CHAPTER 2
INTERFACING THE SCC/ESCC

2.1 INTRODUCTION

This chapter covers the system interface requirements with the SCC Timing requirements for both devices are described in a general sense here, and the user should refer to the SCC Product Specification for detailed AC/DC parametric requirements.

The ESCC has an additional new register, Write Register Seven Prime (WR7'). Its features include the ability to read WR3, WR4, WR5, WR7', and WR10. Both the Z80230 and the Z85230 have the ability to deassert the /DTR//REQ pin quickly to ease DMA interface design. Additionally, the Z85230 features a relaxed requirement for a valid data bus when the /WR pin goes Low. The effects of the deeper data FIFOs should be considered when writing the interrupt service routines. The user should read the sections which follow for details on these features.

2.2 Z80X30 INTERFACE TIMING

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

Two control signals, /AS and /DS, are used by the Z80X30 to time bus transactions. In addition, four other control signals (/CS0, CS1, R/W, and /INTACK) are used to control the type of bus transaction that occurs. A bus transaction is initiated by /AS; the rising edge latches the register address on the Address/Data bus and the state of /INTACK and /CS0.

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

The Z80X30 generates internal control signals in response to a register access. Since /AS and /DS have no phase relationship with PCLK, the circuit generating these internal control signals provides time for metastable conditions to disappear. This results in a recovery time related to PCLK.

This recovery time applies only to transactions involving the Z80X30, and any intervening transactions are ignored. This recovery time is four PCLK cycles, measured from the falling edge of /DS of one access to the SCC, to the falling edge of /DS for a subsequent access.

2.2.1 Z80X30 Read Cycle Timing

The read cycle timing for the Z80X30 is shown in Figure 2-1. The register address on AD7-AD0, as well as the state of /CS0 and /INTACK, are latched by the rising edge of /AS. R/W must be High before /DS falls to indicate a read cycle. The Z80X30 data bus drivers are enabled while CS1 is High and /DS is Low.

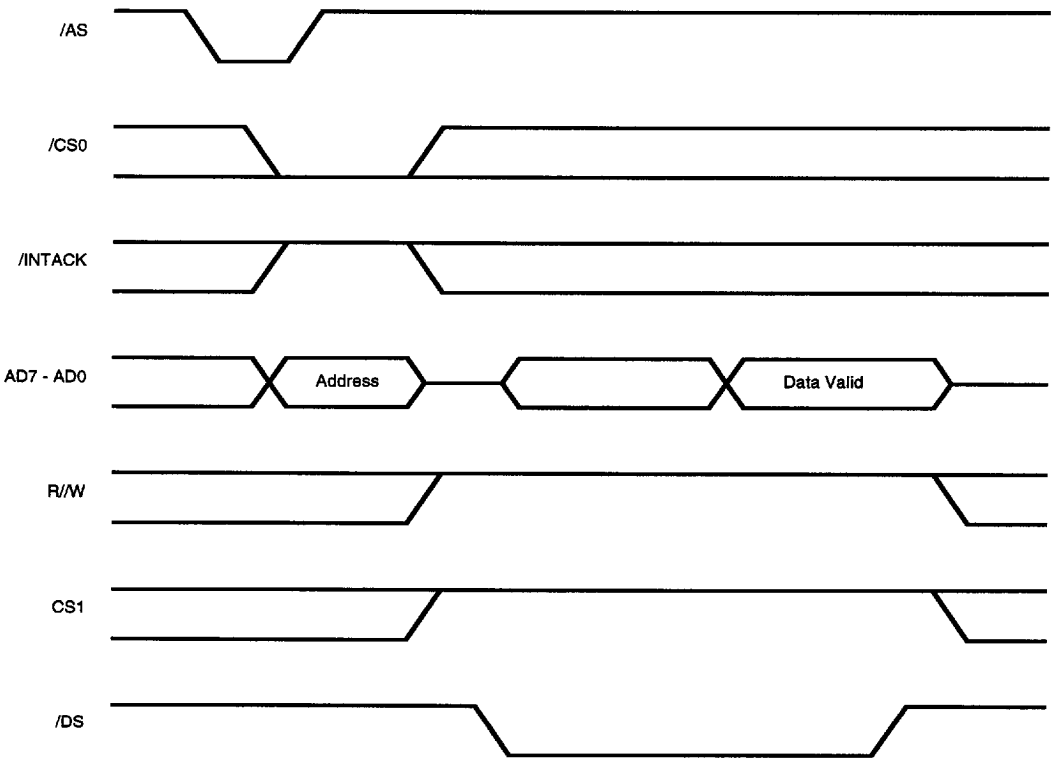

Figure 2-1. Z80X30 Read Cycle

2.2.2 Z80X30 Write Cycle Timing

The write cycle timing for the Z80X30 is shown in Figure 2-2. The register address on AD7-AD0, as well as the state of /CS0 and /INTACK, are latched by the rising edge of /AS. R/W must be Low when /DS falls to indicate a write cycle. The leading edge of the coincidence of CS1 High and /DS Low latches the write data on AD7-AD0, as well as the state of R/W.

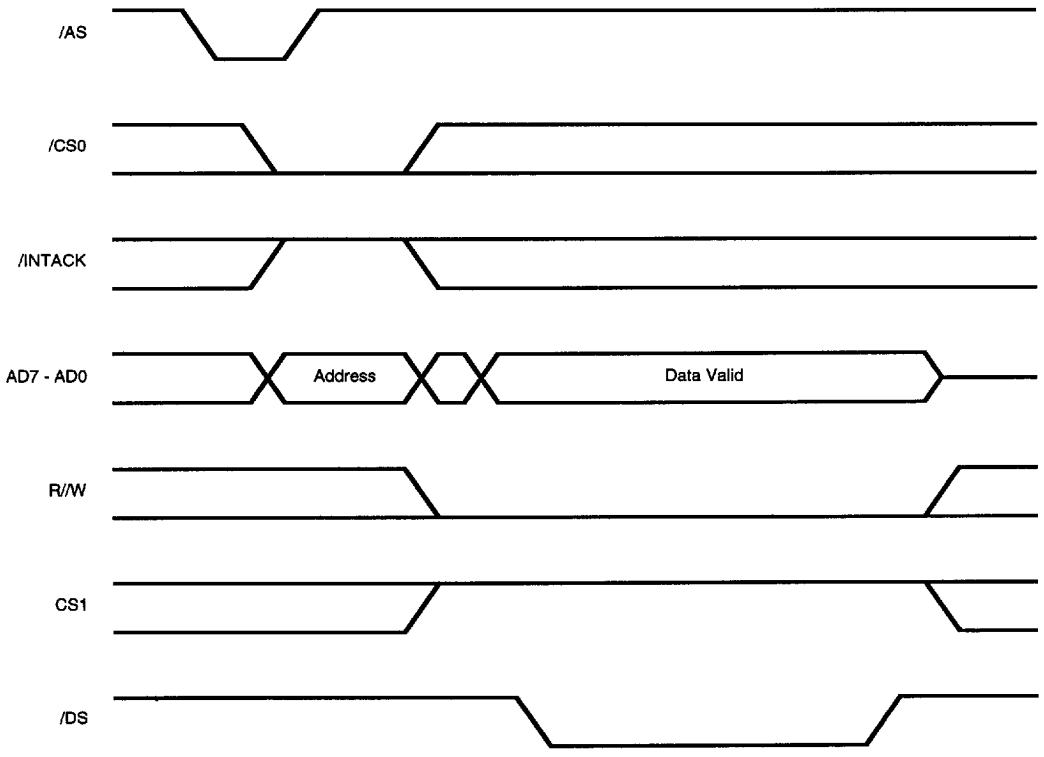

Figure 2-2. Z80X30 Write Cycle

2.2.3 Z80X30 Interrupt Acknowledge Cycle Timing

The interrupt acknowledge cycle timing for the Z80X30 is shown in Figure 2-3. The address on AD7-AD0 and the state of /CS0 and /INTACK are latched by the rising edge of /AS. However, if /INTACK is Low, the address, /CS0, CS1 and R/W are ignored for the duration of the interrupt acknowledge cycle.

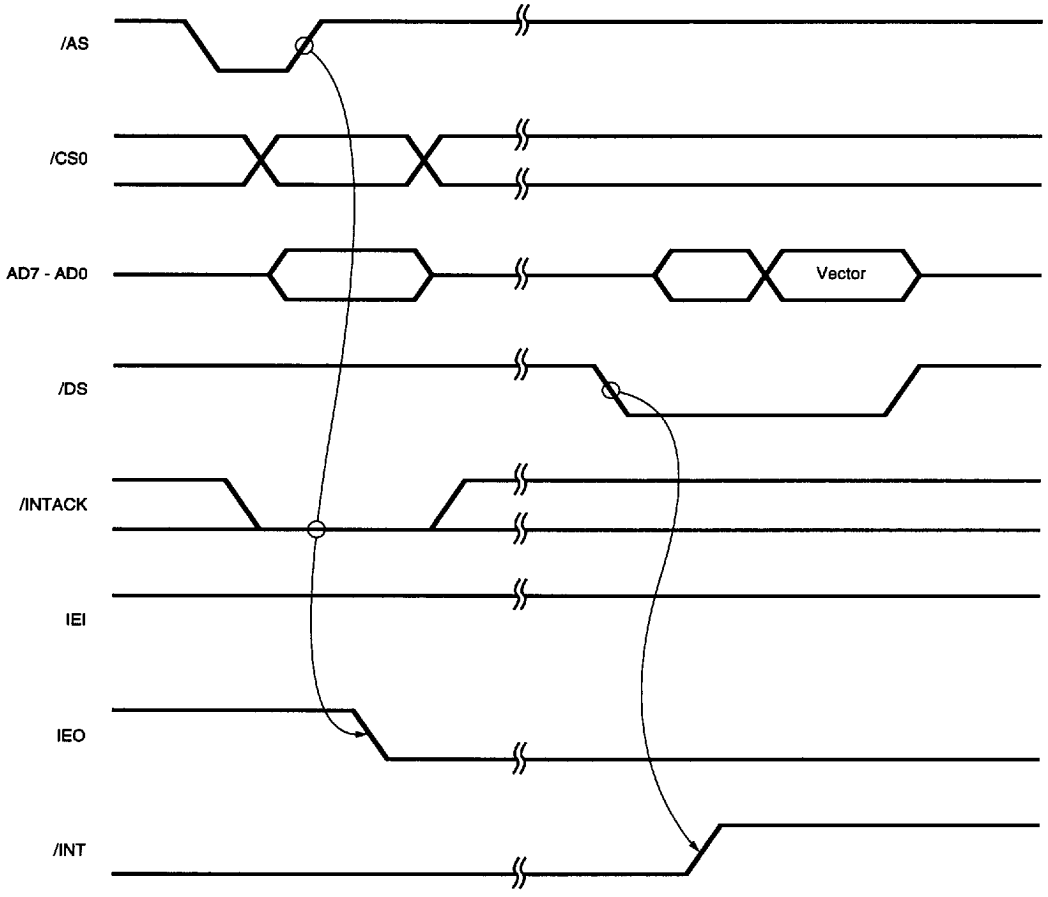

Figure 2-3. Z80X30 Interrupt Acknowledge Cycle

The Z80X30 samples the state of /INTACK on the rising edge of /AS, and AC parameters #7 and #8 specify the setup and hold-time requirements. Between the rising edge of /AS and the falling edge of /DS, the internal and external daisy chains settle (AC parameter #29). A system with no external daisy chain should provide the time specified in spec #29 to settle the interrupt daisy-chain priority internal to the SCC. Systems using an external daisy chain should refer to Note 5 referenced in the Z80X30 Read/Write & Interrupt Acknowledge Timing for the time required to settle the daisy chain.

Note: /INTACK is sampled on the rising edge of /AS. If it does not meet the setup time to the first rising edge of /AS

of the interrupt acknowledge cycle, it is latched on the next rising edge of /AS. Therefore, if /INTACK is asynchronous to /AS, it may be necessary to add a PCLK cycle to the calculation for /INTACK to /RD delay time.

If there is an interrupt pending in the SCC, and IEI is High when /DS falls, the acknowledge cycle was intended for the SCC. This being the case, the Z80X30 sets the Interrupt-Under-Service (IUS) latch for the highest priority pending interrupt, as well as placing an interrupt vector on AD7-AD0. The placing of a vector on the bus can be disabled by setting WR9, D1=1. The /INT pin also goes inactive in response to the falling edge of /DS. Note that there should be only one /DS per acknowledge cycle. Another important fact is that the IP bits in the Z80X30 are updated by /AS, which may delay interrupt requests if the processor does not supply /AS strobes during the time between accesses of the Z80X30.

2.2.4 Z80X30 Register Access

The registers in the Z80X30 are addressed via the address on AD7-AD0 and are latched by the rising edge of /AS. The Shift Right/Shift Left bit in the Channel B WR0 controls which bits are decoded to form the register address. It is placed in this register to simplify programming when the current state of the Shift Right/Shift Left bit is not known.

A hardware reset forces Shift Left mode where the address is decoded from AD5-AD1. In Shift Right mode, the address is decoded from AD4-AD0. The Shift Right/Shift Left bit is written via a command to make the software writing to WR0 independent of the state of the Shift Right/Shift Left bit.

While in the Shift Left mode, the register address is placed on AD4-AD1 and the Channel Select bit, A/B, is decoded from AD5. The register map for this case is shown in Table 2-1. In Shift Right mode, the register address is again placed on AD4-AD1 but the channel select A/B is decoded from AD0. The register map for this case is shown in Table 2-2.

Because the Z80X30 does not contain 16 read registers, the decoding of the read registers is not complete; this is indicated in Table 2-1 and Table 2-2 by parentheses around the register name. These addresses may also be used to access the read registers. Also, note that the Z80X30 contains only one WR2 and WR9; these registers may be written from either channel.

Shift Left Mode is used when Channel A and B are to be programmed differently. This allows the software to sequence through the registers of one channel at a time. The Shift Right Mode is used when the channels are programmed the same. By incrementing the address, the user can program the same data value into both the Channel A and Channel B register.

Table 2-1. Z80X30 Register Map (Shift Left Mode)

AD5	AD4	AD3	AD2	AD1	WRITE	READ 8030 80C30/230* WR15 D2 = 0	80C30/230 WR15 D2=1	80230 WR15 D2=1 WR7' D6=1
0	0	0	0	0	WR0B	RR0B	RR0B	RR0B
0	0	0	0	1	WR1B	RR1B	RR1B	RR1B
0	0	0	1	0	WR2	RR2B	RR2B	RR2B
0	0	0	1	1	WR3B	RR3B	RR3B	RR3B
0	0	1	0	0	WR4B	(RR0B)	(RR0B)	(WR4B)
0	0	1	0	1	WR5B	(RR1B)	(RR1B)	(WR5B)
0	0	1	1	0	WR6B	(RR2B)	RR6B	RR6B
0	0	1	1	1	WR7B	(RR3B)	RR7B	RR7B
0	1	0	0	0	WR8B	RR8B	RR8B	RR8B
0	1	0	0	1	WR9	(RR13B)	(RR13B)	(WR3B)
0	1	0	1	0	WR10B	RR10B	RR10B	RR10B
0	1	0	1	1	WR11B	(RR15B)	(RR15B)	(WR10B)
0	1	1	0	0	WR12B	RR12B	RR12B	RR12B
0	1	1	0	1	WR13B	RR13B	RR13B	RR13B
0	1	1	1	0	WR14B	RR14B	RR14B	(WR7'B)
0	1	1	1	1	WR15B	RR15B	RR15B	RR15B
1	0	0	0	0	WR0A	RR0A	RR0A	RR0A
1	0	0	0	1	WR1A	RR1A	RR1A	RR1A
1	0	0	1	0	WR2	RR2A	RR2A	RR2A
1	0	0	1	1	WR3A	RR3A	RR3A	RR3A
1	0	1	0	0	WR4A	(RR0A)	(RR0A)	(WR4A)
1	0	1	0	1	WR5A	(RR1A)	(RR1A)	(WR5A)
1	0	1	1	0	WR6A	(RR2A)	RR6A	RR6A
1	0	1	1	1	WR7A	(RR3A)	RR7A	RR7A
1	1	0	0	0	WR8A	RR8A	RR8A	RR8A
1	1	0	0	1	WR9A	(RR13A)	(RR13A)	(WR3A)
1	1	0	1	0	WR10A	RR10A	RR10A	RR10A
1	1	0	1	1	WR11A	(RR15A)	(RR15A)	(WR10A)
1	1	1	0	0	WR12A	RR12A	RR12A	RR12A
1	1	1	0	1	WR13A	RR13A	RR13A	RR13A
1	1	1	1	0	WR14A	RR14A	RR14A	(WR7'A)
1	1	1	1	1	WR15A	RR15A	RR15A	RR15A

Notes:
The register names in ( ) are the values read out from that register location
WR15, bit D2 enables status FIFO function (not available on NMOS).
WR7' bit D6 enables extend read function (only on ESCC).
* Includes 80C30/230 when WR15 D2=0.

Table 2-2. Z80X30 Register Map (Shift Right Mode)

AD5	AD4	AD3	AD2	AD1	WRITE	READ 8030 80C30/230* WR15 D2 = 0	80C30/230 WR15 D2=1	80230 WR15 D2=1 WR7' D6=1
0	0	0	0	0	WR0B	RR0B	RR0B	RR0B
0	0	0	0	1	WR0A	RR0A	RR0A	RR0A
0	0	0	1	0	WR1B	RR1B	RR1B	RR1B
0	0	0	1	1	WR1A	RR1A	RR1A	RR1A
0	0	1	0	0	WR2	RR2B	RR2B	RR2B
0	0	1	0	1	WR2	RR2A	RR2A	RR2A
0	0	1	1	0	WR3B	RR3B	RR3B	RR3B
0	0	1	1	1	WR3A	RR3A	RR3A	RR3A
0	1	0	0	0	WR4B	(RR0B)	(RR0B)	(WR4B)
0	1	0	0	1	WR4A	(RR0A)	(RR0A)	(WR4A)
0	1	0	1	0	WR5B	(RR1B)	(RR1B)	(WR5B)
0	1	0	1	1	WR5A	(RR1A)	(RR1A)	(WR5A)
0	1	1	0	0	WR6B	(RR2B)	RR6B	RR6B
0	1	1	0	1	WR6A	(RR2A)	RR6A	RR6A
0	1	1	1	0	WR7B	(RR3B)	RR7B	RR7B
0	1	1	1	1	WR7A	(RR3A)	RR7A	RR7A
1	0	0	0	0	WR8B	RR8B	RR8B	RR8B
1	0	0	0	1	WR8A	RR8A	RR8A	RR8A
1	0	0	1	0	WR9	(RR13B)	(RR13B)	(WR3B)
1	0	0	1	1	WR9A	(RR13A)	(RR13A)	(WR3A)
1	0	1	0	0	WR10B	RR10B	RR10B	RR10B
1	0	1	0	1	WR10A	RR10A	RR10A	RR10A
1	0	1	1	0	WR11B	(RR15B)	(RR15B)	(WR10B)
1	0	1	1	1	WR11A	(RR15A)	(RR15A)	(WR10A)
1	1	0	0	0	WR12B	RR12B	RR12B	RR12B
1	1	0	0	1	WR12A	RR12A	RR12A	RR12A
1	1	0	1	0	WR13B	RR13B	RR13B	RR13B
1	1	0	1	1	WR13A	RR13A	RR13A	RR13A
1	1	1	0	0	WR14B	RR14B	RR14B	(WR7'B)
1	1	1	0	1	WR14A	RR14A	RR14A	(WR7'A)
1	1	1	1	0	WR15B	RR15B	RR15B	RR15B
1	1	1	1	1	WR15A	RR15A	RR15A	RR15A

Notes:
The register names in ( ) are the values read out from that register location
WR15 bit D2 enables status FIFO function (not available on NMOS).
WR7' bit D6 enables extend read function (only on ESCC).
* Includes 80C30/230 when WR15 D2=0

2.2.5 Z80C30 Register Enhancement

The Z80C30 has an enhancement to the NMOS Z8030 register set, which is the addition of a 10x19 SDLC Frame Status FIFO. When WR15 bit D2=1, the SDLC Frame Status FIFO is enabled, and it changes the functionality of RR6 and RR7. See Section 4.4.3 for more details on this feature.

2.2.6 Z80230 Register Enhancements

In addition to the Z80C30 enhancements, the 80230 has several enhancements to the SCC register set. These include the addition of Write Register 7 Prime (WR7'), and the ability to read registers that are read only in the 8030.

Write Register 7' is addressed by setting WR15 bit, D0=1 and then addressing WR7. Figure 2-4 shows the register bit location of the six features enabled through this register. All writes to address seven are to WR7' when WR15, D0=1. Refer to Chapter 5 for detailed information on WR7'.

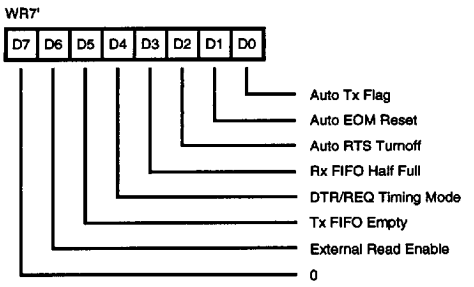

Figure 2-4. Write Register 7 Prime (WR7')

WR7' bit D6=1, enables the extended read register capability. This allows the user to read the contents of WR3, WR4, WR5, WR7' and WR10 by reading RR9, RR4, RR5, RR14 and RR11, respectively. When WR7' D6=0, these write registers are write only.

Table 2-3 shows what functions are enabled for the various combinations of register bit enables. See Table 2-1 (Shift Left) and Table 2-2 (Shift Right) for the register address map with the SDLC FIFO enabled only and the map with both the extended read and SDLC FIFO features enabled.

Table 2-3. Z80230 SDLC/HDLC Enhancement Options

WR15 Bit D2	WR15 Bit D0	WR7' Bit D6	Functions Enabled
0	1	0	WR7' enabled only
0	1	1	WR7' with extended read enabled
1	0	X	10x19 SDLC FIFO enhancement enabled only
1	1	0	10x19 SDLC FIFO and WR7'
1	1	1	10x19 SDLC FIFO and WR7' with extended read enabled

2.2.7 Z80X30 Reset

The Z80X30 may be reset by either a hardware or software reset. Hardware reset occurs when /AS and /DS are both Low at the same time, which is normally an illegal condition. As long as both /AS and /DS are Low, the Z80X30 recognizes the reset condition. However, once this condition is removed, the reset condition is asserted internally for an additional four to five PCLK cycles. During this time, any attempt to access is ignored.

The Z80X30 has three software resets that are encoded into two command bits in WR9. There are two channel resets, which only affect one channel in the device and some bits of the write registers. The command forces the same result as the hardware reset, the Z80X30 stretches the reset signal an additional four to five PCLK cycles beyond the ordinary valid access recovery time. The bits in WR9 may be written at the same time as the reset command because these bits are affected only by a hardware reset. The reset values of the various registers are shown in Table 2-4.

Table 2-4. Z80X30 Register Reset Values

	Hardware RESET								Channel RESET							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
WR0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
WR1	0	0	X	0	0	X	0	0	0	0	X	0	0	X	0	0
WR2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
WR3	X	X	X	X	X	X	X	0	X	X	X	X	X	X	X	0
WR4	X	X	X	X	X	1	X	X	X	X	X	X	X	1	X	X
WR5	0	X	X	0	0	0	0	X	0	X	X	0	0	0	0	X
WR6	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
WR7	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
WR7'*	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
WR9	1	1	0	0	0	0	X	X	X	X	0	X	X	X	X	X
WR10	0	0	0	0	0	0	0	0	0	X	X	0	0	0	0	0
WR11	0	0	0	0	1	0	0	0	X	X	X	X	X	X	X	X
WR12	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
WR13	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
WR14	X	X	1	1	0	0	0	0	X	X	1	0	0	0	X	X
WR15	1	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0
RR0	X	1	X	X	X	1	0	0	X	1	X	X	X	1	0	0
RR1	0	0	0	0	0	1	1	X	0	0	0	0	0	1	1	X
RR3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RR10	0	X	0	0	0	0	0	0	0	X	0	0	0	0	0	0

Notes:
*'WR7' is available only on the Z80230

2.3 Z85X30 INTERFACE TIMING

Two control signals, /RD and /WR, are used by the Z85X30 to time bus transactions. In addition, four other control signals, /CE, D//C, A//B and /INTACK, are used to control the type of bus transaction that occurs. A bus transaction starts when the addresses on D//C and A//B are asserted before /RD or /WR fall (AC Spec #6 and #8). The coincidence of /CE and /RD or /CE and /WR latches the state of D//C and A//B and starts the internal operation. The /INTACK signal must have been previously sampled High by a rising edge of PCLK for a read or write cycle to occur. In addition to sampling /INTACK, PCLK is used by the interrupt section to set the IP bits.

The Z85X30 generates internal control signals in response to a register access. Since /RD and /WR have no phase relationship with PCLK, the circuitry generating these internal control signals provides time for metastable conditions to disappear. This results in a recovery time related to PCLK.

This recovery time applies only between transactions involving the Z85X30, and any intervening transactions are ignored. This recovery time is four PCLK cycles (AC Spec #49), measured from the falling edge of /RD or /WR in the case of a read or write of any register.

2.3.1 Z85X30 Read Cycle Timing

The read cycle timing for the Z85X30 is shown in Figure 2-5. The address on A//B and D//C is latched by the coincidence of /RD and /CE active. /CE must remain Low and /INTACK must remain High throughout the cycle. The Z85X30 bus drivers are enabled while /CE and /RD are both Low. A read with D//C High does not disturb the state of the pointers and a read cycle with D//C Low resets the pointers to zero after the internal operation is complete.

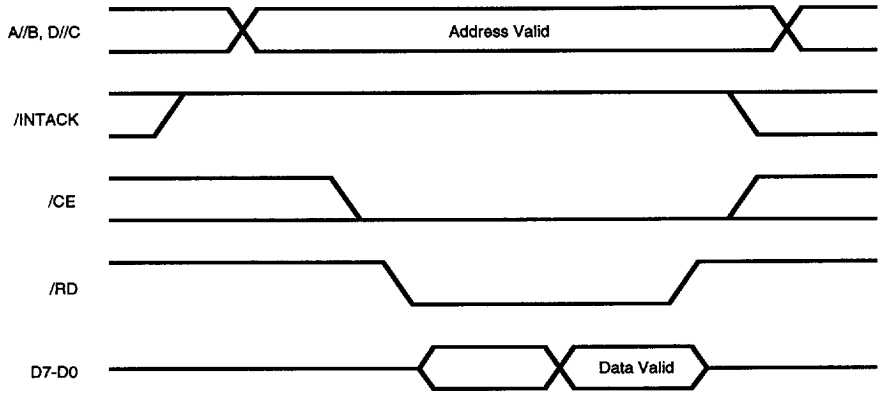

Figure 2-5. Z85X30 Read Cycle Timing

2.3.2 Z85X30 Write Cycle Timing

The write cycle timing for the Z85X30 is shown in Figure 2-6. The address on A//B and D//C, as well as the data on D7-D0, is latched by the coincidence of /WR and /CE active. /CE must remain Low and /INTACK must remain High throughout the cycle. A write cycle with D//C High does not disturb the state of the pointers and a write cycle with D//C Low resets the pointers to zero after the internal operation is complete.

Historically, the NMOS/CMOS version latched the data bus on the falling edge of /WR. However, many CPUs do not guarantee that the data bus is valid at the time when the /WR pin goes low, so the data bus timing was modified to allow a maximum delay from the falling edge of /WR to the latching of the data bus. On the Z85230, the AC Timing parameter #29 TsDW(WR), Write Data to /WR falling minimum, has been changed to: /WR falling to Write Data Valid maximum. Refer to the AC Timing Characteristic section of the Z85230 Product Specification for more information regarding this change

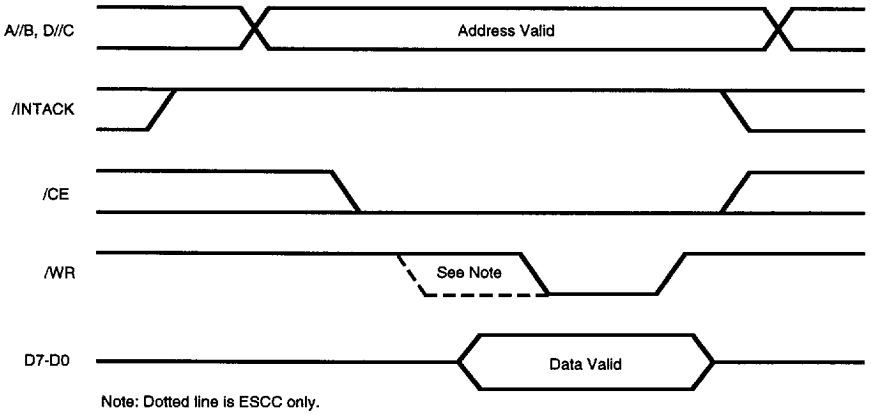

Figure 2-6. Z85X30 Write Cycle Timing

2.3.3 Z85X30 Interrupt Acknowledge Cycle Timing

The interrupt acknowledge cycle timing for the Z85X30 is shown in Figure 2-7. The state of /INTACK is latched by the rising edge of PCLK (AC Spec #10). While /INTACK is Low, the state of A//B, /CE, D//C, and /WR are ignored.

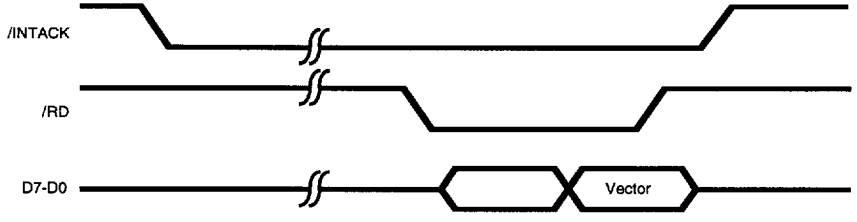

Figure 2-7. Z85X30 Interrupt Acknowledge Cycle Timing

Between the time /INTACK is first sampled Low and the time /RD falls, the internal and external IEI/IEO daisy chain settles (AC parameter #38 TdIAI(RD) Note 5). A system with no external daisy chain must provide the time specified in AC Spec #38 to settle the interrupt daisy chain priority internal to the SCC. Systems using the external IEI/IEO daisy chain should refer to Note 5 referenced in the Z85X30 Read/Write and Interrupt Acknowledge Timing for the time required to settle the daisy chain.

Note: /INTACK is sampled on the rising edge of PCLK. If it does not meet the setup time to the first rising edge of PCLK of the interrupt acknowledge cycle, it is latched on the next rising edge of PCLK. Therefore, if /INTACK is asynchronous to PCLK, it may be necessary to add a PCLK cycle to the calculation for /INTACK to /RD delay time.

If there is an interrupt pending in the Z85X30, and IEI is High when /RD falls, the interrupt acknowledge cycle was intended for the Z85X30. In this case, the Z85X30 sets the appropriate Interrupt-Under-Service latch, and places an interrupt vector on D7-D0.

If the falling edge of /RD sets an IUS bit in the Z85X30, the /INT pin goes inactive in response to the falling edge. Note that there should be only one /RD per acknowledge cycle.

Note 1: The IP bits in the Z85X30 are updated by PCLK. However, when the register pointer is pointing to RR2 and RR3, the IP bits are prevented from changing. This prevents data changing during a read, but will delay interrupt requests if the pointers are left pointing at these registers.

Note 2: The SCC should only receive one INTACK signal per acknowledge cycle. Therefore, if the CPU generates more than one (as is common for the 80X86 family), an external circuit should be used to convert this into a single pulse or does not use Interrupt Acknowledge.

2.3.4 Z85X30 Register Access

The registers in the Z85X30 are accessed in a two step process, using a Register Pointer to perform the addressing. To access a particular register,the pointer bits are set by writing to WR0. The pointer bits may be written in either channel because only one set exists in the Z85X30. After the pointer bits are set, the next read or write cycle of the Z85X30 having D//C Low will access the desired register. At the conclusion of this read or write cycle the pointer bits are reset to 0s, so that the next control write is to the pointers in WR0.

A read to RR8 (the receive data FIFO) or a write to WR8 (the transmit data FIFO) is either done in this fashion or by accessing the Z85X30 having D//C pin High. A read or write with D//C High accesses the data registers directly, and independently of the state of the pointer bits. This allows single-cycle access to the data registers and does not disturb the pointer bits.

The fact that the pointer bits are reset to 0, unless explicitly set otherwise, means that WR0 and RR0 may also be accessed in a single cycle. That is, it is not necessary to write the pointer bits with 0 before accessing WR0 or RR0.

There are three pointer bits in WR0, and these allow access to the registers with addresses 7 through 0. Note that a command may be written to WR0 at the same time that the pointer bits are written. To access the registers with addresses 15 through 8, the Point High command must accompany the pointer bits. This precludes concurrently issuing a command when pointing to these registers.

The register map for the Z85X30 is shown in Table 2-5. If, for some reason, the state of the pointer bits is unknown they may be reset to 0 by performing a read cycle with the D//C pin held Low. Once the pointer bits have been set, the desired channel is selected by the state of the A//B pin during the actual read or write of the desired register.

Table 2-5. Z85X30 Register Map

A/B	PNT2	PNT1	PNT0	WRITE	READ 8530 85C30/230 WR15 D2 = 0	85C30/230 WR15 D2=1	85230 WR15 D2=1 WR7' D6=1
0	0	0	0	WR0B	RR0B	RR0B	RR0B
0	0	0	1	WR1B	RR1B	RR1B	RR1B
0	0	1	0	WR2	RR2B	RR2B	RR2B
0	0	1	1	WR3B	RR3B	RR3B	RR3B
0	1	0	0	WR4B	(RR0B)	(RR0B)	(WR4B)
0	1	0	1	WR5B	(RR1B)	(RR1B)	(WR5B)
0	1	1	0	WR6B	(RR2B)	RR6B	RR6B
0	1	1	1	WR7B	(RR3B)	RR7B	RR7B
0	0	0	0	WR0A	RR0A	RR0A	RR0A
0	0	0	1	WR1A	RR1A	RR1A	RR1A
0	0	1	0	WR2	RR2A	RR2A	RR2A
0	0	1	1	WR3A	RR3A	RR3A	RR3A
0	1	0	0	WR4A	(RR0A)	(RR0A)	(WR4A)
0	1	0	1	WR5A	(RR1A)	(RR1A)	(WR5A)
0	1	1	0	WR6A	(RR2A)	RR6A	RR6A
0	1	1	1	WR7A	(RR3A)	RR7A	RR7A
With Point High Command							
1	0	0	0	WR8B	RR8B	RR8B	RR8B
1	0	0	1	WR9	(RR13B)	(RR13B)	(WR3B)
1	0	1	0	WR10B	RR10B	RR10B	RR10B
1	0	1	1	WR11B	(RR15B)	(RR15B)	(WR10B)
1	1	0	0	WR12B	RR12B	RR12B	RR12B
1	1	0	1	WR13B	RR13B	RR13B	RR13B
1	1	1	0	WR14B	RR14B	RR14B	(WR7'B)
1	1	1	1	WR15B	RR15B	RR15B	RR15B
1	0	0	0	WR8A	RR8A	RR8A	RR8A
1	0	0	1	WR9	(RR13A)	(RR13A)	(WR3A)
1	0	1	0	WR10A	RR10A	RR10A	RR10A
1	0	1	1	WR11A	(RR15A)	(RR15A)	(WR10A)
1	1	0	0	WR12A	RR12A	RR12A	RR12A
1	1	0	1	WR13A	RR13A	RR13A	RR13A
1	1	1	0	WR14A	RR14A	RR14A	(WR7'A)
1	1	1	1	WR15A	RR15A	RR15A	RR15A

Notes:
WR15 bit D2 enables status FIFO function. (Not available on NMOS)
WR7' bit D6 enables extend read function (Only on ESCC)

2.3.5 Z85C30 Register Enhancement

The Z85C30 has an enhancement to the NMOS Z8530 register set, which is the addition of a 10 x 19 SDLC Frame Status FIFO. When WR15 bit D2=1, the SDLC Frame Status FIFO is enabled, and it changes the functionality of RR6 and RR7. See Section 4.4.3 for more details on this feature.

2.3.6 Z85230 Register Enhancements

In addition to the 85C30 enhancements, the 85230 provides several enhancements to the SCC register set. These include the addition of Write Register 7 Prime (WR7'), the ability to read registers that are write-only in the SCC.

Write Register 7' is addressed by setting WR15, D0=1 and then addressing WR7. Figure 2-8 shows the register bit location of the six features enabled through this register. All writes to address seven are to WR7' when WR15 D0=1. Refer to Chapter 5 for detailed information on WR7'.

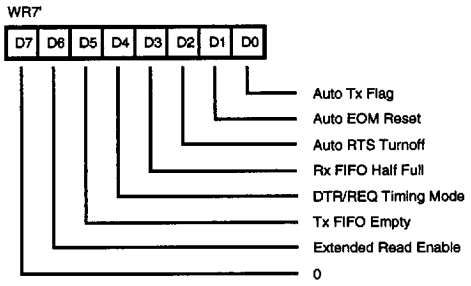

Figure 2-8. Write Register 7 Prime (WR7')

Setting WR7' bit D6=1 enables the extended read register capability. This allows the user to read the contents of WR3, WR4, WR5, WR7' and WR10 by reading RR9, RR4, RR5, RR14 and RR11, respectively. When WR7' D6=0, these write registers are write-only.

Table 2-6 shows what functions are enabled for the various combinations of register bit enables. See Table 2-5 for the register address map with only the SDLC FIFO enabled and with both the extended read and SDLC FIFO features enabled.

Table 2-6. Z85230 Register Enhancement Options

WR15 Bit D2	WR15 Bit D0	WR7' Bit D6	Functions Enabled
0	1	0	WR7' enabled only
0	1	1	WR7' with extended read enabled
1	0	X	10x19 SDLC FIFO enhancement enabled only
1	1	0	10x19 SDLC FIFO and WR7'
1	1	1	10x19 SDLC FIFO and WR7' with extended read enabled

2.3.7 Z85X30 Reset

The Z85X30 may be reset by either a hardware or software reset. Hardware reset occurs when /WR and /RD are both Low at the same time, which is normally an illegal condition. As long as both /WR and /RD are Low, the Z85X30 recognizes the reset condition. However, once this condition is removed, the reset condition is asserted internally

for an additional four to five PCLK cycles. During this time any attempt to access is ignored.

The Z85X30 has three software resets that are encoded into the command bits in WR9. There are two channel resets which only affect one channel in the device and some bits of the write registers. The command forces the same result as the hardware reset, the Z85X30 stretches the reset signal an additional four to five PCLK cycles beyond the ordinary valid access recovery time. The bits in WR9 may be written at the same time as the reset command because these bits are affected only by a hardware reset. The reset values of the various registers are shown in Table 2-7.

Table 2-7. Z85X30 Register Reset Values

	Hardware RESET								Channel RESET							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
WR0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
WR1	0	0	X	0	0	X	0	0	0	0	X	0	0	X	0	0
WR2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
WR3	X	X	X	X	X	X	X	0	X	X	X	X	X	X	X	0
WR4	X	X	X	X	X	1	X	X	X	X	X	X	X	1	X	X
WR5	0	X	X	0	0	0	0	X	0	X	X	0	0	0	0	X
WR6	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
WR7	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
WR7'*	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
WR9	1	1	0	0	0	0	X	X	X	X	0	X	X	X	X	X
WR10	0	0	0	0	0	0	0	0	0	X	X	0	0	0	0	0
WR11	0	0	0	0	1	0	0	0	X	X	X	X	X	X	X	X
WR12	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
WR13	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
WR14	X	X	1	1	0	0	0	0	X	X	1	0	0	0	X	X
WR15	1	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0
RR0	X	1	X	X	X	1	0	0	X	1	X	X	X	1	0	0
RR1	0	0	0	0	0	1	1	X	0	0	0	0	0	1	1	X
RR3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RR10	0	X	0	0	0	0	0	0	0	X	0	0	0	0	0	0

Notes:
* WR7' is only available on the ESCC

2.4 INTERFACE PROGRAMMING

The following subsections explain and illustrate all areas of interface programming.

2.4.1 I/O Programming Introduction

The SCC can work with three basic forms of I/O operations: polling, interrupts, and block transfer. All three I/O types involve register manipulation during initialization and data transfer. However, the interrupt mode also incorporates Z-Bus interrupt protocol for a fast and efficient data transfer.

Regardless of the version of the SCC, all communication modes can use a choice of polling, interrupt and block transfer. These modes are selected by the user to determine the proper hardware and software required to supply data at the rate required.

Note to ESCC Users: Those familiar with the NMOS/CMOS version will find the ESCC I/O operations very similar but should note the following differences: the addition of software acknowledge (which is available in the current version of the CMOS SCC, but not in NMOS); the /DTR//REQ pin can be programmed to be deasserted faster; and the programmability of the data interrupts to the FIFO fill level.

2.4.2 Polling

This is the simplest mode to implement. The software must poll the SCC to determine when data is to be input or output from the SCC. In this mode, MIE (WR9, bit 3), and Wait/DMA Request Enable (WR1, bit 7) are both reset to 0 to disable any interrupt or DMA requests. The software must then poll RR0 to determine the status of the receive buffer, transmit buffer and external status.

During a polling sequence, the status of Read Register 0 is examined in each channel. This register indicates whether or not a receive or transmit data transfer is needed and whether or not any special conditions are present, e.g., errors.

This method of I/O transfer avoids interrupts and, consequently, all interrupt functions should be disabled. With no interrupts enabled, this mode of operation must initiate a read cycle of Read Register 0 to detect an incoming character before jumping to a data handler routine.

2.4.3 Interrupts

Each of the SCC's two channels contain three sources of interrupts, making a total of six interrupt sources. These three sources of interrupts are: 1) Receiver, 2) Transmitter, and 3) External/Status conditions. In addition, there are several conditions that may cause these interrupts. Figure 2-9 shows the different conditions for each interrupt source and each is enabled under program control. Channel A has a higher priority than Channel B with Receive, Transmit, and External/Status Interrupts prioritized, respectively, within each channel as shown in Table 2-8. The SCC internally updates the interrupt status on every PCLK cycle in the Z85X30 and on /AS in the Z80X30.

Table 2-8. Interrupt Source Priority

Receive Channel A	Highest
Transmit Channel A	
External/Status Channel A	
Receive Channel B	
Transmit Channel B	
External/Status Channel B	Lowest

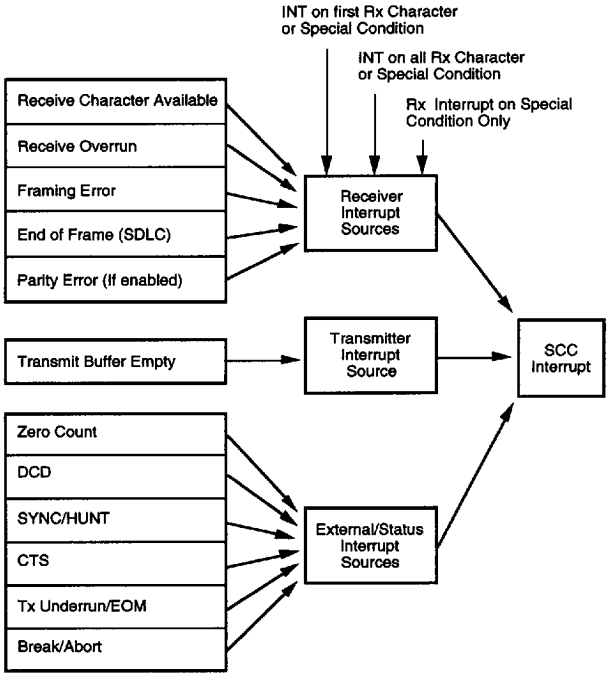

Figure 2-9. ESCC Interrupt Sources

ESCC:
The receive interrupt request is either caused by a receive character available or a special condition. When the receive character available interrupt is generated, it is dependent on WR7' bit D3. If WR7' D3=0, the receive character available interrupt is generated when one character is loaded into the FIFO and is ready to be read. If WR7' D3=1, the receive character available interrupt is generated when four bytes are available to be read in the receive data FIFO. The programmed value of WR7' D5 also affects when DMA requests are generated. See Section 2.5 for details.

Note: If the ESCC is used in SDLC mode, it enables the SDLC Status FIFO to affect how receive interrupts are generated. If this feature is used, read Section 4.4.3 on the SDLC Anti-Lock Feature.

The special conditions are Receive FIFO overrun, CRC/framing error, end of frame, and parity. If parity is included as a special condition, it is dependent on WR1 D2. The special condition status can be read from RR1.

On the NMOS/CMOS versions, set the IP bit whenever the transmit buffer becomes empty. This means that the transmit buffer was full before the transmit IP can be set.

ESCC:
The transmit interrupt request has only one source and is dependent on WR7' D5. If the IP bit WR7' D5=0, it is set when the transmit buffer becomes completely empty. If the IP bit WR7' D5=1, the transmit interrupt is generated when the exit location of the FIFO is empty. Note that in both cases the transmit interrupt is not set until after the first character is written to the ESCC.

For transmit Interrupt, see Section 2.4.8 for details.

The External/status interrupts have several sources which may be individually enabled in WR15. The sources are zero count, /DCD, Sync/Hunt, /CTS, transmitter underrun/EOM and Break/Abort.

2.4.4 Interrupt Control

In addition to the MIE bit that enables or disables all SCC interrupts, each source of interrupt in the SCC has three control/status bits associated with it. They are the Interrupt Enable (IE), Interrupt Pending (IP), and Interrupt-Under-Service (IUS). Figure 2-10 shows the SCC interrupt structure.

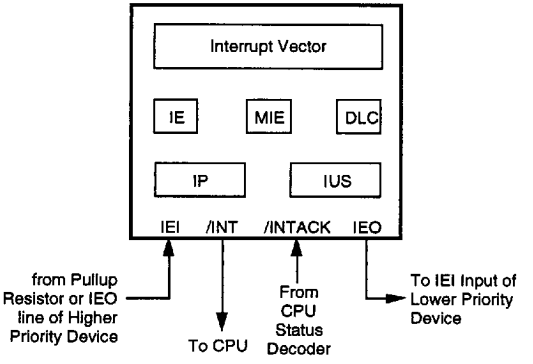

Figure 2-10. Peripheral Interrupt Structure

Figure 2-11 shows the internal priority resolution method to allow the highest priority interrupt to be serviced first. Lower priority devices on the external daisy chain can be prevented from requesting interrupts via the Disable Lower Chain bit in WR9 D2.

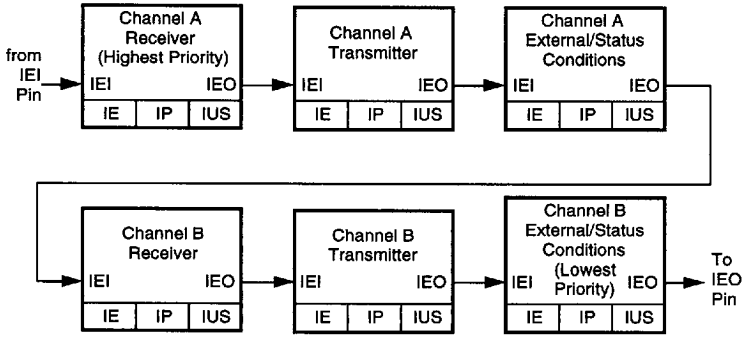

Figure 2-11. Internal Priority Resolution

2.4.4.1 Master Interrupt Enable Bit

The Master Interrupt Enable (MIE) bit, WR9 D3, must be set to enable the SCC to generate interrupts. The MIE bit should be set after intializing the SCC registers and enabling the individual interrupt enables. The SCC requests an interrupt by asserting the /INT pin Low from its open-drain state only upon detection that one of the enabled interrupt conditions has been detected.

2.4.4.2 Interrupt Enable Bit

The Interrupt Enable (IE) bits control interrupt requests from each interrupt source on the SCC. If the IE bit is set to 1 for an interrupt source, that source may generate an interrupt request, providing all of the necessary conditions are met. If the IE bit is reset, no interrupt request is generated by that source. The transmit interrupt IE bit is WR1 D1. The receive interrupt IE bits are WR1 D3 and D4. The external status interrupts are individually enabled in WR15 with the master external status interrupt enable in WR1 D0. Reminder: The MIE bit, WR9 D3, must be set for any interrupt to occur.

2.4.4.3 Interrupt Pending Bit

The Interrupt Pending (IP) bit for a given source of interrupt is set by the presence of an interrupt condition in the SCC. It is reset directly by the processor, or indirectly by some action that the processor may take. If the corresponding IE bit is not set, the IP for that source of interrupt will never be set. The IP bits in the SCC are read only via RR3 as shown in Figure 2-12.

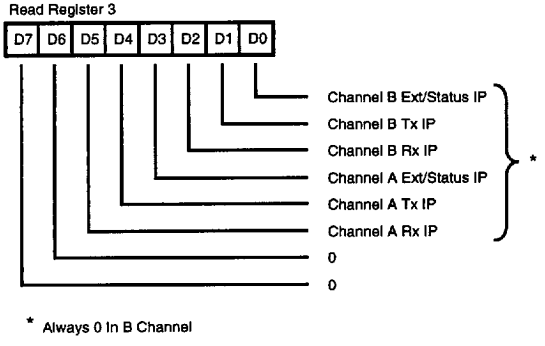

Figure 2-12. RR3 Interrupt Pending Bits

2.4.4.4 Interrupt-Under-Service Bit

The Interrupt-Under-Service (IUS) bits are completely hidden from the processor. An IUS bit is set during an interrupt acknowledge cycle for the highest priority IP. On the CMOS or ESCC the IUS bits can be set by either a hardware acknowledge cycle with the /INTACK pin or through software if WR9 D5=1 and then reading RR2.

The IUS bits control the operation of internal and external daisy-chain interrupts. The internal daisy chain links the six sources of interrupt in a fixed order, chaining the IUS bit of each source. If an internal IUS bit is set, all lower priority interrupt requests are masked off; during an interrupt acknowledge cycle the IP bits are also gated into the daisy chain. This ensures that the highest priority IP selected has its IUS bit set. At the end of an interrupt service routine, the processor must issue a Reset Highest IUS command in WR0 to re-enable lower priority interrupts. This is the only way, short of a software or hardware reset, that an IUS bit may be reset.

Note: It is not necessary to issue the Reset Highest IUS command in the interrupt service routine, since the IUS bits can only be set by an interrupt acknowledge if no hardware acknowledge or software acknowledge cycle (not with NMOS) is executed. The only exception is when the SDLC Frame Status FIFO (not with NMOS) is enabled and "receive interrupt on special condition only" is used. See section 4.4.3 for more details on this mode.

2.4.4.5 Disable Lower Chain Bit

The Disable Lower Chain (DLC) bit in WR9 (D2) is used to disable all peripherals in a lower position on the external daisy chain. If WR9 D2=1, the IEO pin is driven Low and prevents lower priority devices from generating an interrupt request. Note that the IUS bit, when set, will have the same effect, but is not controllable through software.

2.4.5 Daisy-Chain Resolution

The six sources of interrupt in the SCC are prioritized in a fixed order via a daisy chain; provision is made, via the IEI and IEO pins, for use of an external daisy chain as well. All Channel A interrupts are higher priority than any Channel B interrupts, with the receiver, transmitter, and External/Status interrupts prioritized in that order within each channel. The SCC requests an interrupt by pulling the /INT pin Low from its open-drain state. This is controlled by the IP bits and the IEI input, among other things. A flowchart of the interrupt sequence for the SCC is shown in Figure 2-13.

The internal daisy chain links the six sources of interrupt in a fixed order, chaining the IUS bits for each source. While an IUS bit is set, all lower priority interrupt requests are masked off, thus preventing lower priority interrupts, but still allowing higher priority interrupts to occur. Also, during an interrupt acknowledge cycle the IP bits are gated into the daisy chain. This insures that the highest priority IP is selected to set IUS. The internal daisy chain may be controlled by the MIE bit in WR9. This bit, when reset, has the same effect as pulling the IEI pin Low, thus disabling all interrupt requests.

2.4.5.1 External Daisy-Chain Operations

The SCC generates an interrupt request by pulling /INT Low, but only if such interrupt requests are enabled (IE is 1, MIE is 1) and all of the following conditions occur:

- IP is set without a higher priority IUS being set
- No higher priority IUS is being set
- No higher priority interrupt is being serviced (IEI is High)
- No interrupt acknowledge transaction is taking place

IEO is not pulled Low by the SCC at this time, but instead continues to follow IEI until an interrupt acknowledge transaction occurs. Some time after /INT has been pulled Low, the processor initiates an Interrupt Acknowledge transaction. Between the time the SCC recognizes that an Interrupt Acknowledge cycle is in progress and the time during the acknowledge that the processor requests an interrupt vector, the IEI/IEO daisy chain settles. Any peripheral in the daisy chain having an Interrupt Pending (IP is 1) or an Interrupt-Under-Service (IUS is 1) holds its IEO line Low and all others make IEO follow IEI.

When the processor requests an interrupt vector, only the highest priority interrupt source with a pending interrupt (IP is 1) has its IEI input High, its IE bit set to 1, and its IUS bit set to 0. This is the interrupt source being acknowledged, and at this point it sets its IUS bit to 1. If its NV bit is 0, the SCC identifies itself by placing the interrupt vector from WR2 on the data bus. If the NV bit is 1, the SCC data bus remains floating, allowing external logic to supply a vector. If the VIS bit in the SCC is 1, the vector also contains status information, encoded as shown in Table 2-9, which further describes the nature of the SCC interrupt.

Table 2-9. Interrupt Vector Modification

V3 V4	V2 V5	V1 V6	Status High/Status Low = 0 Status High/Status Low = 1
0	0	0	Ch B Transmit Buffer Empty
0	0	1	Ch B External/Status Change
0	1	0	Ch B Receive Character Avail
0	1	1	Ch B Special Receive Condition
1	0	0	Ch A Transmit Buffer Empty
1	0	1	Ch A External/Status Change
1	1	0	Ch A Receive Character Avail
1	1	1	Ch A Special Receive Condition

If the VIS bit is 0, the vector held in WR2 is returned without modification. If the SCC is programmed to include status information in the vector, this status may be encoded and placed in either bits 1-3 or in bits 4-6. This operation is selected by programming the Status High/Status Low bit in WR9. At the end of the interrupt service routine, the processor should issue the Reset Highest IUS command to unlock the daisy chain and allow lower priority interrupt requests. The IP is reset during the interrupt service routine, either directly by command or indirectly through some action taken by the processor. The external daisy chain may be controlled by the DLC bit in WR9. This bit, when set, forces IEO Low, disabling all lower priority devices.

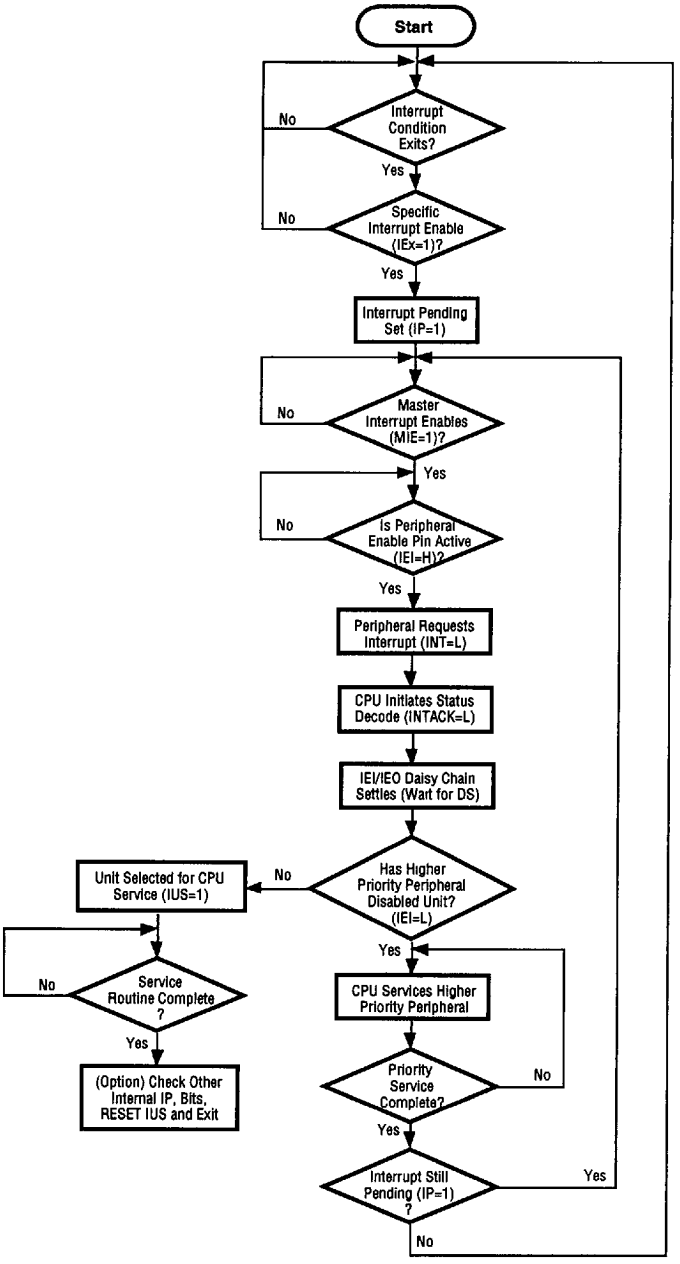

Figure 2-13. Interrupt Flow Chart (for each interrupt source).

2.4.6 Interrupt Acknowledge

The SCC is flexible with its interrupt method. The interrupt may be acknowledged with a vector transferred, acknowledged without a vector, or not acknowledged at all.

2.4.6.1 Interrupt Without Acknowledge

In this mode, the Interrupt Acknowledge signal does not have to be generated. This allows a simpler hardware design that does not have to meet the interrupt acknowledge timing. Soon after the INT goes active, the interrupt controller jumps to the interrupt routine. In the interrupt routine, the code must read RR2 from Channel B to read the vector including status. When the vector is read from Channel B, it always includes the status regardless of the VIS bit (WR9 bit 0). The status given will decode the highest priority interrupt pending at the time it is read. The vector is not latched so that the next read could produce a different vector if another interrupt occurs. The register is disabled from change during the read operation to prevent an error if a higher interrupt occurs exactly during the read operation.

Once the status is read, the interrupt routine must decode the interrupt pending, and clear the condition. Removing the interrupt condition clears the IP and brings /INT inactive (open drain), as long as there are no other IP bits set. For example, writing a character to the transmit buffer clears the transmit buffer empty IP.

When the interrupt IP, decoded from the status, is cleared, RR2 can be read again. This allows the interrupt routine to clear all of the IP's within one interrupt request to the CPU.

2.4.6.2 Interrupt With Acknowledge

After the SCC brings /INT active, the CPU can respond with a hardware acknowledge cycle by bringing /INTACK active. After enough time has elapsed to allow the daisy chain to settle (see AC Spec #38), the SCC sets the IUS bit for the highest priority IP. If the No Vector bit is reset (WR9 D1=0), the SCC then places the interrupt vector on the data bus during a read. To speed the interrupt response time, the SCC can modify 3 bits in the vector to indicate the source of the interrupt. To include the status, the VIS bit, WR9 D0, is set. The service routine must then clear the interrupting condition. For example, writing a character to the transmit buffer clears the transmit buffer empty IP. After the interrupting condition is cleared, the routine can read RR3 to determine if any other IP's are set and take the appropriate action to clear them. At the end of the interrupt routine, a Reset IUS command (WR0) is issued to unlock the daisy chain and allow lower-priority interrupt requests. This is the only way, short of a software or hardware reset, that an IUS bit is reset.

If the No Vector bit is set (WR9 D1=1), the SCC will not place the vector on the data bus. An interrupt controller must then vector the code to the interrupt routine. The interrupt routine reads RR2 from Channel B to read the status. This is similar to an interrupt without an acknowledge, except the IUS is set and the vector will not change until the Reset IUS command in RR0 is issued.

2.4.6.3 Software Interrupt Acknowledge (CMOS/ESCC)

An interrupt acknowledge cycle can be done in software for those applications which use an external interrupt controller or which cannot generate the /INTACK signal with the required timing. If WR9 D5 is set, reading register two, RR2, results in an interrupt acknowledge cycle to be executed internally. Like a hardware INTACK cycle, a software acknowledge causes the /INT pin to return High, the IEO pin to go Low and the IUS latch to be set for the highest priority interrupt pending.

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

2.4.7 The Receiver Interrupt

The sources of receive interrupts consist of Receive Character Available and Special Receive Condition. The Special Receive Condition can be subdivided into Receive Overrun, Framing Error (Asynchronous) or End of Frame (SDLC). In addition, a parity error can be a special receive condition by programming.

As shown in Figure 2-14, Receive Interrupt mode is controlled by three bits in WR1. Two of these bits, D4 and D3, select the interrupt mode; the third bit, D2, is a modifier for the various modes. On the ESCC, WR7' bit D2 affects the receiver interrupt operation mode as well. If the interrupt capability of the receiver in the SCC is not required, polling may be used. This is selected by disabling receive interrupts and polling the Receiver Character Available bit in RR0. When this bit indicates that a received character has reached the exit location (CPU side) of the FIFO, the status in RR1 should be checked and then the data should be read. If status is checked, it must be done before the data is read, because the act of reading the data pops both the data and error FIFOs. Another way of polling SCC is to enable one of the interrupt modes and then reset the MIE bit in WR9. The processor may then poll the IP bits in RR3A to determine when receive characters are available.

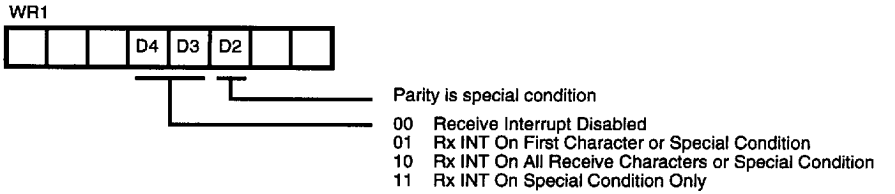

Figure 2-14. Write Register 1 Receive Interrupt Mode Control

2.4.7.1 Receive Interrupt on the ESCC

On the ESCC, one other bit, WR7' bit D2, also affects the interrupt operation.

WR7' D3=0, a receive interrupt is generated when one byte is available in the FIFO. This mode is selected after reset and maintains compatibility with the SCC. Systems with a long interrupt response time can use this mode to generate an interrupt when one byte is received, but still allow up to seven more bytes to be received without an overrun error. By polling the Receive Character Available bit, RR0 D0, and reading all available data to empty the FIFO before exiting the interrupt service routine, the frequency of interrupts can be minimized.

WR7' D3=1, the ESCC generates an interrupt when there are four bytes in the Receive FIFO or when a special condition is received. By setting this bit, the ESCC generates a receive interrupt when four bytes are available to read from the FIFO. This allows the CPU not to be interrupted until at least four bytes can be read from the FIFO, thereby minimizing the frequency of receive interrupts. If four or more bytes remain in the FIFO when the Reset Highest IUS command is issued at the end of the service routine, another receive interrupt is generated.

When a special receive condition is detected in the top four bytes, a special receive condition interrupt is generated immediately. This feature is intended to be used with the Interrupt On All Receive Characters and Special Condition mode. This is especially useful in SDLC mode because the characters are contiguous and the reception of the closing flag immediately generates a special receive interrupt. The generation of receive interrupts is described in the following two cases:

Case 1: Four Bytes Received with No Errors. A receive character available interrupt is triggered when the four bytes in receive data FIFO (from the exit side) are full and no special conditions have been detected. Therefore, the interrupt service routine can read four bytes from the data FIFO without having to read RR1 to check for error conditions.

Case 2: Data Received with Error Conditions. When any of the four bytes from the exit side in the receive error FIFO indicate an error has been detected, a Special Receive condition interrupt is triggered without waiting for the byte to reach the top of the FIFO. In this case, the interrupt service routine must read RR1 first before reading each data byte to determine which byte has the special receive condition and then take the appropriate action. Since, in this mode, the status must be checked before the data is read, the data FIFO is not locked and the Error Reset command is not necessary.

Note: The above cases assume that the receive IUS bit is reset to zero in order for an interrupt to be generated.

WR7' D3 should be written zero when using Interrupt on First Character and Special Condition or Interrupt on Special Condition Only. See the description for Interrupt on All Characters or Special Condition mode for more details on this feature.

Note: The Receive Character Available Status bit, RR0 D0, indicates if at least one byte is available in the Receive FIFO, independent of WR7' D3. Therefore, this bit can be polled at any time for status if there is data in the Receive FIFO.

2.4.7.2 Receive Interrupts Disabled

This mode prevents the receiver from requesting an interrupt. It is used in a polled environment where either the status bits in RR0 or the modified vector in RR2 (Channel B) is read. Although the receiver interrupts are disabled, the interrupt logic can still be used to provide status.

When these bits indicate that a received character has reached the exit location of the FIFO, the status in RR1 should be checked and then the data should be read. If status is to be checked, it must be done before the data is read, because the act of reading the data pops both the data and error FIFOs.

2.4.7.3 Receive Interrupt on First Character or Special Condition

This mode is designed for use with DMA transfers of the receive characters. The processor is interrupted when the SCC receives the first character of a block of data. It reads the character and then turns control over to a DMA device to transfer the remaining characters. After this mode is selected, the first character received, or the first character already stored in the FIFO, sets the receiver IP. This IP is reset when this character is removed from the SCC.

No further receive interrupts occur until the processor issues an Enable Interrupt on Next Receive Character command in WR0 or until a special receive condition occurs. The correct sequence of events when using this mode is to first select the mode and wait for the receive character available interrupt. When the interrupt occurs, the processor should read the character and then enable the DMA to transfer the remaining characters.

ESCC:
WR7' bit D3 should be reset to zero in this mode.

A special receive condition interrupt may occur any time after the first character is received, but is guaranteed to occur after the character having the special condition has been read. The status is not lost in this case, however, because the FIFO is locked by the special condition. In the interrupt service routine, the processor should read RR1 to obtain the status, and may read the data again if necessary. The FIFO is unlocked by issuing an Error Reset command in WR0. If the special condition was End-of-Frame, the processor should now issue the Enable Interrupt on Next Receive Character command to prepare for the next frame. The first character interrupt and special condition interrupt are distinguished by the status included in the interrupt vector. In all other respects they are identical, including sharing the IP and IUS bits.

2.4.7.4 Interrupt on All Receive Characters or Special Condition

This mode is designed for an interrupt driven system. In this mode, the NMOS/CMOS version and the ESCC with WR7' D3=0 sets the receive IP when a received character is shifted into the exit location of the FIFO. This occurs whether or not it has a special receive condition. This includes characters already in the FIFO when this mode is selected. In this mode of operation the IP is reset when the character is removed from the FIFO, so if the processor requires status for any characters, this status must be read before the data is removed from the FIFO.

On the ESCC with D3=1, four bytes are accumulated in the Receive FIFO before an interrupt is generated (IP is set), and reset when the number of the characters in the FIFO is less than four.

The special receive conditions are identical to those previously mentioned, and as before, the only difference between a "receive character available" interrupt and a "special receive condition" interrupt is the status encoded in the vector. In this mode a special receive condition does not lock the receive data FIFO so that the service routine must read the status in RR1 before reading the data.

At moderate to high data rates where the interrupt overhead is significant, time can usually be saved by checking for another character before exiting the service routine. This technique eliminates the interrupt acknowledge and the status processing, saving time, but care must be exercised because this receive character must be checked for special receive conditions before it is removed from the SCC.

2.4.7.5 Receive Interrupt on Special Conditions

This mode is designed for use when a DMA transfers all receive characters between memory and the SCC. In this mode, only receive characters with special conditions will cause the receive IP to be set. All other characters are assumed to be transferred via DMA. No special initialization sequence is needed in this mode. Usually, the DMA is initialized and enabled, then this mode is selected in the SCC. A special receive condition interrupt may occur at any time after this mode is selected, but the logic guarantees that the interrupt will not occur until after the character with the special condition has been read from the SCC. The special condition locks the FIFO so that the status is valid when read in the interrupt service routine, and it guarantees that the DMA will not transfer any characters until the special condition has been serviced.

In the service routine, the processor should read RR1 to obtain the status and unlock the FIFO by issuing an Error Reset command. DMA transfer of the receive characters then resumes. Figure 2-15 shows the special conditions interrupt service routine.

Note: On the CMOS and ESCC, if the SDLC Frame Status FIFO is being used, please refer to Section 4.4.3 on the FIFO anti-lock feature.

Note: Special Receive Condition interrupts are generated *after* the character is read from the FIFO, *not* when the special condition is first detected. This is done so that when using receive interrupt on first or Special Condition or Special Condition Only, data is directly read out of the data FIFO without checking the status first. If a special condition interrupted the CPU when first detected, it would be necessary to read RR1 before each byte in the FIFO to determine which byte had the special condition. Therefore, by not generating the interrupt until after the byte has been read and then locking the FIFO, only one status read is necessary. A DMA can be used to do all data transfers (otherwise, it would be necessary to disable the DMA to allow the CPU to read the status on each byte). Consequently, since the special condition locks the FIFO to preserve the status, it is necessary to issue the Error Reset command to unlock it. Only the exit location of the FIFO is locked allowing more data to be received into the other bytes of the Receive FIFO.

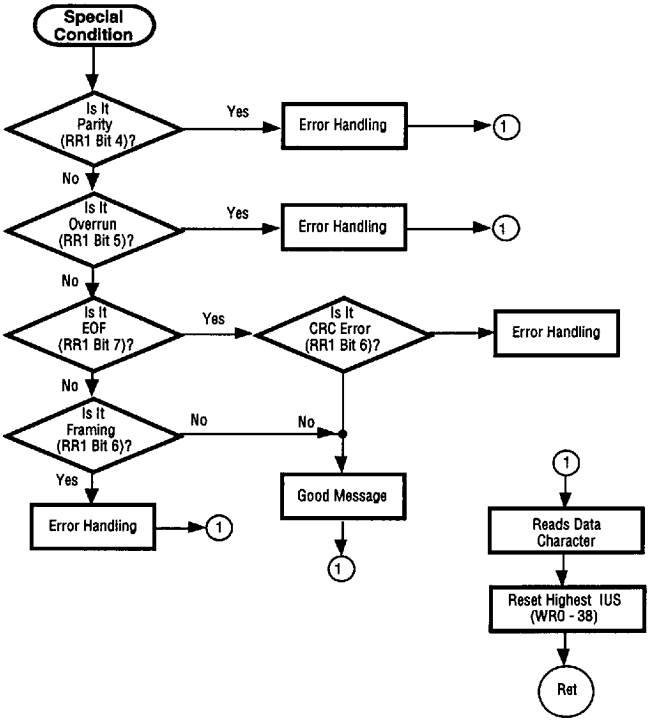

Figure 2-15. Special Conditions Interrupt Service Flow

2.4.8 Transmit Interrupts and Transmit Buffer Empty Bit

Transmit interrupts are controlled by Transmit Interrupt Enable bit (D1) in WR1. If the interrupt capabilities of the SCC are not required, polling may be used. This is selected by disabling transmit interrupts and polling the Transmit Buffer Empty bit (TBE) in RR0. When the TBE bit is set, a character may be written to the SCC without fear of writing over previous data. Another way of polling the SCC is to enable transmit interrupts and then reset Master Interrupt Enable bit (MIE) in WR9. The processor may then poll the IP bits in RR3A to determine when the transmit buffer is empty. Transmit interrupts should also be disabled in the case of DMA transfer of the transmitted data.

Because the depth of the transmitter buffer is different between the NMOS/CMOS version of the SCC and ESCC, generation of the transmit interrupt is slightly different. The following subsections describe transmit interrupts.

Note: For all interrupt sources, the Master Interrupt Enable (MIE) bit, WR9 bit D3, must be set for the device to generate a transmit interrupt.

2.4.8.1 Transmit Interrupts and Transmit Buffer Empty Bit on the NMOS/CMOS

The NMOS/CMOS version of the SCC only has a one byte deep transmit buffer. The status of the transmit buffer can be determined through TBE bit in RR0, bit D2, which shows whether the transmit buffer is empty or not. After a hardware reset (including a hardware reset by software), or a channel reset, this bit is set to 1.

While transmit interrupts are enabled, the NMOS/CMOS version sets the Transmit Interrupt Pending (TxIP) bit whenever the transmit buffer becomes empty. This means that the transmit buffer must be full before the TxIP can be set. Thus, when transmit interrupts are first enabled, the TxIP will not be set until after the first character is written to the NMOS/CMOS. In synchronous modes, one other condition can cause the TxIP to be set. This occurs at the end of a transmission after the CRC is sent. When the last bit of the CRC has cleared the Transmit Shift Register and the flag or sync character is loaded into the Transmit Shift Register, the NMOS/CMOS version sets the TxIP and TBE bit. Data for a second frame or block transmission may be written at this time.

The TxIP is reset either by writing data to the transmit buffer or by issuing the Reset Tx Int command in WR0. Ordinarily, the response to a transmit interrupt is to write more data to the device; however, the Reset Tx Int command should be issued in lieu of data at the end of a frame or a block of data where the CRC is to be sent next.

Note: A transmit interrupt may indicate that the packet has terminated illegally, with the CRC byte(s) overwritten by the data. If the transmit interrupt occurs after the first CRC byte is loaded into the Transmit Shift Register, but before the last bit of the second CRC byte has cleared the Transmit Shift Register, then data was written while the CRC was being sent.

2.4.8.2 Transmit Interrupt and Transmit Buffer Empty bit on the ESCC

The ESCC has a four byte deep Transmit FIFO where the NMOS/CMOS version is just one byte deep. Consequently, the generation of transmit interrupts is slightly different from the NMOS/CMOS version. The ESCC has two modes of transmit interrupt generation, which are programmed by D5 of WR7'. One transmit mode generates interrupts when the entry location (the location the CPU writes data) of the Transmit FIFO is empty, and the other generates interrupts when the FIFO is completely empty. This allows the ESCC response to be tailored to system requirements for the frequency of interrupts and the interrupt response time.

When WR7' D5=1 (the default case), the ESCC generates a transmit interrupt when the Transmit FIFO is completely empty. The transmit data interrupt is generated when the exit location of the Transmit FIFO loads into the Transmit Shift Register and the FIFO becomes empty. WR7' bit D5 is set to one by a hardware or channel reset. This mode minimizes the frequency of transmit interrupts by writing four bytes to the Transmit FIFO on each entry to the interrupt service routine. The TBE bit, RR0 bit D2, is set when at least one byte can be written to the Transmit FIFO. Therefore, this bit may be polled at any time to determine if a byte can be written.

When WR7' D5=0, the TxIP bit is set when the entry location of the Transmit FIFO is empty. This mode provides a system which cannot respond to an interrupt in one character transmission time (a long latency time to respond before the transmitter underruns.)

Note: When WR7' D5=0, TBE bit and transmit interrupt are reset momentarily when data is loaded into the entry location of the Transmit FIFO. Transmit interrupt is not generated when the entry location of the Transmit FIFO is filled. The TBE bit is set and transmit interrupt is generated again when the data is pushed down the FIFO and the entry location becomes empty (approximately one PCLK time).

Note: When WR7' D5=0, only one byte is written to the FIFO and three or fewer bytes are in the FIFO, the ESCC generates multiple interrupts until the FIFO is filled. Therefore, by polling the TBE bit (RR0 D2) after writing each byte, multiple interrupts to fill the FIFO are avoided.

While transmit interrupts are enabled, the ESCC sets the TxIP when the transmit buffer reaches the condition programmed in WR7' bit D5. This means that the transmit buffer must have been written before the TxIP is set. Thus, when transmit interrupts are first enabled, the transmit IP is not set until the programmed interrupting condition is met.

The Tx IP is reset either by writing data to the transmit buffer or by issuing the Reset Tx Int Pending command in WR0. Ordinarily, the response to a transmit interrupt is to write more data to the ESCC; however, if there is no more data to be transmitted at that time, it is the end of the frame. The Reset Tx Int Pending command is used to reset the TxIP and clear the interrupt. For example, at the end of a frame or block of data where the CRC is to be sent next, the Reset Tx Int Pending command should be issued after the last byte of data has been written to the ESCC.

On the ESCC, TBE bit is not directly related to the transmit interrupt status or the state of WR7' bit D5, but it shows the status of the exit location of the Transmit FIFO. This indicates that more characters can be written to the Transmit FIFO without being overwritten, which allows the interrupt/polling handling routine to write more than one byte to fill up the FIFO.

In synchronous modes, one other condition can cause the TxIP to be set. This occurs at the end of a transmission after the CRC is sent. When the last bit of the CRC has cleared the Transmit Shift Register and the flag or sync character is loaded into the Transmit Shift Register, the ESCC sets the TxIP. Data for the new frame or block to be transmitted may be written at this time. In this particular case, the Transmit Buffer Empty bit in RR0 and the TxIP are set.

An enhancement to the ESCC from the NMOS/CMOS version is that the CRC has priority over the data, where on the NMOS/CMOS version data has priority over the CRC bytes. This means on the ESCC the CRC bytes are guaranteed to be sent, even if the data for the next packet has written before the second transmit interrupt, but after the EOM/Underrun condition exists. This helps to increase the system throughput because there is no waiting for the second transmit interrupt. On the NMOS/CMOS version, if the data is written while the CRC is sent, CRC byte(s) are replaced with the flag/sync pattern followed by the data.

Another enhancement of the ESCC is that it latches the transmit interrupt because the CRC is loaded into the Transmit Shift Register even if the transmit interrupt, due to the last data byte, is not yet reset. Therefore, the end of a synchronous frame is guaranteed to generate two transmit interrupts even if a Reset Tx Int Pending command for the data created interrupt is issued after (Time "A" in Figure 2-16) the CRC interrupt had occurred. In this case, two reset Tx Int Pending commands are required. The TxIP is latched if the EOM latch has been reset before the end of the frame.

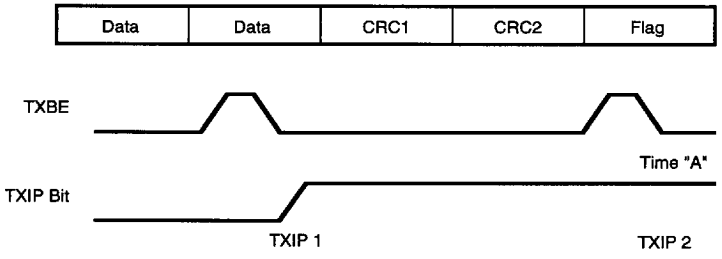

Figure 2-16. TxIP Latching on the ESCC

2.4.8.3 Transmit Interrupt and Tx Underrun/EOM bit in synchronous modes

As described in the section above, the behavior of the NMOS/CMOS version and the ESCC is slightly different, particularly at the end of packet sending. On the NMOS/CMOS version, the data has higher priority over CRC data; writing data before this interrupt would terminate the packet illegally. In this case, the CRC byte(s) are replaced with a Flag or Sync pattern, followed by the data written. On the ESCC, the CRC has priority over the data. That means after the reception of the Underrun/EOM (End Of Message) interrupt, it accepts the data for the next packet without collapsing the packet. On the ESCC, if data was written during the time period described above, the TBE bit (bit D2 of RR0) will *not* be set even if the second TxIP is guaranteed to set when the flag/sync pattern was loaded into the Transmit Shift Register, as mentioned above (Figures 2-17 and 18). Hence, on the ESCC, there is no need to wait for the second TxIP bit to set before writing data for the next packet and reducing the overhead.

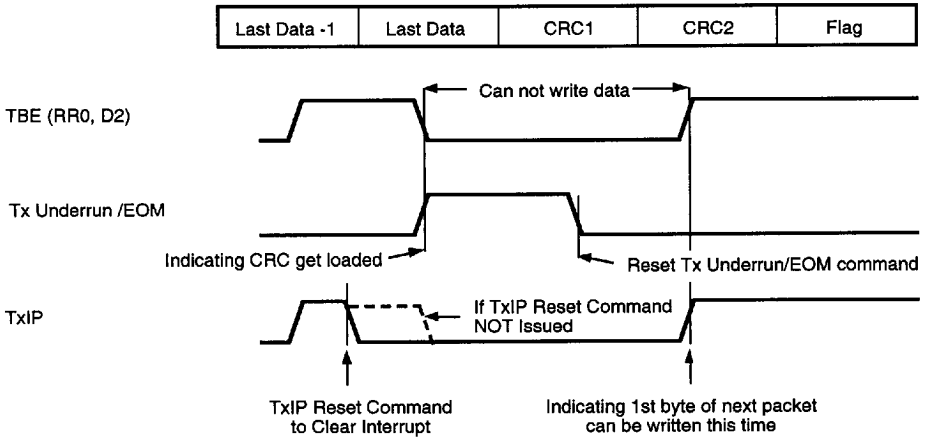

Figure 2-17. Operation of TBE, Tx Underrun/EOM and TxIP on NMOS/CMOS.

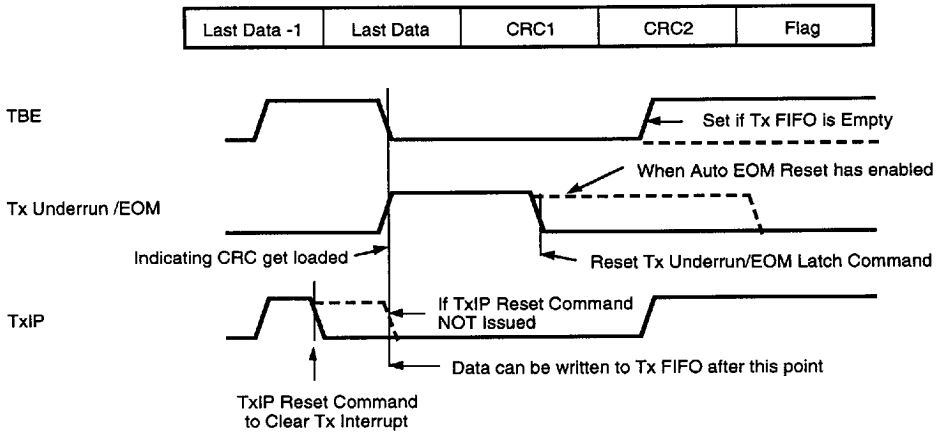

Figure 2-18. Operation of TBE, Tx Underrun/EOM and TxIP on ESCC.

An example flowchart for processing an end of packet is shown in Figure 2-19. The chart includes the differences in processing between the ESCC and NMOS/CMOS version. In this chart, Tx IP and Underrun/EOM INT can be processed by interrupts or by polling the registers. Note that this flowchart does not have the procedures for interrupt handling, such as saving/restoring of registers to be used in the ISR (Interrupt Service Routine), Reset IUS command, or return from interrupt sequence.

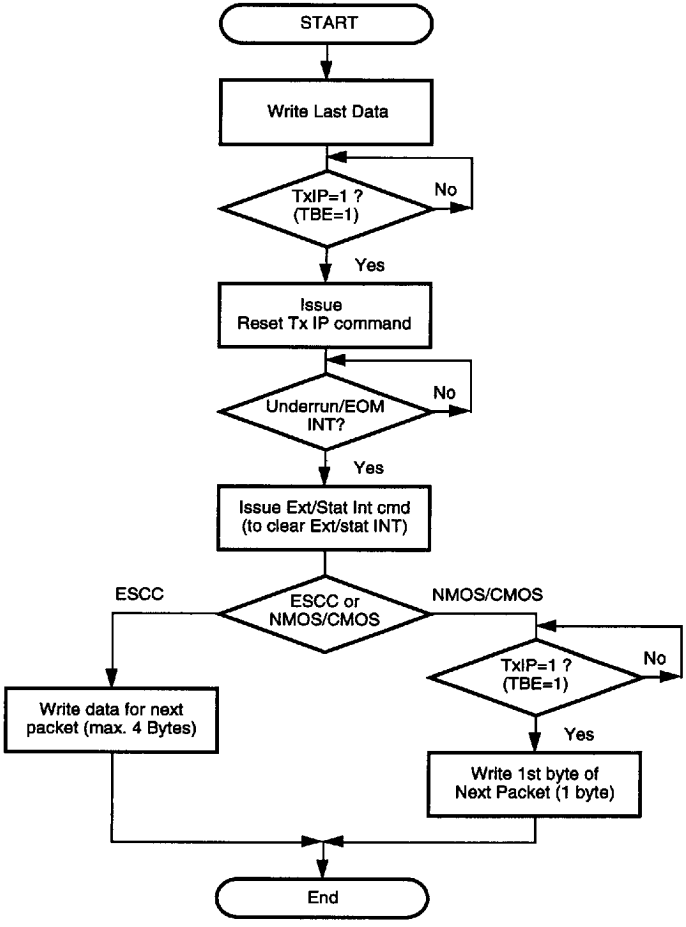

Figure 2-19. Flowchart example of processing an end of packet

2.4.9 External/Status Interrupts

Each channel has six external/status interrupt conditions: BRG Zero Count, Data Carrier Detect, Sync/Hunt, Clear to Send, Tx Underrun/EOM, and Break/Abort. The master enable for external/status interrupts is D0 of WR1, and the individual enable bits are in WR15. Individual enable bits control whether or not a latch is present in the path from the source of the interrupt to the corresponding status bit in RR0. If the individual enable is set to 0, then RR0 reflects the current unlatched status, and if the individual enable is set to 1, then RR0 reflects the latched status.

The latches for the external/status interrupts are not independent. Rather, they all close at the same time as a result of a state change in one of the sources of enabled external/status interrupts. This is shown schematically in Figure 2-20.

The External/Status IP is set by the closing of the latches and remains set as long as they are closed. In order to determine which condition(s) require service when an external/status interrupt is received, the processor should keep an image of RR0 in memory and update this image each time it executes the external/status service routine.

Thus, a read of RR0 returns the current status for any bits whose individual enable is 0, and either the current state or the latched state of the remainder of the bits. To guarantee the current status, the processor should issue a Reset External/Status Interrupts command in WR0 to open the latches. The External/Status IP is set by the closing of the latches and remains set as long as they are closed. If the master enable for the External/Status interrupts is not set, the IP is never set, even though the latches may be present in the signal paths and working as described.

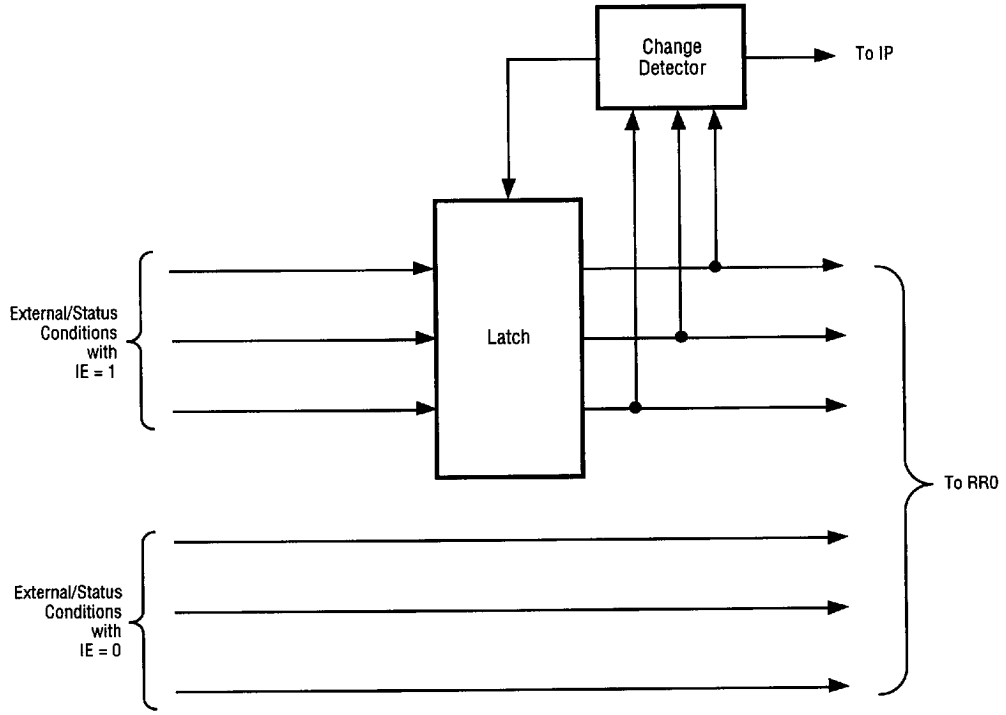

Figure 2-20. RR0 External/Status Interrupt Operation

Because the latches close on the current status, but give no indication of change, the processor must maintain a copy of RR0 in memory. When the SCC generates an External/Status Interrupt, the processor should read RR0 and determine which condition changed state and take appropriate action. The copy of RR0 in memory is then updated and the Reset External/Status Interrupt command issued. Care must be taken in writing the interrupt service routine for the External/Status interrupts because it is possible for more than one status condition to change state at the same time. All of the latch bits in RR0 should be compared to the copy of RR0 in memory. If none have changed and the ZC interrupt is enabled, the Zero Count condition caused the interrupt.

On the ESCC, the contents of RR0 are latched while reading this register. The ESCC prevents the contents of RR0 from changing while the read cycle is active. On the NMOS/CMOS version, it is possible for the status of RR0 to change while a read is in progress, so it is necessary to read RR0 twice to detect changes that otherwise may be missed. The contents of RR0 are latched on the falling edge of /RD and are updated after the rising edge of /RD.

The operation of the individual enable bits in WR15 for each of the six sources of External/Status interrupts is identical, but subtle differences exist in the operation of each source of interrupt. The six sources are Break/Abort, Underrun/EOM, CTS, DCD, Sync/Hunt and Zero Count. The Break/Abort, Underrun/EOM, and Zero Count conditions are internal to the SCC, while Sync/Hunt may be internal or external, and CTS and DCD are purely external signals. In the following discussions, each source is assumed to be enabled so that the latches are present and the External/Status interrupts are enabled as a whole. Recall that the External/Status IP is set while the latches are closed and that the state of the signal is reflected immediately in RR0 if the latches are not present.

2.4.9.1 Break/Abort

The Break/Abort status is used in asynchronous and SDLC modes, but is always 0 in synchronous modes other than SDLC. In asynchronous modes, this bit is set when a break sequence (null character plus framing error) is detected in the receive data stream, and remains set as long as 0s continue to be received. This bit is reset when a 1 is received. A single null character is left in the Receive FIFO each time that the break condition is terminated. This character should be read and discarded.

In SDLC mode, this bit is set by the detection of an abort sequence which is seven or more contiguous 1s in the receive data stream. The bit is reset when a 0 is received. A received abort forces the receiver into Hunt, which is also an external/status condition. Though these two bits change state at roughly the same time, one or two External/Status Interrupts may be generated as a result. The Break/Abort bit is unique in that both transitions are guaranteed to cause the latches to close, even if another External/Status interrupt is pending at the time these transitions occur. This guarantees that a break or abort will be caught. This bit is undetermined after reset.

2.4.9.2 Transmit Underrun/EOM

The Transmit Underrun/EOM bit is used in synchronous modes to control the transmission of the CRC. This bit is reset by issuing the Reset Transmit Underrun/EOM command in WR0. However, this transition does not cause the latches to close; this occurs only when the bit is set. To inform the processor of this fact, the SCC sets this bit when the CRC is loaded into the Transmit Shift Register. This bit is also set if the processor issues the Send Abort command in WR0. This bit is always set in Asynchronous mode.

ESCC:
The ESCC has been modified so that in SDLC mode this interrupt indicates when more data can be written to the Transmit FIFO. When this interrupt is used in this way, the Automatic SDLC Flag Transmission feature must be enabled (WR7' D0=1). On the NMOS/CMOS version, it is necessary to wait for the transmit buffer empty interrupt that is generated when the CRC transmission is completed before more data is written to the transmit buffer. However, on the ESCC, the Transmit Underrun/EOM interrupt can be used to signal when data for a subsequent frame can be written to the Transmit FIFO which more easily supports the transmission of back to back frames.

2.4.9.3 CTS/DCD

The CTS bit reports the state of the /CTS input, and the DCD bit reports the status of the /DCD input. Both bits latch on either input transition. In both cases, after the Reset External/Status Interrupt command is issued, if the latches are closed, they remain closed if there is any odd number of transitions on an input; they open if there is an even number of transitions on the input.

2.4.9.4 Zero Count

The Zero Count bit is set when the counter in the baud rate generator reaches a count of 0 and is reset when the counter is reloaded. The latches are closed only when this bit is set to 1. The status in RR0 always reflects the current status. While the Zero count IE bit in WR15 is reset, this bit is forced to 0.

2.4.9.5 Sync/Hunt

There are a variety of ways in which the Sync/Hunt may be set and reset, depending on the SCC's mode of operation. In the Asynchronous mode this bit reports the state of the /SYNC pin, latching on both input transitions. The same is true of External Sync mode. However, if the crystal oscillator is enabled while in Asynchronous mode, this bit will be forced to 0 and the latches will not be closed. Selecting the crystal option in External Sync mode is illegal, but the result will be the same.

In Synchronous modes other than SDLC, the Sync/Hunt reports the Hunt state of the receiver. Hunt mode is entered when the processor issues the Enter Hunt command in WR3. This forces the receiver to search for a sync character match in the receive data stream. Because both transitions of the Hunt bit close the latches, issuing this command will cause an External/Status interrupt. The SCC resets this bit when character synchronization has been achieved, causing the latches to again be closed.

In these synchronous modes, the SCC will not re-enter the Hunt mode automatically; only the Enter Hunt command will set this bit. In SDLC mode this bit is also set by the Enter Hunt command, but the receiver automatically enters the Hunt mode if an Abort sequence is received. The receiver leaves Hunt upon receipt of a flag sequence. Both transitions of the Hunt bit will cause the latches to be closed. In SDLC mode, the receiver automatically synchronizes on Flag characters. The receiver is in Hunt mode when it is enabled, so the Enter Hunt command is never needed.

2.4.9.6 External/Status Interrupt Handling

If careful attention is paid to details, the interrupt service routine for External/Status interrupts is straightforward. To determine which bit or bits changed state, the routine should first read RR0 and compare it to a copy from memory. For each changed bit, the appropriate action should be taken and the copy in memory updated. The service routine should close with two Reset External/Status interrupt commands to reopen the latches. The copy of RR0 in memory should always have the Zero Count bit set to 0, since this is the state of the bit after the Reset External/Status interrupts command at the end of the service routine. When the processor issues the Reset Transmit Underrun/EOM latch command in WR0, the Transmit Underrun/EOM bit in the copy of RR0 in memory should be reset because this transition does not cause an interrupt.

2.5 BLOCK/DMA 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 Write Register 1. The /W//REQ output can be defined by software as a /WAIT line in the CPU Block Transfer mode or as a /REQ line in the DMA Block Transfer mode. The /DTR//REQ pin can also be programmed through WR14 bit D2 to function as a DMA request for the transmitter.

To a DMA controller, the SCC's /REQ outputs indicate that the SCC is ready to transfer data to or from memory. To the CPU, the /WAIT output indicates that the SCC is not ready to transfer data, thereby requesting the CPU to extend the I/O cycle.

2.5.1 Block Transfers

The SCC offers several alternatives for the block transfer of data. The various options are selected by WR1 (bits D7 through D5) and WR14 (bit D2). Each channel in the SCC has two pins which are used to control the block transfer of data. Both pins in each channel may be programmed to act as DMA Request signals. The /W//REQ pin in each channel may be programmed to act as a Wait signal for the CPU. In either mode, it is advisable to select and enable the mode in two separate accesses of the appropriate register. The first access should select the mode and the second access should enable the function. This procedure prevents glitches on the output pins. Reset forces Wait mode, with /W//REQ open-drain.

2.5.1.1 Wait On Transmit

The Wait On Transmit function is selected by setting both D6 and D5 to 0 and then enabling the function by setting D7 of WR1 to 1. In this mode the /W//REQ pin carries the /WAIT signal, and is open-drain when inactive and Low when active. When the processor attempts to write to the transmit buffer when it is full, the SCC asserts /WAIT until the byte is written (Figure 2-21).

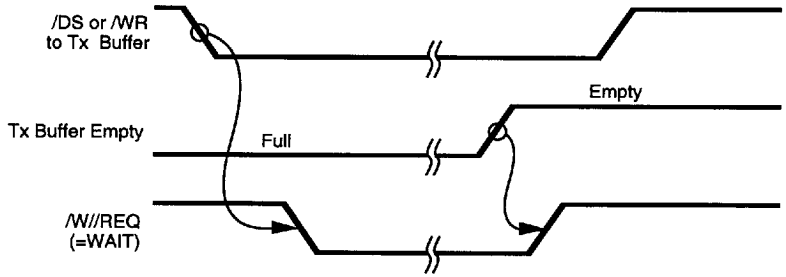

Figure 2-21. Wait On Transmit Timing

This allows the use of a block move instruction to transfer the transmit data. In the case of the Z80X30, /WAIT will go active in response to /DS going active, but only if WR8 is being accessed and a write is attempted. In all other cases, /WAIT remains open-drain. In the case of the Z85X30, /WAIT goes active in response to /WR going active, but only if the data buffer is being accessed, either directly or via the pointers. The /WAIT pin is released in response to the falling edge of PCLK. Details of the timing are shown in Figure 2-22.

Care must be taken when using this function, particularly at slow transmission speed. The /WAIT pin stays active as long as the transmit buffer stays full, so there is a possibility that the CPU may be kept waiting for a long period.

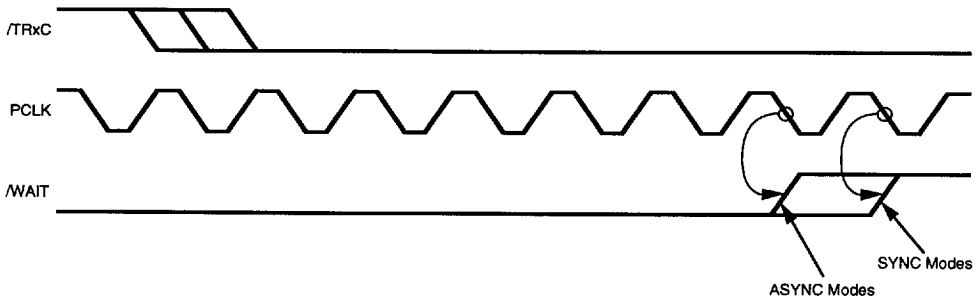

Figure 2-22. Wait On Transmit Timing

2.5.1.2 Wait On Receive

The Wait On Receive function is selected by setting D6 or WR1 to 0, D5 of WR1 to 1, and then enabling the function by setting D7 of WR1 to 1. In this mode, the /W//REQ pin carries the /WAIT signal, and is open-drain when inactive and Low when active. When the processor attempts to read data from the Receive FIFO when it is empty, the SCC asserts /WAIT until a character has reached the exit location of the FIFO (Figure 2-23).

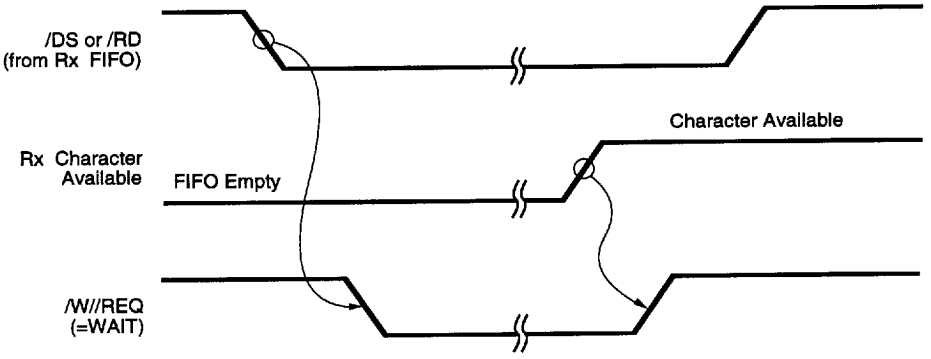

Figure 2-23. Wait On Receive Timing

This allows the use of a block move instruction to transfer the receive data. In the case of the Z80X30, /WAIT goes active in response to /DS going active, but only if RR8 is being accessed and a read is attempted. In all other cases, /WAIT remains open-drain. In the case of the Z85X30, /WAIT goes active in response to /RD going active, but only if the receive data FIFO is being accessed, either directly or via the pointers. The /WAIT pin is released in response to the falling edge of PCLK. Details of the timing are shown in Figure 2-24.

Care must be taken when this mode is used. The /WAIT pin stays active as long as the Receive FIFO remains empty. When the CPU access the SCC, the CPU remains in the wait state until data gets into the Receive FIFO, freezing the system.

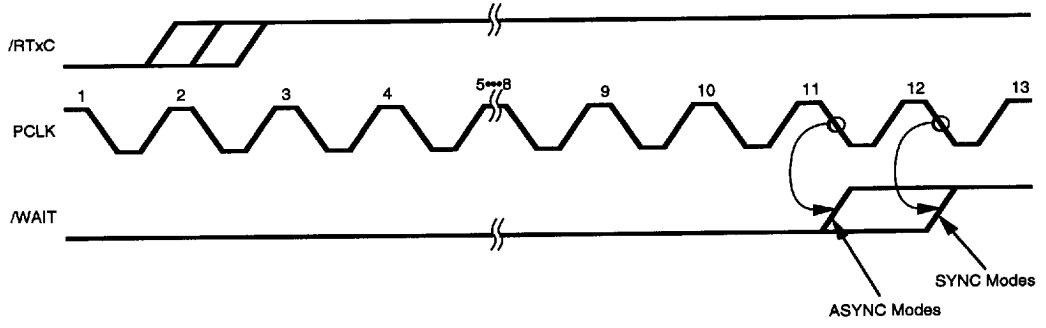

Figure 2-24. Wait On Receive Timing

2.5.2 DMA Requests

The two DMA request pins /W//REQ and /DTR//REQ can be programmed for DMA requests. The /W//REQ pin is used as either a transmit or a receive request, and the /DTR//REQ pin can be used as a transmit request only. For full-duplex operation, the /W//REQ is used for receive, and the /DTR//REQ is used for transmit. These modes are described below.

2.5.2.1 DMA Request on ESCC

Transmit DMA request is also affected by WR7' bit D5. As noted earlier, WR7' D5 affects both the transmit interrupt and DMA request generation similarly.

Note: WR7' D3 is ignored by the Receive Request function. This allows a DMA to transfer all bytes out of the Receive FIFO and still maintain the full advantage of the FIFO when the DMA has a long latency response acquiring the data bus.

Bit D5 of WR7' is set to 1 after reset to maintain maximum compatibility with SCC designs. This is necessary because if WR7' D5=0 when the request function is enabled, requests are made in rapid succession to fill the FIFO. Consequently, some designs which require an edge to be detected for each data transfer may not recover fast enough to detect the edges. This is handled by programming WR7' D5=1, or changing the DMA to be level sensitive instead of edge sensitive. Programming WR7' D5=0 has the advantage of the DMA requesting to keep the FIFO full. Therefore, if the CPU is busy, a significantly longer latency can be tolerated without the transmitter underrunning.

2.5.2.2 DMA Request On Transmit (using /W//REQ)

The Request On Transmit function is selected by setting D6 of WR1 to 1, D5 of WR1 to 0, and then enabling the function by setting D7 of WR1 to 1. In this mode, the /W//REQ pin carries the /REQ signal, which is active Low. When this mode is selected but not yet enabled, the /W//REQ is driven High.

The /REQ pin generates a falling edge for each byte written to the transmit buffer when the DMA controller is to write new data. For the Z80X30, the /REQ pin then goes inactive on the falling edge of the DS that writes the new data (see AC spec #26, TdDSf(REQ)) For the Z85X30, the /REQ pin then goes inactive on the falling edge of the WR strobe that writes the new data (see AC spec #33, TdWRf(REQ)) This is shown in Figure 2-25.

Note: The /REQ pin follows the state of the transmit buffer even though the transmitter is disabled. Thus, if the /REQ is enabled, the DMA writes data to the SCC before the transmitter is enabled. This will not cause a problem in Asynchronous mode, but it may cause problems in Synchronous mode because the SCC sends data in preference to flags or sync characters. It may also complicate the CRC initialization, which cannot be done until after the transmitter is enabled.

On the ESCC, this complication can be avoided in SDLC mode by using the Automatic SDLC Opening Flag Transmission feature and the Auto EOM reset feature, which also resets the transmit CRC (see Section 4.4.1 for details). Applications using other synchronous modes should enable the transmitter before enabling the /REQ function.

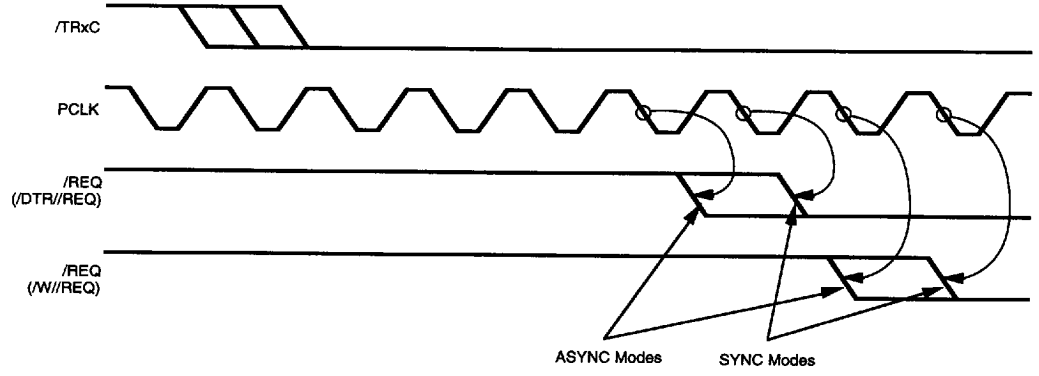

Figure 2-25. Transmit Request Assertion

With only one exception, the /REQ pin directly follows the state of the transmit buffer (for the ESCC as programmed by WR7' D5) in this mode. The SCC generates only one falling edge on /REQ per character requested and the timing for this is shown in Figure 2-26.

The one exception occurs in synchronous modes at the end of a CRC transmission. At the end of a CRC transmission, when the closing flag or sync character is loaded into the Transmit Shift Register, /REQ is pulsed High for one PCLK cycle. The DMA uses this falling edge on /REQ to write the first character of the next frame to the SCC. In the case of the Z80X30, /REQ goes High in response to the falling edge of DS, but only if the appropriate channel transmit buffer in the SCC is accessed. This is shown in Figure 2-25. In the case of the Z85X30, /REQ goes High in response to the falling edge of /WR, but only when the appropriate channel transmit buffer in the SCC is accessed This is shown in Figure 2-27.

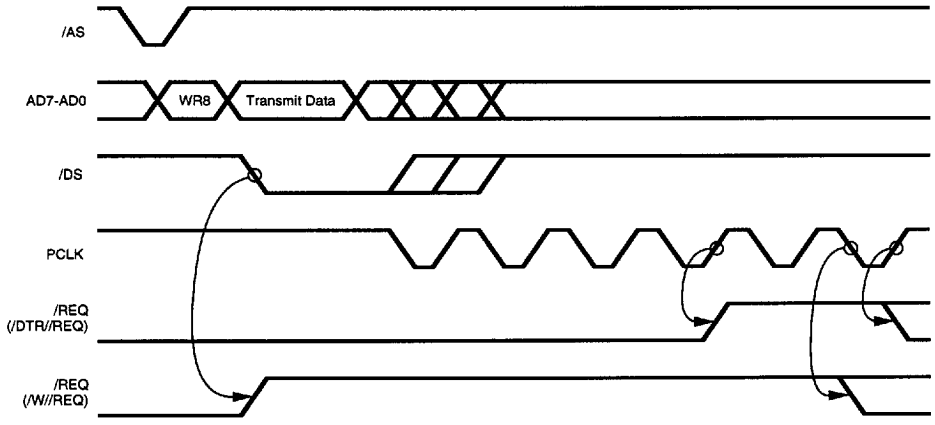

Figure 2-26. Z80X30 Transmit Request Release

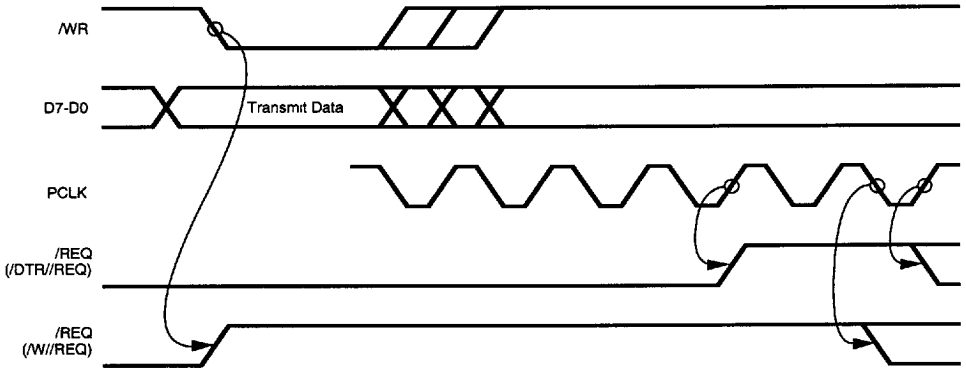

Figure 2-27. Z85X30 Transmit Request Release

2.5.2.3 DMA Request On Transmit (using /DTR//REQ)
A second Request on Transmit function is available on the /DTR//REQ pin. This mode is selected by setting D2 of WR14 to 1. /REQ goes Low when the Transmit FIFO is empty if WR7' D5=1, or when the exit location of the Transmit FIFO is empty if WR7' D5=0. In the Request mode, /REQ follows the state of the Transmit FIFO even though the transmitter is disabled. While D2 of WR14 is set to 0, the /DTR//REQ pin is /DTR and follows the inverted state of D7 in WR5. This pin is High after a channel or hardware reset and in the DTR mode.

The /DTR//REQ pin goes inactive High between each transfer for a minimum of one PCLK cycle (Figure 28).

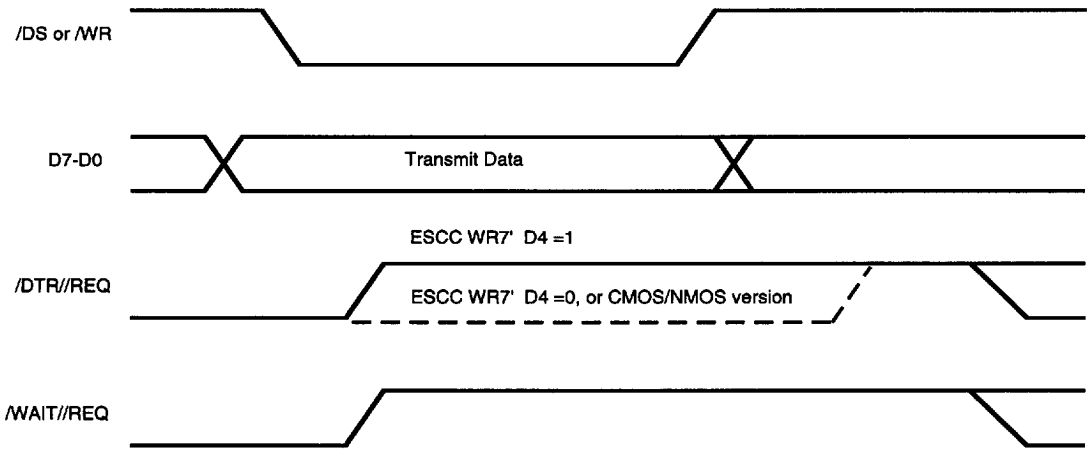

Figure 2-28. /DTR//REQ Deassertion Timing

ESCC:
The timing of deactivation of this pin is programmable through WR7' bit D4. The /DTR//REQ waits until the write operation has been completed before going inactive. Refer to Z85230 AC spec #35a TdWRr(REQ) and Z80230 AC spec #27a TdDSr(REQ). This mode is compatible with the SCC and guarantees that any subsequent access to the ESCC does not violate the valid access recovery time requirement.

If WR7' D4=1, the /DTR//REQ is deactivated with identical timing as the /W/REQ pin. Refer to Z85230 AC spec #35b TdWRr(REQ) and Z80230 AC spec #27b TdDSr(REQ).

This feature is beneficial to applications needing the DMA request to be deasserted quickly. It prevents a full Transmit FIFO from being overwritten due to the assertion of REQUEST being too long and being recognized as a request for more data.

Note: If WR7' D4=1, analysis should be done to verify that the ESCC is not repeatedly accessed in less than four PCLKs. However, since many DMAs require four clock cycles to transfer data, this typically is not a problem.

In the Request mode, /REQ will follow the state of the transmit buffer even though the transmitter is disabled. Thus, if /REQ is enabled before the transmitter is enabled, the DMA may write data to the SCC before the transmitter is enabled. This does not cause a problem in Asynchronous mode, but may cause problems in Synchronous modes because the SCC sends data in preference to flags or sync characters. It may also complicate the CRC initialization, which cannot be done until after the transmitter is enabled. On the ESCC, this complication can be avoided in SDLC mode by using the Automatic SDLC Opening Flag Transmission feature and Auto EOM reset feature which also resets the transmit CRC. (See section 4.4.1.2 for details). Applications using other synchronous modes should enable the transmitter before enabling the /REQ function.

With only one exception, the /REQ pin directly follows the state of the Transmit FIFO (for ESCC, as programmed by WR7' D5) in this mode. The one exception occurs in synchronous modes at the end of a CRC transmission. At the end of a CRC transmission, when the closing flag or sync character is loaded into the Transmit Shift Register, /REQ is pulsed High for one PCLK cycle. The DMA uses this falling edge on /REQ to write the first character of the next frame to the SCC.

2.5.2.4 DMA Request On Receive

The Request On Receive function is selected by setting D6 and D5 of WR1 to 1 and then enabling the function by setting D7 of WR1 to 1. In this mode, the /W//REQ pin carries the /REQ signal, which is active Low. When REQ on Receive is selected, but not yet enabled (WR1 D7=0), the /W//REQ pin is driven High. When the enable bit is set, /REQ goes Low if the Receive FIFO contains a character at the time, or will remain High until a character enters the Receive FIFO. Note that the /REQ pin follows the state of the Receive FIFO even though the receiver is disabled. Thus, if the receiver is disabled and /REQ is still enabled, the DMA transfers the previously received data correctly. In this mode, the /REQ pin directly follows the state of the Receive FIFO with only one exception. /REQ goes Low when a character enters the Receive FIFO and remains Low until this character is removed from the Receive FIFO.

The SCC generates only one falling edge on /REQ per character transfer requested (Figure 2-29). The one exception occurs in the case of a special receive condition in the Receive Interrupt on First Character or Special Condition mode, or the Receive Interrupt on Special Condition Only mode. In these two interrupt modes, any receive character with a special receive condition is locked at the top of the FIFO until an Error Reset command is issued. This character in the Receive FIFO would ordinarily cause additional DMA Requests after the first time it is read. However, the logic in the SCC guarantees only one falling edge on /REQ by holding /REQ High from the time the character with the special receive condition is read, and the FIFO locked, until after the Error Reset command has been issued.

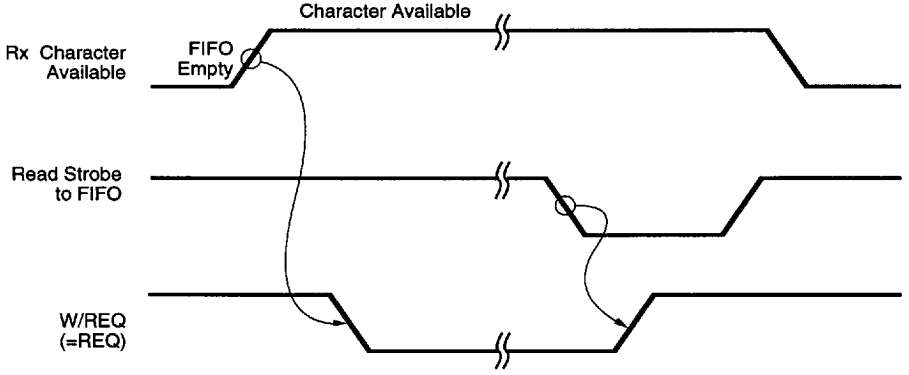

Figure 2-29. DMA Receive Request Assertion

Once the FIFO is locked, it allows the checking of the Receive Error FIFO (RR1) to find the cause of the error. Locking the data FIFO, therefore, stops the error status from popping out of the Receive Error FIFO. Also, since the DMA request becomes inactive, the interrupt (Special Condition) is serviced.

Once the FIFO is unlocked by the Error Reset command, /REQ again follows the state of the receive buffer. In the case of the Z80X30, /REQ goes High in response to the falling edge of /DS, but only if the appropriate receive buffer in the SCC is accessed (Figure 2-30). In the case of the Z85X30, /REQ goes High in response to the falling edge of /RD, but only when the appropriate receive buffer in the SCC is accessed (Figure 2-31).

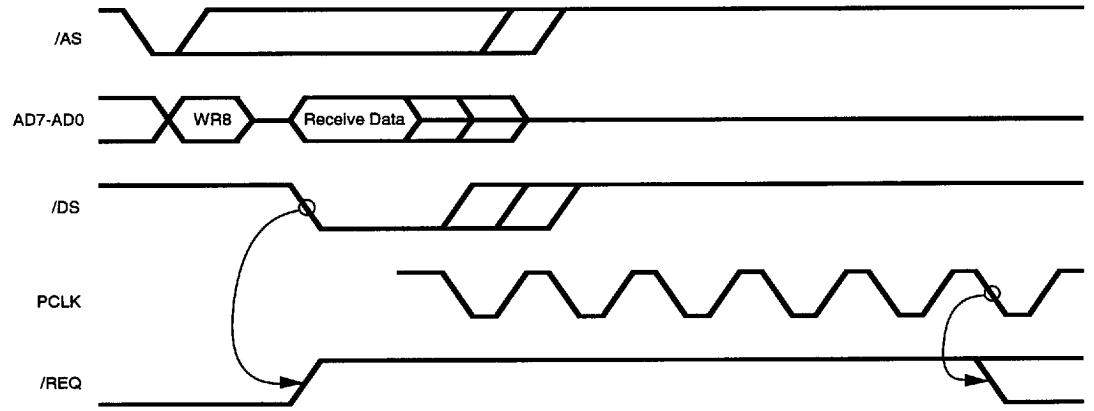

Figure 2-30. Z80X30 Receive Request Release

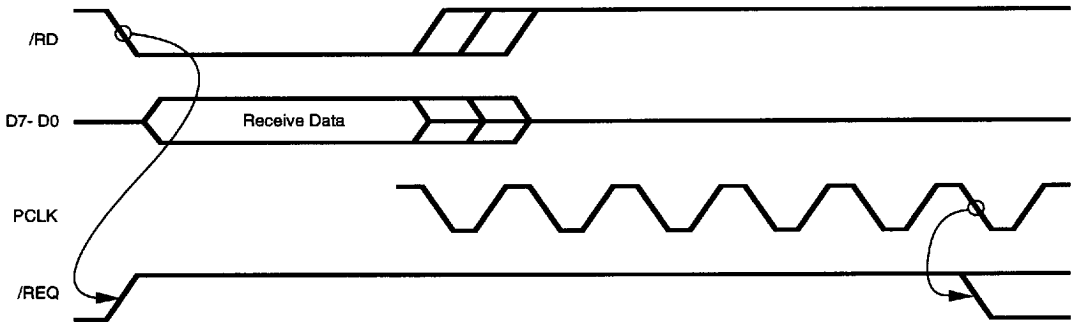

Figure 2-31. Z85X30 Receive Request Release

2.6 TEST FUNCTIONS

The SCC contains two other features useful for diagnostic purposes, controlled by bits in WR14. They are Local Loopback and Auto Echo.

2.6.1 Local Loopback

Local Loopback is selected when WR14 bit D4 is set to 1. In this mode, the output of the transmitter is internally connected to the input of the receiver. At the same time, the TxD pin remains connected to the transmitter. In this mode, the /DCD pin is ignored as a receive enable and the /CTS pin is ignored as a transmitter enable even if the Auto Enable mode has been selected. Note that the DPLL input is connected to the RxD pin, not to the input of the receiver. This precludes the use of the DPLL in Local Loopback. Local Loopback is shown schematically in Figure 2-32.

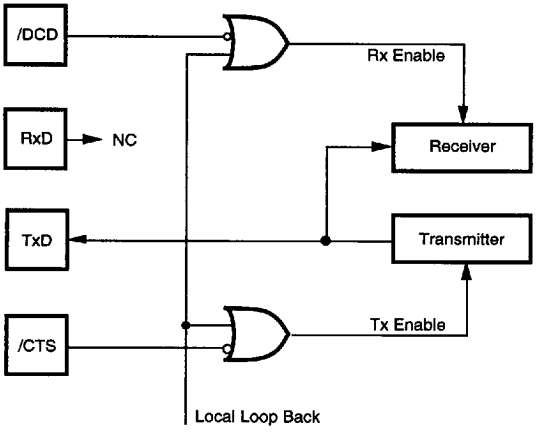

Figure 2-32. Local Loopback

2.6.2 Auto Echo

Auto Echo is selected when bit D3 of WR14 is set to 1. In this mode, the TxD pin is connected directly to the RxD pin, and the receiver input is connected to the RxD pin. In this mode, the /CTS pin is ignored as a transmitter enable and the output of the transmitter does not connect to anything. If both the Local Loopback and Auto Echo bits are set to 1, the Auto Echo mode is selected, but both the /CTS pin and /DCD pin are ignored as auto enables. This should not be considered a normal operating mode, however (Figure 2-33).

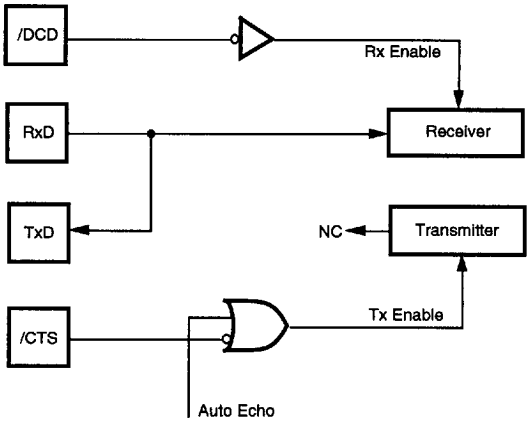

Figure 2-33. Auto Echo