INTERFACING Z80® CPUS TO THE Z8500 PERIPHERAL FAMILY

INTRODUCTION

The Z8500 Family consists of universal peripherals that can interface to a variety of microprocessor systems that use a non-multiplexed address and data bus. Though similar to Z80 peripherals, the Z8500 peripherals differ in the way they respond to I/O and Interrupt Acknowledge cycles. In addition, the advanced features of the Z8500 peripherals enhance system performance and reduce processor overhead.

To design an effective interface, the user needs an understanding of how the Z80 Family interrupt structure works, and how the Z8500 peripherals interact with this structure. This application note provides basic information on the interrupt structures, as well as a discussion of the hardware and software considerations involved in interfacing the Z8500 peripherals to the Z80 CPUs. Discussions center around each of the following situations:

- Z80A 4 MHz CPU to Z8500 4 MHz peripherals
- Z80B 6 MHz CPU to Z8500A 6 MHz peripherals
- Z80H 8 MHz CPU to Z8500 4 MHz peripherals
- Z80H 8 MHz CPU to Z8500A 6 MHz peripherals

This application note assumes the reader has a strong working knowledge of the Z8500 peripherals; it is not intended as a tutorial.

CPU HARDWARE INTERFACING

The hardware interface consists of three basic groups of signals; data bus, system control, and interrupt control, described below. For more detailed signal information, refer to Zilog's DataBook, Universal Peripherals.

Data Bus Signals

D7-D0. Data Bus (bidirectional tri-state). This bus transfers data between the CPU and the peripherals.

System Control Signals

AD-A0. Address Select Lines (optional). These lines select the port and/or control registers.

/CE. Chip Enable (input, active Low). /CE is used to select the proper peripheral for programming. /CE should be gated with /IORQ or /MREQ to prevent spurious chip selects during other machine cycles.

/RD* Read (input, active Low). /RD activates the chip-read circuitry and gates data from the chip onto the data bus.

/WR* Write (input, active Low). /WR strobes data from the data bus into the peripheral.

*Chip reset occurs when /RD and /WR are active simultaneously.

Interrupt Control

/INTACK. Interrupt Acknowledge (input, active Low). This signal indicates an Interrupt Acknowledge cycle and is used with /RD to gate the interrupt vector onto the data bus.

/INT. Interrupt Request (output, open-drain, active Low).

The IUS bit indicates that an interrupt is currently being serviced by the CPU. The IUS bit is set during an Interrupt Acknowledge cycle if the IP bit is set and the IEI line is High. If the IEI line is Low, the IUS bit is not set, and the device is inhibited from placing its vector onto the data bus. In the Z80 peripherals, the IUS bit is normally cleared by decoding the RETI instruction, but can also be cleared by a software command (SIO). In the Z8500 peripherals, the IUS bit is cleared only by software commands.

CPU HARDWARE INTERFACING (Continued)

Z80® Interrupt Daisy-Chain Operation

In the Z80 peripherals, both the IP and IUS bits control the IEO line and the lower portion of the daisy chain.

When a peripheral's IP bit is set, its IEO line is forced Low. This is true regardless of the state of the IEI line. Additionally, if the peripheral's IUS bit is clear and its IEI line High, the /INT line is also forced Low.

The Z80 peripherals sample for both /M1 and /IORQ active, and /RD inactive to identify and Interrupt Acknowledge cycle. When /M1 goes active and /RD is inactive, the peripheral detects an Interrupt Acknowledge cycle and allows its interrupt daisy chain to settle. When the /IORQ line goes active with /M1 active, the highest priority interrupting peripheral places its interrupt vector onto the data bus. The IUS bit is also set to indicate that the peripheral is currently under service. As long as the IUS bit is set, the IEO line is forced Low. This inhibits any lower priority devices from requesting an interrupt. When the Z80 CPU executes the RETI instruction, the peripherals monitor the data bus and the highest priority device under service resets its IUS bit.

Z8500 Interrupt Daisy-Chain Operation

In the Z8500 peripherals, the IUS bit normally controls the state of the IEO line. The IP bit affects the daisy chain only during an Interrupt Acknowledge cycle. Since the IP bit is normally not part of the Z8500 peripheral interrupt daisy chain, there is no need to decode the RETI instruction. To allow for control over the daisy chain, Z8500 peripherals have a Disable Lower Chain (DLC) software command that pulls IEO Low. This can be used to selectively deactivate parts of the daisy chain regardless of the interrupt status. Table 1 shows the truth tables for the Z8500 interrupt daisy-chain control signals during certain cycles. Table 2 shows the interrupt state diagram for the Z8500 peripherals.

Table 1. Z8500 Daisy-Chain Control Signals

Truth Table for Daisy Chain Signals During Idle State				Truth Table for Daisy Chain Signals During /INTACK Cycle			
IEI	**IP**	**IUS**	**IEO**	**IEI**	**IP**	**IUS**	**IEO**
0	X	X	0	0	X	X	0
1	X	0	1	1	1	X	0
1	X	1	0	1	X	1	0
1	0	0	1				

IEI. Interrupt Enable In (Input, active High).

IEO. Interrupt Enable Out (output, active High).

These lines control the interrupt daisy chain for the peripheral interrupt response.

Z8500 I/O Operation

The Z8500 peripherals generate internal control signals from /RD and /WR. Since PCLK has not required phase relationship to /RD or /WR, the circuitry generating these signals provides time for metastable conditions to disappear.

The Z8500 peripherals are initialized for different operating modes by programming the internal registers. These internal registers are accessed during I/O Read and Write cycles, which are described below.

Read Cycle Timing

Figure 1 illustrates the Z8500 Read cycle timing. All register addresses and /INTACK must remain stable throughout the cycle. If /CE goes active after /RD goes active, or if /CE goes inactive before /RD goes inactive, then the effective Read cycle is shortened.

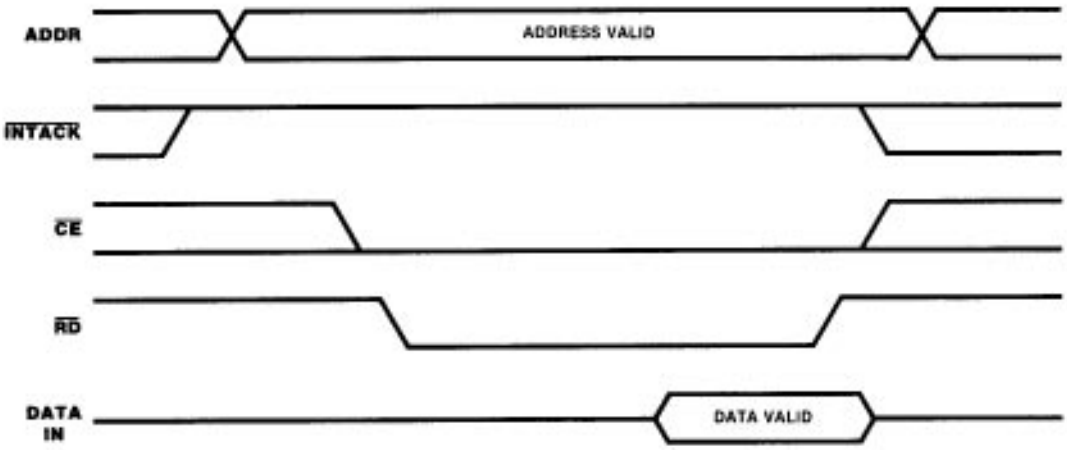

Figure 1. Z8500 Peripheral I/O Read Cycle Timing

Write Cycle Timing

Figure 2 illustrates the Z8500 Write cycle timing. All register addresses and /INTACK must remain stable throughout the cycle. If /CE goes active after /WR goes active, or if /CE goes inactive before /WR goes inactive, then the effective Write cycle is shortened. Data must be available to the peripheral prior to the falling edge of /WR.

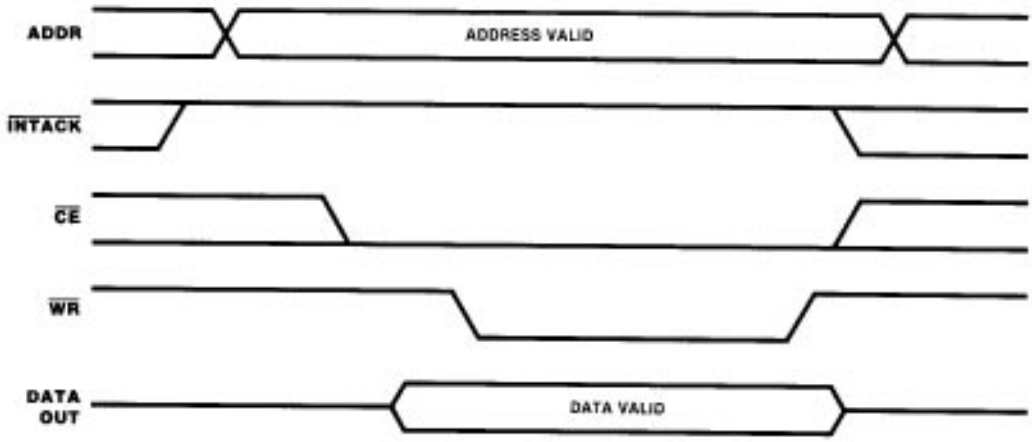

Figure 2. Z8500 Peripheral I/O Write Cycle Timing

PERIPHERAL INTERRUPT OPERATION

Understanding peripheral interrupt operation requires a basic knowledge of the Interrupt Pending (IP) and Interrupt Under Service (IUS) bits in relation to the daisy chain. Both Z80 and Z8500 peripherals are designed in such a way that no additional interrupts can be requested during an Interrupt Acknowledge cycle. This allows that interrupt daisy chain to settle, and ensures proper response of the interrupting device.

The IP bit is set in the peripheral when CPU intervention is required (such conditions as buffer empty, character available, error detection, or status changes). The Interrupt Acknowledge cycle does not necessarily reset the IP bit. This bit is cleared by a software command to the peripheral, or when the action that generated the interrupt is completed (i.e., reading a character, writing data, resetting errors, or changing the status). When the interrupt has been serviced, other interrupts can occur.

The Z8500 peripherals use /INTACK (Interrupt Acknowledge) for recognition of an Interrupt Acknowledge cycle. This pin, used in conjunction with /RD, allows the Z8500 peripheral to gate its interrupt vector onto the data bus. An active /RD signal during an Interrupt Acknowledge cycle performs two functions. First, it allows the highest priority device requesting an interrupt to place its interrupt vector on the data bus. Secondly, it sets the IUS bit in the highest priority device to indicate that the device is currently under service.

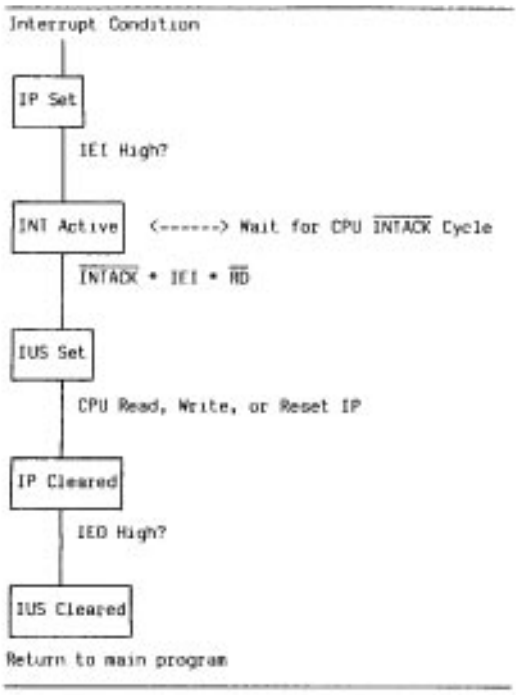

Figure 3. Z8500 Interrupt State Diagram

INPUT/OUTPUT CYCLES

Although Z8500 peripherals are designed to be as universal as possible, certain timing parameters differ from the standard Z80 timing. The following sections discuss the I/O interface for each of the Z80 CPUs and the Z8500 peripherals. Figure 9 depicts logic for the Z80A CPU to Z8500 peripherals (and Z80B CPU to Z8500A peripherals) I/O interface as well as the Interrupt Acknowledge interface. Figures 4 and 7 depict some of the logic used to interface the Z80H CPU to the Z8500 and Z8500A peripherals for the I/O and Interrupt Acknowledge interfaces. The logic required for adding additional Wait states into the timing flow is not discussed in the following sections.

Z80A CPU to Z8500 Peripherals

No additional Wait states are necessary during the I/O cycles, although additional Wait states can be inserted to compensate for timing delays that are inherent in a system. Although the Z80A timing parameters indicate a negative value for data valid prior to /WR, this is a worse than “worst case” value. This parameter is based upon the longest (worst case) delay for data available from the falling edge of the CPU clock minus the shortest (best case) delay for CPU clock High to /WR low. The negative value resulting from these two parameters does not occur because the worst case of one parameter and the best case of the other do not occur within the same device. This indicates that the value for data available prior to /WR will always be greater than zero.

All setup and pulse width times for the Z8500 peripherals are met by the standard Z80A timing. In determining the interface necessary, the /CE signal to the Z8500 peripherals is assumed to be the decoded address qualified with the /IORQ signal.

Figure 4 shows the minimum Z80A CPU to Z8500 peripheral interface timing for I/O cycles. If additional Wait states are needed, the same number of Wait states can be inserted for both I/O Read and Write cycles to simplify interface logic. There are several ways to place the Z80A CPU into a Wait condition (such as counters or shift registers to count system clock pulses), depending upon whether or not the user wants to place Wait states in all I/O cycles, or only during Z8500 I/O cycles. Tables 3 and 4 list the Z8500 peripheral and the Z80A CPU timing parameters (respectively) of concern during the I/O cycles. Tables 5 and 6 list the equations used in determining if these parameters are satisfied. In generating these equations and the values obtained from them, the required number of Wait states was taken into account. The reference numbers in Tables 3 and 4 refer to the timing diagram in Figure 4.

INPUT/OUTPUT CYCLES (Continued)

Table 2. Z8500 Timing Parameters I/O Cycles

Worst Case			Min	Max	Units
6.	TsA(WR)	Address to /WR to Low Setup	80		ns
1.	TsA(RD)	Address to /RD Low Setup	80		ns
2.	TdA(DR)	Address to Read Data Valid		590	
	TsCEI(WR)	/CE Low to /WR Low Setup		ns	
	TsCEI(RD)	/CE Low to /RD Low Setup		ns	
4.	TwRDI	/RD Low Width	390		ns
8.	TwWRI	/WR Low Width	390		ns
3.	TdRDf(DR)	/RD Low to Read Data Valid		255	ns
7.	TsDW(WR)	Write Data to /WR Low Setup	0		ns

Table 3. Z80A Timing Parameters I/O Cycles

Worst Case			Min	Max	Units
	TcC	Clock Cycle Period	250		ns
	TwCh	Clock Cycle High Width	110		ns
	TfC	Clock Cycle Fall Time		30	ns
	TdCr(A)	Clock High to Address Valid		110	ns
	TdCr(RDf)	Clock High to /RD Low		85	ns
	TdCr(IORQf)	Clock High to /IORQ Low		75	ns
	TdCr(WRf)	Clock High to /WR Low		65	ns
5.	TsD(Cf)	Data to Clock Low Setup	50		ns

Table 4. Parameter Equations

Z8500 Parameter	Z80A Equation	Value	Units
TsA(RD)	TcC-TdCr(A)	140 min	ns
TdA(DR)	3TcC+TwCh-TdCr(A)-TsD(Cf)	800 min	ns
TdRDf(DR)	2TcC+TwCh-TsD(Cf)	460 min	ns
TwRD1	2TcC+TwCh+TfC-TdCr(RDf)	525 min	ns
TsA(WR)	TcC-TdCr (A)	140 min	ns
TsDW(WR)		>0 min	ns
TwWR1	2TcC+TwCh+TfC-TdCr(WRf)	560 min	ns

.

Table 5. Parameter Equations

Z80A Parameter	Z8500 Equation	Value	Units
TsD(Cf)	3TcC+TwCh-TdCr(A)-TdA(DR)	160 min	ns
	/RD		
	2TcC+TwCh-TdCr(RDf)-TdRD(DR)	135 min	ns

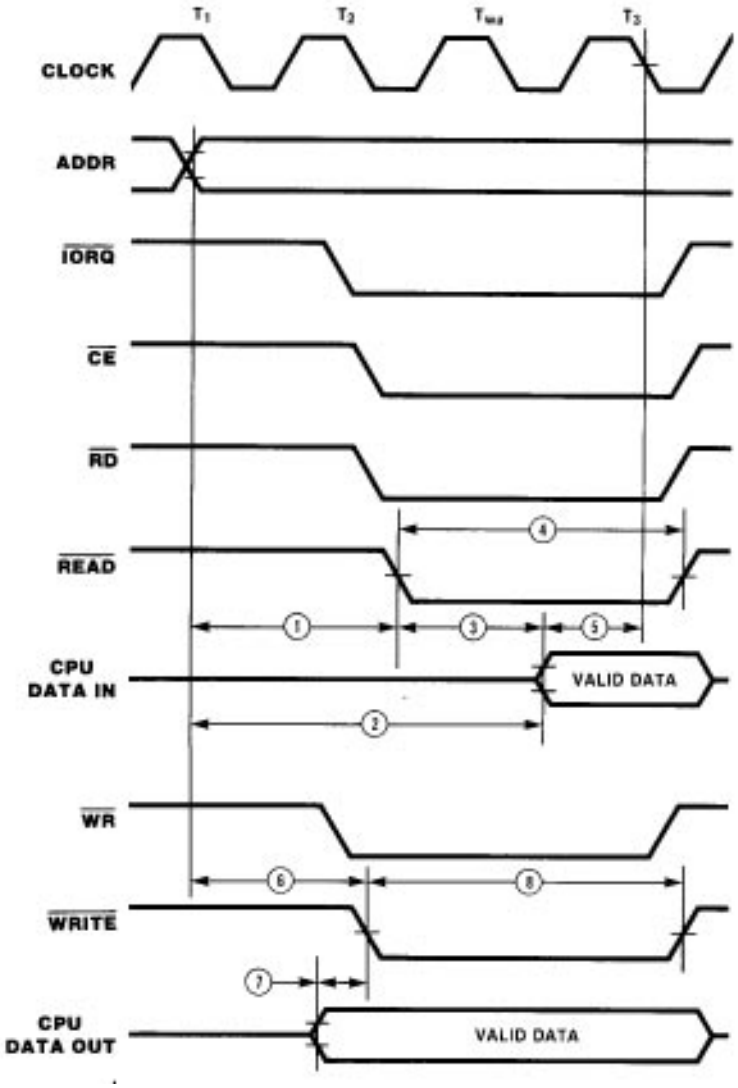

Figure 4. Z80A CPU to Z8500 Peripheral Minimum I/O Cycle Timing

Z80B CPU TO Z8500A PERIPHERALS

No additional Wait states are necessary during I/O cycles, although Wait states can be inserted to compensate for any systems delays. Although the Z80B timing parameters indicate a negative value for data valid prior to /WR, this is a worse than “worst case” value. This parameter is based upon the longest (worst case) delay for data available from the falling edge of the CPU clock minus the shortest (best case) delay for CPU clock High to /WR Low. The negative value resulting from these two parameters does not occur because the worst case of one parameter and best case of the other do not occur within the same device. This indicates that the value for data available prior to /WR will always be greater than zero.

All setup and pulse width times for the Z8500A peripherals are met by the standard Z80B timing. In determining the interface necessary, the /CE signal to the Z8500A peripherals is assumed to be the decoded address qualified with /IORQ signal.

Figure 5 shows the minimum Z80B CPU to Z8500A peripheral interface timing for I/O cycles. If additional Wait states are needed, the same number of Wait states can be inserted for both I/O Read and I/O Write cycles in order to simplify interface logic. There are several ways to place the Z80B CPU into a Wait condition (such as counters or shift registers to count system clock pulses), depending upon whether or not the user wants to place Wait states in all I/O cycles, or only during Z8500A I/O cycles. Tables 6 and 7 list the Z8500A peripheral and Z80B CPU timing parameters (respectively) of concern during the I/O cycles. Tables 8 and 9 list the equations used in determining if these parameters are satisfied. In generating these equations and the values obtained from them, the required number of Wait states was taken into account. The reference numbers in Tables 6 and 7 refer to the timing diagram of Figure 5.

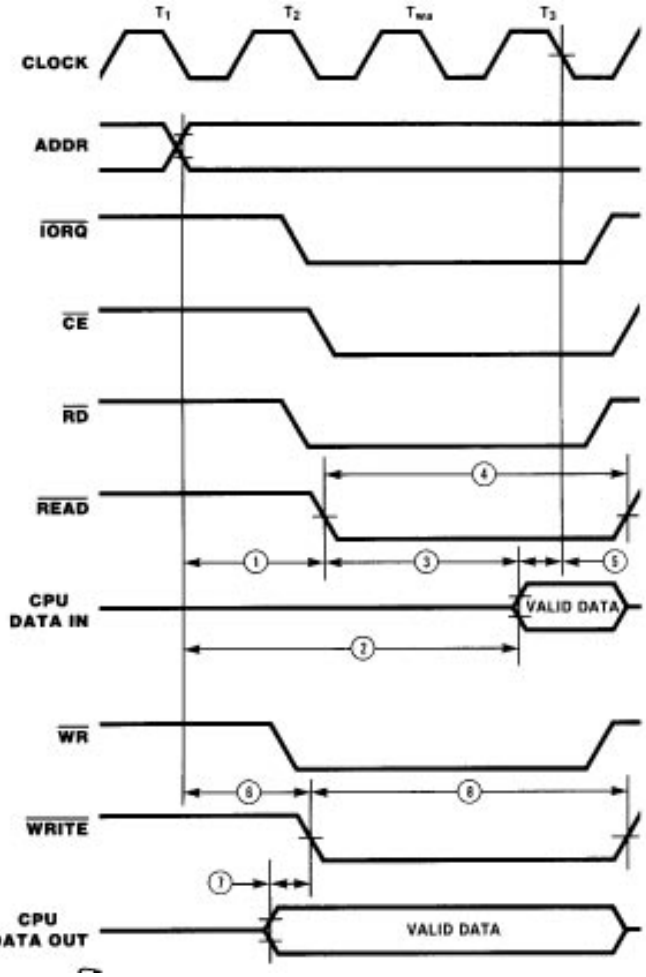

Figure 5. Z80B CPU to Z8500A Peripheral Minimum I/O Cycle Timing

Table 6. Z8500A Timing Parameters I/O Cycles

Worst Case			Min	Max	Units
6.	TsA(WR)	Address to /WR Low Setup	80		ns
1.	TsA(RD)	Address to /RD Low Setup	80		ns
2.	TdA(DR)	Address to Read Data Valid		420	ns
	TsCE1(WR)	/CE Low to /WR Low Setup		ns	
	TsCE1(RD)	/CE Low to /RD Low Setup		ns	
4.	TwRD1	/RD Low Width	250		ns
8.	TwWR1	/WR Low Width	250		ns
3.	TdRDf(DR)	/RD Low to Read Data Valid		180	ns
7.	TsDW(WR)	Write Data to /WR Low Setup	0		ns

Table 7. Z80B Timing Parameters I/O Cycles

Worst Case			Min	Max	Units
	TcC	Clock Cycle Period	165		ns
	TwCh	Clock Cycle High Width	65		ns
	TfC	Clock Cycle Fall Time		20	ns
	TdCr(A)	Clock High to Address Valid		90	ns
	TdCr(RDf)	Clock High to /RD Low		70	ns
	TdCR(IORQf)	Clock High to /IORQ Low		65	ns
	TdCr(WRf)	Clock High to /WR Low		60	ns
5.	TsD(Cf)	Data to Clock Low Setup	40		ns

Table 8. Parameter Equations

Z8500A Parameter	Z80B Equation	Value	Units
TsA(RD)	TcC-TdCr(A)	>75 min	ns
TdA(DR)	3TcC+TwCh-TdCr(A)-TsD(Cf)	430 min	ns
TdRDf(DR)	2TcC+TwCh+TsD(Cf)	345 min	ns
TwRD1	2TcC+TwCh+TfC-TdCr(RDf)	325 min	ns
TsA(WR)	TcC-TdCr(A)	75 min	ns
TsDW(WR)		> 0 min	ns
TwWR1	2 TcC+Twch+TfC-TdCr(WRf)	352 min	ns

Table 9. Parameter Equations

Z8500A Equation		Value	Units
	3TcC+TwCh-TdCr(A)-TdA(DR)	50 min	ns
	2TcC+TwCh-TdCr(RDf)-TdRD(DR)	75 min	ns
	Z80H CPU to Z8500 Peripherals		

Z90H CPU TO Z8500 PERIPHERALS

During an I/O Read cycle, there are three Z8500 parameters that must be satisfied. Depending upon the loading characteristics of the /RD signal, the designer may need to delay the leading (falling) edge of /RD to satisfy the Z8500 timing parameter TsA(RD) (Addresses Valid to /RD Setup). Since Z80H timing parameters indicate that the /RD signal may go Low after the falling edge of T2, it is recommended that the rising edge of the system clock be used to delay /RD (if necessary). The CPU must also be placed into a Wait condition long enough to satisfy TdA(DR) (Address Valid to Read Data Valid Delay) and TdRDf(DR) (/RD Low to Read Data Valid Delay).

During an I/O Write cycle, there are three other Z8500 parameters that must be satisfied. Depending upon the loading characteristics of the /WR signal and the data bus, the designer may need to delay the leading (falling) edge of /WR to satisfy the Z8500 timing parameters TsA(WR) (Address Valid to /WR setup). Since Z80H timing parameters indicate that the /WR signal may go Low after the falling edge of T2, it is recommended that the rising edge of the system clock be used to delay /WR (if necessary). This delay will ensure that both parameters are satisfied. The CPU must also be placed into a Wait condition long enough to satisfy TwWR1 (/WR Low Pulse Width). Assuming that the /WR signal is delayed, only two additional Wait states are needed during an I/O Write cycle when interfacing the Z80H CPU to the Z8500 peripherals.

To simplify the I/O interface, the designer can use the same number of Wait states for both I/O Read and I/O Write cycles. Figure 6 shows the minimum Z80H CPU to Z8500 peripheral interface timing for the I/O cycles (assuming that the same number of Wait states are used for both cycles and that both /RD and /WR need to be delayed). Figure 8 shows two suits that can be used to delay the leading (falling) edge of either the /RD or the /WR signals. There are several ways to place the Z80A CPU into a Wait condition (such as counters or shift registers to count system clock pulses), depending upon whether or not the use wants to place Wait states in all I/O cycles, or only during Z8500 I/O cycles. Tables 3 and 10 list the Z8500 peripheral and the Z80H CPU timing parameters (respectively) of concern during the I/O cycles. Tables 13 and 14 list the equations used in determining if these parameters are satisfied. In generating these equations and the values obtained from them, the required number of Wait states was taken into account. The reference numbers in Tables 3 and 10 refer to the timing diagram of Figure 6.

Table 10. Z80H Timing Parameter I/O Cycles

		Equation	Min	Max	Units
	TcC	Clock Cycle Period	125		
	TwCh	Clock Cycle High Width	55		ns
	TfC	Clock Cycle Fall Time		10	ns
	TdCr(A)	Clock High to Address Valid		80	ns
	TdCr(RDf)	Clock High to /RD Low		60	ns
	TdCr(IORQf)	Clock High to /IORQ Low		55	ns
	TdCr(WRf)	Clock High to /WR Low		55	ns
5.	TsD(Cf)	Data to Clock Low Setup	30		ns

Table 11. Parameter Equations

Z8500 Parameter	Z80H Equation	Value	Units
TsA(RD)	2TcC-TdCr(A)	170 min	ns
TdA(DR)	6TcC+TwCh-TdCr(A)-TsD(Cf)	695 min	ns
TdRDf(DR)	4TcC+TwCh-TsD(Cf)	523 min	ns
TwRD1	4TcC+TwCh+TfC-TdCr(RDf)	503 min	ns
TsA(WR)	/WR - delayed		
	2TcC-TdCr(A)	170 min	ns
TsDW(WR)		>0 min	ns
TwWR1	4TcC+TwCh+TfC	563 min	ns

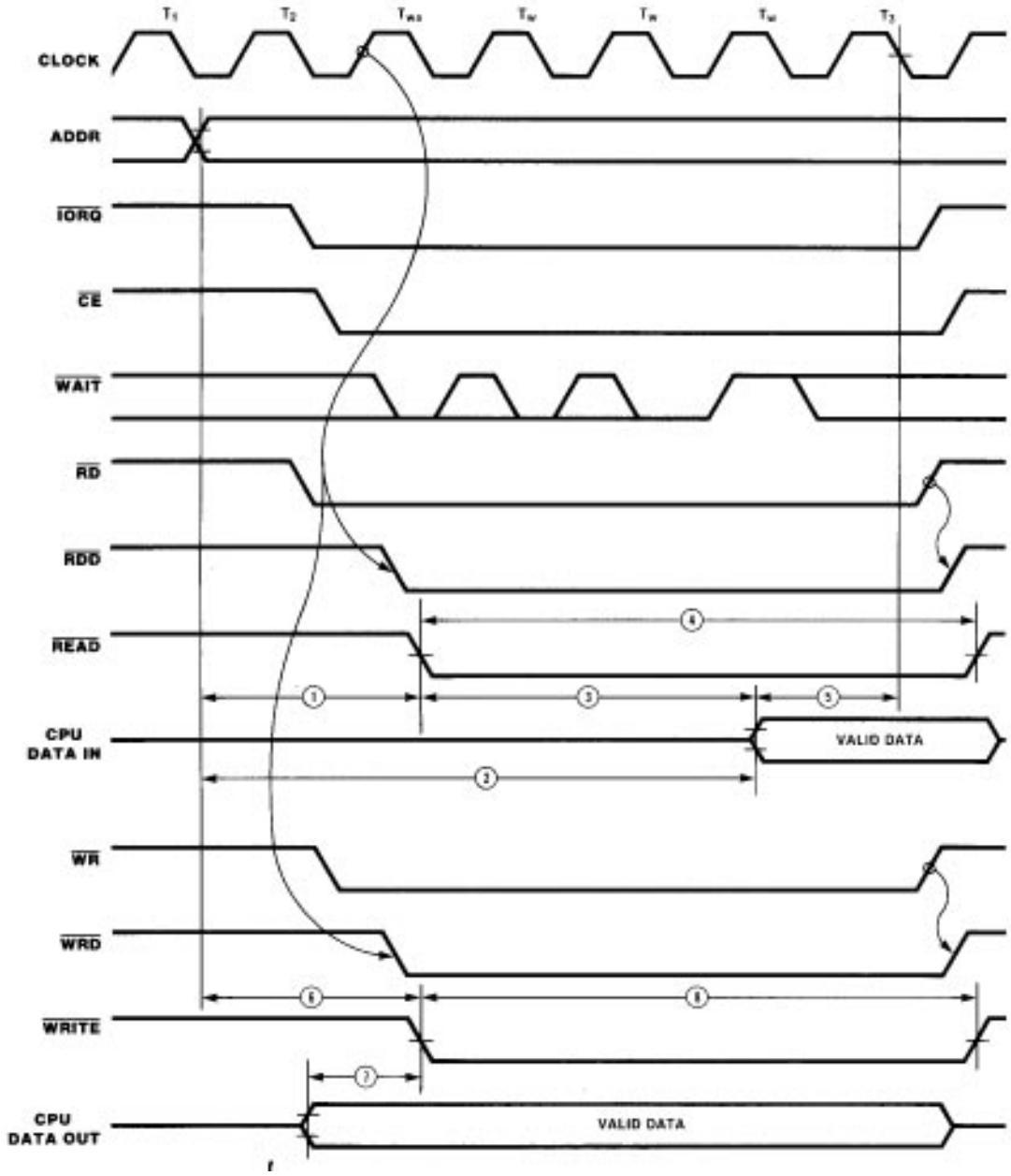

Figure 6. Z80H CPU to Z8500 Peripheral Minimum I/O Cycle Timing

Z80H CPU TO Z8500A PERIPHERALS

During an I/O Read cycle, there are three Z8500A parameters that must be satisfied. Depending upon the loading characteristics of the /RD signal, the designer may need to delay the leading (falling) edge of /RD to satisfy the Z8500A timing parameter TsA(RD) (Address Valid to /RD Setup). Since Z80H timing parameters indicate that the /RD signal may go Low after the falling edge of T2, it is recommended that the rising edge of the system must also be placed into Wait condition long enough to satisfy TdA(DR) (Address Valid to Read Data Valid Delay) and TdRDf(DR) (/RD Low to Read Data Valid Delay). Assuming that the /RD signal is delayed, then only one additional Wait state is needed during an I/O Read cycle when interfacing the Z80H CPU to the Z8500A peripherals.

During an I/O Write cycle, there are three other Z850A parameters that have to be satisfied. Depending upon the loading characteristics of the /WR signal and the data bus, the designer may need to delay the leading (falling) edge of /WR to satisfy the Z8500A timing parameters TsA(WR) (Address Valid to /WR Setup) and TsDW(WR) (Data Valid Prior to /WR Setup). Since Z80H timing parameters indicate that the /WR signal may go Low after the falling edge of T2, it is recommended that the rising edge of the system clock be used to delay /WR (if necessary). This delay will ensure that both parameters are satisfied. The CPU must also be placed into a Wait condition long enough to satisfy TwWR1 (/WR Low Pulse Width). Assuming that the /WR signal is delayed, then only one additional Wait state is needed during an I/O Write cycle when interfacing the Z80H CPU to the Z8500A peripherals.

Figure 7 shows the minimum Z80H CPU to Z8500A peripheral interface timing for the I/O cycles (assuming that the same number of Wait states are used for both cycles and that both /RD and /WR need to be delayed). Figure 8 shows two circuits that may be used to delay leading (falling) edge of either the /RD or the /WR signals. There are several methods used to place the Z80A CPU into a Wait condition (such as counters or shift registers to count system clock pulses), depending upon whether or not the user wants to place Wait states in all I/O cycles, or only during Z8500A I/O cycles, Tables 7 and 11 list the Z8500A peripheral and the Z80H CPU timing parameters (respectively) of concern during the I/O cycles. Tables 14 and 15 list the equations used in determining if these parameters are satisfied. In generating these equations and the values obtained from them, the required number of Wait states was taken into account. The reference numbers in Table 4 and 11 refer to the timing diagram of Figure 7.

Table 12. Parameter Equations

Z80H Parameter	Z8500 Equation	Value	Units
TsD(Cf)	6TcC+TwCh-TdCr(A)-TdA(DR)	135 min	ns
	/RD - delayed		
	4TcC+TwCh+TfC-TdRD(DR)	300 min	ns

Table 13. Parameter Equations

Z8500A Parameter	Z80H Equation	Value	Units
TsA(RD)	2TcC-TdCr(A)	170 min	ns
TdA(DR)	6TcC+TwCh-TdCr(A)-TsD(Cf)	695 min	ns
TdRDf(DR)	4TcC+TwCh-TsD(Cf)	525 min	ns
TwRD1	4TcC+TwCh+TfC-TdCr(RDf)	503 min	ns
TsA(WR)	/WR - delayed		
	2TcC-TdCr(A)	170 min	ns
TsDW(WR)		>0 min	ns
TwWR1	2TcC+TwCh+TfC	313 min	ns

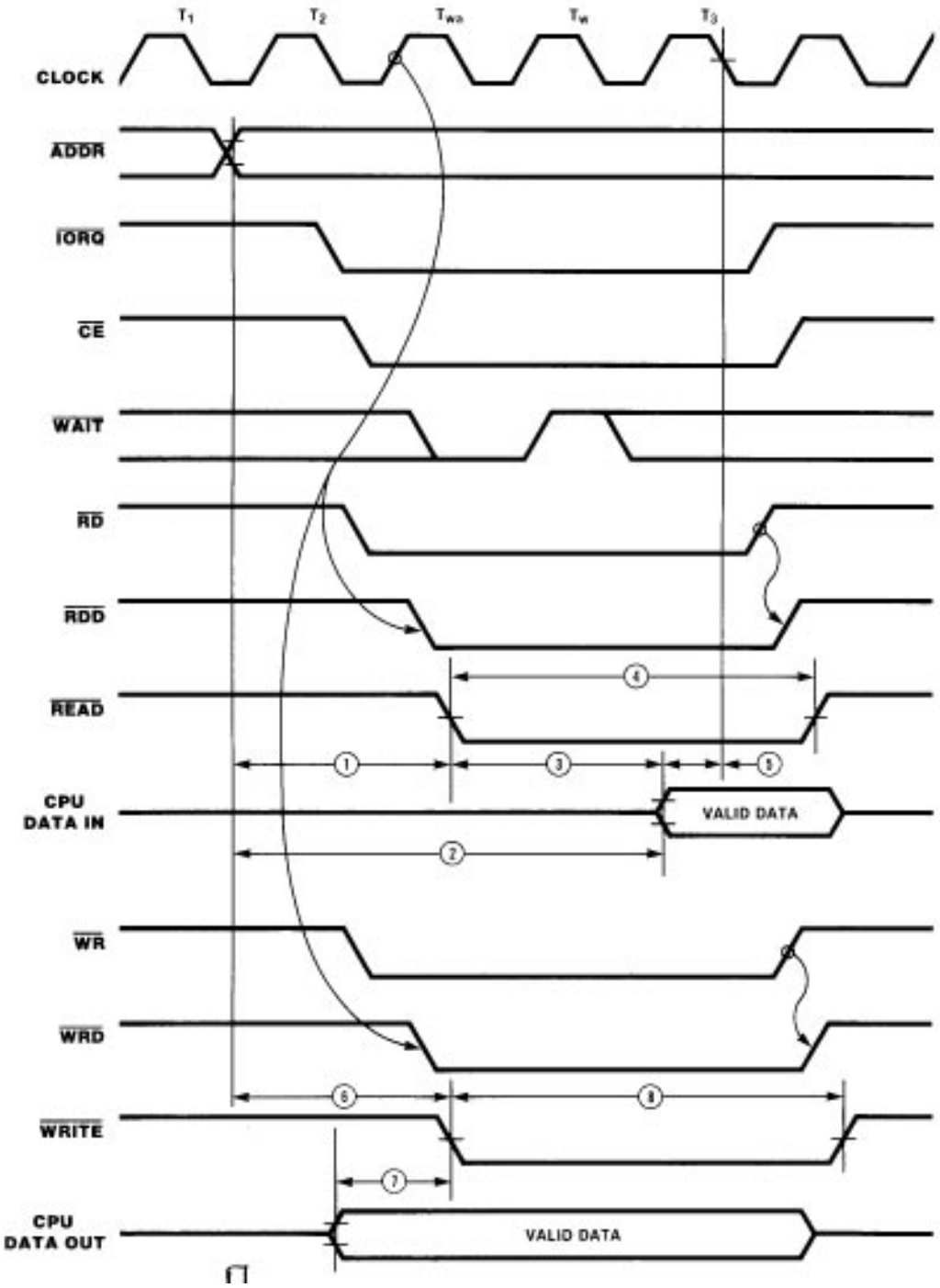

Figure 7. Z80H CPU to Z8500A Peripheral Minimum I/O Cycle Timing

Z80H CPU TO Z8500A PERIPHERALS (Continued)

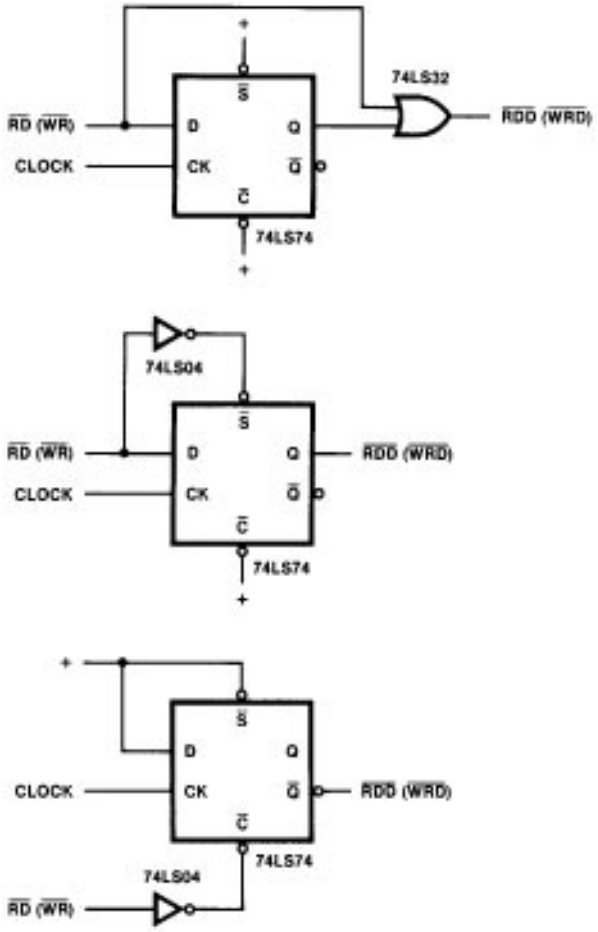

Figure 8. Delaying /RD or /WR

Table 14. Parameter Equations

Z80H Parameter	Z8500A Equation	Value	Units
TsD(Cf)	4TcC+TwCh-TdCr(A)-TdA(DR) /RS - delayed	55 min	ns
	2TcC+TwCh-TdRD(DR)	125 min	ns

INTERRUPT ACKNOWLEDGE CYCLES

The primary timing differences between the Z80 CPUs and Z8500 peripherals occur in the Interrupt Acknowledge cycle. The Z8500 timing parameters that are significant during Interrupt Acknowledge cycles are listed in Table 16, while the Z80 parameters are listed in Table 17. The reference numbers in Tables 16 and 17 refer to Figures 10, 12 and 13.

If the CPU and the peripherals are running at different speeds (as with the Z80H interface), the /INTACK signal must be synchronized to the peripheral clock. Synchronization is discussed in detail under Interrupt Acknowledge for Z80H CPU to Z8500/8500A Peripherals.

During an Interrupt Acknowledge cycle, Z8500 peripherals require both /INTACK and /RD to be active at certain times. Since the Z80 CPUs do not issue either /INTACK or /RD, external logic must generate these signals.

Generating these two signals is easily accomplished, but the Z80 CPU must be placed into a Wait condition until the peripheral interrupt vector is valid. If more peripherals are added to the daisy chain, additional Wait states may be necessary to give the daisy chain time to settle. Sufficient time between /INTACK active and /RD active should be allowed for the entire daisy chain to settle.

Since the Z8500 peripheral daisy chain does not use the IP flag except during interrupt acknowledge, there is no need for decoding the RETI instruction used by the Z80 peripherals. In each of the Z8500 peripherals, there are commands that reset the individual IUS flags.

EXTERNAL INTERFACE LOGIC

The following sections discuss external interface logic required during Interrupt Acknowledge cycles for each interface type.

CPU/Peripheral Same Speed

Figure 9 shows the logic used to interface the Z80A CPU to the Z8500 peripherals and the Z80B CPU to Z8500A peripherals during an Interrupt Acknowledge cycle. The primary component in this logic is the Shift register (74LS164), which generates /INTACK, /READ, and /WAIT.

Table 15. Z8500 Timing Parameters Interrupt Acknowledge Cycles

			4 MHz		6 MHz		
Worst Case			**Min**	**Max**	**Min**	**Max**	**Units**
1.	TsIA(PC)	/INTACK Low to PCLK High Setup	100		100		ns
	ThIA(PC)	/INTACK Low to PCLK High Hold	100		100		ns
2.	TdIAi(RD)	/INTACK Low to RD (Acknowledge) Low	350		250		ns
5.	TwRDA	/RD (Acknowledge) Width	350		250		ns
3.	TdRDA(DR)	/RD (Acknowledge) to Data Valid		250		180	ns
	TsIEI(RDA)	IEI to /RD (Acknowledge) Setup	120		100		ns
	ThIEI(RDA)	IEI to /RD (Acknowledge) Hold	100		70		ns
	TdIEI(IE)	IEI to IEO Delay		150		100	ns

.

Table 16. Z80 CPU Timing Parameters Interrupt Acknowledge Cycles

		4 MHz		6 MHz		8 MHz		
Worst Case		**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**
TdC(M1f)	Clock High to /M1 Low Delay		100		80		70	ns
TdM1f(IORQf)	/M1 Low to /IORQ Low Delay	575*		*345		275*		ns
4. TsD(Cr)	Data to Clock High Setup	35		30		25		ns
	*Z80A: 2TcC + TwCh + TfC - 65 Z80B: 2 TcC + TwCh + TfC - 50 Z80H: 2TcC + TwCh + TfC - 45							

EXTERNAL INTERFACE LOGIC (Continued)

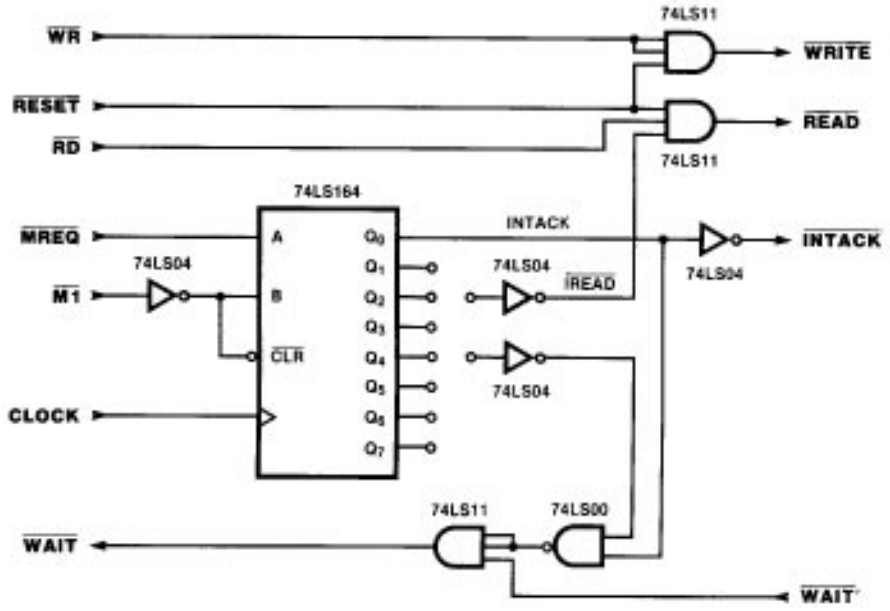

Figure 9. Z80A/Z80B CPU to Z8500/Z8500A Peripheral Interrupt Acknowledge Interface Logic

During I/O and normal memory access cycles, the Shift registers remains cleared because the /M1 signal is inactive. During opcode fetch cycles, also, the Shift register remains cleared, because only 0s can be clocked through the register. Since Shift register outputs are Low, /READ, /WRITE, and /WAIT are controlled by other system logic and gated through the AND gates (74LS11). During I/O and normal memory access cycles, /READ and /WRITE are active as a result of the system /RD and /WR signals (respectively) becoming active. If system logic requires that the CPU be placed into a Wait condition, the /WAIT signal controls the CPU. Should it be necessary to reset the system, /RESET causes the interface logic to generate both /READ and /WRITE (the Z8500 peripheral Reset condition).

Normally an Interrupt Acknowledge cycle is indicated by the Z80 CPU when /M1 and /IORQ are both active (which can be detected on the third rising clock edge after T1). To obtain an early indication of an Interrupt Acknowledge cycle, the Shift register decodes an active /M1 in the presence of an inactive /MREQ on the rising edge of T2.

During an Interrupt Acknowledge cycle, the /INTACK signal is generated on the rising edge of T2.

Since it is the presence of /INTACK and an active /READ that gates the interrupt vector onto the data bus, the logic must also generate /READ at the is Td1Ai(RD) /INTACK to /RD (Acknowledge) Low Delay]. This time delay allows the interrupt daisy chain to settle so that the device requesting the interrupt can place its interrupt vector onto the data bus. The shift register allows a sufficient time delay from the generation of /INTACK before it generates /READ. During this delay, it places the CPU into a Wait state until the valid interrupt vector can be placed onto the data bus. If the time between these two signals is insufficient for daisy chain settling, more time can be added by taking /READ and /WAIT from a later position on the Shift register.

Figure 10 illustrates Interrupt Acknowledge cycle timing resulting from the Z80A CPU to Z8500 peripheral and the Z80B CPU to A8500A peripheral interface. This timing comes from the logic illustrated in Figure 9, which can be used for both interfaces. Should more Wait states be required, the additional time can be calculated in terms of system clocks, since the CPU clock and PCLK are the same.

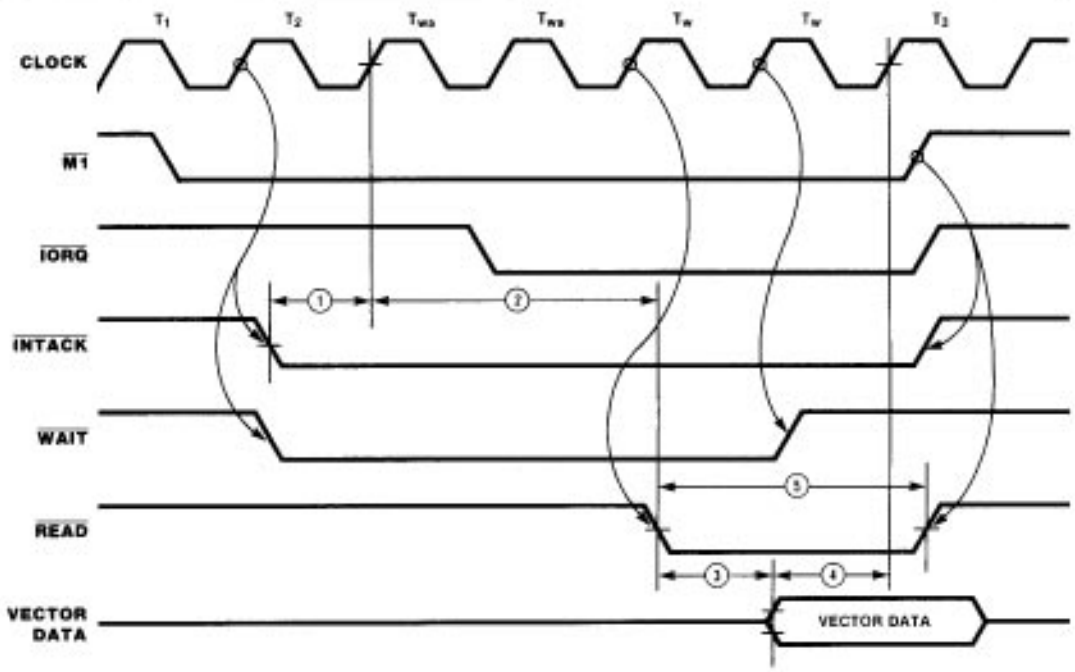

Figure 10. Z80A/Z80B CPU to Z8500/Z8500A Peripheral Interrupt Acknowledge Interface Timing

Z8500/Z8500A Peripherals

Figure 11 depicts logic that can be used in interfacing the Z80H CPU to the Z8500/Z8500A peripherals. This logic is the same as that shown in Figure 5, except that a synchronizing flip-flop is used to recognize an Interrupt Acknowledge cycle. Since Z8500 peripherals do not rely upon PCLK except during Interrupt Acknowledge cycles, synchronization need occur only at that time. Since the CPU and the peripherals are running at different speeds, /INTACK and /RD must be synchronized to the Z8500 peripherals clock.

During I/O and normal memory access cycles, the synchronizing flip-flop and the Shift register remain cleared because the /M1 signal is inactive. During opcode fetch cycles, the flip-flop and the Shift register again remain cleared, but this time because the /MREQ signal is active. The synchronizing flip-flop allows an Interrupt Acknowledge cycle to be recognized on the rising edge of T2 when /M1 is active and /MREQ is inactive, generating the INTA signal. When INTA is active, the Shift register can clock and generate /INTACK to the peripheral and /WAIT to the CPU. The Shift register delays the generation of /READ to the peripheral until the daisy chain settles. The /WAIT signal is removed when sufficient time has been allowed for the interrupt vector data to be valid.

Figure 12 illustrates Interrupt Acknowledge cycle timing for the Z80H CPU to Z8500 peripheral interface. Figure 13 illustrates Interrupt Acknowledge cycle timing for the Z80H CPU to Z8500A peripheral interface. These timing result from the logic in Figure 11. Should more Wait states be required, the needed time should be calculated in terms of PCLKs, not CPU clocks.

Z80 CPU to Z80 and Z8500 Peripherals

In a Z80 system, a combination of Z80 peripherals and Z8500 peripherals can be used compatibly. While there is no restriction on the placement of the Z8500 peripherals in the daisy chain, it is recommended that they be placed early in the chain to minimize propagation delays during RET1 cycles.

During an Interrupt Acknowledge cycle, the IEO line from Z8500 peripherals changes to reflect the interrupt status. Time should be allowed for this change to ripple through the remainder of the daisy chain before activating /IORQ to the Z80 peripherals, or /READ to the Z8500 peripherals.

EXTERNAL INTERFACE LOGIC (Continued)

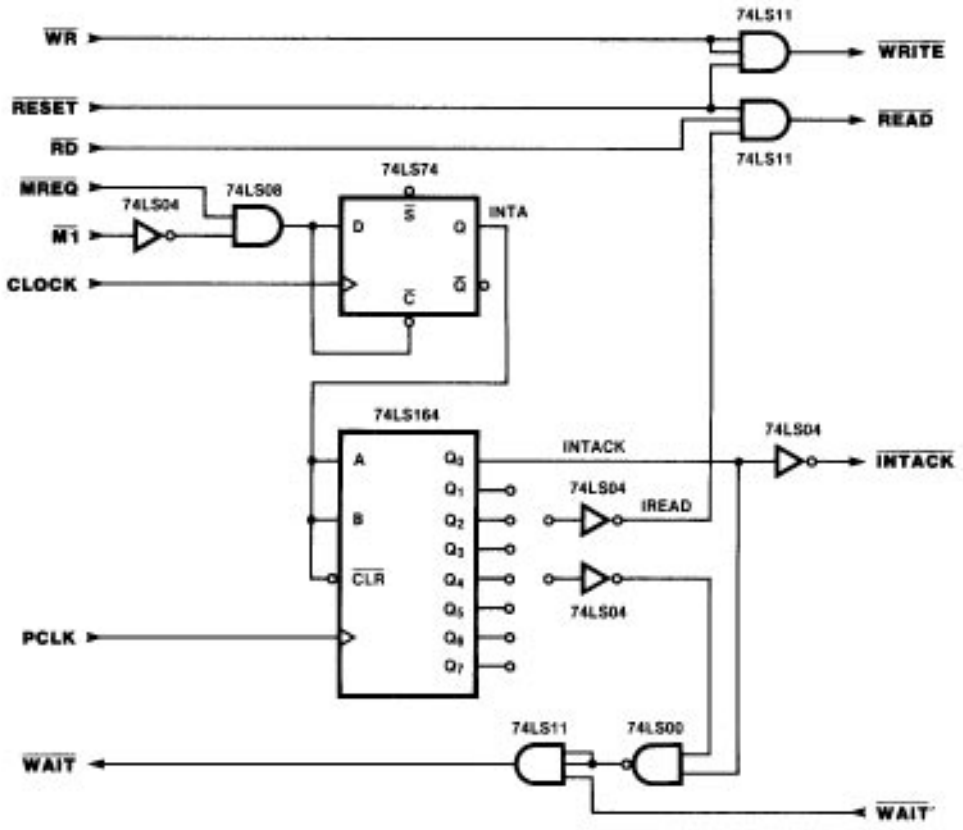

Figure 11. Z80H to Z8500/Z8500A Peripheral Interrupt Acknowledge Interface Logic

During RETI cycles, the IEO line from the Z8500 peripherals does not change state as in the Z80 peripherals. As long as the peripherals are at the top of the daisy chain, propagation delays are minimized.

The logic necessary to create the control signals for both Z80 and Z8500 peripherals is shown in Figure 9. This logic delays the generation of /IORQ to the Z80 peripherals by the same amount of time necessary to generate /READ for the Z8500 peripherals. Timing for this logic during an Interrupt Acknowledge cycle is depicted in Figure 10.

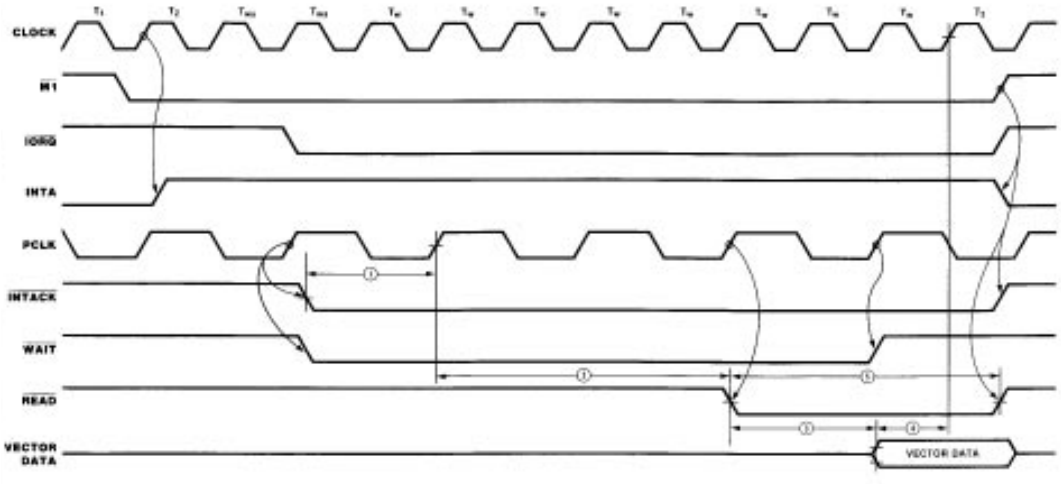

Figure 12. Z80H CPU to Z8500 Peripheral Interrupt Acknowledge Interface Timing

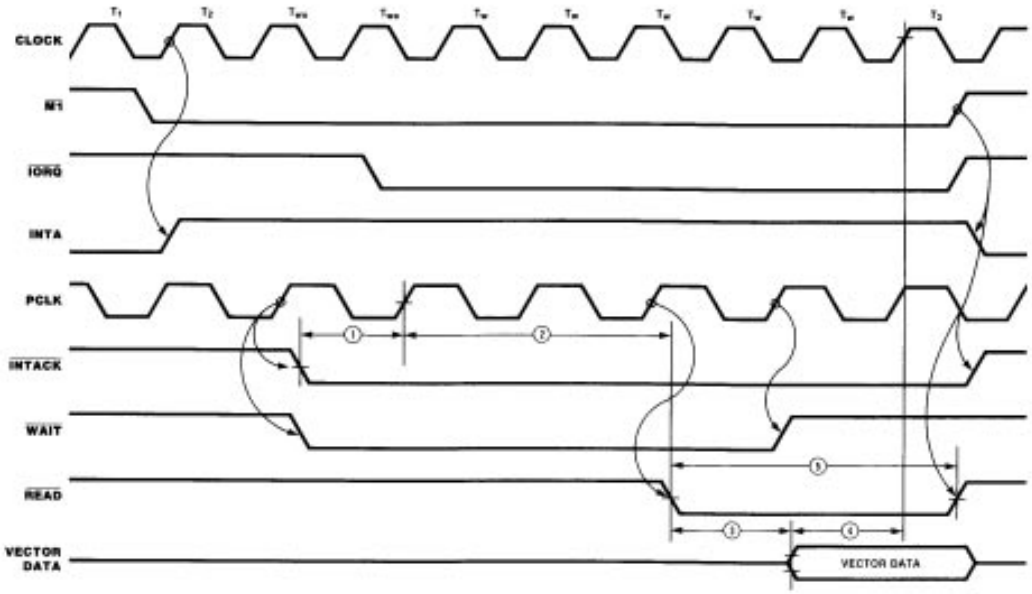

Figure 13. Z80H CPU to Z8500A Peripheral Interrupt Acknowledge Interface Timing

EXTERNAL INTERFACE LOGIC (Continued)

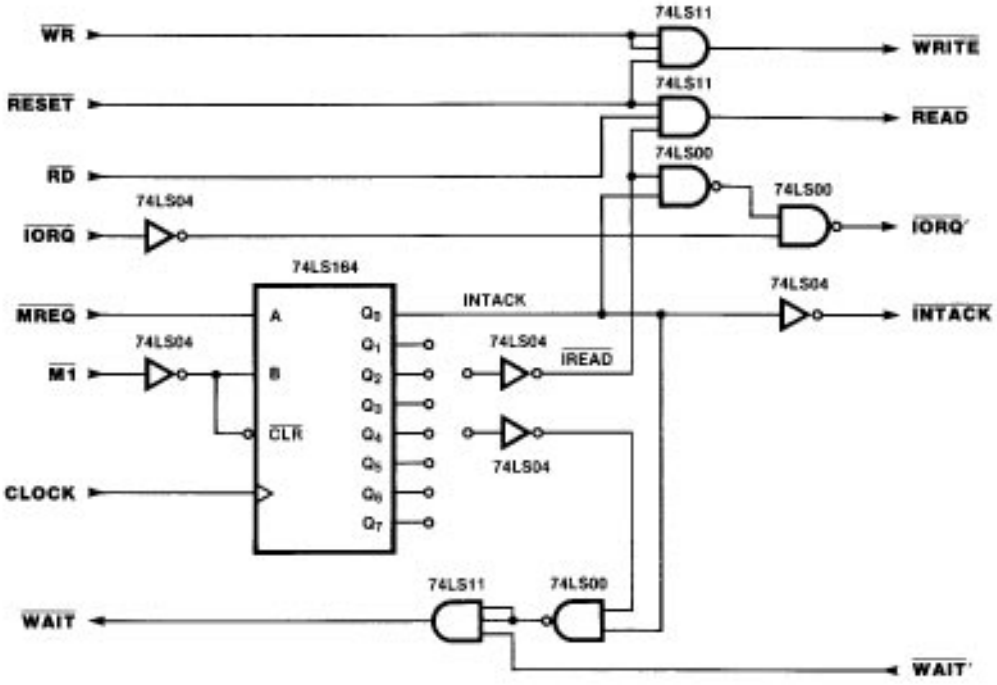

Figure 14. Z80 and Z8500 Peripheral Interrupt Acknowledge Interface Logic

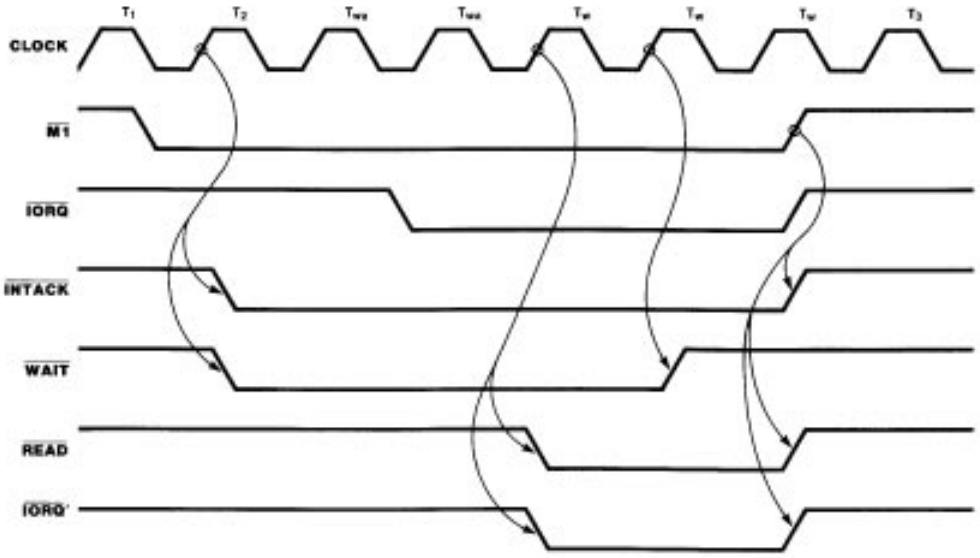

Figure 15. Z80 and Z8500 Peripheral Interrupt Acknowledge Interface Timing

SOFTWARE CONSIDERATIONS - POLLED OPERATION

There are several options available for servicing interrupts on the Z8500 peripherals. Since the vector of IP registers can be read at any time, software can be used to emulate the Z80 interrupt response. The interrupt vector read reflects the interrupt status condition even if the device is programmed to return to vector that does not reflect the status change (SAV or VIS is not set). The code below is a simple software routine that emulates the Z80 vector response operation.

Z80 Vector Interrupt Response, Emulation by Software

```
;This code emulates the Z80 vector interrupt
;operation by reading the device interrupt
;vector and forming an address from a vector
;table. It then executes an indirect jump to
;the interrupt service routine.

INDX:   LD      A,CIVREG        ;CURRENT INT. VECT. REG
        OUT     (CTRL), A       ;WRITE REG. PTR.
        IN      A, (CTRL)       ;READ VECT. REG.
        INC     A               ;VALID VECTOR?
        RET     Z               ;NO INT - RETURN
        AND     00001110B       ;MASK OTHER BITS
        LD      E,A
        LD      D,0             ;FORM INDEX VALUE
        LD      HL,VECTAB
        ADD     HL,DE           ;ADD VECT. TABLE ADDR.
        LD      A, (HL)         ;GET LOW BYTE
        INC     HL
        LD      H, (HL)         ;GET HIGH BYTE
        LD      L,A             ;FORM ROUTINE ADDR.
        JP      (HL)            ;JUMP TO IT

VECTAB:
        DEFW    INT1
        DEFW    INT2
        DEFW    INT3
        DEFW    INT4
        DEFW    INT5
        DEFW    INT6
        DEFW    INT7
        DEFW    INT8
```

A SIMPLE Z80-Z8500 SYSTEM

The Z8500 devices interface easily to the Z80 CPU, thus providing a system of considerable flexibility. Figure 16 illustrates a simple system using the Z80A CPU and Z8536 Counter/Timer and Parallel I/O Unit (CIO) in a mode 1 or non-interrupt environment. Since interrupt vectors are not used, the /INTACK line is tied High and no additional logic is needed. Because the CIO can be used in a polled interrupt environment, the /INT pin is connected to the CPU. The Z80 should not be set for mode 2 interrupts since the CIO will never place a vector onto the data bus. Instead, the CPU should be placed into mode 1 interrupt mode and a global interrupt service routine can poll the CIO to determine what caused the interrupt to occur. In this system, the software emulation procedure described above is effective.

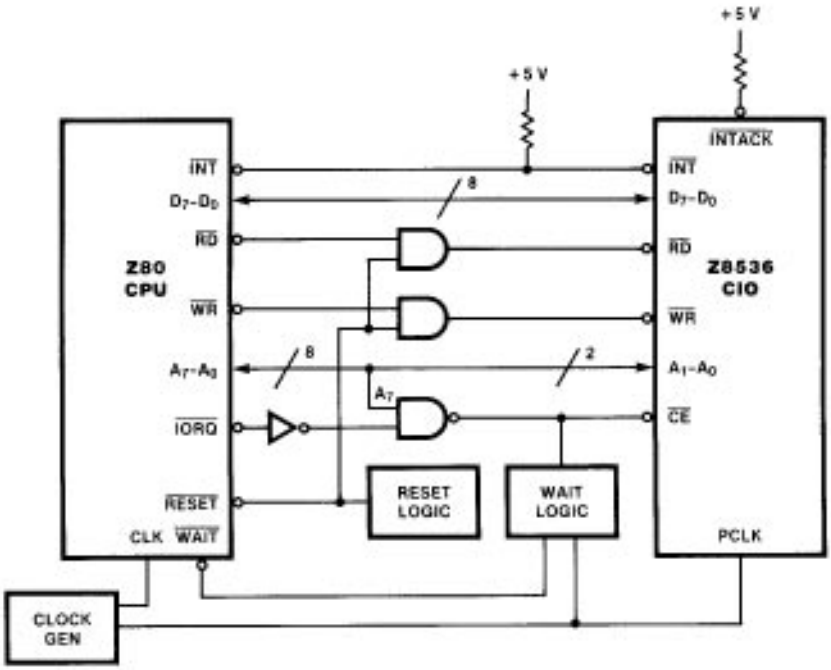

Figure 16. Z80 to Z8500 Simple System Mode 1 Interrupt or Non-Interrupt Structure

Additional Information in Zilog Publications:

The Z80 Family User's Manual includes technical information on the Z80 CPU, DMA, PIO, CTC, and SIO.

Technical information on the Z80 CPU AC Characteristics and the Z80 Family Interrupt Structure Tutorial can be found in the Z80 Databook.

The Z8000 User's Manual features technical information on the Z8536 CIO and Z8038 FIO.