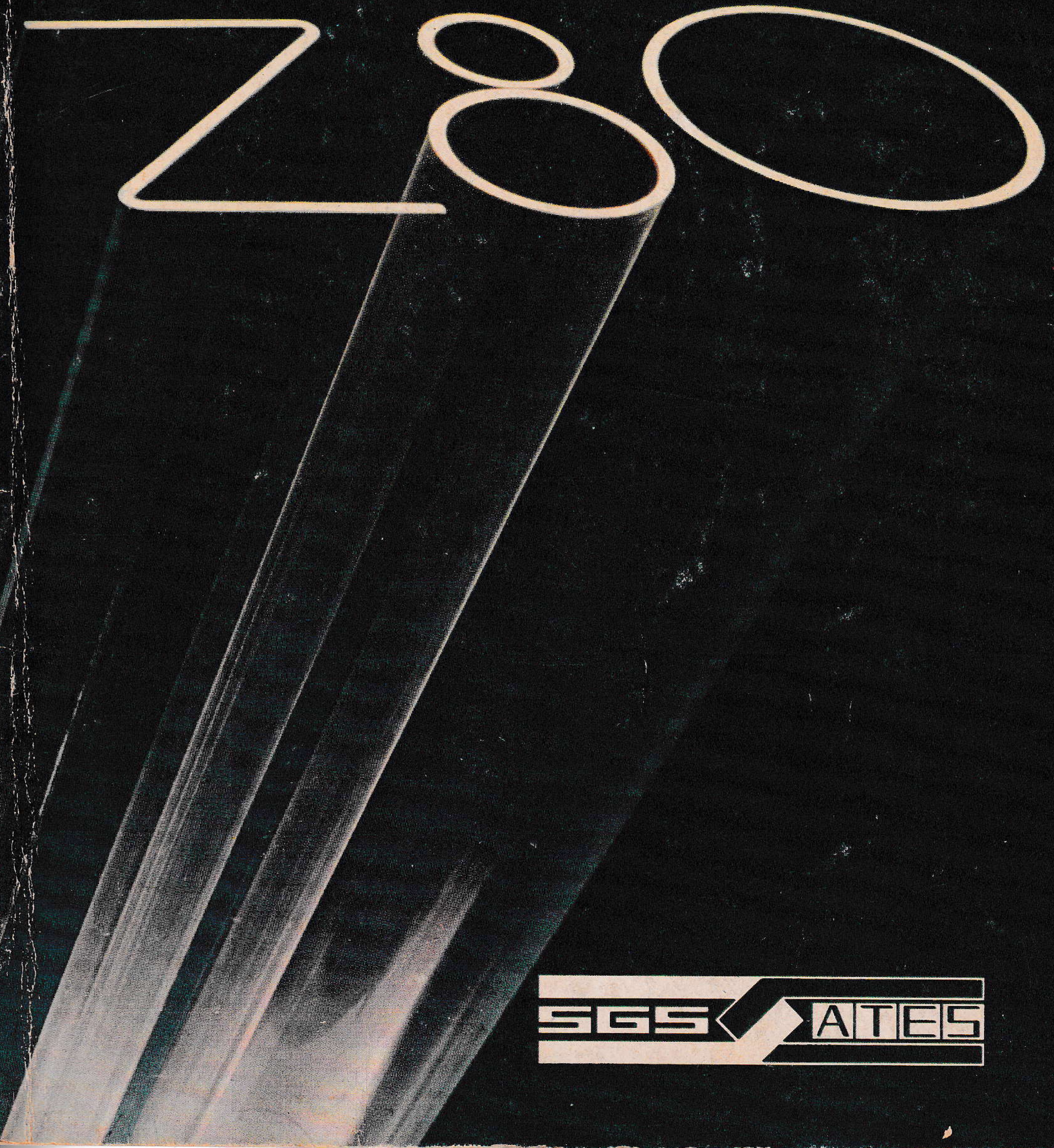
Z80 MICROCOMPUTER SYSTEM



SGS ATES

Z80™ MICROCOMPUTER SYSTEM

EUROCRATIC MOS

Since setting up its MOS department in 1966, SGS-ATES has led the way in European MOS technology.

Between the major landmarks of the first European-designed MOS calculator in 1968 and the F8 microprocessor in 1977, we brought you a full range of memories: 1K static and 4K dynamic RAMs, a 1K x 8 EPROM, a 1K x 8 ROM.....
and now we bring you the Z-80.

Not only the Z-80 but a team of experts dedicated to the development of the Z-80 device family, Z-80 systems, applications and interface devices.

Moreover, we've set up a comprehensive European network of "local" microcomputer application centres packed with the most up-to-date equipment available, staffed with highly-experienced software engineers and located in UK, Sweden, Italy, France and Germany.

SGS-ATES and Zilog: a vast reserve of know-how and resources committed to the advancement of microprocessors - stay with us and be part of the Z-80 conquest.

TM: Z80 is a registered trademark of Zilog, Inc.

CONTENTS

Z–80 MICROCOMPUTER PRODUCT LINE

Introduction

The Z-80 LSI component set includes all of the logic circuits necessary for the user to build high performance microcomputer systems with virtually no external logic and an absolute minimum number of lowest cost standard memory components. The Z-80 component set is backed by advanced software, a disk based hardware/software development system and complete training and support. The entire Z-80 product line has been developed as a single, highly integrated entity to insure that the user can develop his system quickly and still obtain all the performance advantages of the Z-80 component set.

High System Throughput

The architecture of the Z-80 CPU includes a superset of 158 instructions, with more internal registers and addressing modes than second generation microcomputers and extremely fast interrupt response time. All of these features mean that in any given amount of time the Z-80 can perform *far* more work (processor throughput) than any other micro-computer system available today. This throughput advantage allows users to continually expand the features and capabilities of their systems without increasing hardware costs.

Low Memory Costs

One of the major features of the Z-80 CPU is that it greatly reduces system memory costs. The expanded set of 158 software instructions results in a tremendous reduction in the memory required for any typical application. In addition, the Z-80 CPU provides all refresh and timing signals to directly drive dynamic memories so that the Z-80 LSI components can interface to most standard 4K dynamic memories with virtually no external logic. The Z-80 CPU uses a technique whereby the memory address is generated very early in memory cycles, permitting the high speed Z-80 CPU to operate with standard speed memories, again reducing system memory costs. The Z-80 CPU was designed to operate with standard memory products from any source since these devices will always be less expensive than custom memories designed for any particular microcomputer.

Low I/O Costs

The Z-80 LSI component set includes four general purpose programmable I/O circuits that contain all of the logic required to implement fast I/O transfers with minimal CPU overhead. These circuits have a built-in ripple priority interrupt control circuit (the device closest to the CPU has the highest priority) and all the logic necessary for nesting of interrupts to any level. Using the programmable features of these circuits, the user can configure the devices to interface with a wide range of peripheral devices with virtually no other external logic. These features make the peripheral device controllers in a Z-80 system much simpler and therefore lower in cost.

Low System Hardware Costs

The Z-80 component set requires very little support circuitry. All devices require a single +5 volt power supply and a single phase TTL clock. In addition, all control signals are directly compatible with I/O and memory devices so that system control circuits are not required. External interrupt control circuits are not required since these are included in every Z-80 I/O circuit. DMA circuits are generally not required due to an extremely fast interrupt response and powerful I/O block transfer capability within the CPU.

Low Development Costs

SGS-ATES offers more than a fully integrated line of LSI components. Everything is provided that is necessary for the user to easily develop his own proprietary system using the Z-80 components. This includes complete software packages, disk based development systems and training. For example, the expanded Z-80 software instruction set coupled with the easy to learn Z-80 assembly language and reference cards make assembly language programming much easier than previously possible. For larger programs, PL/Z may be used to speed up the development cycle, to enhance program documentation and to improve program maintainability.

Z-80 MICROCOMPUTER SUMMARY

Central Processor Unit/Z-80-CPU

□ Single chip, N-channel processor

□ 158 instructions - Includes all 78 of the 8080A instructions with *total* software compatibility. New instructions include 4-, 8- and 16-bit operations with more useful addressing modes.

□ 17 internal registers (more than twice the 8080A registers), including two real index registers.

□ Three modes of fast interrupt response plus a nonmaskable interrupt.

□ Directly interfaces standard speed static or dynamic memories with virtually no external logic.

□ 1.6μs instruction execution speed.

□ Single 5V supply and single-phase TTL Clock.

□ Out-performs any other microcomputer in 4-4-, 8-, 16-bit applications.

□ Requires 25% to 50% less memory space than the 8080A CPU.

□ Up to 500% more throughput than the 8080A.

□ TTL compatible tri-state data and address busses.

Interface and Control Circuits

Parallel Input/Output Controller/Z-80-PIO

Programmablc circuit that allows for a direct interface to a wide range of parallel interface peripherals without other external logic.

Serial Input/Output Controller/Z-80-SIO

Programmable circuit that allows for a direct interface to a wide range of serial interface peripherals without other external logic.

Counter Timer Circuit/Z-80-CTC

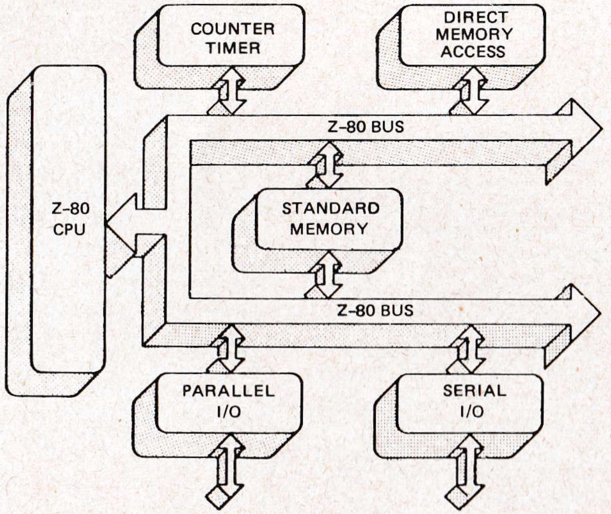
Contains four independent programmable counter timer circuits for control of real time events.

Direct Memory Access Controller/Z-80-DMA

Programmable circuit that can directly transfer data between the SIO or PIO and memory on a CPU cycle steal basis.

All Z-80 controllers have built in nested priority interrupt control and fast interrupt response capability (up to 6 times faster than the 8080A).

All Z-80 controllers monitor peripheral status to eliminate any type of CPU polling.



Z-80 COMPONENTS

Introduction

The SGS-ATES third generation microcomputer components are the most advanced and comprehensive set of LSI microcomputer products available today. The major components in the Z-80 product line are an extremely high performance central processing unit (CPU), a programmable parallel input/output controller (PIO), a programmable serial input/output controller (SIO), a versatile counter timer circuit (CTC) and a high speed direct memory access controller (DMA).

All of the Z-80 components utilize the industry standard N-channel silicon gate technology to provide the highest density at the lowest cost. Depletion load technology is also used to provide high performance with a single 5V power supply.

The CPU, PIO, SIO and DMA are packages in standard 40-pin DIPs; the CTC comes in a standard 28-pin DIP. All require only a single 5V power supply plus the Z-80 single-phase TTL level clock.

Z-80 CPU

The Z-80 CPU is an extremely powerful, third generation CPU which incorporates a number of major features over the standard 8080A CPU while retaining total software compatibility. Major improvements include: □ More than twice as many registers on the CPU chip, including two real index registers □ Many more addressing modes □ More than twice as many instructions □ Three modes of extremely fast interrupt response □ A separate non-maskable interrupt to a fixed location.

Another unique feature of the Z-80 CPU is its ability to generate all of the control signals for standard memory circuits. Static memories can be interfaced using only an external address decoder for chip selects. In addition the Z-80 CPU provides all of the refresh control for dynamic memories, and the Z-80 control bus timing signals are directly compatible with all widely used, standard speed, 18- and 22-pin 4K RAMs (16-pin 4K RAMs require only an external address multiplexer). Thus dynamic RAMs can be interfaced with virtually no additional external logic. This provides the user with the ability to easily interface to the lowest cost dynamic memories without reducing CPU operational speed.

By selecting the best standard memory for a given application, the user can reduce his product manufacturing costs, and the product development expenses will also be much lower.

The Z-80 CPU is designed to be totally software compatible with the standard 8080A microprocessor to facilitate the user's transition to the Z-80. By using the Z-80 component set and the most economical memory for the particular application, the user need only re-layout any 8080 based design and use any existing software programs to obtain an immediate and very significant reduction in sysíem hardware costs. A major advantage is that the same ROMs that are used in the 8080 system can be used in the Z-80 system. At a later date the software programs can be upgraded, taking advantage of the powerful Z-80 instruction set and the full capability of the Z-80 component set to obtain increased performance and even further cost reduction for memory components.

The Z-80 CPU is an extremely fast and versatile device. Full instruction cycle times for non-memory reference instructions are 1.6μs and the CPU responds to interrupts very rapidly (the 8080 requires up to 6 times as long to respond, and uses more than twice as much memory storage). This fast interrupt response, in conjunction with new I/O block transfer instructions, allows the CPU to directly control many peripherals without the costly use of DMA hardware and it greatly reduces the size of software routines required for peripheral control, again saving memory space and costs.

Probably the most important feature of the Z-80 microprocessor family is its repertoire of 158 software instructions. The original 78 instructions of the 8080A CPU are included using the same OP codes; thus, the Z-80 can execute 8080 or 8080A programs stored in existing ROMs. The Z-80 new software instructions provide an expanded capability for the user, such as: □ Additional addressing modes, including indexed and relative □ Memory to memory block transfers and searches □ Bit manipulation and testing in any register or memory location □ Many new I/O instructions, including block I/O transfers □ A wide range of memory or register rotates and shifts (logical and arithmetic) □ Expanded 16-bit arithmetic □ Expanded BCD arithmetic.

Parallel Input/Output (PIO)

The Z-80 PIO circuit uses an advanced interrupt driven, program controlled I/O transfer technique for easy handling of virtually any peripheral with a parallel interface. Without other logic, the PIO can interface most line printers, paper tape readers or punchers, card readers, keyboards, electronic typewriters and other similar devices.

The PIO contains all of the interrupt control logic necessary for nested priority interrupt handling with very fast response time. Thus additional interrupt control circuits are not needed and servicing time is minimized. The parallel I/O can handle two high speed I/O ports, and it interrupts the CPU after each I/O transfer is complete.

The PIO circuit include two independent ports, each with eight I/O lines and two handshake lines which are programmed by the CPU to operate in one of four modes: □ Byte output with interrupt driven handshake □ Byte input with interrupt driven handshake □ Bidirectional byte bus with interrupt driven handshake □ Control mode wherein any bit can be programmed as an input or output.

A major feature of the PIO is its ability to generate an interrupt on any bit pattern at the I/O pins, thus eliminating the need for the processor to constantly test I/O lines for a particular peripheral status condition. This feature greatly enhances the ability of the processor to easily handle peripherals, while also reducing software overhead.

Serial Input/Output (SIO)

The SIO circuit is a programmable I/O device similar in concept to the PIO, except that it is designed to handle peripherals with a serial data interface such as floppy disks, CRTs and communication terminals. Each SIO circuit can handle a full duplex serial I/O channel. The device will handle data that is asynchronous with 5- to 8-bit characters and with 1, $1\frac{1}{2}$ or 2 stop bits. The SIO will handle 5- to 8-bit synchronous data including IBM BiSync and SDL communication channels. CRC generation and parity checking are also included.

Counter Timer Circuit (CTC)

The CTC circuit contains four versatile clocks, each with its own nested priority interrupt control. All clocks have a minimum resolution of 8μs and can generate interrupts in the range of 8μs to 32 ms. The circuit may also be used in a mode in which it counts external events. Another major feature is that an interrupt can be programmed to occur after the occurrence of an external event. The four timing circuits greatly ease the CPU software handling requirements for many real-time control applications. For example, the CTC allows the implementation of a very low-cost TTY or CRT I/O port, and simple sector control of floppy disk subsystems.

Direct Memory Access Controller (DMA)

The DMA circuit is provided for those applications in which data must be transferred directly into memory at a very high rate rather than going through the central processor unit. This circuit is not needed for most applications due to the fast interrupt response and block transfer capabilities of the Z-80 CPU. However, in large systems applications with many high speed peripherals, such as floppy disks, communications channels, etc., the DMA circuit can greatly improve system performance by totally controlling block transfers between I/O circuits and the system memory.

The DMA circuit contains all control for four I/O circuits including a block length counter and a memory address pointer. The circuits also have a ripple priority chain so that virtually any number of DMA channels can be implemented. The DMA circuit communicates directly between the I/O circuits and the system memory after obtaining a DMA acknowledge signal from the CPU.

Product Specification

The SGS-ATES Z80 product line is a complete set of microcomputer components, development systems and support software. The Z80 microcomputer component set includes all of the circuits necessary to build high-performance microcomputer systems with virtually no other logic and a minimum number of low cost standard memory elements.

The Z80 and Z80A CPU's are third generation single chip microprocessors with unrivaled computational power. This increased computational power results in higher system through-put and more efficient memory utilization when compared to second generation microprocessors. In addition, the Z80 and Z80A CPU's are very easy to implement into a system because of their single voltage requirement plus all output signals are fully decoded and timed to control standard memory or peripheral circuits. The circuit is implemented using an N-channel, ion implanted, silicon gate MOS process.

Figure 1 is a block diagram of the CPU, Figure 2 details the internal register configuration which contains 208 bits of Read/Write memory that are accessible to the programmer. The registers include two sets of six general purpose registers that may be used individually as 8-bit registers or as 16-bit register pairs. There are also two sets of accumulator and flag registers. The programmer has access to either set of main or alternate registers through a group of exchange instructions. This alternate set allows foreground/background mode of operation or may be reserved for very fast Interrupt response. Each CPU also contains a 16-bit stack pointer which permits simple implementation of multiple level interrupts, unlimited subroutine nesting and simplification of many types of data handling.

The two 16-bit index registers allow tabular data manipulation and easy implementation of relocatable code. The Refresh register provides for automatic, totally transparent refresh of external dynamic memories. The I register is used in a powerful interrupt response mode to form the upper 8 bits of a pointer to a interrupt service address table, while the interrupting device supplies the lower 8 bits of the pointer. An indirect call is then made to this service address.

FEATURES

- Single chip, N-channel Silicon Gate CPU.
- 158 instructions—includes all 78 of the 8080A instructions with total software compatibility. New instructions include 4-, 8- and 16-bit operations with more useful addressing modes such as indexed, bit and relative.
- 17 internal registers.
- Three modes of fast interrupt response plus a nonmaskable interrupt.
- Directly interfaces standard speed static or dynamic memories with virtually no external logic.
- 1.0 μs instruction execution speed.
- Single 5 VDC supply and single-phase 5 volt Clock.
- Out-performs any other single chip microcomputer in 4-, 8-, or 16-bit applications.
- All pins TTL Compatible
- Built-in dynamic RAM refresh circuitry.

Fig. 1 - Z80, Z80A CPU BLOCK DIAGRAM

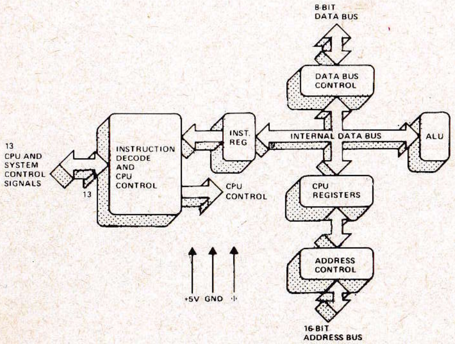
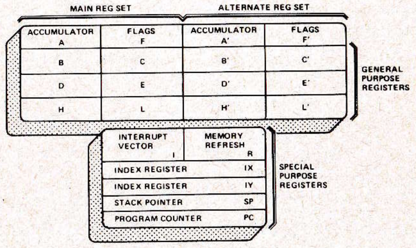
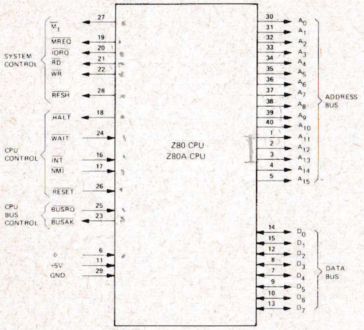


Fig. 2 - Z80, Z80A CPU REGISTERS



Pin Description

PIN CONFIGURATION



A_0-A_{15} (Address Bus) Tri-state output, active high. A_0-A_{15} constitute a 16-bit address bus. The address bus provides the address for memory (up to 64K bytes) data exchanges and for I/O device data exchanges.

D_0-D_7 (Data Bus) Tri-state input/output, active high. D_0 - D_7 constitute an 8-bit bidirectional data bus. The data bus is used for data exchanges with memory and I/O devices.

$\overline{M_1}$ (Machine Cycle one) Output, active low. $\overline{M_1}$ indicates that the current machine cycle is the OP code fetch cycle of an instruction execution.

$\overline{MREQ}$ (Memory Request) Tri-state output, active low. The memory request signal indicates that the address bus holds a valid address for a memory read or memory write operation.

$\overline{IORQ}$ (Input/ Output Request) Tri-state output, active low. The $\overline{IORQ}$ signal indicates that the lower half of the address bus holds a valid I/O address for a I/O read or write operation. An $\overline{IORQ}$ signal is also generated when an interrupt is being acknowledged to indicate that an interrupt response vector can be placed on the data bus.

$\overline{RD}$ (Memory Read) Tri-state output, active low. $\overline{RD}$ indicates that the CPU wants to read data from memory or an I/O device. The addressed I/O device or memory should use this signal to gate data onto the CPU data bus.

$\overline{WR}$ (Memory Write) Tri-state output, active low. $\overline{WR}$ indicates that the CPU data bus holds valid data to be stored in the addressed memory or I/O device.

$\overline{RFSH}$ (Refresh) Output, active low. $\overline{RFSH}$ indicates that the lower 7 bits of the address bus contain a refresh address for dynamic memories and the current $\overline{MREQ}$ signal should be used to do a refresh read to all dynamic memories.

$\overline{HALT}$ (Halt state) Output, active low. $\overline{HALT}$ indicates that the CPU has executed a HALT software instruction and is awaiting either a non-maskable or a maskable interrupt (with the mask enabled) before operation can resume. While halted, the CPU executes NOP's to maintain memory refresh activity.

$\overline{WAIT}$ (Wait) Input, active low. $\overline{WAIT}$ indicates to the Z-80 CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter wait states for as long as this signal is active.

$\overline{INT}$ (Interrupt Request) Input, active low. The Interrupt Request signal is generated by I/O devices. A request will be honored at the end of the current instruction if the internal software controlled interrupt enable flip-flop (IFF) is enabled.

$\overline{NMI}$ (Non Maskable Interrupt) Input, active low. The non-maskable interrupt request line has a higher priority than $\overline{INT}$ and is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop. $\overline{NMI}$ automatically forces the Z-80 CPU to restart to location 0066_H.

$\overline{RESET}$ Input, active low. $\overline{RESET}$ initializes the CPU as follows: reset interrupt enable flip-flop, clear PC and registers I and R and set interrupt to 8080A mode. During reset time, the address and data bus go to a high impedance state and all control output signals go to the inactive state.

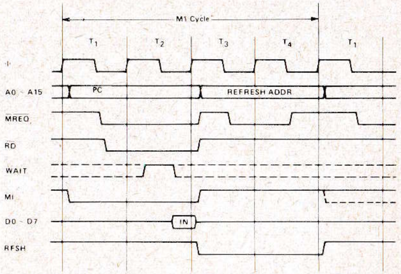
$\overline{BUSRQ}$ (Bus Request) Input, active low. The bus request signal has a higher priority than $\overline{NMI}$ and is always recognized at the end of the current machine cycle and is used to request the CPU address bus, data bus and tri-state output control signals to go to a high impedance state so that other devices can control these busses.

$\overline{BUSAK}$ (Bus Acknowledge) Output, active low. Bus acknowledge is used to indicate to the requesting device that the CPU address bus, data bus and tri-state control bus signals have been set to their high impedance state and the external device can now control these signals.

Timing Waveforms

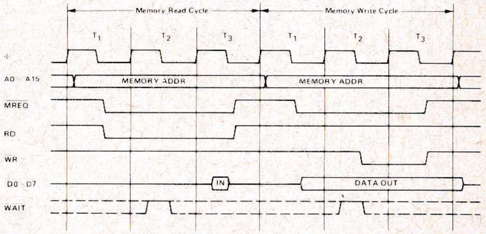
INSTRUCTION OP CODE FETCH

The program counter content (PC) is placed on the address bus immediately at the start of the cycle. One half clock time later $\overline{MREQ}$ goes active. The falling edge of $\overline{MREQ}$ can be used directly as a chip enable to dynamic memories. $\overline{RD}$ when active indicates that the memory data should be enabled onto the CPU data bus. The CPU samples data with the rising edge of the clock state T_3. Clock states T_3 and T_4 of a fetch cycle are used to refresh dynamic memories while the CPU is internally decoding and executing the instruction. The refresh control signal $\overline{RFSH}$ indicates that a refresh read of all dynamic memories should be accomplished.



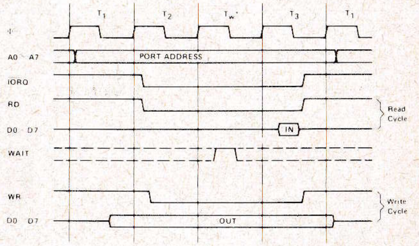
MEMORY READ OR WRITE CYCLES

Illustrated here is the timing of memory read or write cycles other than an OP code fetch (M_1 cycle). The $\overline{MREQ}$ and $\overline{RD}$ signals are used exactly as in the fetch cycle. In the case of a memory write cycle, the $\overline{MREQ}$ also becomes active when the address bus is stable so that it can be used directly as a chip enable for dynamic memories. The $\overline{WR}$ line is active when data on the data bus is stable so that it can be used directly as a R/W pulse to virtually any type of semiconductor memory.



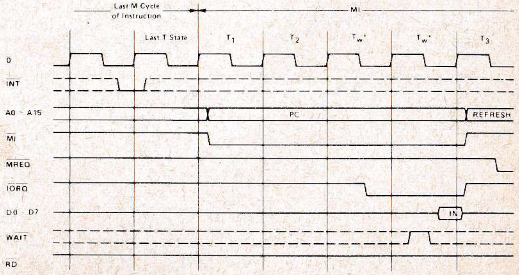
INPUT OR OUTPUT CYCLES

Illustrated here is the timing for an I/O read or I/O write operation. Notice that during I/O operations a single wait state is automatically inserted (Tw*). The reason for this is that during I/O operations this extra state allows sufficient time for an I/O port to decode its address and activate the $\overline{WAIT}$ line if a wait is required.



INTERRUPT REQUEST/ACKNOWLEDGE CYCLE

The interrupt signal is sampled by the CPU with the rising edge of the last clock at the end of any instruction. When an interrupt is accepted, a special M_1 cycle is generated. During this M_1 cycle, the $\overline{IORQ}$ signal becomes active (instead of $\overline{MREQ}$) to indicate that the interrupting device can place an 8-bit vector on the data bus. Two wait states (Tw*) are automatically added to this cycle so that a ripple priority interrupt scheme, such as the one used in the Z80 peripheral controllers, can be easily implemented.



Instruction Set

The following is a summary of the Z80, Z80A instruction set showing the assembly language mnemonic and the symbolic operation performed by the instruction. A more detailed listing appears in the Z80-CPU technical manual, and assembly language programming manual. The instructions are divided into the following categories:

8-bit loads
16-bit loads
Exchanges
Memory Block Moves
Memory Block Searches
8-bit arithmetic and logic
16-bit arithmetic
General purpose Accumulator
& Flag Operations
Miscellaneous Group
Rotates and Shifts
Bit Set, Reset and Test
Input and Output
Jumps
Calls
Restarts
Returns

In the table the following terminology is used.

b ≡ a bit number in any 8-bit register or memory location
cc ≡ flag condition code
NZ ≡ non zero
Z ≡ zero
NC ≡ non carry
C ≡ carry
PO ≡ Parity odd or no over flow
PE ≡ Parity even or over flow
P ≡ Positive
M ≡ Negative (minus)
d ≡ any 8-bit destination register or memory location
dd ≡ any 16-bit destination register or memory location
e ≡ 8-bit signed 2's complement displacement used in relative jumps and indexed addressing
L ≡ 8 special call locations in page zero. In decimal notation these are 0, 8, 16, 24, 32, 40, 48 and 56
n ≡ any 8-bit binary number
nn ≡ any 16-bit binary number
r ≡ any 8-bit general purpose register (A, B, C, D, E, H, or L)
s ≡ any 8-bit source register or memory location
s_b ≡ a bit in a specific 8-bit register or memory location
ss ≡ any 16-bit source register or memory location
subscript "L" ≡ the low order 8 bits of a 16-bit register
subscript "H" ≡ the high order 8 bits of a 16-bit register
( ) ≡ the contents within the ( ) are to be used as a pointer to a memory location or I/O port number

8-bit registers are A, B, C, D, E, H, L, I and R
16-bit register pairs are AF, BC, DE and HL
16-bit registers are SP, PC, IX and IY

Addressing Modes implemented include combinations of the following:

Immediate
Immediate extended
Modified Page Zero
Relative
Extended
Indexed
Register
Implied
Register Indirect
Bit

	Mnemonic	Symbolic Operation	Comments
8-BIT LOADS	LD r, s	r ← s	s ≡ r, n, (HL), (IX+e), (IY+e)
	LD d, r	d ← r	d ≡ (HL), r (IX+e), (IY+e)
	LD d, n	d ← n	d ≡ (HL), (IX+e), (IY+e)
	LD A, s	A ← s	s ≡ (BC), (DE), (nn), I, R
	LD d, A	d ← A	d ≡ (BC), (DE), (nn), I, R
16-BIT LOADS	LD dd, nn	dd ← nn	dd ≡ BC, DE, HL, SP, IX, IY
	LD dd, (nn)	dd ← (nn)	dd ≡ BC, DE, HL, SP, IX, IY
	LD (nn), ss	(nn) ← ss	ss ≡ BC, DE, HL, SP, IX, IY
	LD SP, ss	SP ← ss	ss = HL, IX, IY
	PUSH ss	(SP-1) ← ss_H; (SP-2) ← ss_L	ss = BC, DE, HL, AF, IX, IY
	POP dd	dd_L ← (SP); dd_H ← (SP+1)	dd = BC, DE, HL, AF, IX, IY
EXCHANGES	EX DE, HL	DE ↔ HL	
	EX AF, AF′	AF ↔ AF′	
	EXX	$\begin{pmatrix}BC\\DE\\HL\end{pmatrix} \leftrightarrow \begin{pmatrix}BC'\\DE'\\HL'\end{pmatrix}$	
	EX (SP), ss	(SP) ↔ ss_L, (SP+1) ↔ ss_H	ss ≡ HL, IX, IY

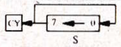
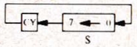
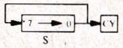
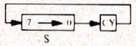
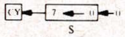
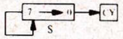
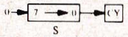
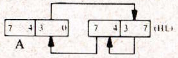
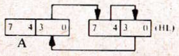
	Mnemonic	Symbolic Operation	Comments
MEMORY BLOCK MOVES	LDI	(DE) ← (HL), DE ← DE+1 HL ← HL+1, BC ← BC-1	
	LDIR	(DE) ← (HL), DE ← DE+1 HL ← HL+1, BC ← BC-1 Repeat until BC = 0	
	LDD	(DE) ← (HL), DE ← DE-1 HL ← HL-1, BC ← BC-1	
	LDDR	(DE) ← (HL), DE ← DE-1 HL ← HL-1, BC ← BC-1 Repeat until BC = 0	
MEMORY BLOCK SEARCHES	CPI	A-(HL), HL ← HL+1 BC ← BC-1	

	CPIR	A-(HL), HL ← HL+1
		BC ← BC-1, Repeat
		until BC = 0 or A = (HL)
			A-(HL) sets the flags only. A is not affected

	CPD	A-(HL), HL ← HL-1 BC ← BC-1	

	CPDR	A-(HL), HL ← HL-1
		BC ← BC-1, Repeat
		until BC= 0 or A = (HL)	

8-BIT ALU	ADD s	A ← A + s	
	ADC s	A ← A + s + CY	CY is the carry flag
	SUB s	A ← A - s	
	SBC s	A ← A - s - CY	s ≡ r, n, (HL) (IX+e), (IY+e)
	AND s	A ← A ∧ s	
	OR s	A ← A ∨ s	
	XOR s	A ← A ⊕ s	

	Mnemonic	Symbolic Operation	Comments
8-BIT ALU	CP s	A - s	s = r, n (HL) (IX+e), (IY+e)
	INC d	d ← d + 1	d = r, (HL) (IX+e), (IY+e)
	DEC d	d ← d - 1	
16-BIT ARITHMETIC	ADD HL, ss	HL ← HL + ss	ss ≡ BC, DE HL, SP
	ADC HL, ss	HL ← HL + ss + CY	
	SBC HL, ss	HL ← HL - ss - CY	
	ADD IX, ss	IX ← IX + ss	ss ≡ BC, DE, IX, SP
	ADD IY, ss	IY ← IY + ss	ss ≡ BC, DE, IY, SP
	INC dd	dd ← dd + 1	dd ≡ BC, DE, HL, SP, IX, IY
	DEC dd	dd ← dd - 1	dd ≡ BC, DE, HL, SP, IX, IY
GP ACC. & FLAG	DAA	Converts A contents into packed BCD following add or subtract.	Operands must be in packed BCD format
	CPL	A ← $\overline{A}$	
	NEG	A ← 00 – A	
	CCF	CY ← $\overline{CY}$	
	SCF	CY ← 1	
MISCELLANEOUS	NOP	No operation	
	HALT	Halt CPU	
	DI	Disable Interrupts	
	EI	Enable Interrupts	
	IM 0	Set interrupt mode 0	8080A mode
	IM 1	Set interrupt mode 1	Call to 0038_H
	IM 2	Set interrupt mode 2	Indirect Call
ROTATES AND SHIFTS	RLC s		
	RL s		
	RRC s		
	RR s		
	SLA s		s ≡ r, (HL) (IX+e), (IY+e)
	SRA s		
	SRL s		
	RLD		
	RRD		

	Mnemonic	Symbolic Operation	Comments
BIT S, R, & T	BIT b, s	Z ← $\overline{s_b}$	Z is zero flag
	SET b, s	s_b ← 1	s ≡ r, (HL) (IX+e), (IY+e)
	RES b, s	s_b ← 0	
INPUT AND OUTPUT	IN A, (n)	A ← (n)	
	IN r, (C)	r ← (C)	Set flags
	INI	(HL) ← (C), HL ← HL + 1 B ← B - 1	
	INIR	(HL) ← (C), HL ← HL + 1 B ← B - 1 Repeat until B = 0	
	IND	(HL) ← (C), HL ← HL - 1 B ← B - 1	
	INDR	(HL) ← (C), HL ← HL - 1 B ← B - 1 Repeat until B = 0	
	OUT(n), A	(n) ← A	
	OUT(C), r	(C) ← r	
	OUTI	(C) ← (HL), HL ← HL + 1 B ← B - 1	
	OTIR	(C) ← (HL), HL ← HL + 1 B ← B - 1 Repeat until B = 0	
	OUTD	(C) ← (HL), HL ← HL - 1 B ← B - 1	
	OTDR	(C) ← (HL), HL ← HL - 1 B ← B - 1 Repeat until B = 0	
JUMPS	JP nn	PC ← nn	
	JP cc, nn	If condition cc is true PC ← nn, else continue	$cc \left\{\begin{matrix}NZ & PO\\ Z & PE\\ NC & P\\ C & M\end{matrix}\right.$
	JR e	PC ← PC + e	
	JR kk, e	If condition kk is true PC ← PC + e, else continue	$kk \left\{\begin{matrix}NZ & NC\\ Z & C\end{matrix}\right.$
	JP (ss)	PC ← ss	ss = HL, IX, IY
	DJNZ e	B ← B - 1, if B = 0 continue, else PC ← PC + e	
CALLS	CALL nn	(SP-1) ← PC_H (SP-2) ← PC_L, PC ← nn	
	CALL cc, nn	If condition cc is false continue, else same as CALL nn	$cc \left\{\begin{matrix}NZ & PO\\ Z & PE\\ NC & P\\ C & M\end{matrix}\right.$
RESTARTS	RST L	(SP-1) ← PC_H (SP-2) ← PC_L, PC_H ← 0 PC_L ← L	
RETURNS	RET	PC_L ← (SP), PC_H ← (SP+1)	
	RET cc	If condition cc is false continue, else same as RET	$cc \left\{\begin{matrix}NZ & PO\\ Z & PE\\ NC & P\\ C & M\end{matrix}\right.$
	RETI	Return from interrupt, same as RET	
	RETN	Return from non- maskable interrupt	

Z80–CPU A.C. Characteristics

T_A= 0°C to 70°C, V_{cc}= +5V ± 5%, Unless Otherwise Noted.

Signal	Symbol	Parameter	Min.	Max.	Unit	Test Condition
Φ	t_c	Clock Period	4	[12]	μsec	
	t_w (Φ H)	Clock Pulse Width, Clock High	180	[E]	nsec	
	t_w (Φ L)	Clock Pulse Width, Clock Low	180	2000	nsec	
	$t_{r,}$ f	Clock Rise and Fall Time		30	nsec	
A0-15	$t_{D\ (AD)}$	Address Output Delay		145	nsec	C_L= 50 pF
	$t_{F\ (AD)}$	Delay to Float		110	nsec	
	t_{acm}	Address Stable Prior to $\overline{\text{MREQ}}$ (Memory Cycle)	[1]		nsec	
	t_{act}	Address Stable Prior to $\overline{\text{IORQ}}$, $\overline{\text{RD}}$ or $\overline{\text{WR}}$ (I/O Cycle)	[2]		nsec	
	t_{ca}	Address Stable from $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{IORQ}}$ or $\overline{\text{MREQ}}$	[3]		nsec	
	t_{caf}	Address Stable From $\overline{\text{RD}}$ or $\overline{\text{WR}}$ During Float	[4]		nsec	
D0-7	$t_{D\ (D)}$	Data Output Delay		230	nsec	C_L= 50 pF
	$t_{F\ (D)}$	Delay to Float During Write Cycle		90	nsec	
	$t_{S\Phi\ (D)}$	Data Setup Time to Rising Edge of Clock During Ml Cycle	50		nsec	
	$t_{S\Phi\ (D)}$	Data Setup Time to Falling Edge of Clock During M2 to M5	60		nsec	
	t_{dcm}	Data Stable Prior to $\overline{\text{WR}}$ (Memory Cycle	[5]		nsec	
	t_{dci}	Data Stable Prior to $\overline{\text{WR}}$ (I/O Cycle)	[6]		nsec	
	t_{cdf}	Data Stable From $\overline{\text{WR}}$	[7]			
	t_H	Any Hold Time for Setup Time	0		nsec	
$\overline{\text{MREQ}}$	$t_{DL\overline{\Phi}\ (MR)}$	$\overline{\text{MREQ}}$ Delay From Falling Edge of Clock, $\overline{\text{MREQ}}$ Low		100	nsec	C_L = 50pF
	$t_{DH\Phi\ (MR)}$	$\overline{\text{MREQ}}$ Delay From Rising Edge of Clock, $\overline{\text{MREQ}}$ High		100	nsec	
	$t_{DH\overline{\Phi}\ (MR)}$	$\overline{\text{MREQ}}$ Delay From Falling Edge of Clock, $\overline{\text{MRE'}}$) High		100	nsec	
	$t_{w\ (\overline{MRL})}$	Pulse Width, $\overline{\text{MREQ}}$ Low	[8]		nsec	
	$t_{w\ (\overline{MRH})}$	Pulse Width, $\overline{\text{MREQ}}$ High	[9]		nsec	
$\overline{\text{IORQ}}$	$t_{DL\Phi\ (IR)}$	$\overline{\text{IORQ}}$ Delay From Rising Edge of Clock, $\overline{\text{IORQ}}$ Low		90	nsec	C_L = 50pF
	$t_{DL\overline{\Phi}\ (IR)}$	$\overline{\text{IORQ}}$ Delay From Falling Edge of Clock, $\overline{\text{IORQ}}$ Low		110	nsec	
	$t_{DH\Phi\ (IR)}$	$\overline{\text{IORQ}}$ Delay From Rising Edge of Clock, $\overline{\text{IORQ}}$ High		100	nsec	
	$t_{DH\overline{\Phi}\ (IR)}$	$\overline{\text{IORQ}}$ Delay From Falling Edge of Clock, $\overline{\text{IORQ}}$ High		110	nsec	
$\overline{\text{RD}}$	$t_{DL\Phi\ (RD)}$	$\overline{\text{RD}}$ Delay From Rising Edge of Clock, $\overline{\text{RD}}$ Low		100	nsec	C_L = 50pF
	$t_{DL\overline{\Phi}\ (RD)}$	$\overline{\text{RD}}$ Delay From Falling Edge of Clock, $\overline{\text{RD}}$ Low		130	nsec	
	$t_{DH\Phi\ (RD)}$	$\overline{\text{RD}}$ Delay From Rising Edge of Clock, $\overline{\text{RD}}$ High		100	nsec	
	$t_{DH\overline{\Phi}\ (RD)}$	$\overline{\text{RD}}$ Delay From Falling Edge of Clock, $\overline{\text{RD}}$ High		110	nsec	
$\overline{\text{WR}}$	$t_{DL\Phi\ (WR)}$	$\overline{\text{WR}}$ Delay From Rising Edge of Clock, $\overline{\text{WR}}$ Low		80	nsec	C_L = 50pF
	$t_{DL\overline{\Phi}\ (WR)}$	$\overline{\text{WR}}$ Delay From Falling Edge of Clock, $\overline{\text{WR}}$ Low		90	nsec	
	$t_{DH\overline{\Phi}\ (WR)}$	$\overline{\text{WR}}$ Delay From Falling Edge of Clock, $\overline{\text{WR}}$ High		100	nsec	
	$t_{w\ (\overline{WR}L)}$	Pulse Width, $\overline{\text{WR}}$ Low	[10]		nsec	
$\overline{\text{MI}}$	$t_{DL\ (MI)}$	$\overline{\text{MI}}$ Delay From Rising Edge of Clock, $\overline{\text{MI}}$ Low		130	nsec	C_L = 50pF
	$t_{DH\ (MI)}$	$\overline{\text{MI}}$ Delay From Rising Edge of Clock, $\overline{\text{MI}}$ High		130	nsec	
$\overline{\text{RFSH}}$	$t_{DL\ (RF)}$	$\overline{\text{RFSH}}$ Delay From Rising Edge of Clock, $\overline{\text{RFSH}}$ Low		180	nsec	C_L = 50pF
	$t_{DH\ (RF)}$	$\overline{\text{RFSH}}$ Delay From Rising Edge of Clock, $\overline{\text{RFSH}}$ High		150	nsec	
$\overline{\text{WAIT}}$	$t_{s\ (WT)}$	$\overline{\text{WAIT}}$ Setup Time to Falling Edge of Clock	70		nsec	
$\overline{\text{HALT}}$	$t_{D\ (HT)}$	$\overline{\text{HALT}}$ Delay Time From Falling Edge of Clock		300	nsec	C_L = 50pF
$\overline{\text{INT}}$	$t_{s\ (IT)}$	$\overline{\text{INT}}$ Setup Time to Rising Edge of Clock	80		nsec	
$\overline{\text{NMI}}$	$t_{w\ (\overline{N}ML)}$	Pulse Width, $\overline{\text{NMI}}$ Low	80		nsec	
$\overline{\text{BUSRQ}}$	$t_{s\ (BQ)}$	$\overline{\text{BUSRQ}}$ Setup Time to Rising Edge of Clock	80		nsec	
$\overline{\text{BUSAK}}$	$t_{DL\ (BA)}$	$\overline{\text{BUSAK}}$ Delay From Rising Edge of Clock, $\overline{\text{BUSAK}}$ Low		120	nsec	C_L = 50pF
	$t_{DH\ (BA)}$	$\overline{\text{BUSAK}}$ Delay From Falling Edge of Clock, $\overline{\text{BUSAK}}$ High		110	nsec	
$\overline{\text{RESET}}$	$t_{s\ (RS)}$	$\overline{\text{RESET}}$ Setup Time to Rising Edge of Clock	90		nsec	
	$t_{F\ (C)}$	Delay to Float ($\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, $\overline{\text{RD}}$ and $\overline{\text{WR}}$)		100	nsec	
	t_{mr}	$\overline{\text{MI}}$ Stable Prior to $\overline{\text{IORQ}}$ (Interrupt Ack.)	[11]		nsec	

[12] $t_c = t_{w(\Phi H)} + t_{w(\Phi L)} + t_r + t_f$

[1] $t_{acm} = t_{w(\Phi H)} + t_f - 75$ 75

[2] $t_{aci} = t_c - 80$

[3] $t_{ca} = t_{w(\Phi L)} + t_r - 40$

[4] $t_{caf} = t_{w(\Phi L)} + t_r - 60$

[5] $t_{dcm} = t_c - 210$

[6] $t_{dci} = t_{w(\Phi L)} + t_r - 210$

[7] $t_{cdf} = t_{w(\Phi L)} + t_r - 80$

[8] $t_{w(MRL)} = t_c - 40$

[9] $t_{w(MRH)} = t_{w(\Phi H)} + t_f -- 30$

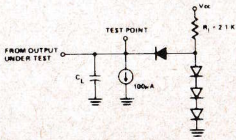
[10] $t_{w(\overline{WR}L)} = t_c - 40$

[11] $t_{mr} = 2t_c + t_{w(\Phi H)} + t_f - 80$

NOTES

A. Data should be enabled onto the CPU data bus when $\overline{\text{RD}}$ is active. During interrupt acknowledge data should be enabled when $\overline{\text{MI}}$ and $\overline{\text{IORQ}}$ are both active.
B. All control signals are internally synchronized, so they may be totally asynchronous with respect to the clock.
C. The $\overline{\text{RESET}}$ signal must be active for a minimum of 3 clock cycles.
D. Output Delay vs. Loaded Capacitance
   TA = 70°C   Vcc = +5V ±5%
   Add 10nsec delay for each 50pf increase in load up to a maximum of 200pf for the data bus & 100pf for address & control lines
E. Although static by design, testing guarantees $t_{w(\Phi H)}$ of 200 μsec maximum

Load circuit for Output



Z80A–CPU A.C. Characteristics

T_A= 0°C to 70°C, V_{cc}= +5V ± 5%, Unless Otherwise Noted.

Signal	Symbol	Parameter	Min	Max	Unit	Test Condition
Φ	t_c	Clock Period	.25	[12]	μsec	
	t_w (ΦH)	Clock Pulse Width, Clock High	110	[E]	nsec	
	t_w (ΦL)	Clock Pulse Width, Clock Low	110	2000	nsec	
	$t_{r,f}$	Clock Rise and Fall Time		30	nsec	
A_{0-15}	$t_{D\ (AD)}$	Address Output Delay		110	nsec	C_L = 50pF
	$t_{F\ (AD)}$	Delay to Float		90	nsec	
	t_{acm}	Address Stable Prior to $\overline{\text{MREQ}}$ (Memory Cycle)	[1]		nsec	
	t_{aci}	Address Stable Prior to $\overline{\text{IORQ}}$, $\overline{\text{RD}}$ or $\overline{\text{WR}}$ (I/O Cycle)	[2]		nsec	
	t_{ca}	Address Stable from $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{IORQ}}$ or $\overline{\text{MREQ}}$	[3]		nsec	
	t_{caf}	Address Stable From $\overline{\text{RD}}$ or $\overline{\text{WR}}$ During Float	[4]		nsec	
D_{0-7}	$t_{D\ (D)}$	Data Output Delay		150	nsec	C_L = 50pF
	$t_{F\ (D)}$	Delay to Float During Write Cycle		90	nsec	
	$t_{S\Phi\ (D)}$	Data Setup Time to Rising Edge of Clock During M1 Cycle	35		nsec	
	$t_{S\overline{\Phi}\ (D)}$	Data Setup Time to Falling Edge of Clock During M2 to M5	50		nsec	
	t_{dcm}	Data Stable Prior to $\overline{\text{WR}}$ (Memory Cycle)	[5]		nsec	
	t_{dci}	Data Stable Prior to $\overline{\text{WR}}$ (I/O Cycle)	[6]		nsec	
	t_{cdf}	Data Stable From $\overline{\text{WR}}$	[7]			
	t_H	Any Hold Time for Setup Time		0	nsec	
$\overline{\text{MREQ}}$	$t_{DL\overline{\Phi}\ (MR)}$	$\overline{\text{MREQ}}$ Delay From Falling Edge of Clock, $\overline{\text{MREQ}}$ Low		85	nsec	C_L = 50pF
	$t_{DH\Phi\ (MR)}$	$\overline{\text{MREQ}}$ Delay From Rising Edge of Clock, $\overline{\text{MREQ}}$ High		85	nsec	
	$t_{DH\overline{\Phi}\ (MR)}$	$\overline{\text{MREQ}}$ Delay From Falling Edge of Clock, MREQ High		85	nsec	
	$t_{w\ (\overline{MRL})}$	Pulse Width, $\overline{\text{MREQ}}$ Low	[8]		nsec	
	$t_{w\ (\overline{MRH})}$	Pulse Width, $\overline{\text{MREQ}}$ High	[9]		nsec	
$\overline{\text{IORQ}}$	$t_{DL\Phi\ (IR)}$	$\overline{\text{IORQ}}$ Delay From Rising Edge of Clock, $\overline{\text{IORQ}}$ Low		75	nsec	C_L = 50pF
	$t_{DL\overline{\Phi}\ (IR)}$	$\overline{\text{IORQ}}$ Delay From Falling Edge of Clock, $\overline{\text{IORQ}}$ Low		85	nsec	
	$t_{DH\Phi\ (IR)}$	$\overline{\text{IORQ}}$ Delay From Rising Edge of Clock, $\overline{\text{IORQ}}$ High		85	nsec	
	$t_{DH\overline{\Phi}\ (IR)}$	$\overline{\text{IORQ}}$ Delay From Falling Edge of Clock, $\overline{\text{IORQ}}$ High		85	nsec	
$\overline{\text{RD}}$	$t_{DL\Phi\ (RD)}$	$\overline{\text{RD}}$ Delay From Rising Edge of Clock, $\overline{\text{RD}}$ Low		85	nsec	C_L = 50pF
	$t_{DL\overline{\Phi}\ (RD)}$	$\overline{\text{RD}}$ Delay From Falling Edge of Clock, $\overline{\text{RD}}$ Low		95	nsec	
	$t_{DH\Phi\ (RD)}$	$\overline{\text{RD}}$ Delay From Rising Edge of Clock, $\overline{\text{RD}}$ High		85	nsec	
	$t_{DH\overline{\Phi}\ (RD)}$	$\overline{\text{RD}}$ Delay From Falling Edge of Clock, $\overline{\text{RD}}$ High		85	nsec	
$\overline{\text{WR}}$	$t_{DL\Phi\ (WR)}$	$\overline{\text{WR}}$ Delay From Rising Edge of Clock, $\overline{\text{WR}}$ Low		65	nsec	C_L = 50pF
	$t_{DL\overline{\Phi}\ (WR)}$	$\overline{\text{WR}}$ Delay From Falling Edge of Clock, $\overline{\text{WR}}$ Low		80	nsec	
	$t_{DH\overline{\Phi}\ (WR)}$	$\overline{\text{WR}}$ Delay From Falling Edge of Clock, $\overline{\text{WR}}$ High		80	nsec	
	$t_{w\ (\overline{WR}L)}$	Pulse Width, $\overline{\text{WR}}$ Low	[10]		nsec	
$\overline{\text{MI}}$	$t_{DL\ (MI)}$	$\overline{\text{MI}}$ Delay From Rising Edge of Clock, $\overline{\text{MI}}$ Low		100	nsec	C_L = 50pF
	$t_{DH\ (MI)}$	$\overline{\text{MI}}$ Delay From Rising Edge of Clock, $\overline{\text{MI}}$ High		100	nsec	
$\overline{\text{RFSH}}$	$t_{DL\ (RF)}$	$\overline{\text{RFSH}}$ Delay From Rising Edge of Clock, $\overline{\text{RFSH}}$ Low		130	nsec	C_L = 50pF
	$t_{DH\ (RF)}$	$\overline{\text{RFSH}}$ Delay From Rising Edge of Clock, $\overline{\text{RFSH}}$ High		120	nsec	
$\overline{\text{WAIT}}$	$t_{s\ (WT)}$	$\overline{\text{WAIT}}$ Setup Time to Falling Edge of Clock	70		nsec	
$\overline{\text{HALT}}$	$t_{D\ (HT)}$	$\overline{\text{HALT}}$ Delay Time From Falling Edge of Clock		300	nsec	C_L = 50pF
$\overline{\text{INT}}$	$t_{s\ (IT)}$	$\overline{\text{INT}}$ Setup Time to Rising Edge of Clock	80		nsec	
$\overline{\text{NMI}}$	$t_{w\ (\overline{N}ML)}$	Pulse Width, $\overline{\text{NMI}}$ Low	80		nsec	
$\overline{\text{BUSRQ}}$	$t_{s\ (BQ)}$	$\overline{\text{BUSRQ}}$ Setup Time to Rising Edge of Clock	50		nsec	
$\overline{\text{BUSAK}}$	$t_{DL\ (BA)}$	$\overline{\text{BUSAK}}$ Delay From Rising Edge of Clock, $\overline{\text{BUSAK}}$ Low		100	nsec	C_L = 50pF
	$t_{DH\ (BA)}$	$\overline{\text{BUSAK}}$ Delay From Falling Edge of Clock, $\overline{\text{BUSAK}}$ High		100	nsec	
$\overline{\text{RESET}}$	$t_{s\ (RS)}$	$\overline{\text{RESET}}$ Setup Time to Rising Edge of Clock	60		nsec	
	$t_{F\ (C)}$	Delay to Float ($\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, $\overline{\text{RD}}$ and $\overline{\text{WR}}$)		80	nsec	
	t_{mr}	MI Stable Prior to IORQ (Interrupt Ack.)	[11]		nsec	

[12] $t_c = t_{w(\Phi H)} + t_{w(\Phi L)} + t_r + t_f$

[1] $t_{acm} = t_{w(\Phi H)} + t_f - 65$

[2] $t_{aci} = t_c - 70$

[3] $t_{ca} = t_{w(\Phi L)} + t_r - 50$

[4] $t_{caf} = t_{w(\Phi L)} + t_r - 45$

[5] $t_{dcm} = t_c - 170$

[6] $t_{dci} = t_{w(\Phi L)} + t_r - 170$

[7] $t_{cdf} = t_{w(\Phi L)} + t_r - 70$

[8] $t_{w\ (\overline{MRL})} = t_c - 30$

[9] $t_{w(\overline{MRH})} = t_{w(\Phi H)} + t_f - 20$

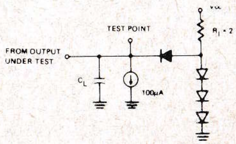
[10] $t_{w(\overline{WR}L)} = t_c - 30$

[11] $t_{mr} = 2t_c + t_{w(\Phi H)} + t_f - 65$

NOTES:

A. Data should be enabled onto the CPU data bus when $\overline{\text{RD}}$ is active. During interrupt acknowledge data should be enabled when $\overline{\text{MI}}$ and $\overline{\text{IORQ}}$ are both active.
B. All control signals are internally synchronized, so they may be totally asynchronous with respect to the clock.
C. The $\overline{\text{RESET}}$ signal must be active for a minimum of 3 clock cycles.
D. Output Delay vs. Loaded Capacitance
   TA = 70°C   Vcc = +5V ±5%
   Add 10nsec delay for each 50pf increase in load up to maximum of 200pf for data bus and 100pf for address & control lines.
E. Although static by design, testing guarantees $t_{w(\Phi H)}$ of 200 μsec maximum

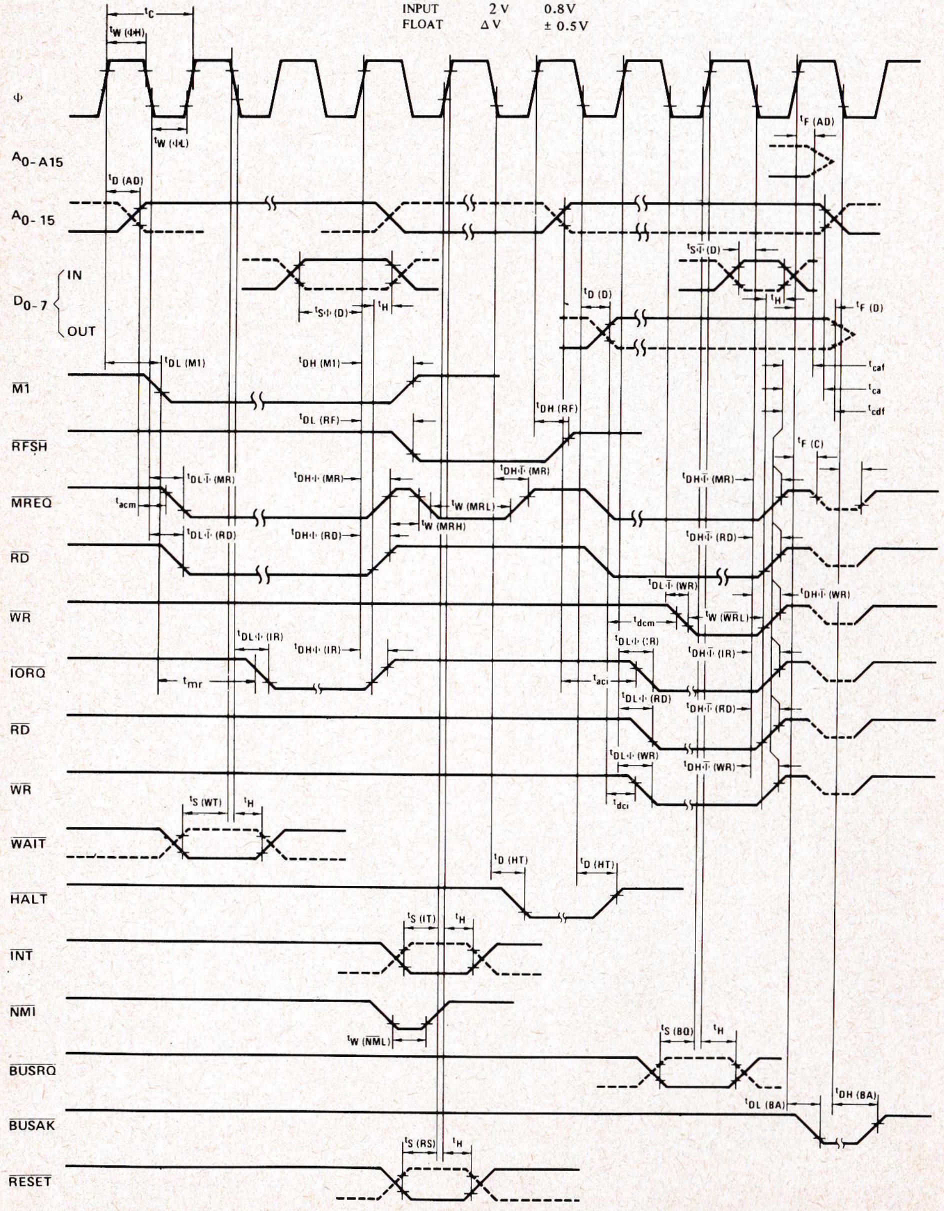
Load circuit for Output



A.C. Timing Diagram

Timing measurements are made at the following voltages, unless otherwise specified:

	"1"	"0"
CLOCK	V_{cc}-0.6V	0.45V
OUTPUT	2 V	0.8V
INPUT	2 V	0.8V
FLOAT	ΔV	± 0.5V



Absolute Maximum Ratings

Temperature Under Bias	Specified operating range
Storage Temperature	-65°C to +150°C
Voltage On Any Pin with Respect to Ground	-0.3V to +7V
Power Dissipation	1.5W

Note: For Z80–CPU all AC and DC characteristics remain the same for the military grade parts except I_{cc}.
I_{cc}= 200 mA

* Comment
Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Z80–CPU D.C. Characteristics

T_A= 0°C to 70°C, V_{cc}= 5V ± 5% unless otherwise specified

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	-0.3		0.45	V	
V_{IHC}	Clock Input High Voltage	V_{cc}-0.6		V_{cc}+0.3	V	
V_{IL}	Input Low Voltage	-0.3		0.8	V	
V_{IH}	Input High Voltage	2		V_{cc}	V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL}= 1.8 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH}= -250 μA
I_{CC}	Power Supply Current			150	mA	
I_{LI}	Input Leakage Current			10	μA	V_{IN}= 0 to V_{cc}
I_{LOH}	Tri-State Output Leakage Current in Float			10	μA	V_{OUT}= 2.4 to V_{cc}
I_{LOL}	Tri-State Output Leakage Current in Float			-10	μA	V_{OUT}= 0.4V
I_{LD}	Data Bus Leakage Current in Input Mode			±10	μA	0 < V_{IN} < V_{cc}

Capacitance

T_A= 25°C, f = 1 MHz,
unmeasured pins returned to ground

Symbol	Parameter	Max.	Unit
$C_Φ$	Clock Capacitance	35	pF
C_{IN}	Input Capacitance	5	pF
C_{OUT}	Output Capacitance	10	pF

Z80A–CPU D.C. Characteristics

T_A= 0°C to 70°C, V_{cc}= 5V ± 5% unless otherwise specified

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
V_{ILC}	Clock Input Low.Voltage	-0.3		0.45	V	
V_{IHC}	Clock Input High Voltage	V_{cc}-0.6		V_{cc}+0.3	V	
V_{IL}	Input Low Voltage	-0,3		0.8	V	
V_{IH}	Input High Voltage	2		V_{cc}	V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL}= 1.8 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH}= -250 μA
I_{CC}	Power Supply Current		90	200	mA	
I_{LI}	Input Leakage Current			10	μA	V_{IN}= 0 to V_{cc}
I_{LOH}	Tri-State Output Leakage Current in Float			10	μA	V_{OUT}= 2.4 to V_{cc}
I_{LOL}	Tri-State Output Leakage Current in Float			-10	μA	V_{OUT}= 0.4V
I_{LD}	Data Bus Leakage Current in Input Mode			±10	μA	0 < V_{IN} < V_{cc}

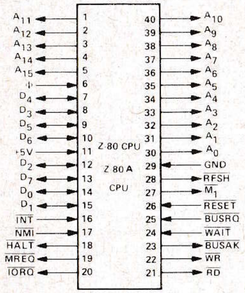
Capacitance

T_A= 25°C, f = 1 MHz;
unmeasured pins returned to ground

Symbol	Parameter	Max.	Unit
$C_Φ$	Clock Capacitance	35	pF
C_{IN}	Input Capacitance	5	pF
C_{OUT}	Output Capacitance	10	pF

Z 80-CPU
Z 80A-CPU

Package Configuration

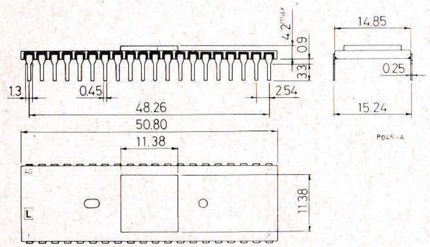


ORDERING NUMBERS:

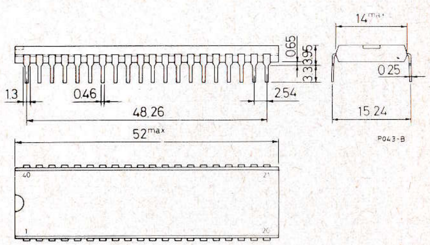
Z80–CPU D1	for dual in–line ceramic slam package
Z80–CPU B1	for dual in–line plastic package
Z80A–CPU D1	for dual in–line ceramic slam package
Z80A–CPU B1	for dual in–line plastic package

MECHANICAL DATA (dimensions in mm)

40–PIN CERAMIC DUAL IN–LINE SLAM PACKAGE



40–PIN PLASTIC DUAL IN–LINE PACKAGE



Product Specification

Z 80-PIO
Z 80A-PIO

The SGS-ATES Z80 product line is a complete set of microcomputer components, development systems and support software. The Z-80 microcomputer component set includes all of the circuits necessary to build high-performance microcomputer systems with virtually no other logic and a minimum number of low cost standard memory elements.

The Z-80 Parallel I/O (PIO) Interface Controller is a programmable, two port device which provides TTL compatible interface between peripheral devices and the Z80-CPU. The Z80-CPU configures the Z80-PIO to interface with standard peripheral devices such as tape punches, printers, keyboards, etc.

Structure

- N-Channel Silicon Gate Depletion Load technology
- 40 Pin DIP
- Single 5 volt supply
- Single phase 5 volt clock
- Two independent 8-bit bidirectional peripheral interface ports with "handshake" data transfer control

Features

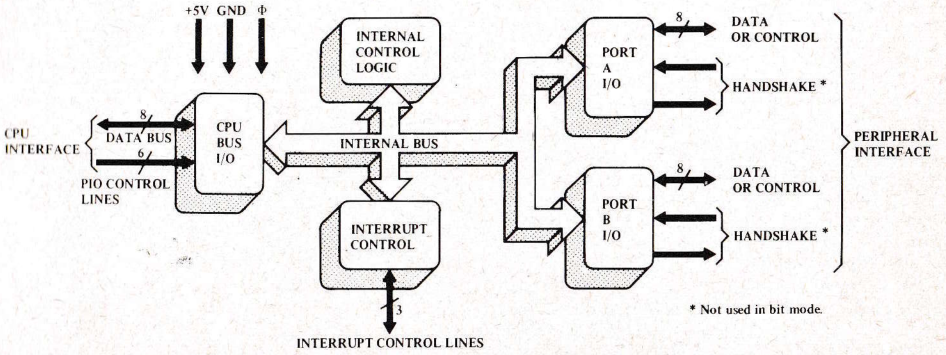
- Interrupt driven "handshake" for fast response
- Any one of the following modes of operation may be selected for either port:
  - Byte output
  - Byte input
  - Byte bidirectional bus (available on Port A only)
  - Bit Mode
- Programmable interrupts on peripheral status conditions.
- Daisy chain priority interrupt logic included to provide for automatic interrupt vectoring without external logic.
- Eight outputs are capable of driving Darlington transistors.
- All inputs and outputs fully TTL compatible.

PIO Architecture

A block diagram of the Z80-PIO is shown in figure 3. The internal structure of the Z80-PIO consists of a Z80-CPU bus interface, internal control logic, Port A I/O logic, Port B I/O logic, and interrupt control logic. A typical application might use Port A as the data transfer channel and Port B for the status and control monitoring.

The Port I/O logic is composed of 6 registers with "handshake" control logic as shown in figure 4. The registers include: an 8-bit input register, an 8-bit output register, a 2-bit mode control register, an 8-bit mask register, an 8-bit input/output select register, and a 2-bit mask control register. The last three registers are used only when the port has been programmed to operate in the bit mode.

Fig. 3 - PIO BLOCK DIAGRAM



* Not used in bit mode.

Register Description

Mode Control Register—2 bits, loaded by CPU to select the operating mode: byte output, byte input, byte bidirectional bus or bit mode.

Data Output Register—8 bits, permits data to be transferred from the CPU to the peripheral.

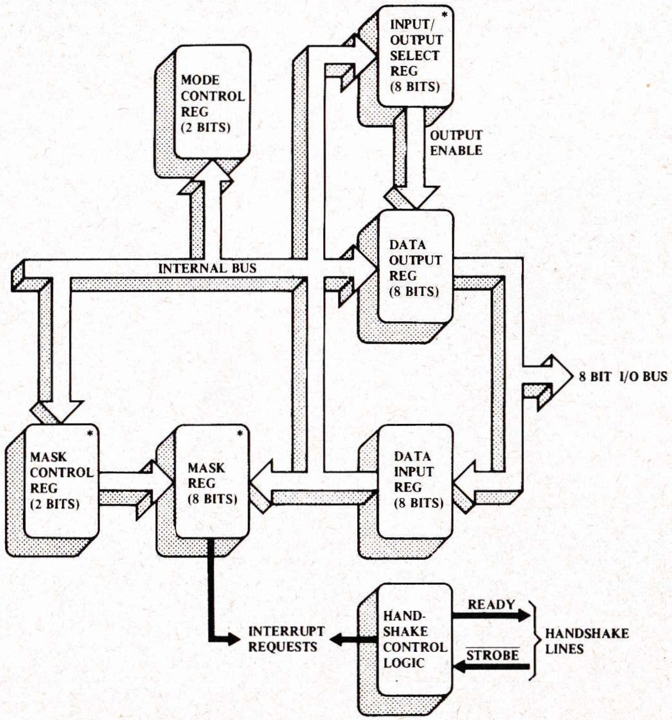
Data Input Register—8 bits, accepts data from the peripheral for transfer to the CPU.

Mask Control Register—2 bits, loaded by the CPU to specify the active state (high or low) of any peripheral device interface pins that are to be monitored and, if an interrupt should be generated when all unmasked pins are active (AND condition), or, when any unmasked pin is active (OR condition).

Mask Register—8 bits, loaded by the CPU to determine which peripheral device interface pins are to be monitored for the specified status condition.

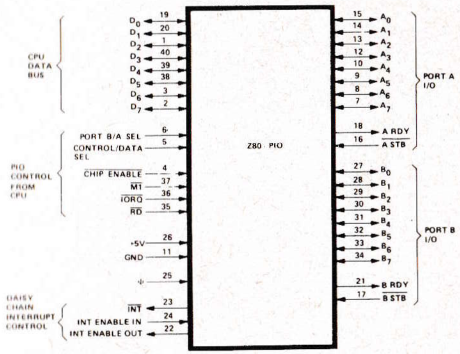
Input/Output Select Register—8 bits, loaded by the CPU to allow any pin to be an output or an input during bit mode operation.

Fig. 4 - A TYPICAL PORT I/O BLOCK DIAGRAM



* Used in the bit mode only to allow generation of an interrupt if the peripheral I/O pins go to the specified state.

Z80-PIO Pin Description

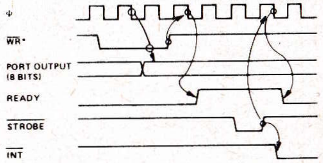


D_7-D_0	Z80-CPU Data Bus (bidirectional, tristate)
B/A Sel	Port B or A Select (input, active high)
C/D Sel	Control or Data Select (input, active high)
$\overline{\text{CE}}$	Chip Enable (input, active low)
Φ	System Clock (input)
$\overline{\text{M1}}$	Machine Cycle One Signal from CPU (input, active low)
$\overline{\text{IORQ}}$	Input/Output Request from Z80-CPU (input, active low)
$\overline{\text{RD}}$	Read Cycle Status from the Z80-CPU (input, active low)
IEI	Interrupt Enable In (input, active high)
IEO	Interrupt Enable Out (output, active high). IEI and IEO form a daisy chain connection for priority interrupt control.
$\overline{\text{INT}}$	Interrupt Request (output, open drain, active low)
A_0-A_7	Port A Bus (bidirectional, tristate)
$\overline{\text{A STB}}$	Port A Strobe Pulse from Peripheral Device (input, active low)
A RDY	Register A Ready (output, active high)
B_0-B_7	Port B Bus (bidirectional, tristate)
$\overline{\text{B STB}}$	Port B Strobe Pulse from Peripheral Device (input, active low)
B RDY	Register B Ready (output, active high)

Timing Waveforms

OUTPUT MODE

An output cycle is always started by the execution of an output instruction by the CPU. The $\overline{\text{WR}}$ pulse from the CPU latches the data from the CPU data bus into the selected port's output register. The write pulse sets the ready flag after a low going edge of Φ, indicating data is available. Ready stays active until the positive edge of the strobe line is received indicating that data was taken by the peripheral. The positive edge of the strobe pulse generates an $\overline{\text{INT}}$ if the interrupt enable flip flop has been set and if this device has the highest priority.

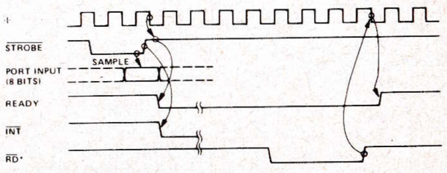


MODE 0 (OUTPUT) TIMING

WR* = $\overline{\text{RD}}$ · CE · $\overline{\text{C/D}}$ · IORQ

INPUT MODE

When $\overline{\text{STROBE}}$ goes low data is loaded into the selected port input register. The next rising edge of strobe activates $\overline{\text{INT}}$ if interrupt enable is set and this is the highest priority requesting device. The following falling edge of Φ resets Ready to an inactive state, indicating that the input register is full and cannot accept any more data until the CPU completes a read. When a read is complete the positive edge of $\overline{\text{RD}}$ will set Ready at the next low going transition of Φ. At this time new data can be loaded into the PIO.



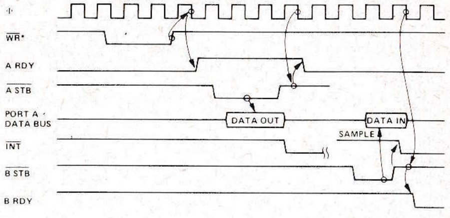
RD* = RD · CE · $\overline{\text{C/D}}$ · IORQ

MODE 1 (INPUT) TIMING

Timing Waveforms (continued)

BIDIRECTIONAL MODE

This is a combination of modes 0 and 1 using all four handshake lines and the 8 Port A I/O lines. Port B must be set to the Bit Mode. The Port A handshake lines are used for output control and the Port B lines are used for input control. Data is allowed out onto the Port A bus only when A STB is low. The rising edge of this strobe can be used to latch the data into the peripheral.

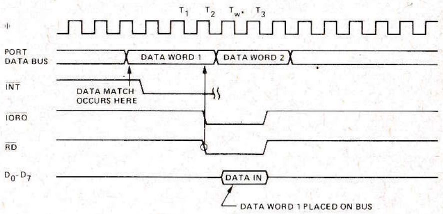


WR* = $\overline{\text{RD}}$ · CE · $\overline{\text{C/D}}$ · IORQ

BIT MODE

The bit mode does not utilize the handshake signals and a normal port write or port read can be executed at any time. When writing, the data will be latched into the output registers with the same timing as the output mode.

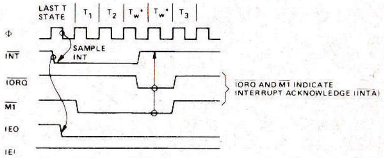
When reading the PIO, the data returned to the CPU will be composed of output register data from those port data lines assigned as outputs and input register data from those port data lines assigned as inputs. The input register will contain data which was present immediately prior to the falling edge of RD. An interrupt will be generated if interrupts from the port are enabled and the data on the port data lines satisfy the logical equation defined by the 8-bit mask and 2-bit mask control registers.



* Timing Diagram Refers to Bit Mode Read.

INTERRUPT ACKNOWLEDGE

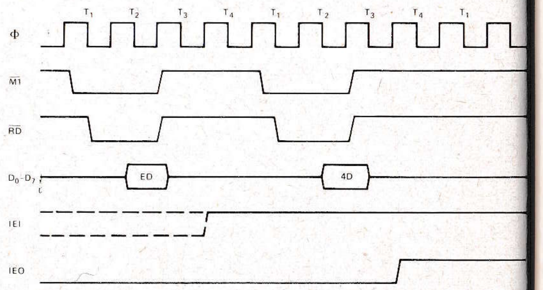
During MI time, peripheral controllers are inhibited from changing their interrupt enable status, permitting the INT Enable signal to ripple through the daisy chain. The peripheral with IEI high and IEO low during INTA will place a preprogrammed 8-bit interrupt vector on the data bus at this time. IEO is held low until a return from interrupt (RETI) instruction is executed by the CPU while IEI is high. The 2-byte RETI instruction is decoded internally by the PIO for this purpose.



RETURN FROM INTERRUPT CYCLE

If a Z80 peripheral device has no interrupt pending and is not under service, then its IEO=IEI. If it has an interrupt under service (i.e., it has already interrupted and received an interrupt acknowledge) then its IEO is always low, inhibiting lower priority chips from interrupting. If it has an interrupt pending which has not yet been acknowledged, IEO will be low unless an "ED" is decoded as the first byte of a two byte opcode. In this case, IEO will go high until the next opcode byte is decoded, whereupon it will again go low. If the second byte of the opcode was a "4D" then the opcode was an RETI instruction.

After an "ED" opcode is decoded, only the peripheral device which has interrupted and is currently under service will have its IEI high and its IEO low. This device is the highest priority device in the daisy chain which has received an interrupt acknowledge. All other peripherals have IEI=IEO. If the next opcode byte decoded is "4D", this peripheral device will reset its "interrupt under service" condition.



PIO Programming

LOAD INTERRUPT VECTOR

The Z80-CPU requires an 8-bit interrupt vector be supplied by the interrupting device. The CPU forms the address for the interrupt service routine of the port using this vector. During an interrupt acknowledge cycle the vector is placed on the Z-80 data bus by the highest priority device requesting service at that time. The desired interrupt vector is loaded into the PIO by writing a control word to the desired port of the PIO with the following format.

D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	V3	V2	V1	0

signifies this control word is an interrupt vector

SELECTING AN OPERATING MODE

When selecting an operating mode, the 2-bit mode control register is set to one of four values. These two bits are the most significant bits of the register, bits 7 and 6; bits 5 and 4 are not used while bits 3 through 0 are all set to 1111 to indicate "set mode."

D7	D6	D5	D4	D3	D2	D1	D0
M1	M0	X	X	1	1	1	1

mode word

signifies mode word to be set

X=unused bit

Mode	M_1	M_0
Output	0	0
Input	0	1
Bidirectional	1	0
Bit	1	1

MODE 0 active indicates that data is to be written from the CPU to the peripheral.

MODE 1 active indicates that data is to be read from the peripheral to the CPU.

MODE 2 allows data to be written to or read from the peripheral device.

MODE 3 is intended for status and control applications. When selected, the next control word must set the I/O Register to indicate which lines are to be input and which lines are to be output.

I/O = 1 sets bit to input.
I/O = 0 sets bit to output.

D7	D6	D5	D4	D3	D2	D1	D0
I/O_7	I/O_6	I/O_5	I/O_4	I/O_3	I/O_2	I/O_1	I/O_0

INTERRUPT CONTROL

Bit 7 = 1	interrupt enable is set—allowing interrupt to be generated.
Bit 7 = 0	indicates the enable flag is reset and interrupts may not be generated.
Bits 6,5,4	are used in the bit mode interrupt operations; otherwise they are disregarded.
Bits 3,2,1,0	signify that this command word is an interrupt control word.

D7	D6	D5	D4	D3	D2	D1	D0
Enable Interrupt	AND/ OR	High/ Low	Mask follows	0	1	1	1

used in Mode 3 only

signifies interrupt control word

If the "mask follows" bit is high (D4 = 1), the next control word written to the port must be the mask.

D7	D6	D5	D4	D3	D2	D1	D0
MB_7	MB_6	MB_5	MB_4	MB_3	MB_2	MB_1	MB_0

Only those port lines whose mask bit is a 0 will be monitored for generating an interrupt.

The interrupt enable flip-flop of a port may be set or reset without modifying the rest of the interrupt control word by the following command.

D7	D6	D5	D4	D3	D2	D1	D0
Int Enable	X	X	X	0	0	1	1

Z80–PIO A.C. Characteristics

TA = 0° C to 70° C, Vcc = +5 V ± 5%, unless otherwise noted

SIGNAL	SYMBOL	PARAMETER	MIN	MAX	UNIT	COMMENTS
Φ	t_c	Clock Period	400	[1]	nsec	
	$t_{W(\Phi H)}$	Clock Pulse Width, Clock High	170	2000	nsec	
	$t_{W(\Phi L)}$	Clock Pulse Width, Clock Low	170	2000	nsec	
	t_r, t_f	Clock Rise and Fall Times		30	nsec	
	t_h	Any Hold Time for Specified Set-Up Time	0		nsec	
CS, $\overline{CE}$ ETC.	$t_{S\Phi}$ (CS)	Control Signal Set-Up Time to Rising Edge of Φ During Read or Write Cycle	280		nsec	
D_0-D_7	t_{DR} (D)	Data Output Delay from Falling Edge of $\overline{RD}$		430	nsec	[2]
	$t_{S\Phi}$ (D)	Data Set-Up Time to Rising Edge of Φ During Write or $\overline{M1}$ Cycle	50		nsec	C_L = 50 pf
	t_{DI} (D)	Data Output Delay from Falling Edge of $\overline{IORQ}$ During INTA Cycle.		340	nsec	[3]
	t_F (D)	Delay to Floating Bus (Output Buffer Disable Time)		160	nsec	
IEI	t_S (IEI)	IEI Set-Up Time to Falling Edge of $\overline{IORQ}$ During INTA Cycle	140		nsec	
IEO	t_{DH} (IO)	IEO Delay Time from Rising Edge of IEI		210	nsec	[5]
	t_{DL} (IO)	IEO Delay Time from Falling Edge of IEI		190	nsec	[5] C_L = 50 pf
	t_{DM} (IO)	IEO Delay from Falling Edge of $\overline{M1}$ (Interrupt Occurring Just Prior to $\overline{M1}$) See Note A.		300	nsec	[5]
$\overline{IORQ}$	$t_{S\Phi}$ (IR)	$\overline{IORQ}$ Set-Up Time to Rising Edge of Φ During Read or Write Cycle	250		nsec	
$\overline{M1}$	$t_{S\Phi}$ (M1)	$\overline{M1}$ Set-Up Time to Rising Edge of Φ During INTA or $\overline{M1}$ Cycle. See Note B.	210		nsec	
$\overline{RD}$	$t_{S\Phi}$ (RD)	$\overline{RD}$ Set-Up Time to Rising Edge of Φ During Read or $\overline{M1}$ Cycle	240		nsec	
A_0-A_7, B_0-B_7	t_S (PD)	Port Data Set-Up Time to Rising Edge of $\overline{STROBE}$ (Mode 1)	260		nsec	
	t_{DS} (PD)	Port Data Output Delay from Falling Edge of $\overline{STROBE}$ (Mode 2)		230	nsec	[5]
	t_F (PD)	Delay to Floating Port Data Bus from Rising Edge of $\overline{STROBE}$ (Mode 2)		200	nsec	C_L = 50 pf
	t_{DI} (PD)	Port Data Stable from Rising Edge of $\overline{IORQ}$ During WR Cycle (Mode 0)		200	nsec	[5]
$\overline{ASTB}$, $\overline{BSTB}$	t_W (ST)	Pulse Width, $\overline{STROBE}$	150 [4]		nsec nsec	
$\overline{INT}$	t_D (IT)	$\overline{INT}$ Delay Time from Rising Edge of $\overline{STROBE}$		490	nsec	
	t_D (IT3)	$\overline{INT}$ Delay Time from Data Match During Mode 3 Operation		420	nsec	
ARDY, BRDY	t_{DH} (RY)	Ready Response Time from Rising Edge of $\overline{IORQ}$		t_c+ 460	nsec	[5] C_L = 50 pf
	t_{DL} (RY)	Ready Response Time from Rising Edge of $\overline{STROBE}$		t_c+ 400	nsec	[5]

A. 2.5 $t_c > (N\text{-}2)\, t_{DL(IO)} + t_{DM(IO)} + t_{S(IEI)}$ + TTL Buffer Delay, if any

B. $\overline{M1}$ must be active for a minimum of 2 clock periods to reset the PIO.

[1] $t_c = t_{W(\Phi H)} + t_{W(\Phi L)} + t_r + t_f$

[2] Increase $t_{DR(D)}$ by 10 nsec for each 50 pf increase in loading up to 200 pf max.

[3] Increase $t_{DI(D)}$ by 10 nsec for each 50 pf increase in loading up to 200 pf max.

[4] For Mode 2: $t_{W(ST)} > t_{S(PD)}$

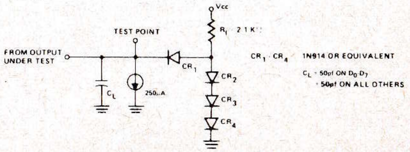
[5] Increase these values by 2 nsec for each 10 pf increase in loading up to 100 pf max.

Capacitance

TA = 25° C, f = 1 MHz

Symbol	Parameter	Max.	Unit	Test Condition
C_Φ	Clock Capacitance	10	pF	Unmeasured Pins
C_{IN}	Input Capacitance	5	pF	Returned to Ground
C_{OUT}	Output Capacitance	10	pF	

Output Load circuit



Z80A–PIO A.C. Characteristics

TA = 0° C to 70° C, Vcc = +5 V ± 5%, unless otherwise noted

SIGNAL	SYMBOL	PARAMETER	MIN	MAX	UNIT	COMMENTS
Φ	t_c	Clock Period	250	[1]	nsec	
	$t_{W(\Phi H)}$	Clock Pulse Width, Clock High	105	2000	nsec	
	$t_{W(\Phi L)}$	Clock Pulse Width, Clock Low	105	2000	nsec	
	t_r, t_f	Clock Rise and Fall Times		30	nsec	
	t_h	Any Hold Time for Specified Set-Up Time	0		nsec	
CS, $\overline{CE}$ ETC.	$t_{S\Phi}$ (CS)	Control Signal Set-Up Time to Rising Edge of Φ During Read or Write Cycle	145		nsec	
D_0-D_7	t_{DR} (D)	Data Output Delay From Falling Edge of $\overline{RD}$		380	nsec	[2]
	$t_{S\Phi}$ (D)	Data Set-Up Time to Rising Edge of Φ During Write or $\overline{M1}$ Cycle	50		nsec	C_L = 50 pf
	t_{DI} (D)	Data Output Delay from Falling Edge of $\overline{IORQ}$ During INTA Cycle		250	nsec	[3]
	t_F (D)	Delay to Floating Bus (Output Buffer Disable Time)		110	nsec	
IEI	t_S (IEI)	IEI Set-Up Time to Falling edge of $\overline{IORQ}$ During INTA Cycle	140		nsec	
IEO	t_{DH} (IO)	IEO Delay Time from Rising Edge of IEI		160	nsec	[5]
	t_{DL} (IO)	IEO Delay Time from Falling Edge of IEI		130	nsec	[5] C_L = 50 pf
	t_{DM} (IO)	IEO Delay from Falling Edge of $\overline{M1}$ (Interrupt Occurring Just Prior to $\overline{M1}$) See Note A.		190	nsec	[5]
$\overline{IORQ}$	$t_{S\Phi}$ (IR)	$\overline{IORQ}$ Set-Up Time to Rising Edge of Φ During Read or Write Cycle.	115		nsec	
$\overline{M1}$	$t_{S\Phi}$ (M1)	$\overline{M1}$ Set-Up Time to Rising Edge of Φ During INTA or $\overline{M1}$ Cycle See Note B	90		nsec	
$\overline{RD}$	$t_{S\Phi}$ (RD)	$\overline{RD}$ Set-Up Time to Rising Edge of Φ During Read or $\overline{M1}$ Cycle	115		nsec	
A_0-A_7, B_0-B_7	t_S (PD)	Port Data Set-Up Time to Rising Edge of $\overline{STROBE}$ (Mode 1)	230		nsec	
	t_{DS} (PD)	Port Data Ourput Delay from Falling Edge of $\overline{STROBE}$ (Mode 2)		210	nsec	[5]
	t_F (PD)	Delay to Floating Port Data Bus from Rising Edge of $\overline{STROBE}$ (Mode 2)		180	nsec	C_L = 50 pf
	t_{DI} (PD)	Port Data Stable from Rising Edge of $\overline{IORQ}$ During WR Cycle (Mode 0)		180	nsec	[5]
$\overline{ASTB}$, $\overline{BSTB}$	t_W (ST)	Pulse Width, $\overline{STROBE}$	150 [4]		nsec nsec	
$\overline{INT}$	t_D (IT)	$\overline{INT}$ Delay time from Rising Edge of $\overline{STROBE}$		440	nsec	
	t_D (IT3)	$\overline{INT}$ Delay Time from Data Match During Mode 3 Operation		380	nsec	
ARDY, BRDY	t_{DH} (RY)	Ready Response Time from Rising Edge of $\overline{IORQ}$		t_c+ 410	nsec	[5] C_L 50 pf
	t_{DL} (RY)	Ready Response Time from Rising Edge of $\overline{STROBE}$		t_c+ 360	nsec	[5]

A. 2.5 $t_c > (N\text{-}2)\, t_{DL(IO)} + t_{DM(IO)} + t_{S(IEI)}$ + TTL Buffer Delay, if any

B. $\overline{M1}$ must be active for a minimum of 2 clock periods to reset the PIO.

[1] $t_c = t_{W(\Phi H)} + t_{W(\Phi L)} + t_r + t_f$

[2] Increase $t_{DR(D)}$ by 10 nsec for each 50 pf increase in loading up to 200 pf max.

[3] Increase $t_{DI(D)}$ by 10 nsec for each 50 pf increase in loading up to 200 pf max.

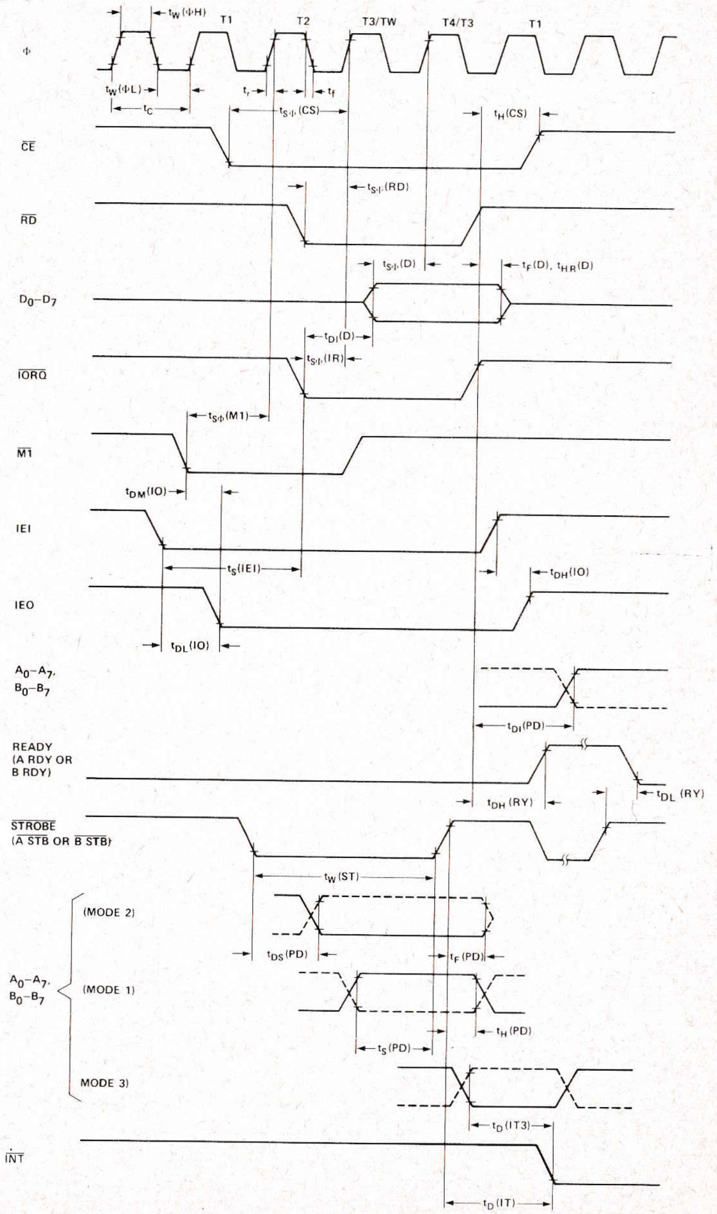
[4] For Mode 2: $t_{W(ST)} > t_{S(PD)}$

[5] Increase these values by 2 nsec for each 10 pf increase in loading up to 100 pf max.

A.C. Timing Diagram

Timing measurements are made at the following voltages, unless otherwise specified:

	"1"	"0"
CLOCK	4.2V	0.8V
OUTPUT	2V	0.8V
INPUT	2V	0.8V
FLOAT	ΔV	= +0.5V



Absolute Maximum Ratings

Temperature Under Bias	Specified operating range
Storage Temperature	-65°C to +150°C
Voltage On Any Pin with Respect to Ground	-0.3V to +7V
Power Dissipation	0.6W

Note: All AC and DC characteristics remain the same for the military grade parts except I_{CC}.
I_{CC}= 130 mA

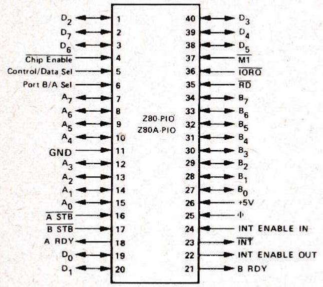
* Comment
Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Z80–PIO and Z80A–PIO D.C. Characteristics

T_A= 0°C to 70°C, V_{CC}= +5V ± 5%, unless otherwise noted

Symbol	Parameter	Min.	Max.	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	-0.3	0.45	V	
V_{IHC}	Clock Input High Voltage	V_{CC}-0.6	V_{CC}+0.3	V	
V_{IL}	Input Low Voltage	-0.3	0.8	V	
V_{IH}	Input High Voltage	2	Vcc	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2.0 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} - 250 μA
I_{CC}	Power Supply Current		70	mA	
I_{LI}	Input Leakage Current		10	μA	V_{IN} = 0 to Vcc
I_{LOH}	Tri-State Output Leakage Current in Float		10	μA	V_{OUT} = 2.4 to Vcc
I_{LOL}	Tri-State Output Leakage Current in Float		-10	μA	V_{OUT} = 0.4 V
I_{LD}	Data Bus Leakage Current in Input Mode		±10	μA	$0 \leq V_{IN} \leq$ Vcc
I_{OHD}	Darlington Drive Current	-1.5		mA	V_{OH} = 1.5 V Port B Only

Package Configuration

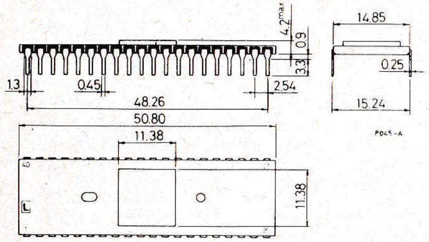


ORDERING NUMBERS:

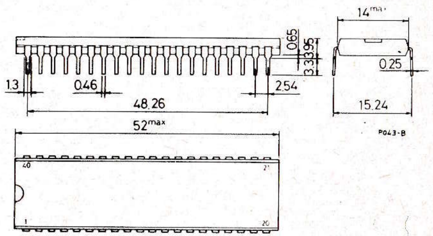
Z80-PIO	**D1**	for dual in-line ceramic slam package
Z80-PIO	**B1**	for dual in-line plastic package
Z80A-PIO	**D1**	for dual in-line ceramic slam package
Z80A-PIO	**B1**	for dual in-line plastic package

MECHANICAL DATA (dimensions in mm)

40-PIN CERAMIC DUAL IN-LINE SLAM PACKAGE



40-PIN PLASTIC DUAL IN-LINE PACKAGE



Product Specification

The SGS-ATES Z80 product line is a complete set of microcomputer components, development systems and support software. The Z80 microcomputer component set includes all of the circuits necessary to build high-performance microcomputer systems with virtually no other logic and a minimum number of low cost standard memory elements.

The Z80-Counter Timer Circuit (CTC) is a programmable, four channel device that provides counting and timing functions for the Z80-CPU. The Z80-CPU configures the Z80-CTC's four independent channels to operate under various modes and conditions as required.

Structure

- N-Channel Silicon Gate Depletion Load Technology
- 28 Pin DIP
- Single 5 volt supply
- Single phase 5 volt clock
- Four independent programmable 8-bit counter/16-bit timer channels

Features

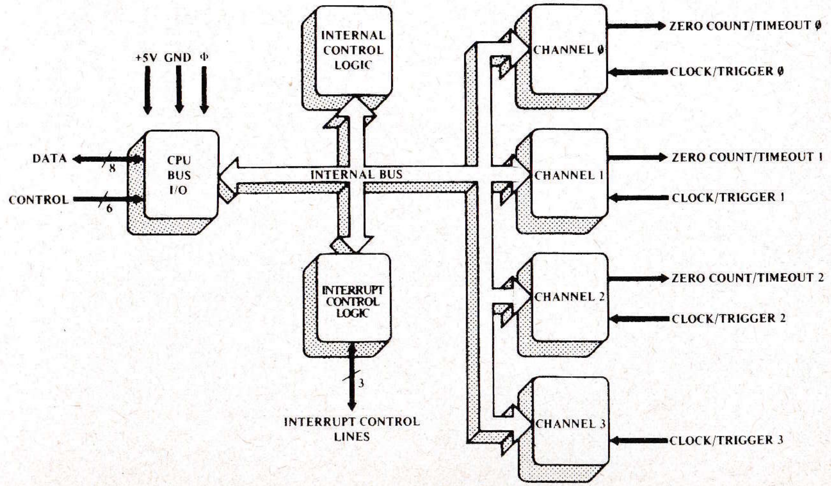
- Each channel may be selected to operate in either a counter mode or timer mode.
- Programmable interrupts on counter or timer states.
- A time constant register automatically reloads the down counter at zero and the cycle is repeated.
- Readable down counter indicates number of counts-to-go until zero.
- Selectable 16 or 256 clock prescaler for each timer channel.
- Selectable positive or negative trigger may initiate timer operation.
- Three channels have zero count/timeout outputs capable of driving Darlington transistors.
- Daisy chain priority interrupt logic included to provide for automatic interrupt vectoring without external logic.
- All inputs and outputs fully TTL compatible.

CTC Architecture

A block diagram of the Z80-CTC is shown in figure 5. The internal structure of the Z80-CTC consists of a Z80-CPU bus interface, internal control logic, four counter channels, and interrupt control logic. Each channel has an interrupt vector for automatic interrupt vectoring, and interrupt priority is determined by channel number with channel Ø having the highest priority.

The channel logic is composed of 2 registers, 2 counters and control logic as shown in figure 6. The registers include an 8-bit time constant register and an 8-bit channel control register. The counters include an 8-bit readable down counter and an 8-bit prescaler. The prescaler may be programmed to divide the system clock by either 16 or 256.

Fig. 5 - CTC BLOCK DIAGRAM



Channel Counter and Register Description

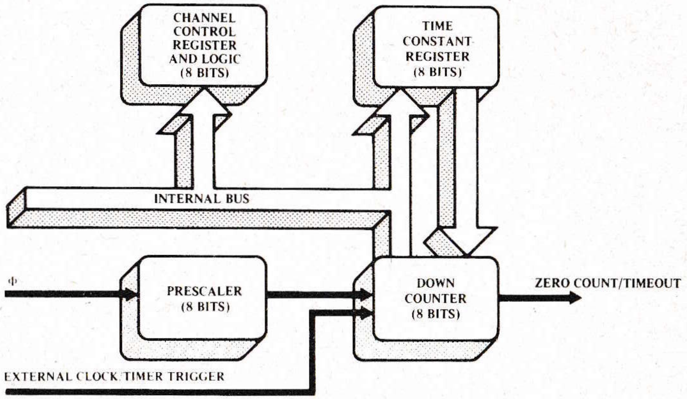
Time Constant Register – 8 bits, loaded by the CPU to initialize and re-load Down Counter at a count of zero.

Channel Control Register – 8 bits, loaded by the CPU to select the mode and conditions of channel operation.

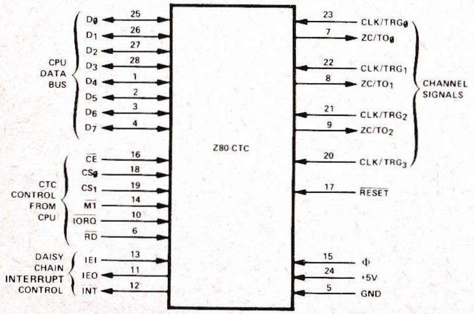
Down Counter – 8 bits, loaded by the Time Constant Register under program control and automatically at a count of zero. At any time, the CPU can read the number of counts-to-go until a zero count. This counter is decremented by the prescaler in timer mode and CLK/TRIG in counter mode.

Prescaler – 8 bit counter, divides system clock by 16 or 256 for decrementing Down Counter. It is used in timer mode only.

Fig. 6 - CHANNEL BLOCK DIAGRAM


CHANNEL CONTROL REGISTER AND LOGIC (8 BITS)
TIME CONSTANT REGISTER (8 BITS)
INTERNAL BUS
Φ
PRESCALER (8 BITS)
DOWN COUNTER (8 BITS)
ZERO COUNT/TIMEOUT
EXTERNAL CLOCK/TIMER TRIGGER

Z80–CTC Pin Description



CLK/TRG∅ Channel ∅ External Clock or Timer Trigger (Input)

CLK/TRG1 Channel 1 External Clock or Timer Trigger (Input)

CLK/TRG2 Channel 2 External Clock or Timer Trigger (Input)

CLK/TRG3 Channel 3 External Clock or Timer Trigger (Input)

ZC/TO∅ Channel ∅ Zero Count or Timeout (output, active high)

Z80–CTC Pin Description (continued)

ZC/TO1 Channel 1 Zero Count or Timeout (output, active high)

ZC/TO2 Channel 2 Zero Count or Timeout (output, active high)

CS1 – CS∅ Channel Select (input, active high). These form a 2-bit binary address of the channel to be accessed.

D7 – D∅ Z80-CPU Data Bus (bidirectional, tristate)

CE Chip Enable (input, active low)

Φ System Clock (input)

M1 Machine Cycle One Signal from Z80-CPU (input, active low)

IORQ Input/Output Request from Z80-CPU (input, active low)

RD Read Cycle Status from the Z80-CPU (input, active low)

IEI Interrupt Enable In (input, active high)

IEO Interrupt Enable Out (output, active high). IEI and IEO form a daisy chain connection for priority interrupt control

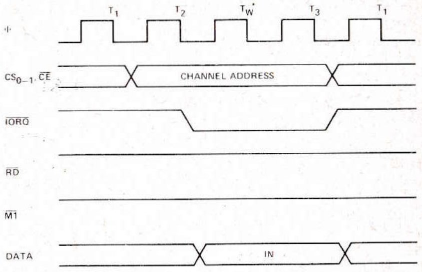
INT Interrupt Request (output, open drain, active low)

RESET RESET stops all channels from counting and resets channel interrupt enable bits in all control registers. During reset time ZC/TO∅-2 and INT go to the inactive states, IEO reflects the state of IEI, and the data bus output drivers go to the high impedance state (input, active low)

Timing Waveforms

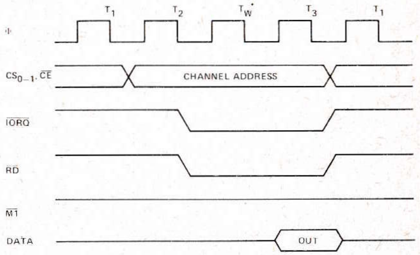
CTC WRITE CYCLE

Illustrated here is the timing for loading a channel control word, time constant and interrupt vector. No wait states are allowed for writing to the CTC other than the automatically inserted (T_W*). Since the CTC does not receive a specific write signal, it internally generates its own from the lack of an RD signal.



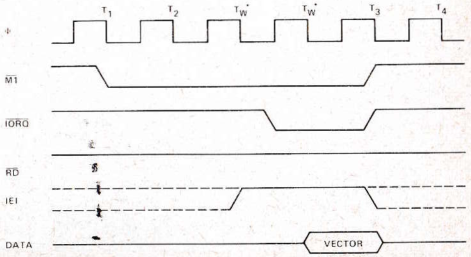
CTC READ CYCLE

Illustrated here is the timing for reading a channel's Down Counter when in Counter Mode. The value read onto the data bus reflects the number of external clock's rising edges prior to the rising edge of cycle (T_2). No wait states are allowed for reading the CTC other than the automatically inserted (T_W*).



INTERRUPT ACKNOWLEDGE CYCLE

Some time after an interrupt is requested by the CTC, the CPU will send out an interrupt acknowledge (M1 and IORQ). During this time the interrupt logic of the CTC will determine the highest priority channel which is requesting an interrupt. To insure that the daisy chain enable lines stabilize, channels are inhibited from changing their interrupt request status when M1 is active. If the CTC Interrupt Enable Input (IEI) is active, then the highest priority interrupting channel places the contents of its interrupt vector register onto the Data Bus when IORQ goes active. Additional wait cycles are allowed.



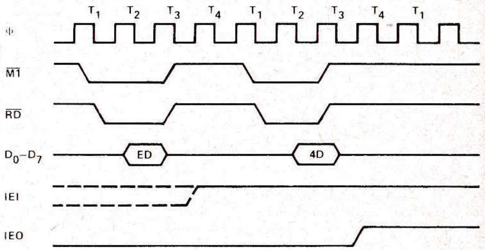
Timing Waveforms (continued)

RETURN FROM INTERRUPT CYCLE

If a Z80 peripheral device has no interrupt pending and is not under service, then its IEO = IEI. If it has an interrupt under service (i.e. it has already interrupted and received an interrupt acknowledge) then its IEO is always low, inhibiting lower priority chips from interrupting. If it has an interrupt pending which has not yet been acknowledged, IEO will be low unless an "ED" is decoded as the first byte of a two byte opcode. In this case, IEO will go high until the next opcode byte is decoded, whereupon it will again go low. If the second byte of the opcode was a "4D" then the opcode was an RETI instruction.

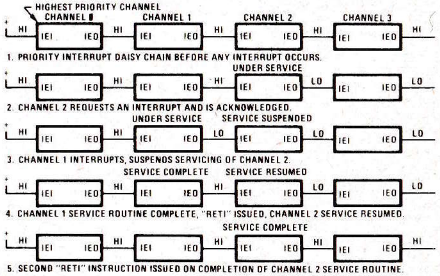
After an "ED" opcode is decoded, only the peripheral device which has interrupted and is currently under service will have its IEI high and its IEO low. This device is the highest priority device in the daisy chain which has received an interrupt acknowledge. All other peripherals have IEI = IEO. If the next opcode byte decoded is "4D", this peripheral device will reset its "interrupt under service" condition.

Wait cycles are allowed in the $\overline{M1}$ cycles.



DAISY CHAIN INTERRUPT SERVICING

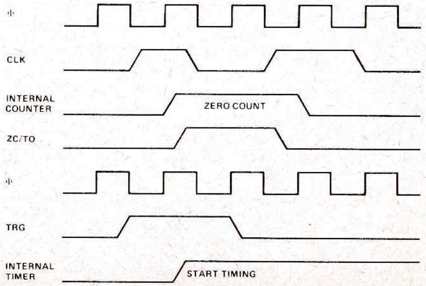
Illustrated at right is a typical nested interrupt sequence which may occur in the CTC. In this sequence channel 2 interrupts and is granted service. While this channel is being serviced, higher priority channel 1 interrupts and is granted service. The service routine for the higher priority channel is completed and a RETI instruction is executed to indicate to the channel that its routine is complete. At this time the service routine of lower priority channel 2 is completed.


HIGHEST PRIORITY CHANNEL
CHANNEL Ø
CHANNEL 1
CHANNEL 2
CHANNEL 3
HI
IEI
IEO
LO
1. PRIORITY INTERRUPT DAISY CHAIN BEFORE ANY INTERRUPT OCCURS.
UNDER SERVICE
2. CHANNEL 2 REQUESTS AN INTERRUPT AND IS ACKNOWLEDGED.
UNDER SERVICE
SERVICE SUSPENDED
3. CHANNEL 1 INTERRUPTS, SUSPENDS SERVICING OF CHANNEL 2.
SERVICE COMPLETE
SERVICE RESUMED
4. CHANNEL 1 SERVICE ROUTINE COMPLETE, "RETI" ISSUED, CHANNEL 2 SERVICE RESUMED.
SERVICE COMPLETE
5. SECOND "RETI" INSTRUCTION ISSUED ON COMPLETION OF CHANNEL 2 SERVICE ROUTINE.

CTC COUNTING AND TIMING

In the counter mode the rising or falling edge of the CLK input causes the counter to be decremented. The edge is detected totally asynchronously and must have a minimum CLK pulse width. However, the counter is synchronous with Φ therefore a setup time must be met when it is desired to have the counter decremented by the next rising edge of Φ.

In the timer mode the prescaler may be enabled by a rising or falling edge on the TRG input. As in the counter mode, the edge is detected totally asynchronously and must have a minimum TRG pulse width. However, when timing is to start with respect to the next rising edge of Φ a setup time must be met. The prescaler counts rising edges of Φ.



CTC Programming

SELECTING AN OPERATING MODE

When selecting a channel's operating mode, bit Ø is set to 1 to indicate this word is to be stored in the channel control register.

D7	D6	D5	D4	D3	D2	D1	D0
INTERRUPT ENABLE	MODE	RANGE	SLOPE	TRIGGER	LOAD TIME CONSTANT	RESET	1
		USED IN TIMER MODE ONLY		USED IN TIMER MODE ONLY			

Bit 7 = Ø   Channel interrupts disabled.

Bit 7 = 1   Channel interrupts enabled to occur every time Down Counter reaches a count of zero. Setting Bit 7 does not let a preceding count of zero cause an interrupt.

Bit 6 = Ø   Timer Mode – Down counter is clocked by the prescaler. The period of the counter is:
$t_C \bullet P \bullet TC$
t_C = system clock period
P = prescale of 16 or 256
TC = 8 bit binary programmable time constant (256 max)

Bit 6 = 1   Counter Mode – Down Counter is clocked by external clock. The prescaler is not used.

Bit 5 = Ø   Timer Mode Only–System clock Φ is divided by 16 in prescaler.

Bit 5 = 1   Timer Mode Only–System clock Φ is divided by 256 in prescaler.

Bit 4 = Ø   Timer Mode – negative edge trigger starts timer operation.
Counter Mode – negative edge decrements the down counter.

Bit 4 = 1   Timer Mode – positive edge trigger starts timer operation.
Counter Mode – positive edge decrements the down counter.

Bit 3 = Ø   Timer Mode Only – Timer begins operation on the rising edge of T_2 of the machine cycle following the one that loads the time constant.

Bit 3 = 1   Timer Mode Only – External trigger is valid for starting timer operation after rising edge of T_2 of the machine cycle following the one that loads the time constant. The Prescaler is decremented 2 clock cycles later if the setup time is met, otherwise 3 clock cycles.

Bit 2 = Ø   No time constant will follow the channel control word. One time constant must be written to the channel to initiate operation.

Bit 2 = 1   The time constant for the Down Counter will be the next word written to the selected channel. If a time constant is loaded into a channel while a channel is counting, the present count will be completed before the new time constant is loaded into the Down Counter.

Bit 1 = Ø   Channel continues counting.

Bit 1 = 1   Stop operation. If Bit 2 = 1 channel will resume operation after loading a time constant, otherwise a new control word must be loaded.

LOADING A TIME CONSTANT

An 8-bit time constant is loaded into the Time Constant register following a channel control word with bit 2 set. All zeros indicate a time constant of 256.

D7	D6	D5	D4	D3	D2	D1	D0
TC_7	TC_6	TC_5	TC_4	TC_3	TC_2	TC_1	TC_0

LOADING AN INTERRUPT VECTOR

The Z80-CPU requires that an 8-bit interrupt vector be supplied by the interrupting channel. The CPU forms the address for the interrupt service routine of the channel using this vector. During an interrupt acknowledge cycle the vector is placed on the Z80 Data Bus by the highest priority channel requesting service at that time. The desired interrupt vector is loaded into the CTC by writing into channel Ø with a zero in DØ. D_7-D_3 contain the stored interrupt vector, D_2 and D_1 are not used in loading the vector. When the CTC responds to an interrupt acknowledge, these two bits contain the binary code of the highest priority channel which requested the interrupt and DØ contains a zero since the address of the interrupt service routine starts at an even byte. Channel Ø is the highest priority channel.

D7	D6	D5	D4	D3	D2	D1	D0
V_7	V_6	V_5	V_4	V_3	X	X	0

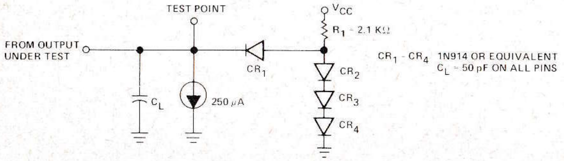
Z80–CTC A.C. Characteristics

TA = 0° C to 70° C, Vcc = +5 V ± 5%, unless otherwise noted

Signal	Symbol	Parameter	Min	Max	Unit	Comments
Φ	t_C	Clock Period	400	[1]	ns	
	$t_W(\Phi H)$	Clock Pulse Width, Clock High	170	2000	ns	
	$t_W(\Phi L)$	Clock Pulse Width, Clock Low	170	2000	ns	
	t_r, t_f	Clock Rise and Fall Times		30	ns	
	t_H	Any Hold Time for Specified Setup Time	0		ns	
CS, $\overline{CE}$, etc.	$t_{S\Phi}(CS)$	Control Signal Setup Time to Rising Edge of Φ During Read or Write Cycle	160		ns	
D_0–D_7	$t_{DR}(D)$	Data Output Delay from Rising Edge of $\overline{RD}$ During Read Cycle		480	ns	[2]
	$t_{S\Phi}(D)$	Data Setup Time to Rising Edge of Φ During Write or M1 Cycle	60		ns	
	$t_{DI}(D)$	Data Output Delay from Falling Edge of IORQ During INTA Cycle		340	ns	[2]
	$t_F(D)$	Delay to Floating Bus (Output Buffer Disable Time)		230	ns	
IEI	$t_S(IEI)$	IEI Setup Time to Falling Edge of $\overline{IORQ}$ During INTA Cycle	200		ns	
IEO	$t_{DH}(IO)$	IEO Delay Time from Rising Edge of IEI		220	ns	[3]
	$t_{DL}(IO)$	IEO Delay Time from Falling Edge of IEI		190	ns	[3]
	$t_{DM}(IO)$	IEO Delay from Falling Edge of $\overline{M1}$ (Interrupt Occurring just Prior to $\overline{M1}$)		300	ns	[3]
$\overline{IORQ}$	$t_{S\Phi}(IR)$	$\overline{IORQ}$ Setup Time to Rising Edge of Φ During Read or Write Cycle	250		ns	
$\overline{M1}$	$t_{S\Phi}(M1)$	$\overline{M1}$ Setup Time to Rising Edge of Φ During INTA or M1 Cycle	210		ns	
$\overline{RD}$	$t_{S\Phi}(RD)$	$\overline{RD}$ Setup Time to Rising Edge of Φ During Read or M1 Cycle	240		ns	
$\overline{INT}$	$t_{DCK}(IT)$	$\overline{INT}$ Delay Time from Rising Edge of CLK/TRG		$2t_C(\Phi) + 200$		Counter Mode
	$t_{D\Phi}(IT)$	$\overline{INT}$ Delay Time from Rising Edge of Φ		$t_C(\Phi) + 200$		Timer Mode
CLK/TRG_{0-3}	$t_C(CK)$	Clock Period	$2t_C(\Phi)$			Counter Mode
	t_r, t_f	Clock and Trigger Rise and Fall Times		50		
	$t_S(CK)$	Clock Setup Time to Rising Edge of Φ for Immediate Count	210			Counter Mode
	$t_S(TR)$	Trigger Setup Time to Rising Edge of Φ for Enabling of Prescaler on Following Rising Edge of Φ	210			Timer Mode
	$t_W(CTH)$	Clock and Trigger High Pulse Width	200			Counter and Timer Modes
	$t_W(CTL)$	Clock and Trigger Low Pulse Width	200			Counter and Timer Modes
ZC/TO_{0-2}	$t_{DH}(ZC)$	ZC/TO Delay Time from Rising Edge of Φ, ZC/TO High		190		Counter and Timer Modes
	$t_{DL}(ZC)$	ZC/TO Delay Time from Falling Edge of Φ, ZC/TO Low		190		Counter and Timer Modes

Notes:
[1] $t_C = t_W(\Phi H) + t_W(\Phi L) + t_r + t_f$.
[2] Increase delay by 10 nsec for each 50 pF increase in loading, 200 pF maximum for data lines and 100 pF for control lines.
[3] Increase delay by 2 nsec for each 10 pF increase in loading, 100 pF maximum
[4] $\overline{RESET}$ must be active for a minimum of 3 clock cycles.

OUTPUT LOAD CIRCUIT



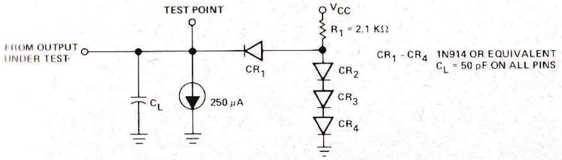
Z80A–CTC A.C. Characteristics

TA = 0° C to 70° C, Vcc = +5 V ± 5%, unless otherwise noted

Signal	Symbol	Parameter	Min	Max	Unit	Comments
Φ	t_C	Clock Period	250	[1]	ns	
	$t_W(\Phi H)$	Clock Pulse Width, Clock High	105	2000	ns	
	$t_W(\Phi L)$	Clock Pulse Width, Clock Low	105	2000	ns	
	t_r, t_f	Clock Rise and Fall Times		30	ns	
	t_H	Any Hold Time for Specified Setup Time	0		ns	
CS, $\overline{CE}$, etc	$t_{S\Phi}(CS)$	Control Signal Setup Time to Rising Edge of Φ During Read or Write Cycle	60		ns	
D_0 D_7	$t_{DR}(D)$	Data Output Delay from Falling Edge of $\overline{RD}$ During Read Cycle		380	ns	[2]
	$t_{S\Phi}(D)$	Data Setup Time to Rising Edge of Φ During Write or M1 Cycle	50		ns	
	$t_{DI}(D)$	Data Output Delay from Falling Edge of IORG During INTA Cycle		160	ns	[2]
	$t_F(D)$	Delay to Floating Bus (Output Buffer Disable Time)		110	ns	
IEI	$t_S(IEI)$	IEI Setup Time to Falling Edge of $\overline{IORQ}$ During INTA Cycle	140		ns	
IEO	$t_{DH}(IO)$	IEO Delay Time from Rising Edge of IEI		160	ns	[3]
	$t_{DL}(IO)$	IEO Delay Time from Falling Edge of IEI		130	ns	[3]
	$t_{DM}(IO)$	IEO Delay from Falling Edge of $\overline{M1}$ (Interrupt Occurring just Prior to $\overline{M1}$)		190	ns	[3]
$\overline{IORQ}$	$t_{S\Phi}(IR)$	$\overline{IORQ}$ Setup Time to Rising Edge of Φ During Read or Write Cycle	115		ns	
$\overline{M1}$	$t_{S\Phi}(M1)$	$\overline{M1}$ Setup Time to Rising Edge of Φ During INTA or M1 Cycle	90		ns	
$\overline{RD}$	$t_{S\Phi}(RD)$	$\overline{RD}$ Setup Time to Rising Edge of Φ During Read or M1 Cycle	115		ns	
$\overline{INT}$	$t_{DCK}(IT)$	$\overline{INT}$ Delay Time from Rising Edge of CLK/TRG		$2t_C(\Phi) + 140$		Counter Mode
	$t_{D\Phi}(IT)$	$\overline{INT}$ Delay Time from Rising Edge of Φ		$t_C(\Phi) + 140$		Timer Mode
CLK/TRG_{0-3}	$t_C(CK)$	Clock Period	$2t_C(\Phi)$			Counter Mode
	t_r, t_f	Clock and Trigger Rise and Fall Times		30		
	$t_S(CK)$	Clock Setup Time to Rising Edge of Φ for Immediate Count	130			Counter Mode
	$t_S(TR)$	Trigger Setup Time to Rising Edge of Φ for enabling of Prescaler on Following Rising Edge of Φ	130			Timer Mode
	$t_W(CTH)$	Clock and Trigger High Pulse Width	120			Counter and Timer Modes
	$t_W(CTL)$	Clock and Trigger Low Pulse Width	120			Counter and Timer Modes
ZC/TO_{0-2}	$t_{DH}(ZC)$	ZC/TO Delay Time from Rising Edge of Φ, ZC/TO High		120		Counter and Timer Modes
	$t_{DL}(ZC)$	ZC/TO Delay Time from Rising Edge of Φ, ZC/TO Low		120		Counter and Timer Modes

Notes:
[1] $t_C = t_W(\Phi H) + t_W(\Phi L) + t_r + t_f$.
[2] Increase delay by 10 nsec for each 50 pF increase in loading, 200 pF maximum for data lines and 100 pF for control lines.
[3] Increase delay by 2 nsec for each 10 pF increase in loading, 100 pF maximum.
[4] $\overline{RESET}$ must be active for a minimum of 3 clock cycles.

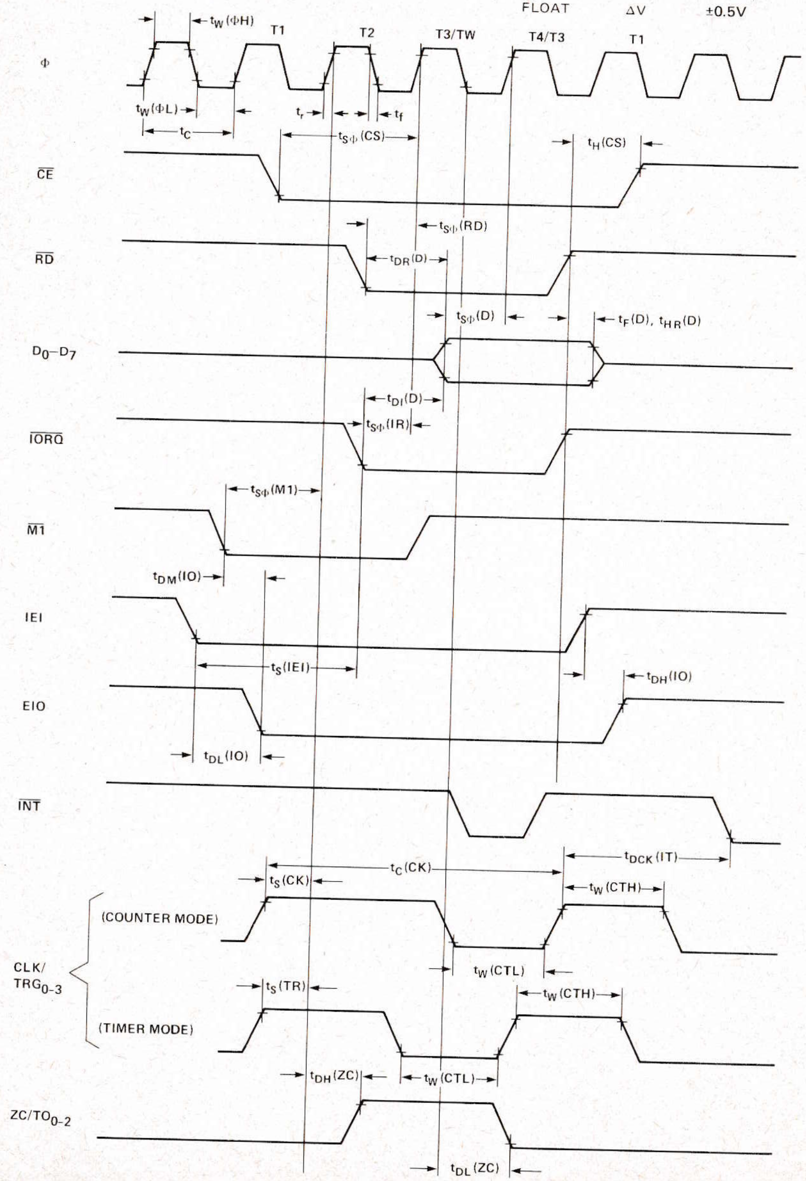
OUTPUT LOAD CIRCUIT



A.C. Timing Diagram

Timing measurements are made at the following voltages, unless otherwise specified:

	"1"	"0"
CLOCK	V_{CC}–0.6V	0.45V
OUTPUT	2 V	0.8V
INPUT	2 V	0.8V
FLOAT	ΔV	±0.5V



Absolute Maximum Ratings

Temperature Under Bias	0°C to 70°C
Storage Temperature	-65°C to +150°C
Voltage On Any Pin with Respect to Ground	-0.3V to +7V
Power Dissipation	0.8W

* Comment
Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Z80-CTC D.C. Characteristics

T_A = 0°C to 70°C, V_{cc}= 5V ± 5% unless otherwise specified

Symbol	Parameter	Min	Max	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	–0.3	0.45	V	
V_{IHC}	Clock Input High Voltage [1]	V_{CC}–0.6	V_{CC}+0.3	V	
V_{IL}	Input Low Voltage	–0.3	0.8	V	
V_{IH}	Input High Voltage	2	V_{CC}	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = –250 µA
I_{CC}	Power Supply Current		120	mA	T_C = 400 nsec
I_{LI}	Input Leakage Current		10	µA	V_{IN} = 0 to V_{CC}
I_{LOH}	Tri-State Output Leakage Current in Float		10	µA	V_{OUT} = 2.4 to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float		–10	µA	V_{OUT} = 0.4V
I_{OHD}	Darlington Drive Current	–1.5		mA	V_{OH} = 1.5V R_{EXT} = 390Ω

Z80A-CTC D.C. Characteristics

T_A = 0°C to 70°C, V_{cc}= 5V ± 5% unless otherwise specified

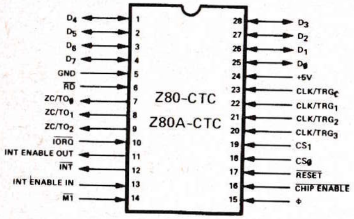
Symbol	Parameter	Min	Max	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	–0.3	0.45	V	
V_{IHC}	Clock Input High Voltage [1]	V_{CC}–0.6	V_{CC}+0.3	V	
V_{IL}	Input Low Voltage	–0.3	0.8	V	
V_{IH}	Input High Voltage	2	V_{CC}	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = –250 µA
I_{CC}	Power Supply Current		120	mA	T_C = 250 nsec
I_{LI}	Input Leakage Current		10	µA	V_{IN} = 0 to V_{CC}
I_{LOH}	Tri-State Output Leakage Current in Float		10	µA	V_{OUT} = 2.4 to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float		–10	µA	V_{OUT} = 0.4V
I_{OHD}	Darlington Drive Current	–1.5		mA	V_{OH} = 1.5V R_{EXT} = 390Ω

Capacitance

TA = 25° C, f = 1 MHz

Symbol	Parameter	Max.	Unit	Test Condition
C_Φ	Clock Capacitance	20	pF	Unmeasured Pins Returned to Ground
C_{IN}	Input Capacitance	5	pF	
C_{OUT}	Output Capacitance	10	pF	

Package Configuration

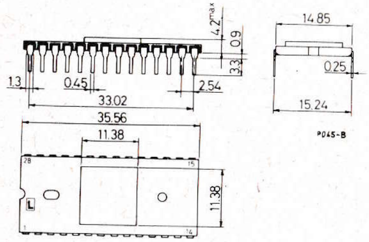


ORDERING NUMBERS:

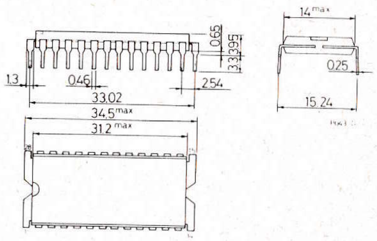
Z80–CTC	D1	for dual in–line ceramic slam package
Z80–CTC	B1	for dual in–line plastic package
Z80A–CTC	D1	for dual in–line ceramic slam package
Z80A–CTC	B1	for dual in–line plastic package

MECHANICAL DATA (dimensions in mm)

28–PIN CERAMIC DUAL IN–LINE SLAM PACKAGE



28–PIN PLASTIC DUAL IN–LINE PACKAGE



Product Specification

SGS-ATES Z80 microcomputer product line includes a third generation LSI component set, development systems and support software. The component set includes all the logic circuits necessary for the user to build high performance microcomputer systems with virtually no external logic and a minimal number of standard low-cost memory components. The Z80-DMA (Direct Memory Access) circuit is a programmable single-channel device which provides all address, timing and control signals to effect the transfer of blocks of data between two ports within a Z80-CPU based system. These ports may be either system main memory or any system peripheral I/O device. The DMA can also search a block of data for a particular byte (bit maskable), with or without a simultaneous transfer.

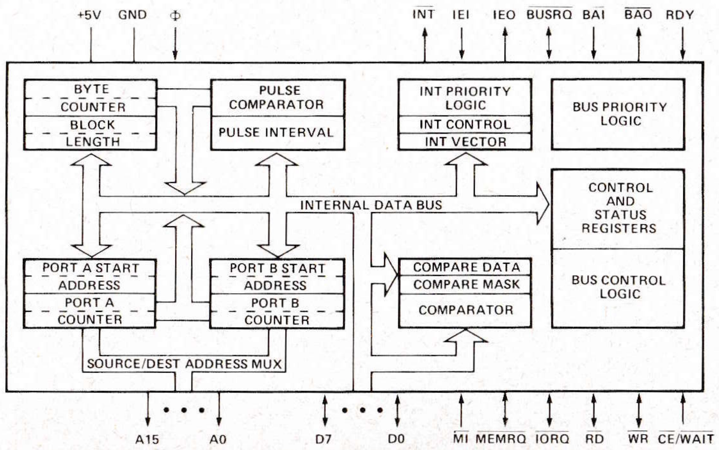
Structure

- N-channel Silicon Gate Depletion Load Technology
- 40 Pin DIP
- Single 5 volt supply
- Single phase 5 volt clock
- Single channel, two port

Features

- Three classes of operation:
  - Transfer Only
  - Search Only
  - Search-Transfer
- Address and Block Length Registers fully buffered. Values for next operation may be loaded without disturbing current values.
- Dual addresses generated during a transfer (one for read port and one for write).
- Programmable data transfers and searches, automatically incrementing or decrementing the port addresses from programmed starting addresses (they can also remain fixed).
- Four modes of operation:
  - –Byte-at-a-time: One byte transferred per request
  - –Burst: Continues as long as ports are ready
  - –Continuous: Locks out CPU until operation complete
  - –Transparent: Steals refresh cycles
- Timing may be programmed to match the speed of any port.
- Interrupts on Match Found, End of Block, or Ready, may be programmed.
- An entire previous operation may be repeated automatically or on command. (Auto restart or Load)
- The DMA can signal when a specified number of bytes has been transferred, without halting transfer.
- Multiple DMA's easily configured for rotating priority.
- The channel may be enabled, disabled or reset under software control.
- Complete channel status upon program (CPU) request.
- Up to 1.25 megabyte Search or Transfer Rate.
- Daisy-chain priority interrupt and bus acknowledge included to provide automatic interrupt vectoring and bus request control, without need for additional external logic.
- TTL compatible inputs and outputs
- The CPU can read current Port counters, byte counters, or status. A mask word can be set which defines which registers can be accessed during read operations.

Fig. 7 - DMA INTERNAL BLOCK DIAGRAM



DMA Architecture

A block diagram of the Z80 DMA is shown in Figure 7. The internal structure consists of the following circuitry:

- *Bus Interface*: provides driver and receiver circuitry to interface to the Z80-CPU Bus.
- *Control Logic and Registers*: set the class, mode and other basic control parameters of the DMA.
- *Address, Byte Count and Pulse Circuitry*: generates the proper port addresses for the read and write operations, with provisions for incrementing or decrementing the address. When zero bytes remain to be handled, the byte count circuitry sets a flag in the status register. Pulse circuitry generates a pulse each time the byte counter lower 8-bits equal the pulse reg.
- *Timing Circuitry*: allows the user to completely specify the read/write timing for both of the channels' addressed ports.
- *Match Circuitry*: holds the match byte and a mask byte which allows for the comparison of only certain bits within the byte. If a match is encountered during a Search or Transfer, this circuitry sets a flag in the status register.
- *Interrupt and $\overline{\text{BUSRQ}}$ Circuitry*: includes a control register which specifies the conditions under which the DMA can generate an interrupt; priority encoding logic to select between the generation of an $\overline{\text{INT}}$ or $\overline{\text{BUSRQ}}$ output under these conditions; and an interrupt vector register for automatic vectoring to the interrupt service routine.
- *Status Register*: holds current status of DMA.

Register Description

The following DMA-internal registers are available to the programmer:

- *Control Registers*: Hold DMA control information: such as, when to initiate an interrupt or pulse, what mode or class of operation to perform, etc. (Write Only) (8 Bits)
- *Timing Registers*: Hold read/write timing parameters for the two ports. (Write Only) (8 bits)
- *Interrupt Vector Register*: Holds the 8-bit vector that the DMA will put onto the data bus after receiving an $\overline{\text{IORQ}}$ during an interrupt acknowledge sequence if it is the highest priority device requesting an interrupt. (This register is readable only during interrupt acknowledge cycles.) (Read/Write) (8 bits)
- *Block Length Register*: Contains total block length of data to be searched and/or transferred. (Write Only) (16 bits)
- *Byte Counter*: Counts number of bytes transferred (or searched). On a Load or Continue the Byte Counter is reset to zero. Thereafter, each byte transfer operation increments it until it matches the contents of the Block Length Register, at which time End of Block is set in the status register and operation is suspended if programmed. Also if so programmed the DMA will generate an interrupt. (Read Only) (16 bits)
- *Compare Register*: Holds the byte for which a match is being sought in Search operations. (Write Only) (8 bits)
- *Mask Register*: Holds the 8 bit mask to determine which bits in the compare register are to be examined for a match. (Write Only) (8 bits)
- *Starting Address Registers (Port A and Port B)*: Hold the starting addresses (upper and lower 8 bits) for the two ports involved in Transfer operations. In Search Only operations, only one port address would have to be specified. Only memory starting addresses require both upper and lower 8-bits; I/O ports are generally addressed with only the lower 8-bits, and in this case the address contained in the register is a generally fixed address. (Write Only) (16 bits each)
- *Address Counters (Port A and Port B)*: These counters are loaded with the contents of the corresponding Starting Address Registers whenever Searches or Transfers are initiated with a Load or Continue. They are incremented, decremented or remain fixed, as programmed. (Read Only) (16 bits each)
- *Pulse Control Register*: Holds program-supplied length (in bytes) of block after which the DMA will provide a signal pulse on the $\overline{\text{INT}}$ pin. (Since this occurs while both $\overline{\text{BUSRQ}}$ and $\overline{\text{BUSAK}}$ are active, the CPU will not interpret this as an interrupt request. Instead, the signal is used to communicate with a peripheral I/O device.) (Write Only) (8 bits)
- *Status Register*: Match, End of Block, Ready Active, Interrupt Pending, and Write Address Valid bits indicate these functions when set. (Read Only) (8 bits)

Modes of Operation

The DMA may be programmed for one of four modes of operation. (See Command Byte 2B).

- *Byte at a time*: control is returned to the CPU after each one-byte cycle.
- *Burst*: operation continues as long as the DMA's RDY input is active, indicating that the relevant port is ready. Control returns to the CPU when RDY is inactive or at end of block or a match if so programmed.
- *Continuous*: the entire Search and/or Transfer of a block of data is completed before control is returned to CPU.
- *Transparent*: DMA operation occurs during normal memory refresh times without visible loss of CPU time.

Classes of Operation

The DMA has three classes of operation: Transfer only, Search Only and a combined Search-Transfer. (See Command Byte 1A.)

During a Transfer, data is read from one port and written to the other port, byte by byte. (The DMA's two ports are termed Port A and Port B.) The ports may be programmed to be either system main memory or peripheral I/O devices. Thus, a block of data might be written from a peripheral to another; or it might be written from one area in main memory to another; or from a peripheral to main memory.

During a Search, data is read only, and compared byte by byte against two DMA-internal registers, one of which contains a match byte and the other an optional mask byte which allows only certain bits to be compared. If any byte of searched data matches, a DMA-internal status bit is set; if programmed to do so, the DMA will then suspend operation and/or generate an interrupt.

The third class of operation is a combined Search-Transfer. In such an operation a block of data is transferred as described above until a match is found; then, as in a Search Only operation, the transfer may be suspended and/or an interrupt generated.

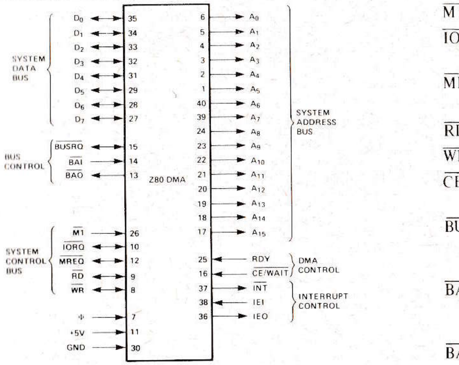
Addressing

The DMA's addressing of ports is either fixed or sequential, incrementing or decrementing from a starting address. The length of the operation (number of bytes) is specified by the programmed contents of a block length register. The DMA can address block lengths of up to 64K bytes. During a transfer two separate port addresses are generated, one during the Read cycle and one during the Write cycle.

Operating Sequence

Once the DMA has been programmed it may be "Enabled" (command byte 2d). In the enabled condition when Ready goes active the DMA will request the bus by bringing $\overline{\text{BUSRQ}}$ low. The CPU will acknowledge this with a $\overline{\text{BUS ACK}}$ which will normally be attached to $\overline{\text{BAI}}$. When the DMA receives $\overline{\text{BAI}}$ it will start its programmed operation releasing $\overline{\text{BUSRQ}}$ to a "high" state when it is through.

Z80-DMA Pin Description

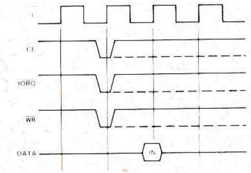


A_0–A_{15}	System Address Bus. All sixteen of these pins are used by the DMA to address system main memory or an I/O port (output)
D_0–D_7	System Data Bus. Commands from the CPU, DMA status and data from memory or peripherals are transferred on these tristate pins (input/output)
+5V	Power
GND	Ground
Φ	System clock (input)
$\overline{\text{M1}}$	Machine cycle One signal from CPU (input)
$\overline{\text{IORQ}}$	Input/Output Request to and from the System Bus (input/output)
$\overline{\text{MREQ}}$	Memory REQuest to the System Bus (input/output)
$\overline{\text{RD}}$	ReaD to and from the System Bus (input/output)
$\overline{\text{WR}}$	WRite to and from the System Bus (input/output)
$\overline{\text{CE/WAIT}}$	Chip Enable; may also be programmed to be $\overline{\text{WAIT}}$ during time when $\overline{\text{BAI}}$ is low (input)
$\overline{\text{BUSRQ}}$	BUS ReQuest. Requests control of the CPU Address Bus, Data Bus and Status/Control Bus (input/output)
$\overline{\text{BAI}}$	Bus Acknowledge In. Signals that the system buses have been released for DMA control (input)
$\overline{\text{BAO}}$	Bus Acknowledge Out. $\overline{\text{BAI}}$ and $\overline{\text{BAO}}$ form a daisy-chain connection for system-wide priority bus control (output)
$\overline{\text{INT}}$	INTerrupt request (output)
IEI	Interrupt Enable In (input)
IEO	Interrupt Enable Out. IEI and IEO form a daisy-chain connection for system-wide priority interrupt control (output)
RDY	ReaDY is monitored by the DMA to determine when a peripheral device associated with a DMA port is ready for a read or write operation (input, programmable as active high or low)

DMA Timing Waveforms

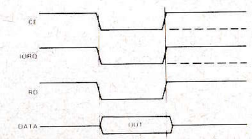
DMA Command Write Cycle

Illustrated here is the timing associated with a command byte or control byte being written to the DMA which is to be loaded into internal registers. Z80 Output instructions satisfy this timing.



DMA Register Read Cycle

This timing is used when a read operation is performed on the DMA to access the contents of the Status Register, Address Counter or other readable registers. Z80 Input instructions satisfy this timing.

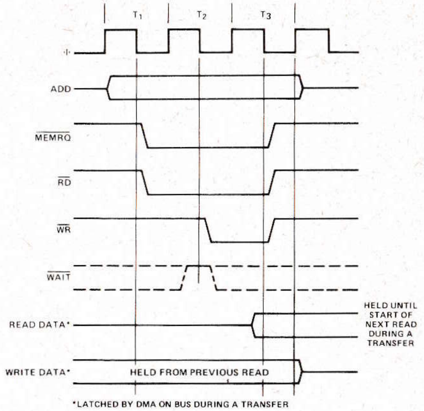


DMA Timing Waveforms (continued)

STD Memory Timing

This timing is exactly the same as used by the Z80-CPU to access system main memory, either in a Read or Write operation. The DMA will default to this timing after a power-on reset, or when a Reset or Reset Timing command is written to it; and unless otherwise programmed, will use this timing during all Transfer or Search operations involving system main memory. During the memory Read portion of a transfer cycle, data is latched in the DMA on the negative edge of Φ during T_3 and held into the following Write cycle. During the memory Write portion of a transfer cycle, data is held from the previous Read cycle and released at the end of the present cycle.

NOTE: The DMA is normally programmed for a 3 T-cycle duration in memory transactions. But $\overline{\text{WAIT}}$ is sampled during the negative transition of T_2, and if it is low, T_2 will be extended another T-cycle, after which $\overline{\text{WAIT}}$ will again be sampled. The duration of a memory transaction cycle may thus be indefinitely extended.

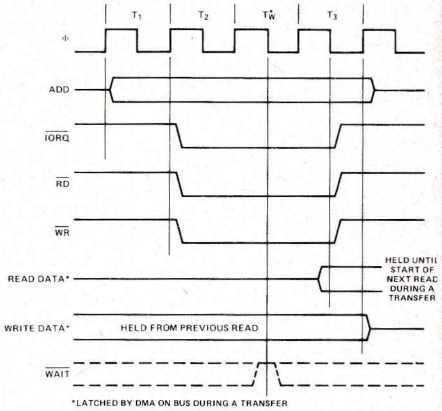


*LATCHED BY DMA ON BUS DURING A TRANSFER

STD Peripheral Timing

This timing is identical to the Z80-CPU's Read/Write timing to I/O peripheral devices. The DMA will default to this timing after a power-on reset, or when a Reset or Reset Timing command is written to it; and unless otherwise programmed, will use this timing during all Transfer or Search operations involving I/O peripherals. During the I/O Read of a transfer cycle, data is latched on the negative edge of Φ during T_3 and is then held into the Write cycle. During an I/O Write, data is held from the previous Read cycle until the end of the Write cycle.

NOTE: If $\overline{\text{WAIT}}$ is low during the negative transition of T_W*, then T_W* will be extended another T-cycle and $\overline{\text{WAIT}}$ will again be sampled. The duration of a peripheral transaction cycle may thus be indefinitely extended.

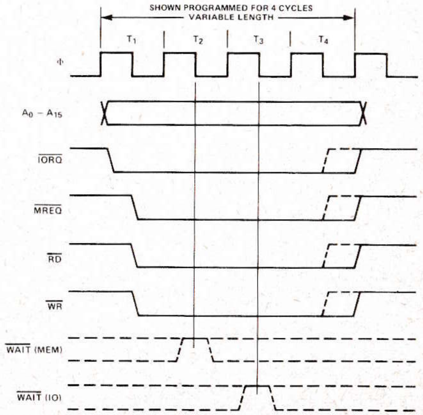


*LATCHED BY DMA ON BUS DURING A TRANSFER

Variable Cycle

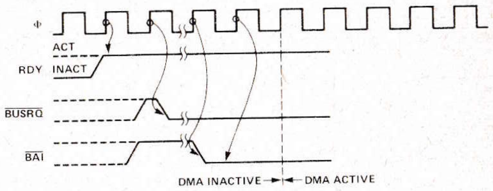
The Variable feature of the DMA allows the user to program the DMA's memory or peripheral transaction timing to values different than given above in the standard default diagrams. This permits the designer to tailor his timing to the particular requirements of his system components, and maximizes the data transfer rate while eliminating external signal conditioning logic. Cycle length can be one to four T-cycles (more if $\overline{\text{WAIT}}$ is used). Signal timing can be varied as shown. During a transfer, data will be latched by the DMA on the clock edge causing the rising edge of $\overline{\text{RD}}$ and will be held on the data lines until the end of the following Write cycle.

(See Timing Control Byte, page 41).



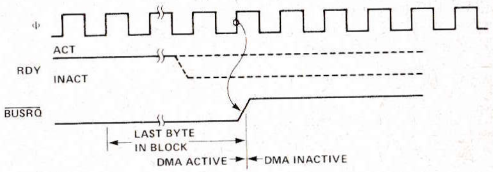
DMA Bus Request and Acceptance for Byte-at-a-Time, Burst, and Continuous Mode

Ready is sampled on every rising edge of Φ. When it is found to be active, the following rising edge of Φ generates $\overline{\text{BUSRQ}}$. After receiving $\overline{\text{BUSRQ}}$ the CPU will grant a $\overline{\text{BUSAK}}$ which will be connected to $\overline{\text{BAI}}$ either directly or through the Bus Acknowledge Daisy Chain. When a low is detected on $\overline{\text{BAI}}$ (sampled on every rising edge of Φ), the next rising edge of Φ will start an active DMA cycle.



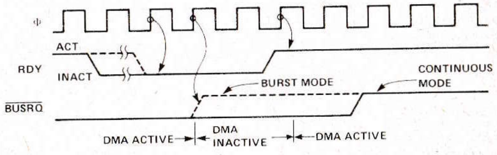
DMA Bus Release at End of Block for Burst or Continuous Mode

Timing for End of Block and DMA not programmed for Auto-restart.



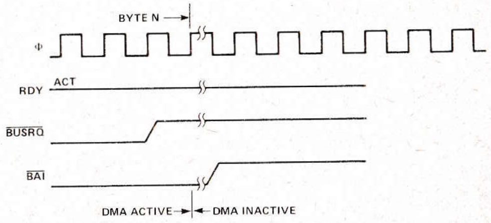
DMA Bus Release with 'Ready' for Burst and Continuous Mode

The DMA will relinquish the bus after RDY has gone inactive (Burst mode) or after an End of Block or a Match is found (Continuous mode). With RDY inactive, the DMA in Continuous mode is inactive but maintains control of the bus ($\overline{\text{BUSRQ}}$ low) until the cycle is resumed when RDY goes active.



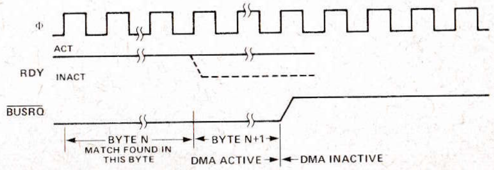
DMA Bus Release for Byte-at-a-Time Mode

In the Byte mode the DMA will release $\overline{\text{BUSRQ}}$ on the rising edge of Φ prior to the end of each Read cycle in Search Only or each Write cycle in a Transfer, regardless of the state of RDY. The next bus request will come after both $\overline{\text{BUSRQ}}$ and $\overline{\text{BAI}}$ have returned high.



DMA Bus Release with Match for Burst or Continuous Modes

When a Match is found and the DMA is programmed to stop on Compare, the DMA performs an operation on the next byte and then releases bus.



Reading from the DMA Internal Registers

Seven registers are available on the DMA for reading. They are: 8 bits of the status register, the upper and lower 8 bits of the block length register, and two port address registers.

These are available to be read sequentially: status, BLK Lower, BLK Upper, Port A Address lower, Port A Address Upper, Port B Address lower, Port B Address upper. An internal pointer points to each register in turn as each READ is accomplished. If a register is not to be read, it may be excluded by programming a 0 in the Read Byte. The internal pointer will skip any register not programmed with a 1 in the Read Byte. After a Reset or a Load, Reset RD must be given to set the internal pointer pointing to the first register programmed to be read by the Read Byte. After RD Status, the pointer will be pointing to the status register regardless of the mask and the next read will be from the status register. The following read will be from the register pointed to before RD Status.

Programming the DMA

Previous sections of this specification have indicated the various functions and modes of the DMA. The diagrams and charts below will show how the DMA is programmed to select among these functions and modes and to adapt itself to the requirements of the user system.

The Z80-DMA chip may be in an "enable" state, in which it can gain control of the system buses and direct the transfer of data between its ports, or in a "disable" state, when it cannot gain control of the bus. Program commands can be written to it in either state, but writing a command to it automatically puts it in the disable state, which is maintained until an enable command is issued to the DMA. The CPU must program it in advance of any data search or transfer by addressing it as an I/O port and sending it a sequence of 8 bit command bytes via the system data bus using Output instructions. When the DMA is powered up or reset by any means, the DMA will automatically be placed into a disable state, in which it can initiate neither bus requests nor data transfers nor interrupts.

The command bytes contain information to be loaded into the DMA's control and other registers and/or information to alter the state of the chip, such as an Enable Interrupt command. The command structure is designed so that certain bits in some commands can be set to alert the DMA to expect the next byte written to it to be for a particular internal register.

The following diagrams and charts give the function of each bit in the six different command bytes. Two of these are defined as being from Group 1, and are termed command bytes 1A and 1B. These Group 1 commands contain the most basic DMA set-up information. The other four are categorized as Group 2, and are termed commands 2A, 2B, 2C and 2D. Group 2 words specify more detailed set-up information.

Command Byte 1A

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	BLOCK LENGTH (UPPER) FOLLOWS	BLOCK LENGTH (LOWER) FOLLOWS	PORT A STARTING ADDRESS (UPPER) FOLLOWS	PORT A STARTING ADDRESS (LOWER) FOLLOWS	SOURCE PORT	CLASS CONTROL C_1	CLASS CONTROL C_0

Specifies Group 1

Byte 1A cannot be 00

C_1	C_0	Function
0	0	Not allowed. (Command Byte 1B)
0	1	Transfer Only.
1	0	Search Only.
1	1	Search and Transfer.

$D_2 = 1$ Port A is read from, Port B is written to (unless the Search Only Mode has been selected, in which case Port B is never addressed).

$D_2 = 0$ Port B is read from, Port A is written to (unless the Search Only Mode has been selected, in which case Port A is never addressed).

Command Byte 1B

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	TIMING BYTE FOLLOWS	ADDRESS FIXED	ADDRESS INCREMENTS/ DECREMENTS	I/O OR MEMORY	PORT A OR B	0	0

Specifies Group 1

Specifies Byte 1B

$D_4 = 1$ Address for this port increments after each byte.

$D_4 = 0$ Address for this port decrements after each byte.

$D_3 = 1$ This port addresses an I/O peripheral.

$D_3 = 0$ This port addresses main memory.

$D_2 = 1$ This word programs Port A.

$D_2 = 0$ This word programs Port B.

Command Byte 2A

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	ENABLE CHIP	ENABLE INTERRUPT	MATCH BYTE FOLLOWS	MASK BYTE FOLLOWS	STOP ON COMPARE	0	0

Specifies Group 2

Specifies Byte 2A

Programming the DMA (continued)

Command Byte 2B

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	MODE M_1	MODE M_0	INTERRUPT CONTROL BYTE FOLLOWS	PORT B UPPER ADDRESS FOLLOWS	PORT B LOWER ADDRESS FOLLOWS	0	1

Specifies Group 2

Specifies Byte 2B

M_1	M_0	Mode
0	0	Byte
0	1	Continuous
1	0	Burst
1	1	Transparent

Command Byte 2C

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	NOT USED	AUTOMATIC RESTART	$\overline{\text{WAIT}}$ MULTIPLEXED	READY HIGH/LOW	NOT USED	1	0

Specifies Group 2

Specifies Byte 2C

$D_5 = 1$ Automatically repeats entire operation when end of block is reached.

$D_5 = 0$ No affect.

$D_4 = 1$ $\overline{\text{CE}}$ and $\overline{\text{WAIT}}$ multiplexed on same pin.

$D_4 = 0$ $\overline{\text{CE}}$ only.

$D_3 = 1$ Ready active high.

$D_3 = 0$ Ready active low.

Command Byte 2D

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	f_4	f_3	f_2	f_1	f_0	1	1

Specifies Group 2

Specifies Byte 2D

Hex	f_4	f_3	f_2	f_1	f_0	
C3	1	0	0	0	0	Reset
C7	1	0	0	0	1	Reset Port A Timing
CB	1	0	0	1	0	Reset Port B Timing
CF	1	0	0	1	1	Load
D3	1	0	1	0	0	Continue
AB	0	1	0	1	0	Enable Int
AF	0	1	0	1	1	Disable Int
A3	0	1	0	0	0	Reset Int
87	0	0	0	0	1	Enable DMA
83	0	0	0	0	0	Disable DMA
BB	0	1	1	1	0	Read Byte Follows
A7	0	1	0	0	1	Reset RD
BF	0	1	1	1	1	RD Status
B3	0	1	1	0	0	Force Ready
B7	0	1	1	0	1	Enable After RETI
8B	0	0	0	1	0	Reset Status

Command Byte 2D Summary

Reset: Resets all interrupt circuitry, disables interrupts and bus req. logic.

Reset Timing A or B: Resets timing for Port A or B to standard Z80-CPU timing.

Load: Zeros Byte Counter and loads Starting Address for both Ports.

Continue: Resets byte counter only. Addresses continue from present location.

Enable Interrupt: Permits interrupt to occur.

Disable Interrupt: Inhibits interrupt from occurring.

Reset Interrupt: Resets and disables all interrupt circuits (similar to RETI).

Enable DMA, Disable DMA: Overall enable or disable for all operations except interrupts; does not reset any functions.

Read Byte Follows: Next write to DMA will contain a mask to program which readable registers are to be read.

Reset RD: Next read will be from 1st register set as readable by response mask.

RD Status: Next read will be from status register.

Force Ready: Ready will be considered active regardless of the state of external RDY pin. Used for Mem-Mem operations where no RDY signal is needed.

Enable after RETI: DMA will not request bus until after it has received a RETI.

RST Status: Resets Match and End of Block status bits.

Read Byte

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
NOT USED	PORT B UPPER ADDR	PORT B LOWER ADDR	PORT A UPPER ADDR	PORT A LOWER ADDR	BYTE UPPER COUNT	BYTE LOWER COUNT	STATUS

A "1" in any bit position enables that register to be read.

Interrupt Control Byte

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
NO EFFECT	INTERRUPT BEFORE REQUESTING BUS	STATUS AFFECTS INTERRUPT VECTOR	INTERRUPT VECTOR FOLLOWS	PULSE COUNT FOLLOWS	PULSE GENERATED	INTERRUPT ON MATCH FOUND	INTERRUPT AT END OF BLOCK

A "1" in a bit position selects the option.

Timing Control Byte

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$\overline{\text{WR}}$ END	$\overline{\text{RD}}$ END	NOT USED	NOT USED	$\overline{\text{MREQ}}$ END	$\overline{\text{IORQ}}$ END	T_1	T_0

T_1	T_0	Cycle Length
0	0	4
0	1	3
1	0	2
1	1	1

A "0" in D_2, D_3, D_6, or D_7 will cause the corresponding control signal to end ½ clock time before the end of the cycle. Note: the total operation (Read and Write in Transfer or Read in Search) must be at least 2 cycles long.

Programming the DMA (continued)

Mask Byte

A zero in a given bit position will cause a compare to be performed between that bit position in the compare word register and the same bit position in the data being read.

Match Byte

Up to an 8-bit word to be compared to $D_0 - D_7$ during a read. See MASK BYTE.

Status Byte

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
NOT USED	NOT USED	END OF BLK	MATCH	INT. PENDING	NOT USED	READY ACTIVE	WRITE ADDRESS VALID

Pulse Count

This 8-bit word is loaded into a register. At the completion of each operation, the register is compared with the lower 8-bits of the byte counter. When it compares, the $\overline{\text{INT}}$ line is pulsed (but no interrupt is generated).

Interrupt Vector

This 8-bit byte is supplied to the CPU during Interrupt acknowledge if the DMA is the highest priority interrupting device.

If bit 5 of the Interrupt Control Byte (see p. 7) has been set and the DMA has been programmed to interrupt on a given status condition then D_1 and D_2 of the vector will be modified as follows:

Vector Bits	D_2	D_1	
	0	0	INT on RDY
	0	1	Match
	1	0	End of Blk
	1	1	Match, End of Blk

DMA Programming Example

The following example will show how the DMA may be programmed to transfer data from a peripheral (Port A) to memory (Port B). The table of bytes may be stored in memory and transferred to the DMA with an output instruction such as an OTIR.

Port A	Data Flow	Port B
Peripheral Address "X"5H	⟶ Block Length 1000H Bytes	Memory Starting Address 1050H

READY from the peripheral is active high
Memory address increments on each write

		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	HEX
1	Command Byte 1a Sets the DMA to receive Block length and Port A address and sets direction of transfer	0 Group 1	1 Blk Length Upper Follows	1 Blk Length Lower Follows	0 No Port A Upper Addr Follows	1 Port A Lower Addr Follows	1 A→B	0 1a Transfer No Search	1	6D
2	Port A Address Lower 8-bits	0	0	0	0	0	1	0	1	01
3	Block Length Lower 8-bits	0	0	0	0	0	0	0	0	00
4	Block Length Upper 8-bits	0	0	0	1	0	0	0	0	10
5	Command Byte 1b Defines Port A as peripheral with fixed addresses	0 Group 1	0 No Timing Follows	1 Fixed Addresses	X	1 Port is IO	0 This is Port "A"	0 1b	0	
6	Command Byte 1b - Defines Port B as a memory with incrementing addresses	0 Group 1	0 No Timing Follows	0 Address Changes	1 Address Increments	0 Port is Memory	1 This is Port "B"	0 1b	0	14
7	Command Byte 2b Sets mode to burst, sets DMA to expect Port B starting address	1 Group 2	1 Burst Mode	0	0 No Int Cont Byte Follows	1 Port B Upper Addr Follows	1 Port B Lower Addr Follows	0 2b	1	CD
8	Port B Address Lower 8-bits	0	1	0	1	0	0	0	0	50
9	Port B Address Upper 8-bits	0	0	0	1	0	0	0	0	10
10	Command Byte 2c Sets Ready Active High	1 Group 2	X	0 No Auto Restart	0 No wait States	1 Rdy Active High	X	1 2c	0	
11	Command Byte 2d loads starting addresses and resets block counter	1 Group 2	1	0	0 Load	1	1	1 2A	1	CF
12	Command Byte 2d Enables DMA to start operation	1 Group 2	0	0	0 ENABLE DMA	0	1	1 2d	1	87

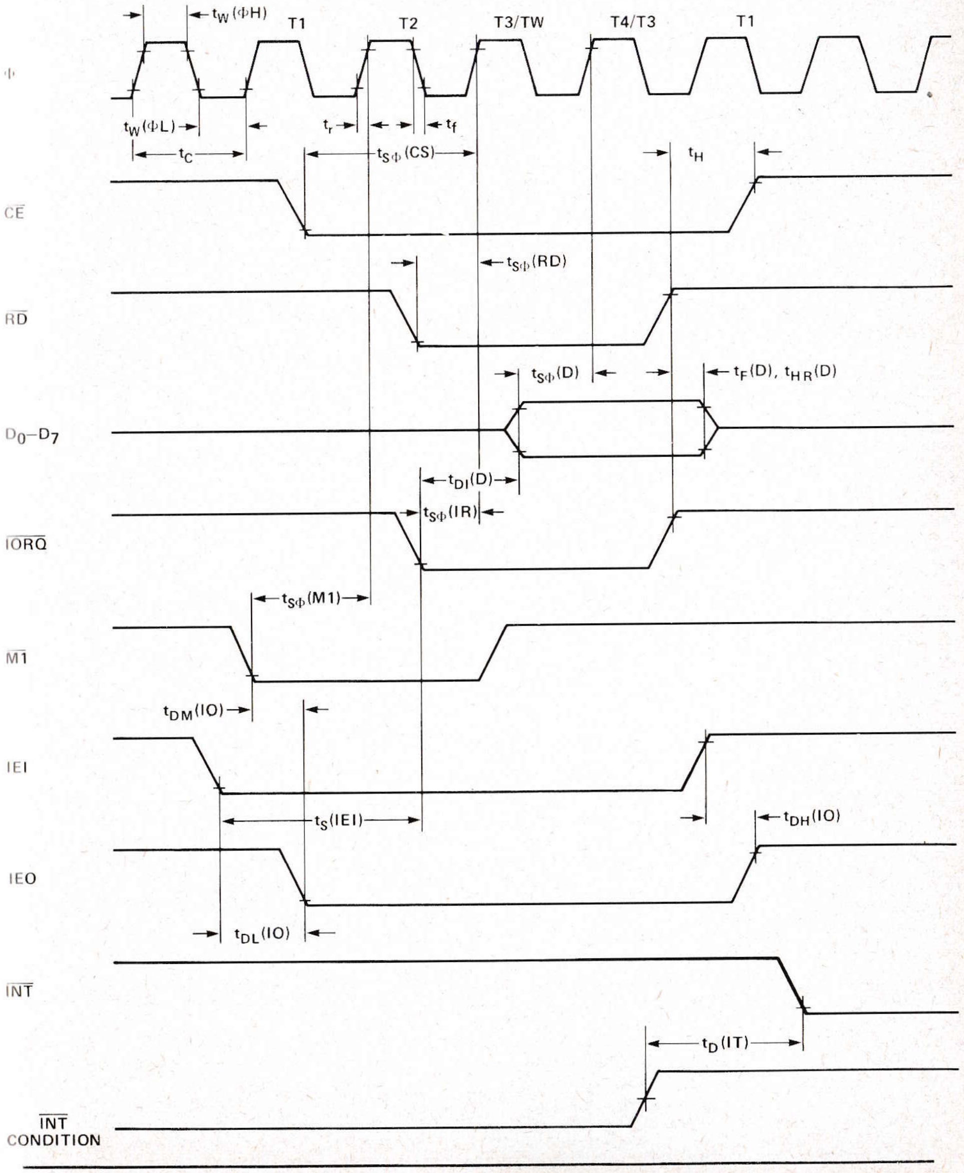
To reload the same addresses and block length for a subsequent operation, only two bytes are needed.

1. Command byte 2d — 1 1 0 0 1 1 1 1
   Reloads port addresses and block length — Load
2. Command byte 2d — 1 0 0 0 1 0 1 1
   Enables DMA — Enable DMA

Z80 and Z80A as a Peripheral Device (Inactive State)

Timing measurements are made at the following voltages, unless otherwise specified:

	"1"	"0"
CLOCK	4.2V	0.8V
OUTPUT	2 V	0.8V
INPUT	2 V	0.8V
FLOAT	ΔV	= +0.5V



Z80–DMA A.C. Characteristics

Z80-DMA as a Peripheral Device (Inactive State).
$T_A = 0°C$ to $70°C$, $Vcc = +5V \pm 5\%$, Unless Otherwise Noted

SIGNAL	SYMBOL	PARAMETER	MIN	MAX	UNIT	COMMENTS
Φ	t_c	Clock Period	400	[1]	nsec	
	$t_{w(\Phi H)}$	Clock Pulse Width, Clock High	170	2000	nsec	
	$t_{w(\Phi L)}$	Clock Pulse Width, Clock Low	170	2000	nsec	
	$t_{r,f}$	Clock Rise and Fall Times		30	nsec	
	t_H	Any Hold Time for Specified Setup Time	0		nsec	
$\overline{CE}$	$t_{S\Phi(CS)}$	Control Signal Setup Time to Rising Edge of Φ During Write Cycle	280		nsec	
D_{0-7}	$t_{DR(D)}$	Data Output Delay from Falling Edge of $\overline{RD}$		430	nsec	[2]
	$t_{S\Phi(D)}$	Data Setup Time to Rising Edge of Φ During Write or $\overline{M1}$ Cycle	50		nsec	$C_L = 50pF$
	$t_{DI(D)}$	Data Output Delay from Falling Edge of $\overline{IORQ}$ During INTA Cycle		340	nsec	[3]
	$t_{F(D)}$	Delay to Floating Bus (Output Buffer Disable Time)		160	nsec	
IEI	$t_{S(IEI)}$	IEI Setup Time to Falling Edge of $\overline{IORQ}$ During INTA Cycle	140		nsec	
IEO	$t_{DH(IO)}$	IEO Delay Time from Rising Edge of IEI		210	nsec	
	$t_{DL(IO)}$	IEO Delay Time from Falling Edge of IEI		190	nsec	
	$t_{DM(IO)}$	IEO Delay from Falling Edge of $\overline{M1}$ (Interrupt Occurring Just Prior to $\overline{M1}$) See Note A.		300	nsec	$C_L = 50pF$
$\overline{IORQ}$	$t_{S\Phi(IR)}$	$\overline{IORQ}$ Setup Time to Rising Edge of ΦDuring Write Cycle	250		nsec	
$\overline{M1}$	$t_{S\Phi(M1)}$	$\overline{M1}$ Setup Time to Rising Edge of Φ During INTA or $\overline{M1}$ Cycle. See Note B.	210		nsec	
$\overline{RD}$	$t_{S\Phi(RD)}$	$\overline{RD}$ Setup Time to Rising Edge of Φ During $\overline{M1}$ Cycle	240		nsec	
$\overline{INT}$	$t_{D(IT)}$	$\overline{INT}$ Delay Time from Condition Causing $\overline{INT}$. $\overline{INT}$ generated only when DMA is inactive.		500	nsec	
BAO	$t_{DH(BO)}$	BAO Delay from Rising Edge of BAI	150	200	nsec	
	$t_{DL(BO)}$	BAO Delay from Falling Edge of BAI	150	200	nsec	

[1] $t_c = t_{w(\Phi H)} + t_{w(\Phi L)} + t_r + t_f$

[2] Increase $t_{DR(D)}$ by 10 nsec for each 50pF increase in loading up to 200pF max.

[3] Increase $t_{DI(D)}$ by 10 nsec for each 50pF increase in loading up to 200pF max.

Z80A–DMA A.C. Characteristics

Z80A-DMA as a Peripheral Device (Inactive State).
$T_A = 0°C$ to $70°C$, $Vcc = +5V \pm 5\%$, Unless Otherwise Noted

SIGNAL	SYMBOL	PARAMETER	MIN	MAX	UNIT	COMMENTS
Φ	t_c	Clock Period	250	[1]	nsec	
	$t_{w(\Phi H)}$	Clock Pulse Width, Clock High	105	2000	nsec	
	$t_{w(\Phi L)}$	Clock Pulse Width, Clock Low	105	2000	nsec	
	$t_{r,f}$	Clock Rise and Fall Times		30	nsec	
	t_H	Any Hold Time for Specified Setup Time	0		nsec	
$\overline{CE}$	$t_{S\Phi(CS)}$	Control Signal Setup Time to Rising Edge of Φ During Write Cycle	145		nsec	
$D_{0\ 7}$	$t_{DR(D)}$	Data Output Delay from Falling Edge of $\overline{RD}$		380	nsec	[2]
	$t_{S\Phi(D)}$	Data Setup Time to Rising Edge of Φ During Write or $\overline{M1}$ Cycle	50		nsec	$C_L = 50pF$
	$t_{DI(D)}$	Data Output Delay from Falling Edge of $\overline{IORQ}$ During INTA Cycle		250	nsec	[3]
	$t_{F(D)}$	Delay to Floating Bus (Output Buffer Disable Time)		110	nsec	
IEI	$t_{S(IEI)}$	IEI Setup Time to Falling Edge of $\overline{IORQ}$ During INTA Cycle	140		nsec	
IEO	$t_{DH(IO)}$	IEO Delay Time from Rising Edge of IEI		160	nsec	
	$t_{DL(IO)}$	IEO Delay Time from Falling Edge of IEI		130	nsec	
	$t_{DM(IO)}$	IEO Delay from Falling Edge of $\overline{M1}$ (Interrupt Occurring Just Prior to $\overline{M1}$) See Note A.		190	nsec	$C_L = 50pF$
$\overline{IORQ}$	$t_{S\Phi(IR)}$	$\overline{IORQ}$ Setup Time to Rising Edge of Φ During Write Cycle	115		nsec	
$\overline{M1}$	$t_{S\Phi(M1)}$	$\overline{M1}$ Setup Time to Rising Edge of Φ During INTA or $\overline{M1}$ Cycle. See Note B.	90		nsec	
$\overline{RD}$	$t_{S\Phi(RD)}$	$\overline{RD}$ Setup Time to Rising Edge of Φ During $\overline{M1}$ Cycle	115		nsec	
$\overline{INT}$	$t_{D(IT)}$	$\overline{INT}$ Delay Time from Condition Causing $\overline{INT}$. $\overline{INT}$ generated only when DMA is inactive.		500	nsec	
BAO	$t_{DH(BO)}$	BAO Delay from Rising Edge of BAI	150	200	nsec	
	$t_{DL(BO)}$	BAO Delay from Falling Edge of BAI	150	200	nsec	

1] $t_c = t_{w(\Phi H)} + t_{w(\Phi L)} + t_r + t_f$

[2] Increase $t_{DR(D)}$ by 10 nsec for each 50pF increase in loading up to 200pF max.

[3] Increase $t_{DI(D)}$ by 10 nsec for each 50pF increase in loading up to 200pF max.

A. $2.5\ t_c > (N-2)\ t_{DL(IO)} + t_{DM(IO)} + t_{S(IEI)}$ + TTL Buffer Delay, if any

Z80–DMA A.C. Characteristics

Z80-DMA as a Bus Controller (Active State)
$T_A = 0°C$ to $70°C$. $Vcc = +5V \pm 5\%$, Unless Otherwise Noted.

SIGNAL	SYMBOL	PARAMETER	MIN	MAX	UNIT	COMMENTS
Φ	t_c	Clock Period	.4	[12]	μsec	
	$t_{w(\Phi H)}$	Clock Pulse Width, Clock High	180	2000	nsec	
	$t_{w(\Phi L)}$	Clock Pulse Width, Clock Low	180	2000	nsec	
	$t_{r,f}$	Clock Rise and Fall Time		30	nsec	
A_{0-15}	$t_{D(AD)}$	Address Output Delay		145	nsec	
	$t_{F(AD)}$	Delay to Float		110	nsec	$C_L = 50pF$
	t_{acm}	Address Stable Prior to $\overline{MREQ}$ (Memory Cycle)	[1]		nsec	D
	t_{aci}	Address Stable Prior to $\overline{IORQ}$, $\overline{RD}$ or $\overline{WR}$ (I/O Cycle)	[2]		nsec	D
	t_{ca}	Address Stable from $\overline{RD}$ or $\overline{WR}$	[3]		nsec	D
	t_{caf}	Address Stable From $\overline{RD}$ or $\overline{WR}$ During Float	[4]		nsec	D
D_{0-7}	$t_{D(D)}$	Data Output Delay		260	nsec	
	$t_{F(D)}$	Delay to Float During Write Cycle		90	nsec	
	$t_{S\Phi(D)}$	Data Setup Time to Rising Edge of Clock During Read When Rising Edge Ends RD	50		nsec	$C_L = 200pF$
	$t_{S\overline{\Phi}(D)}$	Data Setup Time to Falling Edge of Clock During Read When Falling Edge Ends $\overline{RD}$	60		nsec	
	t_{dcm}	Data Stable Prior to $\overline{WR}$ (Memory Cycle)	[5]		nsec	D
	t_{dci}	Data Stable Prior to $\overline{WR}$ (I/O Cycle)	[6]		nsec	D
	t_{cdf}	Data Stable From $\overline{WR}$	[7]		nsec	D
	t_H	Any Hold Time for Setup Time	0		nsec	
$\overline{MREQ}$	$t_{DL\overline{\Phi}(MR)}$	$\overline{MREQ}$ Delay from Falling Edge of Clock, $\overline{MREQ}$ Low		100	nsec	
	$t_{DH\Phi(MR)}$	$\overline{MREQ}$ Delay from Rising Edge of Clock, $\overline{MREQ}$ High		100	nsec	
	$t_{DH\overline{\Phi}(MR)}$	$\overline{MREQ}$ Delay from Falling Edge of Clock, $\overline{MREQ}$ High		100	nsec	
	$t_{DL\Phi(MR)}$	$\overline{MREQ}$ Delay from Falling Edge of Clock, $\overline{MREQ}$ Low		100	nsec	$C_L = 50pF$
	$t_{w(\overline{MRL})}$	Pulse Width, $\overline{MREQ}$ Low	[8]		nsec	D
	$t_{w(\overline{MRH})}$	Pulse Width, $\overline{MREQ}$ High	[9]		nsec	D
$\overline{IORQ}$	$t_{DL\Phi(IR)}$	$\overline{IORQ}$ Delay from Rising Edge of Clock, $\overline{IORQ}$ Low		90	nsec	
	$t_{DL\overline{\Phi}(IR)}$	$\overline{IORQ}$ Delay from Falling Edge of Clock, $\overline{IORQ}$ Low		110	nsec	
	$t_{DH\Phi(IR)}$	$\overline{IORQ}$ Delay from Rising Edge of Clock, $\overline{IORQ}$ High		100	nsec	$C_L = 50pF$
	$t_{DH\overline{\Phi}(IR)}$	$\overline{IORQ}$ Delay from Falling Edge of Clock, $\overline{IORQ}$ High		110	nsec	
$\overline{RD}$	$t_{DL\Phi(RD)}$	$\overline{RD}$ Delay from Rising Edge of Clock, $\overline{RD}$ Low		100	nsec	
	$t_{DL\overline{\Phi}(RD)}$	$\overline{RD}$ Delay from Falling Edge of Clock, $\overline{RD}$ Low		130	nsec	
	$t_{DH\Phi(RD)}$	$\overline{RD}$ Delay from Rising Edge of Clock, $\overline{RD}$ High		100	nsec	$C_L = 50pF$
	$t_{DH\overline{\Phi}(RD)}$	$\overline{RD}$ Delay from Falling Edge of Clock, $\overline{RD}$ High		110	nsec	
$\overline{WR}$	$t_{DL\Phi(WR)}$	$\overline{WR}$ Delay from Rising Edge of Clock, $\overline{WR}$ Low		80	nsec	
	$t_{DL\overline{\Phi}(WR)}$	$\overline{WR}$ Delay from Falling Edge of Clock, $\overline{WR}$ Low		90	nsec	
	$t_{DH\overline{\Phi}(WR)}$	$\overline{WR}$ Delay from Falling Edge of Clock, $\overline{WR}$ High		100	nsec	$C_L = 50pF$
	$t_{DH\Phi(WR)}$	$\overline{WR}$ Delay from Rising Edge of Clock , $\overline{WR}$ High		100	nsec	
	$t_{w(\overline{WRL})}$	Pulse Width, $\overline{WR}$ Low	[10]		nsec	
$\overline{WAIT}$	$t_{s(WT)}$	$\overline{WAIT}$ Setup Time to Falling Edge of Clock	70		nsec	
$\overline{BUSRQ}$	$t_{D(BQ)}$	$\overline{BUSRQ}$ Delay Time from Rising Edge of Clock	100		nsec	
	$t_{F(C)}$	Delay to Float ($\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$ and $\overline{WR}$)		100	nsec	

NOTES:
A. Data should be enabled onto the DMA data bus when $\overline{RD}$ is active.
B. All control signals are internally synchronized, so they may be totally asynchronous with respect to the clock.
C. Output Delay vs. Loaded Capacitance
TA = 70°C Vcc = +5V±5%
(1) $\Delta C_L = +100pF (A_{\emptyset} - A_{15}$ and Control Signals), add 30 nsec to timing shown.
D. During Standard CPU Timing

[1] $t_{acm} = t_{w(\Phi H)} + t_f - 75$

[2] $t_{aci} = t_c - 80$

[3] $t_{ca} = t_{w(\Phi L)} + t_r - 40$

[4] $t_{caf} = t_{w(\Phi L)} + t_r - 60$

[5] $t_{dcm} = t_c - 180$

[6] $t_{dci} = t_{w(\Phi L)} + t_r - 180$

[7] $t_{cdf} = t_{w(\Phi L)} + t_r - 50$

[8] $t_{w(MRL)} = t_c - 40$

[9] $t_{w(MRH)} = t_c - 40$ Std. CPU Timing
$t_{w(MRH)} = t_{w(\Phi H)} + t_f - 30$ Variable 1 Cycle.

[10] $t_{w(WR)} = t_c - 40$ Std. CPU Timing
$t_{w(WR)} = t_{w(\Phi H)} + t_f - 30$ Variable 1 Cycle.

[12] $t_c = t_{w(\Phi H)} + t_{w(\Phi L)} + t_r + t_f$

Z80A–DMA A,C. Characteristics

Z80A-DMA as a Bus Controller (Active State)
$T_A = 0°C$ to $70°C$, Vcc = +5V±5%, Unless Otherwise Noted.

SIGNAL	SYMBOL	PARAMETER	MIN	MAX	UNIT	COMMENTS
Φ	t_c	Clock Period	.25	[12]	μsec	
	$t_{w(\Phi H)}$	Clock Pulse Width, Clock High	110	2000	nsec	
	$t_{w(\Phi L)}$	Clock Pulse Width, Clock Low	110	2000	nsec	
	$t_{r,\ f}$	Clock Rise and Fall Time		30	nsec	
A_{0-15}	$t_{D(AD)}$	Address Output Delay		110	nsec	
	$t_{F(AD)}$	Delay to Float		90	nsec	C_L = 50pF
	t_{acm}	Address Stable Prior to $\overline{\text{MREQ}}$ (Memory Cycle)	[1]		nsec	D
	t_{aci}	Address Stable Prior to $\overline{\text{IORQ}}$, $\overline{\text{RD}}$ or $\overline{\text{WR}}$ (I/O Cycle)	[2]		nsec	D
	t_{ca}	Address Stable from $\overline{\text{RD}}$ or $\overline{\text{WR}}$	[3]		nsec	D
	t_{caf}	Address Stable From $\overline{\text{RD}}$ or $\overline{\text{WR}}$ During Float	[4]		nsec	D
D_{0-7}	$t_{D(D)}$	Data Output Delay		180	nsec	
	$t_{F(D)}$	Delay to Float During Write Cycle			nsec	
	$t_{S\Phi(D)}$	Data Setup Time to Rising Edge of Clock During Read When Rising Edge Ends RD	35		nsec	C_L = 200pF
	$t_{S\overline{\Phi}(D)}$	Data Setup Time to Falling Edge of Clock During Read When Falling Edge Ends $\overline{\text{RD}}$	50		nsec	
	t_{dcm}	Data Stable Prior to $\overline{\text{WR}}$ (Memory Cycle)	[5]		nsec	D
	t_{dci}	Data Stable Prior to $\overline{\text{WR}}$ (I/O Cycle)	[6]		nsec	D
	t_{cdf}	Data Stable From $\overline{\text{WR}}$	[7]		nsec	D
	t_H	Any Hold Time for Setup Time		0	nsec	
$\overline{\text{MREQ}}$	$t_{DL\overline{\Phi}(MR)}$	$\overline{\text{MREQ}}$ Delay from Falling Edge of Clock, $\overline{\text{MREQ}}$ Low		75	nsec	
	$t_{DH\Phi(MR)}$	$\overline{\text{MREQ}}$ Delay from Rising Edge of Clock, $\overline{\text{MREQ}}$ High		75	nsec	
	$t_{DH\overline{\Phi}(MR)}$	$\overline{\text{MREQ}}$ Delay from Falling Edge of Clock, $\overline{\text{MREQ}}$ High		75	nsec	
	$t_{DL\Phi(MR)}$	$\overline{\text{MREQ}}$ Delay from Falling Edge of Clock, $\overline{\text{MREQ}}$ Low		80	nsec	C_L = 50pF
	$t_{w(\overline{MRL})}$	Pulse Width, $\overline{\text{MREQ}}$ Low	[8]		nsec	D
	$t_{w(\overline{MRH})}$	Pulse Width, $\overline{\text{MREQ}}$ High	[9]		nsec	D
$\overline{\text{IORQ}}$	$t_{DL\Phi(IR)}$	$\overline{\text{IORQ}}$ Delay from Rising Edge of Clock, $\overline{\text{IORQ}}$ Low		75	nsec	
	$t_{DL\overline{\Phi}(IR)}$	$\overline{\text{IORQ}}$ Delay from Falling Edge of Clock, $\overline{\text{IORQ}}$ Low		80	nsec	
	$t_{DH\Phi(IR)}$	$\overline{\text{IORQ}}$ Delay from Rising Edge of Clock, $\overline{\text{IORQ}}$ High		80	nsec	C_L = 50pF
	$t_{DH\overline{\Phi}(IR)}$	$\overline{\text{IORQ}}$ Delay from Falling Edge of Clock, $\overline{\text{IORQ}}$ High		80	nsec	
$\overline{\text{RD}}$	$t_{DL\Phi(RD)}$	$\overline{\text{RD}}$ Delay from Rising Edge of Clock, $\overline{\text{RD}}$ Low		75	nsec	
	$t_{DL\overline{\Phi}(RD)}$	$\overline{\text{RD}}$ Delay from Falling Edge of Clock, $\overline{\text{RD}}$ Low		95	nsec	
	$t_{DH\Phi(RD)}$	$\overline{\text{RD}}$ Delay from Rising Edge of Clock, $\overline{\text{RD}}$ High		75	nsec	C_L = 50pF
	$t_{DH\overline{\Phi}(RD)}$	$\overline{\text{RD}}$ Delay from Falling Edge of Clock, $\overline{\text{RD}}$ High		80	nsec	
$\overline{\text{WR}}$	$t_{DL\Phi(WR)}$	$\overline{\text{WR}}$ Delay from Rising Edge of Clock, $\overline{\text{WR}}$ Low		60	nsec	
	$t_{DL\overline{\Phi}(WR)}$	$\overline{\text{WR}}$ Delay from Falling Edge of Clock, $\overline{\text{WR}}$ Low		80	nsec	
	$t_{DH\overline{\Phi}(WR)}$	$\overline{\text{WR}}$ Delay from Falling Edge of Clock, $\overline{\text{WR}}$ High		80	nsec	C_L = 50pF
	$t_{DH\Phi(WR)}$	$\overline{\text{WR}}$ Delay from Rising Edge of Clock, $\overline{\text{WR}}$ High		80	nsec	
	$t_{w(\overline{WRL})}$	Pulse Width, $\overline{\text{WR}}$ Low	[10]		nsec	
$\overline{\text{WAIT}}$	$t_{s(WT)}$	$\overline{\text{WAIT}}$ Setup Time to Falling Edge of Clock	70		nsec	
$\overline{\text{BUSRQ}}$	$t_{D(BQ)}$	$\overline{\text{BUSRQ}}$ Delay Time from Rising Edge of Clock	100		nsec	
	$t_{F(C)}$	Delay to Float ($\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, $\overline{\text{RD}}$ and $\overline{\text{WR}}$)		80	nsec	

NOTES:
A. Data should be enabled onto the DMA data bus when $\overline{\text{RD}}$ is active.
B. All control signals are internally synchronized, so they may be totally asynchronous with respect to the clock.
C. Output Delay vs. Loaded Capacitance
TA = 70°C Vcc = +5V±5%
(1) ΔC_L = +100pF($A_{\emptyset}-A_{15}$and Control Signals), add 30 nsec to timing shown.
D. During Standard CPU Timing

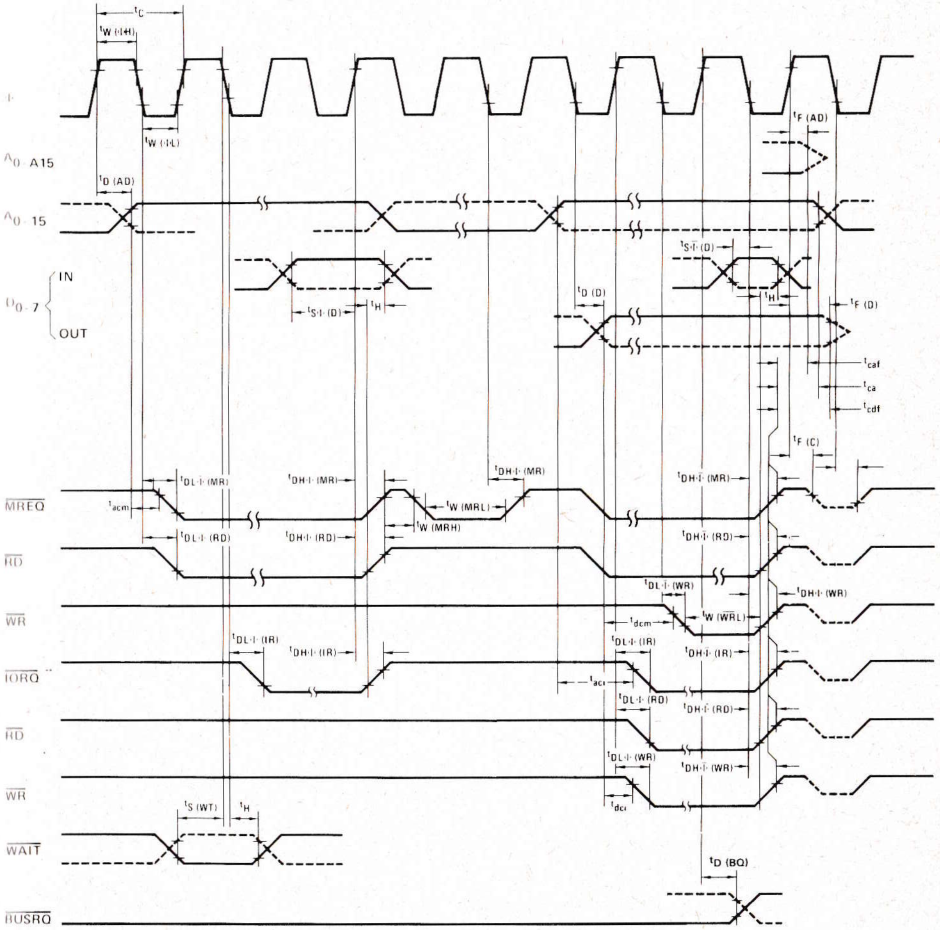
[1] $t_{acm} = t_{w(\Phi H)}+t_f-75$
[2] $t_{aci} = t_c-80$
[3] $t_{ca} = t_{w(\Phi L)}+t_r-40$
[4] $t_{caf} = t_{w(\Phi L)}+t_r-60$
[5] $t_{dcm} = t_c-180$
[6] $t_{dci} = t_{w(\Phi L)}+t_r-180$
[7] $t_{cdf} = t_{w(\Phi L)}+t_r-50$
[8] $t_{w(MRL)} = t_c-40$
[9] $t_{w(MRH)} = t_c-40$ Std. CPU Timing
$t_{w(MRH)} = t_{w(\Phi H)}+t_f-30$ Variable 1 Cycle.
[10] $t_{w(WR)} = t_c-40$ Std. CPU Timing
$t_{w(WR)} = t_{w(\Phi H)}+t_f-30$ Variable 1 Cycle.
[12] $t_c = t_{w(\Phi H)}+t_{w(\Phi L)}+t_r+t_f$

A.C. Timing Diagrams

Z80 and Z80A as a Bus Controller (Active State)

Timing measurements are made at the following voltages, unless otherwise specified:

	"1"	"0"
CLOCK	4.2V	0.8V
OUTPUT	2 V	0.8V
INPUT	2 V	0.8V
FLOAT	ΔV	= +0.5V



Absolute Maximum Ratings

Temperature Under Bias	Specified operating range
Storage Temperature	-65°C to +150°C
Voltage On Any Pin with Respect to Ground	-0.3V to +7V
Power Dissipation	1.5W

Note: All AC and DC characteristics remain the same for the military grade parts except I_{cc}.
I_{cc} = 200 mA

* Comment
Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Z80–DMA D.C. Characteristics

T_A = 0°C to 70°C, V_{CC} = 5V ±5% unless otherwise specified

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	-0.3		0.45	V	
V_{IHC}	Clock Input High Voltage	V_{CC}-0.6		V_{cc}+0.3	V	
V_{IL}	Input Low Voltage	-0.3		0.8	V	
V_{IH}	Input High Voltage	2		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = 2 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = -250 μA
V_{CC}	Power Supply Current			150	mA	t_C = 400 nsec
I_{LI}	Input Leakage Current			10	μA	V_{IN} = 0 to V_{CC}
I_{LOH}	Tri-State Output Leakage Current in Float			10	μA	V_{OUT} = 2.4 to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float			-10	μA	V_{OUT} = 0.4V
I_{LD}	Data Bus Leakage Current in Input Mode			±10	μA	0 < V_{IN} < V_{CC}

Z80A–DMA D.C. Characteristics

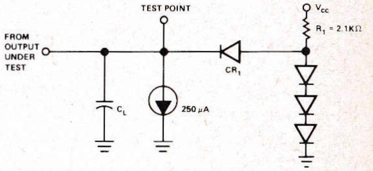
T_A = 0°C to 70°C, V_{CC} = 5V ±5% unless otherwise specified

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	-0.3		0.45	V	
V_{IHC}	Clock Input High Voltage	V_{CC}-0.6		V_{cc}+0.3	V	
V_{IL}	Input Low Voltage	-0.3		0.8	V	
V_{IH}	Input High Voltage	2		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = 2 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = -250 μA
V_{CC}	Power Supply Current		90	200	mA	t_c = 250nsec
I_{LI}	Input Leakage Current			10	μA	V_{IN} = 0 to V_{CC}
I_{LOH}	Tri-State Output Leakage Current in Float			10	μA	V_{OUT} = 2.4 to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float			-10	μA	V_{OUT} = 0.4V
I_{LD}	Data Bus Leakage Current in Input Mode			±10	μA	0 < V_{IN} < V_{CC}

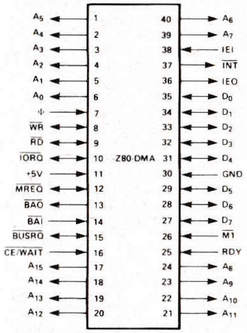
Capacitance

T_A = 25°C, f = 1 MHz

Symbol	Parameter	Max.	Unit	Test Condition
C_Φ	Clock Capacitance	35	pF	Unmeasured Pins
C_{IN}	Input Capacitance	5	pF	Returned to Ground
C_{OUT}	Output Capacitance	10	pF	



Package Configuration

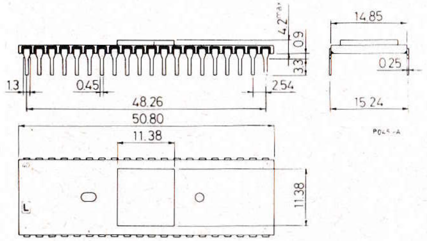


ORDERING NUMBERS:

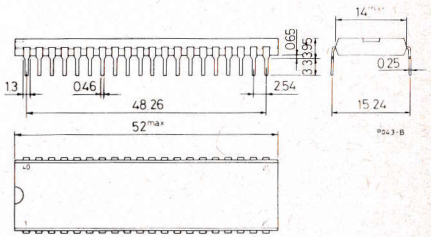
Z80–DMA	D1	for dual in–line ceramic slam package
Z80–DMA	B1	for dual in–line plastic package
Z80A–DMA	D1	for dual in–line ceramic slam package
Z80A–DMA	B1	for dual in–line plastic package

MECHANICAL DATA (dimensions in mm)

40–PIN CERAMIC DUAL IN–LINE SLAM PACKAGE



40–PIN PLASTIC DUAL IN–LINE PACKAGE



Z 80 - SIO

Product Specification

The SGS-ATES Z80 product line is a complete set of microcomputer components, development systems and support software. The Z80 microcomputer component set includes all of the circuits necessary to build high-performance microcomputer systems with virtually no other logic and a minimum number of low cost standard memory elements.

The Z80-SIO (Serial Input/Output) circuit is a programmable, dual-channel device which provides formatting of data for serial data communication. It is capable of handling asynchronous, synchronous and synchronous bit oriented protocols such as IBM BiSync, HDLC, SDLC and virtually any other serial protocol. It can generate CRC codes in any synchronous mode and can be programmed by the CPU for any traditional asynchronous format.

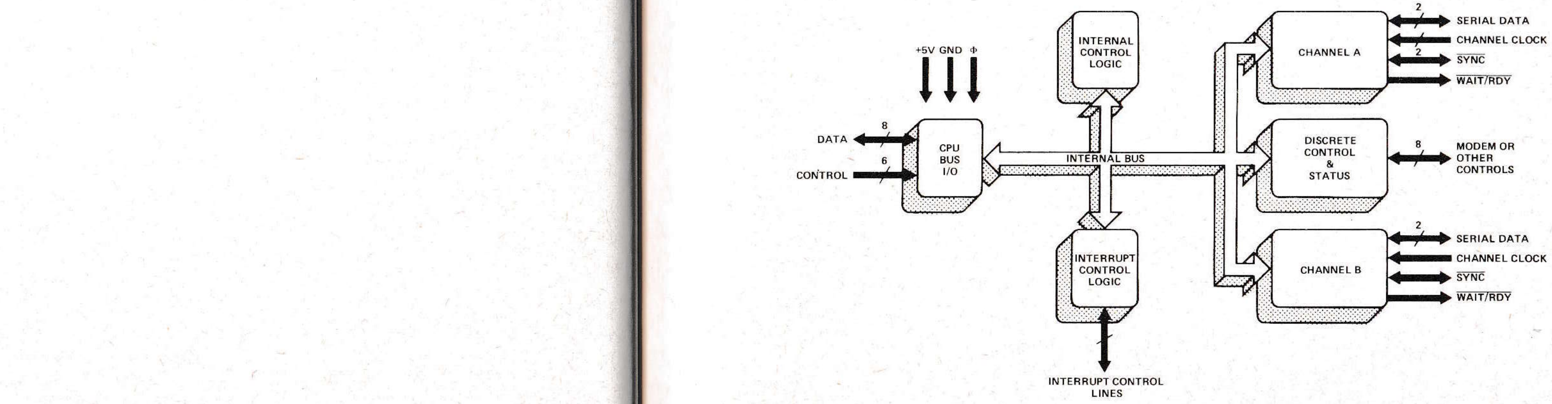
Structure

- N-channel Silicon Gate Depletion Load Technology
- 40 Pin DIP
- Single 5 volt power supply
- Single phase 5 volt clock
- Two Full Duplex channels

Features

- Two independent full duplex channels
- Data rates – 0 to 550K bits/second
- Receiver data registers quadruply buffered; transmitter doubly buffered.
- Asynchronous operation
  - 5, 6, 7 or 8 bits/character
  - 1, 1½ or 2 stop bits
  - Even, odd or no parity
  - x1, x16, x32 and x 64 clock modes
  - Break generation and detection
  - Parity, Overrun and Framing error detection
- Binary Synchronous operation
  - Internal or external character synchronozation
  - One or two Sync characters in separate registers
  - Automatic Sync Character Insertion
  - CRC generation and checking
- HDLC or IBM SDLC operation
  - Automatic Zero insertion and deletion
  - Automatic Flag insertion
  - Address field recognition
  - I-field residue handling
  - Valid receive messages protected from overrun
  - CRC generation and checking
- Eight modem control inputs and outputs
- Both CRC-16 and CRC-CCITT (-0 and -1) are implemented
- Daisy chain priority interrupt logic included to provide for automatic interrupt vectoring without external logic.
- All inputs and outputs fully TTL compatible.

Fig. 8 - SIO BLOCK DIAGRAM

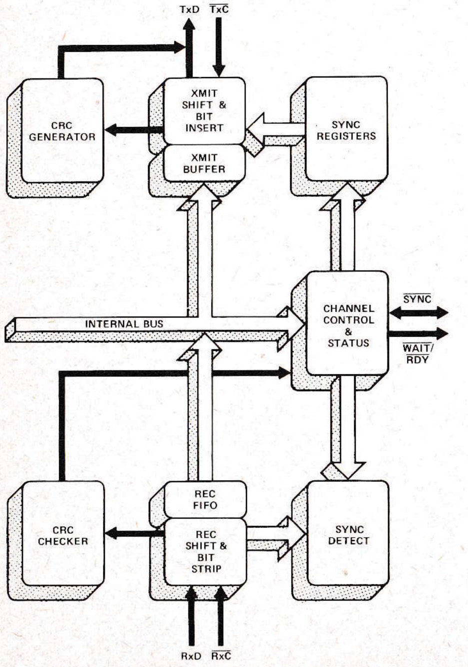


SIO Architecture

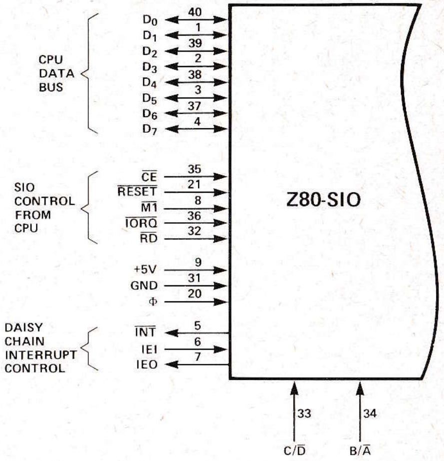
A block diagram of the SIO is shown in Figure 8. The internal structure includes a Z80-CPU bus interface, internal control and interrupt logic and two full duplex channels. The interrupt control logic determines which channel and which device within the channel is the highest priority for purposes of the automatic interrupt vectoring. Priority is fixed with Channel A assigned higher priority than Channel B and the Receiver, Transmitter and External/Status assigned priority in that order within each channel.

The channel logic is shown in block form in Figure 9. Each channel has five 8-bit control registers, two 8-bit status registers and two 8-bit sync character registers. The interrupt vector is written into an additional 8-bit register in Channel B and may also be read thru that channel. The receiver has three 8-bit buffer registers in FIFO arrangement in addition to the 8-bit input shift register. The transmitter has one 8-bit buffer register in addition to the 8-bit output shift register. The CRC generator/checkers are 16-bit shift registers with appropriate internal feedback (programmable) for two different CRC codes.

Fig. 9 - CHANNEL BLOCK DIAGRAM



Pin Description



CPU DATA BUS
D0 40
D1 1
D2 39
D3 2
D4 38
D5 3
D6 37
D7 4
SIO CONTROL FROM CPU
$\overline{\text{CE}}$ 35
RESET 21
$\overline{\text{M1}}$ 8
$\overline{\text{IORQ}}$ 36
$\overline{\text{RD}}$ 32
+5V 9
GND 31
Φ 20
DAISY CHAIN INTERRUPT CONTROL
$\overline{\text{INT}}$ 5
IEI 6
IEO 7
Z80-SIO
33 C/$\overline{\text{D}}$
34 B/$\overline{\text{A}}$

D_0-D_7 System Data Bus (bidirectional, tri-state)

B/$\overline{\text{A}}$ Channel B or A select (input high is Channel B)

C/$\overline{\text{D}}$ Control or Data select (input high is control)

$\overline{\text{CE}}$ Chip Enable (input, active low)

$\overline{\text{M1}}$ Machine Cycle One Signal from Z80-CPU (input, active low)

$\overline{\text{IORQ}}$ Input/Output request from Z80-CPU (input, active low)

$\overline{\text{RD}}$ Read Cycle Status from the Z80-CPU (input, active low)

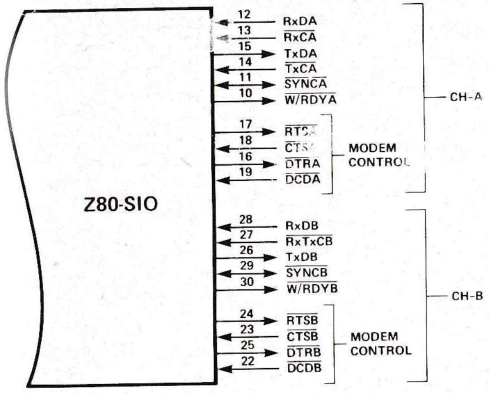
Φ System Clock (input)

$\overline{\text{RESET}}$ Reset (input, active low) disables both receivers and transmitters. TxDA and TxDB are forced marking. Modem controls are forced high. Control registers must be rewritten after SIO is reset and before any data is transmitted or received. All interrupts are disabled.

IEI Interrupt Enable In (input, active high)

IEO Interrupt Enable Out (output, active high) IEI and IEO form a daisy-chain connection for priority interrupt control.

$\overline{\text{INT}}$ Interrupt Request (output, open drain, active low).



Z80-SIO
12 RxDA
13 $\overline{\text{RxCA}}$
15 TxDA
14 $\overline{\text{TxCA}}$
11 $\overline{\text{SYNCA}}$
10 $\overline{\text{W/RDYA}}$
17 $\overline{\text{RTSA}}$
18 $\overline{\text{CTSA}}$
16 $\overline{\text{DTRA}}$
19 $\overline{\text{DCDA}}$
MODEM CONTROL
CH-A
28 RxDB
27 $\overline{\text{RxTxCB}}$
26 TxDB
29 $\overline{\text{SYNCB}}$
30 $\overline{\text{W/RDYB}}$
24 $\overline{\text{RTSB}}$
23 $\overline{\text{CTSB}}$
25 $\overline{\text{DTRB}}$
22 $\overline{\text{DCDB}}$
MODEM CONTROL
CH-B

$\overline{\text{WAIT/READY}}$ A
$\overline{\text{WAIT/READY}}$ B Two pins, one for each channel. They may be programmed to serve as ready lines for use with a DMA Controller or they may serve as wait lines to synchronize the Z80-CPU to the SIO data rate.

$\overline{\text{CTSA}}$, $\overline{\text{CTSB}}$ Clear to Send (2 pins, inputs, active low). When programmed as "auto enables," these inputs inhibit the transmitters of their respective channels. If these'pins are not programmed as transmitter enables, they may be programmed as general-purpose input pins. These inputs are Schmitt-trigger buffered to allow slow-risetim e inputs.

$\overline{\text{DCDA}}$, $\overline{\text{DCDB}}$ Data Carrier Detect (2 pins, inputs, active low.) These pins are similar to the $\overline{\text{CTS}}$ inputs, except that they are usable as receiver inhibits rather than transmitter inhibits.

RxDA, RxDB Receive Data. (2 pins, inputs, active high.)

TxDA, TxDB Transmit Data. (2 pins, outputs, active high.)

$\overline{\text{RxCA}}$, $\overline{\text{RxCB}}$ Receiver Clocks (inputs, active low.) (Two pads, one per channel. See note on Bonding Option.) Clock may be x1, x16, x32 or x64 the data rate in asynchronous modes.

$\overline{\text{TxCA}}$, $\overline{\text{TxCB}}$ Transmitter Clocks (inputs, active high.) (Two pads, one per channel. See note on Bonding Option.) May be x1, x16, x32 or x64 baud rate, but same multiplier must be observed as for receiver. The TxC and RxC inputs are Schmitt-trigger buffered, for relaxed rise and fall time requirements.

$\overline{\text{RTSA}}$, $\overline{\text{RTSB}}$ Request to Send (2 pins, outputs, active low.) When the $\overline{\text{RTS}}$ bit is set, the $\overline{\text{RTS}}$ pin goes low. When the bit is reset in asynchronous mode, the pin goes high, but only after the transmitter is empty. In synchronous modes, $\overline{\text{RTS}}$ is a simple output which strictly follows the state of the $\overline{\text{RTS}}$ bit.

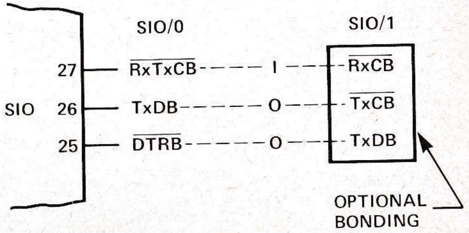
$\overline{\text{DTRA}}$, $\overline{\text{DTRB}}$ Data Terminal Ready (2 pins, output, active low.) Pin follows state programmed with DTR bit. (Two pads, one per channel. See note on Bonding Option.)

$\overline{\text{SYNCA}}$, $\overline{\text{SYNCB}}$ External Character Synchronization (2 pins, input/output, active low.) If the External Synchronization mode is selected, assembly of characters will begin on the next rising edge of $\overline{\text{RxC}}$. If internal character sync modes are selected, the pins are outputs that are active during part of the clock cycles that a sync character is recognized. The sync condition is not latched, so this pin will be active every time a sync pattern is recognized, regardless of character boundaries. In asynchronous modes, these pins are simple inputs to the Hunt/Sync bits in Status Register 0 and may be used for any input function desired.

NOTE: When used as an external synchronization pin, it must not become active for three system clock cycles after the previous rising edge of $\overline{\text{RxC}}$. This requirement normally can be met by allowing $\overline{\text{SYNC}}$ to change only on the falling edge of $\overline{\text{RxC}}$.

Note on Bonding Option:

Due to package constraints, there are only two pins available for the three signals, $\overline{\text{TxCB}}$, $\overline{\text{RxCB}}$ and $\overline{\text{DTRB}}$. They are normally bonded so that $\overline{\text{TxCB}}$ and $\overline{\text{RxCB}}$ are one pin, and $\overline{\text{RxTxCB}}$ and $\overline{\text{DTRB}}$ is an available output. If there is a requirement for different clock rates or phases for $\overline{\text{RxCB}}$ and $\overline{\text{TxCB}}$, they may be bonded independently by sacrificing $\overline{\text{DTRB}}$.

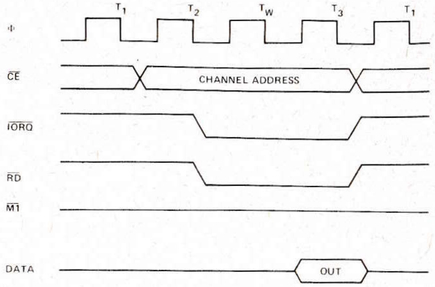


SIO/0
SIO/1
SIO
27 $\overline{\text{RxTxCB}}$ I $\overline{\text{RxCB}}$
26 TxDB O $\overline{\text{TxCB}}$
25 $\overline{\text{DTRB}}$ O TxDB
OPTIONAL BONDING

Timing Waveforms

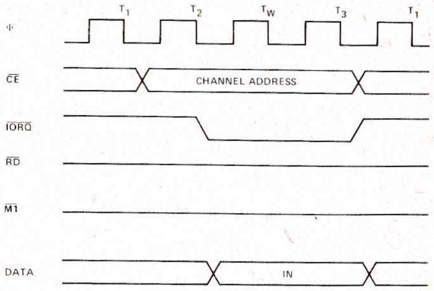
WRITE CYCLE

Illustrated here is the timing associated with a data or control byte being written into the SIO. Z80 Output Instructions satisfy this timing.



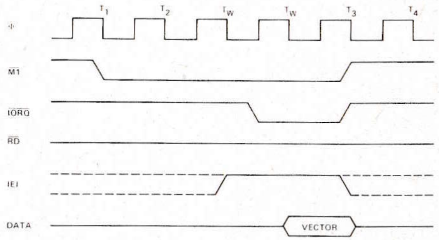
READ CYCLE

The timing associated with reading data or a status register within the SIO is illustrated here. Z80 Input instructions satisfy this timing.



INTERRUPT ACKNOWLEDGE CYCLE

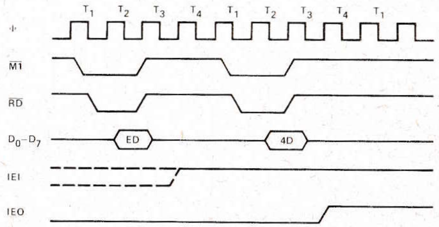
Some time after an interrupt is requested by the SIO, the CPU will send out an interrupt acknowledge ($\overline{M1}$ and $\overline{IORQ}$.) During this time, the interrupt logic of the SIO will determine the highest priority function which is requesting an interrupt. To insure that the daisy chain enable lines stabilize, channels are inhibited from changing their interrupt request status when $\overline{M1}$ is active (low). If the SIO is the highest priority device requesting an interrupt, the SIO will place the appropriate interrupt vector on the data bus when $\overline{IORQ}$ goes active.



RETURN FROM INTERRUPT CYCLE

If a Z80 peripheral device has no interrupt pending and is not under service, then its IEO = IEI. If it has an interrupt under service (i.e. it has already interrupted and received an interrupt acknowledge) then its IEO is always low, inhibiting lower priority chips from interrupting. If it has an interrupt pending which has not yet been acknowledged, IEO will be low unless an "ED" is decoded as the first byte of a two byte opcode. In this case, IEO will go high until the next opcode byte is decoded, whereupon it will again go low. If the second byte of the opcode was a "4D" then the opcode was an RETI instruction.

After an "ED" opcode is decoded, only the peripheral device which has interrupted and is currently under service will have its IEI high and its IEO low. This device is the highest priority device in the daisy chain which has received an interrupt acknowledge. All other peripherals have IEI = IEO. If the next opcode byte decoded is "4D", this peripheral device will reset its "interrupt under service" condition.



Wait cycles are allowed in the $\overline{M1}$ cycles.

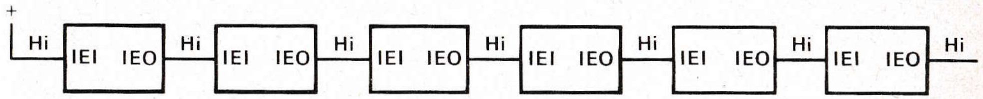
Operation Of SIO

Daisy Chain Interrupt Servicing

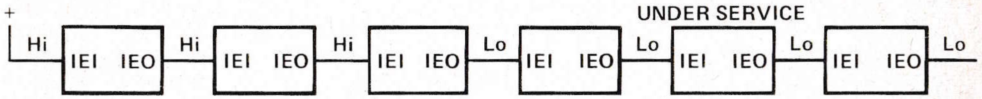
The following illustration is a typical nested interrupt sequence which may occur in the SIO. In a system with several peripheral chips, the other chips may be included in the daisy chain with either higher or lower priority than the SIO channels.

In this sequence, the transmitter of Channel B interrupts and is granted service. While it is being serviced, an external/status interrupt from Channel A occurs and is granted service. The service routine for the Channel A interrupt is completed and either the RETI instruction is executed or the RETI command is written into the SIO to indicate to Channel A that the external/status interrupt routine is complete. At this time, the service routine for the Channel B transmitter is resumed. When this routine is completed, another RETI instruction is executed to complete the service.

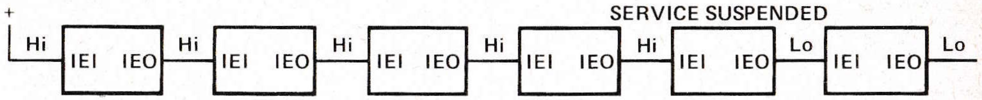
CHANNEL A RECEIVER	CHANNEL A TRANSMITTER	CHANNEL A EXTERNAL/ STATUS	CHANNEL B RECEIVER	CHANNEL B TRANSMITTER	CHANNEL B EXTERNAL/ STATUS



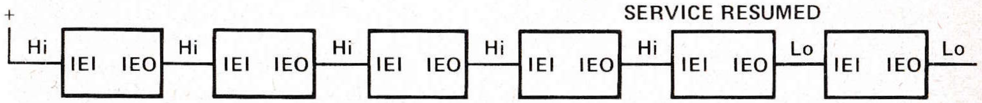
1. Priority Interrupt Daisy Chain before any interrupt occurs.



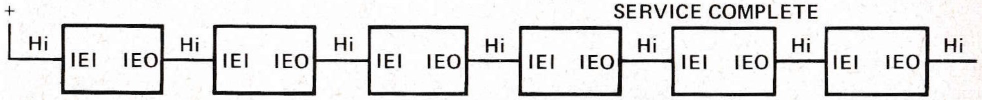
2. Channel B's transmitter interrupts and is acknowledged.



3. External/Status of Channel A interrupts suspending service of Channel B transmitter



4. Channel A External/Status routine complete. RETI issued, Channel B transmitter service resumed.



5. Channel B transmitter's derive routine complete, second RETI issued.

Operation Of SIO (continued)

Operation of the SIO is determined by the contents of the control registers. These must be programmed before any operations can be performed by the SIO. Some commands and modes may be changed during operation. The device status registers can be read at any time.

ASYNCHRONOUS MODES

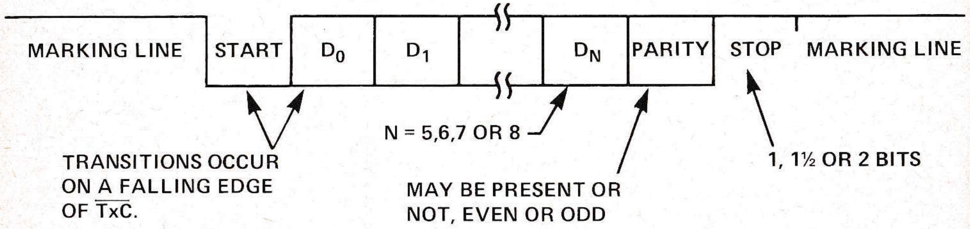
The receiver ports are quadruply buffered, i.e. there are three storage registers in addition to the input shift register. This allows additional time for the CPU to service an interrupt at the beginning of a block of high-speed data transfer. The error flags are also quadruply buffered and are loaded at the same time as the character. The Receiver Overrun and Parity Error flags are not reset unless an Error Reset, (latches) Command (Command 6) is issued. End of Frame and CRC/Framing error always reflect the state of the character currently in the buffer and are not reset by error reset. Thus, when the error status is read, it will reflect an error in the current word in the receive buffer in addition to any parity or overrun errors received since the last Error Reset, (latches) Command. In order to keep correspondence between the state of the error buffer and the contents of the receive registers, the status register should be read before the data (see exception). This is easily accomplished if the vectored interrupts are used since a special interrupt vector is generated for errors or end of frame.

If the status is read after the data is read, the error data for the next data word will also be included if it has been stacked in the buffer. If operations are being performed rapidly enough so that the next character will not yet be received, then the status register will remain valid. The exception occurs when the "Receive Interrupt on First Character Only" mode is selected. A special interrupt in this mode will hold error data and the character itself (even if read from the buffer) until the Error Reset, (latches) Command is issued. This prevents further data from becoming available in the receiver until the Reset is issued.

If the Interrupt on Every Character mode is selected, the interrupt vector will be different if error states exist in the status register. If receiver overrun should occur, despite the quadruple buffering, the most recent character received will be loaded. The character preceding it will be lost. When the character whch has been written over other characters is read, the Overflow bit will be set and the "Special Receive Condition" vector returned if "Status Affects Vector" is enabled.

It is possible to use the SIO in a polled environment. This requires monitoring of the "Receive Character Available" bit to know when to read a character. This bit is reset automatically when the receive buffers are all empty. The "Transmit Buffer Empty" bit is high whenever the transmit buffer is empty. In polled operation, it should be checked before writing data into the transmitter to prevent overwriting of data.

ASYNCHRONOUS FORMAT


MARKING LINE
START
D_0
D_1
D_N
PARITY
STOP
MARKING LINE
N = 5,6,7 OR 8
TRANSITIONS OCCUR ON A FALLING EDGE OF $\overline{TxC}$.
MAY BE PRESENT OR NOT, EVEN OR ODD
1, 1½ OR 2 BITS

TRANSMISSION

A data character sent by the SIO will be assembled as follows in asynchronous modes:

Idle state (no characters being sent) is a marking line (high) unless a break has been programmed in the control register, in which case, the line will remain spacing until the "send break" command has been removed or the chip is reset.

Transmission cannot begin unless the Transmit Enable bit is set. If the Auto Enables option is selected, then $\overline{CTS}$ must be low as well. If the 5 bits/character mode is selected, then unused bits (D_5, D_6 and D_7) must be zero in each data byte written into the SIO.

RECEIVING

Asynchronous reception will begin when the Receiver Enable bit is set. If the Auto Enables option is selected, the $\overline{DCD}$ must be low as well. A low (spacing) condition on RxD indicates a start bit. If the low persists for ½ bit time, the start bit is assumed to be valid and the data input is then sampled at mid-bit time until the entire character is assembled. This method of detecting a start bit improves error rejection when noise spikes exist on an otherwise marking line. If the X1 clock mode is selected, bit synchronization must be accomplished externally.

Synchronous Modes

The various synchronous modes all require a x1 clock for transmission and reception. Data is sampled on the rising edge of $\overline{RxC}$. Transmitter data transitions occur on the falling edge of $\overline{TxC}$.

In all cases, the receiver is in a hunt mode after a reset (internal or external). The hunt can begin only when the receiver is enabled. Only when character synchronization has been achieved can data transfer begin. If there is a loss of character synchronization, the hunt mode can be reentered by writing a control word with the "Enter Hunt Mode" bit set.

The differences in operation of the monosync, bisync and external sync modes are only in the manner in which initial synchronization is achieved. Note: The mode of operation must be selected before the sync characters are loaded, since the registers are used differently in the various modes.

MONOSYNC; (8-BIT SYNC MODE)

Matching of a single sync character, programmed into Write register 7, implies character synchronization, which enables data transfer.

BISYNC: (16-BIT SYNC MODE)

Matching of two adjacent sync characters programmed in Write Registers 6 and 7 implies character synchronization. In both monosync and bisync modes, the $\overline{SYNC}$ pin will be active (low) any time the sync character sequence is detected and will remain low for the clock cycle in which it is detected.

EXTERNAL SYNC MODE

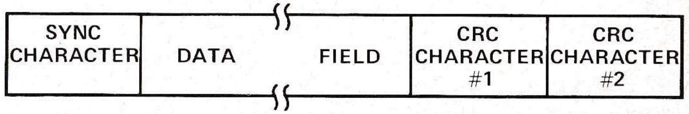
In this mode, character assembly *begins* on the first rising edge of $\overline{RxC}$ after the $\overline{SYNC}$ pin becomes active (low). It should be held active for at least one complete clock cycle.

In Monosync, Bisync and External sync modes, assembly will continue until the SIO is reset (either internally or with the Reset pin) or until the receiver is disabled (by command or with the $\overline{DCD}$ pin in the Auto Enables Mode) or until the CPU sets the "Enter Hunt Mode" bit.

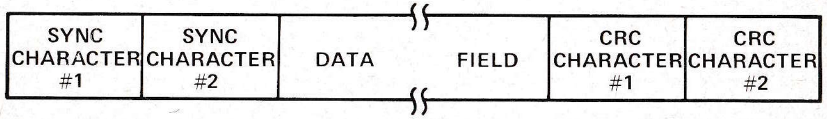
After initial synchronization has been achieved, the Monosync, Bisync, and External Sync modes are very similar. Any differences will be noted in the following, which is meant to apply to all three modes.

SYNCHRONOUS FORMATS

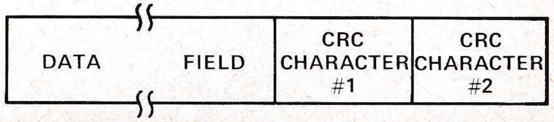
MONOSYNC MESSAGE FORMAT (Internal Sync Detect)



BISYNC MESSAGE FORMAT (Internal Sync Detect)



EXTERNAL SYNC DETECT FORMAT



Synchronous Modes (continued)

Synchronous Modes (Except SDLC) Transmission:

A. Default state (after a Reset or transmitter not enabled) is a marking line. Break may be programmed to generate a spacing line, which begins as soon as programmed, regardless of the contents of the send register. With the transmitter enabled, and after modes have been selected, default is continuous transmission of the 8 or 16 bit sync character.

B. Several Interrupt modes are possible:

1. Transmit interrupts enabled – every time that the transmit buffer becomes empty, an interrupt will be generated if the "Transmit Interrupt Enable" bit is set. The interrupt may be satisfied by either writing another character into the transmitter or by resetting the Transmitter Interrupt pending latch with the "Reset Transmitter Interrupt Pending" command (Command 5). If the interrupt is satisfied with this command and nothing more is written into the transmitter, there will be no further transmitter interrupts, as it is the buffer *becoming* empty that causes the interrupt. When another character is written, the transmitter can again become empty and interrupt again.
2. External/Status interrupts enabled – If the External/Status Interrupt Enable bit is set, Transmitter conditions such as starting to send CRC characters, starting to send Sync characters, and $\overline{\text{CTS}}$ changing state cause interrupts, which have a unique vector if "Status Affects Vector" is set.
3. All interrupts may be disabled for operation in a polled mode or to prevent interrupts at inappropriate times in a program's execution.

C. If CRC is not enabled, sync characters will automatically be inserted when the transmitter has no data to send. An interrupt is generated only after the first automatically inserted sync character has been loaded. If CRC is enabled, the first time the transmitter has no data to send, the 16-bit CRC is automatically sent, followed by sync characters. While sending CRC, the "Sending CRC/SYNCS" bit is set and the "Transmit Buffer Empty" bit indicates full. CRC is not calculated on the automatically inserted sync characters, but it will be calculated on any sync character sent as data unless the CRC generator is disabled when that character is loaded to the transmit shift register from the transmit buffer. When the CRC has been sent, the "Transmit Buffer Empty" bit goes high again, and an interrupt is generated to indicate that another message can begin. Control of the CRC generator may procede as follows:

The CRC generator should be reset by issuing the "RESET TRANSMIT CRC GENERATOR" Command, before any data is loaded. After CRC and the entire transmitter is enabled, data may be loaded. Before CRC is to be sent (but after the first data has been loaded), the CRC/SYNC SENT/SENDING flag must be reset with the "RESET CRC/SYNC SENT SENDING" Command.

Because sending of the CRC is inhibited when the CRC/SYNCS SENT/SENDING flag is set, the SIO can be used to automatically insert fill characters within messages instead of automatically sending the CRC. CRC is not calculated on syncs automatically inserted and when the end of the message is reached, the flag can be reset, thus allowing the CRC to be sent.

D. If the transmitter is disabled while a character is being sent, that character (whether Data, CRC or SYNC) will be sent as normal but will be followed by a marking line rather than CRC or sync characters. A character in the buffer when the transmitter is disabled will remain in the buffer. However, a programmed break will be effective as soon as it is written into the control register. Characters being transmitted, if any, will be lost.

E. In all modes, characters are sent low-order bits first, ie., D_0 before D_1, etc. for as many bits as are programmed. This requires right-hand justification of data to be transmitted if word length is less than 8 bits. If word length is 5 bits or less, the special technique described in the "Transmit Bits/Char" section must be used for the data format.

Synchronous Modes (Except SDLC) Reception:

A. After programming the mode and sync characters (in that order), the receiver may be enabled. It will then be in the Hunt Mode and will stay in that mode until:

1. A match is made with a single sync character (monosync mode) or
2. A match is made with a dual sync character (BiSync mode) or
3. The external $\overline{\text{SYNC}}$ pin is forced low. In cases (1) and (2) the external $\overline{\text{SYNC}}$ pin is an output which indicates that character synchronization has been achieved. In case (3) it is an input.

B. Character assembly begins after sync has been achieved. Four interrupt modes are possible.

1. No interrupts enabled – for a purely polled operation or for "off line" conditions.
2. Interrupt on first character only. This mode would normally be used to start a software polling loop or a block transfer instruction using the $\overline{\text{WAIT/READY}}$ output to synchronize the CPU to the incoming data rate. It could also be used with a DMA device. In this mode, the SIO will interrupt on the first character and thereafter will only interrupt if errors are detected. The mode is reset with the "Reset Receive Interrupt on First Character" command (Command 4).

   The first character received after this command is issued will also cause an interrupt. If External/Status interrupts are enabled, they may interrupt at any time. Parity errors do not cause interrupts in this mode, but End-of-Frame (SDLC Mode) and receiver overrun do cause interrupts.
3. Interrupt on every character – whenever the receiver buffer has a character an interrupt is generated. Error and special conditions generate a special vector if the "Status Affects Vector" mode is selected. A parity error may optionally not generate the special vector.

C. CRC checking generation may be used in the synchronous modes.

1. Calculation of the CRC on a particular character begins 8 bit times after the word has been transferred to the receive buffer. If CRC is enabled before the next character is transferred to the receive buffer, CRC will be calculated on the character. If CRC is disabled before the time of the next transfer, calculation will proceed on the word in progress, but the word just transferred to the buffer will not be include This allows starting and stopping CRC checking on the various characters employed in BiSync.
2. The CRC may be enabled and disabled as many times as necessary for a given calculation.
3. CRC Codes are selected during the mode selection process. Either the CRC-16 polynomial $X^{16} + X^{15} + X^2 + 1$ or the SDLC polynomial $X^{16} + X^{12} + X^5 + 1$ may be used. In all except SDLC mode, the CRC calculator and checker are reset to all 0's. Transmitter and receiver must use the same polynomial.
4. In Monosync, Bisync and External Sync modes, the CRC/FRAMING ERROR bit contains the result of the comparison of the CRC checker to "all zeros" 16 bit times after the character has been loaded from the receive shift register to the buffer. The comparison is made with each load and is valid only as long as the character remains in the buffer. If time increases down the page, then the following holds:

   Character "A" loaded into the buffer

   Character "B" loaded into the buffer...
   If CRC is disabled before "C" is in the buffer it will no be calculated on "B".

   Character "C" loaded into buffer...
   After "C" is loaded the "CRC FRAMING ERROR" bit shows the result of the comparison *thru* Character "A".

   Character "D" loaded into buffer...
   After "D" is in buffer, the CRC ERROR bit shows the result of the comparison thru Character "B".

   Because of the serial operation of the CRC calculation, the receiver clock ($\overline{\text{RxC}}$) must go through 16 cycles after the CRC character has been loaded into the receive buffer (20 cycles after the last bit is at the SIO RxD pin) before the CRC calculation is complete.

Synchronous Modes (continued)

TRANSMISSION
SDLC/HDLC Message Format

FLAG 01111110	ADDRESS 8 BITS	DATA "I" FIELD	CRC #1	CRC #2	FLAG 01111110

SDLC MODE TRANSMISSION:

A. Normally, the CRC generator should be reset (with the "Reset Transmit CRC Generator" command) before a data block is transmitted. Reset may occur any time *after* the CRC of the previous message has been sent. During the time that CRC is being sent, the CRC SYNC SENT/SENDING BIT WILL BE

SDLC MODE TRANSMISSION:

A. Normally, the CRC generator should be reset (with the "Reset Transmit CRC Generator" command) before a data block is transmitted. Reset may occur any time *after* the CRC of the previous message has been sent. During the time that CRC is being sent, the CRC/SYNC SENT/SENDING bit will be set, but the TRANS BUFFER EMPTY bit will not be set. After the CRC has been sent, the TRANS BUFFER EMPTY bit is set again, which will cause an interrupt signifying that the CRC has been sent, if transmit interrupts are enabled.

B. The idle device state (if the transmitter is enabled) is continuous flags being transmitted. If the transmitter is not enabled, a marking line is sent (idle line state).

C. An abort sequence may be sent by issuing the "Send Abort" command (Command 1). This causes at least 8 but less than 14 one's to be sent before the line reverts to continuous flags. Any data being transmitted and any data in the transmit buffer will be lost.

D. One to 8 bits per character may be sent. See the Register Description of Write Register 5, Transmit Bits Char. for an explanation of how this is accomplished. Since the number of bits/character may be changed "on the fly", this feature may be used to fill a data field with any number of bits. When used in conjunction with the Receiver Residue Codes, the SIO may receive a message of any number of bits length and retransmit it exactly as received with no previous information about the character structure of the I-field (if any). A change in the number of bits/character will not affect the character in the process of being shifted out. Characters will be sent with the number of bits programmed at the time that the character is loaded from the buffer to the transmitter.

E. As in other synchronous modes, the two byte CRC sequence will be sent automatically when the transmitter has no more data to send, i.e. when there is no character in the transmit buffer and the transmit shift register is empty. When the CRC sending begins, the CRC/SYNCS SENT/SENDING bit is set and a status change interrupt is generated if external/status interrupts are enabled. This may be used as a transmitter underrun indication. After the CRC has been sent, the line reverts to continuous flags, without shared zeros, i.e. ... 0111111001111110011111100 ....

Control of the CRC generator may proceed as follows:

0. Set up necessary mode (only at initial power on)
1. Reset CRC generator
2. Write first 2 bytes of data (i.e. address and or control bytes)
3. Reset CRC/SYNCS SENT/SENDING bit
4. Write rest of data
5. After data is complete, CRC & flags will be sent automatically, and this sequence can repeat from 1.

F. Extra zeros are automatically inserted in the data stream where required to fulfill the requirement of 5 ones maximum in a row, except for flags or aborts.

G. When SDLC mode is selected, Reset of the CRC generator is actually a preset to all 1's. The SDLC CRC code must be selected.

RECEPTION
SDLC/HDLC Message Format

FLAG 01111110	ADDRESS 8 BITS	DATA "I" FIELD	CRC #1	CRC #2	FLAG 01111110

SDLC OPERATION, RECEIVER

A. Data transfer beings with the first non-flag character received after at least one flag (01111110) has been received if Address Search Mode has not been enabled. If Address Search Mode is enabled, then a flag followed by either the programmed address or the global address (11111111) is required before data transfer will begin.

1. If interrupts are disabled, the presence of characters in the receive buffer can be detected by observing the Receive Character Available bit in Read Register 0.
2. If the "Interrupt on First Character Only" mode has been selected, this would normally be used to initiate a block transfer. If the length of the message is unknown, the "special condition" (End of Frame) interrupt may be used to exit the instruction or software loop. The "Reset Interrupt on first character" command (Command 4) must be issued before an interrupt for a following block's first character can be operated.
3. Flags are not transferred. The extra zeros inserted in transmission are automatically deleted.
4. Aborts are detected as 7 or more one's and cause a status interrupt (if enabled) with the Break/Abort bit set in Read Register 0. After the "Reset External/Status Interrupts" command (Command 2) has been issued, a second interrupt will occur when the continuous one's condition has been cleared.

B. In SDLC mode, control of the receive CRC generator is automatic. It is reset by the leading flag and CRC is calculated up to the final flag. The byte which has the "End-of-Frame" bit set is the byte which contains the result of the CRC check. If the CRC/Framing Error bit is not set, then the CRC indicates a valid message. A special check sequence is used for the SDLC check because of the preset to all one's. The final check must be 0001110100001111.

C. Character length may be changed "on the fly." If address and control bytes are processed as 8-bit characters, the receiver may be switched to a smaller character length during the time that the first information character is being assembled. This change must be made quickly enough so that it is effective before the number of bits specified have been assembled, i.e., if the change is to be from the 8-bit control to a 7-bit information field character length, the change must be made *before* the first 7 bits of the I-field have been assembled.

D. If address search mode is not used, or if messages have multi-byte addresses, an unwanted message need not be completely read by the CPU. Once the determination has been made that the message is not needed, writing the "Enter Hunt Mode" bit will suspend receiption until another message headed by a flag has been received.

E. When the trailing flag is received, an interrupt with a special vector is generated (if enabled). This signals that the byte with the "End of Frame" bit set has been received. In addition to the results of the CRC check, Read Register 1 has 3 bits of Residue Code valid at this time. For those cases in which the number of bits in the I-field is not an integral multiple of the character length used, these bits indicate the boundary between the CRC check bits and the I-field bits. For a detailed description of the meaning of these bits, see the description of the Residue Codes in Read Register 1.

F. Parity checking may be used on data in the information field only if 5-7 bit characters are used and only if a half-duplex protocol is being used. (There are no separate controls for parity on the receiver and transmitter so parity cannot, for example, be simultaneously disabled for transmitting an 8-bit address and enabled for receiving a 5-bit I-field character).

SIO Programming

General

The Z80-SIO is a multi-function peripheral component specifically designed to satisfy a wide variety of serial data communications requirements in microcomputer systems. Its basic role is that of a serial to parallel, parallel to serial converter/controller but within that role it is configured by systems software programming so that its function or "personality" can be optimized for a given serial data communications application.

To program the Z80-SIO the systems software issues a series of commands that initialize the basic mode of operation desired and other commands to qualify conditions within the mode selected i.e. Stop Bits, Bits/Char, Sync Char etc. The command structure of the Z80-SIO is designed to take advantage of the powerful Z80 BLOCK I/O instructions to simplify programming, minimize overhead and optimize CPU interaction activities.

Each of the two channels of the Z80-SIO contain command registers that must be programmed via system software prior to functional operation. The channel select input (B/$\overline{A}$) and the control/data input (C/$\overline{D}$) are the command structure addressing controls, normally controlled by the address bus of the Z80 CPU.

C/$\overline{D}$	B/$\overline{A}$	Function
0	0	Channel A Data
0	1	Channel B Data
1	0	Channel A Commands/Status
1	1	Channel B Commands/Status

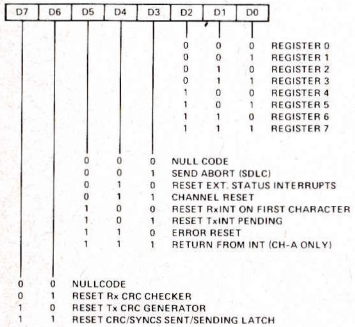
Write Registers

The Z80-SIO contains eight (8) registers that are programmed (written into) by the system software to configure the functional personality of each channel. All Write Registers, with the exception of Write Register 0, require two bytes to be properly programmed. The first byte contains 3 bits which point to the selected register (D0-D2) the second byte is the actual control word that is being written that register to configure the SIO.

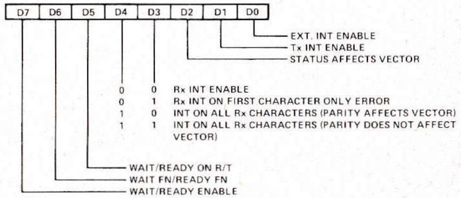
Write Register 0 is a special case. RESET (either internal command or external input) will initialize the SIO to Write Register 0. All basic commands (CMD2-CMD0) and CRC controls (CRC0, CRC1) can be accessed with a single byte using Write Register 0.

Contained in the first byte of any Write Register access are the basic commands (CMD2-CMD0) and the CRC controls (CRC0,CRC1) so that maximum system control and flexibility is maintained.

WRITE REGISTER 0



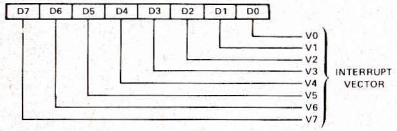
WRITE REGISTER 1



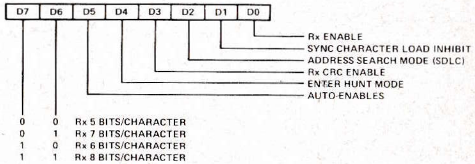
D7 D6 D5 D4 D3 D2 D1 D0
EXT. INT ENABLE
Tx INT ENABLE
STATUS AFFECTS VECTOR
0 0 Rx INT ENABLE
0 1 Rx INT ON FIRST CHARACTER ONLY ERROR
1 0 INT ON ALL Rx CHARACTERS (PARITY AFFECTS VECTOR)
1 1 INT ON ALL Rx CHARACTERS (PARITY DOES NOT AFFECT VECTOR)
WAIT/READY ON R/T
WAIT FN/READY FN
WAIT/READY ENABLE

Write Registers (continued)

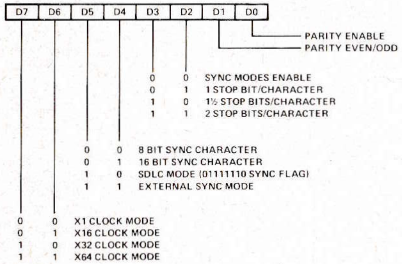
WRITE REGISTER 2



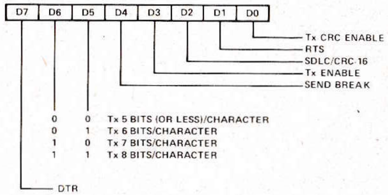
WRITE REGISTER 3



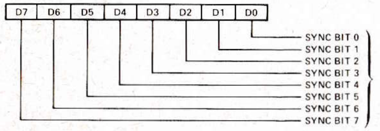
WRITE REGISTER 4



WRITE REGISTER 5

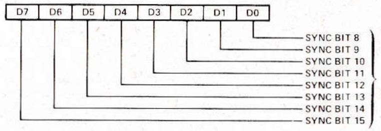


WRITE REGISTER 6



*ALSO SDLC ADDRESS FIELD

WRITE REGISTER 7



*FOR SDLC IT MUST BE PROGRAMMED TO "01111110" FOR FLAG RECOGNITION

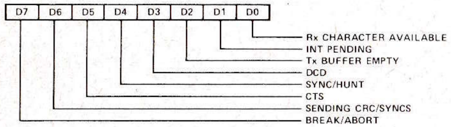
SIO Programming (continued)

Read Registers

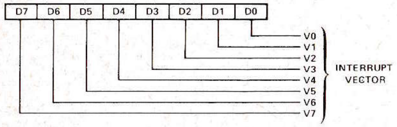
The Z80-SIO contains three (3) registers that can be read to obtain the status of each channel. Status information includes error conditions, interrupt vector, and standard communication interface protocol signals. To read the contents of a selected Read Register the system software must first write out to the SIO the byte containing pointer information (D0-D2) in exactly the same manner as a Write Register operation. Then by issuing a READ operation the contents of the addressed Read/Status Register can be read by the Z80-CPU.

The real power in this type of command structure is that the programmer has complete freedom after pointing to the selected register of either Reading or Writing to initialize or test that register. By designing software to initialize the Z80-SIO in a modular, structured fashion, the programmer can use the powerful Z80 BLOCK I/O instructions to significantly simplify and speed his software development and debug.

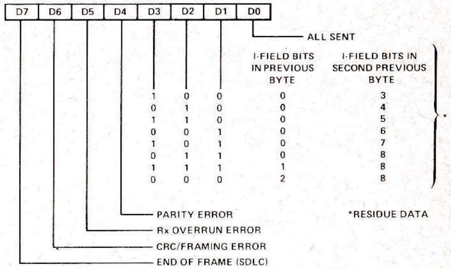
READ REGISTER 0



READ REGISTER 2



READ REGISTER 1



D3	D2	D1	I-FIELD BITS IN PREVIOUS BYTE	I-FIELD BITS IN SECOND PREVIOUS BYTE
1	0	0	0	3
0	1	0	0	4
1	1	0	0	5
0	0	1	0	6
1	0	1	0	7
0	1	1	0	8
1	1	1	1	8
0	0	0	2	8

*RESIDUE DATA

Register Description

Each channel contains the following control registers, addressed as commands (not data):

Write Register 0, a command register:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
CRC Reset Code 1	CRC Reset Code 0	CMD 2	CMD 1	CMD 0	PNT 2	PNT 1	PNT 0

$PNT_0 - PNT_2$ (D_0-D_2)

These are pointer bits which tell the SIO into which register the following byte is to be written. The first byte written into each channel after a reset (either by command or with the external reset pin) will go to write register 0. The byte following a read or write to any register (not register 0) will be to register 0.

CMD_0 to CMD_2 (D_3-D_5)

These are commands:

Command	CMD_2	CMD_1	CMD_0	
0	0	0	0	Null Command (no affect)
1	0	0	1	Send Abort (SDLC Mode)
2	0	1	0	Reset External/Status Interrupts
3	0	1	1	Channel Reset
4	1	0	0	Reset Receive Interrupt on First Character
5	1	0	1	Reset Transmitter Interrupt Pending
6	1	1	0	Error Reset (latches)
7	1	1	1	Return from Interrupt (Channel A only)

COMMAND 0 (The null command) has no affect. It's normal use is to do nothing while setting the pointers for a following byte.

COMMAND 1 (Send Abort) is used only with the SDLC mode to generate a sequence of 8 to 13 ones.

COMMAND 2 (Reset External/Status Interrupts). After an external or status interrupt (indicating a change on a modem line or a break condition, for example) the status bits of Read Register 0 are latched. This command re-enables them and and allows interrupts to occur. The latching allows capture of short pulses on the inputs until such time as the CPU can read the change.

COMMAND 3 (Channel Reset.) This command performs the same operation as an external reset, but only on a single channel. The Channel A Reset also resets the interrupt prioritization logic. All control registers must be rewritten after this command. After this command is written, four extra system (Φ) clock cycles should be allowed for the SIO reset time before any additional commands or controls are written into that channel of the SIO.

COMMAND 4 (Reset Receive Interrupt on First Receive Character.) If the "interrupt only on first receive character" mode of operation is programmed, it needs to be reactivated after each complete message is received, in preparation for the next message.

COMMAND 5 (Reset Transmitter Interrupt Pending.) The transmitter will interrupt when it becomes empty if the "interrupt every character" mode is selected. In those cases when there are no additional characters to be sent, issuing this command will prevent further transmitter interrupts (i.e. until after the next character has been loaded into the transmitter.)

COMMAND 6 (Error Reset, Latches.) Parity and overrun errors are latched in Read Register 1 until reset with this command. This allows errors occuring in block transfers to be examined only at the end of the block.

COMMAND 7 (Return from Interrupt.) This command (which must be issued in Channel A) is interpreted by the SIO in exactly the same way as it would interpret an RETI Command on the data bus, i.e. it would reset the Interrupt Under Service latch of the internal device (receiver, transmitter, etc.) under service and thus, by means of the daisy chain, allow lower priority devices to interrupt. The internal daisy chain may be used even in systems with no external daisy chain and no RETI Command by use of this command.

CRC RESET CODE 0 (D_6) and CRC RESET CODE 1 (D_7)

Together, these bits specify three reset modes.

CRC Reset Code 1	CRC Reset Code 0	
0	0	Null Code (no affect)
0	1	Reset Receive CRC Checker
1	0	Reset Transmit CRC Generator
1	1	Reset CRC/SYNCS Sent Sending latch

Register Description (continued)

WRITE REGISTER 1 contains the control bits for the various interrupt and WAIT/READY modes.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Wait Ready Enable	ReadyFN WaitFN	W Ready On R T	Receive interrupt Mode 1	Receive Interrupt Mode 0	Status Affects Vector	Trans Interrupt Enable	Ext Interrupts Enable

EXT INT ENABLE (D_0)

External Interrupt Enable, allows interrupts to occur as a result of transitions on the $\overline{DCD}$, $\overline{CTS}$ or $\overline{SYNC}$ lines or as a result of a Break Condition or the beginning of sending CRC or sync characters.

TRANS INT ENABLE (D_1)

Transmitter Interrupt Enable. If enabled, interrupts will occur whenever the transmitter buffer becomes empty.

STATUS AFFECTS VECTOR (D_2)

If this mode is selected, the vector returned from an interrupt acknowledge cycle will be variable according to the following:

	V_3	V_2	V_1	
Ch B	0	0	0	Ch B Transmit Buffer Empty
	0	0	1	Ch B External/Status Change
	0	1	0	Ch B Receive Character Available
	0	1	1	Ch B Special Receive Condition
Ch A	1	0	0	Ch A Transmit Buffer Empty
	1	0	1	Ch A External/Status Change
	1	1	0	Ch A Receive Character Available
	1	1	1	Ch A Special Receive Condition

If this bit is 0, the fixed vector programmed in the vector register is returned.

REC INT MODE 0 (D_3), REC INT MODE 1 (D_4)

Receive Interrupt Mode 0 and Receive Interrupt Mode 1 together specify the various character available conditions:

	D_4	D_3	
MODE	REC INT MODE 1	REC INT MODE 0	
0	0	0	Receiver interrupts disabled
1	0	1	Receive interrupt on first character only error
2	1	0	Interrupt on all Receive Characters-Parity/affects Vector
3	1	1	Interrupt on all Receive Characters-Parity error does not affect Vector.

W/READY on R/T (D_5)

When the W/Ready line is enabled, this bit selects whether it will be active when the receiver is empty (bit=1) or when the transmit buffer is full (bit=0).

READY FN/WAIT FN (D_6)

When used with the CPU as a Wait line, this bit should be programmed "0". When used with a DMA as a Ready line, it must be programmed "1". The Ready function can occur any time, regardless of whether the SIO is addressed or not. The Wait function is active only if the CPU attempts to read SIO data that has not yet been received, as would frequently occur if block transfer instructions are used with the SIO, or tries to write data while the transmit buffer is still full.

Also, as a Wait function, the output is open drain and occurs from the negative edge of Φ. As a Ready function, it is actively driven high and occurs from the positive edge of Φ.

WAIT/READY ENABL (D_7)

The Wait/Ready pin will remain high (Ready mode) or floating (Wait mode) until this bit is programmed to one.

WRITE REGISTER 2

Write Register 2 is the interrupt vector register and it exists only in Channel B. V_4-V_7 and V_0 are always returned exactly as written. V_1-V_3 are returned as written if the "Status Affects Vector", Control bit is "0".

WRITE REGISTER 3

Write register 3 contains control bits for some of the receiver logic.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
RCVR Bits Char 0	RCVR Bits/ Char 1	Auto Enables	Enter Hunt Mode	RECVR CRC Enabl	Address Search Mode	Sync Char Load Inhibit	Receiver Enabl

RECEIVER ENABLE (D_0)

A "1" programmed here allows receiver operations to begin.

SYNC CHAR LOAD INHIBIT (D_1)

Sync characters preceding a message will not be loaded into the receiver buffers if this option is selected. The CRC calculation is not stopped by the sync character being stripped.

ADDRESS SEARCH MODE (D_2)

If the SDLC mode is selected, this mode will cause messages with addresses not matching the programmed address or the global (11111111) address to be rejected, i.e., no interrupts occur unless an address match occurs if this mode is selected.

RECVR CRC ENABLE (D_3)

Receiver CRC Enable. If this bit is set, a calculation of CRC begins (or restarts) at the start of the last character transferred from the receive register to the buffer stack regardless of the number of characters in the stack.

ENTER HUNT MODE (D_4)

If character synchronization is lost for any reason, or if in SDLC mode, it is determined that the contents of an incoming message are not needed, Hunt mode may be reentered by writing a "1" to this bit.

AUTO ENABLES (D_5)

If this mode is selected, the $\overline{DCD}$ and $\overline{CTS}$ inputs are receiver and transmitter enables, respectively. If the mode is not selected, $\overline{DCD}$ and $\overline{CTS}$ are only inputs to their corresponding bits in Read Register 0.

RCVR BITS/CHAR 1 (D_6), RCVR BITS/CHAR 0 (D_7)

These bits together determine the number of serial receive bits that will be assembled to form a character.

These bits may be changed during the time that a character is being assembled, if it is done before the number of bits currently programmed is reached.

D_6	D_7	
Receiver Bits/Character 1	Receiver Bits/character 0	Bits/Character
0	0	5
0	1	6
1	0	7
1	1	8

WRITE REGISTER 4

Write Register 4 contains control bits affecting both the receiver and transmitter.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Clock Rate 1	Clock Rate 0	Sync Modes 1	Sync Modes 0	Stop Bits 1	Stop Bits 0	Parity Even Odd	Parity

PARITY (D_0)

If this bit is set, an additional bit position (in addition to those specified in the bits/character control) is added to transmitted data and is expected in receive data.

PARITY EVEN/ODD (D_1)

If parity is specified, this bit determines whether it is sent or checked as even or odd parity.

STOP BITS 0 (D_2), STOP BITS 1 (D_3)

These bits determine the number of stop bits added to each asynchronous character sent. The receiver always checks for one stop bit.

The special (00) mode is used to signify that a synchronous mode is to be selected.

D_3	D_2	
Stop Bits 1	Stop Bits 0	
0	0	Sync Modes
0	1	1 Stop Bit Per Character
1	0	1½ Stop Bits Per Character
1	1	2 Stop Bits Per Character

SYNC MODES 0 (D_4), SYNC MODES (D_5)

These select the various options for character synchronization:

Sync Mode 1	Sync Mode 0	
0	0	8-bit programmed sync
0	1	16-bit programmed sync
1	0	SDLC Mode (01111110 sync pattern)
1	1	External Sync Mode

CLOCK RATE 0 (D_6), CLOCK RATE 1 (D_7)

Specifies the multiplier between clock and data rates. For synchronous modes X1 must be specified. Any rate may be specified for the asynchronous modes. The same multiplier is used for both the receiver and transmitter.

In all modes, the system clock (Φ) must be at least 4.5 X the data rate. If the X1 clock rate is selected, bit synchronization must be accomplished externally.

Clock Rate 1	Clock Rate 0	
0	0	Data Rate X 1 = Clock Rate
0	1	Data Rate X16 = Clock Rate
1	0	Data Rate X32 = Clock Rate
1	1	Data Rate X64 = Clock Rate

Register Description (continued)

WRITE REGISTER 5

Write Register 5 contains mostly control bits affecting the transmitter.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
DTR	Transmit Bits/ Char 0	Transmit Bits/ Char 1	Send Break	Transmit Enable	SDLC/ CRC16	RTS	Transmit CRC Enable

TRANSMIT CRC ENABLE (D_0)

This bit determines whether CRC is to be calculated on any particular send character. If set at the time of loading the character from the transmit buffer to the transmit shift register, CRC will be calculated on the character. CRC will not be automatically sent unless this bit is set when the transmitter is completely empty.

RTS (D_1)

Request to Send is the control bit for the $\overline{\text{RTS}}$ pin. When the RTS bit is set, the $\overline{\text{RTS}}$ goes active (low). When the bit is reset (to 0), the $\overline{\text{RTS}}$ pin will go inactive (high) only after the transmitter is empty.

SDLC/CRC/16 (D_2)

This bit selects the CRC code used by both the transmitter and the receiver. When set, the SDLC polynomial $X^{16} + X^{12} + X^5 + 1$ is used. (In SDLC mode, the registers are preset to "all 1's" and a special check sequence is used.) When set, the CRC-16 polynomial $X^{16} + X^{15} + X^2 + 1$ is used.

TRANSMIT ENABLE (D_3)

Data will not be transmitted and the TxD pin will be held marking (high) until this bit is set. Data or Sync characters in the process of being transmitted will be completely sent if the transmit enable bit is reset after transmission has started. CRC characters will *not* be completely sent if the transmitter is disabled during the sending of a CRC character.

SEND BREAK (D_4)

When set, this bit directly forces the TxD pin spacing, regardless of any data being transmitted. When reset, the TxD pin is released.

TRANSMIT BITS/CHAR 0 (D_5), TRANSMIT BITS/CHAR 1 (D_6)

These bits together control the number of bits that will be sent from each byte transferred to the transmit buffer.

Transmit Bits/Character 1	Transmit Bits/Character 0	Bits/Character
0	0	5 or less
0	1	6
1	0	7
1	1	8

Bits to be sent are assumed to be right justified. Low order bits (D_0) are sent first. The "5 or less" mode allows transmission of 1 to 5 bits in a character.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
1	1	1	1	0	0	0	D	Sends one bit
1	1	1	0	0	0	D	D	Sends two bits
1	1	0	0	0	D	D	D	Sends three bits
1	0	0	0	D	D	D	D	Sends four bits
0	0	0	D	D	D	D	D	Sends five bits

DTR (D_7)

Data Terminal Ready is the control bit for the $\overline{\text{DTR}}$ pin. When set, $\overline{\text{DTR}}$ is active (low). When reset (0) $\overline{\text{DTR}}$ is inactive (high).

WRITE REGISTER 6

This register contains the first 8 bits of a BiSync sequence. It must be programmed with the check address (if used) in SDLC mode, and must contain the sync character in the 8-bit sync mode. It is not used in the external sync mode.

WRITE REGISTER 7

This register contains the second byte of a 16-bit synchronization sequence, or the 8-bit sync character. For SDLC mode, it must be programmed to 01111110. It is not used in the external sync mode.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SYN15	SYN14	SYN13	SYN12	SYN11	SYN10	SYN9	SYN8

READ REGISTER 0

This is the register read if the register pointers are (000).

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Break/ Abort	Sending CRC/ Syncs	CTS	Sync/ Hunt	DCD	Transmit Buffer Empty	Interrupt Pending	Receive Character Available

RECEIVE CHARACTER AVAILABLE (D_0)

This bit is set when at least one character is available in the receive buffers.

INTERRUPT PENDING (D_1)

Any interrupt condition present in the entire SIO will cause this bit to be set, but it is present only in Channel A and is always 0 in Channel B.

TRANSMIT BUFFER EMPTY (D_2)

The Transmit Buffer Empty bit is set whenever the transmit buffer is empty, except when a CRC character is being sent in a synchronous mode.

DCD (D_3)

Shows the state of the $\overline{\text{DCD}}$ pin at the time of the last change of any of the five "external/status" bits. (DCD, CTS, SYNC/HUNT, BREAK/ABORT or SENDING CRC/SYNCS.) To get the current state of the $\overline{\text{DCD}}$ pin, this bit must be read immediately following a "Reset External/Status Interrupts" command. (Command 2.)

SYNC/HUNT (D_4)

In asynchronous modes, this bit is similar to the DCD and the CTS bits, except that it shows the state of the $\overline{\text{SYNC}}$ pin. In synchronous modes, this bit is reset when character synchronization is achieved and is set by writing the "Enter Hunt Mode" bit. Unlike the external pin, the bit remains reset until set by the "Enter Hunt Mode" bit.

CTS (D_5)

This bit is similar to the DCD bit, except that it shows the state of the $\overline{\text{CTS}}$ pin.

BREAK/ABORT (D_6)

In asynchronous modes, this bit is set when a "break" is detected. After the inputs have been re-enabled (by the "Reset External/Status Interrupts" command, Command 2), the bit will be reset when the break stops. If "External/Status" interrupts are enabled, these changes of state cause interrupts. In SDLC mode, this bit is set by the detection of an abort sequence (7 or more 1's). It is not used in other synchronous modes.

SENDING CRC/SYNCS (D_7)

In synchronous modes, CRC is automatically sent when the transmitter is empty for the first time in a message. Interrupts are generated (if enabled) when this bit is set, but not when reset. If this bit is set and the TRANSMIT BUFFER EMPTY bit is not set, then the CRC character is being sent. TRANSMIT BUFFER EMPTY and SENDING CRC/SYNCS both set imply that SYNC characters are being sent.

READ REGISTER 1

This register is read when the register pointers are (001). The pointers automatically reset to (000) after a read from this register.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
End Of Frame (SDLC)	CRC/ Framing Error	Receiver Overrun Error	Parity Error	Residue Code 2	Residue Code 1	Residue Code 0	All Sent

ALL SENT (D_0)

In asynchronous modes, this bit is set when all characters have completely cleared the transmitter. Transitions of this bit do not cause interrupts. It is always set in synchronous modes.

RESIDUE CODE 0 (D_1)—RESIDUE CODE 2 (D_3)

These three bits indicate the length of the I-field in the SDLC mode in those cases where the I-field is not an integral multiple of the character length used. Only on the transfer on which the END OF FRAME (SDLC) bit is set do these codes have meaning.

For a receiver setting of eight bits per character, the codes signify the following:

Residue Code 2	Residue Code 1	Residue Code 0	I-Field Bits In Previous Byte	I-Field Bits In Second Previous Byte
1	0	0	0	3
0	1	0	0	4
1	1	0	0	5
0	0	1	0	6
1	0	1	0	7
0	1	1	0	8
1	1	1	1	8
0	0	0	2	8

I-Field bits are right-justified in all cases.

If a receive character length different from eight bits is used for the I-field, a table similar to the above may be constructed for each different character length. For no residue, i.e., the last character boundary coincides with the boundary of the I-Field and CRC Field, the Residue Code will always be:

Residue Code 2	Residue Code 1	Residue Code 0
1	0	1

PARITY ERROR (D_4)

When parity is enabled, this bit is set for those characters whose parity does not match the sense programmed. The bit is latched so that once an error occurs, the bit remains set until the Error Reset command, Command 6, is given.

RECEIVER OVERRUN ERROR (D_5)

This indicates that more than four characters have been received without a read from the CPU. Only the character that has been written over is flagged with this error, but when this character is read, the error condition is latched until reset by the Error Reset Command, Command 6. If "Status Affects Vector" is enabled, the character that has been overrun will interrupt with the "Special Receive Condition" vector.

CRC/FRAMING ERROR (D_6)

If a framing error occurs (in asynchronous modes), this bit is set (and not latched) only for the character on which it occurred. Detection of a framing error adds an additional ½ bit time to the character time so that the framing error will not also be interpreted as a new start bit. In synchronous modes, this bit indicates the result of comparing the CRC checker to the appropriate check value.

END OF FRAME (SDLC) (D_7)

In SDLC mode, this bit indicates that a valid ending flag has been received and that the CRC error and residue codes are valid.

READ REGISTER 2

This register contains the interrupt vector as written into Write Register 2 if the "Status Affects Vector" control bit is not set. If that control bit is set, it contains the interrupt vector as it would be returned were an interrupt from the SIO to be processed exactly at the time of the read. If no interrupts are pending, $V_3 = 0$, $V_2 = 1$, $V_1 = 1$ and other bits are as programmed. The register may be read only through Channel B.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
V_7	V_6	V_5	V_4	V_3	V_2	V_1	V_0

Variable if "Status Affects Vector" is enabled

Z80-SIO COMMAND STRUCTURE

Reg.	Control			DATA BITS							
#	C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	D7	D6	D5	D4	D3	D2	D1	D0
0	1	1	0	CRC 1	CRC 0	CMD 2	CMD 1	CMD 0	0	0	0
	1	1	0	CRC 1	CRC 0	CMD 2	CMD 1	CMD 0	0	0	0
	1	0	1	Break/Abort	Sendg CRC/SYNC	CTS	SYNC/HUNT	DCD	TxBuffer EMPTY	INT Pending (CH-A Only)	RxChar Avail
1	1	1	0	CRC 1	CRC 0	CMD 2	CMD 1	CMD 0	0	0	1
	1	1	0	Wait/RDY EN	WaitFN/RDYFN	Wait/RDYon R/T	RxINT Mode 1	RxINT Mode 0	Status Effects V (CH-B Only)	TxINT EN	EXT INT EN
	1	0	1	Endo FrameSDLC	CRC FrameError	RxOVRN Error	Parity Error	Res.Code 2	Res.Code 1	Res.Code 0	All Sent
2	1	1	0	CRC 1	CRC 0	CMD 2	CMD 1	CMD 0	0	1	0
CH-B ONLY	1	1	0	V7	V6	V5	V4	V3	V2	V1	V0
	1	0	1	V7	V6	V5	V4	V3	V2	V1	V0
3	1	1	0	CRC 1	CRC 0	CMD 2	CMD 1	CMD 0	0	1	1
	1	1	0	RxBits/Char 0	RxBits/Char 1	Auto Enables	Enter HuntMode	RxCRC EN	AddrssSearch Md	SyncChar LD INH	RxEN
4	1	1	0	CRC 1	CRC 0	CMD 2	CMD 1	CMD 0	1	0	0
	1	1	0	Clock Rate 1	Clock Rate 0	Sync Mode 1	Sync Mode 0	Stop Bits 1	Stop Bits 0	Parity Even/Odd	Parity
5	1	1	0	CRC 1	CRC 0	CMD 2	CMD 1	CMD 0	1	0	1
	1	1	0	DTR	TxBits/Char 0	TxBits/Char 1	Send BREAK	TxEN	$\overline{\text{SDLC}}$/CRC-16	RTS	TxCRC EN
6	1	1	0	CRC 1	CRC 0	CMD 2	CMD 1	CMD 0	1	1	0
	1	1	0	SYNC/SDLC 7	SYNC/SDLC 6	SYNC/SDLC 5	SYNC/SDLC 4	SYNC/SDLC 3	SYNC/SDLC 2	SYNC/SDLC 1	SYNC/SDLC 0
7	1	1	0	CRC 1	CRC 0	CMD 2	CMD 1	CMD 0	1	1	1
	1	1	0	SYNC/SDLC 15	SYNC/SDLC 14	SYNC/SDLC 13	SYNC/SDLC 12	SYNC/SDLC 11	SYNC/SDLC 10	SYNC/SDLC 9	SYNC/SDLC 8

Programming Example

A typical start-up routine following an internal or external reset, would be as follows:

B/$\overline{\text{A}}$	C/$\overline{\text{D}}$	$\overline{\text{RD}}$	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	COMMENTS
1	1	1	0	0	0	0	0	0	1	0	Pointer set to Register 2B
1	1	1	V_7	V_6	V_5	V_4	V_3	V_2	V_1	V_0	Interrupt Vector loaded
1	1	1	0	0	0	0	0	1	0	0	Pointer set to Write Register 4B
1	1	1	0	1	X	X	0	1	1	1	Even parity, 1 stop bit, X16 clock, asynchronous mode selected
1	1	1	0	0	0	0	0	1	0	1	Pointer set to Write Register 5B
1	1	1	0	1	0	0	1	0	1	0	7 bits/transmit character, transmitter enabled
1	1	1	0	0	0	0	0	0	1	1	Pointer set to Write Register 3B
1	1	1	0	1	1	0	0	0	0	1	7 bits/receive character, DCD and CTS enable Receiver and Transmitter, Receiver enabled
1	1	1	0	0	0	0	0	0	0	1	Pointer set to Register 1B
1	1	1	0	0	0	1	0	1	1	1	Interrupt on every character, status affects Vector external/status interrupts enabled

Channel B is now setup to send and receive asynchronous data.

Setup for Channel A follows:

0	1	1	0	0	0	0	0	1	0	0	Pointer set to Write Register 4A
0	1	1	0	0	1	0	0	0	0	0	SDLC mode and X1 clock selected, no parity

Programming Example

B/$\overline{\text{A}}$	C/$\overline{\text{D}}$	$\overline{\text{RD}}$	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	COMMENTS
0	1	1	0	1	0	0	0	1	1	0	Pointer set to Write Register 6A, Reset Receive CRC Checker
0	1	1	AD_7	AD_6	AD_5	AD_4	AD_3	AD_2	AD_1	AD_0	SDLC message address entered
0	1	1	1	0	0	0	0	1	1	1	Pointer set to Write Register 7A, Reset Transmit CRC generator
0	1	1	0	1	1	1	1	1	1	0	SDLC Flag entered
0	1	1	0	0	0	0	0	0	0	1	Pointer set to Register 1A
0	1	1	0	0	0	1	0	1	1	1	Interrupt every character, status affects vector, external/status interrupts enabled
0	1	1	0	0	0	1	0	1	0	1	Pointer set to Write Register 5A, Reset External/Status Interrupts
0	1	1	1	1	1	0	1	0	0	0	SDLC CRC Code selected, 8 bits/transmit character, CRC and transmitter enabled
0	1	1	0	0	0	0	0	0	1	1	Pointer set to Write Register 3A
0	1	1	1	1	1	0	1	1	0	1	8 bits/receive character, DCD and CTS enable receiver and transmitter, receiver is enabled, SIO searches for programmed address

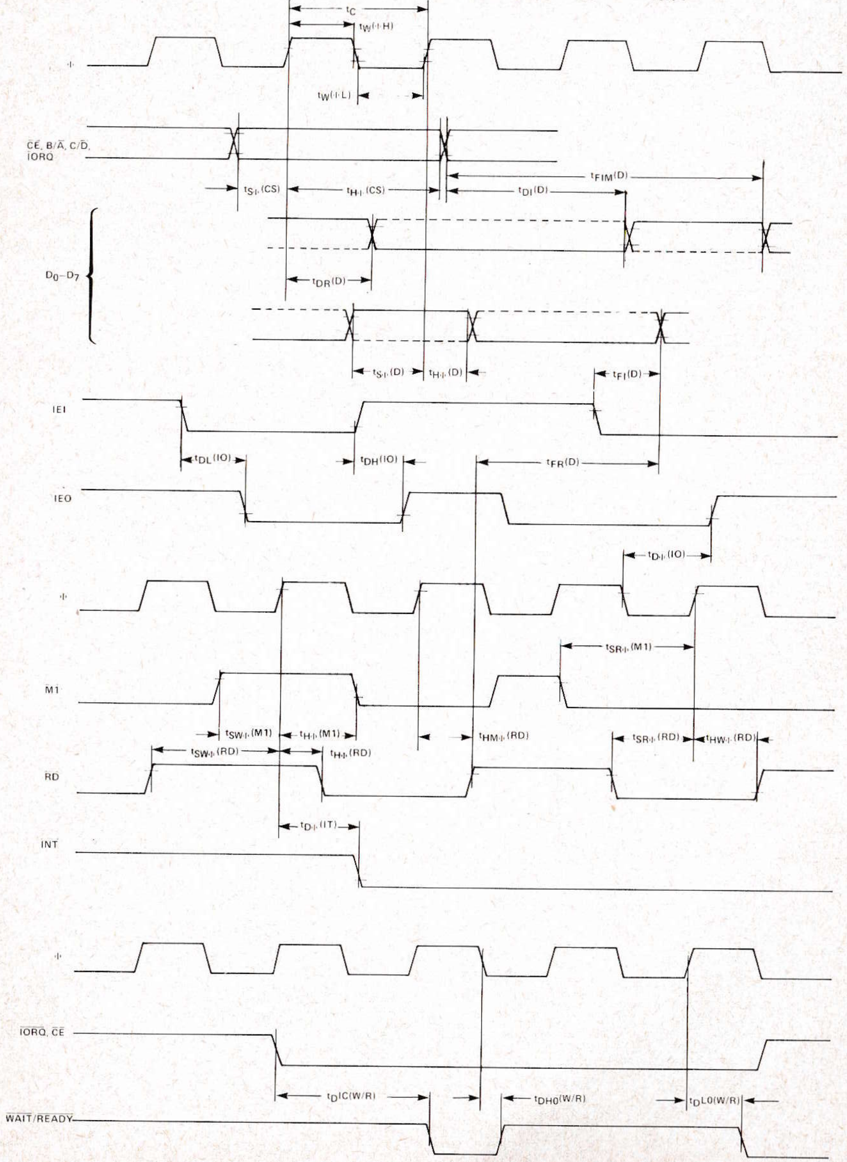
Channel A is now programmed for SDLC transfers.

0	0	1	D	D	D	D	D	D	D	D	Address byte to be sent by Ch. A
0	0	1	D	D	D	D	D	D	D	D	Address or control byte to be sent by Ch. A
0	1	1	1	1	0	0	0	0	0	0	Reset CRC/SYNCS SENT/SENDING, pointer to register 0, so CRC can be automatically sent at end of message

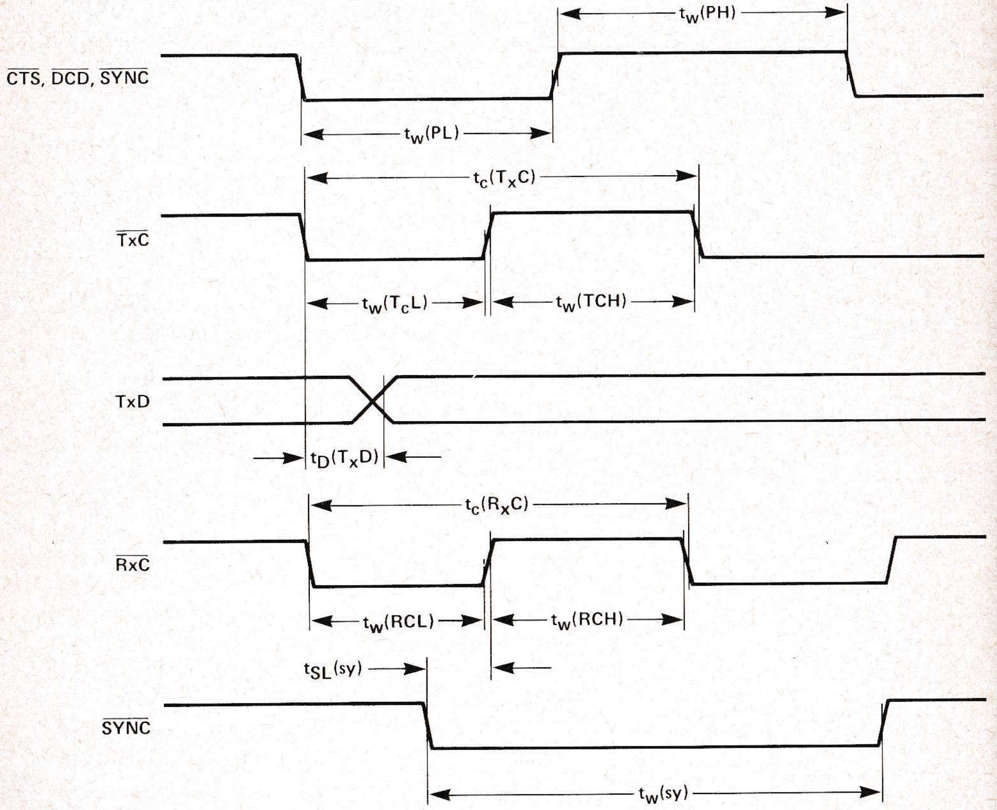
A.C. Timing Diagram

Timing measurements are made at the following voltages, unless otherwise specified:

CLOCK	4.2V	0.8V	**Only for timing measurements**
OUTPUT	2 V	0.8V	
INPUT	2 V	0.8V	
FLOAT	ΔV =	±0.5V	



A.C. Timing Diagram (continued)



T_A = 0° C to 70° C, Vcc = +5V ±5%, unless otherwise noted

Signal	Symbol	Parameter	Min	Max	Unit	Comments
Φ	$t_C(\Phi)$	Clock Period	400		nsec	
	$t_W(\Phi H)$	Clock Pulse Width, Clock High	170	2000	nsec	
	$t_W(\Phi L)$	Clock Pulse Width, Clock Low	170	2000	nsec	
	t_r, t_f	Clock Rise and Fall Times	-0-	30	nsec	
$\overline{CE}$,B/$\overline{A}$	$t_H(CS)$	Control Signal hold time from Rising Edge of Φ	-0-		nsec	NOTE 1
C/$\overline{D}$,$\overline{IORQ}$	$t_S(CS)$	Control Signal setup time from Rising Edge of Φ	160			
	$t_{DR}(D)$	Data Output Delay from Rising Edge of Φ during Read Cycle		480	nsec	
	$t_S\Phi(D)$	Data Setup Time to Rising Edge of Φ during Write Cycle or M1 Cycle	50		nsec	
	$t_H\Phi(D)$	Data Hold Time from Rising Edge of Φ during Write Cycle or M1 Cycle	-0-		nsec	
D_0-D_7	$t_{DI}(D)$	Data Output Delay from Falling Edge of IORQ during INTA Cycle		340	nsec	
	$t_{FIM}(D)$	Delay to Floating Bus from Rising Edge of IORQ during INTA Cycle		230	nsec	
	$t_{FR}(D)$	Delay to Floating Bus from Rising Edge of RD during Read Cycle		230	nsec	
	$t_{FI}(D)$	Delay to Floating Bus from Falling Edge of IEI during INTA Cycle		230	nsec	
IEO	$t_{DL}(IO)$	IEO Delay Time from Falling Edge of IEI		200	nsec	
	$t_{DH}(IO)$	IEO Delay Time from Rising Edge of IEI		200	nsec	
	$t_D\Phi(IO)$	IEO Delay Time from Falling Edge of M1 (when interrupt occurs just prior to M1)		300	nsec	
$\overline{M1}$	$t_{SW}\Phi(M1)$	M1 Setup Time to Rising Edge of Φ during Read or Write Cycle	210		nsec	
	$t_{SR}\Phi(M1)$	M1 Setup Time to Rising Edge of Φ during INTA or M1 Cycle	210		nsec	
	$t_H\Phi(M1)$	M1 Hold Time from Rising Edge of Φ	-0-		nsec	
$\overline{RD}$	$t_{SW}\Phi(RD)$	RD Setup Time to Rising Edge of Φ during Write or INTA Cycle	240		nsec	
	$t_H\Phi(RD)$	RD Hold Time from Rising Edge of Φ during INTA Cycle	-0-		nsec	
	$t_{SR}\Phi(RD)$	RD Setup Time to Rising Edge of Φ during Read or M1 Cycle	240		nsec	
	$t_{HW}\Phi(RD)$	RD Hold Time from Rising Edge of Φ during Write Cycle	-0-		nsec	
	$t_{HM}\Phi(RD)$	RD Hold Time from Rising Edge of Φ during M1 Cycle	-0-		nsec	
$\overline{INT}$	$t_{DRX}(IT)$	INT Delay Time from center of Receive Data Bit	10	13	ΦPeriods	
	$t_{DTX}(IT)$	INT Delay Time from center of Transmit Data Bit	5	9	ΦPeriods	
	$t_D\Phi(IT)$	INT Delay Time from Rising Edge of Φ		200	nsec	
$\overline{WAIT/READY}$	$t_DIC(W/R)$	WAIT/READY Delay Time from IORQ or CE in WAIT Mode		180	nsec	
	$t_DH\Phi(W/R)$	WAIT/READY Delay Time from Falling Edge of Φ, WAIT/READY hIGH, WAIT Mode		150	nsec	
	$t_DRx(W/R)$	WAIT/READY Delay Time from center of Receive Data Bit, Ready Mode	10	13	ΦPeriods	
	$t_DTx(W/R)$	WAIT/READY Delay Time from center of Transmit Data bit, Ready Mode	5	9	ΦPeriods	
	$t_DL\Phi(W/R)$	WAIT/READY Delay from Rising Edge of Φ, WAIT/READY, Low, Ready Mode		120	nsec	
$\overline{CTSA}$,$\overline{CTSB}$ $\overline{DCDA}$,$\overline{DCDB}$, $\overline{SYNCA}$,$\overline{SYNCB}$	$t_W(PH)$	Minimum High Pulse Width for latching states into register and generating interrupt	200		nsec	
	$t_W(PL)$	Minimum Low Pulse Width for latching state into register and generating interrupt	200		nsec	
$\overline{SYNCA}$,$\overline{SYNCB}$	$t_{DI}(SY)$	Sync Pulse Delay Time from Center of Receive Data Bit, Output Modes	4	7	ΦPeriods	
	$t_{SL}(SY)$	Sync Pulse Setup Time to Rising Edge of RxC, External Sync Mode	100		nsec	
	$t_W(SY)$	Sync Pulse Width to Start Character Assembly	1		$\overline{RxC}$ Period	
$\overline{TxCA}$,$\overline{TxCB}$	$t_C(TxC)$	Transmit Clock Period	400	∞	nsec	NOTE 2
	$t_W(TCH)$	Transmit Clock Pulse Width, Clock High	180	∞	nsec	
	$t_W(TCL)$	Transmit Clock Pulse Width, Clock Low	180	∞		
TxDA,TxDB	$t_D(TxD)$	TxD Output Delay from Falling Edge of $\overline{TxC}$ (1x Clock Mode)		400	nsec	
$\overline{RxCA}$,$\overline{RxCB}$	$t_C(RxC)$	Receive Clock Period	400	∞	nsec	NOTE 3
	$t_W(RCH)$	Receive Clock Pulse Width, Clock High	180	∞	nsec	
	$t_W(RCL)$	Receive Clock Pulse Width, Clock Low	180	∞	nsec	

NOTE 1: If WAIT is to be used, $\overline{CE}$, $\overline{IORQ}$, C/$\overline{D}$ and $\overline{M1}$ must be valid for as long as WAIT condition is to persist.

NOTE 2: In all modes, maximum data rate must be less than $\frac{1}{4.5}$ of system clock (Φ) rate.

NOTE 3: The RESET signal must be active a minimum of one complete Φ cycle.

Absolute Maximum Ratings

Temperature Under Bias	Specified operating range
Storage Temperature	-65°C to +150°C
Voltage On Any Pin with Respect to Ground	-0.3V to +7V
Power Dissipation	1.5W

* Comment
Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. Characteristics

T_A = 0°C to 70°C, V_{CC} = 5V ±5% unless otherwise specified

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	-0.3		0.40	V	
V_{IHC}	Clock Input High Voltage	V_{CC}-0.2[1]		V_{CC}	V.	
V_{IL}	Input Low Voltage	-0.3		0.8	V	
V_{IH}	Input High Voltage	2		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = 1.8 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = -250μA
V_{CC}	Power Supply Current			140	mA	t_C = 400 nsec
I_{LI}	Inṕut Leakage Current			10	μA	A_{IN} = 0 to V_{CC}
I_{LOH}	Tri-State Output Leakage Current in Float			10	μA	V_{OUT} = 2.4 to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float			-10	μA	V_{OUT} = 0.4V
I_{LD}	Data Bus Leakage Current in Input Mode			±10	μA	$0 \leqslant V_{IN} \leqslant V_{CC}$

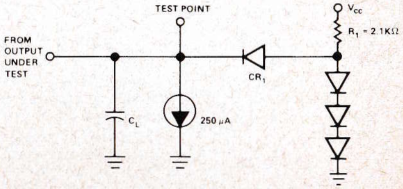
Note 1: An external clock pull-up resistor of (330Ω) will meet both the AC and DC clock requirements.

Capacitance

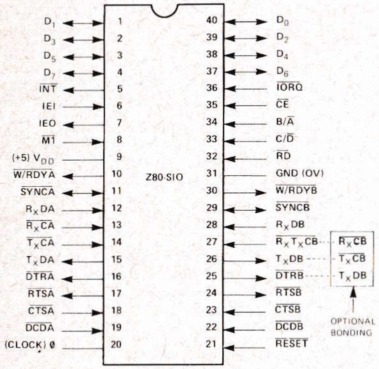
T_A = 25°C, f = 1 MHz

Symbol	Parameter	Max.	Unit	Test Condition
C_Φ	Clock Capacitance	35	pF	Unmeasured Pins
C_{IN}	Input Capacitance	5	pF	Returned to Ground
C_{OUT}	Output Capacitance	10	pF	

Load Circuit for Output



Package Configuration

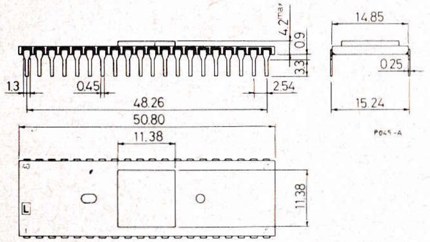


ORDERING NUMBERS:

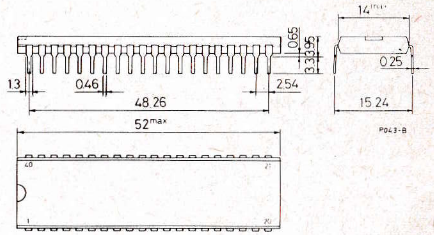
Z80-SIO D1 for dual in-line ceramic slam package
Z80-SIO B1 for dual in-line plastic package

MECHANICAL DATA (dimensions in mm)

40-PIN CERAMIC DUAL IN-LINE SLAM PACKAGE



40-PIN PLASTIC DUAL IN-LINE PACKAGE



Information furnished is believed to be accurate and reliable. However, no responsibility is assumed for the consequences of its use nor for an infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of SGS-ATES. This publication supersedes and substitutes all information previously supplied.

SGS – ATES GROUP OF COMPANIES
Italy – France – Germany – Singapore – Sweden – United Kingdom – U.S.A.
Printed in Italy by permission of Zilog Inc.

SGS-ATES GROUP OF COMPANIES

INTERNATIONAL HEADQUARTERS

SGS-ATES Componenti Elettronici SpA
Via C. Olivetti 2 - 20041 Agrate Brianza - Italy
Tel.: 039-650341÷4/650441÷5/650841÷5
Telex: 36141-36131

BENELUX

SGS-ATES Componenti Elettronici SpA
Benelux Sales Office
-B-1180 Bruxelles
Winston Churchill Avenue, 122
Tel.: 02-3432439
Telex: 24149 B

DENMARK

SGS-ATES Scandinavia AB
Sales Office:
2730 Herlev
Marielundvej 46D
Tel.: 02-948533
Telex: 35280

FRANCE

SGS-ATES France S.A.
75643 Paris Cedex 13
Résidence "Le Palatino"
17, Avenue de Choisy
Tel.: 5842730
Telex: 021-25938

GERMANY

SGS-ATES Deutschland Halbleiter
Bauelemente GmbH
8018 Grafing bei München
Haidling 17
Tel.: 08092-691
Telex: 032-527370
Sales Offices:
1000 Berlin 20
Gatower Strasse 185
Tel.: 030-3622031
Telex: 01 85418
3000 Hannover 1
Lange Laube 19
Tel.: 0511-17522/3
Telex: 09 23195
8000 München 40
Gernotstrasse 10
Tel.: 089-304270/304485
Telex: 05 215784
8500 Nurnberg 15
Parsifalstrasse 10
Tel.: 0911-40645
7000 Stuttgart 80
Kalifenweg 45
Tel.: 0711-713091/2
Telex: 07 255545

ITALY

SGS-ATES Componenti Elettronici SpA
Sales Offices:
50127 Firenze
Via Giovanni Del Pian Dei Carpini 96/1
Tel.: 055-4377763
20149 Milano
Via Correggio 1/3
Tel.: 02-4695651
00199 Roma
Piazza Gondar 11
Tel.: 06-8392848/8312777
10134 Torino
Via La Loggia 51/7
Tel.: 011-634572

NORWAY

SGS-ATES Scandinavia AB
Sales Office:
Oslo 9
Haavard Martinsens Vei 19
Tel.: 10 60 50
Telex: 11796

SINGAPORE

SGS-ATES Singapore (Pte) Ltd.
Singapore 12
Lorong 4 & 6 - Toa Payoh
Tel.: 531411
Telex: ESGIES RS 21412

SWEDEN

SGS-ATES Scandinavia AB
19501 Märsta
Tingvallavaegen 9J
Tel.: 0760-40120
Telex: 042-10932

UNITED KINGDOM

SGS-ATES (United Kingdom) Ltd.
Aylesbury, Bucks
Planar House, Walton Street
Tel.: 0296-5977
Telex: 041-83245