ZiLOG

PRELIMINARY PRODUCT SPECIFICATION

Z85C80

SCSCI SERIAL COMMUNICATIONS AND SMALL COMPUTER INTERFACE

FEATURES

- Low power CMOS
- Two independent, 0 to 2.5M bit/second, full-duplex channels, each with a separate crystal oscillator, baud rate generator, and Digital Phase-Locked Loop for clock recovery.
- Multi-protocol operation under program control; programmable for NRZ, NRZI, or FM data encoding.
- Asynchronous mode with five to eight bits and one, one and one-half, or two stop bits per character, programmable clock factor, break detection and generation; parity, overrun, and framing error detection.
- Synchronous mode with internal or external character synchronization on one or two synchronous characters and CRC generation and checking with CRC-16 or CRC-CCITT preset to either 1s or 0s.
- SDLC/HDLC mode with comprehensive frame-level control, automatic zero insertion and deletion, I-field residue handling, abort generation and detection, CRC generation and checking, and SDLC Loop mode operation.
- Local Loopback and Auto Echo modes.
- Supports T1 digital trunk.
- Enhanced DMA support
  10 X 19-bit status FIFO
  14-bit byte counter
- Arbitration Support
- DMA or Programmed I/O Data Transfers
- Supports Normal DMA
- Memory or I/O Mapped CPU Interface.
- Asynchronous Interface, Supports 3 MB/s
- Direct SCSI Bus Interface with On-Board 48 mA Drivers
- Supports Target and Initiator Roles

GENERAL DESCRIPTION

The Z85C80 CMOS SCSCI is an industry standard 85C30 dual channel Serial Communication Controller (SCC) and an industry standard 53C80 Small Computer System Interface (SCSI) integrated into one monolithic Integrated Circuit. The internal SCC and SCSI share the 8-bit data bus (D7 through D0) and read and write inputs (/RD and /WR).

Note: All Signals with a preceding front slash, "/", are active Low, e.g.: B//W (WORD is active Low); /B/W (BYTE is active Low, only); /N//S (NORMAL and SYSTEM are both active Low).

The Z85C80 is offered in a 68-pin PLCC package. With a few exceptions, all of the internal SCC and SCSI signals are connected to the outside pins.

GENERAL DESCRIPTION (Continued)

The exceptions are:

- IEI input to SCC is internally connected to VDD.
- IEO output from SCC is not internally connected (N/C).
- READY output from SCSI is not internally connected (N/C).
- /SYNCB output from the SCC is not internally connected (N/C).
- /TRXCA and /CTSA inputs to the SCC are internally connected.
- /TRXCB and /CTSB inputs to the SCC are internally connected.

The internal SCC is a dual channel, multi-protocol data communications peripheral that easily interfaces to CPUs with non-multiplexed address/data buses. The programming flexibility of the internal registers allows the SCC to be configured to satisfy a wide variety of serial communications applications. The many on-chip features such as baud rate generators, digital phase locked loops, and crystal oscillators dramatically reduce the need for external logic. Additional features, including a 10x19-bit status FIFO and 14-bit byte counter, were added to support high speed SDLC transfers using DMA controllers.

The SCC handles asynchronous formats, synchronous byte-oriented protocols such as IBM Bisync, and synchronous bit-oriented protocols such as HDLC and IBM SDLC. The internal SCC can generate and check CRC codes in any synchronous mode and can be programmed to check data integrity in various modes. It also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls are used for general-purpose I/O. The daisy-chain interrupt hierarchy is also supported and is standard for Zilog peripheral components.

The internal SCSI is designed to implement the SCSI protocol as defined by the ANSI X3.131-1986 standard, and is fully compatible with the industry standard 53C80. It is capable of operating both as a target and as an initiator. Special high current open-drain outputs enable it to directly interface to, and drive, the SCSI bus. The internal SCSI has the necessary interface hook-ups so the system CPU can communicate with it like with any other peripheral device. The CPU can read from, or write to, the SCSI registers which are addressed as standard or memory-mapped I/Os.

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

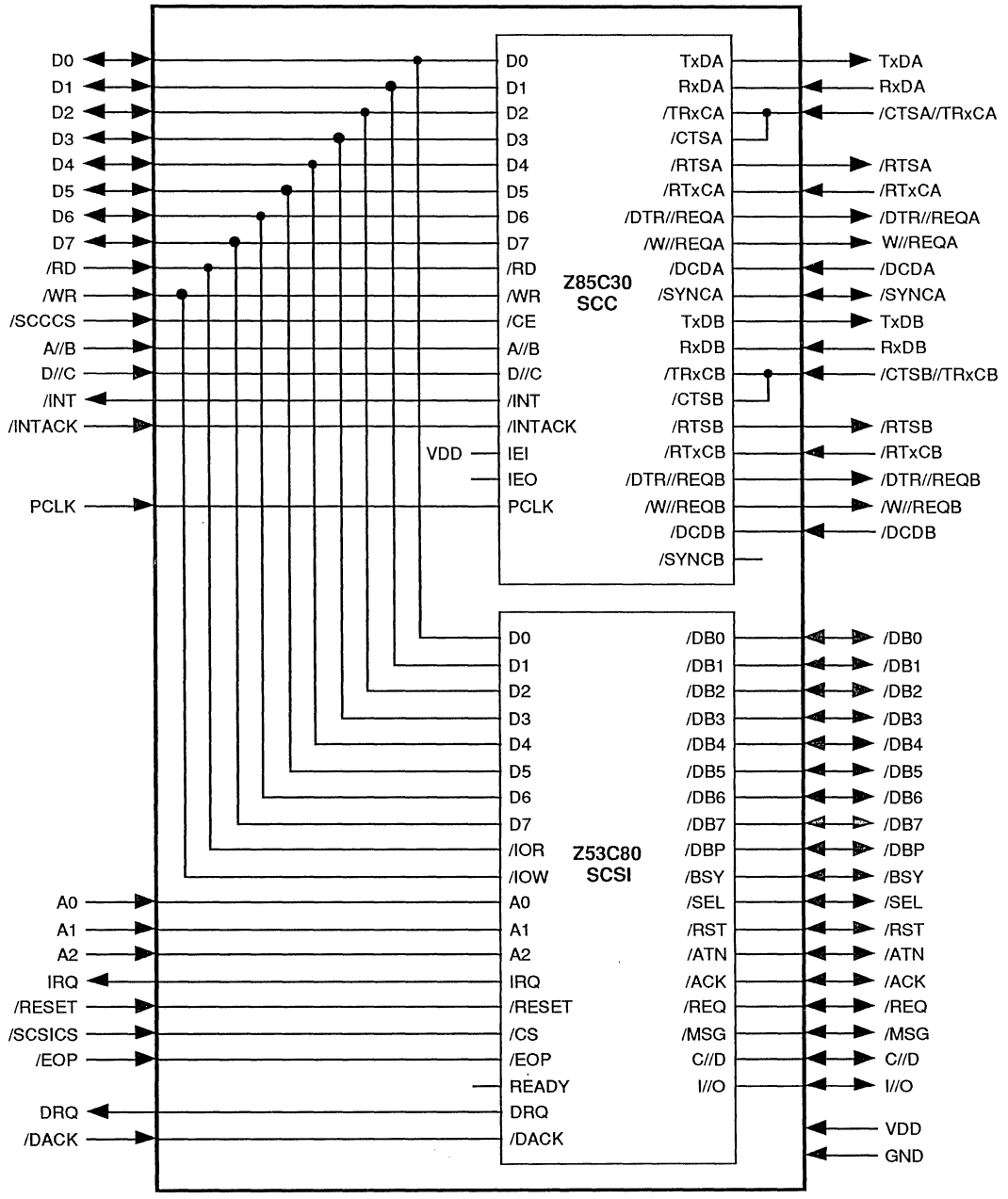
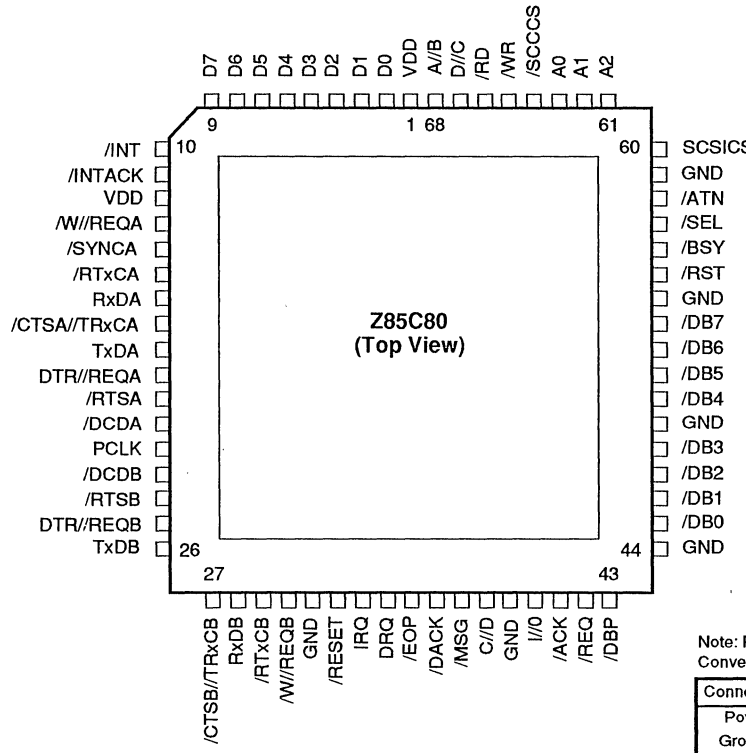


Figure 1. Z85C80 SCSCI Block Diagram



Note: Power connections follow Conventional descriptions below

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

Figure 2. 68-Pin PLCC Pin Diagram

PIN DESCRIPTION

Signal	Pin	Type	Description
A0	63	I	**SCSI Address Line Bit 0** (SCSI). Address lines are used with /SCSICS, /RD, or /WR to address all internal registers.
A1	62	I	**SCSI Address Line Bit 1** (SCSI). Address lines are used with /SCSICS, /RD, or /WR to address all internal registers.
A2	61	I	**SCSI Address Line Bit 2** (SCSI). Address lines are used with /SCSICS, /RD, or /WR to address all internal registers.
A//B	68	I	**Channel A/Channel B** (SCC). This signal selects the SCC channel in which the read or write operation occurs.
/ACK	41	I/O	**Acknowledge** (open-drain, active low, SCSI). Driven by an Initiator, /ACK indicates an acknowledgement for a /REQ//ACK data-transfer handshake. In the Target role, /ACK is received as a response to the /REQ signal.
/ATN	58	I/O	**Attention** (open-drain, active low, SCSI). Driven by an Initiator, received by the Target. /ATN indicates an Attention condition.

PIN DESCRIPTION (Continued)

Signal	Pin	Type	Description
/BSY	56	I/O	**Busy** (open-drain, active low, SCSI). This signal indicates that the SCSI bus is being used and can be driven by both the Initiator and the Target device.
C//D	38	I/O	**Control/Data** (open-drain, SCSI). Driven by the Target and received by the Initiator. C//D indicates whether Control or Data information is on the Data Bus. True indicates control.
/CTSA//TRXCA	17	I	**Clear To Send for channel A; Transmit/Receive Clock for channel A** (active low, SCC). This pin is internally connected to SCC's A Channel /CTS and /TRXC. Receive clock or the transmit clock is supplied via this pin to the SCC's A channel. When programmed as Auto Enables, a low on this pin enables the A-channel transmitter.
/CTSB//TRXCB	27	I	**Clear To Send for channel B/Transmit/Receive Clock for channel B** (active low, SCC). This pin is internally connected to SCC's B-channel /CTS and /TRXC. Receive clock or the transmit clock is supplied via this pin to the SCC's B channel. When programmed as Auto Enables, a low on this pin enables the B-channel transmitter.
D0	2	I/O	**Data bus bit 0** (tri-state, active high, SCC and SCSI). This is the Least Significant Bit of the bus. Data bus lines carry data and commands to and from the SCSCI.
D1	3	I/O	**Data bus bit 1** (tri-state, active high, SCC and SCSI). Data bus lines carry data and commands to and from the SCSCI.
D2	4	I/O	**Data bus bit 2** (tri-state, active high, SCC and SCSI). Data bus lines carry data and commands to and from the SCSCI.
D3	5	I/O	**Data bus bit 3** (tri-state, active high, SCC and SCSCI). Data bus lines carry data and commands to and from the SCSCI.
D4	6	I/O	**Data bus bit 4** (tri-state, active high, SCC and SCSI). Data bus lines carry data and commands to and from the SCSCI.
D5	7	I/O	**Data bus bit 5** (tri-state, active high, SCC and SCSI). Data bus lines carry data and commands to and from the SCSCI.
D6	8	I/O	**Data bus bit 6** (tri-state, active high, SCC and SCSI). Data bus lines carry data and commands to and from the SCSCI.
D7	9	I/O	**Data bus bit 7** (tri-state, active high, SCC and SCSI). This is the most significant bit of the bus. Data bus lines carry data and commands to and from the SCSCI.
/DACK	36	I	**DMA Acknowledge** (active low, SCSI). /DACK resets DRQ and selects the data register for input or output data transfers. /DACK is used by DMA controller instead of /SCSICS.
/DBO	45	I/O	**SCSI Data Bus bit 0** (open-drain, active low, SCSI). Least significant bit in the SCSI data bus.
/DB1	46	I/O	**SCSI Data Bus bit 1** (open-drain, active low, SCSI).
/DB2	47	I/O	**SCSI Data Bus bit 2** (open-drain, active low, SCSI).

PIN DESCRIPTION (Continued)

Signal	Pin	Type	Description
/DB3	48	I/O	**SCSI Data Bus bit 3** (open-drain, active low, SCSI).
/DB4	50	I/O	**SCSI Data Bus bit 4** (open-drain, active low, SCSI).
/DB5	51	I/O	**SCSI Data Bus bit 5** (open-drain, active low, SCSI).
/DB6	52	I/O	**SCSI Data Bus bit 6** (open-drain, active low, SCSI).
/DB7	53	I/O	**SCSI Data Bus bit 7** (open-drain, active low, SCSI). This is the most significant bit in the SCSI data bus.
/DBP	43	I/O	**SCSI Data Bus Parity bit** (open-drain, active low, SCSI). Data parity is odd. Parity is always generated and optionally checked. Parity is not valid during Arbitration.
D//C	67	I	**Data/Control Select** (SCC). This signal defines the type of information transferred to and from the SCC.
/DCDA	21	I	**Data Carrier Detect for A channel** (active low, SCC). This pin functions as receive enable if it is programmed for Auto Enable; otherwise, it may be used as general-purpose input pin. The SCC detects pulses on this pin and can interrupt the CPU on both logic level transactions.
/DCDB	23	I	**Data Carrier Detect for B channel** (active low, SCC). This pin functions as receive enable if it is programmed for Auto Enable; otherwise, it may be used as general-purpose input pin. The SCC detects pulses on this pin and can interrupt the CPU on both logic level transactions.
DRQ	34	O	**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	O	**Data Terminal Ready/Request for channel A** (active low, SCC). This output follows the state programmed into the DTR bit. It can also be used as general-purpose output or as Request line for a DMA controller.
/DTR//REQB	25	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 general-purpose output or as Request line for a DMA controller.
/EOP	35	I	**End of process** (active low, SCSI). EOP is used to terminate a DMA transfer. If asserted during a DMA cycle, the current byte will be transferred, but no additional bytes will be requested.
GND	31, 39,44, 45,54, 59	S	**Ground supply** (SCC and SCSI).
/INT	10	O	**SCC Interrupt Request** (open-drain, active low, SCC). This signal is activated when the SCC requests an interrupt.

PIN DESCRIPTION (Continued)

Signal	Pin	Type	Description
/INTACK	11	I	**Interrupt acknowledge** (active low, SCC). This signal indicates an active Interrupt Acknowledge cycle. /INTACK is latched by the rising edge of PCLK.
I//O	40	I/O	**Input/Output** (open-drain, SCSI). I/O is a signal driven by a Target which controls the direction of data movement on the SCSI bus. TRUE indicates input to the Initiator. This signal is also used to distinguish between Selection and Reselection phases.
IRQ	33	0	**SCSI Interrupt Request** (active high, SCSI). This signal alerts a microprocessor of an error condition or an event completion.
/MSG	37	I/O	**Message** (open-drain, SCSI). This signal is driven by the Target during the Message phase. This signal is received by the Initiator.
PCLK	22	I	**Clock** (SCC). This is the master SCC clock used to synchronize internal signals. PCLK is not required to have any phase relationship with the master system clock.
/RD	66	I	**Read** (active low, SCC and SCSI). When the SCC is selected, it enables the SCC's bus drivers. When the SCSI is selected, it is used in conjunction with /SCSICS and A2-A0 to read an internal register. It also selects the Input Data Register in SCSI when used with /DACK.
/REQ	42	I/O	**Request** (open-drain, active low, SCSI). Driven by a Target and received by the Initiator, this signal indicates a request for a /REQ/ /ACK data-transfer handshake.
/RESET	32	I	**SCSI Reset** (active low, SCSI). This signal clears all registers in the SCSI. It has no effect upon the SCSI /RST signal.
/RST	55	I/O	**SCSI bus Reset** (open-drain, active low, SCSI). This signal indicates a SCSI bus Reset condition.
/RTSA	20	0	**Request To Send for channel A** (active low, SCC). When the RTS bit in Write Register 5 is set, the /RTS signal goes low. When the RTS bit is reset in the Asynchronous mode and Auto Enable is on, the signal goes high after the transmitter is empty. In Synchronous mode or in Asynchronous mode with Auto Enable off, the /RTS pin strictly follows the state of the RTS bit. This pin can be used as a general-purpose output.
/RTSB	24	0	**Request To Send for channel B** (active low, SCC). When the RTS bit in Write Register 5 is set, the /RTS signal goes low. When the RTS bit is reset in the Asynchronous mode and Auto Enable is on, the signal goes high after the transmitter is empty. In Synchronous mode or in Asynchronous mode with Auto Enable off, the /RTS pin strictly follows the state of the RTS bit. This pin can be used as a general-purpose output.

PIN DESCRIPTION (Continued)

Signal	Pin	Type	Description
/RTXCA	15	I	**Receive/Transmit Clock for channel A** (active low, SCC). This pin can be programmed in several modes of operation. It may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the digital phase-locked loop. This pin can also be programmed for use with the /SYNCA pin as a crystal oscillator. The receive clock can be 1, 16, 32, or 64 times the data rate in Asynchronous modes.
/RTXCB	29	I	**Receive/Transmit Clock for channel B** (active low, SCC). This pin can be programmed in several modes of operation. It may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the digital phase-locked loop. The receive clock can be 1, 16, 32, or 64 times the data rate in Asynchronous modes.
RXDA	16	I	**Receive Data for channel A** (active high, SCC). This input signal receives serial data.
RXDB	28	I	**Receive Data for channel B** (active high, SCC). This input signal receives serial data.
/SCCCS	64	I	**SCC Chip Select** (active low, SCSI). This signal selects SCC for a read or write operation.
/SCSICS	60	I	**SCSI Chip Select** (active low, SCSI). This signal, in conjunction with /RD or /WR, enables the internal register selected by A2-A0, to be read from or written to.
/SEL	57	I/O	**Select** (open-drain, active low, SCSI). This signal is used by an Initiator to select a Target, or by a Target to reselect an Initiator.
/SYNCA	14	I/O	**Synchronization for channel A** (active low, SCC). This pin can act as input, output, or part of the crystal oscillator circuit.
TXDA	18	O	**Transmit Data for channel A** (active high, SCC). This output signal transmits serial data at standard TTL levels.
TXDB	26	O	**Transmit Data for channel B** (active high, SCC). This output signal transmits serial data at standard TTL levels.
VDD	1,12	S	**VDD supply** (SCC and SCSI).
/WR	65	I	**Write** (active low, SCC and SCSI). When the SCC is selected, this signal indicates a write operation. The coincidence of /RD and /WR is interpreted as a reset. When the SCSI is selected, it is used in conjunction with /SCSICS and A2-A0 to write an internal register. It also selects the Output Data Register in SCSI, when used with /DACK.
/W//REQA	13	O	**Wait/Request for channel A** (open-drain when programmed for a Wait function, driven high or low when programmed for a Request function, SCC). This dual purpose output may be programmed as request line for a DMA controller or as a Wait line to synchronize the CPU to the SCC data rate. The reset state is Wait.

PIN DESCRIPTION (Continued)

Signal	Pin	Type	Description
/W//REQB	30	O	**Wait/Request for channel B** (open-drain when programmed for a Wait function, driven high or low when programmed for a Request function, SCC). This dual purpose output may be programmed as request line for a DMA controller or as a Wait line to synchronize the CPU to the SCC data rate. The reset state is Wait.

FUNCTIONAL DESCRIPTION

The Z85C80 consists of an industry standard Z85C30 Serial Communication Controller (SCC) and an industry standard Z53C80 Small Computer System Interface (SCSI), sharing the data bus and read and write signals. With the exception of the following special configurations, the internal SCC and SCSI can be used as standard devices.

SCC Configuration

- IEI (Interrupt Enable In) is hardwired to VDD. Thus no external interrupt daisy-chain can be used.
- IEO (Interrupt Enable Out) is not bonded out. Since no daisy-chain interrupt is used, this pin is left unbonded.
- /TRXC and /CTS are connected together in each of the two channels to form /CTS//TRXC. In this configuration, the pin in each channel is sued as receive or transmit clock input.
- /SYNCB (channel B Synchronization) is not bonded.

SCSI Configuration

- Data lines of the SCSI are shared with the SCC's data bus (D7 through D0 on both devices). Care must be taken not to cause bus contention by inappropriately selecting the two internal devices using their respective /CS.
- /IOR of SCSI connected to /RD of SCC to generate Z85C80's /RD pin.
- /IOW of SCSI is connected to /WR of SCC to generate Z85C80's /WR pin.
- READY (Ready) is not bonded out. READY is normally used to control the speed of Block Mode DMA transfers. It goes active to indicate the SCSI is ready to send/receive data.

SCC Functional Description

The functional capabilities of the SCC are described from two different points of view: as a data communications device, it transmits and receives data in a wide variety of data communications protocols; as a microprocessor peripheral, the SCC offers valuable features such as vectored interrupts, polling, and simple handshake capability.

Data Communications Capabilities. The SCC provides two independent full-duplex channels programmable for use in any common Asynchronous or Synchronous data communication protocol. Figure 3 and the following description briefly detail these protocols.

Asynchronous Modes. Transmission and reception can be accomplished independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one and one half, or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break. Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxDA or RxDB in Figure 1). If the Low does not persist (as in the case of a transient), the character assembly process does not start.

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

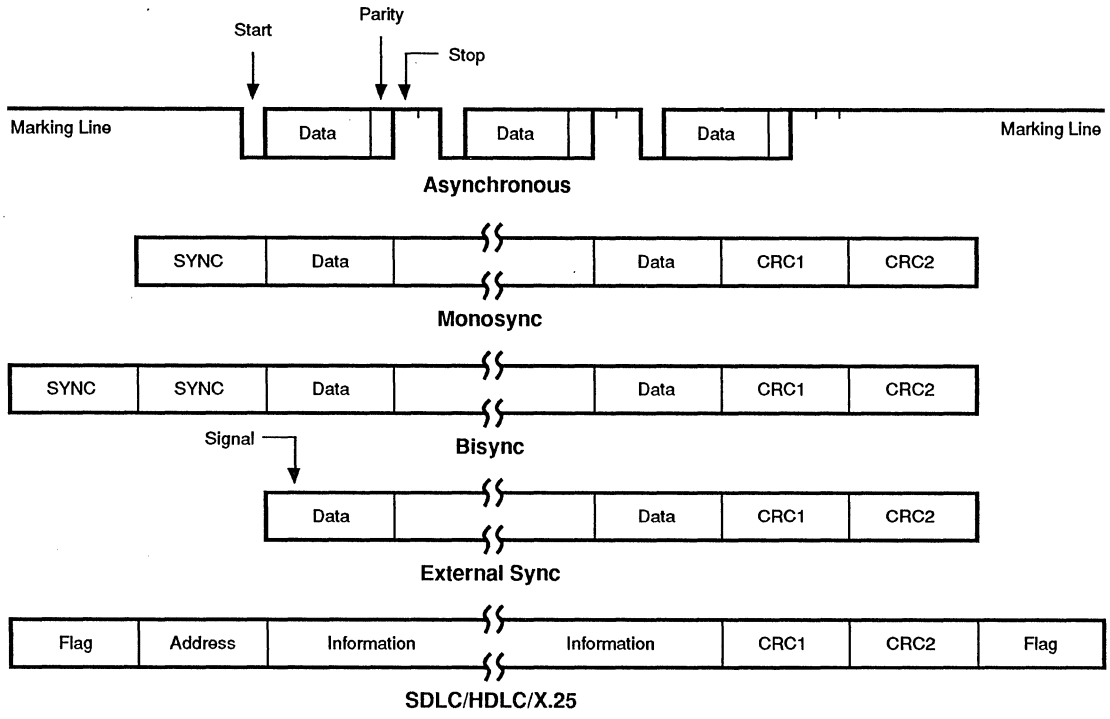


Figure 3. Some SCC Protocols

The SCC does not require symmetric transmit and receive clock signals - a feature allowing the use of a wide variety of clock sources. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32, or 1/64 of the clock rate supplied to the receive and transmit clock inputs. In Asynchronous modes, the /SYNC pin may be programmed as an input used for functions such as monitoring a ring indicator.

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

Five- or 7-bit synchronous characters are detected with 8- or 16-bit patterns in the SCC by overlapping the larger pattern across multiple incoming synchronous characters as shown in Figure 4.

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

Both CRC-16 ($X^{16} + X^{15} + X^{12} + 1$) and CCITT ($X^{16} + X^{15} + X^{12} + 1$) error checking polynomials are supported. Either polynomial may be selected in all Synchronous modes. Users may preset the CRC generator and checker to all 1s or all 0s. The SCC also provides a feature that automatically transmits CRC data when no other data is available for transmission. This allows for high speed transmission under DMA control, with no need for CPU intervention at the end of a message. When there is no data or CRC to send in Synchronous modes, the transmitter inserts 6-, 8-, or 16-bit synchronous characters, regardless of the programmed character length.

The SCC supports Synchronous bit-oriented protocols, such as SDLC and HDLC, by performing automatic flag sending, zero insertion, and CRC generation. A special command can be used to abort a frame in transmission. At

the end of a message, the SCC automatically transmits the CRC and trailing flag when the transmitter underruns. The transmitter may also be programmed to send an idle line consisting of continuous flag characters or a steady marking condition.

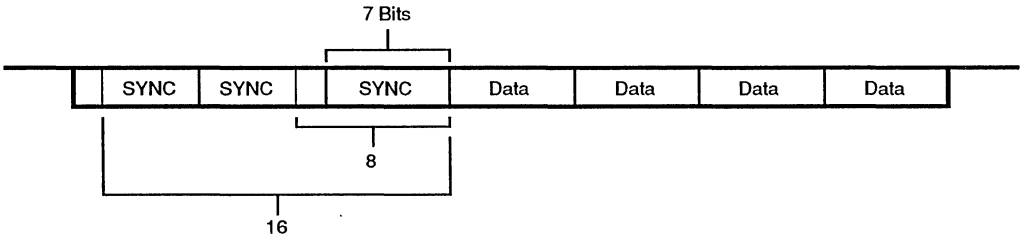


Figure 4. Detecting 5 - or 7-Bit Synchronous Characters

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

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

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

The SCC can be conveniently used under DMA control to provide high speed reception or transmission. In reception, for example, the SCC can interrupt the CPU when the first character of a message is received. The CPU then enables the DMA to transfer the message to memory. The SCC then issues an end-of-frame interrupt and the CPU can check the status of the received message. Thus, the CPU is freed for other service while the message is being received. The CPU may also enable the DMA first and have the SCC interrupt only on end-of-frame. This procedure allows all data to be transferred via the DMA.

SDLC Loop Mode. The SCC supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, there is a primary controller station that manages the message traffic flow on the loop and any number of secondary stations. In SDLC Loop mode, the SCC performs the functions of a secondary station while an SCC operating in regular SDLC mode can act as a controller (Figure 5).

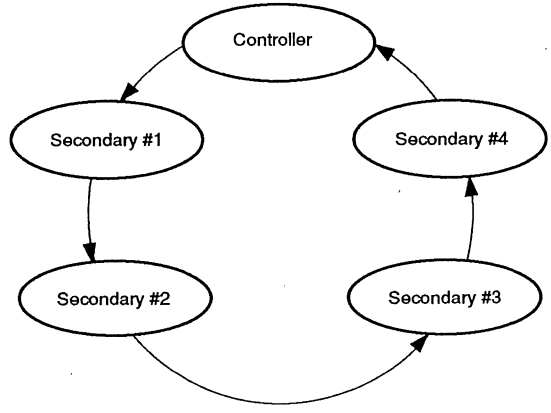


Figure 5. An SDLC Loop

FUNCTIONAL DESCRIPTION (Continued)

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

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

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

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

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

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

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

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

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

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

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

Data Encoding. The SCC may be programmed to encode and decode the serial data in four different ways (Figure 6). In NRZ encoding, a 1 is represented by a High level and a 0 is represented by a Low level. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level. In FM1 (more properly, bi-phase mark), a transition occurs at the beginning of every bit cell. A 1 is represented by an additional transition at the center of the bit cell and a 0 is represented by no additional transition at the center of the bit cell. In FM0 (bi-phase space), a transition occurs at the beginning of every bit cell. A 0 is represented by an additional transition at the center of the bit cell, and a 1 is represented by no additional transition at the center of the bit cell. In addition to these four methods, the SCC can be used to decode Manchester (bi-phase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0 to 1, the bit is a 0. If the transition is 1 to 0, the bit is a 1.

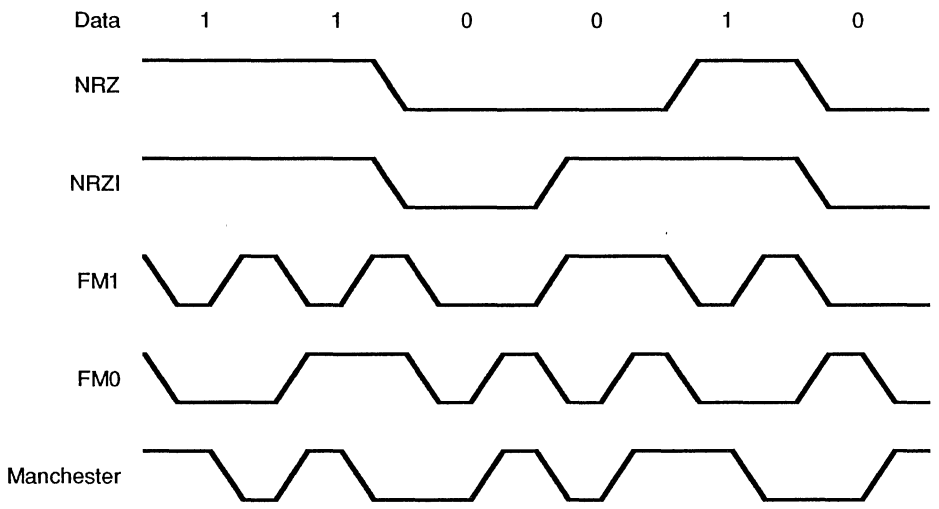


Figure 6. Data Encoding Methods

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

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

I/O Interface Capabilities. The SCC offers the choice of Polling, Interrupt (vectored or nonvectored), and Block Transfer modes to transfer data, status, and control information to and from the CPU. The Block Transfer mode can be implemented under CPU or DMA control.

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

Interrupts

When an SCC responds to an Interrupt Acknowledge signal (/INTACK) from the CPU, an interrupt vector may be placed on the data bus. This vector is written in WR2 and may be read in RR2A or RR2B - Read Register 2, Channel A, or Channel B (Figures 9 and 10).

To speed interrupt response time, the SCC can modify three bits in this vector to indicate status. If the vector is read in Channel A, status is never included; if it is read in Channel B, status is always included.

Each of the six sources of interrupts in the SCC (Transmit, Receive, and External/Status interrupts in both channels)

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

In the SCC, the IP bit signals a need for interrupt servicing. When an IP bit is 1, the /INT output is pulled Low, requesting an interrupt. In the SCC, if the IE bit is not set by enabling interrupts, then the IP for that source can never be set. The IP bits are readable in RR3A.

The IUS bits signal that an interrupt request is being serviced. There are three types of interrupts: Transmit, Receive, and External/Status. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receiver, Transmit, and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted when the transmit buffer becomes empty. (This implies that the transmitter must have had a data character written into it so that it can become empty.) When enabled, the receiver can interrupt the CPU in one of three ways.

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

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

3. Interrupt on Special Receive Condition Only.

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

The main function of the External/Status interrupt is to monitor the signal transactions of the /CTC//TRXC, /DCD, and /SYNC pins; however, an External/Status interrupt is also caused by a Transmit Underrun condition, or a zero count in the baud rate generator, or by the detection, or a zero count in the baud rate generator, or by the detection of a Break (Asynchronous mode), Abort (SDLC mode) or EOP (SDLC Loop mode) sequence in the data stream. The interrupt caused by the Abort or EOP has a special feature allowing the SCC to interrupt when the Abort condition in external logic in SDLC mode. In SDLC Loop mode, this feature allows secondary stations to recognize the wishes of the primary station to regain control of the loop during a poll sequence.

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

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

ARCHITECTURE

The SCC internal structure includes two full-duplex channels, two baud rate generators, internal control and interrupt logic, and a bus interface to a nonmultiplexed bus. Associated with each channel are a number of read and write registers for mode control and status information, as well as logic necessary to interface to modems or other external devices (Figure 7).

The logic for both channels provides formats, synchronization, and validation for data transferred to and from the channel interface. The modem control inputs are monitored by the control logic under program control. All of the modem control signals are general-purpose in nature and can optionally be used for functions other than modem control.

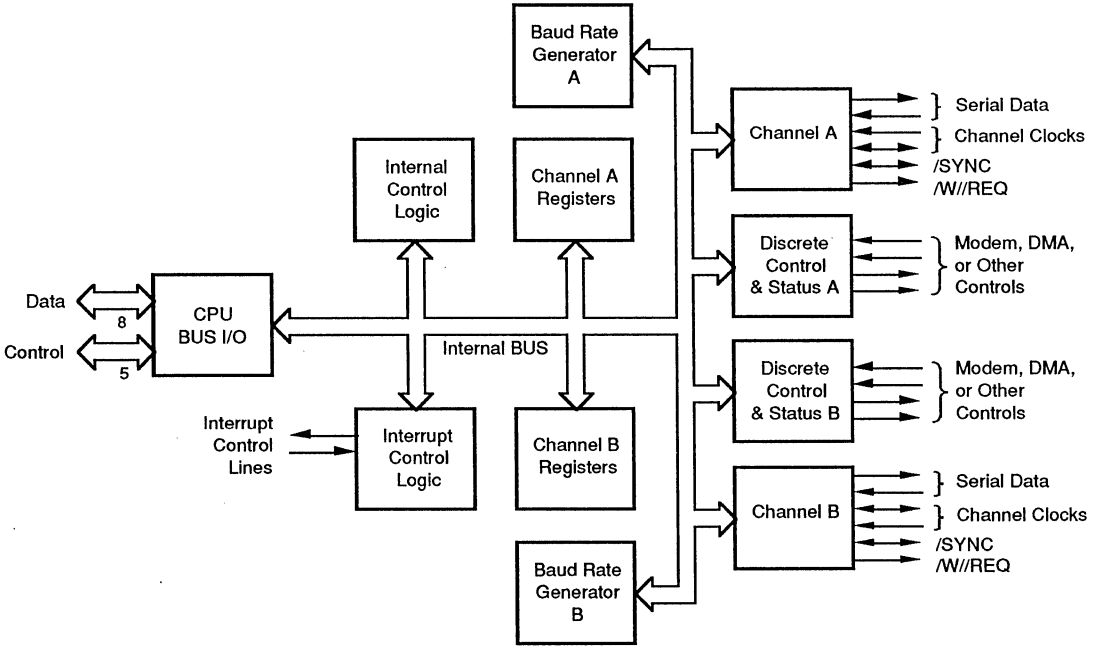


Figure 7. Block Diagram of SCC Architecture

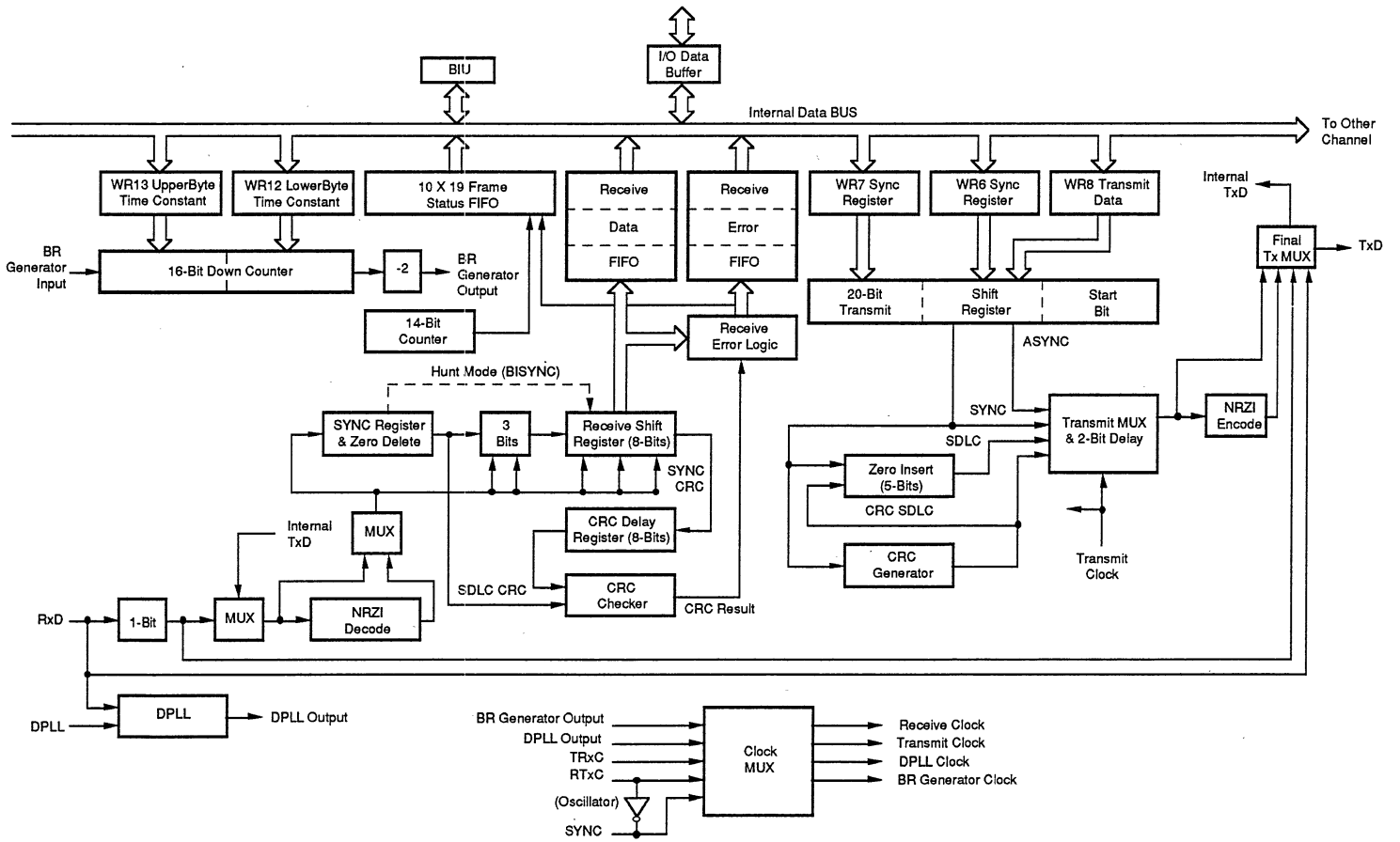


Figure 8. Data Path

ARCHITECTURE (Continued)

The register set for each channel includes ten control (write) registers, two sync-character (write) registers, and four status (read) registers. In addition, each baud rate generator has two (read/write) registers for holding the time constant that determines the baud rate. Finally, associated with the interrupt logic is a write register for the interrupt vector accessible through either channel, a write only Master Interrupt Control register and three read registers: one containing the vector with status information (Channel B only), one containing the vector without status (Channel A only), and one containing the Interrupt Pending bits (Channel A only).

The registers for each channel are designated as follows:

- WR0-WR15 - Write Registers 0 through 15.
- RR0-RR3, RR10, RR12, RR13, RR15 - Read Registers 0 through 3, 10, 12, 13, 15.

Table 1 lists the functions assigned to each read or write register. The SCC contains only one WR2 and WR9, but they can be accessed by either channel. All other registers are paired (one for each channel).

Data Path. The transmit and receive data path illustrated in Figure 8 is identical for both channels. The receiver has three 8-bit buffer registers in a FIFO arrangement, in addition to the 8-bit receive shift register. This scheme creates additional time for the CPU to service an interrupt at the beginning of a block of high speed data. Incoming data is routed through one of several paths (data or CRC) depending on the selected mode (the character length in Asynchronous modes also determines the data path).

The transmitter has an 8-bit Transmit Data buffer register loaded from the internal data bus and a 20-bit Transmit Shift register that can be loaded either from the synchronous character registers or from the Transmit Data register. Depending on the operational mode, outputting data is routed through one of four main paths before it is transmitted from the Transmit Data output (TxD).

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 9 shows the formats for each read register.

The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring; e.g., when the interrupt vector indicates a Special Receive Condition interrupt, all the appropriate error bits can be read from a single register (RR1).

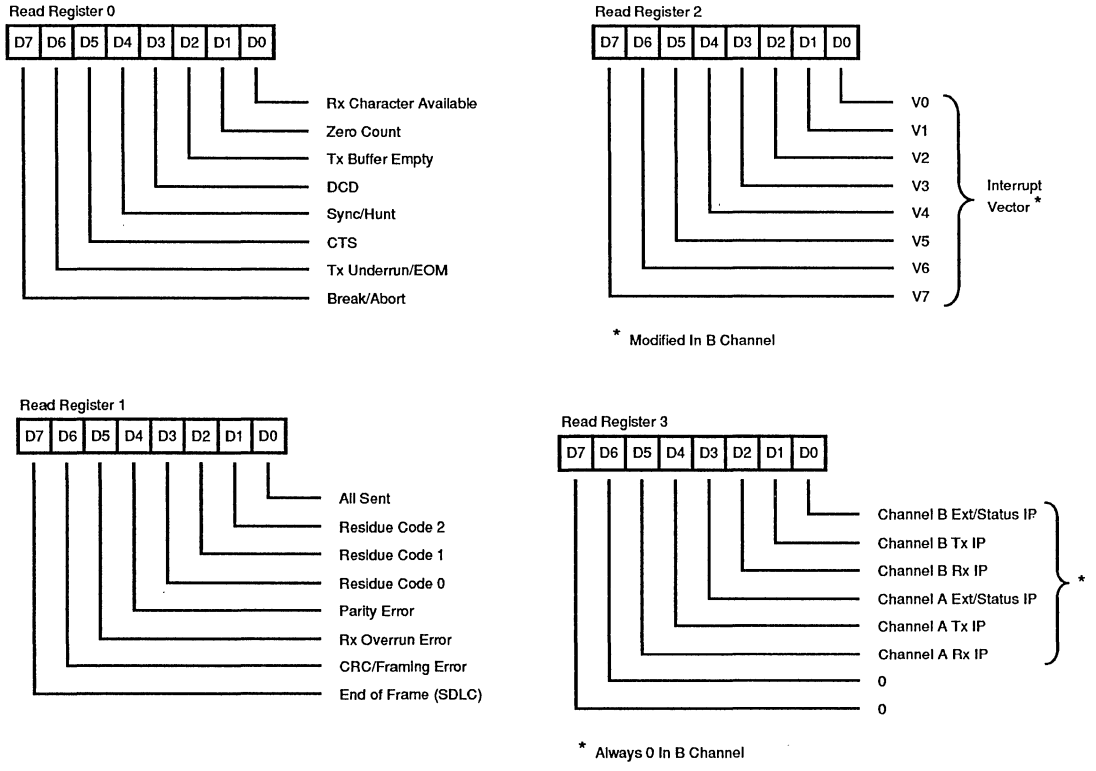


Figure 9. Read Register Bit Functions

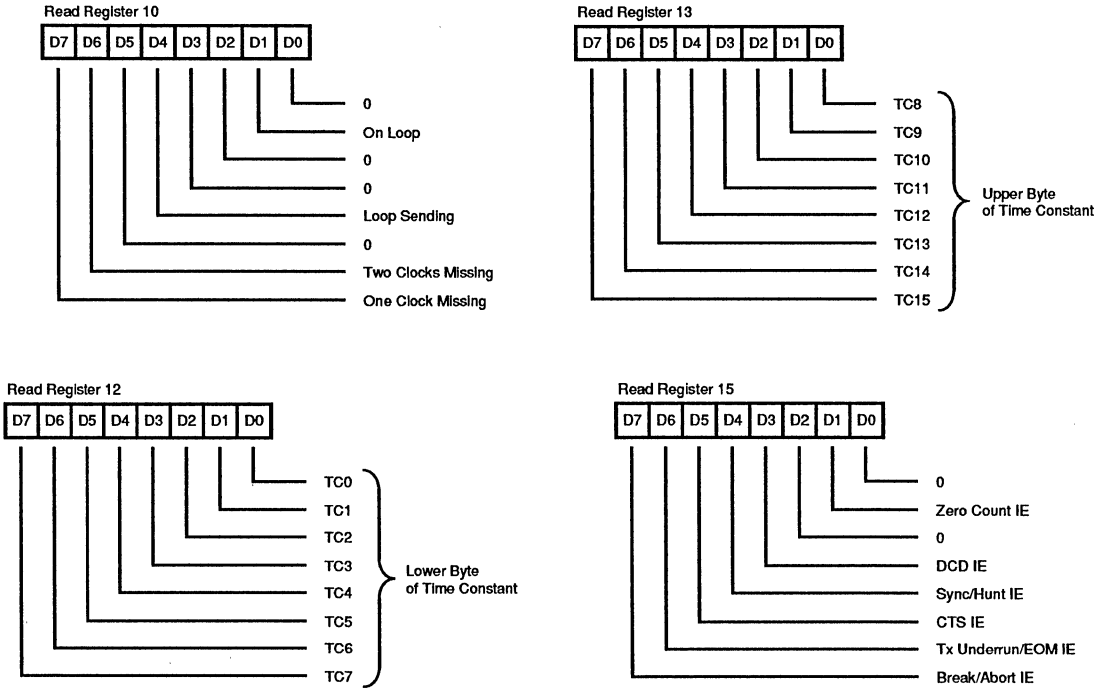


Figure 9. Read Register Bit Functions (Continued)

Write Registers. The SCC contains 13 write registers (14 counting WR8, the transmit buffer) in each channel. These write registers are programmed separately to configure the functional "personality" of the channels. In addition, there are two registers (WR2 and WR9) shared by the two channels that may be accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits. Figure 10 shows the format of each write register.

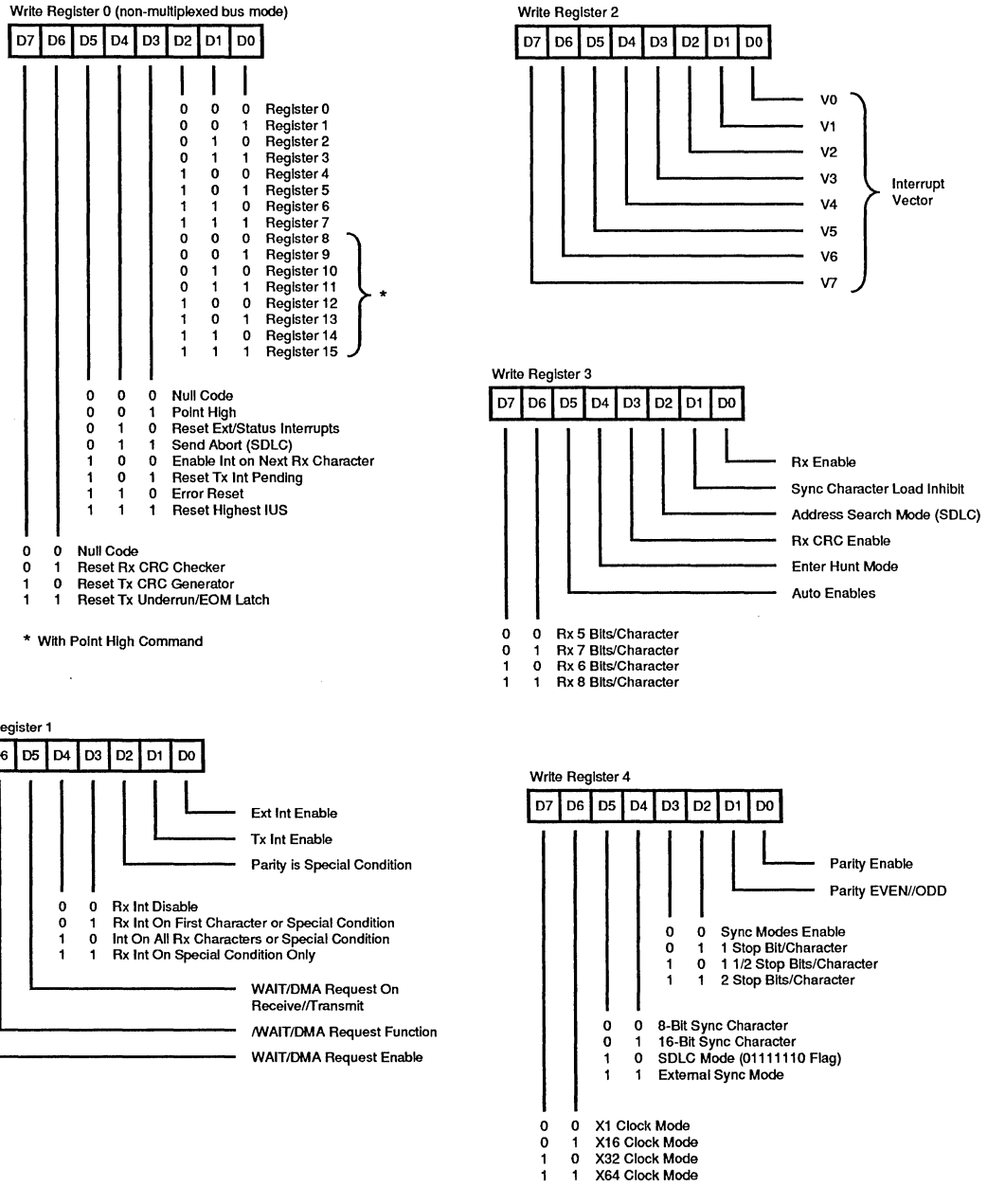
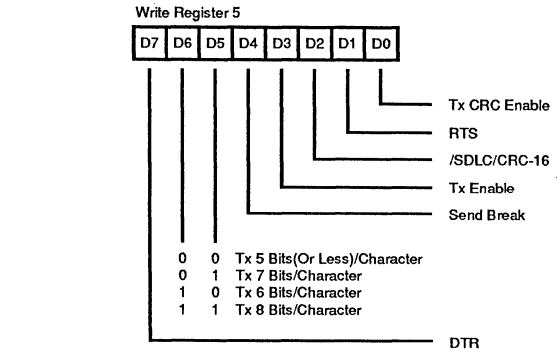
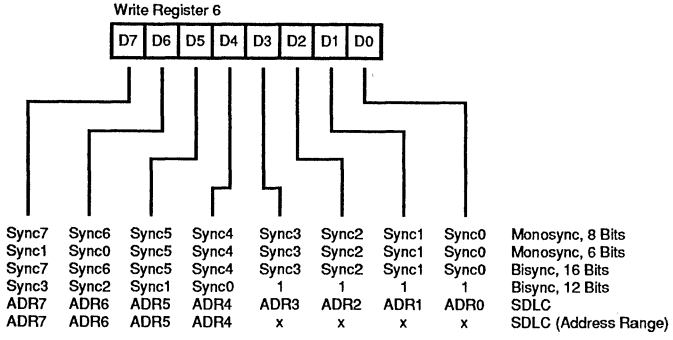


Figure 10. Write Register Bit Functions





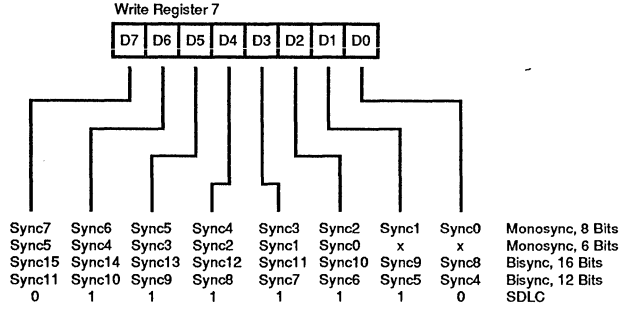


Figure 10. Write Register Bit Functions (Continued)

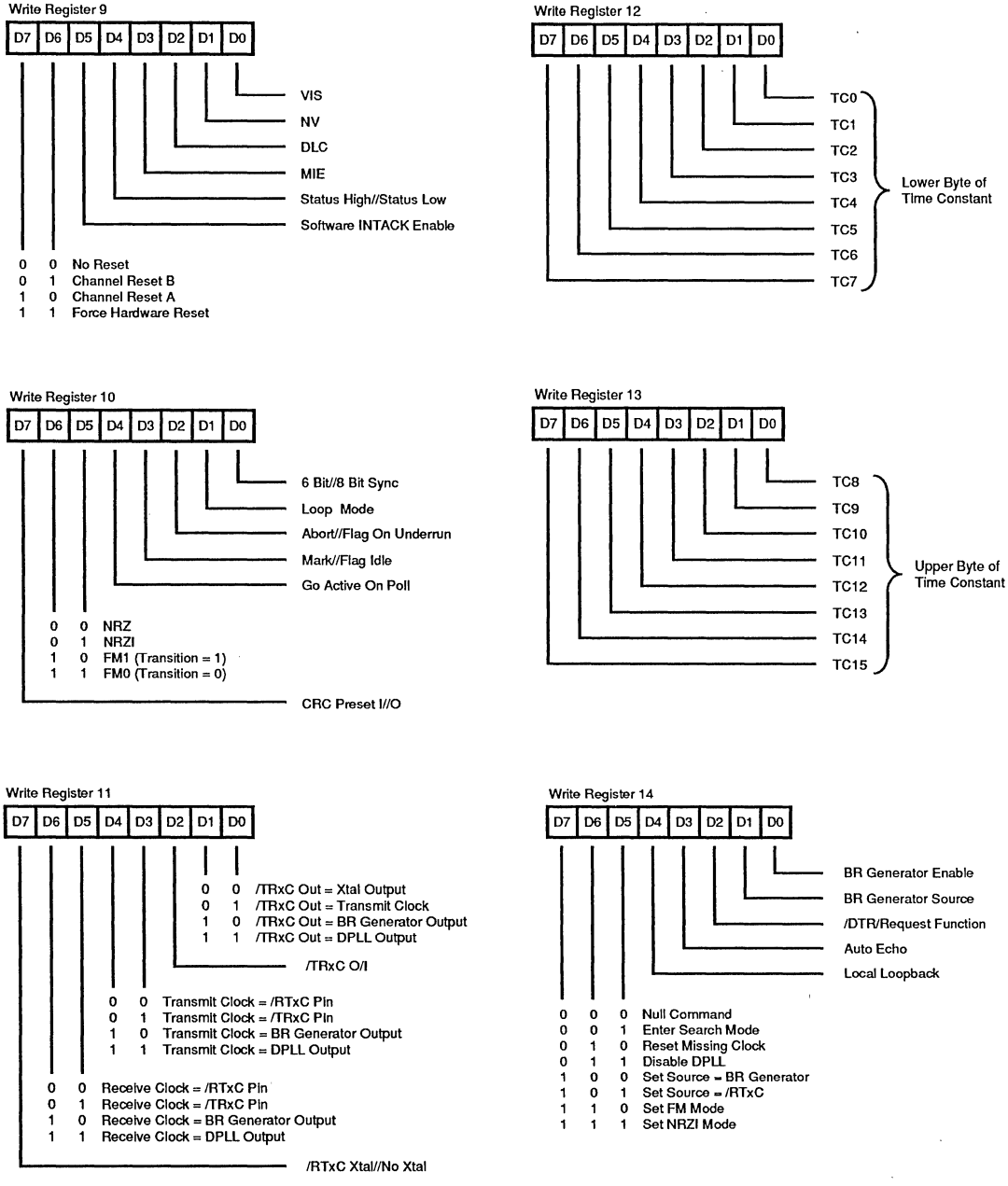


Figure 10. Write Register Bit Functions (Continued)

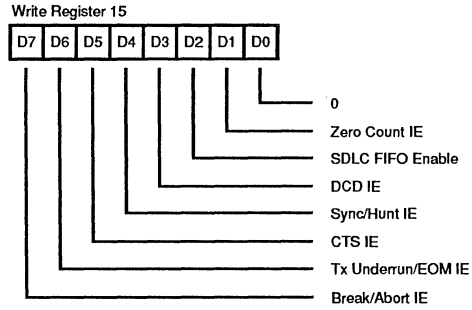


Figure 10. Write Register Bit Functions (Continued)

TIMING

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

Read Cycle Timing. Figure 11 illustrates Read cycle timing. Addresses on A//B and D//C and the status on /INTACK must remain stable throughout the cycle. If /SCCCS falls after /RD falls or if it rises before /RD rises, the effective /RD is shortened.

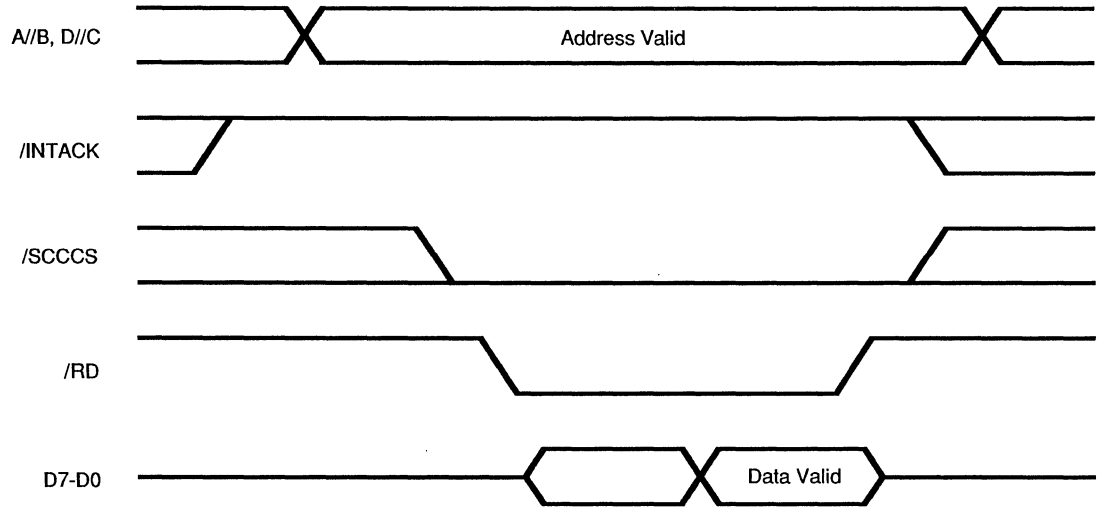


Figure 11. Read Cycle Timing

Write Cycle Timing. Figure 12 illustrates Write cycle timing. Addresses on A//B and D//C and the status on /INTACK must remain stable throughout the cycle. If /SCCCS falls after /WR falls or if it rises before /WR rises, the effective /WR is shortened. Data must be valid before the falling edge of /WR.

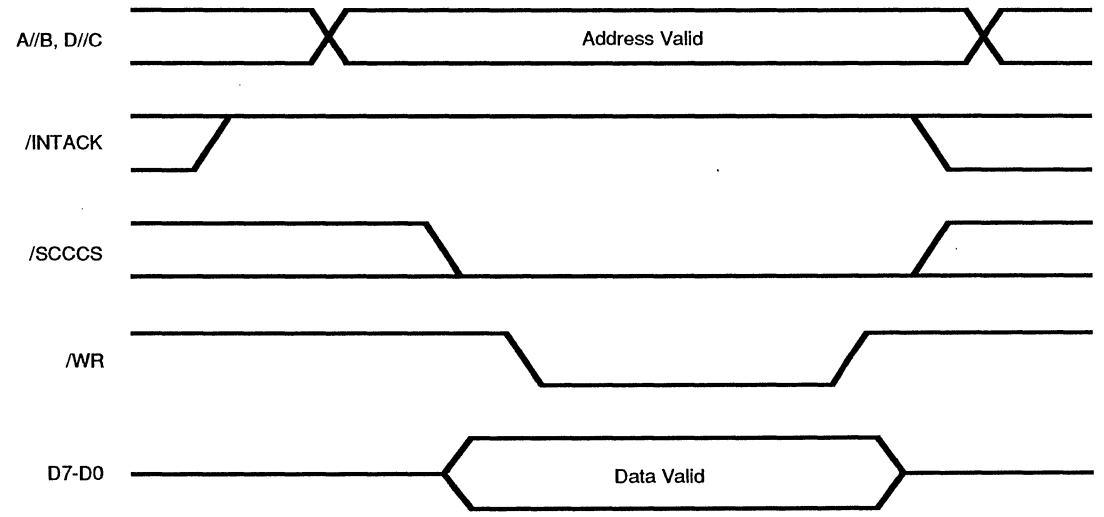


Figure 12. Write Cycle Timing

TIMING (Continued)

Interrupt Acknowledge Cycle Timing. Figure 13 illustrates Interrupt Acknowledge cycle timing.

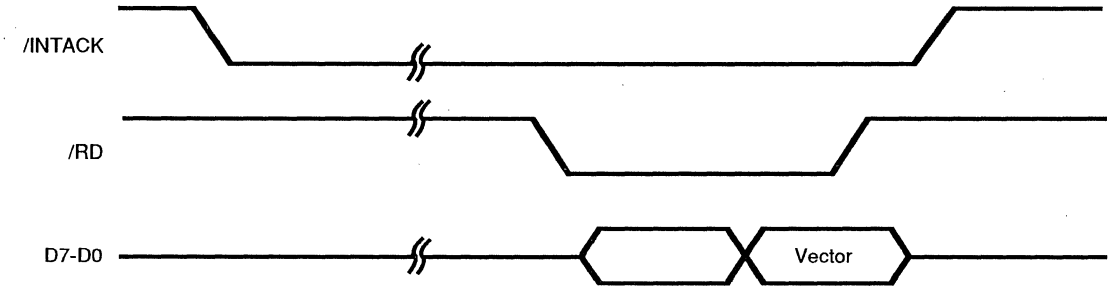


Figure 13. Interrupt Acknowledge Cycle Timing

FIFO

The following text explains the functional operations of the FIFO.

FIFO Enhancements. When used with a DMA controller, the Z85C30 FIFO enhancement maximizes the SCC's ability to receive high speed back-to-back SDLC messages while minimizing frame overruns due to CPU latencies in responding to interrupts.

Additional logic was added to the industry standard NMOS SCC consisting of a 10 deep by 19 bit status FIFO, 14-bit receive byte counter, and control logic as shown in Figure 14. The 10 x 19 bit status FIFO is separate from the existing three byte receive data FIFO.

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

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

FIFO Detail. For a better understanding of details of the FIFO operation, refer to the block diagram contained in Figure 14.

Enable/Disable. This FIFO is implemented so that it is enabled when WR15 bit 2 is set and the SCC is in the SDLC/HDLC mode, otherwise the status register contents bypass the FIFO and go directly to the bus interface (the FIFO pointer logic is reset either when disabled or via a channel or power-on reset). When the FIFO mode is disabled, the SCC is completely downward-compatible with the NMOS 8530. The FIFO mode is disabled on power-up (WR15 bit 2 is set to 0 on reset). The effects of backward compatibility on the register set are that RR4 is an image of RR0, RR5 is an image of RR1, RR6 is an image of RR2 and RR7 is an image of RR3. For the details of the added registers, refer to Figure 16. The status of the FIFO Enable signal can be obtained by reading RR15 bit 2. If the FIFO is enabled, the bit will be set to 1; otherwise, it will be reset.

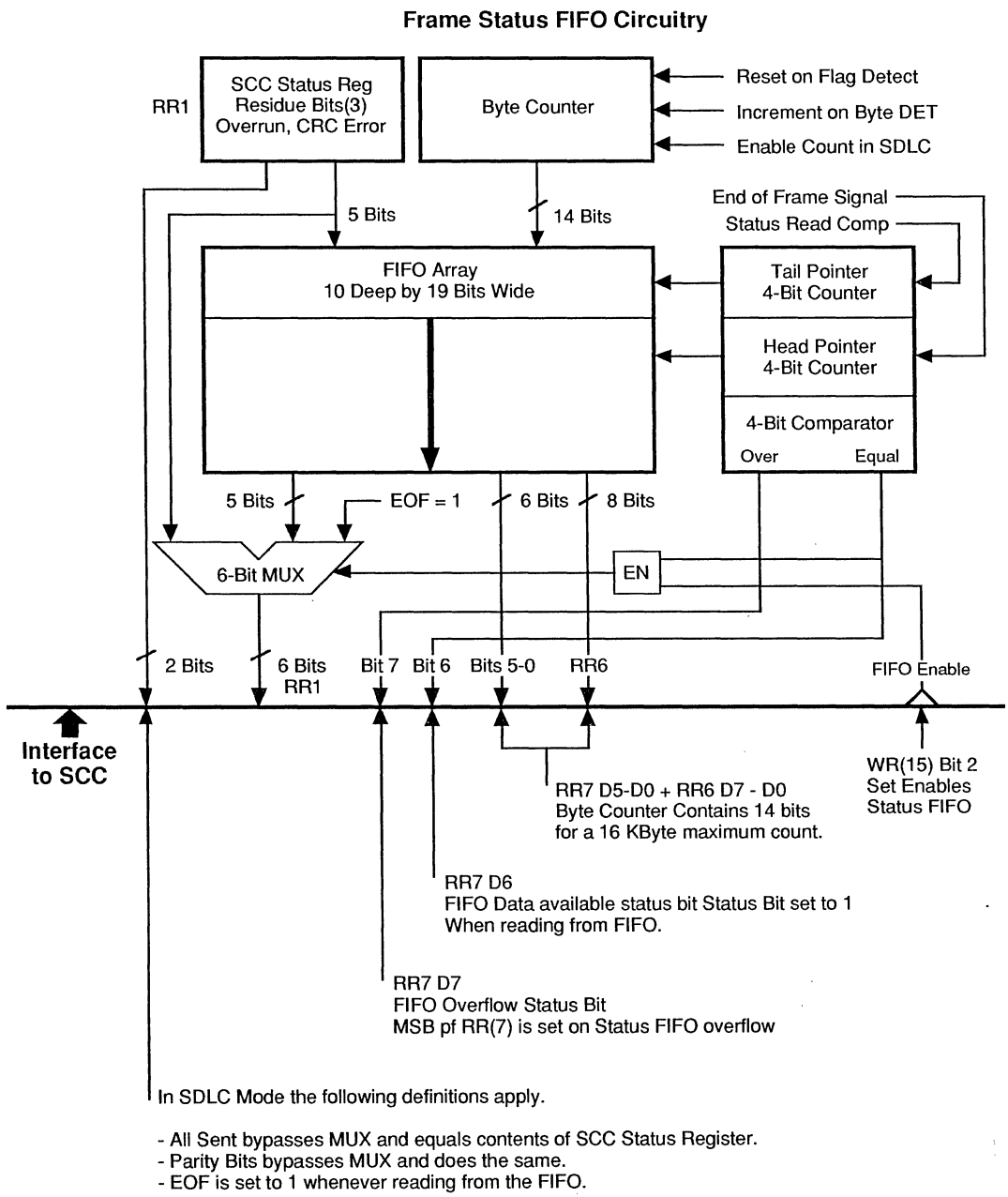


Figure 14. SCC Status Register Modifications

Read Operation. When WR15 bit 2 is set and the FIFO is not empty, the next read to any of status register RR1 or the additional registers RR7 and RR6 will actually be from the FIFO. Reading status register RR1 causes one location of the FIFO to be emptied, so status should be read after reading the byte count, otherwise the count will be incorrect. Before the FIFO underflows, it is disabled. In this case, the multiplexer is switched to allow status to read directly from the status register, and reads from RR7 and RR6 will contain bits that are undefined. Bit 6 of RR7 (FIFO Data Available) can be used to determine if status data is coming from the FIFO or directly from the status register, since it is set to 1 whenever the FIFO is not empty.

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

The sequence for proper operation of the byte count and FIFO logic is to read the registers in the following order: RR7, RR6, and RR1 (reading RR6 is optional). Additional logic prevents the FIFO from being emptied by multiple reads from RR1. The read from RR7 latches the FIFO empty/full status bit (bit 6) and steers the status multiplexer to read from the SCC megacell instead of the status FIFO (since the status FIFO is empty). The read from RR1 allows an entry to be read from the FIFO (if the FIFO was empty, logic is added to prevent a FIFO underflow condition).

Write Operation. When the end of an SDLC frame (EOF) has been received and the FIFO is enabled, the contents of the status and byte-count registers are loaded into the FIFO. The EOF signal is used to increment the FIFO. If the FIFO overflows, the MSB of RR7 (FIFO Overflow) is set to indicate the overflow. This bit and the FIFO control logic is reset by disabling and re-enabling the FIFO control bit (WR15 bit 2). For details of FIFO control timing during an SDLC frame, refer to Figure 15.

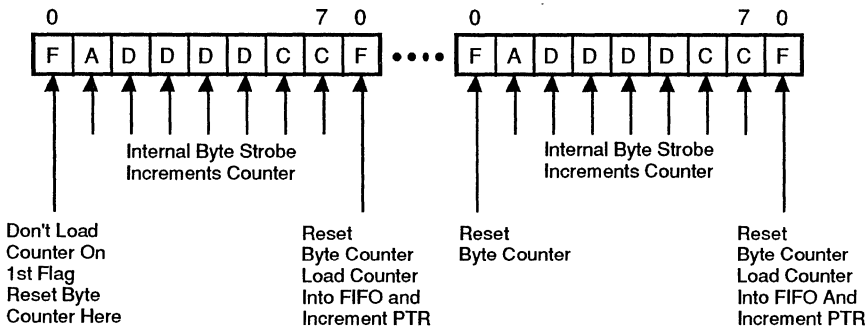


Figure 15. SDLC Byte Counting Detail

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

Enable. The byte counter is enabled in the SDLC/HDLC mode.

Reset. The byte counter is reset whenever an ADLC flag character is received. The reset is timed so that the contents of the byte counter are successfully written into the FIFO.

Increment. The byte counter is incremented by writes to the data FIFO. The counter represents the number of bytes received by the SCC, rather than the number of bytes transferred from the SCC. (These counts may differ by up to the number of bytes in the receive data FIFO contained in the SCC).

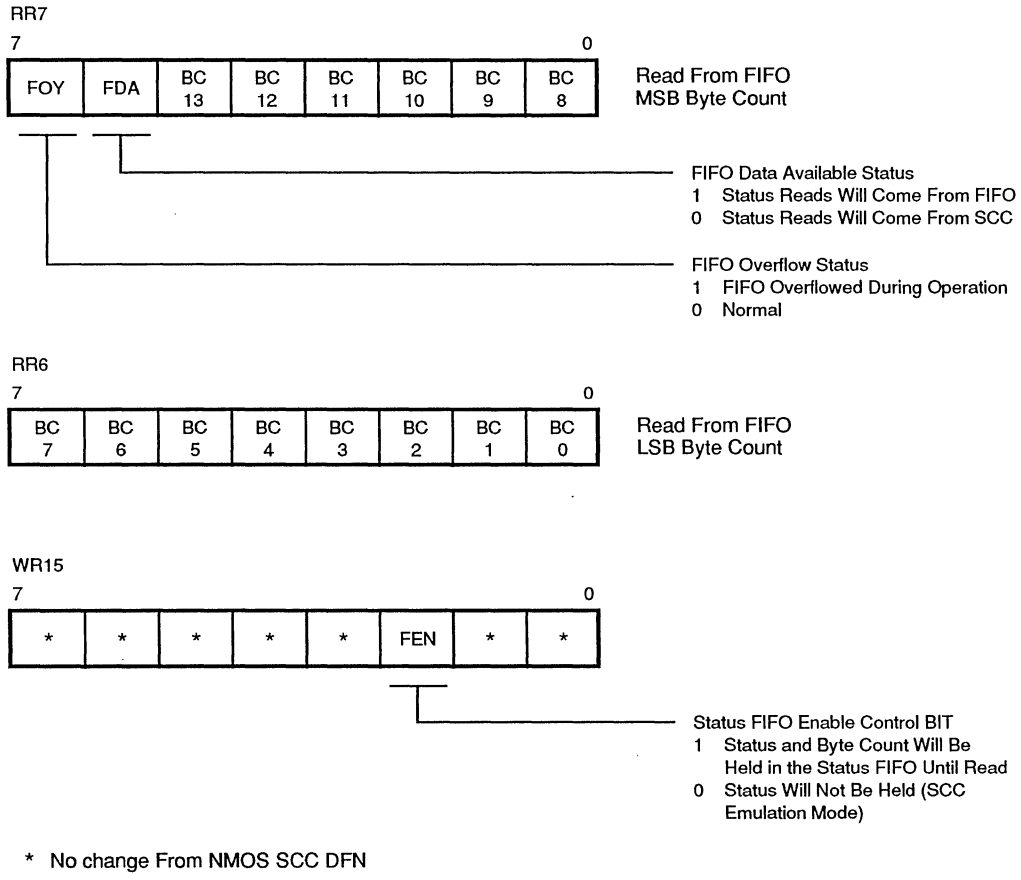


Figure 16. SCC Additional Registers

SOFTWARE INTERRUPT ACKNOWLEDGE

The SCC can do an interrupt acknowledge cycle through software. In some CPU environments it is difficult to create the /INTACK signal with the necessary timing to acknowledge interrupts and allow the nesting of interrupts. In these cases, it would be desirable to create this signal in software.

If bit 5 of Write Register 9 (WR9) is set, reading register 2 (RR2) will result in an interrupt acknowledge cycle to be executed internally. Like a hardware INTACK cycle, a software acknowledge will cause the /INT pin to return high.

Similarly to when the /INTACK signal is used, when a software acknowledge cycle is sued, a Reset Highest IUS command must be issued in the interrupt service routine. If the RR2 is read from channel B, the modified vector will be returned. If the RR2 is read from channel A, then the vector will be returned unmodified. The Vector Includes Status (VIS) and no vector (NV) bits (WR9) and are ignored when bit 5 is set to 1.

When the /INTACK is not being used, it should be pulled up to VDD through a resistor (10K ohm typical).

SCSI FUNCTIONAL DESCRIPTION

General. The Small Computer System Interface (SCSI) device has a set of eight registers that are controlled by the CPU. By reading and writing the appropriate registers, the CPU may initiate any SCSI Bus activity or may sample and assert any signal on the SCSI Bus. This allows the user to implement all or any of the SCSI protocol in software. These registers are read (written) by activating /SCSICS with an address on A2-A0 and then issuing a /RD (/WR) pulse. This section describes the operation of the internal registers (Table 2).

Table 2. Register Summary

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

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

Current SCSI Data Register. *Address 0* (Read Only). The Current SCSI Data Register (Figure 17) is a read-only register which allows the microprocessor to read the active SCSI Data Bus. This is accomplished by activating /SCSICS with an address on A2-A0 and issuing a /RD pulse. If parity checking is enabled, the SCSI Bus parity is checked at the beginning of the read cycle. This register is used during a programmed I/O data read or during Arbitration to check for higher priority arbitrating devices. Parity is not guaranteed valid during Arbitration.

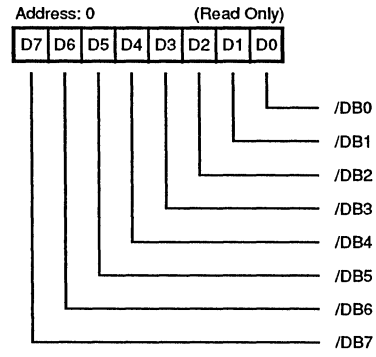


Figure 17. Current SCSI Data Register

Output Data Register. *Address 0* (Write Only). The Output Data Register (Figure 18) is a write-only register that is used to send data to the SCSI Bus. This is accomplished by either using a normal CPU write, or under DMA control, by using /WR and /DACK. This register also asserts the proper ID bits on the SCSI Bus during the Arbitration and Selection phases.

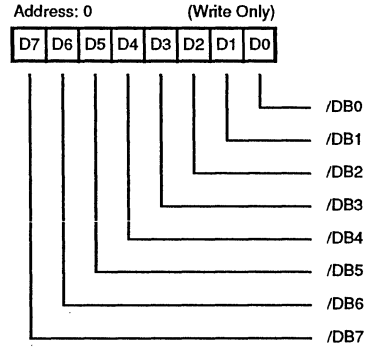


Figure 18. Output Data Register

Input Data Register. *Address 6* (Read Only). The input Data Register (Figure 19) is a read-only register that is used to read latched data from the SCSI Bus. Data is latched either during a DMA Target receive operation

when /ACK goes active or during a DMA Initiator receive when /REQ goes active. The DMA Mode bit (Mode Register bit 1) must be set before data can be latched in the Input Data Register. This register is read under DMA control using /RD and /DACK. Parity is optionally checked when the Input Data Register is loaded.

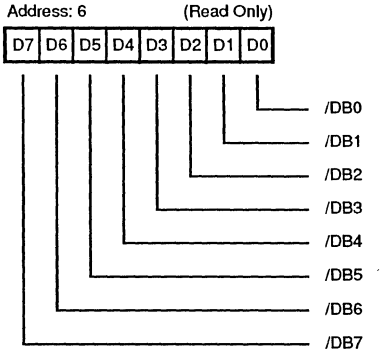


Figure 19. Input Data Register

Initiator Command Register. *Address 1* (read/write). The Initiator Command Register (Figures 20 and 21) are read and write registers which assert certain SCSI Bus signals, monitors those signals, and monitors the progress of bus arbitration. Many of these bits are significant only when being used as an Initiator; however, most can be used during Target role operation.

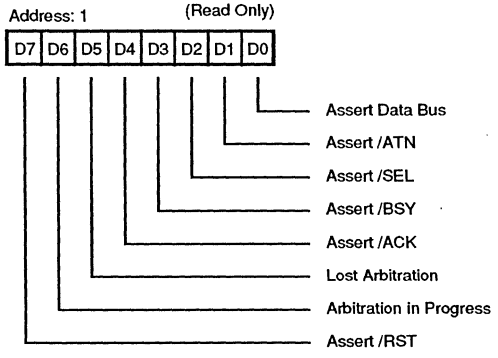


Figure 20. Initiator Command Register (Register Read)

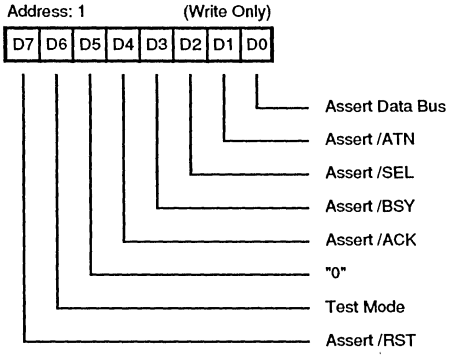


Figure 21. Initiator Command Register (Register Write)

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

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

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

This bit should also be set during DMA send operations.

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

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

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

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

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

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

Bit 6. *TEST MODE* (Write Bit). Bit 6 is written during a test environment to disable all output drivers, effectively removing the Z53C80 from the circuit. Resetting this bit returns the part to normal operation.

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

Bit 7. *ASSERT/RST.* Whenever a one is written to bit 7 of the Initiator Command Register, the /RST signal is asserted on the SCSI Bus. The /RST signal will remain asserted until this bit is reset or until an external /RESET occurs. After this bit is set (1), IRQ goes active and all internal logic and control registers are reset (except for the interrupt latch and the ASSERT/RST bit). Writing a zero to bit 7 of the Initiator Command Register deasserts the /RST signal. The status of this bit is monitored by reading the Initiator Command Register.

Mode Register. *Address 2* (Read/Write). The Mode Register controls the operation of the chip. This register determines whether the SCSI operates as an Initiator or a Target, whether DMA transfers are being used, whether parity is checked, and whether interrupts are generated on various external conditions. This register is read to check the value of these internal control bits (Figure 22).

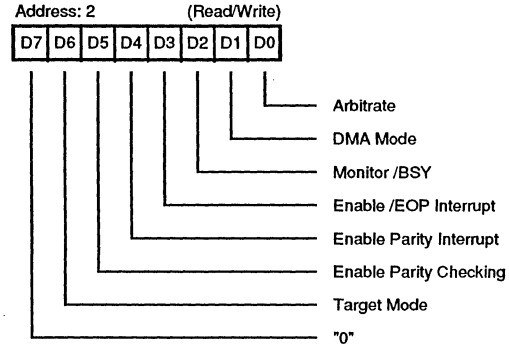


Figure 22. Mode Register

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

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

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

Bit 2. *MONITOR BUSY.* The MONITOR BUSY bit, when TRUE (1), causes an interrupt to be generated for an

unexpected loss of /BSY. When the interrupt is generated due to loss of /BSY, the lower six bits of the Initiator Command Register are reset (0) and all signals are removed from the SCSI Bus.

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

Bit 4. *ENABLE PARITY INTERRUPT.* Tthe ENABLE PARITY INTERRUPT bit, when set (1), will cause an interrupt (IRQ) to occur if a parity error is detected. A parity interrupt will only be generated if the ENABLE PARITY CHECKING bit (bit 5) is also enabled (1).

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

Bit 6. *TARGETMODE.* The TARGETMODE bit allows the SCSI to operate as either a SCSI Bus Initiator, bit reset (0), or as a SCSI Bus Target device, bit set (1). If the signals /ATN and /ACK are to be asserted on the SCSI Bus, the TARGETMODE bit must be reset (0). If the signals C//D, I//O, /MSG, and /REQ are to be asserted on the SCSI Bus, the TARGETMODE bit must be set (1).

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

Target Command Register. *Address 3* (Read/Write). When connected as a target device, the Target Command Register (Figure 23) allows the CPU to control the SCSI Bus Information Transfer phase and/or to assert /REQ by writing this register. The TARGETMODE bit (Mode Register, bit 6) must be TRUE (1) for bus assertion to occur. The SCSI Bus phases are described in Table 3.

Table 3. SCSI Information Transfer Phase

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

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

Bits 4, 5, and 6 are not used.

Bit 7. *LAST BYTE SENT* (Read Only). The END OF DMA TRANSFER bit (Bus and Status Register, bit 7) only indicates when the last byte was received from the DMA controller. The LAST BYTE SENT bit can be used to flag that the last byte of the DMA send operation has been transferred on the SCSI Data Bus.

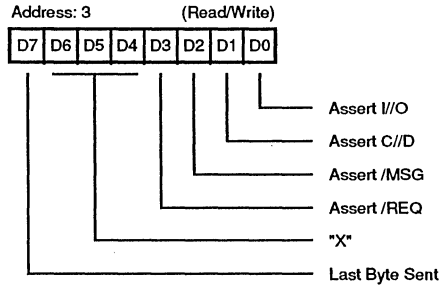


Figure 23. Target Command Register

Current SCSI Bus Status Register. *Address 4* (Read Only). The Current SCSI Bus Register is a read-only register which is used to monitor seven SCSI Bus control signals, plus the Data Bus parity bit. For example, an Initiator device can use this register to determine the current bus phase and to poll /REQ for pending data transfers. This register may also be used to determine why a particular interrupt occurred. Figure 24 describes the Current SCSI Bus Status Register.

Select Enable Register. *Address 4* (Write Only). The Select Enable Register (Figure 25) is a write-only register which is used as a mask to monitor a signal ID during a selection attempt. The simultaneous occurrence of the correct ID bit, /BSY FALSE, and /SEL TRUE will cause an interrupt. This interrupt can be disabled by resetting all bits in this register. If the ENABLE PARITY CHECKING bit (Mode Register, bit 5) is active (1), parity is checked during selection.

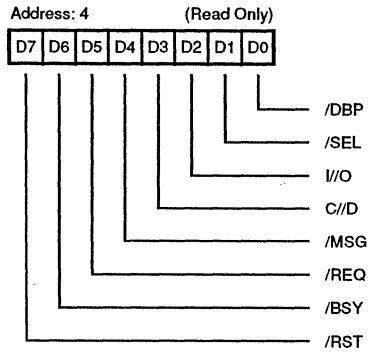


Figure 24. Current SCSI Bus Status Register

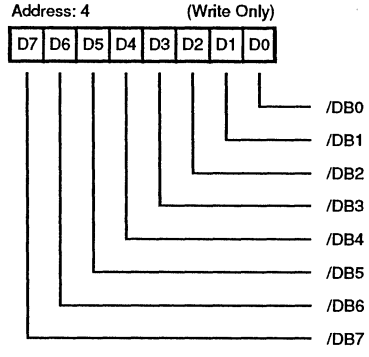


Figure 25. Select Enable Register

Bus and Status Register. *Address 5* (Read Only). The Bus and Status Register (Figure 26) is a read-only register which can be used to monitor the remaining SCSI control signals not found in the Current SCSI Bus Status Registers (/ATN and /ACK), as well as six other status bits. The following describes each bit of the Bus Status Register individually.

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

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

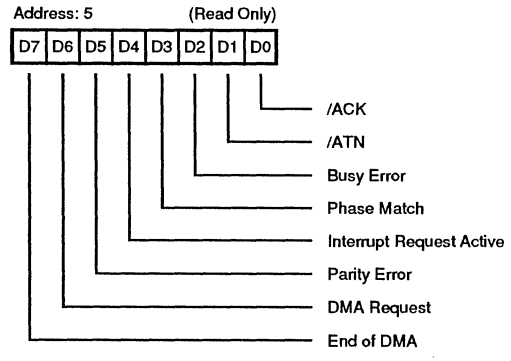


Figure 26. Bus and Status Register

Bit 2. *BUSY ERROR.* The BUSY ERROR bit is active if an unexpected loss of the /BSY signal has occurred. This latch is set whenever the MONITOR BUSY bit (Mode Register, bit 2) is TRUE and /BSY is FALSE. An unexpected loss of /BSY disables any SCSI outputs and resets the DMA MODE bit (Mode Register, bit 1).

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

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

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

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

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

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

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

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

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

Reset Parity/Interrupt. *Address 7* (Read Only). Reading this register resets the PARITY ERROR bit (bit 5), the INTERRUPT REQUEST bit (bit 4), and the BUSY ERROR bit (bit 2) in the Bus and Status Register.

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

Arbitration is accomplished using a Bus-Free filter to continuously monitor /BSY. If /BSY remains inactive for at least 400ns, the SCSI is considered free and Arbitration may begin. Arbitration will begin if the bus is free, /SEL is inactive, and the ARBITRATE bit (Mode Register, bit 0) is active. Once arbitration has begun (/BSY asserted), an arbitration delay of 2.2 µs must elapse before the Data Bus can be examined to determine if Arbitration is enabled. This delay is implemented in the controlling software driver.

The Z53C80 is a clockwise device. Delays such as bus-free delay, bus-set delay, and bus-settle delay are implemented using gate delays. These delays may differ between devices because of inherent process variations, but are well within the proposed ANSI X3.131 - 1986 specification.

INTERRUPTS. The Z53C80 provides an interrupt output (IRQ) to indicate a task completion or an abnormal bus occurrence. The use of interrupts is optional and may be disabled by resetting the appropriate bits in the Mode Register or the Select Enable Register.

When an interrupt occurs, the Bus and Status Register and the Current SCSI Bus Status Register (Figures 26 and 24) must be read to determine which condition created the interrupt. IRQ can be reset simply by reading the Reset Parity/Interrupt Register or by an external chip reset /RESET active for 100ns.

Assuming the Z53C80 has been properly initialized, an interrupt will be generated if the chip is selected or reselected, if an /EOP signal occurs, if a parity error occurs during a data transfer, if a bus phase mismatch occurs, or if a SCSI Bus disconnection occurs.

Selection Reselection. The Z53C80 generates a select interrupt if SEL is active (0), its device ID is TRUE and /BSY is FALSE for at least a bus-settle delay. If I//O is active, this is considered a reselect interrupt. The correct ID bit is determined by a match in the Select Enable Register. Only a single bit match is required to generate an interrupt. This interrupt may be disabled by writing zeros into all bits of the Select Enable Register.

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

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

The proper values for the Bus and Status Register and the Current SCSI Bus Status Register are displayed in Figures 27 and 28, respectively.

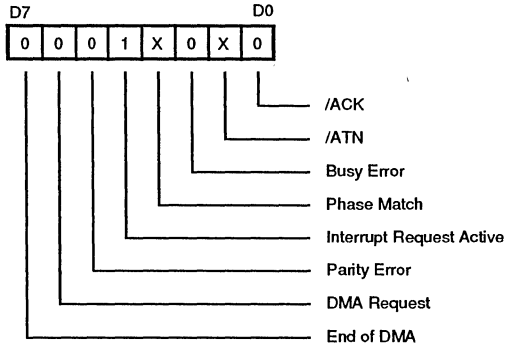


Figure 27. Bus and Status Register

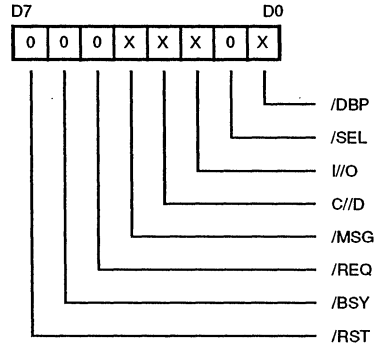


Figure 28. Current SCSI Bus Status Register

End of Process (EOP) Interrupt. An End Of Process signal (EOP) which occurs during a DMA transfer (DMA MODE TRUE) will set the END OF DMA Status bit (Bus and Status Register bit 7) and will optionally generate an interrupt if ENABLE EOP INTERRUPT bit (Mode Register, bit 3) is TRUE. The /EOP pulse will not be recognized (END OF DMA bit set) unless /EOP, /DACK, and either /RD or /WR are concurrently active for at least 50 ns. DMA transfers can still occur if /EOP was not asserted at the correct time. This interrupt is disabled by resetting the ENABLE EOP INTERRUPT bit.

The proper values for the Bus and Status Register and the Current SCSI Bus Status Register for this interrupt are shown in Figures 29 and 30.

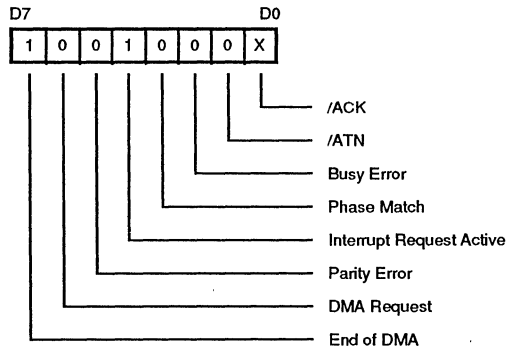


Figure 29. Bus and Status Register

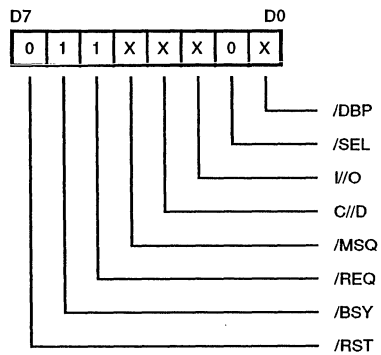


Figure 30. Current SCSI Bus Status Register

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

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

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

The proper values for the Bus and Status Register and the Current SCSI Bus Status Register are displayed in Figures 31 and 32, respectively.

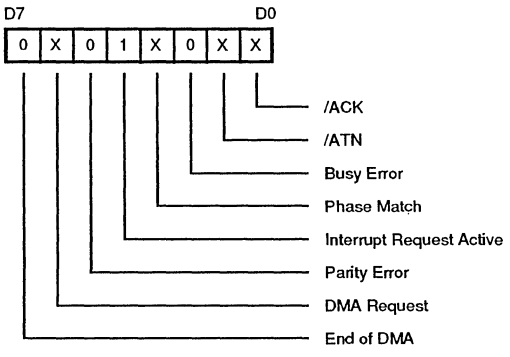


Figure 31. Bus and Status Register

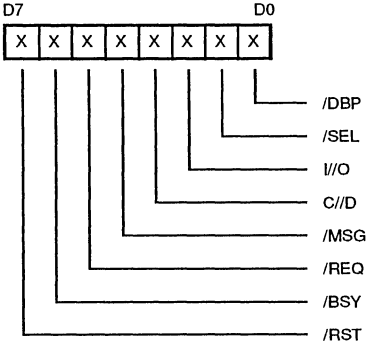


Figure 32. Current SCSI Bus Status Register

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

The proper values for the Bus and Status Register and the Current SCSI Bus Status Register are displayed in Figures 33 and 34, respectively.

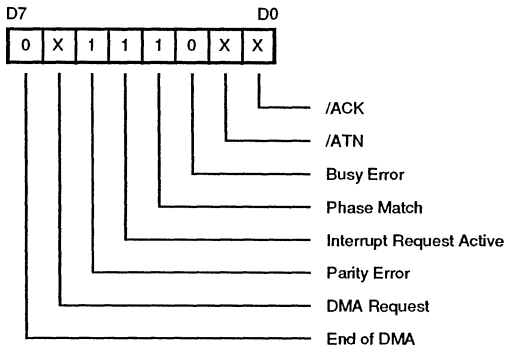


Figure 33. Bus and Status Register

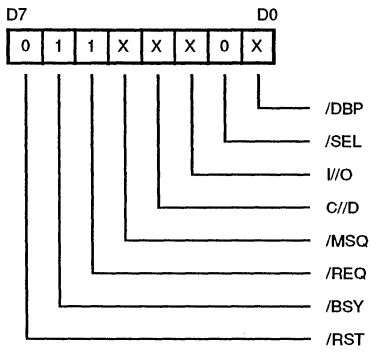


Figure 34. Current SCSI Bus Status Register

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

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

The proper values for the Bus and Status Register and the Current SCSI Bus Status Register are displayed in Figures 35 and 36, respectively.

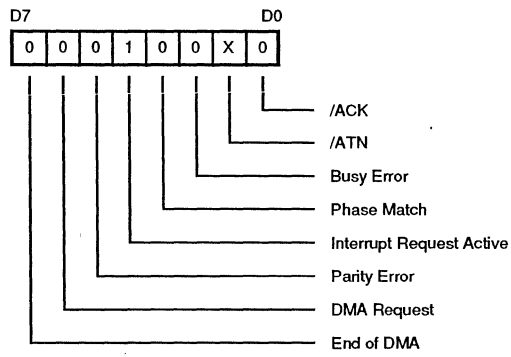


Figure 35. Bus and Status Register

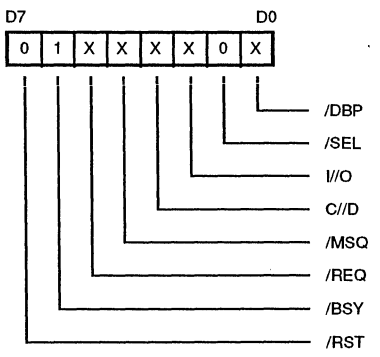


Figure 36. Current SCSI Bus Status Register

Loss of BSY. If the MONITOR BUSY bit (bit 2) in the Mode Register is active, an interrupt is generated if the BSY signal goes FALSE for at least a bus-settle delay. This interrupt is disabled by resetting the MONITOR BUSY bit. Register values are displayed in Figures 37 and 38.

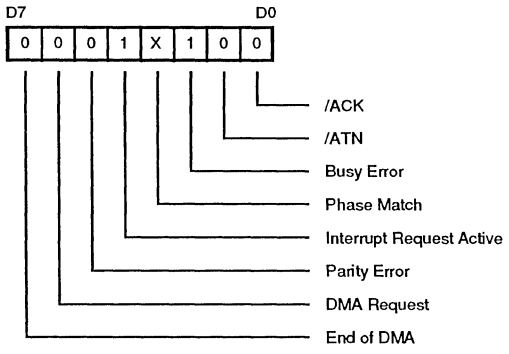


Figure 37. Bus and Status Register

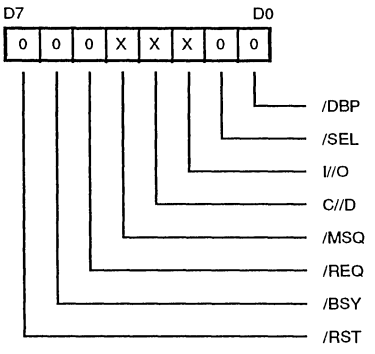


Figure 38. Current SCSI Bus Status Register

Reset Conditions. Three possible reset situations exist with the Z85C80, as follows:

Hardware Chip Reset. When the signal RST is active for at least 100 ns, the Z53C80 device is re-initialized and all internal logic and control registers are cleared. This is a chip reset only and does not create a SCSI Bus-Reset condition.

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

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

Data Transfers. Data is transferred between SCSI Bus devices in one of four modes: 1) Programmed I/O, 2) Normal DMA, 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.

Pseudo DMA Mode. To avoid the tedium of monitoring and asserting the request/acknowledgement handshake signals for programmed I/O transfers, the system may be designed to implement a pseudo DMA mode. This mode is implemented by programming the Z53C80 to operate in the DMA mode, but using the CPU to emulate the DMA handshake. DRQ may be detected by polling the DMA REQUEST bit (bit 6) in the Bus and Status Register, by sampling the signal through an external port, or by using it to generate a CPU interrupt. Once DRQ is detected, the CPU can perform a read or write data transfer. This CPU read/write is externally decoded to generate the appropriate /DACK and /RD or /WR signals.

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

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

Using the /EOP Signal. If /EOP is used, it should be asserted for at least 50 ns while /DACK and /RD or /WR are simultaneously active. Note, however, that if /RD or /WR is not active, an interrupt is generated, but the DMA activity continues. The /EOP signal does not reset the DMA MODE bit. Since the /EOP signal can occur during the last byte sent to the Output Data Register, the /REQ and /ACK signals are monitored to ensure that the last byte has transferred.

Bus Phase Mismatch Interrupt. A bus phase mismatch interrupt is used to halt the transfer if operating as an Initiator. Using this method frees the host from maintaining a data length counter and frees the DMA logic from providing the /EOP signal. If performing an Initiator send operation, the Z53C80 requires /DACK to cycle before /ACK goes inactive. Since phase changes cannot occur if /ACK is active, either /DACK must be cycled after the last byte is sent or the DMA MODE bit must be reset in order to receive the phase mismatch interrupt.

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

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

READ REGISTERS

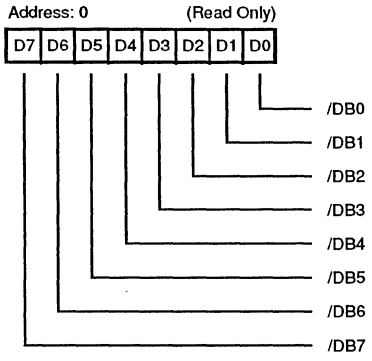


Figure 39. Current SCSI Data Register

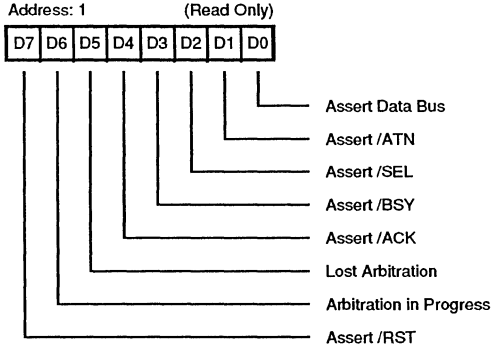


Figure 40. Initiator Command Register

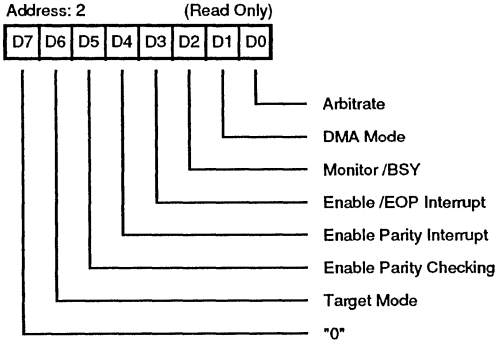


Figure 41. Mode Register

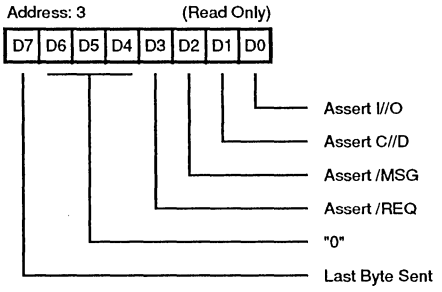


Figure 42. Target Command Register

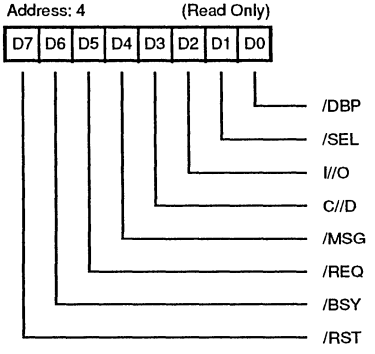


Figure 43. Current SCSI Bus Status Register

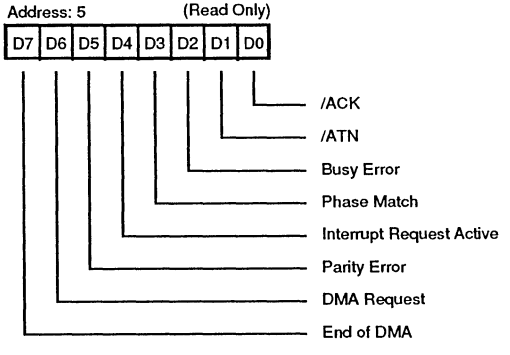


Figure 44. Bus and Status Register

READ REGISTERS (Continued)

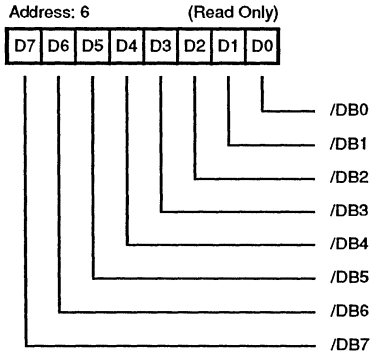


Figure 45. Input Data Register

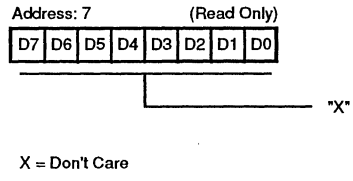


Figure 46. Reset Parity/Interrupt

WRITE REGISTERS

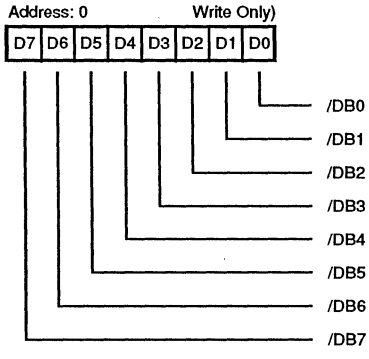


Figure 47. Output Data Register

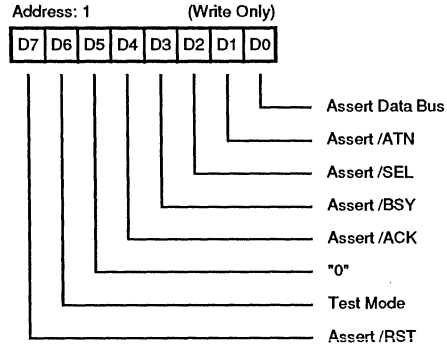


Figure 48. Initiator Command Register

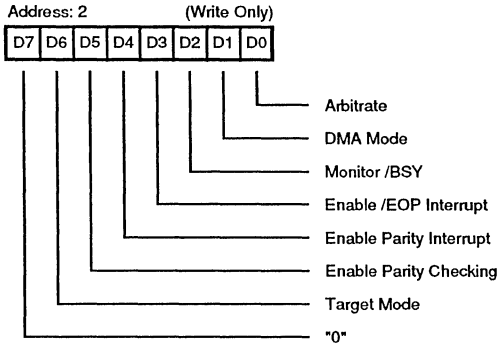


Figure 49. Mode Register

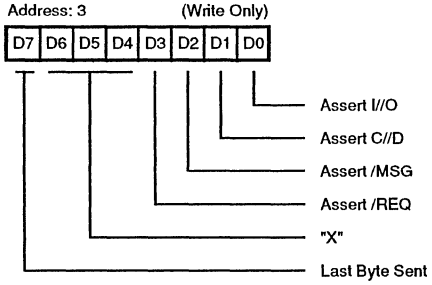


Figure 50. Target Command Register

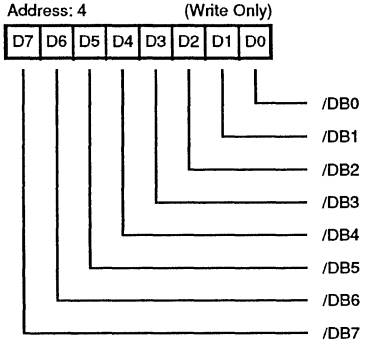


Figure 51. Select Enable Register

Note: X = Don't care

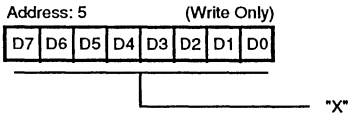


Figure 52. Start DMA Send

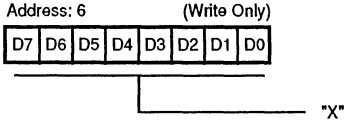


Figure 53. Start DMA Target Receive

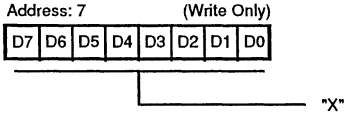


Figure 54. Start DMA Initiator Receive

ABSOLUTE MAXIMUM RATINGS

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

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

STANDARD TEST CONDITIONS

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

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

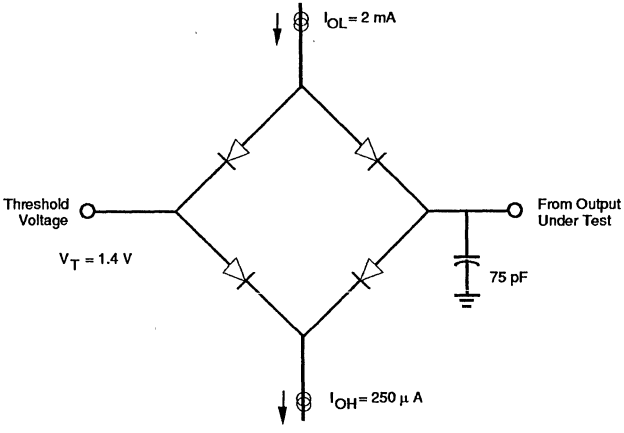


Figure 55. Standard Test Dynamic Load Circuit

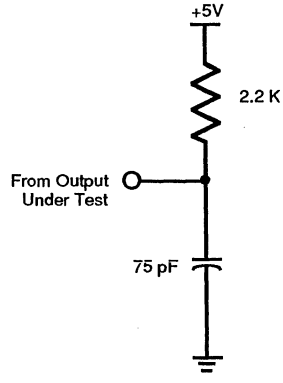


Figure 56. Open-Drain Test Load

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
$I_{IH}1$	High-Level Input Current	V_{IH} = 5.5V			
	SCSI Bus Pins	V_{IL} = 0V		50	µA
$I_{IH}2$	High-Level Input Current	V_{IH} = 5.5V			
	All Other Pins	V_{IL} = 0V		10	µA
$I_{IL}1$	Low-Level Input Current	V_{IH} = 5.5V			
	SCSI Bus Pins	V_{IL} = 0V		-50	µA
$I_{IL}2$	Low-Level Input Current	V_{IH} = 5.5V			
	All Other Pins	V_{IL} = 0V		-10	µA
$V_{OH}1$	High-Level Output Voltage	I_{OH} = -3mA	2.4		
$V_{OH}2$	High-Level Output Voltage	I_{OH} = -250 µA	V_{DD} -0.8	V	
$V_{OL}1$	Low-Level Output Voltage				
	SCSI Bus Pins	I_{OL} = 48 mA		0.5	V
$V_{OL}2$	Low-Level Output Voltage				
	All Other Pins	I_{OL} = 7 mA		0.5	V
I_{DD}	Supply Current			40	mA
C_{IN}	Input Capacitance			10	pf
C_{OUT}	Output Capacitance			15	pf
$C_{I/O}$	Bidirectional Capacitance			20	pf
T_A	Operating Free-Air Temperature		0	70	°C

AC CHARACTERISTICS
General Timing

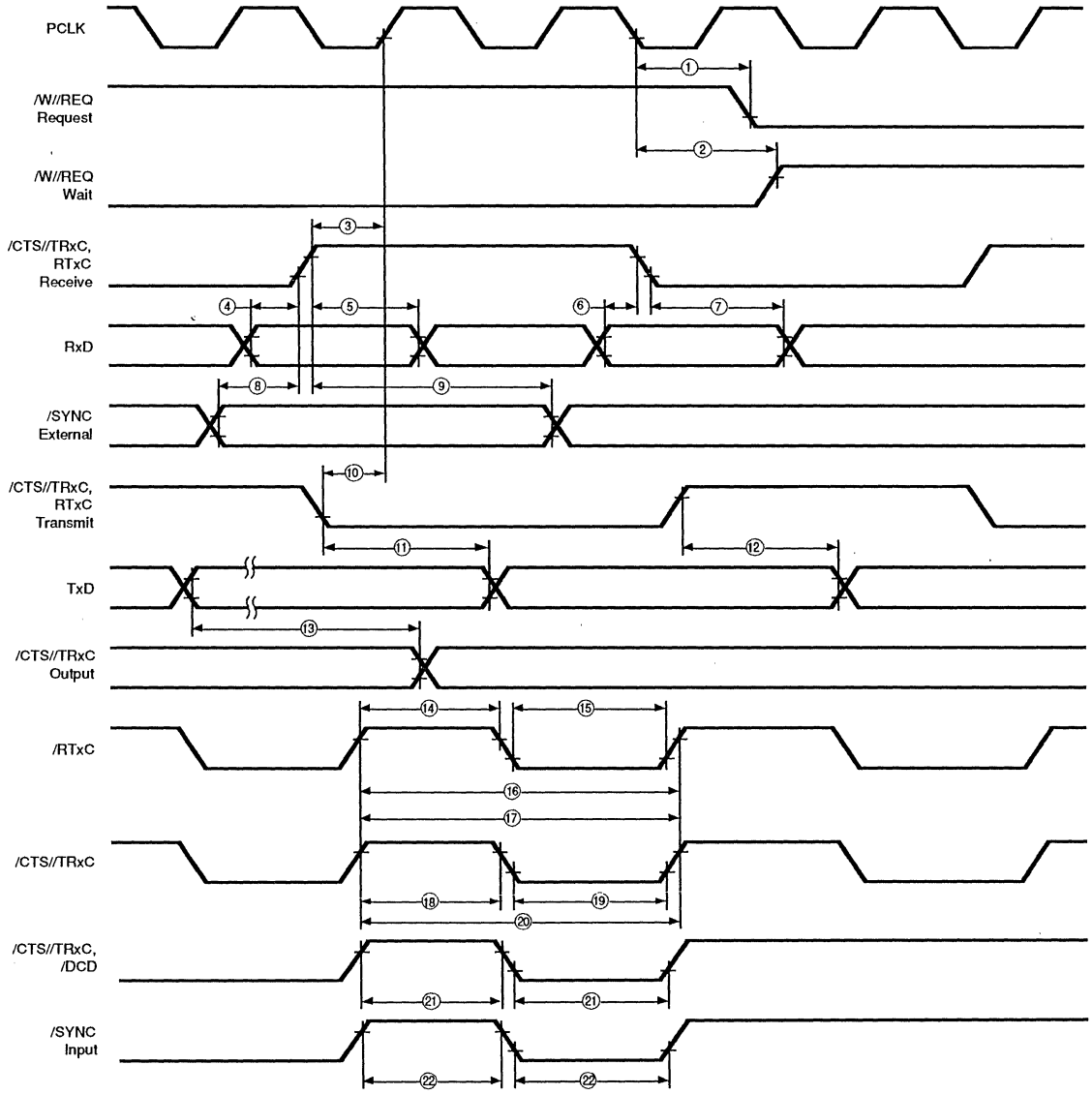


Figure 58. General Timing

AC CHARACTERISTICS (Continued)
Z85C80 General Timing

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

Notes:
[1] /RxC is /RTxC or TRxC, whichever is supplying the receive clock.
[2] /TxC is /TRxC or RTxC, whichever is supplying the transmit clock.
[3] Both /RTxC and /SYNC have 300 pf capacitors to ground connected to them.
[4] Synchronization of /RxC to PCLK is eliminated in divide by four operation.
[5] Parameter applies only to FM encoding/decoding.
[6] Parameter applies only for transmitter and receiver; DPLL and baud rate timing requirements are identical to case PCLK requirements.
[7] The maximum receive or transmit data is 1/4 PCLK.
[8] Applies to DPLL clock source only. Maximum data rate of 1/4 PCLK still applies. DPLL clock should have a 50% duty cycle.

† Units in nanoseconds (ns)

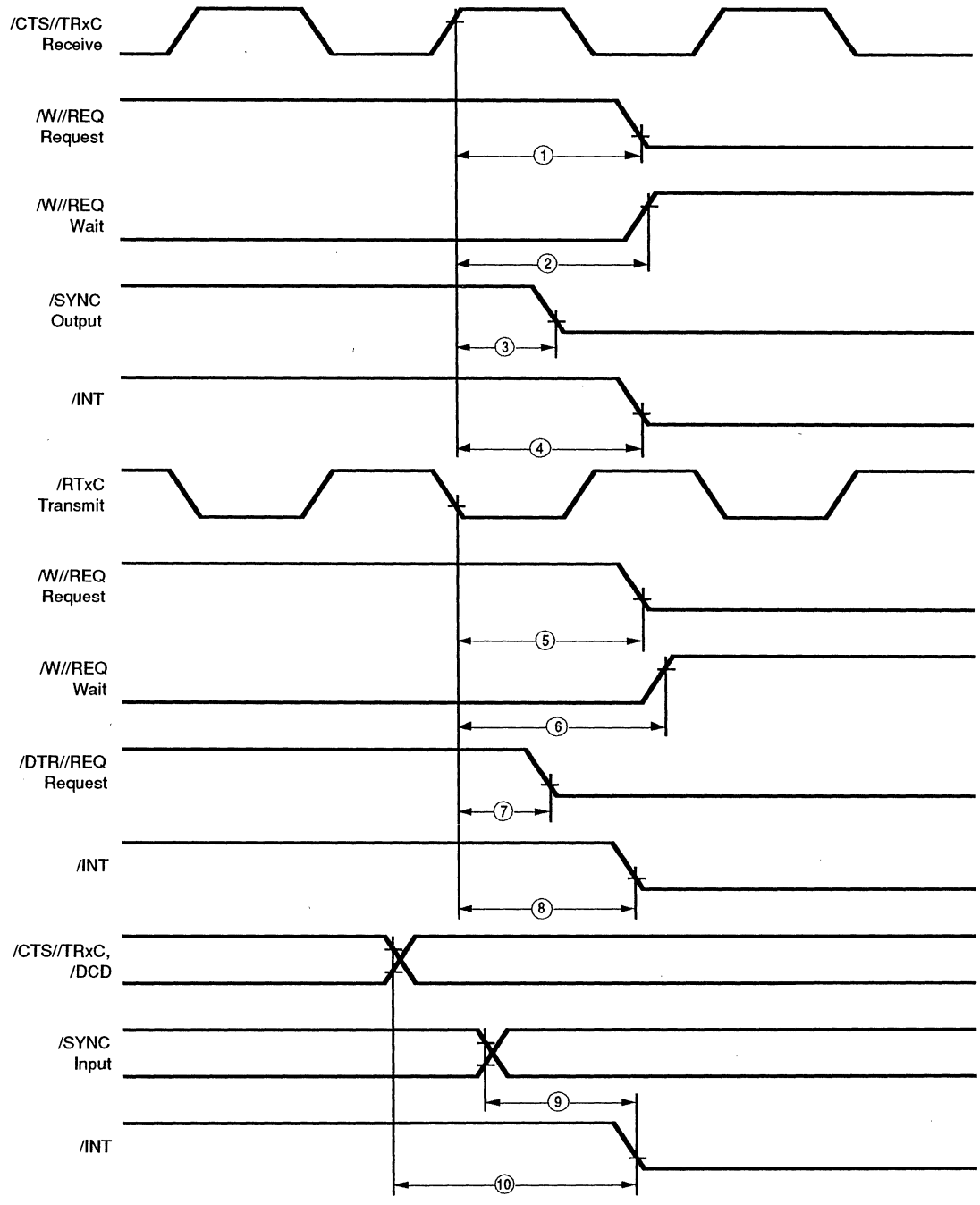


Figure 59. System Timing

AC CHARACTERISTICS (Continued)
Z85C80 System Timing

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

Notes:
[1] Open-drain output measured with open-drain test load.
[2] /RxC is /RTxC or /CTS//TRxC, whichever is supplying the receive clock.
[3] /TxC is /CTS//TRxC or RTxC, whichever is supplying the transmit clock.

† Units equal to TcPC

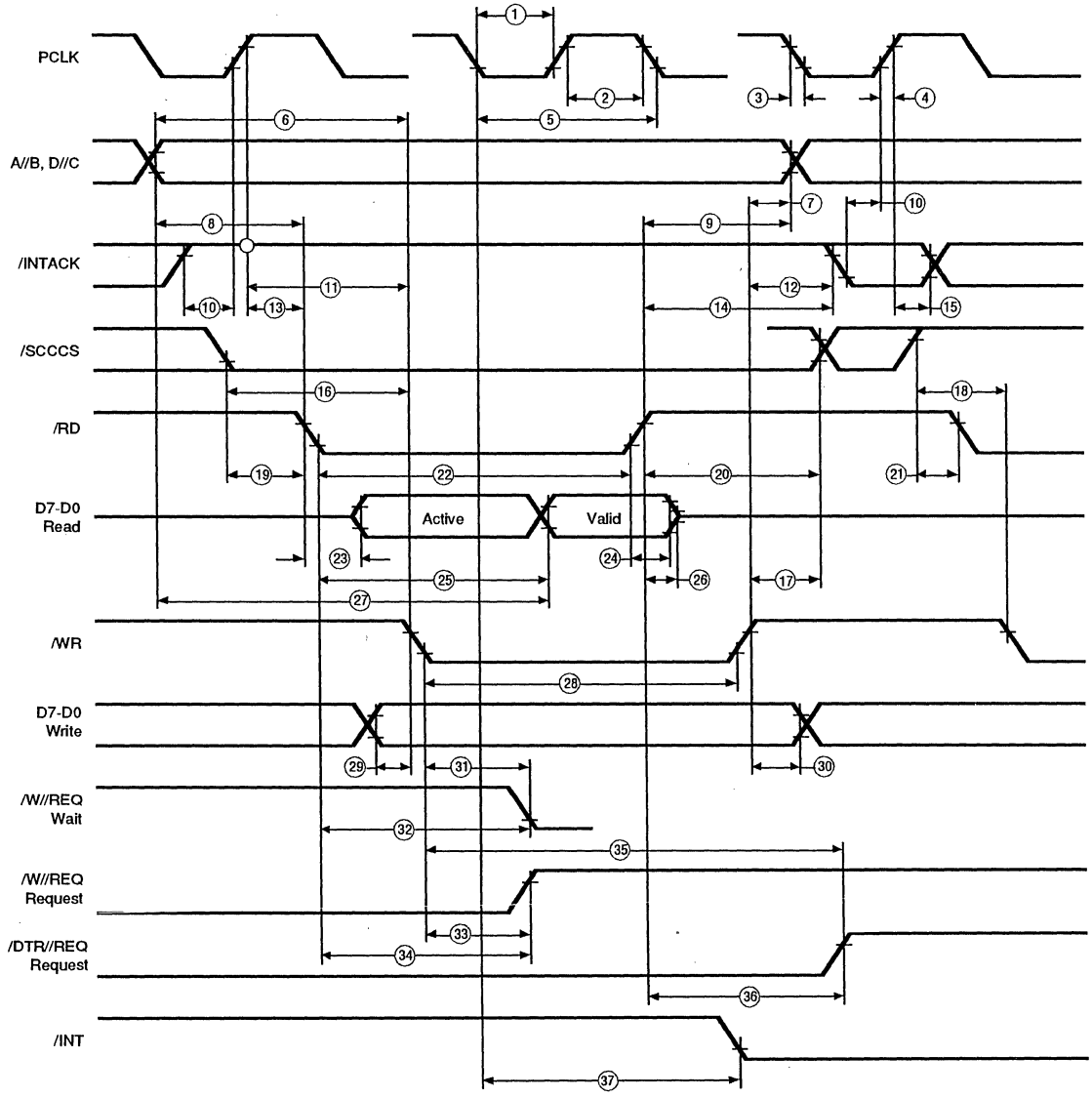


Figure 60. Read/Write Timing

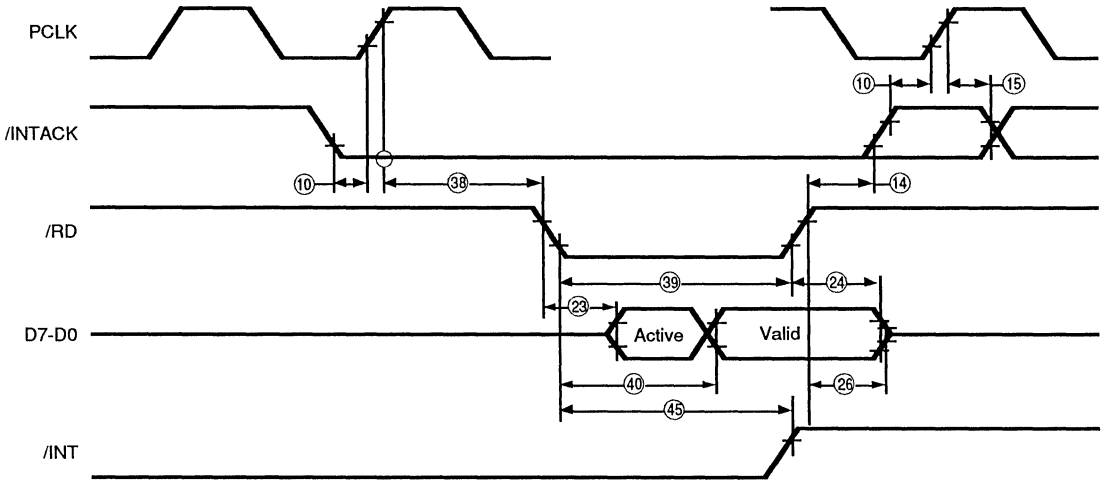


Figure 61. Interrupt Acknowledge Timing

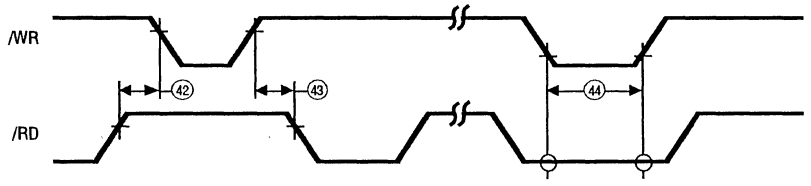


Figure 62. Reset Timing

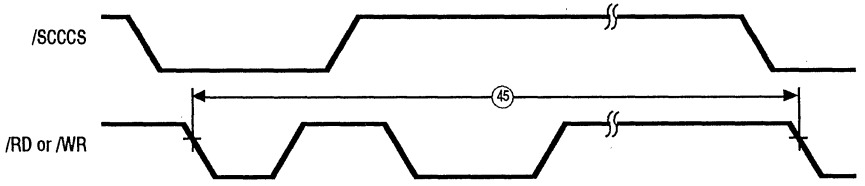


Figure 63. Cycle Timing

AC CHARACTERISTICS
Additional Timing

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

AC CHARACTERISTICS (Continued)
Additional Timing

No	Symbol	Parameter	Min	Max	Notes †
43	TdWRQ(RD)	/WR Rise to /RD Fall Delay for No Reset	15		
44	TwRES	/WR and /RD Coincident Low for Reset	100		
45	Trc	Valid Access Recovery Time	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)

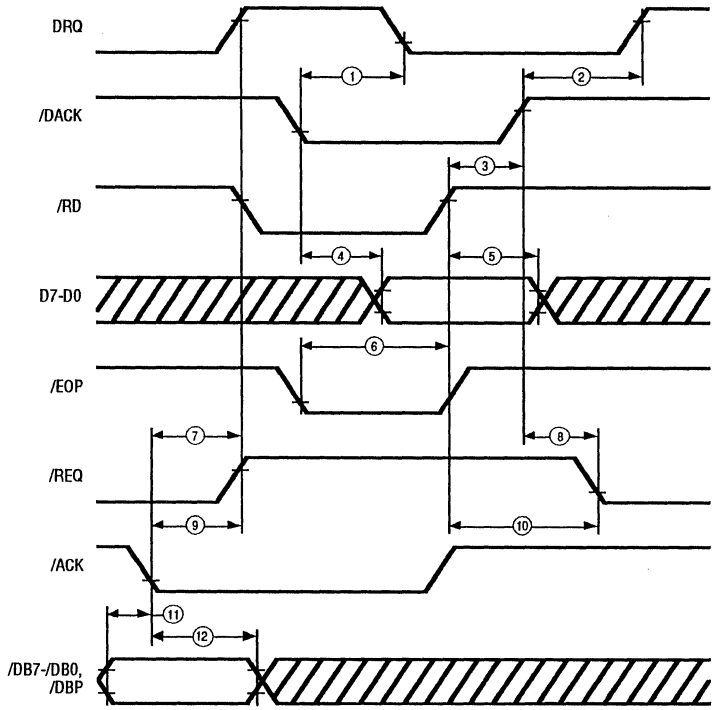


Figure 64. DMA Read Target Receive Cycle

AC CHARACTERISTICS
DMA Read Target Receive Cycle

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low		60	ns
2	/DACK High to DRQ High	30		ns
3	/DACK Hold Time from End of /RD	0		ns
4	Data Access Time from Read Enable*		70	ns
5	Data Hold Time from End of /RD	10		ns
6	Width of /EOP Pulse [1]	50		ns
7	/ACK Low to DRQ High		70	ns
8	/DACK High to /REQ Low (/ACK High)		90	ns
9	/ACK Low to /REQ High		80	ns
10	/ACK High to /REQ Low (/DACK High)		100	ns
11	Data Setup Time to /ACK	20		ns
12	Data Hold Time from /ACK	30		ns

Notes:
[1] /EOP, /RD, and /DACK must be concurrently Low for at least T6 for proper recognition of the /EOP pulse.
* Read Enable is the occurrence of /RD and /DACK.

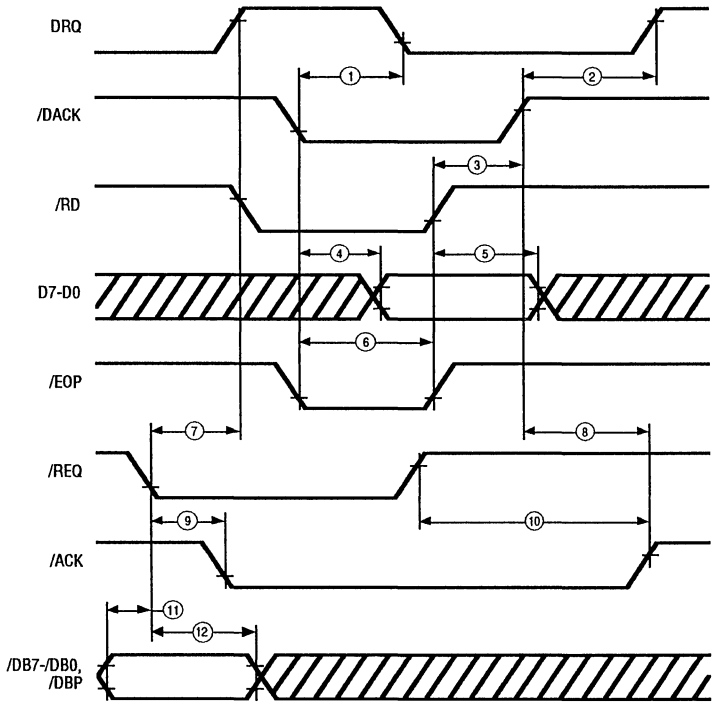


Figure 65. DMA Read Initiator Receive Cycle

AC CHARACTERISTICS
DMA Read Initiator Receive Cycle

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low		60	ns
2	/DACK High to DRQ High	30		ns
3	/DACK Hold Time from End of /RD	0		ns
4	Data Access Time from Read Enable*		70	ns
5	Data Hold Time from End of /RD	10		ns
6	Width of /EOP Pulse [1]	50		ns
7	/REQ Low to DRQ High		130	ns
8	/DACK High to /ACK High (/REQ High)		90	ns
9	/REQ Low to /ACK Low		130	ns
10	/REQ High to /ACK High (/DACK High)		80	ns
11	Data Setup Time to /REQ	20		ns
12	Data Hold Time from /REQ	110		ns

Notes:
[1] /EOP, /RD, and /DACK must be concurrently Low for at least T6 for proper recognition of the /EOP pulse.
* Read Enable is the occurrence of /RD and /DACK.

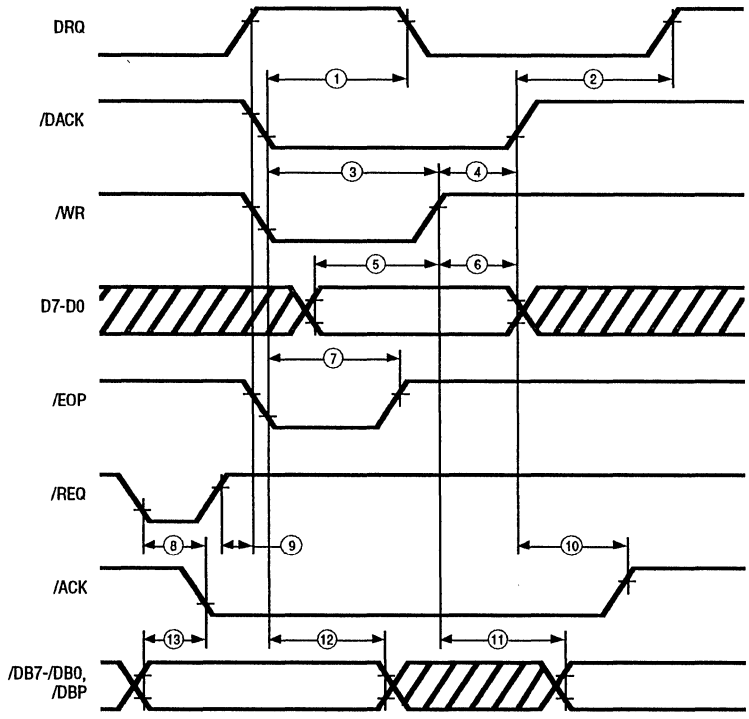


Figure 66. DMA Write Initiator Send Cycle

AC CHARACTERISTICS

DMA Write Initiator Send Cycle

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low		60	ns
2	/DACK High to DRQ High	30		ns
3	Write Enable Width*	50		ns
4	/DACK Hold from End of /WR	0		ns
5	Data Setup to End of Write Enable*	50		ns
6	Data Hold Time from End of /WR	25		ns
7	Width of /EOP Pulse [1]	50		ns
8	/REQ Low to /ACK Low		130	ns
9	/REQ High to DRQ High		70	ns
10	/DACK High to /ACK High		90	ns
11	/WR High to Valid SCSI Data		50	ns
12	Data Hold from Write Enable*	15		ns
13	Data Setup to /ACK Low	55		ns

Notes:

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

* Write Enable is the occurrence of /WR and /DACK.

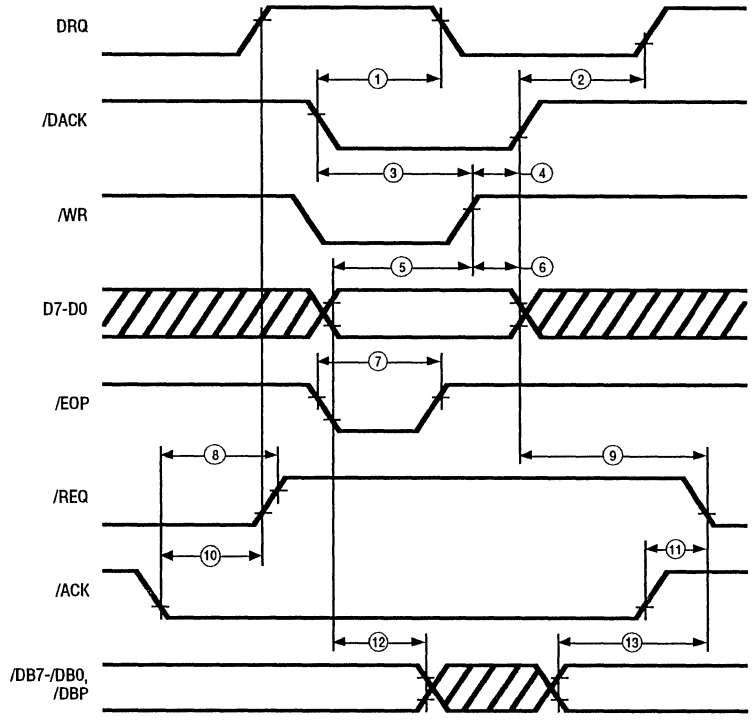


Figure 67. DMA Write Target Send Cycle

AC CHARACTERISTICS

DMA Write Target Send Cycle

No	Description	Min	Max	Units
1	DRQ Low from /DACK Low		60	ns
2	/DACK High to DRQ High	30		ns
3	Write Enable Width*	50		ns
4	/DACK Hold from /WR High	0		ns
5	Data Setup to End of Write Enable*	50		ns
6	Data Hold Time from End of /WR	25		ns
7	Width of /EOP Pulse [1]	50		ns
8	/ACK Low to /REQ High		80	ns
9	/REQ from End of /DACK (/ACK High)		90	ns
10	/ACK Low to DRQ High (Target)		70	ns
11	/ACK High to /REQ Low (/DACK High)		100	ns
12	Data Hold from Write Enable	15		ns
13	Data Setup to /REQ Low (Target)	55		ns

Notes:

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

* Write Enable is the occurrence of /IOW and /DACK

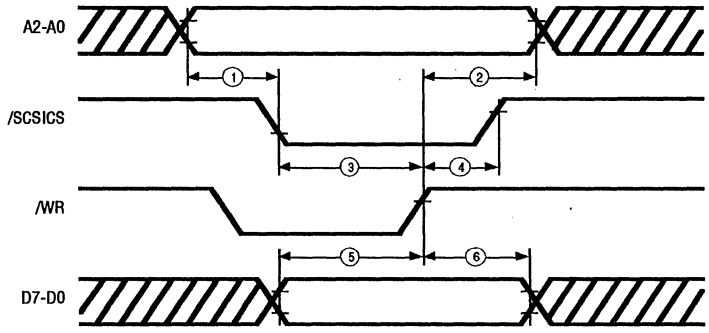


Figure 68. CPU Write Cycle

AC CHARACTERISTICS

CPU Write Cycle

No	Description	Min	Max	Units
1	Address Setup to Write Enable*	10		ns
2	Address Hold from End Write Enable*	10		ns
3	Write Enable Width*	40		ns
4	Chip Select Hold from End of /IOW	0		ns
5	Data Setup to end of Write Enable*	20		ns
6	Data Hold Time form End of /IOW	20		ns

Note:

* Write Enable is the occurrence of /WR and /SCSICS

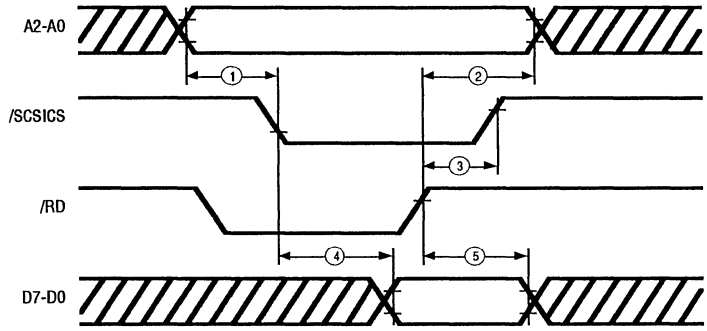


Figure 69. CPU Read Cycle

AC CHARACTERISTICS
CPU Read Cycle

No	Description	Min	Max	Units
1	Address Setup to Read Enable*	10		ns
2	Address Hold from End Read Enable*	10		ns
3	Chip Select Hold from End of /RD	0		ns
4	Data Access Time from Read Enable*		70	ns
5	Data Hold Time from End of Read Enable*	10		ns

Note:
* Read Enable is the occurrence of /RD and /SCSICS

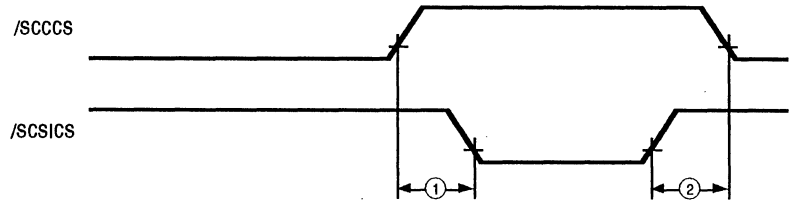


Figure 70. Selection

AC CHARACTERISTICS
Selection

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

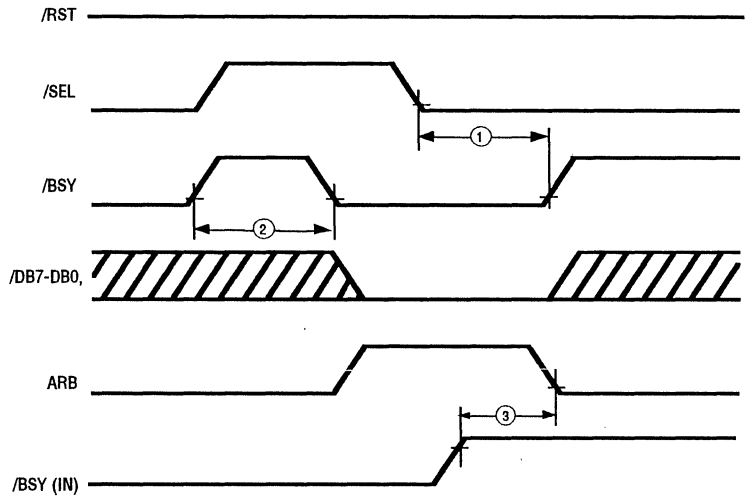


Figure 71. Arbitration

AC CHARACTERISTICS
Arbitration

No	Description	Min	Max	Units
1	Bus Clear from /SEL Low		600	ns
2	Arbitrate Start from /BSY False	1200	2200	ns
3	Bus Clear from /BSY High		1100	

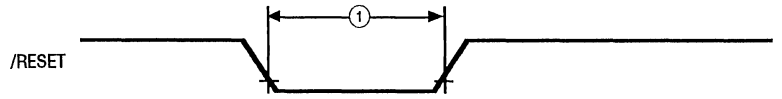


Figure 72. Reset

AC CHARACTERISTICS
Reset

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