIGH and /DS is LOW.

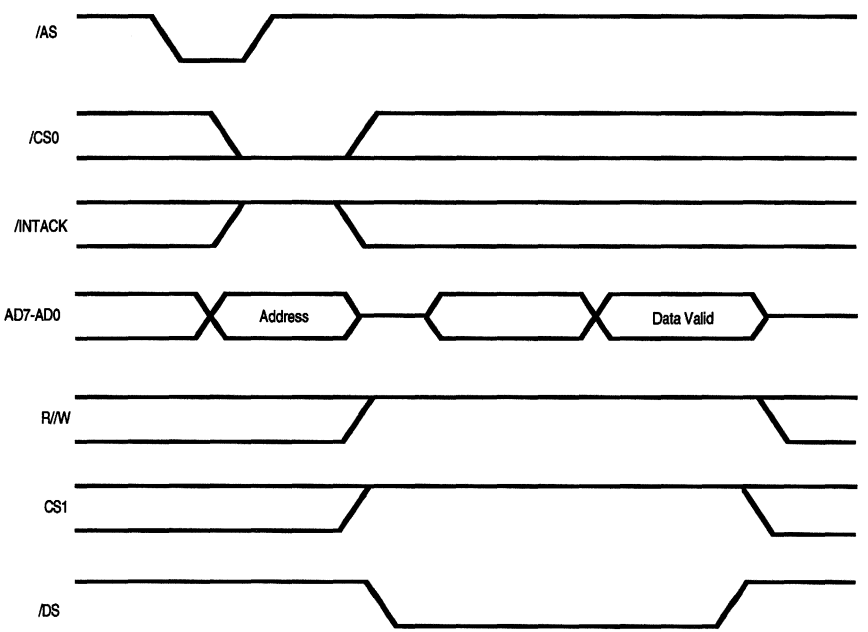


Figure 20. Read Cycle Timing

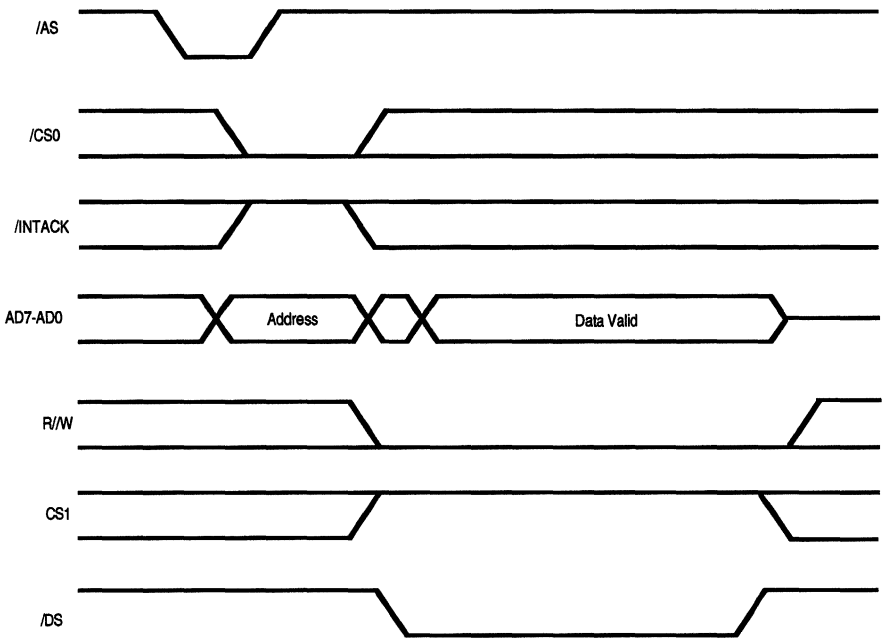


Figure 21. Write Cycle Timing

Z80230 Interrupt Acknowledge Cycle Timing

The Interrupt Acknowledge cycle timing for the Z80230 is shown in Figure 22. 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 /CS0, CS1 and R/W are ignored for the duration of the interrupt acknowledge cycle.

The Z80230 samples the state of /INTACK on the rising edge of /AS and AC parameters #7 and #8 specify the setup and hold time requirements. Between the rising edge of /AS and the falling edge of /DS, the internal and external daisy chains settle (AC parameter #29). A system with no external daisy chain should provide the time priority internal to the ESCC. Systems using an external daisy chain should refer to Note 5 referenced in the Z80230 Read/Write and Interrupt Acknowledge Timing for the time required to settle the daisy chain.

If there is an interrupt pending in the ESCC, and IEI is HIGH when /DS falls, the acknowledge cycle was intended for the ESCC. This being the case, the Z80230 sets the Interrupt Under Service (IUS) latch for the highest priority pending interrupt, as well as placing an interrupt vector on AD7-AD0. The placing of a vector on the bus can be disabled by setting WR9, D1=1. The /INT pin also goes inactive in response to the falling edge of /DS. Note that there should be only one /DS per acknowledge cycle. Another important fact is that the IP bits in the Z80230 are updated by /AS, which may delay interrupt requests if the processor does not supply /AS strobes during the time between accesses of the Z80230.

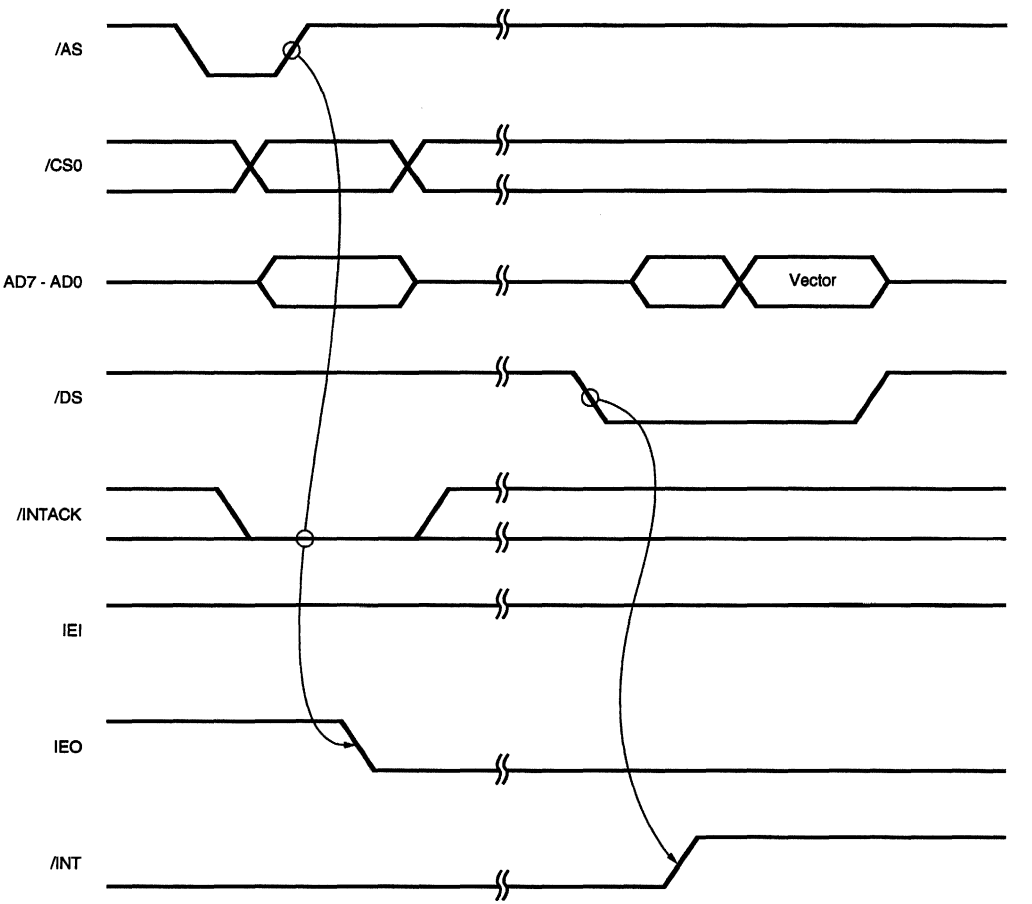


Figure 22. Interrupt Acknowledge Cycle Timing

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 buses.
Z85C30 SCC	CMOS SCC at speeds up to 16 MHz. NMOS compatible.
Z80C30 SCC	CMOS SCC for multiplexed buses.
Z85230 ESCC™	CMOS ESCC for non-multiplexed buses.
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 with ISDN Time Slot Assigner.
Z16C31 IUSC™	MUSC with high performance dual channel DMA.
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 ..... –0.3V 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

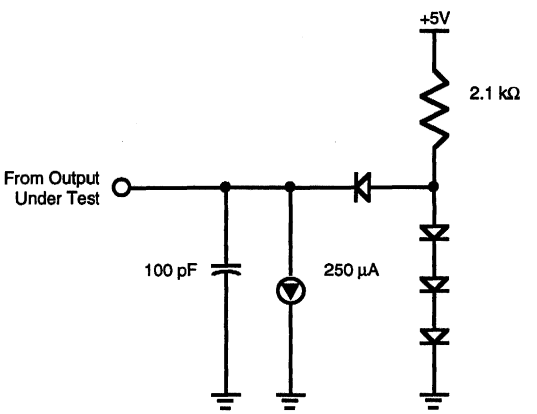


Figure 23. Standard Test Load

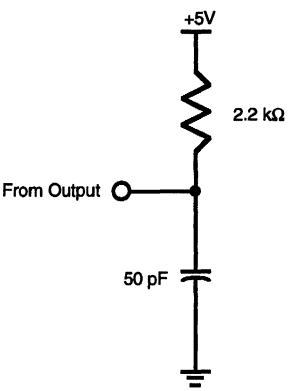


Figure 24. Open-Drain Test Load

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.

MISCELLANEOUS

Gate Count - 11,000

DC CHARACTERISTICS
Z80230

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	$0.4 \le V_{IN} \le +2.4V$
I_{OL}	Output Leakage			±10.0	µA	$0.4 \le V_{OUT} \le +2.4V$
I_{CC1}	V_{CC} Supply Current		4	10 (8.5 MHz)	mA	
			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	
$I_{CC(OSC)}$	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.

3

AC CHARACTERISTICS

Z80230 Read and Write Timing Diagrams

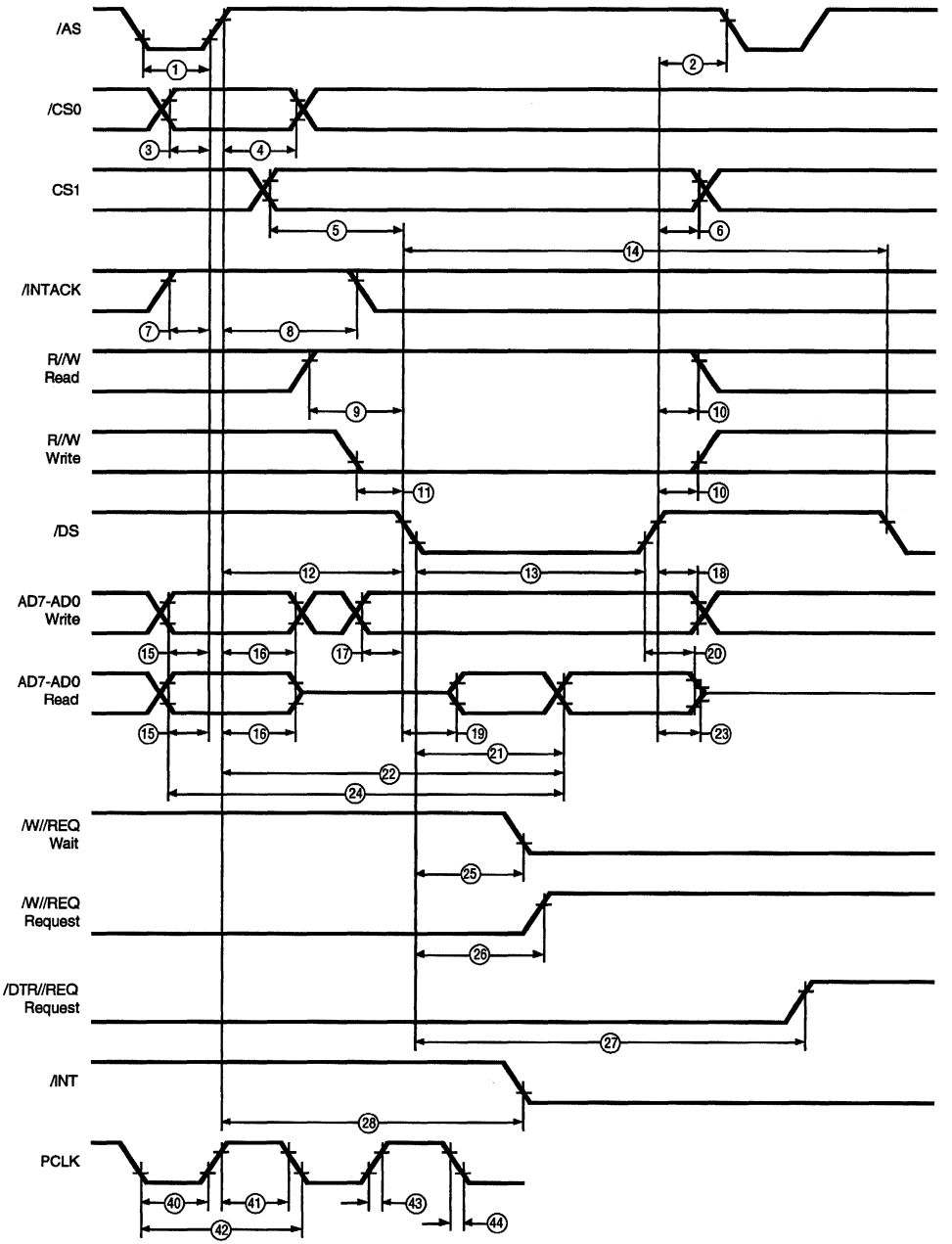


Figure 25. Read and Write Timing Diagram

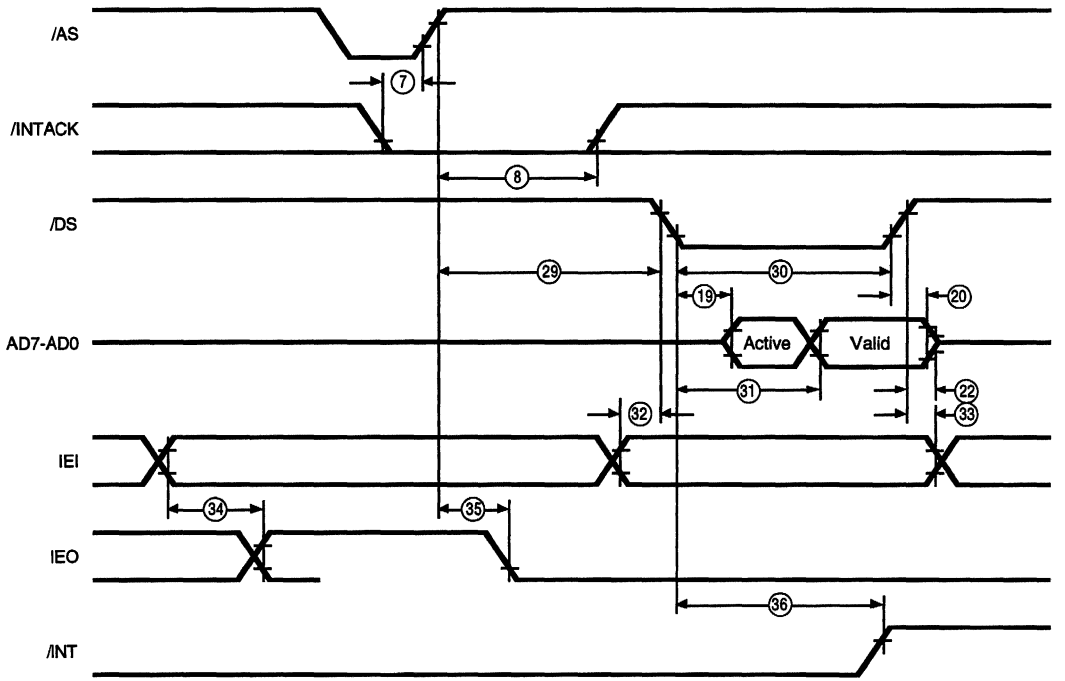


Figure 26. Interrupt Acknowledge Timing Diagram

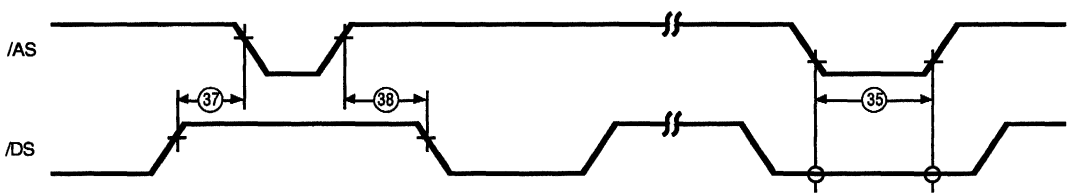


Figure 27. Reset Timing Diagram

AC CHARACTERISTICS

Z80230 Read/Write Timing Table

No	Symbol	Parameter	10 MHz Min	10 MHz Max	16 MHz Min	16 MHz Max	Notes*
1	TwAS	/AS Low Width	30		20		
2	TdDS(AS)	/DS Rise to /AS Fall Delay	10		10		[1]
3	TsCSO(AS)	/CS0 to /AS Rise Setup Time	0		0		[1]
4	ThCSO(AS)	/CS0 to /AS Rise Hold Time	20		15		[1]
5	TsCS1(DS)	CS1 to /DS Fall Setup Time	50		35		[1]
6	ThCS1(DS)	CS1 to /DS Rise Hold Time	20		10		[1]
7	TsIA(AS)	/INTACK to /AS Rise Setup Time	10		10		
8	ThIA(AS)	/INTACK to /AS Rise Hold Time	125		100		
9	TsRWR(DS)	R/W (Read) to /DS Fall Setup Time	50		30		
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	20		15		
13	TwDSI	/DS Low Width	125		80		
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	20		10		[1]
17	TsDW(DS)	Write Data to /DS Fall Setup Time	10		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		120		70	
22	TdAS(DR)	/AS Rise to Read Data Valid Delay		190		110	
23	TdDS(DRz)	/DS Rise to Read Data Float Delay		35		20	[3]
24	TdA(DR)	Address Required Valid to Read Data Valid Delay		210		100	
25	TdDS(W)	/DS Fall to Wait Valid Delay		160		60	[4]
26	TdDSf(REQ)	/DS Fall to /W//REQ Not Valid Delay		160		60	
27	TdDSr(REQ)	/DS Fall to /DTR//REQ Not Valid Delay		4TcPc		4TcPc	
28	TdAS(INT)	/AS Rise to /INT Valid Delay		500		175	[4]
29	TdAS(DSA)	/AS Rise to /DS Fall (Acknowledge) Delay	225		50		[5]
30	TwDSA	/DS (Acknowledge) Low Width	125		75		
31	TdDSA(DR)	/DS Fall (Acknowledge) to Read Data Valid Delay	120		70		
32	TsIEI(DSA)	IEI to /DS Fall (Acknowledge) Setup Time	80		50		
33	ThIEI(DSA)	IEI to /DS Rise (Acknowledge) Hold Time	0		0		
34	TdIEI(IEO)	IEI to IEO Delay		90		45	
35	TdAS(IEO)	/AS Rise to IEO Delay		175		80	[6]
36	TdDSA(INT)	/DS Fall (Acknowledge) to /INT Inactive Delay		450		200	[4]
37	TdDS(ASQ)	/DS Rise to /AS Fall Delay for No Reset	15		10		
38	TdASQ(DS)	/AS Rise to /DS Fall Delay for No Reset	15		10		
39	TwRES	/AS and /DS Coincident Low for Reset	100		75		[7]
40	TwPCI	PCLK Low Width	40	100	26	1000	

AC CHARACTERISTICS

Z80230 Read/Write Timing Table (Continued)

No	Symbol	Parameter	10 MHz Min	10 MHz Max	16 MHz Min	16 MHz Max	Notes*
41	TwPCh	PCLK High Width	40	1000	26	1000	
42	TcPC	PCLK Cycle Time	100	2000	61	2000	
43	TrPC	PCLK Rise Time		10		5	
44	TfPC	PCLK Fall Time		10		5	

Notes:

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

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

[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 Zilog ESCC in the daisy chain. TdAS(DSA) must be greater than the sum of TdAS(IEO) for the highest priority device in the daisy chain. TslEI(DSA) for the Zilog ESCC, and TdlEIf(IEO) for each device separating them in the daisy chain.

[6] Parameter applies only to a Zilog ESCC 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-ESCC. All timing references assume 2.0V for a logic 1 and 0.8V for a logic 0.

* Units in nanoseconds (ns).

AC CHARACTERISTICS

Z80230 General Timing Diagram

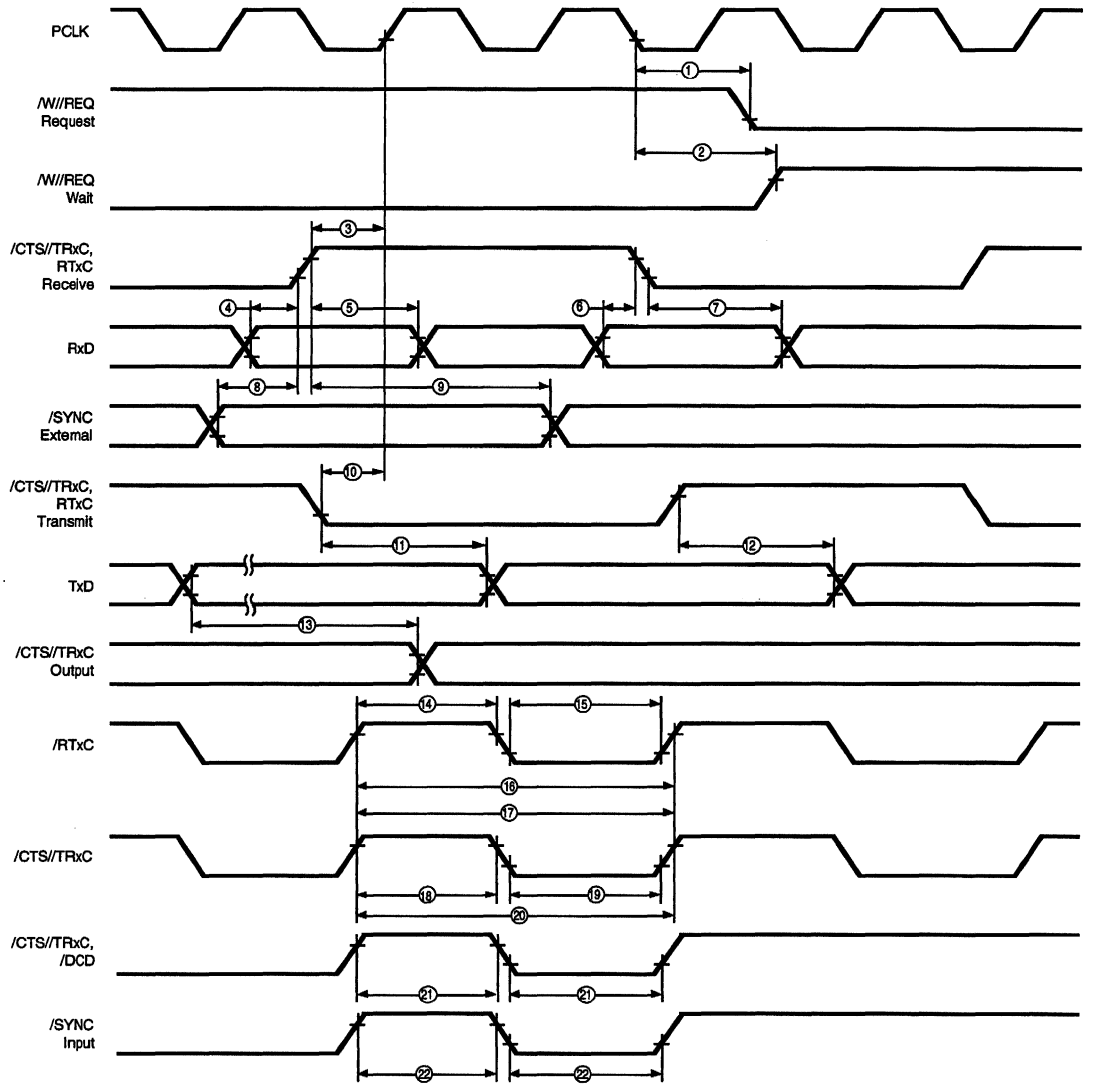


Figure 28. General Timing Diagram

AC CHARACTERISTICS

Z80230 General Timing Table

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

Z80230 System Timing Table

No	Symbol	Parameter	10 MHz Min	10 MHz Max	16 MHz Min	16 MHz Max	Notes*
1	TdRXC(REQ)	/RxC High to W/REQ Valid	13	17	13	17	[2]
2	TdRXC(W)	/RxC High to Wait Inactive	13	19	13	19	[1,2]
3	TdRdXC(SY)	/RxC High to SYNC Valid	9	12	9	12	[2]
4b	TdRXC(INT), Z80230	/RxC High to INT Valid	13	17	13	17	[1,2]
			2	3	+2	+3	[4]
5	TdTXC(REQ)	/TxC Low to W/REQ Valid	8	11	8	11	[3]
6	TdTXC(W)	/TxC Low to Wait Inactive	8	14	8	14	[1,3]
7	TdTXC(DRQ)	/Txc Low to DTR/REQ Valid	7	10	7	10	[3]
8b	TdTXC(INT), Z80230	/TxC Low to /INT Valid	7	9	7	9	[1,3]
			+2	+3	+2	+3	[4]
9a	TdSY(INT)	SYNC to INT Valid	2	6	2	6	[1]
9b	TdSY(INT)	SYNC to INT Valid	2	3	2	3	[1,4]
10b	TdEXT(INT), Z80230		2	3	3	8	[1,4]

Notes:

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

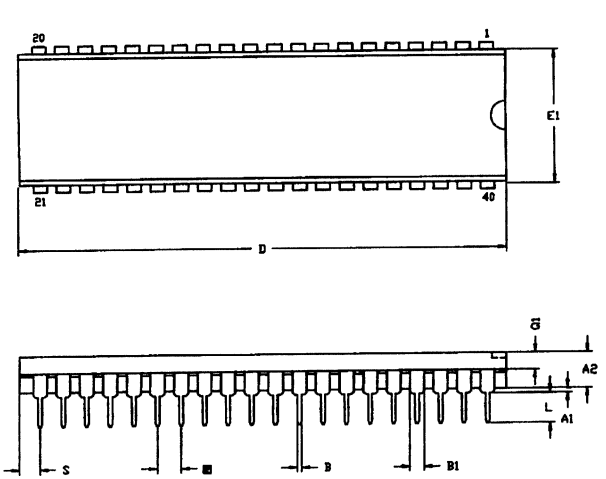
[2] /RxC is /RTxC or /TRxC, whichever is supplying the receive clock.

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

[4] Units equal to /AS.

* Units equal to TcPc.

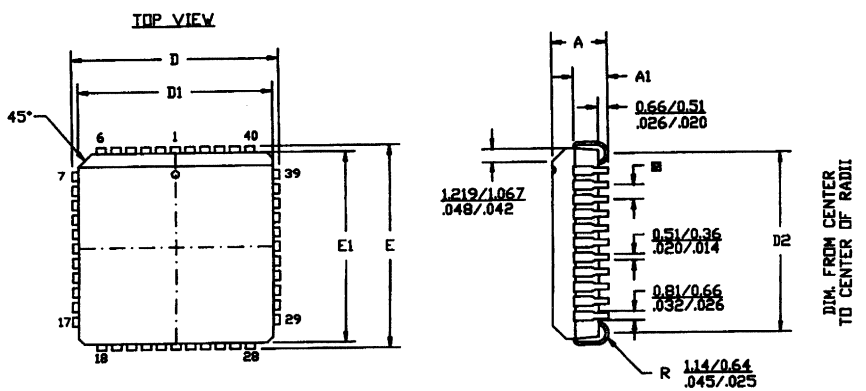
PACKAGE INFORMATION



SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A1	0.51	0.81	.020	.032
A2	3.25	3.43	.128	.135
B	0.38	0.53	.015	.021
B1	1.02	1.52	.040	.060
C	0.23	0.38	.009	.015
D	52.07	52.58	2.050	2.070
E	15.24	15.75	.600	.620
E1	13.59	14.22	.535	.560
e	2.54 TYP		.100 TYP	
eA	15.49	16.51	.610	.650
L	3.18	3.81	.125	.150
Q1	1.52	1.91	.060	.075
S	1.52	2.29	.060	.090

CONTROLLING DIMENSIONS : INCH

40-Pin DIP Package Diagram



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

SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	4.27	4.57	.168	.180
A1	2.67	2.92	.105	.115
D/E	17.40	17.65	.685	.695
D1/E1	16.51	16.66	.650	.656
D2	15.24	16.00	.600	.630
e	1.27 TYP		.050 TYP	

44-Pin PLCC Package Diagram

ORDERING INFORMATION

Z80230

10 MHz	16 MHz
Z8023010PSC	Z8023016PSC
Z8023010VSC	Z8023016VSC

Package
P = Plastic DIP
V = Plastic LCC
C = Ceramic DIP

Temperature
E = –40°C to +100°C
S = 0°C to +70°C

Speeds
10 = 10 MHz
16 = 16.384 MHz

Environmental
C = Plastic Standard
D = Plastic Stressed
E = Hermetic Standard

Example:

Z 80230 16 P S C is a Z80230, 16 MHz, PLCC, 0°C to +70°C, Plastic Standard Flow

- C — Environmental Flow
- S — Temperature
- P — Package
- 16 — Speed
- 80230 — Product Number
- Z — Zilog Prefix

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:
  - 8.0 MHz for Data Rates up to 2.0 Mbit/sec.
  - 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.
- 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.

GENERAL DESCRIPTION

The Zilog Enhanced Serial Communications Controller, Z85230 ESCC, is a pin and software compatible CMOS member of the SCC™ Family (The SCC was 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 (Figure 1).

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

Power connections follow conventional descriptions below:

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

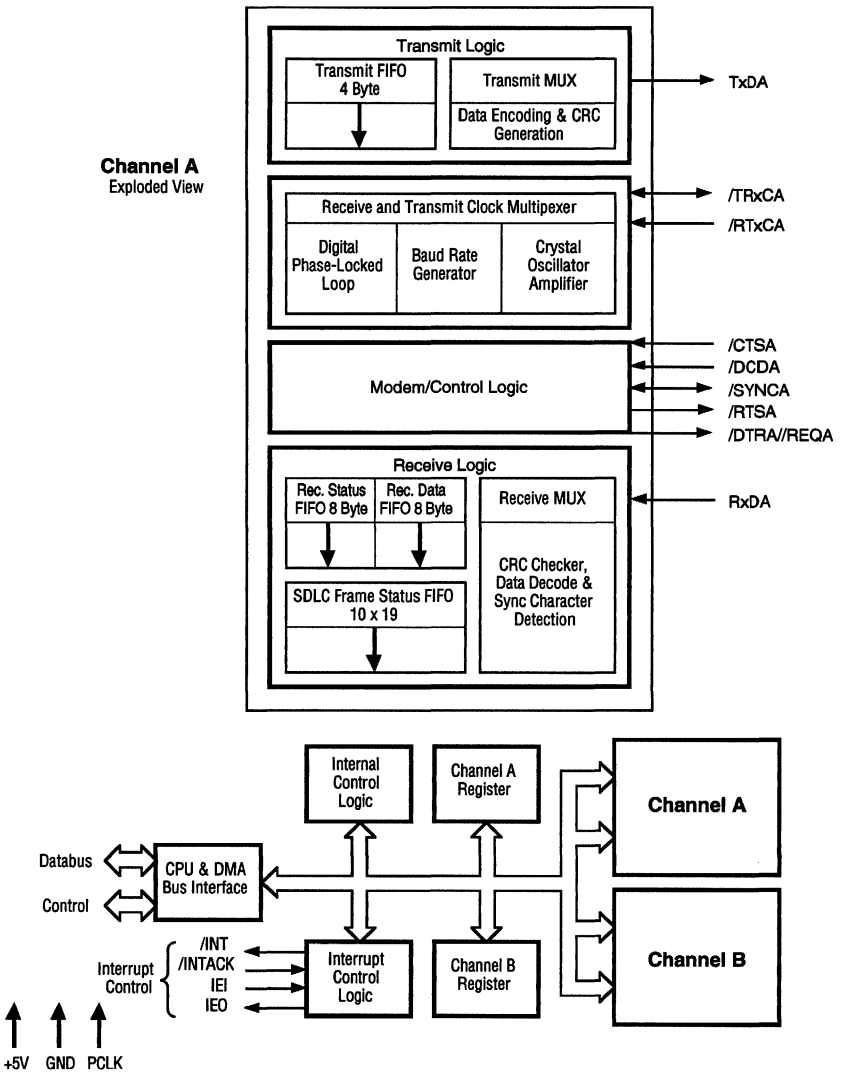


Figure 1. ESCC Block Diagram

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.

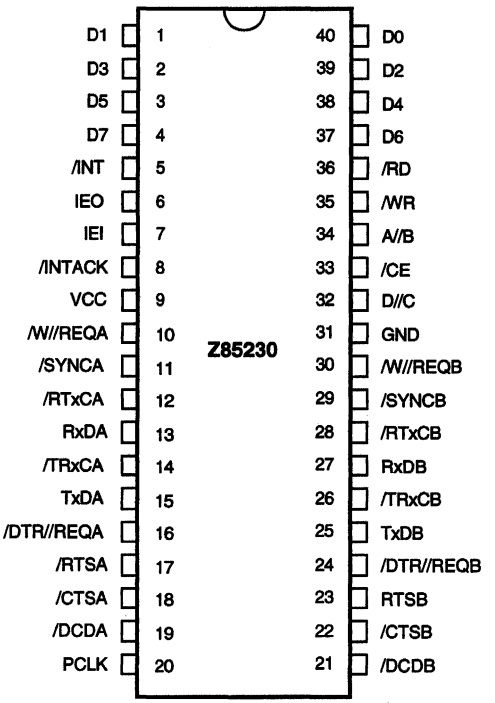


Figure 2. Z85230 DIP Pin Assignments

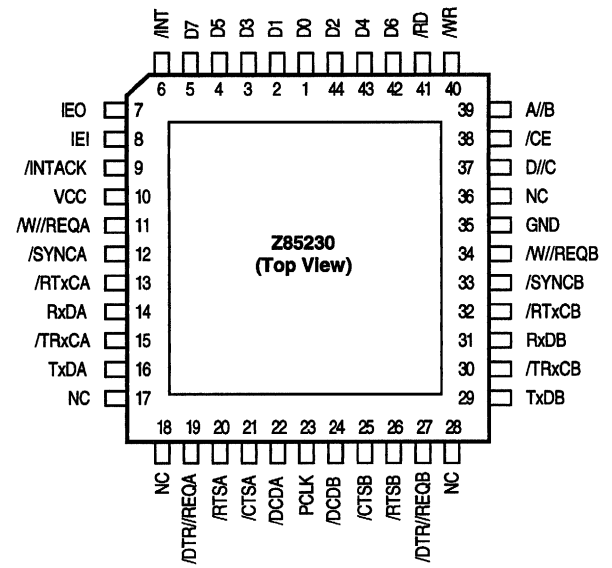


Figure 3. Z85230 PLCC Pin Assignments

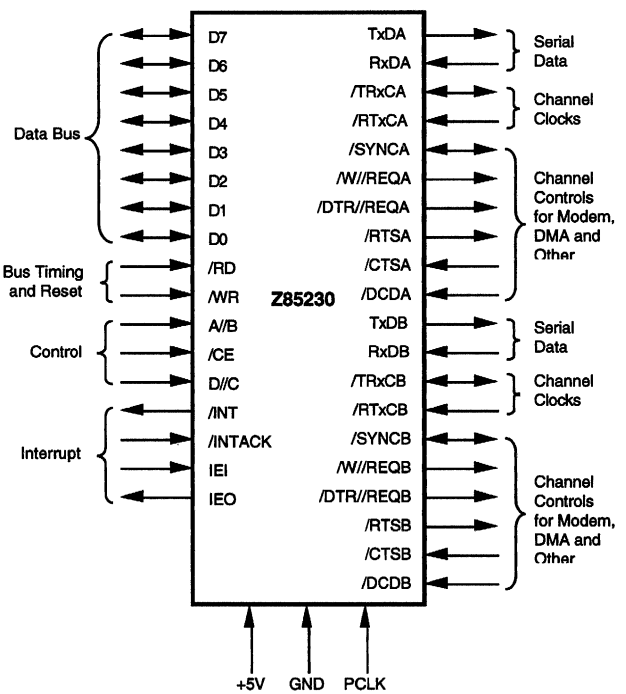


Figure 4. Z85230 Pin Functions

4

PIN DESCRIPTIONS

/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 a 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* (bi-directional, 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 are 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.

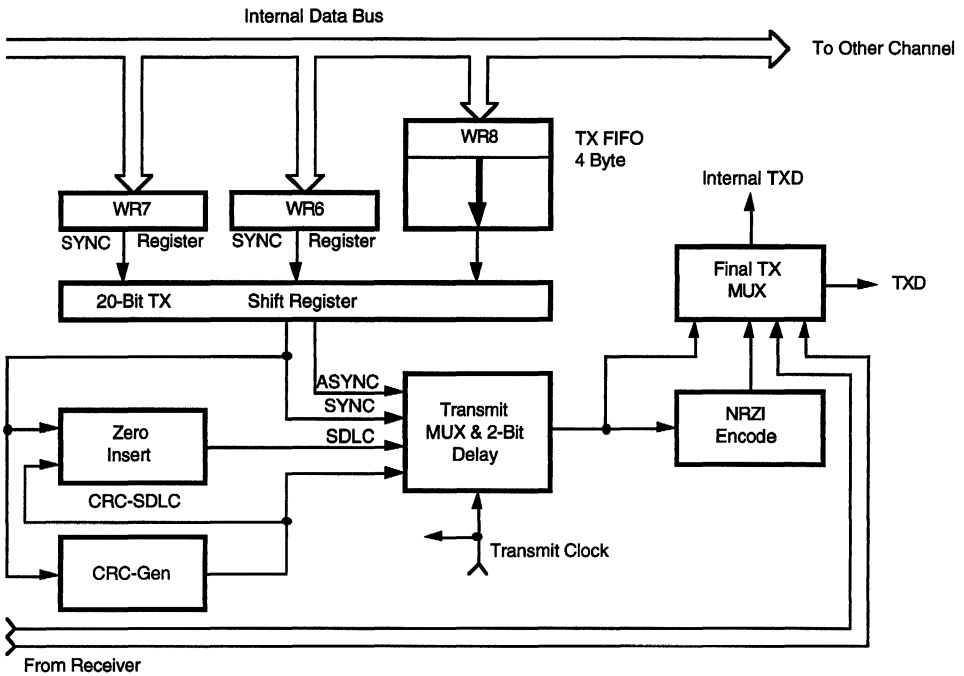


Figure 5. ESCC Transmit Data Path

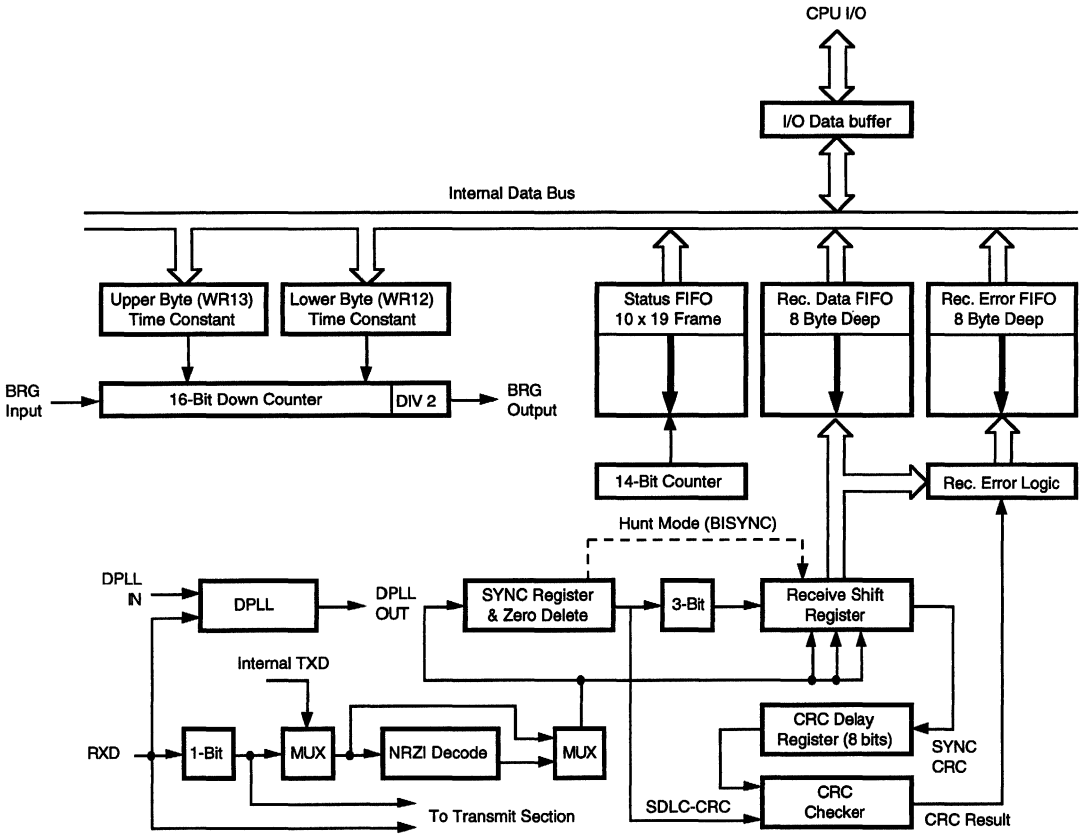


Figure 6. ESCC Receive Data Path

4

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.

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.

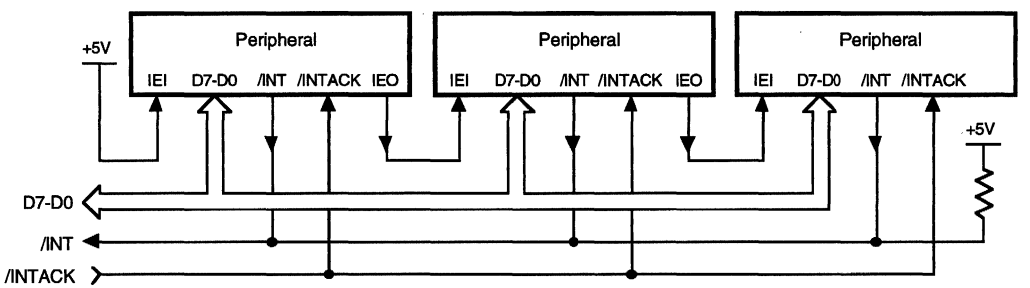


Figure 7. ESCC Interrupt Priority Schedule

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 Z85230 Enhancements 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:

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.

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.

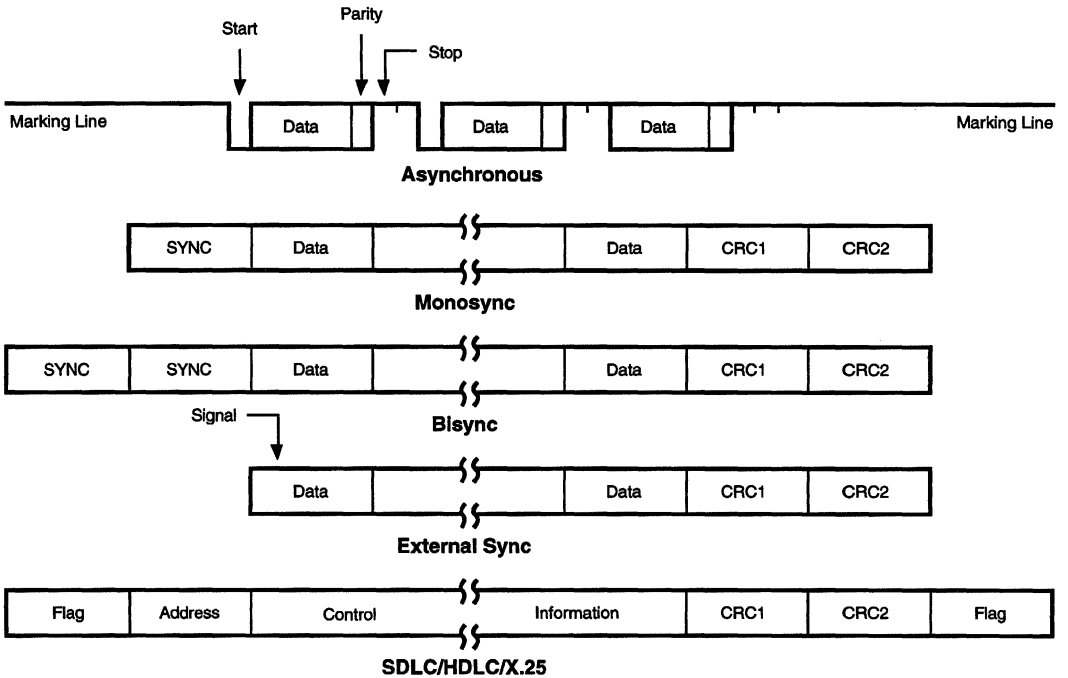


Figure 8. Some ESCC Protocols

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 Z85230 Enhancements 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.

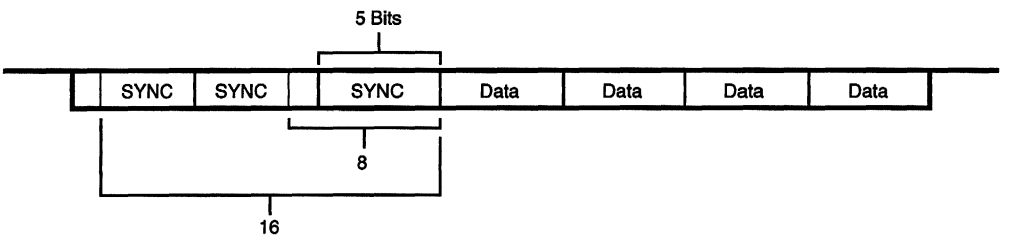


Figure 9. Detecting 5- or 7-Bit Synchronous Characters

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.

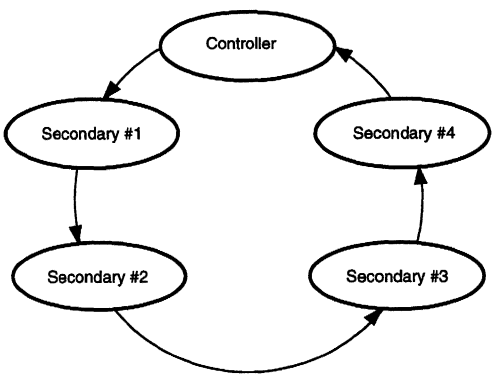


Figure 10. An SDLC Loop

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-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 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

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

selected in Write Register 4, bits D6 and D7. Synchronous operation modes should select 1 and Asynchronous should select 16, 32 or 64.

ESCC DATA COMMUNICATIONS CAPABILITIES (Continued)

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.

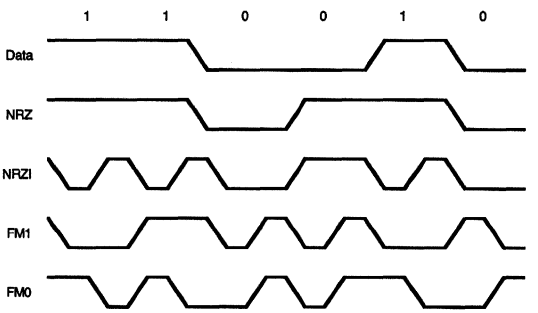


Figure 11. Data Encoding Methods

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.

Z85230 ENHANCEMENTS

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 IS 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.

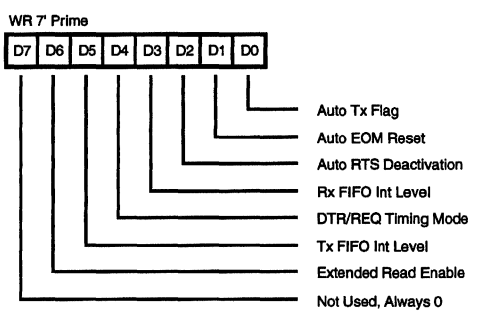


Figure 12. Write Register 7' (7 prime)

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.

Z85230 ENHANCEMENTS (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.

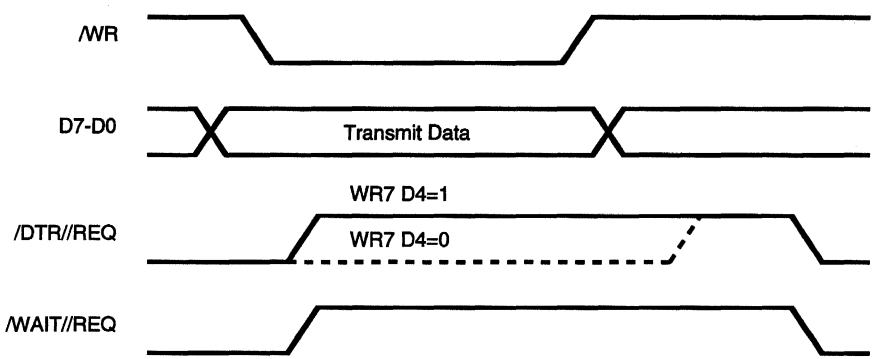


Figure 13. DMA Request on Transmit Deactivation Timing

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 20 ns 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.

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.

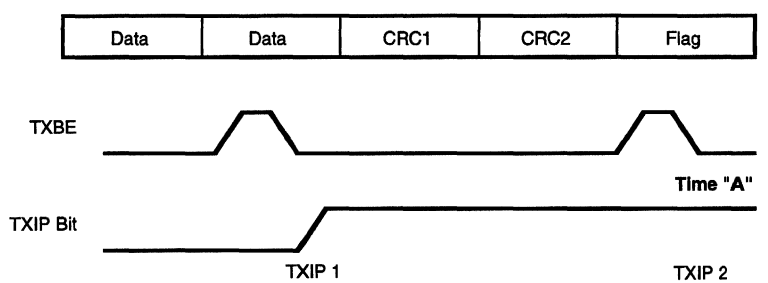


Figure 14. TxIP Latching

NEW FEATURE DESCRIPTION (Continued)

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).

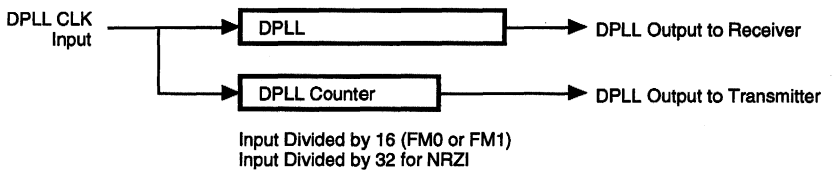


Figure 15. DPLL Outputs

Read Register 0 Status Latched During Read Cycle

The contents of Read Register 0, 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 V_{CC} 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-bit deep by 19-bit wide 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 8-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 and status will be 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 details on 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. 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 (Figure 16). 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. Reading the frame status (CRC, byte count and other status stored in the status FIFO) to determine EOF is not required.

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.

NEW FEATURE DESCRIPTION (Continued)

Frame Status FIFO Circuitry

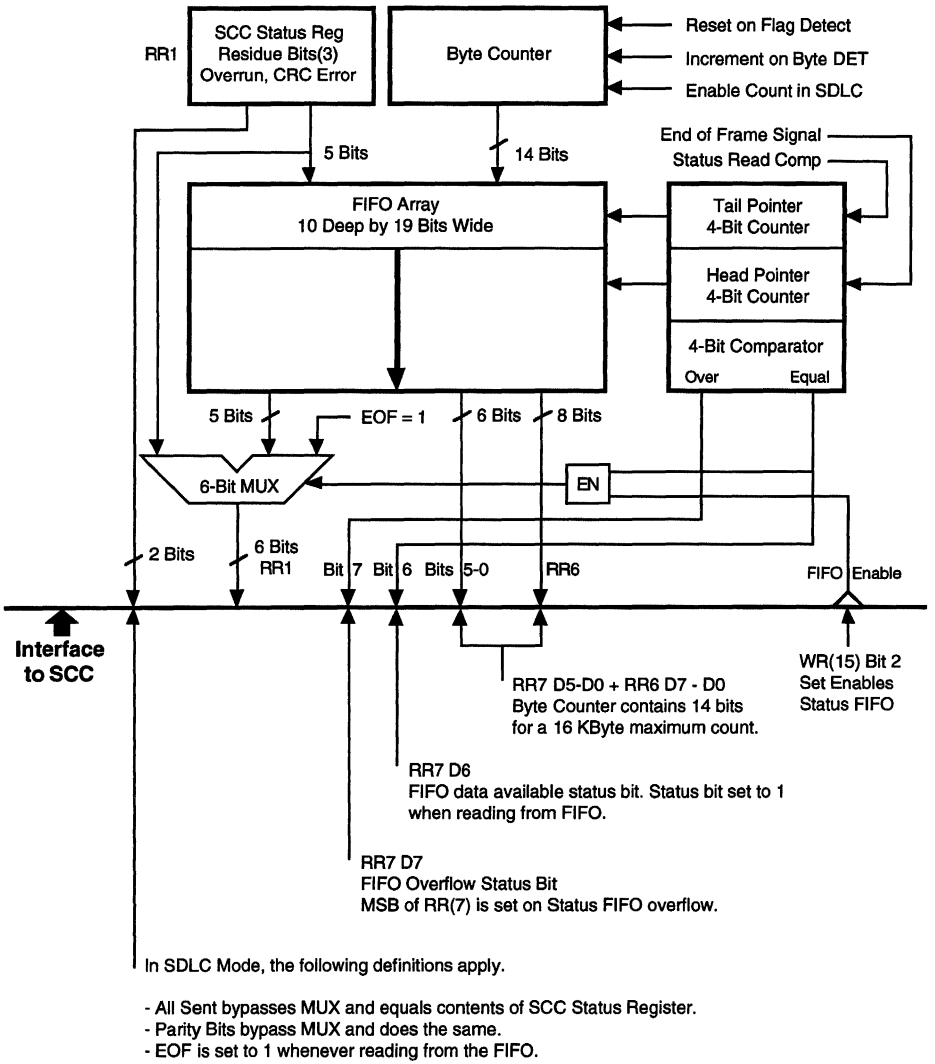


Figure 16. SDLC Frame Status FIFO

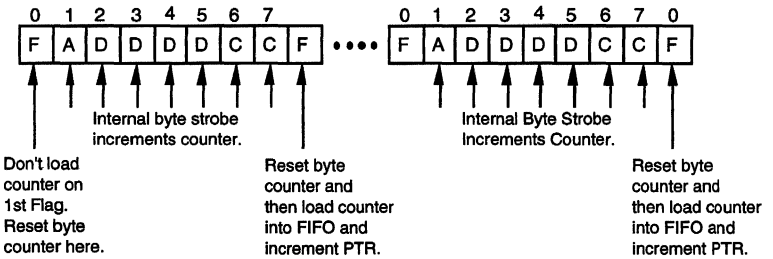


Figure 17. SDLC Byte Counting Detail

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 19 shows the format of each Read register.

CONTROL REGISTERS

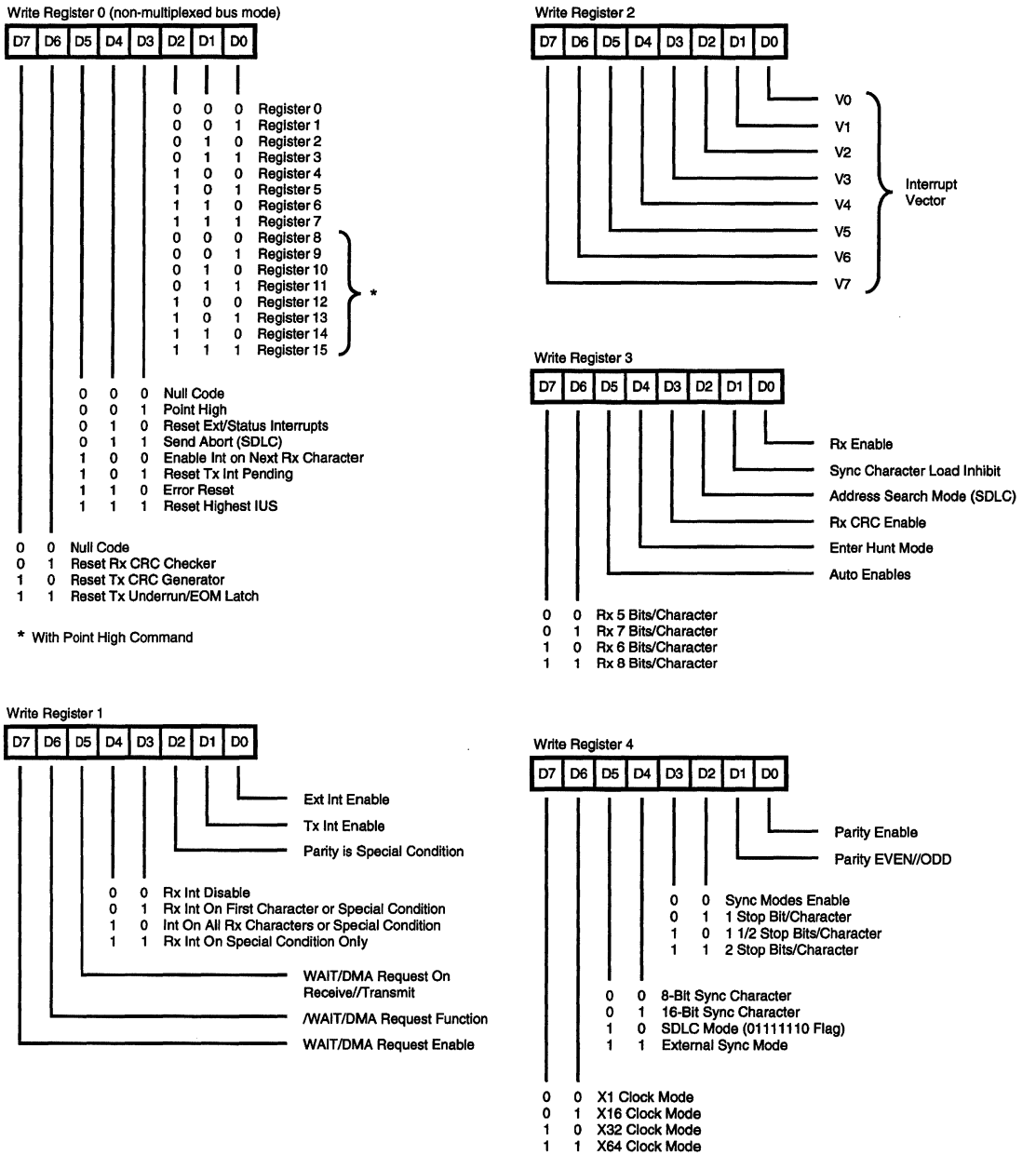


Figure 18. Write Register Bit Functions

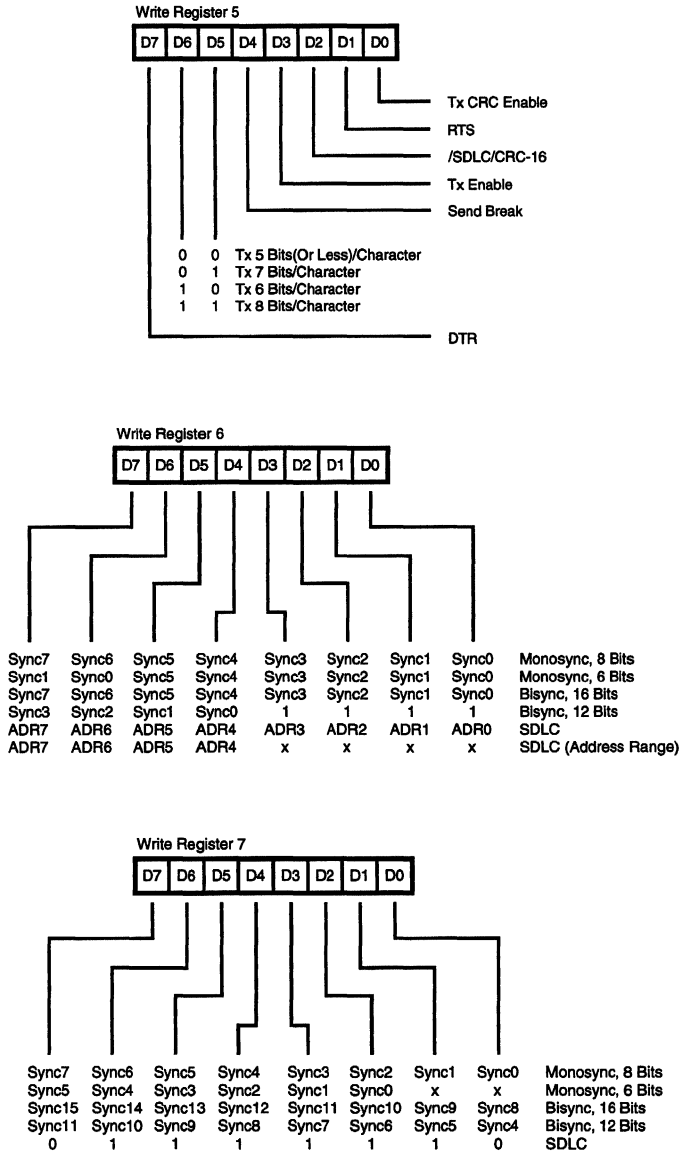


Figure 18. Write Register Bit Functions (Continued)

CONTROL REGISTERS (Continued)

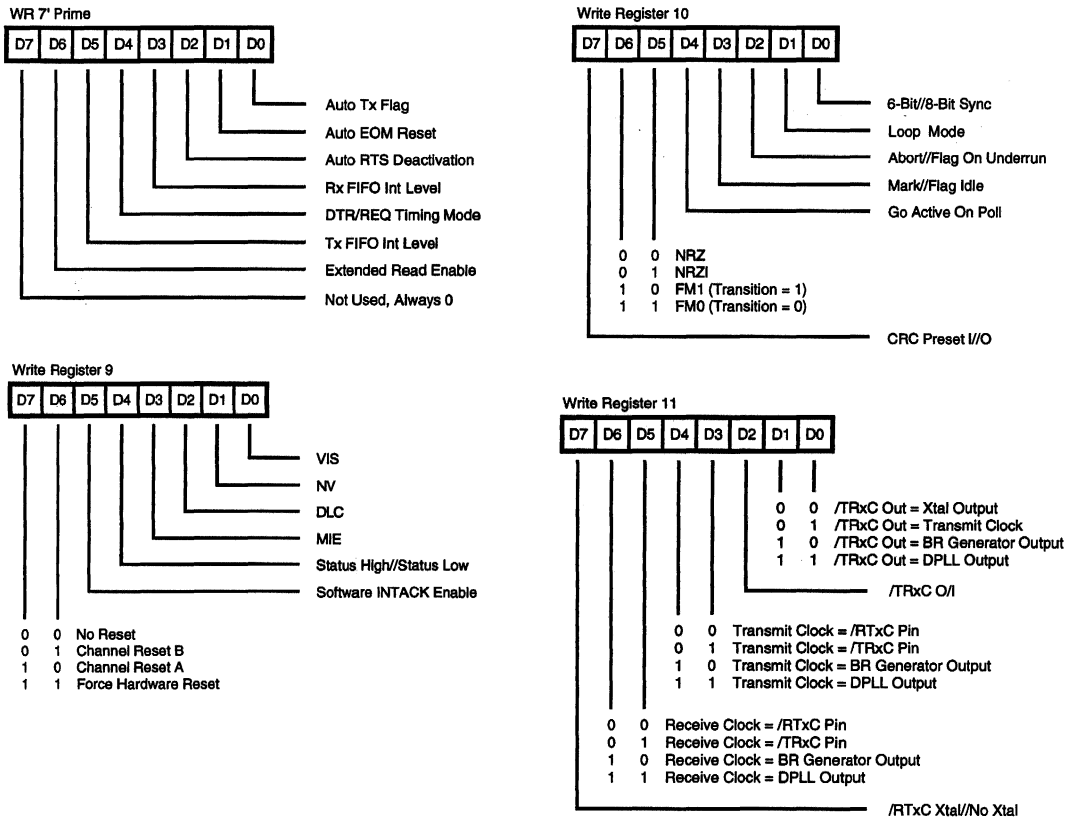


Figure 18. Write Register Bit Functions (Continued)

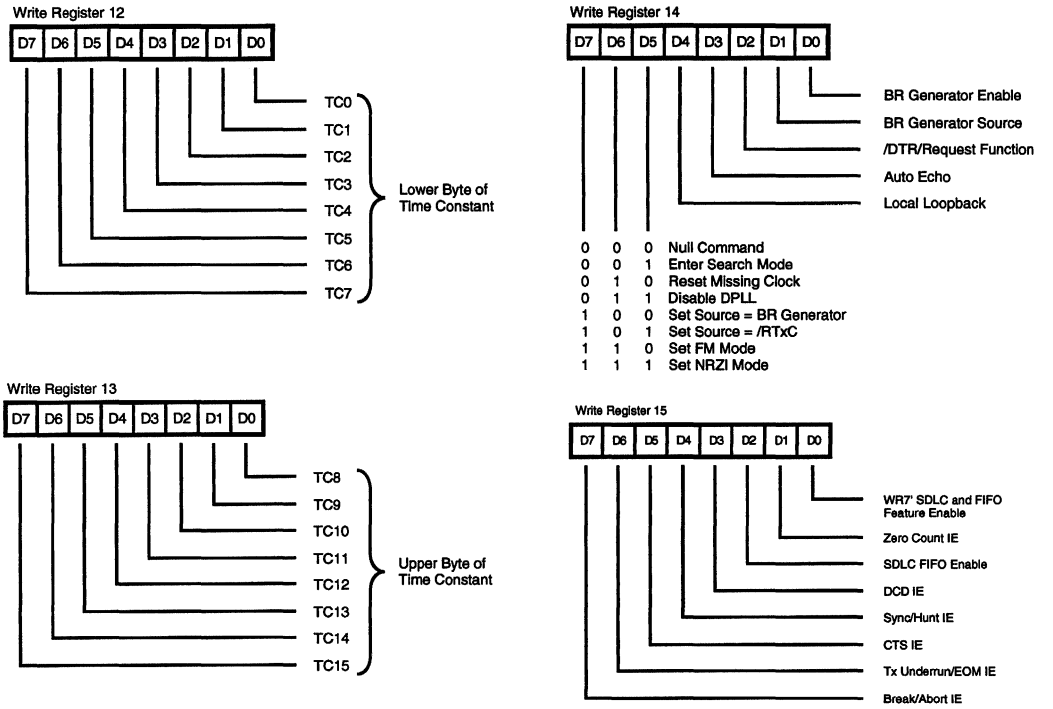


Figure 18. Write Register Bit Functions (Continued)

4

CONTROL REGISTERS (Continued)

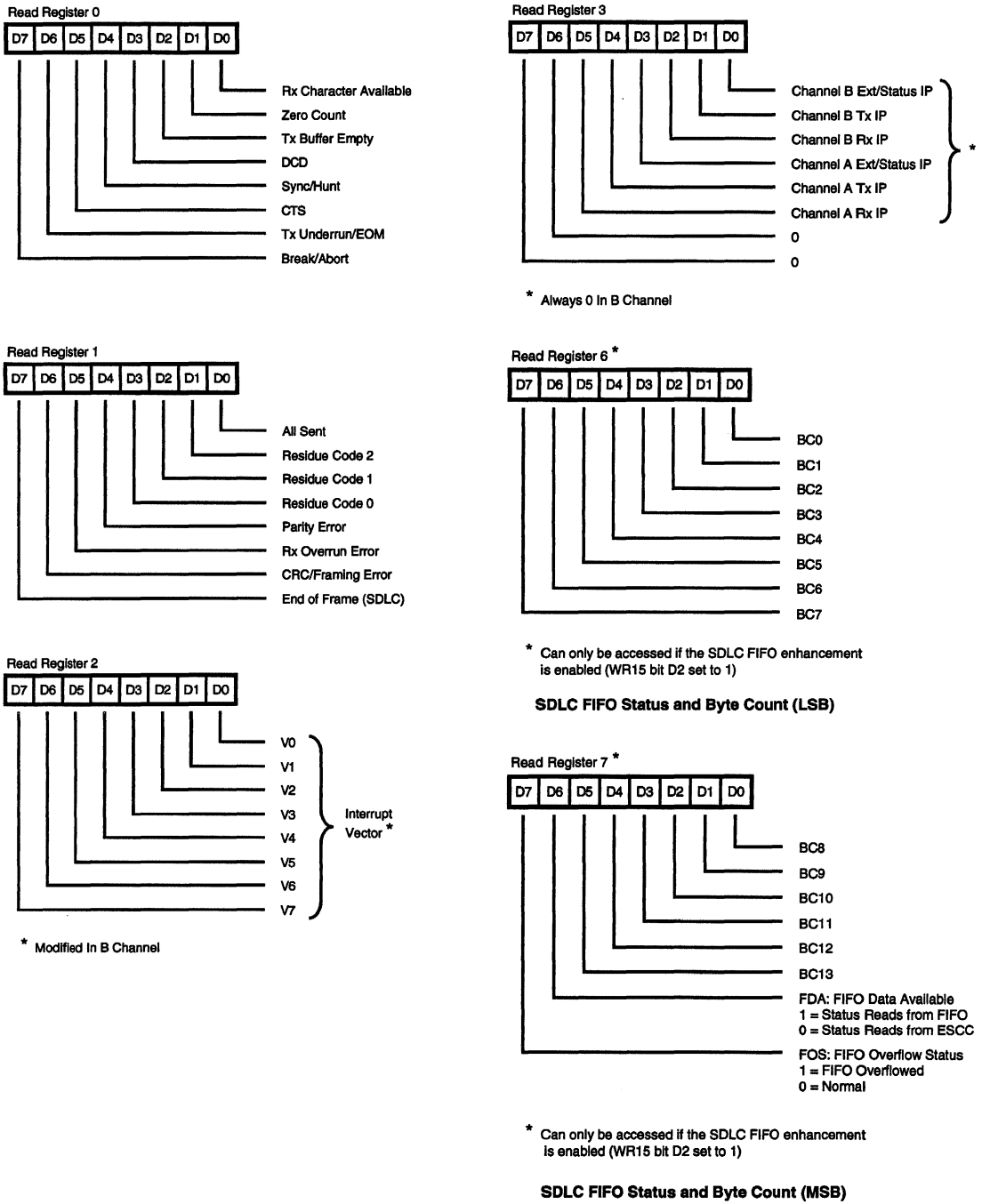


Figure 19. Read Register Bit Functions

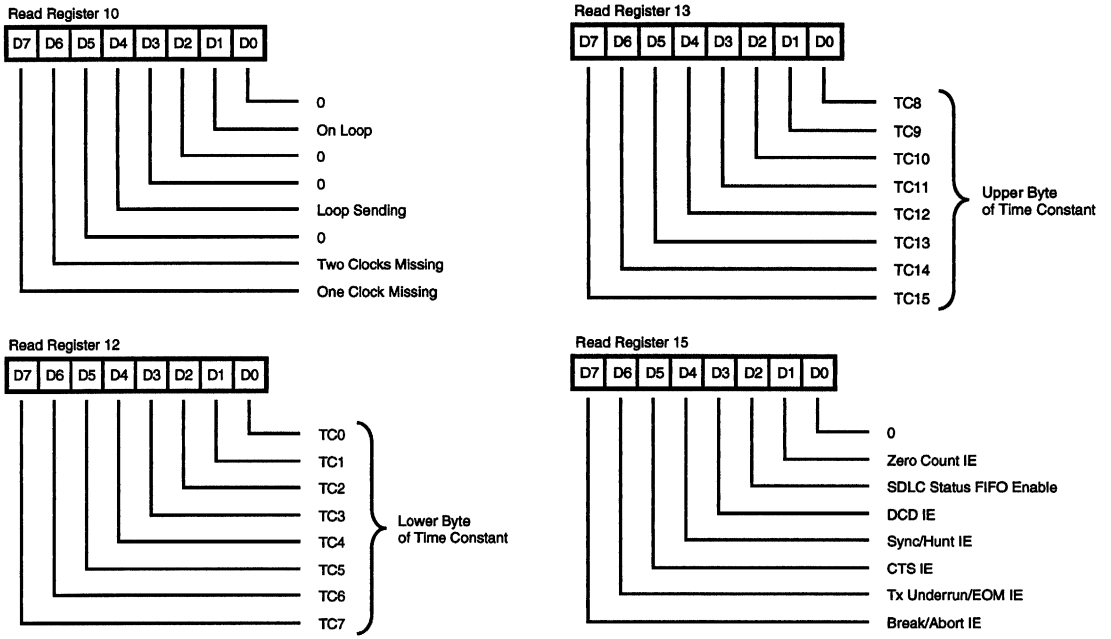


Figure 19. Read Register Bit Functions (Continued)

4

Z85230 TIMING

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.

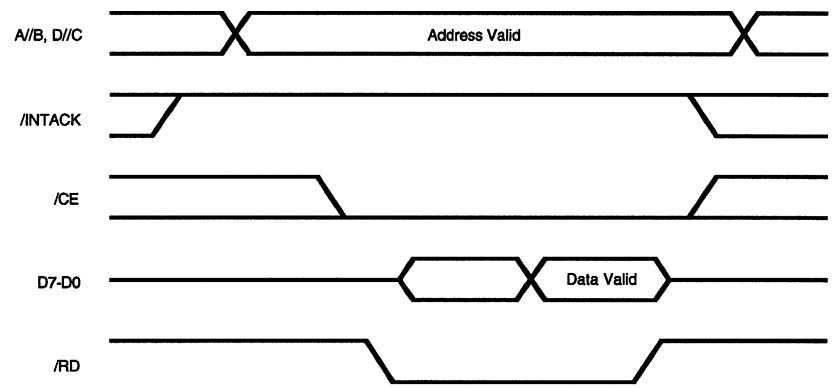


Figure 20. Read Cycle Timing

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.

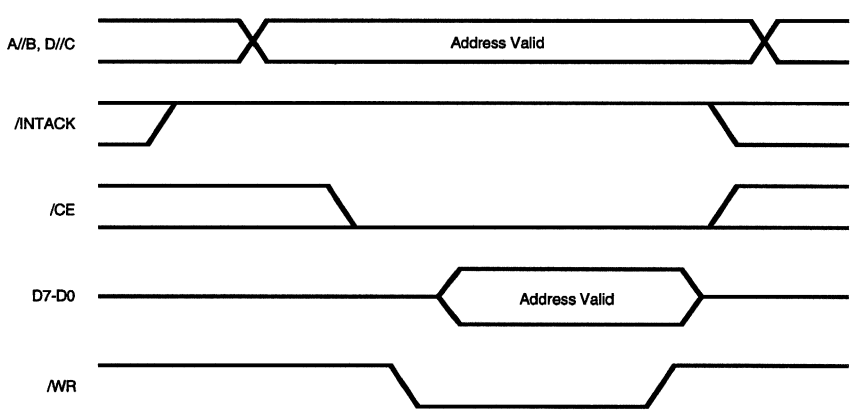


Figure 21. Write Cycle Timing

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.

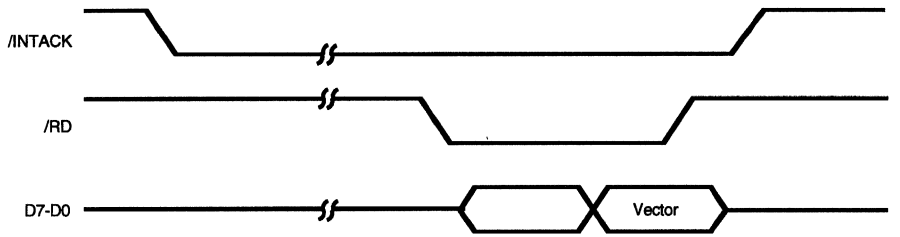


Figure 22. Interrupt Acknowledge Cycle Timing

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 buses.
Z85C30 SCC	CMOS SCC at speeds up to 16 MHz. NMOS compatible.
Z80C30 SCC	CMOS SCC for multiplexed buses.
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.
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 ..... –0.3V to V_{CC} +0.3V
Operating Ambient ..... –0.3V to V_{CC} +0.3V
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

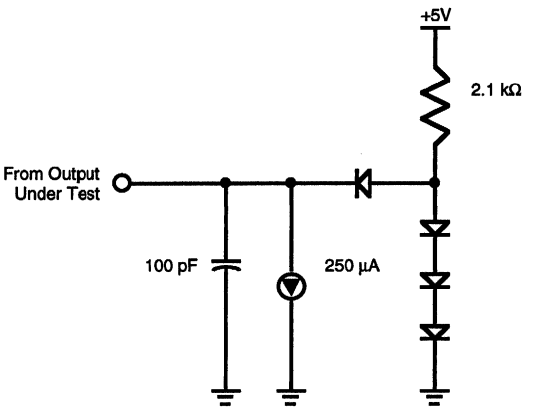


Figure 23. Standard Test Load

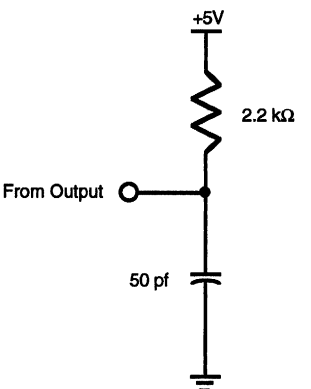


Figure 24. Open-Drain Test Load

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.

MISCELLANEOUS

Gate Count - 11,000

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		4	10 (8.5 MHz)	mA	
			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	
$I_{CC(OSC)}$	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 Timing Diagrams

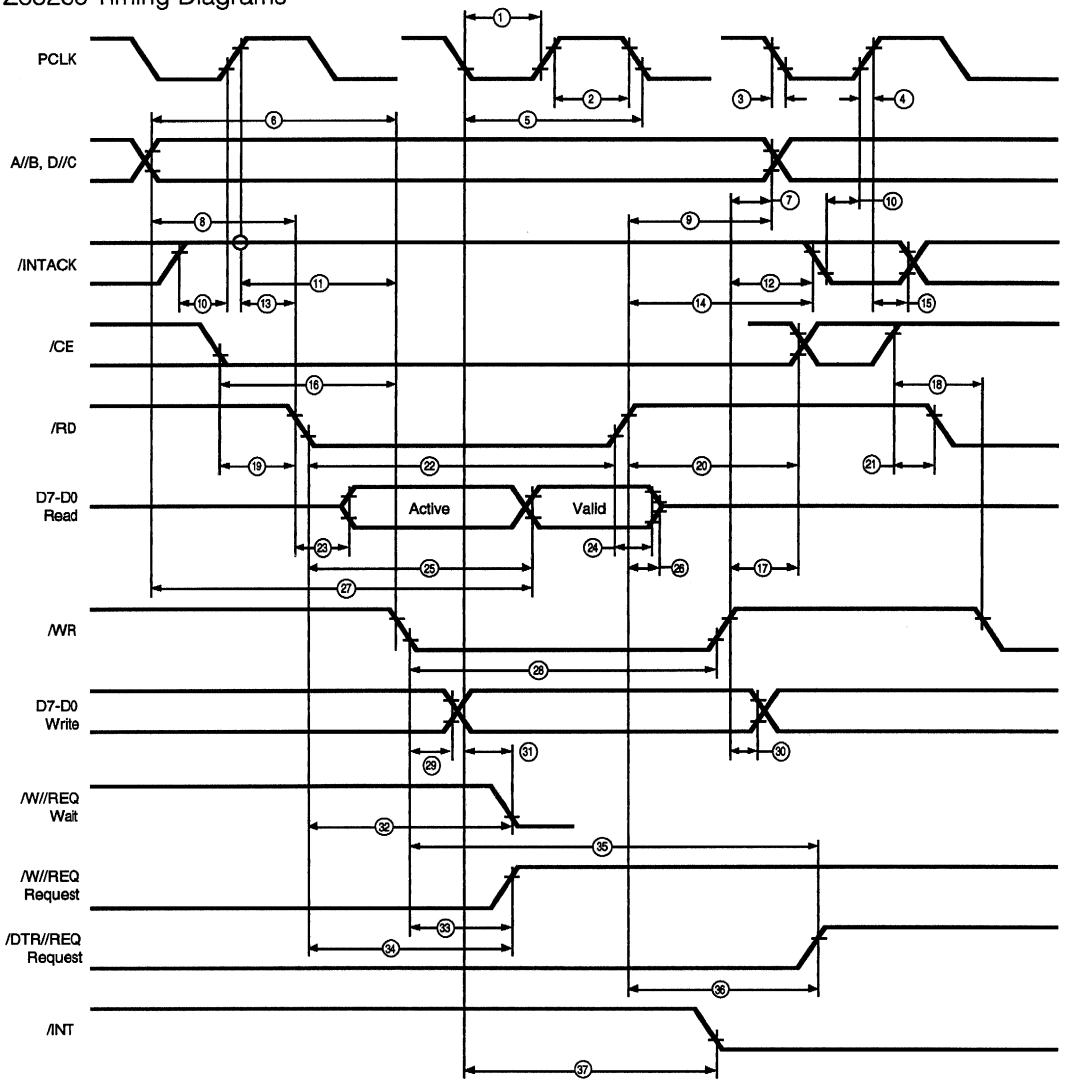


Figure 25. Read and Write Timing Diagram

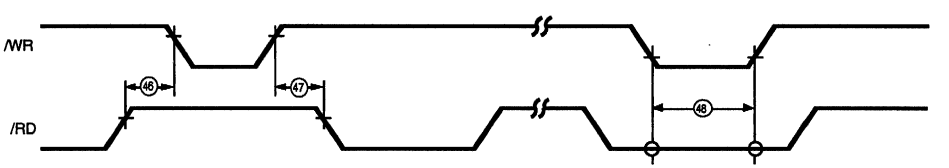


Figure 26. Reset Timing Diagram

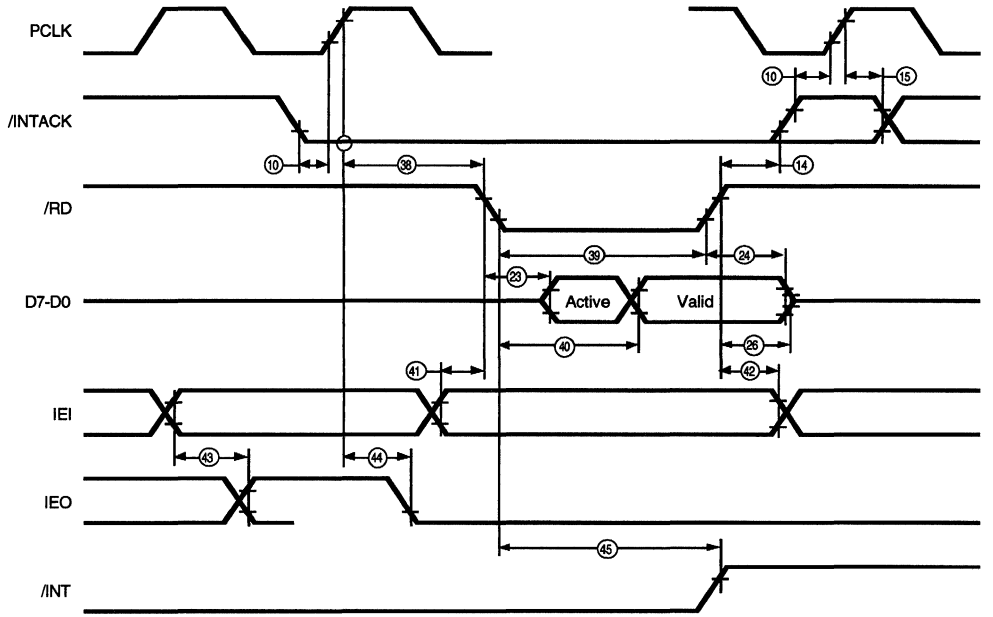


Figure 27. Interrupt Acknowledge Timing Diagram

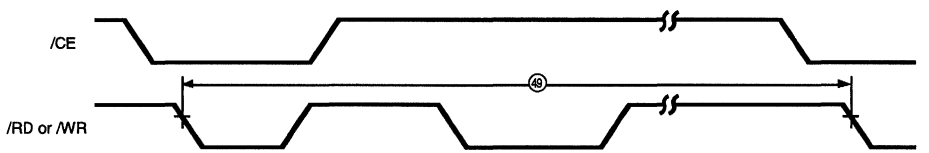


Figure 28. Cycle Timing Diagram

4

AC CHARACTERISTICS

Z85230 Read and 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	20 MHz Min	20 MHz Max	Notes
1	TwPCl	PCLK Low Width	45	1000	40	1000	26	1000	22	1000	
2	TwPCh	PCLK High Width	45	1000	40	1000	26	1000	22	1000	
3	TfPC	PCLK Fall Time		10		10		5		5	
4	TrPC	PCLK Rise Time		10		10		5		5	
5	TcPC	PCLK Cycle Time	118	2000	100	2000	61	2000	50	2000	
6	TsA(WR)	Address to /WR Fall Setup Time	66		50		35		30		
7	ThA(WR)	Address to /WR Rise Hold Time	0		0		0		0		
8	TsA(RD)	Address to /RD Fall Setup Time	66		50		35		30		
9	ThA(RD)	Address to /RD Rise Hold Time	0		0		0		0		
10	TslA(PC)	/INTACK to PCLK Rise Setup Time	20		20		15		15		

AC CHARACTERISTICS

Z85230 Read and 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	20 MHz Min	20 MHz Max	Notes
11	TslAi(WR)	/INTACK To /WR Fall Setup Time	140		130		70		65		[1]
12	ThlA(WR)	/INTACK To /WR Rise Hold Time	0		0		0		0		
13	TslAi(RD)	/INTACK To /RD Fall Setup Time	140		130		70		65		[1]
14	ThlA(RD)	/INTACK To /RD Rise Hold Time	0		0		0		0		
15	ThlA(PC)	/INTACK To PCLK Rise Hold Time	38		30		15		15		
16	TsCEl(WR)	/CE Low To /WR Fall Setup Time	0		0		0		0		
17	ThCE(WR)	/CE To /WR Rise Hold Time	0		0		0		0		
18	TsCEh(WR)	/CE High To /WR Fall Setup Time	58		50		30		25		
19	TsCEl(RD)	/CE Low To /RD Fall Setup Time	0		0		0		0		[1]
20	ThCE(RD)	/CE To /RD Rise Hold Time	0		0		0		0		[1]
21	TsCEh(RD)	/CE High To /RD Fall Setup Time	58		50		30		25		[1]
22	TwRDl	/RD Low Width	145		125		70		65		[1]
23	TdRD(DRA)	/RD Fall To Read Data Active Delay	0		0		0		0		
24	TdRDr(DR)	/RD Rise To Data Not Valid Delay	0		0		0		0		
25	TdRDl(DR)	/RD Fall To Read Data Valid Delay		135		120		70		65	
26	TdRD(DRz)	/RD Rise To Read Data Float Delay		38		35		30		30	
27	TdA(DR)	Addr To Read Data Valid Delay		210		180		100		90	
28	TwWRl	/WR Low Width	145		125		75		65		
29	TdWR(DW)	/WR Fall To Write Data Valid Delay		20		20		20		20	
30	ThDW(WR)	Write Data To /WR Rise Hold Time	0		0		0		0		
31	TdWR(W)	/WR Fall To Wait Valid Delay		168		100		50		50	[4]
32	TdRD(W)	/RD Fall To Wait Valid Delay		168		100		50		50	[4]
33	TdWRf(REQ)	/WR Fall To /W//REQ Not Valid Delay		168		120		70		65	
34	TdRDf(REQ)	/RD Fall To /W//REQ Not Valid Delay		168		120		70		65	[6]
35a	TdWRr(REQ)	/WR Fall To /DTR//REQ Not Valid		4TcPc		4TcPc		4TcPc		4TcPc	
35b	TdWRr(REQ)	/WR Fall To /DTR//REQ Not Valid		168		100		70		65	[6]
36	TdRDr(REQ)	/RD Rise To /DTR//REQ Not Valid Delay		NA		NA		NA		NA	
37	TdPC(INT)	PCLK Fall To /INT Valid Delay		500		320		175		160	
38	TdIAi(RD)	/INTACK To /RD Fall (Ack) Delay	145		90		50		45		[5]
39	TwRDA	/RD (Acknowledge) Width	145		125		75		65		
40	TdRDA(DR)	/RD Fall(Ack) To Read Data Valid Delay		135		120		70		60	
41	TslEl(RDA)	IEI To /RD Fall (Ack) Setup Time	95		95		50		45		
42	ThlEl(RDA)	IEI To /RD Rise (Ack) Hold Time	0		0		0		0		
43	TdIEl(IEO)	IEI To IEO Delay Time		95		90		45		40	
44	TdPC(IEO)	PCLK Rise To IEO Delay		195		175		80		80	
45	TdRDA(INT)	/RD Fall To /INT Inactive Delay		480		320		200		180	[4]
46	TdRD(WRQ)	/RD Rise To /WR Fall Delay For No Reset	15		15		10		10		
47	TdWRQ(RD)	/WR Rise To /RD Fall Delay For No Reset	15		15		10		10		
48	TwRES	/WR And /RD Low For Reset	145		100		75		65		
49	Trc	Valid Access Recovery Time	4TcPc		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. TsIEl(RDA) for the ESCC and TdIEl(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

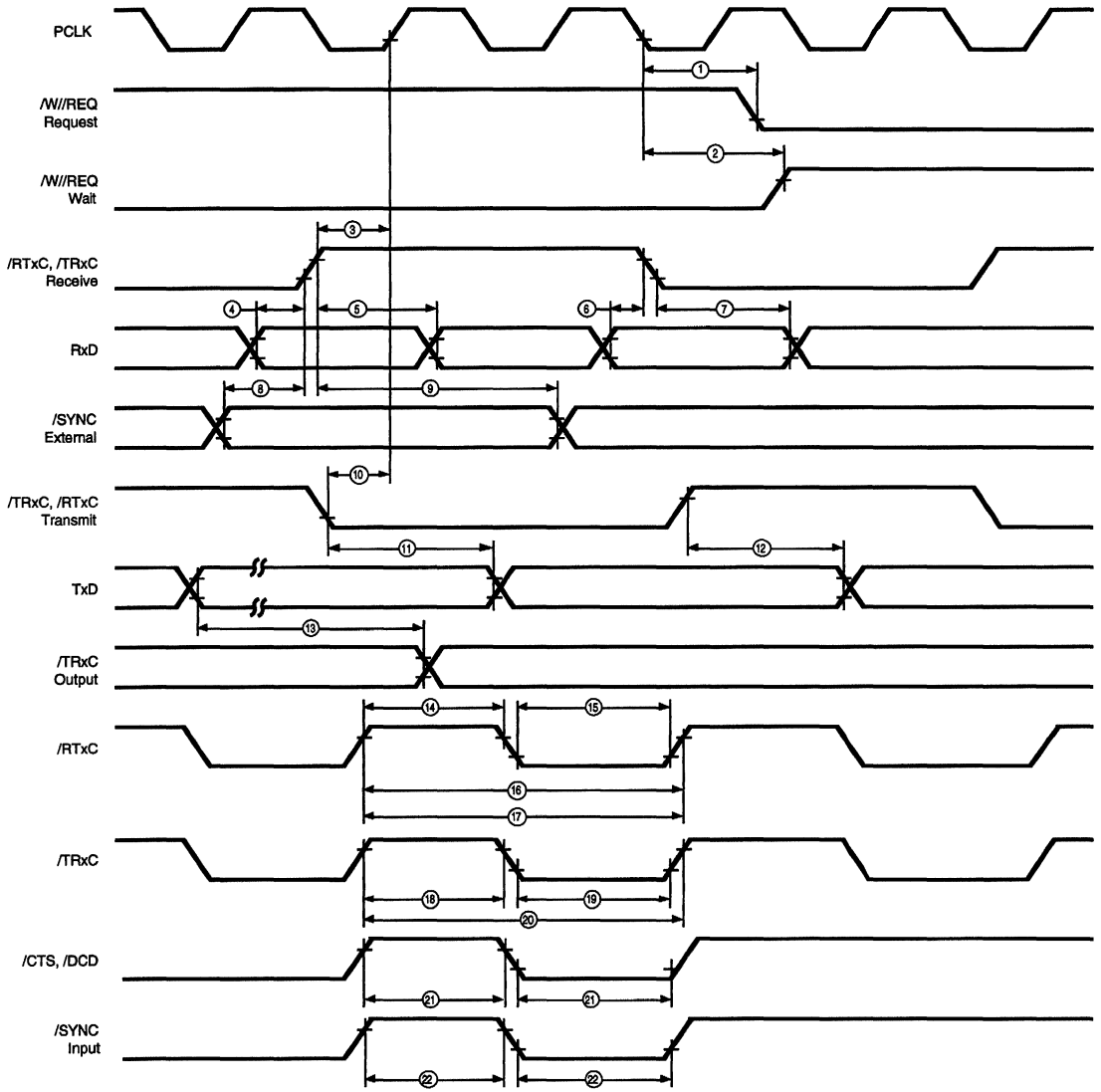


Figure 29. General Timing Diagram

AC CHARACTERISTICS

Z85230 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	20 MHz Min	20 MHz Max	Notes
1	TdPC(REQ)	/PCLK to W/REQ Valid		250		200		80		70	
2	TdPC(W)	/PCLK to Wait Inactive		350		300		180		170	
3	TsRXC(PC)	/RxC to /PCLK Setup Time	N/A		N/A		N/A		N/A		[1,4]
4	TsRXD(RXCr)	RxD to /RxC Setup Time		0		0		0		0	[1]
5	ThRXD(RxCr)	RxD to /RXC Hold Time	150		125		50		45		[1]
6	TsRXD(RXCf)	RxD to /RXC Setup Time	0		0		0		0		[1,5]
7	ThRXD(RXCf)	RXD to /RXC Hold Time	150		125		50		45		[1,5]
8	TsSY(RXC)	/SYNC to /RxC Setup Time	-200		-150		-100		-90		[1]
9	ThSY(RXC)	/SYNC to/RXC Hold Time	5TcPc		5TcPc		5TcPc		5TcPc		[1]
10	TsTXC(PC)	/TxC to /PCLK Setup Time	N/A		N/A		N/A		N/A		[2,4]
11	TdTXCf(TXD)	/TxC to TxD Delay		190		150		80		70	[2]
12	TdTxCr(TXD)	/TxC to TxD Delay		190		150		80		70	[2,5]
13	TdTXD(TRX)	TxD to TRxC Delay		200		140		80		70	
14	TwRTXh	RTxC High Width	130		120		80		70		[6]
15	TwRTXl	TRxC Low Width	130		120		80		70		[6]
16a	TcRTX	RTxC Cycle Time	472		400		244		200		[6,7]
16b	TxRX(DPLL)	DPLL Cycle Time Min	50		50		31		31		[7,8]
17	TcRTXX	Crystal Osc. Period	125	1000	100	1000	61	1000	61	1000	[3]
18	TwTRXh	TRxC High Width	130		120		80		70		[6]
19	TwTRXl	TRxC Low Width	130		120		80		70		[6]
20	TcTRX	TRxC Cycle Time	472		400		244		200		[6,7]
21	TwEXT	DCD or CTS Pulse Width	200		120		70		60		
22	TwSY	SYNC Pulse Width	200		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

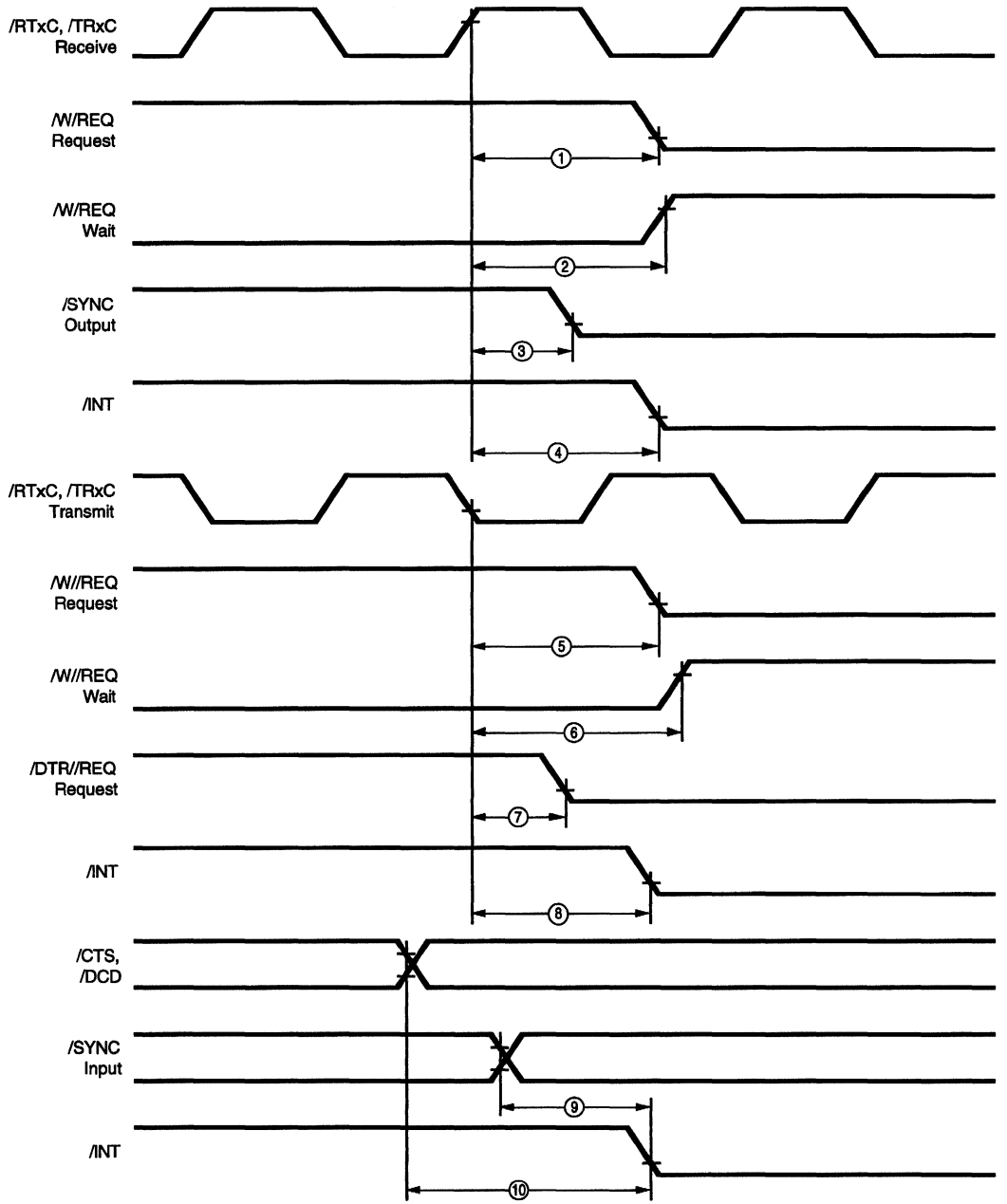


Figure 30. Z85230 System Timing

AC CHARACTERISTICS

Z85230 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	20 MHz Min	20 MHz Max	Notes [4]
1	TdRXC(REQ)	/RXC to /W//REQ Valid	13	17	13	17	13	17	13	18	[2]
2	TdRXC(W)	/RxC to /Wait Inactive	13	17	13	17	13	17	13	18	[1,2]
3	TdRXC(SY)	/RxC to /SYNC Valid	4	7	4	7	4	7	4	8	[2]
4	TdRXC(INT)	/RxC to /INT Valid	15	21	15	21	15	21	15	22	[1,2]
5	TdTXC(REQ)	/TxC to /W//REQ Valid	8	11	8	11	8	11	8	12	[3]
6	TdTXC(W)	/TxC to /Wait Inactive	8	14	8	14	8	14	8	15	[1,3]
7	TdTXC(DRQ)	/TxC to /DTR//REQ Valid	7	10	7	10	7	10	7	11	[3]
8	TdTXC(INT)	/TxC to /INT Valid	9	13	9	13	9	13	9	14	[1,3]
9	TdSY(INT)	/SYNC to /INT Valid	2	6	2	6	2	6	2	7	[1]
10	TdEXT(INT)	/DCD or /CTS to /INT Valid	3	8	3	8	3	8	3	9	[1]

Notes:

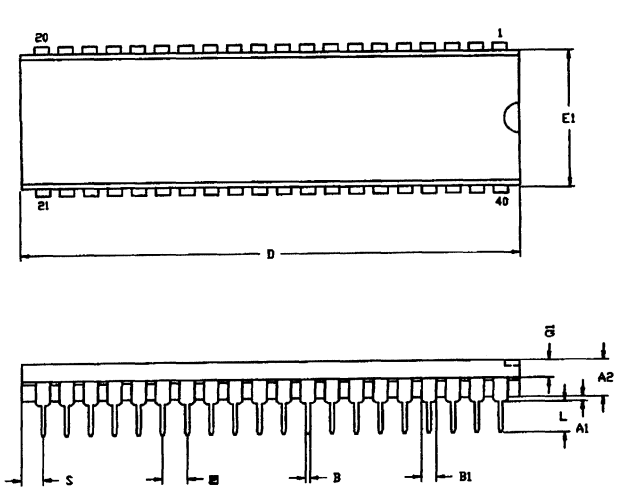
[1] Open-drain output, measured with open-drain test load.

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

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

[4] Units equal to TcPc.

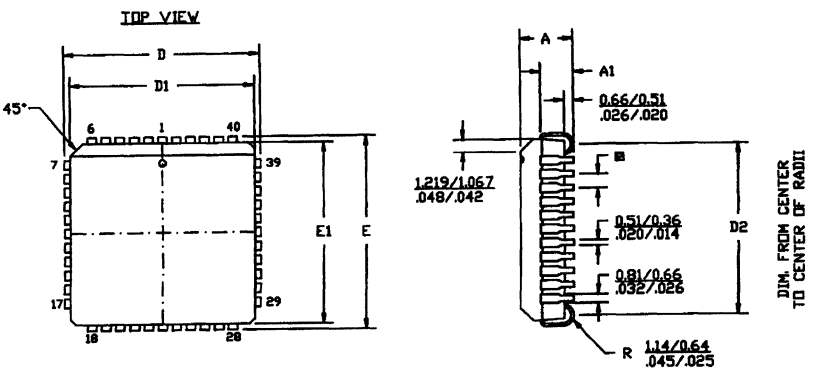
PACKAGE INFORMATION



SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A1	0.51	0.81	.020	.032
A2	3.25	3.43	.128	.135
B	0.38	0.53	.015	.021
B1	1.02	1.52	.040	.060
C	0.23	0.38	.009	.015
D	52.07	52.58	2.050	2.070
E	15.24	15.75	.600	.620
E1	13.59	14.22	.535	.560
e	2.54 TYP		.100 TYP	
eA	15.49	16.51	.610	.650
L	3.18	3.81	.125	.150
Q1	1.52	1.91	.060	.075
S	1.52	2.29	.060	.090

CONTROLLING DIMENSIONS : INCH

40-Pin DIP Package Diagram



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

SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	4.27	4.57	.168	.180
A1	2.67	2.92	.105	.115
D/E	17.40	17.65	.685	.695
D1/E1	16.51	16.66	.650	.656
D2	15.24	16.00	.600	.630
e	1.27 TYP		.050 TYP	

44-Pin PLCC Package Diagram

ORDERING INFORMATION

Z85230

8 MHz	10 MHz	16 MHz	20 MHz
Z8523008PSC/PEC	Z8523010PSC	Z8523016PSC	Z8523020PSC
Z8523008VSC/VEC	Z8523010VSC	Z8523016VSC	Z8523020VSC
	Z8523010PEC	Z8523016PEC	
	Z8523010VEC	Z8523016VEC	

Package
P = Plastic DIP
V = Plastic LCC

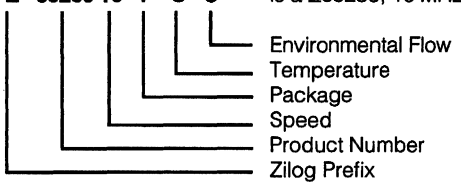
Temperature
E = -40°C to +100°C
S = 0°C to +70°C

Speeds
10 = 10.0 MHz
16 = 16.384 MHz
20 = 20.0 MHz

Environmental
C = Plastic Standard

Example:

Z 85230 16 P S C is a Z85230, 16 MHz, DIP, 0°C to +70°C, Plastic Standard Flow.



Z85233

EMSCC™ ENHANCED MONO SERIAL COMMUNICATION CONTROLLER

FEATURES

- Deeper Data FIFOs
  - 4-Byte Transmit FIFO
  - 8-Byte Receive FIFO
- Programmable FIFO Interrupt Levels Provide Flexible Interrupt Response
- 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.
- 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
- A Full-Duplex Channel with a Crystal Oscillator, Baud Rate Generator, and Digital Phase-Locked Loop.
- Multi-Protocol Operation Under Program Control
- Asynchronous Mode/Synchronous Mode

GENERAL DESCRIPTION

The Zilog Enhanced Mono Serial Communication Controller, Z85233 EMSCC, is a software compatible CMOS member of the SCC family introduced by Zilog in 1981. The EMSCC is a full-duplex datacommunications controller capable of supporting a wide range of popular protocols. The Z85233 EMSCC is a single channel version (Channel A) of Zilog's Z85230 ESCC. Based on Zilog's unique Superintegration™ Technology, the EMSCC is compatible with designs using Zilog's SCC and ESCC 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 transmitter and receiver.

GENERAL DESCRIPTION (Continued)

The EMSCC also has many features that improve packet handling in SDLC mode. The EMSCC 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 data bus 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 EMSCC is improved over the SCC with faster response of the /INT and /DTR//REQ pins.

The many enhancements added to the EMSCC permits a system design that increases overall system performance with better data handling and less interface logic (Figure 1).

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

Power connections follow conventional descriptions below:

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

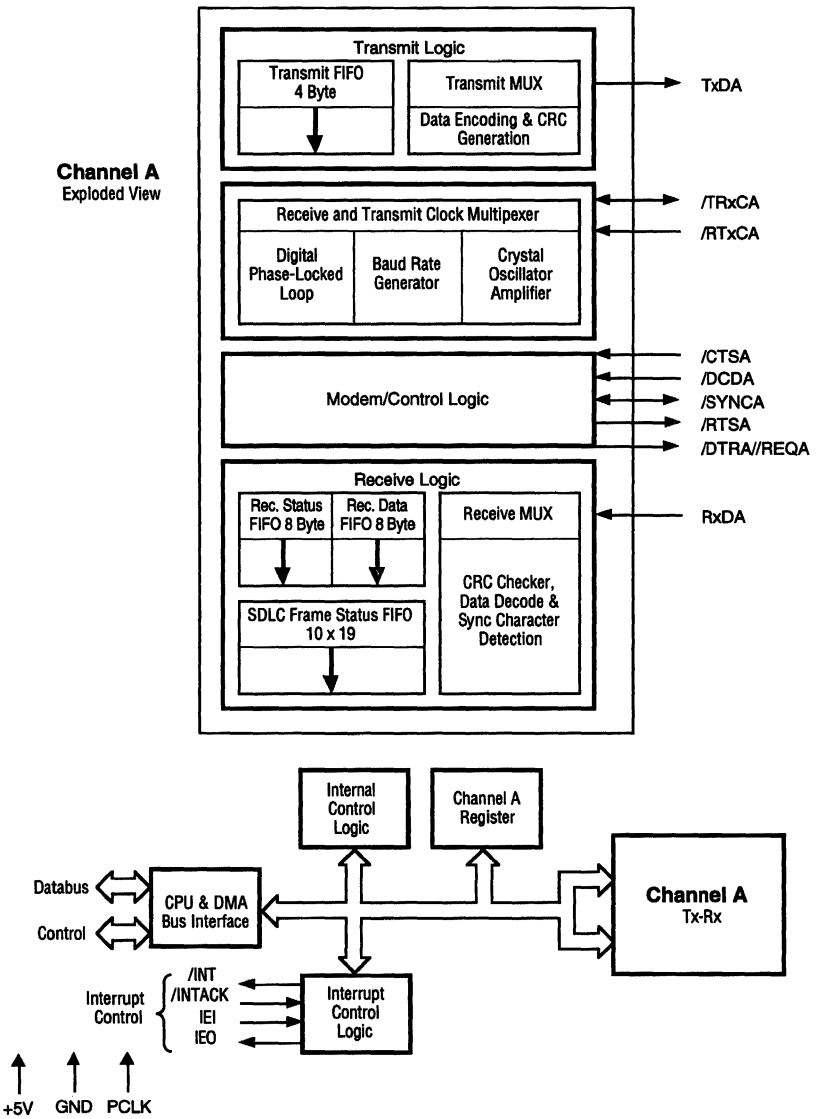


Figure 1. EMSCC Block Diagram

5

PIN DESCRIPTIONS

The following section describes the Z85233 pin functions. Figure 2 details the pin functions of the EMSCC and Figures 3 and 4 are the pin assignments for the 44-pin PQFP and 44-pin PLCC packages, respectively. The pin electrical characteristics are the same as the Z85230 ESCC. Any unused input pins should be pulled up to the +5V supply.

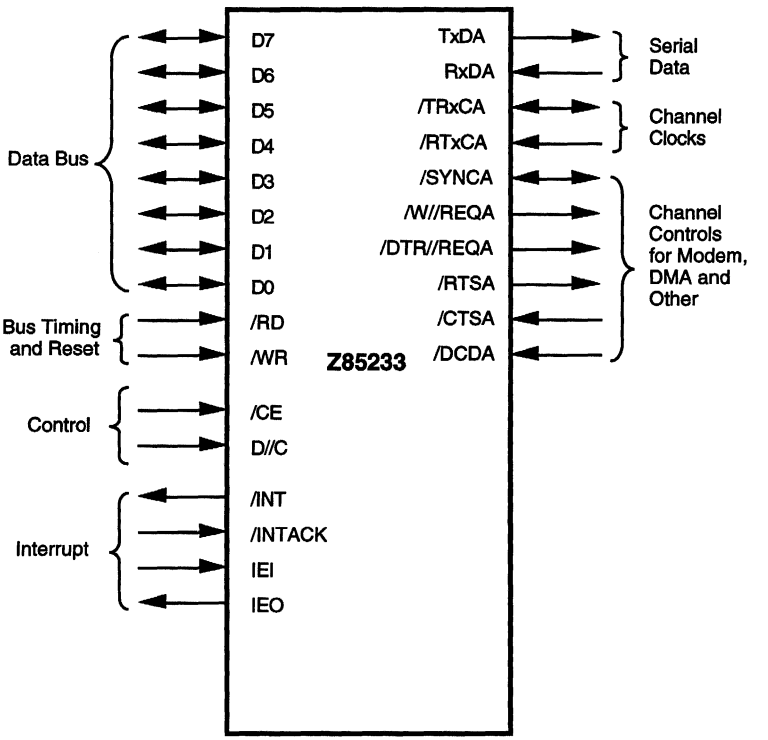


Figure 2. Z85233 Pin Functions

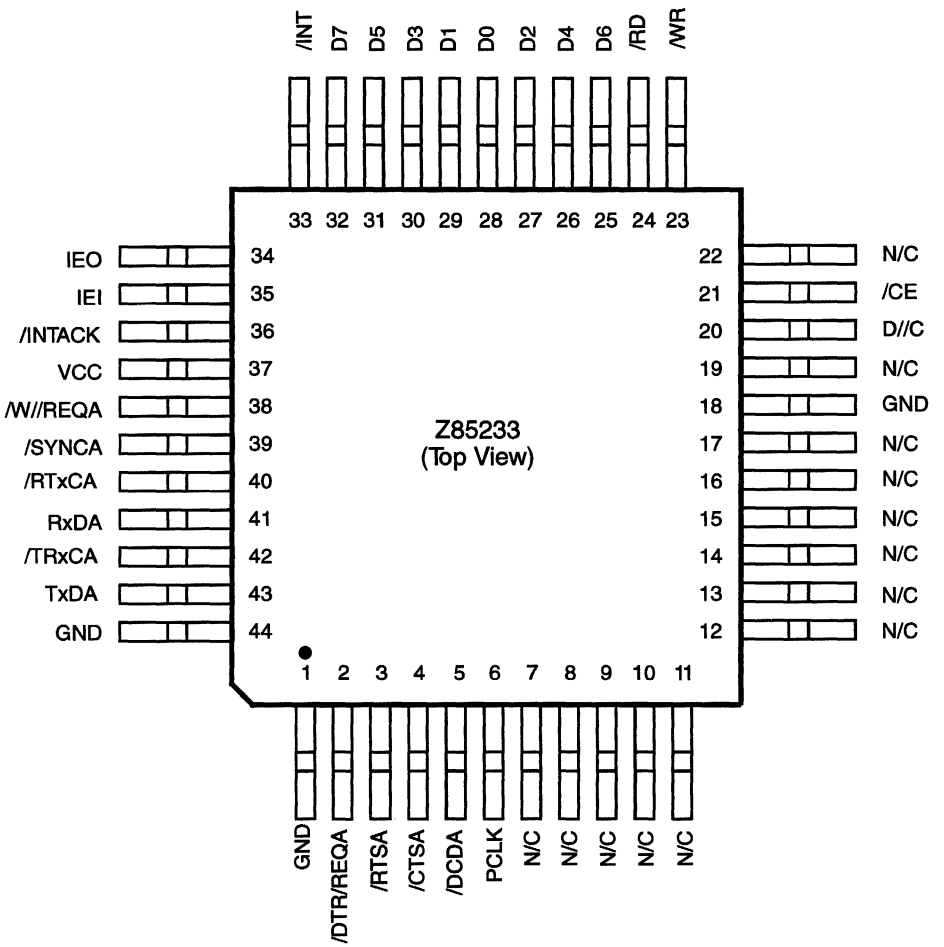


Figure 3. Z85233 PQFP Pin Assignments

5

PIN DESCRIPTIONS (Continued)

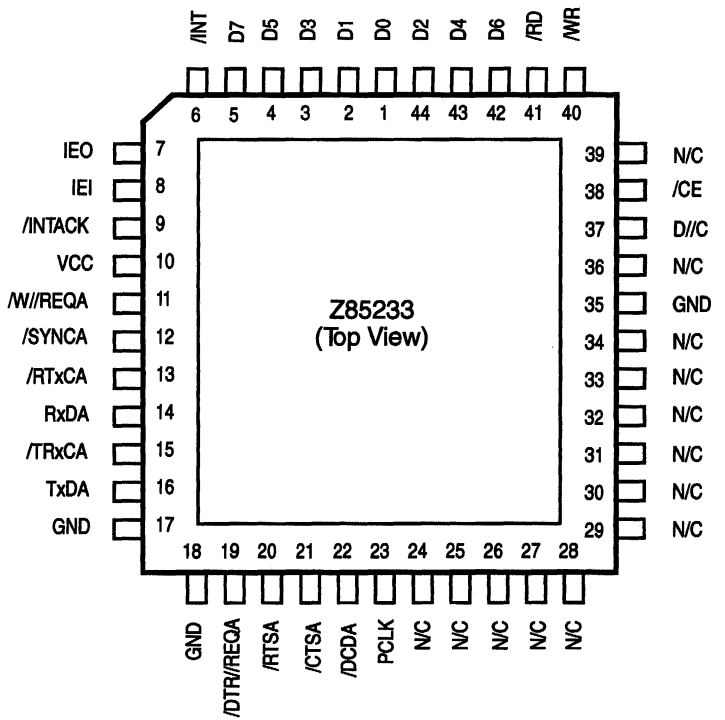


Figure 4. Z85233 PLCC Pin Assignments

Table 1. QFP Pin Identification

Pin No.	Symbol	Function	Direction
37	V_{CC}	Power Supply	Input
1, 18, 37	GND	Ground	Input
25-32	D7-D0	Data Bus	Input/Output
21	/CE	Chip Enable	Input
20	D//C	Data/Control Set	Input
24	/RD	Read	Input
23	/WR	Write	Input
33	/INT	Interrupt	Output
36	/INTACK	Interrupt Acknowledge	Input
35	IEI	Interrupt Enable In	Input
34	IEO	Interrupt Enable Out	Output
4	/CTSA	Clear To Send	Input
5	/DCDA	Data Carrier Detect	Input
3	/RTSA	Request To Send	Output
2	/DTR//REQA	Data Terminal Ready/Request	Output
39	/SYNCA	Synchronization	Input/Output
38	W//REQA	Wait/Request	Output
41	RxDA	Receive Data	Input
40	/RTxCA	Receive/Transmit Clock	Input
43	TxDA	Transmit Data	Output
42	/TRxCA	Transmit/Receive Clock	Input/Output
6	PCLK	Clock	Input

PIN DESCRIPTIONS (Continued)

Table 1. PLCC Pin Identification (Continued)

Pin No.	Symbol	Function	Direction
10	V_{CC}	Power Supply	Input
17, 18, 35	GND	Ground	Input
1-5, 42-44	D7-D0	Data Bus	Input/Output
38	/CE	Chip Enable	Input
37	D//C	Data/Control Set	Input
41	/RD	Read	Input
40	/WR	Write	Input
6	/INT	Interrupt	Output
9	/INTACK	Interrupt Acknowledge	Input
8	IEI	Interrupt Enable In	Input
7	IEO	Interrupt Enable Out	Output
21	/CTSA	Clear To Send	Input
22	/DCDA	Data Carrier Detect	Input
20	/RTSA	Request To Send	Output
19	/DTR//REQA	Data Terminal Ready/Request	Output
12	/SYNCA	Synchronization	Input/Output
11	W//REQA	Wait/Request	Output
14	RxDA	Receive Data	Input
13	/RTxCA	Receive/Transmit Clock	Input
16	TxDA	Transmit Data	Output
15	/TRxCA	Transmit/Receive Clock	Input/Output
23	PCLK	Clock	Input

/CTSA *Clear To Send* (input, active Low). This pin functions as transmitter enable if it is programmed for Auto Enable (WR3, D5 = 1). A Low on the input enables the transmitter. If not programmed as Auto Enable, it may be used as a general-purpose input pin. The input is Schmitt-trigger buffered to accommodate a slow rise time input. The EMSCC detects pulses on this input and can interrupt the CPU on both logic level transitions.

/DCDA *Data Carrier Detect* (input, active Low). This pin functions as receiver enable if it is programmed for Auto Enable (WR3, D5 = 1); otherwise it is used as a general purpose input pin. The pin is Schmitt-trigger buffered to accommodate a slow rise-time signal. The EMSCC detects pulses on this pin and can interrupt the CPU on both logic level transitions.

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

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

/SYNCA *Synchronization* (input or output, active Low). This pin can act either as an input, output, or part of the crystal oscillator circuit. In the 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 affects 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, this line also acts as an input. 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, this pin acts as an output and is active only during the part of the receive clock cycle in which the synchronous condition is latched. This output is active each time a synchronization pattern is recognized (regardless of character boundaries). In SDLC mode, this pin acts as an output and is valid on receipt of a flag. The /SYNC pin switches from input to output when monosync, bisync, or SDLC is programmed in WR4 and sync modes are enabled.

/DTR//REQA *Data Terminal Ready/Request* (output, active Low). This pin is programmed (WR14, D2) to serve either as a general-purpose output or as a DMA Request line. When programmed for the DTR function (WR14, D2 = 0), this output follows the state programmed into the DTR bit of Write Register 5 (WR5, D7). When programmed for Request mode (WR14, D2 = 1), this pin serves as a DMA Request for the transmitter.

When used as a DMA request line, 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 *Wait/Request* (output, open-drain when programmed for Wait function, driven High or Low when programmed for Ready function). This dual-purpose output may be programmed as a Request line for a DMA controller or as a Wait line which synchronizes the CPU to the EMSCC data rate. The reset state is Wait.

RxDA *Receive Data* (input, active High). This input signal receives serial data at standard TTL levels.

/RTxCA *Receive/Transmit Clock* (input, active Low). This pin can be programmed to several 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 clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.

TxDA *Transmit Data* (output, active High). This output signal transmits serial data at standard TTL levels.

/TRxCA *Transmit/Receive Clock* (input or output, active Low). This pin 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 EMSCC 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.

PIN DESCRIPTIONS (Continued)

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 EMSCC interrupt, or the EMSCC 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 EMSCC 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 EMSCC 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 EMSCC places the interrupt vector on the data bus when /RD goes active. /INTACK is latched by the rising edge of PCLK.

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

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

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

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

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

Note: All ground signals must be connected and must not be left floating.

FUNCTIONAL DESCRIPTION

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

The EMSCC's peripheral and datacommunication features are described in the following sections. The 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. See the ESCC Technical Manual for full details on using the EMSCC.

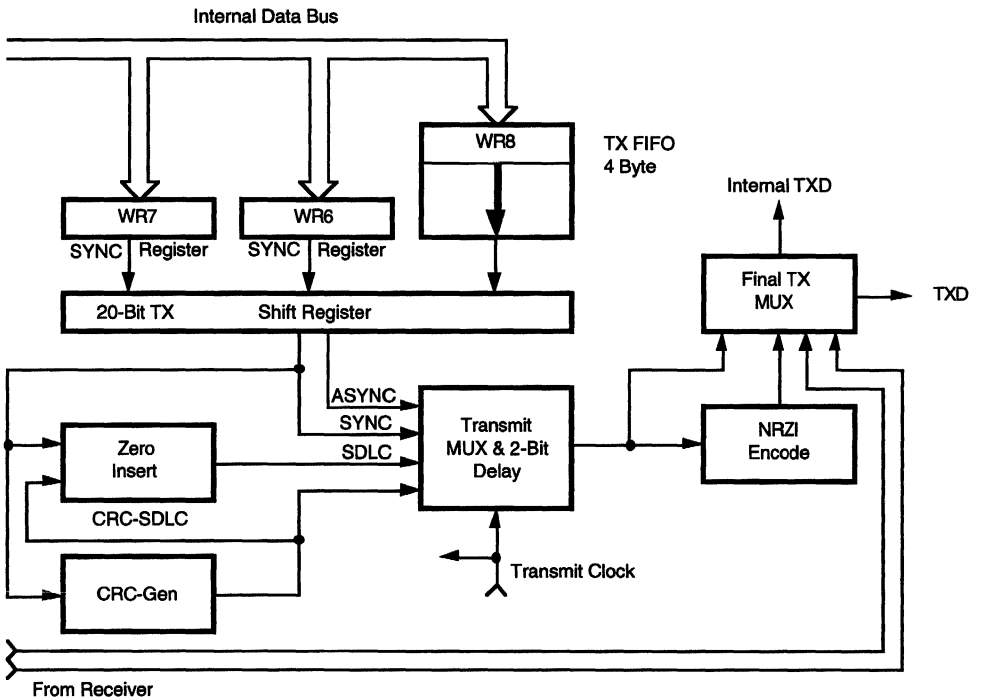


Figure 5. EMSCC Transmit Data Path

5

FUNCTIONAL DESCRIPTION (Continued)

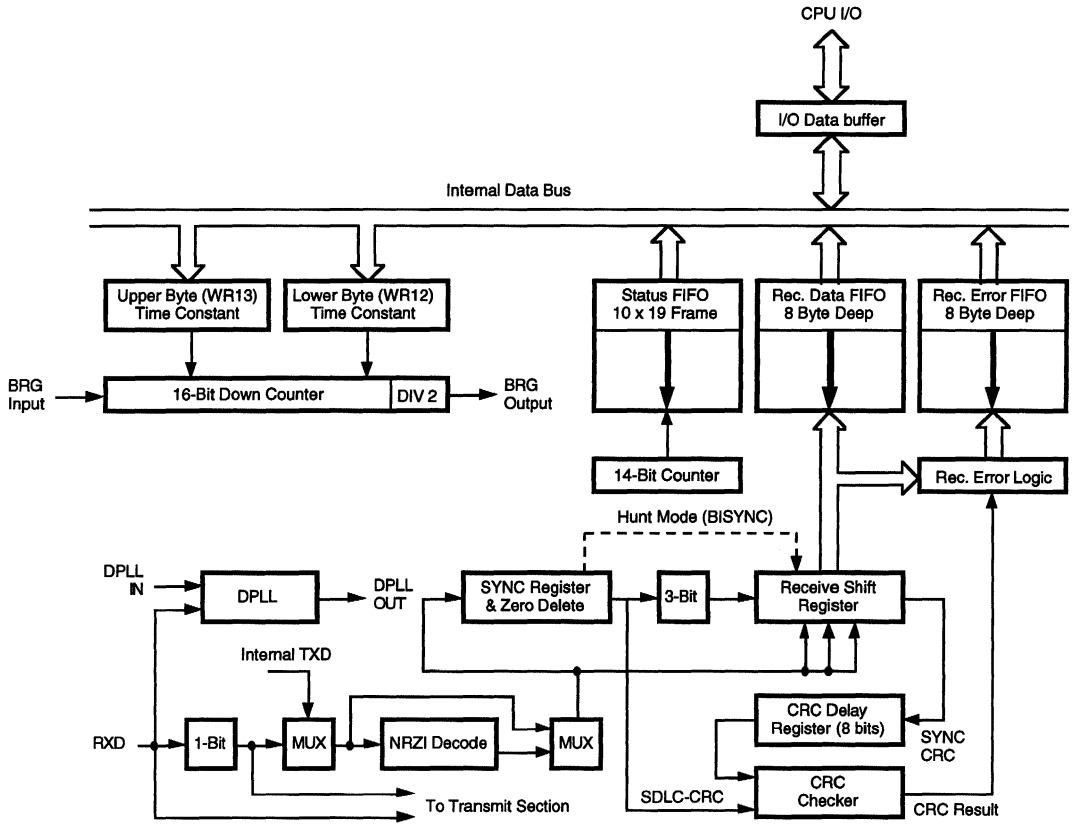


Figure 6. EMSCC Receive Data Path

I/O INTERFACE CAPABILITIES

System communication to and from the EMSCC is done through the EMSCC's register set. There are 17 write registers and 15 read registers. Many of the new features on the EMSCC are enabled through a new register in the EMSCC: Write Register 7 Prime (WR7'). This new register can be accessed if bit D0 of WR15 is set. Table 1 lists all of the EMSCC'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:

WR4	Write Register 4
RR3	Read Register 3

Table 2. EMSCC 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.
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.
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.
RR2	Unmodified interrupt vector (if VIS = 0). Modified interrupt vector (if VIS = 1).
RR3	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 10 x 19 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).

I/O INTERFACE CAPABILITIES (Continued)

There are three choices to move data into and out of the EMSCC: 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 EMSCC 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 the channel resides in one register.

Interrupts. The EMSCC'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 EMSCC's requests for data transfers to be tuned to the system interrupt response time.

Nested interrupts are supported with the interrupt acknowledge feature (/INTACK pin) of the EMSCC. 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 EMSCC interrupt or another higher priority device can interrupt the CPU. When an EMSCC 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 EMSCC can modify three bits in this vector to indicate status by setting the VIS bit. (WR9, D0); vector read will have status included.

Each of the three sources of interrupts in the EMSCC (Transmit, Receive, and External/Status interrupts) 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 EMSCC 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.

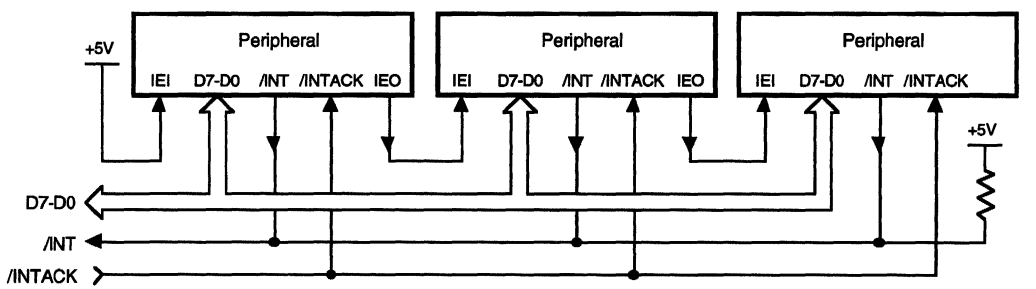


Figure 7. EMSCC Interrupt Priority Schedule

The EMSCC 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 EMSCC. See the Z85233 Enhancements section for more details .

In the EMSCC, 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 EMSCC, 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 RR3.

The IUS bits signal that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority in the EMSCC and external to the EMSCC 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 EMSCC 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 Receiver, Transmit, and External/Status interrupts prioritized in that order. 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:

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.

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 EMSCC 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 EMSCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers. The Block Transfer mode uses 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 EMSCC REQUEST output indicates the EMSCC is ready to transfer data to or from memory. To the CPU, the WAIT line indicates that the EMSCC 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 EMSCC can be programmed to deassert the /DTR//REQUEST pin with the same timing as the /WAIT//REQUEST pin if WR7' D4 is set.

EMSCC DATA COMMUNICATIONS CAPABILITIES

The EMSCC provides a full-duplex programmable channel for use in any common asynchronous or synchronous data communication protocols (Figure 8).

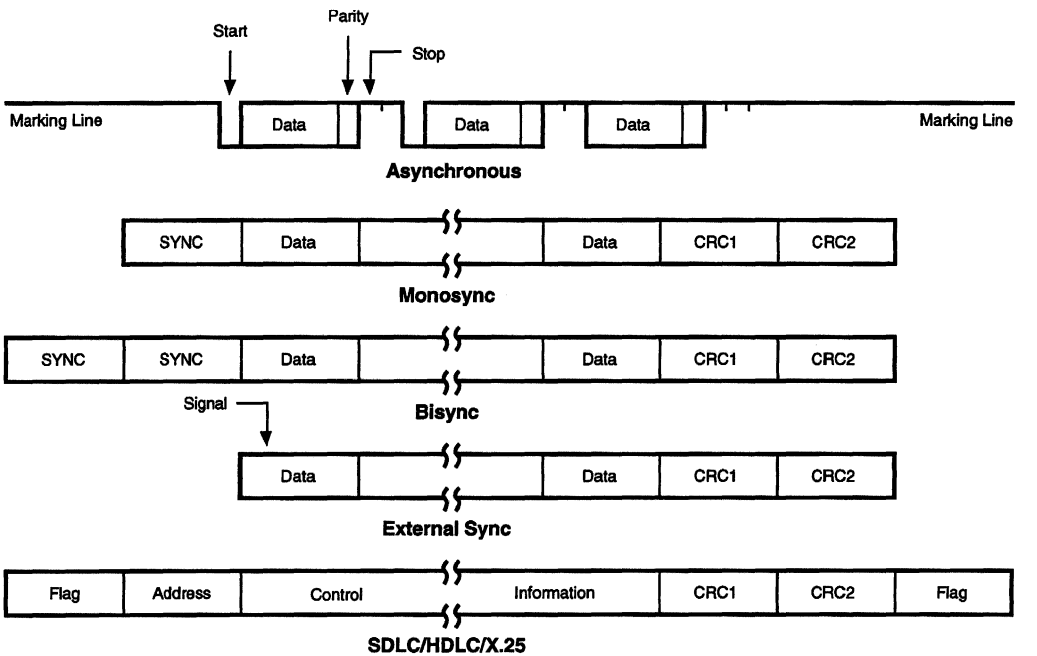


Figure 8. Some EMSCC Protocols

The EMSCC has significant improvements to its data communication 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 Z85233 Enhancements 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). 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 EMSCC 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 EMSCC 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 EMSCC by overlapping the larger pattern across multiple incoming synchronous characters as shown in Figure 9.

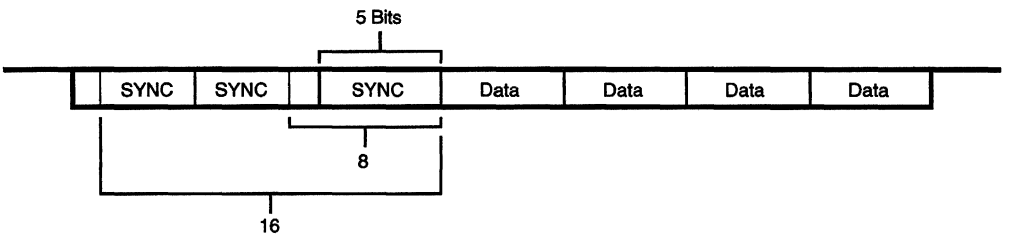


Figure 9. Detecting 5- or 7-Bit Synchronous Characters

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 1s or all 0s. The EMSCC 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 EMSCC 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 EMSCC 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 EMSCC 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 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

5

EMSCC DATA COMMUNICATIONS CAPABILITIES (Continued)

frame is available in the status registers. In SDLC mode, the EMSCC 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.

SDLC Loop Mode. The EMSCC 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 EMSCC performs the functions of a secondary station while an EMSCC operating in regular SDLC mode acts as a controller (Figure 10). SDLC loop mode can be selected by setting WR10 bit D1.

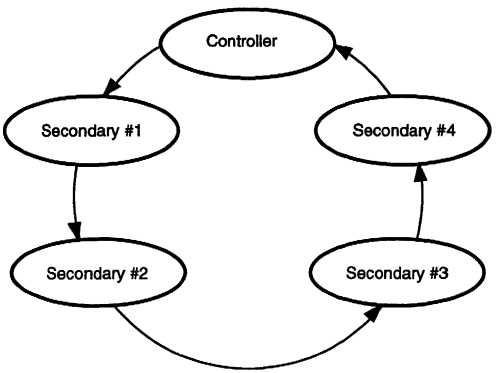


Figure 10. An SDLC Loop

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 EMSCC'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 6 and 7 are only accessible when the SDLC FIFO is enabled. The 10 x 19 status FIFO is separate from the 8-byte receive data FIFO.

Baud Rate Generator. The EMSCC contains a programmable baud rate generator. The generator consists of two 8-bit time constant registers which 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 EMSCC 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 EMSCC 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 EMSCC via the TRxC pin (if this pin is not being used as an input).

Data Encoding. The EMSCC 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 EMSCC 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 EMSCC 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 EMSCC 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.

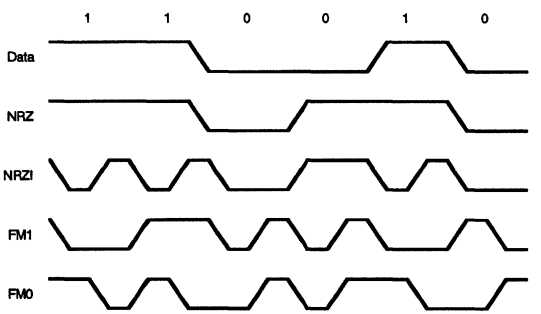


Figure 11. Data Encoding Methods

5

NEW FEATURE DESCRIPTION

The following is a detailed description of the enhancements to the Z85233, EMSCC from the standard SCC.

4-Byte Deep Transmit FIFO
The EMSCC 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 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 IS NOT the reset state.

For applications where the frequency of interrupts is important, the transmit interrupt service routine can be optimized by programming the EMSCC 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 EMSCC 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 EMSCC 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 EMSCC to facilitate the programming of six new features. The format of this register is shown in Figure 12.

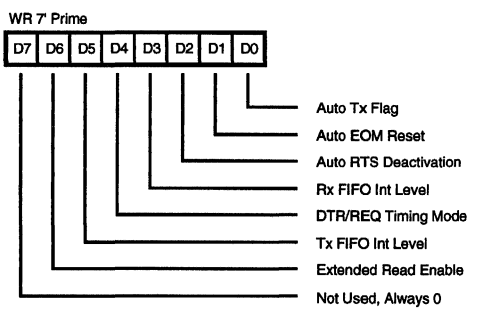


Figure 12. Write Register 7' (7 prime)

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.

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.

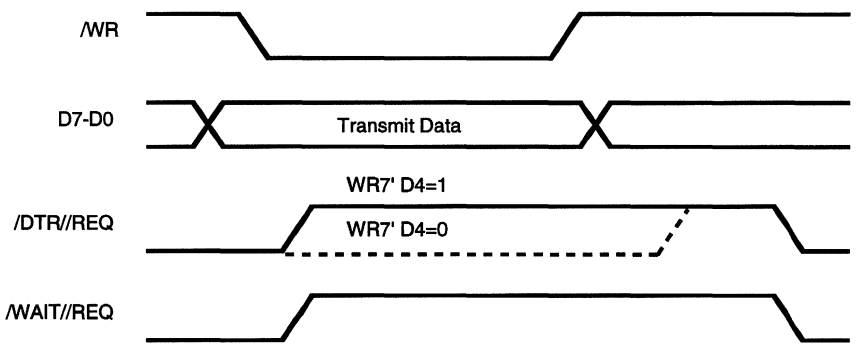


Figure 13. DMA Request on Transmit Deactivation Timing

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 Receive FIFO Interrupt Level 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 EMSCC 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 EMSCC 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 EMSCC 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.

5

NEW FEATURE DESCRIPTION (Continued)

Modified Data Bus Timing

The EMSCC's latching of the Data Bus 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 Z85233 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 20 ns after the falling edge of /WR regardless of the system (PCLK) clock rate. The databus hold time, spec #30, remains at 0 ns.

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 EMSCC 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 EMSCC, 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 EMSCC 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 EMSCC 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.

Improved Transmit Interrupt Handling in Synchronous Modes

The EMSCC 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.

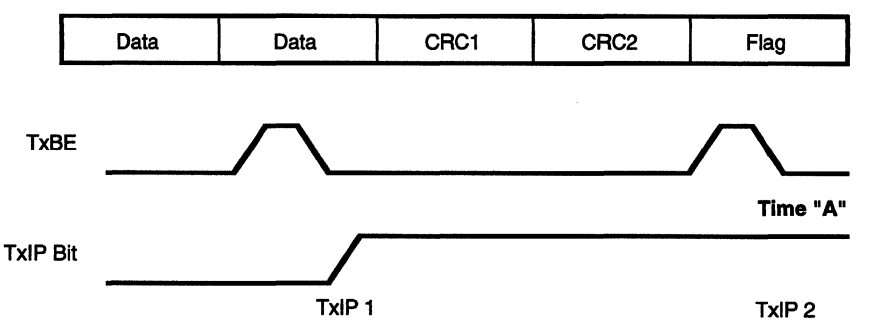


Figure 14. TxIP Latching

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).

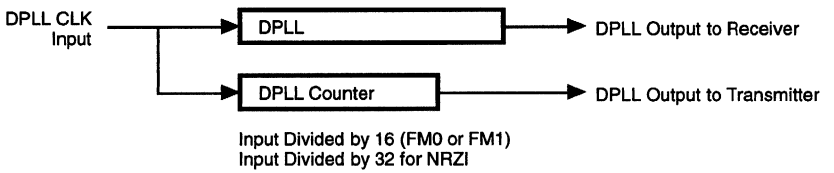


Figure 15. DPLL Outputs

Read Register 0 Status Latched During Read Cycle

The contents of Read Register 0, RR0, are latched during a read to this register. The EMSCC 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 Z85233 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 Vector Include Status (VIS) is reset (WR9, D0 = 0), and RR2 is read, the vector returned is unmodified. If VIS is set (WR9, D0 = 1), and the vector returned in RR2 is modified to indicate the source of the interrupt, the Non 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).

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 EMSCC 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 Z85233 SDLC Frame Status FIFO enhancement maximizes the EMSCC'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-bit deep by 19-bit wide 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.

NEW FEATURE DESCRIPTION (Continued)

Summarizing the operation; data is received, assembled, and loaded into the 8-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 EMSCC 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 power-on reset). 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 details on the added registers, refer to Figure 17. 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. 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 EMSCC 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 18.

SDLC Status FIFO Anti-Lock Feature. When the Frame Status FIFO is enabled and the EMSCC 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 (Figure 17). 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.

Frame Status FIFO Circuitry

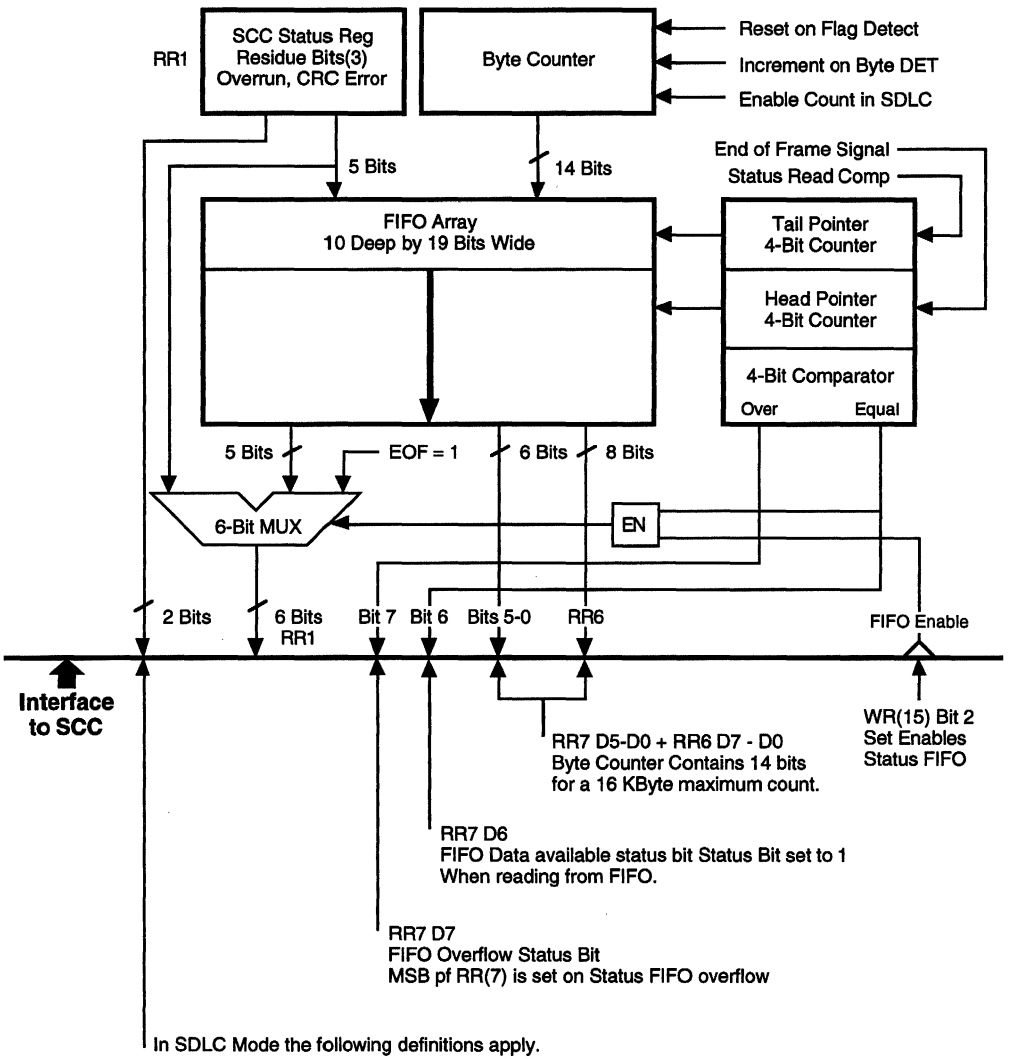


Figure 16. SDLC Frame Status FIFO

5

NEW FEATURE DESCRIPTION (Continued)

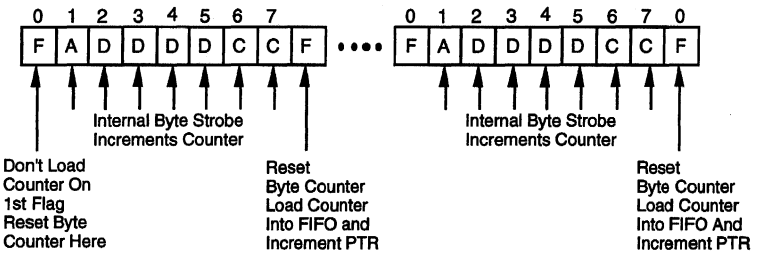


Figure 17. SDLC Byte Counting Detail

PROGRAMMING

The EMSCC contains write registers which are programmed by the system.

In the EMSCC, 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 EMSCC 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 EMSCC contains 16 write registers (17 counting the transmit buffer). A new register, WR7', was added to the EMSCC and may be written to if WR15 D0 is set. Figure 18 shows the format of each write register.

Read Registers. The EMSCC contains ten read registers 11, counting the receive buffer RR8). 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 if VIS bit is reset (VIS = 0) or the vector modified by status information if VIS bit is set (if VIS = 1). 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 19 shows the format of each read register.

CONTROL REGISTERS

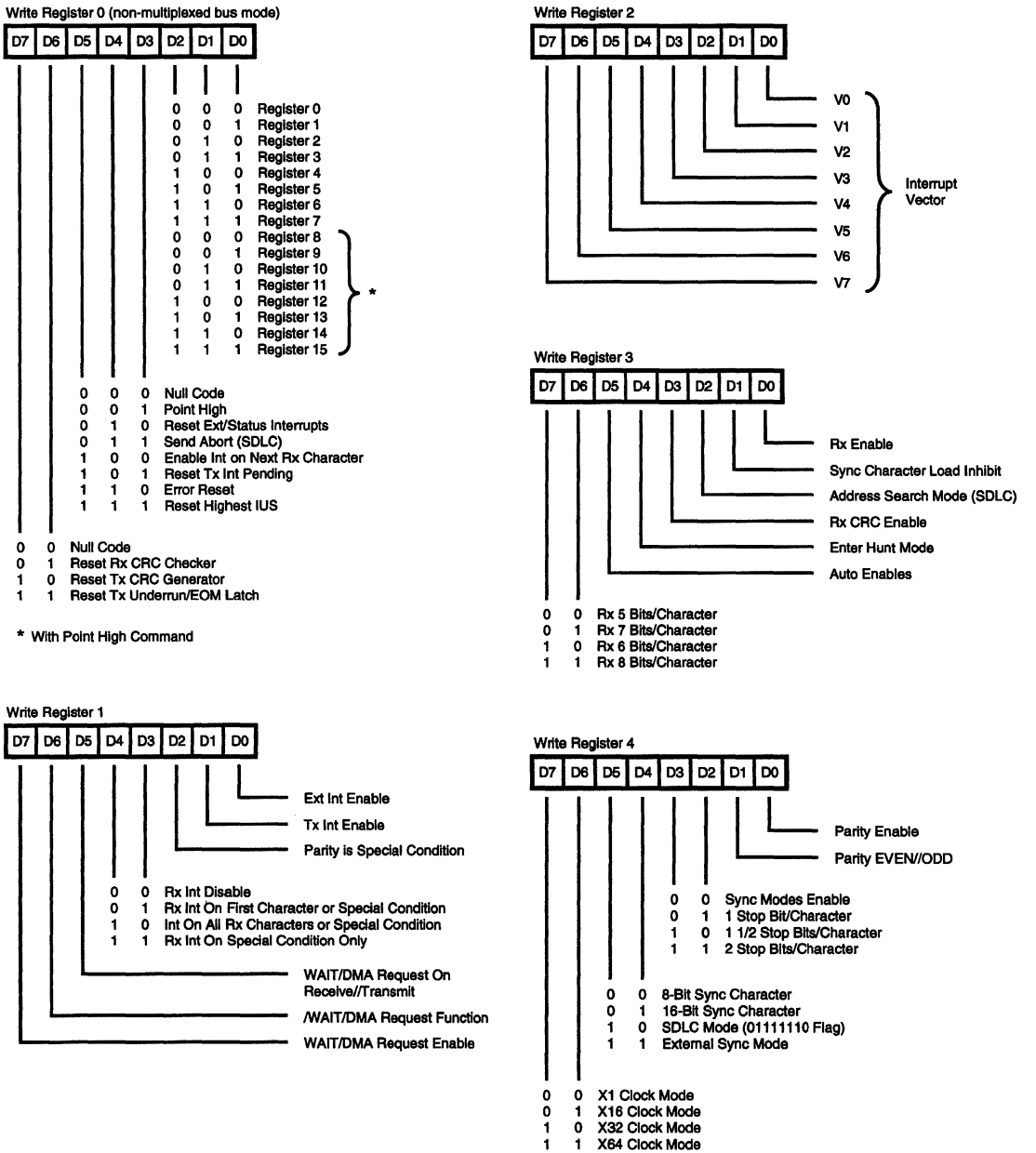
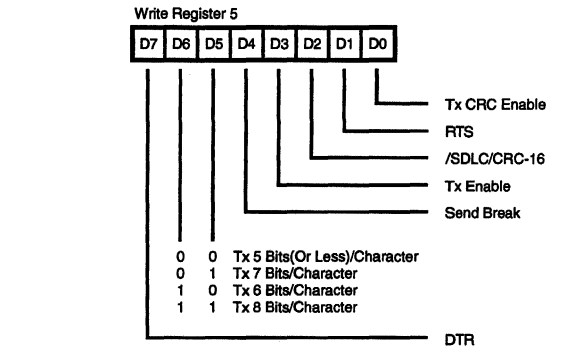
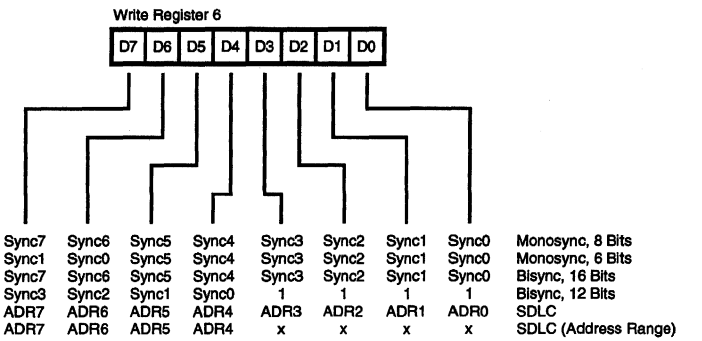


Figure 18. Write Register Bit Functions

5

CONTROL REGISTERS (Continued)





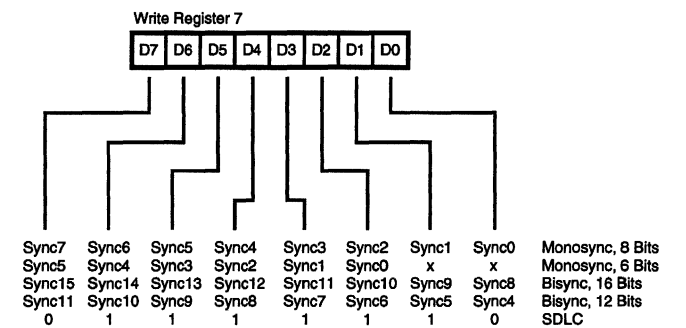


Figure 18. Write Register Bit Functions (Continued)

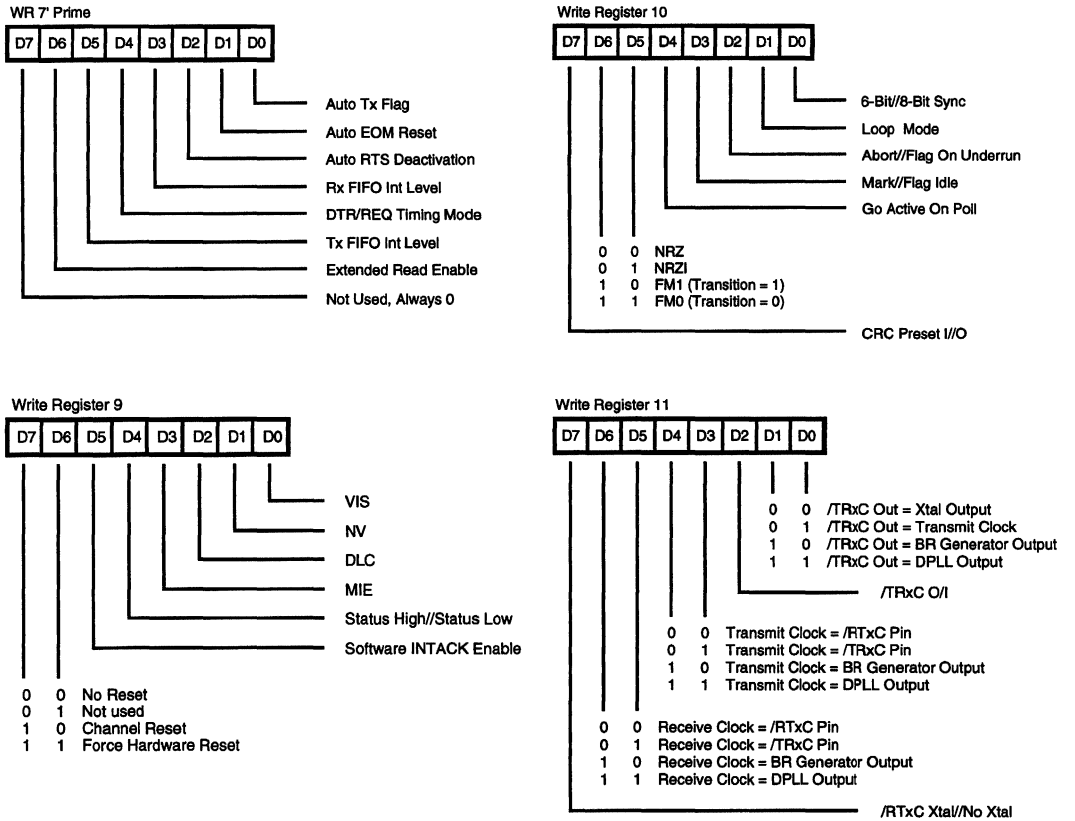


Figure 18. Write Register Bit Functions (Continued)

5

CONTROL REGISTERS (Continued)

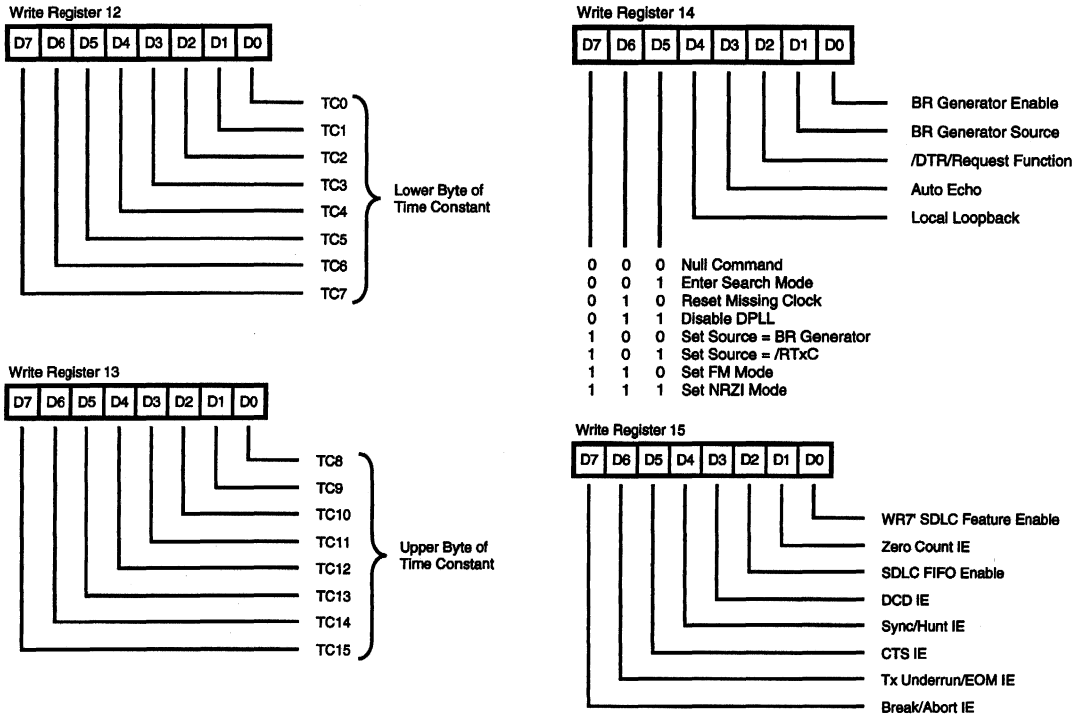
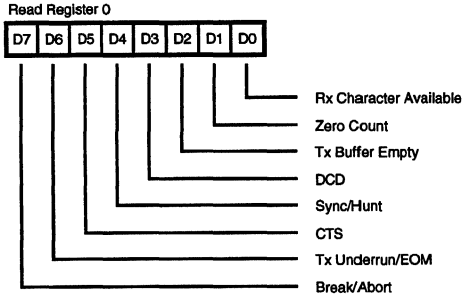
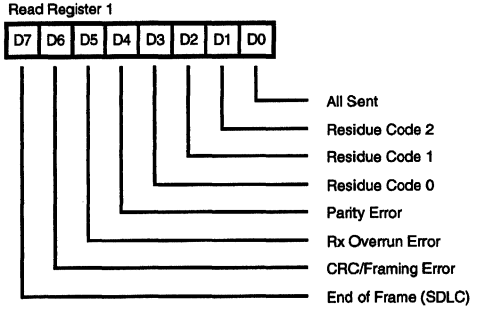


Figure 18. Write Register Bit Functions (Continued)

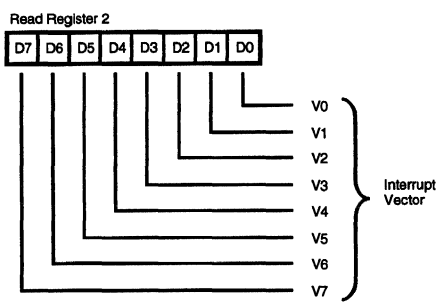
Read Register 0



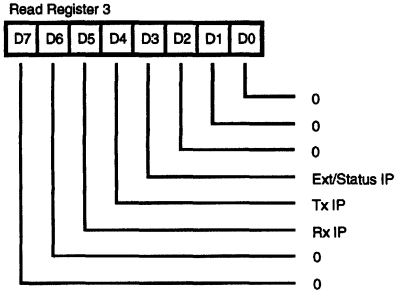
Read Register 1



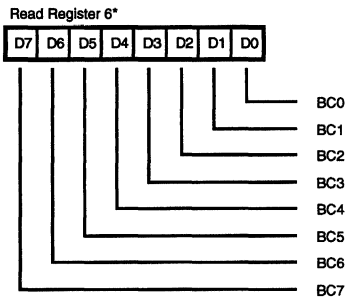
Read Register 2



Read Register 3



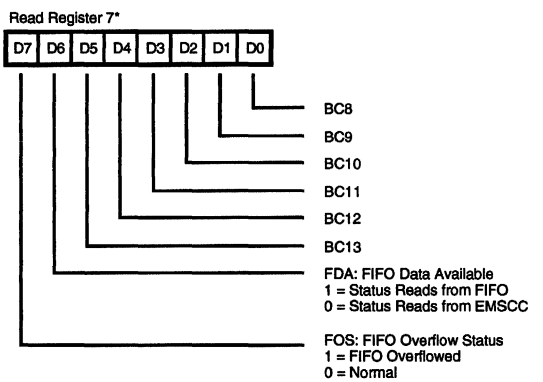
Read Register 6*



*Can only be accessed if the SDLC FIFO enhancement is enabled (WR15 bit D2 set to 1)

SDLC FIFO Status and Byte Count (LSB)

Read Register 7*



*Can only be accessed if the SDLC FIFO enhancement is enabled (WR15 bit D2 set to 1)

SDLC FIFO Status and Byte Count (LSB)

Figure 19. Read Register Bit Functions

5

CONTROL REGISTERS (Continued)

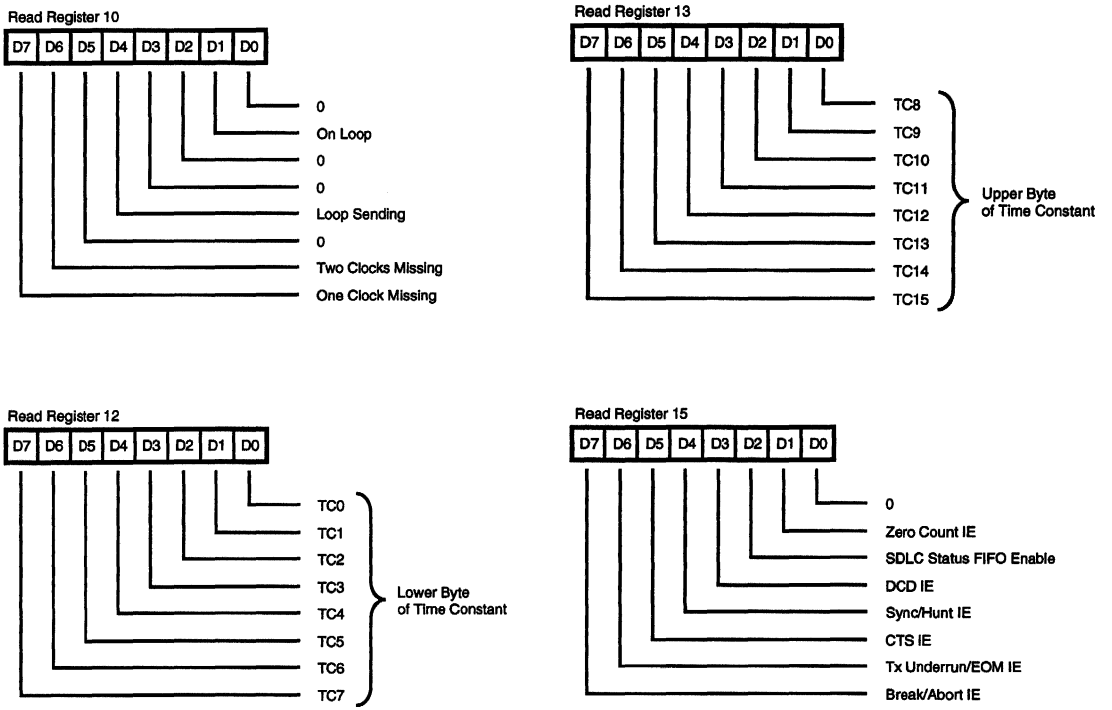


Figure 19. Read Register Bit Functions (Continued)

Z85233 TIMING

The EMSCC 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 EMSCC. The recovery time required for proper operation is specified from the falling edge of /WR or /RD in the first transaction involving the EMSCC to the falling edge of /WR or /RD in the second transaction involving the EMSCC. 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 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.

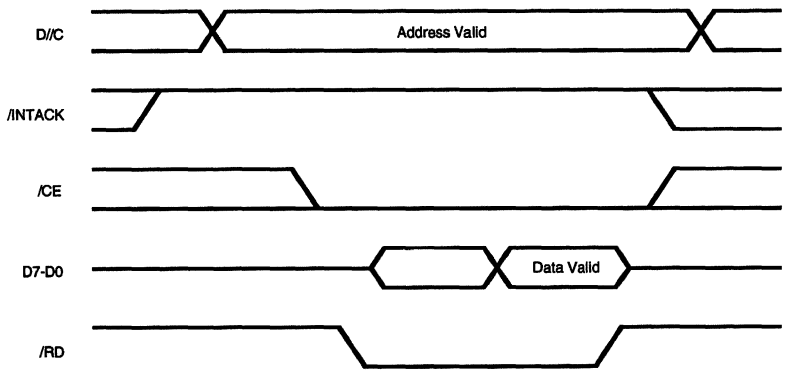


Figure 20. Read Cycle Timing

Write Cycle Timing. Figure 21 illustrates Write cycle timing. Addresses on 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 EMSCC 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.

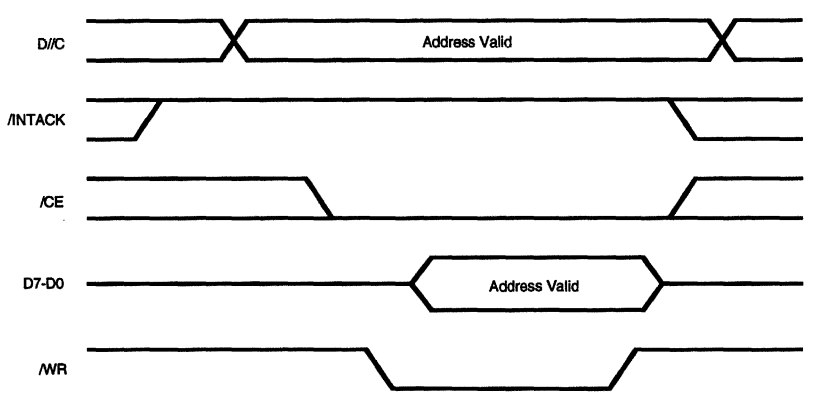


Figure 21. Write Cycle Timing

Z85233 TIMING (Continued)

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 EMSCC and IEI is High when /RD falls, the Acknowledge cycle is intended for the EMSCC. In this case, the EMSCC 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 EMSCC. 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.

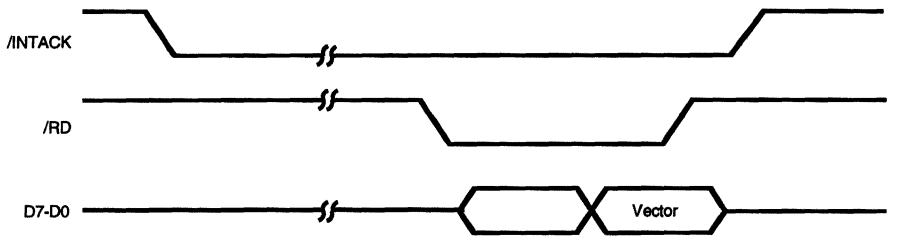


Figure 22. Interrupt Acknowledge Cycle Timing

ABSOLUTE MAXIMUM RATINGS

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

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

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

STANDARD TEST CONDITIONS

The DC Characteristics and capacitance 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:

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

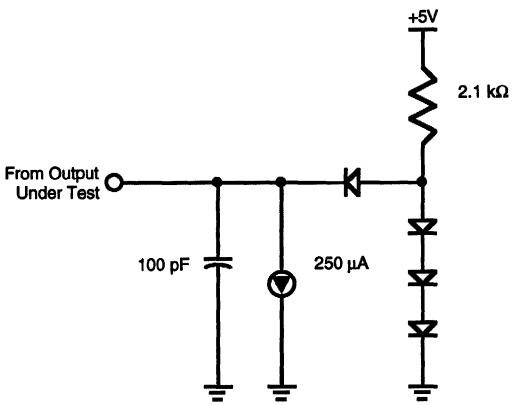


Figure 23. Standard Test Load

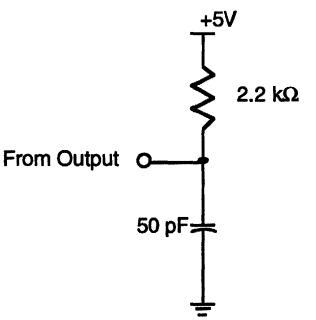


Figure 24. Open-Drain Test Load

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.

MISCELLANEOUS

Gate Count - 7000

DC CHARACTERISTICS
Z85233

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		4	8 (10 MHz)	mA	V_{CC} = 5V V_{IH} = 4.8 V_{IL} = 0.2V
			5	10 (16 MHz)	mA	Crystal Oscillators off
			6	12 (20 MHz)	mA	
$I_{CC(OSC)}$	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

Z85233 Read and Write Timing Diagram

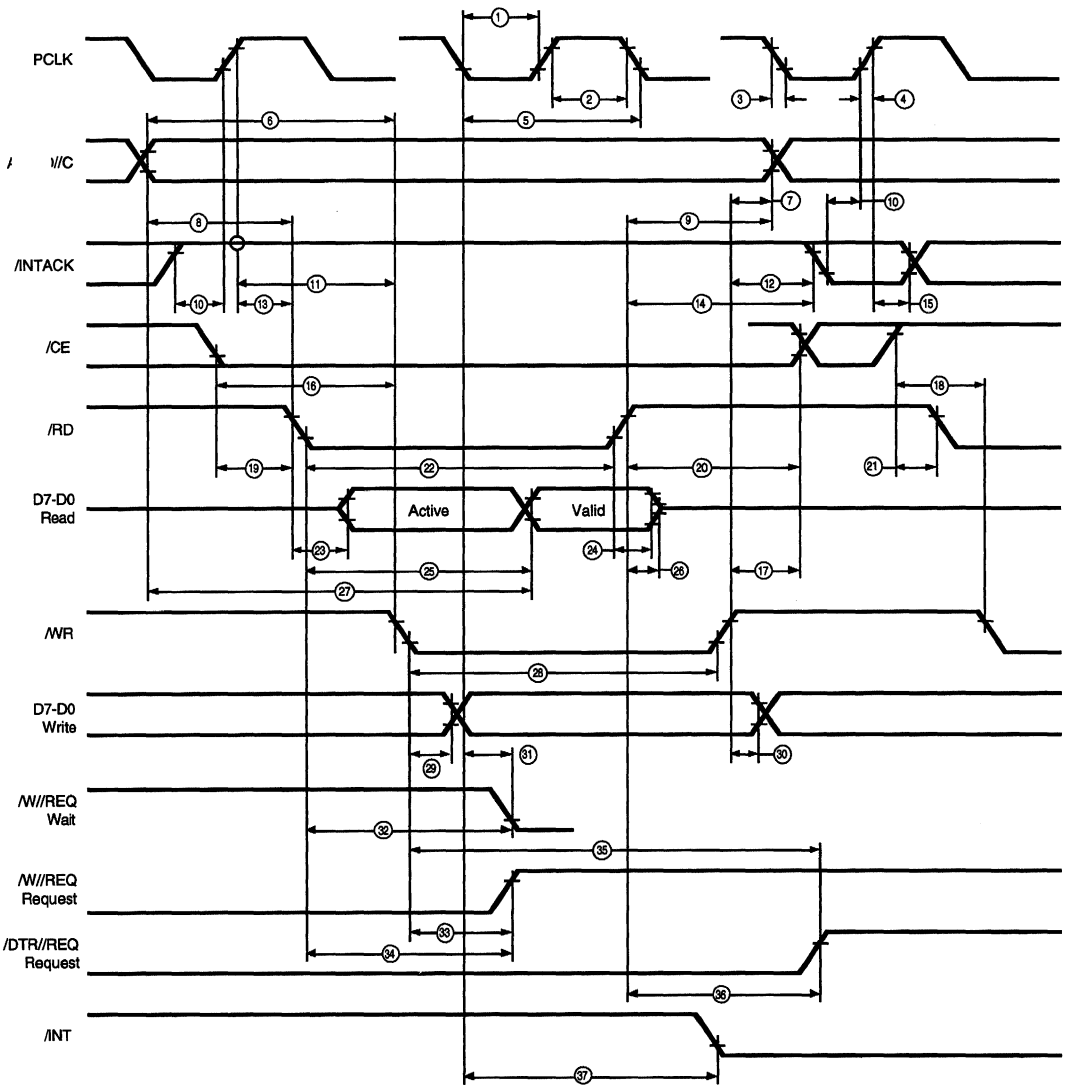


Figure 25. Read and Write Timing Diagram

AC CHARACTERISTICS

Z85233 Timing Diagrams (Continued)

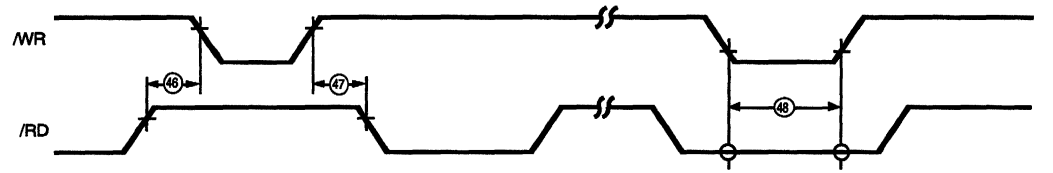


Figure 26. Reset Timing Diagram

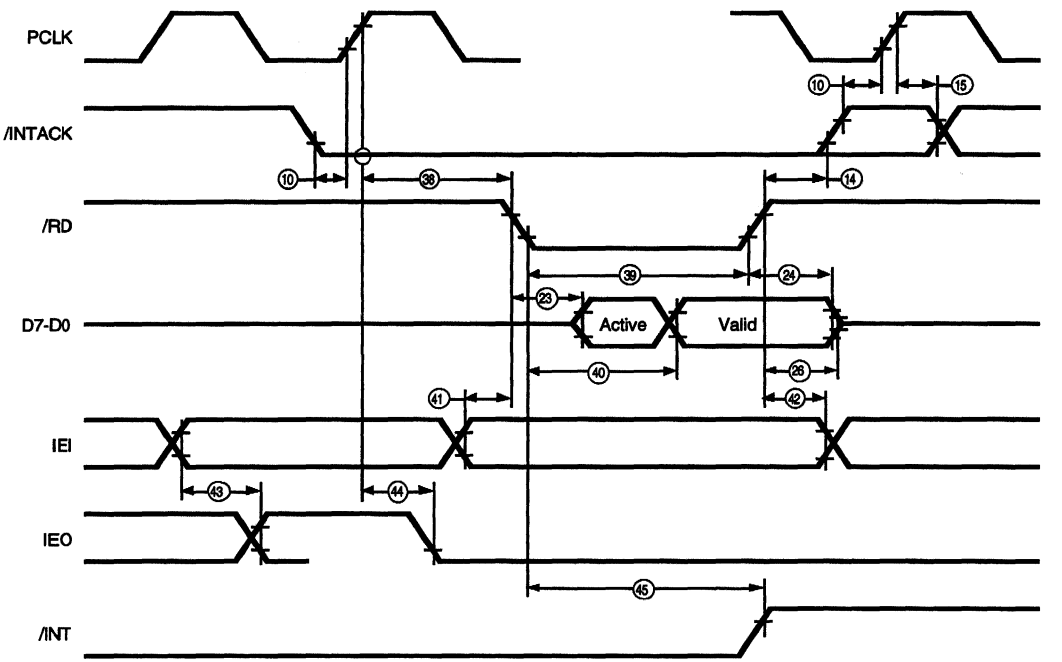


Figure 27. Interrupt Acknowledge Timing Diagram

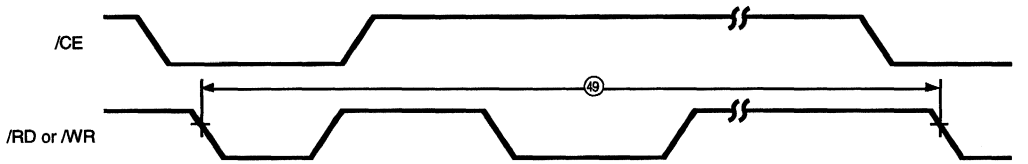


Figure 28. Cycle Timing Diagram

AC CHARACTERISTICS

Z85233 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	TslA(PC)	/INTACK to PCLK Rise Setup Time	20		15		15		
11	TslAi(WR)	/INTACK to /WR Fall Setup Time	130		70		65		[1]
12	ThlA(WR)	/INTACK to /WR Rise Hold Time	0		0		0		
13	TslAi(RD)	/INTACK to /RD Fall Setup Time	130		70		65		[1]
14	ThlA(RD)	/INTACK to /RD Rise Hold Time	0		0		0		
15	ThlA(PC)	/INTACK to PCLK Rise Hold Time	30		15		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	50		30		25		
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	50		30		25		[1]
22	TwRDl	/RD Low Width	125	2TcPc	70	2TcPc	65	2TcPc	[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	TdRDl(DR)	/RD Fall to Read Data Valid Delay		120		70		65	
26	TdRD(DRz)	/RD Rise to Read Data Float Delay		35		30		30	
27	TdA(DR)	Addr to Read Data Valid Delay		180		100		90	
28	TwWRl	/WR Low Width	125		75		65		

5

AC CHARACTERISTICS

Z85233 Read and Write Timing Table (Continued)

No	Symbol	Parameter	10 MHz Min	10 MHz Max	16 MHz Min	16 MHz Max	20 MHz Min	20 MHz Max	Notes
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	
35b	TdWRr(REQ)	/WR Fall to /DTR//REQ Not Valid		100		70		65	[6]
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	TsIEI(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		80	
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 EMSCC.

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

[5] Parameter is system dependent. For any EMSCC 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 EMSCC 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

Z85233 General Timing Diagram

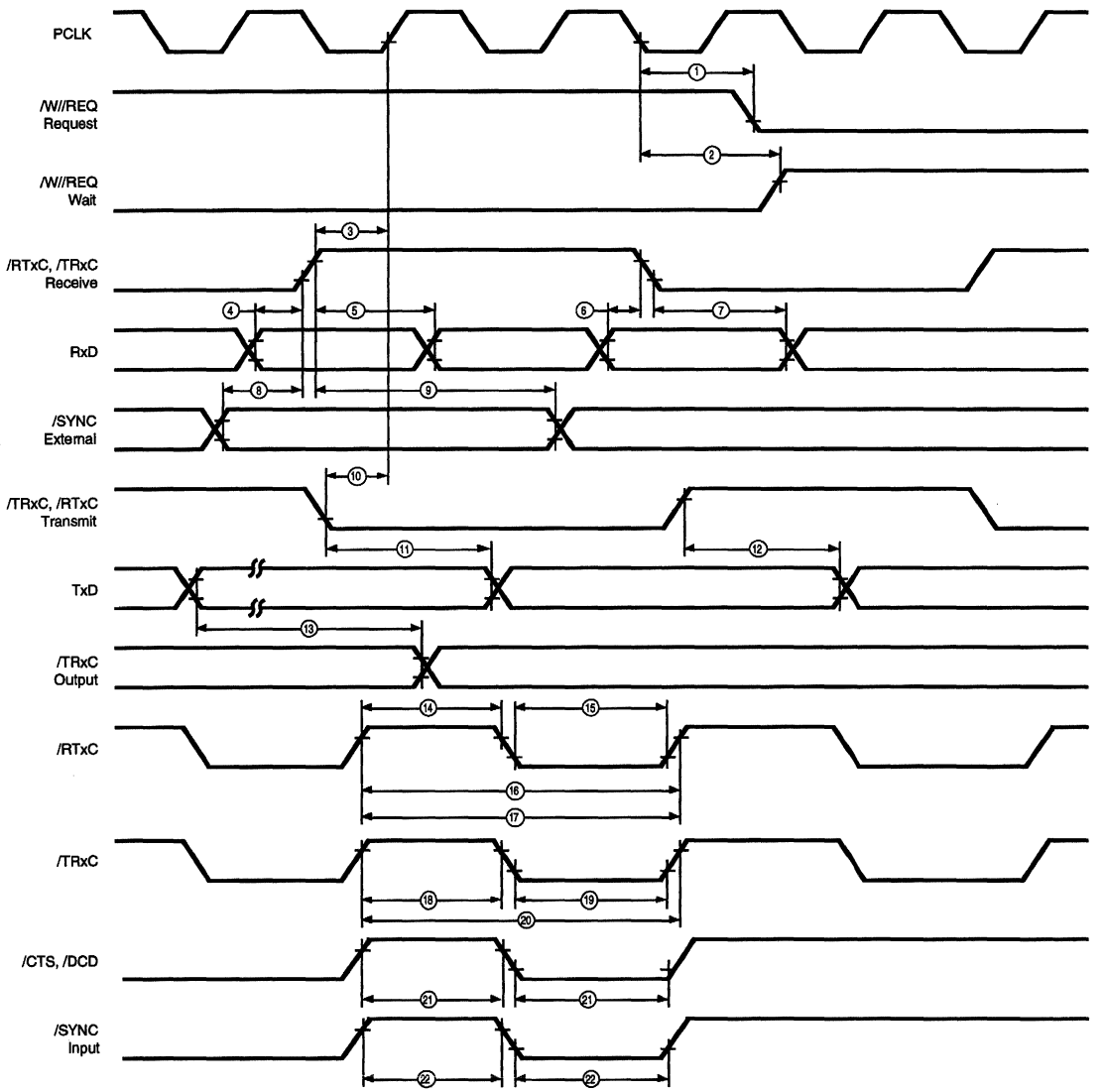


Figure 29. General Timing Diagram

AC CHARACTERISTICS

Z85233 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
1a	TdPC(REQ)	/PCLK to W/REQ Valid		200		80		70	
1b	TdPC(REQ)	/PCLK to DTR/REQ Valid		200		80		70	[9]
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.
[9] Parameter applies only when WRT' D4 is set to 1.

AC CHARACTERISTICS

Z85233 System Timing Diagram (Preliminary)

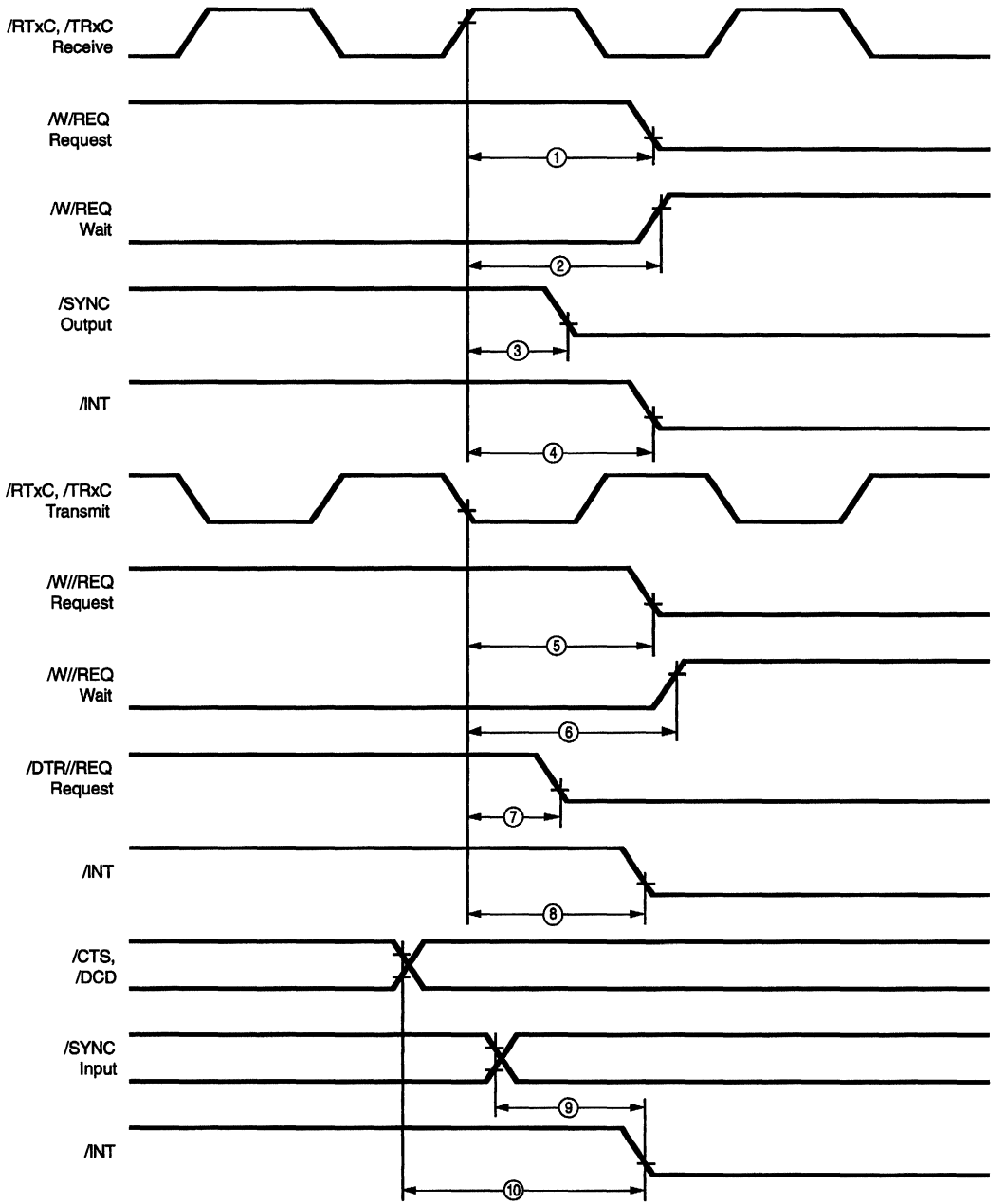


Figure 30. System Timing

AC CHARACTERISTICS

Z85233 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 [4]
1	TdRXC(REQ)	/RXC to /W//REQ Valid	13	17	13	17	13	18	[2]
2	TdRXC(W)	/RxC to /Wait Inactive	13	17	13	17	13	18	[1,2]
3	TdRXC(SY)	/RxC to /SYNC Valid	9	12	9	12	9	13	[2]
4	TdRXC(INT)	/RxC to /INT Valid	15	21	15	21	15	22	[1,2]
5	TdTXC(REQ)	/TxC to /W//REQ Valid	8	11	8	11	8	12	[3]
6	TdTXC(W)	/TxC to /Wait Inactive	8	14	8	14	8	15	[1,3]
7	TdTXC(DRQ)	/TxC to /DTR//REQ Valid	7	10	7	10	7	11	[3]
8	TdTXC(INT)	/TxC to /INT Valid	9	13	9	13	9	14	[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	3	8	3	8	3	9	[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.

[4] Units equal to TcPc

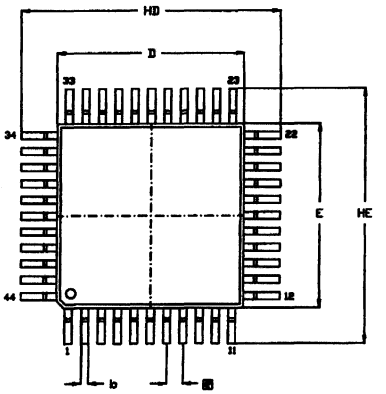
OTHER ZILOG DATA COMMUNICATIONS PRODUCTS

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.
Z85230	ESCC 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 with ISDN Time Slot Assigner.
Z16C31 IUSC™	MUSC with high performance dual channel DMA.
Z16C50 DDPLL™	Dual channel DPLL cell from the USC.

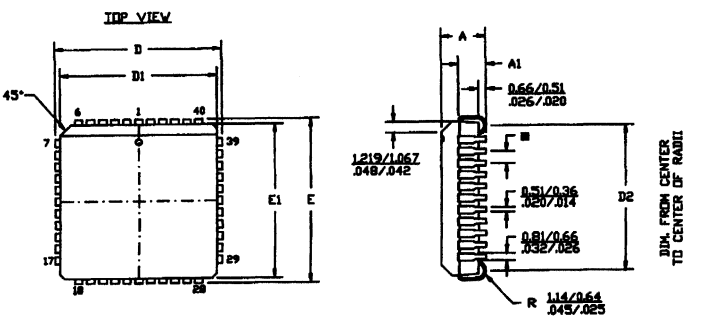
PACKAGE INFORMATION



NOTES:
1. CONTROLLING DIMENSIONS : MILLIMETER
2. LEAD COPLANARITY : MAX .10 mm / .004"

SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A1	0.05	0.25	.002	.010
A2	2.00	2.25	.078	.089
b	0.25	0.45	.010	.018
c	0.13	0.20	.005	.008
HD	13.70	14.30	.539	.563
D	9.90	10.10	.390	.398
HE	13.70	14.30	.539	.563
E	9.90	10.10	.390	.398
e	0.80 TYP		.031 TYP	
L	0.60	1.20	.024	.047

44-Pin QFP Package Diagram



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

SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	4.27	4.57	.168	.180
A1	2.67	2.92	.105	.115
D/E	17.40	17.65	.685	.695
D1/E1	16.51	16.66	.650	.656
D2	15.24	16.00	.600	.630
e	1.27 TYP		.050 TYP	

44-Pin PLCC Package Diagram

ORDERING INFORMATION

Z85233	10 MHz	16 MHz	20 MHz
44-Pin PQFP	Z8523310FSC	Z8523316FSC	Z8523320FSC
44-Pin PLCC	Z8523310VSC	Z8523316VSC	Z8523320VSC

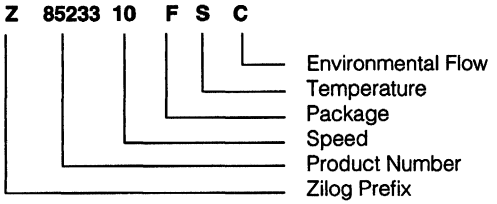
Package
V=Plastic Chip Carrier
C=Ceramic DIP
L=Ceramic LCC
F=Plastic Quad Flat Pack

Temperature
E=-40°C to +100°C
S=0°C to +70°C

Speeds
10=10 MHz
16=16 MHz
20=20 MHz

Environmental
C=Plastic Standard
D=Plastic Stressed
E=Hermetic Stressed

Example:
Z8523310FSC is a CMOS Single Channel 85230, 10 MHz, Plastic QFP, 0°C to +70°C, Plastic Standard Flow.



5

Z85C80

SCSCI™ SERIAL COMMUNICATIONS AND SMALL COMPUTER INTERFACE

FEATURES

- Low Power CMOS
- Two Independent, SCC Interface, Full-Duplex Channels, Each with a Separate Crystal Oscillator, Baud Rate Generator, and Digital Phase-Locked Loop for Clock Recovery.
  - 10 MHz @ 2.5 Mbit/sec
  - 16 MHz @ 4 Mbit/sec
- 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
- SCSI Interface Arbitration Support
- DMA or Programmed I/O Data Transfers
- Supports Normal DMA
- Memory or I/O Mapped CPU Interface
- Asynchronous Interface, Supports 3.0 MB/sec (Twice as Fast as the Competition)
- 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).

The Z85C80 is offered in a 68-pin PLCC package in both 10 MHz and 16 MHz speed grades for the SCC interface, and in 100-pin VQFP in the 16 MHz speed grade. The SCSI interface data transfer rate is 3.0 MB/sec. 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 V_{DD}.
- IEO output from SCC is not internally connected (N/C).
- READY output from SCSI is not internally connected (NC).
- /SYNCB output from the SCC is not internally connected (NC).
- /TRXCA and /CTSA inputs to the SCC are internally connected.
- /TRXCB and /CTSB inputs to the SCC are internally connected on the 68-pin PLCC package.

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 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. 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 as 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 handshake signals to support normal DMA operations with most DMA controllers available.

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

Power connections follow conventional descriptions below:

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

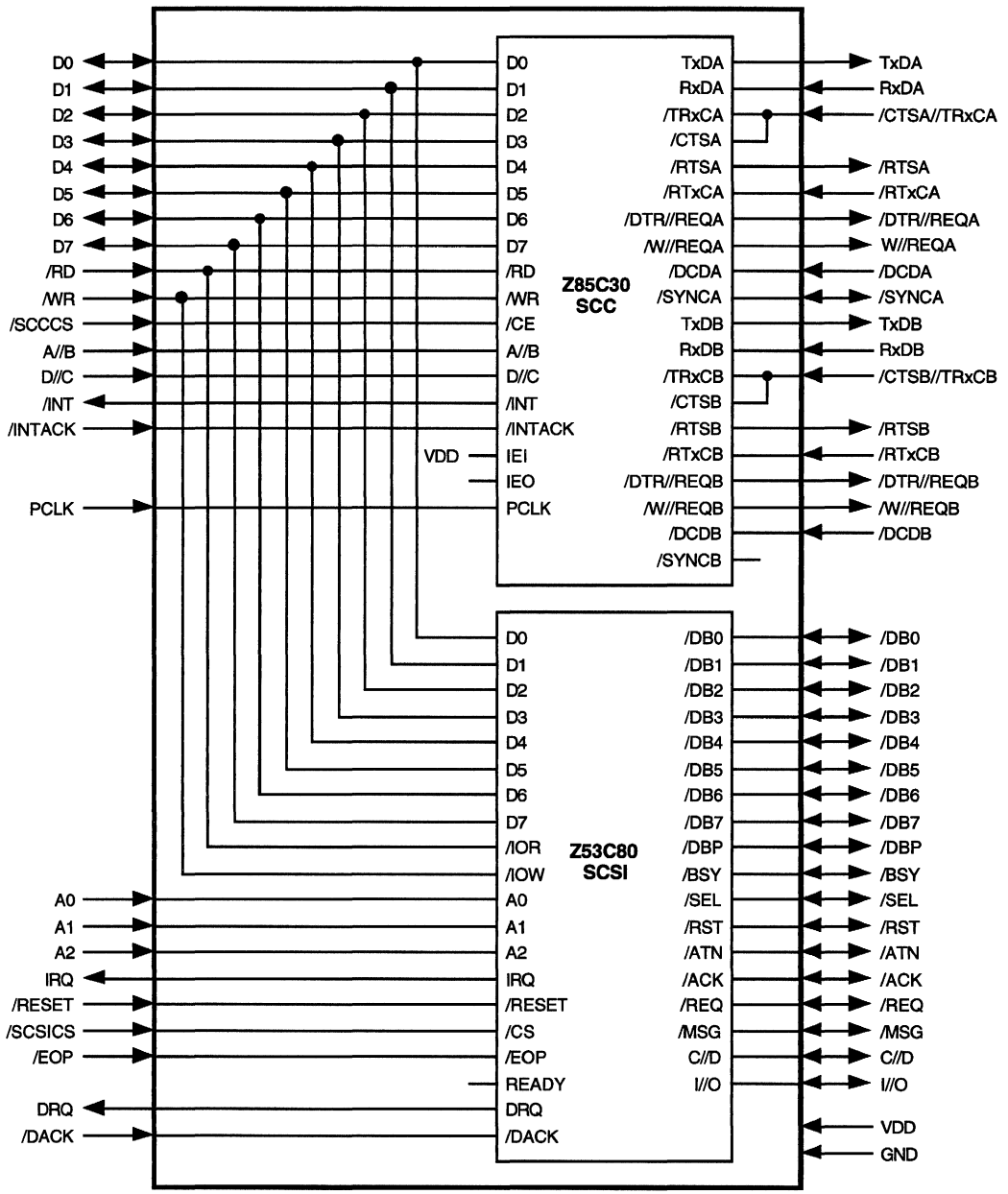


Figure 1. Z85C80 SCSCI Block Diagram

Note 1: Pins /CTSB and /TRxCB are bonded out separately in the 100-pin VQFP package and should be externally tied together by the user to meet the pin-out specification.

GENERAL DESCRIPTION (Continued)

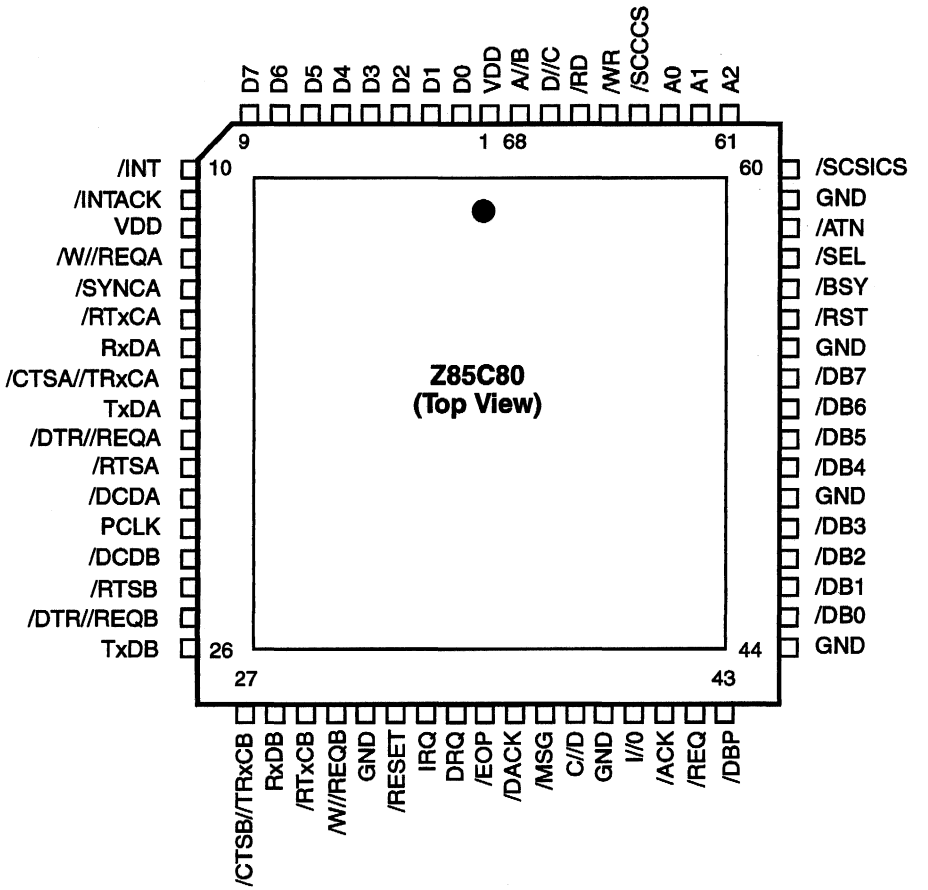


Figure 2. 68-Pin PLCC Pin Assignments

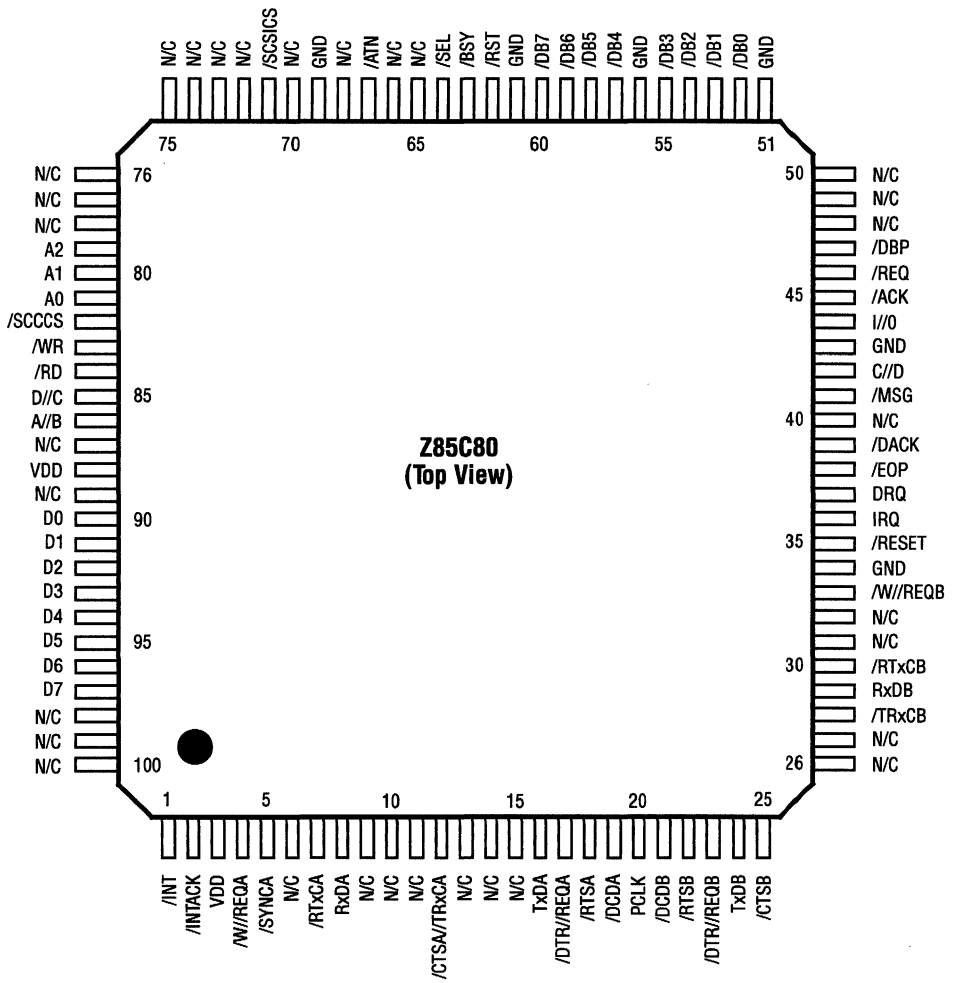


Figure 3. 100-Pin VQFP Pin Assignments

PIN DESCRIPTION

Signal	68-pin PLCC	100-pin VQFP	Type	Description
A0	63	81	I	**SCSI Address Line Bit 0** (SCSI). Address lines are used with /SCSICS, /RD, or /WR to address all internal registers.
A1	62	80	I	**SCSI Address Line Bit 1** (SCSI). Address lines are used with /SCSICS, /RD, or /WR to address all internal registers.
A2	61	79	I	**SCSI Address Line Bit 2** (SCSI). Address lines are used with /SCSICS, /RD, or /WR to address all internal registers.
A//B	68	86	I	**Channel A/Channel B** (SCC). This signal selects the SCC channel in which the read or write operation occurs.
/ACK	41	45	I/O	**Acknowledge** (open-drain, active Low, SCSI). Driven by an Initiator, /ACK indicates an acknowledgment for a /REQ//ACK data-transfer handshake. In the Target role, /ACK is received as a response to the /REQ signal.
/ATN	58	67	I/O	**Attention** (open-drain, active Low, SCSI). Driven by an Initiator, received by the Target. /ATN indicates an Attention condition.
/BSY	56	63	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	42	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	12	I	**Clear To Send for Channel A; Transmit/Receive Clock for Channel A** (active Low, SCC). This pin is internally connected to the SCC's A Channel /CTS and /TRXC. Receive clock or the transmit clock is supplied through 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	25/28	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 through 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	90	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	91	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	92	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	93	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	94	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.

Signal	68-pin PLCC	100-pin VQFP	Type	Description
D5	7	95	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	96	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	97	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	39	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	52	I/O	**SCSI Data Bus bit 0** (open-drain, active Low, SCSI). Least Significant Bit in the SCSI data bus.
/DB1	46	53	I/O	**SCSI Data Bus bit 1** (open-drain, active Low, SCSI).
/DB2	47	54	I/O	**SCSI Data Bus bit 2** (open-drain, active Low, SCSI).
/DB3	48	55	I/O	**SCSI Data Bus bit 3** (open-drain, active Low, SCSI).
/DB4	50	57	I/O	**SCSI Data Bus bit 4** (open-drain, active Low, SCSI).
/DB5	51	58	I/O	**SCSI Data Bus bit 5** (open-drain, active Low, SCSI).
/DB6	52	59	I/O	**SCSI Data Bus bit 6** (open-drain, active Low, SCSI).
/DB7	53	60	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	47	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	85	I	**Data/Control Select** (SCC). This signal defines the type of information transferred to and from the SCC.
/DCDA	21	19	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 a general-purpose input pin. The SCC detects pulses on this pin and can interrupt the CPU on both logic level transactions.
/DCDB	23	21	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	37	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	17	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 a general-purpose output or as a Request line for a DMA controller.

PIN DESCRIPTION (Continued)

Signal	68-pin PLCC	100-pin VQFP	Type	Description
/DTR//REQB	25	23	O	**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 a general-purpose output or as a Request line for a DMA controller.
/EOP	35	38	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	34, 43, 51 56, 61 69	S	**Ground supply** (SCC and SCSI).
/INT	10	1	O	**SCC Interrupt Request** (open-drain, active Low, SCC). This signal is activated when the SCC requests an interrupt.
/INTACK	11	2	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	44	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	36	O	**SCSI Interrupt Request** (active High, SCSI). This signal alerts the microprocessor of an error condition or an event completion.
/MSG	37	41	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	20	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	84	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	46	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	35	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	62	I/O	**SCSI bus Reset** (open-drain, active Low, SCSI). This signal indicates a SCSI bus Reset condition.

Signal	68-pin PLCC	100-pin VQFP	Type	Description
/RTSA	20	18	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	22	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.
/RTxCA	15	7	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	30	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	8	I	**Receive Data for Channel A** (active High, SCC). This input signal receives serial data.
RXDB	28	29	I	**Receive Data for Channel B** (active High, SCC). This input signal receives serial data.
/SCCCS	64	82	I	**SCC Chip Select** (active Low, SCSI). This signal selects SCC for a read or write operation.
/SCSICS	60	71	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	64	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	5	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	16	0	**Transmit Data for Channel A** (active High, SCC). This output signal transmits serial data at standard TTL levels.
TxDB	26	24	0	**Transmit Data for Channel B** (active High, SCC). This output signal transmits serial data at standard TTL levels.

PIN DESCRIPTION (Continued)

Signal	68-pin PLCC	100-pin VQFP	Type	Description
V_{DD}	1,12	3, 88	S	**V_{DD} supply** (SCC and SCSI).
/WR	65	83	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	4	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 a 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.
/W//REQB	30	33	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 a 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 V_{DD}. 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 used 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 datacommunications device, it transmits and receives data in a wide variety of datacommunications protocols; as a microprocessor peripheral, the SCC offers valuable features such as vectored interrupts, polling, and simple handshake capability.

Datacommunications Capabilities. The SCC provides two independent full-duplex channels programmable for use in any common Asynchronous or Synchronous datacommunication protocol. Figure 4 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.

FUNCTIONAL DESCRIPTION (Continued)

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.

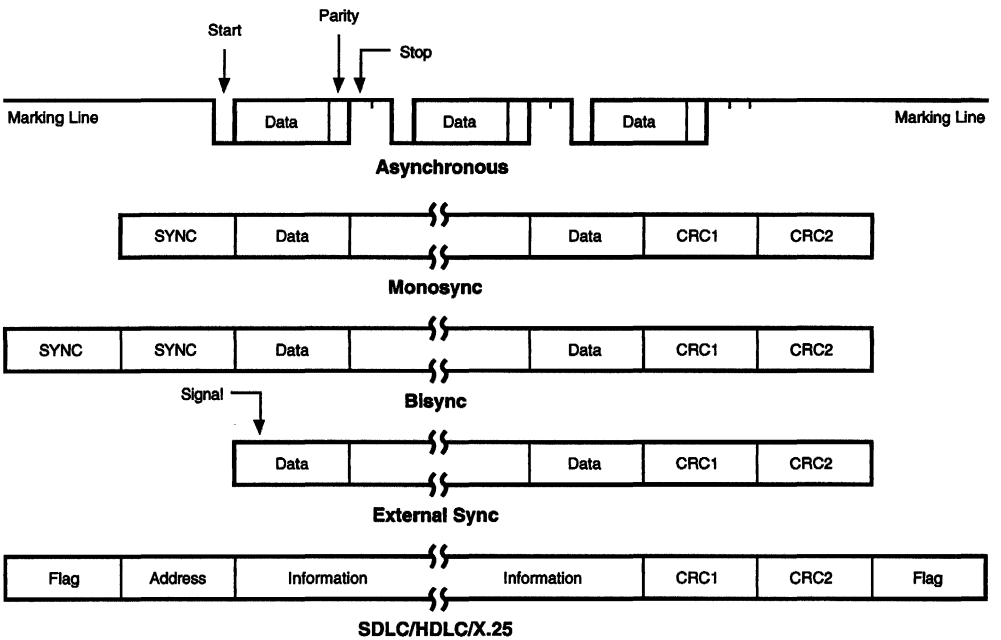


Figure 4. Some SCC Protocols

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, 16, 32, or 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 5.

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.

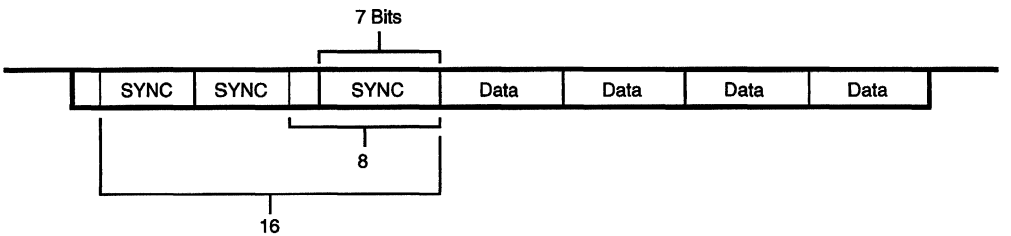


Figure 5. Detecting 5 - or 7-Bit Synchronous Characters

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 through 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 6).

6

FUNCTIONAL DESCRIPTION (Continued)

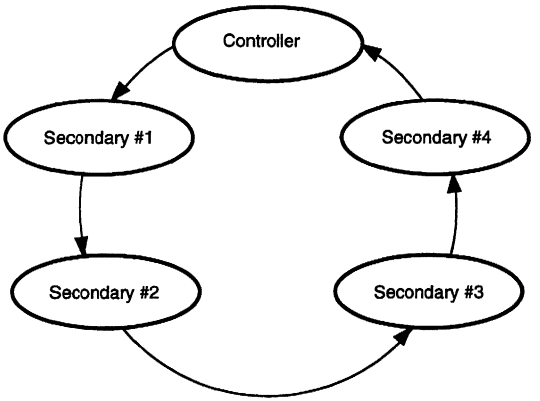


Figure 6. An SDLC Loop

A secondary station in an SDLC Loop is always listening to the messages being sent around the loop, and in fact 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 start-up, 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).

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 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 through 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 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 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.

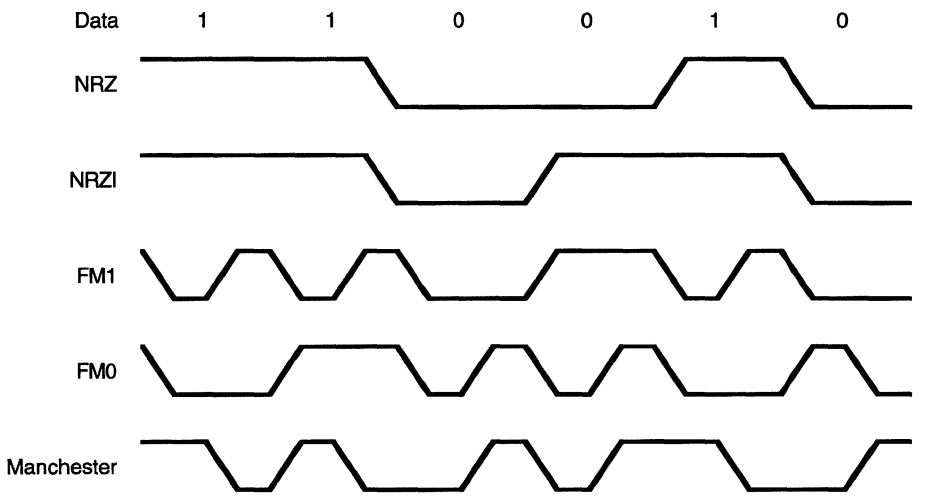


Figure 7. Data Encoding Methods

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 data 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 through 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,

FUNCTIONAL DESCRIPTION (Continued)

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 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.

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 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.

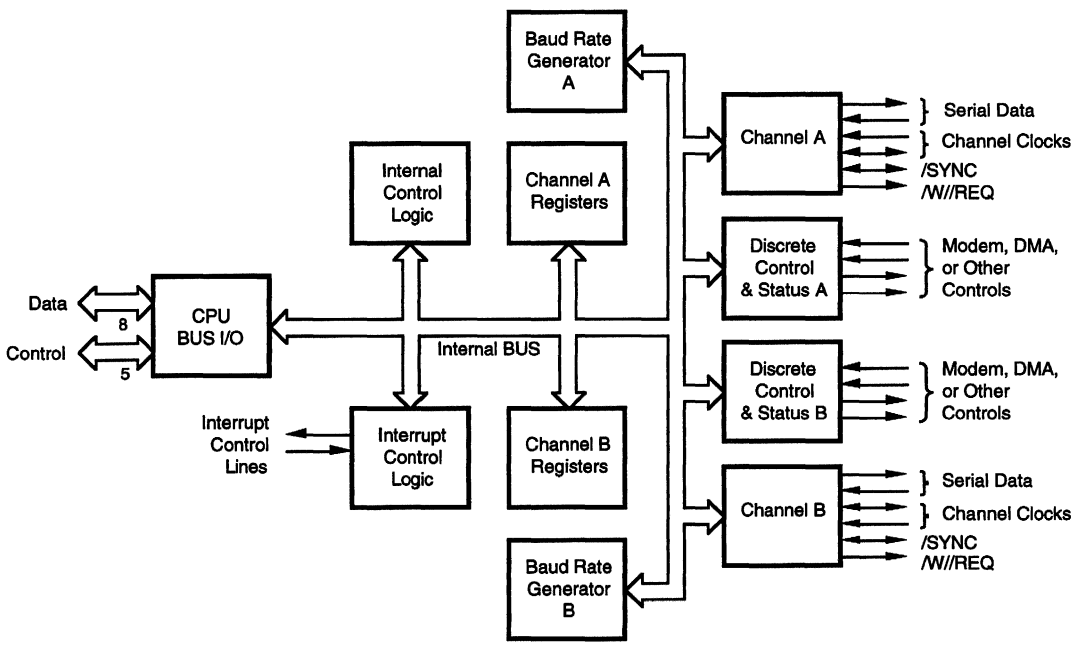


Figure 8. Block Diagram of SCC Architecture

ARCHITECTURE (Continued)

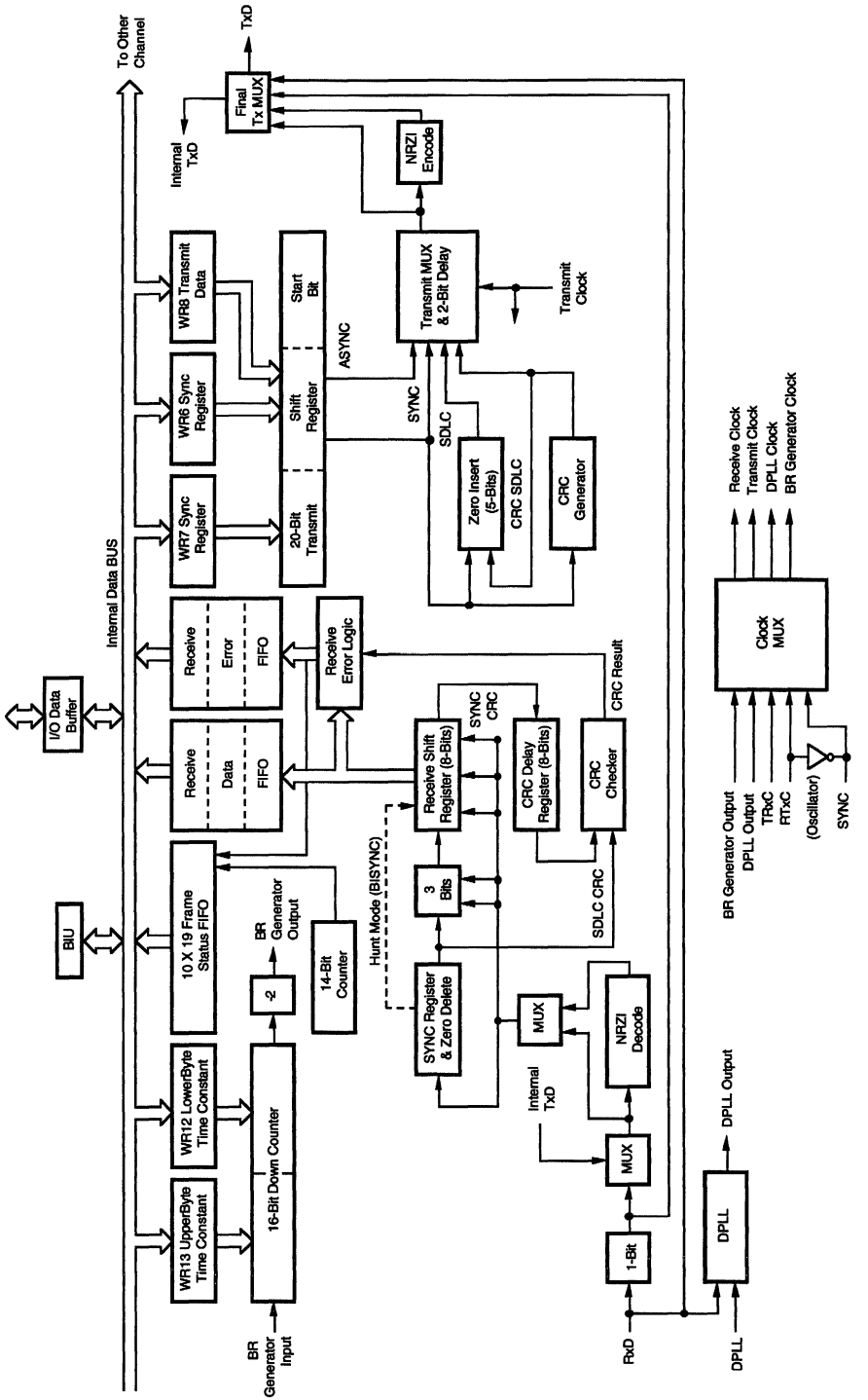


Figure 9. Data Path

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 9 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).

ARCHITECTURE (Continued)

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 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).

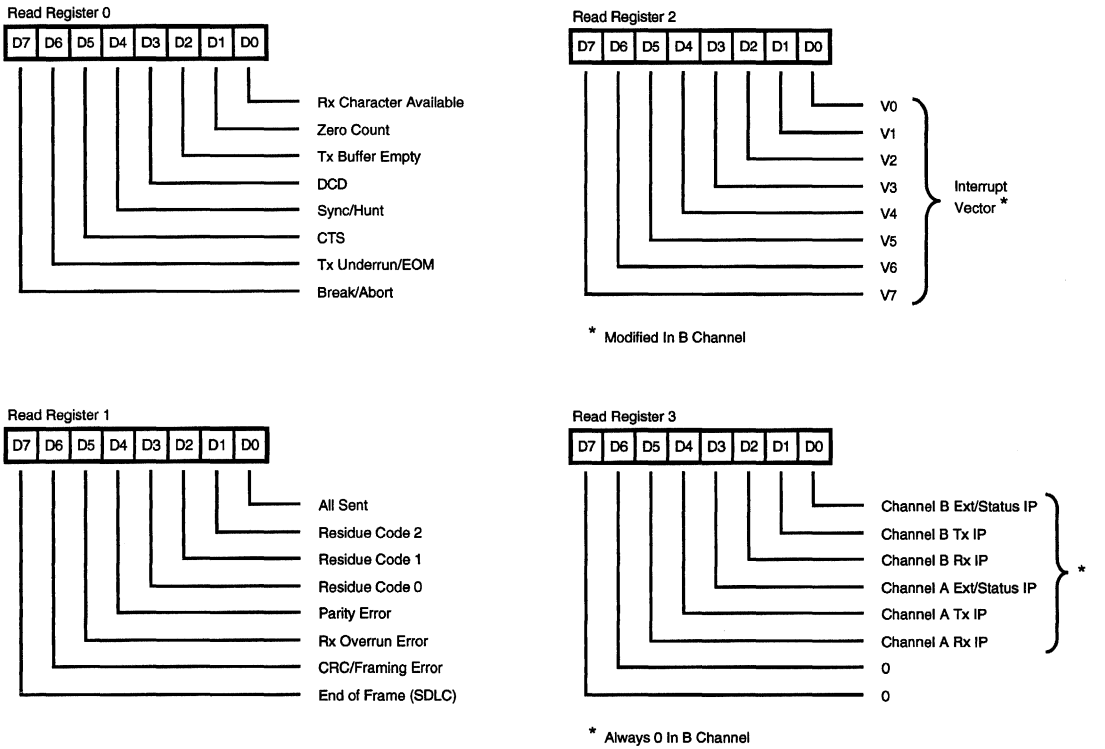


Figure 10. Read Register Bit Functions

PROGRAMMING (Continued)

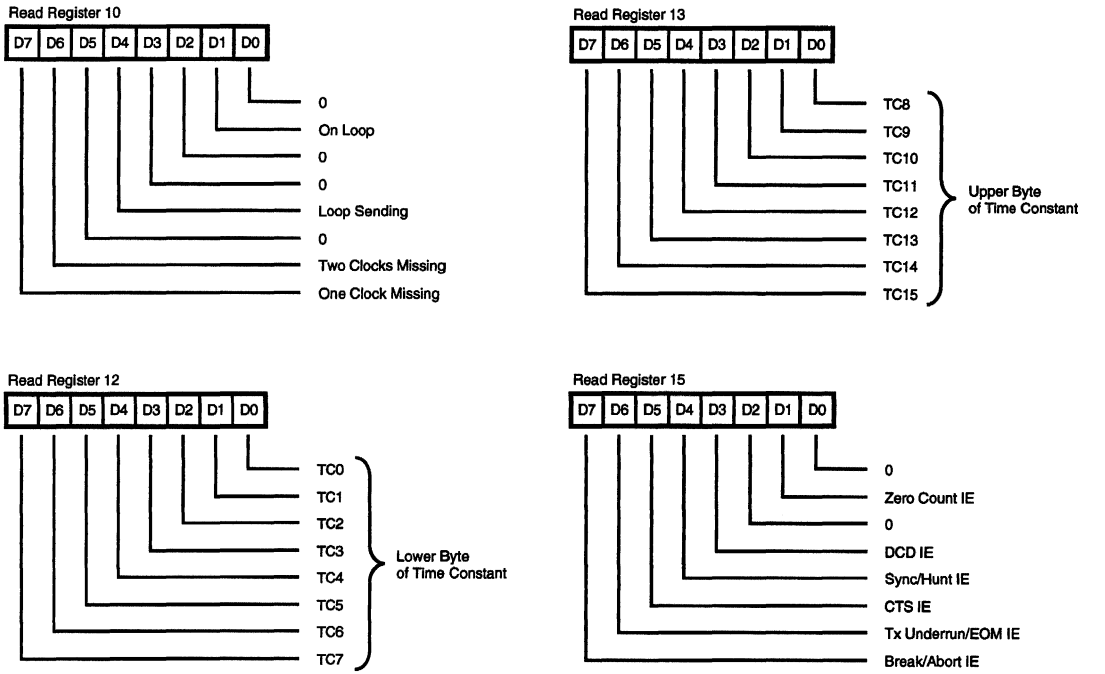


Figure 10. Read Register Bit Functions (Continued)

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 11 shows the format of each write register.

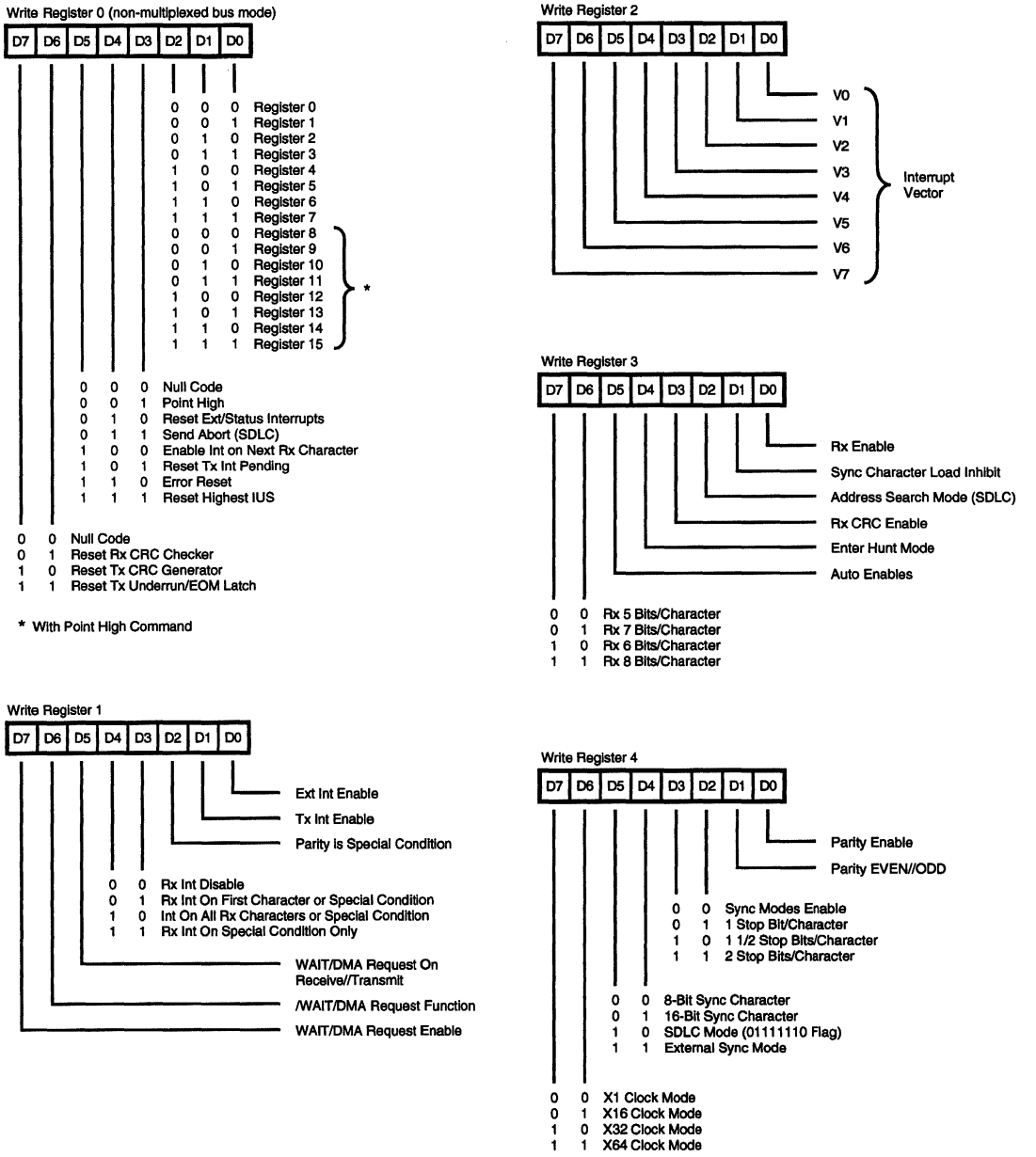
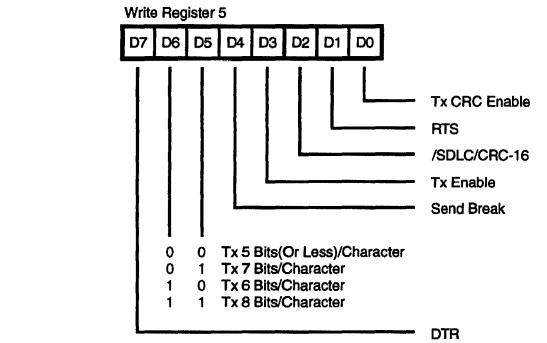
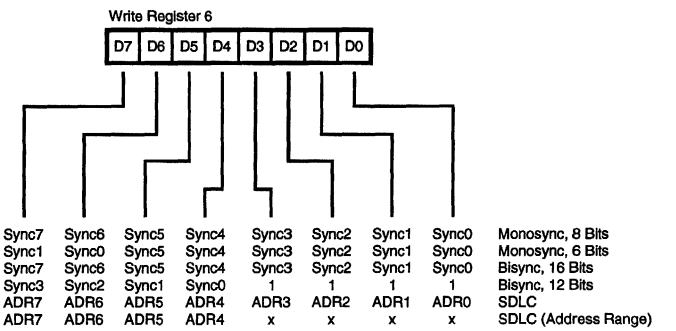


Figure 11. Write Register Bit Functions

6

PROGRAMMING (Continued)





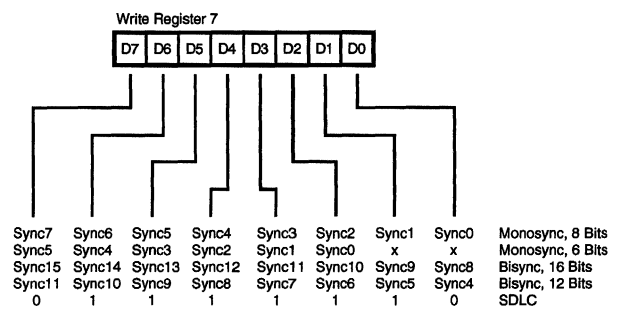


Figure 11. Write Register Bit Functions (Continued)

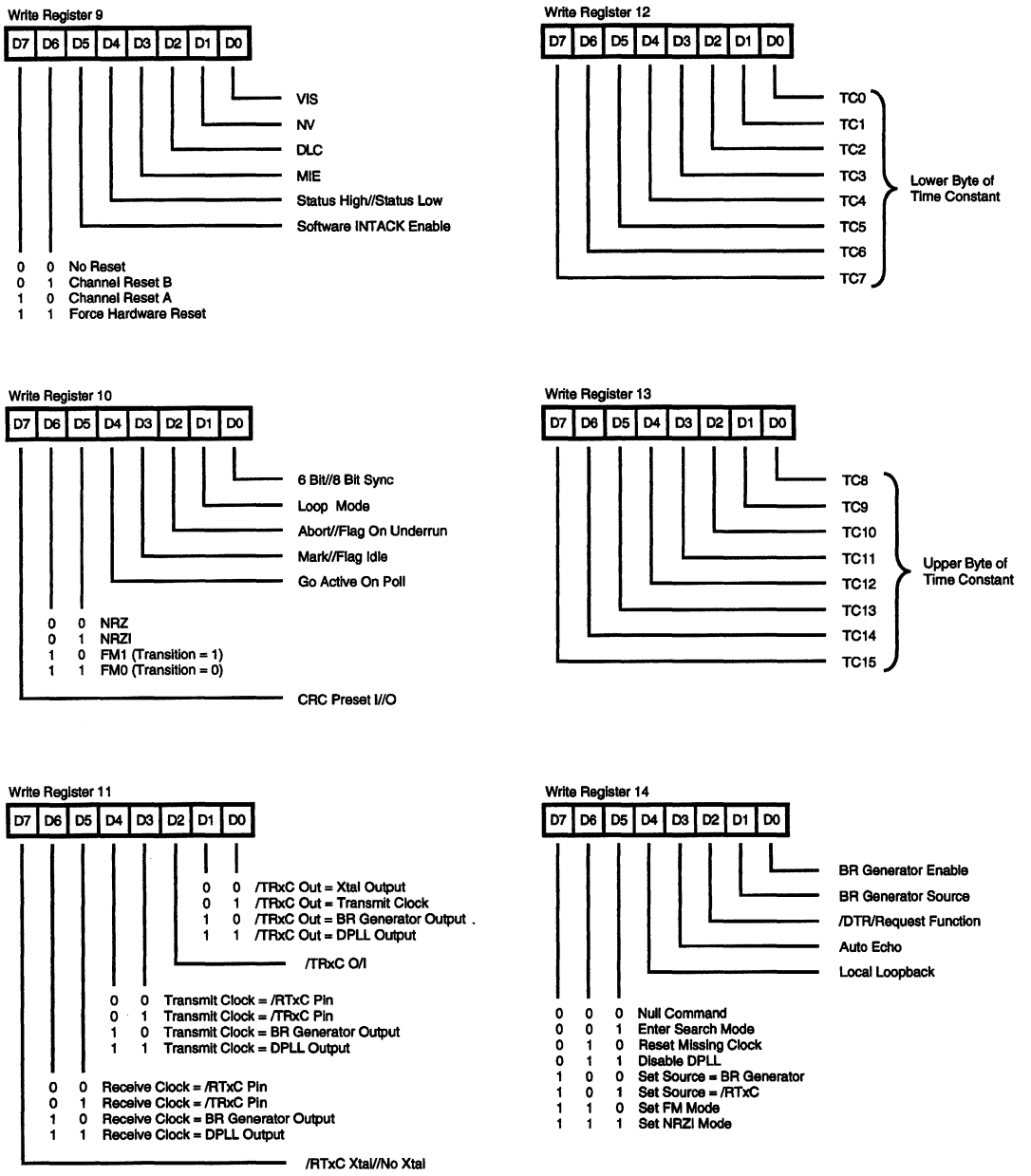


Figure 11. Write Register Bit Functions (Continued)

6

PROGRAMMING (Continued)

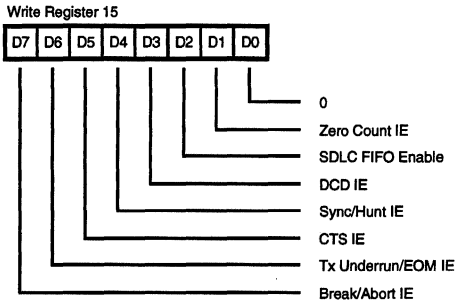


Figure 11. Write Register Bit Functions (Continued)

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 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 12 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.

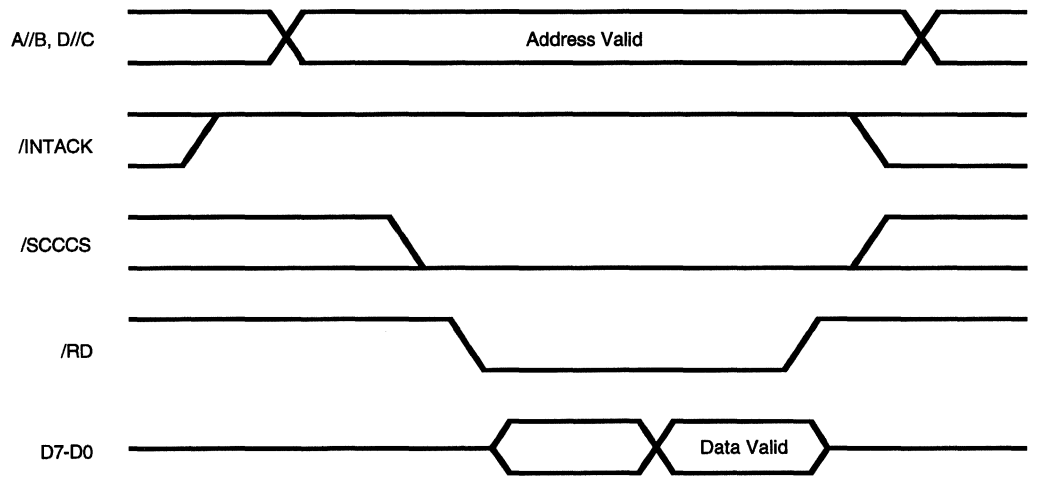


Figure 12. Read Cycle Timing

Write Cycle Timing. Figure 13 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.

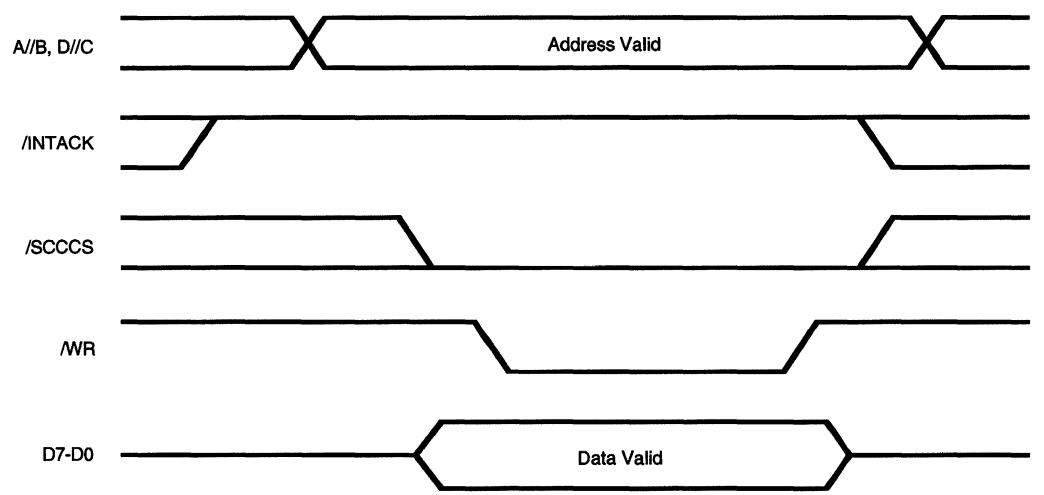


Figure 13. Write Cycle Timing

6

TIMING (Continued)

Interrupt Acknowledge Cycle Timing. Figure 14 illustrates Interrupt Acknowledge cycle timing.

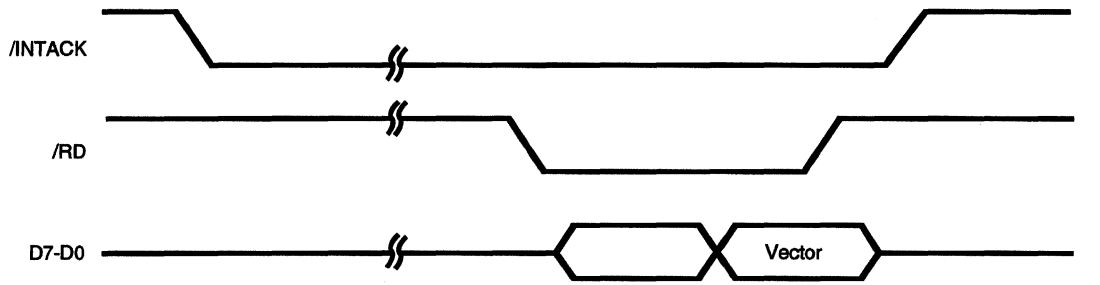


Figure 14. Interrupt Acknowledge Cycle Timing

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 15. The 10 x 19 bit status FIFO is separate from the existing 3-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 3-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 15.

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 through 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.

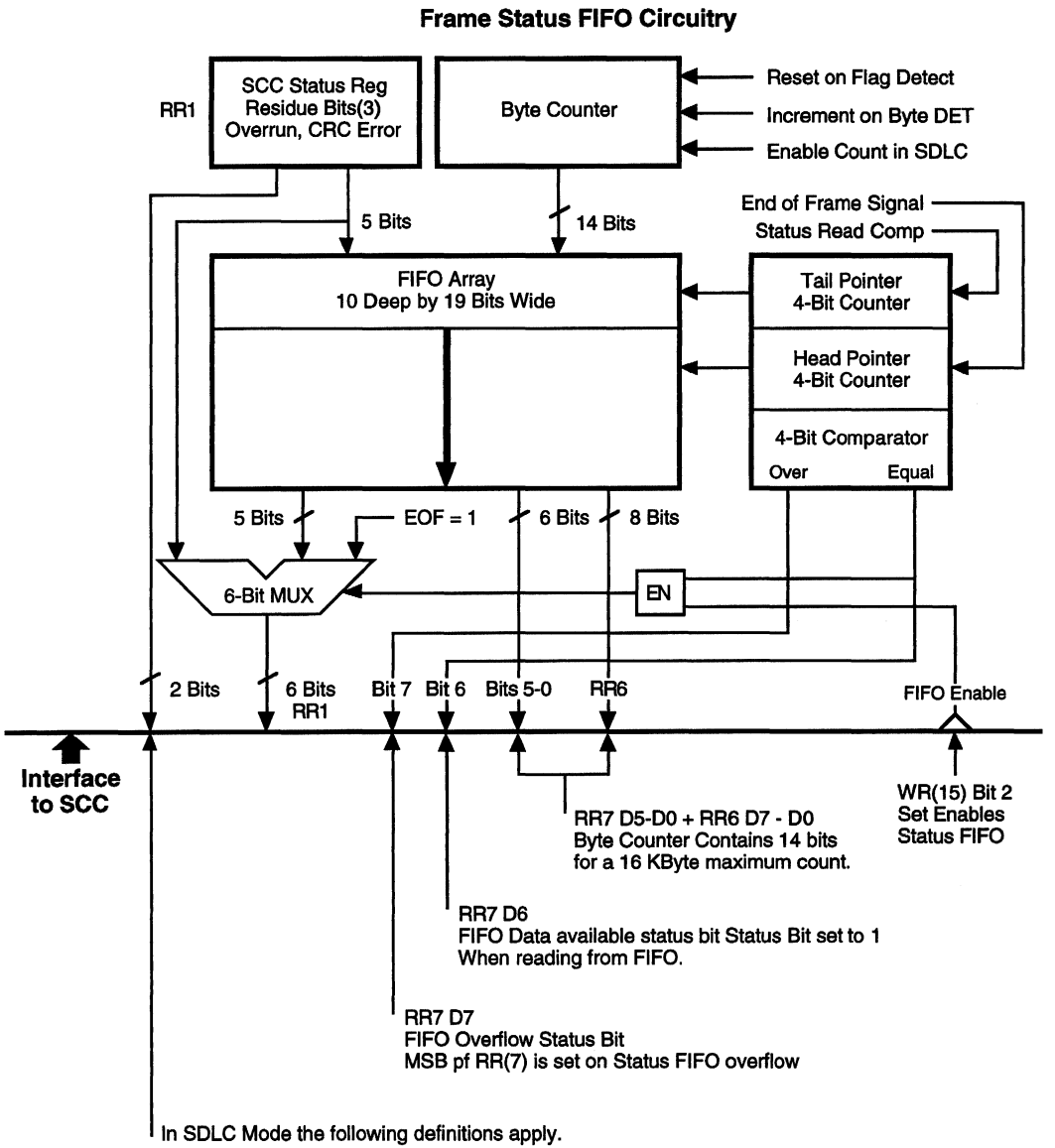


Figure 15. SCC Status Register Modifications

FIFO (Continued)

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 16.

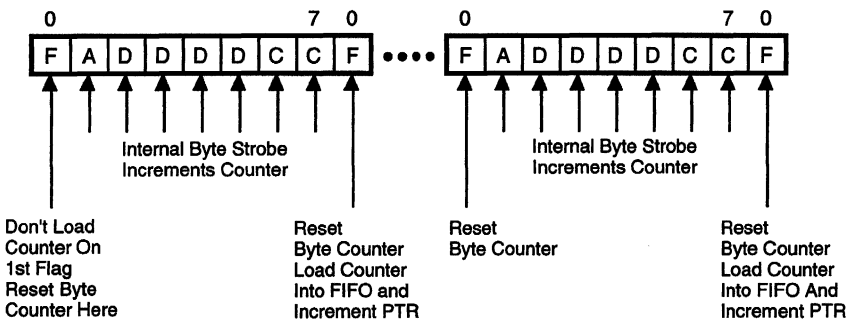


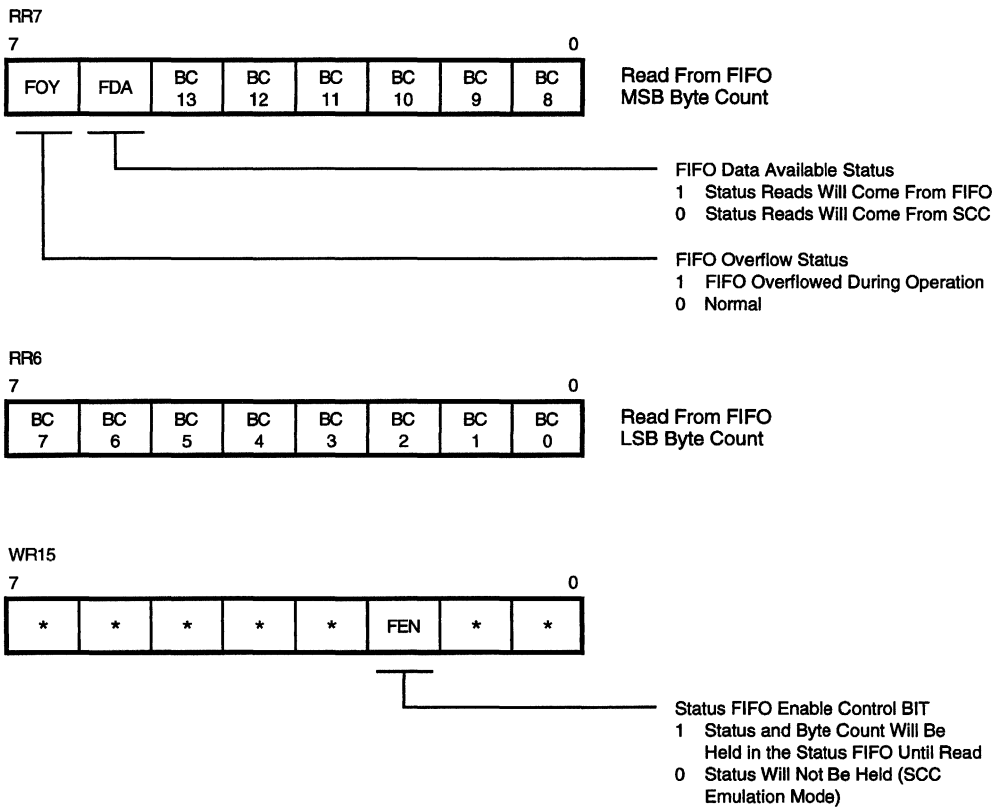
Figure 16. SDLC Byte Counting Detail

Byte Counter Detail. The 14-bit byte counter allows for packets up to 16 Kbytes to be received. For a better understanding of its operation refer to Figures 15 and 16.

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.)



* No change From NMOS SCC DFN

Figure 17. 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 issued, 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 V_{DD} through a resistor (10 Kohm typical).

6

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

Note: The 85C80 does not support or spec DMA Block Mode.

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 18) 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.

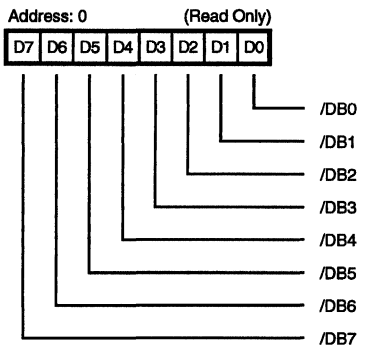


Figure 18. Current SCSI Data Register

Output Data Register *Address 0* (Write Only). The Output Data Register (Figure 19) is a write-only register that is used to send data to the SCSI Bus. This is accomplished by using either 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.

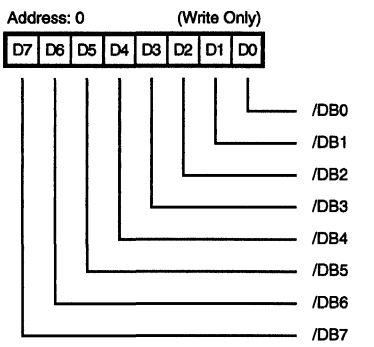


Figure 19. Output Data Register

Input Data Register *Address 6* (Read Only). The input Data Register (Figure 20) 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.

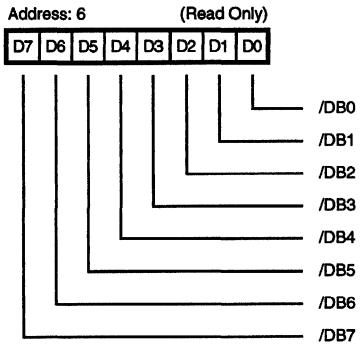


Figure 20. Input Data Register

Initiator Command Register *Address 1* (read/write). The Initiator Command Register (Figures 21 and 22) 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.

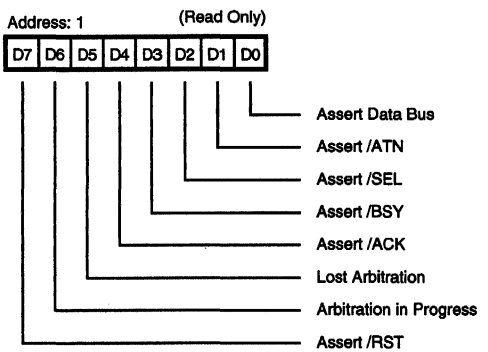


Figure 21. Initiator Command Register (Register Read)

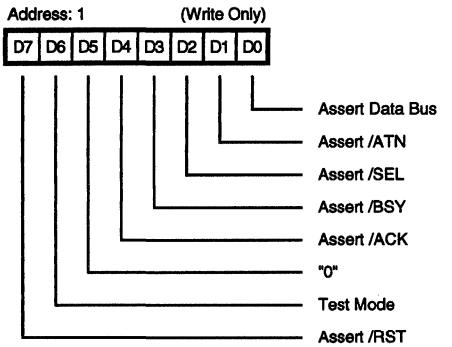


Figure 22. Initiator Command Register (Register Write)

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 Target Mode 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//O, 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 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 0. A read on this register simply reflects the status of this bit.

Bit 2 *Assert/SEL.* Writing a 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 0. A read of this register reflects the status of this bit.

SCSI FUNCTIONAL DESCRIPTION (Continued)

Bit 3 *Assert/BSY.* Writing a 1 into this bit position asserts /BSY onto the SCSI Bus. Conversely, a 0 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 0 to this bit deasserts /ACK. Reading this register reflects bit status.

Bit 5 *0* (Write Bit). Bit 5 should be written with a 0 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 1 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 0 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 23).

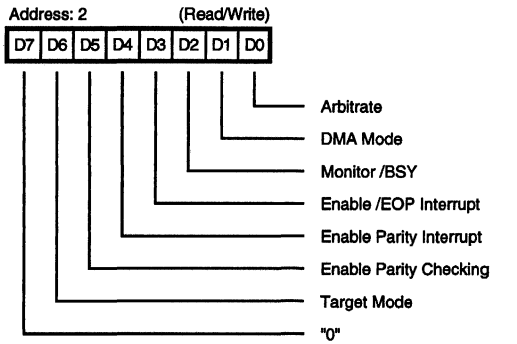


Figure 23. Mode Register

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 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 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 0 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.* 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 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 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 *0.* Bit 7 should be written with a 0 for proper operation.

Target Command Register *Address 3* (Read/Write). When connected as a target device, the Target Command Register (Figure 24) 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 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.

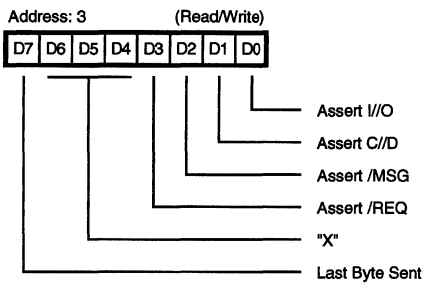


Figure 24. Target Command Register

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 25 describes the Current SCSI Bus Status Register.

Select Enable Register *Address 4* (Write Only). The Select Enable Register (Figure 26) 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.

6

SCSI FUNCTIONAL DESCRIPTION (Continued)

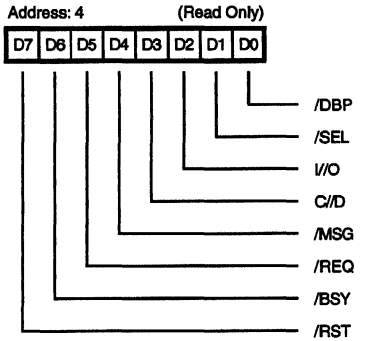


Figure 25. Current SCSI Bus Status Register

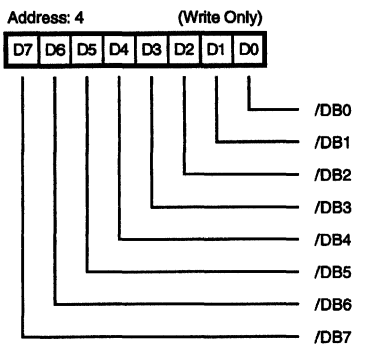


Figure 26. Select Enable Register

Bus and Status Register *Address 5* (Read Only). The Bus and Status Register (Figure 27) 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.

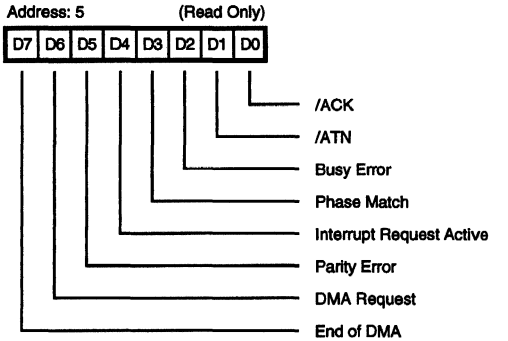


Figure 27. Bus and Status Register

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 100 ns. 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 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 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 400 ns, 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 27 and 25) 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 100 ns.

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.

SCSI FUNCTIONAL DESCRIPTION (Continued)

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

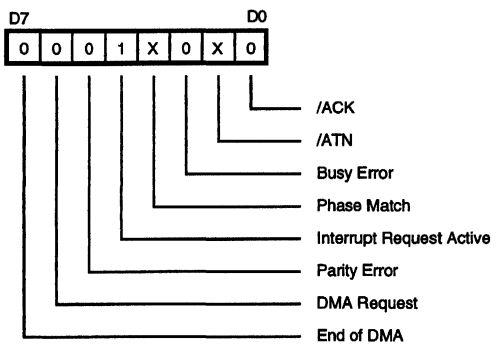


Figure 28. Bus and Status Register

The proper values for the Bus and Status Register and the Current SCSI Bus Status Register are displayed in Figures 28 and 29, respectively.

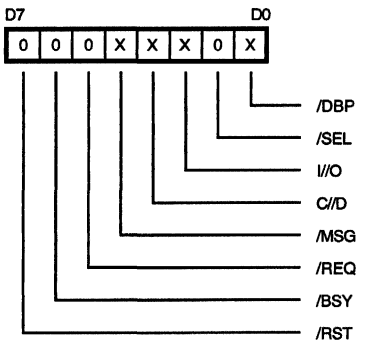


Figure 29. Current SCSI Bus Status Register

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 30 and 31.

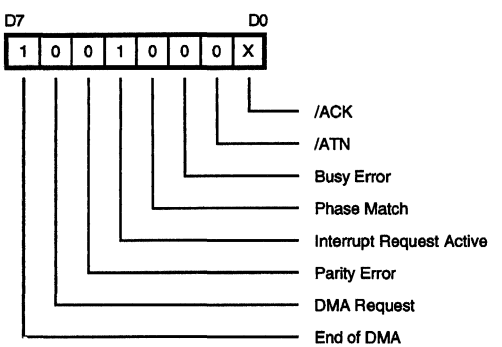


Figure 30. Bus and Status Register

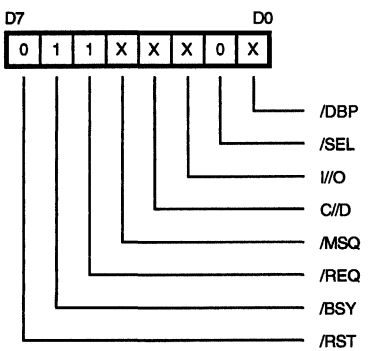


Figure 31. Current SCSI Bus Status Register

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 32 and 33, respectively.

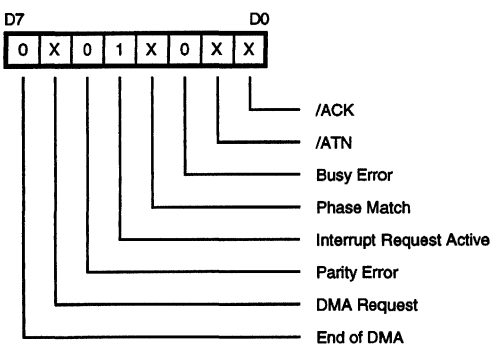


Figure 32. Bus and Status Register

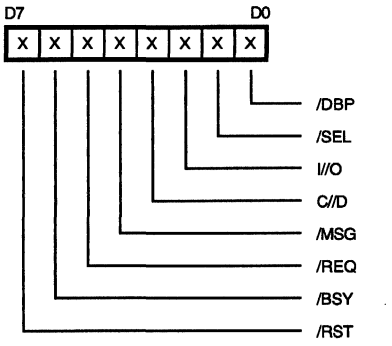


Figure 33. Current SCSI Bus Status Register

Parity Error. 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 34 and 35, respectively.

SCSI FUNCTIONAL DESCRIPTION (Continued)

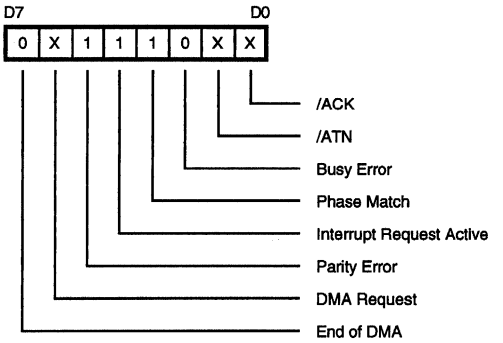


Figure 34. Bus and Status Register

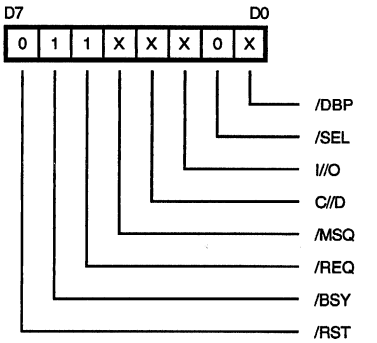


Figure 35. Current SCSI Bus Status Register

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 36 and 37, respectively.

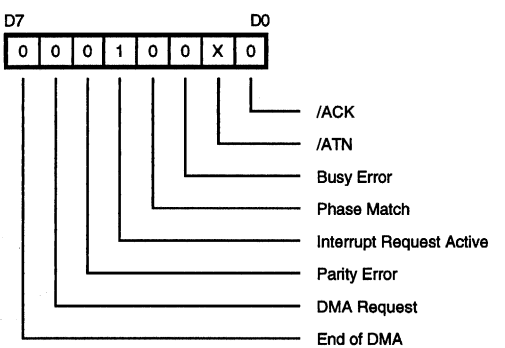


Figure 36. Bus and Status Register

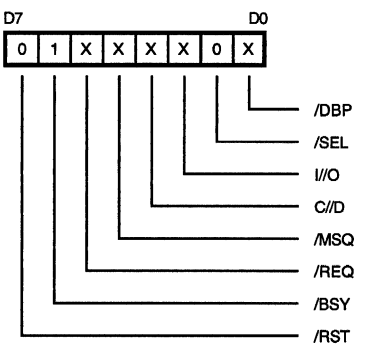


Figure 37. Current SCSI Bus Status Register

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 38 and 39.

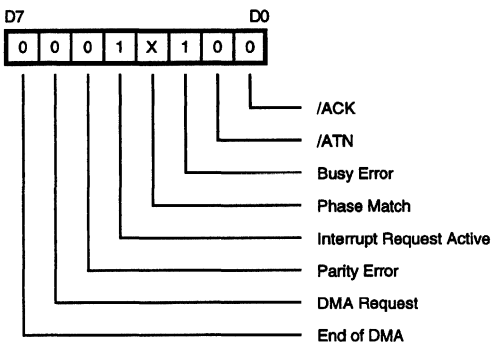


Figure 38. Bus and Status Register

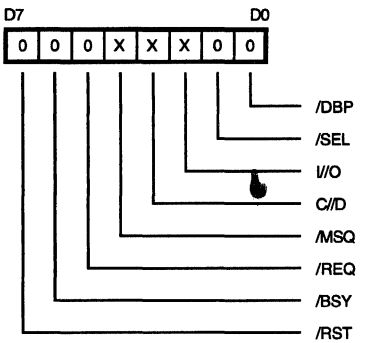


Figure 39. Current SCSI Bus Status Register

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, or 3) 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.

6

SCSI FUNCTIONAL DESCRIPTION (Continued)

Pseudo DMA Mode. To avoid the tedium of monitoring and asserting the request/acknowledgment 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.

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

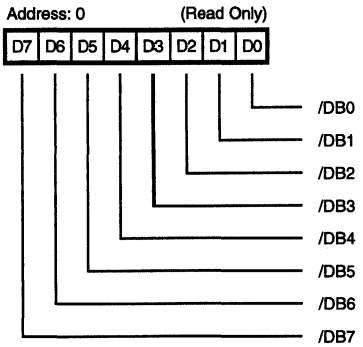


Figure 40. Current SCSI Data Register

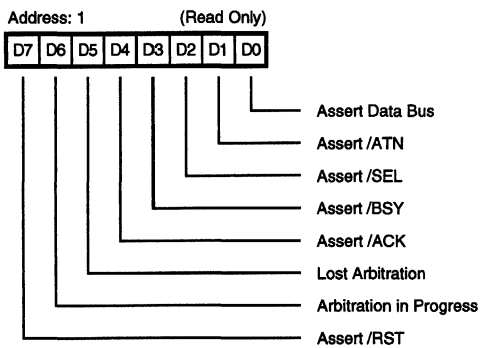


Figure 41. Initiator Command Register

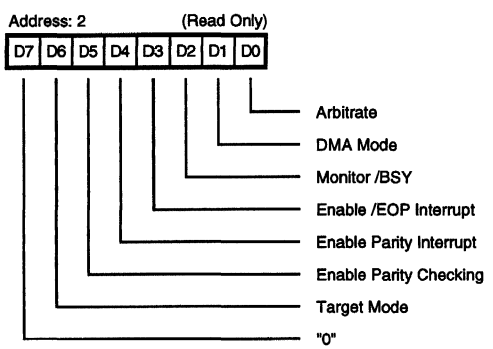


Figure 42. Mode Register

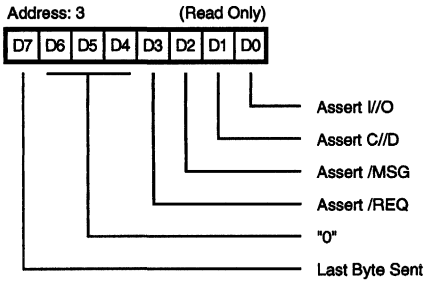


Figure 43. Target Command Register

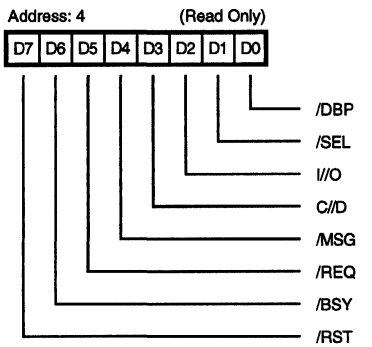


Figure 44. Current SCSI Bus Status Register

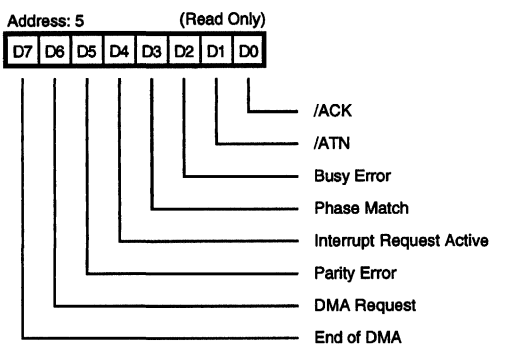


Figure 45. Bus and Status Register

6

READ REGISTERS (Continued)

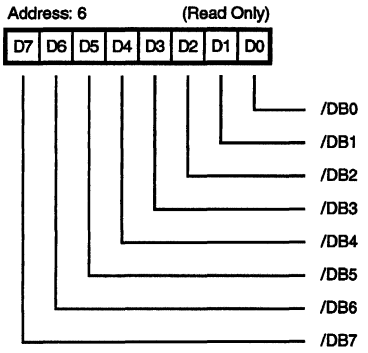


Figure 46. Input Data Register

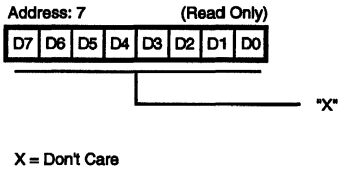


Figure 47. Reset Parity/Interrupt

WRITE REGISTERS

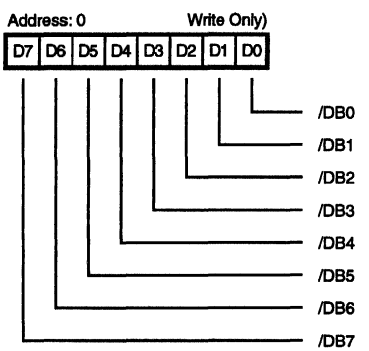


Figure 48. Output Data Register

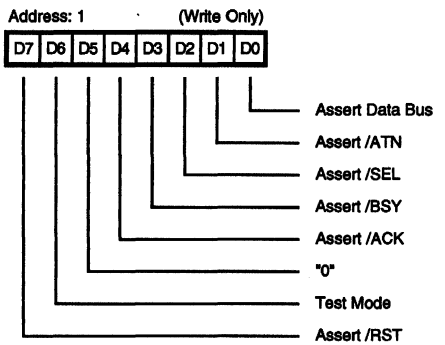


Figure 49. Initiator Command Register

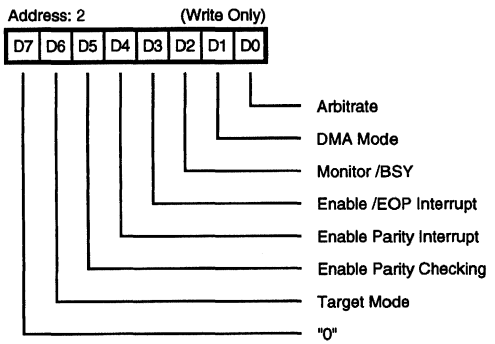


Figure 50. Mode Register

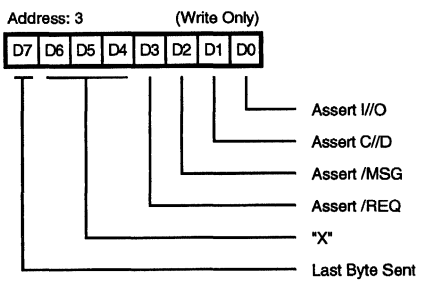


Figure 51. Target Command Register

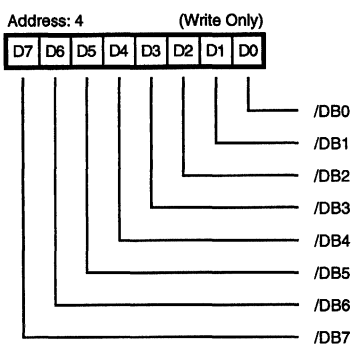


Figure 52. Select Enable Register

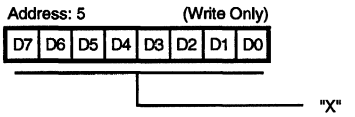


Figure 53. Start DMA Send

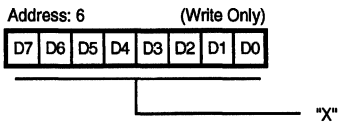


Figure 54. Start DMA Target Receive

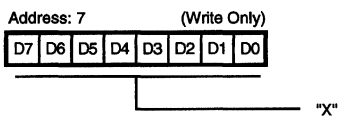


Figure 55. Start DMA Initiator Receive

Note: X = Don't care

6

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:

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

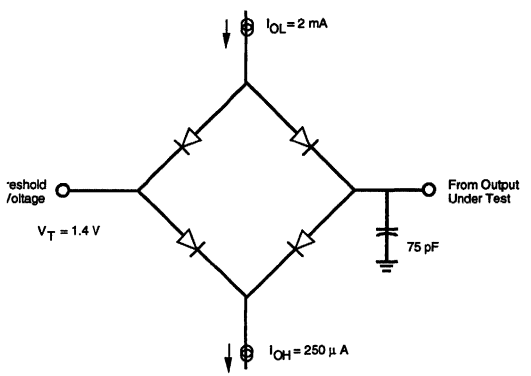


Figure 56. Standard Test Dynamic Load Circuit

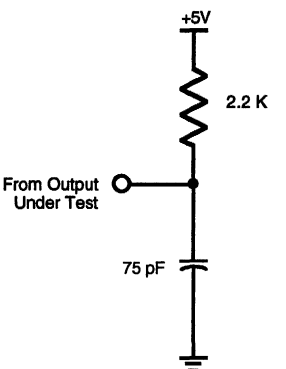


Figure 57. Open-Drain Test Load

DC CHARACTERISTICS

Symbol	Parameter	Condition	Min	Max	Units
V_{DD}	Supply Voltage		4.75	5.25	V
V_{IH}	High-Level Input Voltage		2.0	5.5	V
V_{IL}	Low-Level Input Voltage		–0.3	0.8	V
I_{IH1}	High-Level Input Current	V_{IH} = 5.25V			
	SCSI Bus Pins	V_{IL} = 0V		50	µA
I_{IH2}	High-Level Input Current	V_{IH} = 5.25V			
	All Other Pins	V_{IL} = 0V		10	µA
I_{IL1}	Low-Level Input Current	V_{IH} = 5.25V			
	SCSI Bus Pins	V_{IL} = 0V		–50	µA
I_{IL2}	Low-Level Input Current	V_{IH} = 5.25V			
	All Other Pins	V_{IL} = 0V		–10	µA
V_{OH1}	High-Level Output Voltage	I_{OH} = –3 mA	2.4		
V_{OH2}	High-Level Output Voltage	I_{OH} = –250 µA	V_{DD} –0.8	V	
V_{OL1}	Low-Level Output Voltage				
	SCSI Bus Pins	I_{OL} = 48 mA		0.5	V
V_{OL2}	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

6

AC CHARACTERISTICS
General Timing

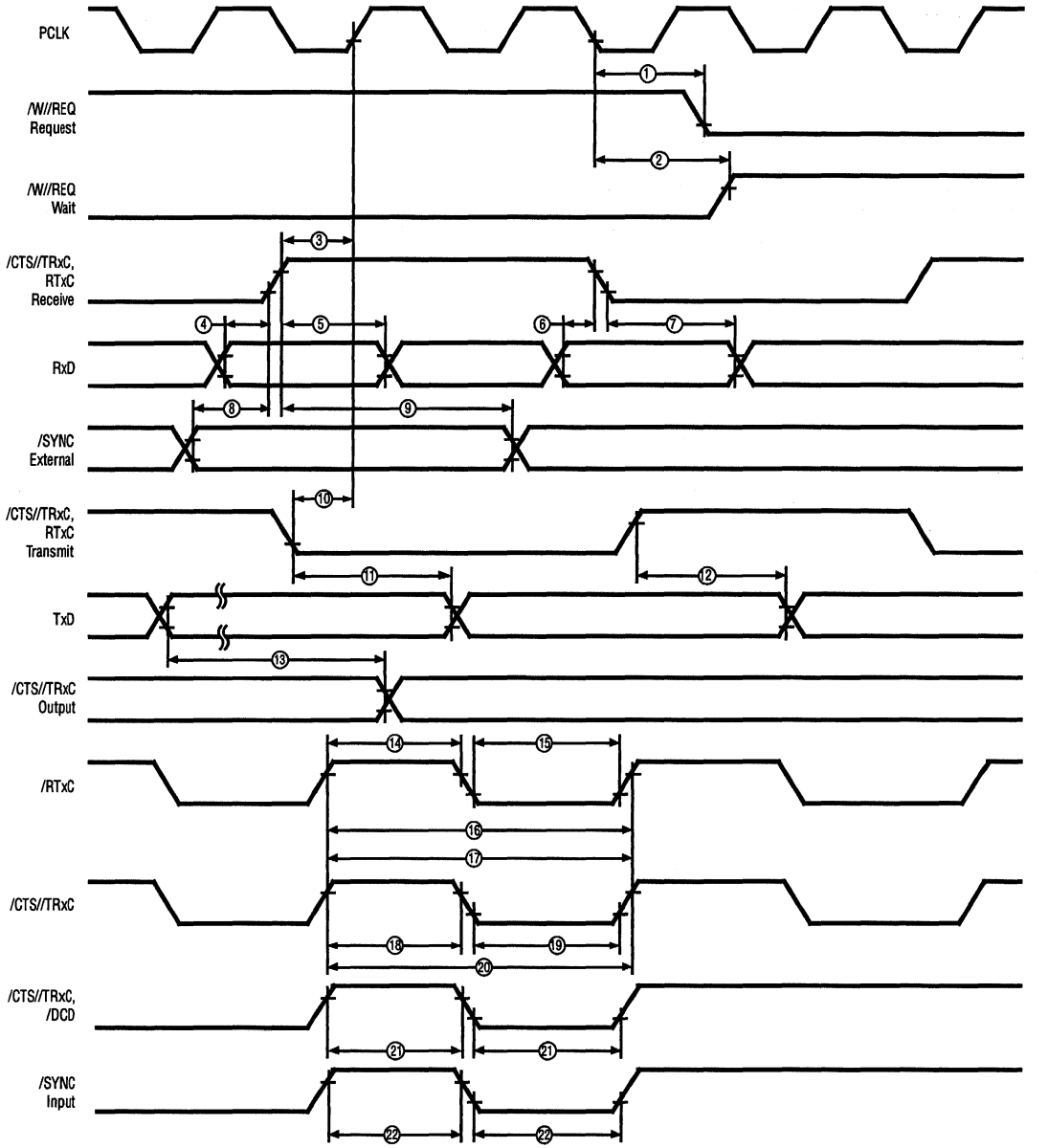


Figure 58. General Timing

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

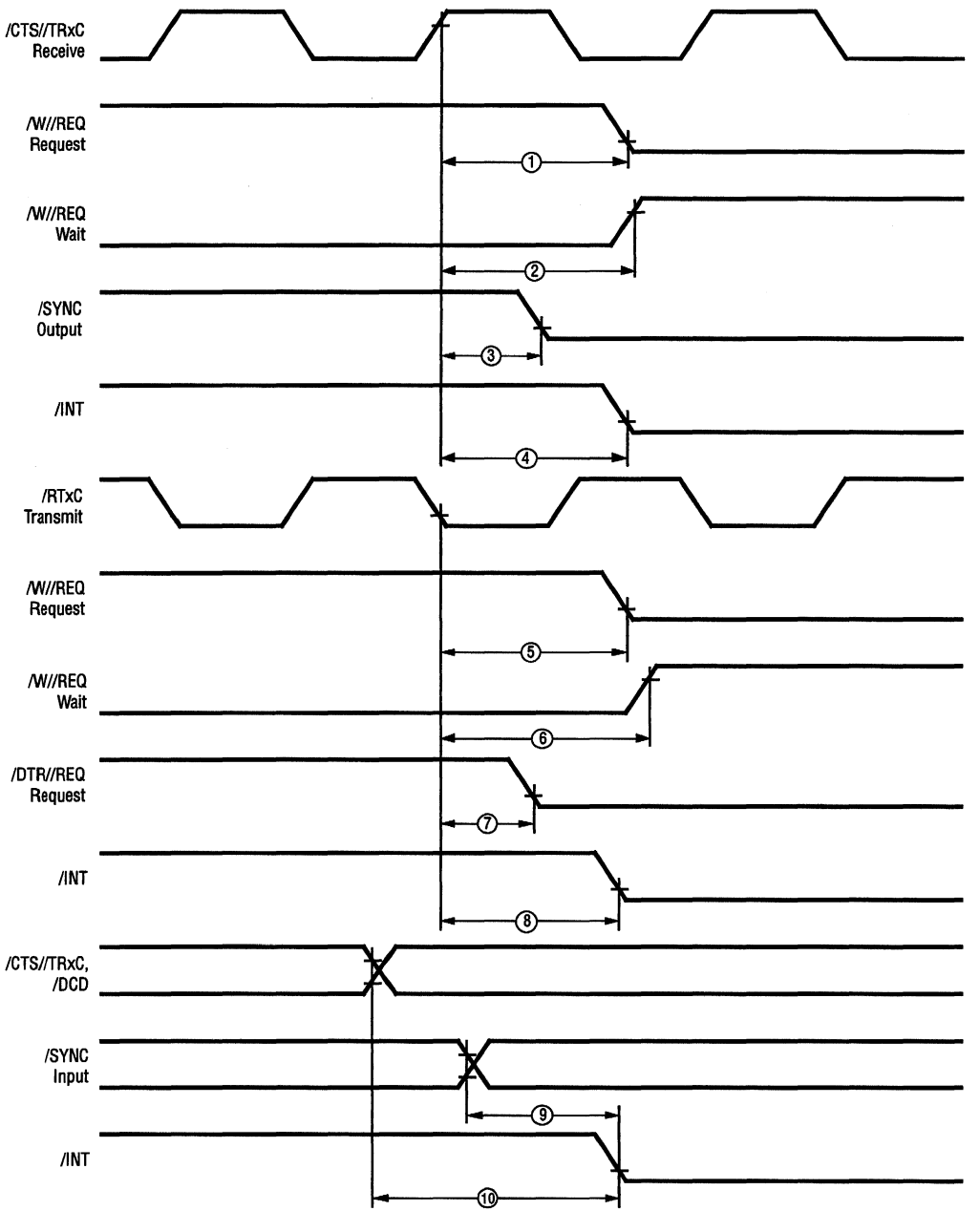


Figure 59. System Timing

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

AC CHARACTERISTICS
Z85C80 Additional Timing

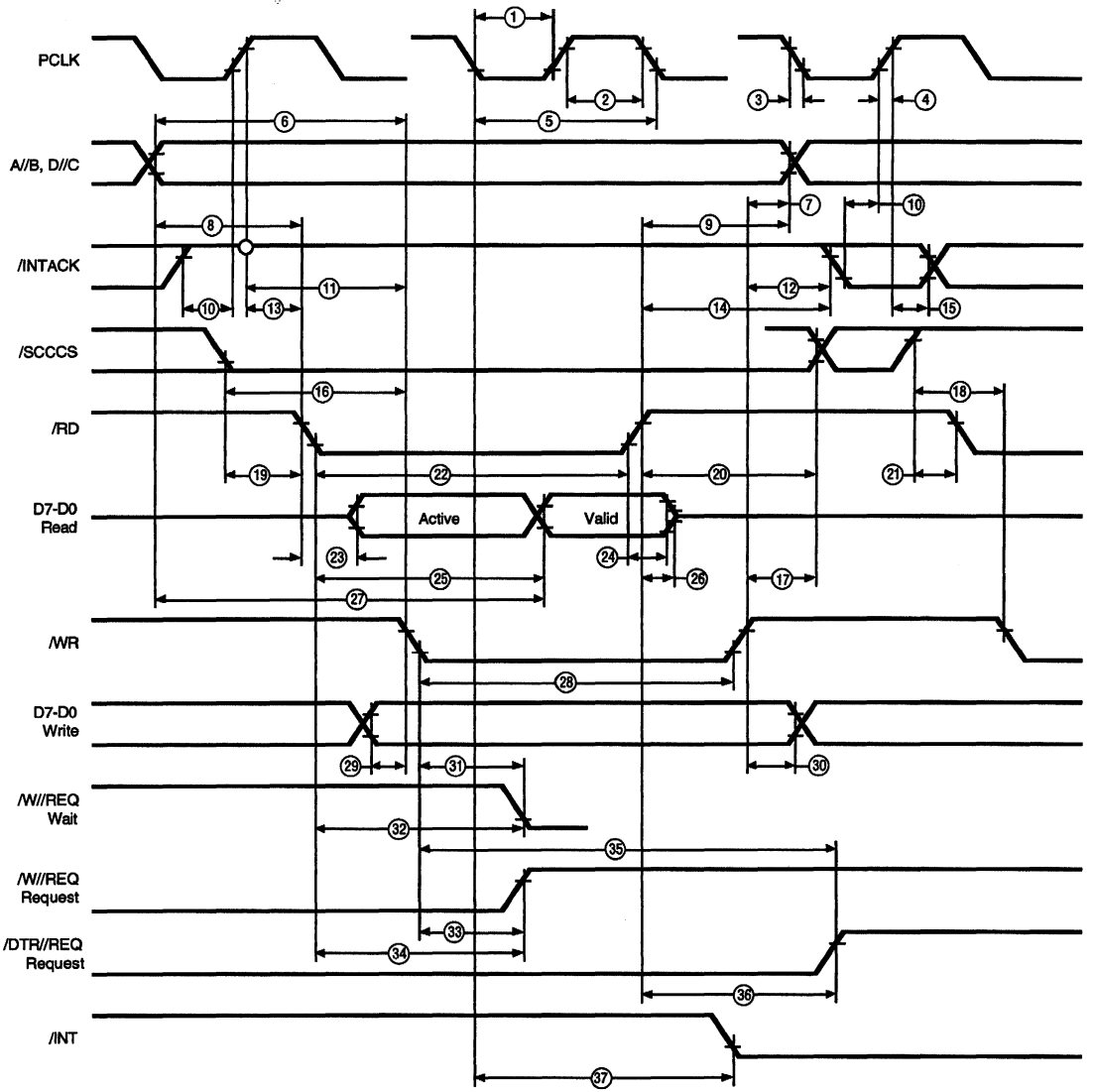


Figure 60. Read/Write Timing

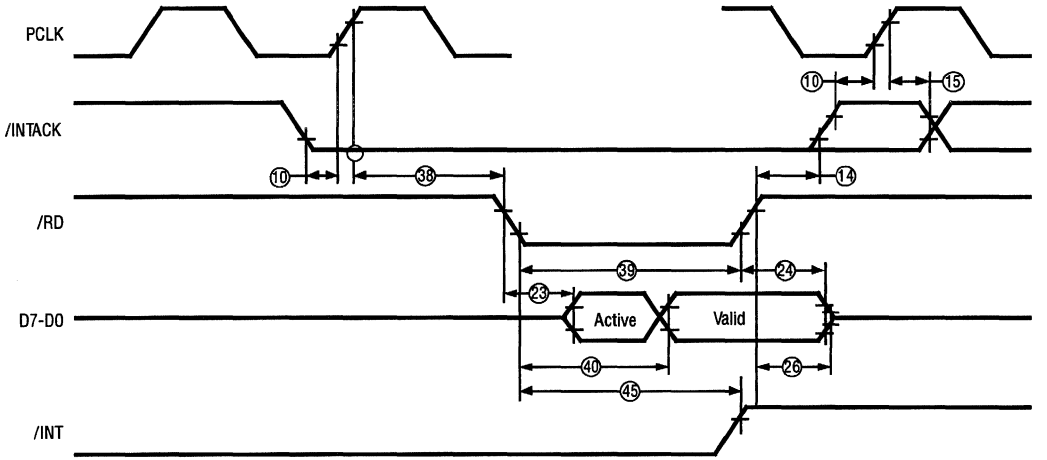


Figure 61. Interrupt Acknowledge Timing

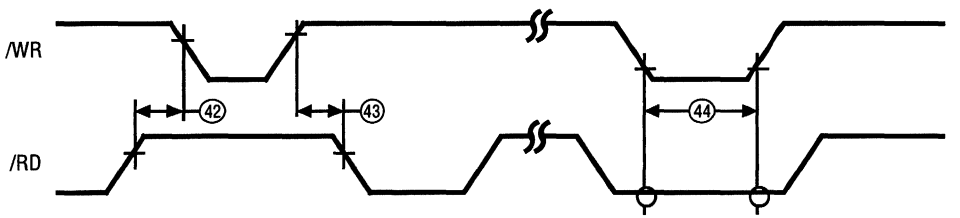


Figure 62. Reset Timing

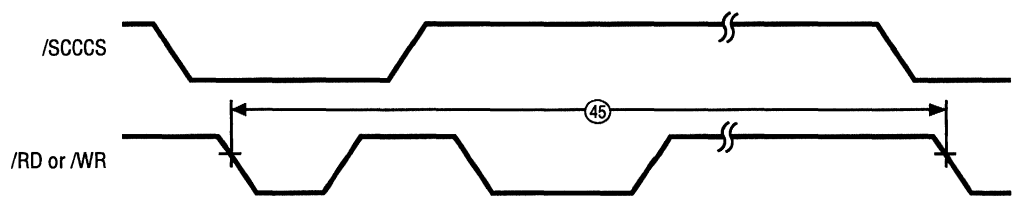


Figure 63. Cycle Timing

6

AC CHARACTERISTICS
Additional Timing

No	Symbol	Parameter	10 MHz Min	10 MHz Max	16 MHz Min	16 MHz Max	Notes †
1	TwPCl	PCLK Low Width	40	1000	26	1000	
2	TwPCh	PCLK High Width	40	1000	26	1000	
3	TfPC	PCLK Fall Time		10		5	
4	TrPC	PCLK Rise Time		10		5	
5	TcPC	PCLK Cycle Time	100	2000	61	2000	
6	TsA(WR)	Address to /WR Fall Setup Time	50		35		
7	ThA(WR)	Address to /WR Rise Hold Time	0		0		
8	TsA(RD	Address to /RD Fall Setup Time	50		35		
9	ThA(RD)	Address to /RD Rise Hold Time	0		0		
10	TslA(PC)	/INTACK to PCLK Rise Setup Time	20		15		
11	TslAi(WR)	/INTACK to /WR Fall Setup Time	130		75		[1]
12	ThlA(WR)	/INTACK to /WR Rise Hold Time	0		0		
13	TslAi(RD)	/INTACK to /WR Fall Setup Time	130		75		[1]
14	ThlA(RD)	/INTACK to /RD Rise Hold Time	0		0		
15	ThlA(PC)	/INTACK to PCLK Rise Hold Time	30		15		
16	TsCEl(WR)	/SCCCS Low to /WR Fall Setup Time	0		0		
17	ThCE(WR)	/SCCCS to /WR Rise Hold Time	0		0		
18	TsCEh(WR)	/SCCCS High to /WR Fall Setup Time	50		30		
19	TsCEl(RD)	/SCCCS Low to /RD Fall Setup Time	0		0		[1]
20	ThCE(RD)	/SCCCS to /RD Rise Hold Time	0		0		[1]
21	TsCEh(RD)	/SCCCS High to /RD Fall Setup Time	50		30		[1]
22	TwRDl	/RD Low Width	125		70		[1]
23	TdRD(DRA)	/RD Fall to Read Data Active Delay	0		0		
24	TdRDr(DR)	/RD Rise to Read Data Not Valid Delay	0		0		
25	TdRDl(DR)	/RD Fall to Read Data Valid Delay		120		65	
26	TdRD(DRz)	/RD Rise to Read Data Float Delay		35		20	
27	TdA(DR)	Address to Read Data Valid Delay		180		100	
28	TwWRl	/WR Low Width	125		70		
29	TsDW(WR)	Write Data to /WR Fall Setup Time	10		10		
30	ThDW(WR)	Write Data to /WR Rise Hold Time	0		0		
31	TdWR(W)	/WR Fall to Wait Valid Delay		160		80	[2]
32	TdRD(W)	/RD Fall to Wait Valid Delay		160		80	[2]
33	TdWRf(REQ)	/WR Fall to /W//REQ Not Valid Delay		160		80	
34	TdRDf(REQ)	/RD Fall to /W//REQ Not Valid Delay		160		80	
35	TdWRr(REQ)	/WR Fall /DTR//REQ Not Valid Delay		4TcPC		4TcPc	
36	TdRDr(REQ)	/RD Rise to /DTR//REQ Not Valid Delay		N/A		N/A	
37	TdPC(INT)	PCLK Fall to /INT Valid Delay		450		175	
38	TdlAiRD)	/INTACK to /RD Fall (Acknowledge) Delay	125		75		[3]
39	TwRDA	/RD (Acknowledge) Width	125		70		[3]
40	TdRDA(DR)	/RD Fall (Acknowledge) to Read Data Valid Delay	120		70		
41	TdRDA(INT)	/RD Fall to /INT Inactive Delay		320		200	[2]
42	TdRD(WRQ)	/RD Rise to /WR Fall Delay for No Reset	15		10		

No	Symbol	Parameter	10 MHz Min	10 MHz Max	16 MHz Min	16 MHz Max	Notes †
43	TdWRQ(RD)	/WR Rise to /RD Fall Delay for No Reset	15		10		
44	TwRES	/WR and /RD Coincident Low for Reset	100		75		
45	Trc	Valid Access Recovery Time	3.5TcPc		3.5TcPc		[1]

Notes:
[1] Parameter is guaranteed by design and 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)

AC CHARACTERISTICS
DMA Read Target Receive Cycle

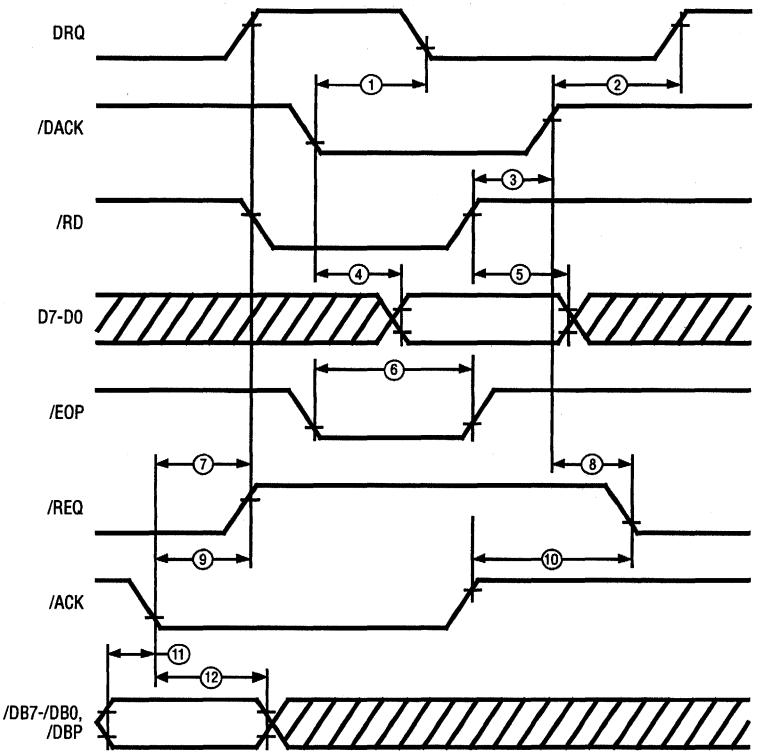


Figure 64. DMA Read Target Receive Cycle

AC CHARACTERISTICS
DMA Read Target Receive Cycle

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low		100	ns
2	/DACK High to DRQ High	N/A		ns
3	/DACK Hold Time from End of /RD	0		ns
4	Data Access Time from Read Enable*		100	ns
5	Data Hold Time from End of /RD	20		ns
6	Width of /EOP Pulse [1]	70		ns
7	/ACK Low to DRQ High		110	ns
8	/DACK High to /REQ Low (/ACK High)		120	ns
9	/ACK Low to /REQ High		125	ns
10	/ACK High to /REQ Low (/DACK High)		120	ns
11	Data Setup Time to /ACK	20		ns
12	Data Hold Time from /ACK	65		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.
N/A - Not Applicable. Zilog does not test this spec.

AC CHARACTERISTICS
DMA Read Initiator Receive Cycle

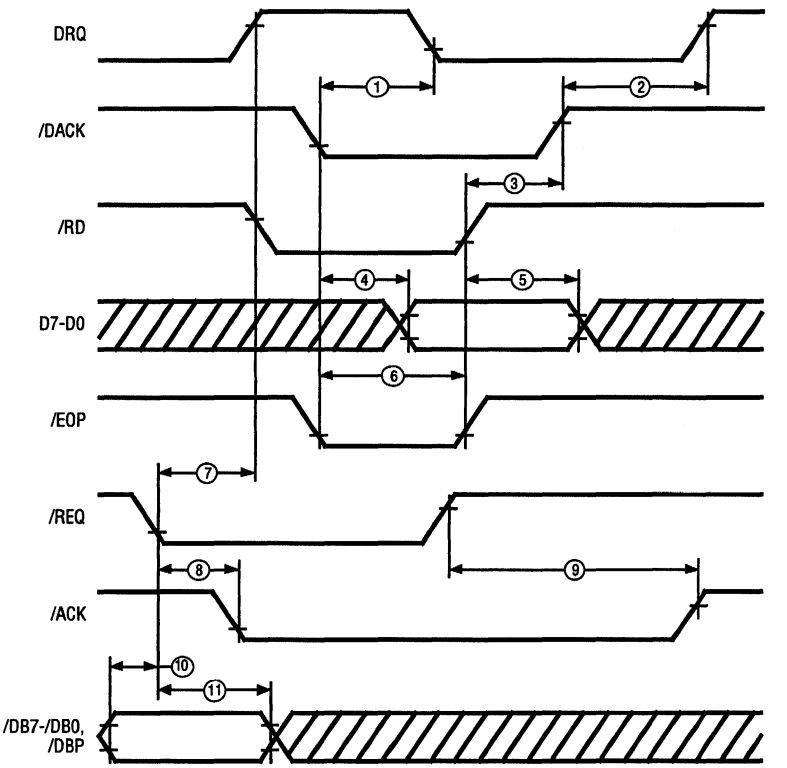


Figure 65. DMA Read Initiator Receive Cycle

AC CHARACTERISTICS

DMA Read Initiator Receive Cycle

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low		100	ns
2	/DACK High to DRQ High	N/A		ns
3	/DACK Hold Time from End of /RD	0		ns
4	Data Access Time from Read Enable*		100	ns
5	Data Hold Time from End of /RD	20		ns
6	Width of /EOP Pulse [1]	70		ns
7	/REQ Low to DRQ High		140	ns
8	/REQ Low to /ACK Low [3]		115	ns
9	/REQ High to /ACK High [2]		100	ns
10	Data Setup Time to /REQ	20		ns
11	Data Hold Time from /REQ	65		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.

[2] For Fast Ack operation. /ACK will be automatically released after /REQ.

[3] Guaranteed by design

N/A - Not Applicable. Zilog does not test this spec.

AC CHARACTERISTICS
DMA Write Initiator Send Cycle

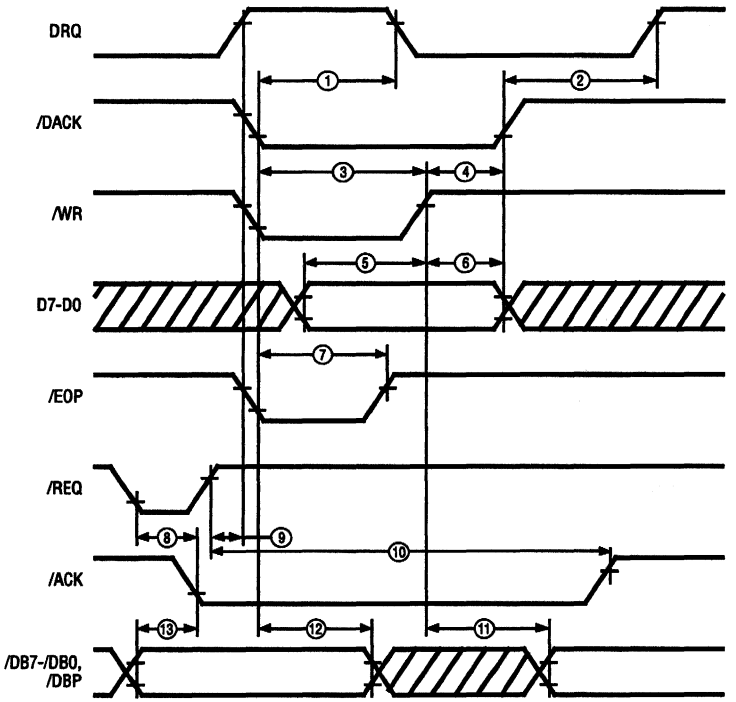


Figure 66. DMA Write Initiator Send Cycle

AC CHARACTERISTICS

DMA Write Initiator Send Cycle

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low		100	ns
2	/DACK High to DRQ High	N/A		ns
3	Write Enable Width*	70		ns
4	/DACK Hold from End of /WR	0		ns
5	Data Setup to End of Write Enable* [3]	30		ns
6	Data Hold Time from End of /WR	40		ns
7	Width of /EOP Pulse [1]	70		ns
8	/REQ Low to /ACK Low	55	115 [3]	ns
9	/REQ High to DRQ High		70	ns
10	/REQ High to /ACK High		90	ns
11	/WR High to Valid SCSI Data		100	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.

[2] For fast ACK operations, /ACK should be released after /REQ.

[3] Guaranteed by design

N/A - Not Applicable. Zilog does not test this spec.

AC CHARACTERISTICS
DMA Write Target Send Cycle

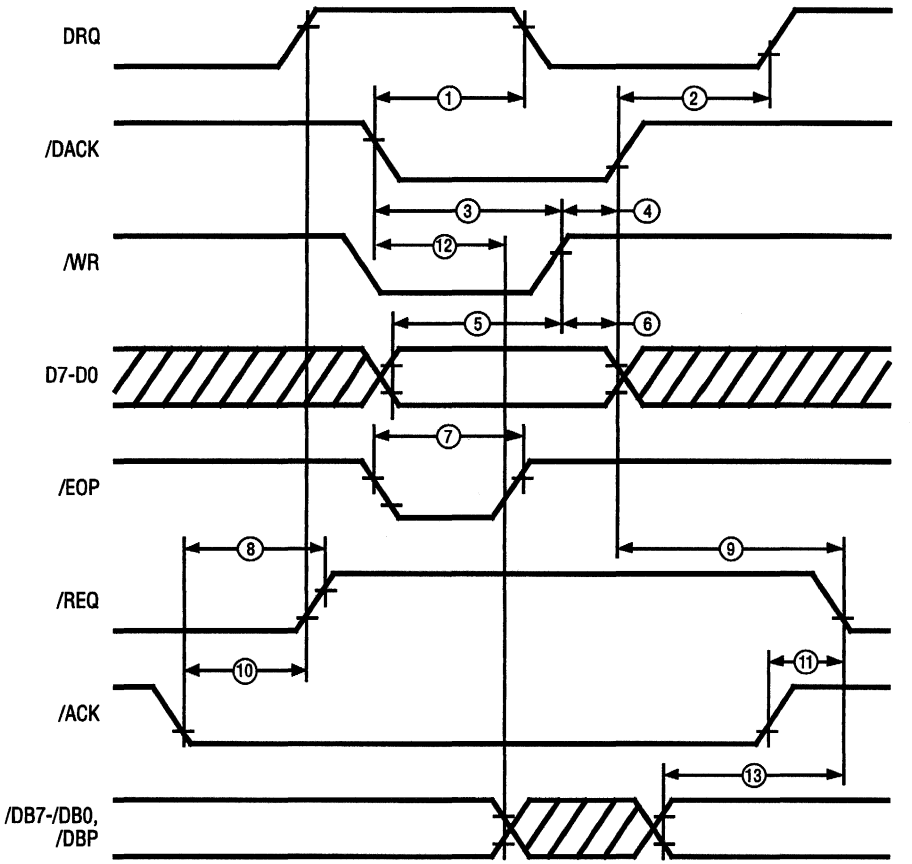


Figure 67. DMA Write Target Send Cycle

AC CHARACTERISTICS
DMA Write Target Send Cycle

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low		100	ns
2	/DACK High to DRQ High	N/A		ns
3	Write Enable Width*	70		ns
4	/DACK Hold from /WR High	0		ns
5	Data Setup to End of Write Enable* [2]	30		ns
6	Data Hold Time from End of /WR	40		ns
7	Width of /EOP Pulse [1]		70	ns
8	/ACK Low to /REQ High		125	ns
9	/REQ from End of /DACK (/ACK High)		130	ns
10	/ACK Low to DRQ High (Target)		110	ns
11	/ACK High to /REQ Low (/DACK High)		130	ns
12	Data Hold from Write Enable	15		ns
13	Data Setup to /REQ Low (Target)	60		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
[2] Guaranteed by design
N/A Not Applicable - Zilog does not test this spec.

AC CHARACTERISTICS
CPU Write Cycle

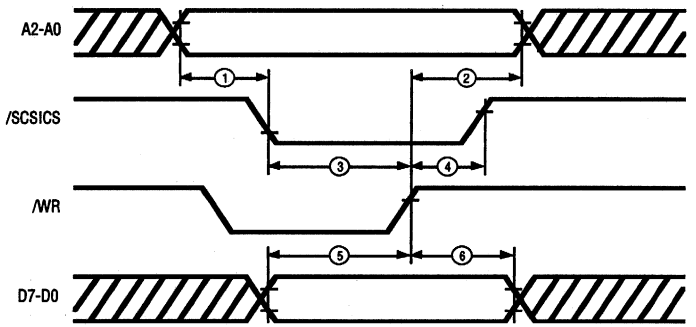


Figure 68. CPU Write Cycle

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* [1]	0		ns
3	Write Enable Width*	40		ns
4	Chip Select Hold from End of /WR	0		ns
5	Data Setup to end of Write Enable*	20		ns
6	Data Hold Time form End of /WR	20		ns

Note:
* Write Enable is the occurrence of /WR and /SCSICS.
[1] Guaranteed by design

AC CHARACTERISTICS
CPU Read Cycle

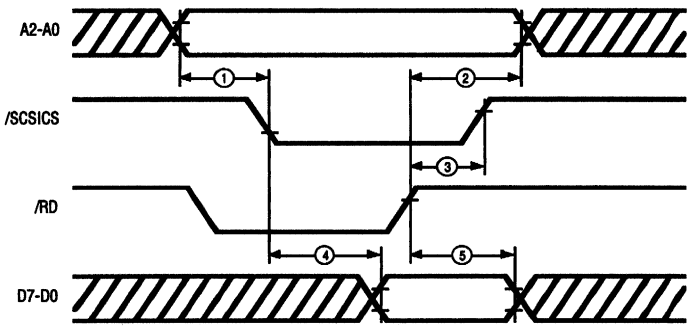


Figure 69. CPU Read Cycle

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* [1]	0		ns
3	Chip Select Hold from End of /RD	0		ns
4	Data Access Time from Read Enable*		100	ns
5	Data Hold Time from End of Read Enable*	20		ns

Note:
* Read Enable is the occurrence of /RD and /SCSICS.
[1] Guaranteed by design.

6

AC CHARACTERISTICS
Selection

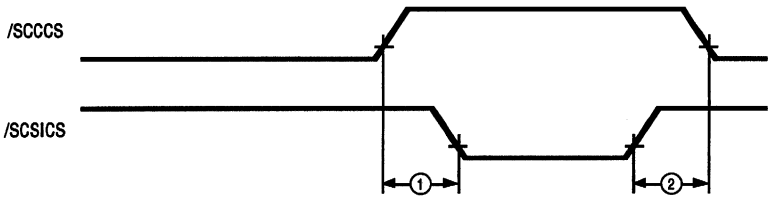


Figure 70. Selection

AC CHARACTERISTICS
Selection

No	Description	Min	Max	Units
1	/SCCCS to /SCSICS	100		ns
2	/SCSICS to /SCCCS	100		ns

AC CHARACTERISTICS
Arbitration

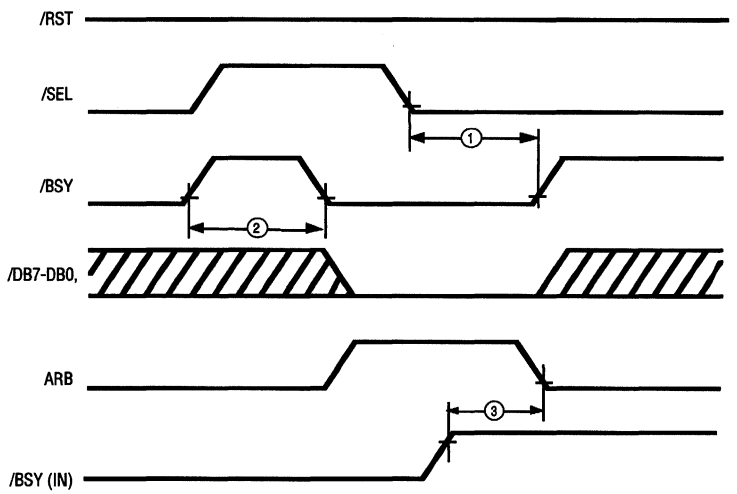


Figure 71. Arbitration

AC CHARACTERISTICS
Arbitration

No	Description	Min	Max	Units	Note
1	Bus Clear from /SEL Low		600	ns	
2	Arbitrate Start from /BSY False	1200	2400	ns	[1]
3	Bus Clear from /BSY High		1100		

Note:
[1] Guaranteed by design, not tested.

AC CHARACTERISTICS
Reset

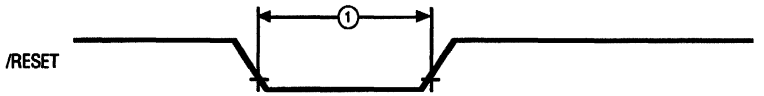
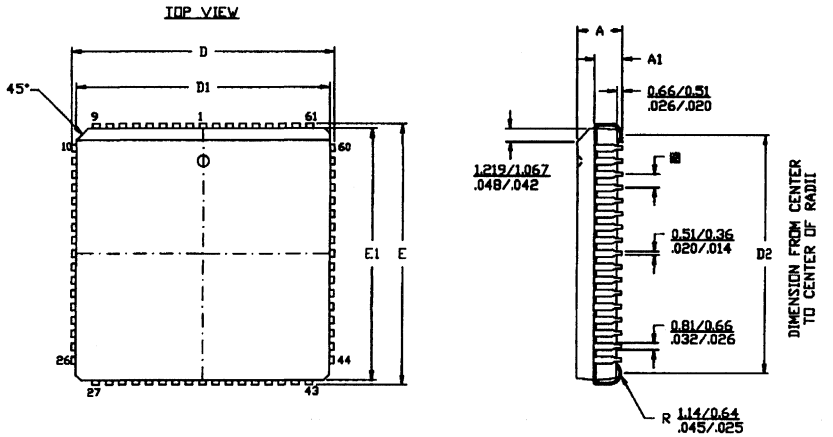


Figure 72. Reset

AC CHARACTERISTICS
Reset

No	Description	Min	Max	Units
1	Minimum Width of /RESET	100		ns

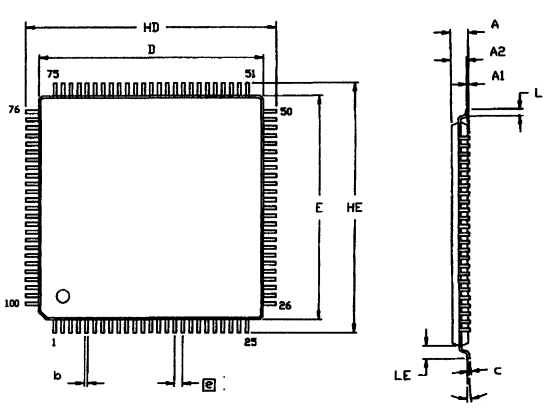
PACKAGE INFORMATION



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

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

68-Pin PLCC Package Diagram



SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	1.35	1.60	.053	.063
A1	0.05	0.20	.002	.008
A2	1.30	1.50	.051	.059
b	0.15	0.26	.006	.010
c	0.10	0.20	.004	.008
HD	15.85	16.15	.624	.636
D	13.90	14.10	.547	.555
HE	15.85	16.15	.624	.636
E	13.90	14.10	.547	.555
e	0.50 TYP		.020 TYP	
L	0.35	0.65	.014	.026
LE	0.90	1.10	.035	.043

1. CONTROLLING DIMENSIONS : MM
2. MAX COPLANARITY : .10mm / .004"

100-Pin VQFP Package Diagram

6

ORDERING INFORMATION

Z85C80

68-Pin PLCC		100-Pin VQFP
10 MHz	**16 MHz**	**16 MHz**
Z85C8010VSC	Z85C8016VSC	Z85C8016ASC

Codes

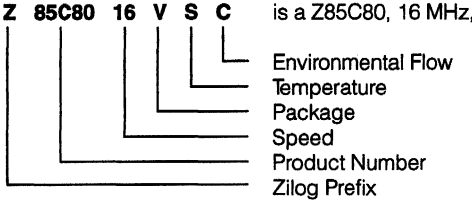
Package
V = Plastic Chip Carrier
A = Very Small Plastic Quad Flat Pack

Temperature
S = 0°C to +70°C

Environmental
C = Plastic Standard
E = Hermetic Standard

Example:

Z 85C80 16 V S C is a Z85C80, 16 MHz, VQFP, 0°C to +70°C, Plastic Standard Flow



Z16C35/Z85C35 CMOS ISCC™ Integrated Serial Communication Controller 7

SCC Application Notes 8

SCC Support Products 9

Superintegration™ Products Guide S

Zilog Sales Offices Representatives & Distributors Z

Zilog's Literature Guide Ordering Information L

Z16C35/Z85C35

CMOS ISCC™ INTEGRATED SERIAL COMMUNICATIONS CONTROLLER

FEATURES

- 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 680 x 0 and 8 x 86 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 10 x 19 Status FIFO for DMA Support

GENERAL DESCRIPTION

The Z85C35, directly equivalent to 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 10 x 19 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.

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

Power connections follow conventional descriptions below:

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

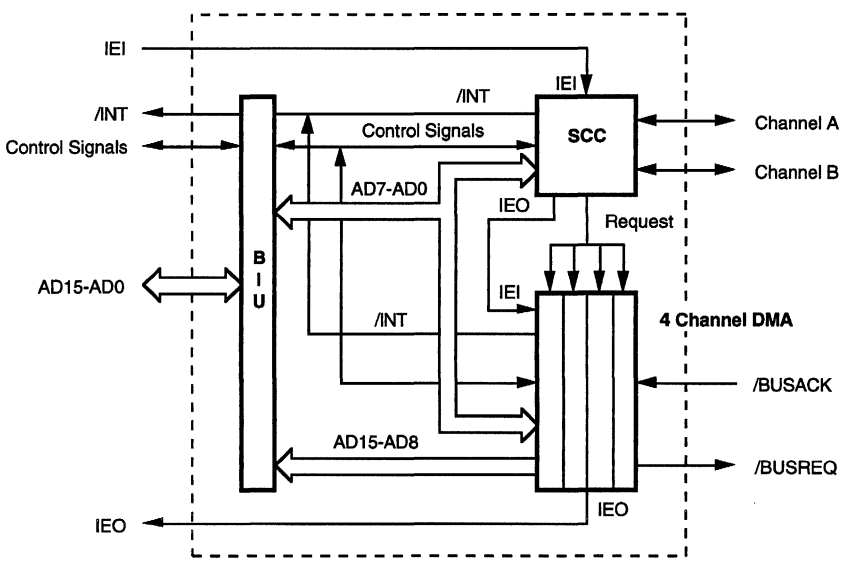


Figure 1. Block Diagram

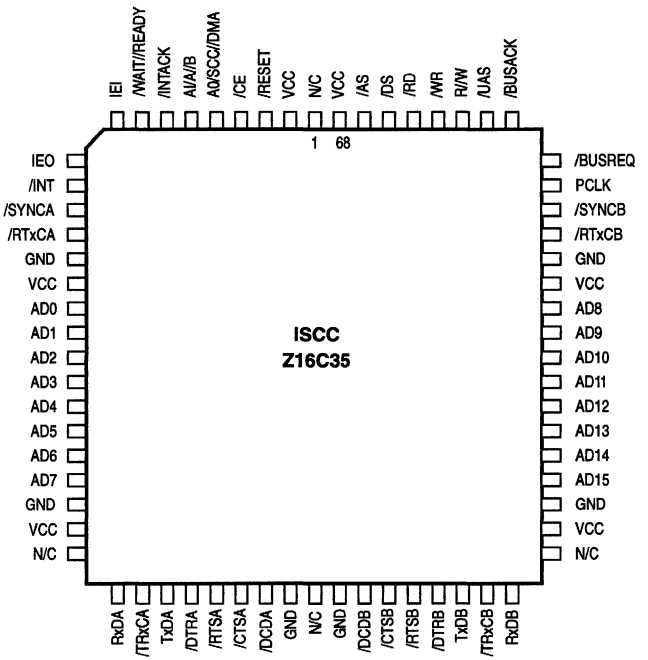


Figure 2. Pin Assignments

PIN DESCRIPTION

The following section describes the Z16C35 pin functions. Figure 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. This signal tristates at reset.

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 8 x 86 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 Low). 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, tri-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 8 x 86 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 680 x 0 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. This signal tri-states at reset.

/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.

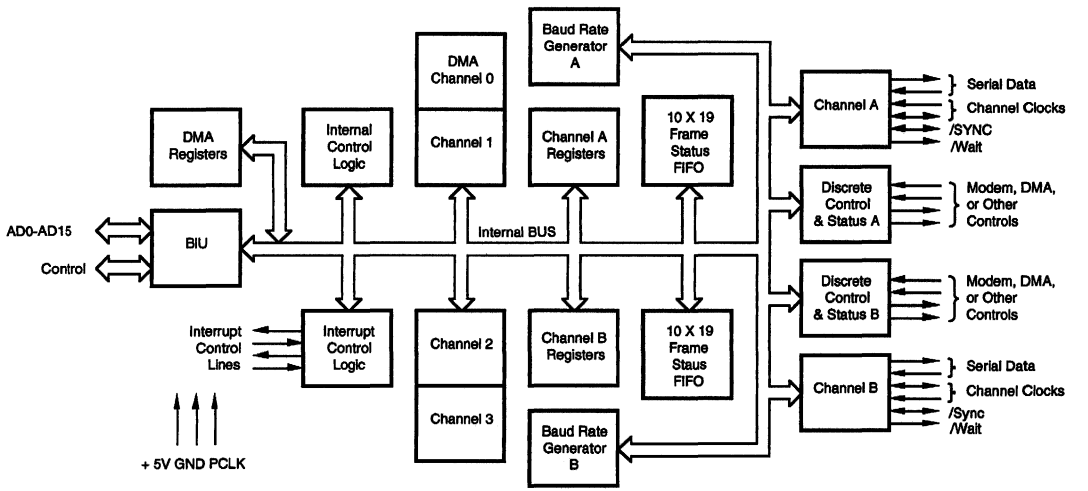


Figure 3. Block Diagram of ISCC Architecture

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.

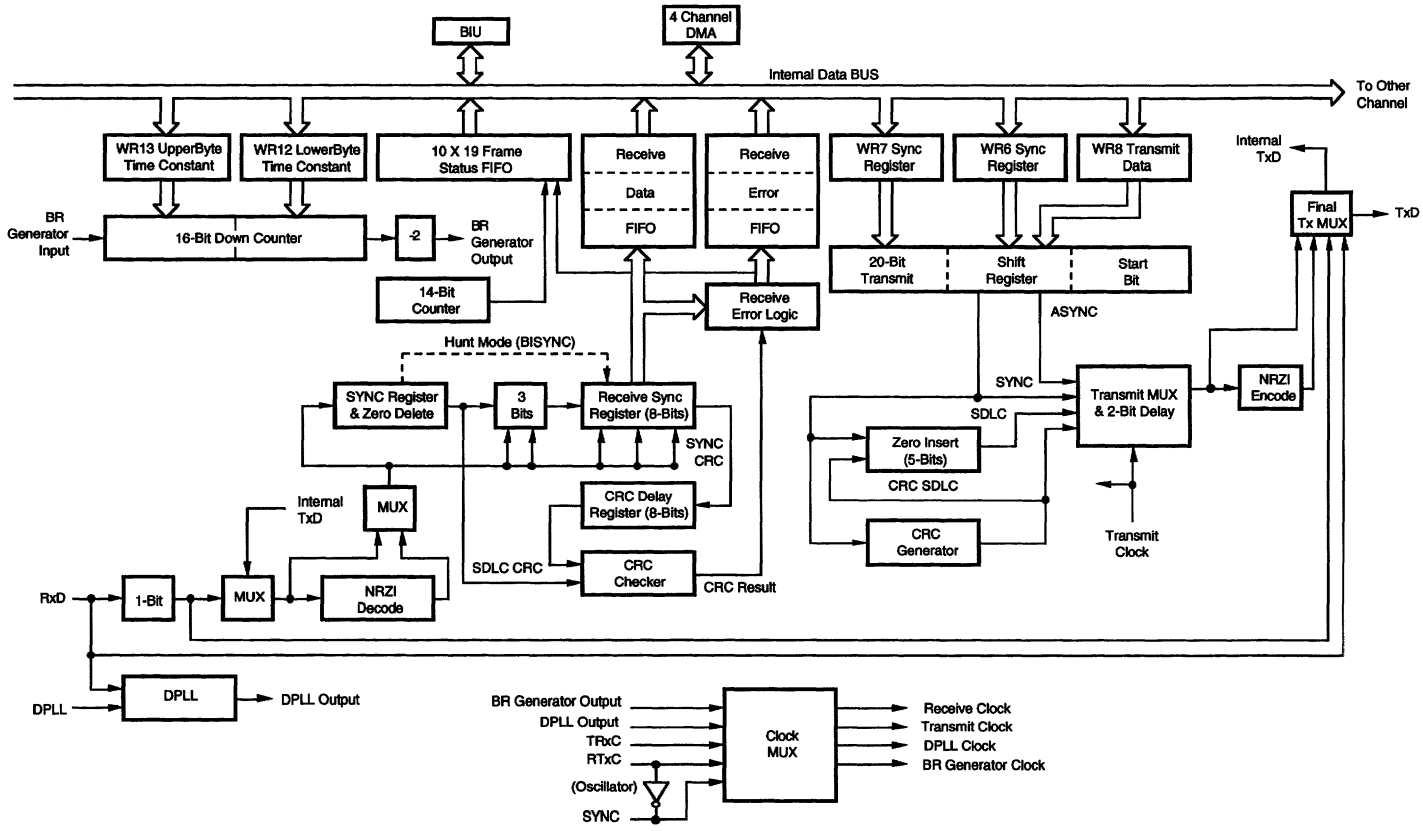


Figure 4. ISCC Data Path

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.

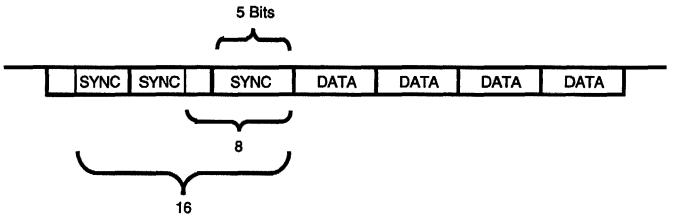


Figure 5. Detecting 5- or 7-Bit Synchronous Characters

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 $8 \times 86 + X15 + X2 + 1$) and CCITT $8 \times 86 + X12 + X5 + 1$) 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).

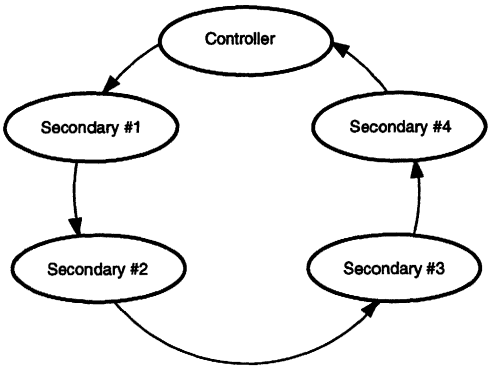


Figure 6. An SDLC Loop

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 10 x 19 status FIFO is separate from the 3 byte receive data FIFO.

Notes on the SDLC FIFO. When using the SDLC FIFO enhancement in channel B, it is necessary to enable the enhancement 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 enhancement 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 start-up, 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.

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

FUNCTIONAL DESCRIPTION (Continued)

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.

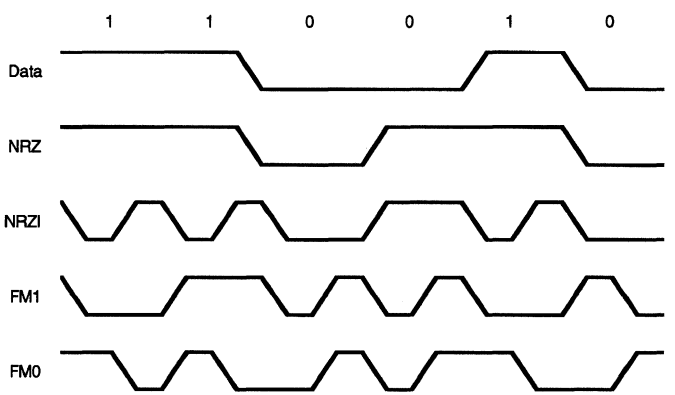


Figure 7. Data Encoding Methods

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.

7

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).

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).

FUNCTIONAL DESCRIPTION (Continued)

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:

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.

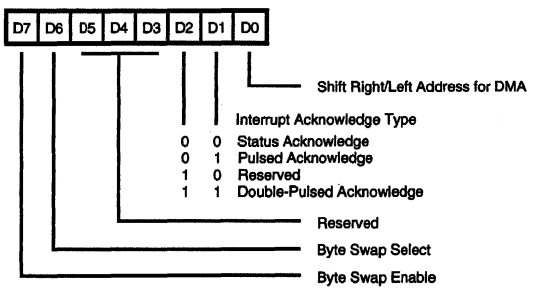


Figure 8. Bus Configuration Register (BCR)

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

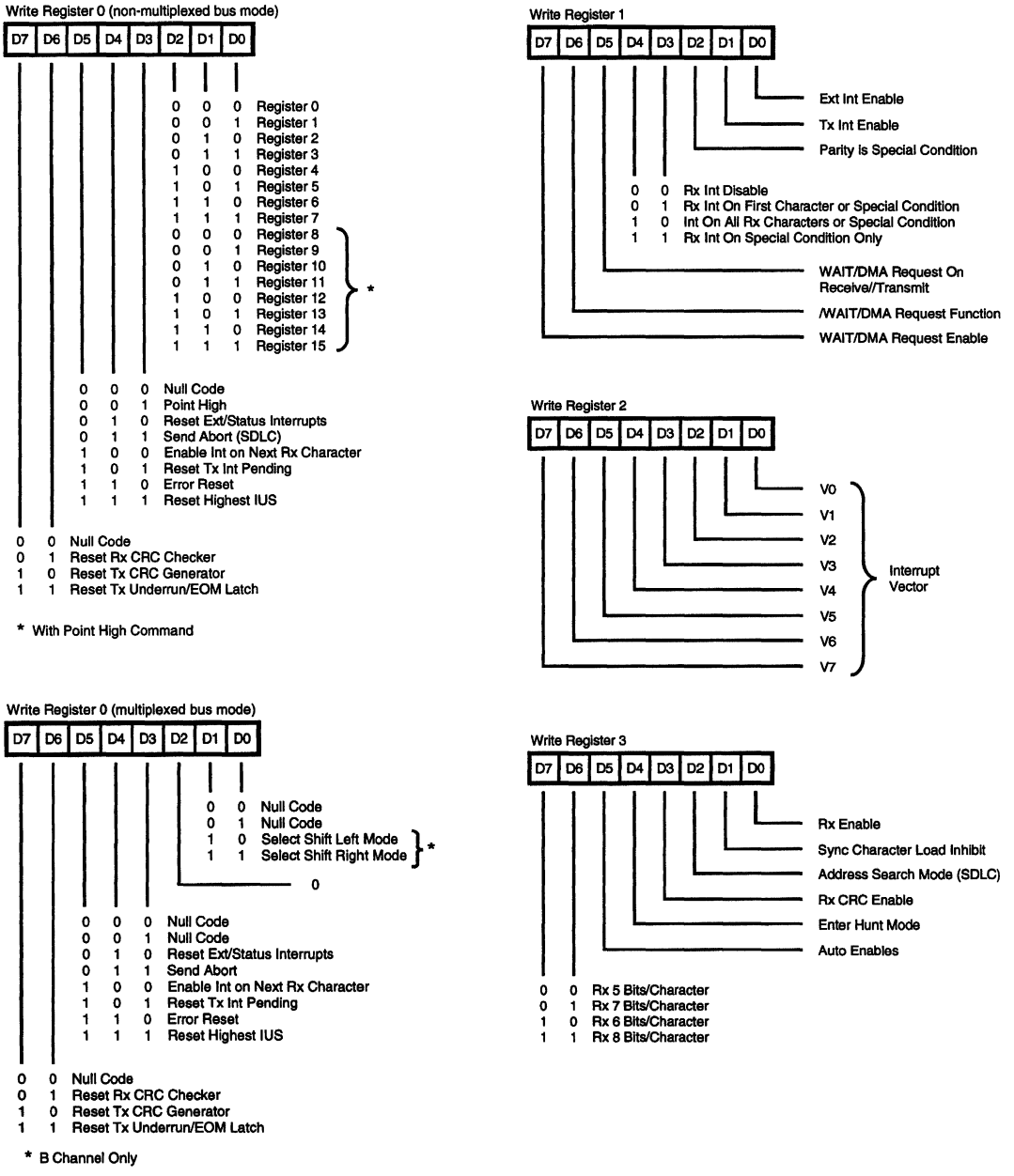


Figure 9. Write Register Bit Functions

CONTROL REGISTERS (Continued)

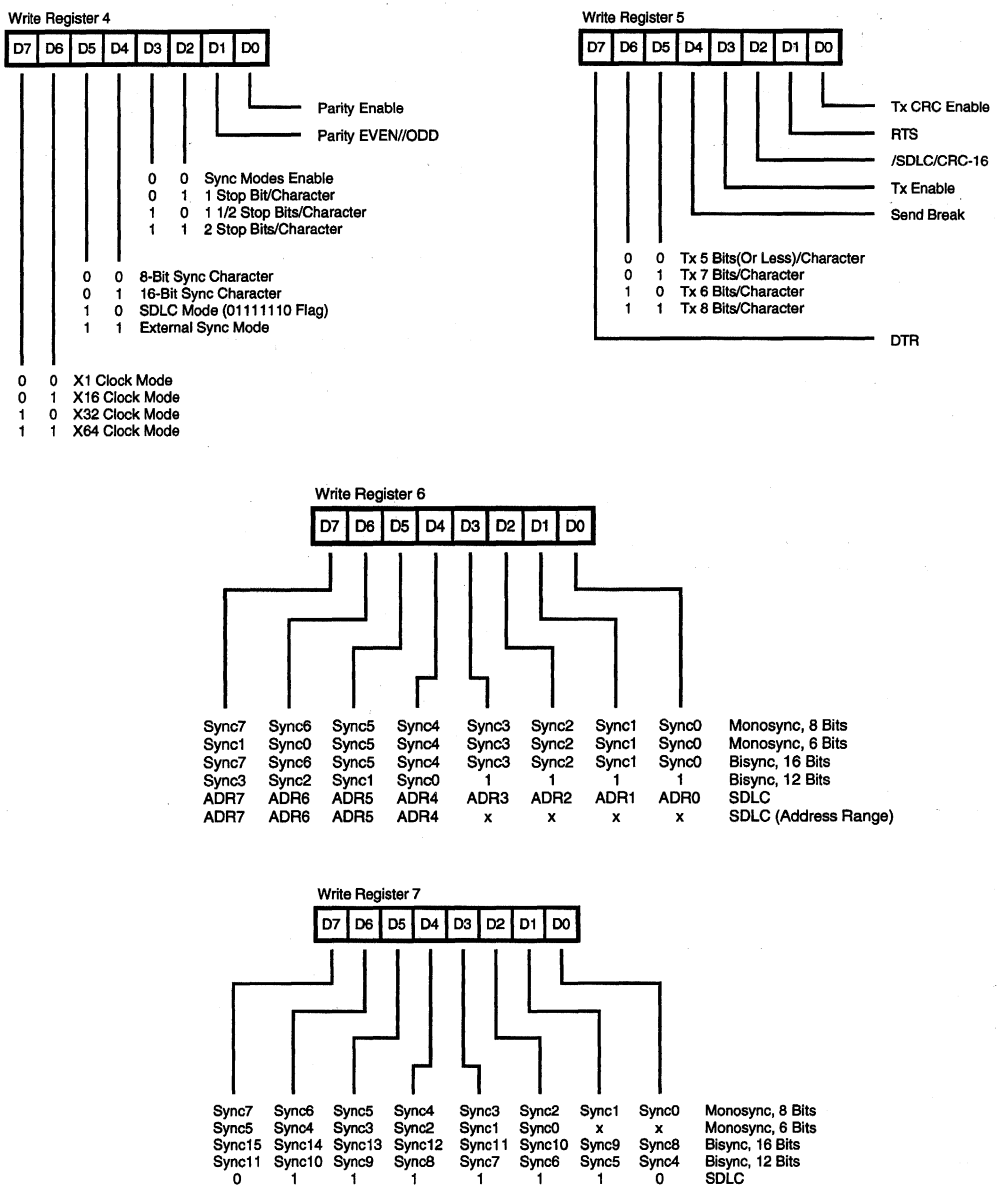


Figure 9. Write Register Bit Functions (Continued)

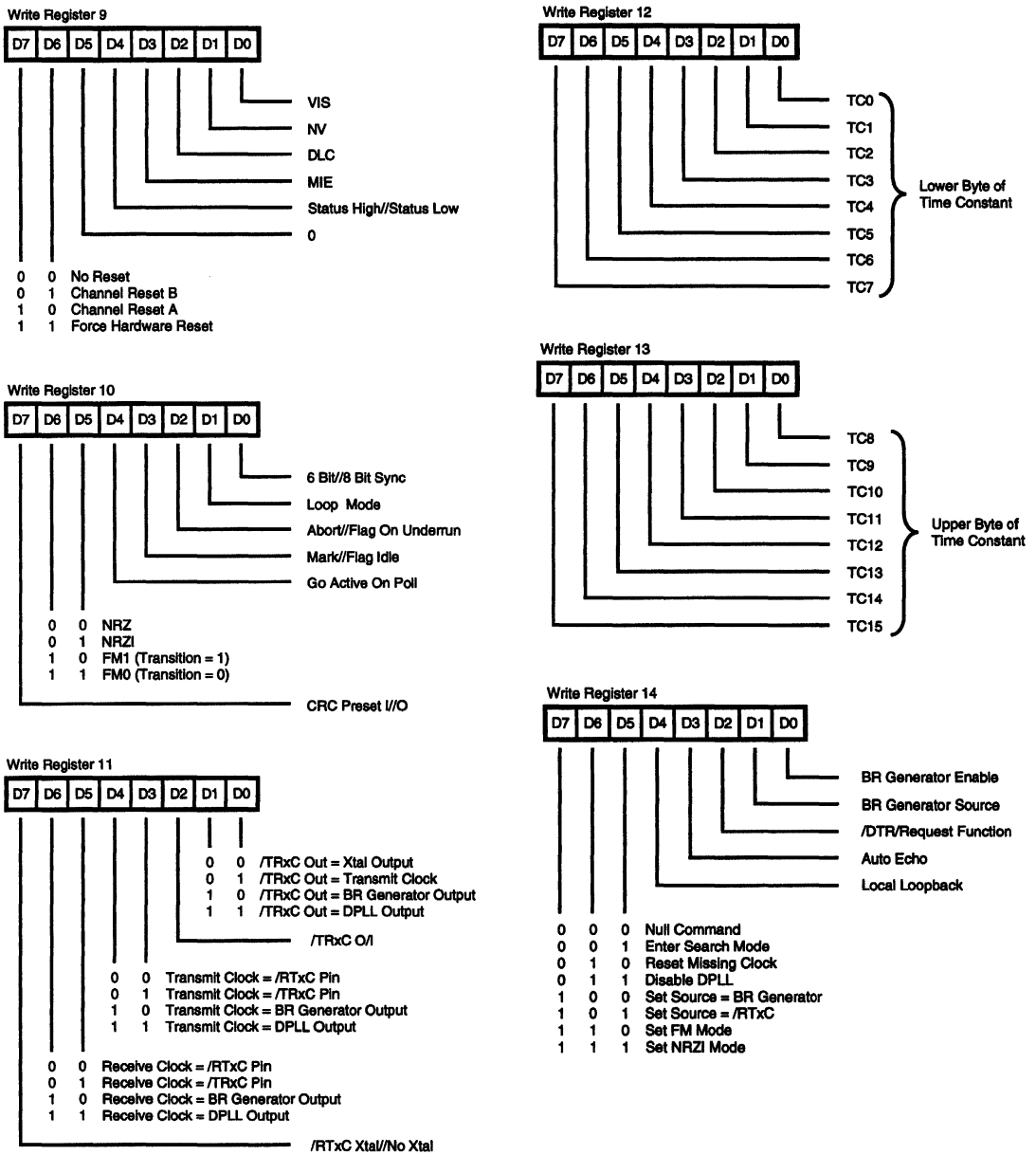


Figure 9. Write Register Bit Functions (Continued)

CONTROL REGISTERS (Continued)

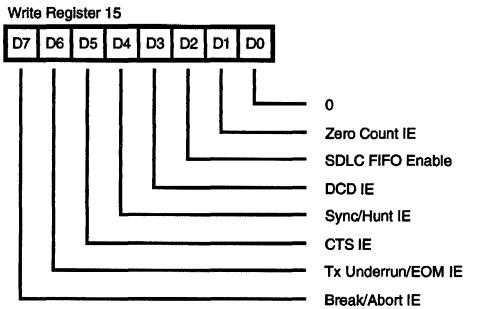


Figure 9. Write Register Bit Functions (Continued)

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

Read Register 0

D7	D6	D5	D4	D3	D2	D1	D0

- D0: Rx Character Available
- D1: Zero Count
- D2: Tx Buffer Empty
- D3: DCD
- D4: Sync/Hunt
- D5: CTS
- D6: Tx Underrun/EOM
- D7: Break/Abort

Read Register 1

D7	D6	D5	D4	D3	D2	D1	D0

- D0: All Sent
- D1: Residue Code 2
- D2: Residue Code 1
- D3: Residue Code 0
- D4: Parity Error
- D5: Rx Overrun Error
- D6: CRC/Framing Error
- D7: End of Frame (SDLC)

Read Register 2

D7	D6	D5	D4	D3	D2	D1	D0

- D0: V0
- D1: V1
- D2: V2
- D3: V3
- D4: V4
- D5: V5
- D6: V6
- D7: V7

Interrupt Vector *

* Modified In B Channel

Read Register 3

D7	D6	D5	D4	D3	D2	D1	D0

- D0: Channel B Ext/Status IP
- D1: Channel B Tx IP
- D2: Channel B Rx IP
- D3: Channel A Ext/Status IP
- D4: Channel A Tx IP
- D5: Channel A Rx IP
- D6: 0
- D7: 0

* *

* Always 0 In B Channel

Read Register 6 *

D7	D6	D5	D4	D3	D2	D1	D0

- D0: BC0
- D1: BC1
- D2: BC2
- D3: BC3
- D4: BC4
- D5: BC5
- D6: BC6
- D7: BC7

* Can only be accessed if the SDLC FIFO enhancement is enabled (WR15 bit D2 set to 1)

SDLC FIFO Status and Byte Count (LSB)

Read Register 7 *

D7	D6	D5	D4	D3	D2	D1	D0

- D0: BC8
- D1: BC9
- D2: BC10
- D3: BC11
- D4: BC12
- D5: BC13
- D6: FDA: FIFO Available Status
1 = Status Reads from FIFO
- D7: FOS: FIFO Overflow Status
1 = FIFO Overflowed
0 = Normal

* Can only be accessed if the SDLC FIFO enhancement is enabled (WR15 bit D2 set to 1)

SDLC FIFO Status and Byte Count (MSB)

Figure 10. Read Register Bit Functions

CONTROL REGISTERS (Continued)

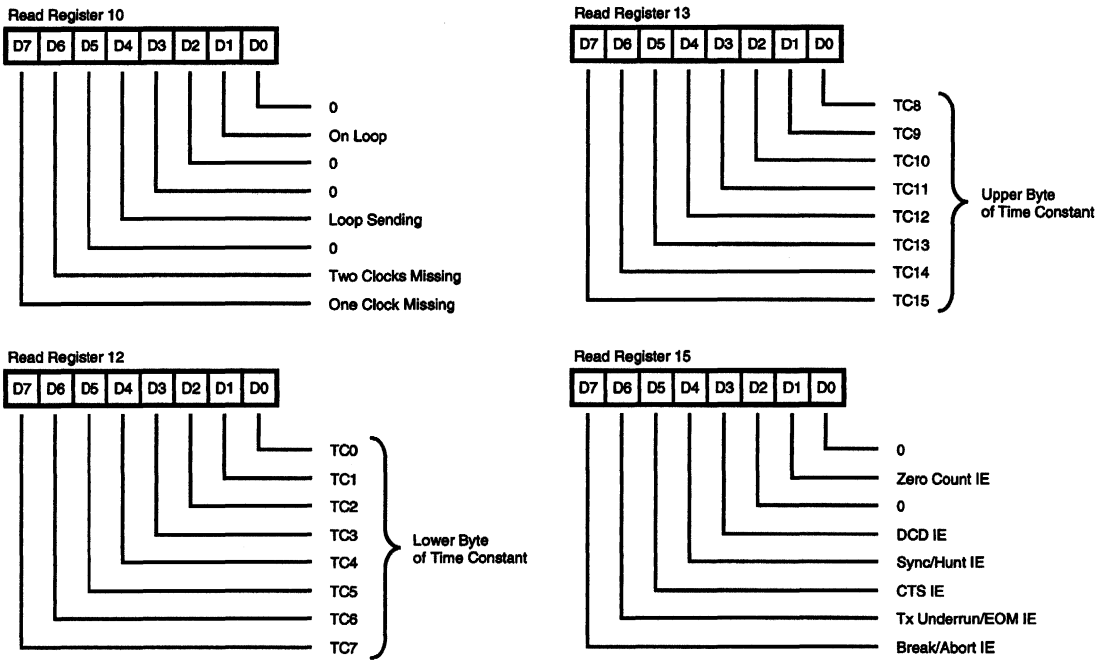


Figure 10. Read Register Bit Functions (Continued)

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

CONTROL REGISTERS (Continued)

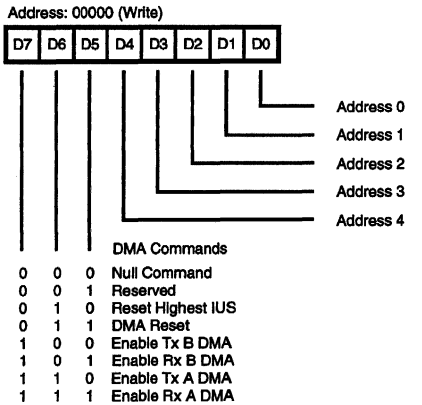


Figure 11. Channel Command/Address Register

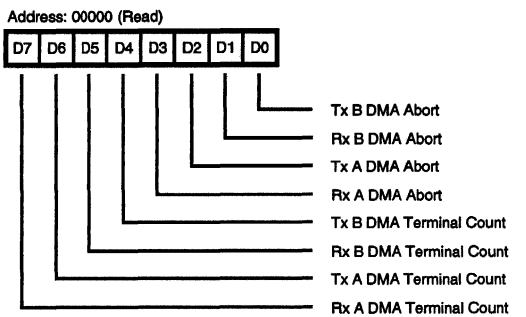


Figure 12. DMA Status Register

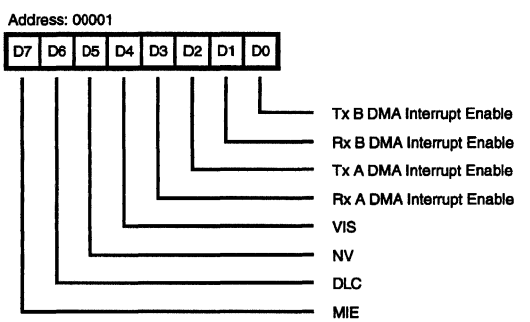


Figure 13. Interrupt Control Register

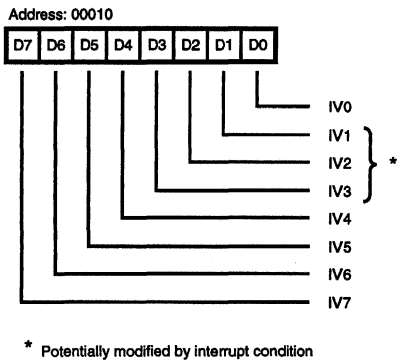


Figure 14. Interrupt Vector Register

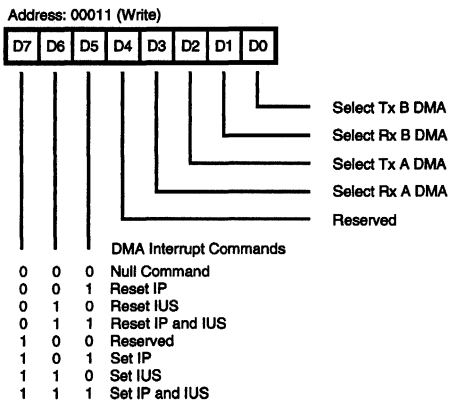


Figure 15. Interrupt Command/Register

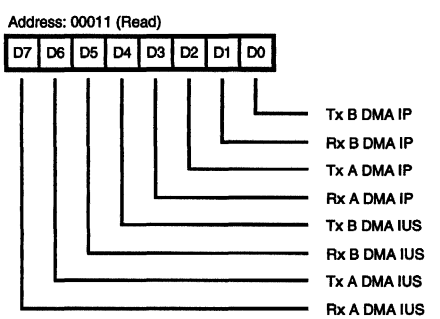


Figure 16. Interrupt Status Register

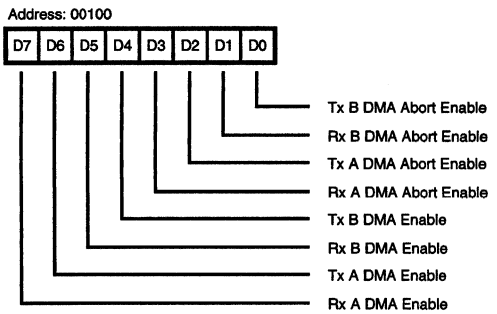


Figure 17. DMA Enable Register

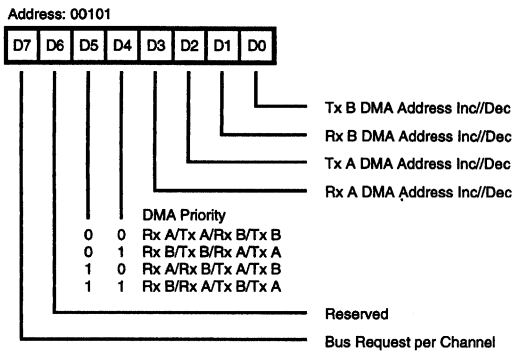


Figure 18. DMA Control Register

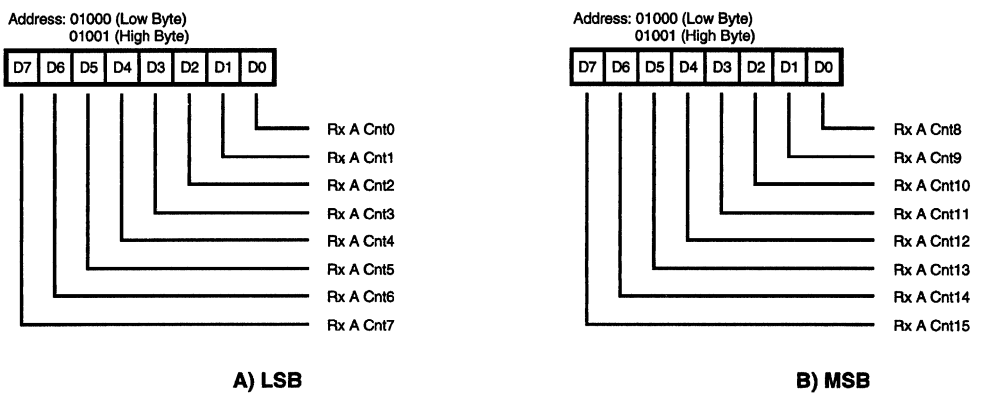


Figure 19. Receive DMA Count Register Channel A

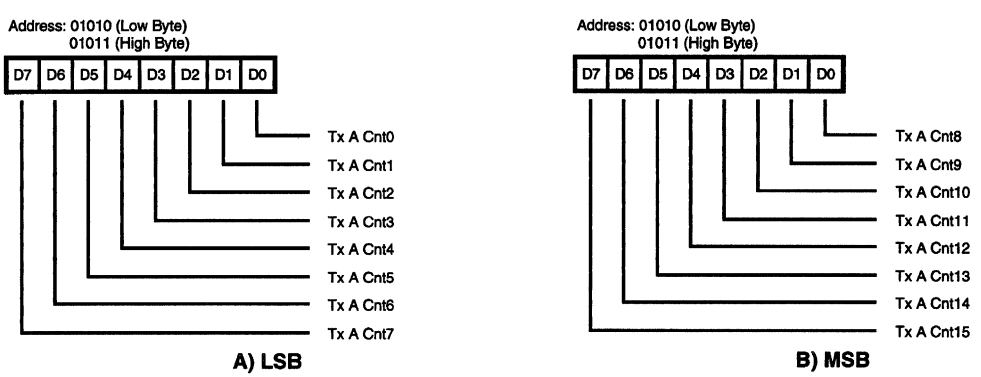


Figure 20. Transmit DMA Count Register Channel A

CONTROL REGISTERS (Continued)

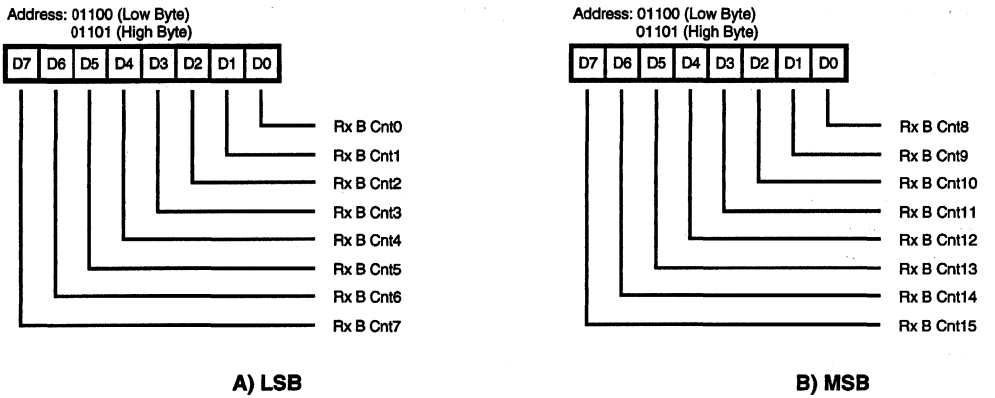


Figure 21. Receive DMA Count Register Channel B

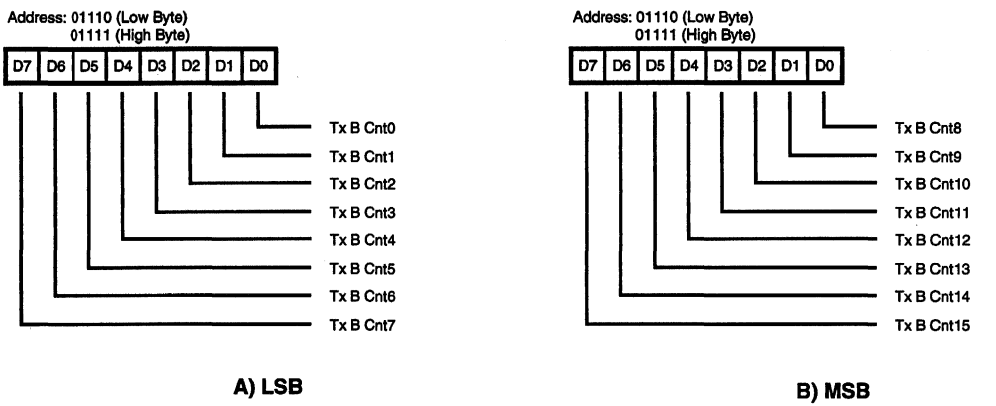


Figure 22. Transmit DMA Count Register Channel B

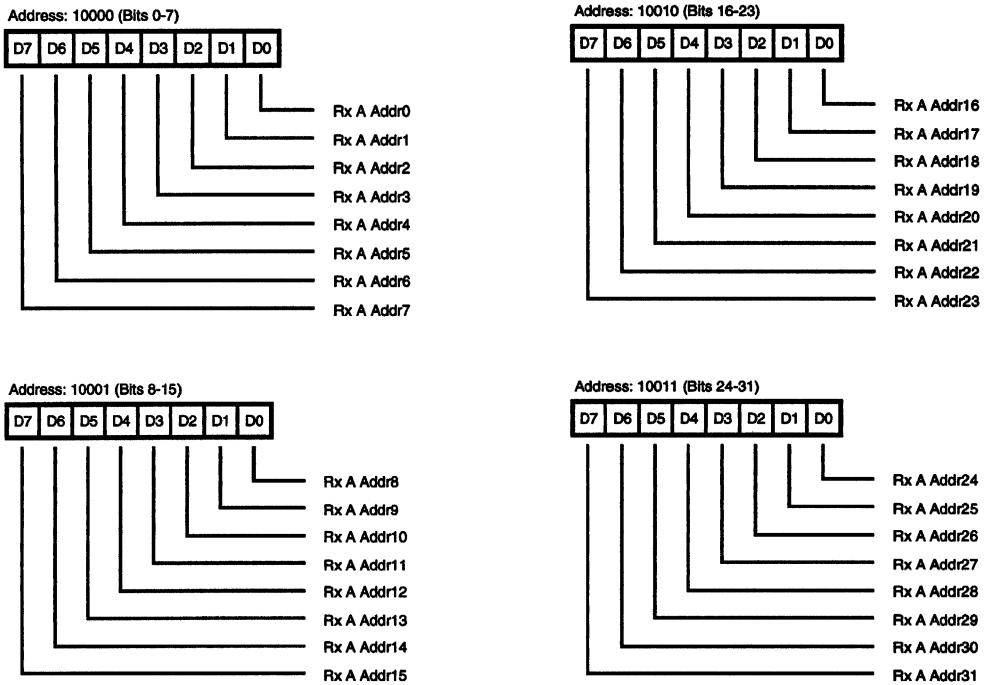


Figure 23. Receive DMA Address Register Channel A

CONTROL REGISTERS (Continued)

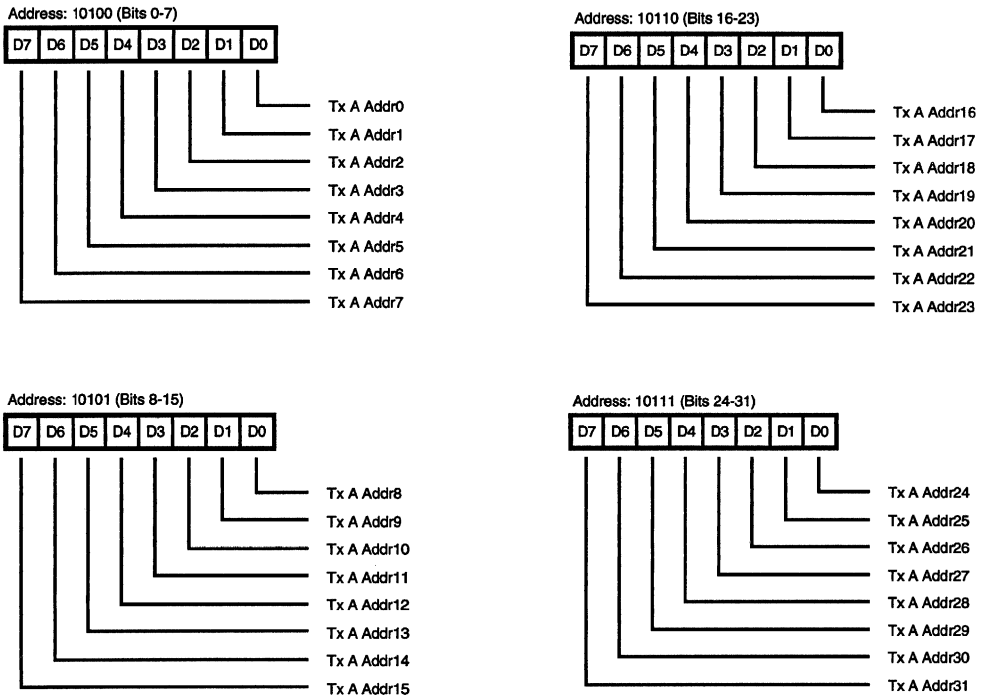


Figure 24. Transmit DMA Address Register Channel A

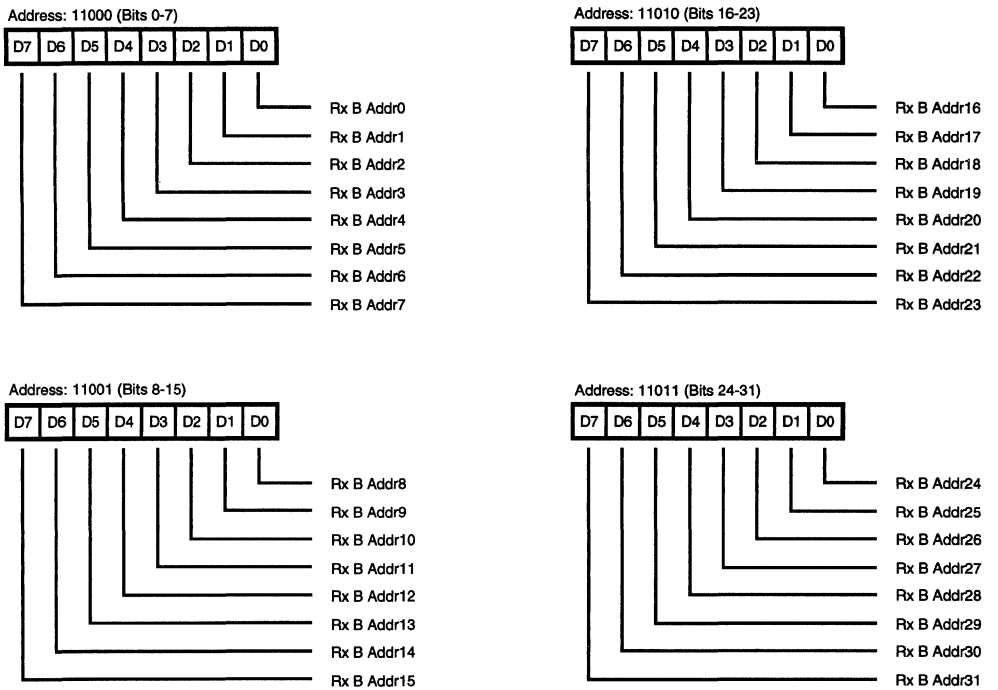


Figure 25. Receive DMA Address Register Channel B

CONTROL REGISTERS (Continued)

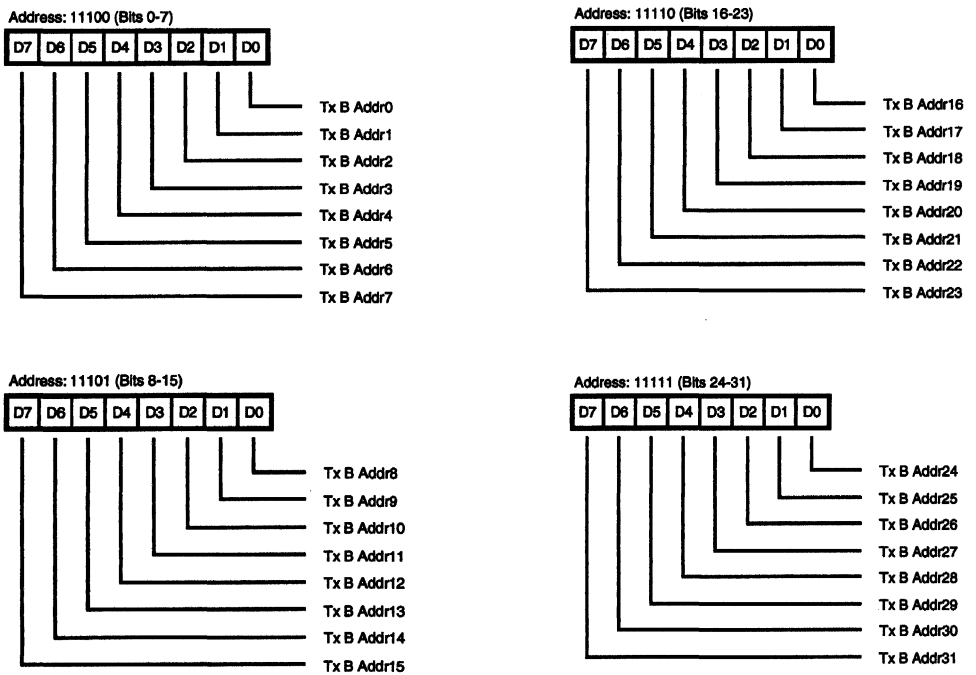


Figure 26. Transmit DMA Address Register Channel B

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.

7

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

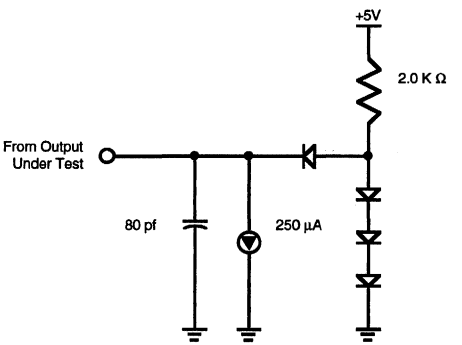


Figure 27. Standard Test Load

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).

			10 MHz		16 MHz		
No	Symbol	Parameter	Min	Max	Min	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	TwRESl	/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	TwRDl	/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	TwWRl	/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 + TwPCl + 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.
[11] Timings in nanoseconds

AC CHARACTERISTICS (Continued)

Timing Diagrams

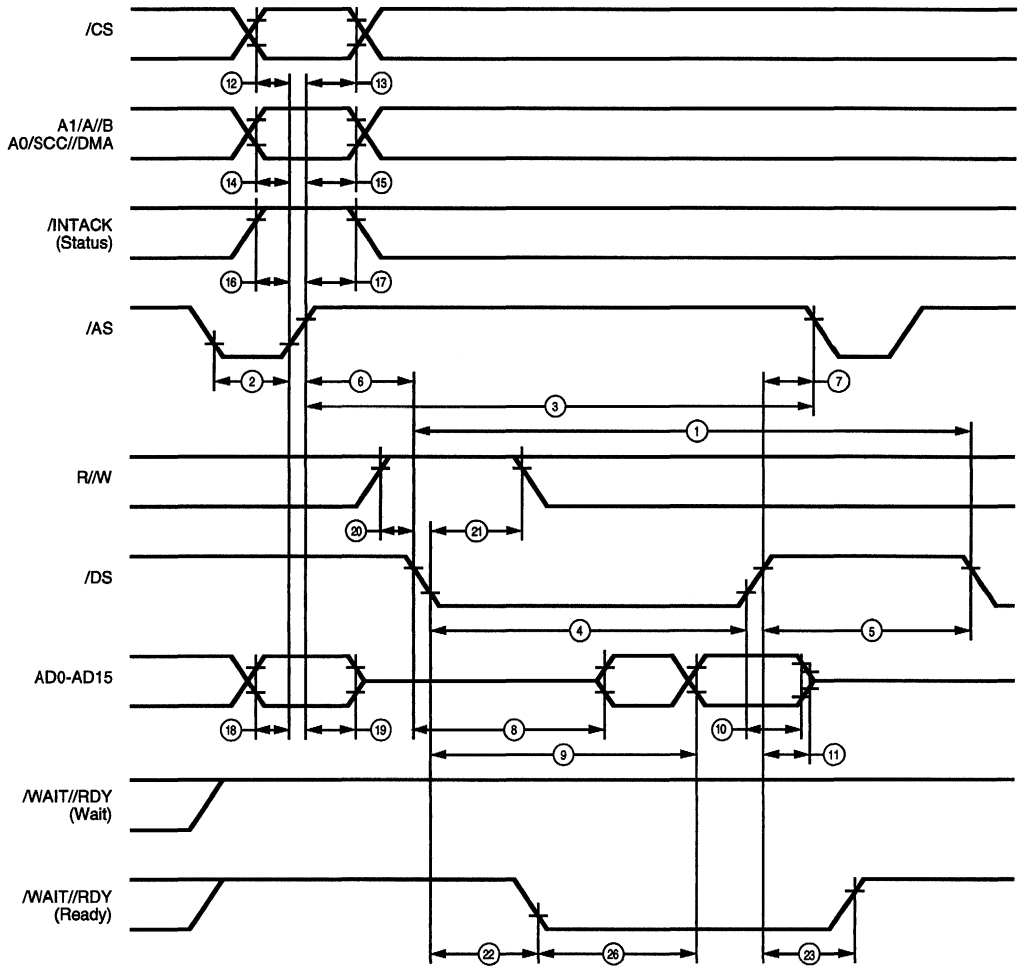


Figure 28. Multiplexed /DS Read Cycle

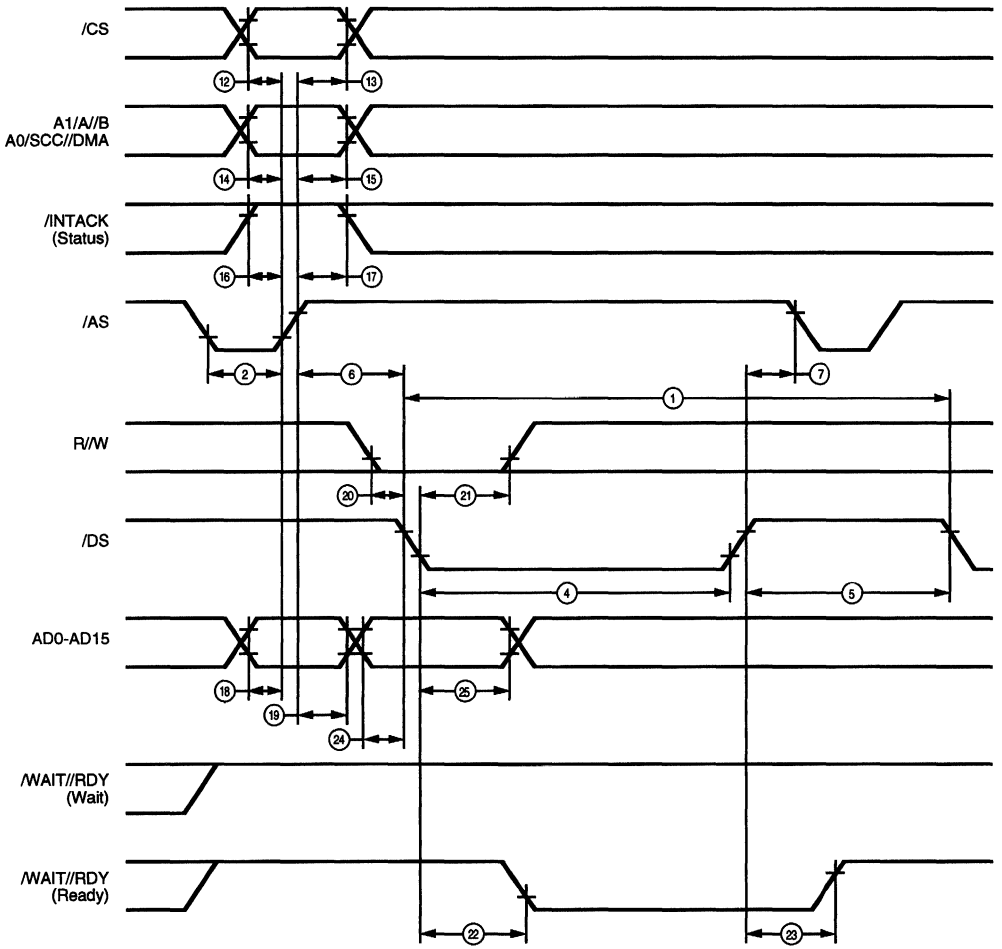


Figure 29. Multiplexed /DS Write Cycle

AC CHARACTERISTICS (Continued)

Timing Diagrams

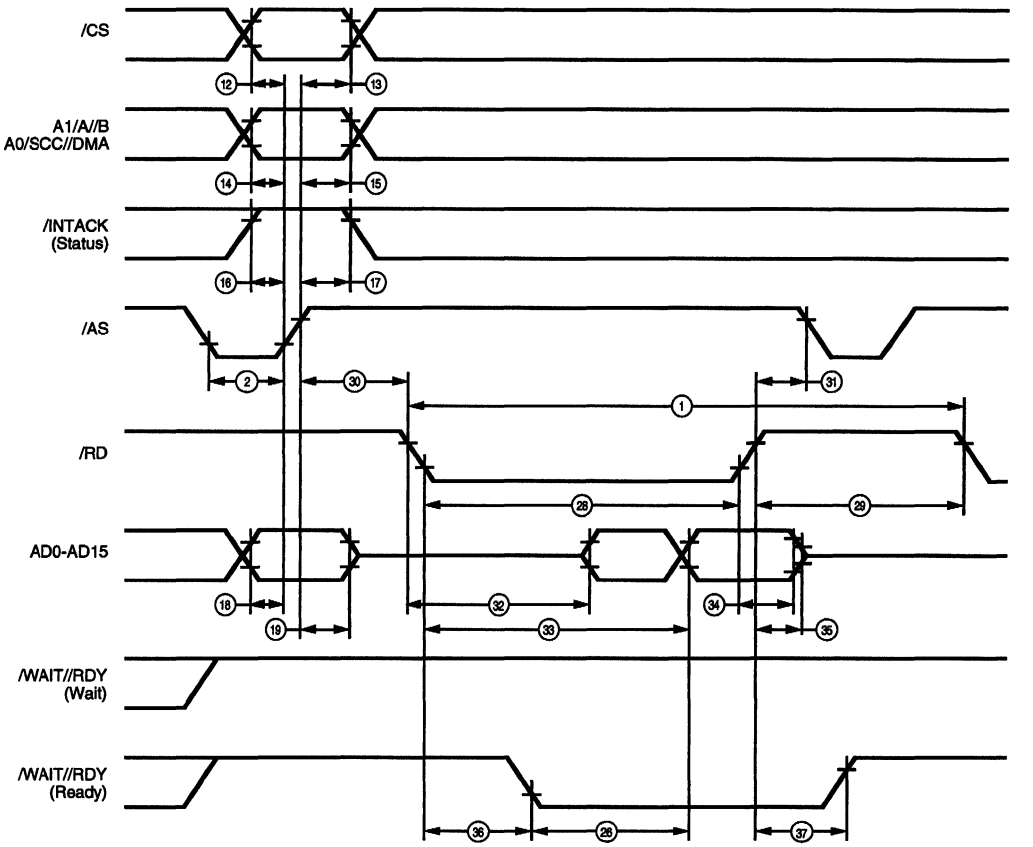


Figure 30. Multiplexed /RD Read Cycle

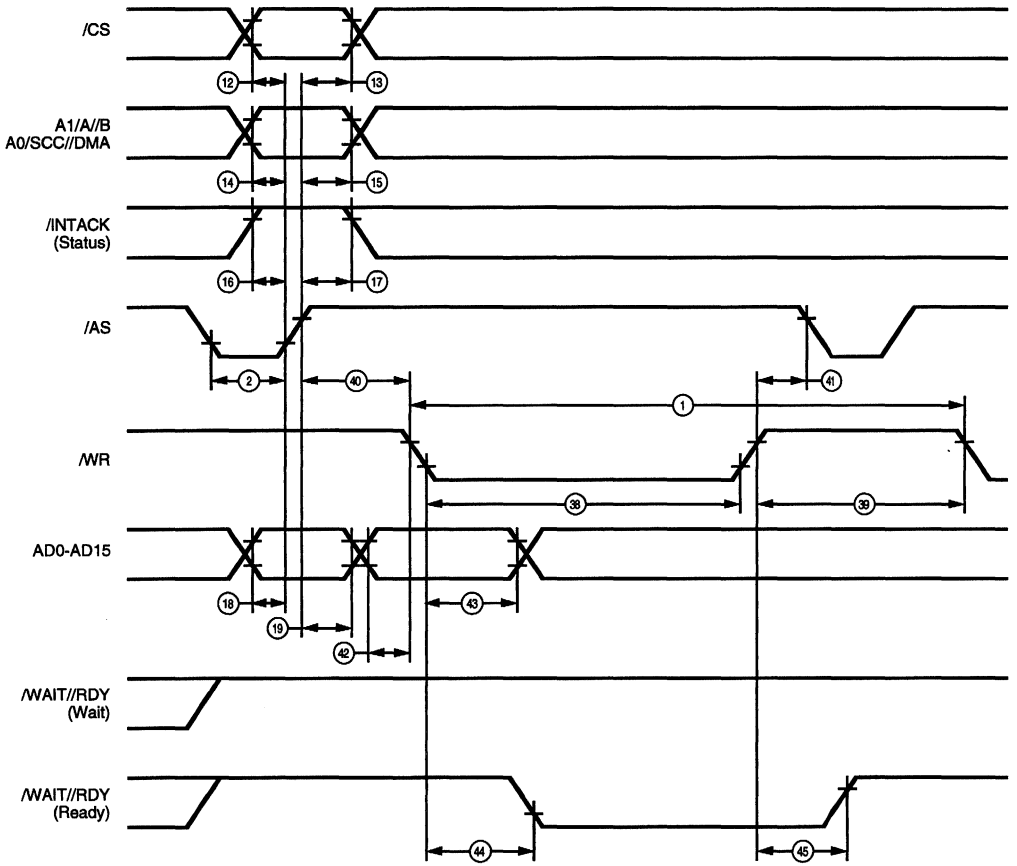


Figure 31. Multiplexed /WR Write Cycle

AC CHARACTERISTICS (Continued)
Timing Diagrams

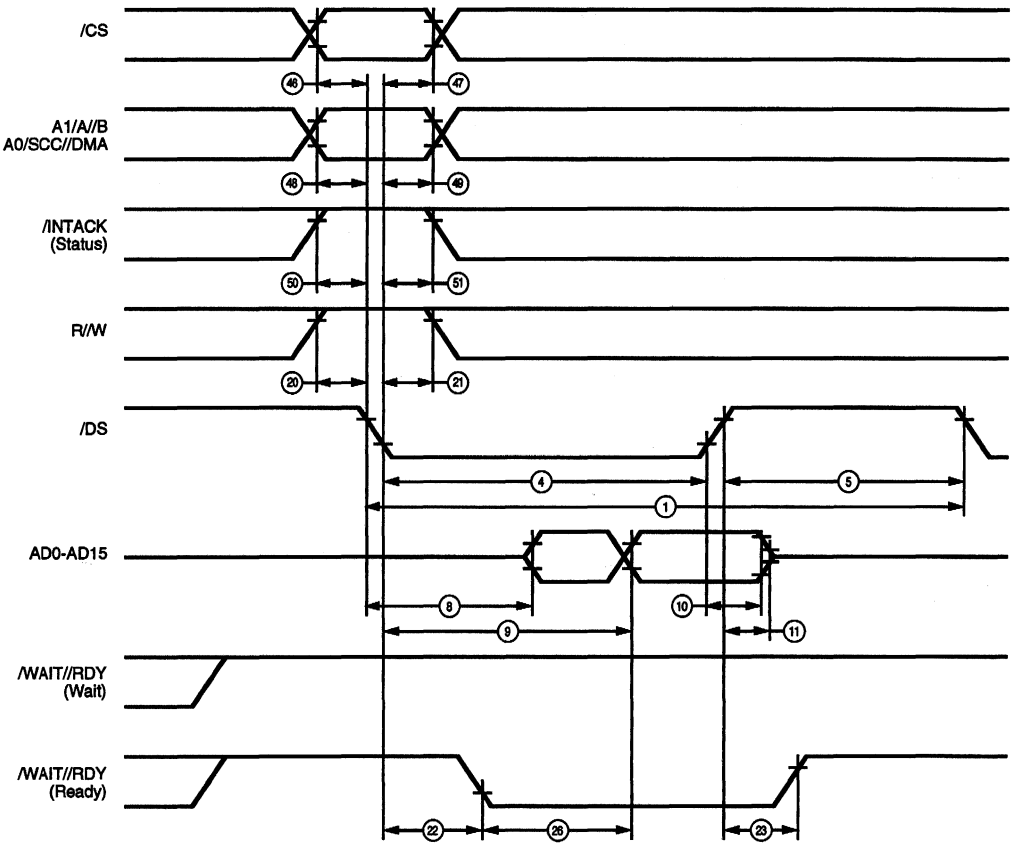


Figure 32. Non-multiplexed /DS Read Cycle

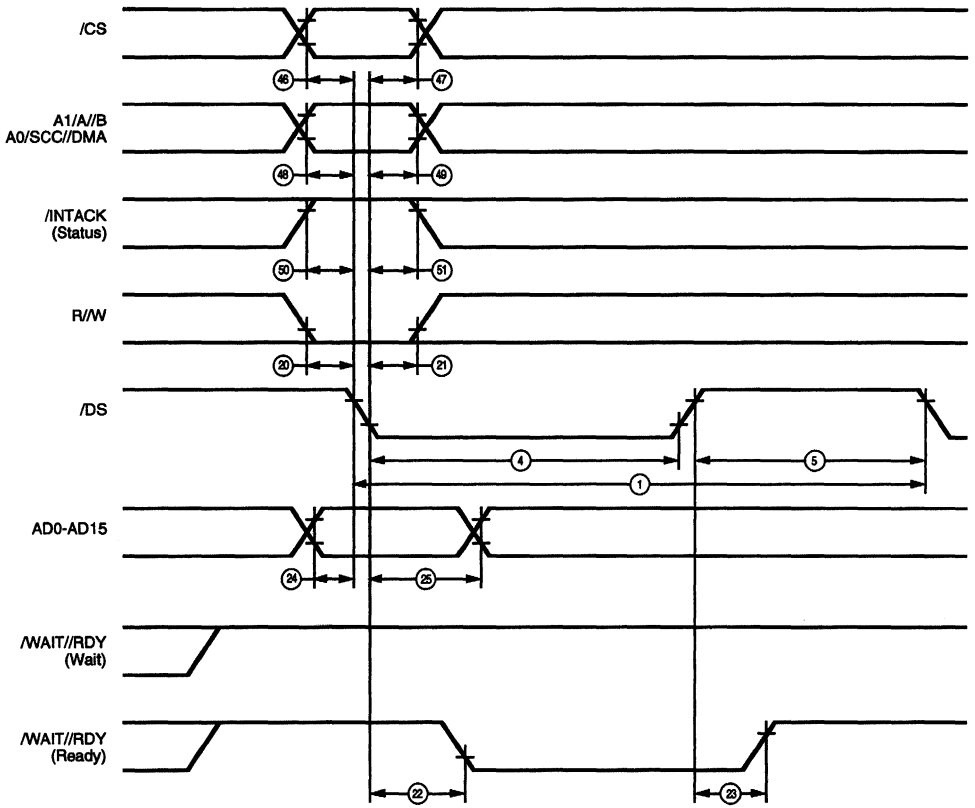


Figure 33. Non-multiplexed /DS Write Cycle

AC CHARACTERISTICS (Continued)

Timing Diagrams

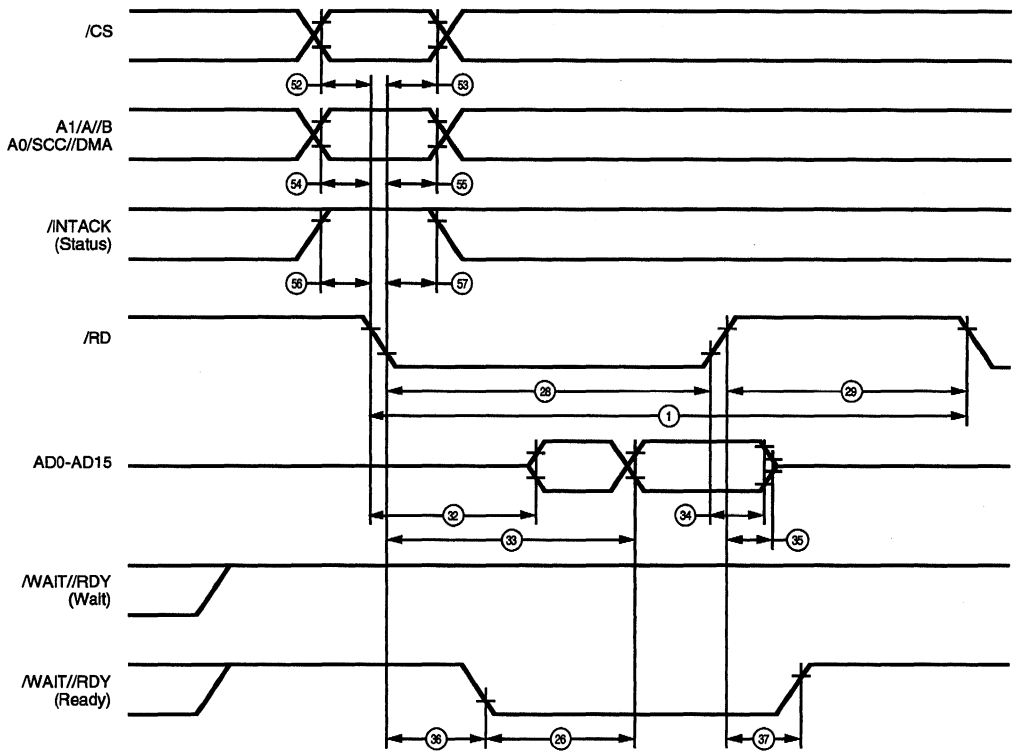


Figure 34. Non-multiplexed /RD Read Cycle

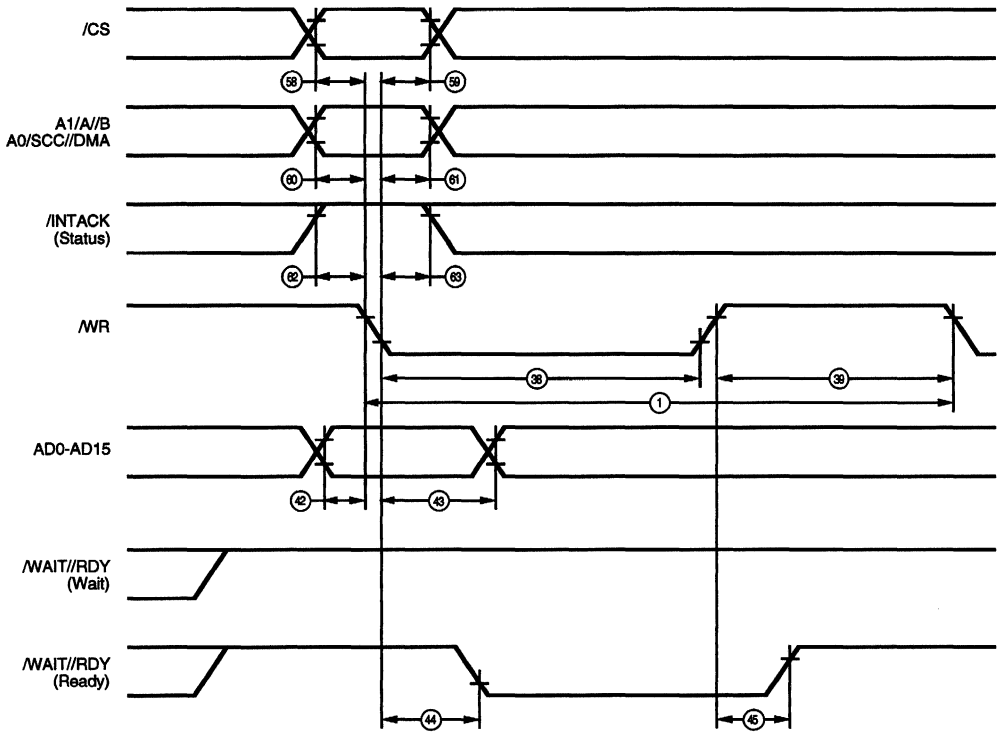


Figure 35. Non-multiplexed /WR Write Cycle

AC CHARACTERISTICS (Continued)

Timing Diagrams

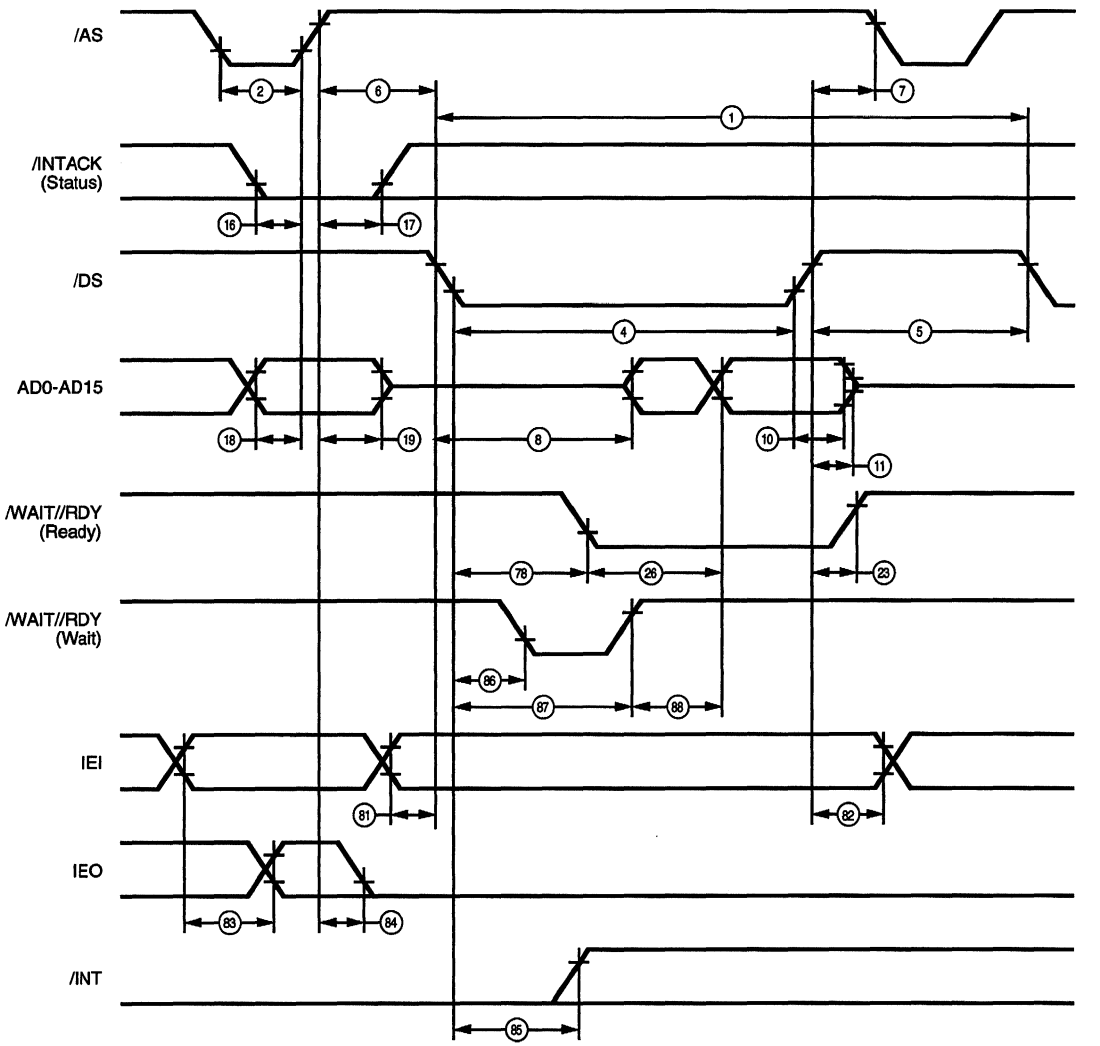


Figure 36. Multiplexed /DS Status INTACK Cycle

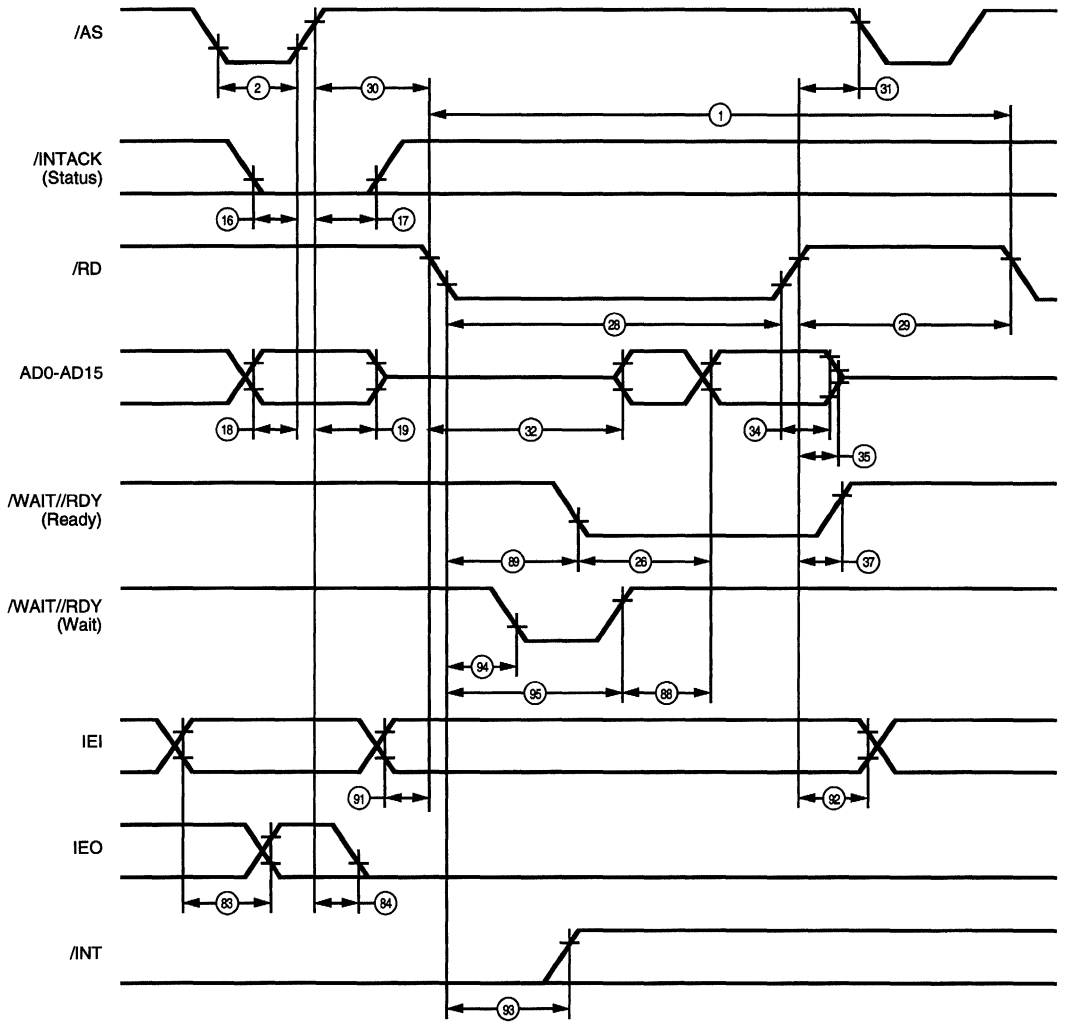


Figure 37. Multiplexed /RD Status INTACK Cycle

AC CHARACTERISTICS (Continued)

Timing Diagrams

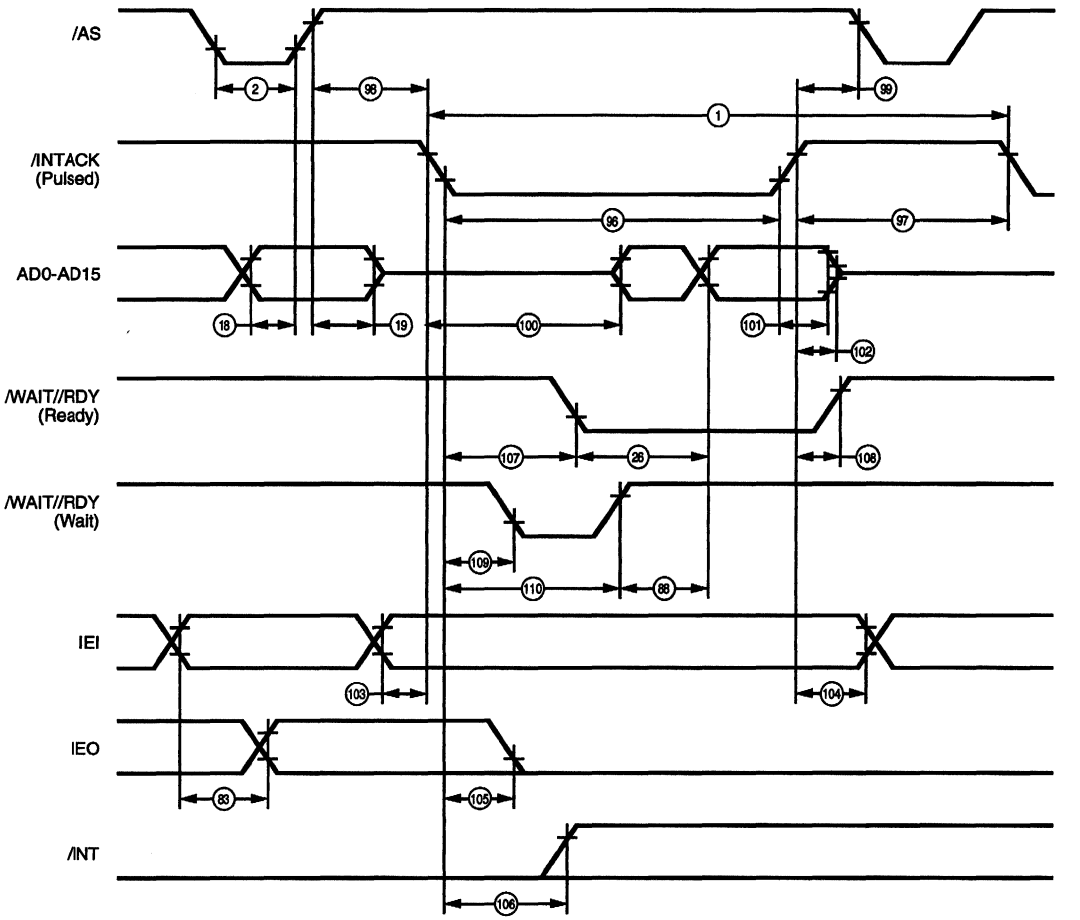


Figure 38. Multiplexed Pulsed INTACK Cycle

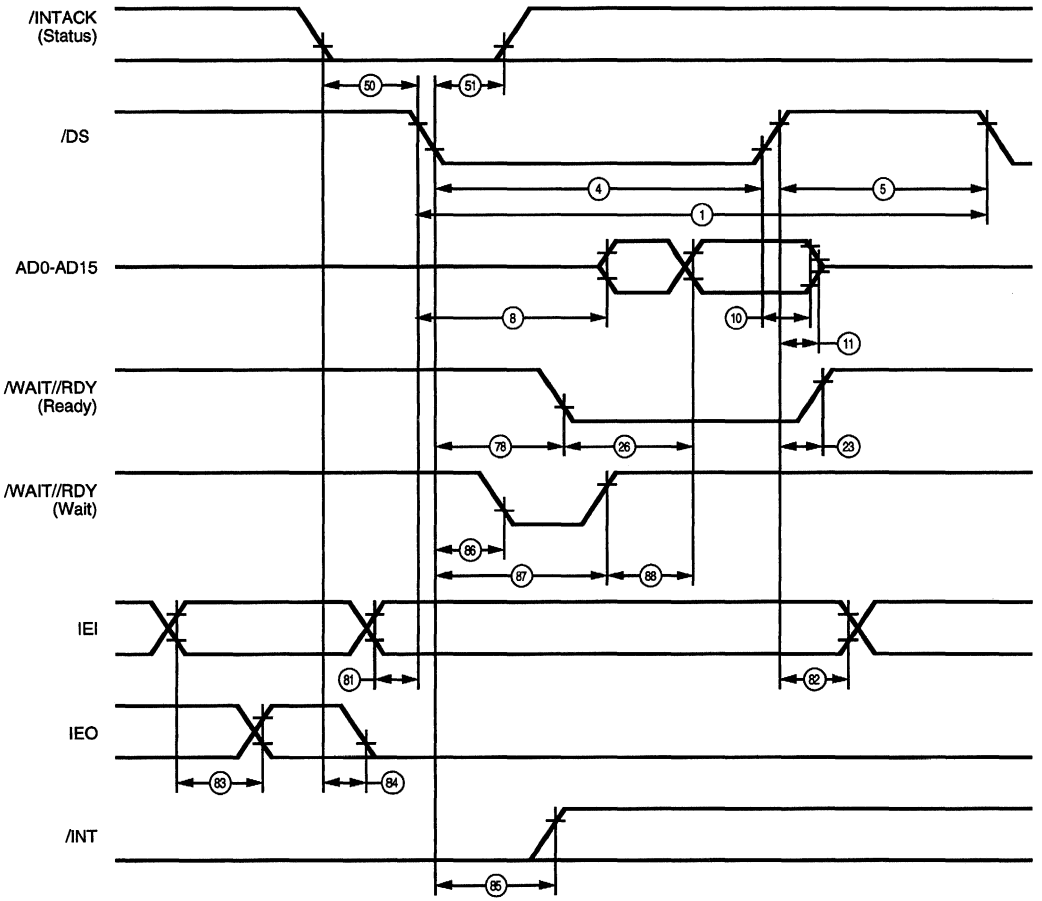


Figure 39. Non-multiplexed /DS INTACK Cycle

AC CHARACTERISTICS (Continued)

Timing Diagrams

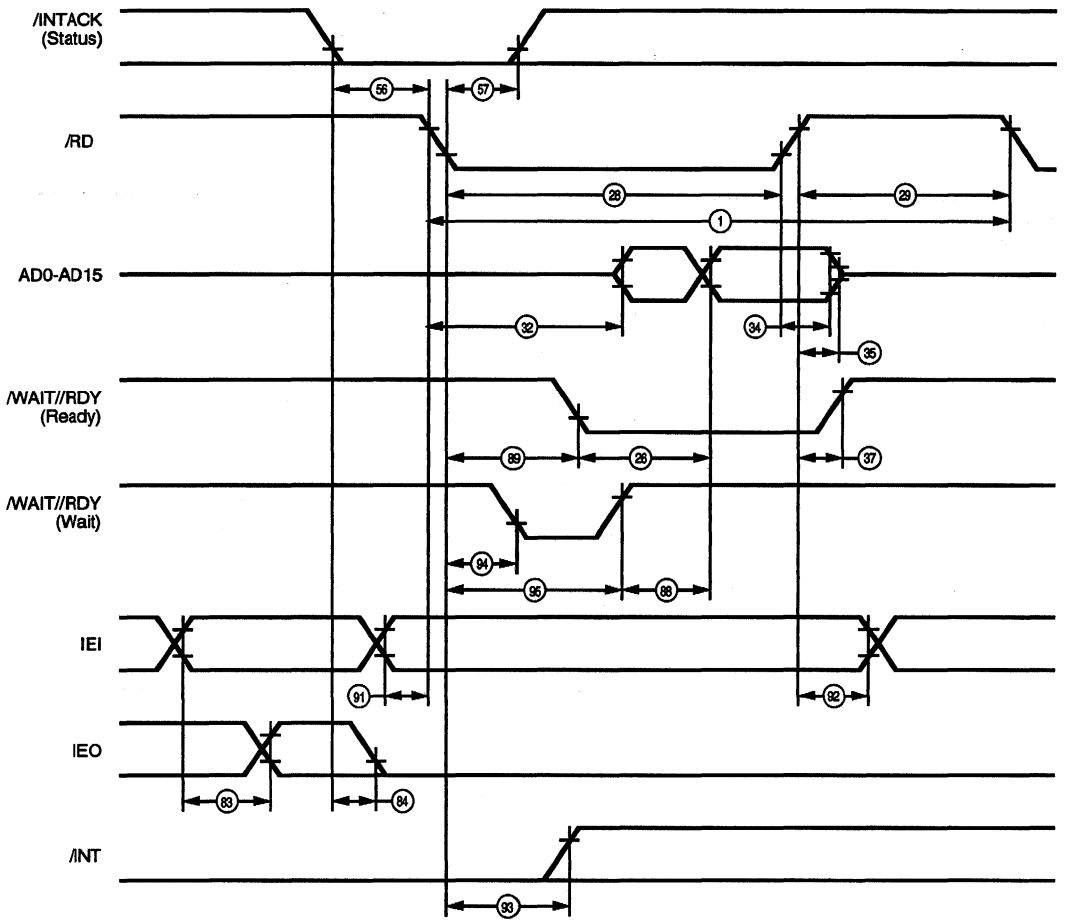


Figure 40. Non-multiplexed /RD Status INTACK Cycle

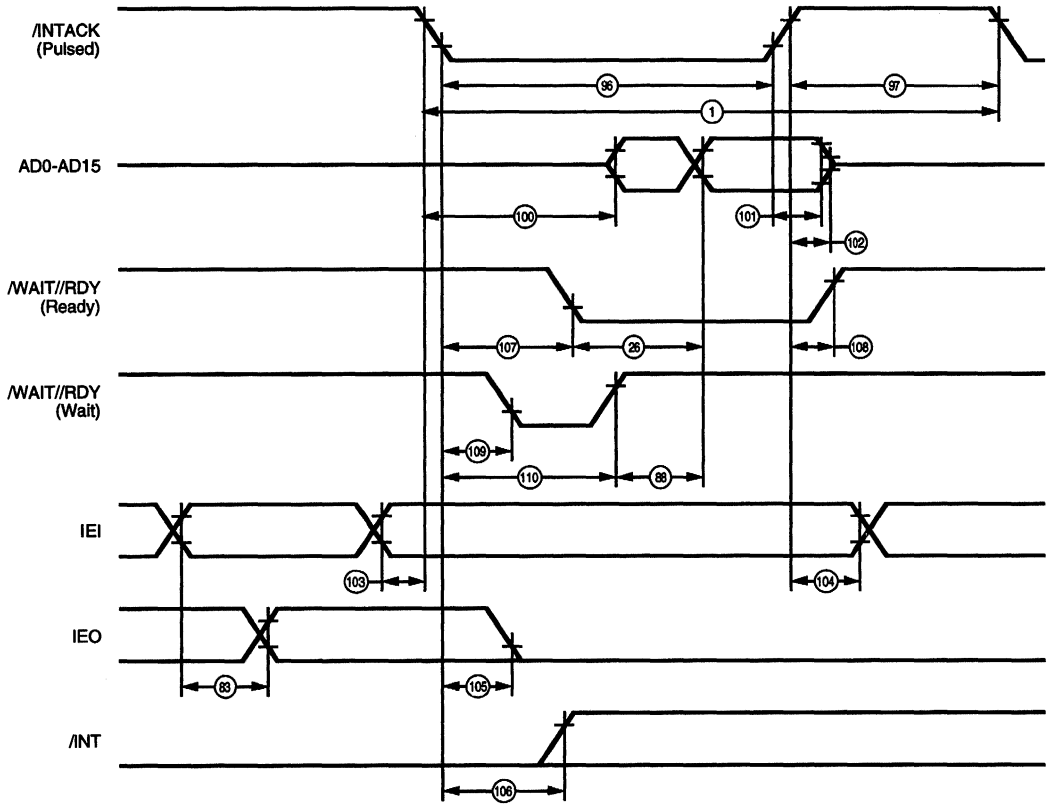


Figure 41. Non-multiplexed Pulsed INTACK Cycle

AC CHARACTERISTICS (Continued)
Timing Diagrams

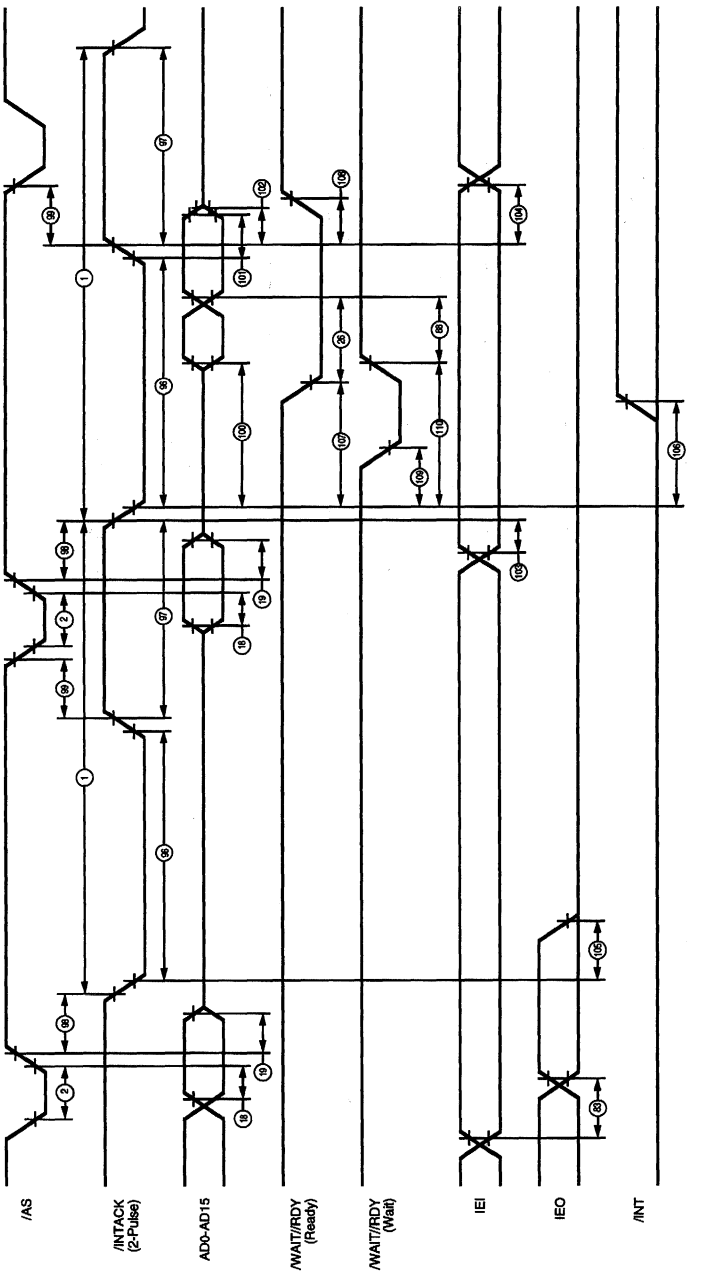


Figure 42. Multiplexed Double-Pulse INTACK Cycle

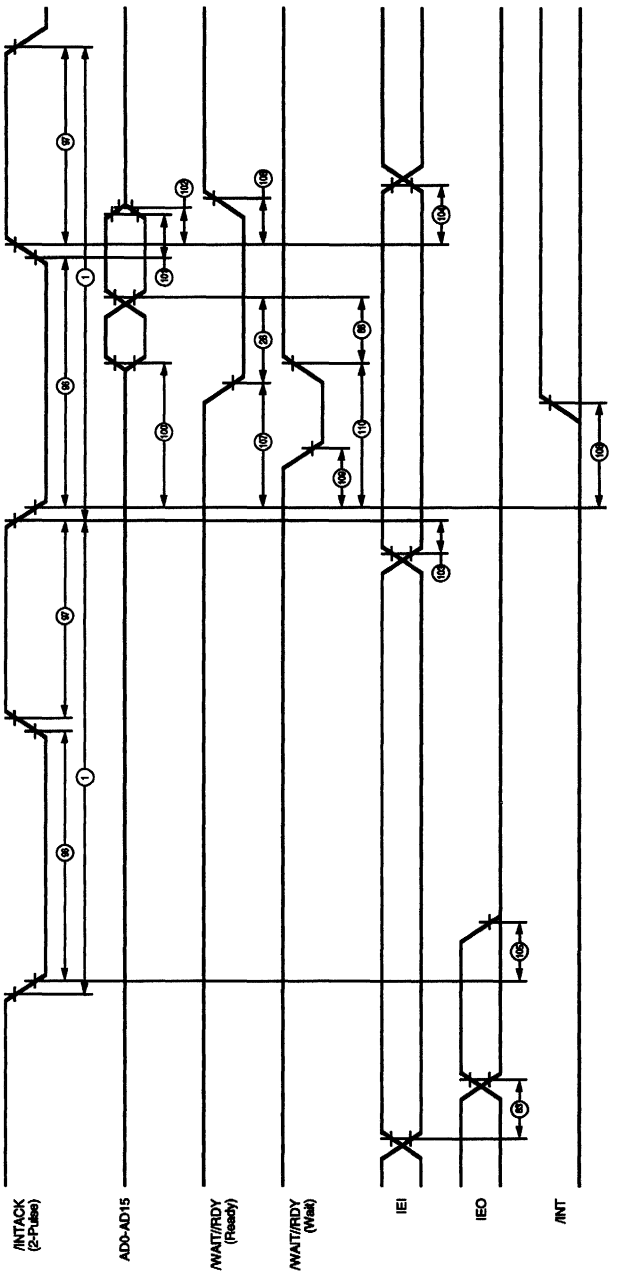


Figure 43. Non-multiplexed Double-Pulsed INTACK Cycle

AC CHARACTERISTICS (Continued)
Timing Diagrams

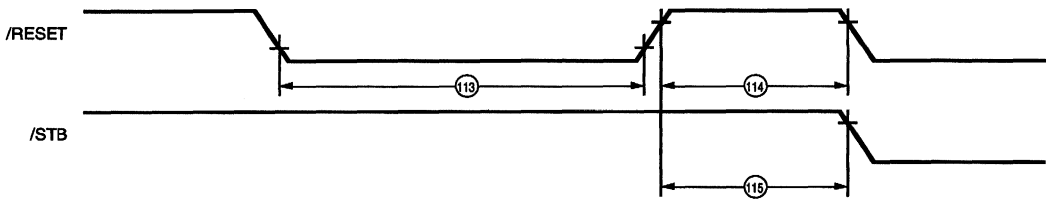


Figure 44. Reset

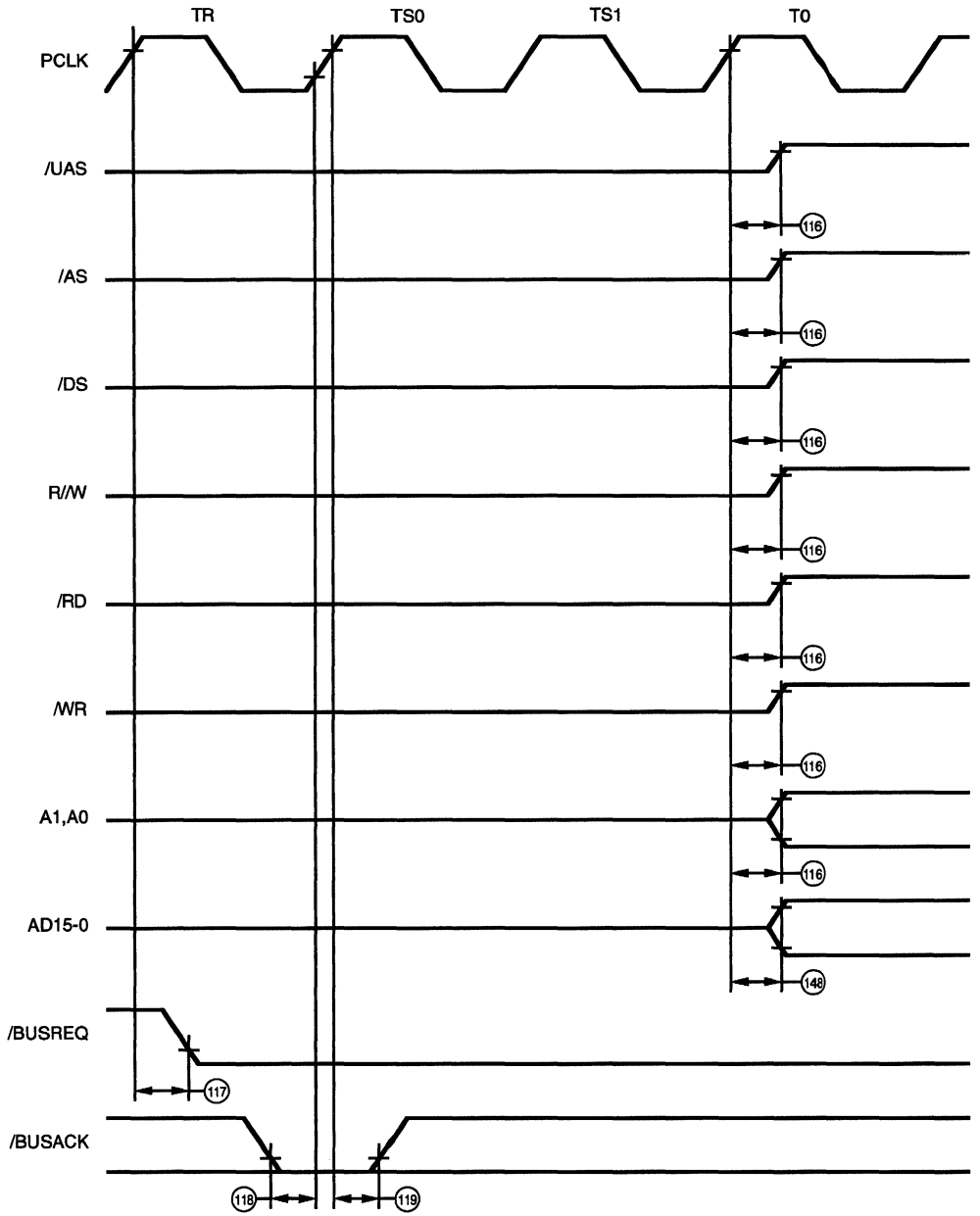


Figure 45. Z16C35 Start-up

AC CHARACTERISTICS (Continued)
Timing Diagrams

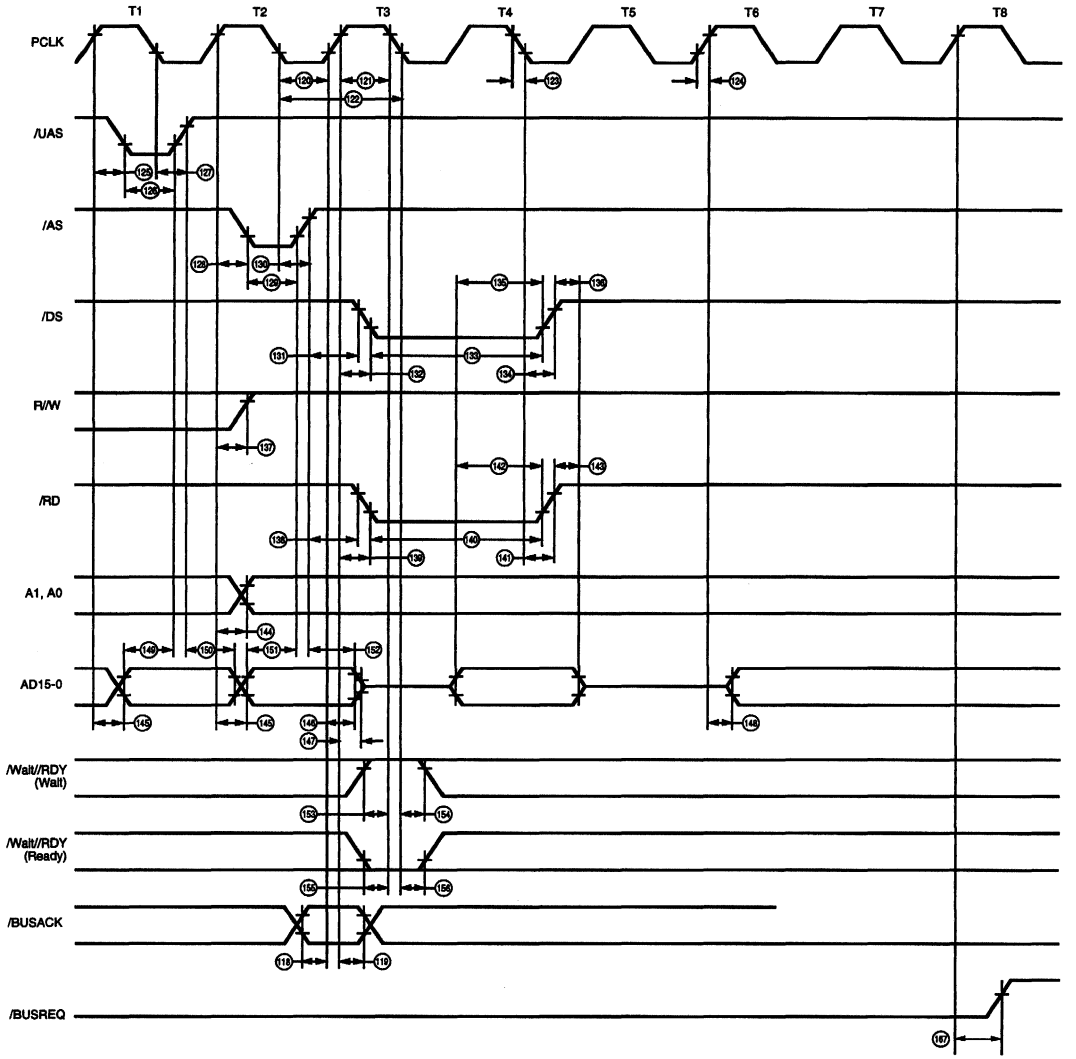


Figure 46. Z16C35 Memory Read

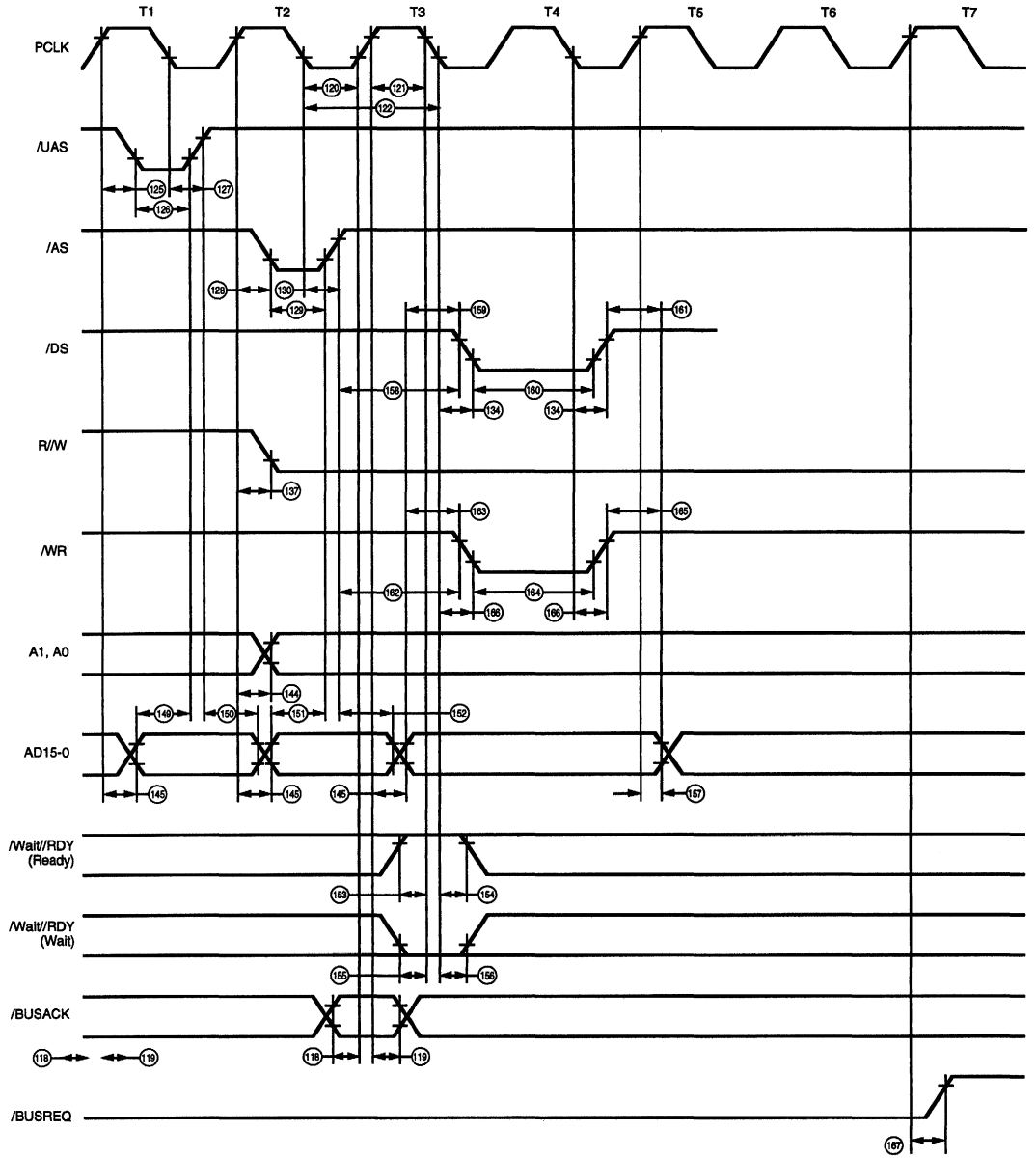


Figure 47. Z16C35 Memory Write

AC CHARACTERISTICS (Continued)

Timing Diagrams

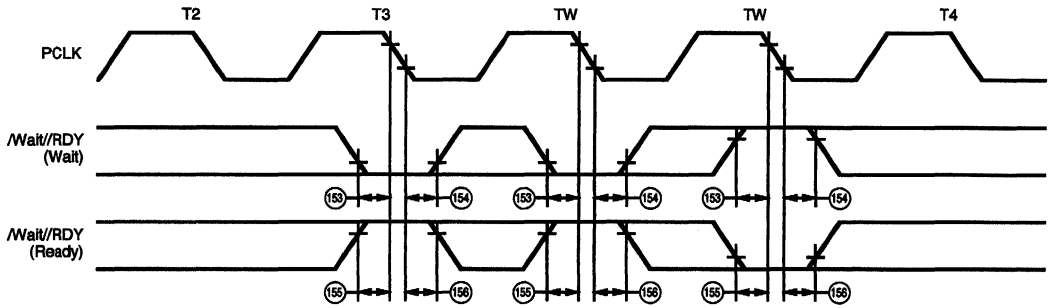


Figure 48. Wait and Ready Timing

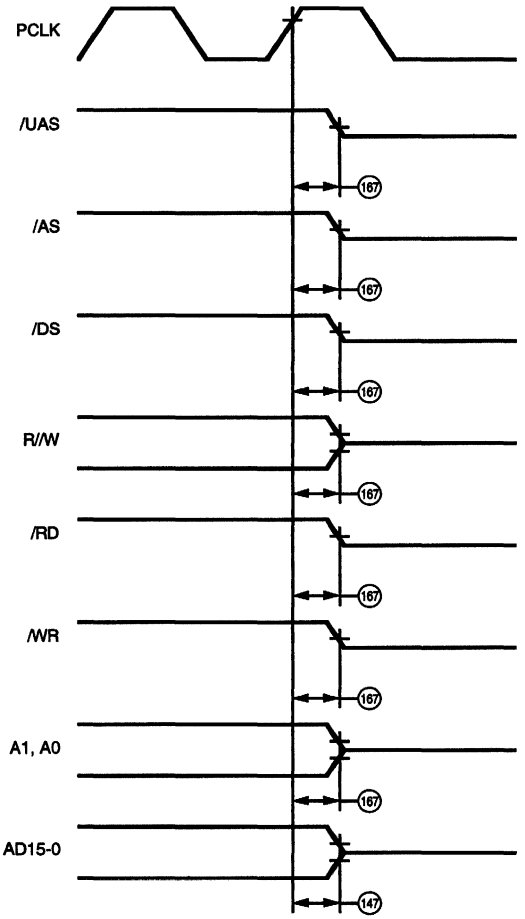


Figure 49. BUS Release

AC CHARACTERISTICS (Continued)
General Timing

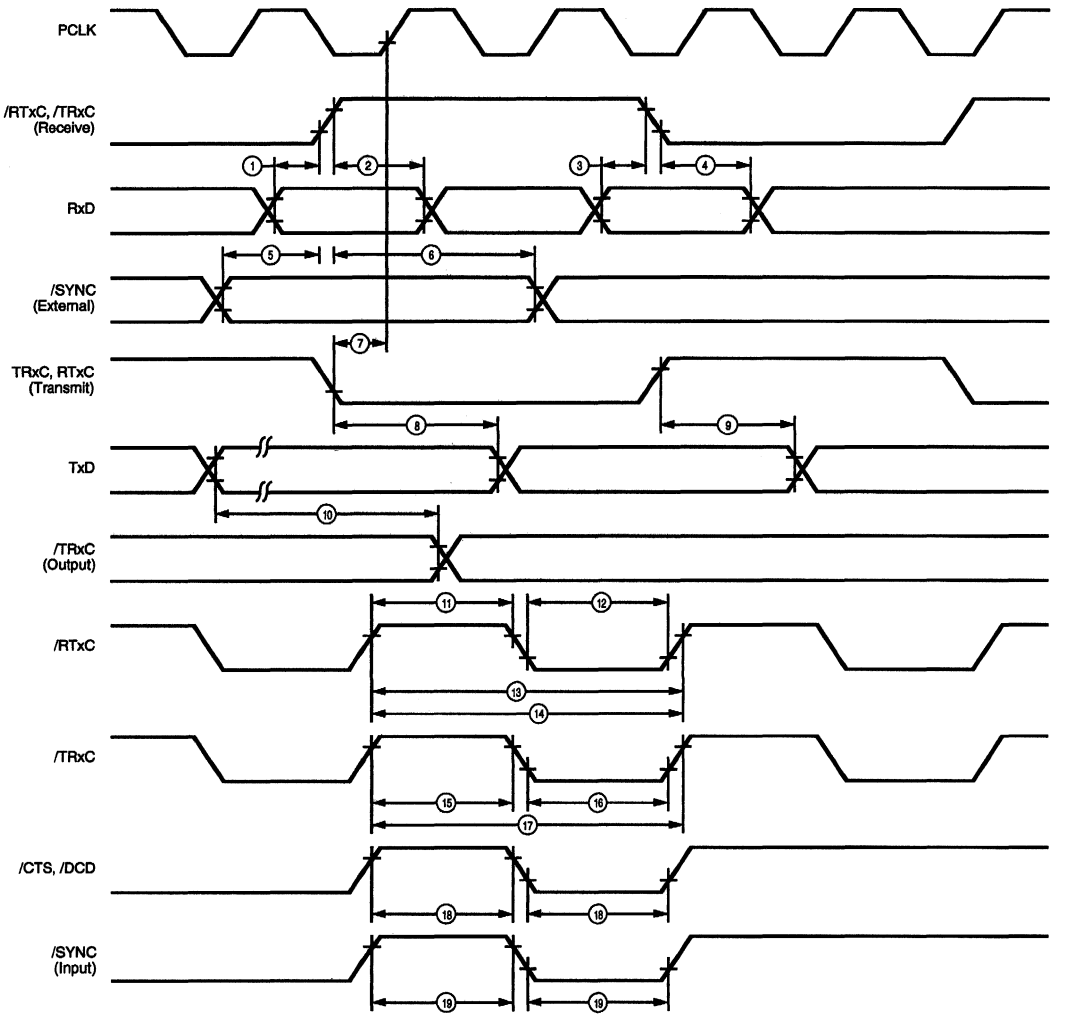


Figure 50. Z16C35 General Timing

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.
[8] Timings in nanoseconds.

AC CHARACTERISTICS (Continued)
System Timing

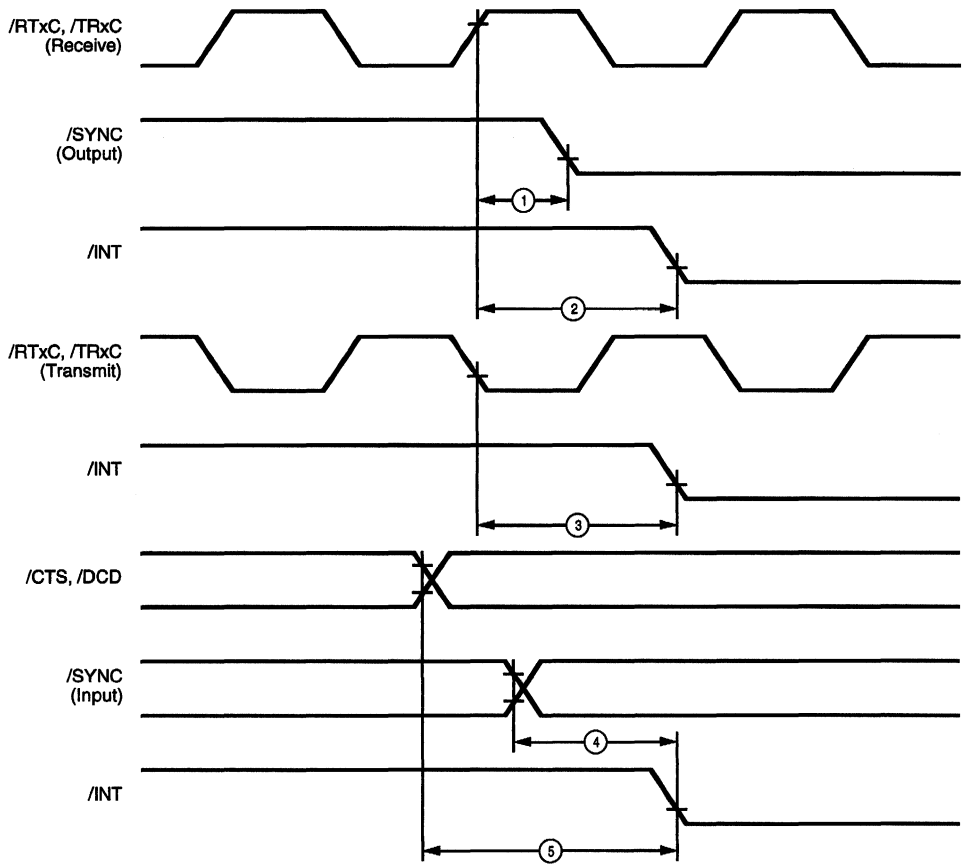
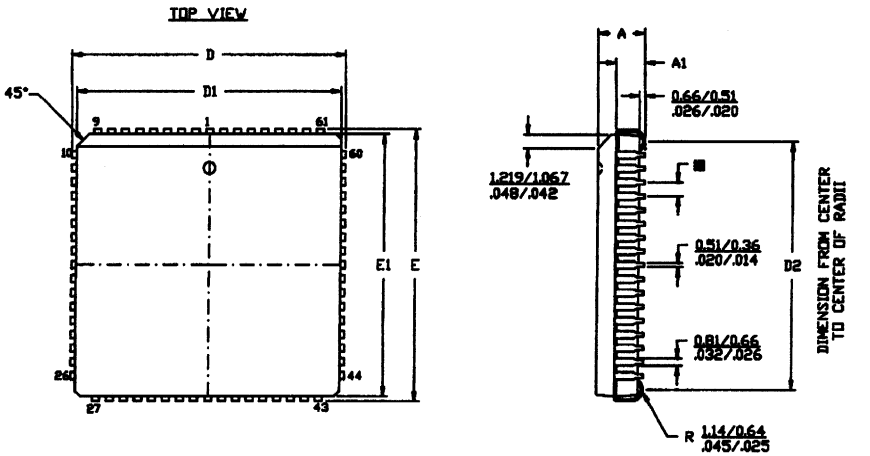


Figure 51. Z16C35 System Timing

No	Symbol	Parameter	10 MHz Min	10 MHz Max	16 MHz Min	16 MHz Max	Notes[3]
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.
[3] Units equal to TcPc.

PACKAGE INFORMATION



NOTES:

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

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

68-Pin PLCC Package Diagram

ORDERING INFORMATION

Z16C35	10 MHz	16 MHz
68-Pin PLCC	Z16C3510VSC	Z16C3516VSC

Package
V=Plastic Leaded Chip Carrier
C=Ceramic DIP
L=Ceramic LCC
F=Plastic Quad Flat Pack

Temperature
E=-40°C to +105°C
S=0°C to +70°C

Speeds
10=10 MHz
16=16 MHz

Environmental
C=Plastic Standard
D=Plastic Stressed
E=Hermetic Stressed

Example:

Z 16C35 34 A S C is a Z16C35, 34 MHz, PLCC, 0°C to +70°C, Plastic Standard Flow

- C — Environmental Flow
- S — Temperature
- A — Package
- 34 — Speed
- 16C35 — Product Number
- Z — Zilog Prefix

7

ZILOG

APPLICATION NOTE

INTERFACING 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 is a supplement to 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 DC-8293-02)
- Motorola 16-Bit Microprocessor User's Manual. 3rd ed. Englewood Cliffs, N.J., Prentice-Hall, Inc. 1979.
- Monolithic Memories Bipolar LSI 1982 Databook

8

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 show 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 cycles: 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 non-multiplexed 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 registers: Control and Data. Depending on the peripheral, registers are selected with either the A0, A1, A//B, or D//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.

A0, A1, A//B, D//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 A1 and A2 Address bus lines.

/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 (/AS). The active Low /AS guarantees that the 68000 addresses are valid.

D7-D0. The Z8500 Data bus can be directly connected to the lowest byte (D7-D0) 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 all the peripherals are connected, as shown in Figure 1.

/INT. The interrupt request pins for each peripheral in the daisy chain can be wire-ORed and connected to the 68000's ILPN pins. The 68000 has seven interrupt levels that can be encoded into the ILP0, ILP1, and ILP2 pins. Multiple 68000 interrupt levels can be implemented by using a multiplexer like the 74LS148.

/INTACK. The INTACK pin signals the peripherals that an Interrupt Acknowledge cycle is occurring. The following equation describes how /INTACK is generated:

$$/INTACK = /(FC0)\bullet(FC1)\bullet(FC2)\bullet(AS)$$

The 68000 FC2-FC0 are status pins that indicate an Interrupt Acknowledge when they are all High. They should be ANDed with inverted /AS to guarantee their validity. The /INTACK signal must be synchronized with PCLK to guarantee setup and hold times. This can be accomplished by changing the state of /INTACK on the falling edge of PCLK. If the /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.

/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 /RD is generated:

$$/RD = /[\ (R//W0)\bullet(AS) = RESET]$$

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 or /RD sets the Interrupt Under Service (IUS) bits during an Interrupt Acknowledge cycle.

Table 1. Z8500 and 68000 Pin Functions

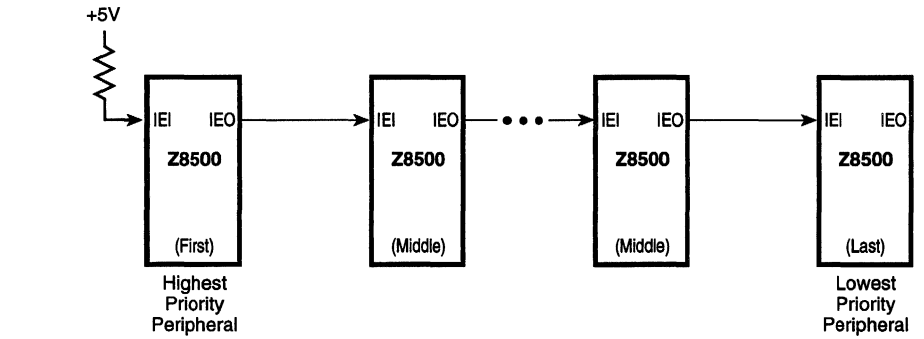
68000 Signals		Z8500 Signals	
Mnemonic	**Function**	**Mnemonic**	**Function**
A23-A1	Address Bus	A0,A1,A//B, D//C	Register Select
/AS	Address Strobe	/CE	Chip Enable
CLK	68000 clock (8 MHz)	D7-D0	Data Bus
D15-D0	Data Bus	IEI,IEO	Interrupt Daisy Chain
/DATACK	Data Transfer Acknowledge		Control
FC2-FC0	Processor Status	/INT	Interrupt Request
ILP2-ILP0	Interrupt Request	/INTACK	Interrupt Acknowledge
R/W	Read/Write	PLCK	Peripheral Clock
/VMA	Valid Memory Address	/RD	Read Strobe
/VPA	Valid Peripheral Address	/WR	Write Strobe

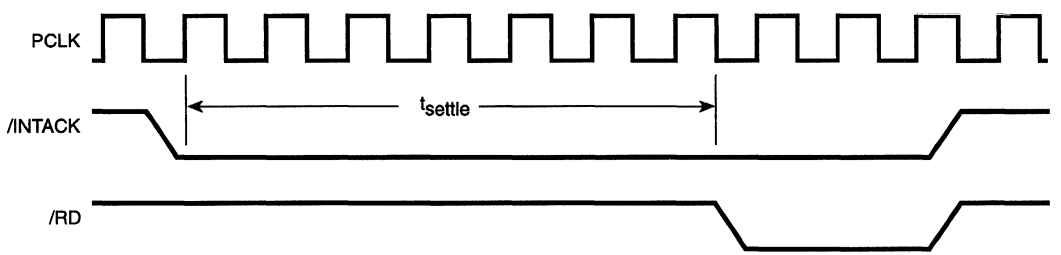
*The register select pins on each peripheral have different names.

/WR. This signal strobes data into the peripheral. A data-to-write setup time requires that data be valid before /WR goes active Low. The equation for generating the /WR strobe is made up of two components: an active reset and a normal Write cycle, as shown in the following equation:

$$/WR = /[/(R/W)\bullet(AS) + RESET]$$

Forcing /RD and /WR simultaneously Low resets the peripherals.





t_{settle} (ns)

Peripheral (4 MHz)	First	Middle	Last
CIO	350	150	100
FIO	350	150	100
SCC	250	120	120

Figure 1. Peripheral Interrupt Daisy Chain

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 through three types of bus cycle: Read, Write, and Interrupt Acknowledge. The discussion that follows pertains to the 4 MHz peripherals, but the 6 MHz devices have similar 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 /RD line. A setup time of 80 ns from the time the register select inputs (A//B, C//D, A0, A1) are stable to the falling edge of /RD guarantees that the proper register is accessed. The access time specification is usually measured from the falling edge of /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.

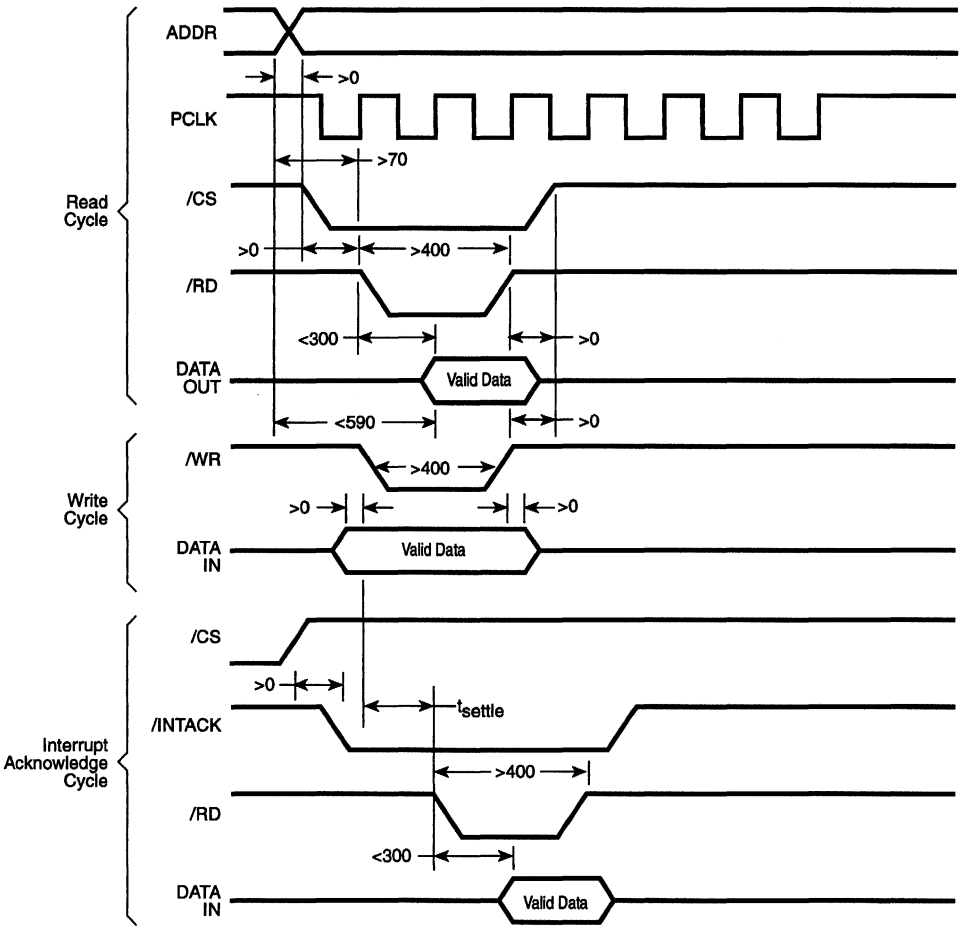


Figure 2. Z8500 Interface Timing (4 MHz)

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 /WR is required. The data must be valid prior to the falling edge of /WR. The /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 interrupt, the timing shown in Figure 2 should be observed. An interrupt sequence begins with an /INT going active because of an interrupt condition. The CPU acknowledges the interrupt with ana /INTACK signal.

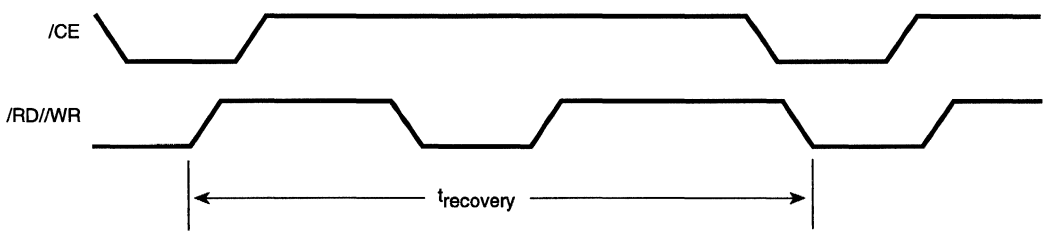
A daisy chain settle time (dependent upon the number of devices in the chain) ensures that the interrupts are prioritized. The falling edge of /RD causes the IUS bit to be set and enables a vector to go out on the bus.

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 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 /CE is deselected when Read or Write occurs.

8



Peripheral (4 MHz)	Recovery Time
CIO	Greater than 3 PCLK cycles or 1000 ns
FIO	Greater than 1000 ns
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

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 (/VPA) cycle meets the Z8500 timing requirements. In this mode the 68000 accepts nonvectored interrupts. The second example uses the Data Transfer Acknowledge (/DTACK) pin. This interface allows faster operation and makes use of the Z8500's 8-bit vectored interrupts. The third example also uses a /DTACK cycle and is similar to the second, except the external logic is integrated into a single chip, the PAL20 x 10 programmable array logic.

EXAMPLE 1: A TTL INTERFACE USING A VPA CYCLE

The 68000 has a special input pin, Valid Peripheral Address (/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.

The timing is shown in Figure 4. Figure 5 shows how the hardware can be implemented. PCLK is generated by dividing down the 68000 CLK. /RD, /WR, and /INTACK are simply ANDed 68000 signals. The worst-case daisy-chain settle time is 450 ns. Connecting /INT to IPL0 generates a level 1 interrupt. The internal registers are accessed by A0, A1, D//C, and A//B, which can be the 68000 lowest order addresses.

Functional Description

/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//W)]

WR = [(CE)•(VMA)•/(R//W)]

INTACK = [(FCO)•(FC1)•(FC2)•(AS)]

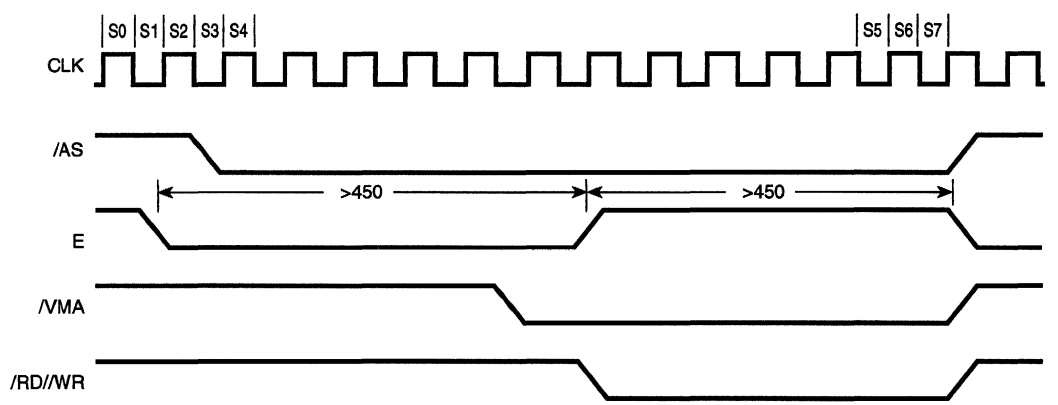


Figure 4. /VPA Cycle Timing

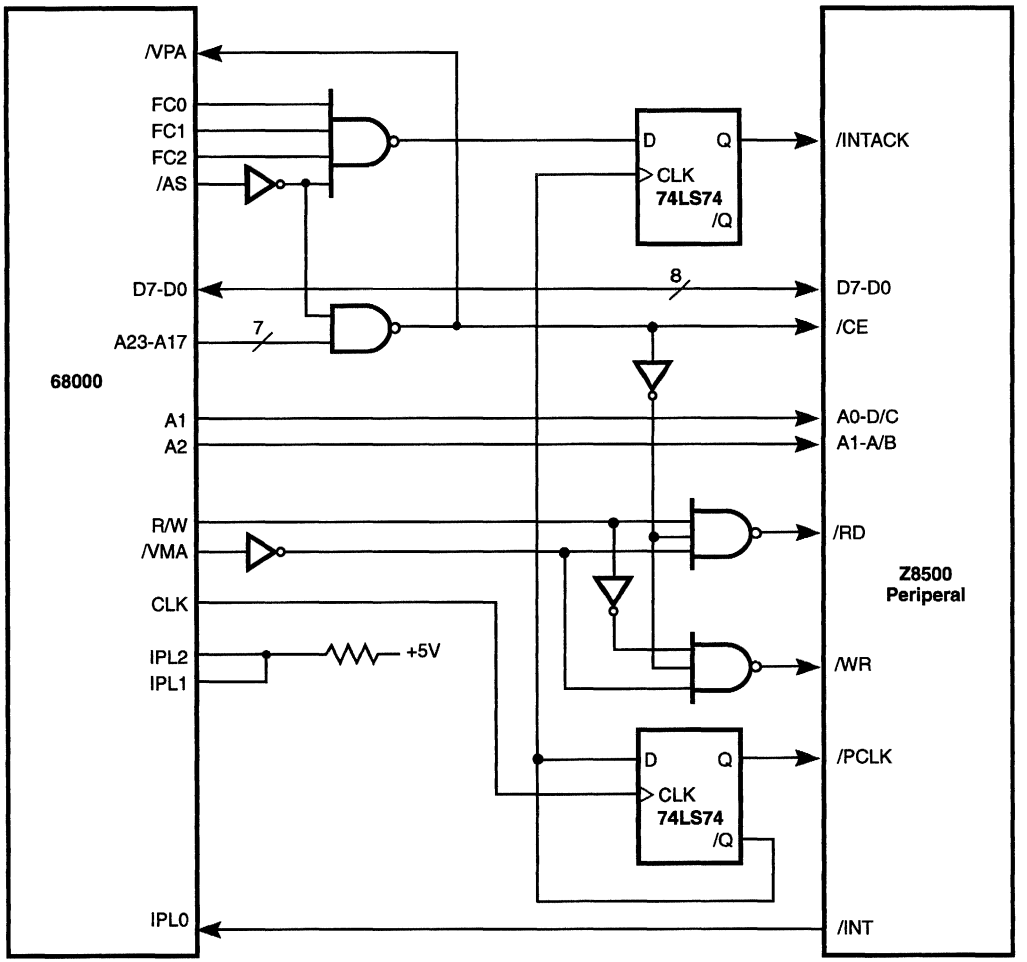


Figure 5. Interface Using the /VPA Cycle

EXAMPLE 2: A TTL INTERFACE USING DTACK CYCLES

Using the 68000 Data Transfer Acknowledge (/DTACK) cycle is a second way of interfacing to the Z8500 peripherals. The 68000 inserts Wait states until the /DTACK input is strobed Low to complete the transfer. In addition to generating the control signals, the interface logic must also generate /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, QA (Figure 7) is set High for one PCLK cycle and then reset. This pulse is shifted through the QA-QH outputs and is used to generate /RD, /WR, and /DTACK signals. Some of the extra Wait states can be eliminated by tapping the Shift register sooner (e.g., QC).

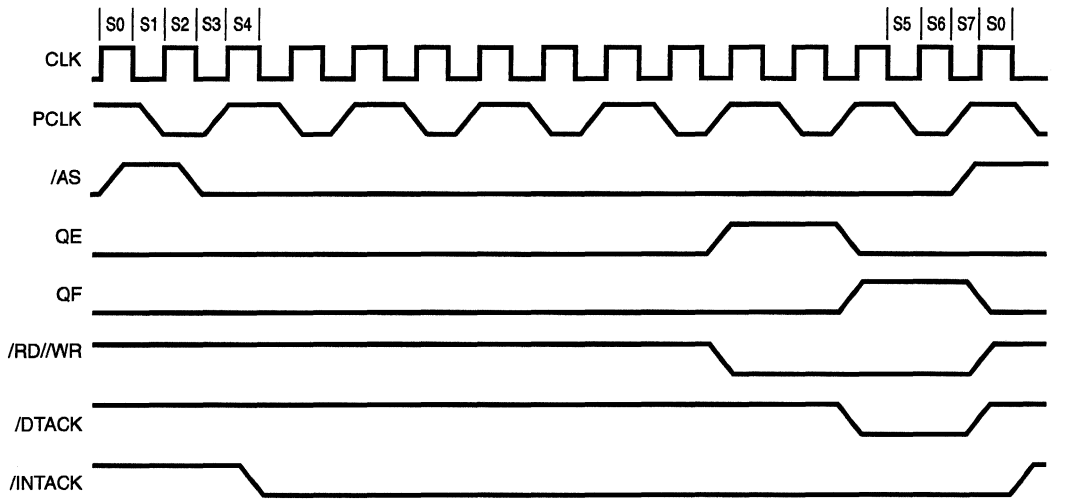


Figure 6. Timing for /DTACK Interface

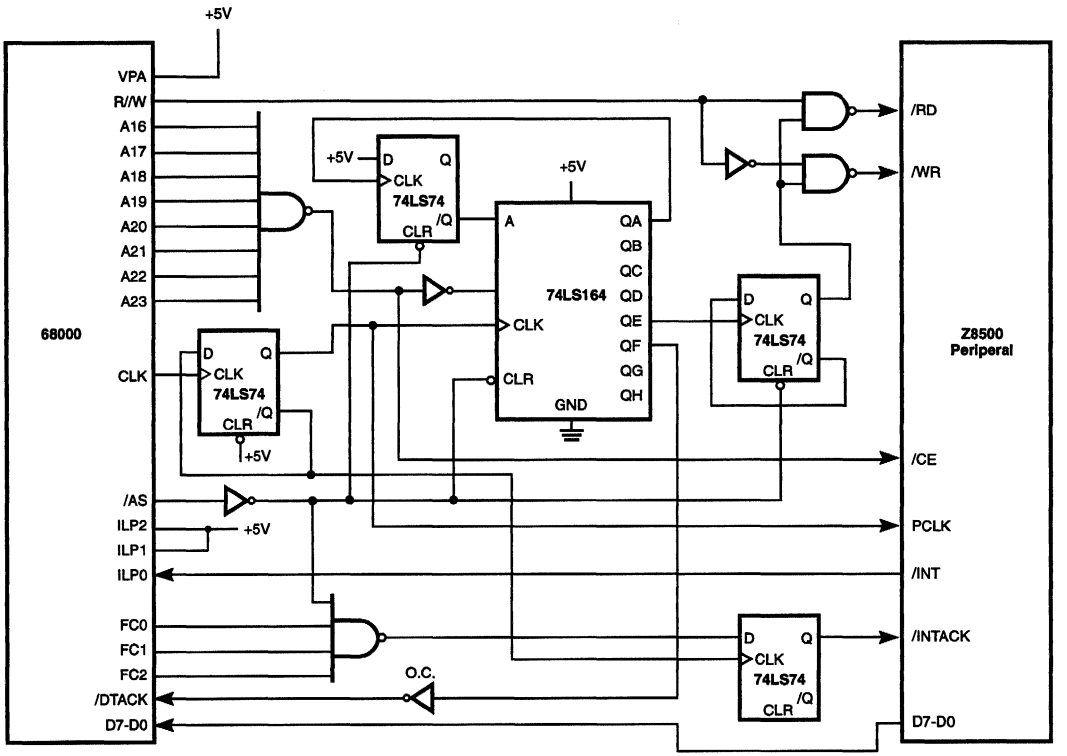


Figure 7. Hardware Diagram for /DTACK Interface

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 /DTACK to inform the 68000 of a completed data transfer cycle. This allows the 68000 to use the peripheral's vectored interrupts.

Functional Description

Figure 8 shows the PAL's pin functions. The PAL generates five control signals, of which four (/WR, /RD, C0, and /INTACK) go to the Z8500 and one (/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 C0 is the least significant bit and C3 is the 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 /AS goes active. The counter goes back to toggling when /AS goes inactive. CYC goes active Low at the same time the counter starts counting down. The equations in Table 2 can be entered into a development board to program the PAL.

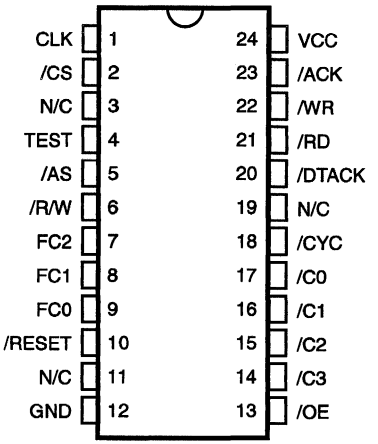


Figure 8. PAL Pinout

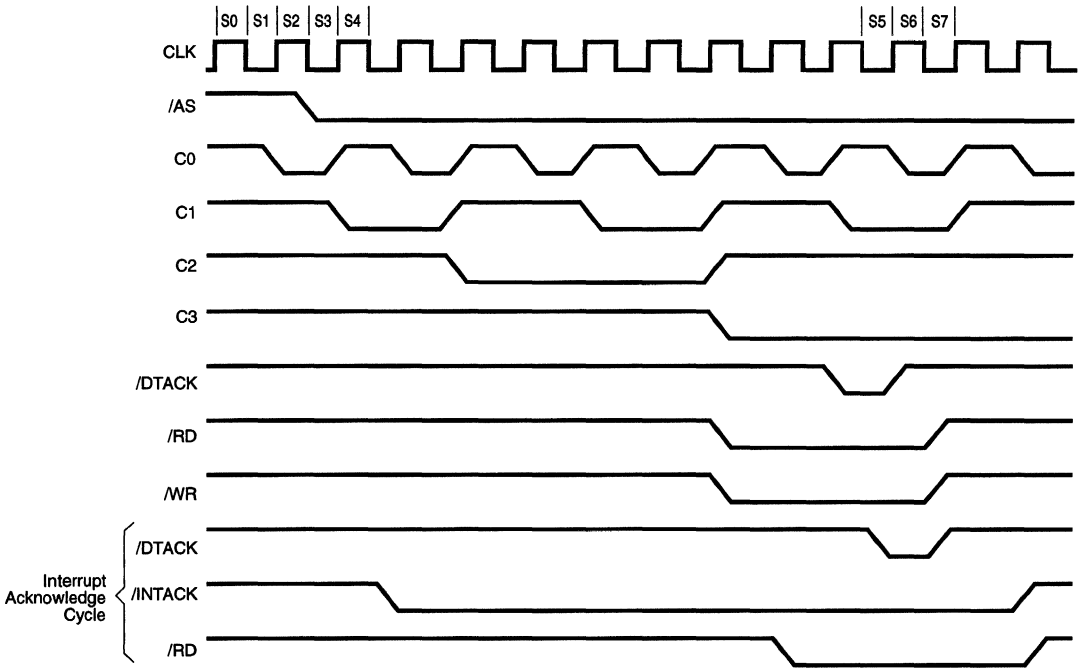


Figure 9. PAL Interface Timing

Functional Description (Continued)

Table 2. PAL Equations

PAL 20X10 **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

Hardware Diagram

The hardware diagram of the PAL interface is shown in Figure 10. The 68000 signals CLK, /CS, /AS, R/W, FC0, FC1, and FC2 are used to generate the Z8500 control signals. The control signals are synchronous with the rising edge of the 68000's CLK. TEST and /OE must be grounded. /CS is used to enable /DTACK, /RD, and /WR as shown in the equations. The Z8500 /INT is connected to the /ILP0, which generates a 68000 level 1 interrupt. The peripherals are memory-mapped into the highest 64 Kbyte block of memory, where A32-A17 equals "FFH". Addresses A6-A4 are used to select the peripheral; A3-A1 select the internal registers. Table 3 shows the peripherals memory map.

Table 3. Peripheral Memory Map

Peripheral	Register	Hex Addr
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

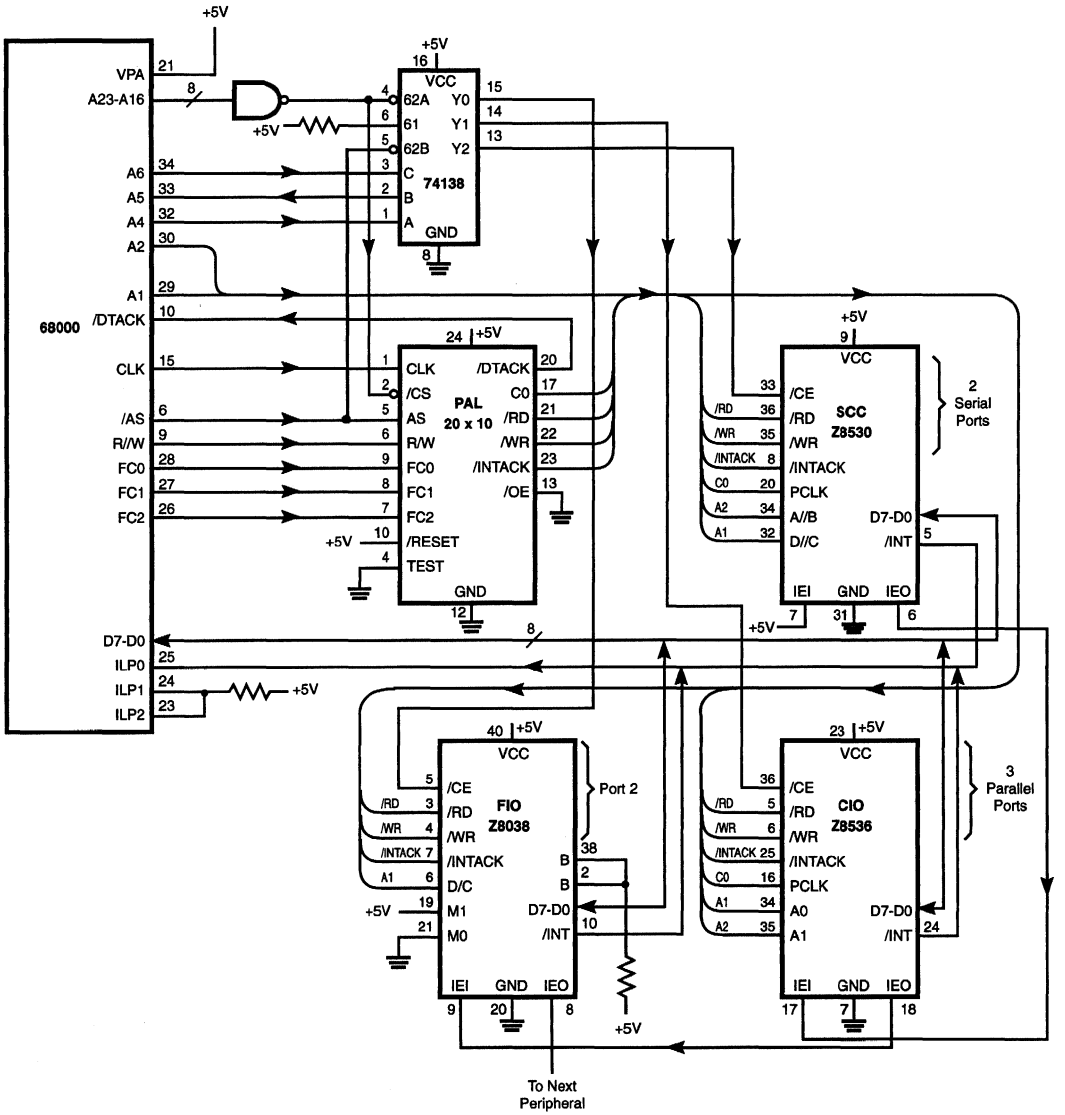


Figure 10. PAL Hardware Diagram

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 /RD and/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 01H.

The SCC's Read cycle can be verified by reading the bits in RR0. Bits D2 and D6 are set to 1 and bits D0, D1, and D7 are 0. Bits D5-D3 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 00H 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 (/INTACK) cycle consists of several steps. First, the peripheral makes an Interrupt Request (/INT) to the CPU. When the processor is ready to service the interrupt, it initiates an Interrupt Acknowledge (/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 through 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 non-multiplexed 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

SCC IN BINARY SYNCHRONOUS COMMUNCATIONS

INTRODUCTION

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.

8

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 (/W//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: Mono-sync, 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.

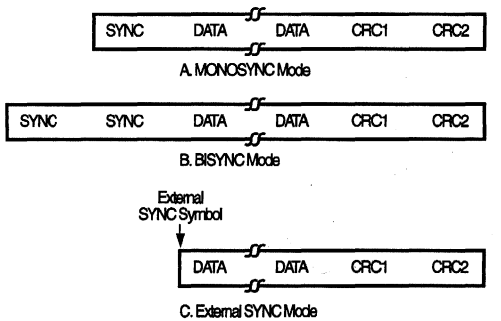


Figure 1. Synchronous Modes of Communication

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 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 stop lower in interrupt priority compared to the other device.

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.

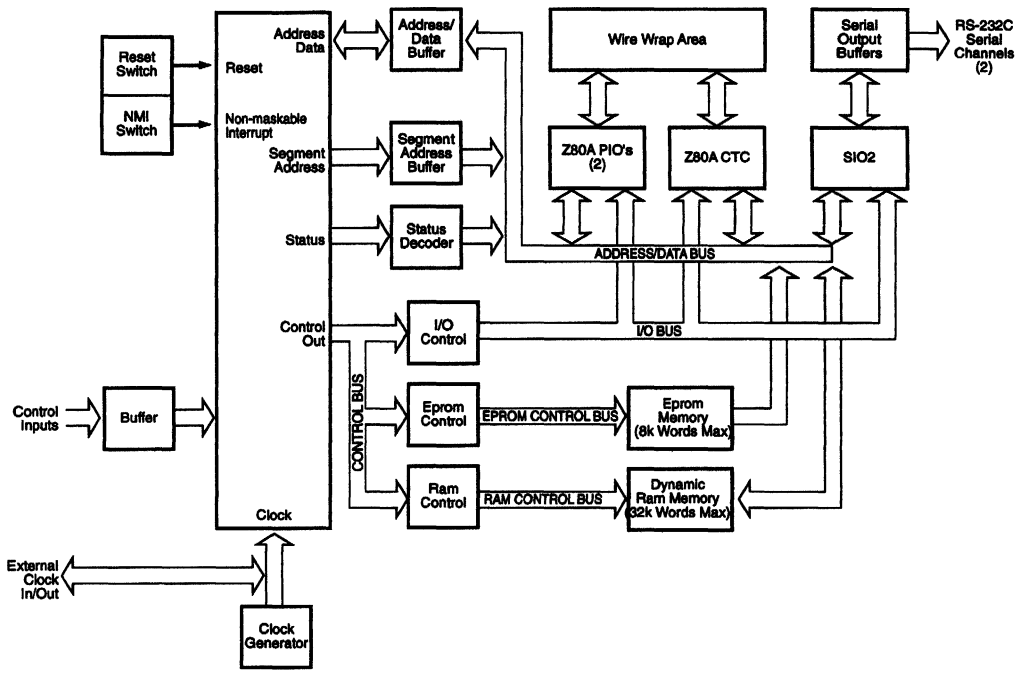


Figure 2. Block Diagram of Z8000 DM

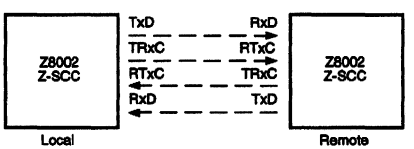


Figure 3. Block Diagram of Two Z8000 Development Modules

SYSTEM INTERFACE (Continued)

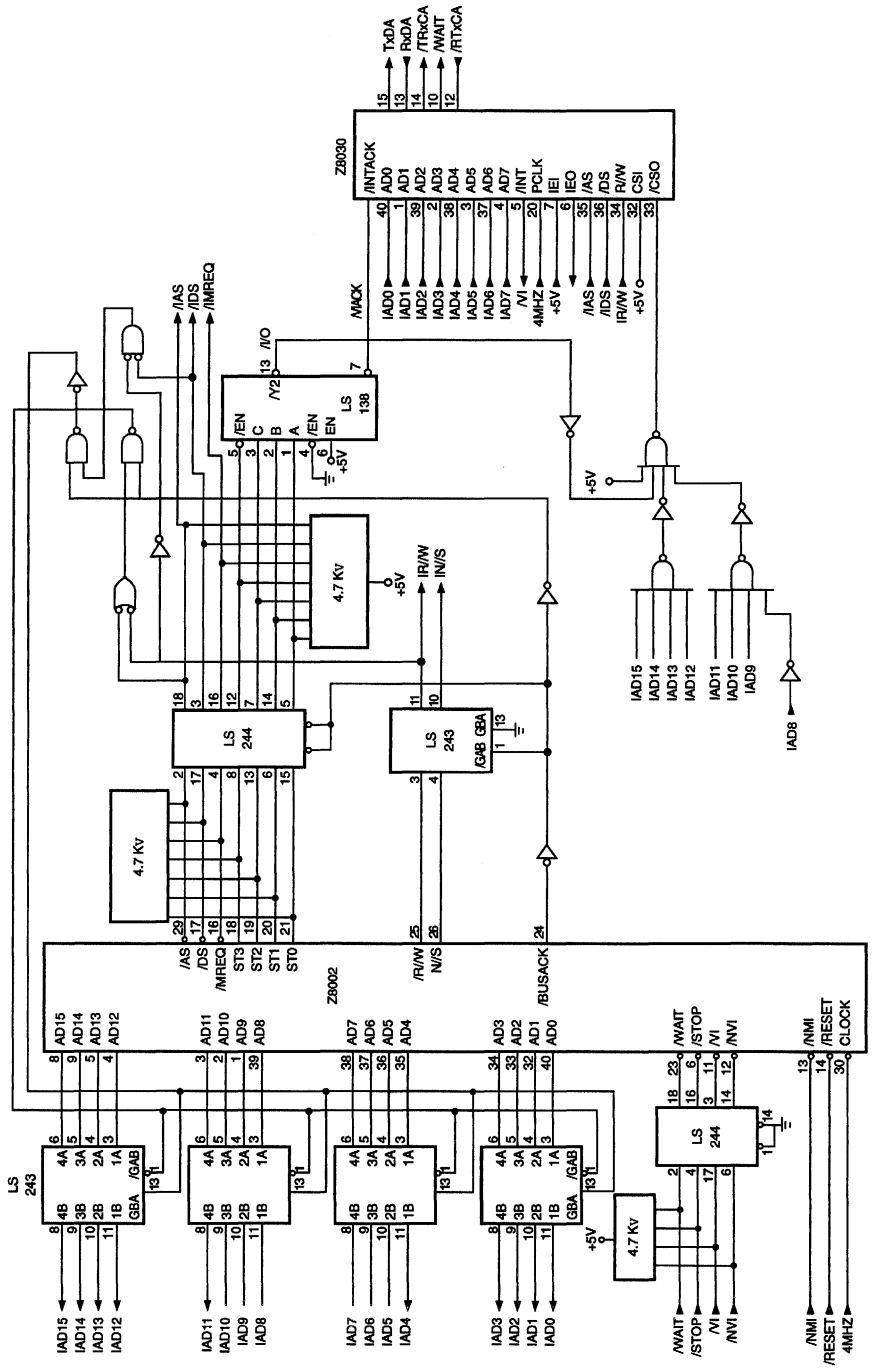


Figure 4. Z8002 with SCC

When the Z8002 CPU uses the lower half of the Address/Data bus (AD0-AD7 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 AD5-AD1. 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 (/CS0) is derived by decoding I/O address 'FE' hex from lines AD15-AD8 of the controller. The Read/Write registers are automatically selected by the Z-SCC when internally decoding lines AD5-AD1 in Shift Left mode. To select the Read/Write registers automatically, the Z-SCC decodes lines AD5-AD1 in Shift Left mode. The register map for the Z-SCC is depicted in Table 1.

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 XI 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.

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 (Continued)

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//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	Rx8 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 Receive 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 AREAS!
                    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      RI(#%IC),R0 !FCW VALUE(%5000) AT %441C FOR VECTORED!
000C  001C
                                                   !INTERRUPTS!
000E  7600                     LDA     R0,REC
0010  00F4'
0012  3310                     LD      RI(#%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 ENTRY	INIT	PROCEDURE	
0634	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	@RI, @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	CO		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	!WRIO=CRC PRESET ZERO, NRZ,16 BIT SYNC!
0054	OC		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=NT 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	IA		BVAL.	2*13	
005F	00		BVAL	0	!WR13= UPPER TC=01
0060	1C		BVAL	2*14	
0061	03		BVAL	%03	!WRI4=BRG ON, ITS SRC=PCLK!
0062	1E		BVAL	2*15	
0063	00		BVAL	%00	!WRI5=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	CI		BVAL	&CI	IWR3=RX 8 BITS/CHAR, REC ENABLE!
0068	02		BVAL	2*1	
0069	08		BVAL	%C1	!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,#428        !WAIT ON RECV.!
006C 3A86                   OUTB        WR0A+2,RL0
0070 FE23
0072 6000                   LDB         RL0,%A8
0074 00AB
0076 3A86                   OUTB        WR0A+2,RL0      !ENABLE WAIT 1ST CHAR,SP.COND. INT!
0078 FE23
007A 2101                   LD          Rl,#RR0A+16
007C FE31
007E 3Cl8                   INB         RL0,@R1         !READ STX CHARACTER!
0080 C8C9                   LDB         RL0,#%C9
0082 3AB6                   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,#I
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 TBUP

OOA6		GLOBAL ENTRY	TRANSMIT	PROCEDURE	
00A6	2102		LD	R2, #TBUF	!PTR TO START OF BUFFER!
00AB	0028'				
00AA	C86C		LDB	RL0, #%6C	
00AC	3AB6		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	3AB6		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	@RI, @R2,R0	!SEND START OF TEXT!
00D2	0010				
00D4	C8C0		LDB	RL0, #%C0	
00D6	3AB6		OUTB	WR0A, RL0	!RESET TxUND/EOM LATCH!
00D8	FE21				
00DA	2100		LD	R0, #COUNT-1	
00DC	000B				
00DE	3A22		OTIRB	@RI, @R2, R0	!SEND MESSAGE!
00E0	0010				
00E2	C804		LDB	RL0, #%04	
00E4	3EI8		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	3AB6		OUTB	WR0A+10, RL0	!DISABLE TRANSMITTER!
00F0	FE2B				
00F2	9E0B		RET		
00F4		END TRANSMIT			

RECEIVE INT. SERVICE ROUTINE

00F4		GLOBAL ENTRY	REC	PROCEDURE	
00F4	93F0		PUSH	@RI5, 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, @RI5	
011C	7B00		IRET		
0IIE		END REC			

8

SPECIAL CONDITION INTERRUPT SERVICE ROUTINE

```
011E            GLOBAL      SPCOND      PROCEDURE
                ENTRY
011E  93F0                  PUSH        @RI5, R0
0120  3A84                  INB         RL0, RR0A+2     !READ ERRORS!
0122  FE23
                            !PROCESS ERRORS!
0124  C830                  LDB         RL0, #%30
0126  3A8B6                 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  C0D1                  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, @RI5
013E  7B00                  IRET

0140            END SPCOND

                END BISYNC
```

0 errors
Assembly complete

ZILOG SCC
Z8030/Z8530
QUESTIONS AND ANSWERS

March 1992

This document contains the most commonly asked questions about the Zilog SCC. They are divided into five sections:

- Hardware Considerations
- Interrupts and Polling
- Asynchronous Mode
- Synchronous Mode
- Miscellaneous Questions

HARDWARE CONSIDERATIONS

This section includes questions and answers on the hardware interface, the clocks, the FIFO, special modes (Local Loopback, DPLL, Manchester), and internal timing consideration.

Hardware (Includes DMA Interface)

Q. What is the SCC transistor count?

A. Approximately 6000 gates, or 18,000 transistors.

Q. What is the difference between the Z8030 and the Z8530?

A. The Z8030 and Z8530 are packaged from the same die. The multiplexed bus (Z8030) or non-multiplexed bus (Z8530) version of the chip is selected at packaging time by an internal bonding option.

Q. Can /AS be active only when the Z8030 is being accessed and High all other times?

A. Since the interrupt pending bits (IPs) are updated on address strobes, interrupts will not occur unless /AS is continuous.

Q. How do /WR and /CE interact on the Z8530?

A. /WR and /CE are ANDed to enable a transparent latch. Data is latched on the falling edge when both /CE and /WR go Low.

Q. How many register pointers does the Z8530 have?

A. The SCC has only one register pointer for both channels. The SIO (Z844X) has two, one for each channel.

Q. Do you have to write to the pointer with the Z8530 to access WR0 or RR0?

A. No. Both registers are accessed automatically without first writing to the pointer.

Q. Does /CE (/CS) have to be High during an interrupt acknowledge cycle?

A. No.

Q. Does the SCC support full duplex DMA?

A. The SCC allows full duplex DMA transfers by using the DTR/REQ and W/REQ as two separate DMA control lines for transmit request and receive request on each channel.

Q. When using full duplex DMA, how do you program W/REQ?

A. W/REQ should be programmed for receive and DTR/REQ pin should be programmed for transmit.

Q. Can both channels make simultaneous DMA requests?

A. Yes.

Q. Do you have to reset the SCC in hardware?

A. No. A software reset is the same as a hardware reset, (WR9 CO). It also does not matter whether the Z8030 is in shift right or shift left mode because the address is the same in either.

Hardware (Includes DMA Interface)

Q. Do you need to clear the reset bit in WR0 after a software reset?

A. The reset is clocked with PCLK; so it must be active during reset.

Q. How long after a hardware reset should you wait before programming the SCC.

A. Four PCLKs.

Q. Why does the SCC initialization require that the External Status Interrupts be reset twice?

A. Because of the possibility of noise causing an interrupt pending bit (IP) to be set. The second reset guarantees that the latch is clear. If the latch is closed high and the external signal is low, the first reset will open the latch at the high-to-low transition causing an interrupt.

Clocks

Q. Does PCLK have to have a 50% duty cycle?

A. The duty cycle doesn't have to be 50% as long as the minimum specification is met.

Q. Can the SCC PCLK be stretched?

A. Yes, as long as the pertinent specification is met. However, this could cause a problem if PCLK is used to generate the bit rate.

Q. The bit rate generator is driven from what sources?

A. It may be driven from the RTxC pin or PCLK, or from a crystal.

Q. How do you connect a bit rate crystal to the SCC?

A. A crystal can be connected between RTxC and SYNC to supply the clock if the SCC is programmed for WR11 D7-1.

Q. What is the crystal specification?

A. It is a fundamental, parallel resonant crystal. For further details see the "Design Considerations Using Quartz Crystals with Zilog's Components" Application Note.

Q. Can RTxC on both channels be driven from the same crystal.

A. No. A separate crystal should be used for each channel. The crystal should be connected between /SYNC and RTxC of the respective channels. The alternate solution may be to use crystal on one channel and reflect the clock out of the TRxC output and feed it into another channel.

Q. How do you select a crystal frequency?

A. Time constant: (Clock Frequency/2 x Bit rate x clock factor) - 2 (the SCC Technical Manual assumed a clock factor of one in the formula). Two examples are given below:

For PCLK = 3.6864 MHz			For PCLK = 3.9936 MHz		
Bit Rate	**TC**	**Error**	**Bit Rate**	**TC**	**Error**
38400	46	-	19200	102	-
19200	94	-	9600	206	-
9600	190	-	7200	275	12%
7200	254	-	4800	414	-
4800	382	-	3600	553	.06%
3600	510	-	2400	830	-
2400	766	-	2000	996	.04%
1200	1534	-	1800	1107	.03%
			1200	1662	-
			600	3326	-
			300	6654	-
			150	13310	-
			134.5	14844	.0007%
			110	18151	.0015%
			75	26622	-
			50	39934	-

Q. Why are there different Clock factors?

A. These clock factors enable the SCC to sample the center of the data cell. In the 16x mode, the SCC divides the bit cell into 16 counts and samples on count 8. Clock factors are generally only used with Asynchronous modes.

Q. How is the error in the receive/transmit clock reduced?

A. The ideal way to reduce this error is by adjusting the crystal frequency such that only an integer value of TC is yielded when the equation is used.

Q. What are the maximum transfer rates?

A. The following table shows the PCLK rates (in bps).

	4 MHz	6 MHz	8 MHz	10 MHz	16 MHz	20 MHz
Asynchronous mode:						
External clock						
6x mode (no BRG)	250K	375K	500K	635K	1M	1.25M
BRG						
16x mode (TX + 0)	62.5K	93.75K	125K	156.5K	250K	312.5K
Synchronous mode:						
Using external clock	1M	1.5M	2M	2.5M	4M	5M
Using DPLL, FM encoding	250K	375K	500K	625K	1M	1.25M
Using DPLL, MRZ/NRZI encoding	125K	187.5K	250K	312.5K	500K	625K
Using DPLL, FM, BRG	62.5K	93.75K	125K	156.25K	250K	312.5K
Using DPLL, NRZ/NRZI, BRG	32.25K	46.88K	62.5K	78.125K	125K	156.25K

Q. Can the maximum transfer rate using an external clock be achieved?

A. Yes, but it is not trivial. In order to achieve the maximum rate on transmit, the SCC should have a dedicated processor or DMA. For example, at a 1 MHz rate, a byte must be loaded into the SCC every 8 microseconds. To achieve the maximum rate on receive, requires that the receive clock and the SCC PCLK be synchronized. (RTxC to PCLK setup time at maximum rate in the Product Specification.) It is probably easier to use a slightly faster PCLK SCC, or back off slightly from the maximum rate.

FIFO

Q. How do you avoid an overrun in the received FIFO?

A. The receive buffer must be read before the recently received data character on the serial input is shifted into the receive data FIFO. This FIFO is three bytes deep. Thus, if the buffer is not read, the fifth character just arrived causes an overrun condition. There is no bit that can be set or reset to disable the buffering.

Q. What happens when you read an empty FIFO?

A. You read the last character in the buffer.

Q. When the FIFO gets locked due to an error condition, can it still receive?

A. The SCC continues to receive until an overrun occurs.

Q. Assuming that there are characters available in the FIFO, what happens to them if the receiver goes into the hunt mode?

A. They will remain in the FIFO until they are either read by the CPU or DMA, or until the channel is reset.

Special Modes (Local, Loopback, DPLL, Manchester)

Q How are the Local, Loopback, and Auto Echo modes implemented?

A. The TxD and RxD pins are connected through drivers. If both modes are simultaneously enabled, then Auto Echo overrides.

Q. Can the SCC transmit when the Auto Echo mode is enabled?

A. No, the transmitter is logically disconnected from the TxD pin.

Q. Can the Digital Phase Lock Loop (DPLL) be used with NRZ?

A. The DPLL simply generates the receive clock which is the same for both NRZ and NRZI.

Q. Do you have to use the DPLL with NRZI and FM encoding?

A. If the DPLL is not used, a properly phased external clock must be supplied.

Q. What is the error tolerance for the DPLL?

A. The DPLL can only tolerate a + or - 1/32 deviation in frequency, or about 3%.

Q. Can you receive and transmit between two channels on the same SCC using the DPLL to generate both the transmit and receive clocks?

A. To transmit and receive using the same clock, you need to divide the transmit clock by 16 or 32 to be the same rate for transmitting and receiving, because the DPLL requires a divide-by-16 or -32 on the receiver, depending on the encoding. An external divide-by-16 or -32 is required, and can be connected by outpouring the bit rate generator on the /TRxC pin, through the external divide circuit, and back in the /RTxC pin as an input to the transmitter.

Q. How fast will Manchester be decoded?

A. The SCC can decode Manchester data by using the DPLL in the FM mode and programming the receiver for NRZ data. Hence, the 125K bit/s is the maximum rate for decoding at 8MHz SCC. A circuit for encoding Manchester is available from Zilog.

Q. When will the Time Constant be loaded into the BRG counter?

A. After a S/W reset or a Zero Count is reached.

Q. How to run NRZ data using the DPLL?

A. Use NRZI for DPLL (WR14) but set to NRZ (WR10).

INTERNAL TIMING

Q. When does data transfer from the transmit buffer to the shift register?

A. About 3 PCLK's after the last bit is shifted out.

Q. How long does it take for a write operation to get to the transmit buffer?

A. It takes about 5 PCLK's for the data to get to the buffer.

Q. What is Valid Access Recovery Time?

A. Since WR/ and RD/ (AS/ and DS/ on the Z8030) have no phase relationship with PCLK, 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.

Q. How long is Valid Access Recovery Time?

A. On the current device from Zilog introduced in 1986, the recovery time is 4 PCLK's. Earlier SCC's required 6 PCLK's plus some additional nanoseconds.

Q. Why does the Z8030 require that the PCLK be "at least 90% of the CPU clock frequency for Z8000?"

A. If the clocks are within 90%, then the setup and hold times will be met. Otherwise, the setup and hold times must be met by the user.

Q. Does Valid Access Recovery Time apply to all successive accesses to the SCC?

A. Any access to the SCC requires that the recovery time be observed before a new access. This includes reading several bytes from the receive FIFO, accessing separate bytes on two different channels, etc. When using DMA or block transfer methods, the recovery time must be considered.

Q. Do the DMA request and wait lines on the SCC take the Valid Access Recovery time into account before they make a request?

A. No, they are not that intelligent. The user must take this into account, and program the DMA accordingly. For example, be inserting wait states during the memory access between SCC accesses, which will lengthen the time in between SCC accesses, or by requiring the DMA to release the bus between accesses to the SCC, to prevent simultaneous data requests from two channels from violating the recovery time.

Q. What happens if Valid Access Recovery Time is violated?

A. Invalid data can result.

Q. Does Valid Access Recovery Time affect the interrupt acknowledge cycle?

A. No. The interrupt vector is put on the bus by the SCC during the interrupt acknowledge cycle, but does not require any recovery time.

Q. Why can some systems violate the recovery time by 1 or 2 PCLK's without affecting the data to the SCC?

A. This violation may or may not matter to the SCC. This phase relationship between PCLK, /RD, /WR, (/AS, /DS for Z8030) can by ASYNC. The SCC requires some time internally to synchronize these signals. The electrical specs for the SCC indicate a recovery time, which is the worst case maximum.

INTERRUPT CONSIDERATIONS

8

Q. What conditions must exist for the SCC to generate an interrupt request?

A. Interrupts must be enabled (MIE = 1 and IE = 1). The Interrupt Enable Input (IEI) must be high. The interrupt pending bit (IP) must be set and its interrupt under service bit (IUS) must be reset. No interrupt acknowledge cycle may be active.

Q. How can the /INTACK signal be synchronized with PCLK?

A. /INTACK needs to be synchronized with PCLK. This can be accomplished by changing /INTACK only on the falling edge of PCLK by using a D flip-flop that is clocked with the inverted PCLK.

Q. Is /CE required during an Interrupt Acknowledge cycle?

A. No.

Q. How long does /INT stay active low when requesting an interrupt?

A. If the SCC is operated in a polled mode, the /INT will remain active until the IP bit is reset. For an interrupt acknowledge cycle, the /INT will go inactive shortly after the falling edge of /RD or /DS when the IUS bit is set.

Q. Can you use the SCC without a hardware interrupt acknowledge?

A. Yes. If you are not using the hardware daisy chain, you don't need to give an interrupt acknowledge. Tie the intack pin high, enable interrupts, and on responding to an interrupt, check RR3 for the cause, and special receive conditions if you are in receive mode. The internal daisy-chain settling time must still be met. (IEI to IEO delay time specification.)

Q. How do you acknowledge an interrupt without a hardware interrupt acknowledge?

A. Reset the responsible interrupt pending bit (IP). The /INT line follows the IP bit.

Q. When are the IP bits cleared?

A. A transmitter empty IP is cleared by writing to the data register. A receive character available IP is cleared by reading the data register. The external/status interrupt IP is cleared by the command Reset Ext/Status Interrupts.

Q. Can the IP bits be set while the SCC is servicing other interrupts?

A. Yes. If the interrupting condition has a higher priority than the interrupt currently being serviced, it causes another interrupt, thus nesting the interrupt services.

Q. Can the IUS bits be accessed?

A. No. They are not accessible.

Q. When do IUS bits get set?

A. The IUS bits are set during an interrupt acknowledge cycle on the falling edge or /RD or /DS.

Q. How do you reset interrupts on the SCC?

A. The interrupt under service bit (IUS) can be reset by the command "Reset Highest IUS" or 38 Hex to WR0. Reset Highest IUS should be the last command issued in the interrupt service routine.

Q. Why is the interrupt daisy chain settle time required?

A. This mechanism allows the peripheral with the highest priority interrupt pending in the hardware interrupt daisy chain to have its interrupt serviced.

INTERRUPT CONSIDERATIONS (Continued)

Q. Is there still a settle time if the peripherals are not chained?

A. Even if only one SCC is used, there still is a minimum daisy-chain settle time due to the internal chain.

Q. How should the vectors be read when utilizing the /INTACK?

A. /INTACK should be tied to 5 volts through a register. Erroneous reads can result from a floating INTACK. The interrupt vectors can be read after an interrupt from RR2.

Q. How is the vector register different from the other registers?

A. The vector register is shared between both channels. The Write register can be accessed from either channel. Reading "Read Register 2" on Channel A (RR2A) returns the unmodified vector, and RR2B returns the modified vector that includes status. The vector includes the status bit (VIS, WR9) and determines which vector register is put out on the bus during an interrupt cycle.

Q. How do you poll the external/status IP?

A. Set the IE bits in WR15 so the conditions are latched and set ext/status master interrupt enable bit in WR1. To guarantee the current status, the processor should issue a Reset External/Status interrupts command in WR0 to open the latches before reading the register. For further details see the SCC Technical Manual, section 3.4.7.

Q. When should the status in RR1 be checked?

A. Always read RR1 before reading the data.

Q. What conditions cause the transmit IP to be set?

A. Either the buffer is empty, or the flag after CRC is being loaded.

Q. How do you tell if you have a Zero Count (ZC) interrupt?

A. This bit is not latched like the other external IP bits. If an external interrupt occurs and none of the other IP bits have changed since the last ext/status interrupt, then the ZC condition caused it. A ZC interrupt will not be generated if there are other ext/status (IP) pending. The ZC stays active for each time only when the count reached zero, approximately two PCLK time periods.

Q. How do you poll the bits in RR3A?

A. Enable interrupts in WR1 and disable MIE before polling.

Q. What happens when the SCC is programmed to interrupt on transmit buffer empty and also to request DMA activity on transmit buffer empty?

A. This would not be a wise thing to do. The interrupt would occur but the DMA could gain control of the bus and remove the interrupting condition before the interrupt acknowledge could take place. When the CPU recovers control of the bus and starts the interrupt acknowledge cycle, bus confusion results because the peripheral no longer has a reason to interrupt.

Q. Will IP bit (s) for external status be cleared by the Reset Ext/Status Interrupt?

A. Yes.

ASYNCHRONOUS MODE

Q. Can the Sync Character Load Inhibit function strip characters in Asynchronous mode if not disabled?

A. Yes. If not disabled it will strip any characters which match the value in the sync character register. Always disable this function in asynchronous mode (WR3, bit D1).

Q. What controls the DTR/WREQ pin?

A. The DTR pin follows the D7 bit in WR5 (inverse) as a Data Terminal Ready pin, or it is a DMA request line (WREQ). The bit can be set or reset by writing to WR5.

Q. How is the Asynchronous mode selected?

A. The Asyn mode is selected by programming the number of stop bits in write register 4.

Q. How are receiver breaks handled?

A. The SCC should monitor the break condition and wait for it to terminate. When the break condition stops, the single NULL character in the receive buffer should be read and discarded.

Q. Where can you get the DTR input if the DTR/REQ pin is being used for DMA?

A. The SYNC can be used as an input if operating in the Async mode. It will cause an interrupt on both transitions.

Q. When a special condition occurs due to a parity error, will a receive interrupt for that byte still be generated?

A. No. In the case of Receive interrupt on Special Condition Only mode, the interrupt will not occur until after the character with the special condition is read. In the case of Receive Interrupt on All Characters or Special Condition Only mode, the interrupt is generated on every character whether or not it has a special condition.

Q. In the Auto Enable mode, what happens when CTS/ goes inactive (high) in the middle of transferring a byte?

A. If the Auto Enable mode is selected, the CTS/ pin is an enable for the transmitter. So, when CTS/ is inactive, transmit stops immediately.

Q. Can X1 clock mode really be used for the Async operation?

A. X1 mode cannot be used unless the receive and transmit clocks are synchronized. Using a synchronous modem is one way of satisfying this requirement.

Q. When does the FIFO buffer lock on an error condition?

A. The receive data FIFO gets locked only in cases where the following receiver interrupt modes are selected:

- Receive Interrupt on Special Condition only
- Receive Interrupt on First Character or Special Condition

In both of these modes, the Special Condition interrupt occurs after the character with the special condition has been read. The error status has to be valid when read in the service routine. The Special Condition locks the FIFO and guarantees that the DMA will not transfer any characters until the Special Condition has been serviced.

SYNCHRONOUS MODES
(SDLC, HDLC, Bysync, And Monosync Modes Included)

Q. For what are the cyclical redundancy check (CRC) residue codes used?

A. The residue codes provide a secondary method to check the reception of the message.

Q. Why is the second byte of the CRC incorrect when read from the receiving SCC?

A. The second byte of the CRC actually consists of the last two bits of the first byte or CRC, and the first six bits of the second byte of CRC. For more details there is an Application Note available from Zilog called "SDLC Residue Codes for the Z80 SIO" which applies to the SCC as well.

Q. How does the SCC send CRC?

A. The SCC can be programmed to automatically send the CRC. First, write the first byte of the message to be sent. This guarantees the transmitter is full. Then reset the Transmit Underrun/EOM latch (WR0 10). Write the rest of the data frame. When the transmit buffer underruns, the CRC is sent. The following table describes the action taken by the SCC for the bit-oriented protocols:

Tx Underrun EOM Latch Bit	Abort/Flag Bit	Action Upon Tx Underrun	Comment
0	0	Sends CRC + Flags	Valid Frame
0	0	Sends Abort + Flags	Aborted Frame
1	X	Sends Flags	Software CRC

The SCC sets the Tx Underrun/EOM latch when the CRC or Abort is loaded into the shift register for transmission. This event causes an interrupt (if enabled).

Q. In SDLC, when do you reset the CRC generator and checker?

A. The Reset TxCRC Generator command should be issued when the transmitter is enabled and idling (WR0). This needs to be done only once at initialization time for SDLC mode.

Q. How can you make sure that a flag is transmitted after CRC?

A. Use the external status end of message (EOM) interrupt to start the CRC transmission, then enable the transmit buffer empty interrupt. When you get the interrupt, it means that the buffer is empty, a flag is loaded in the shift register, and you can send the next packet of information.

Q. If the SCC is idling flags, and a byte of data is loaded into the transmit buffer, what will be transmitted?

A. Data takes priority over flags and will be loaded in the shift register and transmitted.

Q. Since data is preferred, can this cause a problem?

A. This allows you to append on the end of a message, but it can cause problems with DMA. A character could be transmitted without an opening flag. To make sure that a flag has been transmitted, watch for the W/REQ line to toggle when the flag is loaded into the shift register.

Q. Can you gate data by stretching the receive clock?

A. You can hold the clock until you have valid data. There are no maximum specs on the RxC period, and the edges are used to sample the data. If there are no edges, no data is sampled.

Q. How do you synchronize the DPLL in SDLC mode?

A. There are two methods to synchronize the DPLL. Supply at least 16 transitions at the beginning of each message so the DPLL has time to make adjustments, or use the DPLL search mode in WR14 to cause the SCC to synchronize on first transition. The first edge must be guaranteed to be a cell boundary.

Q. In SDLC, is the flag and address stripped-off?

A. No, only the flag is stripped. The address will be the 1st character received.

Q. Does IBM® SDLC specify parity?

A. No.

Q. Can the SCC include parity in SDLC mode?

A. Yes. It is appended at the end of the character.

Q. How does the SCC operate in transparent mode?

A. The transparentness, as defined by IBM SNA, should be provided by the software. The SCC does not perform any automatic insertion and deletion of link control nor does it automatically exclude the characters from the CRC calculation. This also applies to other high level protocols.

Q. When does the Abort function take effect?

A. The abort takes place immediately by inserting eight consecutive 1's.

Q. Can the SCC detect multiple aborts?

A. The SCC searched for seven consecutive 1's on the receive data line for the abort detection. This condition may be allowed to cause an external status interrupt. After these seven 1's are received, the receiver automatically enters Hunt mode, where it looks for flags. So, even if more than seven 1's are received in case of multiple aborts, only the first sequence of 1's is significant.

Q. How do you send an end of poll (EOP) flag in SDLC loop mode?

A. To send the EOP message, simply toggle the bit which idles flags or ones to mark flags, then mark ones. This produces a zero and more than seven 1's; an EOP condition.

Q. When the SCC is programmed for 6 bit sync, how are bits sent?

A. Six bits are sent. The 12-bit sync character sends 12 bits.

Q. Do sync patterns (or flags) in data transmissions get stripped and still cause interrupts?

A. All leading sync patterns (and all flags) are automatically stripped if the Sync Character Load Inhibit feature is programmed. Any data stripped from the transmission stream cannot cause a receive character available interrupt but may cause other interrupts (such as External/Status for Sync/Hunt and special receive condition for EOM).

Q. How are the sync characters sent at the beginning of a Bisync frame?

A. Load the transmit buffer with the first byte and the sync characters are automatically sent out.

Q. How can you determine when the flag has been completely sent?

A. There are several ways to determine if the flag has been completely sent. This allows the transmitter to be shut off, or in half duplex the line can be turned around. This requires a little work by the user because the SCC does not know when the last flag bit has been shifted our. The following are some suggestions:

- Once the flag is loaded into the transmit shift register, start an external clock. Use the baud rate generator as the counter.

- Tie the transmit line into DCD or an available input pin, and watch for a zero, or end of flag. If you are running half-duplex, use the local loopback mode and watch for the flag to end.

- Allow an abort, although this destroys the last character. Be sure to send a dummy character - then idle flags after the abort latch is set.

Q. How do the DMA W/REQ lines operate?

A. DMA request lines follow the state of the transmit buffer.

Q. How does the SCC handle messages less than four bytes in length?

A. A 4-byte message consists of an address, control word, no data, and 2 bytes of CRC. SDLC defines messages of less than 4-bytes as an error. It is not defined how the SCC will react, however, as tested by a SCC user, 4-, 3-, and 2-byte messages cause an interrupt on end of frame, but a 1 byte message does not cause an interrupt.

MISCELLANEOUS QUESTIONS

Q. Can the SCC support MARK and SPACE parity in async?

A. The SCC can transmit-end the equivalent of MARK parity by setting WR4 to select two STOP bits. The receiver always checks for only one STOP bit; therefore, the receiver does not verify the MARK parity bit.

The SCC (and products using the SCC cell) does not support SPAC parity for transmitting or receiving. The Zilog USC Family of serial datacom controllers do support odd, even, mark, & space parity types.

Q. Since both D7 and D1 bits in RR0 are not latched, it is possible that the receiver detected an Abort condition, set D7 to 1, initiated an external/status interrupt and before the processor entered the service routine, termination of the abort was detected, which reset the Break Abort bit. Currently in the TM (page 7-20), the description for Bit1: Zero Count states if the interrupt service routine does not see any changes in the External/Status conditions, it should assume that a zero count transition occurred when in fact, an Abort condition occurred and was missed. What could be done to correct this and not miss the fact that an Abort occurred?

A. Very few people actually use the Zero Count interrupt. This interrupt is generated TWICE during each bit time and is usually used to count a specific number of bits that are sent or received. If this interrupt is not used by your customer, then what is said in the TM about the Zero Count is true for the Abort Condition. If no other changes occurred in the external/status conditions and the Zero Count is not used, then the source of the interrupt was the Abort condition.

Q. Can the SCC resynchronize independent clocks (at the same frequency, but could be out of phase), one for Rx data and one for Tx data?

A. No, the two clocks are independent of each other. However, the SCC provides a special transmitter-to-receiver synchronization function that may be used to guarantee that the character boundaries for the received and transmitted data are the same. This function can be found in the SCC Technical Manual (Q3/90 Section 5.2.3).

Q. When is EOM and EOF asserted?

A. EOM is asserted when it detects depletion of data in the Tx buffer; EOF is asserted when it detects a closing flag.

Q. After powering up the SCC, are the reset values in the write and read registers guaranteed?

A. No. You must perform a hardware or software reset. A list of the hardware and software reset values are listed in the SCC Technical Manual (Q3/90) on page 3-9.

Q. Can you read the status of a write register, such as the MIE bit in WR9?

A. No, in order to retain the status of a write register, you must keep its status in a separate memory for later use. However, the only exception is that WR15 is a mirror image of RR15. Also, the ESCC has a new feature to allow the user to read some of the write registers (see the ESCC Product Specification or Technical Manual for more details).

Q. Is there a signal to indicate that a closing SDLC flag is completely shifted out of the TxD pin? This is needed to indicate that the frame is completely free of the output to allow carrier cut off without disrupting the CRC or closing flag.

A. No, the only way to find this timing is to count the number of clocks from Tx Underrun Interrupt to the closing flag. The ESCC contains the feature by deasserting the /RTS pin after the closing flag. Upgrade to the ESCC!

Q. Does the SCC detect a loss of the receive clock signal?

A. No, if the clock stops, the SCC senses that the bit time is very long. Use a watch-dog timer to detect a loss in the receive clock signal.

Q. Is there any harm in grounding the "NO CONNECT" (NC) pins in the PLCC package (pin #17,18,28,36)?

A. These NC pins are not physically connected inside the die. Therefore, it is safe to tie them to ground.

Q. Can the SCC be used as a shift register in one of the synchronous modes with only data sent to the Tx register with no CRC and no sync characters?

A. CRC is optional in Mono-, Bi-, and External Sync Modes only. The sync characters can be stripped out via software.

INTEGRATING SERIAL DATA AND SCSI PERIPHERAL CONTROL ON ONE CHIP

The Z85C80 SCSCI has the fit, form and function for the size-conscious arena of Laptop computers, portable printers, portable copiers and other electronic systems where serial communications and peripheral control are design requirements.

INTRODUCTION

The SCSCI™ combines on one chip the industry standard 10 MHz SCC (Serial Communications Controller) and a fast SCSI (Small Computer System Interface) controller through Zilog's Superintegration™ methodology. These highly integrated circuits perform their systems tasks as efficiently and reliably as their discrete counterparts and require less board space to accomplish their functions. (Figures 1 and 2). This compactness is an important feature in the size-conscious electronic world of laptop computers, portable printers, portable copiers, and other small peripherals. Included with these primary features of reliability and density is the overriding fact of cost savings.

This App Note (Application Note) defines two different customer application examples for Zilog's Z85C80 SCSCI; an Apple Macintosh laptop compatible computer and its peripherals (Figure 1); a high-end typesetting system (Figure 2).

The SCSCI low current requirements are not provided by the sleep modes discussed in this App Note.

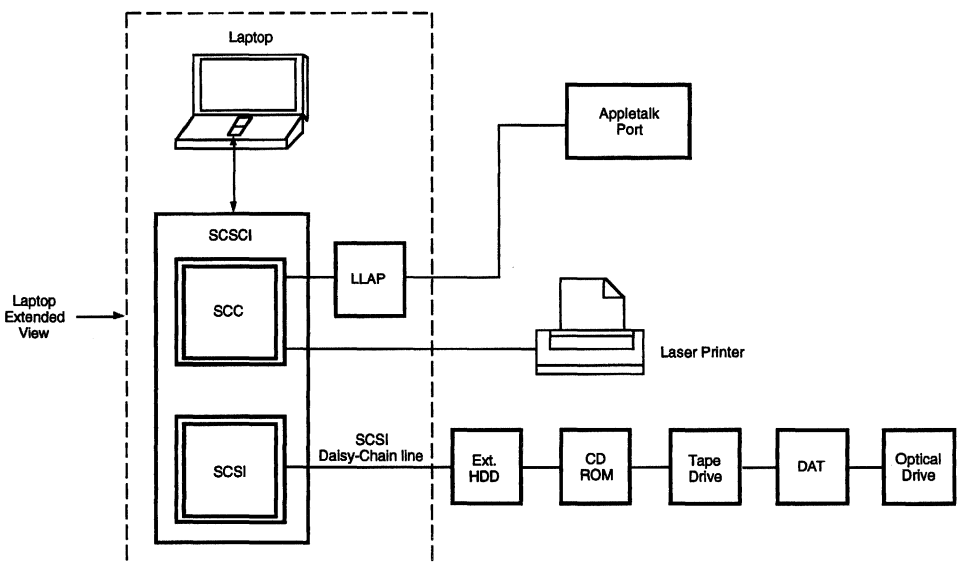


Figure 1. Apple Macintosh Laptop Compatible Computer System

INTRODUCTION (Continued)

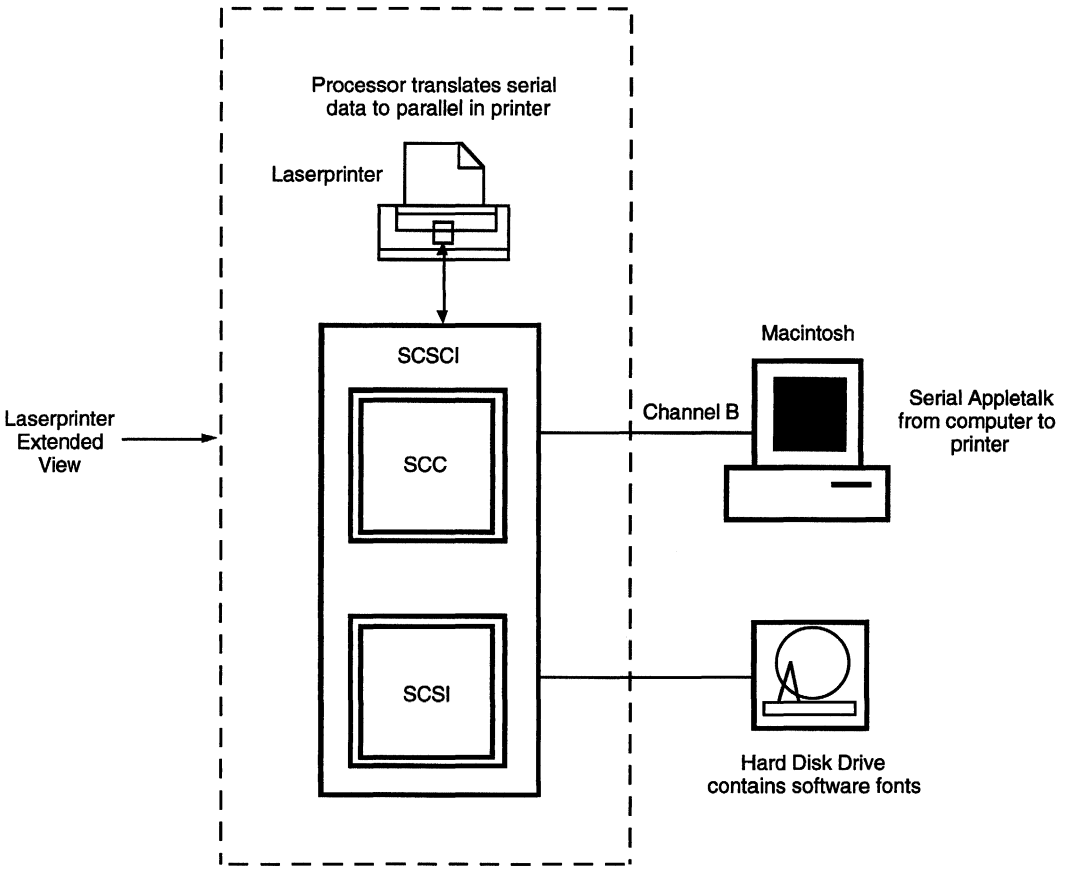


Figure 2. Texas Instruments High-End Typesetting System

Z85C80 - LAPTOP COMPATIBLE COMPUTER SYSTEM

The SCSCI incorporates all communication requirements needed to build a personal computer. The computer includes a hard disk and interface network and/or other serial communications ports. The SCSCI is particularly useful in the laptop system because of its critical space requirements. It can be used in applications where the SCC provides the Appletalk® interface and the on-chip SCSI is the interface to hard disks, CD ROMs, tape drives, DATs and optical drives.

Z85C80 - LASER PRINTER

The high-end typesetting example of a laser printer system shows the SCSCI handling both the network (Appletalk) interfaces and the optional hard disk that is used for buffering font bit-maps. Data movement is between the host computer, laser printer and hard disk drive (Reference Figure 2). Disk caching of bit-maps, as font characters are drawn, can significantly improve performance when large type faces are used.

In this example, one of the SCC channels provides the data link interface for the Appletalk Network that ties the system together. The SCC provides the necessary interrupts and acknowledgments for high speed synchronous serial communications.

8

SCC/SCSI BRIEF DESCRIPTIONS

The following subsections give a brief refresher on the basic descriptions and features of the SCC and SCSI. If more detailed information is needed, see Zilog's Datacom ICs Databook (DC-2503-02) sections on the SCC, SCSI or SCSCI.

On-Chip SCC Functions

The SCC is a dual-channel, multiprotocol data communications peripheral that easily interfaces to CPUs with either multiplexed or non-multiplexed 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 like baud rate generators, digital phase-lock loops, and crystal oscillators dramatically reduce the need for external logic (Figure 3).

The SCC handles asynchronous formats, synchronous byte-oriented protocols such as IBM Bisync, and synchronous bit-oriented protocols like HDLC and IBM SDLC. It supports virtually any serial data transfer application. The SCC also generates and checks CRC codes in any synchronous mode and can be programmed to check data integrity in various modes. A daisy-chain interrupt hierarchy is also supported.

SCC/SCSI BRIEF FUNCTIONS (Continued)

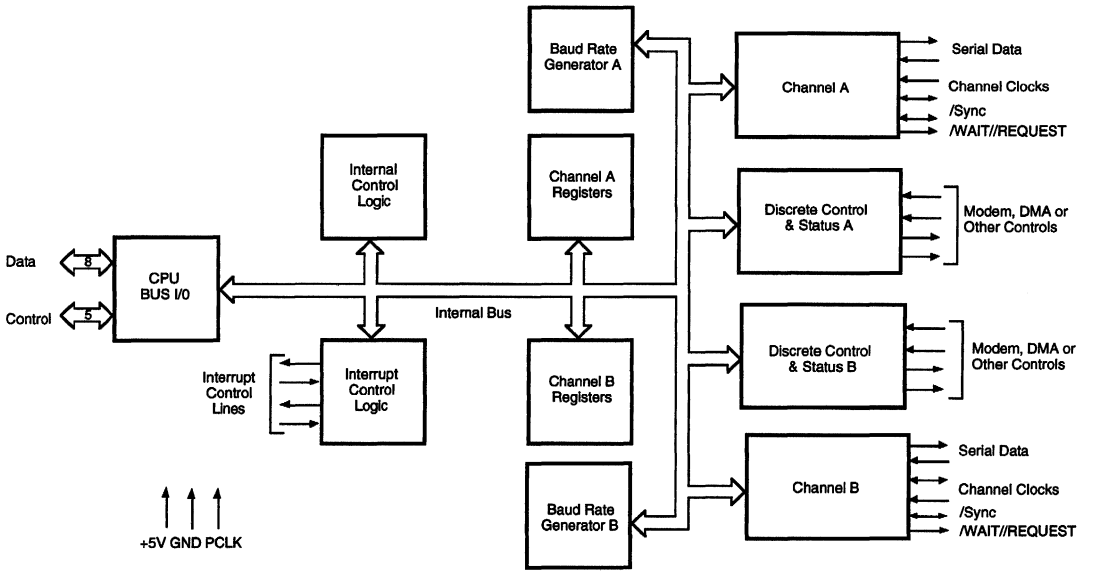


Figure 3. SCC Portion of the SCSCI

On-Chip SCSI Functions

The SCSI controller portion of the SCSCI is designed to implement the SCSI protocol as defined by the ANSI X3.131 - 1986 standard. The industry standard SCSI provides the flexibility of working 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. Like any other device, the SCSI has the necessary hookups to interface to the system CPU (Figure 4).

The CPU reads from and writes to the SCSI registers that can be addressed as standard or memory-mapped I/O. The SCSI increases system performance by minimizing the CPU intervention in DMA operations (controlled by the SCSI). The CPU is interrupted by the SCSI when the SCSI detects a bus condition that requires attention. It also supports arbitration and reselection and has the proper handshake signals to support normal mode DMA operations.

The maximum number of peripheral devices the SCSI can control is eight with a maximum usable distance of 6 meters (approximately 20 feet). Each peripheral is given a device number according to the system scheme of the designer. The SCSI bus provides for connections of large capacity storage with fast byte/parallel operation including quick Acknowledge handshake timing (Figure 5).

8

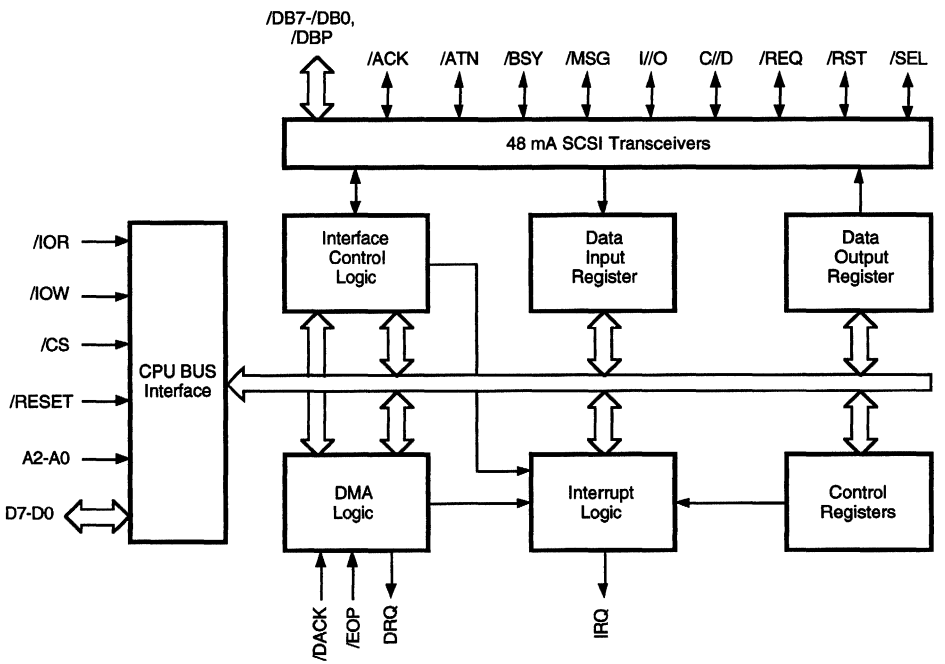


Figure 4. SCSI Portion of the SCSCI

SCC/SCSI BRIEF FUNCTIONS (Continued)

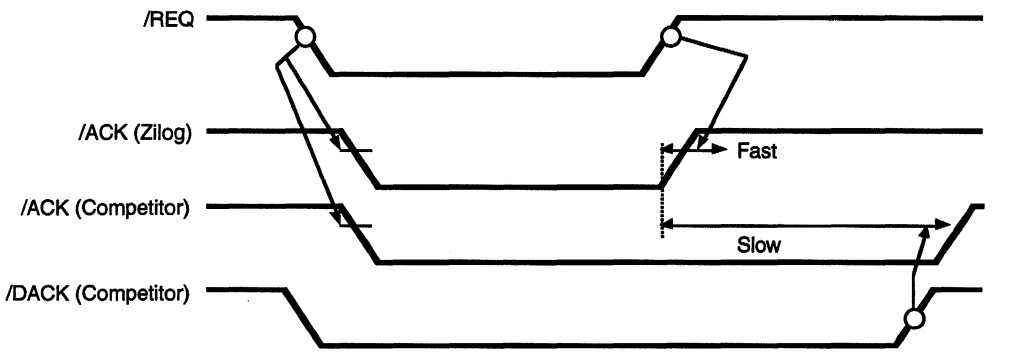


Figure 5. Fast Acknowledge Cycle Timing

SAVING POWER

The SCSCI has a "Sleep Mode" feature that is extremely desirable in the low power systems under discussion. Sleep mode saves power consumption by shutting down all active circuits not in use. While in sleep mode, any incoming clock is stopped, all internal Voltage Controlled Oscillators (VCO) and Ring Oscillators are disabled, and to avoid driving any resistive load, SCSI outputs are tri-stated. A key design issue for systems with clock stop/start capability is to make sure no glitches occur when the system clock is being stopped or started.

The SCSCI sleep mode is designed so that the SCC and SCSI devices are powered down separately or together depending on user needs. The SCSCI is in Deep Sleep when both the SCC and SCSI are powered down. In this case, it utilizes the least current.

Clock and Host Interface

The SCSCI clock and host interface are crucial from a design point of view. The SCSI timing generator (Figure 6) has an internal oscillator for Arbitration and Selection timing defined in the ANSI X3.131-1986 standard. The SCC cannot be written to while PCLK is stopped, but any read or write to SCSI is unaffected by PCLK since this part of the device is asynchronous. An application may shut down the SCSI disk while serial communication is active, or shut down the serial port while data is being transferred to/from the disk.

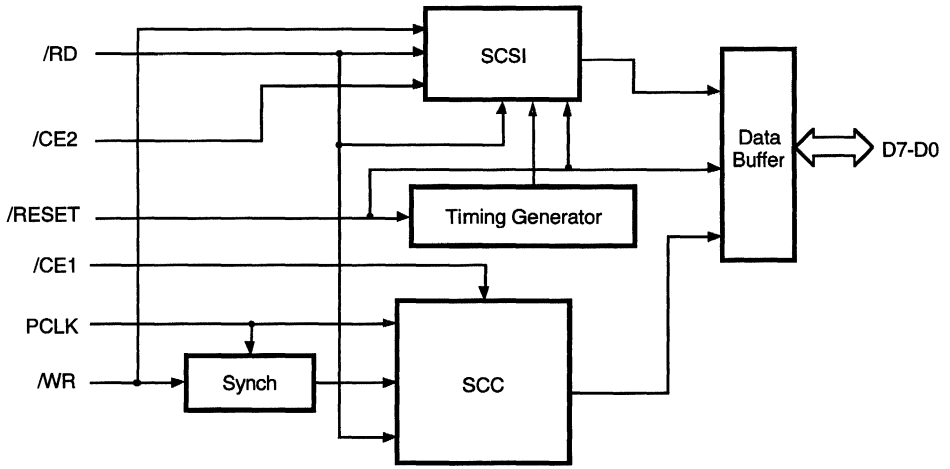


Figure 6. Simplified Block Diagram of the SCSCI

Design Hints for Battery-Powered Computers

The following list of design hints should help the engineer in designing battery-powered computers using the SCSCI.

- To save power, no Power-On-Reset (POR) circuitry is implemented in the SCSCI. The user needs to reset the SCC cell with either the software command "Hardware Reset" or through hardware by asserting the /RD and /WR pins simultaneously.
- In order to ensure that databus outputs are inactive, either:
  1. /RD should be kept at logic 1, or
  2. /CE1 and /CE2 should be kept at logic 1.
- To go to Deep Sleep mode, PCLK is stopped at the logic 1 level and /RESET is at logic level 0. Also, the user should ensure that no double clocking or glitches occur on the PCLK going into or coming out of this state.
- To stop the internal timing generator, /RESET of the SCSI portion should be kept Low (Figure 7).
- A minimum of two clocks is required after /RESET is deasserted before any normal read or write is allowed.
- Using the on-chip crystal oscillator amplifiers can represent a significant portion of the total current usage of the SCC portion of the SCSCI. Therefore, use an alternate choice of clock options when power consumption is important.
- Power-On Reset should be applied to the SCC when coming out of Sleep Mode before restarting communications.

SAVING POWER (Continued)

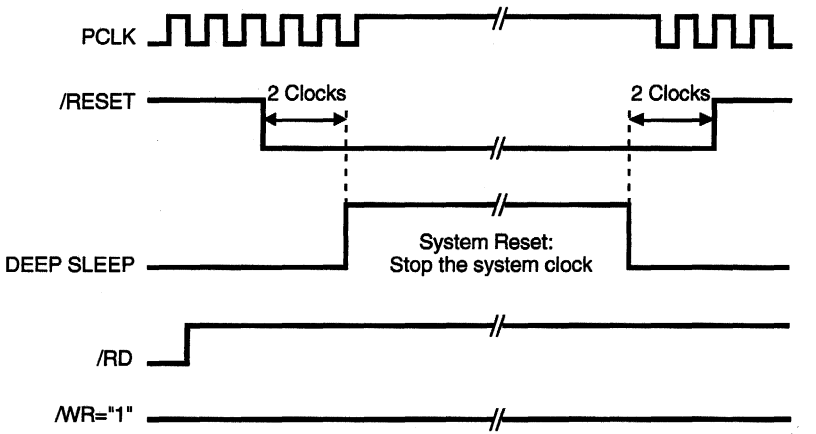
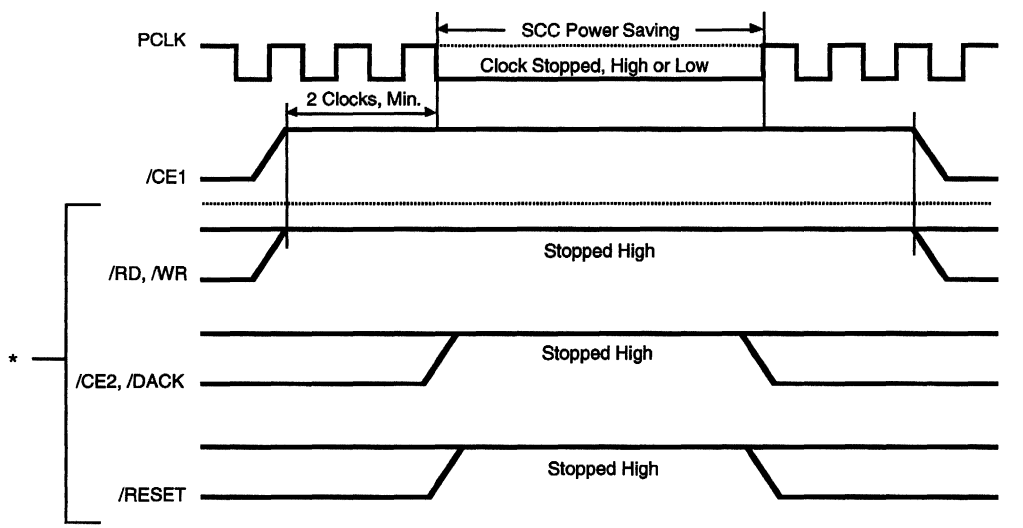


Figure 7. Typical Timing for Low Power Sleep Modes

Figures 8 and 9 are typical SCC and SCSI power saving timing diagrams. It is recommended that while the chip is in Deep Sleep mode, /CE1, /CE2, /RD and /WR are at logic 1 (PCLK stopped).



* If the inputs to the SCSCI are inactive High, power consumption is further reduced.

Figure 8. SCC Power Saving Modes

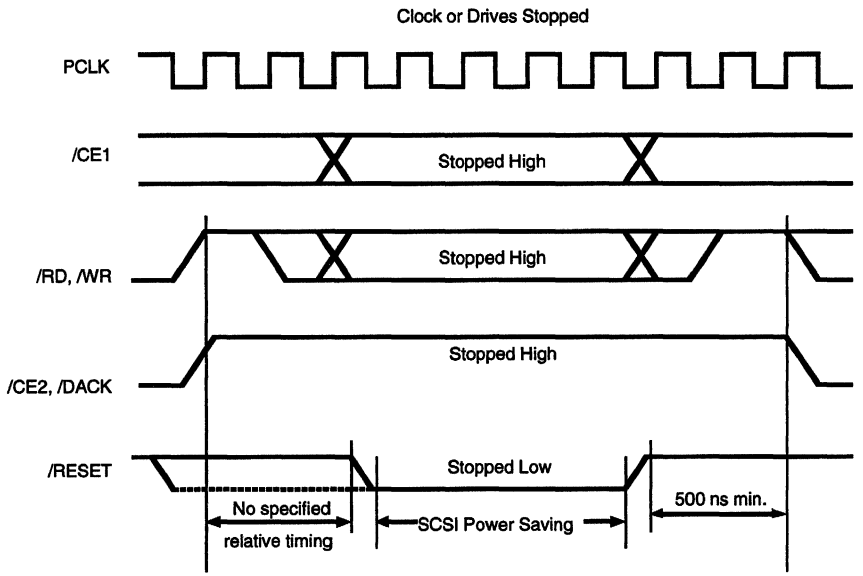


Figure 9. SCSI Power Saving Mode (SCSI held in RESET)

Note:
The SCC and SCSI power saving modes can be used independently, or in combination. The SCC mode alone reduces the SCSCI current consumption to approximately 1mA. Both modes together reduce current consumption to approximately 100 µA.

INITIALIZATION PROCEDURES

The SCC initialization procedures for either one of the two system examples in this App Note can be asynchronous or synchronous. The following discussion concentrates on the polled asynchronous mode.

Asynchronous Initialization

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

Initialization Procedures

The SCC initialization procedure is divided into three parts; programming the operation modes, e.g., bits-per-character, parity, loading the constants; e.g., interrupt vector, time constants. The second part enables the hardware function; e.g., transmitter, receiver and baud-rate generator. It is important that the operating modes are programmed before the hardware functions are enabled. The third part, if required, consists of enabling the different interrupts.

Table 1 shows the order (from top to bottom) in which the SCC registers should be programmed. Those registers that do not need programming are listed as optional in the commands column. The bits in the registers marked with an X are programmed by the user. The bits marked with an S are set to their previous programming value. For example, in Part 3, Write Register 3, bits D7-D1 are shown with an S because they have been programmed in Part 1 and must remain set to the same value.

Table 1. SCC Initialization Order

Part 1. Modes and Constants

WR9	1100000	Hardware Reset
WR4	XXXXXXXX	Tx/Rx con, Async or Sync Mode
WR1	0XX00X00	Select W/REQ (opt)
WR2	XXXXXXXX	Program Interrupt Vector (opt)
WR3	XXXXXXX0	Select Rx Control
WR5	XXXX0XXX	Select Tx Control
WR6	XXXXXXXX	Program sync character (opt)
WR7	XXXXXXXX	Program sync character (opt)
WR9	00X0XXX	Select Interrupt Control
WR10	XXXXXXXX	Miscellaneous Control (opt)
WR11	XXXXXXXX	Clock Control
WR12	XXXXXXXX	Time constant lower byte (opt)
WR13	XXXXXXXX	Time constant upper byte (opt)
WR14	XXXXXXX0	Miscellaneous Control
WR14	XXXSSSS	Commands (opt)

Part 2. Interrupt Status

WR15	XXXXXXXX	Enable External/Status
WR0	00010000	Reset External Status
WR0	00010000	Reset External Status twice
WR1	XSSXXSXX	Enable Rx, Tx and Ext/Status
WR9	000SXSSS	Enable Master Interrupt Enable

1=Set to one X=User defined
0=Reset to zero S=Same as previously prog.

Part 3. Enables

WR14	000SSSS1	Baud Rate Enable
WR3	SSSSSSS1	Rx Enable
WR5	SSSS1SSS	Tx Enable
WR0	10000000	Reset Tx CRG (opt)
WR1	SSS00S00	DMA Enable (opt)

Initialization Table Generation

Figures 10a and 10b provide a worksheet for the initialization of the SCC. All the bits that must be programmed as either a 0 or a 1 are already filled in; the remaining bits are left blank and are programmed by the user according to the desired mode of operation. The binary value can then be converted to a hexadecimal number and placed in the table following the Write register notation in the column labeled HEX.

Register		Hex		Binary 7 6 5 4 3 2 1 0	Comments
Modes	WR9	—	—	1 0 0 0 0 0 0 0	Software Reset
	WR0	—	—	0 0 0 0 0 0	
	WR4	—	—		
	WR1	—	—	0 0 0 0 0	
	WR2	—	—		
	WR3	—	—	0	
	WR5	—	—	0	
	WR6	—	—		
	WR7	—	—		
	WR9	—	—	0 0 0 0	
	WR10	—	—		
	WR11	—	—		
	WR12	—	—		
	WR13	—	—		
	WR14	—	—	0	
	WR14	—	—	0	

Figure 10a. SCC Initialization Worksheet

INITIALIZATION PROCEDURES (Continued)

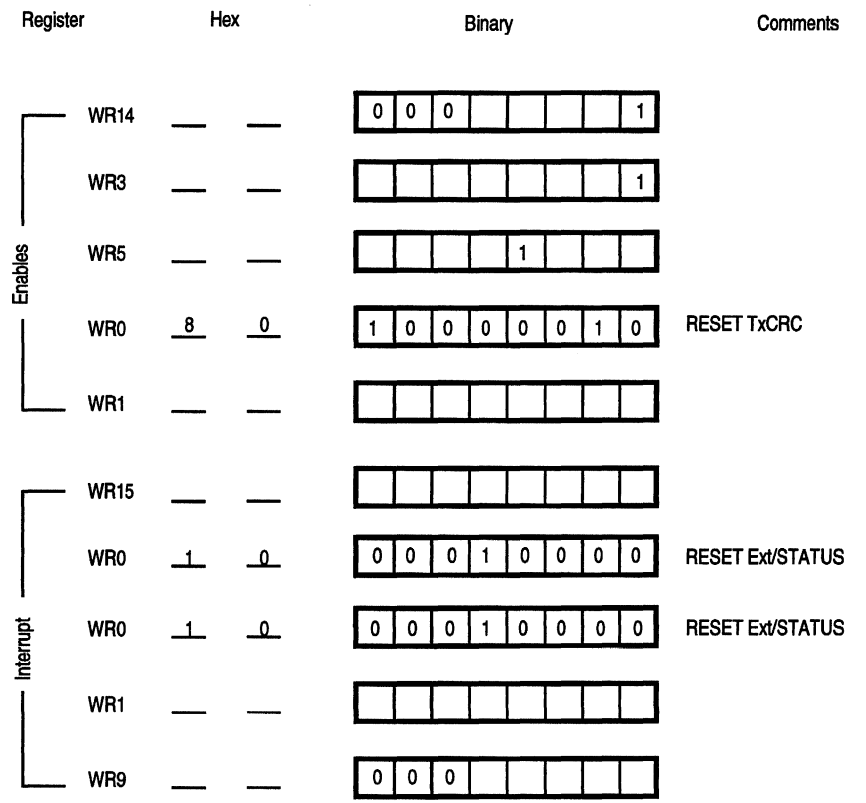


Figure 10b. SCC Initialization Worksheet (Continued)

SCC Polled Asynchronous Mode

The SCC, in polled asynchronous mode, can be set with five to eight bits per character, 1, 1.5, or 2 stop bits and a wide range of baud rates. In this particular example, 8 bits per character, 2 stop bits and a 9600 baud rate are used. The SCC can be programmed to local loopback for on-board diagnostics. The user can make use of this feature to test-program the part without additional hardware to simulate an actual transmit and receive environment.

In Figure 11, the 8-bit data bus and control lines all come from the user's CPU. PCLK comes from the system clock, or an external crystal, up to the maximum rate of the SCC. The IEI and the /INTACK pins should be pulled up. The baud rate generator clock is connected to the /RTxC pin.

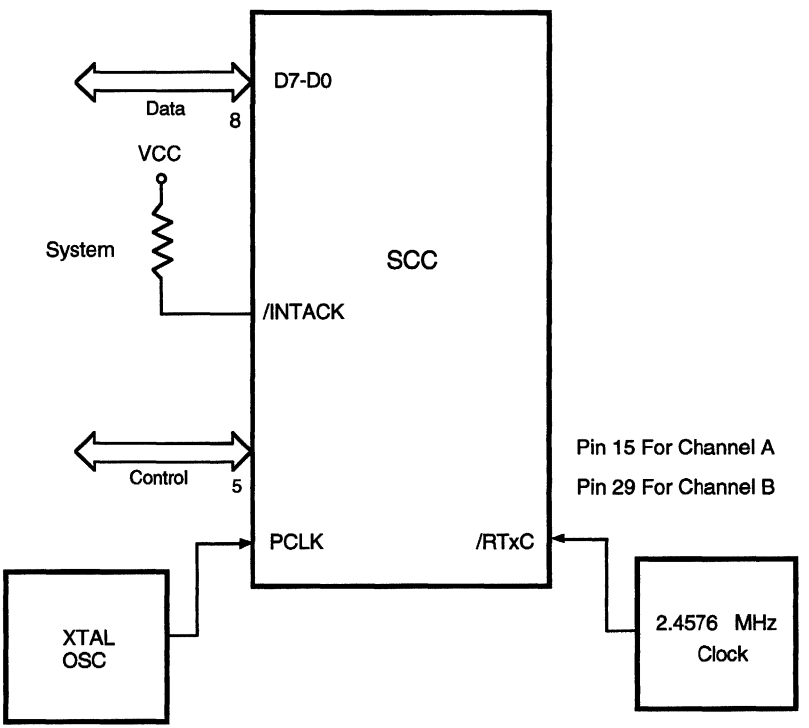


Figure 11. SCC to CPU Interface

SCC Initialization

Initialization of the SCC for polled asynchronous communication is divided into two parts; part one programs the operating modes of the SCC and part two enables them. Care must be taken when writing the software to meet the SCC's Cycle and Reset Recovery times. The cycle recovery time of six PCLK cycles applies to the period between any read or write cycles affecting the SCC. The reset recovery time is the period after a hardware reset caused either by hardware or software; this recovery time extends the cycle recovery time to 11 PCLK cycles.

Table 2. Polled Asynchronous Initialization Procedure

Register	Value	Comments
WR9	C0H	Force Hardware Reset
WR4	4CH	x16 clock, 2 stop bits, no parity
WR3	C0H	R x 8 bits, Rx disabled
WR5	60H	T x 8 bits, DTR, RTS, Tx off
WR9	00H	Int. Disabled
WR10	00H	NRZ
WR11	56H	Tx & Rx = BRG out, TR x C = BRG out
WR12	06H	Time constant = 6
WR13	00H	Time constant high = 0
WR14	10H	BRG in = RT x C, BRG off, loopback
		Enables
WR14	11H	BRG enable
WR3	C1H	Rx enable
WR5	68H	Tx enable

SCC Polled Asynchronous Register Descriptions

The following are the descriptions of the register settings for polled asynchronous operations that are given in Table 2:

WR9 resets the SCC to a known state by writing C0H. The command, Force Hardware Reset, is identical to a hardware reset.

WR4 selects the asynchronous x16 mode with two stop bits and no parity. The x16 mode means that the clock rate is 16 times the data rate.

WR3 selects 8 bits per received character, but does not enable the receiver fully at this time because the SCC has not been initialized.

WR5 selects 8 bits per transmitted character, but does not enable the transmitter at this time because the SCC has not been fully initialized.

WR9 determines that there are no interrupts enabled. This inhibits the SCC from requesting an interrupt from the CPU.

WR10 selects NRZ encoding. This NRZ coding is used on the transmitter as well as the receiver.

WR11 selects the /RTxC pin as the basic clock input; the baud rate generator is the source for the transmit and receive clocks.

WR12 & WR13 selects the baud rate generator's time constant. The WR13 time constant is determined by the equation:

Time Constant =
Clock Frequency/2 x Baud Rate x clock mode

In this example, the clock frequency is 2.4576 MHz, the baud rate is 9600 and the clock mode is 16; therefore, the time constant is 6. Expressed as a 16-bit, hexadecimal number, it is 0006H. The time constant LOW (WR12) is, therefore, 06H and the time constant HIGH (WR13) is 00H. The baud rate for this example can be varied as long as the data rate is less than 1/4 of the PCLK rate. Table 3 shows the time constants for other common baud rates.

Table 3. Time Constants for Common Baud Rates*

Baud Rate	Divider Dec	Divider Hex
38400	0	0000H
19200	2	0002H
9600	6	0006H
4800	14	000EH
2400	30	001EH
1200	62	003EH
600	126	007EH
300	254	00FEH
150	510	01FEH

*For 2.4576 MHz Clock, X16 Mode

R14 selects the baud-rate generator as the /RTxC pin, the baud-rate generator disabled, and the internal loopback. The baud-rate generator uses the /RTxC pin as the clock source and is not enabled at this time because the SCC initialization is not complete.

SCC Operating Mode Enables

WR14 enables the baud-rate generator. Bit 0 (LSB) is changed to a 1 to enable the baud-rate generator; all other bits must maintain the value selected during initialization.

WR3 enables the receiver. Bit 0 (LSB) is changed to a 1 to enable the receiver, all other bits must maintain the value selected during initialization.

WR5 enables the transmitter. Bit 3 is changed to a 1 to enable the transmitter, all other bits must maintain the value selected during initialization.

March 1992

ISCC QUESTIONS AND ANSWERS

Q. Is the interrupt vector present on both the lower 8 bits and the upper 8 bits in an interrupt cycle (See Figure 40 of the Z16C35 CMOS ISCC Product Spec)?

A. Both halves of the AD bus are driven during an interrupt acknowledge cycle by the ISCC. In fact, both halves of the data bus are never driven individually.

Q. In DMA mode, must the /WAIT//READY and /BUSACK signals be externally synchronized to PCLK (See Figure 46 of the Z16C35 CMOS ISCC Product Spec)?

A. No, not exactly. The documented timing shows when the ISCC samples these coming back from memory.

Q. Can the address and data bus be outputted before /BUSACK is received (See Figure 46 of Z16C35 CMOS ISCC Product Spec)?

A. No.

Q. What causes the Terminal Count to be Reset?

A. Refer to P.5-26 TM, Sec. 5.6.2, "the status in this register is automatically cleared after a Read." In other words the bits are Reset when you Read the contents of the register.

Q. Which Rev of the SCC is in the ISCC?

A. It is the D Rev (but without the oscillator fix).

Q. Does the ISCC allow software interrupt acknowledge (WR9 bit D5)?

A. Yes, it does. It is not required to use the /INTACK signal of the ISCC to process interrupts. The source of the interrupt can be determined by reading the interrupt vector just like a normal interrupt is determined by reading the interrupt vector (like a normal register Read). The SCC RR2B is modified to reflect the source. RR2A is not modified. Also, the other status registers could be used to figure out who interrupted. SCC interrupts can be Reset by reading RR2B if software interrupt acknowledge is enabled (WR9 D5=1).

Q. Does the software interrupt acknowledge support DMA operation?

A. No. Unlike the SCC core, the DMA core does not support this feature. The DMA has two sources of the interrupt, i.e., IP and IUS bits.

Q. When the ISCC is used on a multiplexed bus, the ISCC does not interrupt when the SCC source interrupts occur until after another Write to the ISCC. Why?

A. When programmed for multiplexed bus operation, similar to the Z8030/Z80C30, the /AS signal is used to update the interrupt status of the SCC. Consequently, if no /AS is present, the interrupt status is not updated until an /AS occurs. If /AS of the ISCC is tied to the /AS of the processor, sufficient /AS signals will occur to keep the ISCC interrupts up to date. However, if /AS is only generated to the ISCC when it is being accessed, any pending SCC interrupts will not assert the /INT pin until after the /AS of an access to the part. This typically occurs when a PAL is used to generate the access signals to the ISCC and /AS is only generated to the ISCC when it is being accessed.

Q. Can the Upper Address Strobe be defeated (to shorten the transfer cycle time)?

A. No. But this is possible in the IUSC!

Q. How many clock cycles does it take to do a DMA transfer, after BUSACK is granted?

A. By looking at Figure 45 of Z16C35 CMOS ISCC Product Spec, it takes TS0, TS1, T0, T1, T2, T3, T4, T5, about 8 cycles total.

Q. Is there any reason why the ISCC couldn't use pclk twice as fast as the processor, in order to cut access recovery times?

A. No, as long as the required timings are met!

Q. What's the recovery time required for the ISCC?

A. A recovery time may apply to ANY access of the ISCC. Thus, a bus transaction before or after an access of the ISCC looks like it requires that the recovery time be met for those accesses. The timing for /Strobe signals, i.e. /DS, /WR, /RD or Pulsed /INTACK relative to CLK is three clocks if /Strobe, synched to the /INTACK relative to CLK, is three clocks if /Strobe is synched to the rising edge of CLK; or four clocks otherwise. The Recovery time is independent of /CS. Please note, if in any design application with the ISCC the reads and writes are unreliable, this recovery timing should be checked very carefully and as this could be a bug with the ISCC.

Q: Is the SDLC FIFO available in ISCC?

A: Yes, the SDLC FIFO is available in the SCC cell of the ISCC. There is a mistake in our ISCC Technical Manual, P.5-20, on Register Description. The statement 'Bit 2 is not used and must be programmed "0" is wrong. Bit 2 of WR15 is used for enabling the SDLC FIFO.

Q: Will DMA be enabled by writing the Enable Command in the Channel Command/Address Register?

A: Yes, DMA operation is triggered by the command, 'Enable DMA' on Channel Command/Address Register. This is another mistake in our ISCC Technical Manual, P.5-25, on Registe Description. The statement 'DMA operation is not triggered by this command' is wrong, e.g., Writing "100" to bits 7 through 5 enables and triggers TxB DMA operation.

Q: Will DMA operation be triggered by the DMA enable command in the DMA Enable Register?

A: Yes, DMA operation will also be triggered by setting corresponding DMA Enable bits in the DMA Enable Register (P.5-29 sec 5.6.7, DMA Enable Register in ISCC Technical Manual). Note that this is a read/write register. Read-modify-write instructions should be used in writing this register to avoid the register value to be overwritten and cause accidental enabling/disabling of the DMA operations.

BOOST YOUR SYSTEM PERFORMANCE USING THE ZILOG ESCC™

For expanded testing, larger interface flexibility, and increased CPU/DMA offloading, upgrade from the SCC to the ESCC™ Controller ... and benefit from the ESCC's full potential.

INTRODUCTION

This App Note (Application Note) describes the differences between the SCC (Z8030/8530, Z80C30/85C30) and ESCC (Z80230/85230). It outlines the procedures in utilizing the ESCC to its full potential. Application details such as Schematics and Program Listings are not included since these materials are in our various application support products.

Note: The author assumes the audience has fundamental Datacommunications knowledge and basic familiarity with Zilog SCC products.

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

Power connections follow conventional descriptions below:

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

ESCC/SCC DIFFERENCES

The differences between the ESCC and SCC are shown below:

ESCC ENHANCEMENT	PERFORMANCE BENEFITS
1. Extended Read Enable of Write Registers	- Improves Testing Capabilities - Ability to examine SDLC status on-the-fly
2. Hardware Improvement - Modified WRITE Timing - Modified DMA Request on - Transmit Deactivation Timing	- Improves Interface to 80 x 86 CPU - Improves Interface DMA-driven system
3. Throughput improvement - Deeper Transmit FIFO - Deeper Receive FIFO - FIFO Interrupt Level	- Reduces TBE Interrupt Frequency by 3/4 - Reduces RCA Interrupt Frequency by 3/4 - Flexibility in Adapting CPU Workload
4. SDLC End Of Frame Improvement - Automatic RTS Deassertion after Closing Flag - Automatic Opening Flag Transmission - Automatic TxD Forced High in SDLC with NRZI Encoding When Marking Idle After End Of Frame - Improvement to Allow Transmission of Back-to-Back Frames with a Shared Flag - Status FIFO Anti-Lock Feature in DMA-Driven System	- Reduces CPU and DMA Controller Overhead after End Of Frame - Allows Optimal SDLC Line Utilization

ESCC/SCC DIFFERENCES (Continued)

The differences between the ESCC and SCC are summarized by a new register, WR7' (Figure 1).

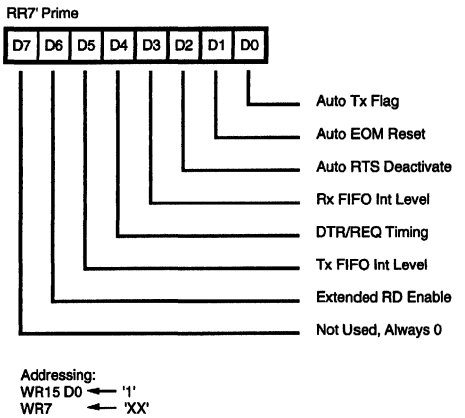


Figure 1. WR7' Definition

The advantages of the new features are illustrated in the following examples.

One of the features that is offered by the ESCC, but not the SCC, is Extended Read Enable. Write Register values from the WR3, WR4, WR5, WR7', and WR10 can be examined in the ESCC but not the SCC. This feature improves system testability. It is also crucial for SCC/ESCC differentiation and allows generic software structures for all SCC/ESCC devices.

Flowchart 1 (Figure 2) shows a generic software structure applicable for all SCC/ESCC initializations. Flowchart 2 (Figure 3) suggests a method for determining which type of SCC/ESCC™ device is in the socket. This software structure helps the development of software drivers independent of the device type.

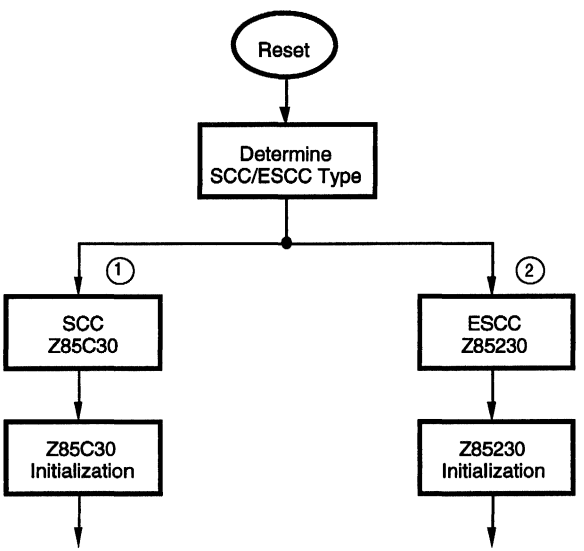


Figure 2. Generic SCC/ESCC Drivers

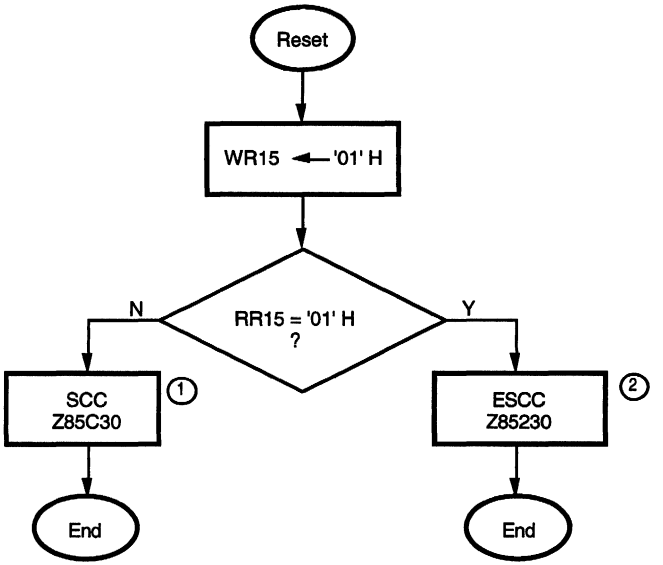


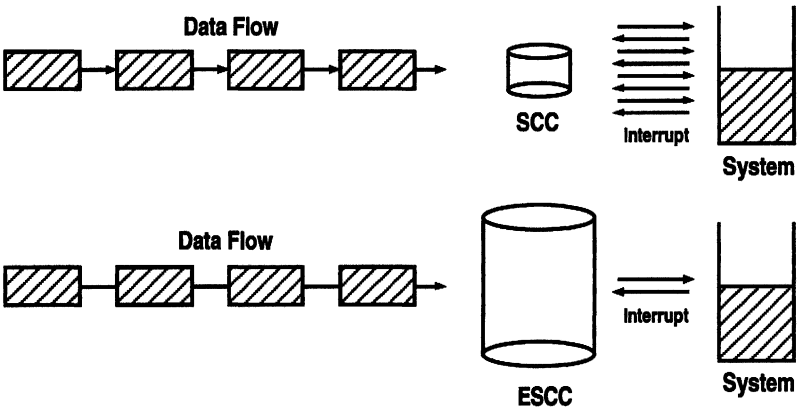
Figure 3. SCC/ESCC Differentiation Flowchart

ESCC SYSTEM BENEFITS

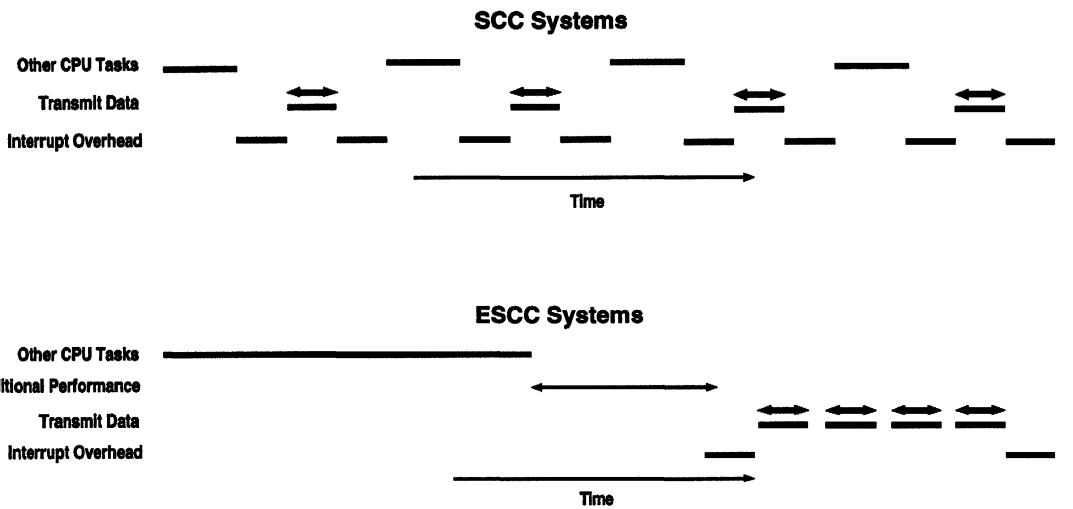
The Software Overhead sets the System Performance Limits. The ESCC's deeper FIFOs and other features significantly reduce the software overhead for each channel. This allows:

- More Channels Per System
- Faster Data Rates on Channels
- More CPU bandwidth available for other tasks
- Lower CPU Costs

ESCC SYSTEM BENEFITS (Continued)



Interrupt Frequency Reduction



ESCC Reduces System Workload and Allows Extra Performance

TRANSMIT FIFO INTERRUPT

In the ESCC, transmit interrupt frequencies are reduced by a deeper Transmit FIFO and the revised transmit interrupt structure. If the WR7' D5 Transmit FIFO Interrupt Level bit is reset, the transmit interrupt is generated when the entry location of the FIFO is empty, i.e., more data can be written. This is downward compatible with a SCC Transmit Interrupt since the SCC only has a one-byte transmit buffer instead of a four-byte Transmit FIFO.

If WR7' D5 is set, the transmit buffer empty interrupt is generated when the transmit FIFO is completely empty. Enabling the transmit FIFO interrupt level, together with polling the Transmit Buffer Empty (TBE) bit in RR0, causes significant transmit interrupt frequency reduction. Transmit data is sent in blocks of four bytes (algorithm is illustrated in Figure 4). This helps to off-load those systems which have long interrupt latency or a fully loaded Operating System.

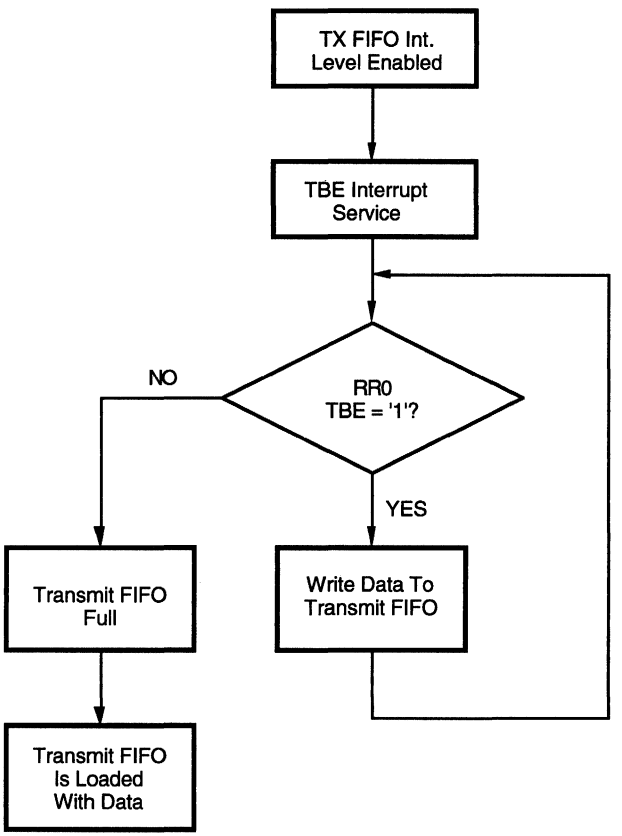


Figure 4. Flowchart of Transmit Interrupt Service Routine to Reduce Transmit Interrupt Frequencies

RECEIVE FIFO INTERRUPT

In the ESCC, receive interrupt frequencies are reduced due to a deeper Receive FIFO and the revised receive interrupt structure.

If WR7' D3 Receive FIFO Interrupt Level bit is reset, the ESCC generates the receive character available interrupt on every received character. This is compatible with SCC Receive Character Available Interrupt. If WR7' D3 is set, the Receive Character Available Interrupt is triggered when the Receive FIFO is half full; the first four locations from the entry are still empty. By enabling the receive FIFO interrupt level, together with polling the Receive Character Available (RCA) bit in RR0, the receive interrupt frequencies are reduced significantly. Receive data is read in blocks of four bytes (Figure 5). This would help to off-load systems which have a long interrupt latency and heavily loaded Operating Systems.

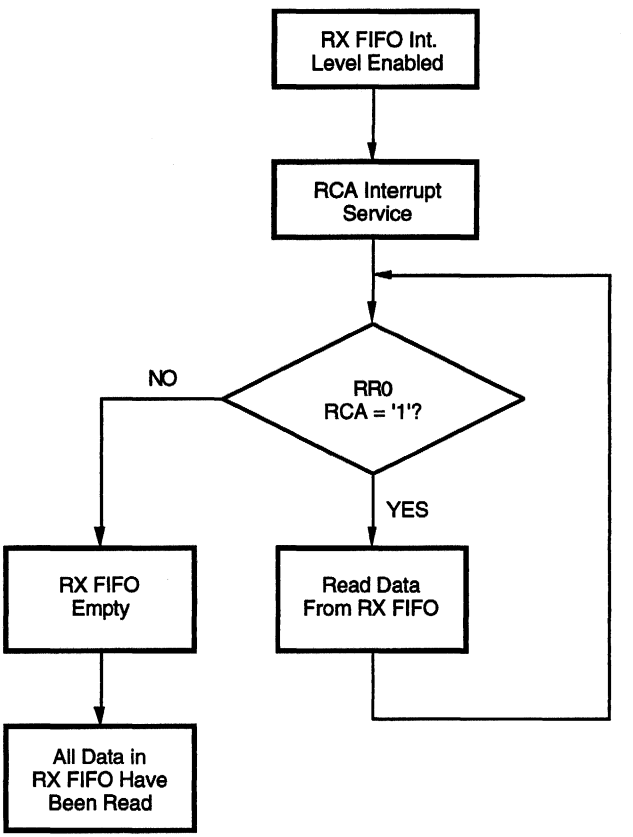


Figure 5. Flowchart of Receive Interrupt Service Routine to Reduce Receive Interrupt Frequencies

AUTOMATIC /RTS DEASSERTION

Several SDLC enhancements are provided in the ESCC. The ESCC allows automatic /RTS deassertion at End Of Frame (EOF). The automatic /RTS deassertion is enabled by setting WR7' D2. If ESCC is programmed for SDLC mode and the Flag-On-Underrun bit (WR10 D2) is reset, with the RTS bit (WR5 D1) reset, /RTS is deasserted automatically at the last bit of the closing flag. It is triggered by the rising edge of the Transmit Clock (TxC - Figures 6 and 7).

/RTS is normally used in SDLC for switching the direction of line drivers. Automatic /RTS deassertion allows optimal line switching without any software intervention. The typical procedures are as follows:

1. Enable Automatic /RTS Deassertion
2. Before frame transmission, set RTS bit
3. Enable frame transmission
4. Reset RTS bit
5. /RTS pin deassertion is delayed until the last rising TxC edge closing flag.

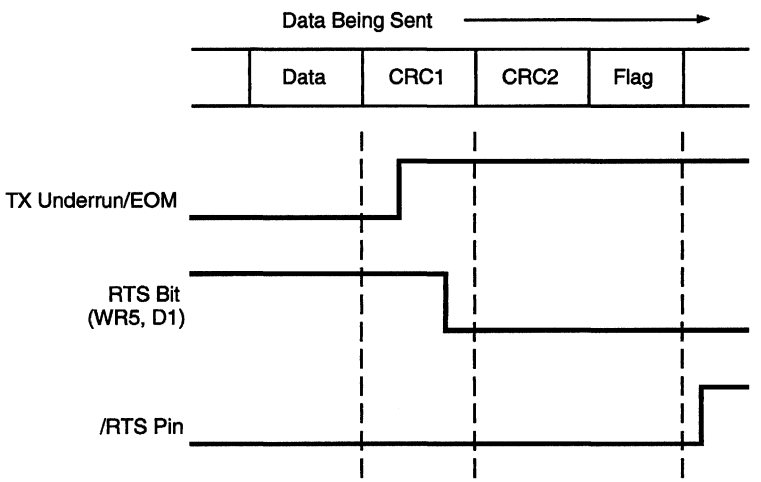


Figure 6. /RTS Deassertion Timing

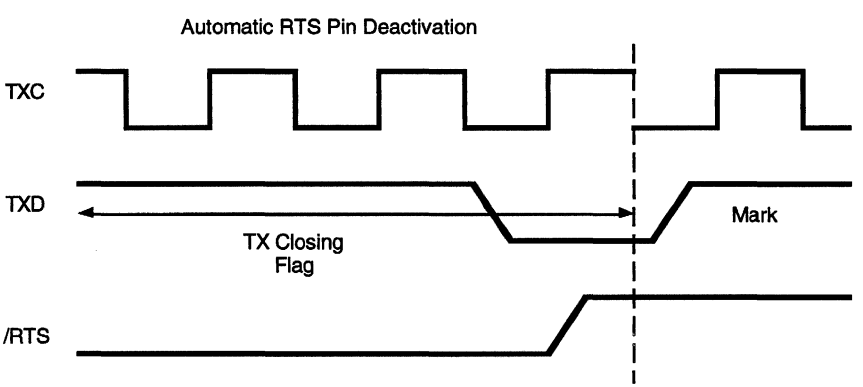


Figure 7. /RTS Deassertion Sequence

AUTOMATIC OPENING FLAG TRANSMISSION

When Auto Tx Flag (WR7', D0) is enabled, the ESCC automatically transmits a SDLC opening flag before transmitting data. This removes:

1. Requirements to reset the mark idle bit (WR10 D3) before writing data to the transmitter, or;
2. Waiting for eight bit times to load the opening flag.

TxD Forced High In SDLC With NRZI Encoding When Marking Idle After End Of Frame

When the ESCC is programmed for SDLC mode with NRZI encoding and mark idle (WR10 D6=0,D5=1,D3=1), TxD is automatically forced high when the transmitter goes to the mark idle state at EOF or when Abort is detected. This feature is used in combination with the automatic SDLC opening flag transmission to format the data packets between successive frames properly without any requirement in software intervention.

Status FIFO Enhancement

ESCC SDLC Frame Status FIFO implementation has been improved to maximize ESCC ability to interface with a DMA-driven system (Technical Manual, 4.4.3). The Status FIFO and its relationship with RR1, RR6 and RR7 is shown in Figure 8.

Other special conditions (e.g., Overrun) generates special receive conditions and lock the Receiver FIFO (Figures 9 and 10).

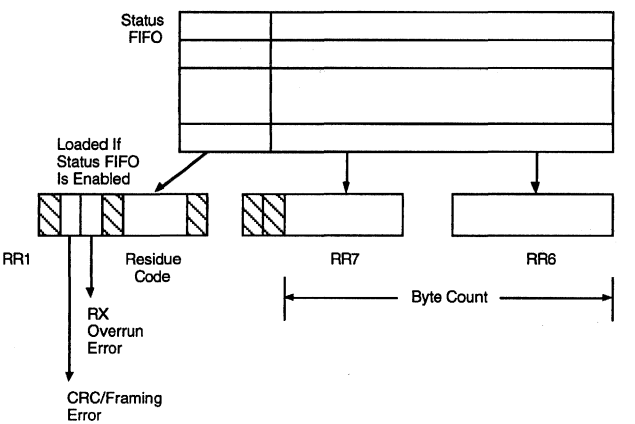


Figure 8. Status FIFO

SDLC Frame Status FIFO enhancement is enabled by setting WR15 D2. If it is enabled when EOF is detected, byte count and status from the Status FIFO are loaded into RR6, RR7 and RR1. This is used in DMA-driven systems. Historically, EOF is treated as a special condition. Special condition interrupts are triggered if any one of the below interrupts is enabled:

1. Receive Interrupt on First Character or Special Condition.
2. Interrupt on All Receive Characters or Special Conditions.
3. Special Receive Condition Only.

If 1 or 3 (above) is enabled, the data FIFO is locked after the interrupt is serviced by reading RR1 in the Status FIFO, as shown in Figure 11. This is commonly used in a DMA-driven system to avoid delivering useless information (e.g., EOF) to the data buffer. Locking the data FIFO is not desirable in systems with long interrupt latency and high data rate communications. The reason is the ERROR RESET command is necessary to unlock the FIFO. Data from the next frame may be lost if ERROR RESET fails to issue early.

This drawback is improved in the ESCC for a DMA driven system. By enabling interrupts on "Special Receive Conditions only" and SDLC status FIFO, EOF is treated differently from other special conditions. When EOF status reached the exit location of the FIFO:

1. A "Receive Data Available" interrupt is generated to signal that EOF has been reached.
2. Receive Data FIFO is not locked.

Because of these changes, the data from the next frame is securely loaded and the system processes the EOF interrupt. The only responsibility of the software is issuing the Reset Highest IUS before resuming normal operation (Figure 12).

8

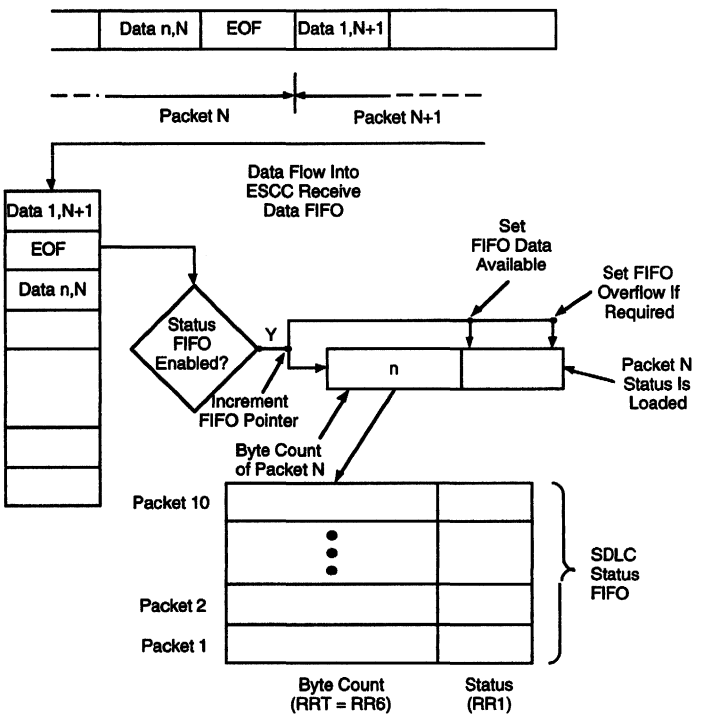
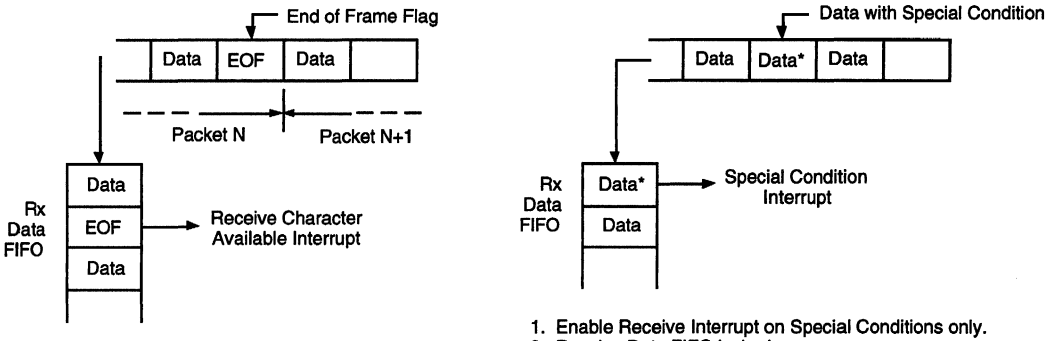


Figure 9. Status FIFO Operation at End Of Frame

Status FIFO Enhancement (Continued)



1. Enable Receive Interrupt on Special Conditions only.
2. Receive Data FIFO not locked.
3. Receive Character Available Interrupt generated even if it has not been enabled to indicate detection of EOF.

1. Enable Receive Interrupt on Special Conditions only.
2. Receive Data FIFO locked.
3. Special Condition Interrupt generated.

Figure 10. SDLC Status FIFO Anti-Lock

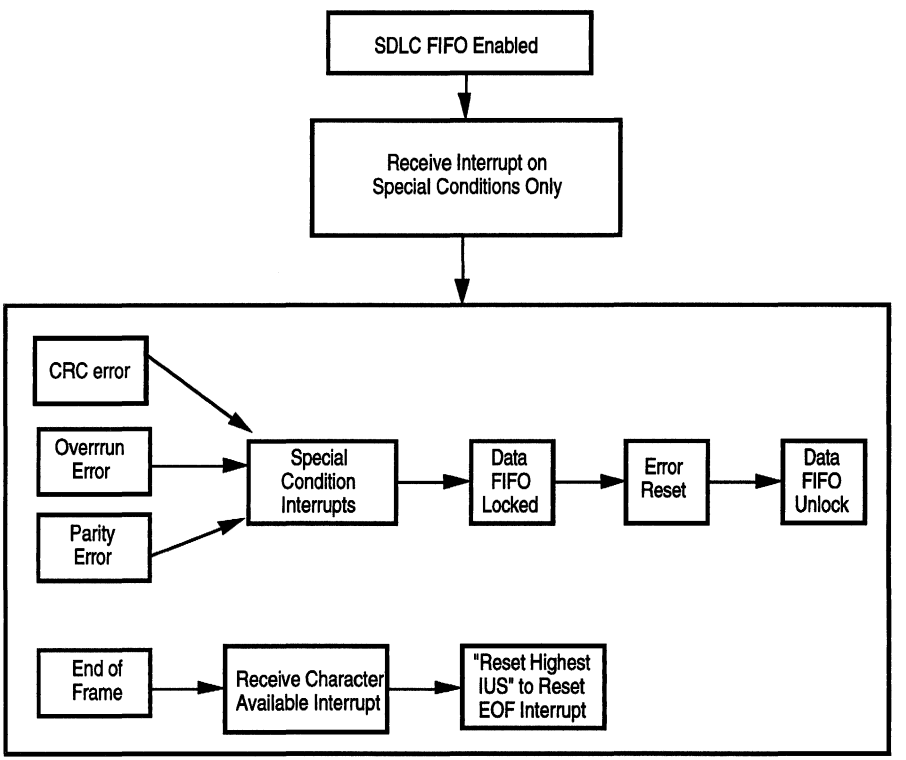


Figure 11. Receive Interrupt Mechanism 1

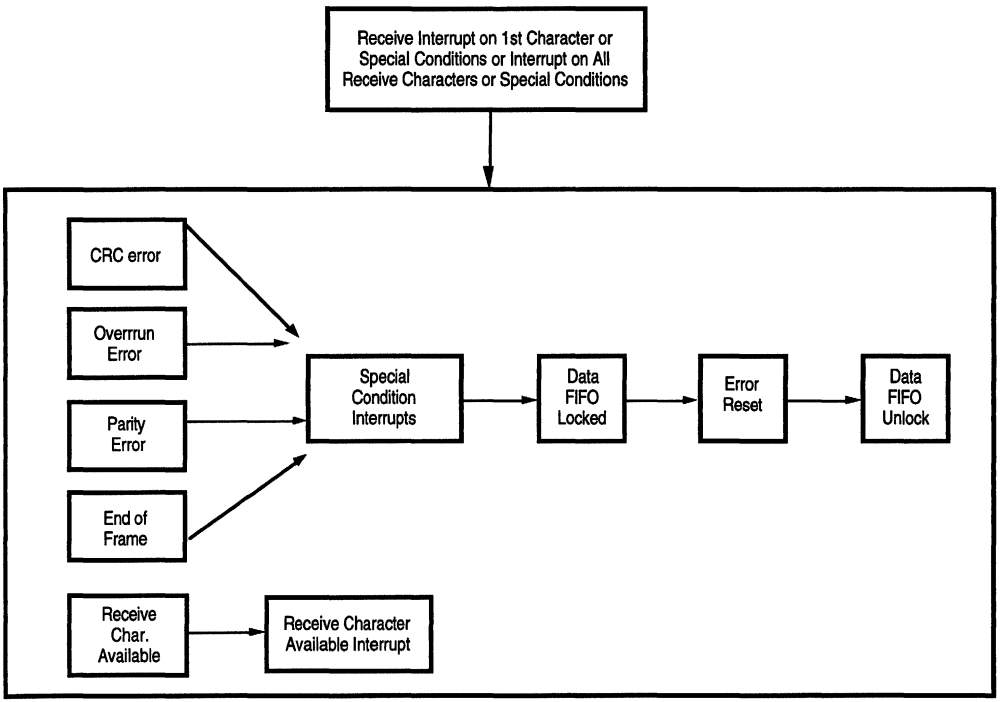


Figure 12. Receive Interrupt Mechanism 2

DMA Request on Transmit Deactivation Timing /DTR//REQ.

Timing implementation in the ESCC has been improved to make it more compatible with the DMA cycle timing (Reference Tech Manual, Section 2.5.2; DMA Request on Transmit).

Transmission of Back-To-Back Frames With a Shared Flag.

The ESCC provides facilities to allow transmission of back-to-back frames with a shared flag between frames (Figure 13).

In the ESCC, if the Automatic End Of Message (EOM) Reset feature is enabled (WR7' D1=1), data for a second frame is written to the transmit FIFO when Tx Underrun/EOM interrupt has occurred. This allows application software sufficient time to write the data to the transmit FIFO while allowing the current frame to be concluded with CRC and flag.

In the SCC, Transmission of Back-to-Back Frames is more difficult because (Figure 14):

1. Data cannot be written to the transmitter at EOF until a Transmit Buffer Empty interrupt is generated after CRC has completed transmission.

2. Automatic EOM Reset is not available in the SCC. Application software has to issue the "Reset Tx/Underrun EOM" command manually. The software overhead limits the next frame data to deliver immediately after the preceding frame has been concluded with CRC and Flag.

Transmission of Back-To-Back Frames With a Shared Flag (Continued)

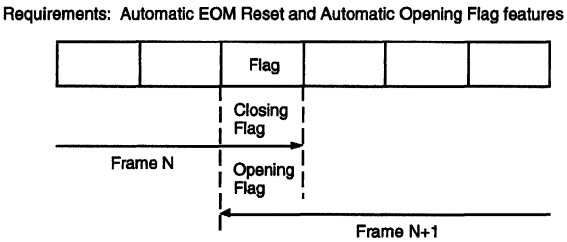


Figure 13. Transmission of Back-to-Back Frames with a Shared Flag

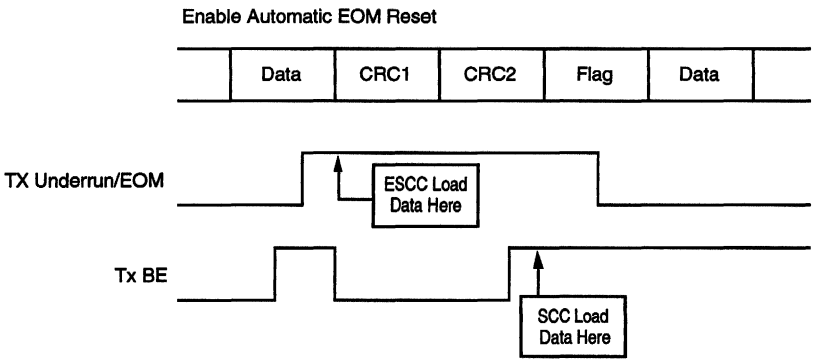
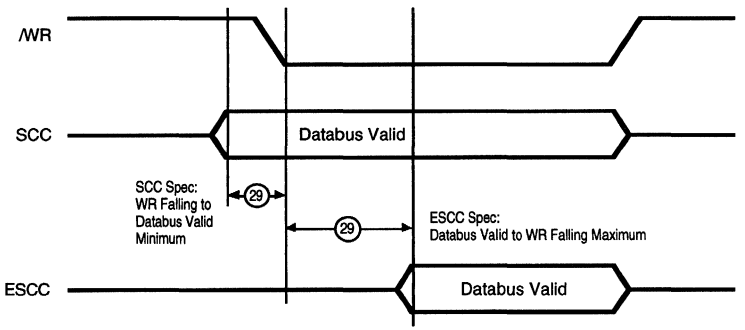


Figure 14. Operation of Shared Flag Transmission

Modified Write Timing

In the SCC write cycle, the SCC assumes the data is valid when /WR is asserted (Figure 15). This assumption is not valid for some CPUs, e.g., the Intel 80 x 86 The /WR signal from this CPU needs to delay for one more clock to initiate the write cycle. Additional hardware is required.

In the ESCC, write cycle timing has been modified so that data becomes valid a short time after write (approx. 20 ns). Therefore, if the data pins from the Intel CPU are connected directly to the ESCC, no additional logic is required.



Databus latched after falling edge of WR saves external logic required to delay WR until databus is valid. Typically needed with Intel CPUs.

Figure 15. Modified Write Timing

March 1992

PRODUCT DESCRIPTION

Q. Which of the following is the major factor in differentiating the ESCC from the USC Family?

(a) The ESCC has less communications channels than the USC

(b) The protocols supported by ESCC and USC are different

(c) The ESCC is limited in operation to less than 5 Mbps, but the USC Family can operate up to 10 Mbps

(d) The USC supports the T1 data rate, not the ESCC

A. (c) Most ESCC and USC Family members have two channels and protocols. Support by the SCC is a subset of ESCC. Both ESCC and USC can support T1 data rates so (a), (b), (d) are not correct.

Q. Which of the following is not an improvement from the SCC to the ESCC?

(a) The ESCC has deeper FIFOs

(b) The ESCC has new SDLC enhancements

(c) The ESCC has added new READ Registers

(d) The ESCC has added new WRITE Registers

A. (c) No new READ register addressing is added in the ESCC although we allowed some Write Registers to become readable through the existing READ Register.

The ESCC has 4 bytes of Tx FIFO and 8 bytes of Rx FIFO, while the SCC has 1 byte for the Tx and 3 bytes for the Rx.

The ESCC has many new SDLC enhancements, such as automatic EOM reset, automatic opening flag generation, etc.

The ESCC has added WR7' as a new WRITE Register to configure the new options, therefore, (a), (b), (d) are all differences between the SCC and ESCC.

APPLICATIONS

Q. Which of the following is a benefit from deeper FIFOs offered by the ESCC?

(a) More CPU bandwidths available for other system tasks

(b) Can support faster data rates on each channel

(c) Can support more channels for the same CPU

(d) All of the above

A. (d) (a), (b) and (c) are consequences of reduction in interrupt frequency that allows more horsepower to be delivered from the CPU.

Q. Which of the following CRC polynomials is supported in ESCC?

(a) CRC-16

(b) CRC-32

(c) CRC-CCITT

(d) (a) and (c)

(e) (b) and (c)

A. (d) CRC-32 is not supported in ESCC.

Q. How long does it usually take for the customer to migrate from SCC to ESCC in order to take the advantage of the FIFO?

(a) Less than 3 month

(b) About 6 month

(c) About a year

A. (a) Since the ESCC is a drop-in replacement to the SCC and using the deeper FIFO only requires minimal efforts.

Q. Which of the following is an applications support the tool for ESCC:

(a) Sealevel Board

(b) Electronic Programmers Manual

(c) Application Note "Boost Your System Performance Using the Zilog ESCC"

(d) All of the above

A. (d)

Q. Which of the following is a target application for the ESCC?

(a) AppleTalk-LocalTalk Peripherals

(b) X.25 Packet Switches

(c) SNA connectivity products

(d) All of the above

A. (d) ESCC could support the data rate and protocol required in the above applications.

THE ZILOG DATACOM FAMILY WITH THE 80186 CPU

Zilog's datacom family evaluation board features the 80186 along with four multiprotocol serial controllers, and allows customers to evaluate these components in an Intel environment.

INTRODUCTION

Zilog's customers need a way to evaluate its serial communications controllers with a central CPU. This App Note (Application Note) explains and illustrates how the datacom family interfaces and communicates with the 80186 on this evaluation board. The board helps the potential customer to evaluate Zilog's data communications controllers in an Intel environment.

The most advanced and complex component of the serial family is the IUSC. One of the highlights of this App Note is how the IUSC adapts to the 80186 CPU with a minimum of difficulty and a maximum of bus and functional flexibility.

8

GENERAL DESCRIPTION

The evaluation board includes the following hardware. (Reference two page Schematic diagram at rear of the App Note - Figures 5A and 5B.)

- Intel 80186 Integrated 16-bit Microprocessor
- Zilog Z16C32 Integrated Universal Serial Controller (IUSC™)
- Zilog Z16C33 Monochannel Universal Serial Controller (MUSC™) or USC®
- Zilog Z16C35 Integrated Serial Communications Controller (ISCC™)
- Zilog Z85230 Enhanced Serial Communications Controller (ESCC™) or SCC
- Two 28-pin EPROM sockets, suitable for 2764's through 27512's
- Six 32-pin (or 28-pin) SRAM sockets, suitable for 32K x 8 or 128K x 8 devices
- Four Altera EPLD circuits comprising the glue logic (Figures 1-4 at rear of the App Note) and Evaluation Board Schematic (Figures 5a, 5b).
- RS-232 and RS-422 line drivers and receivers
- Pin headers for configuring and interconnecting the above to serial applications

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

Power connections follow conventional descriptions below:

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

Processor

The 80186 may be operated at rates up to 16 MHz. To use the CPU clock for accurate serial bit clocking, a 9.8304 MHz CPU clock can be used. The crystal connected to the processor is 2X the operating frequency.

The processor's 1 Mbyte address space is well filled if the maximum RAM complement is installed. Of the integrated Chip Select outputs provided by the 80186, the /UCS output is used for the EPROMs, and all of the /PCS6-/PCS0 outputs are used for the datacom controllers. A hardware address decoder is used for the SRAMs instead of the 80186's /LCS and /MCS3-/MCS0 outputs because the RAMs must be accessible to the on-chip DMA functions of the ISCC and IUSC as well as the 80186. The 80186 does not decode addresses from external bus masters. Both 8-bit and 16-bit accesses are provided for RAM. The EPROMs are only accessible to the 80186.

The 80186's mid-range memory chip select feature (specifically, the /MCS2 output) is used to give the software a way to hardware Reset the ISCC, IUSC, and (M)USC. This allows a customer's program to operate as if it were in a target system starting from Reset, including the initial write to the Bus Configuration Register (BCR).

The 80186's two integrated DMA channels can be used for any two of the four or six serial data streams in the B side of the (E)SCC and the (M)USC. The "DMA EPLD" derives requests for the 80186's two DMA channels from six inputs, two each for (E)SCC channel B and the one or two channels in the (M)USC. It asserts DREQ0 or DREQ1 (High) if any of the inputs for that channel is low, and the 80186 is not performing an Interrupt Acknowledge cycle. Jumper blocks J22, J23, J24, and J29 control the assignment of the 80186's internal DMA controllers, including provision for a clipped Tx request that is needed if a standard SCC is installed in place of the ESCC. The various possibilities are summarized in Table 1.

Table 1. 80186 DMA Jumper Connections

To enable the following to use 80186 DMA Channel 0:	Install this jumper:
(E)SCC B Rx	J23-1 to J23-2
MUSC Rx or USC A Rx	J22-1 to J22-2
MUSC Tx or USC A Tx	J22-4 to J22-2
USC B Rx	J29-1 to J29-2
USC B Tx	J29-4 to J29-2

To enable the following to use 80186 DMA Channel 1:	Install this Jumper:
ESCC B Tx	J24-1 to J24-3
(E)SCC B Tx w/early release	J24-1 to J24-2
MUSC Rx or USC A Rx	J22-1 to J22-3
MUSC Tx or USC A Tx	J22-4 to J22-3
USC B Rx	J29-1 to J29-3
USC B Tx	J29-4 to J29-3

If more than one channel among the ESCC B and (M)USC are enabled for one of the 80186's internal DMA channels, software must ensure that only one of the enabled devices makes requests during a given block transfer. This can be done by leaving an entire Receiver or Transmitter idle or disabled, or by programming the device so that the DMA request is not output on the pin.

The ISCC and IUSC handle their own DMA transfers via the 80186's HOLD/HLDA facility.

Note: Either a Z16C33 MUSC or a Z16C30 USC can be installed in socket U5. If this is done, references to the (M)USC herein after may mean the USC as a whole or just its channel A; which one should be clear from the context.

The inputs and outputs associated with the processor's integrated counter/timer facility are brought to the pin header labelled J26 so that they can be used in applications (Table 2).

Table 2. Counter/Timer Signal Locations

J26 pin	Signal
1	Timer In 1
2	Timer Out 1
3	Timer In 0
4	Timer Out 0
5	N/C
6	Ground

The 80186's integrated interrupt controller is largely bypassed in favor of the traditional Zilogical interrupt daisy-chain structure.

Push buttons are provided for Reset and Non-Maskable Interrupt (NMI). A means to generate an NMI, in response to a Start bit received from the user's PC or terminal, is also provided. The first transmitted Start bit on the RS-232. Console connector J1, after a Reset, also produces an NMI; this feature can be used to find which serial controller channel is connected to the Console connector.

Address Map

EPROM is located at the highest addresses, and its size is programmable in the 80186 for the /UCS output. The addresses of the datacom controllers are programmed in the 80186 for the /PCS6-/PCS0 outputs, as a block of 128x7=896 bytes starting at a 1 Kbyte boundary. The block can be in I/O space or in a part of memory space that is not used for SRAM or EPROM. The starting 1 Kbyte boundary is called (PBA) in the following sections.

RAM extends upward from address 0.

Using 128K x 8 SRAMs and 64K x 8 EPROMs, the address map might be as shown in Table 3.

Table 3. Suggested Address Map

RAM	00000-BFFFF
(E)SCC	D8000, 2, 4, 6 or D8000-D803E (even addrs only)
ISCC	D8080-D80FE (even addrs only)
(M)USC	D8100-D81FF
IUSC	D8200-D837F
ISCC-IUSC-(M)USC Reset	DB000-DB7FF (if enabled)
27512 EPROM	E0000-FFFFF

EPROM

Two 28-pin EPROM sockets are provided; both must be populated in order to handle the 80186's 16-bit instruction fetches. Jumper header J18 allows the sockets to be compatible with 2764s, 27128s, 27256s, or 27512s; it is jumpered at the factory to match the EPROMs provided. For 27512s only, jumper J18-J2 to J18-J3 and leave J18-J1 open. For 2764s, 27128s, or 27256s, jumper J18-J2 to J18-J1 and leave J18-J3 open.

Note: J18 connects pin 1 of both sockets to either A16 or Vcc. This is done because for 2764s, 27128s, and 27256s, pin 1 is Vpp which may require a high voltage and/or draw more current than a normal logic input. For 2764s and 27128s, a similar jumper might be provided in some designs for pin 27 (/PGM). As long as the address for /UCS is programmed as described in the next paragraph, A15 (which is connected to pin 27) is High whenever /UCS is Low, so that 2764s and 27128s operate correctly.

The first code executed after Reset should program the 80186's Chip Select Control Registers to set up the address ranges for which outputs like /UCS and /PCS6-/PCS0 are asserted. In particular, the UMCS register (address A0H within the 80186's Peripheral Control Block) must be programmed to correspond to the size of EPROMs used (Table 4).

EPROM (Continued)

Table 4. EPROM Address Ranges

EPROM Type	UMCS Value	EPROM	Address Range
	2764	FC3C	FC000-FFFFF
	27128	F83C	F8000-FFFFF
	27256	F03C	F0000-FFFFF
	27512	E03C	E0000-FFFFF

The three LSBs of the above UMCS values are all 100, which signifies no external Ready/WAIT is used and no wait states are required. If the EPROMs are not fast enough for no-wait-state operation, making the three LSBs 101, 110, or 111 extends EPROM cycles by 1, 2, or 3 wait states, respectively.

RAM

Six 32-pin sockets are provided; they should be populated in pairs, starting with the lower-numbered sockets, to allow for 16-bit accesses. V_{CC} is provided at both pin 32 and pin 30 so that 28-pin 32K x 8 SRAMs can be installed in pins 3-30 of the sockets. Jumper block J19 allows decoding of the Chip Select signals from A17-A16 for 32K x 8 SRAMs or from A19-A18 for 128K x 8 SRAMs. The six standard memory populations are:

One pair of 32K x 8 devices:	64 Kbytes at 00000-0FFFF
Two pairs of 32K x 8 devices:	128 Kbytes at 00000-1FFFF
Three pairs of 32K x 8 devices:	192 Kbytes at 00000-2FFFF
One pair of 128K x 8 devices:	256 Kbytes at 00000-3FFFF
Two pairs of 128K x 8 devices:	512 Kbytes at 00000-7FFFF
Three pairs of 128K x 8 devices:	768 Kbytes at 00000-BFFFF

J19 is factory set according to the size of the SRAMs provided. For 32K x 8 SRAMs, jumpers are installed between J19-J2 and J19-J3, and between J19-J5 and J19-J6, with J19-J1 and J19-J4 left open. For 128K x 8 SRAMs, jumpers are installed between J19-J1 and J19-J2, and between J19-J4 and J19-J5, with J19-J3 and J19-J6 left open.

32K x 8 SRAMs have cyclic/redundant addressing starting at 40000, 80000, and C0000. The only configuration in which this causes problems is with three pairs of 32K x 8 SRAMs and 27512 EPROMs; in this case, there is a conflict in the range E0000-EFFFF. This conflict can be avoided by any of the following means:

- Using two pairs of 32K x 8 SRAMs;
- Using one pair of 128K x 8 SRAMs;
- Using 27256 EPROMs, or
- Using 27512 EPROMs but programming the size of /UCS like they are 27256s.

Since the /LCS output of the 80186 is not used, the LMCS register in the 80186 is not written with any value.

Programming the Peripheral Chip Selects

The 80186 allows the /PCS6-/PCS0 pins, which in this case select the various datacom controllers, to be asserted for a selected 896-byte block of addresses. The block may reside in either memory or I/O space depending on the values programmed into the PACS and MPCS registers, locations A4H and A8H of the 80186's Peripheral Control Block, respectively. The choice of address space depends on the needs of the customer's application and the configuration of software supplied with the board (Table 5).

Table 5. Three Standard Alternatives for Serial Controller Addressing

Basic Requirement	Base Address (PBA)	PACS value	MPCS value
I/O Space	8000	0838	81B8
Memory Space, 32K x 8 SRAMS used	38000	3838	81F8
Memory Space, 128K x 8 SRAMs used	D8000	D838	81F8

The three LSBs of the PACS value specify the Ready/WAIT handling for the /PCS3-/PCS0 lines which select the (E)SCC, ISCC, and (M)USC. The three LSBs of the MPCS value specify the Ready/WAIT handling for the /PCS4, 5, and 6 lines, which select the IUSC. Both fields are shown here with the LSB's 000, signifying that the 80186 should honor a WAIT on the external Ready/WAIT signal, but that it should not provide any minimum wait.

Programming the Mid-Range Memory to Reset the ISCC, IUSC, and (M)USC

A Reset puts the ISCC, IUSC, and (M)USC in a special and unique state in which the first write to each device implicitly goes to a Bus Configuration Register (BCR) that controls the device's basic bus operation; the BCR is not accessible thereafter. So that this board can serve as a complete development environment for customers' software, it includes a means whereby software (e.g., the debug monitor) can assert the /RESET input of these three devices. Specifically, assertion of the /MCS2 output of the 80186 causes such a Reset.

The 81 in the MS Byte of the MPCS values, shown in Table 5, makes each of the /MCS3-/MCS0 pins correspond to a 2 Kbyte block of addresses in memory space. The actual active pin addresses are determined by the value written into the MMCS register; location A6H of the 80186' Peripheral Control Block. Table 6 shows suggested MMCS values as a function of the RAM chip size, and the corresponding range of addresses for which any read or write access causes the three controllers to be reset.

8

Table 6. Address Ranges for Reset

RAM Size	MMCS value	Address Range for which ISCC, IUSC, and (M)USC are Reset:
32K x 8	3BFF	3B000-3B7FF
128K x 8	DBFF	DB000-DB7FF

The three LSBs of the above MMCS values are 111 so that the longest possible Reset pulse is generated when any of the locations in the indicated range are accessed.

Note that if this feature is not needed, it can be disabled by simply not programming the MMCS register.

Interrupt Daisy Chain (Priority) Order

Jumper block J25 selects whether the (E)SCC device is at the start or the end of the interrupt daisy chain.

To make the interrupt priority be:	Jumper J25 as follows:
(E)SCC highest, IUSC, ISCC, (M)USC lowest	J25-J2 to J25-J3, J25-J4 to J25-J5 (J25-J1, J25X open)
IUSC highest, ISCC, MUSC, (E)SCC lowest	J25-J1 to J25-J2, J25-J to J25-J4 (J25-5J, J25X open)
IUSC highest, ISCC, USC, (E)SCC lowest	J25X to J25-J2, J25-J3 to J25-J4 (J25-J1, J25-J5 open)

This variability is provided in part because early versions of the 85230 ESCC had trouble passing an interrupt acknowledge down the daisy chain if it occurred in response to a lower-priority device's request just as the ESCC was starting to make its own request. Current 85230's don't have the problem.

(E)SCC

Socket U2 can be configured for either an ESCC or SCC, and for versions thereof that use either multiplexed or non-multiplexed address and data. Jumper blocks J20 and J21 select certain signals accordingly. For a part with multiplexed addresses and data (80x30), jumper J20-J1 to J20-J2 and leave J20-J3 open, and jumper J21-J1 to J21-J2 and J21-J4 to J21-J5, leaving J21-J3 and J21-J6 open. With such a part, software can directly address the (E)SCC's registers, and need not concern itself with writing register addresses to Write Register 0 (WR0).

For a part having a non-multiplexed bus (85x30), jumper J20-J2 to J20-J3, J21-J2 to J21-J3, and J21-J5 to J21-J6, leaving J20-J1, J21-J1, and J21-J4 open. In this case, software must handle the (E)SCC by writing register addresses into its WR0 in order to access any register other than WR0, RR0, or the data registers.

Channels A and B can be handled on a polled or interrupt-driven basis. Channel A of the (E)SCC is suggested for connecting the user's PC or terminal for use with the Debug Monitor included in this evaluation kit. Channel B (but not A) can be handled on a DMA basis using the 80186's internal DMA channels, or on a polled or interrupt driven basis.

Jumper block J23 allows channel B's /W//REQB output to be used for either a Wait function or a Receive DMA Request function. To use the output for Wait, jumper J23-J2 to J23-J3 and leave J23-J1 open. The Wait function is only significant if the software wants to delay completion of a Read from the (E)SCC's Receive Data register until data is available, and/or if it wants to delay completion of a Write to the Transmit Data register until the previously-written character has been transferred to the Transmit Shift register. These modes are alternatives to checking the corresponding status flags and can be used to achieve operating speeds higher than those possible with such traditional polling, although not as fast as the speeds possible with a DMA approach.

To use the /W//REQB output as a Receive DMA Request, jumper J23-J1 to J23-J2 and leave J23-J3 open.

Jumper block J24 determines how channel B's /DTR/ /REQB output is used. To use this output for the Data Terminal Ready function, jumper J24-J3 to J24-J4 and leave J24-J1 and J24-J2 open. To use this output directly as a Transmit DMA Request (using the ESCC's early-release capability), jumper J24-J1 to J24-J3 and leave J24-J2 and J24-J4 open. To drive the Transmit DMA Request with a clipped version of this signal that is forced High earlier than a standard SCC drives it High, jumper J24-J1 to J24-J2 and leave J24-J3 and J24-J4 open.

The "SCC EPLD" handles the (E)SCC's signalling requirements. Among other things, this EPLD configures the (E)SCC socket's pins 35 and 36 for either a multiplexed or non-multiplexed part, based on whether J20 is jumpered to connect the 80186 ALE signal to one of its input pins. If the device detects high-going pulses on this input, it drives corresponding low-going Address Strobe pulses onto (E)SCC pin 35 and drives low-going Data Strobe pulses onto (E)SCC pin 36.

If the SCC EPLD's pin 9 stays at Ground, the part drives Read strobes onto pin 36 and drives delayed Write strobes onto pin 35, for a non-multiplexed 85x30 device.

While the ESCC's relaxed timing capability allows the 80186's /WR output to be connected directly to the /WR input of a non-multiplexed ESCC, the SCC EPLD delays start of an SCC's write cycle until write data is valid, even though this is not necessary for an ESCC.

The SCC EPLD also generates the clipped-DMA-request signal mentioned in connection with J24, and logically ORs Reset onto pins 35 and 36. The device also tracks the two IACK cycles provided by the 80186 for each Interrupt Acknowledge cycle. For a multiplexed address/data port, it drives the address strobe (only) on the first cycle, and it provides the /RD or /DS pulse needed by the (E)SCC (only) on the second cycle. The "DMA EPLD" provides the INTACK signal needed by the (E)SCC.

The (E)SCC is only accessible at even addresses. For a non-multiplexed part (85x30), the following four register locations are repeated throughout the even addresses from (PBA) through (PBA)+126:

(PBA), (PBA)+8, ... (PBA)+120	Channel B Command/Status register
(PBA)+2, +10, ... (PBA)+122	Channel B Data register
(PBA)+4, +12, ... (PBA)+124	Channel A Command/Status register
(PBA)+6, +14, ... (PBA)+126	Channel A Data register

For a multiplexed part (80 x 30), the Select Shift Left command (D1-0=11) should be written to Channel B's WR0 before any other registers are accessed. Then the basic (E)SCC register map occurs twice in the even addresses from (PBA) through (PBA)+126:

(PBA), (PBA)+2, ... (PBA)+30	Channel B registers 0-15
(PBA)+32, +34, ... (PBA)+62	Channel A registers 0-15
(PBA)+64, +66, ... (PBA)+94	Channel B registers 0-15
(PBA)+96, +98, ... (PBA)+126	Channel A registers 0-15

The redundant addressing of the (E)SCC is used to control a feature that can be used by software to allow the user to interrupt software execution from his keyboard. If the (E)SCC is read at an address with A6-A5=11 (for a multiplexed part this means in the higher-addressed A channel), a mode is set in which a low on the console Received Data line (i.e., a Start bit on pin 3 of the J1 connector) causes a Non-Maskable Interrupt on the 80186. The mode is cleared by Reset, or when the (E)SCC is read at an address with A6-A5=10 (on a multiplexed part, in the higher-addressed B channel). The NMI handler should do the latter fairly quickly to prevent subsequent data bits on Received Data from causing further NMIs.

8

ISCC

Since the 80186 processor provides multiplexed addresses and data, the ISCC is configured to use the addresses on the AD lines. Therefore, software can address the various ISCC registers directly, and need not be concerned with writing register addresses into the indirect address fields of the ISCC's WR0 and CCAR.

Because the ISCC includes four DMA channels, its Channel A and B Transmitters and Receivers can be handled on a polled, interrupt-driven, and/or DMA basis, in any mixture.

Since the ISCC can only be programmed as an 8-bit device on the AD7-AD0 lines, it occupies only the even-addressed bytes within its address range, (PBA)+128 through (PBA)+254.

The first write to this address range, after a Reset, implicitly writes the ISCC's Bus Configuration Register (BCR). To match up with the rest of the board's hardware, this first write should be a byte write that stores the hexadecimal value C6 in any even address in the first half of the ISCC's address range [(PBA)+128 through (PBA)+190]. Details of this transaction are as follows:

- The High induced by a pull-up resistor on the ISCC's A/B input selects the WAIT protocol on the /WAIT//RDY pin, which corresponds to how the 80186 works. (In subsequent register accesses, the A/B selection is taken from A5 of the multiplexed address.)
- A Low on the ISCC's SCC//DMA input, which is connected to A6, is required by the internal logic of the ISCC. This is why the BCR write is restricted to the first half of the ISCC's address range.
- As with all transactions between the 80186 and ISCC, the address must be even because the ISCC only accepts slave-mode data on the AD7-AD0 pins.
- The MSB of the data (D7) is 1 to enable the Byte Swap feature, so that when the ISCC's DMA controller is reading transmit data from RAM, it takes alternate bytes from AD7-AD0 and AD15-AD8.
- D6 of the data is 1 so that when the ISCC's DMA controller is reading transmit data from RAM, it takes even-addressed bytes from D7-D0 and odd-addressed bytes from D15-D8 (same function as the 80186).
- D2-D1 of the data are 11 to select double-pulsed mode for the ISCC's /INTACK input. Again, this is how the 80186 works.
- D0 of the data is 0 to select Shift Left Address mode so that the ISCC subsequently takes register addressing from the AD5-AD1 lines rather than from AD4-AD0. This is because the 80186 is a 16-bit processor that locates even-addressed bytes on AD7-AD0 and odd-addressed bytes on AD15-AD8, but the ISCC only accepts slave-mode writes on the AD7-AD0 pins.

ISCC (Continued)

- The fact that the ISCC's internal logic sees activity on its /AS pin, which is inverted from the 80186' ALE signal, automatically conditions it for a multiplexed Address/Data bus.

Given that the BCR is written as above, the ISCC's slave-mode address map is as follows:

(PBA)+128, 130, ..., (PBA)+190	DMA Controller Registers
(PBA)+192, 194, ..., (PBA)+222	ISCC Serial Channel B registers 0-15
(PBA)+224, 226, ..., (PBA)+254	ISCC Serial Channel A registers 0-15

(M)USC

Since the 80186 processor provides multiplexed addresses and data, the (M)USC is configured to use the addresses on the AD lines. Therefore, the software need not write register addresses into the indirect address field of the (M)USC's CCAR.

The (M)USC's Transmitter and Receiver can be handled on a polled or interrupt-driven basis. In addition, any two of the Receivers and Transmitters in the (M)USC and Channel B of the (E)SCC can be handled on a DMA basis, using the 80186's integrated DMA controllers.

Jumper block J22 connects the (M)USC's /RxREQ and /TxREQ outputs to the "DMA EPLD" that makes the DMA Requests to the 80186. As shipped from the factory, jumpers are installed between J22-J1 and J22-J2, and between J22-J3 and J22-J4. In this configuration, the (M)USC's /RxREQ drives the 80186 DREQ0, and (M)USC /TxREQ drives the 80186 DREQ1. To reverse this assignment, jumper J22-J1 to J22-J3 and J22-J2 to J22-J4. To disconnect the (M)USC from one or both of the 80186's DMA channels, remove one or both jumpers (put them in a safe place in case you change your mind). Jumper block J29 provides the same connection-variability for the /RxREQ and /TxREQ outputs of Channel B of a USC.

Since the 80186's DMA channels are not capable of fly-by operation, the (M)USC's /RxACK and /TxACK pins have no dedicated function. They can be used for Request to Send and Data Terminal Ready; the two signals are lightly pulled up since they are not driven after Reset.

The (M)USC can be programmed using 16-bit data on the AD15-AD0 lines or 8-bit data on AD15-AD8 and AD7-AD0. It makes the distinction between 8-bit and 16-bit operations as part of its address map rather than through a control input. The PS pin of an MUSC, or the A//B pin of a USC, is connected to a latched version of 80186 A7. The D//C pin of the (M)USC is grounded. The overall address range of the (M)USC is 256 bytes, between (PBA)+256 and (PBA)+511.

The first write to this address range, after a Reset, implicitly writes the (M)USC's Bus Configuration Register (BCR). To match the rest of the board's hardware, this first write should be a 16-bit write, storing the hex value 0007 at any address in the second half of the (M)USC's range [any address in (PBA)+384 through 510, i.e., in the A channel of a USC]. Details of this transaction are as follows:

- The High on the PS or A//B input, which is connected to A7, selects the WAIT protocol on the /WAIT//RDY pin, corresponding to how the 80186 works.
- The MSB of the data (D15) is 0 because a separate non-multiplexed address is not wired to pins AD13:8 of the (M)USC.
- Bits 14-3 are required to be all zeros by the (M)USC's internal logic.
- D2 of the data is 1 to tell the (M)USC that the data bus is 16 bits wide.
- D1 of the data is 1 to select double-pulsed mode for the (M)USC's /INTACK input. This is how the 80186 works.
- D0 of the data is 1 to select Shift Right Address mode so that the (M)USC subsequently takes register addressing from the AD6-AD0 lines rather than from AD7-AD1.
- The fact that the (M)USC's internal logic sees activity on its /AS pin, which is inverted from the 80186' ALE signal, automatically conditions it for a multiplexed Address/Data bus.

Given that the BCR is written as above, the (M)USC address map is as follows:

Starting Addr	Ending Addr	Registers Accessed
(PBA)+256	(PBA)+319	16-bit access to MUSC regs or USC channel B regs
(PBA)+320	(PBA)+383	8-bit access to MUSC regs or USC channel B regs
(PBA)+384	(PBA)+447	16-bit access to MUSC regs or USC channel A regs
(PBA)+448	(PBA)+511	8-bit access to MUSC regs or USC channel A regs

Note:
To maximize compatibility, program an MUSC using the second half of this range, (PBA)+384 through (PBA)+511.

While the ESCC and ISCC can drive their Baud Rate Generators from their PCLK inputs, the (M)USC has no such input. The 80186 clock output SYSCLK is brought to pins 7 of J9, J10, and J12, at which point it can be jumpered to pin 9 or 8 so that it is routed to the /TxC or /RxC pin of the device.

IUSC

Since the 80186 processor provides multiplexed addresses and data on the AD lines, the IUSC is configured to use these addresses. Software need not write register addresses into the indirect address fields of the IUSC's CCAR and DCAR.

The IUSC's two DMA channels allow its Receiver and Transmitter to be handled on a polled, interrupt-driven, or DMA basis, in any combination.

The IUSC can be programmed using 16-bit data on the AD15-AD0 lines or 8-bit data on AD15-AD8 and AD7-AD0. The distinction between 8-bit and 16-bit operations is made as part of the address map rather than via a control input. The D//C pin of the IUSC is driven from A7 during slave cycles, and the S//D pin is driven from A8. The overall address range of the IUSC is 384 bytes from (PBA)+512 through (PBA)+895.

The first write to this address range, after a Reset, implicitly writes the IUSC's Bus Configuration Register (BCR). To match up with the rest of the board's hardware, this first write is a 16-bit write, storing the recommended hex value 00F7 at any word address in the range (PBA)+768 through (PBA)+830. Details of this transaction are as follows:

- The High on the IUSC's S//D input, which is connected to A8, selects the WAIT protocol on the /WAIT//RDY pin, which is how the 80186 works.
- It may not be required for this initial write, but it is good programming form for A6 to be zero since this is a word write. This and the previous point determine the recommended address range.
- The MSB of the data (D15) is 0 because a separate non-multiplexed address is not wired to pins AD13:8 of the IUSC.
- Bits 14-8 are more or less required to be all 0 by the IUSC's internal logic.
- D7-D6 are 11 to allow the DMA controllers to do either 16-bit transfers, or alternating byte transfers on AD7-AD0 for even-addressed bytes and on AD15-AD8 for odd-addressed bytes. This is compatible with 80186 byte ordering.
- D5-D4 of the data are 11 to select double-pulsed mode for the IUSC's /INTACK input. Again, this is how the 80186 works.
- D3 of the data is 0 to select open-drain mode on the IUSC's /BUSREQ pin. The board's control logic also drives this signal low when the ISCC asserts its Bus Request output.
- D2 of the data is 1 to tell the IUSC that the data bus is 16 bits wide.

IUSC (Continued)

- D1 of the data is 1 to select open-drain mode on the IUSC's /INT pin which is OR-tied with the interrupt request from the (E)SCC.
- D0 of the data is 1 to select Shift Right Address mode, so that the IUSC subsequently takes register addressing from the AD6-AD0 lines rather than from AD7-AD1.
- The fact that the IUSC's internal logic sees activity on its /AS pin, which is inverted from the 80186' ALE signal, automatically conditions it for a multiplexed Address/Data bus.

Given that the BCR is written as above, the IUSC slave-mode address map is as follows:

Starting Addr	Ending Addr	Registers Accessed
(PBA)+512	(PBA)+575	16-bit access to IUSC Transmit DMA registers
(PBA)+576	(PBA)+639	8-bit access to IUSC Transmit DMA registers
(PBA)+640	(PBA)+703	16-bit access to IUSC Receive DMA registers
(PBA)+704	(PBA)+767	8-bit access to IUSC Receive DMA registers
(PBA)+768	(PBA)+831	16-bit access to IUSC Serial Controller registers
(PBA)+832	(PBA)+895	8-bit access to IUSC Serial Controller registers

While the ESCC and ISCC can drive their Baud Rate Generators from their PCLK inputs, the IUSC cannot do this from its CLK input. The 80186 clock output SYSCLK is brought to pins 7 of J9, J10, and J12 at which point it can be jumpered to pin 9 or 8 so that it is routed to the /TxC or /RxC pin of the device.

Since the IUSC contains its own DMA channels, its /RxREQ and /TxREQ pins have no dedicated function. They can be used for Request to Send and Data Terminal Ready; the two signals are lightly pulled up to allow for the fact that they are not driven after Reset.

SERIAL INTERFACING

The serial I/O pins of the four serial controllers are connected to the six connector blocks labelled J5 through J10. In addition, the port pins of the IUSC are connected to the J11 connector block, and the port pins of an MUSC or the B channel of a USC are connected to J12. These connector blocks can be interconnected for communication between on-board serial controllers, or they can be connected to the user's custom communications hardware on another board. As a third option, they can be connected to three on-board serial interfaces via the connector blocks labelled J13 through J15.

Two of the on-board serial interfaces use EIA-RS-232 signal levels and pin arrangement. 25-pin D connectors J1A or J2A are configured as DTE, while J1B and J2B are configured as DCE. These serial interfaces are used by connecting one of J5-J10 to J13 or J14, respectively. J1B is typically used for connection to the user's PC or terminal.

The third on-board serial interface uses EIA-422 signal levels on connector J3A,J3B, or J4, and is used by connecting one of J5-J10 to J15. The 25-pin D connector J3A uses the DTE pin arrangement put forth in the EIA-530 standard. J3B is a DCE version of EIA-530, while the 8-pin circular DIN connector, J4, is compatible with the Apple Macintosh Plus and later Macintoshes, and thus with AppleTalk/LocalTalk equipment.

The serial interface connectors are summarized in the following tables:

Table 7. Controller Port Connectors

To use the following serial controller channel with off-board or on-board serial hardware:	Connect to this (these) 10-pin connector block(s):
(E)SCC Channel A	J5
(E)SCC Channel B	J6
ISCC Channel A	J7
ISCC Channel B	J8
IUSC	J9 (J11 for Port pins)
(M)USC	J10 (J12 for MUSC Port pins or USC channel B)

Table 8. On-Board Line Driver/Receiver Connectors

To use a serial controller with the following on-chip serial interface:	Connect the connector(s) from the previous table to:
J1A or J1B EIA-RS-232 Console	J13
J2A or J2B EIA-RS-232	J14
RS-422 differential: J3A or J3B EIA-530 or J4 Circular-8 (DIN)	J15

The pin-out of the J5-J10 connectors is fairly consistent, but of necessity not identical because of differences among the various serial controllers:

Table 9. Pin Assignments of Standard Controller Connectors

Pin#	J5: (E)SCC A pin	J6: (E)SCC B pin	J7,8: ISCC pin	J9: IUSC pin	J10: MUSC or USC A pin	J12: USC B pin
1	TxD	TxD	TxD	TxD	TxD	TxD
2	RxD	RxD	RxD	RxD	RxD	RxD
3	/RTS	/RTS	/RTS	(N/C)	/RxACK	/RxACK
4	/CTS	/CTS	/CTS	/CTS	/CTS	/CTS
5	/DTR	/DTR or (N/C) [1]	/DTR	(N/C)	/TxACK	/TxACK
6	/DCD	/DCD	/DCD	/DCD	/DCD	/DCD
7	/SYNC	/SYNC	/SYNC	(SYSCLK)	(SYSCLK)	(SYSCLK)
8	/RTxC	/RTxC	/RTxC	/RxC	/RxC	/RxC
9	/TRxC	/TRxC	/TRxC	/TxC	/TxC	/TxC
10	GND	GND	GND	GND	GND	GND
11	NA	NA	NA	/TxREQ	/TxREQ	/TxREQ
12	NA	NA	NA	/RxREQ	/RxREQ	/RxREQ

Notes:
[1] Controlled by the J24 jumper block: must be N/C if (E)SCC channel B transmitter is to be handled by an 80186 DMA channel.

SERIAL INTERFACING (Continued)

The ground pins are included as signal references with off-board hardware.

When interconnecting between two connectors among J5-J10, DO NOT jumper corresponding pins straight across, as this connects outputs to outputs and inputs to inputs. Rather, connect at least each pin 1 to the other pin 2, and enough opposing inputs and outputs as needed to make the communication protocol meaningful.

The pin-out of the 12-pin J13-J15 connectors is similar to that of J5-J10, but more extensive. To allow for the "DCE" connectors that were added in revision "B" of the board, J13 and J14 are 16-pin headers and J15 is a 14-pin one:

Table 10. Pin Assignments of Line Driver/Receiver Connectors

Pin #	J13-J14 DTE signal	J13-J14 DCE signal	J15 DTE signal	J15 DCE signal	Direction/where used
1	TxD	RxD	TxD	RxD	Output to J1-J4
2	RxD	TxD	RxD	TxD	Input from J1-J4
3	/RTS	/CTS	/RTS	/CTS	Output to J1-J3
4	/CTS	/RTS	/CTS	/RTS	Input from J1-J4 [3]
5	/DTR	/DSR	/DTR	/DSR	Output to J1-J4
6	/DSR	/DTR	/DSR	/DTR	Input from J1-J4
7		/DCD		/DCD	Output to J1B, J2B, J3B
8	/DCD		/DDC		Input from J1A, J2A, J3A, J4
9					
10	GND	GND	GND	GND	
11		/RxC		/RxC	Output to J1B, J2B, J3B
12	/RxC		/RxC		Input from J1A, J2A, J3A
13	/TxCO	/TxCI	/TxCO	/TxCI	Output to J1-3
14	/TxCI	/TxCO	/TxCI	/TxCO	Input from J1-3 [3]
15		/RI			Output to J1B, J2B
16	/RI				Input from J1A, J2A

Note:
[3] Various conventions have been used to combine synchronous clock inputs and modem control inputs on Apple Macintosh connectors similar to J4, as described in a later section.

Comparison of the two preceding charts leads to several conclusions:

- Pins 1-5 can always be jumpered straight across from a J5-J10 connector block to a J13-J15 connector block.
- In a synchronous environment, the Transmit clock can be either driven or received and the Receive clock can be received from the DTE connector or sent on the DCE connector.

The 10-pin J11 and J12 jumper blocks provide for connections to the Port pins of the IUSC and (M)USC, respectively. As with J5-J10, these connections may be to the customer's off-board custom circuits and/or to certain pins in the J13-J15 blocks. The following pin assignment is determined so that if a 2-channel USC is plugged into the (M)USC socket, J12 has the same pin-out for the USC's B channel as do J5-J10 for other channels.

Table 11. Pin Assignments of Controller Port Connectors

Pin #	J11: IUSC Signal	J12: (M)USC Signal
1	PORT1 (Clock 1 In)	PORT1
2	PORT4 (Xmit TSA Gate Out)	PORT4 (Xmit TSA Gate Out)
3	(N/C)	(N/C)
4	PORT0 (Clock 0 In)	PORT0
5	(N/C)	(N/C)
6	PORT3 (Rcv TSA Gate Out)	PORT3 (Rcv TSA Gate Out)
7	(N/C)	(SYSCLK)
8	PORT5 (Rcv Sync Out)	PORT5 (Rcv Sync Out)
9	PORT2	PORT2
10	GND	GND
11	PORT6 (Rcv Sync In)	PORT6 (Rcv Sync In)
12	PORT7 (Xmit Complete Out)	PORT7 (Xmit Complete Out)

Finally, an unpopulated 4-pin oscillator socket is included on the board with its output connected to a single jumper/wire-wrap pin. This socket can be populated with a user-supplied oscillator and connected to various clock pin(s) among J5-J15.

Sensing which Serial Controller Channel is connected to the Console

In order to use the software provided with this evaluation board, one of the serial controller channels must be connected to a Personal Computer (or a dumb terminal) via the J1 and J13 connectors. Some versions of this software may restrict the choice to (E)SCC Channel A or the (M)USC, depending on the user's applications needs, but there is nothing in the hardware that limits the choice of which serial channel is used for the Console. However, on the J1-J4 (J13-J15) side there are two things that are special about the J1/J13 section as compared to the others. One is the provision for a Non-Maskable Interrupt in response to a received Start bit, as described earlier in the section on (E)SCC addressing.

Software can use the other special feature of the J1/J13 section, after a Reset, to sense which serial channel is connected to the Console port. A Reset signal (from power-on or the Reset button, but not from the Reset-the-ISCC, etc., address decode as described earlier) puts the "NMI" EPLD in a special mode wherein the first Start bit on the Console's Transmit Data lead causes an NMI. This feature can be used in a start-up procedure like the following, to tell which serial controller channel is used for the Console:

For each serial controller channel that the software can use for the Console:

1. Initialize the channel.
2. Send a NUL character to the channel.
3. Wait a short time to see if an NMI occurs. If so, the current channel is the Console. If not, go on to the next serial channel and try again.

If none of the allowed serial channels produces an NMI, the user has not properly jumpered any J5-J10 connector block to the J13 block.

Basic software should use the serial controller channel for the Console in a very basic, polled way. Because of this and because of similarities between the (E)SCC and the ISCC, and between the (M)USC and the IUSC, note that software allows the Console to be connected to either the (E)SCC channel A or to the (M)USC; in fact, it includes most of the code necessary to use any of the six serial controller channels for the Console.

Notes on J4/Macintosh/AppleTalk/LocalTalk

The J4 connector is similar to that offered on various Macintosh systems. The ESCC and ISCC are particularly well adapted for use with this port, and development of USC family capability for AppleTalk/LocalTalk is of interest.

SERIAL INTERFACING (Continued)

The J3 and J4 connectors cannot be used simultaneously. The J16 jumper block controls whether the RS-422 driver for Transmit Data is turned "on" and "off" under control of the associated Request to Send signal, as on the Mac, or is "on" full time, which is more suitable for the use of J3. To put the TxD driver under control of RTS, jumper J16-1 to J16-J2 and leave J16-J3 open. For full-time drive on TxD (and also the J3 RTS pins), jumper J16-J2 to J16-J3 and leave J16-J1 open.

The J17 jumper block controls whether the reception of Data Carrier Detect and Clear to Send is differential (on J3) or unbalanced, as on J4. To use differential signalling from J3, remove all jumpers from J17.

On the initial Macintosh and subsequent ones as well, Apple did the unbalanced signalling backward from standard RS-423 and RS-232 polarity for the CTS lead (also called HSK and HSKI). If you are developing code for Macintosh hardware, you can preserve Mac compatibility by jumpering J17-J3 to J17-J5 and J17-J4 to J17-J6. This grounds the CTS- lead and connects the CTS+ lead to J4-J2. It also (assuming a standard source at the other end) inverts CTS to the opposite sense from that expected by the serial controller for functions such as auto-enabling. To make the CTS input of the serial controller have its normal (low-true) sense, jumper J17-J3 to J17-J4, and J17-J5 to J17-J6– this grounds the CTS+ lead and connects the CTS- lead to J4-J2.

The DTR (HSKO) output is provided in Apple systems from Mac Plus onward and has standard RS-423 (and RS-232) polarity.

The DCD input on J4-J7 is provided in Apple systems from the Mac II and SE onward, and also has standard polarity on Apple hardware. Jumper J17-J1 to J17-J2 to ground the "+" input of the receiver; the "–" lead is connected to J4-J7.

With jumpers installed to make DCD and CTS unbalanced, J4 can also be used for an additional RS-232 serial link. Connect a "Mac to Hayes modem" cable to J4, and optionally a null modem interconnect module to the other end. The cable internally grounds the RxD+ and TxD+ leads so that RxD– and TxD– act like RS-232 signals.

Macintosh systems also include provisions for synchronous clock inputs. It is not known whether these features are used by any applications, or attached hardware. On all known Macs, the SCC's TRxC pin is driven from the same signal as CTS; to be compatible with this feature, connect J15-J4 to pins 4 and 9 of the selected connector among J5-J10.

On the Mac SE, Mac II, and later models, a multiplexing scheme is provided on SCC channel A's RTxC pin to drive from either the same signal as DCD, or from an on-board 3.672 MHz clock. (Channel B always had the 3.672 MHz clock.) The former capability can be provided by connecting J15-J6 to pins 6 and 8 of the selected connector among J5-J10. The latter capability can be only approximated using the 80186 clock with different baud rate divisors, or by using another oscillator. (The board includes an unpopulated 4-pin oscillator socket that might be useful in this regard.)

JUMPER SUMMARY

Table 12 includes only those connector blocks intended to be populated by 2-pin option jumpers. J1-J15 and J26 are actual connectors meant for use with cables, jumper wires, or wire-wrapped connections.

Table 12. Two-Pin Option Jumpers

Jumpers	Installed	Open
J9-J7 thru -9	7 to 8: 80186 SYSCLK is IUSC /RxC 7 to 9: 80186 SYSCLK is IUSC /TxC	8: Something else on /RxC, or N/C 9: Something else on /TxC, or N/C
J10-J7 thru -9	7 to 8: 80186 SYSCLK is MUSC (USC A) /RxC 7 to 9: 80186 SYSCLK is MUSC (USC A) /TxC	8: Something else on /RxC, or N/C 9: Something else on /TxC, or N/C
J12-J7 thru -9	7 to 8: 80186 SYSCLK is USC B /RxC 7 to 9: 80186 SYSCLK is USC B /TxC	8: Something else on /RxC, or N/C 9: Something else on /TxC, or N/C
J16-J1 thru -3	1 to 2: J3, J4 TxD driven when RTS 2 to 3: J3, J4 TxD, RTS driven full-time	Must install one or the other
J17-J1 to -2	Unbalanced DCD- on J3 or J4	Differential DCD+, DCD- on J3
J17-J3 thru -6	3 to 5 and 4 to 6: CTS+ on J4-J2 3 to 4 and 5 to 6: CTS- on J3 or J4	Differential CTS+, CTS- on J3
J18-J1 thru -3	1 to 2: 2764, 27128, 27256 EPROMs 2 to 3: 27512 EPROMs	Must install one or the other
J19-J1 thru -6	1 to 2 and 4 to 5: 128K x 8 SRAMs 2 to 3 and 5 to 6: 32K x 8 SRAMs	Must install one way or the other
J20-J1 thru -3	1 to 2: U2 contains 80C30 or 80230 2 to 3: U2 contains 85C30 or 85230	Must install one way or the other
J21-J1 thru -6	1 to 2 and 4 to 5: U2 contains 80C30 or 80230 2 to 3 and 5 to 6: U2 contains 85C30 or 85230	Must install one way or the other
J22-J1 thru -4	1 to 2: MUSC (USC A) RxREQ on DMA 0 1 to 3: MUSC (USC A) RxREQ on DMA 1 2 to 4: MUSC (USC A) TxREQ on DMA 0 3 to 4: MUSC (USC A) TxREQ on DMA 1	 1: MUSC (USC A) Rx no DMA 4: MUSC (USC A) Tx no DMA
J23-J1 thru -3	1 to 2: (E)SCC B RxRQ on DMA 0 2 to 3: (E)SCC B Wait function	(E)SCC B neither Rx DMA nor Wait
J24-J1 thru -4	1 to 2: clipped SCC B TxREQ on DMA 1 1 to 3: direct ESCC B TxREQ on DMA 1 3 to 4: /DTR output from ESCC B	(E)SCC B neither Tx DMA nor /DTR
J25-J1 thru - 5 and J25X	1 to 2 and 3 to 4: (E)SCC last on IACK chain, MUSC second to last J25X to 2 and 3 to 4: (E)SCC last, USC 2nd to last 2 to 3 and 4 to 5: (E)SCC first on IACK chain	Must be one of these three ways
J28-J1 thru -6	1 to 2: 80186 SYSCLK is (E)SCC PCLK 3 to 4: 80186 SYSCLK is ISCC PCLK 5 to 6: 80186 SYSCLK is IUSC CLK	Connect some other clock to 2, 4, or 6
J29-J1 thru -4	1 to 2: USC B RxREQ on DMA 0 1 to 3: USC B RxREQ on DMA 1 2 to 4: USC B TxREQ on DMA 0 3 to 4: USC B TxREQ on DMA 1	 1: USC B Rx no DMA 4: USC B Tx no DMA

8

EPLD LOGIC

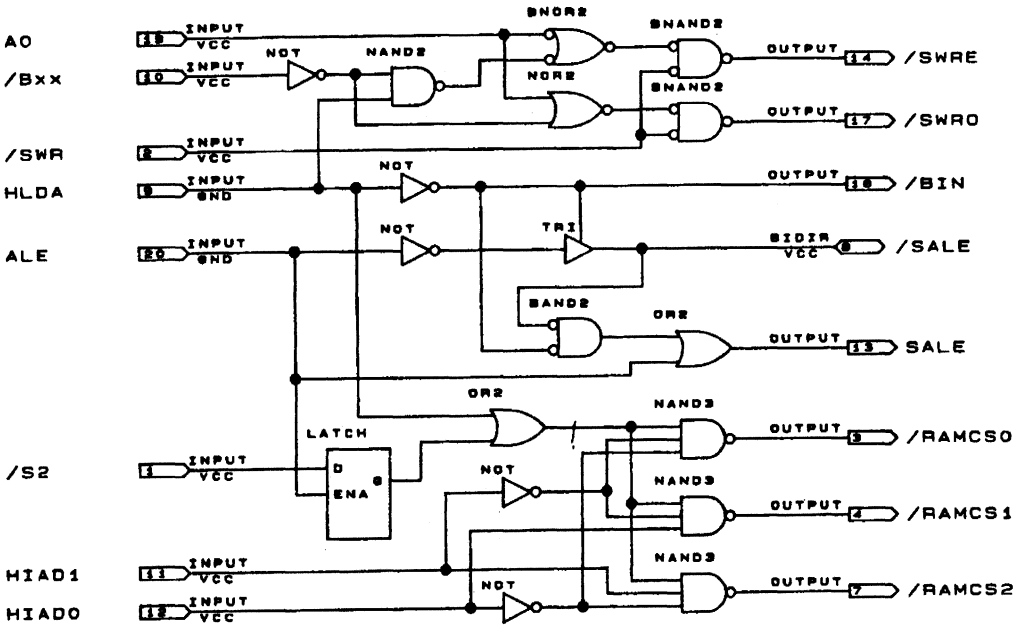
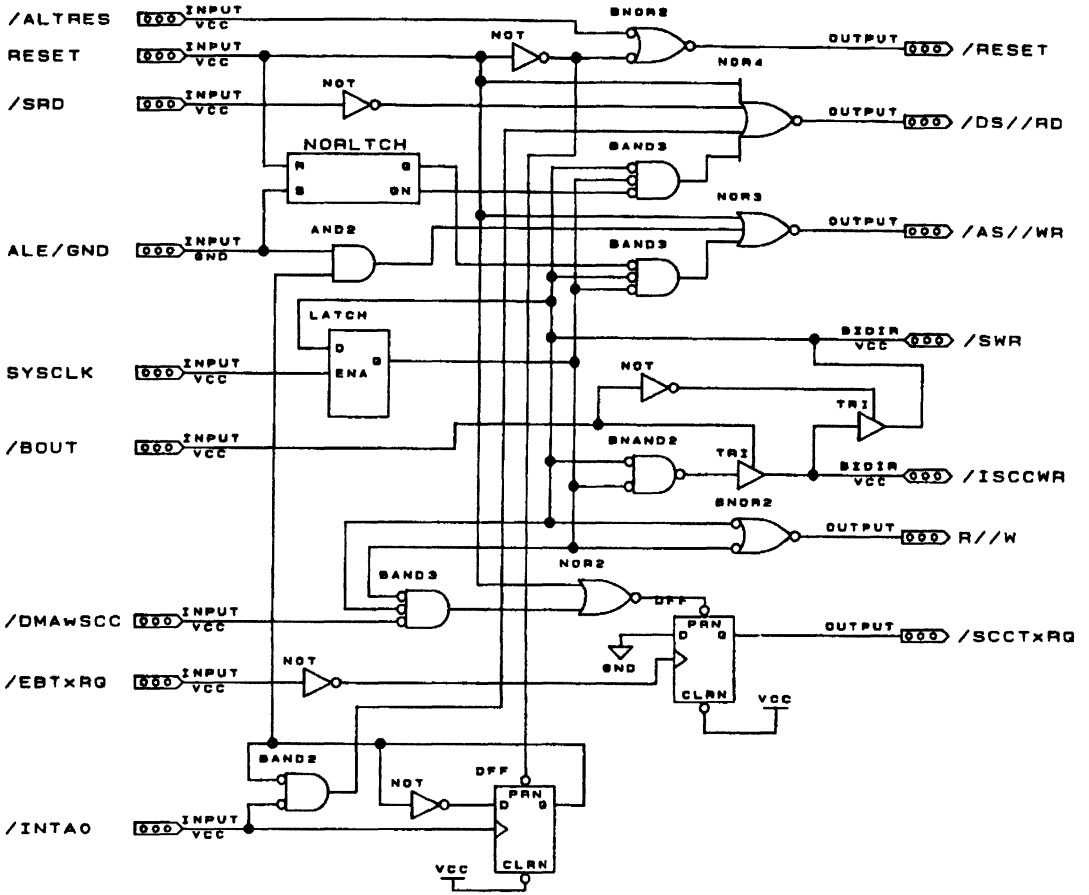


Figure 1. Control EPLD for 186 Board



Figuree 2. SCC EPLD for 186 Board

DMA EPLD LOGIC

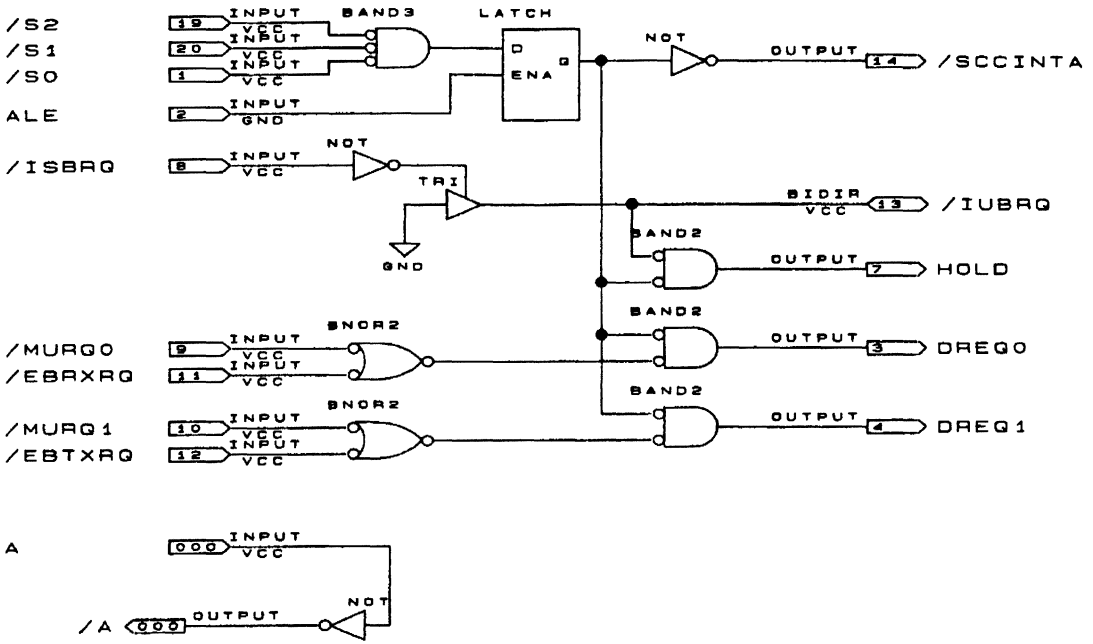


Figure 3. DMA EPLD for 186 Board

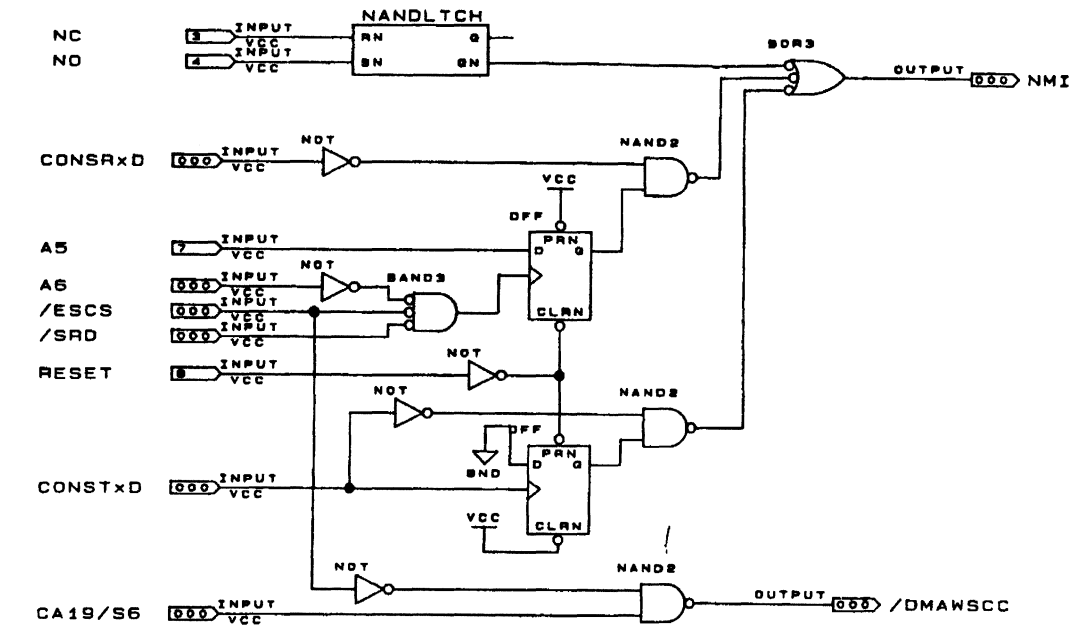


Figure 4. NMI EPLD for 186 Board

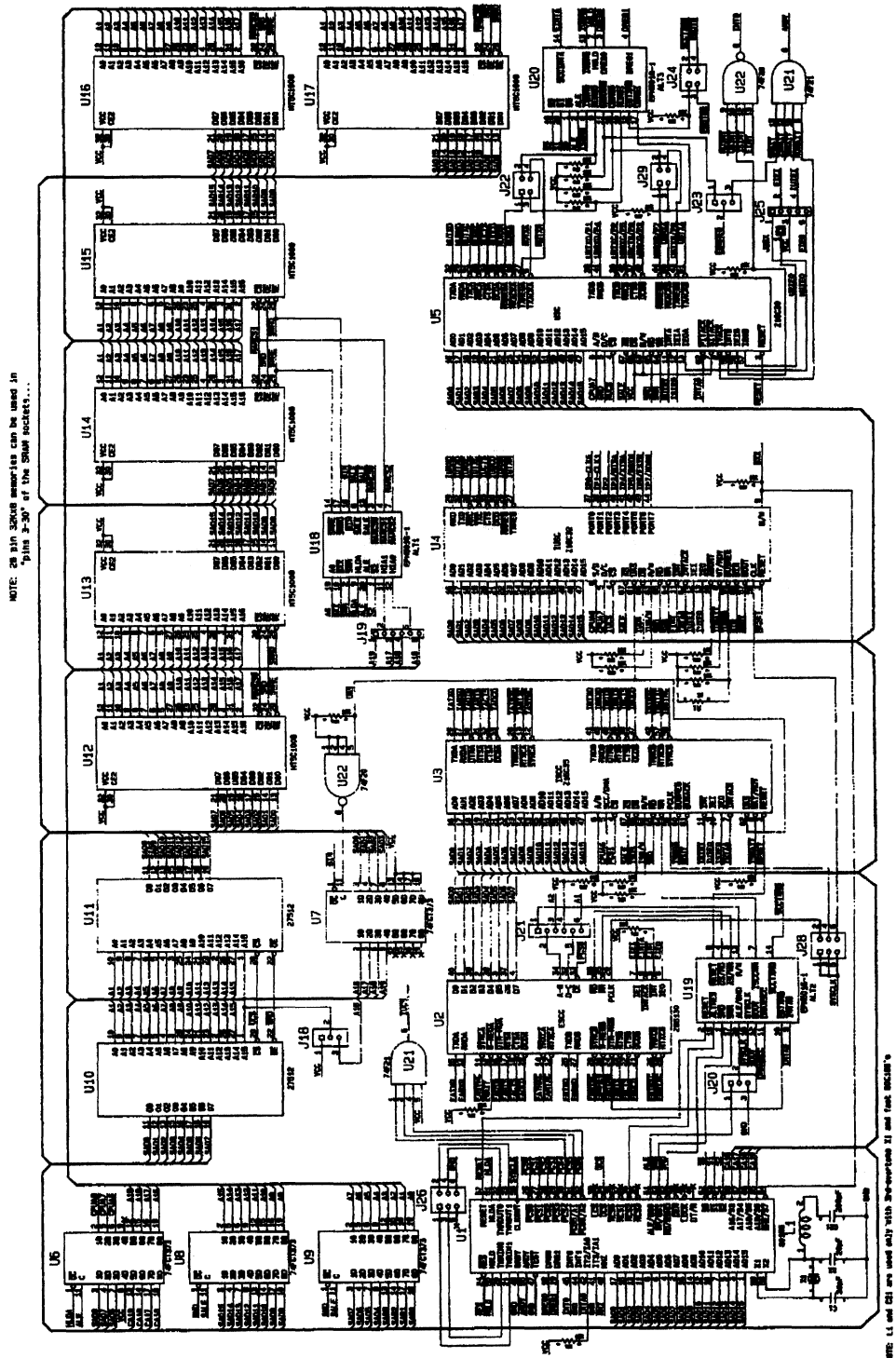


Figure 5. Schematic of the Evaluation Board

ON-CHIP OSCILLATOR DESIGN

Design and Build Reliable, Cost-Effective, Trouble-Free On-Chip Oscillator Circuits for a More Practical Design and 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 start-up, has resulted in an unreliable design and subsequent field problems (See on page 10 for reference materials and acknowledgments).

OSCILLATOR THEORY OF OPERATION

The circuit under discussion is called the Pierce Oscillator (Figures 1 and 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.

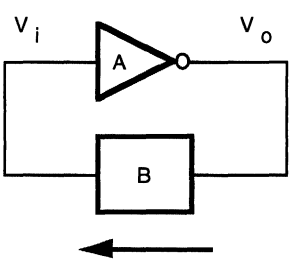


Figure 1. Basic Circuit and Loop Gain

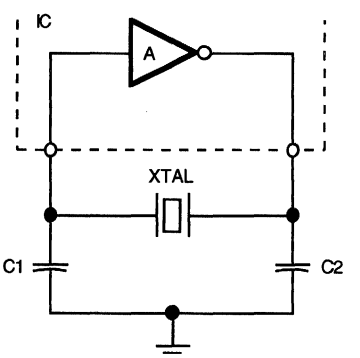


Figure 2. Zilog Pierce Oscillator

OSCILLATOR THEORY OF OPERATION (Continued)

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.

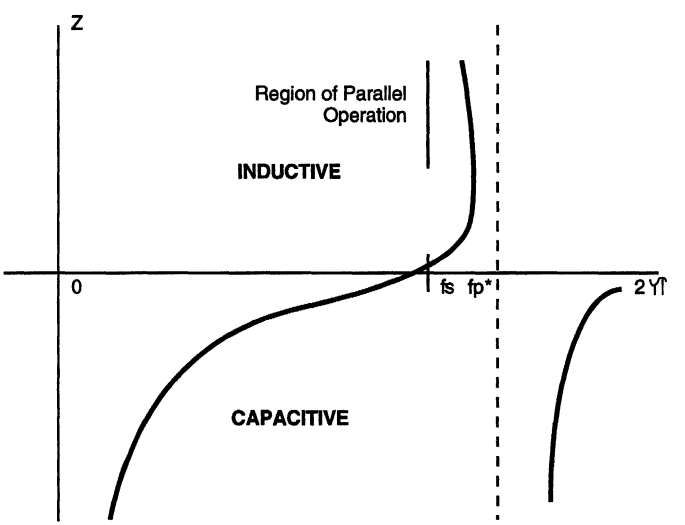
The described oscillator amplifies its own noise at start-up until it settles at the frequency which satisfies the gain/phase requirement AB = 1. This means loop gain equals one, and loop phase equals zero (360 degrees). To do this, the loop gain at points around the frequency of oscillation must be 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).

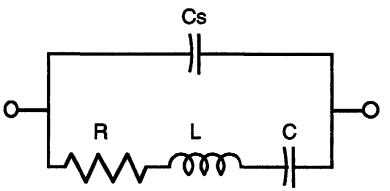
The series resonant frequency is given by:

Fs = 1/(2π x sqrt of LC),
where Xc and Xl are equal.

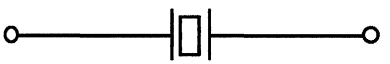
Thus, they cancel each other and the crystal is then R shunted by Cs with zero phase shift.

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.

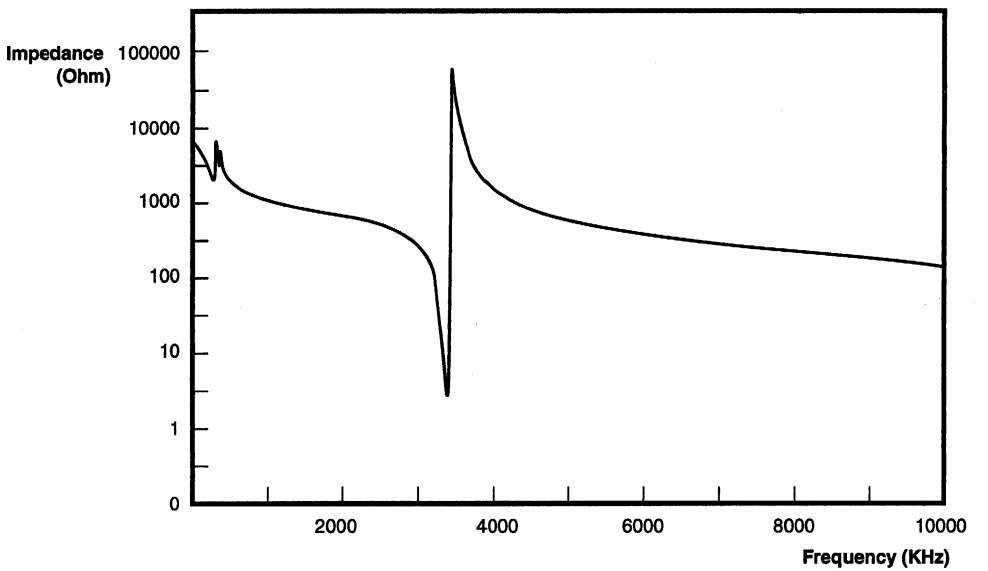


Figure 5. Ceramic Resonator Reactance

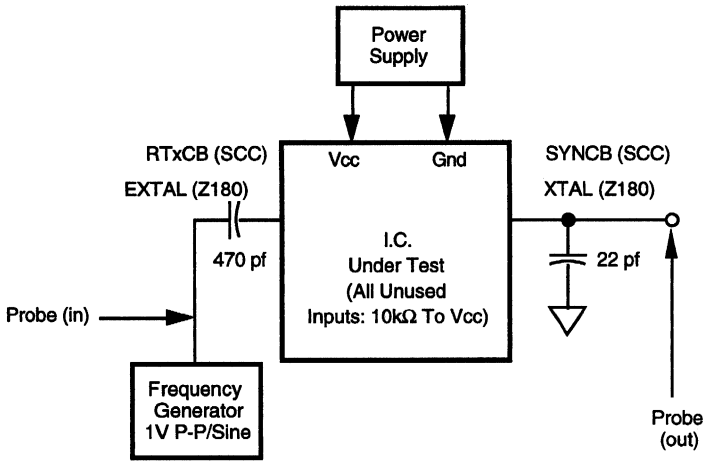


Figure 6. Gain Measurement

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 start-up time; if C1 and C2 increase, start-up time increases to the point at which the oscillator will not start. Hence, for fast and reliable start-up, 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 lot, V_{cc}, Process 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, lot, 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 start-up 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.

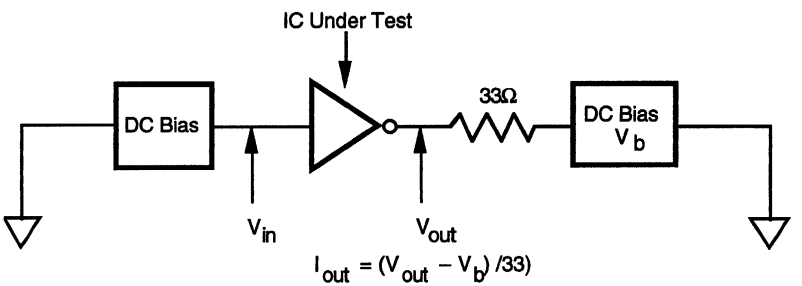
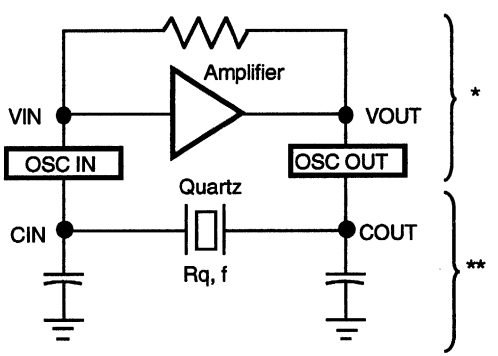


Figure 7. Transconductance (gm) Measurement



* 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 start-up.

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 100 pF.

Summary. For reliable and fast start-up, 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 start-up, 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 start-up 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 start-up 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).

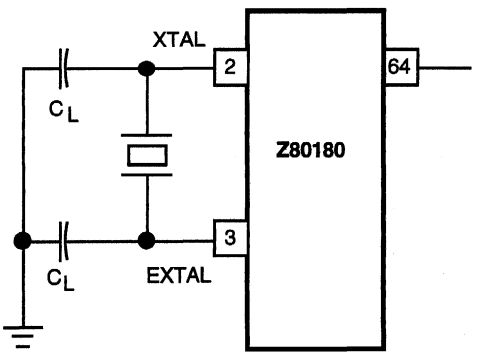
PRACTICE: CIRCUIT ELEMENT AND LAY OUT CONSIDERATIONS (Continued)

Measurement and Observation

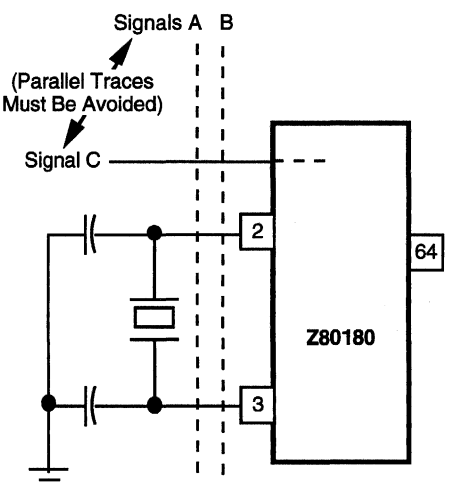
Connection of a scope to either of the circuit nodes is likely to affect operation because the scope adds 3-30 pF of capacitance and 1M-10M ohms of resistance to the circuit.

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:

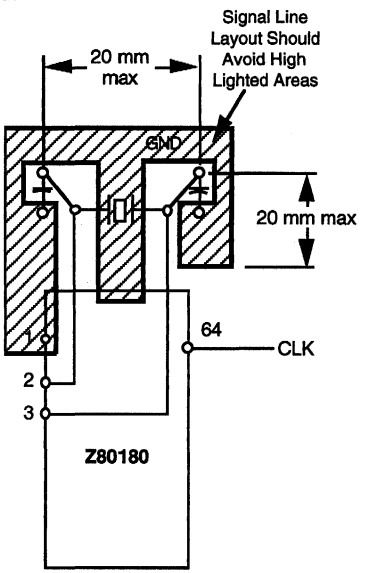


Clock Generator Circuit



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.



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: < 150 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 ACKNOWLEDGMENTS

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.

Z8S18000ZCO Evaluation Board
Product Specification

DEVICES SUPPORTED: Z8S180, Z85230

DESCRIPTION

The kit contains an assembled circuit board, software and documentation to support software and hardware development for the Z8S180 and Z85230 system at 18.432 MHz.

The supplied cross assembler and link/loader package allows full assembly language programming support. A board resident debug monitor program allows executable code to be down-loaded and subsequently debugged.

The board comes with sample code to illustrate the use of Zilog's Z8S180 and Z85230 in a variety of communication applications.

SPECIFICATIONS

Power Requirements

+5Vdc @ 5A

Dimensions

Width: 5.65 in.
Length: 4.0 in.

Serial Interface

RS-232 @ 9600 baud

KIT CONTENTS

Z8S180/ESCC Evaluation Board

CMOS Z8S180 MPU
CMOS Z85230 ESCC
18.432 MHz Crystal
Socketed 64K/(8K) x 8 EPROM
(Programmed with Debug Monitor and Device Driver Demonstration Software)
Socketed 32K/(8K) x 8 Static RAM
RS-232C PC Interface
Z8S180 Expansion Header
Z85230 Expansion Header
Reset Switch
NMI Switch

Cables

25-pin RS-232 Cable

Software (IBM® PC Platform)

ASM800 Z800 Cross Assembler
MOBJ Link/Loader
Resident Debug Monitor and Device Drivers Demonstration Software Source Code
Z8S180 Example Software
(a) In ASM800 Assembly
(b) In Microtec MCC80 C and Microtec ASM80 Assembly

Note: Zilog is not responsible for support and maintenance of the above software.

Documentation

Z8S180/ESCC Kit User's Guide
Z80180/Z8S180 Product Specification
Z80180/Z180 Technical Manual
Z85230 Product Specification
Z85230/Z80230 Technical Manual
ASM800 Z800 Cross Assembler User's Guide
MOBJ Link/Loader User's Guide

ORDERING INFORMATION

Part No: Z8S18000ZCO

*IBM is a registered trademark of International Business Machines Corp.

9

Z8523000ZCO Evaluation Board
Product Specification

DEVICES SUPPORTED: Z8530, Z85C30, Z85230

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 ESCC™ device.

The board illustrates the use of Zilog's ESCC 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

Z85230 Evaluation Board

CMOS Z85230 ESCC
RS-232C and RS-422 line drivers
DB25 connector

Software (IBM® PC Platform)

Source and executable codes to run the ESCC 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

Z85230 Product Specifications
Z85230 Technical Manuals
Z8523000ZCO User's Guide
Sealevel™ User's Manual

ORDERING INFORMATION

Part No: Z8523000ZCO

Z8018600ZCO Evaluation Board
Product Specification

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

DESCRIPTION

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

SPECIFICATIONS

Power Requirements

+5 Vdc @ .50A

Dimensions

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

Serial Interfaces

RS-232C, RS-422

KIT CONTENTS

Z8018600ZCO Evaluation Board

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

Cables

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

Software (IBM® PC Platform)

Resident Monitor for download and execution (80186 Assembler source code)
PC-board terminal emulator
Z85230, Z16C30, Z16C32, and Z16C35 Examples Software (All codes written in "C" and compiled using the Microtec® C compiler.)

Documentation

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

ORDERING INFORMATION

Part No: Z8018600ZCO

ZEPMDC00002 EPM™
PRODUCT SPECIFICATION

DEVICES SUPPORTED: Z8X30, Z8XC30, Z8X230, Z16C35

DESCRIPTION

The EPM™ Electronic Programmer's Manual provides on-line documentation on Zilog's Serial Communications Controller family of devices (Z08X30 NMOS SCC, Z8XC30 CMOS SCC, Z8X230 ESCC, Z16C35 ISCC™ controller): register set and operation of the device. Its code generation features make it a most valuable tool for the programmer. The EPM Manual helps you set the registers to ensure that the device operates with your specified settings. Once you have selected values for the registers, the EPM Manual lets you save the field values as a series of C function calls or as an assembler table. You can include this output in any software that utilizes the device.

SPECIFICATIONS

Minimum Hardware Requirements

IBM® PC/AT with available 512K RAM
5.25 inch, high density, or 3.5 inch, high density floppy disk drive
Hard disk drive
Color monitor

Minimum Operating System

MS-DOS, version 3.0 or later

KIT CONTENTS

Software (IBM® PC Platform)

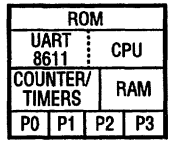
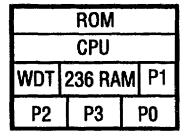
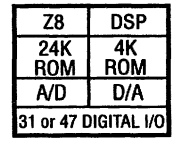
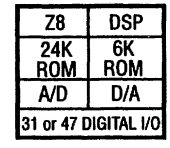
2 EPM Floppy Diskettes: 5.25 inch, high density and 3.5 inch, high density

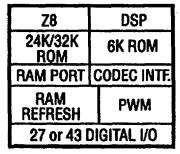
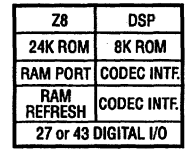
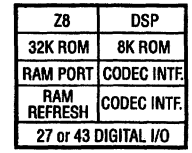
Documentation

EPM User's Guide
SCC User's Manual
ESCC User's Manual
ISCC User's Manual
EPM Registration Reply Card

ORDERING INFORMATION

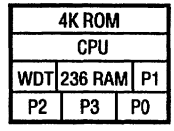
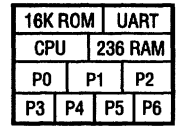
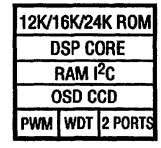
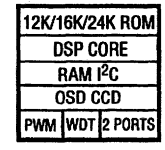
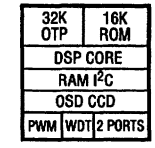
Part No: ZEPMDC00002

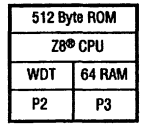
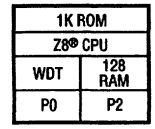
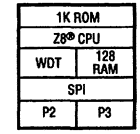
BLOCK DIAGRAM				
PART NUMBER	**Z8600/Z8611**	**Z86C30/E30/C31/E31**	**Z89C65/Z89C66**	**Z89165/Z89166**
DESCRIPTION	Z8® NMOS (CCP™) Z8600 = 2K ROM Z8611 = 4K ROM	Z8® Consumer Controller Processor (CCP™) Z86C30 = 28-Pin, 4K ROM Z86C31 = 28-Pin, 2K ROM Z86C40 = 40-Pin, 4K ROM Z86E30, Z86E31, Z86E40 = OTP Version	Telephone Answering Controller Z89C66 = ROMLess with 31 I/O Pins	Low-Cost DTAD Controller Z89166 = ROMLess with 31 I/O Pins
PROCESS/SPEED	NMOS: 8,12 MHz	CMOS: 12 MHz	CMOS: 20 MHz	CMOS: 20 MHz
FEATURES	■ 2K/4K ROM ■ 128 Bytes RAM ■ 22/32 I/O Lines ■ On-Chip Oscillator ■ Two Counter/Timers ■ Six Vectored, Priority Interrupts ■ UART (Z8611 Only)	■ 4K ROM/236 RAM ■ Two Standby Modes ■ Two Counter/Timers ■ ROM/RAM Protect ■ Four Ports (Z86C40/E40) ■ Three Ports (Z86C30/E30/C31/E31) ■ Low-Voltage Protection ■ Two Analog Comparators ■ Low-EMI Option ■ Watch-Dog Timer (WDT) ■ Auto Power-On Reset ■ Low-Power Option	■ 24K ROM (Z89C65 Only) ■ 16-Bit DSP ■ 4K Word ROM ■ 8-Bit A/D with Automatic Gain Control (AGC) ■ DTMF Macro Available ■ LPC Macro Available ■ 10-Bit PWM D/A ■ Other DSP Software Options Available ■ 47 I/O Pins (Z89C65 Only)	■ 24K ROM (Z89165 Only) ■ 16-Bit DSP ■ 6K Word DSP ROM ■ 8-Bit A/D with Automatic Gain Control (AGC) ■ DTMF Macro Available ■ LPC Macro Available ■ 10-Bit PWM D/A ■ Other DSP Software Options Available ■ 47 I/O Pins (Z89165 Only)
PACKAGE	28-Pin DIP 40-Pin DIP 44-Pin PLCC	28-Pin DIP 40-Pin DIP 44-Pin PLCC, QFP	68-Pin PLCC	68-Pin PLCC 80-Pin QFP
SUPPORT PRODUCTS	Z86C1200ZEM - Emulator Z0860000ZCO - Evaluation Board Z0860000ZDP - Adaptor Kit	Z86CCP00ZEM - Emulator Z86CCP00ZAC - Emulator Z86C5000ZEM - Emulator Z86E3000ZDP - Adaptor Kit Z86E4000ZDP - Program Adaptor Kit	Z89C6501ZEM - Emulator Z89C6500ZDB - Emulator	Z89C6501ZEM - Emulator Z89C6500ZDB - Emulator Z8916500ZCO - Evaluation Board

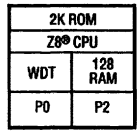
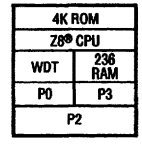
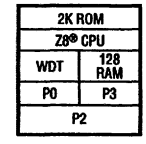
BLOCK DIAGRAM			
PART NUMBER	**Z89C67/Z89C68/Z89C69**	**Z89167/Z89168**	**Z89169**
DESCRIPTION	Telephone Answering Controller Z89C67 = 24 Kbytes of Program ROM Z89C68 = ROMLess with 27 I/O Pins Z89C69 = 32 Kbytes of Program ROM	Enhanced Telephone Answering Controller Z89168 = ROMLess with 27 I/O Pins	Enhanced Telephone Answering Controller
PROCESS/SPEED	CMOS: 20 MHz	CMOS: 24 MHz	CMOS: 24 MHz
FEATURES	■ 16-Bit DSP ■ 6K Word ROM ■ DTMF Macro Available ■ LPC Macro Available ■ 10-Bit PWM D/A ■ Other DSP Software Options Available ■ ARAM/DRAM/ROM Controller and Interface ■ Dual CODEC Interface ■ 43 I/O (Z89C67 Only)	■ 24K ROM (Z89167 Only) ■ 16-Bit DSP ■ 8K Word ROM ■ DTMF Macro Available ■ LPC Macro Available ■ 10-Bit PWM D/A ■ Other DSP Software Options Available ■ ARAM/DRAM/ROM ■ Dual CODEC Interface ■ 43 I/O (Z89167 Only)	■ 32K ROM ■ 16-Bit DSP ■ 8K Word ROM ■ DTMF Macro Available ■ LPC Macro Available ■ 10-Bit PWM D/A ■ Other DSP Software Options Available ■ ARAM/DRAM/ROM ■ Dual CODEC Interface ■ 43 I/O
PACKAGE	84-Pin PLCC	84-Pin PLCC 100-Pin QFP	84-Pin PLCC 100-Pin QFP
SUPPORT PRODUCTS	Z89C5900ZEM - Emulator Z89C6700ZEM - Emulator Z89C6700ZDB - Emulator Z8916902ZCO - Evaluation Board	Z89C5900ZEM - Emulator Z89C6700ZEM - Emulator Z89C6700ZDB - Emulator Z8916902ZCO - Evaluation Board	Z89C5900ZEM - Emulator Z89C6700ZEM -Emulator Z89C6700ZDB - Emulator Z8916902ZCO - Evaluation Board

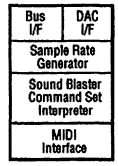
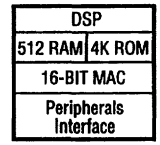
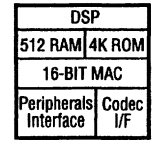
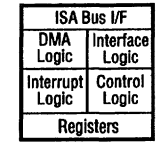
BLOCK DIAGRAM	16/8K ROM 4K CHAR ROM Z8 CPU | RAM OSD 13 PWM | TIMER WDT | 5 PORTS	6K ROM 3K CHAR ROM Z8 CPU | RAM OSD 7 PWM | TIMER WDT | 3 PORTS	CHAR ROM COMMAND INTERPRETER ANALOG SYNC/DATA SLICER | OSD CTRL	1K/6K ROM Z8 CPU WDT | 124 RAM P2 | P3	2K/8K/16K ROM Z8 CPU WDT | 128,256, 768 RAM P0 | P1 | P2 | P3
PART NUMBER	**Z86C27/127/97/47/E47**	**Z86227**	**Z86128/Z86228/Z86129**	**Z86L06/Z86L29**	**Z86L70/71/72/73/74 75/76/77/78**
DESCRIPTION	Digital Television Controller (DTC™) Television, VCRs, and Cable Z86E47 = OTP Version	Standard DTC™ Features with Reduced ROM, RAM, PWM Outputs for Greater Economy	Z86128/228 = Line 21 Closed Caption Controller (L21C™) Z86129/228 = Line 21 Closed Caption and EDS Controller	Z86L06 = Low-Voltage CMOS Consumer Controller Processor Z86L29 = 6K Infrared Remote Controller	Zilog Infrared Remote Controllers (ZIRC™) for IR Remote/Battery Operated Applications Ranging in ROM: L70=2K, L71=8K, L72&78=16K, L73&74=32K, L75=4K, L76=12K, L77=24K
PROCESS/SPEED	CMOS: 4 MHz	CMOS: 4 MHz	CMOS: 12 MHz	Low-Voltage CMOS: 8 MHz	Low-Voltage CMOS: 8 MHz
FEATURES	■ 8K/16K/OTP ROM ■ 256 Byte RAM ■ 160x7-Bit Video RAM ■ On-Screen Display (OSD) Video Controller ■ Programmable – Color – Size – Position Attributes ■ 13 PWMs for D/A Conversion ■ 128-Character Set ■ 4Kx6-Bit Char. Gen. ROM ■ Watch-Dog Timer (WDT) ■ Low-Voltage Protection ■ Five Ports/36 Pins ■ Two Standby Modes ■ Low-EMI Mode	■ 6K ROM, 256 Byte RAM ■ 120x7-Bit Video RAM ■ OSD On-Board Programmable – Color – Size – Position Attributes ■ 7 PWMs ■ 96 Character Set ■ 3Kx6-Bit Char. Gen. ROM ■ Watch-Dog Timer (WDT) ■ Low-Voltage Protection ■ Three Ports/20 Pins ■ Two Standby Modes ■ Low-EMI Mode	■ Conforms to FCC Line 21 Format ■ Parallel or Serial Modes ■ Stand-Alone Operation ■ On-Board Data Sync and Slicer ■ On-Board Character Generator – Color – Blinking – Italic – Underline – Extended Data Services	■ 1K ROM and 6K ROM ■ Watch-Dog Timer (WDT) ■ Two Analog Comparators with Output Option ■ Two Standby Modes ■ Two Counter/Timers ■ Auto Power-On Reset ■ 2V Operation ■ RC Oscillator Option ■ Low-Noise Option ■ Low-Voltage Protection ■ High-Current Drivers (2, 4)	■ Watch-Dog Timer (WDT) ■ Two Analog Comparators with Output Option ■ Two Standby Modes ■ Two Enhanced Counter/Timers – Auto Pulse – Reception/Generation ■ Auto Power-On Reset ■ 2V Operation ■ RC Oscillator Option ■ Low-Voltage Protection ■ High-Current Drivers – Three OTP Versions Available – Z86E72/73/74
PACKAGE	64-Pin DIP	40-Pin DIP	18-Pin DIP	18-Pin DIP 18-Pin SOIC	Z86L71=20-Pin DIP/SOIC Z86L70/L75=18-Pin DIP, SOIC Z86L72/L76/L77=40,44-Pin DIP, PLCC, QFP Z86L74=64/68-Pin
SUPPORT PRODUCTS	Z86C2700ZCO - Evaluation Board Z86C2700ZDB - Emulator Z86C2700ZEM - Emulator	Z86C2700ZDB - Emulator Z86C2702ZEM - Emulator Z86C2700ZCO - Evaluation Board	Support Documentation Provided with the device	Z86C5000ZEM - Emulator	Z86L7200TSC - Emulator Z86L7100ZEM - Emulator Z86L7100ZDB - Emulator

ZiLOG TV/Video Products — SuperIntegration™ Products Guide

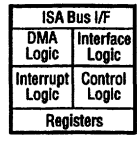
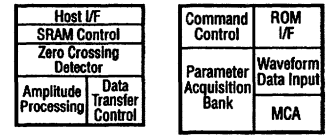
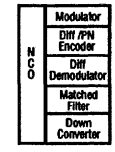
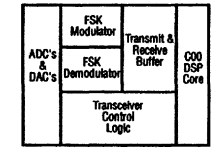
Block Diagram					
Part Number	**Z86C40/Z86E40**	**Z86C61/Z86C62**	**Z89300/02/04/06/14**	**Z89301/03/05/07/13**	**Z89331/Z89336**
Description	Z8® Consumer Controller Processor (CCP™) Z86E40 = OTP Version	Z8® MCU with Expanded I/Os	Advanced TV Controller with Closed Caption Decoder (CCD), StarSight*, OSD for TV, VCR, Cable, Satellite Z89301 = OTP Version	Advanced TV Controller with CCD StarSight, for TV, VCR, Cable, Satellite Z89301 = OTP Version	Advanced TV Controller with CCD StarSight, OSD for TV, VCR, Cable, Satellite Z89301 = OTP Version
Process/Speed	CMOS: 12 MHz	CMOS: 16, 20 MHz	CMOS: 12 MHz	CMOS: 12 MHz	CMOS: 12 MHz
Features	■ 4K ROM, 236 RAM ■ Two Standby Modes ■ Two Counter/Timers ■ ROM Protect ■ RAM Protect ■ Four Ports ■ Low-Voltage Protection ■ Two Analog Comparators ■ Low-EMI Mode ■ Watch-Dog Timer (WDT) ■ Auto Power-On Reset ■ Low-Power Option	■ 16K ROM ■ Full-Duplex UART ■ Two Standby Modes (STOP and HALT) ■ Two Counter/Timers ■ ROM Protect Option ■ RAM Protect Option ■ Pin Compatible to Z86C21 ■ Z86C61 = Four Ports ■ Z86C62 = Seven Ports	■ StarSight Capability ■ Closed-Captioning ■ DSP 12 MHz ■ 16-Bit, 512 Byte (Z89314) ■ 640 Byte RAM ■ 12K/16K/24K ROM ■ Programmable OSD ■ I²C*, 7 PWM ■ 3-Channel ADC ■ Watch-Dog Timer (WDT) ■ Two Ports ■ 32 kHz, XTAL ■ Low-Power Mode *Not Available on Z89314	■ StarSight Capability ■ Closed-Captioning ■ DSP 12 MHz ■ 16-Bit, 640 Byte RAM ■ 12K/16K/24K ROM ■ Programmable OSD ■ I²C, 9 PWM ■ 4-Channel ADC ■ Watch-Dog Timer (WDT) ■ Two Ports ■ 32 kHz, XTAL ■ Low-Power Mode	■ StarSight Capability ■ Closed-Captioning ■ DSP 12 MHz ■ 16-Bit, 640 Byte RAM ■ 12K/16K/24K ROM ■ Programmable OSD ■ I²C, 7 PWM ■ 5-Channel ADC ■ Watch-Dog Timer (WDT) ■ Two Ports ■ 32 kHz, XTAL ■ Low-Power Mode
Package	40-Pin DIP 44-Pin PLCC	Z86C61 = 40-Pin DIP Z86C61 = 44-Pin PLCC,QFP Z86C62 = 68-Pin PLCC	40-Pin SDIP	52-Pin SDIP	42-Pin SDIP
Support Products	Z86C5000ZEM - Emulator Z86CCP00ZEM - Emulator Z86E4000ZDP - Adaptor Kit Z86E4000ZDV - Adaptor Kit	Z86C5000ZEM - Emulator Z86CCP00ZEM - Emulator	Z8930900ZEM - Emulator Z8930900TSC - Emulator Z8930901TSC - Emulator	Z8930900ZEM - Emulator Z8930900TSC - Emulator Z8930901TSC - Emulator	Z8930900ZEM - Emulator Z8930900TSC - Emulator Z8930901TSC - Emulator

BLOCK DIAGRAM			
PART NUMBER	**Z86C03**	**Z86C04/Z86E04**	**Z86C06**
DESCRIPTION	Consumer Controller Processor (CCP™) with 512 Byte ROM	Z86C04 = 8-Bit Low Cost 1 Kbyte ROM MCU Z86E04 = OTP Version	Consumer Controller Processor (CCP™) with 1 Kbyte ROM
PROCESS/SPEED	CMOS: 8 MHz	CMOS: 8 MHz	CMOS: 12 MHz
FEATURES	■ 512 Byte ROM ■ 64 Byte RAM ■ Two Standby Modes ■ One Counter/Timer ■ ROM Protect ■ Two Analog Comparator ■ Auto Power-On Reset ■ Low-Voltage Protection ■ 14 I/O ■ RC Oscillator Option ■ Low-Noise Option	■ 1 Kbyte ROM ■ 128 Byte RAM ■ Two Standby Modes ■ Two Counter/Timer ■ ROM Protect ■ Two Analog Comparator ■ Auto Power-On Reset ■ Low-Voltage Protection (ROM Only) ■ 14 I/O ■ Low-Noise Option	■ 1 Kbyte ROM ■ 128-Byte RAM ■ Two Standby Modes ■ Two Counter/Timer ■ ROM Protect ■ Two Analog Comparator ■ Auto Power-On Reset ■ Low-Voltage Protection (ROM Only) ■ 14 I/O ■ RC Oscillator Option ■ Serial Peripheral Interface (SPI)
PACKAGE	18-Pin DIP 18-Pin SOIC	18-Pin DIP 18-Pin SOIC	18-Pin DIP 18-Pin SOIC
SUPPORT PRODUCTS	Z86CCP00ZEM - Emulator Z86CCP00ZAC - Emulator	Z86C0800ZCO - Evaluation Board Z86C0800ZDP - Adaptor Kit Z86C1200ZEM - Emulator Z86C1200ZPD - Adaptor Kit Z86CCP00ZEM - Emulator Z86CCP00ZAC - Emulator	Z86E0600ZDP - Adaptor Kit Z86C5000ZEM - Emulator Z86C5000ZDP - Adaptor Kit Z86CCP00ZEM - Emulator Z86CCP00ZAC - Emulator

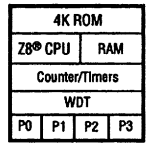
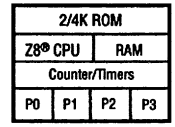
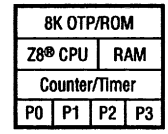
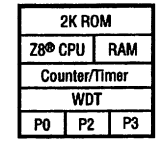
BLOCK DIAGRAM			
PART NUMBER	**Z86C08/Z86E08**	**Z86C30/Z86E30**	**Z86C31/Z86E31**
DESCRIPTION	Z86C08 = Z8® MCU with 2 Kbyte ROM Z86E08 = OTP Version	Z86C30 = Z8® (CCP™) with 4 Kbyte ROM Z86E30 = OTP Version	Z86C31 = 8-Bit MCU with 2 Kbyte ROM Z86E31 = OTP Version
PROCESS/SPEED	CMOS: 12 MHz	CMOS: 12 MHz	CMOS: 8 MHz
FEATURES	■ 2 Kbyte ROM ■ 128 Byte RAM ■ Two Standby Modes ■ Two Counter/Timer ■ ROM Protect ■ Two Analog Comparators ■ Auto Power-On Reset ■ Low-Voltage Protection (ROM Only) ■ 14 I/O ■ Low-Noise Option	■ 4 Kbyte ROM ■ 236 Byte RAM ■ Two Standby Modes ■ Two Counter/Timer ■ ROM Protect ■ Two Analog Comparators ■ Auto Power-On Reset ■ Low-Voltage Protection (ROM Only) ■ 24 I/O ■ RC Oscillator Option ■ Low-Noise Option	■ 2 Kbyte ROM ■ 128 Byte RAM ■ Two Standby Modes ■ Two Counter/Timer ■ ROM Protect ■ Two Analog Comparators ■ Auto Power-On Reset ■ Low-Voltage Protection (ROM Only) ■ 24 I/O ■ RC Oscillator Option ■ Low-Noise Option
PACKAGE	18-Pin DIP 18-Pin SOIC	28-Pin DIP	28-Pin DIP 28-Pin PLCC
SUPPORT PRODUCTS	Z86C0800ZCO - Evaluation Board Z86C0800ZDP - Adaptor Kit Z86C1200ZEM - Emulator Z86C1200ZDP - Adaptor Kit Z86CCP00ZEM - Emulator Z86CCP00ZAC - Emulator	Z86E3000ZDP - Adaptor Kit Z86C5000ZEM - Emulator Z86C5000ZPD - Emulator Pod Z86CCP00ZEM - Emulator Z86CCP00ZAC - Emulator	Z86E3000ZDP - Adaptor Kit Z86C5000ZEM - Emulator Z86C5000ZPD - Emulator Pod Z86CCP00ZEM - Emulator Z86CCP00ZAC - Emulator

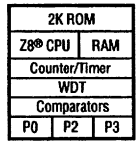
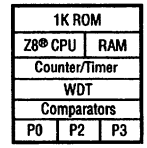
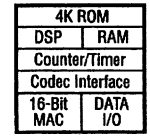
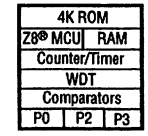
BLOCK DIAGRAM				
PART NUMBER	**Z86321**	**Z89320**	**Z89321/Z89371**	**Z5380**
DESCRIPTION	8-Bit Digital Audio Processor	16-Bit Digital Signal Processor	16-Bit Digital Signal Processor Z89371= OTP Version	Small Computer System Interface (SCSI)
PROCESS/SPEED	CMOS: 12 MHz	CMOS: 10 MHz	CMOS: 20 MHz	Clock: 1.5 Mb/s
FEATURES	■ Sound Blaster™ Compatible ■ ADPCM Decompression ■ 8-Bit DAC Interface ■ Successive Approximation ADC Algoritant ■ MIDI Interface	■ 16-Bit Multiply/Accumulate ■ 100 ns ■ 512 Word RAM ■ 4K Word RAM ■ Peripherals Interface Bus ■ 74 Instruction Set	■ 16-Bit Multiply/Accumulate ■ 50 µs ■ 512 Word RAM ■ 4K Word ROM ■ Peripherals Interface Bus ■ CODEC Interface	■ Compatible 5380 Pin-out ■ CMOS ■ Asynchronous I/F Supports 1.5 Mb/s ■ 48 mA Drivers ■ Arbitration Support ■ Support Normal or Block Mode DMA
PACKAGE	40-Pin DIP 44-Pin PLCC	40-Pin DIP 44-Pin PLCC	40-Pin DIP 44-Pin PLCC	40-Pin DIP 44-Pin PLCC
SUPPORT PRODUCTS	Support Documentation Provided with Device	Z89C0000ZEM - Emulator	Z8937100ZEM - Emulator	Support Documentation Provided with Device

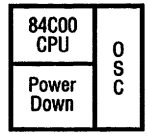
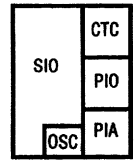
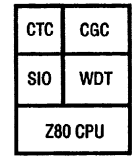
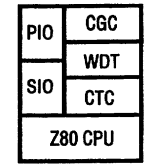
SoundBlaster™ is a Trademark of Creative Labs, Inc.

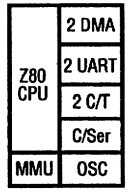
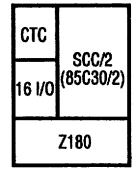
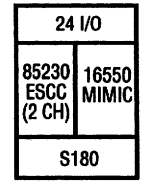
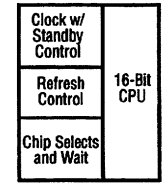
ZILOG MULTIMEDIA/PC AUDIO			WIRELESS DEVICES	
BLOCK DIAGRAM				
PART NUMBER	**Z53C80**	**Z89341/Z89342**	**Z2000***	**Z87000**
DESCRIPTION	SCSI Adaptor	Wave Synthesis Chip Set	Spread Spectrum Burst Processor	Cordless Phone Transceiver/Controller
SPEED MHz	Clock: 3 Mb/s	CMOS: 36 MHz	CMOS: 45 MHz Clock: 2.048 Mb/s	CMOS: 16.384 MHz
FEATURES	■ ANSI X3, 131-1986 Standard ■ DMA or Programmed I/O Data Transfers ■ Asynchronous Interface Support ■ 3 Mb/s ■ ISA Bus I/F ■ Glitch Eater	■ 4-Channel ■ 16-Bit Linear ■ PCM Sound Generator ■ Sampling Rates 20 kHz to 44.1 kHz ■ Support 16-, 18-, and 20-Bit DAC ■ Audio Bandwidth 0 Hz to 20,000 Hz ■ Direct Interface with PC ISA Bus ■ Direct Support 4Mx16 ROM	■ Operates up to 11.1264 Mchips Second in Transmit and Receive Modes ■ Maximum Data Rate of 2.048 Mbps in Conformance with FCC Regulations ■ Supports Differentially Encoded BPSK or QPSK Modulation ■ Full-or Half-Duplex Operation for FDD or TDD Implementations ■ Two Independent PN Sequences ■ Power Management Features	■ Supports 900 MHz Spread Spectrum Cordless Phone Design ■ Adaptive Frequency Hopping ■ Transmit Power Control ■ Bus Interface to ADPCM Processor ■ 12K Words of RAM for Transceiver and Phone Control Software ■ 32 Pins of Program I/O ■ ROM Code, OTP and ICEBOX™ Version to be Available Q3/94
PACKAGE	40-Pin DIP 44-Pin PLCC	84-Pin PLCC	100-Pin VQFP	84-Pin PLCC
SUPPORT PRODUCTS	Support Documentation Provided with Device	Support Documentation Provided with Device	Z02000000ZC0 - Evaluation Board	Z870000ZEM - Emulator

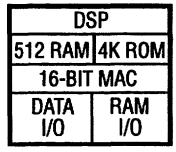
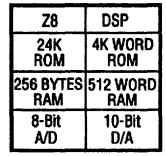
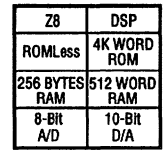
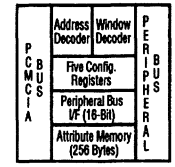
*Z2000 is sold under license from Stanford Telecommunications, Inc. ASIC and Custom Products Division

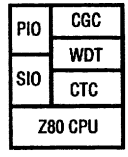
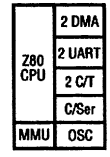
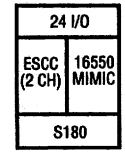
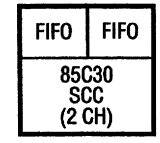
BLOCK DIAGRAM				
PART NUMBER	**Z8615**	**Z8614/Z8602**	**Z86E23**	**Z86C17**
DESCRIPTION	Keyboard MCU	Z8602 = 2K ROM Keyboard MCU Z8614 = 4K ROM Keyboard MCU	Keyboard OTP MCU	Mouse MCU
PROCESS/SPEED	NMOS: 4, 5 MHz	NMOS: 4 MHz	CMOS: 4 MHz	CMOS: 4 MHz
FEATURES	■ 4K ROM ■ 124-Byte RAM ■ 32 I/O Lines ■ Two Counter/Timers ■ Watch-Dog Timer (WDT) ■ RC Oscillator ■ Dedicated Row Column Pins ■ Data/Clock Pins ■ Direct Connect LED Pins	■ 4K ROM ■ 124 Byte RAM ■ 32 I/O Lines ■ Two Counter/Timers ■ Dedicated Row Column Pins	■ 8K ROM ■ 256 Byte RAM ■ 32 I/O Lines ■ Two Counter/Timers ■ Dedicated Row Column Pins	■ 2K ROM ■ 124 Byte RAM ■ 14 I/O Lines ■ Two Counter/Timers ■ Dedicated Opto-Transistor Pins ■ Integrated Pull-up Resistors ■ Power-Down Modes ■ Power-On Reset (POR) ■ Watch-Dog Timer (WDT)
PACKAGE	40-Pin DIP 44-Pin PLCC	40-Pin DIP 44-Pin PLCC	40-Pin DIP 44-Pin PLCC	18-Pin DIP 18-Pin SOIC
SUPPORT PRODUCTS	Z0861500ZCO - Evaluation Board Z86C1200ZEM -Emulator Z0861500ZDP - Adaptor Kit	Z0860200ZCO - Evaluation Board Z86C1200ZEM - Emulator Z0860200ZDP - Adaptor Kit Z86C1200ZPD - Emulator Pod	Z0860200ZCO - Evaluation Board Z86C1200ZEM - Emulator Z0860200ZDP - Adaptor Kit	Z86C1200ZEM - Emulator

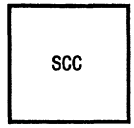
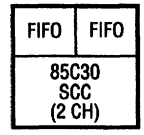
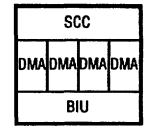
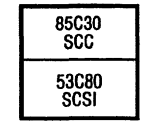
BLOCK DIAGRAM				
PART NUMBER	**Z86C08/Z86C07/Z86E08**	**Z86C04/Z86E04**	**Z89321/Z89371**	**Z86C30/Z86E30**
DESCRIPTION	Pointing Device Z8® MCU Z86E08 = OTP Version	Discrete MCU Z86E04 = OTP Version	16-Bit Digital Signal Processor Z89371 = OTP Version	Z8® MCU Z86E30 = OTP Version
PROCESS/SPEED	CMOS: 4,8,12 MHz	CMOS: 4 MHz	CMOS: 15, 20 MHz	CMOS: 8, 12 MHz
FEATURES	■ 2K ROM ■ 124 Byte RAM ■ 14 I/O Lines ■ Two Counter/Timers ■ Power-Down Modes ■ Two Comparators ■ Power-On Reset (POR) ■ Watch-Dog Timer (WDT) ■ Auto Latch (Z86C07 Only)	■ 1K ROM ■ 124 Byte RAM ■ 14 I/O Lines ■ Two Counter/Timers ■ Power-Down Modes ■ Two Comparators ■ Power-On Reset (POR) ■ Watch-Dog Timer (WDT)	■ 4K Word ROM ■ 512 Word RAM ■ 16 Bit I/O Bus ■ Two Counter/Timers ■ CODEC Interface ■ 50/75 ns Cycle Timer ■ 4K OTP ROM (Z89371 Only)	■ 4K Word ROM ■ 256 Byte RAM ■ 24 I/O Lines ■ 2 Counter/Timers ■ Power-Down Mode ■ Two Comparators ■ Power-On Reset (POR) ■ Watch-Dog Timer (WDT)
PACKAGE	18-Pin DIP 18-Pin SOIC	18-Pin DIP 18-Pin SOIC	40-Pin DIP 44-Pin PLCC	28-Pin DIP 28-Pin SOIC
SUPPORT PRODUCTS	Z86C1200ZEM - Emulator	Z86C1200ZEM - Emulator Z86CCP00ZEM - Emulator	Z8937100ZEM - Emulator Z8937100TSC - Emulator	Z86C5000ZEM - Emulator

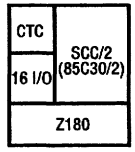
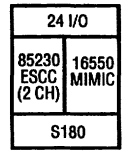
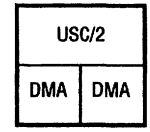
BLOCK DIAGRAM				
PART NUMBER	**Z84C01**	**Z84C90**	**Z84013/Z84C13**	**Z84015/Z84C15**
DESCRIPTION	Z80® CPU with Clock Generator/Clock	Killer I/O (Three Z80® Peripherals)	Intelligent Peripheral Controller	Enhanced Intelligent Peripheral
PROCESS/SPEED	CMOS: 10 MHz	CMOS: 8, 10, 12 MHz	Z84013 = CMOS: 6, 10 MHz Z84C13 = CMOS: 6, 10 MHz	Z84015 = CMOS: 6, 10 MHz Z84C15 = CMOS: 16 MHz
FEATURES	■ Clock Generator/Controller ■ Four Power Down Modes	■ Serial Input/Output (SIO) ■ Counter/Timer Circuit (CTC) ■ Plus Eight I/O Lines ■ Three 8-Bit Ports	■ Serial Input/Output (SIO) ■ Counter/Timer Circuit (CTC) ■ Watch-Dog Timer (WDT) ■ Clock Generator Circuit (CGC) ■ Wait State Generator (WSG) ■ Power-On Reset (POR) ■ Two Chip Selects ■ Evaluation Mode	■ Serial Input/Output (SIO) ■ Counter/Timer Circuit (CTC) ■ Watch-Dog Timer (WDT) ■ Clock Generator Circuit (CGC) ■ Four Power-Down Modes ■ Power-On Reset ■ Two Chip Selects ■ 32-Bit CRC ■ Wait State Generator (WSG) ■ Evaluation Mode
PACKAGE	44-Pin QFP 44-Pin PLCC	84-Pin PLCC 80-Pin QFP	84-Pin PLCC	100-Pin QFP 100-Pin VQFP
SUPPORT PRODUCTS	Z84C9000ZCO - Evaluation Board	Z84C9000ZCO - Evaluation Board	Z84C1500ZCO - Evaluation Board	Z84C1500ZCO - Evaluation Board

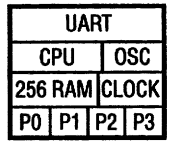
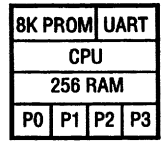
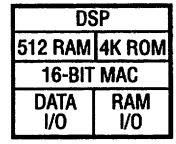
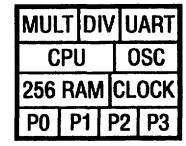
BLOCK DIAGRAM				
PART NUMBER	**Z80180/Z8S180/Z8L180**	**Z80181**	**Z80182/Z8L182**	**Z80380/Z8L380**
DESCRIPTION	High-Performance Z80® CPU with Peripherals Z8S180 = Static Version Z8L180 = Low-Voltage Version	Smart Access Controller	Zilog Intelligent Peripheral (ZIP™) Z8L182 = Low-Voltage Version	Z380™ Microprocessor Z8L380 = Low-Voltage Z380
PROCESS/SPEED	Z80180 = CMOS: 6, 8, 10, MHz Z8S180 = CMOS: 16 MHz Z8L180 = CMOS: 20, 33 MHz	CMOS: 10, 12 MHz	Z80182 = CMOS: 16, 33 MHz Z8L182 = CMOS: 20 MHz	Z8L380 = CMOS: 10 MHz Z80380 = CMOS: 16, 18 MHz
FEATURES	■ Enhanced Z80® CPU ■ 1 Mbyte MMU ■ 2 DMAs ■ 2 UARTs with Baud Rate Generators ■ C/Serial I/O Port Oscillator ■ Z8S180 Includes; – Power-Down – Programmable EMI – Divide-By-One – Clock Option – 3.3V and 5V Version	■ Complete Z180™ plus SCC/2 Counter/Timer Circuit ■ 16 I/O Lines ■ Emulation Mode	■ Static Version of Z180™ plus ESCC (2 Channels of Z85230 with 32-Bit CRC Not Available for 16 MHz) ■ 16550 MIMIC ■ 24 Parallel I/O ■ Emulation Mode ■ 3.3V and 5V Version	■ 16/32-Bit MPU ■ Internal 32-Bit Datapaths and ALU ■ 2 Clocks/Cycle Instruction Execution up to 4 Gbytes of Linear Addressing ■ Enhanced Instruction Set ■ 4 Banks of On-Chip Register Files ■ Object-Code Compatible with Z80/Z180 Microprocessors up to 6 Programmable Memory Chip Selects ■ 3.3V and 5V Version
PACKAGE	64-Pin DIP 68-Pin PLCC 80-Pin QFP	100-Pin QFP	100-Pin QFP 100-Pin VQFP	100-Pin QFP
SUPPORT PRODUCTS	Z8S18000ZCO - Evaluation Board ZEPMIP00001 - EPM™ Manual	Z8018100ZCO - Evaluation Board Z8018100ZDP - Adaptor Kit Z8018101ZCO* - Evaluation Board * Includes LLAP software that can be licensed (Z80181ZA6). ZEPMIP00001- EPM™ Manual	Z8018200ZCO - Evaluation Board ZEPMIP00002 - EPM™ Manual	Z8038000ZCO - Evaluation Board ZEPMIP00003 - EPM™ Manual

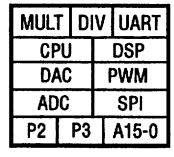
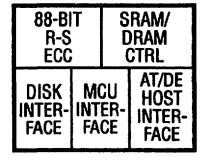
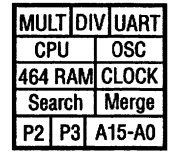
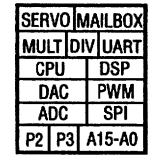
BLOCK DIAGRAM				
PART NUMBER	**Z89C00**	**Z89120**	**Z89920**	**Z86017**
DESCRIPTION	16-Bit Digital Signal Processor	Zilog Modem/Fax Controller	Zilog Modem/Fax Controller	PCMCIA Interface Adaptor
PROCESS/SPEED	CMOS: 10, 15 MHz	CMOS: 20 MHz	CMOS: 20 MHz	CMOS: 20 MHz
FEATURES	■ 16-Bit Multiply/Accumulate ■ 75 ns ■ Two Data RAMs (256 Words each) ■ 4K Word ROM ■ 64Kx16 Ext. ROM ■ 16-Bit I/O Port ■ 74 Instructions ■ Most Single Cycle ■ Two Conditional Branch Inputs, Two User Outputs ■ Library of Macros ■ Zero Overhead Pointers	■ Z8® with 24 Kbyte ROM ■ 16-Bit DSP with 4K Word ROM ■ 8-Bit A/D ■ 10-Bit D/A (PWM) ■ Library of Macros ■ 47 I/O Pins ■ Two Comparators Independent Z8® and DSP Operations Power-Down Mode	■ Z8 with 64K External Memory ■ DSP with 4K Word ROM ■ 8-Bit A/D ■ 10-Bit D/A ■ Library of Macros ■ 47 I/O Pins ■ Two Comparators Independent Z8® and DSP Operations Power-Down Mode	■ 256 Bytes of Attribute Memory ■ Five Configuration Registers ■ EEPROM Sequencer or SPI Interface ■ PCMCIA to I/O, Memory or Both ■ PCMCIA to ATA/IDE ■ ATA/IDE to ATA/IDE ■ 3.0V to 5.5V Operation ■ 8- or 16-Bit Peripheral Support
PACKAGE	68-Pin PLCC 60-Pin VQFP	68-Pin PLCC	68-Pin PLCC	100-Pin VQFP
SUPPORT PRODUCTS	Z89C0000ZEM - Emulator Z89C0000ZCC - Emulator	Z89C6501ZEM - Emulator Z89C6500ZDP - Emulator	Z89C6501ZEM - Emulator Z89C6500ZDB - Emulator	Z8601700ZCO - Evaluation Board

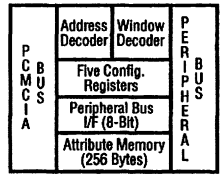
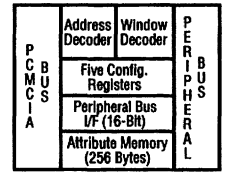
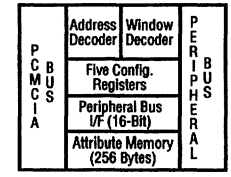
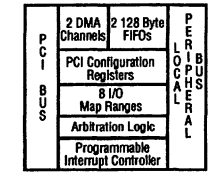
BLOCK DIAGRAM				
PART NUMBER	**Z84C15/Z84015**	**Z80180/Z8S180/Z8L180**	**Z80182/Z8L182**	**Z85230**
DESCRIPTION	Enhanced Intelligent Peripheral Controller	High-Performance Z80® CPU with Peripherals Z8S180 = Static Version Z8L180 = Low-Voltage Version	Zilog Intelligent Peripheral (ZIP™) Z8L182 = Low-Voltage Version	Enhanced Serial Communication Controller
PROCESS/SPEED	Z84015 = CMOS: 6, 10 MHz Z84C15 = CMOS: 16 MHz	Z80180 = CMOS: 6, 8, 10, MHz Z8S180 = CMOS: 16 MHz Z8L180 = CMOS: 20, 33 MHz	Z80182 = CMOS: 16, 18, 33 MHz Z8L182 = CMOS: 20 MHz	CMOS: 8, 10,16, 20 MHz
FEATURES	■ Z80® CPU, Serial Input/Output (SIO) ■ Counter/Timer Circuit (CTC) ■ Watch-Dog Timer (WDT) ■ Clock Generator Circuit (CGC) ■ Four Power-Down Modes Z84C15 Enhancements Include: ■ Power-On Reset ■ Two Chip Selects ■ 32-Bit CRC ■ Wait State Generator (WSG) ■ Evaluation Mode	■ Enhanced Z80® CPU ■ 1 Mbyte MMU ■ 2 DMAs ■ 2 UARTs with Baud Rate Generators ■ C/Serial I/O Port Oscillator ■ Z8S180 Includes; – Power Down – Programmable EMI – Divide-By-One – Clock Option – 3.3V and 5V Version	■ Static Version of Z180™ plus ESCC (Two Channels of Z85230 with 32-Bit CRC Not Available for 16 MHz) ■ 16550 MIMIC ■ 24 Parallel I/O ■ Emulation Mode ■ 3.3V and 5V Version	■ Full Dual-Channel ■ SCC Plus Deeper FIFOs: – 4 Bytes on Transceivers – 8 Bytes on Receivers ■ DPLL Counter Per Channel ■ Software Compatible to SCC
PACKAGE	100-Pin QFP 100-Pin VQFP	64-Pin DIP 68-Pin PLCC 80-Pin QFP	100-Pin QFP 100-Pin VQFP	40-Pin DIP 44-Pin PLCC
SUPPORT PRODUCTS	Z84C1500ZCO - Evaluation Board	Z8S18000ZCO - Evaluation Board ZEPMIP00001- EPM™ Manual	Z8018200ZCO - Evaluation Board ZEPMIP00002 - EPM™ Manual	Z8S18000ZCO - Evaluation Board Z8038000ZCO - Evaluation Board Z8523000ZCO - Evaluation Board Z8018600ZCO - Evaluation Board ZEPMDC00002 - EPM™ Manual

BLOCK DIAGRAM				
PART NUMBER	**Z8030/Z80C30** **Z8530/Z85C30**	**Z85230/Z80230** **Z85233**	**Z16C35**	**Z85C80**
DESCRIPTION	Serial Communication Controller Z8030/Z80C30 = Multiplexed Bus Z8530/Z85C30 = Non-Multiplexed Bus	Enhanced Serial Communication Controller Z8230/Z80230 = Dual Channel Z85233 = Single Channel	Integrated Serial Communication Controller	SCSCI Serial Communication and Small Computer Interface
PROCESS/SPEED	Z8030/Z8530 = NMOS: 4, 6, 8 MHz Z80C30/Z85C30 = CMOS: 8,10 16 MHz Clock: 2, 2.5, 4 Mb/s	CMOS: 10, 16 20 MHz Clock: 2.5, 4.0, 5.0 Mb/s	CMOS: 10, 16 MHz Clock: 2.5, 4.0 Mb/s	CMOS: 10, 16 MHz Clock: 2.5 Mb/s
FEATURES	■ Two Independent Full-Duplex Channels ■ Enhanced DMA Support: ■ 10x19 Status FIFO ■ 14-Bit Byte Counter ■ NRZ/NRZI/FM Encoding Modes	■ Full Dual-Channel SCC Plus Deeper FIFOs: – 4 Bytes on Transmitters – 8 Bytes on Receivers ■ DPLL Counter Per Channel ■ Software Compatible to SCC	■ Full Dual-Channel SCC ■ Four DMA Controllers ■ Bus Interface Unit	■ Two Independent Full-Duplex Channels ■ Direct SCSI Bus Interface ■ Supports SCSI ANSI-X3.131-1986 Standard
PACKAGE	40-Pin DIP 44-Pin CERDIP 44-Pin PLCC	40-Pin DIP 44-Pin PLCC 44-Pin QFP (Z85233 Only)	68-Pin PLCC	68-Pin PLCC 100-Pin VQFP
SUPPORT PRODUCTS	Z8018600ZCO - Evaluation Board Z8523000ZCO - Evaluation Board Z8018100ZCO - Evaluation Board ZEPMD000002 - EPM™ Manual	Z8018600ZCO - Evaluation Board Z8S18000ZCO - Evaluation Board Z8038000ZCO - Evaluation Board Z8523000ZCO - Evaluation Board ZEPMDC00002 - EPM™ Manual	Z8018600ZCO - Evaluation Board	ZEPMD00002 - EPM™ Manual

BLOCK DIAGRAM				
PART NUMBER	**Z80181**	**Z80182/Z8L182**	**Z16C30**	**Z16C32**
DESCRIPTION	Smart Access Controller	Zilog Intelligent Peripheral (ZIP™) Z80L182 = Low-Voltage Version	Universal Serial Controller (USC®)	Integrated Universal Serial Controller
PROCESS/SPEED	CMOS: 10, 12 MHz	Z80182 = CMOS: 16,18, 33 MHz Z8L182 = CMOS: 20 MHz	CMOS: 10 MHz CPU Bus 10 Mb/s	CMOS: 20 MHz DMA Clock 20 Mb/s
FEATURES	■ Complete Z180™ plus SCC/2CTC ■ 16 I/O Lines ■ Emulation Mode	■ Complete Static Version of Z180™ plus ESCC (2 Channels of Z85230 with 32-Bit CRC not Available for 16 MHz) ■ 16550 MIMIC ■ 24 Parallel I/O ■ Emulation Mode ■ 3.3V and 5V Version	■ Two Dual-Channel 32-Byte Receive and Transmit FIFOs ■ 16-Bit Bus B/W:18.2 Mb/s ■ Two BRGs Per Channel ■ Flexible 8/16-Bit Bus Interface ■ 12 Serial Protocols ■ Eight Data Encoding Bits	■ Single-Channel (Half of USC) plus two DMA Controllers ■ Array Chained and Linked-List Modes with Ring Buffer Support
PACKAGE	100-Pin QFP	100-Pin QFP 100-Pin VQFP	68-Pin PLCC	68-Pin PLCC
SUPPORT PRODUCTS	Z8018100ZCO - Evaluation Board Z8018100ZDP - Adaptor Kit Z8018101ZCO* - Evaluation Board ZEPMIP00001 - EPM™ Manual * Includes LLAP software that can be licensed (Z80181ZA6)	Z8018200ZCO - Evaluation Board ZEPMIP00002 - EPM™ Manual	Z16C3001ZCO - Evaluation Board Z8018600ZCO - Evaluation Board ZEPMDC00001 - EPM™ Manual	Z16C3200ZCO - Evaluation Board Z8018600ZCO - Evaluation Board ZEPMDC00001 - USC® EPM™ Manual

BLOCK DIAGRAM				
PART NUMBER	**Z86C91/Z8691**	**Z86E21/Z86C21**	**Z89C00**	**Z86C93**
DESCRIPTION	ROMLess Z8®	Z86E21 = 8K OTP Z86C21 = 8K ROM	16-Bit Digital Signal Processor	ROMLess Enhanced Z8® Mult/Div
PROCESS/SPEED	Z86C91 = CMOS: 16 MHz Z8691 = NMOS: 12 MHz	CMOS: 12, 16 MHz	CMOS: 10, 15 MHz	CMOS: 20, 25, 33 MHz
FEATURES	■ Full-Duplex UART ■ Two Standby Modes (STOP and HALT) ■ 2x8 Bit ■ Counter/Timer	■ 256 Byte RAM ■ Full-Duplex UART ■ Two Standby Modes (STOP and HALT) ■ Two Counter/Timers ■ ROM Protect Option ■ RAM Protect Option ■ Low-EMI Option	■ 16-Bit Multiply/Accumulate ■ 75 ns ■ Two Data RAMs (256 Words Each) ■ 4K Word ROM ■ 64Kx16 Ext. ROM ■ 16-Bit I/O Port ■ 74 Instructions ■ Most Single Cycle ■ Two Conditional Branch Inputs, Two User Outputs ■ Library of Macros ■ Zero Overhead Pointers	■ 16x16 Multiply 17 Clocks ■ 32x16 Divide 20 Clocks ■ Full-Duplex UART ■ Two Standby Modes (STOP and HALT) ■ Three 16-Bit Counter/Timers
PACKAGE	40-Pin DIP 44-Pin PLCC 44-Pin QFP	40-Pin DIP 44-Pin PLCC 44-Pin QFP	68-Pin PLCC	40-Pin DIP 44-Pin PLCC 44-Pin QFP
SUPPORT PRODUCTS	Z0860000ZCO - Evaluation Board Z86C0000ZUSP064 - Signum Emulator Z86C1200ZPD - Signum Emulator Pod	Z0860000ZCO - Evaluation Board Z86C0000ZUSP064 - Signum Emulator Z86C1200ZPD - Signum Emulator Pod	Z89C00ZEM - Emulator	Z0860000ZCO - Evaluation Board Z86C0000ZUSP064 - Signum Emulator Z86C0001ZUSP064 - Signum Emulator Z86C9300ZPD - Signum Emulator Pod Z86C9301ZPD - Signum Emulator Pod

BLOCK DIAGRAM				
PART NUMBER	**Z86C95**	**Z86018**	**Z86193**	**Z86295**
DESCRIPTION	ROMLess Enhanced Z8® with DSP	Zilog Datapath Controller	ROMLess Enhanced Z8® Multiply/Divide	ROMLess Enhanced Z8® DSP Servo Timer
PROCESS/SPEED	CMOS: 24, 33 MHz	CMOS: 40 MHz	CMOS: 40 MHz	CMOS: 40 MHz
FEATURES	■ Eight Channel ■ 8-Bit ADC ■ 8-Bit DAC ■ 16-Bit Multiply/Divide ■ Full-Duplex UART ■ Serial Peripheral Interface (SPI) ■ Three Standby Modes (STOP/HALT/PAUSE) ■ Pulse Width Modulator (PWM) ■ 3x16-Bit Timer ■ 16-Bit DSP Slave Processor ■ 83 ns Multiply/Accumulate	■ Full-Track Read ■ Automatic Data Transfer (Point & Go®) ■ 88-Bit Reed Solomon ECC "On The Fly" ■ Full AT/IDE Bus Interface ■ 64 Kbytes SRAM Buffer ■ 1 Mbytes DRAM Buffer ■ Split Data Field Support ■ Joint Test Action Group (JTAG) Boundary Scan Option ■ 8 Kbytes Buffer RAM Reserved for MCU	■ 16x16 Multiply 17 Clocks ■ 32x16 Divide 38 Clocks ■ Full-Duplex UART ■ Two Standby Modes (STOP & HALT) ■ Three 16-Bit Counter/Timers ■ SEARCH Machine ■ MERGE Machine ■ Bus Request Mode ■ Evaluation Mode	■ Eight Channel ■ 8-Bit ADC ■ 8-Bit DAC ■ Serial Peripheral Interface (SPI) ■ Pulse Width Modulator (PWM) ■ Three 16-Bit Counter/Timer ■ Full-Duplex UART ■ 16-Bit Z8® Multiply/Divide ■ Full 16-Bit DSP ■ Programmable Servo Timer ■ Z8® - DSP Mail Box
PACKAGE	80-Pin QFP 84-Pin PLCC 100-Pin VQFP	100-Pin VQFP	64-Pin VQFP	100-Pin VQFP 144-Pin QFP
SUPPORT PRODUCTS	Z86C9500ZCO - Evaluation Board Z86C9500ZUSP064 - Signum Emulator Z86C9501ZUSP064 - Signum Emulator Z86C9500ZPD - Signum Emulator POD Z86C9501ZPD - Signum Emulator POD Z86ZIA00ZCO - Evaluation Board	Z86C9900ZCO - Evaluation Board	Z8619200ZME - Emulator Z8619300ZCO - Evaluation Board	Z86ZIA01ZCO - Evaluation Board

BLOCK DIAGRAM				
PART NUMBER	**Z86016**	**Z86017**	**Z86M17**	**Z86020**
DESCRIPTION	8-Bit PCMCIA Interface Adaptor	PCMCIA Interface Adaptor	PCMCIA Interface Adaptor	PCI/Multifunction Bridge
PROCESS/SPEED	CMOS: 20 MHz	CMOS: 20 MHz	CMOS: 20 MHz	CMOS: 33 MHz
FEATURES	■ Z86017 with 8-Bit Peripheral Bus Only	■ 256 Bytes of Attribute Memory ■ Five Configuration Registers ■ EEPROM Sequencer or SPI Interface ■ PCMCIA to I/O, Memory or Both ■ PCMCIA to ATA/IDE ■ ATA/IDE to ATA/IDE ■ 3.0V to 5.5V Operation ■ 8- or 16-Bit Peripheral Support	■ Mirror Image Pin-Out of Z86017 for Opposite PCB - Surface Layout	■ 256 Bytes of Configuration Memory ■ 64 PCI Configuration Registers ■ Eight Programmable Memory or I/O Map Ranges with Independent Timing Control ■ 128 Byte FIFO's ■ Two Full Featured DMA Channels ■ PCI Initiator/Target Operations ■ On-Chip Peripheral Bus Arbitration
PACKAGE	48-Pin VQFP 64-Pin VQFP	100-Pin VQFP	100-Pin VQFP	160-Pin QFP
SUPPORT PRODUCTS	Z8601600ZCO - Evaluation Board (Available Q494)	Z8601700ZCO -Evaluation Board	Z8601700ZCO - Evaluation Board	Available Q494

ZILOG DOMESTIC SALES OFFICES AND TECHNICAL CENTERS

CALIFORNIA
Agoura 818-707-2160
Campbell 408-370-8120
Irvine 714-453-9701
San Diego 619-658-0391

COLORADO
Boulder 303-494-2905

FLORIDA
Clearwater 813-725-8400

GEORGIA
Duluth 404-931-4022

ILLINOIS
Schaumburg 708-517-8080

MINNESOTA
Minneapolis 612-944-0737

NEW HAMPSHIRE
Nashua 603-888-8590

OHIO
Independence 216-447-1480

OREGON
Portland 503-274-6250

PENNSYLVANIA
Horsham 215-784-0805

TEXAS
Austin 512-343-8976
Dallas 214-987-9987

INTERNATIONAL SALES OFFICES

CANADA
Toronto 905-850-2377

CHINA
Shenzhen 86-755-2220869
86-755-2220873
Shanghai 86-21-4370050, x5204
86-21-4331020
86-21-4150691

GERMANY
Munich 49-8967-2045
Sömmerda 49-3634-23906

JAPAN
Tokyo 81-3-5272-0230

HONG KONG
Kowloon 85-2-723-8979

KOREA
Seoul 82-2-577-3272

SINGAPORE
Singapore 65-2357155

TAIWAN
Taipei 886-2-741-3125

UNITED KINGDOM
Maidenhead 44-628-392-00

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

SALES REPRESENTATIVES AND DISTRIBUTORS

U.S., CANADIAN & PUERTO RICAN REPRESENTATIVES

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

CALIFORNIA
Santa Clara
Phase II Technical Sales ..... (408) 980-0414
Irvine
Infinity Sales ..... (714) 833-0300

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

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

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

GEORGIA
Norcross
BITS ..... (404) 564-5599

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

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

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

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

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

MICHIGAN
Novi
Rathsburg Associates, Inc. ..... (810) 615-4000

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

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

NORTH CAROLINA
Huntsville
BITS ..... (205) 881-2900
Raleigh
BITS ..... (919) 676-1880

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

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

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

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

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

SALES REPRESENTATIVES AND DISTRIBUTORS

OREGON

Portland
Phase II Technical Sales ..... (503) 643-6455

TEXAS

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

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

Houston
Nova Marketing, Inc. ..... (713) 240-6082

UTAH

Salt Lake City
Thorson Rocky Mountain ..... (801) 942-1683

WASHINGTON

Kirkland
Phase II Technical Sales ..... (206) 823-3874

WISCONSIN

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

CANADA

British Columbia
J-Squared Technologies, Inc. ..... (604) 473-4666

Ontario
J-Squared Technologies, Inc. ..... (905) 672-2030

Ottawa
J-Squared Technologies, Inc. ..... (613) 592-9540

Quebec
J-Squared Technologies, Inc. ..... (514) 694-8330

PUERTO RICO

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

SALES REPRESENTATIVES AND DISTRIBUTORS

U.S. AND CANADIAN DISTRIBUTORS

NATIONWIDE
Newark Electronics 1-800-367-3573
Zeus Electronics 1-800-524-4735

ALABAMA
Birmingham
Newark Electronics ..... (205) 979-7003
Huntsville
Anthem Electronics ..... (205) 890-0302
Arrow Electronics ..... (205) 837-6955
Newark Electronics ..... (205) 837-9091
Mobile
Newark Electronics ..... (205) 471-6500

ARKANSAS
Little Rock
Newark Electronics ..... (501) 225-8130

ARIZONA
Phoenix
Anthem Electronics ..... (602) 966-6600
Arrow Electronics ..... (602) 431-0030
Newark Electronics ..... (602) 864-9905
Tempe
Anthem Electronics ..... (602) 966-6600
Arrow Electronics ..... (602) 431-0030
Newark Electronics ..... (602) 966-6340

CALIFORNIA
Arcadia
Newark Electronics ..... (818) 445-1420
Calabasas
Arrow Electronics ..... (818) 880-9686
Chatsworth
Anthem Electronics ..... (818) 775-1333
Chula Vista
Newark Electronics ..... (619) 691-0141
Fremont
Arrow Electronics ..... (510) 490-9477
Garden Grove
Newark Electronics ..... (714) 893-4909
Hayward
Arrow Electronics ..... (510) 487-8416
Irvine
Anthem Electronics ..... (714) 768-4444
Arrow Electronics ..... (714) 587-0404
Zeus Electronics ..... (714) 581-4622
Palo Alto
Newark Electronics ..... (415) 812-6300
Riverside
Newark Electronics ..... (909) 784-1101
Sacramento
Anthem Electronics ..... (916) 624-9744
Newark Electronics ..... (916) 565-1760
San Diego
Anthem Electronics ..... (619) 453-9005
Arrow Electronics ..... (619) 565-4800
Newark Electronics ..... (619) 453-8211
San Jose
Anthem Electronics ..... (408) 453-1200
Arrow Electronics ..... (408) 441-9700
Zeus Electronics ..... (408) 629-4789
Santa Clara
Newark Electronics ..... (408) 988-7300
Santa Fe Springs
Newark Electronics ..... (310) 929-9722
Ventura
Newark Electronics ..... (805) 644-2265
West Hills
Newark Electronics ..... (818) 888-3718

COLORADO
Denver
Newark Electronics ..... (303) 373-4540
Englewood
Anthem Electronics ..... (303) 790-4500
Arrow Electronics ..... (303) 799-0258

CONNECTICUT
Bloomfield
Newark Electronics ..... (203) 243-1731
Norwalk
Zeus Electronics ..... (203) 852-5411
Wallingford
Arrow Electronics ..... (203) 265-7741
Waterbury
Anthem Electronics ..... (203) 575-1575

SALES REPRESENTATIVES AND DISTRIBUTORS

FLORIDA

Altamonte Springs
Anthem Electronics ..... (407) 831-0007

Clearwater
Anthem Electronics ..... (813) 538-4157
(800) 359-3522

Fort Lauderdale
Anthem Electronics ..... (305) 484-0990

Deerfield Beach
Arrow Electronics ..... (305) 429-8200

Jacksonville
Newark Electronics ..... (904) 399-5041

Orlando
Newark Electronics ..... (407) 896-8350

Plantation
Newark Electronics ..... (305) 424-4400

Tampa
Newark Electronics ..... (813) 287-1578

Lake Mary
Arrow Electronics ..... (407) 333-9300
Zeus Electronics ..... (407) 333-3055

GEORGIA

Duluth
Anthem Electronics ..... (404) 931-3900
(800) 293-0023
Arrow Electronics ..... (404) 497-1300

Norcross
Newark Electronics ..... (404) 448-1300

IDAHO

Boise
Newark Electronics ..... (208) 342-4311

ILLINOIS

Addison
Newark Electronics ..... (708) 495-7740

Arlington Heights
Newark Electronics ..... (708) 956-9270

Itasca
Arrow Electronics ..... (708) 250-0500
Zeus Electronics ..... (708) 595-9730

Rockford
Newark Electronics ..... (815) 229-0225

Schaumberg
Anthem Electronics ..... (708) 884-0200
Newark Electronics ..... (708) 310-8980

Springfield
Newark Electronics ..... (217) 787-9972

Willowbrook
Newark Electronics ..... (708) 789-4780
(708) 654-8250

INDIANA

Ft. Wayne
Newark Electronics ..... (219) 484-0766

Indianopolis
Arrow Electronics ..... (317) 299-2071
Newark Electronics ..... (317) 259-0085
(317) 884-0047

IOWA

Bettendorf
Newark Electronics ..... (319) 359-3711

Cedar Rapids
Arrow Electronics ..... (319) 395-7230
Newark Electronics ..... (319) 393-3800

West Des Moines
Newark Electronics ..... (515) 222-0700

KANSAS

Lenexa
Anthem Electronics ..... (913) 599-1528
Arrow Electronics ..... (913) 541-9542

Overland Park
Newark Electronics ..... (913) 677-0727

KENTUCKY

Louisville
Newark Electronics ..... (502) 423-0280

LOUISIANA

Metarie
Newark Electronics ..... (504) 838-9771

MARYLAND

Columbia
Anthem Electronics ..... (410) 995-6640
Arrow Electronics ..... (410) 596-7800

Hanover
Newark Electronics ..... (410) 712-6922

MASSACHUSETTS

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

Marlborough
Newark Electronics ..... (508) 229-2200

Methuen
Newark Electronics ..... (508) 683-0913

Wilmington
Anthem Electronics ..... (508) 657-5170
Arrow Electronics ..... (508) 658-0900
Zeus Electronics ..... (508) 658-4776

Woburn
Newark Electronics ..... (617) 935-8350

SALES REPRESENTATIVES AND DISTRIBUTORS

U.S. AND CANADIAN DISTRIBUTORS

MICHIGAN

Grand Rapids
Newark Electronics ..... (616) 954-6700

Livonia
Anthem Electronics ..... (313) 347-4090
(800) 359-3526
Arrow Electronics ..... (313) 462-2290

Oak Park
Newark Electronics ..... (810) 967-0600
(810) 968-2950

Plymouth
Arrow Electronics ..... (313) 462-2290

Saginaw
Newark Electronics ..... (517) 799-0480

MINNESOTA

Eden Prairie
Anthem Electronics ..... (612) 946-4826
Arrow Electronics ..... (612) 941-5280

Minneapolis
Newark Electronics ..... (612) 331-6350

St. Paul
Newark Electronics ..... (612) 631-2683

MISSISSIPPI

Ridgeland
Newark Electronics ..... (601) 956-3834

MISSOURI

Maryland Heights
Newark Electronics ..... (314) 298-2505

St. Louis
Arrow Electronics ..... (314) 567-6888

MONTANA

Helena
Newark Electronics ..... (406) 443-6192

NEBRASKA

Omaha
Newark Electronics ..... (402) 592-2423

NEVADA

Las Vegas
Newark Electronics ..... (702) 597-0330

Reno
Newark Electronics ..... (702) 322-6090

Sparks
Arrow Electronics ..... (702) 331-5000

NEW HAMPSHIRE

Nashua
Newark Electronics ..... (603) 888-5790

NEW JERSEY

East Brunswick
Newark Electronics ..... (908) 937-6600

Marlton
Arrow Electronics ..... (609) 596-8000

Pinebrook
Anthem Electronics ..... (201) 227-7960
Arrow Electronics ..... (201) 227-7880

Union
Newark Electronics ..... (908) 851-2290

SALES REPRESENTATIVES AND DISTRIBUTORS

NEW MEXICO

Albuquerque
Newark Electronics ..... (505) 828-1878

NEW YORK

Bohemia
Newark Electronics ..... (516) 567-4200

Brookhaven
Arrow Electronics ..... (516) 924-9400

Cheektowaga
Newark Electronics ..... (716) 862-9700

Commack
Anthem Electronics ..... (516) 864-6600

Hauppauge
Arrow Electronics ..... (516) 231-1000

Latham
Newark Electronics ..... (518) 783-0983

Liverpool
Newark Electronics ..... (315) 457-4873

Long Island
Anthem Electronics ..... (516) 864-6600

Melville
Arrow Electronics ..... (516) 391-1300

Rochester
Arrow Electronics ..... (716) 427-0300

Pittsford
Newark Electronics ..... (716) 381-4244

Port Chester
Zeus Electronics ..... (914) 937-7400

Wappingers Falls
Newark Electronics ..... (914) 298-2810

NORTH CAROLINA

Charlotte
Newark Electronics ..... (704) 535-5650

Greensboro
Newark Electronics ..... (910) 294-2142

Raleigh
Anthem Electronics ..... (919) 782-3550
(800) 359-3532
Arrow Electronics ..... (919) 876-3132
Newark Electronics ..... (919) 781-7677

OHIO

Centerville
Arrow Electronics ..... (513) 435-5563

Cincinnati
Newark Electronics ..... (513) 772-8181

Cleveland
Newark Electronics ..... (216) 391-9330

Columbus
Newark Electronics ..... (614) 326-0352

Dayton
Newark Electronics ..... (513) 294-8980

Solon
Arrow Electronics ..... (216) 248-3990

Toledo
Newark Electronics ..... (419) 866-0404

Youngstown
Newark Electronics ..... (216) 793-6134

OKLAHOMA

Oklahoma City
Newark Electronics ..... (405) 843-3301

Tulsa
Arrow Electronics ..... (918) 252-7537
Newark Electronics ..... (918) 252-5070

OREGON

Beaverton
ALMAC/Arrow Electronics ..... (503) 629-8090
Anthem Electronics ..... (503) 643-1114

Portland
Newark Electronics ..... (503) 297-1984

PENNSYLVANIA

Allentown
Newark Electronics ..... (610) 434-7171

Fort Washington
Newark Electronics ..... (215) 654-1434

Horsham
Anthem Electronics ..... (215) 443-5150

Pittsburgh
Arrow Electronics ..... (412) 856-9490
Newark Electronics ..... (412) 788-4790

SALES REPRESENTATIVES AND DISTRIBUTORS

U.S. AND CANADIAN DISTRIBUTORS

SOUTH CAROLINA

Greenville
Newark Electronics ..... (803) 288-9610

TENNESSEE

Brentwood
Newark Electronics ..... (615) 371-1341

Knoxville
Newark Electronics ..... (615) 588-6493

Memphis
Arrow Electronics ..... (901) 367-0540
Newark Electronics ..... (901) 396-7970

TEXAS

Austin
Anthem Electronics ..... (512) 388-0049
Arrow Electronics ..... (512) 835-4180
Newark Electronics ..... (512) 338-0287

Carrollton
Arrow Electronics ..... (214) 380-9049
Zeus Electronics ..... (214) 380-4330

Corpus Christi
Newark Electronics ..... (512) 857-5621

Dallas
Newark Electronics ..... (214) 458-2528

El Paso
Newark Electronics ..... (915) 772-6367

Houston
Arrow Electronics ..... (713) 647-6868
Newark Electronics ..... (713) 894-9334

Richardson
Anthem Electronics ..... (214) 238-7100

San Antonio
Newark Electronics ..... (210) 734-7960

UTAH

Salt Lake City
Anthem Electronics ..... (801) 973-8555
Arrow Electronics ..... (801) 973-6913
Newark Electronics ..... (801) 261-5660

VIRGINIA

Herndon
Newark Electronics ..... (703) 707-9010

Richmond
Newark Electronics ..... (804) 282-5671

Roanoke
Newark Electronics ..... (703) 772-6821

WASHINGTON

Bellevue
ALMAC/Arrow Electronics ..... (206) 643-9992
Newark Electronics ..... (206) 641-9800

Bothell
Anthem Electronics ..... (206) 483-1700

Spokane
ALMAC/Arrow Electronics ..... (509) 924-9500
Newark Electronics ..... (509) 327-1935

WEST VIRGINIA

Charleston
Newark Electronics ..... (304) 345-3086

WISCONSIN

Brookfield
Arrow Electronics ..... (414) 792-0150

Green Bay
Newark Electronics ..... (414) 494-1400

Madison
Newark Electronics ..... (608) 221-4738

Milwaukee
Newark Electronics ..... (414) 453-9100

CANADA

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

British Columbia
Arrow Electronics ..... (604) 421-2333
Future Electronics ..... (604) 294-1166

Manitoba
Future Electronics ..... (204) 944-1446

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

Ontario
Arrow Electronics ..... (613) 226-6903
Arrow Electronics ..... (905) 670-7769
Future Electronics ..... (905) 612-9200
Future Electronics ..... (613) 820-8313
Newark Electronics ..... (519) 685-4280
Newark Electronics ..... (905) 670-2888

Toronto
Arrow Electronics ..... (416) 670-2010

Quebec
Arrow Electronics ..... (418) 871-7500
Future Electronics ..... (418) 877-6666
Newark Electronics ..... (514) 738-4488

SALES REPRESENTATIVES AND DISTRIBUTORS

CENTRAL AND SOUTH AMERICA

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

ARGENTINA
Buenos Aires
YEL SRL 011-541-440-1532

BRAZIL
Sao Paulo
Nishicom 011-55-11-535-1755

ASIA-PACIFIC

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

CHINA
Beijing
China Electronics Appliance Corp. 86-755-335-4214
TLG Electronics, Ltd. 85-2-388-7613

HONG KONG
Electrocon Products, Ltd. 852-481-6022
Components Agent, Ltd. 852-487-8826
Maxisum, Ltd. 852-410-2780
MEMEC, Ltd. 852-410-2777

INDIA
Bangalore
Maxvale (S) Pte. Ltd. 91-802-215308
Zenith Technologies Pvt. Ltd. 91-812-586782
Bombay
Zenith Technologies Pvt. Ltd. 91-22-4947457
New Delhi
Maxvale (S) Pte. Ltd. 91-11-643-9872

INDONESIA
Jakarta
Cinergi Asiamaju 62-21-7982762

JAPAN
Tokyo
Teksel Co., Ltd. 81-3-5467-9000
Internix Incorporated 81-3-3369-1105
Kanematsu Elec. Components Corp. 81-3-3779-7811
Osaka
Teksel Co., Ltd. 81-6368-9000

KOREA
ENC-Korea 822-523-2220
MEMEC, Ltd. 822-518-8181

MALAYSIA
L.T. Electronics Ltd. 60-4-882895

NEW ZEALAND
GEC Electronics Division 64-9-526-0107

PHILIPPINES
Alexan Commercial 63-2-241-9493

SINGAPORE
Cinergi Technology & Devices Pte. Ltd. 65-778-9331
Eltee Electronics Ltd. 65-283-0888
MEMEC, Ltd. 65-222-4962

TAIWAN (ROC)
Acer Sertek, Inc. 886-2-501-0055
Asec Int'l. Corporation 886-2-786-6677
MEMEC, Ltd. 886-2-760-2028
Promate Electronics Co. Ltd. 886-2-659-0303

THAILAND
Eltee Electronics Ltd. 66-2-933-7565

SALES REPRESENTATIVES AND DISTRIBUTORS

EUROPE

AUSTRIA

Vienna
EBV Elektronik GMBH 43-222-8941-774
Avnet/Electronic 2000 0043-1-9112847

BELGIUM

Antwerp
D & D Electronics PVBA 32-3-8277934
Zaventem
EBV Elektronik 322-7209936

DENMARK

Brondby
Ditz Schweitzer AS 4542-453044
Lynge
Rep Delco 45-35-821200

ENGLAND

Berkshire
Future Electronics 44-753-521193
Gothic Crellon 44-734-787848
Macro Marketing 44-628-604383
Kent
Arrow Electronics 44-732-74039
Lancashire
Complementary Technologies Ltd. 44-942-274731

FINLAND

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

FRANCE

Cedex
A2M 331-395-49-113
CCI Electronique 331-46744700
Champs sur Marne
EBV Elektronik 331-646-88600
Massy
Reptronic SA 331-60139300

GERMANY

Berlin
EBV Elektronik GMBH 030-3421041
Avnet/Electronic 2000 030-2110761
Burgwedel
EBV Elektronik GMBH 05139-80870
Camberg
Thesys A/E 49-6434-5041
Castrop
Future GMBH 02305-42051
Dortmund
Future GMBH 02305-42051
Duesseldorf
Avnet/Electronic 2000 0211-9200385
Thesys/AE 0211-536020
Erfurt
Thesys 0361-4278100
Erkrath
Avnet/Electronic 2000 211-92003-85
Frankfurt
EBV Elektronik GMBH 069-785037
Avnet/Electronic 2000 069-9738041
Future GMBH 06121-54020
Thesys/AE 06434-5041
Gerlingen
Avnet/Electronic 2000 7156-356190
Hamburg
Avnet/Electronic 2000 040-64557021
Leonberg
EBV Elektronik GMBH 07152-30090
Muenchen
Avnet/Electronic 2000 089-4511004
EBV Elektronik GMBH 089-456100
Future GMBH 089-957270
Thesys A/E 89-99355866
Nuernberg
Avnet/Electronic 2000 0911-9951610
Neuss
EBV Elektronik GMBH 02131-96770
Quickborn
Future GMBH 4106-71022
Rauxel
Future GMBH 02305-42051
Stuttgart
Avnet/Electronic 2000 07156-356190
Future GMBH 0711-830380
Thesys/AE 0711-9889100
Weissbach
EBV Elektronik GMBH 036-426486

SALES REPRESENTATIVES AND DISTRIBUTORS

ISRAEL
RDT 972-35483137

ITALY
Milano
Avnet EMG S.R.L. 0039-295-343600
EBV Elektronik 0039-2-66017111
Silver Star 02-66-125-1
Firenze
EBV Elektronik 0039-55-350792
Roma
EBV Elektronik 0039-6-2253367
Modena
EBV Elektronik 0039-59-344752
Napoli
EBV Elektronik 0039-81-2395540
Torino
EBV Elektronik 0039-11-2161531

NETHERLANDS
EBV Elektronik 313-465-2353

NORWAY
Bexab Norge 47-63833800

POLAND
Warsaw
Gamma Ltd. 004822-330853

PORTUGAL
Amadora
Amitron-Arrow. 0035-1-4714806

RUSSIA
Woronesh
Thesys/Intechna 0732553697
Vyborg
Gamma Ltd. 81278-31509
St. Petersburg
Gamma Ltd. 812-5311402

SPAIN
Barcelona
Amitron-Arrow S.A. 0034-3-4907494
Madrid
Amitron-Arrow S.A. 0034-1-3043040

SWEDEN
Bexab Sweden AB 46-8-63088-00
Rep Delco Sweden AB 46-8-63086-00

SWITZERLAND
Dietikon
EBV Elektronik GMBH 0041-1-7401090
Lausanne
EBV Elektronik AG 0041-21-3112804
Regensdorf
Eurodis AG 0041-1-8433111

UKRAINE
Kiev
Thesys/Mikropribor 44-434-9533

Zilog's Customer Development Center

The Zilog Customer Development Center is available to provide product information, answer preliminary technical questions, and review application needs and requirements for existing and potential customers.

By utilizing an automated database program, the Customer Development Center team can track all phone and written inquiries to develop future ongoing customer relationships. Our services currently target nationwide markets, including Canada. Zilog's Application Specific Products can meet your design requirements to shorten your time to market, and the Customer Development Center is ready to offer prompt assistance.

For immediate assistance in the U.S. or Canada, contact our Customer Development Center for product information:

408-370-8016 (Eastern, Southern U.S.)

408-370-8358 (California, Arizona, New Mexico, Texas, Louisiana, Arkansas and Oklahoma)

408-370-8357 (Northwest, Central U.S. and Canada)

For international assistance:

CHINA
Shenzhen 86-755-2236089
Shanghai 86-21-4370050, x5204
86-21-4331020

GERMANY
Munich 49-8967-2045
Sömmerda 49-3634-23906

JAPAN
Tokyo 81-3-5272-0203

HONG KONG
Kowloon 852-7238979

KOREA
Seoul 82-2-577-3272

SINGAPORE
Singapore 65-2357155

TAIWAN
Taipei 886-2-741-3125

UNITED KINGDOM
Maidenhead 44-628-392-00

Z8® MICROCONTROLLERS - CONSUMER FAMILY OF PRODUCTS

Databooks By Market Niche	Part No	Unit Cost
Z8® Microcontrollers Databook	**DC-8305-03**	**$ 5.00**

Product Specifications

Z86B07 CMOS Z8 8-Bit MCU for Battery Charging and Monitoring
Z86C05/C07 CMOS Z8 8-Bit Microcontroller
Z86E07 CMOS Z8 8-Bit OTP Microcontroller
Z86C11 CMOS Z8 Microcontroller
Z86C12 CMOS Z8 In-Circuit Microcontroller Emulator
Z86C21 8K ROM Z8 CMOS Microcontroller
Z86E21 CMOS Z8 8K OTP Microcontroller
Z86C61/62/96 CMOS Z8 Microcontrollers
Z86E61/63 16K/32K EPROM CMOS Z8 Microcontrollers
Z86C63/64 32K ROM Z8 CMOS Microcontrollers
Z86C91 CMOS Z8 ROMless Microcontroller
Z86C93 CMOS Z8 Multiply/Divide Microcontroller
Z86117/717 Z8 8-Bit CMOS OTP/ROM Microcontrollers

Application Notes

On-Chip Oscillator Design
Designing a Low-Cost Thermal Printer

Support Product Specifications

Z0860000ZCO Evaluation Board
Z86C1200ZEM Emulator
Z86E0700ZDP Adaptor Kit
Z86E2100ZDF Adaptor Kit
Z86E2100ZDP Adaptor Kit
Z86E2100ZDV Adaptor Kit
Z86E2101ZDP Adaptor Kit
Z86E2101ZDV Adaptor Kit
Z86C6100TSC Emulator
Z86C6200ZEM Emulator
Z86C9300ZEM Emulator
Z8 S Series Emulators, Base Units and Pods

Additional Information

Zilog's Superintegration™ Products Guide
General Terms and Conditions of Sale
Zilog's Sales Offices, Representatives and Distributors
Literature Guide & Third Party Support Vendors

Z8® MICROCONTROLLERS - CONSUMER FAMILY OF PRODUCTS

Databooks By Market Niche	Part No	Unit Cost
Infrared Remote (IR) Controllers Databook	**DC-8301-04**	**$ 5.00**

Product Specifications

Z86L03/L06 Low Voltage CMOS Consumer Controller Processor
Z86L29 6K Infrared (IR) Remote (ZIRC™) Controller
Z86L70/L71/L72/L75/L76 Zilog IR (ZIRC™) CCP™ Controller Family
Z86L73/74/77 24/32K ROM Infrared Remote Controller (ZIRC™)
Z86E72/E73/E74/77 Zilog IR (ZIRC™) CCP™ Controller Family
Z86C72/76 Zilog Infrared Remote Controller Family (ZIRC™)
Z86L78 16K, 20-Pin Zilog Infrared Remote Controller (ZIRC™)

Application Note

Beyond the 3 Volt Limit
X-10 Compatible Infrared Remote Control

Support Product Specifications

Z86C50000ZEM Emulator
Z86L7100ZDB Emulator Board
Z86L7100ZEM ICEBOX™ In-Circuit Emulator Board

Additional Information

Zilog's Superintegration™ Products Guide
Literature Ordering Guide
Zilog's Sales Offices, Representatives and Distributors

Z8® MICROCONTROLLERS - CONSUMER FAMILY OF PRODUCTS

Databooks By Market Niche	Part No	Unit Cost
Discrete Z8® Microcontrollers	**DC 8318-02**	**$ 5.00**

Product Specifications

Z86C03/C06 CMOS Z8® 8-Bit Consumer Controller Processors
Z86E03/E06 CMOS Z8® 8-Bit OTP Consumer Controller Processors
Z86C04/C08 CMOS Z8® 8-Bit Low Cost 1K/2K ROM Microcontrollers
Z86E04/E08 CMOS Z8® 8-Bit OTP Microcontrollers
Z86C07 CMOS Z8® 8-Bit Microcontroller
Z86E07 CMOS Z8® 8-Bit OTP Microcontroller
Z86C30/C31 CMOS Z8® 8-Bit Consumer Controller Processors
Z86E30/E31 CMOS Z8® 8-Bit OTP Consumer Controller Processors
Z86C40 CMOS Z8® 4K ROM Consumer Controller Processor
Z86E40 CMOS Z8® 8-Bit OTP Consumer Controller Processor

Z8® Microcontrollers Application Notes

Timekeeping with the Z8®
Using The Zilog Z86C06 SPI Bus
DTMF Tone Generation Using the Z8® CCP™
Serial Communications Using the Z8® CCP™ Software UART
The Versatile Z86C08: Three Key Features of this Z8® MCU
The Z86C08 Controls a Scrolling LED Message Display
Interfacing LCDs to the Z8® Microcontroller

Support Product Specifications and Third-Party Vendors

Z86C0800ZCO Evaluation Board
Z86C0800ZDP Adaptor Kit
Z86C1200ZEM Emulator
Z86E0600ZDP Adaptor Kit
Z86E0700ZDP Adaptor Kit
Z86E3000ZDP Adaptor Kit
Z86E4000ZDF Adaptor Kit
Z86E4000ZDP Adaptor Kit
Z86E4000ZDV Adaptor Kit
Z86E4001ZDF Adaptor Kit
Z86E4001ZDV Adaptor Kit
Z86CCP00ZEM Emualtor
Z86CCP00ZAC Emulator Kit
Z8® S Series Emulators, Base Units and Pods
Third-Party Support Vendors

Additional Information

Zilog's Superintegration™ Products Guide
Literature Guide and Ordering Information
Zilog's Sales Offices, Representatives and Distributors

Z8® MICROCONTROLLERS - CONSUMER FAMILY OF PRODUCTS

Databooks By Market Niche	Part No	Unit Cost
Digital Television Controllers	**DC-8308-01**	**$ 5.00**

Product Specifications
- Z89300 Series Digital Television Controller
- Z86C27/97 CMOS Z8® Digital Signal Processor
- Z86C47/E47 CMOS Z8® Digital Signal Processor
- Z86127 Low Cost Digital Television Controller
- Z86128/228 Line 21 Closed-Caption Controller (L21C™)
- Z86227 40-Pin Low Cost (4LDTC™) Digital Television Controller

Support Product Specifications
- Z86C2700ZCO Application Kit
- Z86C2700ZDB Emulation Board
- Z86C2702ZEM In-Circuit Emulator

Additional Information
- Zilog's Superintegration™ Products Guide
- Literature Guide and Ordering Information
- Zilog's Sales Offices, Representatives and Distributors

Telephone Answering Device Databook	**DC-8300-03**	**$ 5.00**

Product Specifications
- Z89165/166 (ROMless) Low-Cost DTAD Controller (Preliminary)
- Z89167/169 Z89168 (ROMless) Enhanced Dual Processor Tapeless TAM Controller (Preliminary)

Development Guides
- Z89165 Software Developer's Manual
- Z89167/169 Software Developer's Manual

Technical Notes
- Z89165/167/169 Design Guidelines
- Z89167/169 Codec Interfacing Preliminary
- Controlling the Out -5V and Codec Clock Signals for Low-Power Halt Mode
- Z89165/166 Input A/D and Electronic Hybrid
- Z89C67/C69/167/169 Low-Power Halt Mode Sequence
- Samsung KT8554 Codec
- Watch-Dog Timer For TAD Applications
- Zilog LPC Words Listing

Support Product Specifications
- Z89C5900ZEM Emulation Module
- Z89C6500ZDB Emulation Board
- Z89C6501ZEM ICEBOX™ In-Circuit Emulator
- Z89C6700ZDB Emulator Board
- Z89C6700ZEM ICEBOX™ Emulator Board

Additional Information
- Zilog's Superintegration™ Products Guide
- Literature Ordering Guide
- Zilog's Sales Offices, Representatives and Distributors

Z8® MICROCONTROLLERS - PERIPHERALS MULTIMEDIA FAMILY OF PRODUCTS

Databooks By Market Niche	Part No	Unit Cost
Digital Signal Processor Databook	**DC-8299-04**	**$ 5.00**

Product Specifications

Z89321/371 16-Bit Digital Signal Processor (Preliminary)
Z89C00 16-Bit Digital Signal Processor (Preliminary)
Z89320 16-Bit Digital Signal Processor (Preliminary)
Z86C95 Z8® Digital Signal Processor (Preliminary)
Z89120, Z89920 (ROMless) 16-Bit Mixed Signal Processor (Preliminary)
Z89121, Z89921 (ROMless) 16-Bit Mixed Signal Processor (Preliminary)

Application Note

Using the Z89371/321 CODEC Interface
Z89371 Inter Processor Communication
Understanding Q15 Two's Complement Fractional Multiplication (Z89C00 DSP)

Support Product Specifications

Z8937100ZEM In-Circuit Emulator -C00
Z8937100TSC Emulation Module
Z89C0000ZAS Z89C00 Assembler, Linker and Librarian
Z89C0000ZCC Z89C00 C Cross Compiler
Z89C0000ZEM In-Circuit Emulator -C00
Z89C0000ZHP Logic Analyzer Adaptor Board
Z89C0000ZSD Z89C00 Simulator/Debugger
Z89C0000ZTR Z89C00 Translator

Additional Information

Zilog's Superintegration™ Products Guide
Literature Guide and Third Party Support
Zilog's Sales Offices, Representatives and Distributors

Z8® MICROCONTROLLERS - PERIPHERALS MULTIMEDIA FAMILY OF PRODUCTS

Databooks By Market Niche	Part No	Unit Cost
Keyboard/Mouse/Pointing Devices Databook	**DC-8304-01**	**$ 5.00**

Product Specifications

Z8602/14 NMOS Z8® 8-Bit Keyboard Controller
Z8615 NMOS Z8® 8-Bit Keyboard Controller
Z86C15 CMOS Z8® 8-Bit MCU Keyboard Controller
Z86E23 Z8® 8-Bit Keyboard Controller with 8K OTP
Z86C04/C08 CMOS Z8® 8-Bit Microcontroller
Z86E08 CMOS Z8® 8-Bit Microcontroller
Z88C17 CMOS Z8® 8-Bit Microcontroller
Z86C117/717 Z8® 8-Bit Microcontroller
Z86217 Z8® 8-Bit Microcontroller

Application Notes

Z8602 Keyboard
Z86C17 In-Mouse Applications

Support Product Specifications and Third Party Support

Z0860200ZCO Evaluation Board
Z0860200ZDP Adaptor Kit
Z86C0800ZCO Evaluation Board
Z86C0800ZDP Adaptor Kit
Z86C1200ZEM Emulator
Z86E2300ZDP Adaptor Kit
Z86E2301ZDP Adaptor Kit
Z86E2300ZDV Adaptor Kit
Z86E2301ZDV Adaptor Kit

Additional Information

Zilog's Superintegration™ Products Guide
Literature Guide and Ordering Information
Zilog's Sales Offices, Representatives and Distributors

PC Audio Databook	**DC-8317-00**	**$ 5.00**

Product Specifications

Z86321 Digital Audio Processor (Preliminary)
Z89320 16-Bit Digital Signal Processor (Preliminary)
Z89321/371 16-Bit Digital Signal Processor (Preliminary)
Z89331 16-Bit PC ISA Bus Interface (Advance Information)
Z89341/42/43 Wave Synthesis Chip Set (Advance Information)
Z5380 Small Computer System Interface

Additional Information

Zilog's Superintegration™ Products Guide
Literature Guide

Z8® MICROCONTROLLERS - PERIPHERALS MEMORY FAMILY OF PRODUCTS

Databooks By Market Niche	Part No	Unit Cost
Mass Storage Solutions	**DC-8303-01**	**$ 5.00**

Product Specifications

Z86C21 8K ROM Z8 CMOS Microcontroller
Z86E21 CMOS Z8 8K OTP Microcontroller
Z86C91 CMOS Z8 ROMless Microcontroller
Z86C93 CMOS Z8 Multiply/Divide Microcontroller
Z86C95 Z8 Digital Signal Processor
Z86018 Data Path Controller
Z89C00 16-Bit Digital Signal Processor

Application Note

Understanding Q15 Two's Complement Fractional Multiplication (Z89C00 DSP)

Support Product Specifications

Z8060000ZCO Development Kit
Z86C1200ZEM In-Circuit Emulator
Z86E2100ZDF Adaptor Kit
Z86E2100ZDP Adaptor Kit
Z86E2100ZDV Adaptor Kit
Z86E2101ZDF Conversion Kit
Z86E2101ZDV Conversion Kit
Z86C9300ZEM ICEBOX™ Emulator
Z86C9500ZCO Evaluation Board
Z8® S Series Emulators, Base Units and Pods
Z89C0000ZAS Z89C00 Assembler, Linker and Librarian
Z89C0000ZCC Z89C00 C Cross Compiler
Z89C0000ZEM In-Circuit Emulator -C00
Z89C0000ZSD Z89C00 Simulator/Debugger
ZPCMCIA0ZDP PCMCIA Extender Card

Additional Information

Zilog's Superintegration™ Products Guide
Zilog's Literature Guide
Zilog's Sales Offices, Representatives and Distributors

Z8® MICROCONTROLLERS LITERATURE (Continued)

Technical Manuals and Users Guides	Part No.	Unit Cost
Z8® Microcontrollers Technical Manual	DC-8291-02	5.00
Z86018 Preliminary User's Manual	DC-8296-00	N/C
Digital TV Controller User's Manual	DC-8284-01	5.00
Z89C00 16-Bit Digital Signal Processor User's Manual/DSP Software Manual	DC-8294-02	5.00
Z86C95 16-Bit Digital Signal Processor User Manual	DC-8595-02	5.00
Z86017 PCMCIA Adaptor Chip User's Manual and Databook	DC-8298-03	5.00
PLC Z89C00 Cross Development Tools Brochure	DC-5538-01	N/C

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

Z80®/Z8000® DATACOMMUNICATIONS FAMILY OF PRODUCTS

Databooks By Market Niche	Part No	Unit Cost
High-Speed Serial Communication Controllers	**DC-8314-01**	**5.00**

Product Specifications
- Z16C30 CMOS Universal Serial Controller (USC™) (Preliminary)
- Z16C32 Integrated Universal Serial Controller (IUSC™) (Preliminary)

Application Notes
- Using the Z16C30 Universal Serial Controller with MIL-STD-1553B
- Design a Serial Board to Handle Multiple Protocols
- Datacommunications IUSC™/MUSC™ Time Slot Assigner

Support Products and Third Party Vendor Support
- Z16C3001ZCO Evaluation Board Product Specification
- Z16C3200ZCO Evaluation Board Product Specification
- Z8018600ZCO Evaluation Board Product Specification
- ZEPMDC00001 EPM™ Electronic Programmer's Manual Product Specification
- Third Party Vendors

Additional Information
- Zilog's Superintegration™ Products Guide
- General Terms and Conditions of Sale
- Sales Offices, Representatives and Distributors
- Literature Guide

Serial Communication Controllers	**DC-8316-01**	**5.00**

Product Specifications
- Z8030/Z8530 Z-Bus® SCC Serial Communication Controller
- Z80C30/Z85C30 CMOS Z-Bus® SCC Serial Communication Controller
- Z80230 Z-Bus® ESCC™ Enhanced Serial Communication Controller (Preliminary)
- Z85230 ESCC™ Enhanced Serial Communication Controller
- Z85233 EMSCC™ Enhanced Mono Serial Communication Controller
- Z85C80 SCSCI™ Serial Communications and Small Computer Interface
- Z16C35/Z85C35 CMOS ISCC™ Integrated Serial Communications Controller

Application Notes
- Interfacing Z8500 Peripherals to the 68000
- SCC in Binary Synchronous Communications
- Zilog SCC Z8030/Z8530 Questions and Answers
- Integrating Serial Data and SCSI Peripheral Control on One Chip
- Zilog ISCC™ Controller Questions and Answers
- Boost Your System Performance Using the Zilog ESCC™
- Zilog ESCC™ Controller Questions and Answers
- The Zilog Datacom Family with the 80186 CPU
- On-Chip Oscillator Design

Support Products
- Z8S18000ZCO Evaluation Board Product Specification
- Z8523000ZCO Evaluation Board Product Specification
- Z8018600ZCO Evaluation Board Product Specification
- ZEPMDC00002 Electronic Programmer's Manual Software

Additional Information
- Zilog's Superintegration™ Products Guide
- Sales Offices, Representatives and Distributors
- Literature Guide

Z80®/Z8000® DATACOMMUNICATIONS FAMILY OF PRODUCTS

Databooks	Part No	Unit Cost
Z80 Family Databook	**DC-8321-00**	**5.00**

Discrete Z80® Family

Z8400/C00 NMOS/CMOS Z80® CPU Product Specification
Z8410/C10 NMOS/CMOS Z80 DMA Product Specification
Z8420/C20 NMOS/CMOS Z80 PIO Product Specification
Z8430/C30 NMOS/CMOS Z80 CTC Product Specification
Z8440/C40 NMOS/CMOS Z80 SIO Product Specification

Embedded Controllers

Z84C01 Z80 CPU with CGC Product Specification
Z8470 Z80 DART Product Specification
Z84C90 CMOS Z80 KIO™ Product Specification
Z84013/015 Z84C13/C15 IPC/EIPC Product Specification

Application Notes and Technical Articles

Z80® Family Interrupt Structure
Using the Z80® SIO with SDLC
Using the Z80® SIO in Asynchronous Communications
Binary Synchronous Communication Using the Z80® SIO
Serial Communication with the Z80A DART
Interfacing Z80® CPUs to the Z8500 Peripheral Family
Timing in an Interrupt-Based System with the Z80® CTC
A Z80-Based System Using the DMA with the SIO
Using the Z84C11/C13/C15 in Place of the Z84011/013/015
On-Chip Oscillator Design
A Fast Z80® Embedded Controller
Z80® Questions and Answers

Additional Information

Zilog's Superintegration™ Products Guide
Literature Guide
Third Party Support Vendors
Zilog's Sales Offices, Representatives and Distributors

Z80®/Z8000® DATACOMMUNICATIONS FAMILY OF PRODUCTS

Databooks	Part No	Unit Cost
Z180™ Microprocessors and Peripherals Databook	**DC-8322-01**	**5.00**

Product Specifications

Z80180/Z8S180/Z8L180 Z180™ Microprocessor
Z80181 Z181™ Smart Access Controller (SAC™)
Z80182/Z8L182 Zilog Intelligent Peripheral Controller (ZIP™)

Application Notes and Technical Articles

Z180™ Questions and Answers
Z180™/SCC Serial Communication Controller Interface at 10 MHz
Interfacing Memory and I/O to the 20 MHz Z8S180 System
Break Detection on the Z80180 and Z181™
Local Talk Link Access Protocol Using the Z80181
Z182 Programming the MIMIC Autoecho ECHOZ182 Sample Code
High Performance PC Communication Port Using the Z182
Improving Memory Access Timing in Z182 Applications

Support Products

Z8S18000ZCO Evaluation Board
Z8018100ZCO Evaluation Board
Z8018101ZCO Evaluation Board
Z8018101ZA6 Driver Software
Z8018100ZDP Adaptor Kit
Z8018200ZCO Evaluation Board
ZEPMIP00001 EPM™ Electronic Programmer's Manual
ZEPMIP00002 EPM Electronic Programmer's Manual
Z80® and Z80180 Hardware and Software Support

Additional Information

Zilog's Superintegration™ Products Guide
Literature Guide
Zilog's Sales Offices, Representatives and Distributors

Z80®/Z8000® DATACOMMUNICATIONS FAMILY OF PRODUCTS

Databooks and User's Manuals	Part No	Unit Cost
Z8000 Family of Products	**DC-8319-00**	**5.00**
Z8000 Family Databook		
Zilog's Z8000 Family Architecture		
Z8001/Z8002 Z8000 CPU Product Specification		
Z8016 Z8000 Z-DTC Product Specification		
Z8036 Z8000 Z-CIO Product Specification		
Z8536 CIO Counter/Timer and Parallel I/O Unit Product Specification		
Z8038/Z8538 FIO FIFO Input/Output Interface Unit Product Specification		
Z8060/Z8560 FIFO Buffer Unit		
Z8581 Clock Generator and Controller Product Specification		
User's Manuals		
Z8000 CPU Central Processing Unit User's Manual		
Z8010 Memory Management Unit (MMU) User's Manual		
Z8036 Z-CIO/Z8536 CIO Counter/Timer and Parallel Input/Output User's Manual		
Z8038 Z8000 Z-FIO FIFO Input/Output Interface User's Manual		
Z8000 Application Notes and Military Products		
Application Notes		
Using SCC with Z8000 in SDLC Protocol		
SCC in Binary Synchronous Communication		
Zilog's Military Products Overview		
Additional Information		
Zilog's Superintegration™ Products Guide		
Literature Guide		
Zilog's Sales Offices, Representatives and Distributors		
Z80 Family Microprocessor Family User's Manual	**DC-8309-01**	**5.00**
User's Manuals		
Z80 Central Processing Unit (CPU)		
Z80 Counter Timer Channels (CTC)		
Z80 Direct Memory Access (DMA)		
Z80 Parallel Input/Output (PIO)		
Z80 Serial Input/Output (SIO)		
Additional Information		
Zilog's Superintegration™ Products Guide		
Zilog's Sales Offices, Representatives and Distributors		
Literature Guide		

Databooks and User's Manuals	Part No	Unit Cost
Z80180 Z180 MPU Microprocessor Unit Technical Manual	DC-8276-04	5.00
Z280 MPU Microprocessor Unit Technical Manual	DC-8224-03	5.00
Z380™ Preliminary Product Specification	DC-6003-03	N/C
Z380™ User's Manual	DC-8297-03	5.00
Z2000 Spread-Spectrum Transceiver Advance Information Product Specification	DC-6021-00	N/C
ZNW2000 User's Manual for PC WAN Adaptor Board Development Kit	DC-8315-00	N/C
SCC Serial Communication Controller User's Manual	DC-8293-02	5.00
High-Speed SCC, Z16C30/Z16C32 User's Manual	DC-8350-00	5.00
Z16C35 ISCC Integrated Serial Communication Controller Technical Manual	DC-8286-01	5.00
Z16C35 ISCC Integrated Serial Communication Controller Addendum	DC-8286-01A	N/C

MILITARY COMPONENTS FAMILY

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

GENERAL LITERATURE

Catalogs, Handbooks, Product Flyers and Users Guides	Part No	Unit Cost
Superintegration Master Selection Guide 1994-1995	DC-5634-01	N/C
Superintegration Products Guide	DC-5676-00	N/C
Quality and Reliability Report	DC-8329-01	N/C
ZIA™ 3.3-5.5V Matched Chip Set for AT Hard Disk Drives Datasheet	DC-5556-01	N/C
ZIA ZIA00ZC0 Disk Drive Development Kit Datasheet	DC-5593-01	N/C
Zilog Hard Disk Controllers - Z86C93/C95 Datasheet	DC-5560-01	N/C
Zilog Infrared (IR) Controllers - ZIRC™ Datasheet	DC-5558-01	N/C
Zilog V. Fast Modem Controller Solutions	DC-5525-02	N/C
Zilog Digital Signal Processing - Z89320 Datasheet	DC-5547-01	N/C
Zilog Keyboard Controllers Datasheet	DC-5600-01	N/C
Z380™ - Next Generation Z80®/Z180™ Datasheet	DC-5580-02	N/C
Fault Tolerant Z8® Microcontroller Datasheet	DC-5603-01	N/C
32K ROM Z8® Microcontrollers Datasheet	DC-5601-01	N/C
Zilog Datacommunications Brochure	DC-5519-00	N/C
Z89300 DTC Controller Family Brochure	DC-5608-01	N/C
Zilog Digital Signal Processing Brochure	DC-5536-02	N/C
Zilog ASSPs - Partnering With You Product Brochure	DC-5553-01	N/C
Zilog Wireless Products Datasheet	DC-5630-00	N/C
Zilog Z8604 Cost Efficient Datasheet	DC-5662-00	N/C
Zilog Chip Carrier Device Packaging Datasheet	DC-5672-00	N/C
Zilog Database of IR Codes Datasheet	DC-5631-00	N/C
Zilog PCMCIA Adaptor Chip Z86017 Datasheet	DC-5585-01	N/C
Zilog Television/Video Controllers Datasheet	DC-5567-01	N/C
Zilog TAD Controllers - Z89C65/C67/C69 Datasheet	DC-5561-02	N/C
Zilog Z87000 Z-Phone Datasheet	DC-5632-00	D/C
Zilog 1993 Annual Report	DC-1993-AR	N/C
Zilog 1994 First Quarter Financial Report	DC-1994-Q1	N/C

ZiLOG LITERATURE GUIDE

ORDERING INFORMATION

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

MINIMUM ORDER REQUIREMENTS

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

SHIPMENT

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

RETURNS ARE NOT ACCEPTED.

PLEASE PRINT OR TYPE

NAME	PHONE ( ) -
COMPANY	**Method of Payment (Check One)**
ADDRESS	☐ Check ☐ Money Order Credit Card ☐ VISA ☐ M/C ☐ P.O. (over $300.00)
CITY / STATE / ZIP	COUNTRY

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

Credit Card or Purchase Order # _____ SUBTOTAL

Expiration Date _____ ADD APPLICABLE SALES TAX (CA ONLY)

Signature _____ ADD 10% SHIPPING AND HANDLING

TOTAL

Mail To:

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

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