DOC1013

SB8010

Z80 CPU CARD

REFERENCE MANUAL

APPLICABILITY D

REVISION DATE 12/20/83

MICRO/SYS

1011 Grand Central Avenue. Glendale. California 91201-3010 (818) 244-4600

1.0 INTRODUCTION

The Micro/sys SB8010 is a highly versatile card comprised of a Z-80 microprocessor, 4 general purpose 28 pin sockets for byte-wide ROM and/or RAM chips, a Z-80 CTC counter-timer, and a serial I/O port (USART) using Intel's 8251. A bootstrap option is provided for loading an operating system into main memory using an on-board PROM.

On-board memory can be mapped in various ways to match the memory chips selected. The mapping options are accomplished using a PROM. The User may install his own PROM if a different mapping is desired. However, the six mappings provided should satisfy most users.

The serial I/O chip uses the counter-timer chip as the clock source for both receiver and transmitter to that baud rate is programmable by software. RS-232 buffers are provided for all the serial outputs and inputs. In an interrupt driven system, the serial chip (8251) uses the CTC as an interrupt controller, enabling the serial chip (8085 peripheral) to benefit from the sophisticated Z-80 interrupt scheme.

The CTC can also be used as an interrupt controller for external events as well as a general purpose counter and/or timer.

Detailed card operation is discussed in section 2. Section 3 explains all user options. Section 4 provides register addresses of all the on-board I/O functions as well as rough guidelines to several modes of operation.

Appendix B contains a Zilog Z-80 data sheet with a mnemonics summary. Detailed Z-80 programming information is not provided as this is commonly available elsewhere. Appendix C and D contain 8251 and CTC data sheets and/or programming information. Refer to these appendices for I/O programming guidelines while obtaining specific register addresses from Section 4. Appendix E and F contain the data programmed into the memory decoder PROM and the I/O decoder PROM. Finally, appendix G contains the PAL (Programmable Array Logic) equations.

TABLE OF CONTENTS

1.0 INTRODUCTION

2.0 THEORY OF OPERATION

- 2.1 BUFFERS
- 2.2 RESET CIRCUIT
- 2.3 MAIN LOGIC
- 2.4 on-board MEMORY
- 2.5 BOOT CIRCUIT
- 2.6 on-board I/0
- 2.7 INTERRUPT SYSTEM

3.0 CONFIGURATION AND OPTIONS

- 3.1 EXTERNAL CONNECTORS
- 3.2 ON-BOARD MEMORY
  - 3.2.1 RESTRICTIONS
  - 3.2.2 CONFIGURING THE MEMORY SOCKETS
  - 3.2.3 ENABLING THE SOCKETS
  - 3.2.4 MEMORY MAP AND BOOT OPTION
- 3.3 MEMEX AND IOEXP
  - 3.3.1 MEMEX
  - 3.3.2 IOEXP
- 3.4 ON-BOARD I/O
  - 3.4.1 OUTPUT CONNECTORS
  - 3.4.2 I/O CONFIGURATIONS
    - 3.4.2.1 USING THE SERIAL CHIP
    - 3.4.2.2 USING THE CTC TO CONTROL EXTERNAL INTERRUPTS

4.0 PROGRAMMING

- 4.1 I/O ADDRESSES
  - 4.1.1 THE SERIAL CHIP ADDRESSES
  - 4.1.2 THE CTC CHIP ADDRESSES
- 4.2 SERIAL CHIP BAUD RATE PROGRAMMING
  - 4.2.1 GUIDELINES FOR BAUD RATE PROGRAMMING
- 4.3 USING THE CTC AS AN INTERRUPT CONTROLLER
- 4.4 RESETTING THE SERIAL CHIP

APPENDICES

- A ASSEMBLY, SCHEMATICS DIAGRAMS
- B Z-80 DATA SHEET
- C SERIAL CHIP - 8251 - DATA SHEETS
- D CTC (COUNTER/TIMER CHIP) DATA SHEETS
- E MEMORY DECODER PROM
- F I/O DECODER PROM
- G PAL (PROGRAMMABLE LOGIC ARRAY) EQUATIONS

2.0 THEORY OF OPERATION

2.1 BUFFERS

All address lines, data lines and output control lines are buffered to provide sufficient drive to a multi-module system. The address and control buffers are unidirectional, pointing towards the bus. The only time they are turned off (in high impedance state) is during a DMA cycle.

The data buffer is bidirectional. It is pointing out (to the bus) during a write operation or an on-board read. It is pointing in during a read which is not on-board. It is turned off during a DMA cycle.

Four input CPU control lines, namely BUSREQ/, WAIT/, INT/ and NMI/, are driven directly off the bus. These lines are static protected by pull up resistors. The advantage of this arrangement is greater response speeds. This is especially important with the WAITRQ/ signal when operating with dynamic memories.

2.2 RESET CIRCUIT

The SB8010 reset circuit provides a reset pulse to the CPU and the SYSRESET/ line on the bus when either the push button reset is activated or power is turned on. The pulse generated is synchronized with the beginning of a fetch cycle and has a fixed duration irrespective of the duration of the push button reset signal. The purpose of this scheme is to avoid the possibility of erasing dynamic memories using CPU refresh. Upon reset the CPU starts fetching instructions from address 0.

2.3 MAIN LOGIC

Most of the logic functions on the card are performed using a PAL (Programmable Array Logic) at U1 and 2 PROMS (U8 and U7). The PROMS provide the PAL with decoded address signals for on-board memory and I/O. The PAL, along with CPU control signals and address lines A1 and A2, does the following:

a. Determines direction of the data buffer
b. Enables the serial I/O port
c. Enables the timer/counter ship
d. Enables the boot flip flop
e. Generates STD bus signals MCSYNC/ and INTAK/

Appendices E and F contain the data programmed into the decoder PROM's. Appendix G contains the PAL equation.

2.4 ON-BOARD MEMORY

Four general purpose memory sockets are provided that can be configured to accept ROM/PROM/EPROM and/or RAM chips of sizes 2,4 or 8K bytes. (See restrictions in section 3.2.1). A memory map PROM (U7) enables these chips according to one of 6 different combinations of sizes and mappings selectable via option posts. The memory maps selected all start at address zero. The 6 mappings provided will satisfy most users. However, the user may program his own PROM if a different mapping is desired (Up to 8 different mappings can be programmed.). Appendix D contains the data programmed into this PROM. The 6 mappings mentioned here may be conditioned with a signal from the boot flip flop (U2 pin 6) if J6 pins 8 and 10 are connected. When this signal is high (pin 19 of U7 is high), a different set of memory map is enabled. These additional maps are used for bootstrapping. See next section.

Whenever on-board memory is accessed, the MEMEX line is activated, disabling off-board memory that occupies the same memory space. Thus a 64 K RAM card can be used as main memory without any conflict with on-board memory. It should be noted, however, that the memory cards that you use should comply with the STD specifications with respect to MEMEX if you wish to use overlapping on-board memory. (A few manufacturers have "forgotten" MEMEX in their design.)

2.5 BOOT CIRCUIT

The boot circuit allows a program on disc, cassette, etc. to be loaded into memory. When power is switched on or when a reset pulse is generated, the boot flip flop (U2 pin 6) outputs a high logic level. If J6 pins 8 and 10 are connected, this level is routed to U7 pin 19.

This selects a different set of memory maps at U7 as discussed in the previous section. This memory mapping will enable a PROM at address zero which can be used (in most cases) to boot a program from, say, disc into memory.

Once the program in the boot PROM has been executed, the PROM can be switched off via an I/O operation. (It can also be switched back on with an I/O operation). Two memory map options are provided for booting, both of which are starting at address zero. This again will satisfy most users. If desired, the user may custom program his own memory map PROM, in which case 8 different maps can be programmed.

2.6 ON-BOARD I/O

Two LSI chips, the 8251 and Z80-CTC, provide serial I/O and timer/counter functions, respectively. The 8251 is a general purpose, single channel USART capable of being programmed to operate in most serial data transmission techniques presently in use. The SB8010 card supports only asynchronous transmission and reception. Most asynchronous baud rates from 50 to 4800 are programmable. Data and control signals CTS, RTS and DTR are buffered using RS-232 line drivers and receivers. The serial I/O signals are available at connector J2. The pins are arranged so that a standard RS232 "D" type connector can be directly connected to J2 using flat ribbon cable.

The Z80-CTC is a counter/timer chip having four independent channels, each with a readable 8 bit downcounter and a selectable 16 or 256 prescaler (for timer mode only). Three channels have "zero count" outputs. All channels have clock/trigger inputs that start the timer in the timer mode or decrement the counter in the counter mode. The 4 clock/trigger inputs and the 3 "zero count" outputs are available unbuffered at connector J3.

When using the 8251, one channel of the CTC has to be programmed to provide the receive and transmit clocks. The other three channels may be used as general purpose timers or counters. Since the CTC is a Z-80 peripheral, a common use of it is as an interrupt controller (3 channels in this case). This is done by programming a channel to operate in the counter mode with a time constant of 1. If interrupts are enabled, the first pulse at the respective trigger input will generate an interrupt. The 8251, not being a Z-80 peripheral can thus use the CTC to operate as an interrupt driven serial I/O. The CTC is placed at the top of the interrupt priority chain. (Refer to Section 2.7). I/O address decoding is done using a PROM. The card uses 8 I/O addresses starting at address F0 HEX. If these I/O addresses interfere with other I/O addresses, the user may program his own PROM. Appendix F contains the data programmed into the I/O decoder PROM.

2.7 INTERRUPT SYSTEM

This section will discuss briefly the Z-80 interrupt system and the interrupt management particular to this card. A more thorough Z-80 interrupt description is widely available in manufacturer publications and the unfamiliar user should refer to those.

There are three modes of maskable interrupts programmable by software: Mode 0 - the interrupting device places an instruction on the data bus. This is normally a restart instruction which will initiate a call to one of 8 restart locations. Mode 1 - In this mode no vector is placed on the bus and the CPU goes to location 0038H. Mode 2 - In this mode the CPU uses the byte transferred by the peripheral as the lower byte of a pointer directing the CPU to a table of service routine addresses. The high byte of the pointer is furnished by the previously loaded interrupt register. The CPU then jumps to the routine at that address.

There is also one non-maskable interrupt input. Here the CPU performs a restart 0066H.

No interrupt is recognized when BUSREQ/ signal is active.

Z-80 peripherals resolve interrupt priority by means of their own logic, unlike 8080/8085 peripherals that require an interrupt controller. Each peripheral has IEI (Interrupt Enable Inable) and IEO (Interrupt Enable Output). These pins are daisy chained by connecting them as shown in figure 2.1.

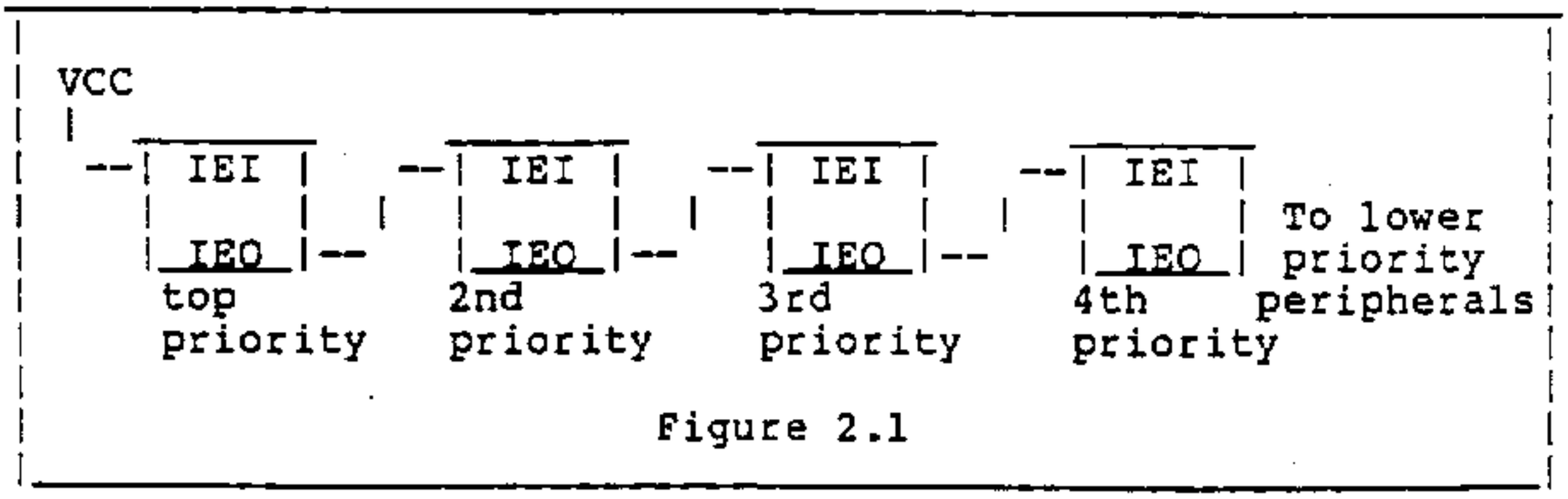


Figure 2.1

A device will not request an interrupt if its IEI is low in which case it will also turn its IEO low. It will also turn IEO low when it requests

an interrupt itself. Thus IEO/IEI propagate downwards, disabling all the lower priority interrupts. When servicing the interrupt the CPU automatically disables all other interrupts unless an "enable interrupt" command has been issued in the interrupt service routine (which will enable nesting the interrupts). Upon completion of the service routine an RETI instruction is executed. This instruction is recognized by the interrupting peripheral causing it to turn IEO high again (if no higher priority requests an interrupt at this instance).

As outlined in Section 2.6 the time/counter chip can be used as an interrupt controller to service non A-80 peripherals (e.g., the on-board 8251 and/or external events). If the serial chip is not used, up to four interrupt channels are available, otherwise only three interrupt channels are available. (One channel is used as a baud rate generator for the serial chip.) To use a channel for generating interrupts, the CTC is programmed for the counter mode with interrupts enabled, a value of 1 is loaded as a time constant, and an interrupt vector is loaded. When a level changes at the TRG/CLK input, the counter will decrement to zero and issue an interrupt. Channel 0 is the highest priority and channel 3 is the lowest. The IEI pin of the CTC is tied high, thus placing it on top of the priority chain.

3.0 CONFIGURATION AND OPTIONS

3.1 EXTERNAL CONNECTORS

There are three external connectors - J1, J2, and J3. J1 is the STD bus edge connector. For pinout description refer to STD bus specifications. J2 and J3 are the serial I/O and the timing/counting I/O interface respectively. Tables 3.7 and 3.8 describe J2 and J3 respectively. Table 3.1 provides connector part numbers from ANSLEY as a reference. Similar connectors from different manufacturers may be used.

TABLE 3.1 SUGGESTED CONNECTORS

CONNECTOR	NO. OF PINS	ANSLEY P/N
J2	14	609-1401M
J3	16	609-1601M

3.2 ON-BOARD MEMORY

3.2.1 RESTRICTIONS

On-board 2K, 4K, and 8K memory chips of any kind may be used provided their access time from chip enable or chip select (not output enable!) does not exceed 215ns for a 4MHZ CPU card version or 405ns for a 2.5MHZ CPU card version. Wait states cannot be generated on-board to alleviate this restriction.

3.2.2 CONFIGURING THE MEMORY SOCKETS

The memory sockets are 28 pin universal sockets capable of accepting various 24 or 28 pin memory chips. The following chips may be used: Static RAMS and ROM/PROM/EPROM chips of 2K x 8, 4K x 8 and 8K x 8. To configure a socket for a chip, follow the following procedure:

a. Determine which socket you need to place the chip in by selecting the appropriate memory map to be used. (Refer to Table 3.7).

b. Determine which option post group corresponds to that socket by using Table 3.2.

TABLE 3.2 OPTION POSTS TO SOCKET DESIGNATION

SOCKET	CORRESPONDING OPTION POSTS
U13	J7
U14	J8
U15	J9
U16	J10

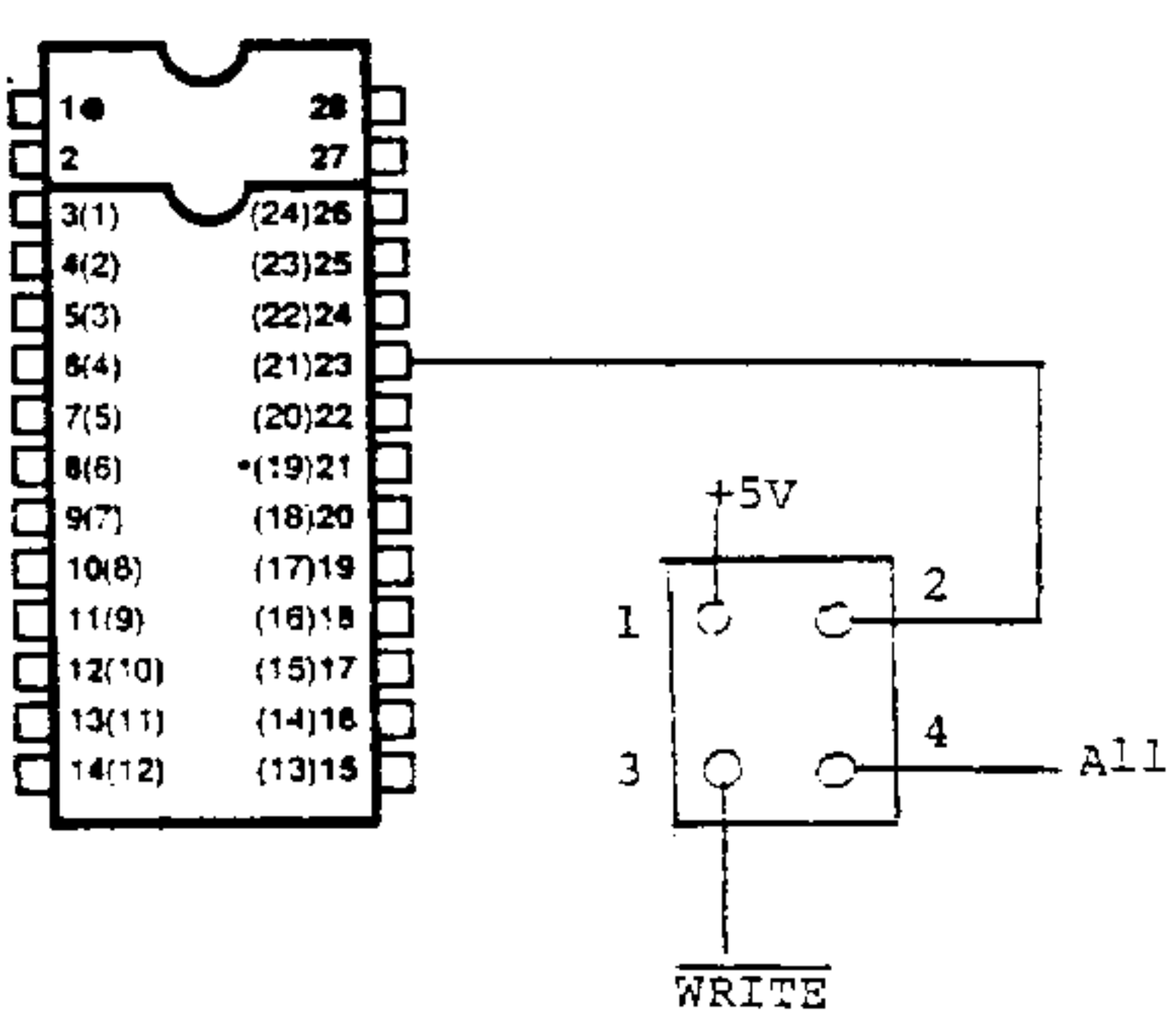
c. Use Table 3.3 to determine the connection at the option post from step b.

TABLE 3.3 CONFIGURATION OF THE OPTION POST (from b)

MEMORY SIZE / TYPE	2KX8	4KX8	8KX8
RAM	3&2	2&4	2&4
ROM/PROM/ EPROM	1&2	2&4	2&4

The memory chip is then placed in the lower part of the socket if it is a 24 pin device. For further clarification refer to Figure 3.1

FIG. 3.1 A TYPICAL SOCKET AND ITS CORRESPONDING OPTION POST



SOCKETS:
U13,U14,U15,U16

OPTION POSTS:
J7,J8,J9,J10

3.2.3 ENABLING THE SOCKETS

Each chip can be individually enabled by a connection at J5. With any memory map selection (Table 3.7) the user may select only part of the memory map by not connecting all the jumpers at J5. For example, the user can have an 8K PROM located between 4000H and 6000H-1 by selecting map option 7 (Table 3.7) but only enabling U15 by connecting J5 pins 5 and 6. So only at these addresses will there be an on-board access, and external overlapping memory will be masked by MEMEX.

Refer to Table 3.4 to determine connections at J5.

TABLE 3.4 enabling the memory sockets

TO ENABLE SOCKET:	CONNECT J5 PINS:
U13	1 & 2
U14	3 & 4
U15	5 & 6
U16	7 & 8

3.2.4 MEMORY MAPS AND BOOT OPTION

There are two different sets of memory maps selectable via the boot flip-flop if J6 pins 8 and 10 are connected. If instead, J6 pins 7 and 8 are connected the boot flip-flop does not play any role and only the first set (Set #1) of memory maps are available. In either case the particular map within a set is selectable on J6.

The boot flip-flop is set after power-on reset or a push button reset. This is its main function -- to enable a PROM initially so that the PROM can load an operating system. The boot flip-flop can then be reset by an I/O operation. Refer to Table 3.5.

TABLE 3.5 BOOT FLIP FLOP CONTROL

OPERATION DESIRED	DO THIS I/O OPERATION	RESULT IF J6 8 & 10 CONNECTED
SET BOOT FLIP-FLOP	OUTPUT 0 TO PORT F6H (See Note 2)	MEMORY MAP SET #2 ENABLED
RESET BOOT FLIP-FLOP	OUTPUT 1 TO PORT F6H	MEMORY MAP SET #1 ENABLED (See Note #1)

Note 1: If only memory map set #1 is desired, connect J6 7&8 in which case the boot flip-flop effect is disabled.

Note 2: Boot flip-flop is also set as a result of a power-on or push button reset.

Memory map set #1 contains 6 different memory maps that take care of various combinations of ROMS/PROMS/EPROMS and RAMS. Memory map set #2, (intended to load an operating system into main memory) contains 2 different maps. The 2 memory map options of set #2 are configured on J6 in such a way as to not have any identical configuration for set #1 on J6. Thus if a map on set #2 is selected there can only be an on-board memory operation if the boot flip-flop is on. Once it is turned off, set #1 is enabled, as explained above, but the configuration on J6 does not select any map from this set.

Table 3.6 shows how to configure J6 for selecting the particular memory map. The table contains boxes showing memory size at each of the four sockets (U13 through U16). Also absolute location in the memory space is shown.

TABLE 3.6 ON BOARD MEMORY MAP SELECTION

#	J6 PINS 1 THROUGH 6	MEMORY MAP SET #1 SELECTED WHEN J6 7&8 CONNECTED OR J6 8&10 CONNECTED BUT BOOT FLIP FLOP OFF	MEMORY MAP SET #2 SELECTED WHEN J6 8&10 CONNECTED AND BOOT FLIP FLOP ON
0	1&2, 3&4, 5&6		U13: 4K
1	1&2, 3&4		U13: 4K; U14: 2K
2	1&2, 5&6	U13: 2K; U14: 2K; U15: 2K; U16: 2K	
3	1&2	U13: 4K; U14: 4K; U15: 2K; U16: 2K	
4	3&4, 5&6	U13: 4K; U14: 4K; U15: 4K; U16: 4K	
5	3&4	U13: 8K; U14: 8K; U15: 2K; U16: 2K	
6	5&6	U13: 8K; U14: 8K; U15: 4K; U16: 4K	
7	NONE	U13: 8K; U14: 8K; U15: 8K; U16: 8K	

MEMORY SPACE 0 4K 8K 12K 16K 20K 24K 28K 0 4K 8K 12K 16K 20K 24K 28K

3.3 MEMEX AND IOEXP

3.3.1 MEMEX

Option posts E and F are connected when this card is to drive the MEMEX line. This card must drive the MEMEX line if there is an on-board accessible memory or if no other card in the system drives MEMEX.

3.3.2 IOEXP

Option posts D and C must normally be connected. It can be left open for the rare occasion that an external circuit would control this line.

3.4 ON-BOARD I/O

The on-board CTC (counter/timer controller) is used partly to provide the baud rate for the on-board serial I/O chip. (Channel zero of the CTC is dedicated for this purpose.) Refer to the programming section for further detail. Also refer to the programming section for the I/O addresses. The flip-flop is also an I/O addressable device; it is completely described in section 3.2.3.

3.4.1 OUTPUT CONNECTORS

Tables 3.7 and 3.8 show the pin assignments for the output connectors J2 and J3, used for the serial I/O and the timer/counter controller respectively.

3.4.2 I/O CONFIGURATIONS

The timer/counter controller can be configured on option post J4 to accept at its four CLK/TRG inputs one of six possible on-board sources. Three of the six sources are the CTC's own ZC/TO outputs so that the counters can be cascaded. Two of the sources are the 8251's (the serial chip) TXRDY and RXRDY signals for interrupt operation of the 8251. See section 2.6. The remaining source is the maximum frequency that the CTC can count -- 1/2 CPU clock. Table 3.9 provides the pin description of J4.

TABLE 3.7 OUTPUT CONNECTOR J2 - 14 pins
(SERIAL I/O CONNECTOR)

PIN	SIGNAL	DESCRIPTION
1, 13	GND	For use as ground reference
3	$\overline{\text{TXD}}$	Transmitted Data see note
5	$\overline{\text{RXD}}$	Received Data " "
7	RTS	Request-to-send " "
9	CTS	Clear-to-send " "
14	DTR	Data terminal ready " "
All other	NC	No connection

Note: This card is configured as DCE thus all signals are referenced to the opposite side (DTE).

TABLE 3.8 OUTPUT CONNECTOR J3 - 16 pins
(TIMER/COUNTER CONNECTOR)

PIN	SIGNAL	DESCRIPTION
2	CLK/TRG0	Trigger or clock input to channel zero of CTC
4	CLK/TRG1	Trigger or clock input to channel one of CTC
6	CLK/TRG2	Trigger or clock input to channel two of CTC
8	CLK/TRG3	Trigger or clock input to channel three of CTC
10	ZC/TO0	Zero count or time-out output channel zero of CTC
12	ZC/TO1	Zero count or time-out output channel one of CTC
14	ZC/TO2	Zero count or time-out output channel two of CTC
16	NC	No connection
All Odd Pins	GND	

TABLE 3.9 CONFIGURATION OPTION POST J4

J4 PIN	SIGNAL	DESCRIPTION
1	CLK/TRG0	Clock/trigger input of CTC channel 1
2	1/2 CPU CLOCK	1.25 MHZ OR 2 MHZ FOR 2.5 MHZ or 4 MHZ CPU respectively
3	CLK/TRG1	Clock/trigger input of CTC Channel 1
4	ZC/TO0	Zero count/time-out of CTC Channel 9
5	CLK/TRG2	Clock/trigger input of CTC Channel 2
6	ZC/TO1	Zero count/time-out of CTC Channel 1
7	TXRDY	Output of serial chip (8251)
8	ZC/TO2	Zero count/time-out of CTC Channel 2
9	RXRDY	Output of serial chip (8251)
10	CLK/TRG3	Clock/trigger of CTC Channel 3

Note: These signals are also available at the output connector J3. see table 3.8

3.4.2.1 USING THE SERIAL CHIP

J4 pins 1 and 2 must be connected when using the serial chip since channel zero of the CTC provides its baud rate. J4 pin 4 is the output of channel zero. From here it is divided by two and made 50% duty cycle before reaching the serial chip.

Refer to Table 3.7 for the serial connector pin description. All the inputs and outputs are buffered by RS-232 buffers. The pins are conveniently positioned so that a cable with an RS-232 "D" connector at one end and a flat cable connector at the other can easily by made.

To use the serial chip inthe interrupt mode the signals RXRDY and TXRDY (J4 pins 9 and 7 respectively) are connected each to one of the three available CLK/TRG signals (J4 pins 3, 4, or 10). Also jumpers A and B must be connected. Refer to the programming section and Section 2.6.

3.4.2.2 USING THE CTC TO CONTROL EXTERNAL INTERRUPTS

The available CLK/TRG inputs of the CTC are connected via the output connector J3 to the interrupting source. Also jumpers A and B must be connected. It should be noted that these inputs are not buffered and go directly to the MOS inputs of the CTC. For further clarification refer to Section 2.6 and the programming section.

4.0 PROGRAMMING

This section provides I/O register addresses for the serial chip and the CTC, as well as guidelines to programming the CTC to provide the baud rate for the serial chip and interrupt control.

This section will not serve to instruct the user how to program the three LSI chips on-board. For this purpose the user should refer to the attached manufacturer's data sheets in appendices B, C, and E as well as other manufacturer publications.

Programming the boot flip-flop is not given here as it was more conveniently presented in Section 3.2.3.

4.1 I/O ADDRESSES

4.1.1 SERIAL CHIP ADDRESSES

The serial chip has two addresses selected according to the C/D (Command/Data) input being '1' or '0'. Refer to Table 4.1.

TABLE 4.1 - Serial Chip (8251) Register Address

ADDRESS	$C/\overline{D}$	REGISTER
F4	0	DATA
F5	1	CONTROL/STATUS

4.1.2 CTC CHIP ADDRESSES

The CTC has four addresses. Each address selects a different channel. Refer to Table 4.2

TABLE 4.2 CTC addresses

ADDRESS	CHANNEL
F0	0
F1	1
F2	2
F3	3

4.2 SERIAL CHIP BAUD RATE PROGRAMMING

CHANNEL 0 of the CTC is used to generate the baud rate for the serial chip. All the standard asynchronous baud rates from 50 to 4800 can be programmed (except 50 baud for a 4MHZ CPU). Use Table 4.3 to determine the CTC time constant and the serial chip baud rate factor for a particular baud rate selection.

TABLE 4.3 BAUD RATE SELECTION

BAUD RATE DESIRED	CTC CHANNEL TIME CONSTANT		SERIAL CHIP BAUD RATE FACTOR	ACTUAL BAUD RATE	
	2.5 MHz CPU	4 MHZ CPU		2.5 MHz MHz	4 MHz CPU
50	195	---	X64	50.08	---
110	89	142	X64	109.73	110.04
150	65	104	X64	150.24	150.24
300	32	52	X64	305.18	300.48
600	65	104	X16	600.96	600.96
1200	32	52	X16	1220.70	1201.92
2400	16	26	X16	2441.41	2403.85
4800	8	13	X16	4882.80	4807.69
9600	4	___	X16	9765.60	---

4.2.1 GUIDELINES FOR BAUD RATE PROGRAMMING

The following two operations on the CTC set the baud rate:

(1) Output to port -- 01XXX111 disable interrupts/counter mode/time constant follows/reset/control word
F0 (channel 0)

(2) Output to port
F0 (channel 0) -- TIME CONSTANT (From Table 4.3)

What still remains in order to set the baud rate is to program the serial chip for the desired baud rate factor (16x or 64x) according to Table 4.3.

4.3 USING THE CTC AS AN INTERRUPT CONTROLLER

The CTC can be used as an interrupt controller for the serial chip and/or external events. See Section 3.4.2 for configurations. The following set of operations accomplishes this (for a particular channel).

(1)	Output to selected channel	11X*X111	interrupt enable / counter mode / time constant follows / reset / control word
(2)	Output to selected channel	-- 00000001	time constant of 1
(3)	Output to channel 0(F0)**	VVVVV000	five significant bits of the interrupt vector

X - Don't care
* - In the case of the serial chip this should be 1. This will trigger an interrupt on the rising edge of a signal.

** - Only one interrupt vector is used for all the channels and is loaded at channel 0.

For further clarification refer to Appendix D -- Manufacturer's Data sheet

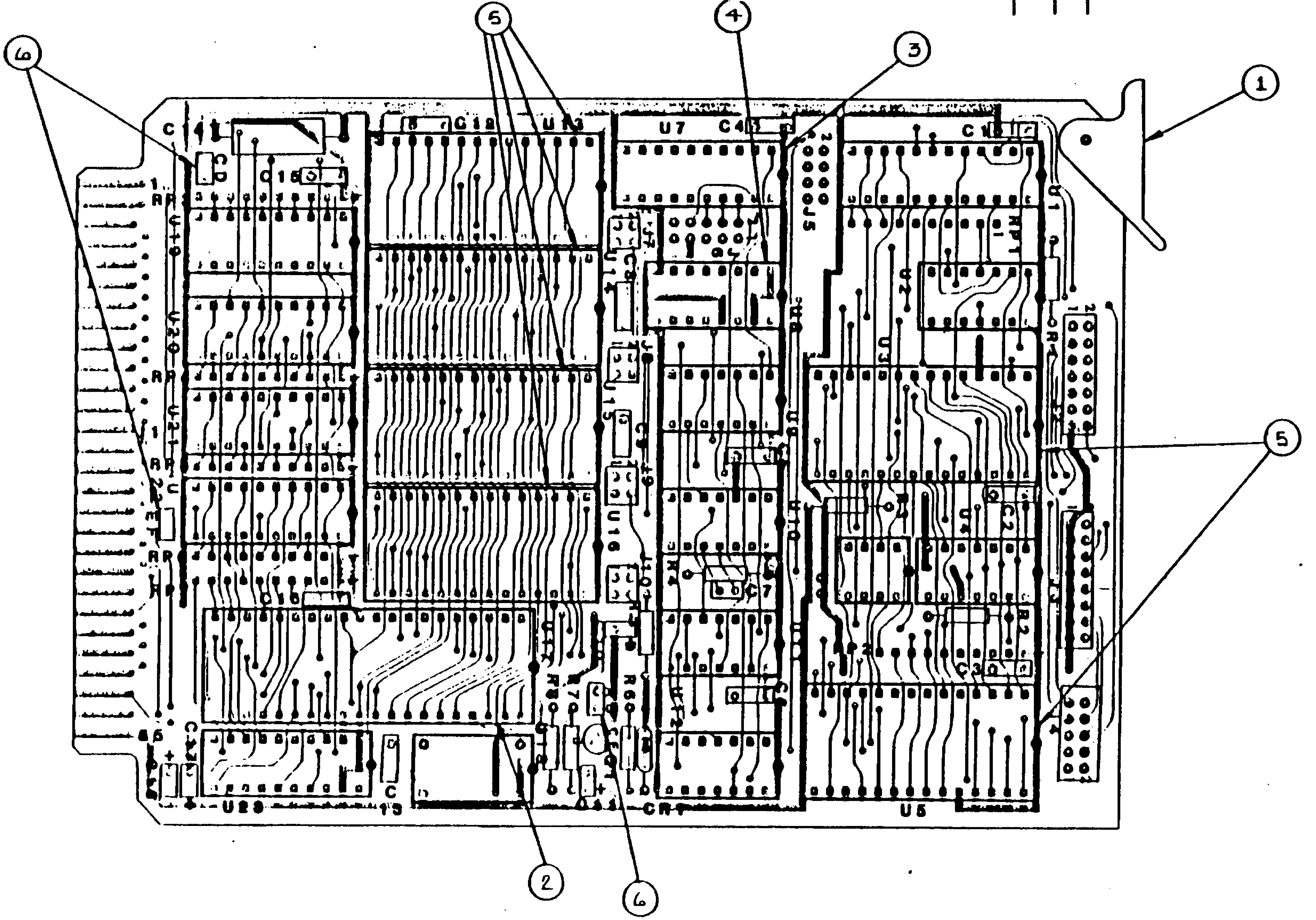
4.4 RESETTING THE SERIAL CHIP

For a safe reset of the serial chip always load the following sequence to port F5 (C/D=1) rather than just the reset command itself: 00H, 00H, 00H, and 40H. A software reset 40H on its own does not always work.

A P P E N D I C E S

A	ASSEMBLY, SCHEMATICS DIAGRAMS
B	Z-80 DATA SHEET
C	SERIAL CHIP - 8251 - DATA SHEETS
D	CTC (COUNTER/TIMER CHIP) DATA SHEETS
E	MEMORY DECODER PROM
F	I/O DECODER PROM
G	PAL (PROGRAMMABLE LOGIC ARRAY) EQUATIONS

ZONE	LTR	REVISIONS DESCRIPTION	DATE	APPROVED



SEE SEPARATE PARTS LIST

	QTY REQD	CODE IDENT	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	
PARTS LIST					

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE:
FRACTIONS ± DECIMALS XX ± XXX ± ANGLES ±

MATERIAL

FINISH

DO NOT SCALE DRAWING

NEXT ASSY	USED ON
APPLICATION	

CONTRACT NO

APPROVALS		DATE
DRAWN	CIRTECH, INC.	2/83
CHECKED	R. E.	2/10/83

REV. 2/10/83 R.E

microsys

ASSEMBLY DRAWING
SB 8010 Z-80 CPU CARD

SIZE	CODE IDENT NO	DRAWING NO
C		SB 8010D

SCALE 2/1 SHEET 2 OF 5

- XSB8010 DOES NOT HAVE U5, U17 & U18

NOTES: UNLESS OTHERWISE SPECIFIED

REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED

+5V
RP1 470Ω
U9 LS74
1/2 CPUCLK
U18 OSC
U12 74LS04
CLOCK
CTCCLK
U12
INT
RG 2.2K
Q1
RP7 4.7K
J1
U17 Z80
A0 - A15
U20 LS244
U21 LS244
U22 LS244
U19 LS245
RP5 RP4 4.7K
RP6 RP7
RP3 4.7K
IORQ
MREQ
WR
RD
MCSYNC
INTAK
BUSAK
M1
DATAOUT
D0 - D7
BUSRQ *
WAITRQ *
INTRQ *
NMIRQ *
SYSRESET *
BUSAK *
CLOCK *
U23 LS3244
WRITE
READ
RESET
RFSH
+12V
+5VDC
GND
-12V
C17 1μF
C14 47μF
C1-C6, C8, C9, C12, C13, C15, C16 .1μF
C18 1μF

QTY REQD	CODE IDENT	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION

PARTS LIST

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES ARE.
FRACTIONS ± DECIMALS .XX ± .XXX ± ANGLES ±

MATERIAL

FINISH

DO NOT SCALE DRAWING

CONTRACT NO.

APPROVALS	DATE
DRAWN CIRTECH, INC	12/82
CHECKED R.G.	1/11/83

REV. 2/10/83 R.G.

NEXT ASSY | USED ON
APPLICATION

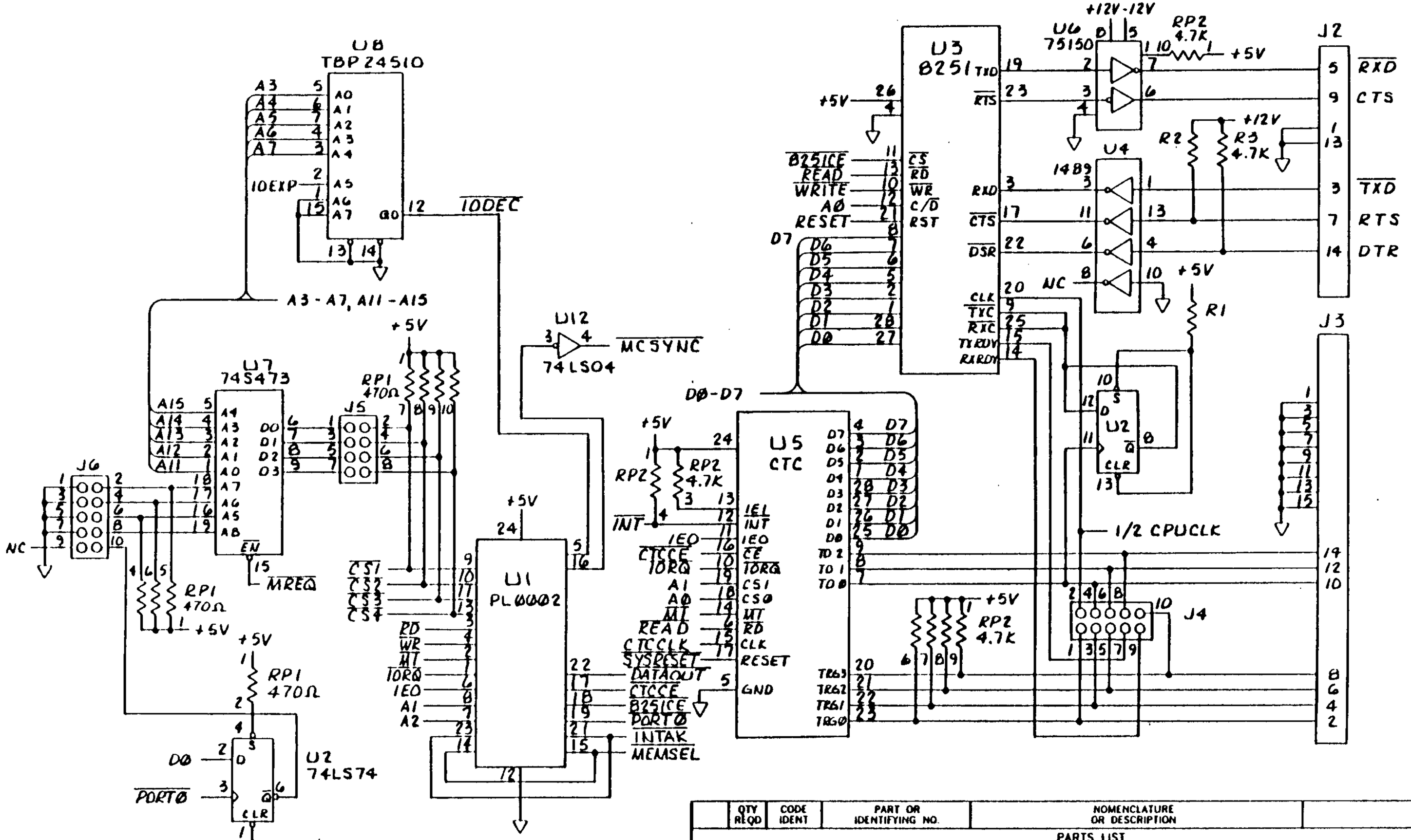
microsys

SCHEMATIC DIAGRAM - SBB010 Z-80 CPU CARD

SIZE	CODE IDENT NO	DRAWING NO
C		SBB010D

SCALE NONE | SHEET 3 OF 5

REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED



QTY REQD	CODE IDENT	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION

PARTS LIST

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES ARE.
FRACTIONS ± DECIMALS XX ± XXX ± ANGLES ±

MATERIAL

FINISH

DO NOT SCALE DRAWING

CONTRACT NO

APPROVALS	DATE
DRAWN CIRTECH,INC	12/82
CHECKED R.E.	1/11/83

REV. 2/10/83 A.E

NEXT ASSY	USED ON

APPLICATION

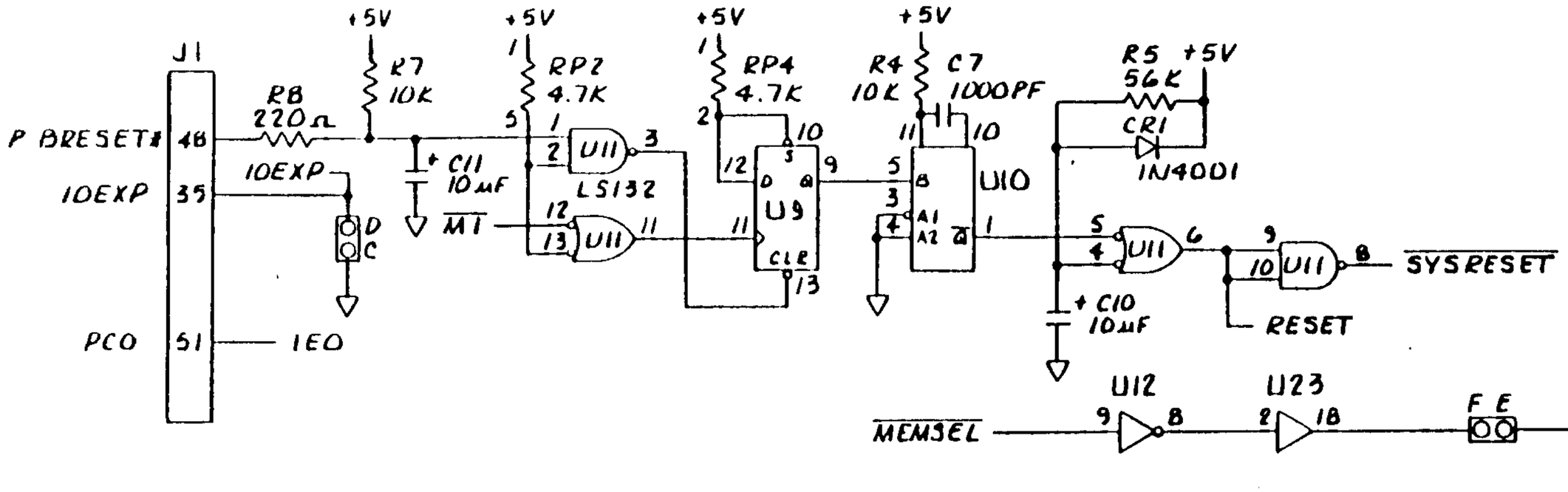
microsys

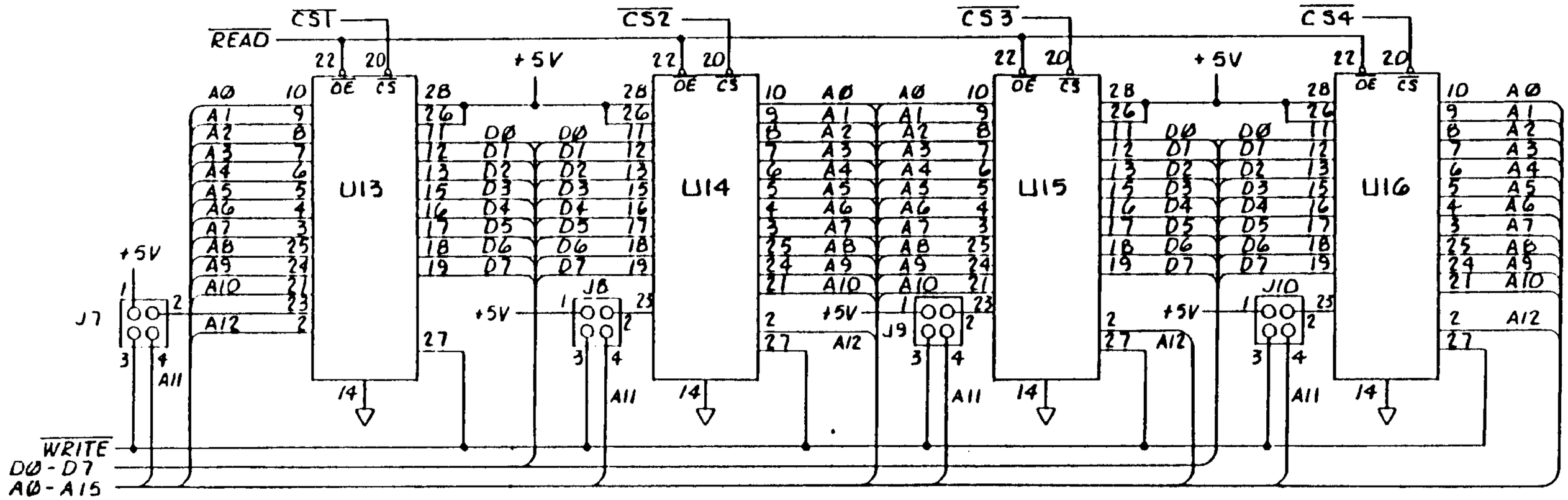
SCHEMATIC DIAGRAM -
SB8010 2-80 CPU CARD

SIZE	CODE IDENT NO	DRAWING NO
C		SB8010D

SCALE NONE SHEET 4 OF 5

REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED





QTY REQD	CODE IDENT	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION

PARTS LIST

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES ARE:
FRACTIONS ± DECIMALS .XX ± .XXX ± ANGLES ±

MATERIAL

FINISH

DO NOT SCALE DRAWING

CONTRACT NO.

APPROVALS	DATE
DRAWN CIRTECH, INC	12/82
CHECKED R.F.	1/31/83

REV. 2/10/83 R.F.

microsys

SCHEMATIC DIAGRAM -
SB8010 Z-80 CPU CARD

SIZE	CODE IDENT NO	DRAWING NO
C		SBP 10D

SCALE NONE SHEET 5 OF 5

NEXT ASSY	USED ON

APPLICATION

A P P E N D I C E S

A ASSEMBLY, SCHEMATICS DIAGRAMS
B Z-80 DATA SHEET
C SERIAL CHIP - 8251 - DATA SHEETS
D CTC (COUNTER/TIMER CHIP) DATA SHEETS
E MEMORY DECODER PROM
F I/O DECODER PROM
G PAL (PROGRAMMABLE LOGIC ARRAY) EQUATIONS

Z8400/Z84C00 NMOS/CMOS Z80® CPU Central Processing Unit

FEATURES

The extensive instruction set contains 158 instructions, including the 8080A instruction set as a subset.

- NMOS version for low cost high performance solutions, CMOS version for high performance low power designs.
- NMOS Z0840004 - 4 MHz, Z0840006 - 6.17 MHz, Z0840008 - 8 MHz.
- CMOS Z84C0006 - DC to 6.17 MHz, Z84C008 - DC to 8 MHz, Z84C0010 - DC to 10 MHz, Z84C0020 - DC - 20 MHz
- 6 MHz version can be operated at 6.144 MHz clock.
- The Z80 microprocessors and associated family of peripherals can be linked by a vectored interrupt system. This system can be daisy-chained to allow implementation of a priority interrupt scheme.
- Duplicate set of both general-purpose and flag registers.
- Two sixteen-bit index registers.
- Three modes of maskable interrupts:
  Mode 0—8080A similar;
  Mode 1—Non-Z80 environment, location 38H;
  Mode 2—Z80 family peripherals, vectored interrupts.
- On-chip dynamic memory refresh counter.

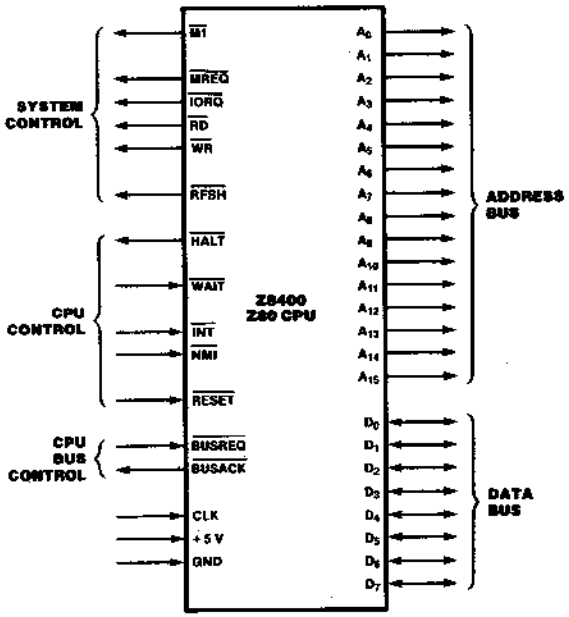


Figure 1. Pin Functions

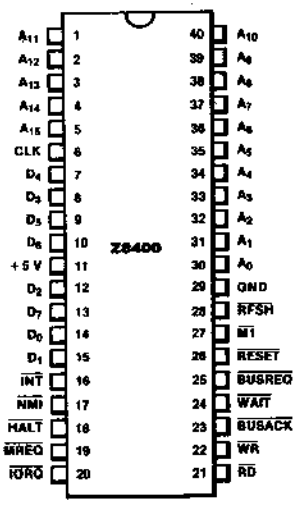
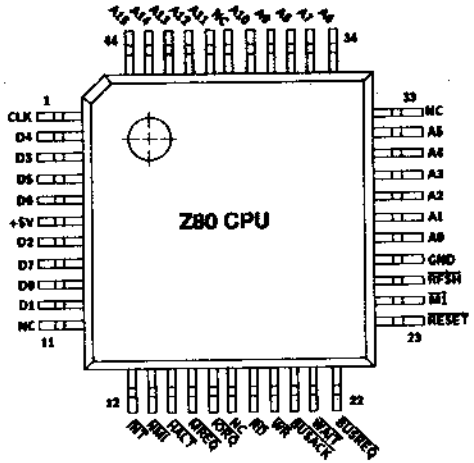


Figure 2. 40-pin Dual-In-Line (DIP), Pin Assignments



44 pin Quad Flat Pack (QFP), Pin Assignments (Only available for 84C00)

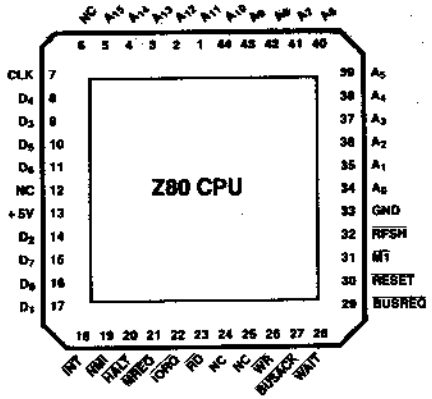


Figure 2b. 44-Pin Chip Carrier Pin Assignments

GENERAL DESCRIPTION

The CPUs are fourth-generation enhanced microprocessors with exceptional computational power. They offer higher system throughput and more efficient memory utilization than comparable second- and third-generation microprocessors. The internal registers contain 208 bits of read/write memory that are accessible to the programmer. These registers include two sets of six general-purpose registers which may be used individually as either 8-bit registers or as 16-bit register pairs. In addition, there are two sets of accumulator and flag registers. A group of "Exchange" instructions makes either set of main or alternate registers accessible to the programmer. The alternate set allows operation in foreground-background mode or it may be reserved for very fast interrupt response.

The CPU also contains a Stack Pointer, Program Counter, two index registers, a Refresh register (counter), and an Interrupt register. The CPU is easy to incorporate into a system since it requires only a single +5V power source. All output signals are fully decoded and timed to control standard memory or peripheral circuits; the CPU is supported by an extensive family of peripheral controllers. The internal block diagram (Figure 3) shows the primary functions of the processors. Subsequent text provides more detail on the I/O controller family, registers, instruction set, interrupts and daisy chaining, and CPU timing.

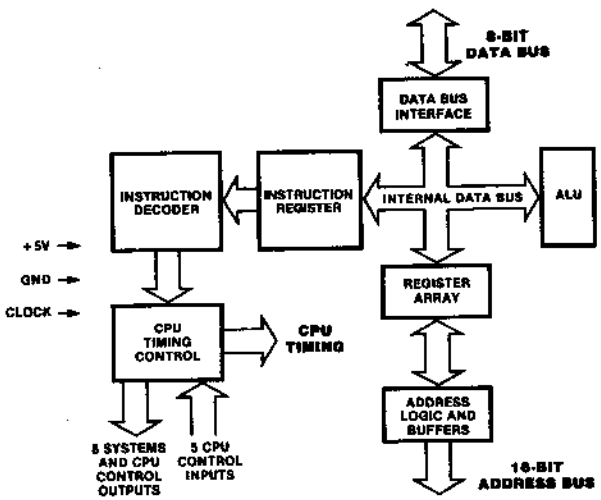


Figure 3. Z80C CPU Block Diagram

Table 1. Z80C CPU Registers

Register		Size (Bits)	Remarks
A, A'	Accumulator	8	Stores an operand or the results of an operation.
F, F'	Flags	8	See Instruction Set.
B, B'	General Purpose	8	Can be used separately or as a 16-bit register with C.
C, C'	General Purpose	8	Can be used separately or as a 16-bit register with C.
D, D'	General Purpose	8	Can be used separately or as a 16-bit register with E.
E, E'	General Purpose	8	Can be used separately or as a 16-bit register with E.
H, H'	General Purpose	8	Can be used separately or as a 16-bit register with L.
L, L'	General Purpose	8	Can be used separately or as a 16-bit register with L.
			Note: The (B,C), (D,E), and (H,L) sets are combined as follows: B — High byte C — Low byte D — High byte E — Low byte H — High byte L — Low byte
I	Interrupt Register	8	Stores upper eight bits of memory address for vectored interrupt processing.
R	Refresh Register	8	Provides user-transparent dynamic memory refresh. Automatically incremented and placed on the address bus during each instruction fetch cycle.
IX	Index Register	16	Used for indexed addressing.
IY	Index Register	16	Used for indexed addressing
SP	Stack Pointer	16	Holds address of the top of the stack. See Push or Pop in instruction set.
PC	Program Counter	16	Holds address of next instruction.
IFF_1-IFF_2	Interrupt Enable	Flip-Flops	Set or reset to indicate interrupt status (see Figure 4).
IMFa-IMFb	Interrupt Mode	Flip-Flops	Reflect Interrupt mode (see Figure 4).

failure has been detected. After recognition of the $\overline{NMI}$ signal (providing $\overline{BUSREQ}$ is not active), the CPU jumps to restart location 0066H. Normally, software starting at this address contains the interrupt service routine.

Maskable Interrupt ($\overline{INT}$). Regardless of the interrupt mode set by the user, the CPU response to a maskable interrupt input follows a common timing cycle. After the interrupt has been detected by the CPU (provided that interrupts are enabled and $\overline{BUSREQ}$ is not active) a special interrupt processing cycle begins. This is a special fetch ($\overline{M1}$) cycle in which $\overline{IORQ}$ becomes active rather than $\overline{MREQ}$, as in a normal $\overline{M1}$ cycle. In addition, this special $\overline{M1}$ cycle is automatically extended by two $\overline{WAIT}$ states, to allow for the time required to acknowledge the interrupt request.

Mode 0 Interrupt Operation. This mode is similar to the 8080 microprocessor interrupt service procedures. The interrupting device places an instruction on the data bus. This is normally a Restart instruction, which will initiate a call **to the selected one of eight restart locations in page zero of memory. Unlike the 8080, the Z80 CPU responds to the Call instruction with only one interrupt acknowledge cycle followed by two memory read cycles.**

Mode 1 Interrupt Operation. Mode 1 operation is very similar to that for the $\overline{NMI}$. The principal difference is that the Mode 1 interrupt has only one restart location, 0038H.

Mode 2 Interrupt Operation. This interrupt mode has been designed to most effectively utilize the capabilities of the Z80 microprocessor and its associated peripheral family. The interrupting peripheral device selects the starting address of the interrupt service routine. It does this by placing an 8-bit vector on the data bus during the interrupt acknowledge cycle. The CPU forms a pointer using this byte as the lower 8 bits and the contents of the I register as the upper 8 bits. This points to an entry in a table of addresses for interrupt service routines. The CPU then jumps to the routine at that

address. This flexibility in selecting the interrupt service routine address allows the peripheral device to use several different types of service routines. These routines may be located at any available location in memory. Since the interrupting device supplies the low-order byte of the 2-byte vector, bit 0 (A_0) must be a zero.

Interrupt Enable/Disable Operation. Two flip-flops, IFF_1 and IFF_2, referred to in the register description, are used to signal the CPU interrupt status. Operation of the two flip-flops is described in Table 2. For more details, refer to the *Z80 CPU Technical Manual* (03-0029-01) and *Z80 Assembly Language Programming Manual* (03-0002-01).

Table 2. State of Flip-Flops

Action	IFF_1	IFF_2	Comments
CPU Reset	0	0	Maskable interrupt $\overline{INT}$ disabled
DI instruction execution	0	0	Maskable interrupt $\overline{INT}$ disabled
EI instruction execution	1	1	Maskable interrupt $\overline{INT}$ enabled
LD A,I instruction execution	•	•	IFF_2 → Parity flag
LD A,R instruction execution	•	•	IFF_2 → Parity flag
Accept $\overline{NMI}$	0	•	Maskable interrupt $\overline{INT}$ disabled
RETN instruction execution	IFF_2	•	IFF_2 → IFF_1 at completion of an $\overline{NMI}$ service routine.

INSTRUCTION SET

The microprocessor has one of the most powerful and versatile instruction sets available in any 8-bit microprocessor. It includes such unique operations as a block move for fast, efficient data transfers within memory, or between memory and I/O. It also allows operations on any bit in any location in memory.

The following is a summary of the instruction set which shows the assembly language mnemonic, the operation, the flag status, and gives comments on each instruction. For an explanation of flag notations and symbols for mnemonic tables, see the Symbolic Notations section which follows these tables. The *Z80 CPU Technical Manual* (03-0029-01), the *Programmer's Reference Guide* (03-0012-03), and *Assembly Language Programming Manual* (03-0002-01) contain significantly more details for programming use.

The instructions are divided into the following categories:

- 8-bit loads
- 16-bit loads
- Exchanges, block transfers, and searches
- 8-bit arithmetic and logic operations
- General-purpose arithmetic and CPU control
- 16-bit arithmetic operations
- Rotates and shifts
- Bit set, reset, and test operations
- Jumps
- Calls, returns, and restarts
- Input and output operations

A variety of addressing modes are implemented to permit efficient and fast data transfer between various registers, memory locations, and input/output devices. These addressing modes include:

- Immediate
- Immediate extended
- Modified page zero
- Relative
- Extended
- Indexed
- Register
- Register indirect
- Implied
- Bit

8-BIT LOAD GROUP

Mnemonic	Symbolic Operation	Flags S	Z	H	P/V	N	C	Opcode 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD r, r'	r ← r'	•	•	X	•	X	•	•	•	01	r	r'		1	1	4	r, r' Reg.
LD r, n	r ← n	•	•	X	•	X	•	•	•	00	r	110		2	2	7	000 B
											← n →						001 C
LD r, (HL)	r ← (HL)	•	•	X	•	X	•	•	•	01	r	110		1	2	7	010 D
LD r, (IX + d)	r ← (IX + d)	•	•	X	•	X	•	•	•	11	011	101	DD	3	5	19	011 E
										01	r	110					100 H
											← d →						101 L
LD r, (IY + d)	r ← (IY + d)	•	•	X	•	X	•	•	•	11	111	101	FD	3	5	19	111 A
										01	r	110					
											← d →						
LD (HL), r	(HL) ← r	•	•	X	•	X	•	•	•	01	110	r		1	2	7	
LD (IX + d), r	(IX + d) ← r	•	•	X	•	X	•	•	•	11	011	101	DD	3	5	19	
										01	110	r					
											← d →						
LD (IY + d), r	(IY + d) ← r	•	•	X	•	X	•	•	•	11	111	101	FD	3	5	19	
										01	110	r					
											← d →						
LD (HL), n	(HL) ← n	•	•	X	•	X	•	•	•	00	110	110	36	2	3	10	
											← n →						
LD (IX + d), n	(IX + d) ← n	•	•	X	•	X	•	•	•	11	011	101	DD	4	5	19	
										00	110	110	36				
											← d →						
											← n →						

8-BIT LOAD GROUP (Continued)

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD (IY+d), n	(IY+d) ← n	•	•	X	X	•	•	11	111	101	FD	4	5	19	
								00	110	110	36				
									← d →						
									← n →						
LD A, (BC)	A ← (BC)	•	•	X	X	•	•	00	001	010	0A	1	2	7	
LD A, (DE)	A ← (DE)	•	•	X	X	•	•	00	011	010	1A	1	2	7	
LD A, (nn)	A ← (nn)	•	•	X	X	•	•	00	111	010	3A	3	4	13	
									← n →						
									← n →						
LD (BC), A	(BC) ← A	•	•	X	X	•	•	00	000	010	02	1	2	7	
LD (DE), A	(DE) ← A	•	•	X	X	•	•	00	010	010	12	1	2	7	
LD (nn), A	(nn) ← A	•	•	X	X	•	•	00	110	010	32	3	4	13	
									← n →						
									← n →						
LD A, I	A ← I	‡	‡	X	X	IFF	0	•	11	101	101	ED	2	2	9	
									01	010	111	57			
LD A, R	A ← R	‡	‡	X	X	IFF	0	•	11	101	101	ED	2	2	9	
									01	011	111	5F			
LD I, A	I ← A	•	•	X	X	•	•	•	11	101	101	ED	2	2	9	
									01	000	111	47			
LD R, A	R ← A	•	•	X	X	•	•	•	11	101	101	ED	2	2	9	
									01	001	111	4F			

NOTE: IFF, the content of the interrupt enable flip-flop, (IFF_2), is copied into the P/V flag.

16-BIT LOAD GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD dd, nn	dd ← nn	•	•	X	X	•	•	00	dd0	001		3	3	10	dd Pair
									← n →						00 BC
									← n →						01 DE
LD IX, nn	IX ← nn	•	•	X	X	•	•	11	011	101	DD	4	4	14	10 HL
								00	100	001	21				11 SP
									← n →						
									← n →						
LD IY, nn	IY ← nn	•	•	X	X	•	•	11	111	101	FD	4	4	14	
								00	100	001	21				
									← n →						
									← n →						
LD HL, (nn)	H ← (nn+1)	•	•	X	X	•	•	•	00	101	010	2A	3	5	16	
	L ← (nn)									← n →					
										← n →					
LD dd, (nn)	dd_H ← (nn+1)	•	•	X	X	•	•	•	11	101	101	ED	4	6	20	
	dd_L ← (nn)								01	dd1	011				
										← n →					
										← n →					

NOTE: $(PAIR)_H$, $(PAIR)_L$ refer to high order and low order eight bits of the register pair respectively. e.g., BC_L = C, AF_H = A.

16-BIT LOAD GROUP (Continued)

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD IX, (nn)	$IX_H \leftarrow (nn+1)$ $IX_L \leftarrow (nn)$	•	•	X	X	•	•	11 00	011 101 ←n→ ←n→	101 010	DD 2A	4	6	20	
LD IY, (nn)	$IY_H \leftarrow (nn+1)$ $IY_L \leftarrow (nn)$	•	•	X	X	•	•	11 00	111 101 ←n→ ←n→	101 010	FD 2A	4	6	20	
LD (nn), HL	$(nn+1) \leftarrow H$ $(nn) \leftarrow L$	•	•	X	X	•	•	00	100 ←n→ ←n→	010	22	3	5	16	
LD (nn), dd	$(nn+1) \leftarrow dd_H$ $(nn) \leftarrow dd_L$	•	•	X	X	•	•	11 01	101 dd0 ←n→ ←n→	101 011	ED	4	6	20	
LD (nn), IX	$(nn+1) \leftarrow IX_H$ $(nn) \leftarrow IX_L$	•	•	X	X	•	•	11 00	011 100 ←n→ ←n→	101 010	DD 22	4	6	20	
LD (nn), IY	$(nn+1) \leftarrow IY_H$ $(nn) \leftarrow IY_L$	•	•	X	X	•	•	11 00	111 100 ←n→ ←n→	101 010	FD 22	4	6	20	
LD SP, HL	$SP \leftarrow HL$	•	•	X	X	•	•	11	111	001	F9	1	1	6	
LD SP, IX	$SP \leftarrow IX$	•	•	X	X	•	•	11 11	011 111	101 001	DD F9	2	2	10	
LD SP, IY	$SP \leftarrow IY$	•	•	X	X	•	•	11 11	111 111	101 001	FD F9	2	2	10	
PUSH qq	$(SP-2) \leftarrow qq_L$ $(SP-1) \leftarrow qq_H$ $SP \rightarrow SP-2$	•	•	X	X	•	•	11	qq0	101		1	3	11	
PUSH IX	$(SP-2) \leftarrow IX_L$ $(SP-1) \leftarrow IX_H$ $SP \rightarrow SP-2$	•	•	X	X	•	•	11 11	011 100	101 101	DD E5	2	4	15	
PUSH IY	$(SP-2) \leftarrow IY_L$ $(SP-1) \leftarrow IY_H$ $SP \rightarrow SP-2$	•	•	X	X	•	•	11 11	111 100	101 101	FD E5	2	4	15	
POP qq	$qq_H \leftarrow (SP+1)$ $qqL \leftarrow (SP)$ $SP \rightarrow SP+2$	•	•	X	X	•	•	11	qq0	001		1	3	10	
POP IX	$IX_H \leftarrow (SP+1)$ $IX_L \leftarrow (SP)$ $SP \rightarrow SP+2$	•	•	X	X	•	•	11 11	011 100	101 001	DD E1	2	4	14	
POP IY	$IY_H \leftarrow (SP+1)$ $IY_L \leftarrow (SP)$ $SP \rightarrow SP+2$	•	•	X	X	•	•	11 11	111 100	101 001	FD E1	2	4	14	

qq	Pair
00	BC
01	DE
10	HL
11	AF

NOTE: $(PAIR)_H$, $(PAIR)_L$ refer to high order and low order eight bits of the register pair respectively, e.g., BC_L = C, AF_H = A.

EXCHANGE, BLOCK TRANSFER, BLOCK SEARCH GROUPS

Mnemonic	Symbolic Operation	Flags S	Z	H	P/V	N	C	Opcode 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
EX DE, HL	DE ↔ HL	•	•	X	•	X	•	•	•	11	101	011	EB	1	1	4	
EX AF, AF'	AF ↔ AF'	•	•	X	•	X	•	•	•	00	001	000	08	1	1	4	
EXX	BC ↔ BC' DE ↔ DE' HL ↔ HL'	•	•	X	•	X	•	•	•	11	011	001	D9	1	1	4	Register bank and auxiliary register bank exchange
EX (SP), HL	H ↔ (SP + 1) L ↔ (SP)	•	•	X	•	X	•	•	•	11	100	011	E3	1	5	19	
EX (SP), IX	IX_H ↔ (SP + 1) IX_L ↔ (SP)	•	•	X	•	X	•	•	•	11 11	011 100	101 011	DD E3	2	6	23	
EX (SP), IY	IY_H ↔ (SP + 1) IYL ↔ (SP)	•	•	X	•	X	•	•	•	11 11	111 100	101 011	FD E3	2	6	23	
LDI	(DE) ← (HL) DE ← DE + 1 HL ← HL + 1 BC ← BC − 1	•	•	X	0	X	↕ ①	0	•	11 10	101 100	101 000	ED A0	2	4	16	Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)
LDIR	(DE) ← (HL) DE ← DE + 1 HL ← HL + 1 BC ← BC − 1 Repeat until BC = 0	•	•	X	0	X	0 ②	0	•	11 10	101 110	101 000	ED B0	2 2	5 4	21 16	If BC ≠ 0 If BC = 0
LDD	(DE) ← (HL) DE ← DE − 1 HL ← HL − 1 BC ← BC − 1	•	•	X	0	X	↕ ①	0	•	11 10	101 101	101 000	ED A8	2	4	16	
LDDR	(DE) ← (HL) DE ← DE − 1 HL ← HL − 1 BC ← BC − 1 Repeat until BC = 0	•	•	X	0	X	0 ②	0	•	11 10	101 111	101 000	ED B8	2 2	5 4	21 16	If BC ≠ 0 If BC = 0
CPI	A − (HL) HL ← HL + 1 BC ← BC − 1	↕	↕ ③	X	↕	X	↕ ①	1	•	11 10	101 100	101 001	ED A1	2	4	16	

NOTE: ① P/V flag is 0 if the result of BC − 1 = 0, otherwise P/V = 1.
② P/V flag is 0 only at completion of instruction.
③ Z flag is 1 if A = HL, otherwise Z = 0.

EXCHANGE, BLOCK TRANSFER, BLOCK SEARCH GROUPS (Continued)

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
			③		①										
CPIR	A − (HL)	‡	‡	X	X	1	•	11	101	101	ED	2	5	21	If BC ≠ 0 and A ≠ (HL)
	HL ← HL + 1 BC ← BC − 1 Repeat until A = (HL) or BC = 0							10	110	001	B1	2	4	16	If BC = 0 or A = (HL)
			③		①										
CPD	A − (HL) HL ← HL − 1 BC ← BC − 1	‡	‡	X	X	1	•	11 10	101 101	101 001	ED A9	2	4	16	
			③		①										
CPDR	A − (HL)	‡	‡	X	X	1	•	11	101	101	ED	2	5	21	If BC ≠ 0 and A ≠ (HL)
	HL ← HL − 1 BC ← BC − 1 Repeat until A = (HL) or BC = 0							10	111	001	B9	2	4	16	If BC = 0 or A = (HL)

NOTE: ① P/V flag is 0 if the result of BC − 1 = 0, otherwise P/V = 1.
② P/V flag is 0 only at completion of instruction.
③ Z flag is 1 if A = (HL), otherwise Z = 0.

8-BIT ARITHMETIC AND LOGICAL GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
ADD A, r	A ← A + r	‡	‡	X	V	0	‡	10	[000]	r		1	1	4	r Reg.
ADD A, n	A ← A + n	‡	‡	X	V	0	‡	11	[000]	110		2	2	7	000 B
									← n →						001 C
															010 D
ADD A, (HL)	A ← A + (HL)	‡	‡	X	V	0	‡	10	[000]	110		1	2	7	011 E
ADD A, (IX + d)	A ← A + (IX + d)	‡	‡	X	V	0	‡	11	011	101	DD	3	5	19	100 H
								10	[000]	110					101 L
									← d →						111 A
ADD A, (IY + d)	A ← A + (IY + d)	‡	‡	X	V	0	‡	11	111	101	FD	3	5	19	
								10	[000]	110					
									← d →						
ADC A, s	A ← A + s + CY	‡	‡	X	V	0	‡		[001]						s is any of r, n, (HL), (IX + d), (IY + d) as shown for ADD instruction. The indicated bits replace the [000] in the ADD set above.
SUB s	A ← A − s	‡	‡	X	V	1	‡		[010]						
SBC A, s	A ← A − s − CY	‡	‡	X	V	1	‡		[011]						
AND s	A ← A ∧ s	‡	‡	X	P	0	0		[100]						
OR s	A ← A ∨ s	‡	‡	X	P	0	0		[110]						
XOR s	A ← A ⊕ s	‡	‡	X	P	0	0		[101]						
CP s	A − s	‡	‡	X	V	1	‡		[111]						

8-BIT ARITHMETIC AND LOGICAL GROUP (Continued)

Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	Opcode 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
INC r	r ← r + 1	‡	‡	X	‡	X	V	0	•	00	r	100		1	1	4	
INC (HL)	(HL) ← (HL) + 1	‡	‡	X	‡	X	V	0	•	00	110	100		1	3	11	
INC (IX + d)	(IX + d) ← (IX + d) + 1	‡	‡	X	‡	X	V	0	•	11 00	011 110 ← d →	101 100	DD	3	6	23	
INC (IY + d)	(IY + d) ← (IY + d) + 1	‡	‡	X	‡	X	V	0	•	11 00	111 110 ← d →	101 100	FD	3	6	23	
DEC m	m ← m − 1	‡	‡	X	‡	X	V	1	•			101					

NOTE: m is any of r, (HL), (IX + d), (IY + d) as shown for INC. DEC same format and states as INC. Replace 100 with 101 in opcode.

GENERAL-PURPOSE ARITHMETIC AND CPU CONTROL GROUPS

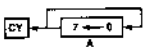
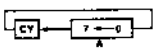
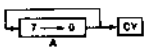
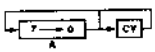
Mnemonic	Symbolic Operation	Flags S	Z		H		P/V	N	C	Opcode 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
DAA	@	‡	‡	X	‡	X	P	•	‡	00	100	111	27	1	1	4	Decimal adjust accumulator.
CPL	A ← A	•	•	X	1	X	•	1	•	00	101	111	2F	1	1	4	Complement accumulator (one's complement).
NEG	A ← 0 − A	‡	‡	X	‡	X	V	1	‡	11 01	101 000	101 100	ED 44	2	2	8	Negate acc. (two's complement).
CCF	CY ← CY	•	•	X	X	X	•	0	‡	00	111	111	3F	1	1	4	Complement carry flag.
SCF	CY ← 1	•	•	X	0	X	•	0	1	00	110	111	37	1	1	4	Set carry flag.
NOP	No operation	•	•	X	•	X	•	•	•	00	000	000	00	1	1	4	
HALT	CPU halted	•	•	X	•	X	•	•	•	01	110	110	76	1	1	4	
DI ★	IFF ← 0	•	•	X	•	X	•	•	•	11	110	011	F3	1	1	4	
EI ★	IFF ← 1	•	•	X	•	X	•	•	•	11	111	011	FB	1	1	4	
IM 0	Set interrupt mode 0	•	•	X	•	X	•	•	•	11 01	101 000	101 110	ED 46	2	2	8	
IM 1	Set interrupt mode 1	•	•	X	•	X	•	•	•	11 01	101 010	101 110	ED 56	2	2	8	
IM 2	Set interrupt mode 2	•	•	X	•	X	•	•	•	11 01	101 011	101 110	ED 5E	2	2	8	

NOTES: @ converts accumulator content into packed BCD following add or subtract with packed BCD operands.
IFF indicates the interrupt enable flip-flop.
CY indicates the carry flip-flop.
★ indicates interrupts are not sampled at the end of EI or DI.

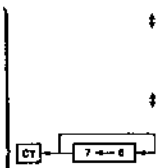
16-BIT ARITHMETIC GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
ADD HL, ss	HL ← HL + ss	•	•	X	X	X	•	0	‡	00	ss1	001		1	3	11	ss Reg. 00 BC 01 DE
ADC HL, ss	HL ← HL + ss + CY	‡	‡	X	X	X	V	0	‡	11 01	101 ss1	101 010	ED	2	4	15	10 HL 11 SP
SBC HL, ss	HL ← HL − ss − CY	‡	‡	X	X	X	V	1	‡	11 01	101 ss0	101 010	ED	2	4	15	
ADD IX, pp	IX ← IX + pp	•	•	X	X	X	•	0	‡	11 01	011 pp1	101 001	DD	2	4	15	pp Reg. 00 BC 01 DE 10 IX 11 SP
ADD IY, rr	IY ← IY + rr	•	•	X	X	X	•	0	‡	11 00	111 rr1	101 001	FD	2	4	15	rr Reg. 00 BC
INC ss	ss ← ss + 1	•	•	X	•	X	•	•	•	00	ss0	011		1	1	6	01 DE
INC IX	IX ← IX + 1	•	•	X	•	X	•	•	•	11 00	011 100	101 011	DD 23	2	2	10	10 IY 11 SP
INC IY	IY ← IY + 1	•	•	X	•	X	•	•	•	11 00	111 100	101 011	FD 23	2	2	10	
DEC ss	ss ← ss − 1	•	•	X	•	X	•	•	•	00	ss1	011		1	1	6	
DEC IX	IX ← IX − 1	•	•	X	•	X	•	•	•	11 00	011 101	101 011	DD 2B	2	2	10	
DEC IY	IY ← IY − 1	•	•	X	•	X	•	•	•	11 00	111 101	101 011	FD 2B	2	2	10	

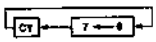
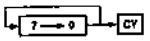
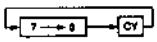
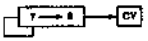
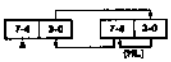
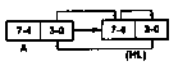
ROTATE AND SHIFT GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
RLCA		•	•	X	0	X	•	0	‡	00	000	111	07	1	1	4	Rotate left circular accumulator.
RLA		•	•	X	0	X	•	0	‡	00	010	111	17	1	1	4	Rotate left accumulator.
RRCA		•	•	X	0	X	•	0	‡	00	001	111	0F	1	1	4	Rotate right circular accumulator.
RRA		•	•	X	0	X	•	0	‡	00	011	111	1F	1	1	4	Rotate right accumulator.

ROTATE AND SHIFT GROUP (Continued)

Mnemonic	Symbolic Operation	S	Z	Flags H	P/V	N	C	Opcode 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
RLC r		‡	‡	X	0	X	P	0	‡	11	001	011	CB	2	2	8	Rotate left circular register r.
										00	000	r					
RLC (HL)		‡	‡	X	0	X	P	0	‡	11	001	011	CB	2	4	15	
										00	000	110					
RLC (IX + d)		‡	‡	X	0	X	P	0	‡	11	011	101	DD	4	6	23	
	r,(HL),(IX + d),(IY + d)									11	001	011	CB				
											← d →						
										00	000	110					
RLC (IY + d)		‡	‡	X	0	X	P	0	‡	11	111	101	FD	4	6	23	
										11	001	011	CB				
											← d →						
										00	000	110					

r	Reg.
000	B
001	C
010	D
011	E
001	H
101	L
111	A

RL m	 m = r,(HL),(IX + d),(IY + d)	‡	‡	X	0	X	P	0	‡		010					
RRC m	 m = r,(HL),(IX + d),(IY + d)	‡	‡	X	0	X	P	0	‡		001					
RR m	 m = r,(HL),(IX + d),(IY + d)	‡	‡	X	0	X	P	0	‡		011					
SLA m	 m = r,(HL),(IX + d),(IY + d)	‡	‡	X	0	X	P	0	‡		100					
SRA m	 m = r,(HL),(IX + d),(IY + d)	‡	‡	X	0	X	P	0	‡		101					
SRL m	 m = r,(HL),(IX + d),(IY + d)	‡	‡	X	0	X	P	0	‡		111					
RLD		‡	‡	X	0	X	P	0	•	11	101	101	ED	2	5	18	Rotate digit left and right between the accumulator and location (HL). The content of the upper half of the accumulator is unaffected.
										01	101	111	6F				
RRD		‡	‡	X	0	X	P	0	•	11	101	101	ED	2	5	18	
										01	100	111	67				

Instruction format and states are as shown for RLCs. To form new opcode replace 000 or RLCs with shown code.

BIT SET, RESET AND TEST GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	7 6	5 4 3	2 1 0	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
BIT b, r	$Z \leftarrow r_b$	X	‡	X	1	X	X	11	001	011	CB	2	2	8	r Reg.
								01	b	r					000 B
BIT b, (HL)	$Z \leftarrow (HL)_b$	X	‡	X	1	X	X	11	001	011	CB	2	3	12	001 C
								01	b	110					010 D
BIT b,(IX+d)$_b$	$Z \leftarrow (IX+d)_b$	X	‡	X	1	X	X	11	011	101	DD	4	5	20	011 E
								11	001	011	CB				100 H
									←d→						101 L
								01	b	110					111 A
															b Bit Tested
BIT b, (IY+d)$_b$	$Z \leftarrow (IY+d)_b$	X	‡	X	1	X	X	11	111	101	FD	4	5	20	000 0
								11	001	011	CB				001 1
									←d→						010 2
								01	b	110					011 3
SET b, r	$r_b \leftarrow 1$	•	•	X	•	X	•	•	•	11	001	011	CB	2	2	8	100 4
								[11]	b	r					101 5
SET b, (HL)	$(HL)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11	001	011	CB	2	4	15	110 6
								[11]	b	110					111 7
SET b, (IX+d)	$(IX+d)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11	011	101	DD	4	6	23	
								11	001	011	CB				
									←d→						
								[11]	b	110					
SET b, (IY+d)	$(IY+d)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11	111	101	FD	4	6	23	
								11	001	011	CB				
									←d→						
								[11]	b	110					
RES b, m	$m_b \leftarrow 0$ m≡r, (HL), (IX+d), (IY+d)	•	•	X	•	X	•	•	•	[10]					To form new opcode replace [11] of SET b, s with [10]. Flags and time states for SET instruction.

NOTE: The notation m_b indicates location m, bit b (0 to 7).

JUMP GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
JP nn	PC ← nn	•	•	X	•	X	•	•	•	11	000	011	C3	3	3	10	cc Condition
											← n →						000 NZ (non-zero)
											← n →						001 Z (zero)
JP cc, nn	If condition cc	•	•	X	•	X	•	•	•	11	cc	010		3	3	10	010 NC (non-carry)
	is true PC ← nn,										← n →						011 C (carry)
	otherwise										← n →						100 PO (parity odd)
	continue																101 PE (parity even)
JR e	PC ← PC + e	•	•	X	•	X	•	•	•	00	011	000	18	2	3	12	110 P (sign positive)
											← e − 2 →						111 M (sign negative)
JR C, e	If C = 0,	•	•	X	•	X	•	•	•	00	111	000	38	2	2	7	If condition not met.
	continue										← e − 2 →						
	If C = 1,													2	3	12	If condition is met.
	PC ← PC + e																
JR NC, e	IF C = 1,	•	•	X	•	X	•	•	•	00	110	000	30	2	2	7	If condition not met.
	continue										← e − 2 →						
	If C = 0,													2	3	12	If condition is met.
	PC ← PC + e																
JP Z, e	If Z = 0	•	•	X	•	X	•	•	•	00	101	000	28	2	2	7	If condition not met.
	continue										← e − 2 →						
	If Z = 1,													2	3	12	If condition is met.
	PC ← PC + e																
JR NZ, e	If Z = 1,	•	•	X	•	X	•	•	•	00	100	000	20	2	2	7	If condition not met.
	continue										← e − 2 →						
	If Z = 0,													2	3	12	If condition is met.
	PC ← PC + e																
JP (HL)	PC ← HL	•	•	X	•	X	•	•	•	11	101	001	E9	1	1	4	
JP (IX)	PC ← IX	•	•	X	•	X	•	•	•	11	011	101	DD	2	2	8	
										11	101	001	E9				
JP (IY)	PC ← IY	•	•	X	•	X	•	•	•	11	111	101	FD	2	2	8	
										11	101	001	E9				
DJNZ, e	B ← B − 1	•	•	X	•	X	•	•	•	00	010	000	10	2	2	8	If B = 0
	If B = 0,										← e − 2 →						
	continue																
	If B≠0,													2	3	13	If B≠0.
	PC ← PC + e																

NOTES: e represents the extension in the relative addressing mode.
e is a signal two's complement number in the range < − 126, 129 >.
e − 2 in the opcode provides an effective address of pc + e as PC is incremented by 2 prior to the addition of e.

CALL AND RETURN GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
CALL nn	$(SP-1)\leftarrow PC_H$ $(SP-2)\leftarrow PC_L$ $PC \leftarrow nn$,	•	•	X	•	X	•	•	•	11	001 ←n→ ←n→	101	CD	3	5	17	
CALL cc,nn	If condition cc is false	•	•	X	•	X	•	•	•	11	cc ←n→	100		3	3	10	If cc is false.
	continue, otherwise same as CALL nn										←n→			3	5	17	If cc is true.
RET	$PC_L \leftarrow (SP)$ $PC_H \leftarrow (SP+1)$	•	•	X	•	X	•	•	•	11	001	001	C9	1	3	10	
RET cc	If condition cc is false	•	•	X	•	X	•	•	•	11	cc	000		1	1	5	If cc is false.
	continue, otherwise same as RET													1	3	11	If cc is true.
RETI	Return from interrupt	•	•	X	•	X	•	•	•	11 01	101 001	101 101	ED 4D	2	4	14	
RETN[1]	Return from non-maskable interrupt	•	•	X	•	X	•	•	•	11 01	101 000	101 101	ED 45	2	4	14	
RST p	$(SP-1)\leftarrow PC_H$ $(SP-2)\leftarrow PC_L$ $PC_H \leftarrow 0$ $PC_L \leftarrow p$	•	•	X	•	X	•	•	•	11	t	111		1	3	11	

cc	Condition
000	NZ (non-zero)
001	Z (zero)
010	NC (non-carry)
011	C (carry)
100	PO (parity odd)
101	PE (parity even)
110	P (sign positive)
111	M (sign negative)

t	p
000	00H
001	08H
010	10H
011	18H
100	20H
101	28H
110	30H
111	38H

NOTE: [1]RETN loads $IFF_2 \rightarrow IFF_1$

INPUT AND OUTPUT GROUP

Mnemonic	Symbolic Operation	Flags S	Z	H	P/V	N	C	Opcode 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
IN A, (n)	A ← (n)	•	•	X	•	X	•	•	•	11	011 ← n →	01	DB	2	3	11	n to $A_0 \sim A_7$ Acc. to $A_8 \sim A_{15}$
IN r, (C)	r ← (C) if r = 110 only the flags will be affected	‡	‡	X	‡	X	P	0	•	11 01	101 r	101 000	ED	2	3	12	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INI	(HL) ← (C) B ← B – 1 HL ← HL + 1	X	‡ ①	X	X	X	X	1	X	11 10	101 100	101 010	ED A2	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INIR	(HL) ← (C) B ← B – 1 HL ← HL + 1 Repeat until B = 0	X	1 ②	X	X	X	X	1	X	11 10	101 110	101 010	ED B2	2 2	5 (If B≠0) 4 (If B = 0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
IND	(HL) ← (C) B ← B – 1 HL ← HL – 1	X	‡ ①	X	X	X	X	1	X	11 10	101 101	101 010	ED AA	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INDR	(HL) ← (C) B ← B – 1 HL ← HL – 1 Repeat until B = 0	X	1 ②	X	X	X	X	1	X	11 10	101 111	101 010	ED BA	2 2	5 (If B≠0) 4 (If B = 0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUT (n), A	(n) ← A	•	•	X	•	X	•	•	•	11	010 ← n →	011	D3	2	3	11	n to $A_0 \sim A_7$ Acc. to $A_8 \sim A_{15}$
OUT (C), r	(C) ← r	•	•	X	•	X	•	•	•	11 01	101 r	101 001	ED	2	3	12	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUTI	(C) ← (HL) B ← B – 1 HL ← HL + 1	X	‡ ①	X	X	X	X	1	X	11 10	101 100	101 011	ED A3	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OTIR	(C) ← (HL) B ← B – 1 HL ← HL + 1 Repeat until B = 0	X	1 ②	X	X	X	X	1	X	11 10	101 110	101 011	ED B3	2 2	5 (If B≠0) 4 (If B = 0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUTD	(C) ← (HL) B ← B – 1 HL ← HL – 1	X	‡ ①	X	X	X	X	1	X	11 10	101 101	101 011	ED AB	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OTDR	(C) ← (HL) B ← B – 1 HL ← HL – 1 Repeat until B = 0	X	1 ②	X	X	X	X	1	X	11 10	101 111	101 011	ED BB	2 2	5 (If B≠0) 4 (If B = 0)	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$

NOTES: ① If the result of B – 1 is zero, the Z flag is set; otherwise it is reset.
② Z flag is set upon instruction completion only.

SUMMARY OF FLAG OPERATION

Instructions	D_7 S	Z		H		P/V	N	D_0 C	Comments
ADD A, s; ADC A, s	‡	‡	X	‡	X	V	0	‡	8-bit add or add with carry.
SUB s; SBC A, s; CP s; NEG	‡	‡	X	‡	X	V	1	‡	8-bit subtract, subtract with carry, compare and negate accumulator.
AND s	‡	‡	X	1	X	P	0	0	Logical operation.
OR s, XOR s	‡	‡	X	0	X	P	0	0	Logical operation.
INC s	‡	‡	X	‡	X	V	0	•	8-bit increment.
DEC s	‡	‡	X	‡	X	V	1	•	8-bit decrement.
ADD DD, ss	•	•	X	X	X	•	0	‡	16-bit add.
ADC HL, ss	‡	‡	X	X	X	V	0	‡	16-bit add with carry.
SBC HL, ss	‡	‡	X	X	X	V	1	‡	16-bit subtract with carry.
RLA; RLCA; RRA; RRCA	•	•	X	0	X	•	0	‡	Rotate accumulator.
RL m; RLC m; RR m; RRC m; SLA m; SRA m; SRL m	‡	‡	X	0	X	P	0	‡	Rotate and shift locations.
RLD; RRD	‡	‡	X	0	X	P	0	•	Rotate digit left and right.
DAA	‡	‡	X	‡	X	P	•	‡	Decimal adjust accumulator.
CPL	•	•	X	1	X	•	1	•	Complement accumulator.
SCF	•	•	X	0	X	•	0	1	Set carry.
CCF	•	•	X	X	X	•	0	‡	Complement carry.
IN r (C)	‡	‡	X	0	X	P	0	•	Input register indirect.
INI; IND; OUTI; OUTD	X	‡	X	X	X	X	1	•	Block input and output. Z = 1 if B ≠ 0, otherwise Z = 0.
INIR; INDR; OTIR; OTDR	X	1	X	X	X	X	1	•	Block input and output. Z = 1 if B ≠ 0, otherwise Z = 0.
LDI; LDD	X	X	X	0	X	‡	0	•	Block transfer instructions. P/V = 1 if BC ≠ 0, otherwise P/V = 0.
LDIR; LDDR	X	X	X	0	X	0	0	•	Block transfer instructions. P/V = 1 if BC ≠ 0, otherwise P/V = 0.
CPI; CPIR; CPD; CPDR	X	‡	X	X	X	‡	1	•	Block search instructions. Z = 1 if A = (HL), otherwise Z = 0. P/V = 1 if BC ≠ 0, otherwise P/V = 0.
LD A; I, LD A, R	‡	‡	X	0	X	IFF	0	•	IFF, the content of the interrupt enable flip-flop, (IFF_2), is copied into the P/V flag.
BIT b, s	X	‡	X	1	X	X	0	•	The state of bit b of location s is copied into the Z flag.

SYMBOLIC NOTATION

Symbol	Operation
S	Sign flag. S = 1 if the MSB of the result is 1.
Z	Zero flag. Z = 1 if the result of the operation is 0.
P/V	Parity or overflow flag. Parity (P) and overflow (V) share the same flag. Logical operations affect this flag with the parity of the result while arithmetic operations affect this flag with the overflow of the result. If P/V holds parity: P/V = 1 if the result of the operation is even; P/V = 0 if result is odd. If P/V holds overflow, P/V = 1 if the result of the operation produced an overflow. If P/V does not hold overflow, P/V = 0.
H*	Half-carry flag. H = 1 if the add or subtract operation produced a carry into, or borrow from, bit 4 of the accumulator.
N*	Add/Subtract flag. N = 1 if the previous operation was a subtract.
C	Carry/Link flag. C = 1 if the operation produced a carry from the MSB of the operand or result.

Symbol	Operation
‡	The flag is affected according to the result of the operation.
•	The flag is unchanged by the operation.
0	The flag is reset by the operation.
1	The flag is set by the operation.
X	The flag is indeterminate.
V	P/V flag affected according to the overflow result of the operation.
P	P/V flag affected according to the parity result of the operation.
r	Any one o the CPU registers A, B, C, D, E, H, L.
s	Any 8-bit location for all the addressing modes allowed for the particular instruction.
ss	Any 16-bit location for all the addressing modes allowed for that instruction.
ii	Any one of the two index registers IX or IY.
R	Refresh counter.
n	8-bit value in range < 0, 255 >.
nn	16-bit value in range < 0, 65535 >.

* H and N flags are used in conjunction with the decimal adjust instruction (DAA) to properly correct the result into packed BCD format following addition or subtraction using operands with packed BCD format.

CPU REGISTERS

Figure 4 shows three groups of registers within the CPU. The first group consists of duplicate sets of 8-bit registers: a principal set and an alternate set [designated by ' (prime), e.g., A']. Both sets consist of the Accumulator register, the Flag register, and six general-purpose registers. Transfer of data between these duplicate sets of registers is accomplished by use of "Exchange" instructions. The result is faster response to interrupts and easy, efficient implementation of such versatile programming techniques as background-foreground data processing. The second set of registers consists of six registers with assigned functions. These are the I (Interrupt register), the R (Refresh register), the IX and IY (Index registers), the SP (Stack Pointer), and the PC (Program Counter). The third group consists of two interrupt status flip-flops, plus an additional pair of flip-flops which assists in identifying the interrupt mode at any particular time. Table 1 provides further information on these registers.

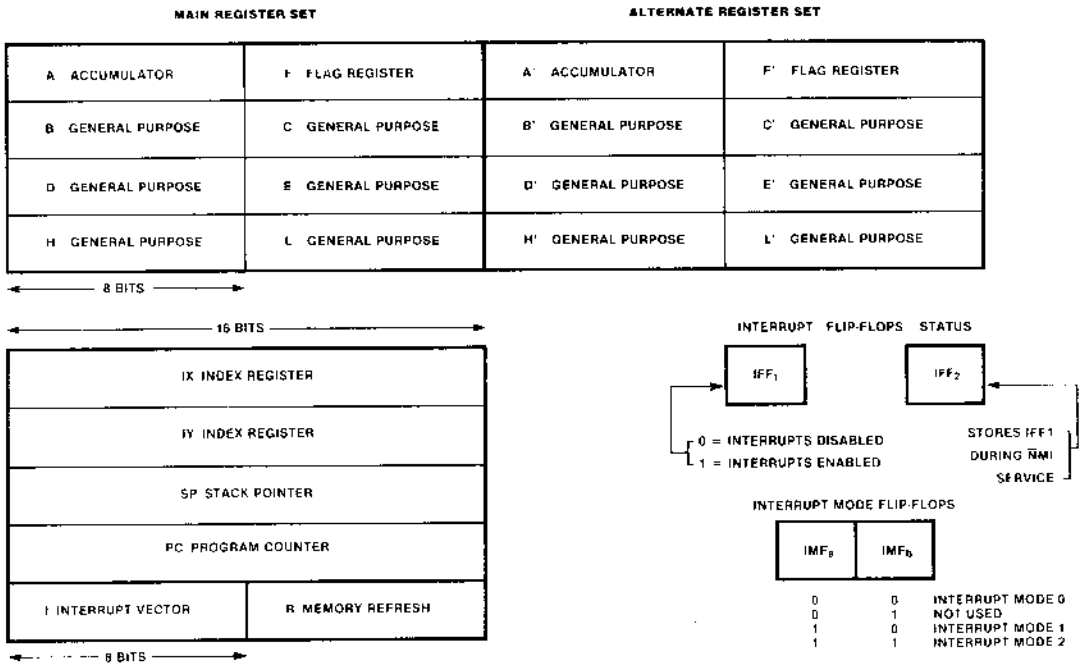


Figure 4. CPU Registers

INTERRUPTS: GENERAL OPERATION

The CPU accepts two interrupt input signals: $\overline{NMI}$ and $\overline{INT}$. The $\overline{NMI}$ is a non-maskable interrupt and has the highest priority. $\overline{INT}$ is a lower priority interrupt and it requires that interrupts be enabled in software in order to operate. $\overline{INT}$ can be connected to multiple peripheral devices in a wired-OR configuration.

The Z80 has a single response mode for interrupt service on the non-maskable interrupt. The maskable interrupt, $\overline{INT}$, has three programmable response modes available. These are:

- Mode 0 — similar to the 8080 microprocessor.
- Mode 1 — Peripheral Interrupt service, for use with non-8080/Z80 systems.
- **Mode 2 - a vectored interrupt scheme, usually daisy-chained, for use with the Z80 Family and compatible peripheral devices.**

The CPU services interrupts by sampling the $\overline{NMI}$ and $\overline{INT}$ signals at the rising edge of the last clock of an instruction. Further interrupt service processing depends upon the type of interrupt that was detected. Details on interrupt responses are shown in the CPU Timing Section.

Non-Maskable Interrupt ($\overline{NMI}$). The nonmaskable interrupt cannot be disabled by program control and therefore will be accepted at all times by the CPU. $\overline{NMI}$ is usually reserved for servicing only the highest priority type interrupts, such as that for orderly shutdown after power

PIN DESCRIPTIONS

A_0-A_{15}. *Address Bus* (output, active High, 3-state). A_0-A_{15} form a 16-bit address bus. The Address Bus provides the address for memory data bus exchanges (up to 64K bytes) and for I/O device exchanges.

$\overline{BUSACK}$. *Bus Acknowledge* (output, active Low). Bus Acknowledge indicates to the requesting device that the CPU address bus, data bus, and control signals $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$ have entered their high-impedance states. The external circuitry can now control these lines.

$\overline{BUSREQ}$. *Bus Request* (input, active Low). Bus Request has a higher priority than $\overline{NMI}$ and is always recognized at the end of the current machine cycle. $\overline{BUSREQ}$ forces the CPU address bus, data bus, and control signals $\overline{MREQ}$, $\overline{IORQ}$, RD, and WR to go to a high-impedance state so that other devices can control these lines. $\overline{BUSREQ}$ is normally wired-OR and requires an external pullup for these applications. Extended $\overline{BUSREQ}$ periods due to extensive DMA operations can prevent the CPU from properly refreshing dynamic RAMs.

D_0-D_7. *Data Bus* (input/output, active High, 3-state). D_0-D_7 constitute an 8-bit bidirectional data bus, used for data exchanges with memory and I/O.

$\overline{HALT}$. *Halt State* (output, active Low). $\overline{HALT}$ indicates that the CPU has executed a Halt instruction and is awaiting either a nonmaskable or a maskable interrupt (with the mask enabled) before operation can resume. While halted, the CPU executes NOPs to maintain memory refresh.

$\overline{INT}$. *Interrupt Request* (input, active Low). Interrupt Request is generated by I/O devices. The CPU honors a request at the end of the current instruction if the internal software-controlled interrupt enable flip-flop (IFF) is enabled. $\overline{INT}$ is normally wired-OR and requires an external pullup for these applications.

$\overline{IORQ}$. *Input/Output Request* (output, active Low, 3-state). $\overline{IORQ}$ indicates that the lower half of the address bus holds a valid I/O address for an I/O read or write operation. $\overline{IORQ}$ is also generated concurrently with $\overline{M1}$ during an interrupt acknowledge cycle to indicate that an interrupt response vector can be placed on the data bus.

$\overline{M1}$. *Machine Cycle One* (output, active Low). $\overline{M1}$, together with $\overline{MREQ}$, indicates that the current machine cycle is the opcode fetch cycle of an instruction execution. $\overline{M1}$, together with $\overline{IORQ}$, indicates an interrupt acknowledge cycle.

$\overline{MREQ}$. *Memory Request* (output, active Low, 3-state). $\overline{MREQ}$ indicates that the address bus holds a valid address for a memory read or memory write operation.

$\overline{NMI}$. *Non-Maskable Interrupt* (input, negative edge-triggered). $\overline{NMI}$ has a higher priority than $\overline{INT}$. $\overline{NMI}$ is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop, and automatically forces the CPU to restart at location 0066H.

$\overline{RD}$. *Read* (output, active Low, 3-state). $\overline{RD}$ 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.

$\overline{RESET}$. *Reset* (input, active Low). $\overline{RESET}$ initializes the CPU as follows: it resets the interrupt enable flip-flop, clears the PC and Registers I and R, and sets the interrupt status to Mode 0. During reset time, the address and data bus go to a high-impedance state, and all control output signals go to the inactive state. Note that $\overline{RESET}$ must be active for a minimum of three full clock cycles before the reset operation is complete.

$\overline{RFSH}$. *Refresh* (output, active Low). $\overline{RFSH}$, together with $\overline{MREQ}$, indicates that the lower seven bits of the system's address bus can be used as a refresh address to the system's dynamic memories.

$\overline{WAIT}$. *Wait* (input, active Low). $\overline{WAIT}$ indicates to the CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter a Wait state as long as this signal is active. Extended $\overline{WAIT}$ periods can prevent the CPU from properly refreshing dynamic memory.

$\overline{WR}$. *Write* (output, active Low, 3-state). $\overline{WR}$ indicates that the CPU data bus holds valid data to be stored at the addressed memory or I/O location.

CPU TIMING

The Z80 CPU executes instructions by proceeding through a specific sequence of operations:

- Memory read or write
- I/O device read or write
- Interrupt acknowledge

The basic clock period is referred to as a T time or cycle, and three or more T cycles make up a machine cycle (M1, M2 or M3 for instance). Machine cycles can be extended either by the CPU automatically inserting one or more Wait states or by the insertion of one or more Wait states by the user.

Instruction Opcode Fetch. The CPU places the contents of the Program Counter (PC) on the address bus at the start of the cycle (Figure 5). Approximately one-half clock cycle later, $\overline{MREQ}$ goes active. When active, $\overline{RD}$ indicates that the memory data can be enabled onto the CPU data bus.

The CPU samples the $\overline{WAIT}$ input with the falling edge of clock state T_2. During clock states T_3 and T_4 of an $\overline{M1}$ cycle, dynamic RAM refresh can occur while the CPU starts decoding and executing the instruction. When the Refresh Control signal becomes active, refreshing of dynamic memory can take place.

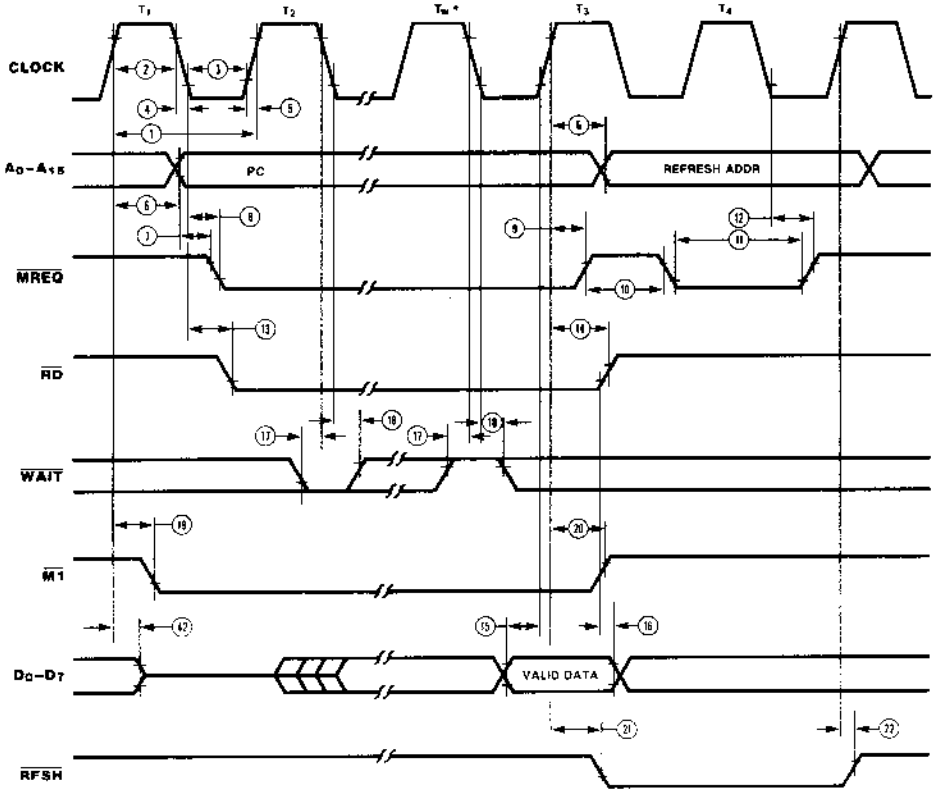


Figure 5. Instruction Opcode Fetch

Memory Read or Write Cycles. Figure 6 shows the timing of memory read or write cycles other than an opcode fetch ($\overline{M1}$) cycle. The $\overline{MREQ}$ and $\overline{RD}$ signals function exactly as in the fetch cycle. In a memory write cycle, $\overline{MREQ}$ also becomes active when the address bus is stable. The $\overline{WR}$ line is active when the data bus is stable, so that it can be used directly as an R/$\overline{W}$ pulse to most semiconductor memories.

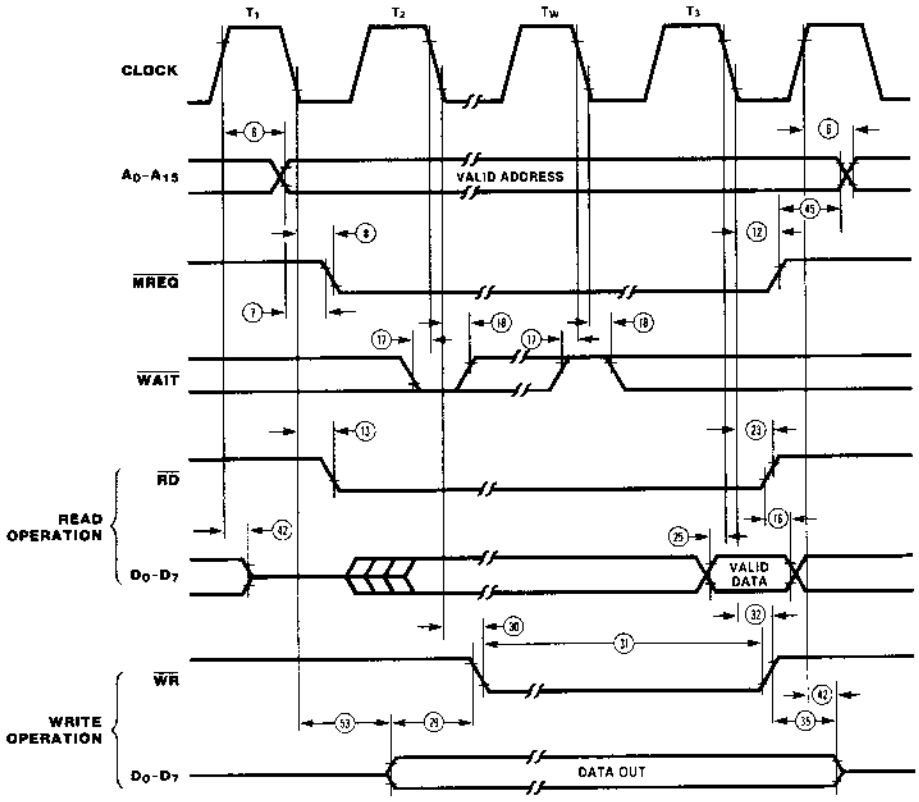
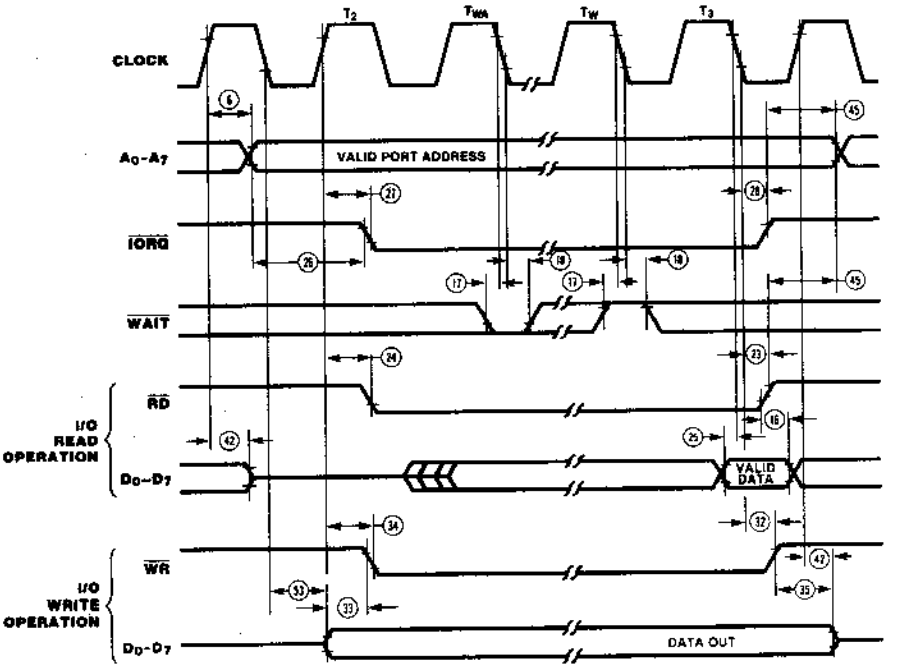


Figure 6. Memory Read or Write Cycles

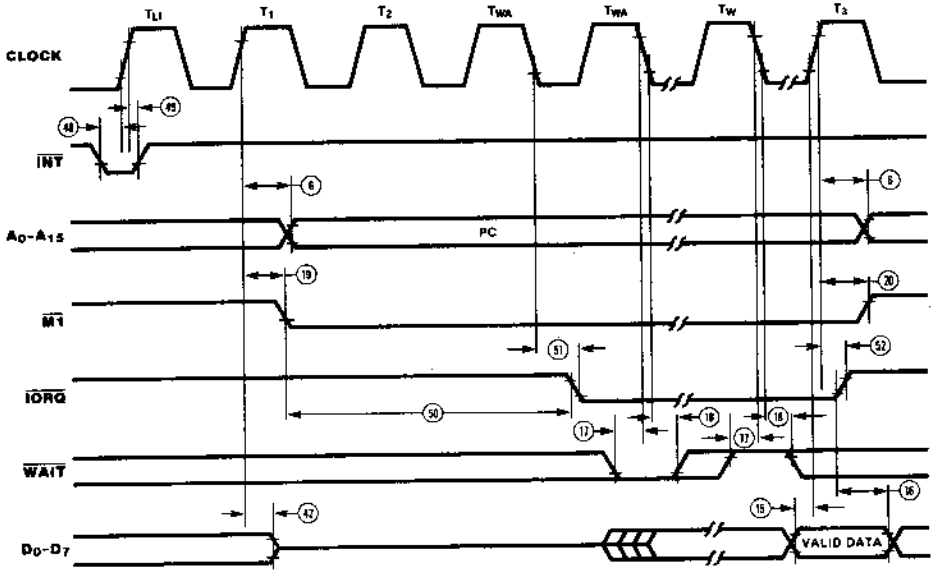
Input or Output Cycles. Figure 7 shows the timing for an I/O read or I/O write operation. During I/O operations, the CPU automatically inserts a single Wait state (T_{WA}). This extra Wait state allows sufficient time for an I/O port to decode the address from the port address lines.



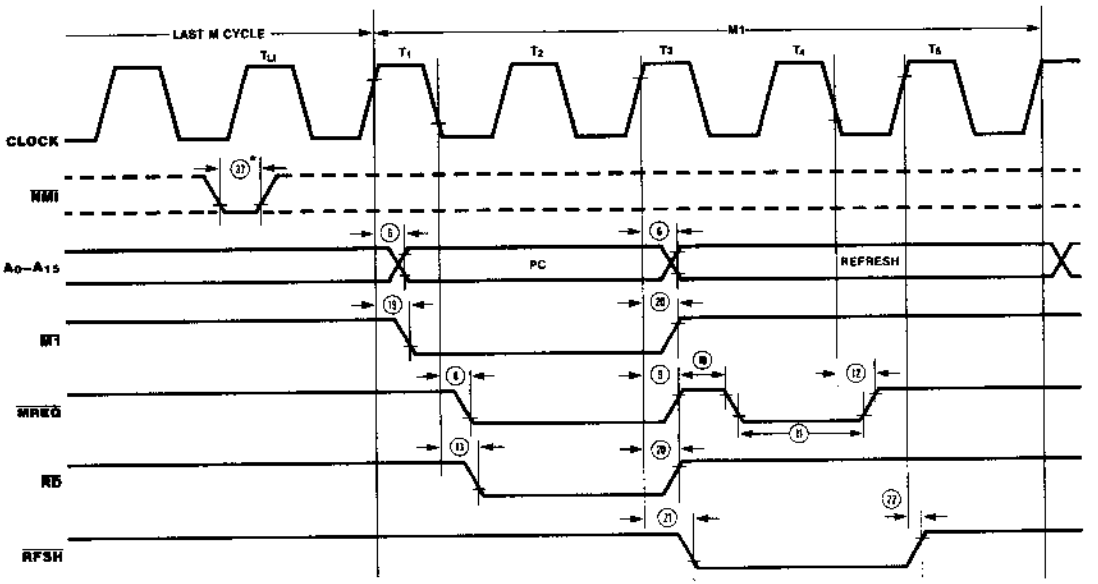
T_{WA} = One wait cycle automatically inserted by CPU.

Figure 7. Input or Output Cycles

Interrupt Request/Acknowledge Cycle. The CPU samples the interrupt signal with the rising edge of the last clock cycle at the end of any instruction (Figure 8). When an interrupt is accepted, a special $\overline{M1}$ cycle is generated. During this $\overline{M1}$ cycle, $\overline{IORQ}$ becomes active (instead of $\overline{MREQ}$) to indicate that the interrupting device can place an 8-bit vector on the data bus. The CPU automatically adds two Wait states to this cycle.



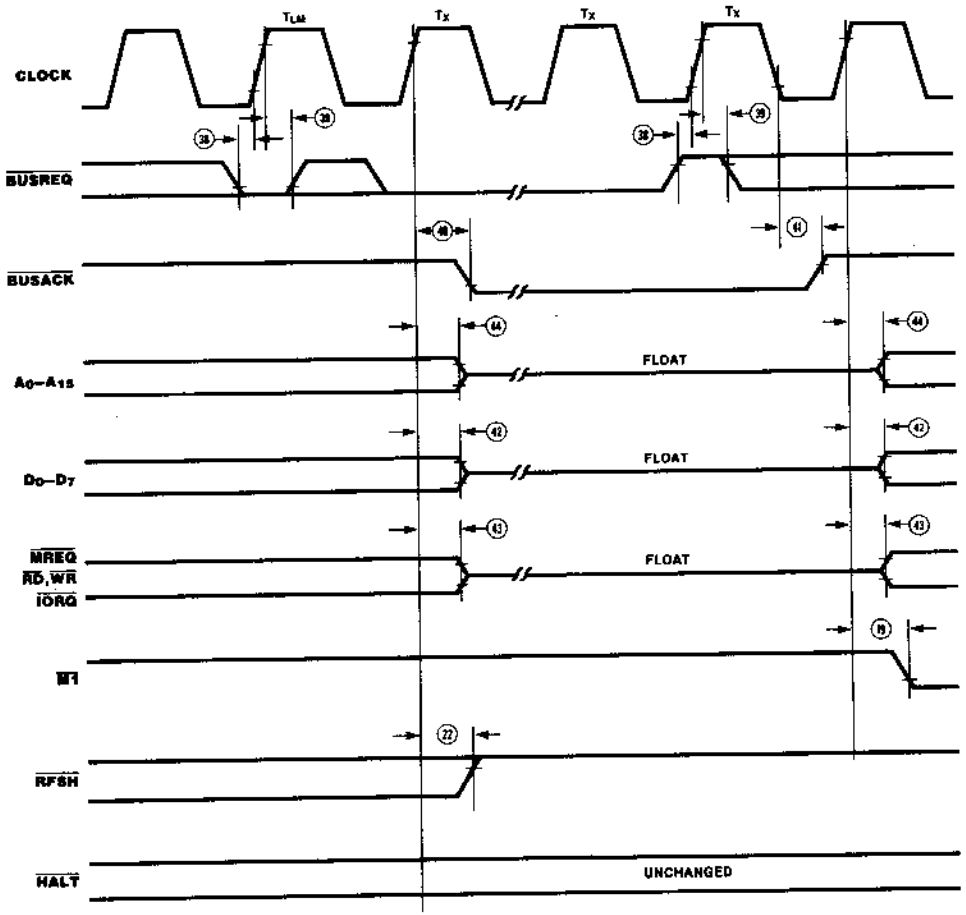
Non-Maskable Interrupt Request Cycle. $\overline{NMI}$ is sampled at the same time as the maskable interrupt input $\overline{INT}$ but has higher priority and cannot be disabled under software control. The subsequent timing is similar to that of a normal memory read operation except that data put on the bus by the memory is ignored. The CPU instead executes a restart (RST) operation and jumps to the $\overline{NMI}$ service routine located at address 0066H (Figure 9).



*Although $\overline{NMI}$ is an asynchronous input, to guarantee its being recognized on the following machine cycle, $\overline{NMI}$'s falling edge must occur no later than the rising edge of the clock cycle preceding the last state of any instruction cycle (T_{LI}).

Figure 9. Non-Maskable Interrupt Request Operation

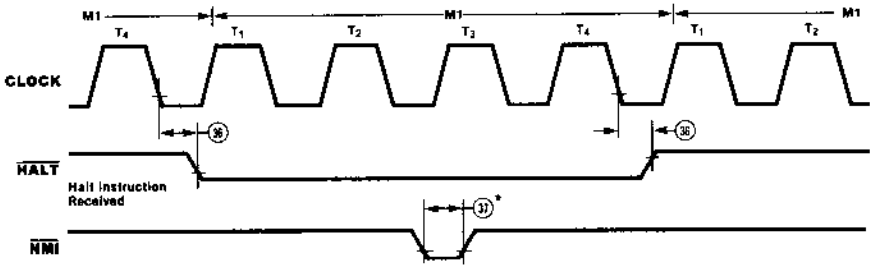
Bus Request/Acknowledge Cycle. The CPU samples $\overline{BUSREQ}$ with the rising edge of the last clock period of any machine cycle (Figure 10). If $\overline{BUSREQ}$ is active, the CPU sets its address, data, and $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$ lines to a high-impedance state with the rising edge of the next clock pulse. At that time, any external device can take control of these lines, usually to transfer data between memory and I/O devices.



NOTES: 1) T_{LM} = Last state of any M cycle.
2) T_X = An arbitrary clock cycle used by requesting device.

Figure 10. BUS Request/Acknowledge Cycle

Halt Acknowledge Cycle. When the CPU receives a HALT instruction, it executes NOP states until either an $\overline{INT}$ or $\overline{NMI}$ input is received. When in the Halt state, the $\overline{HALT}$ output is active and remains so until an interrupt is received (Figure 11). $\overline{INT}$ will also force a Halt exit.



*Although $\overline{NMI}$ is an asynchronous input, to guarantee its being recognized on the following machine cycle, $\overline{NMI}$'s falling edge must occur no later than the rising edge of the clock cycle preceding the last state of any instruction cycle (T_{Li}).

Figure 11. Halt Acknowledge

Reset Cycle. $\overline{RESET}$ must be active for at least three clock cycles for the CPU to properly accept it. As long as $\overline{RESET}$ remains active, the address and data buses float, and the control outputs are inactive. Once $\overline{RESET}$ goes inactive, two internal T cycles are consumed before the CPU resumes normal processing operation. $\overline{RESET}$ clears the PC register, so the first opcode fetch will be to location 0000H (Figure 12).

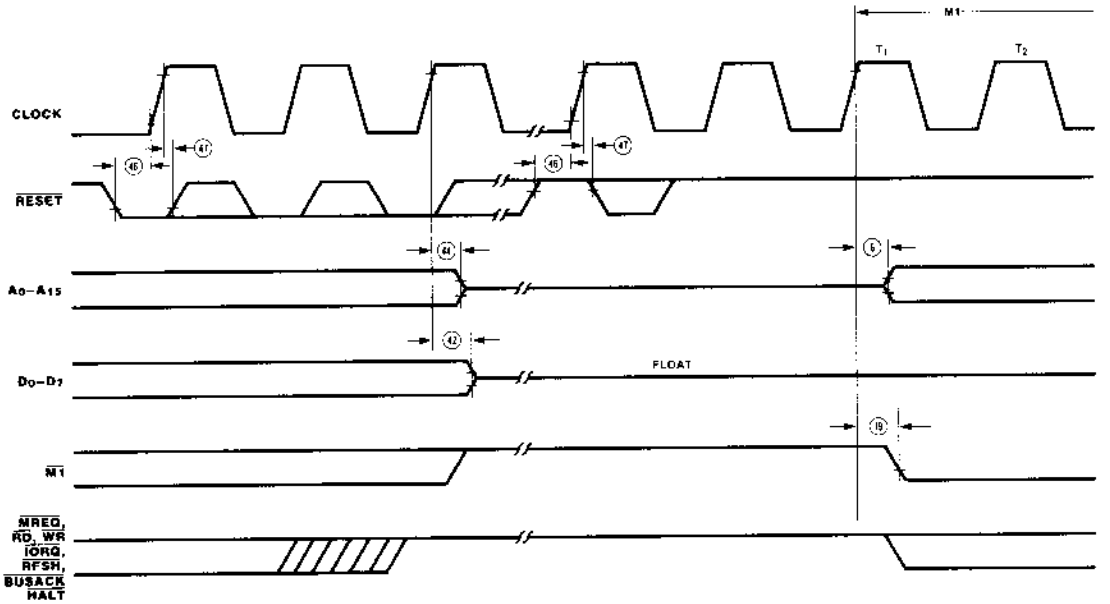


Figure 12. Reset Cycle

Power-Down mode of operation (Only applies to CMOS Z80 CPU).

CMOS Z80 CPU supports Power-Down mode of operation.

This mode is also referred to as the "standby mode", and supply current for the CPU goes down as low as 10 uA (Where specified as Icc_2).

Power-Down Acknowledge Cycle. When the clock input to the CPU is stopped at either a High or Low level, the CPU stops its operation and maintains all registers and control signals. However, I_{CC2} (standby supply current) is guaranteed only when the system clock is stopped at a Low level during T_4 of the machine cycle following the execution of the HALT instruction. The timing diagram for the power-down function, when implemented with the HALT **instruction, is shown in Figure 13.**

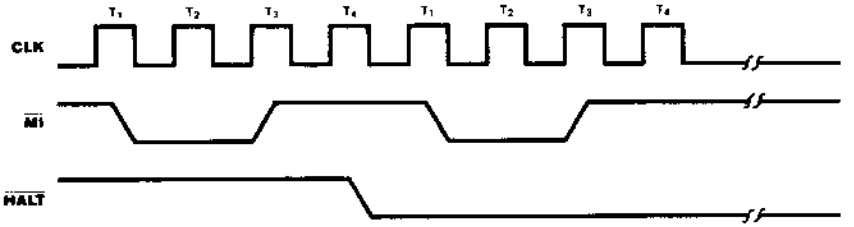


Figure 13. Power-Down Acknowledge

Power-Down Release Cycle. The system clock must be supplied to the CPU to release the power-down state. When the system clock is supplied to the CLK input, the CPU restarts operations from the point at which the power-down state was implemented.

The timing diagrams for the release from power-down mode are shown in Figure 14.

NOTES:

1) When the external oscillator has been stopped to enter the power-down state, some warm-up time may be required to obtain a stable clock for the release.
2) When the HALT instruction is executed to enter the power-down state, the CPU will also enter the Halt state. An interrupt signal (either $\overline{NMI}$ or $\overline{INT}$) or a $\overline{RESET}$ signal must be applied to the CPU after the system clock is supplied in order to release the power-down state.

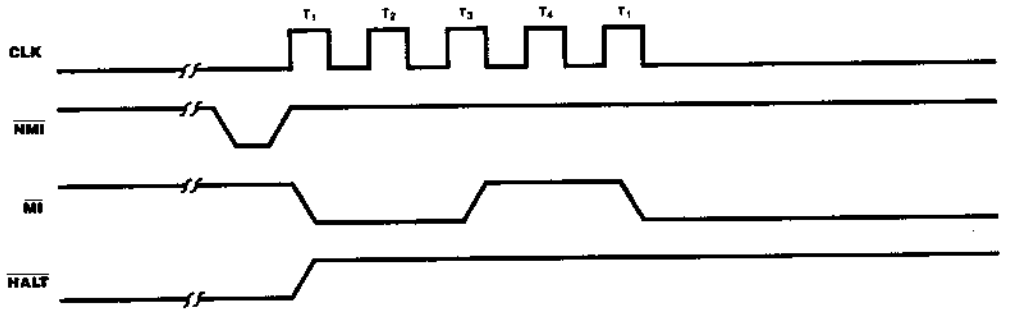


Figure 14a.

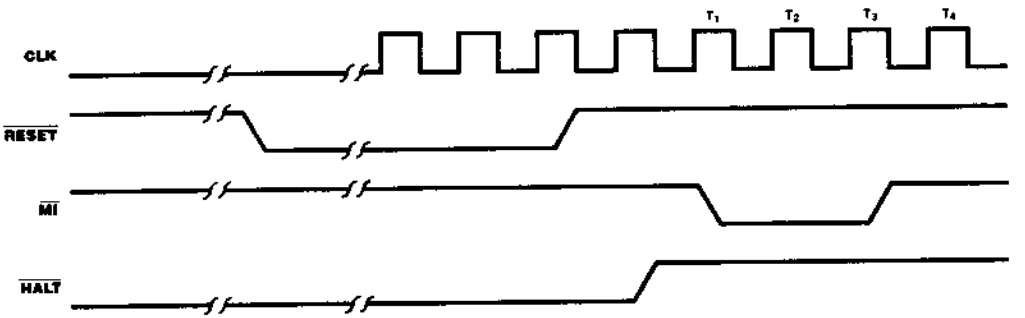


Figure 14b.

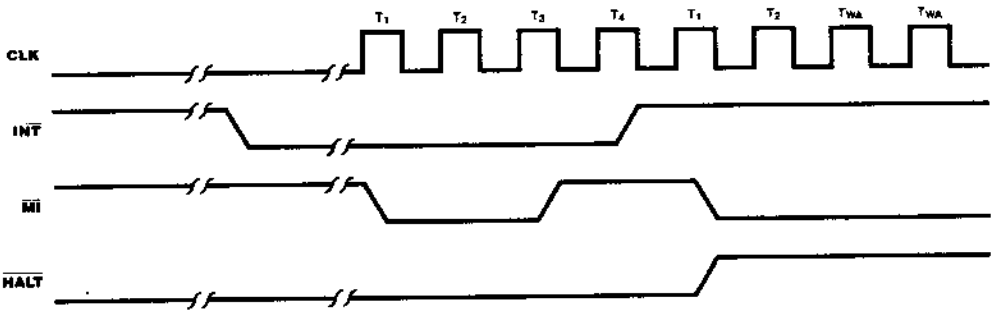


Figure 14c.

Figure 13. Power-Down Release

ABSOLUTE MAXIMUM RATINGS

Voltage on V_{CC} with respect to V_{SS} . . . . . −0.3V to +7V
Voltages on all inputs with respect to V_{SS} . . . . . −0.3V to V_{CC} + 0.3V
Operating Ambient Temperature . . . . . See Ordering Information
Storage Temperature . . . . . −65°C to +150°C

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

STANDARD TEST CONDITIONS

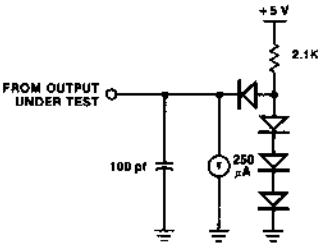
The DC Characteristics and capacitance sections below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND (0V). Positive current flows into the referenced pin.

Available operating temperature ranges are:

- **S = 0°C to +70°C**
  Voltage Supply Range:
  NMOS: +4.75V ≤ VCC ≤ +5.25V
  CMOS: +4.50V ≤ VCC ≤ +5.50V
- E= -40°C to 100°C, +4.50V ≤ VCC ≤ +5.50V

All ac parameters assume a load capacitance of 100 pf. Add 10 ns delay for each 50 pf increase in load up to a maximum of 200 pf for the data bus and 100 pf for address and control lines. AC timing measurements are referenced to 1.5 volts (except for clock, which is referenced to the 10% and 90% points).

The Ordering Information section lists temperature ranges and product numbers. Package drawings are in the Package Information section. Refer to the Literature List for additional documentation.



DC CHARACTERISTICS (Z84C00/CMOS Z80 CPU)

Symbol	Parameter	Min	Max	Unit	Condition
V_{ILC}	Clock Input Low Voltage	−0.3	0.45	V	
V_{IHC}	Clock Input High Voltage	V_{CC} −.6	V_{CC} +.3	V	
V_{IL}	Input Low Voltage	−0.3	0.8	V	
V_{IH}	Input High Voltage	2.2	V_{CC}	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2.0 mA
V_{OH_1}	Output High Voltage	2.4		V	I_{OH} = −1.6 mA
V_{OH_2}	Output High Voltage	V_{CC} − 0.8		V	I_{OH} = −250 μA
I_{CC_1}	Power Supply Current **4 MHz**		**20**	**mA**	V_{CC} = 5V
	6 MHz		**30**	**mA**	V_{IH} = V_{CC} − 0.2V
	8 MHz		**40**	**mA**	V_{IL} = 0.2V
	10 MHz		**50**	**mA**	
	20 MHz		100	mA	V_{cc} = 5V
I_{CC_2}	Standby Supply Current		10	μA	V_{CC} = 5V
					$\overline{CLK}$ = (0)
					V_{IH} = V_{CC} − 0.2V
					V_{IL} = 0.2V
I_{LI}	Input Leakage Current	−10	10	μA	V_{IN} = 0.4 to V_{CC}
I_{LO}	3-State Output Leakage Current in Float	−10	10[2]	μA	V_{OUT} = 0.4 to V_{CC}

1. Measurements made with outputs floating.
2. A_{15}-A_0, D_7-D_0, $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$.
3. I_{CC_2} standby supply current is guaranteed only when the supplied clock is stopped at a low level during T_4 of the machine cycle immediately following the execution of a HALT instruction.

CAPACITANCE

Symbol	Parameter	Min	Max	Unit
C_{CLOCK}	Clock Capacitance		**10**	pf
C_{IN}	Input Capacitance		5	pf
C_{OUT}	Output Capacitance		15	pf

T_A = 25°C, f = **1** MHz.
Unmeasured pins returned to ground.

AC CHARACTERISTICS† (Z84C00/CMOS Z80 CPU)

V_{CC}=5.0V ± 10%, unless otherwise specified

No	Symbol	Parameter	Z84C0004** Min	Z84C0004** Max	Z84C0006 Min	Z84C0006 Max	Z84C0008 Min	Z84C0008 Max	Z84C0010 Min	Z84C0010 Max	Z84C0020[1] Min	Z84C0020[1] Max	Unit	Note
1	TcC	Clock Cycle time	250*	DC	162*	DC	125*	DC	100*	DC	50*	DC	nS	
2	TwCh	Clock Pulse width (high)	110	DC	65	DC	55	DC	40	DC	20	DC	nS	
3	TwCl	Clock Pulse width (low)	110	DC	65	DC	55	DC	40	DC	20	DC	nS	
4	TfC	Clock Fall time		30		20		10		10		10	nS	
5	TrC	Clock Rise time		30		20		10		10		10	nS	
6	TdCr(A)	Address vaild from Clock Rise		110		90		80		65		57	nS	[2]
7	TdA(MREQf)	Address valid to /MREQ Fall	65*		35*		20*		5*		-15*		nS	
8	TdCf(MREQf)	Clock Fall to /MREQ Fall delay		85		70		60		55		40	nS	
9	TdCr(MREQr)	Clock Rise to /MREQ Rise delay		85		70		60		55		40	nS	
10	TwMREQh	/MREQ pulse width (High)	110*		65*		45**		30*		10*		nS	[3]
11	TwMREQl	/MREQ pulse width (low)	220*		132*		100*		75*		25*		nS	[3]
12	TdCf(MERQr)	Clock Fall to /MREQ Rise delay		85		70		60		55		40	nS	
13	TdCf(RDf)	Clock Fall to /RD Fall delay		95		80		70		65		40	nS	
14	TdCr(RDr)	Clock Rise to /RD Rise delay		85		70		60		55		40	nS	
15	TsD(Cr)	Data setup time to Clock Rise	35		30		30		25		12		nS	
16	ThD(RDr)	Data hold time after /RD Rise	0		0		0		0		0		nS	
17	TsWAIT(Cf)	/WAIT setup time to Clock Fall	70		60		50		20		7.5		nS	
18	ThWAIT(Cf)	/WAIT hold time after Clock Fall	10		10		10		10		10		nS	
19	TdCr(M1f)	Clock Rise to /M1 Fall delay		100		80		70		65		45	nS	
20	TdCr(M1r)	Clock Rise to /M1 Rise delay		100		80		70		65		45	nS	
21	TdCr(RFSHf)	Clock Rise to /RFSH Fall delay		130		110		95		80		60	nS	
22	TdCr(RFSHr)	Clock Rise to /RFSH Rise delay		120		100		85		80		60	nS	
23	TdCf(RDr)	Clock Fall to /RD Rise delay		85		70		60		55		40	nS	
24	TdCr(RDf)	Clock Rise to /RD Fall delay		85		70		60		55		40	nS	
25	TsD(Cf)	Data setup to Clock Fall during M2, M3, M4 or M5 cycles	50		40		30		25		12		nS	
26	TdA(IORQf)	Address stable prior to /IORQ Fall	180*		107*		75*		50*		0*		nS	
27	TdCr(IORQf)	Clock Rise to /IORQ Fall delay		75		65		55		50		40	nS	
28	TdCf(IORQr)	Clock Fall to /IORQ Rise delay		85		70		60		55		40	nS	
29	TdD(WRf)Mw	Data stable prior to /WR Fall	80*		22*		5*		40*		-10*		nS	
30	TdCf(WRf)	Clock Fall to /WR Fall delay		80		70		60		55		40	nS	
31	TwWR	/WR pulse width	220*		132*		100*		75*		25*		nS	
32	TdCf(WRr)	Clock Fall to /WR Rise delay		80		70		60		55		40	nS	
33	TdD(WRf)IO	Data stable prior to /WR Fall	-10*		-55*		-55*		-10*		-30*		nS	
34	TdCr(WRf)	Clock Rise to /WR Fall delay		65		60		60		50		40	nS	
35	TdWRr(D)	Data stable from /WR Rise	60*		30*		15*		10*		0*		nS	
36	TdCf(HALT)	Clock Fall to /HALT 'L' or 'H'		300		260		225		90		70	nS	
37	TwNMI	/NMI pulse width	80		60		60		60		60		nS	
38	TsBUSREQ (Cr)	/BUSREQ setup time to Clock Rise	50		50		40		30		15		nS	

*For clock periods other than the minimums shown, calculate parameters using the table on the following page. Calculated values above assumed TrC = TfC = 20 ns.

†Units in nanoseconds (ns)

†† **For loading ≥ 50 pf. Decrease width by 10 ns for each additional 50 pf.**

**4 MHz CMOS Z80 is obsoleted and replaced by 6 MHz

AC CHARACTERISTICS† (Z84C00/CMOS Z80 CPU; Continued)

V_{cc}=5.0V ± 10%, unless otherwise specified

No	Symbol	Parameter	Z84C0004** Min	Z84C0004** Max	Z84C0006 Min	Z84C0006 Max	Z84C0008 Min	Z84C0008 Max	Z84C0010 Min	Z84C0010 Max	Z84C0020[1] Min	Z84C0020[1] Max	Unit	Note
39	ThBUSREQ (Cr)	/BUSREQ hold time after Clock Rise	10		10		10		10		10		nS	
40	TdCr (BUSACKf)	Clock Rise to /BASACK Fall delay		100		90		80		75		40	nS	
41	TdCf (BUSACKr)	Clock Fall to /BASACK Rise delay		100		90		80		75		40	nS	
42	TdCr(Dz)	Clock Rise to Data float delay		90		80		70		65		40	nS	
43	TdCr(CTz)	Clock Rise to Control Outputs Float Delay (/MREQ, /IORQ, /RD and /WR)		80		70		60		65		40	nS	
44	TdCr(Az)	Clock Rise to Address float delay		90		80		70		75		40	nS	
45	TdCTr(A)	Address Hold time from /MREQ, /IORQ, /RD or /WR	80*		35*		20*		20*		0*		nS	
46	TsRESET(Cr)	/RESET to Clock Rise setup time	60		60		45		40		15		nS	
47	ThRESET(Cr)	/RESET to Clock Rise Hold time	10		10		10		10		10		nS	
48	TsINTf(Cr)	/INT Fall to Clock Rise Setup Time	80		70		55		50		15		nS	
49	ThINTr(Cr)	/INT Rise to Clock Rise Hold Time	10		10		10		10		10		nS	
50	TdM1f (IORQf)	/M1 Fall to /IORQ Fall delay	565*		359*		270*		220*		100*		nS	
51	TdCf(IORQf)	/Clock Fall to /IORQ Fall delay		85		70		60		55		45	nS	
52	TdCf(IORQr)	Clock Rise to /IORQ Rise delay		85		70		60		55		45	nS	
53	TdCf(D)	Clock Fall to Data Valid delay		150		130		115		110		75	nS	

Notes:

* For Clock periods other than the minimum shown, calculate parameters using the following table.
Calculated values above assumed TrC = TfC = maximum.

** 4 MHz CMOS Z80 is obsoleted and replaced by 6 MHz

[1] Z84C0020 parameters are guuaranteed with 50pF load Capacitance.

[2] If Capacitive Load is other than 50pF, please use Figure 1. to calculate the value.

[3] Increasing delay by 10nS for each 50pF increase in loading, 200pF max for data lines, and 100pF for control lines.

FOOTNOTES TO AC CHARACTERISTICS

No	Symbol	Parameter	Z84C0004**	Z84C0006	Z84C0008	Z84C0010	Z84C0020
1	TcC	TwCh + TwCl + TrC + TfC					
7	TdA(MREQf)	TwCh + TfC	-65	-50	-45	-45	-45
10	TwMREQh	TwCh + TfC	-20	-20	-20	-20	-20
11	TwMREQl	TcC	-30	-30	-25	-25	-25
26	TdA(IORQf)	TcC	-70	-55	-50	-50	-50
29	TdD(WRf)	TcC	-170	-140	-120	-60	-60
31	TwWR	TcC	-30	-30	-25	-25	-25
33	TdD(WRf)	TwCl + TrC	-140	-140	-120	-60	-60
35	TdWRr(D)	TwCl + TrC	-70	-55	-50	-40	-25
45	TdCTr(A)	TwCl + TrC	-50	-50	-45	-30	-30
50	TdM1f(IORQf)	2TcC + TwCh + TfC	-65	-50	-45	-30	-30

AC Test Conditions: V_{IH} = 2.0 V, V_{IL} = 0.8 V; V_{OH} = 1.5 V, V_{OL} = 1.5 V; V_{IHC} = V_{CC} − 0.6 V, V_{ILC} = 0.45 V; FLOAT = ±0.5 V

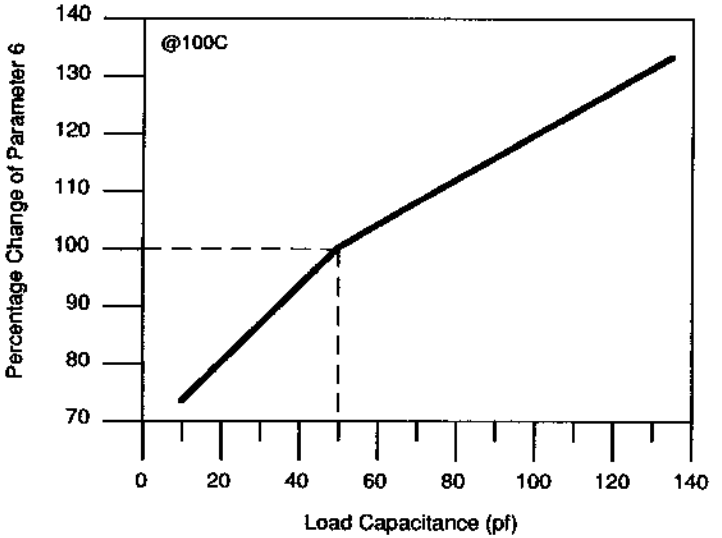


Figure 1. Address Delay Characteristics
(Parameter 6)

DC CHARACTERISTICS (Z8400/NMOS Z80 CPU)

All parameters are tested 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} − .6	V_{CC} + .3	V	
V_{IL}	Input Low Voltage	−0.3	0.8	V	
V_{IH}	Input High Voltage	2.0[1]	V_{CC}	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2.0 mA
V_{OH}	Output High Voltage	2.4[1]		V	I_{OH} = −250 μA
I_{CC}	Power Supply Current		200	mA	Note 3
I_{LI}	Input Leakage Current		10	μA	V_{IN} = 0 to V_{CC}
I_{LO}	3-State Output Leakage Current in Float	−10	10[2]	μA	V_{OUT} = 0.4 to V_{CC}

1. For military grade parts, refer to the Z80 Military Electrical Specification.
2. A_{15}-A_0, D_7-D_0, $\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$.
3. Measurements made with outputs floating

CAPACITANCE

Guaranteed by design and characterization.

Symbol	Parameter	Min	Max	Unit
C_{CLOCK}	Clock Capacitance		35	pf
C_{IN}	Input Capacitance		5	pf
C_{OUT}	Output Capacitance		15	pf

NOTES:
T_A = 25°C, f = 1 MHz.
Unmeasured pins returned to ground.

AC CHARACTERISTICS† (Z8400/NMOS Z80 CPU)

Number	Symbol	Parameter	Z0840004 Min	Z0840004 Max	Z0840006 Min	Z0840006 Max	Z0840008 Min	Z0840008 Max
1	TcC	Clock Cycle Time	250*		162*		125*	
2	TwCh	Clock Pulse Width (High)	110	2000	65	2000	55	2000
3	TwCl	Clock Pulse Width (Low)	110	2000	65	2000	55	2000
4	TfC	Clock Fall Time		30		20		10
5	TrC	Clock Rise Time		30		20		10
6	TdCr(A)	Clock ↑ to Address Valid Delay		110		90		80
7	TdA(MREQf)	Address Valid to $\overline{MREQ}$ ↓ Delay	65*		35*		20*	
8	TdCf(MREQf)	Clock ↓ to $\overline{MREQ}$ ↓ Delay		85		70		60
9	TdCr(MREQr)	Clock ↑ to $\overline{MREQ}$ ↑ Delay		85		70		60
10	TwMREQh	$\overline{MREQ}$ Pulse Width (High)	110*‡		65*‡		45*‡	
11	TwMREQl	$\overline{MREQ}$ Pulse Width (Low)	220*‡		135*‡		100*‡	
12	TdCf(MREQr)	Clock ↓ to $\overline{MREQ}$ ↑ Delay		85		70		60
13	TdCf(RDf)	Clock ↓ to $\overline{RD}$ ↓ Delay		95		80		70
14	TdCr(RDr)	Clock ↑ to $\overline{RD}$ ↑ Delay		85		70		60
15	TsD(Cr)	Data Setup Time to Clock ↑	35		30		30	
16	ThD(RDr)	Data Hold Time to $\overline{RD}$ ↑		0		0		0
17	TsWAIT(Cf)	$\overline{WAIT}$ Setup Time to Clock ↓	70		60		50	
18	ThWAIT(Cf)	$\overline{WAIT}$ Hold Time after Clock ↓		0		0		0
19	TdCr(M1f)	Clock ↑ to $\overline{M1}$ ↓ Delay		100		80		70
20	TdCr(M1r)	Clock ↑ to $\overline{M1}$ ↑ Delay		100		80		70
21	TdCr(RFSHf)	Clock ↑ to $\overline{RFSH}$ ↓ Delay		130		110		95
22	TdCr(RFSHr)	Clock ↑ to $\overline{RFSH}$ ↑ Delay		120		100		85
23	TdCf(RDr)	Clock ↓ to $\overline{RD}$ ↑ Delay		85		70		60
24	TdCr(RDf)	Clock ↑ to $\overline{RD}$ ↓ Delay		85		70		60
25	TsD(Cf)	Data Setup to Clock ↓ during M_2, M_3, M_4, or M_5 Cycles	50		40		30	
26	TdA(IORQf)	Address Stable prior to $\overline{IORQ}$ ↓	180*		110*		75*	
27	TdCr(IORQf)	Clock ↑ to $\overline{IORQ}$ ↓ Delay		75		65		55
28	TdCf(IORQr)	Clock ↓ to $\overline{IORQ}$ ↑ Delay		85		70		60
29	TdD(WRf)	Data Stable prior to $\overline{WR}$ ↓	80*		25*		5*	
30	TdCf(WRf)	Clock ↓ to $\overline{WR}$ ↓ Delay		80		70		60
31	TwWR	$\overline{WR}$ Pulse Width	220*		135*		100*	
32	TdCf(WRr)	Clock ↓ to $\overline{WR}$ ↑ Delay		80		70		60
33	TdD(WRf)	Data Stable prior to $\overline{WR}$ ↓	−10*		−55*		55*	
34	TdCr(WRf)	Clock ↑ to $\overline{WR}$ ↓ Delay		65		60		55
35	TdWRr(D)	Data Stable from $\overline{WR}$ ↑	60*		30*		15*	
36	TdCf(HALT)	Clock ↓ to $\overline{HALT}$ ↑ or ↓		300		260		225
37	TwNMI	$\overline{NMI}$ Pulse Width	80		70		60*	
38	TsBUSREQ(Cr)	$\overline{BUSREQ}$ Setup Time to Clock ↑	50		50		40	

*For clock periods other than the minimums shown, calculate parameters using the table on the following page. Calculated values above assumed TrC = TfC = 20 ns.

†Units in nanoseconds (ns).

‡ For loading ≥ 50 pf., Decrease width by 10 ns for each additional 50 pf.

AC CHARACTERISTICS† (Z8400/NMOS Z80 CPU; Continued)

Number	Symbol	Parameter	Z0840004 Min	Z0840004 Max	Z0840006 Min	Z0840006 Max	Z0840008 Min	Z0840008 Max
39	ThBUSREQ(Cr)	$\overline{BUSREQ}$ Hold Time after Clock ↑	0		0		0	
40	TdCr(BUSACKf)	Clock ↑ to $\overline{BUSACK}$ ↓ Delay		100		90		80
41	TdCf(BUSACKr)	Clock ↓ to $\overline{BUSACK}$ ↑ Delay		100		90		80
42	TdCr(Dz)	Clock ↑ to Data Float Delay		90		80		70
43	TdCr(CTz)	Clock ↑ to Control Outputs Float Delay ($\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$, and $\overline{WR}$)		80		70		60
44	TdCr(Az)	Clock ↑ to Address Float Delay		90		80		70
45	TdCTr(A)	$\overline{MREQ}$ ↑, $\overline{IORQ}$ ↑, $\overline{RD}$ ↑, and $\overline{WR}$ ↑ to Address Hold Time	80*		35*		20*	
46	TsRESET(Cr)	$\overline{RESET}$ to Clock ↑ Setup Time	60		60		45	
47	ThRESET(Cr)	$\overline{RESET}$ to Clock ↑ Hold Time		0		0		0
48	TsINTf(Cr)	$\overline{INT}$ to Clock ↑ Setup Time	80		70		55	
49	ThINTr(Cr)	$\overline{INT}$ to Clock ↑ Hold Time		0		0		0
50	TdM1f(IORQf)	$\overline{M1}$ ↓ to $\overline{IORQ}$ ↓ Delay	565*		365*		270*	
51	TdCf(IORQf)	Clock ↓ to $\overline{IORQ}$ ↓ Delay		85		70		60
52	TdCf(IORQr)	Clock ↑ $\overline{IORQ}$ ↑ Delay		85		70		60
53	TdCf(D)	Clock ↓ to Data Valid Delay		150		130		115

*For clock periods other than the minimums shown, calculate parameters using the following table. Calculated values above assumed TrC = TfC = 20 ns.
†Units in nanoseconds (ns).

FOOTNOTES TO AC CHARACTERISTICS

Number	Symbol	General Parameter	Z0840004	Z0840006	Z0840008
1	TcC	TwCh + TwCl + TrC + TfC			
7	TdA(MREQf)	TwCh + TfC	− 65	− 50	− 45
10	TwMREQh	TwCh + TfC	− 20	− 20	− 20
11	TwMREQl	TcC	− 30	− 30	− 25
26	TdA(IORQf)	TcC	− 70	− 55	− 50
29	TdD(WRf)	TcC	− 170	− 140	− 120
31	TwWR	TcC	− 30	− 30	− 25
33	TdD(WRf)	TwCl + TrC	− 140	− 140	− 120
35	TdWRr(D)	TwCl + TrC	− 70	− 55	− 50
45	TdCTr(A)	TwCl + TrC	− 50	− 50	− 45
50	TdM1f(IORQf)	2TcC + TwCh + TfC	− 65	− 50	− 45

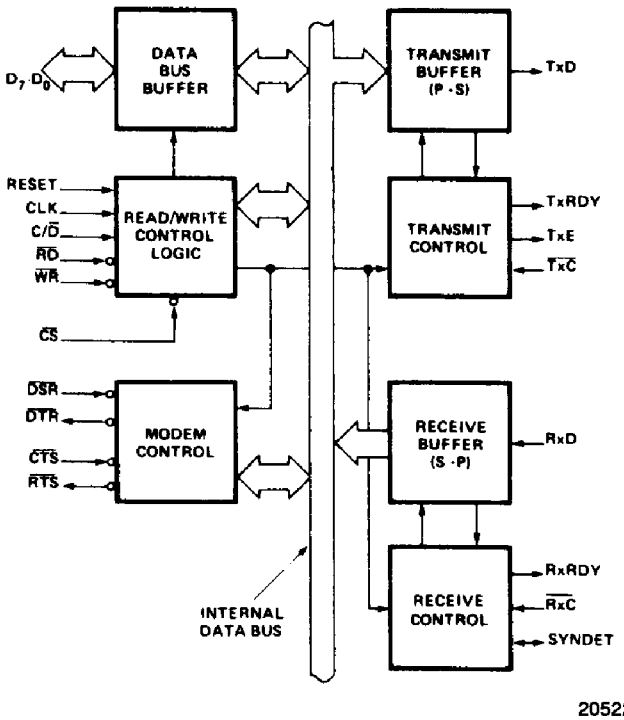
AC Test Conditions:
V_{IH} = 2.0 V
V_{IL} = 0.8 V
$V_{IHC} = V_{CC} - 0.6$ V
V_{ILC} = 0.45 V
V_{OH} = 1.5 V
V_{OL} = 1.5 V
FLOAT = ±0.5 V

8251A
PROGRAMMABLE COMMUNICATION INTERFACE

- **Synchronous and Asynchronous Operation**
- **Synchronous 5-8 Bit Characters; Internal or External Character Synchronization; Automatic Sync Insertion**
- **Asynchronous 5-8 Bit Characters; Clock Rate—1, 16 or 64 Times Baud Rate; Break Character Generation; 1, 1½, or 2 Stop Bits; False Start Bit Detection; Automatic Break Detect and Handling**
- **Synchronous Baud Rate—DC to 64K Baud**
- **Asynchronous Baud Rate—DC to 19.2K Baud**
- **Full-Duplex, Double-Buffered Transmitter and Receiver**
- **Error Detection—Parity, Overrun and Framing**
- **Compatible with an Extended Range of Intel Microprocessors**
- **28-Pin DIP Package**
- **All Inputs and Outputs are TTL Compatible**
- **Available in EXPRESS and Military Versions**

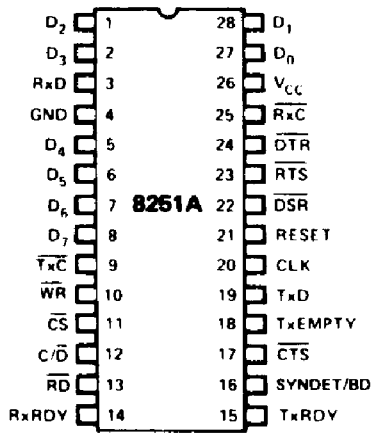
2

The Intel® 8251A is the industry standard Universal Synchronous/Asynchronous Receiver/Transmitter (USART), designed for data communications with Intel's microprocessor families such as MCS-48, 80, 85, and iAPX-86, 88. The 8251A is used as a peripheral device and is programmed by the CPU to operate using virtually any serial data transmission technique presently in use (including IBM "bi-sync"). The USART accepts data characters from the CPU in parallel format and then converts them into a continuous serial data stream for transmission. Simultaneously, it can receive serial data streams and convert them into parallel data characters for the CPU. The USART will signal the CPU whenever it can accept a new character for transmission or whenever it has received a character for the CPU. The CPU can read the complete status of the USART at any time. These include data transmission errors and control signals such as SYNDET, TxEMPTY. The chip is fabricated using Intel's high performance HMOS technology.



205222-1

Figure 1. Block Diagram



205222-2

Figure 2. Pin Configuration

November 1986
Order Number: 205222-002

FEATURES AND ENHANCEMENTS

The 8251A is an advanced design of the industry standard USART, the Intel® 8251. The 8251A operates with an extended range of Intel microprocessors and maintains compatibility with the 8251. Familiarization time is minimal because of compatibility and involves only knowing the additional features and enhancements, and reviewing the AC and DC specifications of the 8251A.

The 8251A incorporates all the key features of the 8251 and has the following additional features and enhancements:

- 8251A has double-buffered data paths with separate I/O registers for control, status, Data In, and Data Out, which considerably simplifies control programming and minimizes CPU overhead.
- In asynchronous operations, the Receiver detects and handles "break" automatically, relieving the CPU of this task.
- A refined Rx initialization prevents the Receiver from starting when in "break" state, preventing unwanted interrupts from a disconnected USART.
- At the conclusion of a transmission, TxD line will always return to the marking state unless SBRK is programmed.
- Tx Enable logic enhancement prevents a Tx Disable command from halting transmission until all data previously written has been transmitted. The logic also prevents the transmitter from turning off in the middle of a word.
- When External Sync Detect is programmed, Internal Sync Detect is disabled, and an External Sync Detect status is provided via a flip-flop which clears itself upon a status read.
- Possibility of false sync detect is minimized by ensuring that if double character sync is programmed, the characters be contiguously detected and also by clearing the Rx register to all ones whenever Enter Hunt command is issued in Sync mode.
- As long as the 8251A is not selected, the $\overline{RD}$ and $\overline{WR}$ do not affect the internal operation of the device.
- The 8251A Status can be read at any time but the status update will be inhibited during status read.
- The 8251A is free from extraneous glitches and has enhanced AC and DC characteristics, providing higher speed and better operating margins.
- Synchronous Baud rate from DC to 64K.

FUNCTIONAL DESCRIPTION

General

The 8251A is a Universal Synchronous/Asynchronous Receiver/Transmitter designed for a wide range of Intel microcomputers such as 8048, 8080, 8085, 8086 and 8088. Like other I/O devices in a microcomputer system, its functional configuration is programmed by the system's software for maximum flexibility. The 8251A can support most serial data techniques in use, including IBM "bi-sync".

In a communication environment an interface device must convert parallel format system data into serial format for transmission and convert incoming serial format data into parallel system data for reception. The interface device must also delete or insert bits or characters that are functionally unique to the communication technique. In essence, the interface should appear "transparent" to the CPU, a simple input or output of byte-oriented system data.

Data Bus Buffer

This 3-state bidirectional, 8-bit buffer is used to interface the 8251A to the system Data Bus. Data is transmitted or received by the buffer upon execution of INput or OUTput instructions of the CPU. Control words, Command words and Status information are also transferred through the Data Bus Buffer. The Command Status, Data-In and Data-Out registers are separate, 8-bit registers communicating with the system bus through the Data Bus Buffer.

This functional block accepts inputs from the system Control bus and generates control signals for overall device operation. It contains the Control Word Register and Command Word Register that store the various control formats for the device functional definition.

RESET (Reset)

A "high" on this input forces the 8251A into an "Idle" mode. The device will remain at "Idle" until a new set of control words is written into the 8251A to program its functional definition. Minimum RESET pulse width is 6 t_{CY} (clock must be running).

A command reset operation also puts the device into the "Idle" state.

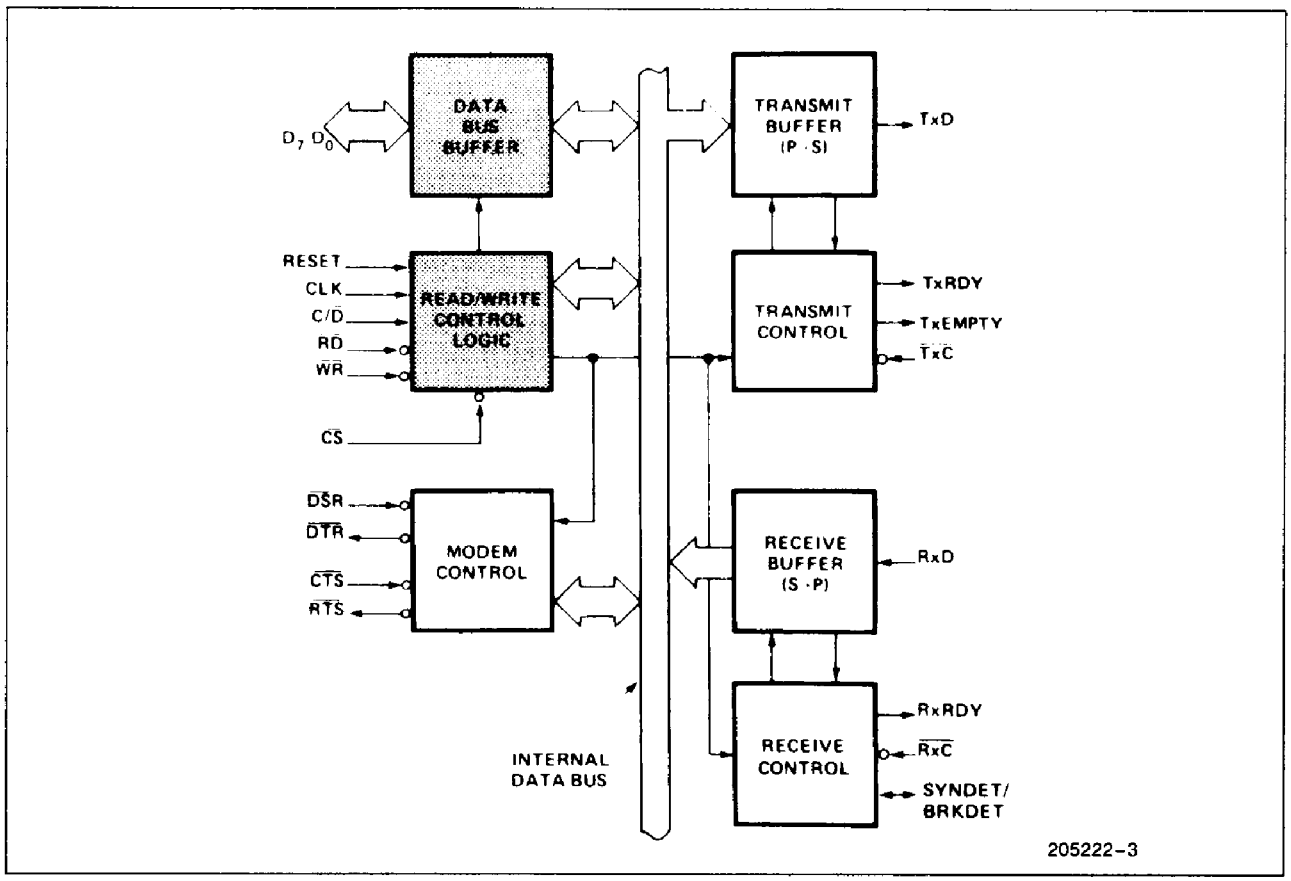


Figure 3. 8251A Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

CLK (Clock)

The CLK input is used to generate internal device timing and is normally connected to the Phase 2 (TTL) output of the Clock Generator. No external inputs or outputs are referenced to CLK but the frequency of CLK must be greater than 30 times the Receiver or Transmitter data bit rates.

$\overline{WR}$ (Write)

A "low" on this input informs the 8251A that the CPU is writing data or control words to the 8251A.

$\overline{RD}$ (Read)

A "low" on this input informs the 8251A that the CPU is reading data or status information from the 8251A.

C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
0	0	1	0	8251A DATA → DATA BUS
0	1	0	0	DATA BUS → 8251A DATA
1	0	1	0	STATUS → DATA BUS
1	1	0	0	DATA BUS → CONTROL
X	1	1	0	DATA BUS → 3-STATE
X	X	X	1	DATA BUS → 3-STATE

C/$\overline{D}$ (Control/Data)

This input, in conjunction with the $\overline{WR}$ and $\overline{RD}$ inputs, informs the 8251A that the word on the Data Bus is either a data character, control word or status information.

1 = CONTROL/STATUS; 0 = DATA.

$\overline{CS}$ (Chip Select)

A "low" on this input selects the 8251A. No reading or writing will occur unless the device is selected. When $\overline{CS}$ is high, the Data Bus is in the float state and $\overline{RD}$ and $\overline{WR}$ have no effect on the chip.

Modem Control

The 8251A has a set of control inputs and outputs that can be used to simplify the interface to almost any modem. The modem control signals are general purpose in nature and can be used for functions other than modem control, if necessary.

$\overline{DSR}$ (Data Set Ready)

The $\overline{DSR}$ input signal is a general-purpose, 1-bit inverting input port. Its condition can be tested by the CPU using a Status Read operation. The $\overline{DSR}$ input is normally used to test modem conditions such as Data Set Ready.

$\overline{DTR}$ (Data Terminal Ready)

The $\overline{DTR}$ output signal is a general-purpose, 1-bit inverting output port. It can be set "low" by programming the appropriate bit in the Command Instruction word. The $\overline{DTR}$ output signal is normally used for modem control such as Data Terminal Ready.

$\overline{RTS}$ (Request to Send)

The $\overline{RTS}$ output signal is a general-purpose, 1-bit inverting output port. It can be set "low" by programming the appropriate bit in the Command Instruction word. The $\overline{RTS}$ output signal is normally used for modem control such as Request to Send.

$\overline{CTS}$ (Clear to Send)

A "low" on this input enables the 8251A to transmit serial data if the Tx Enable bit in the Command byte is set to a "one". If either a Tx Enable off or $\overline{CTS}$ off condition occurs while the Tx is in operation, the Tx will transmit all the data in the USART, written prior to Tx Disable command before shutting down.

Transmitter Buffer

The Transmitter Buffer accepts parallel data from the Data Bus Buffer, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin on the falling edge of $\overline{TxC}$. The transmitter will begin transmission upon being enabled if $\overline{CTS}$ = 0. The TxD line will be held in the marking state immediately upon a master Reset or when Tx Enable or $\overline{CTS}$ is off or the transmitter is empty.

Transmitter Control

The Transmitter Control manages all activities associated with the transmission of serial data. It accepts and issues signals both externally and internally to accomplish this function.

TxRDY (Transmitter Ready)

This output signals the CPU that the transmitter is ready to accept a data character. The TxRDY output pin can be used as an interrupt to the system, since it is masked by TxEnable; or, for Polled operation, the CPU can check TxRDY using a Status Read operation. TxRDY is automatically reset by the leading edge of $\overline{WR}$ when a data character is loaded from the CPU.

Note that when using the Polled operation, the TxRDY status bit is *not* masked by TxEnable, but will only indicate the Empty/Full Status of the Tx Data Input Register.

TxE (Transmitter Empty)

When the 8251A has no characters to send, the TxEMPTY output will go "high". It resets upon receiving a character from CPU if the transmitter is enabled. TxEMPTY remains high when the transmitter is disabled. TxEMPTY can be used to indicate the end of a transmission mode, so that the CPU "knows" when to "turn the line around" in the half-duplex operational mode.

In the Synchronous mode, a "high" on this output indicates that a character has not been loaded and the SYNC character or characters are about to be or are being transmitted automatically as "fillers". Tx EMPTY does not go low when the SYNC characters are being shifted out.

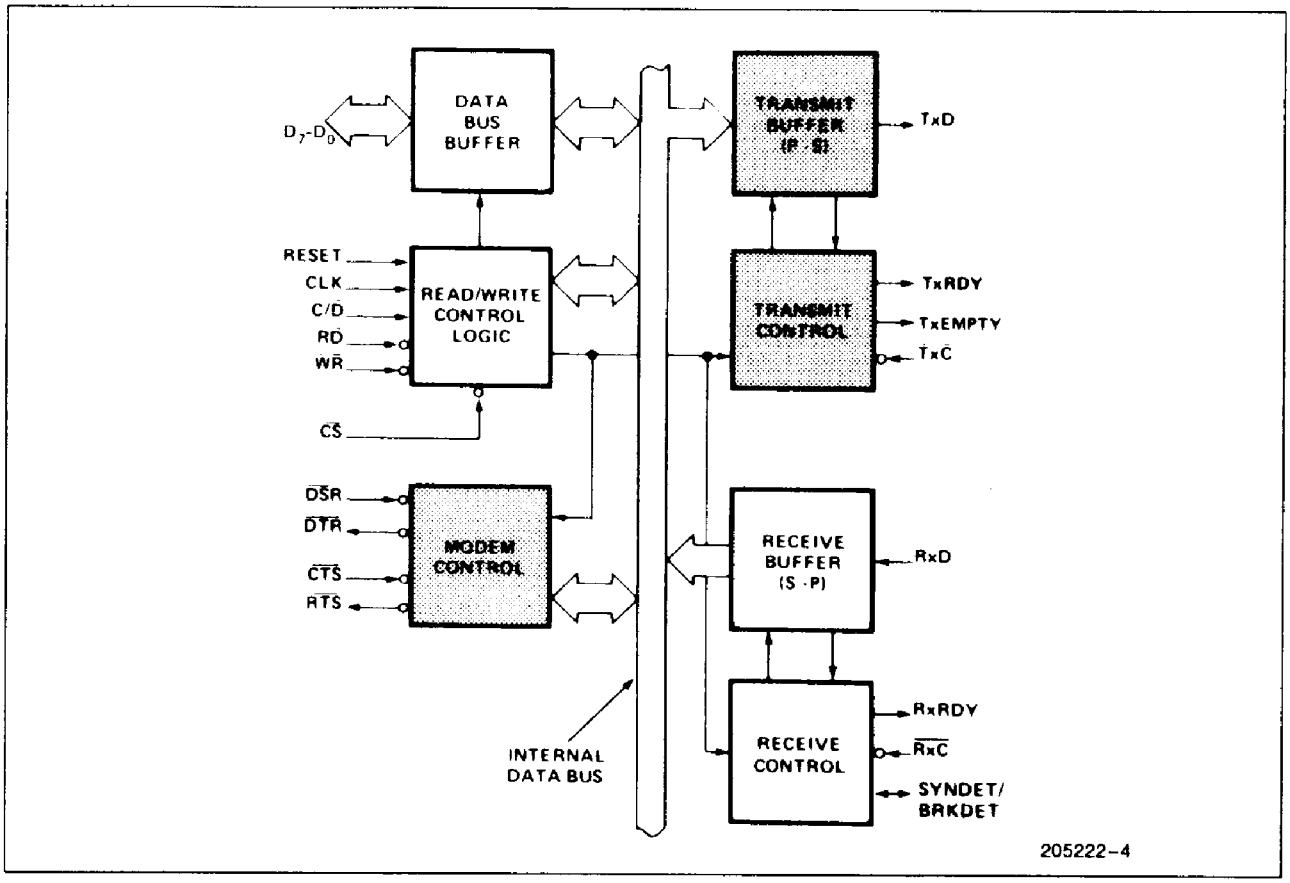


Figure 4. 8251A Block Diagram Showing Modem and Transmitter Buffer and Control Functions

TxC (Transmitter Clock)

The Transmitter Clock controls the rate at which the character is to be transmitted. In the Synchronous transmission mode, the Baud Rate (1x) is equal to the TxC frequency. In Asynchronous transmission mode, the baud rate is a fraction of the actual TxC frequency. A portion of the mode instruction selects this factor; it can be 1, 1/16 or 1/64 the TxC.

For Example:

If Baud Rate equals 110 Baud,
TxC equals 110 Hz in the 1x mode.
TxC equals 1.72 kHz in the 16x mode.
TxC equals 7.04 kHz in the 64x mode.

The falling edge of TxC shifts the serial data out of the 8251A.

Receiver Buffer

The Receiver accepts serial data, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU. Serial data is input to RxD pin, and is clocked in on the rising edge of RxC.

Receiver Control

This functional block manages all receiver-related activities which consists of the following features.

The RxD initialization circuit prevents the 8251A from mistaking an unused input line for an active low data line in the "break condition". Before starting to receive serial characters on the RxD line, a valid "1" must first be detected after a chip master Reset. Once this has been determined, a search for a valid low (Start bit) is enabled. This feature is only active in the asynchronous mode, and is only done once for each master Reset.

The False Start bit detection circuit prevents false starts due to a transient noise spike by first detecting the falling edge and then strobing the normal center of the Start bit (RxD = low).

Parity error detection sets the corresponding status bit.

The Framing Error status bit is set if the Stop bit is absent at the end of the data byte (asynchronous mode).

RxRDY (Receiver Ready)

This output indicates that the 8251A contains a character that is ready to be input to the CPU. RxRDY can be connected to the interrupt structure of the CPU or, for polled operation, the CPU can check the condition of RxRDY using a Status Read operation.

RxEnable, when off, holds RxRDY in the Reset Condition. For Asynchronous mode, to set RxRDY, the Receiver must be enabled to sense a Start Bit and a complete character must be assembled and transferred to the Data Output Register. For Synchronous mode, to set RxRDY, the Receiver must be enabled and a character must finish assembly and be transferred to the Data Output Register.

Failure to read the received character from the Rx Data Output Register prior to the assembly of the next Rx Data character will set overrun condition error and the previous character will be written over and lost. If the Rx Data is being read by the CPU when the internal transfer is occurring, overrun error will be set and the old character will be lost.

$\overline{\text{RxC}}$ (Receiver Clock)

The Receiver Clock controls the rate at which the character is to be received. In Synchronous Mode, the Baud Rate (1x) is equal to the actual frequency of $\overline{\text{RxC}}$. In Asynchronous Mode, the Baud Rate is a fraction of the actual $\overline{\text{RxC}}$ frequency. A portion of the mode instruction selects this factor: 1, 1/16 or 1/64 the $\overline{\text{RxC}}$.

For Example:

Baud Rate equals 300 Baud, if
$\overline{\text{RxC}}$ equals 300 Hz in the 1x mode;
$\overline{\text{RxC}}$ equals 4800 Hz in the 16x mode;
$\overline{\text{RxC}}$ equals 19.2 kHz in the 64x mode.

Baud Rate equals 2400 Baud, if
$\overline{\text{RxC}}$ equals 2400 Hz in the 1x mode;
$\overline{\text{RxC}}$ equals 38.4 kHz in the 16 mode;
$\overline{\text{RxC}}$ equals 153.6 kHz in the 64 mode.

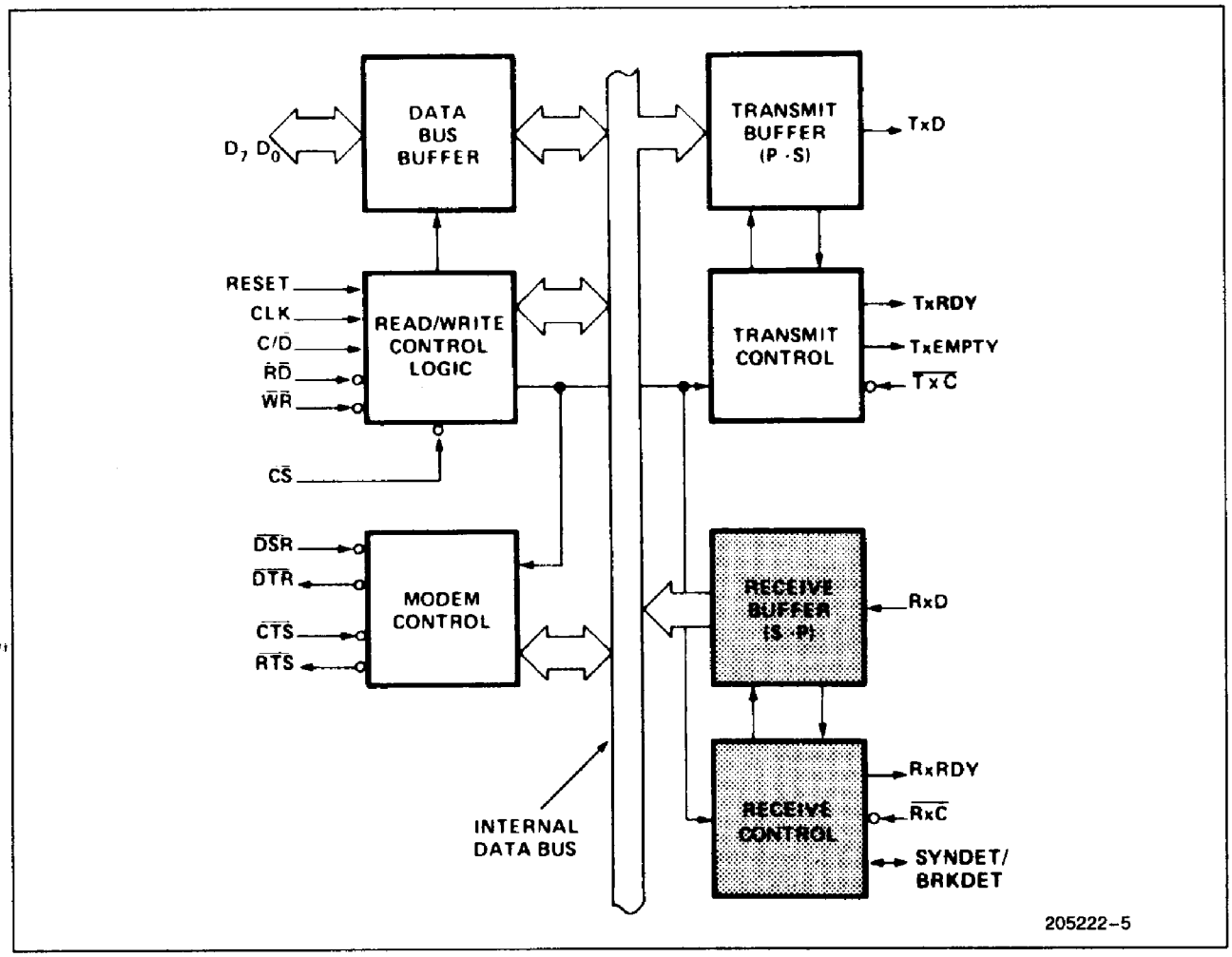


Figure 5. 8251A Block Diagram Showing Receiver Buffer and Control Functions

Data is sampled into the 8251A on the rising edge of $\overline{RxC}$.

NOTE:
In most communication systems, the 8251A will be handling both the transmission and reception operations of a single link. Consequently, the Receive and Transmit Baud Rates will be the same. Both $\overline{TxC}$ and $\overline{RxC}$ will require identical frequencies for this operation and can be tied together and connected to a single frequency source (Baud Rate Generator) to simplify the interface.

SYNDET (SYNC Detect/ BRKDET Break Detect)

This pin is used in Synchronous Mode for SYNDET and may be used as either input or output, programmable through the Control Word. It is reset to output mode low upon RESET. When used as an output (internal Sync mode), the SYNDET pin will go "high" to indicate that the 8251A has located the SYNC character in the Receive mode. If the 8251A is programmed to use double Sync characters (bi-sync), then SYNDET will go "high" in the middle of the last bit of the second Sync character. SYNDET is automatically reset upon a Status Read operation.

When used as an input (external SYNC detect mode), a positive going signal will cause the 8251A to start assembling data characters on the rising edge of the next $\overline{RxC}$. Once in SYNC, the "high" input signal can be removed. When External SYNC Detect is programmed, Internal SYNC Detect is disabled.

BREAK (Async Mode Only)

This output will go high whenever the receiver remains low through two consecutive stop bit sequences (including the start bits, data bits, and parity bits). Break Detect may also be read as a Status bit. It is reset only upon a master chip Reset or Rx Data returning to a "one" state.

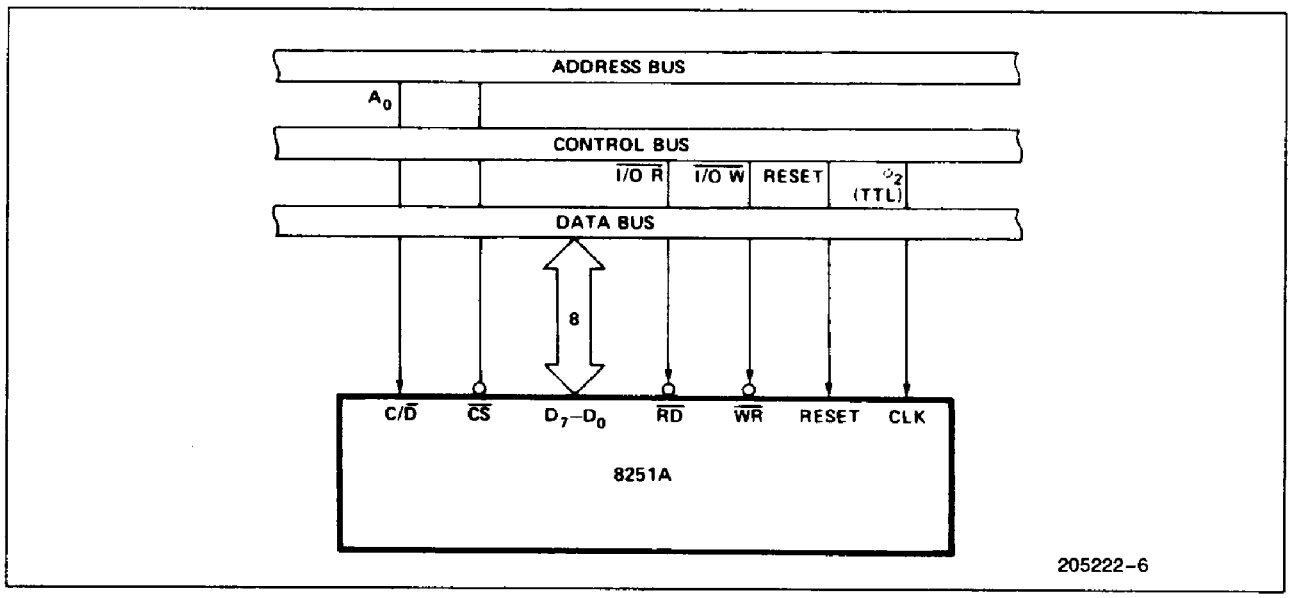


Figure 6. 8251A Interface to 8080 Standard System Bus

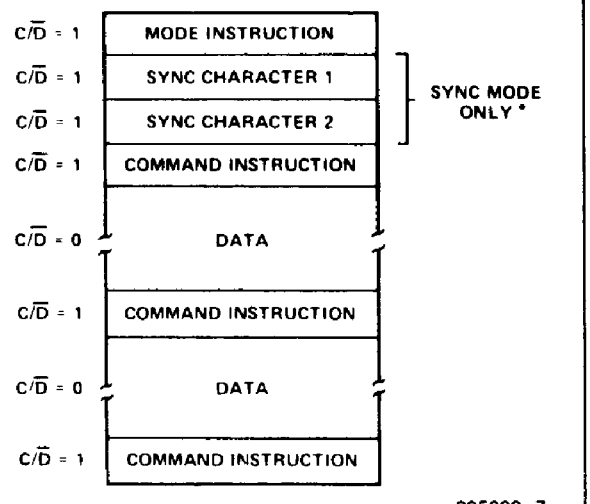
DETAILED OPERATION DESCRIPTION

General

The complete functional definition of the 8251A is programmed by the system's software. A set of control words must be sent out by the CPU to initialize the 8251A to support the desired communications format. These control words will program the: BAUD RATE, CHARACTER LENGTH, NUMBER OF STOP BITS, SYNCHRONOUS or ASYNCHRONOUS OPERATION, EVEN/ODD/OFF PARITY, etc. In the Synchronous Mode, options are also provided to select either internal or external character synchronization.

Once programmed, the 8251A is ready to perform its communication functions. The TxRDY output is raised "high" to signal the CPU that the 8251A is ready to receive a data character from the CPU. This output (TxRDY) is reset automatically when the CPU writes a character into the 8251A. On the other hand, the 8251A receives serial data from the MODEM or I/O device. Upon receiving an entire character, the RxRDY output is raised "high" to signal the CPU that the 8251A has a complete character ready for the CPU to fetch. RxRDY is reset automatically upon the CPU data read operation.

The 8251A cannot begin transmission until the Tx Enable (Transmitter Enable) bit is set in the Command Instruction and it has received a Clear To Send ($\overline{CTS}$) input. The TxD output will be held in the marking state upon Reset.



*The second sync character is skipped if mode instruction has programmed the 8251A to single character sync mode. Both sync characters are skipped if mode instruction has programmed the 8251A to async mode.

Figure 7. Typical Data Block

Programming the 8251A

Prior to starting data transmission or reception, the 8251A must be loaded with a set of control words generated by the CPU. These control signals define the complete functional definition of the 8251A and must immediately follow a Reset operation (internal or external).

The control words are split into two formats:

1. Mode Instruction
2. Command Instruction

Mode Instruction

This instruction defines the general operational characteristics of the 8251A. It must follow a Reset operation (internal or external). Once the Mode Instruction has been written into the 8251A by the CPU, SYNC characters or Command Instructions may be written.

Command Instruction

This instruction defines a word that is used to control the actual operation of the 8251A.

Both the Mode and Command Instructions must conform to a specified sequence for proper device operation (see Figure 7). The Mode Instruction must be written immediately following a Reset operation, prior to using the 8251A for data communication.

All control words written into the 8251A after the Mode Instruction will load the Command Instruction. Command Instructions can be written into the 8251A at any time in the data block during the operation of the 8251A. To return to the Mode Instruction format, the master Reset bit in the Command Instruction word can be set to initiate an internal Reset operation which automatically places the 8251A back into the Mode Instruction format. Command Instructions must follow the Mode Instruction or Sync characters.

Mode Instruction Definition

The 8251A can be used for either Asynchronous or Synchronous data communication. To understand how the Mode Instruction defines the functional operation of the 8251A, the designer can best view the device as two separate components, one Asynchronous and the other Synchronous, sharing the same package. The format definition can be changed only after a master chip Reset. For explanation purposes the two formats will be isolated.

NOTE:
When parity is enabled it is not considered as one of the data bits for the purpose of programming word length. The actual parity bit received on the Rx Data line cannot be read on the Data Bus. In the case of a programmed character length of less than 8 bits, the least significant Data Bus bits will hold the data; unused bits are "don't care" when writing data to the 8251A, and will be "zeros" when reading the data from the 8251A.

Asynchronous Mode (Transmission)

Whenever a data character is sent by the CPU the 8251A automatically adds a Start bit (low level) followed by the data bits (least significant bit first), and the programmed number of Stop bits to each character. Also, an even or odd Parity bit is inserted prior to the Stop bit(s), as defined by the Mode Instruction. The character is then transmitted as a serial data stream on the TxD output. The serial data is shifted out on the falling edge of $\overline{TxC}$ at a rate equal to 1, 1/16, or 1/64 that of the $\overline{TxC}$, as defined by the Mode Instruction. BREAK characters can be continuously sent to the TxD if commanded to do so.

When no data characters have been loaded into the 8251A the TxD output remains "high" (marking) unless a Break (continuously low) has been programmed.

Asynchronous Mode (Receive)

The RxD line is normally high. A falling edge on this line triggers the beginning of a START bit. The validity of this START bit is checked by again strobing this bit at its nominal center (16X or 64X mode only). If a low is detected again, it is a valid START bit, and the bit counter will start counting. The bit counter thus locates the center of the data bits, the parity bit (if it exists) and the stop bits. If parity error occurs, the parity error flag is set. Data and parity bits are sampled on the RxD pin with the rising edge of the $\overline{RxC}$. If a low level is detected as the STOP bit the Framing Error flag will be set. The STOP bit signals the end of a character. Note that the *receiver* requires only *one* stop bit, regardless of the number of stop bits programmed. This character is then loaded into the parallel I/O buffer of the 8251A. The RxRDY pin is raised to signal the CPU that a character is ready to be fetched. If a previous character has not been fetched by the CPU, the present character replaces it in the I/O buffer, and the OVERRUN Error flag

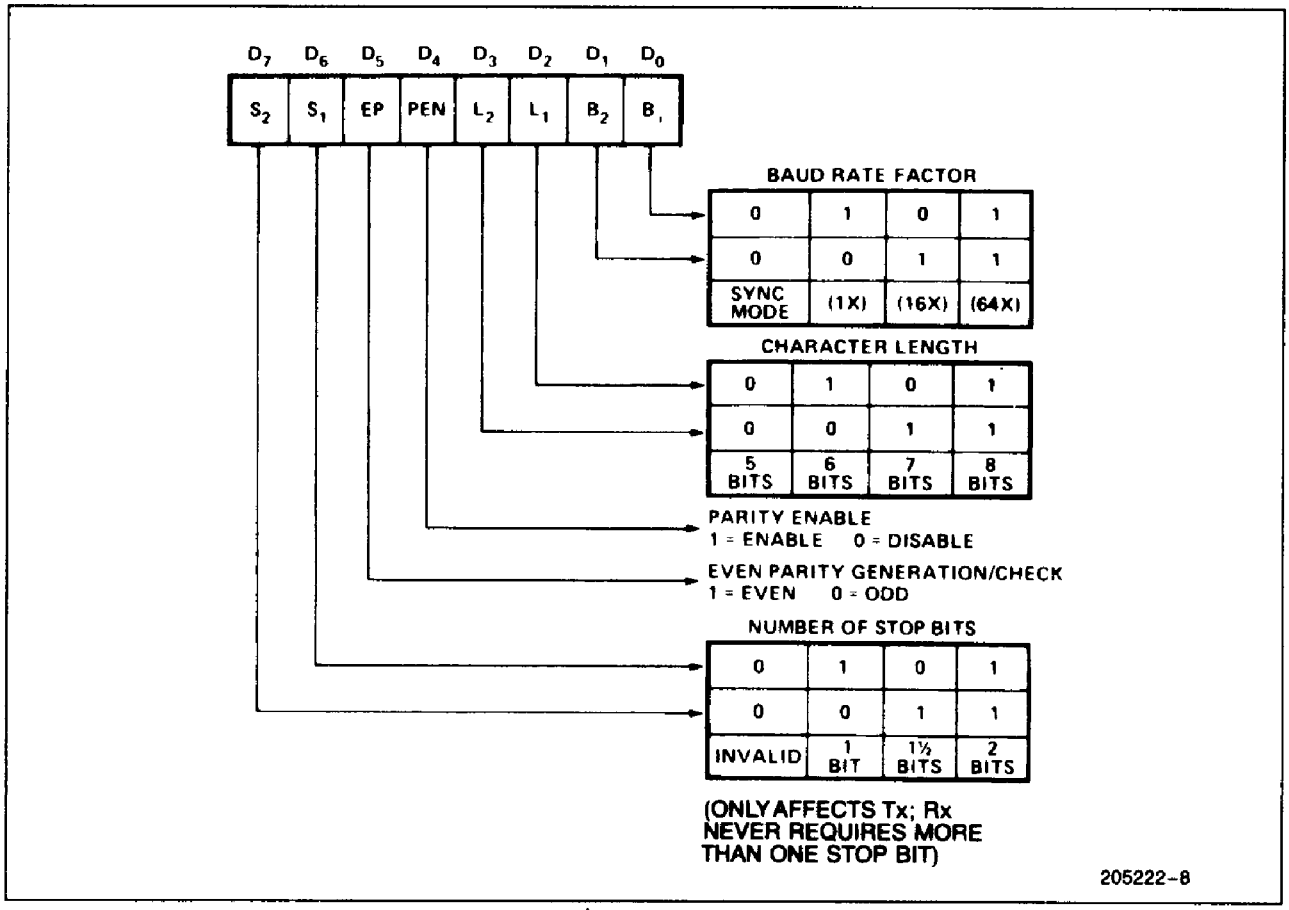


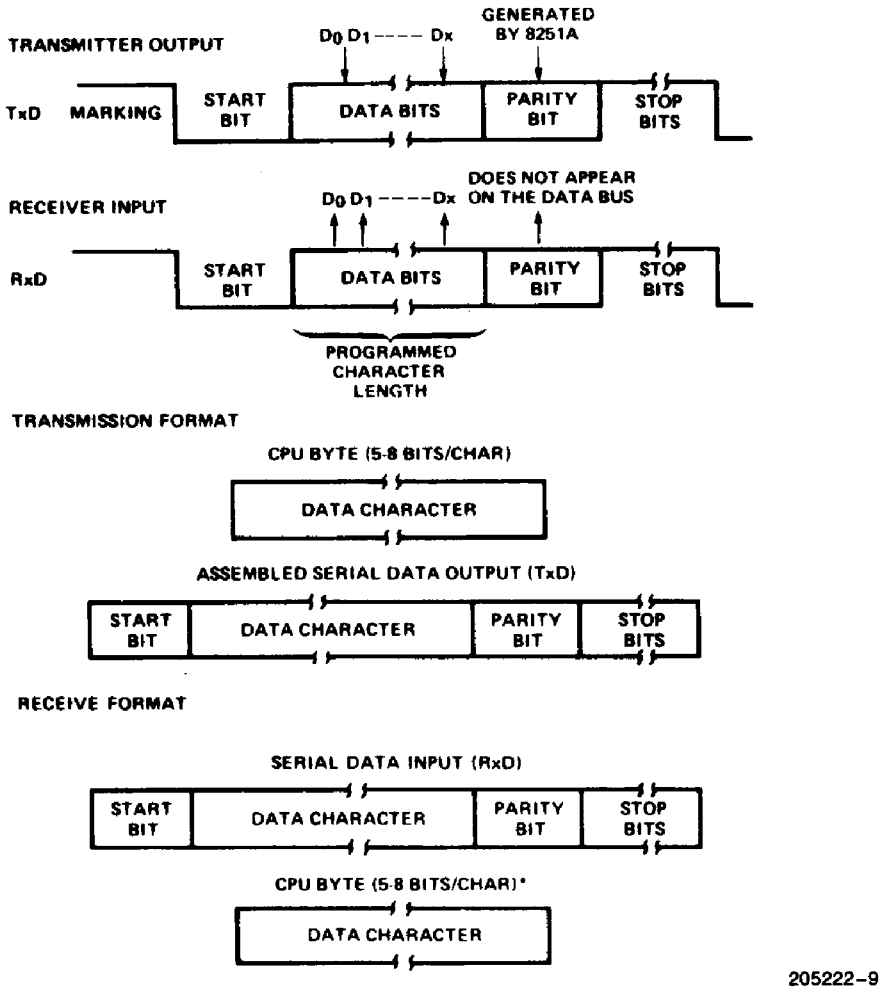
Figure 8. Mode Instruction Format, Asynchronous Mode

is raised (thus the previous character is lost). All of the error flags can be reset by an Error Reset Instruction. The occurrence of any of these errors will not affect the operation of the 8251A.

Synchronous Mode (Transmission)

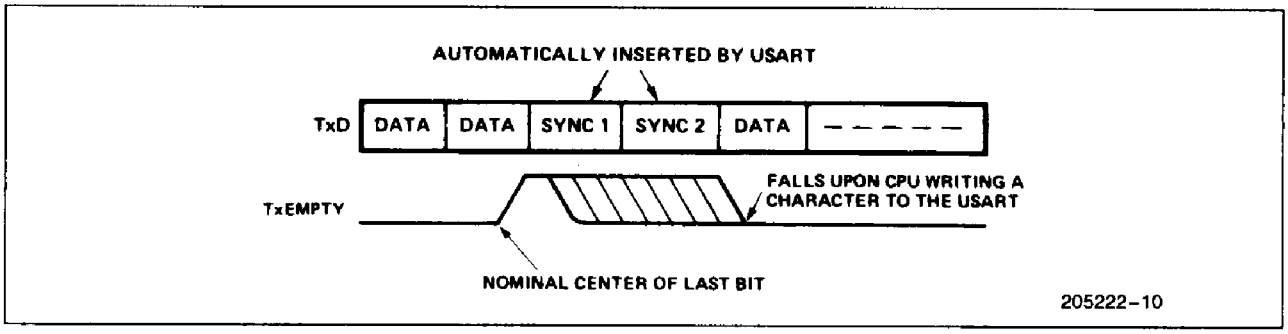
The TxD output is continuously high until the CPU sends its first character to the 8251A which usually is a SYNC character. When the $\overline{CTS}$ line goes low, the first character is serially transmitted out. All characters are shifted out on the falling edge of $\overline{TxC}$. Data is shifted out at the same rate as the $\overline{TxC}$.

Once transmission has started, the data stream at the TxD output must continue at the $\overline{TxC}$ rate. If the CPU does not provide the 8251A with a data character before the 8251A Transmitter Buffers become empty, the SYNC characters (or character if in single SYNC character mode) will be automatically inserted in the TxD data stream. In this case, the TxEMPTY pin is raised high to signal that the 8251A is empty and SYNC characters are being sent out. TxEMPTY does not go low when the SYNC is being shifted out (see figure below). The TxEMPTY pin is internally reset by a data character being written into the 8251A.



*NOTE:
If character length is defined as 5, 6, or 7 bits the unused bits are set to "zero".

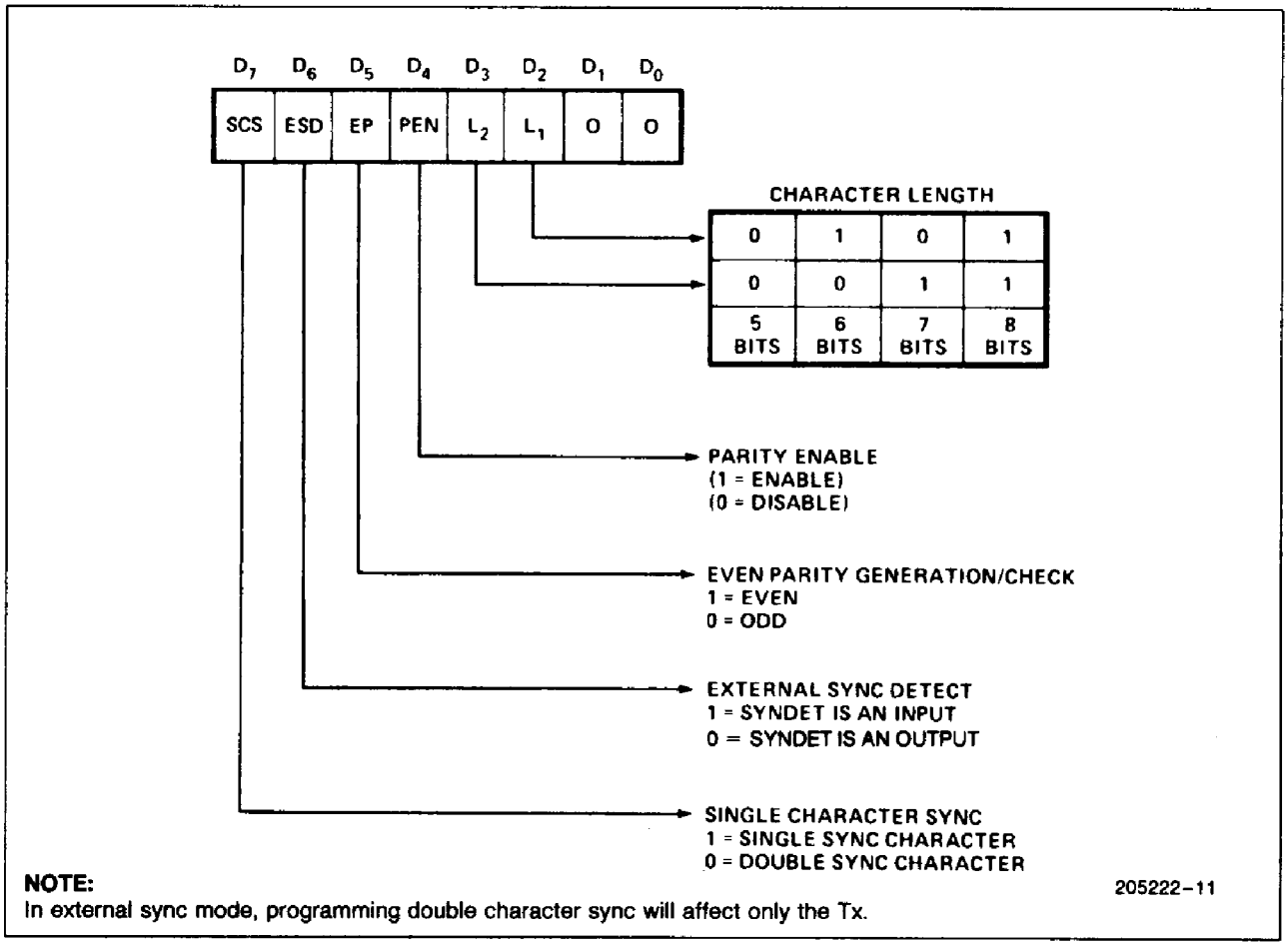
Figure 9. Asynchronous Mode



Synchronous Mode (Receive)

In this mode, character synchronization can be internally or externally achieved. If the SYNC mode has been programmed, ENTER HUNT command should be included in the first command instruction word written. Data on the RxD pin is then sampled on the rising edge of $\overline{RxC}$. The content of the Rx buffer is compared at every bit boundary with the first SYNC character until a match occurs. If the 8251A has been programmed for two SYNC characters, the subsequent received character is also compared; when both SYNC characters have been detected, the USART ends the HUNT mode and is in character synchronization. The SYNDET pin is then set high, and is reset automatically by a STATUS READ. If parity is programmed, SYNDET will not be set until the middle of the parity bit instead of the middle of the last data bit.

In the external SYNC mode, synchronization is achieved by applying a high level on the SYNDET pin, thus forcing the 8251A out of the HUNT mode. The high level can be removed after one $\overline{RxC}$ cycle. An ENTER HUNT command has no effect in the asynchronous mode of operation.



NOTE:
In external sync mode, programming double character sync will affect only the Tx.

Figure 10. Mode Instruction Format, Synchronous Mode

Parity error and overrun error are both checked in the same way as in the Asynchronous Rx mode. Parity is checked when not in Hunt, regardless of whether the Receiver is enabled or not.

The CPU can command the receiver to enter the HUNT mode if synchronization is lost. This will also set all the used character bits in the buffer to a "one," thus preventing a possible false SYNDET caused by data that happens to be in the Rx Buffer at ENTER HUNT time. Note that the SYNDET F/F is reset at each Status Read, regardless of whether internal or external SYNC has been programmed. This does not cause the 8251A to return to the HUNT mode. When in SYNC mode, but not in HUNT, Sync Detection is still functional, but only occurs at the "known" word boundaries. Thus, if one Status Read indicates SYNDET and a second Status Read also indicates SYNDET, then the programmed SYNDET characters have been received since the previous Status Read. (If double character sync has been programmed, then both sync characters have been contiguously received to gate a SYNDET indication). When external SYNDET mode is selected, internal Sync Detect is disabled, and the SYNDET F/F may be set at any bit boundary.

COMMAND INSTRUCTION DEFINITION

Once the functional definition of the 8251A has been programmed by the Mode Instruction and the Sync characters are loaded (if in Sync Mode) then the device is ready to be used for data communication. The Command Instruction controls the actual operation of the selected format. Functions such as: Enable Transmit/Receive, Error Reset and Modem Controls are provided by the Command instruction.

Once the Mode Instruction has been written into the 8251A and Sync characters inserted, of necessary, then all further "control writes" ($C/\overline{D} = 1$) will load a Command Instruction. A Reset Operation (internal or external) will return the 8251A to the Mode Instruction format.

NOTE:
Internal Reset on Power-up:

When power is first applied, the 8251A may come up in the Mode, Sync character or Command format. To guarantee that the device is in the Command Instruction format before the Reset command is issued, it is safest to execute the worst-case initialization sequence (sync mode with two sync characters). Loading three 00Hs consecutively into the device with $C/\overline{D} = 1$ configures sync operation and writes two dummy 00H sync characters. An Internal Reset command (40H) may then be issued to return the device to the "idle" state.

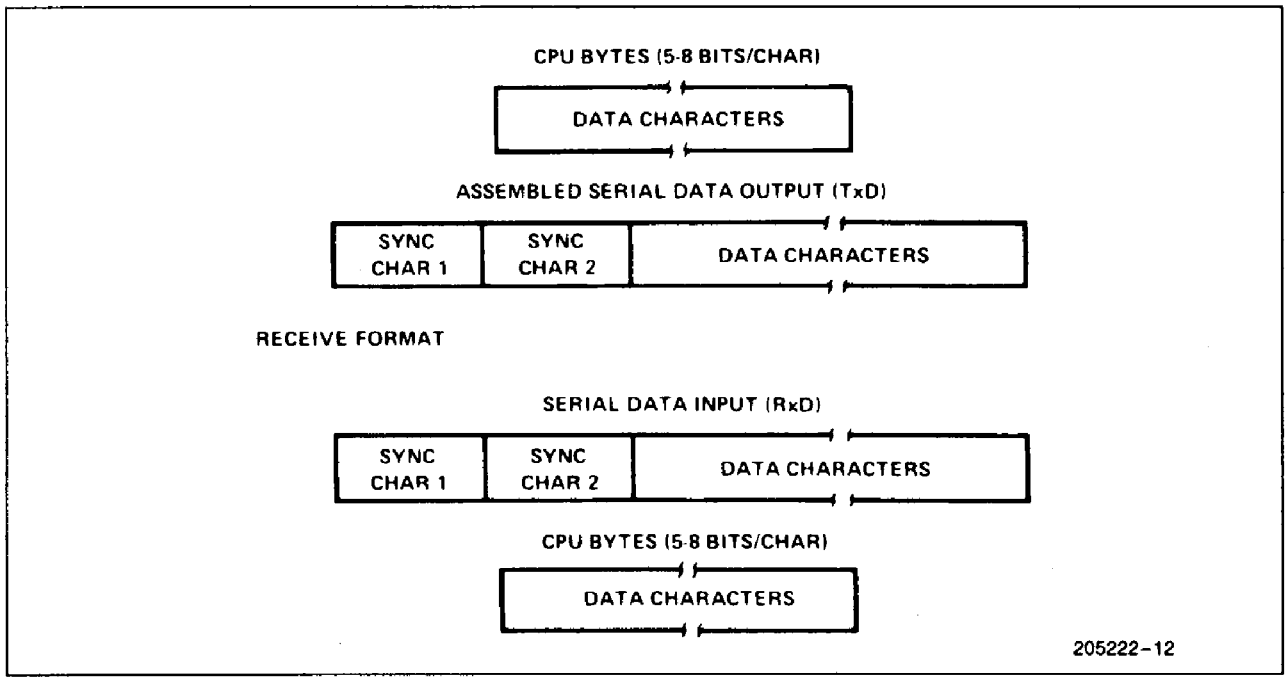
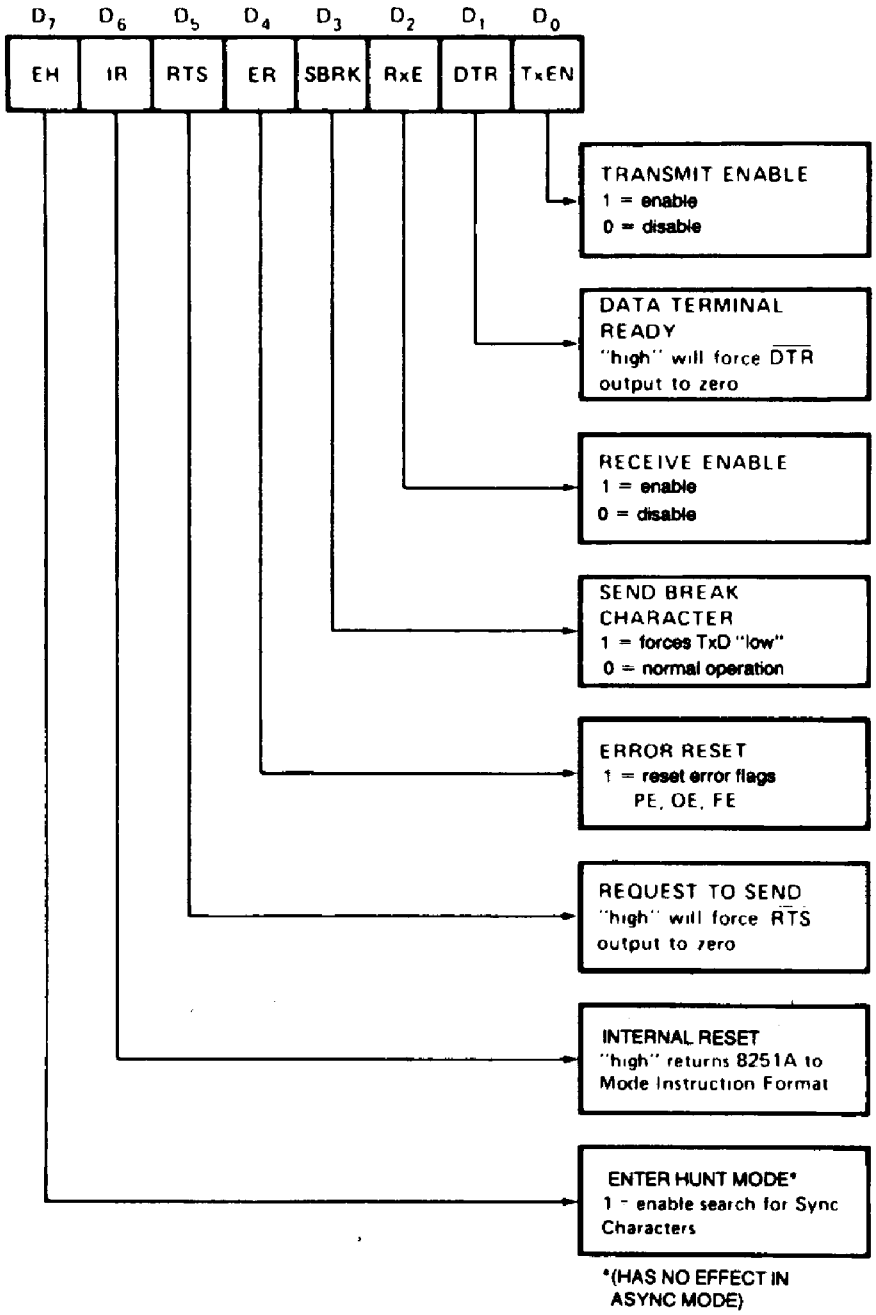


Figure 11. Data Format, Synchronous Mode



NOTE:
Error Reset must be performed whenever RxEnable and Enter Hunt are programmed.

205222-13

Figure 12. Command Instruction Format

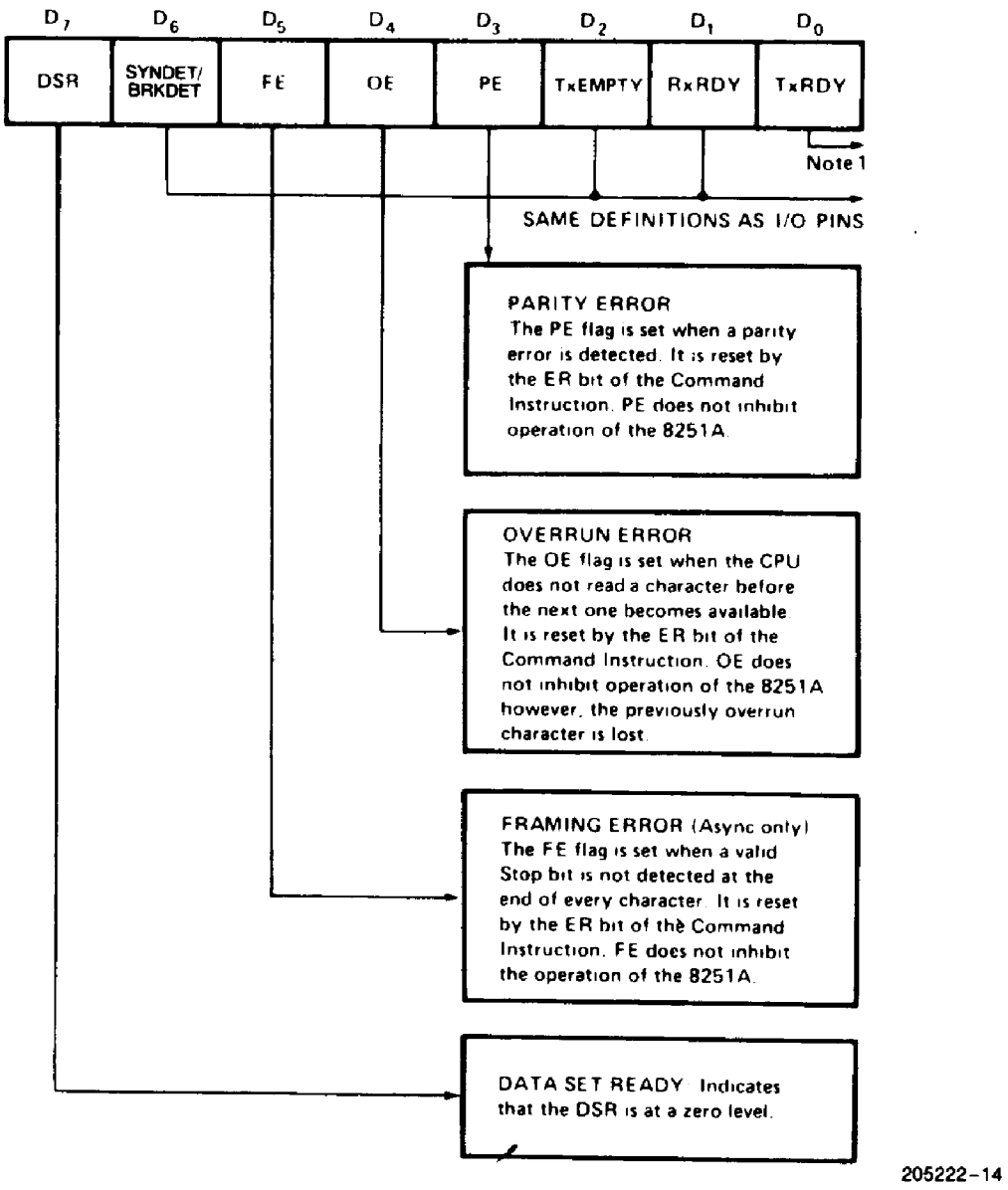
STATUS READ DEFINITION

In data communication systems it is often necessary to examine the "status" of the active device to ascertain if errors have occurred or other conditions that require the processor's attention. The 8251A has facilities that allow the programmer to "read" the status of the device at any time during the functional operation. (Status update is inhibited during status read.)

A normal "read" command is issued by the CPU with C/$\overline{D}$ = 1 to accomplish this function.

Some of the bits in the Status Read Format have identical meanings to external output pins so that the 8251A can be used in a completely polled or interrupt-driven environment. TxRDY is an exception.

Note that status update can have a maximum delay of 28 clock periods from the actual event affecting the status.



NOTE:
1. TxRDY status bit has different meanings from the TxRDY output pin. The former is not conditioned by $\overline{CTS}$ and TxEN; the latter is conditioned by both $\overline{CTS}$ and TxEN.
i.e. TxRDY status bit = DB Buffer Empty
TxRDY pin out = DB Buffer Empty • ($\overline{CTS}$ = 0) • (TxEN = 1)

Figure 13. Status Read Format

APPLICATIONS OF THE 8251A

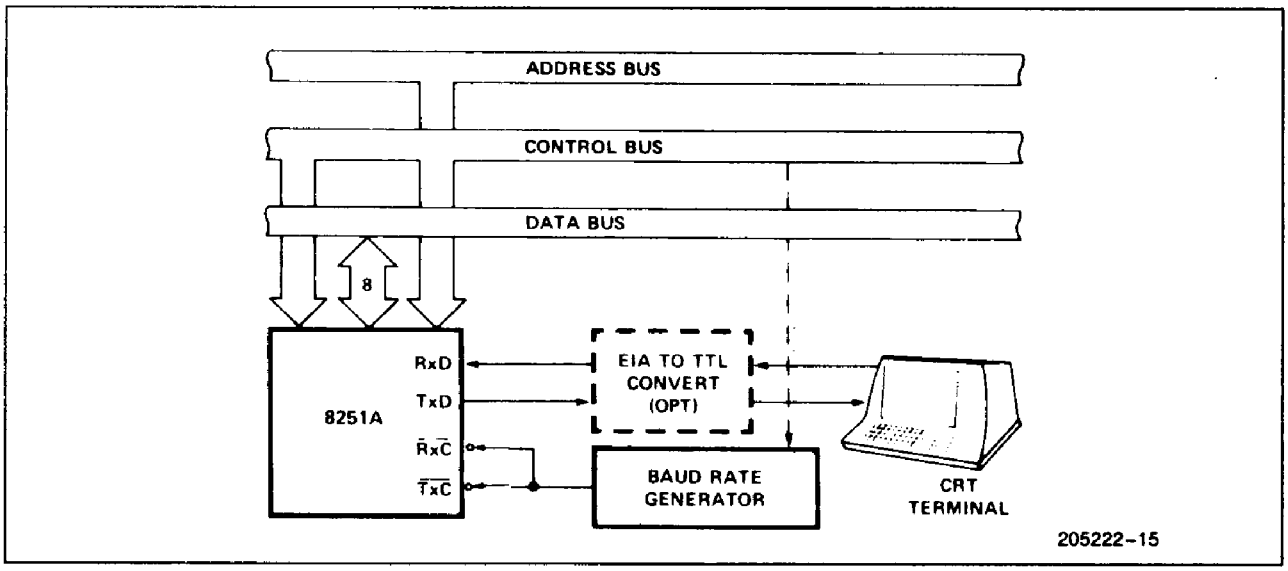


Figure 14. Asynchronous Serial Interface to CRT Terminal, DC—9600 Baud

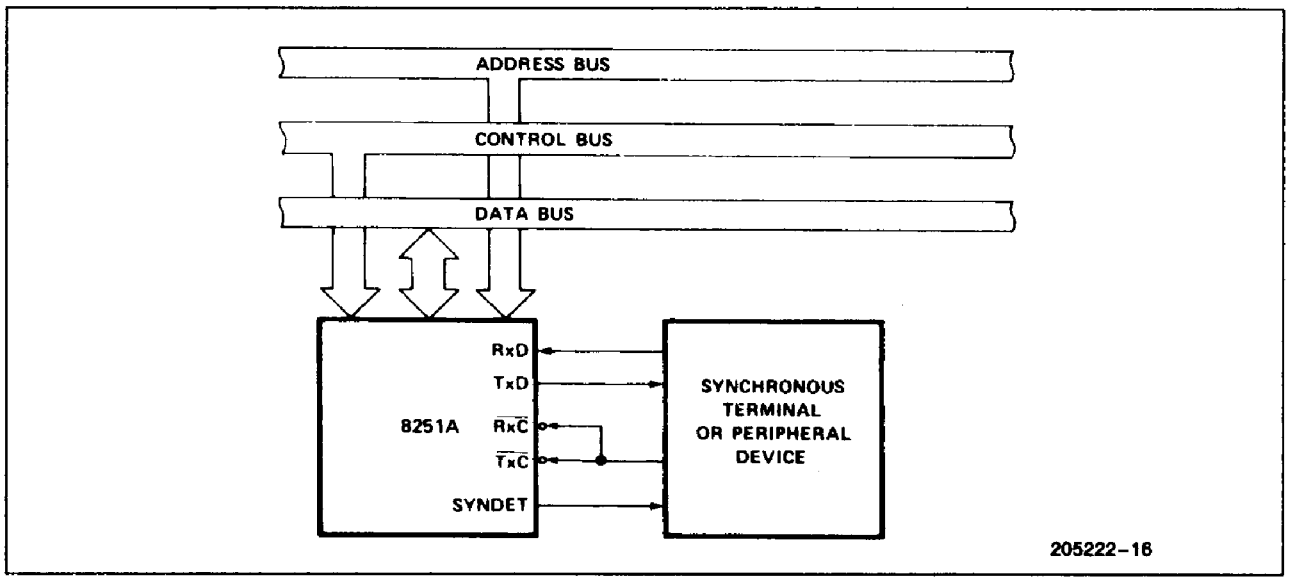


Figure 15. Synchronous Interface to Terminal or Peripheral Device

APPLICATIONS OF THE 8251A (Continued)

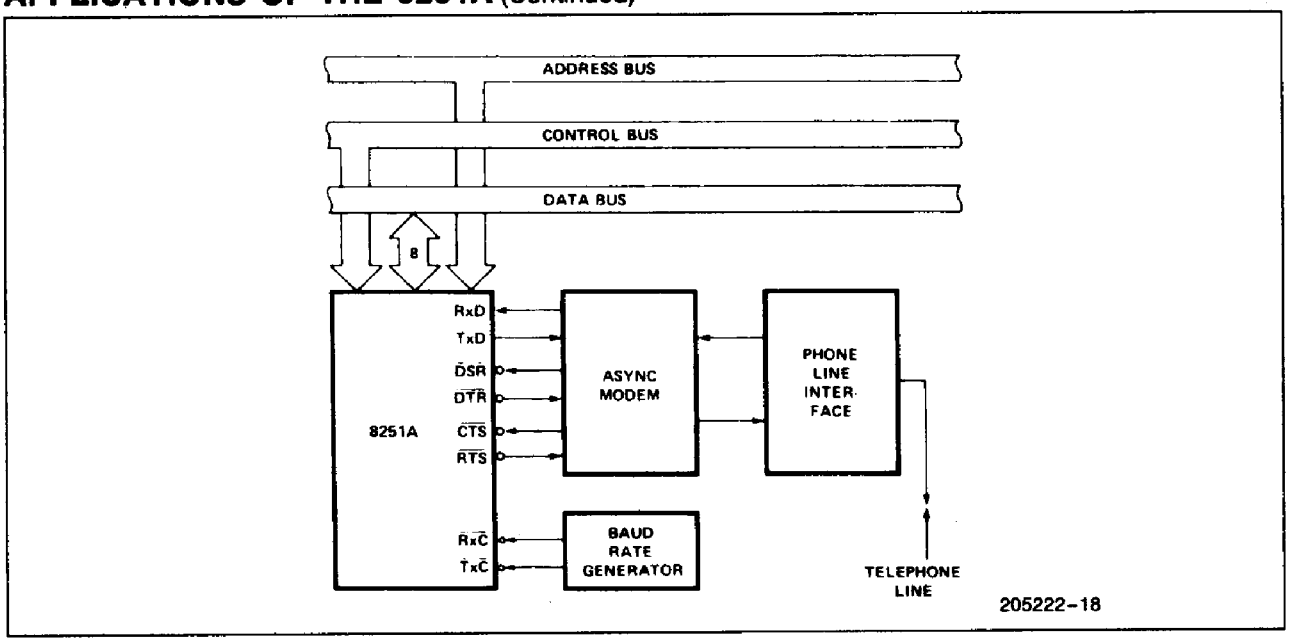


Figure 16. Asynchronous Interface to Telephone Lines

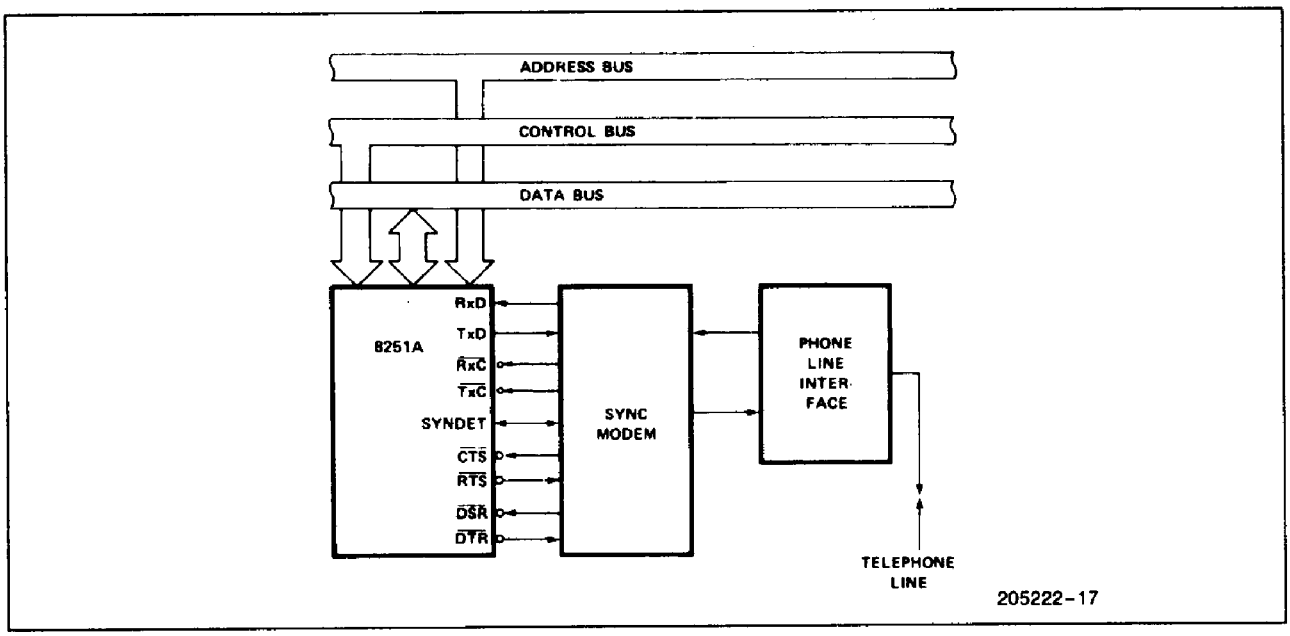


Figure 17. Synchronous Interface to Telephone Lines

NOTES:
1. AC timings measured V_{OH} = 2.0 V_{OL} = 0.8, and with load circuit of Figure 18.
2. Chip Select (CS) and Command/Data (C/D) are considered as Addresses.
3. Assumes that Address is valid before $R_D \downarrow$.
4. This recovery time is for Mode Initialization only. Write Data is allowed only when TxRDY = 1. Recovery Time between Writes for Asynchronous Mode is 8 t_{CY} and for Synchronous Mode is 16 t_{CY}.
5. The TxC and RxC frequencies have the following limitations with respect to CLK: For 1x Baud Rate, f_{Tx} or $f_{Rx} \leq 1/(30\ t_{CY})$: For 16x and 64x Baud Rate, f_{Tx} or $f_{Rx} \leq 1/(4.5\ t_{CY})$. This applies to Baud Rates less than or equal to 64K Baud.
6. Reset Pulse Width = 6 t_{CY} minimum; System clock must be running during Reset.
7. Status update can have a maximum delay of 28 clock periods from the event affecting the status.
8. In external sync mode the tes spec. requires the ratio of the system clock (clock) to receive or transmit bit ratios to be greater than 34.
9. A float is defined as the point where the data bus falls below a logic 1 (2.0V @ I_{OH} limit) or rises above a Logic 0 (0.8V @ I_{OL} limit).

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias .....0°C to 70°C
Storage Temperature .....−65°C to +150°C
Voltage on Any Pin
with Respect to Ground.....−0.5V to +7V
Power Dissipation 1W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5.0V ±10%, GND = 0V*

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −400 μA
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to 0.45V
I_{IL}	Input Leakage		±10	μA	V_{IN} = V_{CC} to 0.45V
I_{CC}	Power Supply Current		100	ma	All Outputs = High

CAPACITANCE T_A = 25°C, V_{CC} = GND = 0V

Symbol	Parameter	Min	Max	Unit	Test Conditions
C_{IN}	Input Capacitance		10	pF	fc = 1 MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins returned to GND

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5.0V ±10%, GND = 0V*

Bus Parameters (Note 1)

READ CYCLE

Symbol	Parameter	Min	Max	Unit	Test Conditions
t_{AR}	Address Stable Before $\overline{READ}$ ($\overline{CS}$, C/$\overline{D}$)	0		ns	(Note 2)
t_{RA}	Address Hold Time for $\overline{READ}$ ($\overline{CS}$, C/$\overline{D}$)	0		ns	(Note 2)
t_{RR}	$\overline{READ}$ Pulse Width	250		ns	
t_{RD}	Data Delay from $\overline{READ}$		200	ns	3, C_L = 150 pF
t_{DF}	$\overline{READ}$ to Data Floating	10	100	ns	(Note 1, 9)

WRITE CYCLE

Symbol	Parameter	Min	Max	Unit	Test Conditions
t_{AW}	Address Stable Before $\overline{WRITE}$	0		ns	
t_{WA}	Address Hold Time for $\overline{WRITE}$	0		ns	
t_{WW}	$\overline{WRITE}$ Pulse Width	250		ns	
t_{DW}	Data Set-Up Time for $\overline{WRITE}$	150		ns	
t_{WD}	Data Hold Time for $\overline{WRITE}$	20		ns	
t_{RV}	Recovery Time Between WRITES	6		t_{CY}	(Note 4)

A.C. CHARACTERISTICS (Continued)

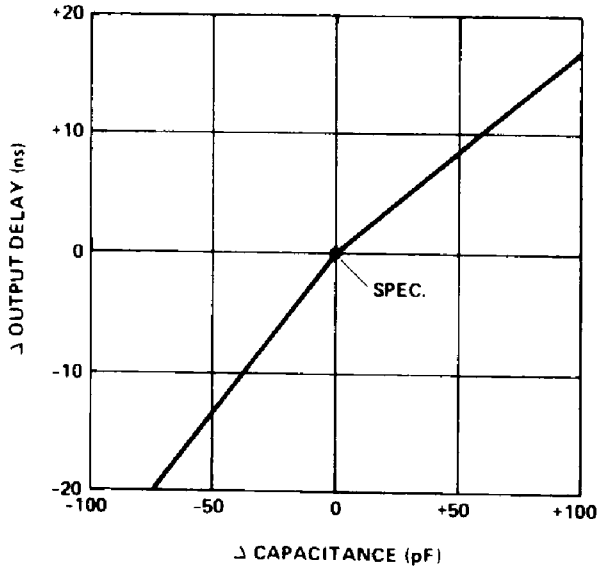
OTHER TIMINGS

Symbol	Parameter	Min	Max	Unit	Test Conditions
t_{CY}	Clock Period	320	1350	ns	(Note 5, 6)
t_{ϕ}	Clock High Pulse Width	120	$t_{CY}-90$	ns	
$\overline{t_{\phi}}$	Clock Low Pulse Width	90		ns	
t_R, t_F	Clock Rise and Fall Time		20	ns	
t_{DTx}	TxD Delay from Falling Edge of $\overline{TxC}$		1	μs	
f_{Tx}	Transmitter Input Clock Frequency 1x Baud Rate 16x Baud Rate 64x Baud Rate	 DC DC DC	 64 310 615	 kHz kHz kHz	
t_{TPW}	Transmitter Input Clock Pulse Width 1x Baud Rate 16x and 64x Baud Rate	 12 1		 t_{CY} t_{CY}	
t_{TPD}	Transmitter Input Clock Pulse Delay 1x Baud Rate 16x and 64x Baud Rate	 15 3		 t_{CY} t_{CY}	
f_{Rx}	Receiver Input Clock Frequency 1x Baud Rate 16x Baud Rate 64x Baud Rate	 DC DC DC	 64 310 615	 kHz kHz kHz	
t_{RPW}	Receiver Input Clock Pulse Width 1x Baud Rate 16x and 64x Baud Rate	 12 1		 t_{CY} t_{CY}	
t_{RPD}	Receiver Input Clock Pulse Delay 1x Baud Rate 16x and 64x Baud Rate	 15 3		 t_{CY} t_{CY}	
t_{TxRDY}	TxRDY Pin Delay from Center of Last Bit		14	t_{CY}	(Note 7)
$t_{TxRDY\ CLEAR}$	TxRDY ↓ from Leading Edge of $\overline{WR}$		400	ns	(Note 7)
t_{RxRDY}	RxRDY Pin Delay from Center of Last Bit		26	t_{CY}	(Note 7)
$t_{RxRDY\ CLEAR}$	RxRDY ↓ from Leading Edge of $\overline{RD}$		400	ns	(Note 7)
t_{IS}	Internal SYNDET Delay from Rising Edge of $\overline{RxC}$		26	t_{CY}	(Note 7)
t_{ES}	External SYNDET Set-Up Time After Rising Edge of $\overline{RxC}$	16 t_{CY}	$t_{RPD}-tCY$	ns	(Note 7)
$t_{TxEMPTY}$	TxEMPTY Delay from Center of Last Bit		20	t_{CY}	(Note 7)
t_{WC}	Control Delay from Rising Edge of WRITE (TxEn, $\overline{DTR}$, $\overline{RTS}$)		8	t_{CY}	(Note 7)
t_{CR}	Control to READ Set-Up Time ($\overline{DSR}$, $\overline{CTS}$)	20		t_{CY}	(Note 7)

***NOTE:**
For Extended Temperature EXPRESS, use MIL 8251A electrical parameters.

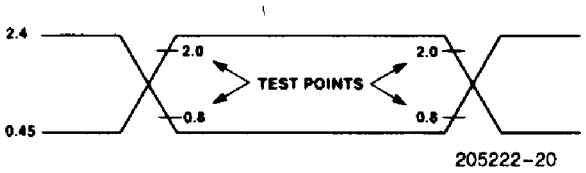
A.C. CHARACTERISTICS (Continued)

TYPICAL Δ OUTPUT DELAY VS. Δ CAPACITANCE (pF)



205222-19

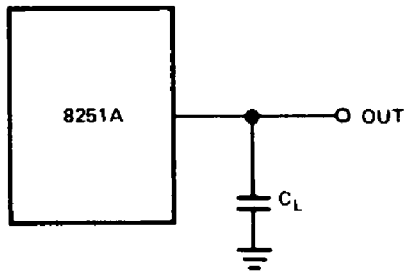
A.C. TESTING INPUT, OUTPUT WAVEFORM



205222-20

AC Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

A.C. TESTING LOAD CIRCUIT



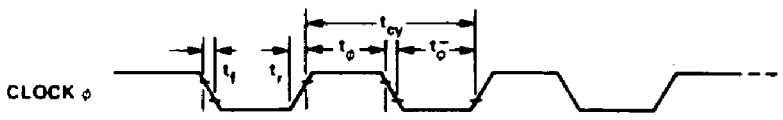
C_L = 150 pF

205222-21

Figure 18

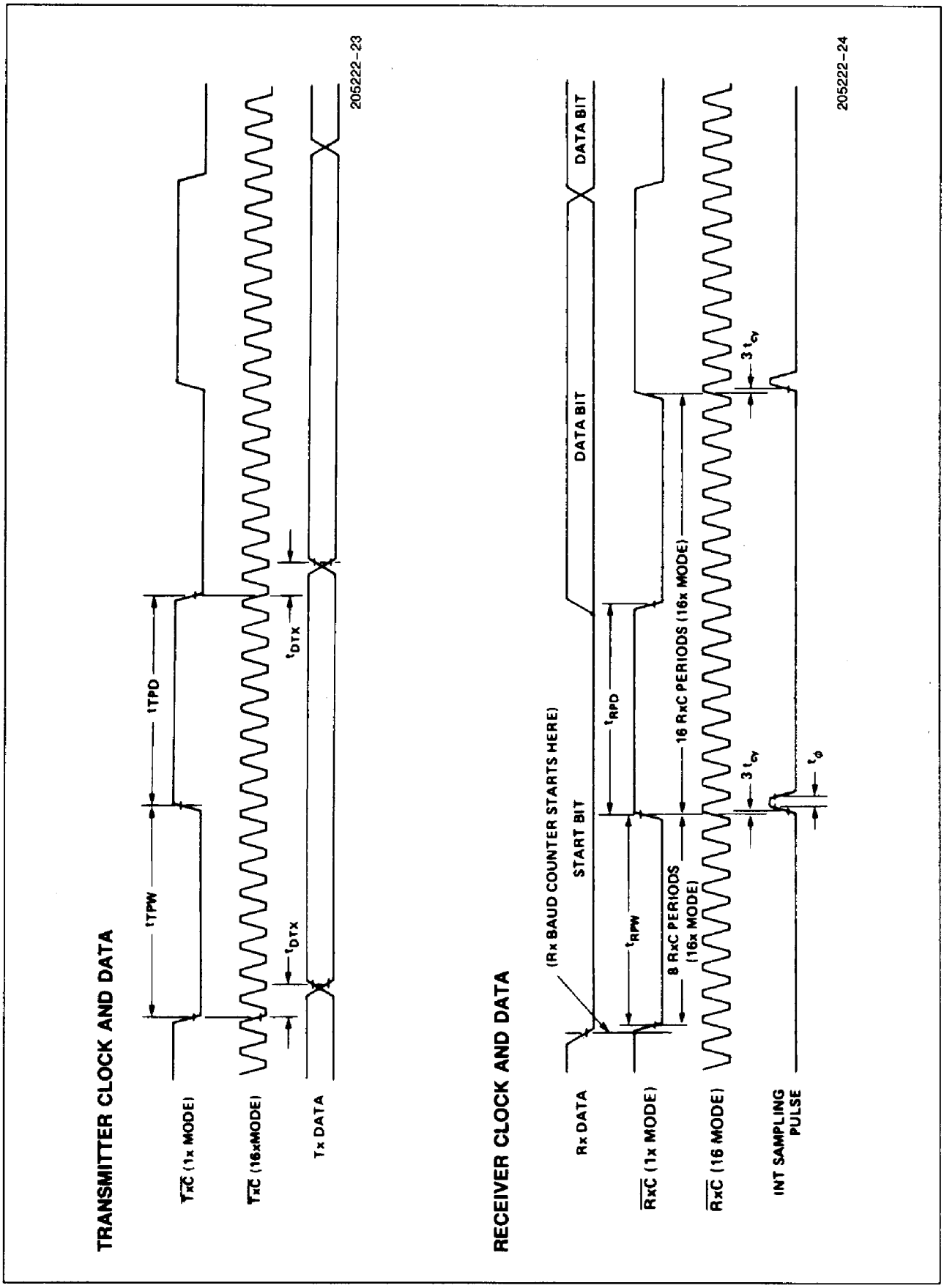
WAVEFORMS

SYSTEM CLOCK INPUT



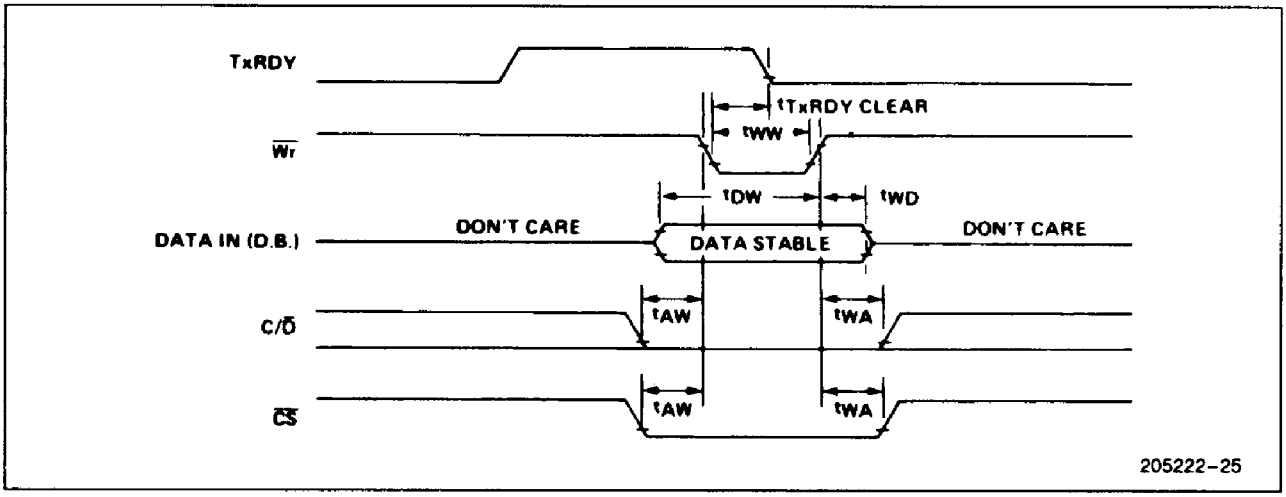
205222-22

WAVEFORMS (Continued)

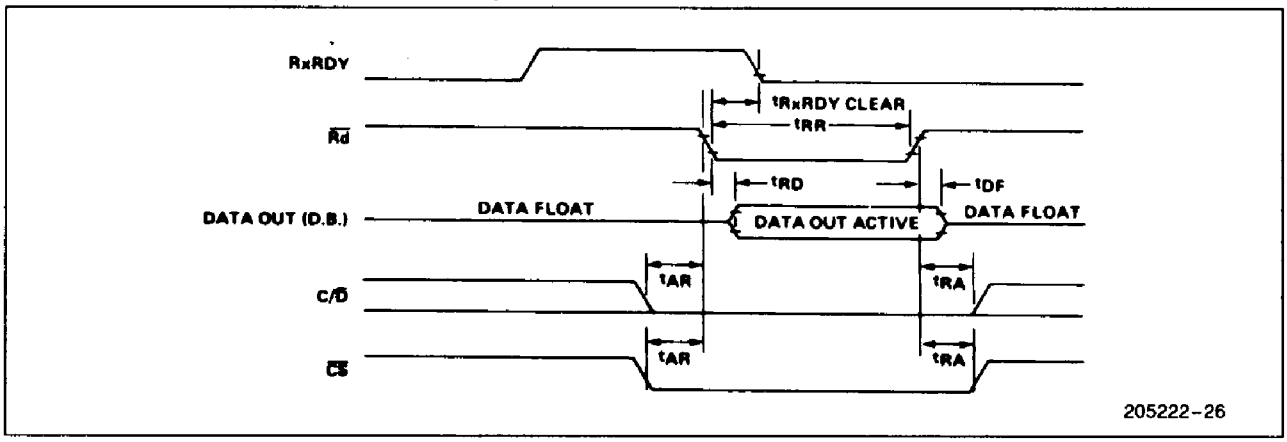


WAVEFORMS (Continued)

WRITE DATA CYCLE (CPU → USART)



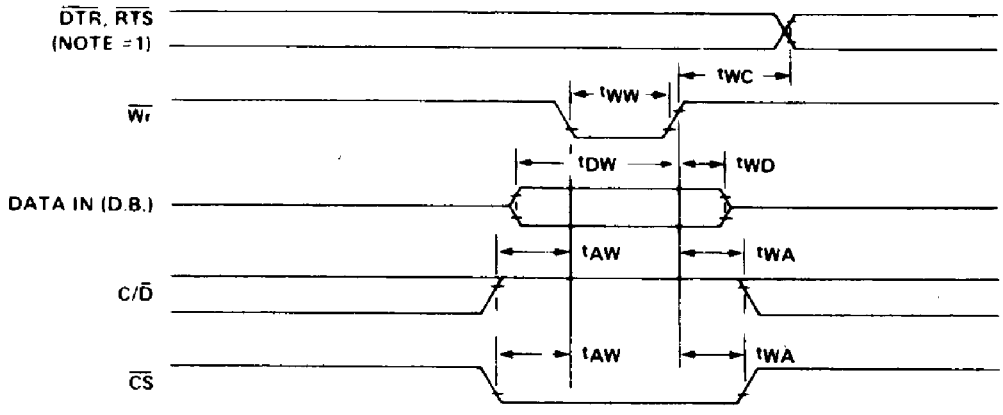
READ DATA CYCLE (CPU ← USART)



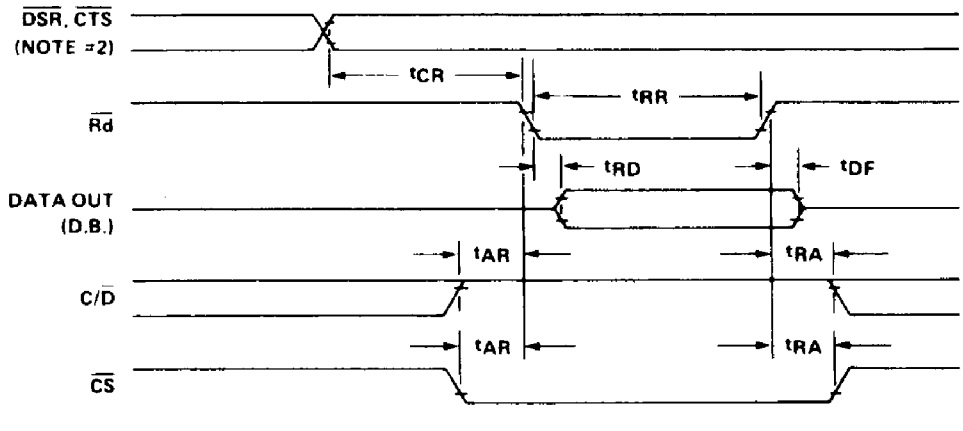
2

WAVEFORMS (Continued)

WRITE CONTROL OR OUTPUT PORT CYCLE (CPU → USART)

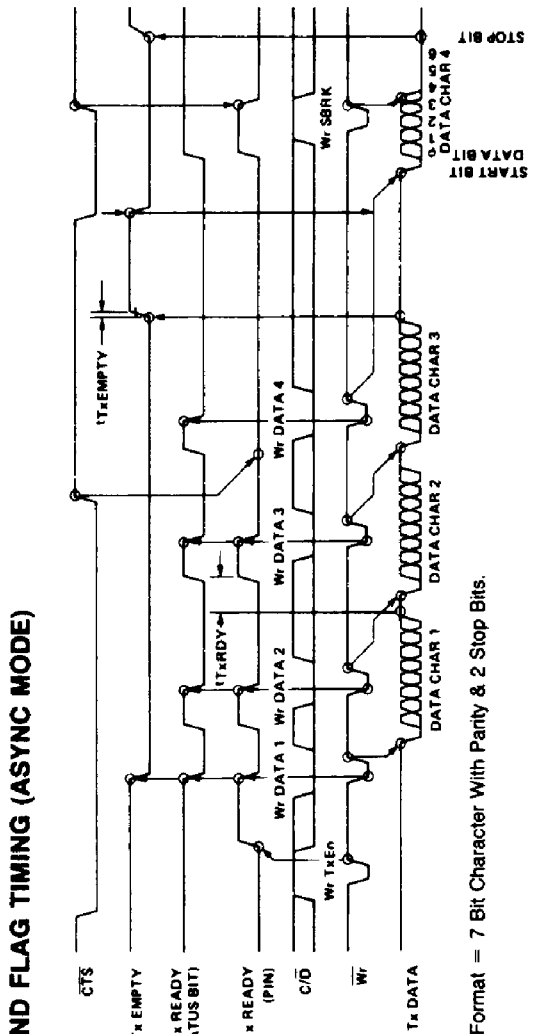


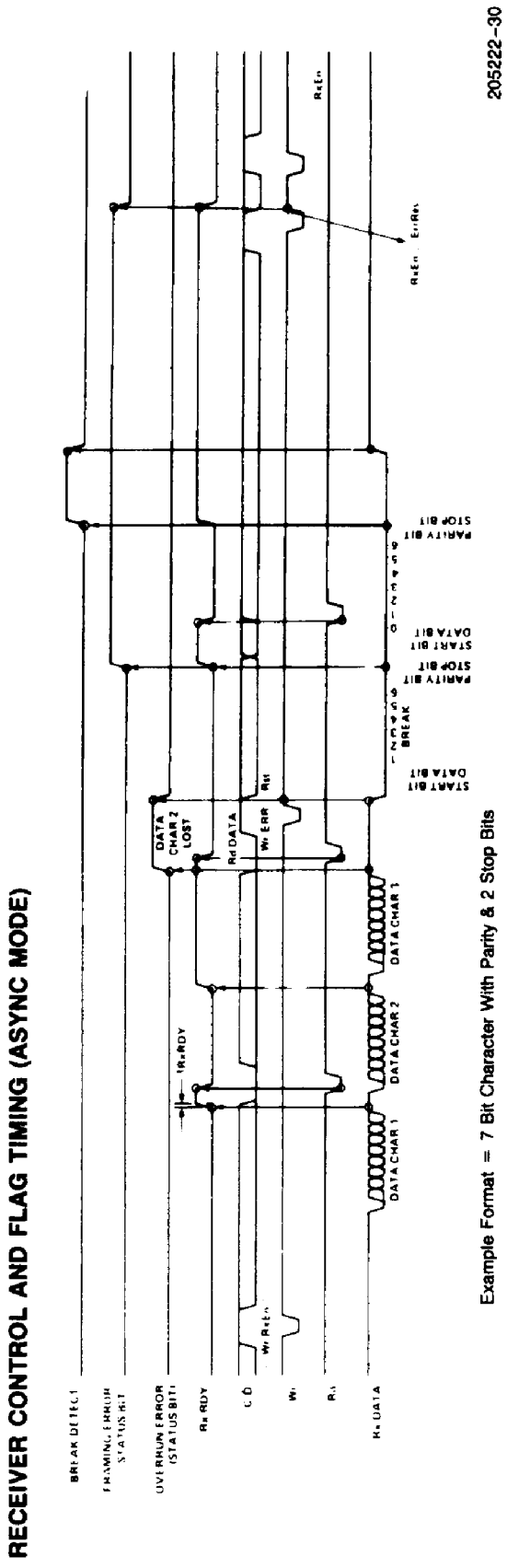
READ CONTROL OR INPUT PORT (CPU ← USART)



NOTES:
1. T_{WC} includes the response timing of a control byte.
2. T_{CR} includes the effect of CTS on the TxENBL circuitry.

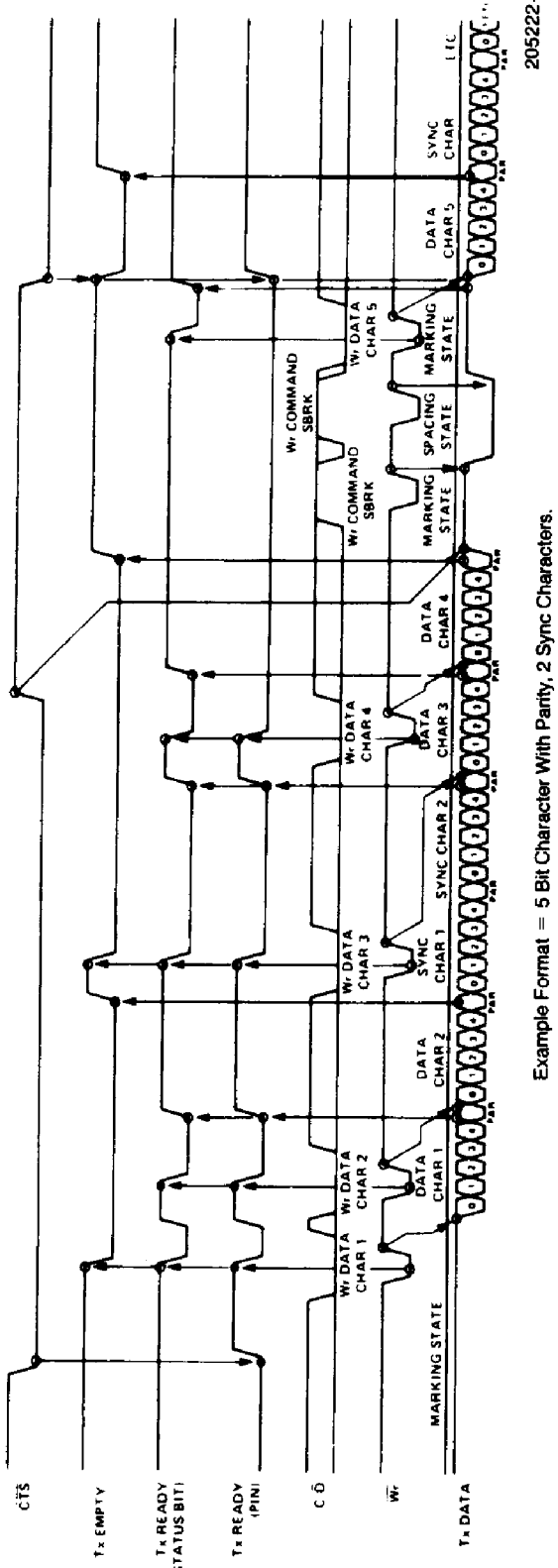
WAVEFORMS (Continued)





WAVEFORMS (Continued)

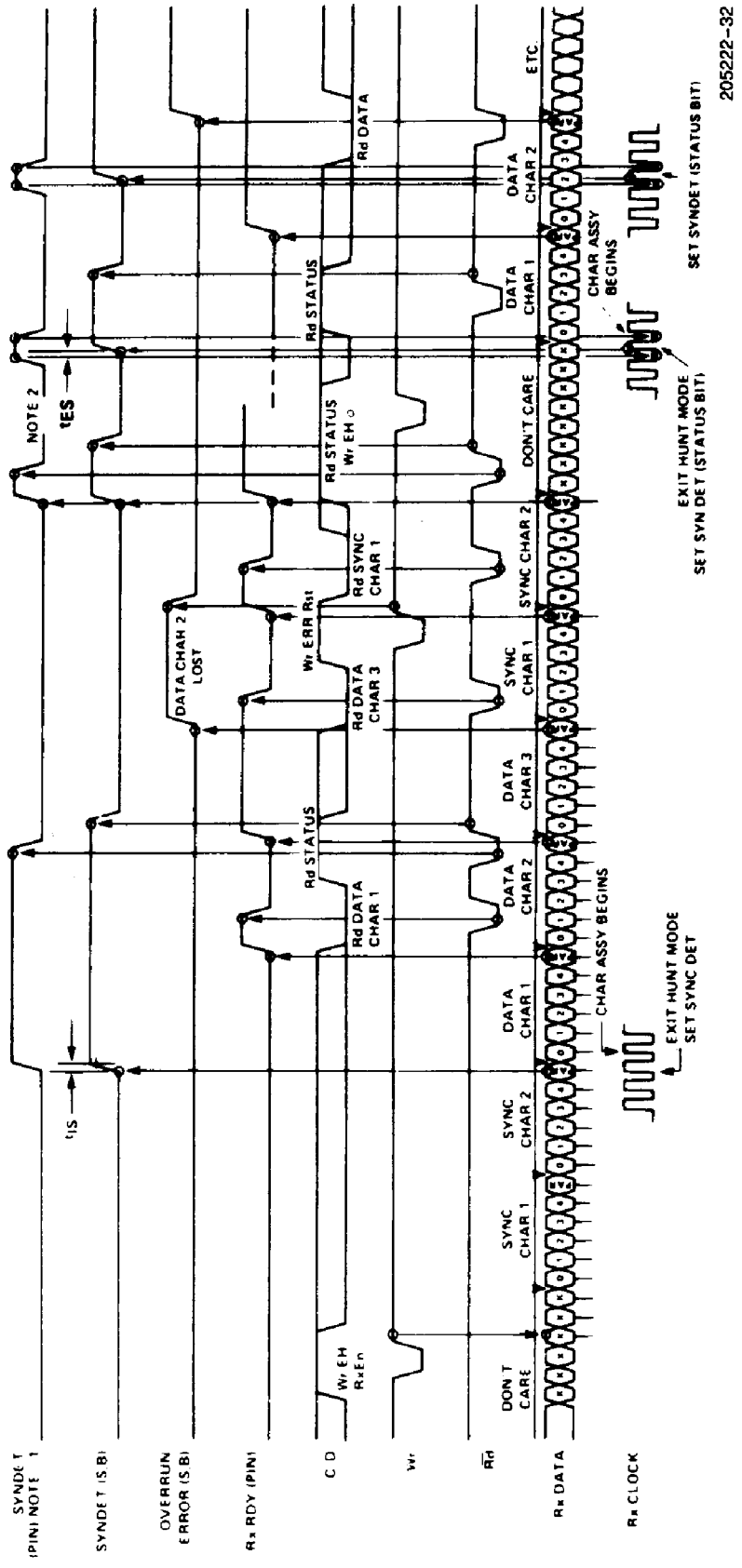
TRANSMITTER CONTROL AND FLAG TIMING (SYNC MODE)



Example Format = 5 Bit Character With Parity, 2 Sync Characters.

WAVEFORMS (Continued)

RECEIVER CONTROL AND FLAG TIMING (SYNC MODE)



NOTES:
1. Internal Sync, 2 Sync Characters, 5 Bits With Parity.
2. External Sync, 5 Bits, With Parity.

Z8430
Z80® CTC Counter/ Timer Circuit

Zilog

Product Specification

March 1981

Features

- Four independently programmable counter/timer channels, each with a readable downcounter and a selectable 16 or 256 prescaler. Downcounters are reloaded automatically at zero count.
- Three channels have Zero Count/Timeout outputs capable of driving Darlington transistors.
- Selectable positive or negative trigger initiates timer operation.
- Standard Z-80 Family daisy-chain interrupt structure provides fully vectored, prioritized interrupts without external logic. The CTC may also be used as an interrupt controller.
- Interfaces directly to the Z-80 CPU or—for baud rate generation—to the Z-80 SIO.

General Description

The Z-80 CTC four-channel counter/timer can be programmed by system software for a broad range of counting and timing applications. The four independently programmable channels of the Z-80 CTC satisfy common microcomputer system requirements for event counting, interrupt and interval timing, and general clock rate generation.

System design is simplified because the CTC connects directly to both the Z-80 CPU and the Z-80 SIO with no additional logic. In larger systems, address decoders and buffers may be required.

Programming the CTC is straightforward: each channel is programmed with two bytes; a third is necessary when interrupts are enabled. Once started, the CTC counts down, reloads its time constant automatically, and resumes counting. Software timing loops are completely eliminated. Interrupt processing is simplified because only one vector need be specified; the CTC internally generates a unique vector for each channel.

The Z-80 CTC requires a single +5 V power supply and the standard Z-80 single-phase system clock. It is fabricated with n-channel silicon-gate depletion-load technology, and packaged in a 28-pin plastic or ceramic DIP.

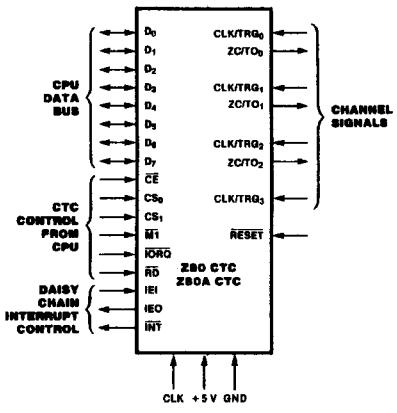


Figure 1. Pin Functions

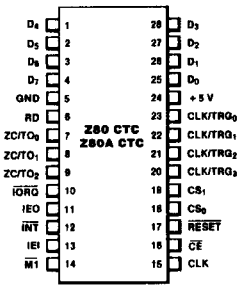


Figure 2. Pin Assignments

Functional Description

The Z-80 CTC has four independent counter/timer channels. Each channel is individually programmed with two words: a control word and a time-constant word. The control word selects the operating mode (counter or timer), enables or disables the channel interrupt, and selects certain other operating parameters. If the timing mode is selected, the control word also sets a prescaler, which divides the system clock by either 16 or 256. The time-constant word is a value from 1 to 256.

During operation, the individual counter channel counts down from the preset time constant value. In counter mode operation the counter decrements on each of the CLK/TRG input pulses until zero count is reached. Each decrement is synchronized by the system clock. For counts greater than 256, more than one counter can be cascaded. At zero count, the down-counter is automatically reset with the time constant value.

The timer mode determines time intervals as small as 4 μs (Z-80A) or 6.4 μs (Z-80) without additional logic or software timing loops. Time intervals are generated by dividing the system clock with a prescaler that decrements a preset down-counter.

Thus, the time interval is an integral multiple of the clock period, the prescaler value (16 or 256) and the time constant that is preset in the down-counter. A timer is triggered automatically when its time constant value is programmed, or by an external CLK/TRG input.

Three channels have two outputs that occur at zero count. The first output is a zero-count/timeout pulse at the ZC/TO output. The fourth channel (Channel 3) does not have a ZC/TO output; interrupt request is the only output available from Channel 3.

The second output is Interrupt Request ($\overline{INT}$), which occurs if the channel has its interrupt enabled during programming. When the Z-80 CPU acknowledges Interrupt Request, the Z-80 CTC places an interrupt vector on the data bus.

The four channels of the Z-80 CTC are fully prioritized and fit into four contiguous slots in a standard Z-80 daisy-chain interrupt structure. Channel 0 is the highest priority and Channel 3 the lowest. Interrupts can be individually enabled (or disabled) for each of the four channels.

Architecture

The CTC has four major elements, as shown in Figure 3.

- CPU bus I/O
- Channel control logic
- Interrupt logic
- Counter/timer circuits

CPU Bus I/O. The CPU bus I/O circuit decodes the address inputs, and interfaces the CPU data and control signals to the CTC for distribution on the internal bus.

Internal Control Logic. The CTC internal control logic controls overall chip operating functions such as the chip enable, reset, and read/write logic.

Interrupt Logic. The interrupt control logic ensures that the CTC interrupts interface properly with the Z-80 CPU interrupt system. The logic controls the interrupt priority of the CTC as a function of the IEI signal. If IEI is High, the CTC has priority. During interrupt

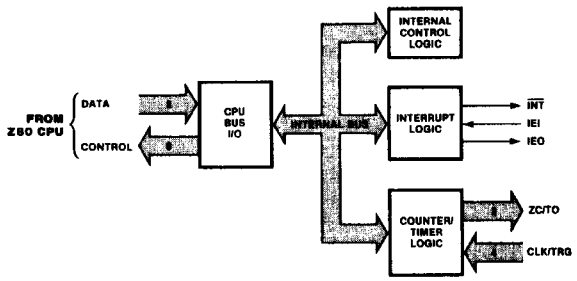


Figure 3. Functional Block Diagram

Architecture (Continued)

processing, the interrupt logic holds IEO Low, which inhibits the interrupt operation on lower priority devices. If the IEI input goes Low, priority is relinquished and the interrupt logic drives IEO Low.

If a channel is programmed to request an interrupt, the interrupt logic drives IEO Low at the zero count, and generates an $\overline{INT}$ signal to the Z-80 CPU. When the Z-80 CPU responds with interrupt acknowledge ($\overline{M1}$ and $\overline{IORQ}$), then the interrupt logic arbitrates the CTC internal priorities, and the interrupt control logic places a unique interrupt vector on the data bus.

If an interrupt is pending, the interrupt logic holds IEO Low. When the Z-80 CPU issues a Return From Interrupt (RETI) instruction, each peripheral device decodes the first byte (ED_{16}). If the device has a pending interrupt, it raises IEO (High) for one $\overline{M1}$ cycle. This ensures that all lower priority devices can decode the entire RETI instruction and reset properly.

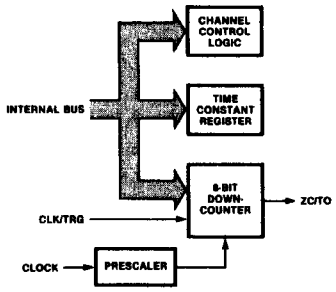


Figure 4. Counter/Timer Block Diagram

Counter/Timer Circuits. The CTC has four independent counter/timer circuits, each containing the logic shown in Figure 4.

Channel Control Logic. The channel control logic receives the 8-bit channel control word when the counter/timer channel is programmed. The channel control logic decodes the control word and sets the following operating conditions:

- Interrupt enable (or disable)
- Operating mode (timer or counter)
- Timer mode prescaler factor (16 or 256)
- Active slope for CLK/TRG input
- Timer mode trigger (automatic or CLK/TRG input)
- Time constant data word to follow
- Software reset

Time Constant Register. When the counter/timer channel is programmed, the time constant register receives and stores an 8-bit time constant value, which can be anywhere from 1 to 256 (0 = 256). This constant is automatically loaded into the down-counter when the counter/timer channel is initialized, and subsequently after each zero count.

Prescaler. The prescaler, which is used only in timer mode, divides the system clock frequency by a factor of either 16 or 256. The prescaler output clocks the down-counter during timer operation. The effect of the prescaler on the down-counter is a multiplication of the system clock period by 16 or 256. The prescaler factor is programmed by bit 5 of the channel control word.

Down-Counter. Prior to each count cycle, the down-counter is loaded with the time constant register contents. The counter is then decremented one of two ways, depending on operating mode:

- By the prescaler output (timer mode)
- By the trigger pulses into the CLK/TRG input (counter mode)

Without disturbing the down-count, the Z-80 CPU can read the count remaining at any time by performing an I/O read operation at the port address assigned to the CTC channel. When the down-counter reaches the zero count, the ZC/TO output generates a positive-going pulse. When the interrupt is enabled, zero count also triggers an interrupt request signal ($\overline{INT}$) from the interrupt logic.

Programming

Each Z-80 CTC channel must be programmed prior to operation. Programming consists of writing two words to the I/O port that corresponds to the desired channel. The first word is a control word that selects the operating mode and other parameters; the second word is a time constant, which is a binary data word with a value from 1 to 256. A time constant word must be preceded by a channel control word.

After initialization, channels may be reprogrammed at any time. If updated control and time constant words are written to a channel during the count operation, the count continues to zero before the new time constant is loaded into the counter.

If the interrupt on any Z-80 CTC channel is enabled, the programming procedure should also include an interrupt vector. Only one vector is required for all four channels, because the interrupt logic automatically modifies the vector for the channel requesting service.

A control word is identified by a 1 in bit 0. A 0 in bit 2 indicates a time constant word is to follow. Interrupt vectors are always addressed to Channel 0, and identified by a 0 in bit 0.

Addressing. During programming, channels are addressed with the channel select pins CS_1 and CS_2. A 2-bit binary code selects the appropriate channel as shown in the following table.

Channel	CS_1	CS_0
0	0	0
1	0	1
2	1	0
3	1	1

Reset. The CTC has both hardware and software resets. The hardware reset terminates all down-counts and disables all CTC interrupts by resetting the interrupt bits in the control registers. In addition, the ZC/TO and Interrupt outputs go inactive, IEO reflects IEI, and D_0-D_7 go to the high-impedance state. All channels must be completely reprogrammed after a hardware reset.

The software reset is controlled by bit 1 in the channel control word. When a channel receives a software reset, it stops counting. When a software reset is used, the other bits in the control word also change the contents of the channel control register. After a software reset a new time constant word must be written to the same channel.

If the channel control word has both bits D_1 and D_2 set to 1, the addressed channel stops operating, pending a new time constant word. The channel is ready to resume after the new constant is programmed. In timer mode, if $D_3 = 0$, operation is triggered automatically when the time constant word is loaded.

Channel Control Word Programming. The channel control word is shown in Figure 5. It sets the modes and parameters described below.

Interrupt Enable. D_7 enables the interrupt, so that an interrupt output ($\overline{INT}$) is generated at zero count. Interrupts may be programmed in either mode and may be enabled or disabled at any time.

Operating Mode. D_6 selects either timer or counter mode.

Prescaler Factor. (Timer Mode Only). D_5 selects factor—either 16 or 256.

Trigger Slope. D_4 selects the active edge or slope of the CLK/TRG input pulses. Note that reprogramming the CLK/TRG slope during operation is equivalent to issuing an active edge. If the trigger slope is changed by a control word update while a channel is pending operation in timer mode, the result is the same as a CLK/TRG pulse and the timer starts. Similarly, if the channel is in counter mode, the counter decrements.

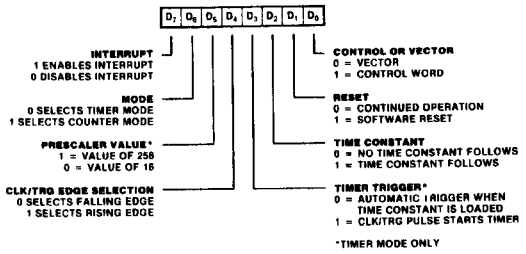


Figure 5. Channel Control Word

Programming (Continued)

Trigger Mode (Timer Mode Only). D_3 selects the trigger mode for timer operation. When D_3 is reset to 0, the timer is triggered automatically. The time constant word is programmed during an I/O write operation, which takes one machine cycle. At the end of the write operation there is a setup delay of one clock period. The timer starts automatically (decrements) on the rising edge of the second clock pulse (T_2) of the machine cycle following the write operation. Once started, the timer runs continuously. At zero count the timer reloads automatically and continues counting without interruption or delay, until stopped by a reset.

When D_3 is set to 1, the timer is triggered externally through the CLK/TRG input. The time constant word is programmed during an I/O write operation, which takes one machine cycle. The timer is ready for operation on the rising edge of the second clock pulse (T_2) of the following machine cycle. Note that the first timer decrement follows the active edge of the CLK/TRG pulse by a delay time of one clock cycle if a minimum setup time to the rising edge of clock is met. If this minimum is not met, the delay is extended by another clock period. Consequently, for immediate triggering, the CLK/TRG input must precede T_2 by one clock cycle plus its minimum setup time. If the minimum time is not met, the timer will start on the third clock cycle (T_3).

Once started the timer operates continuously, without interruption or delay, until stopped by a reset.

Time Constant to Follow. A 1 in D_2 indicates that the next word addressed to the selected channel is a time constant data word for the time constant register. The time constant word may be written at any time.

A 0 in D_2 indicates no time constant word is to follow. This is ordinarily used when the channel is already in operation and the new channel control word is an update. A channel will not operate without a time constant value. The only way to write a time constant value is to write a control word with D_2 set.

Software Reset. Setting D_1 to 1 causes a software reset, which is described in the Reset section.

Control Word. Setting D_0 to 1 identifies the word as a control word.

Time Constant Programming. Before a channel can start counting it must receive a time constant word from the CPU. During programming or reprogramming, a channel control word in which bit 2 is set must precede the time constant word to indicate that the next word is a time constant. The time constant word can be any value from 1 to 256 (Figure 6). Note that 00_{16} is interpreted as 256.

In timer mode, the time interval is controlled by three factors:

- The system clock period (ϕ)
- The prescaler factor (P), which multiplies the interval by either 16 or 256
- The time constant (T), which is programmed into the time constant register

Consequently, the time interval is the product of $\phi \times P \times T$. The minimum timer resolution is $16 \times \phi$ (4 μs with a 4 MHz clock). The maximum timer interval is $256 \times \phi \times 256$ (16.4 ms with a 4 MHz clock). For longer intervals timers may be cascaded.

Interrupt Vector Programming. If the Z-80 CTC has one or more interrupts enabled, it can supply interrupt vectors to the Z-80 CPU. To do so, the Z-80 CTC must be pre-programmed with the most-significant five bits of the interrupt vector. Programming consists of writing a vector word to the I/O port corresponding to the Z-80 CTC Channel 0. Note that D_0 of the vector word is always zero, to distinguish the vector from a channel control word. D_1 and D_2 are not used in programming the vector word. These bits are supplied by the interrupt logic to identify the channel requesting interrupt service with a unique interrupt vector (Figure 7). Channel 0 has the highest priority.

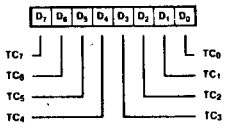


Figure 6. Time Constant Word

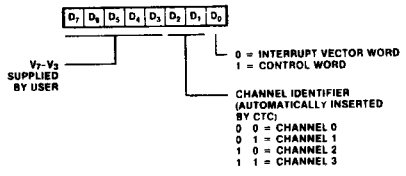


Figure 7. Interrupt Vector Word

Pin Description

$\overline{CE}$. *Chip Enable* (input, active Low). When enabled the CTC accepts control words, interrupt vectors, or time constant data words from the data bus during an I/O write cycle; or transmits the contents of the down-counter to the CPU during an I/O read cycle. In most applications this signal is decoded from the eight least significant bits of the address bus for any of the four I/O port addresses that are mapped to the four counter-timer channels.

CLK. *System Clock* (input). Standard single-phase Z-80 system clock.

CLK/TRG_0-CLK/TRG_3. *External Clock/Timer Trigger* (input, user-selectable active High or Low). Four pins corresponding to the four Z-80 CTC channels. In counter mode, every active edge on this pin decrements the down-counter. In timer mode, an active edge starts the timer.

CS_0-CS_1. *Channel Select* (inputs active High). Two-bit binary address code selects one of the four CTC channels for an I/O write or read (usually connected to A_0 and A_1).

D_0-D_7. *System Data Bus* (bidirectional, 3-state). Transfers all data and commands between the Z-80 CPU and the Z-80 CTC.

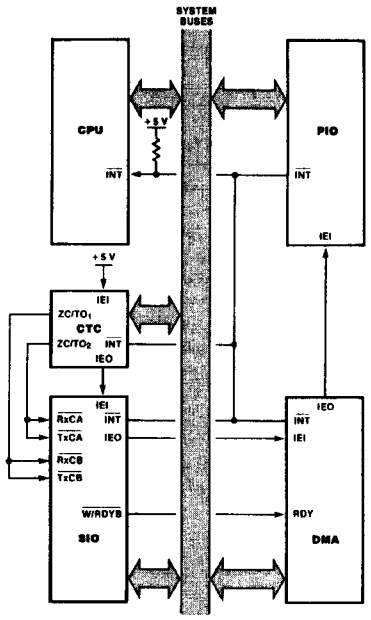


Figure 8. A Typical Z-80 Environment

$\overline{IEI}$. *Interrupt Enable In* (input, active High). A High indicates that no other interrupting devices of higher priority in the daisy chain are being serviced by the Z-80 CPU.

IEO. *Interrupt Enable Out* (output, active High). High only if IEI is High and the Z-80 CPU is not servicing an interrupt from any Z-80 CTC channel. IEO blocks lower priority devices from interrupting while a higher priority interrupting device is being serviced.

$\overline{INT}$. *Interrupt Request* (output, open drain, active Low). Low when any Z-80 CTC channel that has been programmed to enable interrupts has a zero-count condition in its down-counter.

$\overline{IORQ}$. *Input/Output Request* (input from CPU, active Low). Used with $\overline{CE}$ and $\overline{RD}$ to transfer data and channel control words between the Z-80 CPU and the Z-80 CTC. During a write cycle, $\overline{IORQ}$ and $\overline{CE}$ are active and $\overline{RD}$ inactive. The Z-80 CTC does not receive a specific write signal; rather, it internally generates its own from the inverse of an active $\overline{RD}$ signal. In a read cycle, $\overline{IORQ}$, $\overline{CE}$ and $\overline{RD}$ are active; the contents of the down-counter are read by the Z-80 CPU. If $\overline{IORQ}$ and $\overline{M1}$ are both true, the CPU is acknowledging an interrupt request, and the highest priority interrupting channel places its interrupt vector on the Z-80 data bus.

$\overline{M1}$. *Machine Cycle One* (input from CPU, active Low). When $\overline{M1}$ and $\overline{IORQ}$ are active, the Z-80 CPU is acknowledging an interrupt. The Z-80 CTC then places an interrupt vector on the data bus if it has highest priority, and if a channel has requested an interrupt ($\overline{INT}$).

$\overline{RD}$. *Read Cycle Status* (input, active Low). Used in conjunction with $\overline{IORQ}$ and $\overline{CE}$ to transfer data and channel control words between the Z-80 CPU and the Z-80 CTC.

$\overline{RESET}$. *Reset* (input active Low). Terminates all down-counts and disables all interrupts by resetting the interrupt bits in all control registers; the ZC/TO and the Interrupt outputs go inactive; IEO reflects IEI; D_0-D_7 go to the high-impedance state.

ZC/TO_0-ZC/TO_2. *Zero Count/Timeout* (output, active High). Three ZC/TO pins corresponding to Z-80 CTC channels 2 through 0 (Channel 3 has no ZC/TO pin). In both counter and timer modes the output is an active High pulse when the down-counter decrements to zero.

Timing

Read Cycle Timing. Figure 9 shows read cycle timing. This cycle reads the contents of a down-counter without disturbing the count. During clock cycle T_2, the Z-80 CPU initiates a read cycle by driving the following inputs Low: $\overline{RD}$, $\overline{IORQ}$, and $\overline{CE}$. A 2-bit binary code at inputs CS_1 and CS_0 selects the channel to be read. $\overline{M1}$ must be High to distinguish this cycle from an interrupt acknowledge. No additional wait states are allowed.

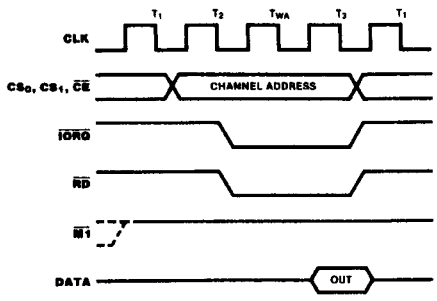


Figure 9. Read Cycle Timing

Write Cycle Timing. Figure 10 shows write cycle timing for loading control, time constant or vector words.

The CTC does not have a write signal input, so it generates one internally when the read ($\overline{RD}$) input is High during T_1. During T_2 $\overline{IORQ}$ and $\overline{CE}$ inputs are Low. $\overline{M1}$ must be High to distinguish a write cycle from an interrupt acknowledge. A 2-bit binary code at inputs CS_1 and CS_0 selects the channel to be addressed, and the word being written is placed on the Z-80 data bus. The data word is latched into the appropriate register with the rising edge of clock cycle T_{WA}. No additional wait states are allowed.

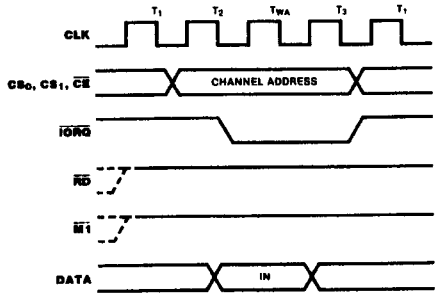


Figure 10. Write Cycle Timing

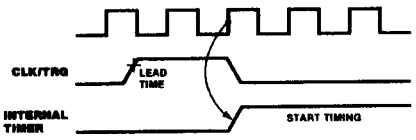


Figure 11. Timer Mode Timing

Timer Operation. In the timer mode, a CLK/TRG pulse input starts the timer (Figure 11) on the second succeeding rising edge of CLK. The trigger pulse is asynchronous and it must have a minimum width. A minimum lead time (210 ns) is required between the active edge of the CLK/TRG and the next rising edge of CLK to enable the prescaler on the following clock edge. If the CLK/TRG edge occurs closer than this, the initiation of the timer function is delayed one clock cycle. This corresponds to the startup timing discussed in the programming section. The timer can also be started automatically if so programmed by the channel control word.

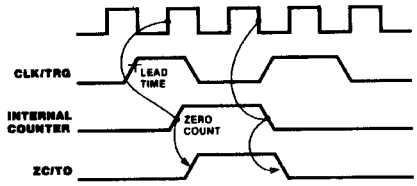


Figure 12. Counter Mode Timing

Counter Operation. In the counter mode, the CLK/TRG pulse input decrements the down-counter. The trigger is asynchronous, but the count is synchronized with CLK. For the decrement to occur on the next rising edge of CLK, the trigger edge must precede CLK by a minimum lead time as shown in Figure 12. If the lead time is less than specified, the count is delayed by one clock cycle. The trigger pulse must have a minimum width, and the trigger period must be at least twice the clock period.

The ZC/TO output occurs immediately after zero count, and follows the rising CLK edge.

Interrupt Operation

The Z-80 CTC follows the Z-80 system interrupt protocol for nested priority interrupts and return from interrupt, wherein the interrupt priority of a peripheral is determined by its location in a daisy chain. Two lines—IEI and IEO—in the CTC connect it to the system daisy chain. The device closest to the +5 V supply has the highest priority (Figure 13). For additional information on the Z-80 interrupt structure, refer to the *Z-80 CPU Product Specification* and the *Z-80 CPU Technical Manual.*

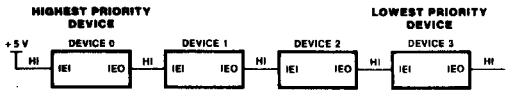


Figure 13. Daisy-Chain Interrupt Priorities

Within the Z-80 CTC, interrupt priority is predetermined by channel number: Channel 0 has the highest priority, and Channel 3 the lowest. If a device or channel is being serviced with an interrupt routine, it cannot be interrupted by a device or channel with lower priority until service is complete. Higher priority devices or channels may interrupt the servicing of lower priority devices or channels.

A Z-80 CTC channel may be programmed to request an interrupt every time its down-counter reaches zero. Note that the CPU must be programmed for interrupt mode 2. Some time after the interrupt request, the CPU sends an interrupt acknowledge. The CTC interrupt control logic determines the highest priority channel that is requesting an interrupt. Then, if the CTC IEI input is High (indicating that it has priority within the system daisy chain) it places an 8-bit interrupt vector on the system data bus. The high-order five bits of this vector were written to the CTC during the programming process; the next two bits are provided by the CTC interrupt control logic as a binary code that identifies the highest priority channel requesting an interrupt; the low-order bit is always zero.

Interrupt Acknowledge Timing. Figure 14 shows interrupt acknowledge timing. After an interrupt request, the Z-80 CPU sends an interrupt acknowledge ($\overline{M1}$ and $\overline{IORQ}$). All channels are inhibited from changing their interrupt request status when $\overline{M1}$ is active—about two clock cycles earlier than $\overline{IORQ}$. $\overline{RD}$ is High to distinguish this cycle from an instruction fetch.

The CTC interrupt logic determines the highest priority channel requesting an interrupt. If the CTC interrupt enable input (IEI) is High, the highest priority interrupting channel within the CTC places its interrupt vector on the data bus when $\overline{IORQ}$ goes Low. Two wait states (T_{WA}) are automatically inserted at this time to allow the daisy chain to stabilize. Additional wait states may be added.

Return from Interrupt Timing. At the end of an interrupt service routine the RETI (Return From Interrupt) instruction initializes the daisy chain enable lines for proper control of nested priority interrupt handling. The CTC decodes the 2-byte RETI code internally and determines whether it is intended for a channel being serviced. Figure 15 shows RETI timing.

If several Z-80 peripherals are in the daisy chain, IEI settles active (High) on the chip currently being serviced when the opcode ED_{16} is decoded. If the following opcode is $4D_{16}$, the peripheral being serviced is released and its IEO becomes active. Additional wait states are allowed.

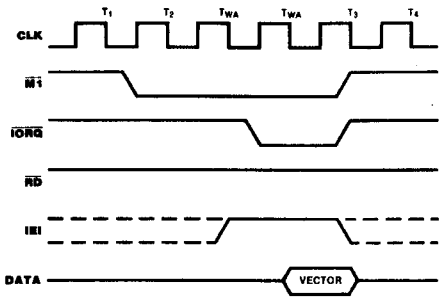


Figure 14. Interrupt Acknowledge Timing

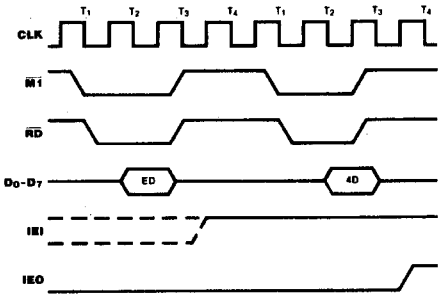


Figure 15. Return From Interrupt Timing

APPENDIX E : MEMORY DECODER PROM

P/N 745473

MICRO/SYS P/N DP0002

PROM ADDRESS	DATA	PROM ADDRESS	DATA
000	FF	080	FE
↓	↓	081	FE
01F	FF	082	FD
020	FF	083	FD
↓	↓	084	FB
03F	FF	085	FB
040	FE	086	F7
041	FD	087	F7
042	FB	088	FF
043	F7	↓	↓
044	FF	09F	FF
↓	↓	0A0	FE
05F	FF	0A1	FE
060	FE	0A2	FE
061	FE	0A3	FE
062	FD	0A4	FD
063	FD	0A5	FD
064	FB	0A6	FD
065	F7	0A7	FD
066	FF	0A8	FB
↓	↓	0A9	F7
07F	FF	0AA	FF
		↓	↓
		0BF	FF

PROM ADDRESS	DATA
0C0	FE
0C1	
0C2	
0C3	FE
0C4	FD
0C5	
0C6	
0C7	FD
0C8	FB
0C9	FB
0CA	F7
0CB	F7
0CC	FF
↓	↓
0DF	FF
0E0	FE
0E1	
0E2	↓
0E3	FE
0E4	FD
0E5	
0E6	↓
0E7	FD
0E8	FB
0E9	
0EA	↓
0EB	FB
0EC	F7
0ED	
0EE	↓
0EF	F7

PROM ADDRESS	DATA
0F0	FF
↓	↓
0FF	FF
100	FE
101	FE
102	FF
11F	FF
120	FE
121	FE
122	FD
123	FF
↓	↓
13F	FF
140	FF
↓	↓
1FF	FF

APPENDIX G : PAL EQUATIONS

P/N PL0002A

$\overline{MCSYNC} = \overline{RD} * \overline{WR} * \overline{ACK}$(pin 23)

DATAOUT = $\overline{RD} * \overline{ACK}$(pin 23) + MEMSEL(pin 14) * RD +
IORQ * IODEC * RD + ACK(pin 23) * $\overline{IEO}$

CTCCE = IODEC * $\overline{A2}$

8251CE = IORQ * IODEC * $\overline{A1}$ * A2

PORT0 = IORQ * IODEC * WR * A1 * A2

SELECT(pin 15) = CS1 + CS2 + CS3 + CS4

INTAK(pin 21) = IORQ * M1