

MPC-4

UNIT PUBLICATION

SD #7140061 REV. C
AUGUST, 1981

SD
SYSTEMS
P.O. Box 28810, Dallas, Texas 75228

MPC-4

(MULTI-PORT COMMUNICATOR WITH FOUR SERIAL PORTS)

UNIT PUBLICATION

REVISION C
AUGUST, 1981
SDSystems Inc.

TABLE OF CONTENTS

PARAGRAPH	TITLE	PAGE
SECTION I INTRODUCTION		
1.1	<u>GENERAL DESCRIPTION</u>	1-1
1.2	<u>FUNCTIONAL DESCRIPTION</u>	1-1
1.2.1	Z80 CPU.....	1-1
1.2.2	CTC.....	1-1
1.2.3	OSCILLATOR.....	1-1
1.2.4	PROGRAMMABLE BAUD RATE GENERATORS.....	1-1
1.2.5	SERIAL I/O.....	1-2
1.2.6	ADDRESS BUFFER.....	1-2
1.2.7	DATA IN BUFFER.....	1-2
1.2.8	DATA OUT BUFFER.....	1-2
1.2.9	STATUS PORT.....	1-2
1.2.10	INTERRUPT CONTROL.....	1-2
1.2.11	ADDRESS DECODER.....	1-2
1.2.12	RANDOM ACCESS MEMORY.....	1-2
1.2.13	PROM/RAM.....	1-3
1.2.14	BOARD SELECT LOGIC.....	1-3
1.2.15	PORT DECODE.....	1-3
1.2.16	MUX.....	1-3
1.2.17	WAIT STATE GENERATOR.....	1-3
1.3	<u>SIGNALS USED</u>	1-3
SECTION II THEORY OF OPERATION		
2.1	<u>INTRODUCTION</u>	2-1
2.2	<u>OVERALL OPERATION</u>	2-1
SECTION III TECHNICAL DESCRIPTION		
3.1	<u>INTRODUCTION</u>	3-1
3.2	<u>CTC AS REAL TIME CLOCK</u>	3-1
3.3	<u>PROGRAMMABLE BAUD RATE GENERATORS</u>	3-1
3.4	<u>SERIAL I/O</u>	3-2
3.5	<u>STATUS PORT</u>	3-3
3.6	<u>INTERRUPT CONTROL</u>	3-3
SECTION IV DATA MANIPULATION		
4.1	<u>MEMORY ORGANIZATION</u>	4-1
4.2	<u>INPUTTING DATA FROM USER</u>	4-2
4.3	<u>OUTPUTTING DATA TO USER</u>	4-2

TABLE OF CONTENT (Continued)

PARAGRAPH	TITLE	PAGE
SECTION V OPTIONS		
5.1	<u>INTRODUCTION</u>	5-1
5.2	<u>BOARD SELECT OPTIONS (X10)</u>	5-1
5.3	<u>MEMORY OPTIONS (E1-E9,X9)</u>	5-1
5.4	<u>CTC OPTIONS (X11)</u>	5-3
5.5	<u>SERIAL I/O OPTIONS (X1-X8)</u>	5-3
5.6	<u>INTERRUPT OPTIONS (E10-E14)</u>	5-3
SECTION VI CONSTRUCTION		
6.1	<u>INTRODUCTION</u>	6-1
6.2	<u>ASSEMBLY PROCEDURE</u>	6-1
6.3	<u>VOLTAGE CHECK</u>	6-2
SECTION VII TESTING		
7.1	<u>INTRODUCTION</u>	7-1
7.2	<u>SUPPLY VOLTAGE</u>	7-1
7.3	<u>BOARD SELECT</u>	7-1
7.4	<u>MEMORY TEST</u>	7-1
7.5	<u>SETUP</u>	7-1
SECTION VIII TROUBLESHOOTING GUIDE		
8.1	<u>INTRODUCTION</u>	8-1
8.2	<u>POWER SUPPLY</u>	8-1
8.3	<u>BOARD SELECT</u>	8-2
8.4	<u>MEMORY TEST</u>	8-2
8.5	<u>SETUP</u>	8-3
APPENDIX A	MPC-4 SCHEMATIC DIAGRAM.....	A-1
APPENDIX B	MPC-4 BILL OF MATERIALS.....	B-1
APPENDIX C	ASSEMBLY DRAWING.....	C-1
APPENDIX D	MPC CONTROL SOFTWARE LISTING.....	D-1
APPENDIX E	MEMORY MAP.....	E-1
APPENDIX F	MPC-4 PORT DESIGNATION.....	F-1
APPENDIX G	CTC PROGRAMMING.....	G-1
APPENDIX H	Z80 DART PROGRAMMING.....	H-1
APPENDIX I	MPC-4 HOST PROGRAM.....	I-1

LIST OF ILLUSTRATIONS

FIGURE	TITLE	PAGE
2-1	Instructional OP Code Fetch.....	2-2
2-2	Memory Read and Write Cycle.....	2-2
2-3	Interrupt Request/Acknowledge Cycle.....	2-3
2-4	Non Maskable Interrupt Request Operation.....	2-3
2-5	Bus Reques/Acknowledge Cycle.....	2-4
7-1	MPC-4 Random Access Memory.....	7-2
7-2	MPC-4 Random Access Memory.....	7-2
7-3	MPC-4 Random Access Memory.....	7-3
7-4	MPC-4 Random Access Memory.....	7-3

LIST OF TABLES

TABLE	TITLE	PAGE
1-1	MPC-4 S-100 Bus Signals.....	1-4
3-1	Baud Rate Constants.....	3-2
5-1	Board Select Jumper Options.....	5-1
5-2	Jumper Options For U19.....	5-2
5-3	Jumper Options For U20.....	5-2
5-4	Block Size Jumpers.....	5-2
5-5	Boundary Jumpers.....	5-2

SECTION I
INTRODUCTION

1.1 GENERAL DESCRIPTION

The MPC-4 is an S-100 Bus processor board which functions as a controller and interface for four serial terminals.

The MPC-4 is controlled by an on-board Z80 microprocessor. There are two 24 pin sockets, one of which can be configured to hold 2K or 4K PROMS, while the other can be configured for 1K or 2K PROMS or RAMS (1K or 2K X 8). The board provides a programmable baud rate for each terminal, two DART's (dual asynchronous receiver/transmitter) for serial interface, a four channel counter timer chip, and 1K of static RAM.

A real time clock is also provided.

1.2 FUNCTIONAL DESCRIPTION

Figure 1-1 is a block diagram of the MPC-4. The following sections contain a description of each block.

1.2.1 Z80 CPU

The Z80 CPU is the heart of the MPC-4. The Z80 CPU controls the DART's, which handle I/O to all of the four serial ports. The CPU also helps control interface between the board and the system.

1.2.2 CTC

The CTC provides one of its four channels for use as a real time clock. The other three channels are available to the user via connections on X11. See Appendix G for programming details.

1.2.3 OSCILLATOR

The OSCILLATOR provides a 4.9152 MHZ clock. This is used directly by the baud rate generators and is divided by 2 to provide the system clock.

1.2.4 PROGRAMMABLE BAUD RATE GENERATORS

The PROGRAMMABLE BAUD RATE GENERATORS (PBRG's) provide a baud rate to each of the four serial channels. The baud rate varies as a function of the constants output to the PBRG's.

1.2.5 SERIAL I/O

Serial I/O is provided by two Z80 DART's (dual asynchronous receiver/transmitter). They are programmed to generate interrupts for data received. See Appendix H for DART programming details.

1.2.6 ADDRESS BUFFER

Serves as an interface between the address bus on the MPC-4 and the addresses on the S-100 bus.

1.2.7 DATA IN BUFFER

Serves as an interface between the data bus on the MPC-4 and the data out lines on the S-100 bus.

1.2.8 DATA OUT BUFFER

Serves as an interface between the data bus on the MPC-4 and the data in lines on the S-100 bus.

1.2.9 STATUS PORT

Serves to inform host system of each users input and output status.

1.2.10 INTERRUPT CONTROL

Generates host system interrupt whenever data is output to status port (if jumper installed). Interrupt is cleared when data is input by the host system. Also responsible for generating NMI when system outputs to port FXH where X=the board number.

1.2.11 ADDRESS DECODER

Uses addresses to form memory enables and signals according to PROM and RAM sizes and boundaries.

1.2.12 RANDOM ACCESS MEMORY (RAM)

Consists of 1K X 8 static RAM (2114-3). This RAM is where the data buffers, stacks, etc., are located.

1.2.13 PROM/RAM

Consists of two sockets, one being PROM, selectable as 2K or 4K, and the other being RAM or PROM, selectable as 1K or 2K (see Options, section 5.0).

1.2.14 BOARD SELECT LOGIC

Generates BUSRQ when board is selected and also enables address buffer.

1.2.15 PORT DECODE

Responsible for decoding the address lines to select the appropriate ports.

1.2.16 MUX

Multiplexer to select appropriate signals depending on whether host system or onboard CPU has control of the board.

1.2.17 WAIT STATE GENERATOR

Puts host system in wait state from the time the board is selected, causing a BUSRQ, until the onboard CPU issues a BUSAK, causing its address, data, and control lines to TRI-STATE. This prevents conflicts between the two CPU's (host and MPC).

1.3 SIGNALS USED

Table 1-1 shows the S-100 Bus signals used by the MPC-4.

Table 1-1. MPC-4 S-100 Bus Signals.

SIGNAL	PIN #	DIRECTION
DI0-DI7	95,94,41,42,91,92,93,43	OUTPUT
DO0-DO7	36,35,88,89,38,38,40,90	INPUT
A0-A7	79,80,81,31,30,29,82,83	INPUT
A8-A15	84,34,37,87,33,85,86,32	INPUT
PRDY	72	OUTPUT
SOUT	45	INPUT
SINP	46	INPUT
VI1	5	OUTPUT (OPTIONAL)
VI2	6	OUTPUT (OPTIONAL)
VI3	7	OUTPUT (OPTIONAL)
VI4	8	OUTPUT (OPTIONAL)
MEMR	47	INPUT
MEMW	68	INPUT
POC	99	INPUT
+8	1,51	
+16	2	
-16	52	
GND	50,100	

SECTION II
THEORY OF OPERATION

2.1 INTRODUCTION

The following discussion assumes that the reader has a basic understanding of digital electronics and of the Z80. Figures 2-1, 2-2, 2-3, 2-4, and 2-5 contain the timing diagrams for the Z80 CPU.

2.2 OVERALL OPERATION

The MPC-4 is controlled by a Z80 CPU which executes a firmware program located at address 0000H. This program is responsible for I/O and housekeeping. All interrupt routines are contained here also.

Data is input to the MPC from the terminals via two Z80 DART's, (dual asynchronous receiver/transmitter). As a user inputs data the DART interrupts the CPU. The CPU in turn executes the appropriate interrupt service routine, which places this data in a separate memory buffer for each user and flags the system to show data available.

Data is output to the terminals by the DART's in much the same manner. The system outputs data to the MPC and this data is entered into a memory buffer (one for each user). This data is then output to the appropriate user via the DART's.

The CTC furnishes one channel for use as a real time clock. The other three channels may be utilized using jumpers. The system clock is divided down by the CTC and generates timed interrupts. These interrupts are counted to provide seconds, minutes, hours, day, month and year.

The MPC-4 causes the host system to enter a wait state when the board is selected. A board select causes a BUSRQ to the Z80 CPU and this BUSRQ generates the wait state. The wait state remains until the Z80 CPU responds with a BUSAK signal. The BUSAK signal is used as a multiplex signal, as well as a direct enable for the address buffers and an indirect enable for the data out and data in buffers.

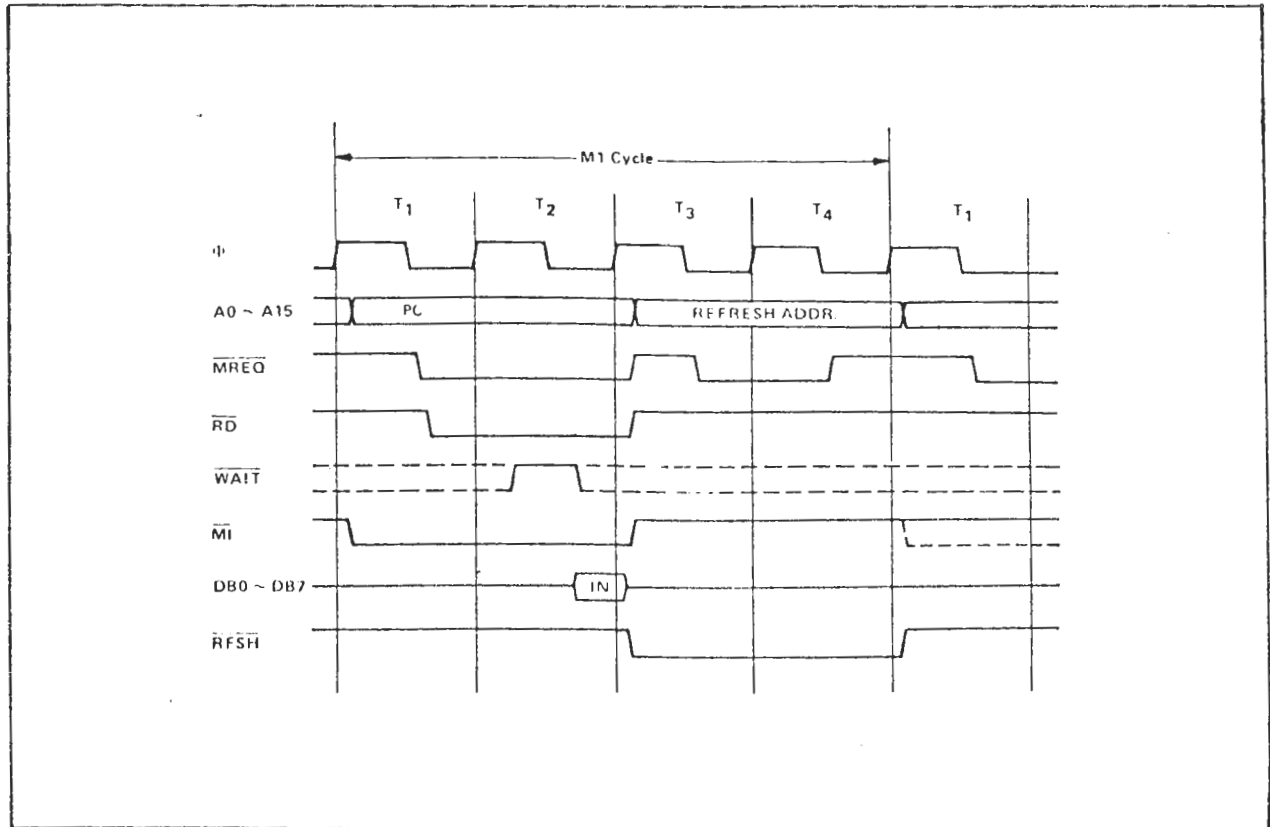


Figure 2-1. Instruction OP Code Fetch.

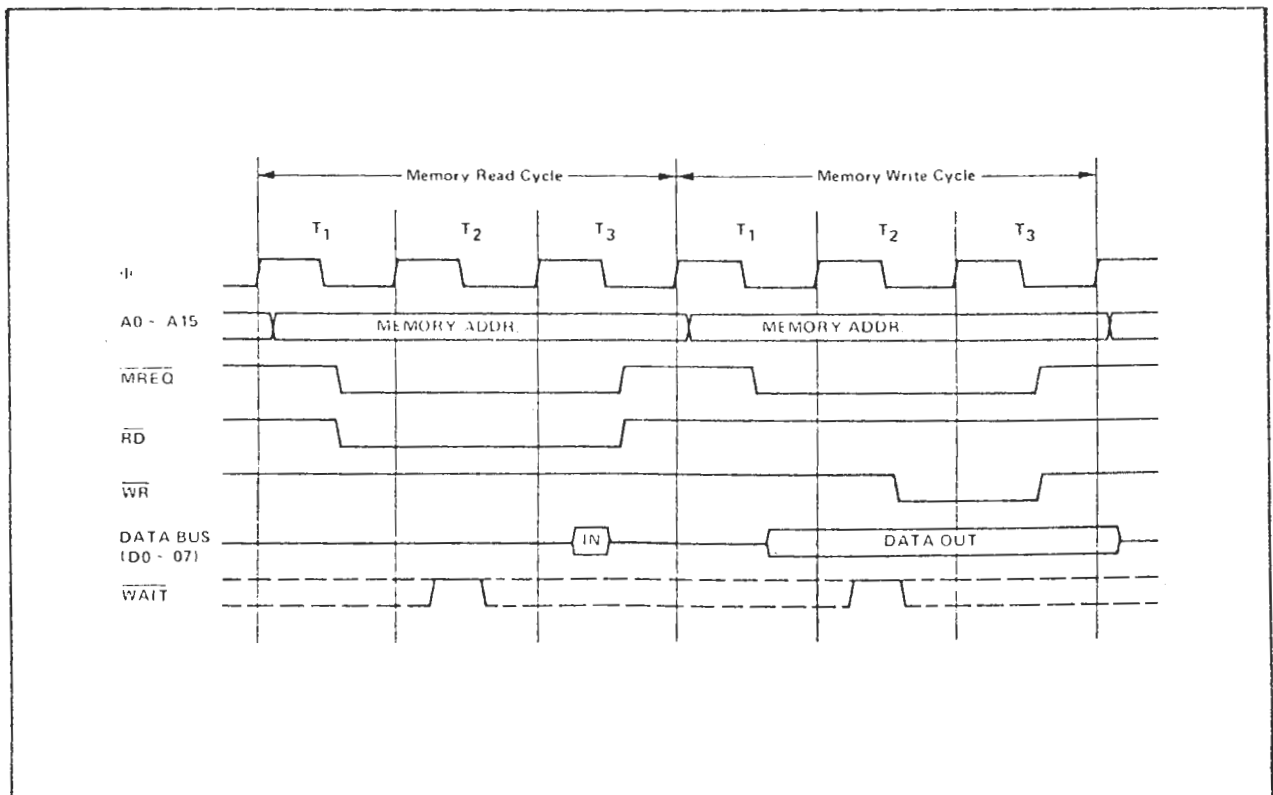


Figure 2-2. Memory Read and Write Cycle.

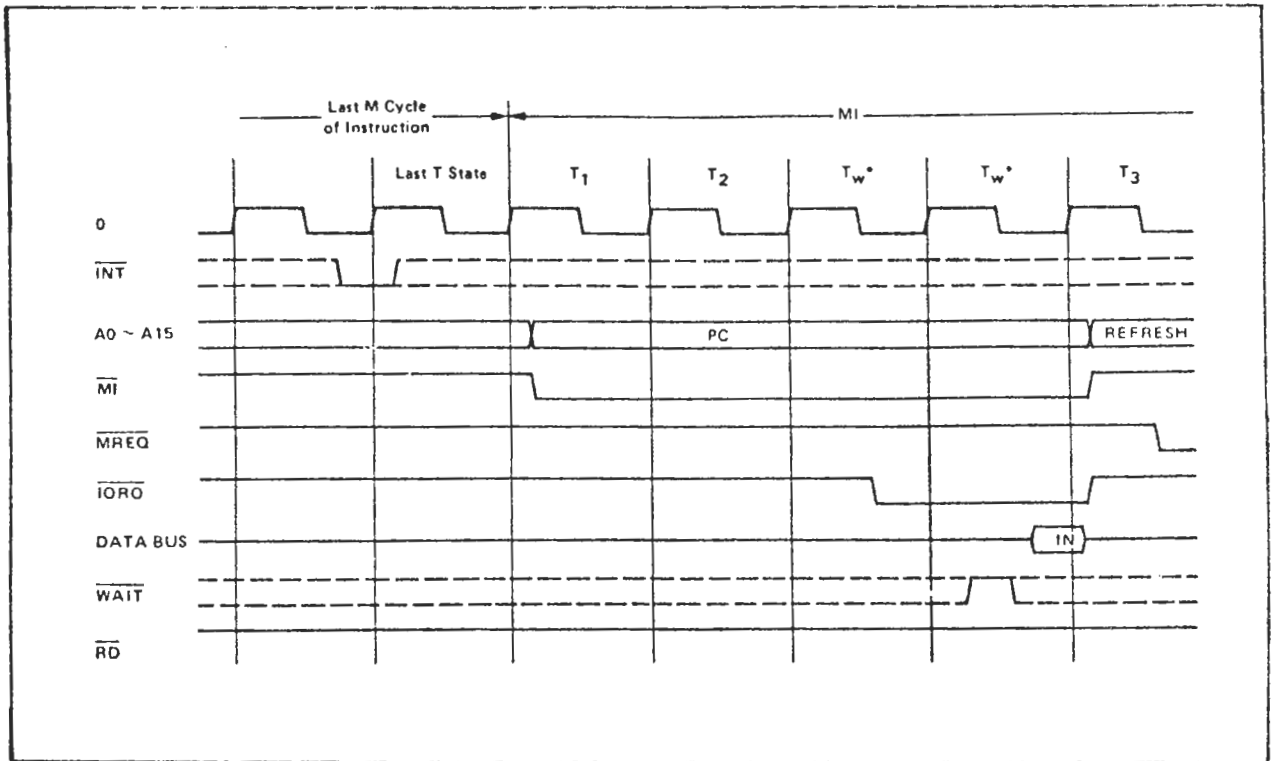


Figure 2-3. Interrupt Request/Acknowledge Cycle.

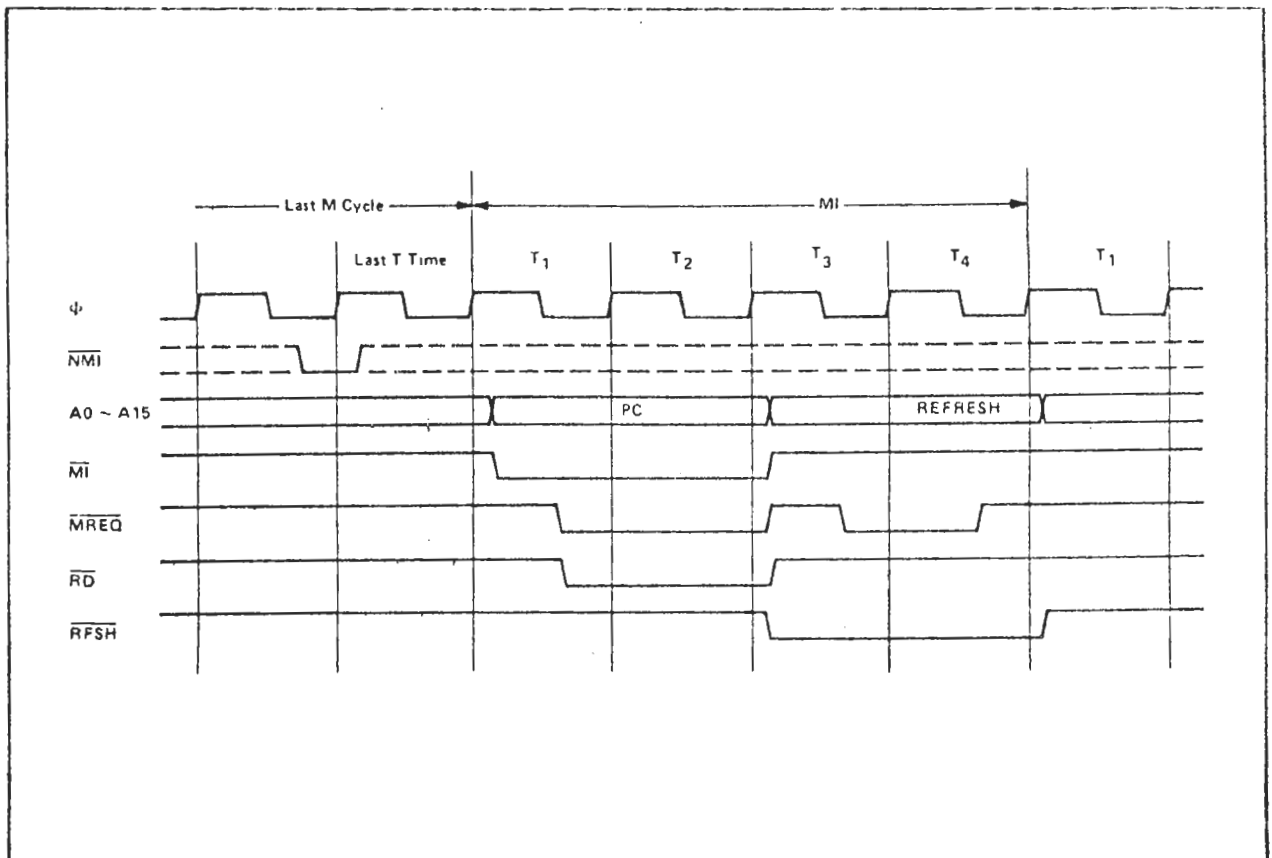


Figure 2-4. Non Maskable Interrupt Request Operation.

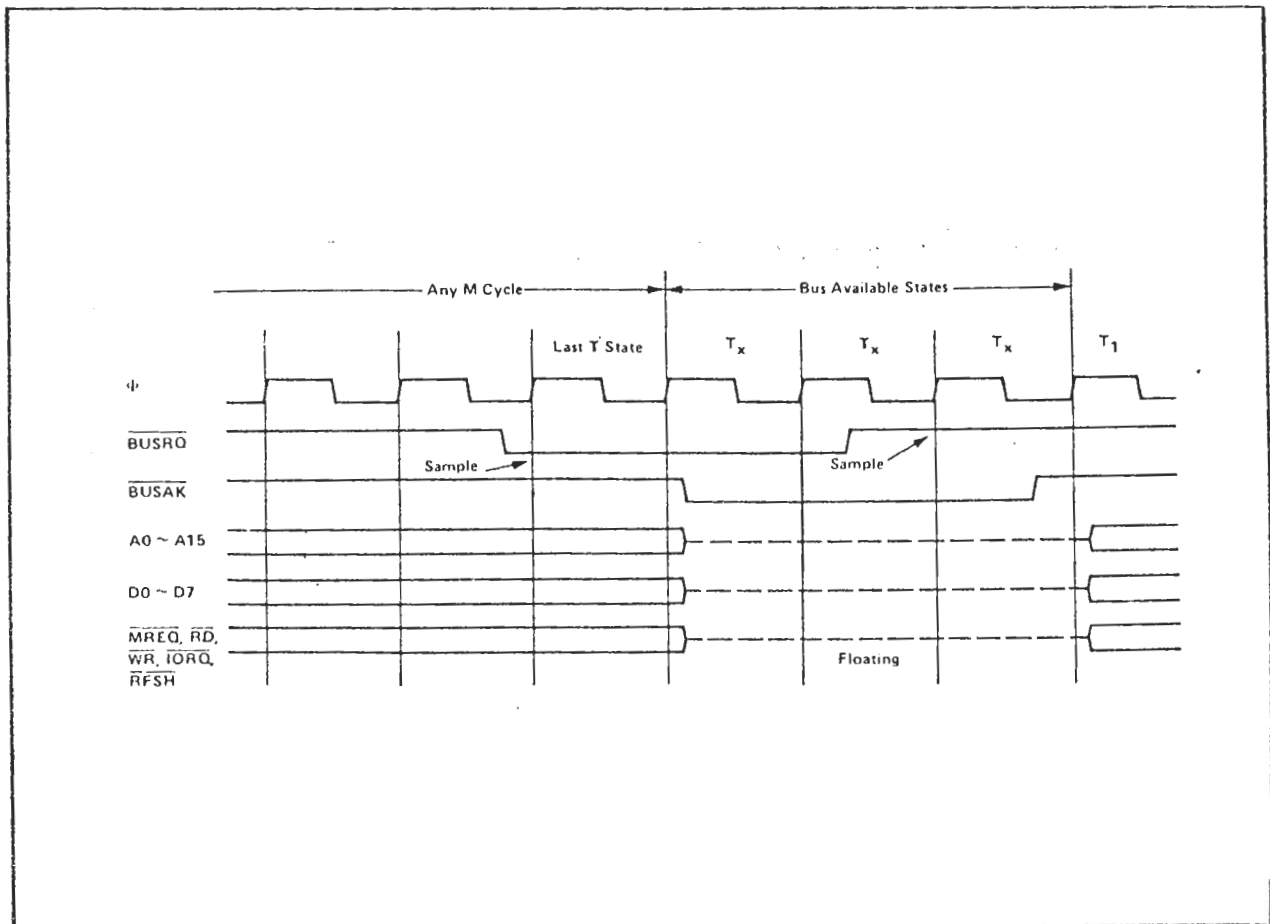


Figure 2-5. Bus Request/Acknowledge Cycle.

SECTION III

TECHNICAL DESCRIPTION

3.1 INTRODUCTION

This description is meant to explain the operation of those blocks in the block diagram which warrant further explanation. It may prove helpful for the reader to also use the software listing (Appendix D) during this discussion.

3.2 CTC AS REAL TIME CLOCK

In this application, the CTC is used in the timer mode. It is programmed with a prescaler of 256 and a time constant of 128. These divisions are performed on the system clock (2.4576 MHz). The CTC generates an interrupt every time this division is completed or with a frequency of $2.4576\text{MHz} - 256 - 128 = 75\text{ Hz}$.

The remainder of the process is taken care of through software. The year, month, day, hour, minute and second counts are kept in memory in BCD form. See the Memory Map in Appendix E for these locations.

3.3 PROGRAMMABLE BAUD RATE GENERATORS

The four baud rates are provided by two COM 5016T-5 PROGRAMMABLE BAUD RATE GENERATORS (PBRG's). Each PBRG has two channels and each can provide any of sixteen (16) standard baud rates between 50 and 19200 Baud, according to a one half byte constant which is latched into it. By outputting a byte to a port corresponding to one of the PBRG's, it is possible to program both channels at the same time. The constants set the baud rate as shown in Table 3-1.

Table 3-1. Baud Rate Constants.

1/2 BYTE CODE	BAUD RATE
0	50
1	75
2	1
3	134.5
4	150
5	300
6	600
7	1200
8	1800
9	2000
A	2400
B	3600
C	4800
D	7200
E	9600
F	19,200

The BRG for users 1 and 2 is accessed as port 0DH. D4-D7 determine the baud rate for user 1. D0-D3 correspond to user 2. The BRG for users 3 and 4 is accessed as port 0EH. D4-D7 correspond to user 3 and D0-D3 correspond to user 4.

3.4 SERIAL I/O

The serial I/O section utilizes two Z80 DART's (dual asynchronous receiver/transmitter) to provide four (4) asynchronous serial interfaces. The Z80 DART's are versatile in that they may be programmed to perform a variety of functions. The DARTS in this application are programmed to interrupt whenever any of their channels has a receive character available. Transmission occurs as an output to the port corresponding to the channel where the data is to be output. It is recommended that the user read Appendix H to better understand the programming of the Z80 DART.

Whenever data is input to a channel of one of the DART's, that channel generates an interrupt and places a vector on the data bus. This vector will be changed according to whether channel A or channel B is interrupting. By programming a different vector into each DART the service routine for each user will have a different starting location. Using the 'Status Affects Vector' mode, the vectors are changed as follows:

	VECTOR WRITTEN	VECTOR PROVIDED AT INTERRUPT
USER 1 (CHANNEL A)	D0H	D4H
USER 2 (CHANNEL B)	D0H	DCH
USER 3 (CHANNEL A)	C0H	C4H
USER 4 (CHANNEL B)	C0H	CCH

3.5 STATUS PORT

The Status Port is seen as port 0FH by the CPU on the MPC-4 and as FXH by the system CPU where X=board number.

The Status Port contains information pertaining to each user's input and output status. The information available to the host system by reading port FXH (Status Port) is arranged as follows:

BIT	7	6	5	4	3	2	1	0
PURPOSE	User 4 Output	User 3 Output	User 2 Output	User 1 Output	User 4 Input	User 3 Input	User 2 Input	User 1 Input

BIT DEFINITIONS

OUTPUT "0" = BUFFER FULL NOT READY FOR DATA
 "1" = READY FOR DATA
 INPUT "0" = NO DATA TO INPUT
 "1" = DATA TO INPUT

When the host system issues an output to the status port, the port is not changed but an NMI (non-maskable interrupt) is generated to the CPU. The service routine for the NMI updates the status byte and outputs the revised status to the status port. (See Appendix I - host system driver for MPC-4)

3.6 INTERRUPT CONTROL

If bus interrupts are selected an interrupt is generated whenever the status port is written to. If the data in the status port is then needed, the interrupt service routine may read the status port. It must be noted that if this interrupt option is used the vector must be written into the CTC on the host system's SBC.

SECTION IV

DATA MANIPULATION

4.1 MEMORY ORGANIZATION

The RAM is located at 1000H on the board as shipped. The memory map for this RAM is contained in Appendix E and will be a useful aid in this discussion.

The first 40H bytes are divided into eight (8) separate sections as follows:

1000H-1007H - USER 1 IN	1008H-100FH - USER 1 OUT
1010H-1017H - USER 2 IN	1018H-101FH - USER 2 OUT
1020H-1027H - USER 3 IN	1028H-102FH - USER 3 OUT
1030H-1037H - USER 4 IN	1038H-103FH - USER 4 OUT

Each of these sections is represented by a buffer section in memory (see Memory Map). The first two bytes of each section contain a pointer which points to the beginning address of that section's buffer.

The third byte contains the buffer size for each section. The input buffers are 80 bytes long and the output buffers are 128 bytes long.

The fourth byte is called the buffer insert offset. When this value is added to the beginning address of the buffer, the result points to the location where the next byte of data is to be stored.

The fifth byte is called the buffer extract offset. When it is added to the beginning address of the buffer the result points to the location from which the next byte of data is to be taken.

The sixth byte is the data byte. During input this byte represents the data just taken from the extract address in the buffer for input to the system. During output it represents the data output by the system for storage in the insert address of the buffer.

The seventh byte is the number of the data port for a particular user.

USER 1 IN & OUT - PORT 0
USER 2 IN & OUT - PORT 2
USER 3 IN & OUT - PORT 4
USER 4 IN & OUT - PORT 6

4.2 INPUTTING DATA FROM USER

Appendix I is the host system driver for the MPC-4 and should be referenced as an example for sections 4.2 and 4.3. The CPU takes data from the DART and places it in the location pointed to by the sum of the buffer beginning address and the buffer insert offset. The buffer insert offset is then incremented. The insert offset will always be greater than the extract offset. The software guards against overwriting the extract with the insert. The location (buffer begin + extract) points to the next byte to be output to the host system. This byte is put in the data byte location and the extract offset is incremented. The host system waits for the specified user's input ready bit on the status port, then takes the data from the data byte location and replaces it with 0FFH. An NMI is then generated by the host writing to port 0FXH. The MPC-4 then repeats with the next byte and continues until data is all output or buffer is full.

4.3 OUTPUTTING DATA TO USER

This procedure is basically the same as the input procedure. When the host system has data to output it waits for the specified output ready bit on the status port, then it selects the MPC-4. This gives the system access to the memory of the MPC-4. The data to be output is placed in the user's data byte location and control is returned to the MPC-4. This data byte is transferred to the data insert location and the insert offset is incremented. The data byte is zeroed. Next the byte in the data extract location is output to the DART for the particular user. The extract offset is then incremented and the procedure repeated until all data is output or the buffer is full.

SECTION V

OPTIONS

5.1 INTRODUCTION

There are certain options available to the user of the MPC-4. They fall in the following categories:

- A. BOARD SELECT
- B. MEMORY
- C. CTC
- D. SERIAL I/O
- E. BUS INTERRUPTS

5.2 BOARD SELECT OPTIONS (X10)

In the board select logic X10 defines the board number. This number may range from 00 to 0FH. The chart in Table 5-1 shows how X10 should be jumpered for each board number.

Table 5-1. Board Select Jumper Options.

BOARD JUMPERS TO INSTALL	
#	
00	X10-1 TO X10-8, X10-2 TO X10-7, X10-3 TO X10-6, X10-4 TO X10-5
01	X10-1 TO X10-8, X10-2 TO X10-7, X10-3 TO X10-6
02	X10-1 TO X10-8, X10-2 TO X10-7 X10-4 TO X10-5
03	X10-1 TO X10-8, X10-2 TO X10-7
04	X10-1 TO X10-8, X10-3 TO X10-6, X10-4 TO X10-5
05	X10-1 TO X10-8, X10-3 TO X10-6
06	X10-1 TO X10-8, X10-4 TO X10-5
07	X10-1 TO X10-8
08	X10-2 TO X10-7, X10-3 TO X10-6, X10-4 TO X10-5
09	X10-2 TO X10-7, X10-3 TO X10-6
0A	X10-2 TO X10-7 X10-4 TO X10-5
0B	X10-2 TO X10-7
0C	X10-3 TO X10-6, X10-4 TO X10-5
0D	X10-3 TO X10-6
0E	X10-4 TO X10-5
0F	

The MPC-4 comes to the user configured as board #7. This is for use with COSMOS in the SD Microcomputers family. If a different board number is desired the etch jumper between X10-1 and X10-8 may need to be cut.

5.3 MEMORY OPTIONS (E1-E9, X9)

There are a number of options available to the user as to the size, type and boundaries of the memory on the MPC-4.

The socket at U20 may contain either RAM, ROM, or EPROM in sizes of 1K X 8 or 2K X 8. The socket at U19 may contain EPROM or ROM in sizes of 2K X 8 or 4K X 8. Tables 5-2 and 5-3 show the necessary jumpers for each case.

Table 5-2. Jumper Options for U19. U20

1K RAM	E4-E5, E8-E9	MK4808
2K RAM	E5-E6, E8-E9	MK4816
1K ROM/ EPROM	E4-E5, E7-E8	Intel 2308/2758
2K ROM/ EPROM	E5-E6, E7-E8	Intel 2316/2716

Table 5-3. Jumper Options for U20. U19

2K ROM/ EPROM	E1-E3	Intel 2316/2716
4K ROM/ EPROM	E2-E3	Intel 2332/2732

X9 contains the jumpers for the block sizes and the lower block boundaries.* Block sizes may be 1K, 2K, or 4K**. The lower block boundary may be set at 0000, 4000H, 8000H, or C000H. Tables 5-4 and 5-5 give all of the options for X9.

Table 5-4. Block Size Jumpers.

BLOCK SIZE	NECESSARY JUMPERS
1K	X9-1 TO X9-16, X9-2 TO X9-14, X9-3 TO X9-13
2K	X9-2 TO X9-15, X9-3 TO X9-14, X9-4 TO X9-13
4K	X9-3 TO X9-15, X9-4 TO X9-14, X9-5 TO X9-13

*It should be noted that the MPC-4 will auto start only at 0000H.

**NOTE - Make block size equal to largest memory device. e.g. If U19 has a 2K RAM and U20 has a 4K ROM, the 4K block size should be selected.

Table 5-5. Boundary Jumpers.

BOUNDARY BEGINNING AT:	NECESSARY JUMPERS
0000H	X9-5 TO X9-12, X9-6 TO X9-11
4000H	X9-6 TO X9-11, X9-7 TO X9-10
8000H	X9-5 TO X9-12, X9-8 TO X9-9
C000H	X9-7 TO X9-10, X9-8 TO X9-9

The MPC-4 has the jumpers etched to configure the board as follows:

- A. 2K blocks beginning at 0000H.
- B. U19 - 2K EPROM
- C. U20 - 1K RAM

The user may need to cut some or all of these etched jumpers to reconfigure.

5.4 CTC OPTIONS (X11)

These pins represent points of access to the channel inputs and outputs on the CTC.

5.5 SERIAL I/O OPTIONS (X1-X8)

These jumpers may be used to convert to synchronous serial interface. It should be noted that if this is desired the Z80 DARTS would have to be replaced with Z80 SIO's.

The board is etch jumpered so that the transmit and receive clocks come directly from the on-board baud rate generators.

5.6 INTERRUPT OPTIONS (E10-E14)

These jumpers provide access to the interrupt chain of the S-100 xbus. It must be noted that if mode 2 interrupts are to be used the vector must be written into the CTC on the SBC.

SECTION VI
CONSTRUCTION

6.1 INTRODUCTION

The MPC-4 kit is intended for those who have had some prior experience with kit building and digital electronics. If you do not fall into this category, it is highly recommended that you find an experienced person to help you in assembly and check out of the board.

Appendix B shows the parts list for the MPC-4. Double check all the parts against the parts list.

6.2 ASSEMBLY PROCEDURE

1. Install and solder the resistors as follows:

- A. R2 22 Ohm 1/4W 5% (RED, RED, BLACK, GOLD)
- B. R3 220 Ohm 1/4W 5% (RED, RED, BROWN, GOLD)
- C. R9,10 820 Ohm 1/4W 5% (GRAY, RED, BROWN, GOLD)
- D. R8 1.2K 1/4W 5% (BROWN, RED, RED, GOLD)
- E. R1,4,5,6 2.4K 1/4W 5% (RED, YELLOW, RED, GOLD)
- F. R7,11,12 4.7K 1/4W 5% (YELLOW, VIOLET, RED, GOLD)
- G. RP1 RESISTOR PACK 4.7K 6 PIN SIP

NOTE: Pin 1 of the Sip is designated by a notch or dot on the end of the package.

2. Install and solder IC sockets as follows:

- 40 PIN - U17, U18, U27
- 28 PIN - U16
- 24 PIN - U19, U20
- 20 PIN - U36, U37, U38, U40, U41
- 18 PIN - U13, U14, U23, U24
- 16 PIN - U2, U25, U28, U29, U34, U39
- 14 PIN - U1, U3-U12, U15, U21, U22, U26, U30-U33, U35, U42, U43

3. Install and solder Berg headers as follows with long pin up:

- X1-X4 (2X11)
- X10 (2X4)

4. Install and solder the voltage regulators as follows:

- A. VR1 (LM323 5V 3A) is installed with the 6103-B heat sink. The metal screws and nuts are to be used.
- B. VR2 (LM7812 12V 1A) is installed with one of the mylar insulators between it and the PC board, a nylon nut and screw are used.

C. VR3 (LM7912 -12V 1A) is installed the same as VR2.

5. Install and solder capacitors as follows:

- A. C3 10 pf
- B. C1 33 pf 50V
- C. C2 .01 uf
- D. C4-C16,21 .1 uf 50V
- E. C17-20 10 uf 25V Tantalum (Note polarity)

6. Install and solder transistor Q1 according to silkscreen.

7. Install and solder crystal Y1.

8. Double check all solder connections for cold solder joints, unsoldered connections, or shorted connections.

6.3 VOLTAGE CHECK

1. The following check should be performed before any IC's are installed.

2. Plug board into an S-100 bus and apply power. Check the following voltages at the designated points:

- +5V Check at positive (+) terminal of C17
- +12V Check right hand pin of VR2
- 12V Check right hand pin of VR3

Each voltage should be within 10% of the specified voltage. If not, refer to the power supply section of the troubleshooting guide.

3. Once the supply voltages have been verified the IC's may be inserted.

- | | |
|-------------------|-------------------------------|
| A. U27 | Z80 CPU |
| B. U17,18 | Z80 DART |
| C. U16 | Z80 CTC |
| D. U19 | MPC-4 PROM-2716 |
| E. U13,14 | 2114 1KX4 RAM |
| F. U23,24 | COM 5016T BAUD RATE GENERATOR |
| G. U3,7,8,12 | 75188 (1488) DRIVER |
| H. U4,5,6,9,10,11 | 75189 (1489) RECEIVER |
| I. U30,33 | 74LS00 |
| J. U21,22 | 74LS02 |
| K. U1 | 74LS04 |
| L. U15 | 7406 |
| M. U31 | 74LS14 |
| N. U35 | 74LS20 |
| O. U26 | 74LS27 |
| P. U32 | 74LS32 |
| Q. U42,43 | 74LS74 |

R.	U25,34	74LS85
S.	U28	74LS138
T.	U2	74LS139
U.	U29	74LS158
V.	U39	74LS174
W.	U36,37,38,40	74LS244
X.	U41	74LS374

4. This completes the construction phase of your MPC-4. It is now time to move to the testing section.

NOTE: Do not be alarmed because U20 is empty and is not called out on the parts list. U20 is a socket provided for additional RAM if desired.

5. Install two PCB ejectors with mounting pins (see Assembly Drawing).

SECTION VII

TESTING

7.1 INTRODUCTION

This section outlines the procedures for verifying the operation of the MPC-4.

7.2 SUPPLY VOLTAGE

With all IC's now inserted, check voltages again at the points given in the construction section. If these voltages are correct move on to section 7.3. Otherwise, refer to the Power Supply section of the Troubleshooting Guide.

7.3 BOARD SELECT

The MPC-4 comes selected as board 7. With the board installed in a system, use the monitor to output a 07H to port 0FFH. The monitor prompt should return. If so, move on to section 7.4. Otherwise, refer to the Board Select section of the Troubleshooting Guide.

7.4 MEMORY TEST

Dump the contents of the memory from 0 to 0FFH. Spot check to see if it compares with the data in the listing provided in Appendix D. If the data does not compare refer to the Memory Test section of the Troubleshooting Guide.

Test the memory from 1000H to 13FFH. If it passes, move on to section 7.5. If not, refer to the Memory test section of the Troubleshooting Guide.

7.5 SETUP

Fill memory from 1000H to 13FFH with zeros. Dump to see that the memory is indeed filled with zeros. Reset the system.

Output a 07H to port 0FFH. Dump memory starting at 1000H. The contents should appear as shown in figures 7-1 thru 7-4 except for the circled locations. If not, refer to the Setup section of the Troubleshooting Guide.

Locations 13E0-13FF are reserved for the stack. These locations will not necessarily be the same as shown.

Place 0EEH in locations 13ADH and 13AEH. These are the baud rate

constants. Make a note of the value of the data in location 13AAH. This is the second counter. This location should contain some number other than zero. If not, see Setup section in Troubleshooting Guide.

1000	40	10	50	00	00	00	00	00	80	11	80	00	00	00	00	00	@.P.....
1010	90	10	50	00	00	00	02	00	00	12	80	00	00	00	02	00	..P.....
1020	E0	10	50	00	00	00	04	00	80	12	80	00	00	00	06	00	0.P.....
1030	30	11	50	00	00	00	06	00	00	13	80	00	00	00	06	00
1040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figure 7-1. MPC-4 Random Access Memory.

1100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
11A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
11B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
11C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
11D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
11E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
11F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figure 7-2. MPC-4 Random Access Memory.

1200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1210	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1220	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1230	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1240	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1250	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1260	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1270	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1280	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1290	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
12A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
12B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
12C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
12D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
12E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
12F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figure 7-3. MPC-4 Random Access Memory.

1300	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1310	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1320	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1330	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1340	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1350	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1360	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1370	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1380	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1390	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
13A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
13B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	A6	00
13C0	C5	02	C5	02	E9	02	0F	02	C5	02	C5	02	DE	02	06	03	E.E.i..E.E.
13D0	C5	02	C5	02	D9	02	FD	02	C5	02	C5	02	C8	02	F4	02	E.E.S..E.E.H.t.
13E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
13F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00;S.T.

Figure 7-4. MPC-4 Random Access Memory.

Place a 0FFH in location 100DH. Output a 00 to port 0FFH. Wait 2-3 seconds. Output a 07H to port 0FFH.

Check location 13AAH. The previously noted value should have been incremented by the numbers of seconds waited in the previous step. If not, see the Setup section of the Troubleshooting Guide.

Using an oscilloscope, look at the following four points. At each point there should be a square wave with an amplitude of at least 4.5v and with a frequency of about 156 KHz.

- U23 - 3
- U23 - 17
- U24 - 3
- U24 - 17

If everything is fine to this point you are ready to boot up COSMOS. Reset and boot up.

SECTION VIII

TROUBLESHOOTING GUIDE

8.1 INTRODUCTION

The Troubleshooting Guide is provided so that the user may find and correct problems which may occur without having to return the MPC-4 to SD SYSTEMS for repair.

Before referring to any of the information in this section reexamine the complete board for etch shorts, solder shorts, cold solder joints, etc. Make sure the capacitors are installed with the correct polarities. Check all IC's to see that no pins are bent under.

After all of these checks are made you are ready to proceed.

8.2 POWER SUPPLY

1. If the voltages were correct upon construction and the IC's have been inserted, move on to section 8.4.
2. Check the input voltage to the regulator for the incorrect voltage. These are the points to check and the expected values for each regulator. Points are on component side of board:

REGULATOR	POINT OF MEASUREMENT	EXPECTED VOLTAGE
VR1	+TERMINAL OF C18	+8V - +10V
VR2	LEFT TERMINAL OF VR2	+16V - +20V
VR3	CENTER TERMINAL OF VR3	-16V - -20V

The case of VR1 can be used for the ground point.

3. Remove the regulator for the bad supply. If the input voltage was incorrect in the previous step, recheck it. If it is still not correct, the problem is either with the mother board power supply or the mother board itself. If it was not correct and is now, install a new regulator.

If the input voltage was correct check for a short between ground and the supply bus. If there is no short, replace the regulator with a new one. If there is a short it must be found and removed.

4. If the IC's have been inserted and one of the supplies is incorrect then one of the IC's is loading the regulator down. Below are listed the IC's and the supply they would affect.

+5	+12	-12
All 74 series parts	COM 5016 T-5	1488 (75188)
1489 (75189)	1488 (75188)	
COM 5016 T-5		
Z80 DART		
Z80 CPU		
Z80 CTC		
2716 EPROM		
2114 RAM		

5. There are two ways to find the bad IC.

- (1) Remove all associated IC's and re-insert them one by one until the supply voltage is wrong again. Replace this last IC.
- (2) Remove the associated IC's one by one until the supply returns to its correct value. Replace the last IC removed with a new one.

NOTE: No IC should be removed while power is on.

8.3 BOARD SELECT

1. If the system hangs up after a 07H is output to port 0FFH it means a BUSRQ has been generated to the CPU on the MPC-4, but no BUSAK has been returned to remove the wait state generated by BUSRQ.
2. Check pin 25 of U27 to see that BUSRQ is going low when the board is selected. If so check BUSAK to see if it ever goes low in response to the BUSRQ. If there is no BUSRQ, check pin 6 of U25 for a compare. When the board number is output to port 0FFH, pin 6 will go high and generate BUSRQ. If a BUSAK is generated the problem is between U33 pin 3 and pin 72 of the S-100 bus.
3. If there was no BUSAK in response to BUSRQ, check pin 6 of U27 with the scope to see if there is a system clock. There should be a 2.4576 MHz square wave with an amplitude of 4.5-5V. If this clock is not present or is of insufficient amplitude it must be corrected for the board to function. If the clock is allright and no BUSAK followed BUSRQ, replace the Z80 CPU.

8.4 MEMORY TEST

1. Load the following program at 0FC00H and execute:

```
3A 00 00 LD A, (0000H)
C3 00 FC JP 0FC00H
```

Use an oscilloscope to be sure that the signals on U19 are similar to those shown in Figure 8.1. Be sure you have output a 07H to port 0FFH. Pay particular attention to the chip select (pin 18). If there is no chip select the difficulty may lie with the address decoder (U28). If there is no output enable (pin 20), check the multiplexer (U29).

2. If the PROM reads okay but the RAM test fails, load the following program at 0FC00H and execute:

```
3e 55      LD  A,55H
32 00 10   LD  (1000H), A
3A 00 10   LD  A,(1000H)
C3 00      FC
```

Use an oscilloscope to see that the signals on U13 and U14 are similar to those shown in Figure 8.2. Pay particular attention to OE (pin 8) and WE (pin 10). If all of the signals are okay, the RAM must be defective. If not, check the multiplexer and the address decoder.

8.5 SETUP

1. At this time the memory has been tested and is functional. If, after the reset, none of the data matches what was expected there is a communications problem between the on-board Z80 and the memory. Observation of signal levels with scope should prove helpful. Signals should rise to at least 4.5V and drop to at least .8V.
2. If most of the data is correct after a reset but there is one byte incorrect every third line or so this points to a problem with interrupts. Either there is a short or a bad connection on the interrupt line or one of the interrupting devices is generating spurious interrupts.
3. If 13AAH contains a zero it means that the CTC is not interrupting. Check the IEI signal (pin 13) of the CTC (U16). It should be a high (1) level. If it is not then U17 or U18 have requested an interrupt incorrectly. If it is high, check IEO (pin 11). If this is low, it means the CTC has requested an interrupt but it has not been serviced. If IEO is high, it means that the CTC interrupt has been serviced or the CTC never interrupted. The data and address lines to the CTC should be checked both with a scope and visually.
4. If the 156K Hz waveforms are not present, check the data lines to the baud rate generators (U23,U24). If these appear okay, repeat the setup procedure and monitor the strobe lines of the baud rate generators. There should be a pulse at each one after the 0FFH is loaded in location 100DH and a 00 is output to port 0FFH. If not, check U2 and U22.

REVISION		DATE	APP
LTR	DESCRIPTION		

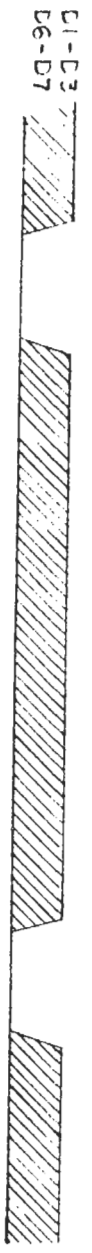
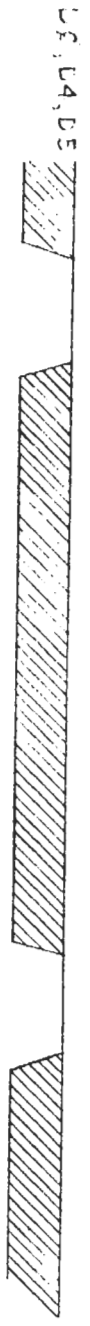
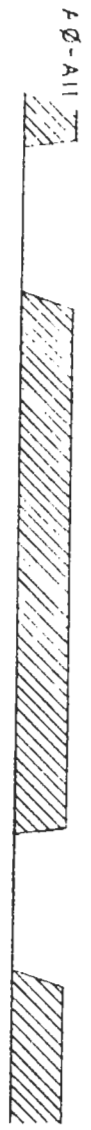
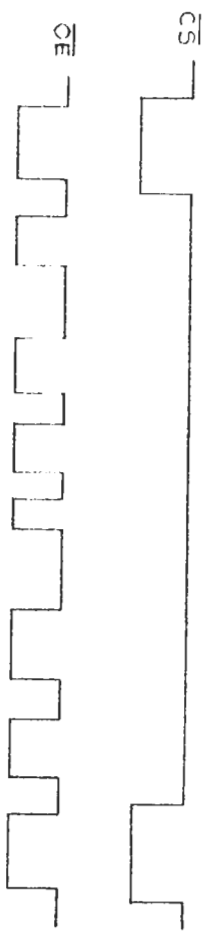


FIG. 8.1
READ CYCLE FOR MPC-4 PROM

TOLERANCES		DRAWN BY D.O.		SD SYSTEMS GARLAND TEXAS
MATERIAL		CHECKED BY J.F.L.		
FINISH		DESIGN ENGR 6-9-72		
		PROD ENGR		
NEXT ASSY USED ON		APPROVED		SIZE CODE IDENT
APPLICATION				DRAWING NO. 0100300
				REV
				SHEET 1 OF 1

READ CYCLE
FOR MPC-4 PROM

FIG. 8.1

REVISION		DESCRIPTION	DATE	APP
LTR				

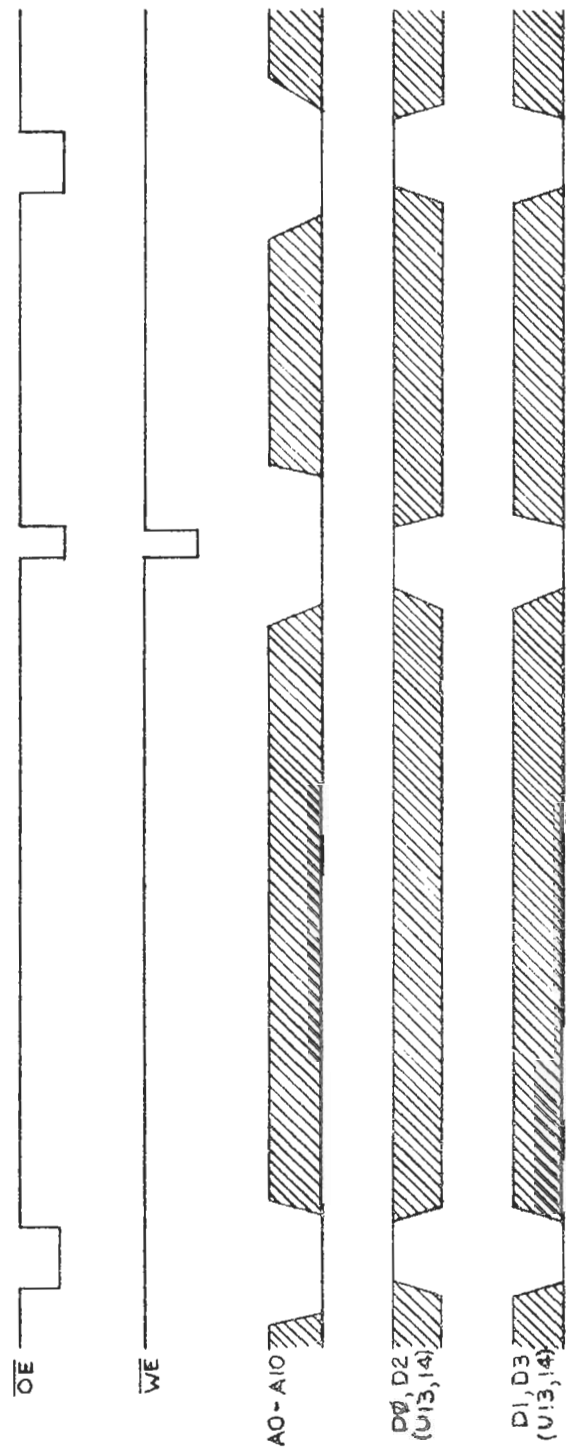
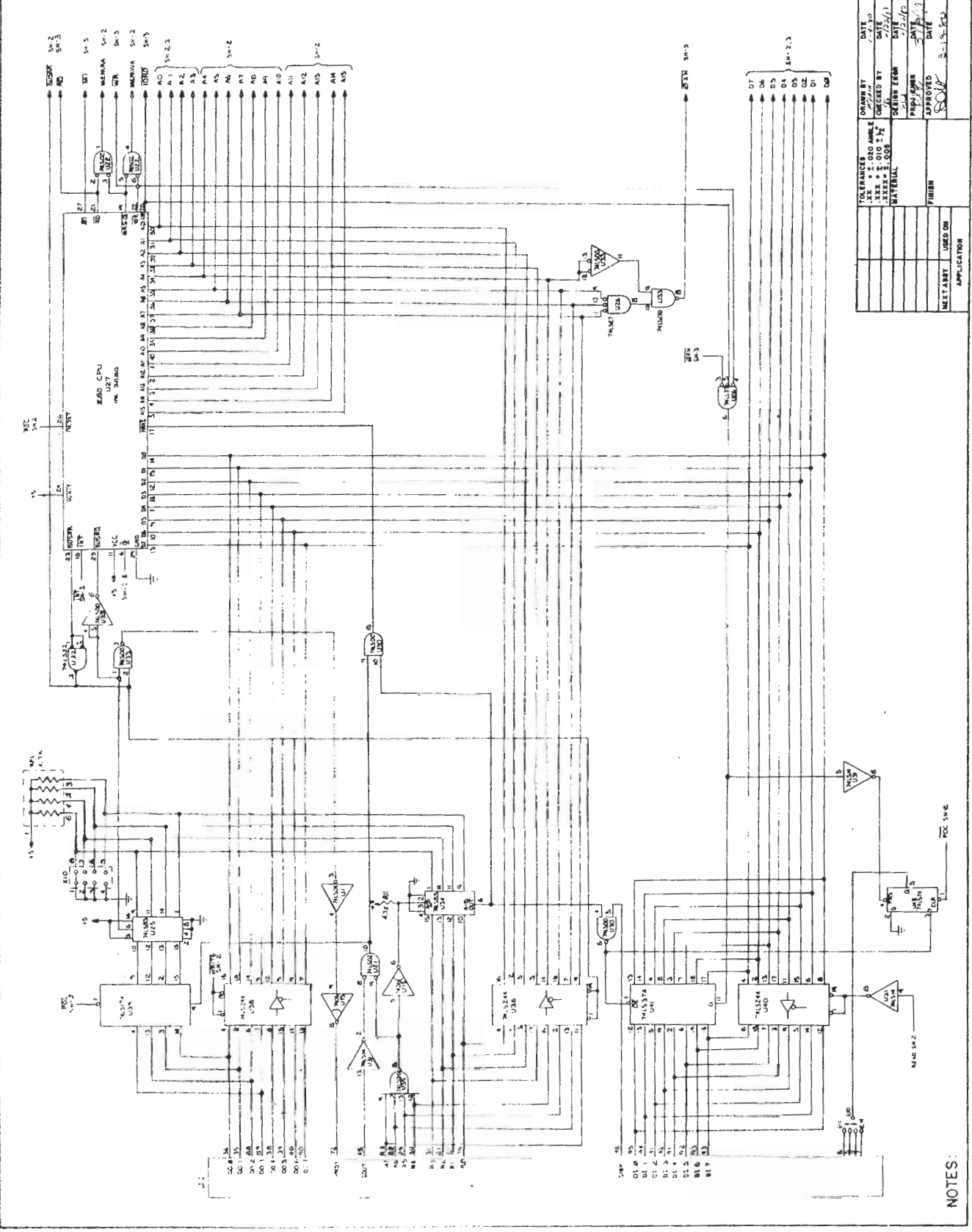


FIG. 8.2
READ/WRITE CYCLE FOR MPC-4 RAM

DRAWN BY D.O.		SD SYSTEMS GARLAND, TEXAS	
CHECKED BY S.C. 6-4-8		READ/WRITE CYCLE FOR MPC-4 RAM	
DESIGN ENGR S.C.		FIG. 8.2	
PROJ ENGR S.C.		SIZE CODE IDENTIFY DRAWING NO.	
APPROVED S.C.		REV	
TOLERANCES		SCALE	
MATERIAL		8	
FINISH		0100299	
HEAT TREATMENT		SHEET 1 OF 1	
APPLICATION			

APPENDIX A
MPC-4 SCHEMATIC DIAGRAM

LTR	DESCRIPTION	DATE	APP
A	PER ECN 132	11/15/77	CCP

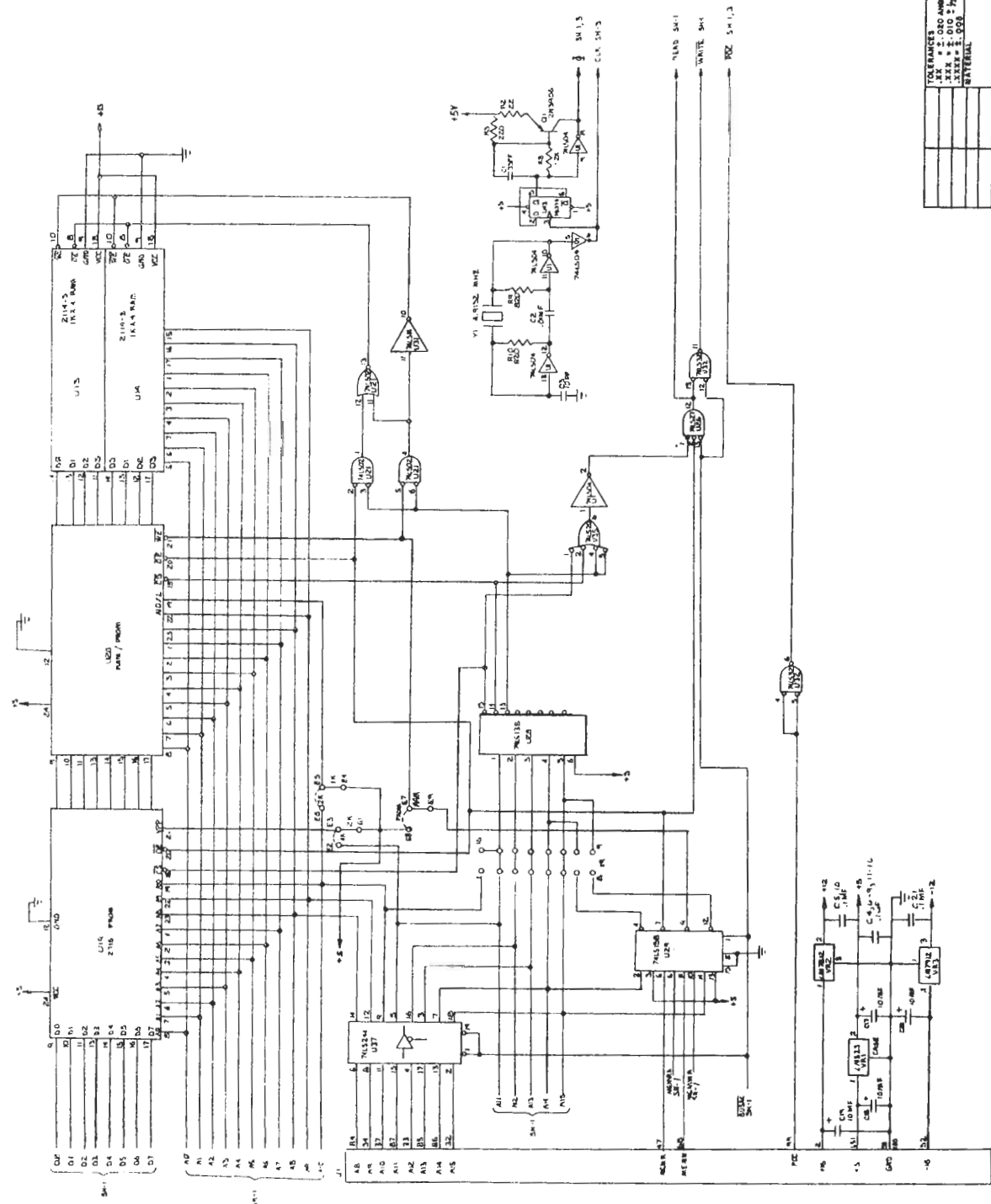


NOTES:

TOLERANCES	DRAWN BY	DATE	REV
XXX ± 0.010 ANGLE	CCP	11/15/77	A
XXX ± 0.010 ± 1/2	CHECKED BY	DATE	
MATERIAL CODE	DESIGN TEAM	DATE	
	PREPARED BY	DATE	
	APPROVED	DATE	
	TURIN		
	USED ON		
	APPLICATION		

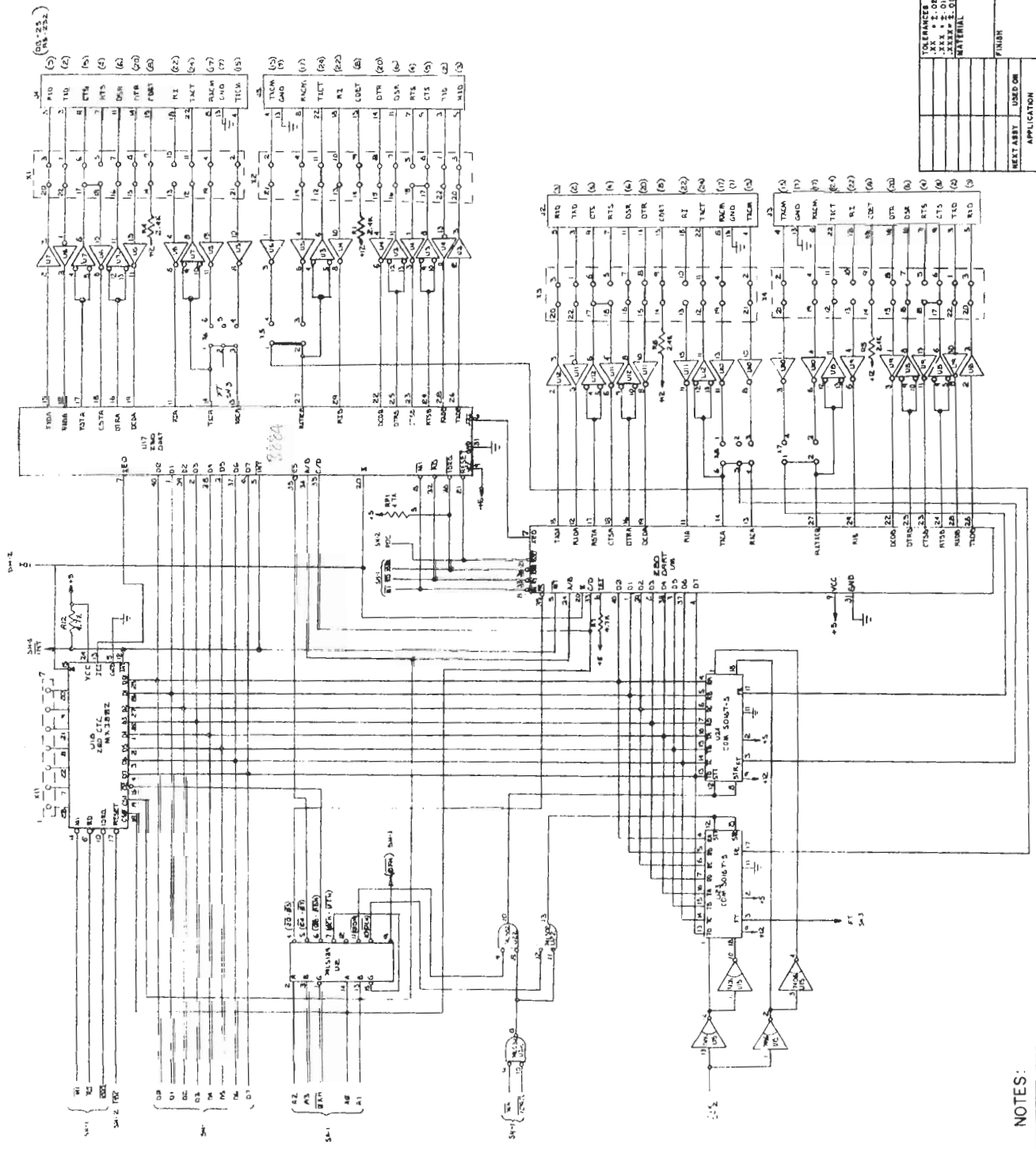
SD SYSTEMS	
GARLAND, TEXAS	
MFC-4 SCHEMATIC	
SIZE	CODE
D	108T
SCALE	DO NOT SCALE DRAWING
	SHEET / DP

LTR	DESCRIPTION	DATE	APP



NOTES:

DESIGNER	DATE	SCALE	DO NOT SCALE DRAWING	SHEET	OF
SD SYSTEMS					
AWP-C-4					
SIZE CODE IDENT	DWGING NO.				
D	0100.157				
REV					
A					



TOLERANCES XXX ± 0.00 ANGLE XXXX ± 0.005 XXXXX ± 0.005 MATERIAL		DRAWN BY CHECKED BY DESIGN ENGR DATE	DATE DATE DATE
FINISH		APPROVED	DATE
SIZE	CODE	ISERT	DRAWING NO.
D			0100157
SCALE	DO NOT SCALE DRAWING		SHEET 3 OF 3
APPLICATION		REV	
		A	

SD SYSTEMS
DESIGNING - TEXAS
APC-4


NOTES:

APPENDIX B
MPC-4 BILL OF MATERIALS

SD Systems

P.O. Box 28810 • Dallas, Texas 75228 214-271-4887

BILL OF MATERIALS

Title:		PL No.		Rev.	
MPC - 4		0100156		A	
Date Released:		Approved:		Sheet	
3-19-80		EHL 3/13/80 		2 of 4	
Item No.	Qty	SD-P/N	Description	Unit Cost	Extension
1	1	7000020	PC BOARD MPC-4		
2	3	7060009	SOCKET 40 PIN DIP		
3	1	7060008	SOCKET 28 PIN DIP		
4	2	7060007	SOCKET 24 PIN DIP		
5	5	7060005	SOCKET 20 PIN DIP		
6	4	7060004	SOCKET 18 PIN DIP		
7	6	7060003	SOCKET 16 PIN DIP		
8	22	7060002	SOCKET 14 PIN DIP		
9	1	7010318	IC MK3880 Z80 CPU U27		
10	2	7010394	IC Z80 DART U17, 18		
11	1	7010320	IC MK3882 Z80 CTC U16		
12	1	7010395	IC 2716 PROM (CONTROL) U19		
13	2	7010396	IC 2114-3 RAM U13, 14		
14	2	7010397	IC COM 5016T-5 BRG. U23, 24		
15	4	7010332	IC 75188 U3, 7, 8, 12		
16	6	7010333	IC 75189 U4, 5, 6, 9, 10, 11		
17	2	7010160	IC 74LS00 U30, 33		
18	2	7010162	IC 74LS02 U21, 22		
19	1	7010164	IC 74LS04 U1		
20	1	7010007	IC 7406 U15		
21	1	7010172	IC 74LS14 U31		
22	1	7010174	IC 74LS20 U35		
23	1	7010178	IC 74LS27 U26		
24	1	7010181	IC 74LS32 U32		

SD Systems

P.O. Box 28810 • Dallas, Texas 75228 214-271-4667

BILL OF MATERIALS

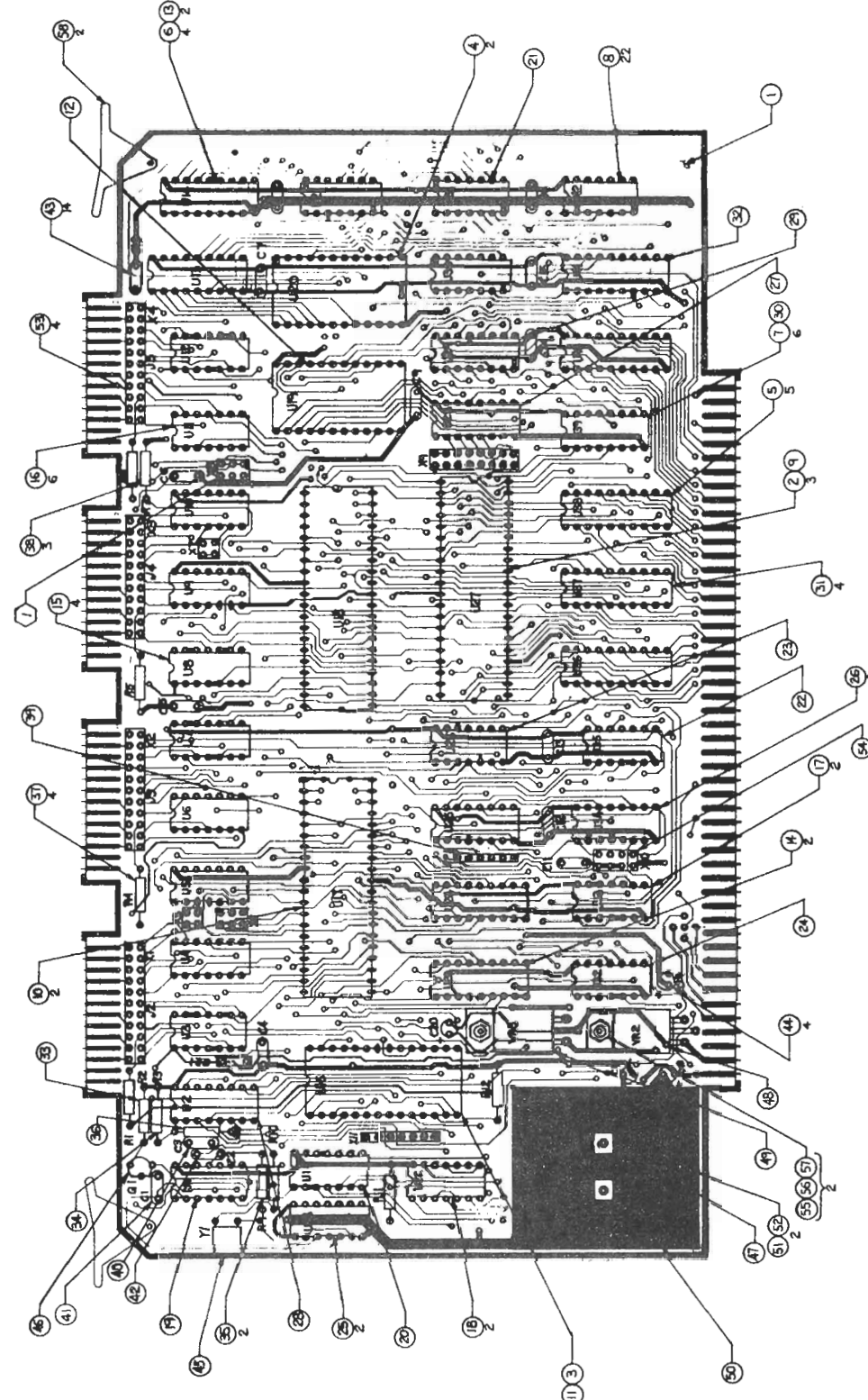
Title: MPC 4	PL No. 0100156	Rev. A
--------------	----------------	--------

Date Released:	Approved:	Sheet 3 of 4
----------------	-----------	--------------

Item No	Qty	SD-P/N	Description	Unit Cost	Extension
25	2	7010195	IC 74LS74 U42 43		
26	2	7010200	IC 74LS85 U25, 34		
27	1	7010219	IC 74LS138 U28		
28	1	7010220	IC 74LS139 U2		
29	1	7010229	IC 74LS158 U29		
30	1	7010241	IC 74LS174 U39		
31	4	7010264	IC 74LS244 U36, 37, 38, 40		
32	1	7010305	IC 74LS374 U41		
33	1	7020033	RESISTOR 22Ω 1/4 W 5% R2		
34	1	7020057	RESISTOR 220Ω 1/4 W 5% R3		
35	2	7020071	RESISTOR 820Ω 1/4 W 5% R9,10		
36	1	7020075	RESISTOR 1.2K 1/4 W 5% R8		
37	4	7020082	RESISTOR 2.4K 1/4 W 5% R1, 4, 5, 6		
38	3	7020089	RESISTOR 4.7K 1/4 W 5% R7, 11, 12		
39	1	7010348	RESISTOR SIP 4.7K 6 PIN RPI		
40	1	7030001	CAPACTOR 10PF C3		
41	1	7030003	CAPACTOR 33PF C1		
42	1	7030008	CAPACTOR .01MF C2		
43	14	7030007	CAPACTOR .1MF C4-16, 21		
44	4	7030009	CAPACTOR 10MF C17-20		
45	1	7080003	CRYSTAL 4.9152 MHZ Y1		
46	1	7040006	TRANSISTOR 2N3906 Q1		
47	1	760002	REGULATOR LM323 5V 3A VRI		
48	1	760003	REGULATOR LM7812 12V 1A VR2		

APPENDIX C
ASSEMBLY DRAWING

REVISIONS		DATE	APP
LTR	DESCRIPTION		
A	PER TECN 150		



SD SYSTEMS SARASOTA, FLORIDA		DATE	
MPC-4		DATE	
ASSY DRAWING		DATE	
DRAWN BY		DATE	
CHECKED BY		DATE	
DESIGNED BY		DATE	
MATERIAL		DATE	
PROJ. NO.		DATE	
APPROVED		DATE	
/MESH		DATE	
USED ON		DATE	
APPLICATION		DATE	
SIZE CODE IDENT	DRAWING NO.	REV	
D	O100156	A	
SCALE			SHEET / OF 4

1 BERG HEADERS NOT INSTALLED AT X5-X9 AND X11
NOTES:

APPENDIX D
MPC CONTROL SOFTWARE LISTING

ADDR CODE

STMT SOURCE STATEMENT

```

0001 ;
0003     NAME MPC
0004 ;
0005 ;           VERSION 1.0           2/19/80 ERS
0006 ;           VERSION 1.1           4/22/80 ERS
0007 ;           VERSION 1.2           7/28/80 ERS
0008 ;
0009 ;           VERSION 1.1 CHECKS THE OUTPUT DATA FOR A NULL.  IF ONE
0010 ;           IS FOUND IT IS CHANGED TO AN 'FF'.
0011 ;           ALSO IF A SPECIAL CONDITION INTERRUPT IS RECEIVED, THE
0012 ;           STATUS IS STORED IN THE ERROR STATUS BYTE IN THE USER'
0013 ;           CONTROL MEMORY.  ERS
0014 ;
0015 ;           VERSION 1.2 INITIALIZES THE DATA IN BYTE OF EACH
0016 ;           USER TO A 0FFH.  ERS
0017 ;
0018 ;           MEMORY DESIGNATIONS
0019 ;
0020 ;
>1000 0021 MPCIM EQU 1000H ; BEGINNING OF MEMORY
>0050 0022 IBUFF EQU 80 ; SIZE OF INPUT BUFFER
>0080 0023 OBUFF EQU 128 ; SIZE OF OUTPUT BUFFER
>1002 0024 ISIZAD EQU MPCIM+2 ; ADDR OF INPUT BUFFER SIZE
>100A 0025 OSIZAD EQU MPCIM+0AH ; " " OUTPUT " "
>1006 0026 PORT1 EQU MPCIM+6 ; USER 1 PORT ADDRESS
>0000 0027 PTNUM1 EQU 0 ; USER 1 DATA PORT
>0004 0028 PTSTAT EQU 4 ; MASK FOR TRANSMIT
>1040 0029 BUFAD1 EQU MPCIM+40H ; USER 1 DATA BUFFER
>13B8 0030 CTCV0 EQU MPCIM+3B8H ; CTC CH 0 INT VECTOR
>13BA 0031 CTCV1 EQU CTCV0+2 ; CTC CH 1 INT VECTOR
>13BC 0032 CTCV2 EQU CTCV1+2 ; CTC CH 2 INT VECTOR
>13BE 0033 CTCV3 EQU CTCV2+2 ; CTC CH 3 INT VECTOR
>13BE 0034 RTINT EQU CTCV3 ; REAL TIME INTERRUPT
>13A4 0035 TIMER EQU MPCIM+3A4H ; COUNT TO 75
>13A5 0036 YEAR EQU TIMER+1 ; YEAR COUNTER
>13A6 0037 MONTH EQU YEAR+1 ; MONTH COUNTER
>13A7 0038 DAY EQU MONTH+1 ; DAY COUNTER
>13A8 0039 HOUR EQU DAY+1 ; HOUR COUNTER
>13A9 0040 MINUTE EQU HOUR+1 ; MINUTE COUNTER
>13AA 0041 SECOND EQU MINUTE+1 ; SECOND COUNTER
>13C0 0042 VTABL EQU MPCIM+3C0H ; VECTOR TABLE
>1000 0043 MPCIM1 EQU MPCIM ; INPUT MEMORY INDEX USER 1
>1010 0044 MPCIM2 EQU MPCIM1+10H ; " " " USER 2
>1020 0045 MPCIM3 EQU MPCIM2+10H ; " " " USER 3
>1030 0046 MPCIM4 EQU MPCIM3+10H ; " " " USER 4
>1008 0047 MPCOM1 EQU MPCIM1+08H ; OUTPUT MEMORY INDEX USER 1
>1018 0048 MPCOM2 EQU MPCOM1+10H ; " " " USER 2
>1028 0049 MPCOM3 EQU MPCOM2+10H ; " " " USER 3
>1038 0050 MPCOM4 EQU MPCOM3+10H ; " " " USER 4
>13AB 0051 OUTFLG EQU MPCIM+3ABH ; FLAG FOR STATUS OUTPUT
>13AC 0052 UBYTE EQU MPCIM+3ACH ; USER STATUS FROM SYSTEM
>13AD 0053 BR1 EQU MPCIM+3ADH ; BAUD RATE CONSTANT USR 1,2
>13AE 0054 BR2 EQU MPCIM+3AEH ; " " " " 3,4
>000D 0055 BAUD1 EQU 0DH ; " " PORT " 1,2
>000E 0056 BAUD2 EQU 0EH ; " " " " 3,4
>13AF 0057 PTBYTE EQU MPCIM+3AFH ; PORT STATUS BYTE
>1400 0058 STACK EQU MPCIM+400H ; TOP OF STACK
>0002 0059 SIZE EQU 2 ; BUFFER SIZE OFFSET

```

ADDR	CODE	STMT	SOURCE	STATEMENT		
>0003		0060	IOFF	EQU	3	; INSERT BUFFER OFFSET
>0004		0061	EOFF	EQU	4	; EXTRACT BUFFER OFFSET
>0005		0062	DATA	EQU	5	; DATA BYTE OFFSET
>0006		0063	PORT	EQU	6	; CMD/STAT PORT OFFSET
>0007		0064	ERROR	EQU	7	; ERROR STATUS BYTE
>0008		0065	CTC0	EQU	08H	; PORT FOR CH 0 OF CTC
>0009		0066	CTC1	EQU	09H	; PORT FOR CH 1 OF CTC
>000A		0067	CTC2	EQU	0AH	; PORT FOR CH 2 OF CTC
>000B		0068	CTC3	EQU	0BH	; PORT FOR CH 3 OF CTC
>000F		0069	STATUS	EQU	0FH	; STATUS PORT
		0070				;

```

MPC          MPC CONTROL SOFTWARE          SD SYSTEMS Z80 ASSEMBLER PAGE 0003
ADDR CODE    STMT SOURCE STATEMENT

0072 ;                      SYSTEM INITIALIZATION
0073 ;                      THIS PORTION OF THE PROGRAM TAKES CARE OF INITIALIZI
0074 ;                      THE INDEX POINTER MEMORY ON THE MPC-4.  INITIALLY, THE
0075 ;                      ERRUPTS FROM THE DARTS ARE DISABLED.  THEY REMAIN SO U
0076 ;                      GIVEN A COMMAND FROM THE SYSTEM.  EACH USER REQUIRES A
0077 ;                      SEPARATE COMMAND.
0078 ;

0000 310014 0079 INIT:  LD      SP,STACK
0003 210010 0080      LD      HL,MPCIM1
0006 AF      0081      XOR     A                      ; ZERO STATUS MEMORY
0007 0640    0082      LD      B,40H                    ; OF WHICH 40H LOCATIONS
0009 77      0083 FILL:  LD      (HL),A                      ; EXIST
000A 23      0084      INC     HL
000B 10FC    0085      DJNZ   FILL
000D 32AF13 0086      LD      (PTBYTE),A          ; ZERO STATUS BYTE
0010 3E13    0087      LD      A,13H                    ; SET UP I REGISTER FOR
0012 ED47    0088      LD      I,A                      ; INTERRUPT VECTOR
0014 111000 0089      LD      DE,10H                   ; PUT BUFFER SIZES IN
0017 0604    0090      LD      B,4                      ; EACH OF THE INPUT
0019 210210 0091      LD      HL,ISIZAD                ; POINTER LOCATIONS. THEY
001C 3E50    0092      LD      A,IBUFF                    ; ARE LOCATED EVERY 10H
001E 77      0093 ISZFIL: LD      (HL),A          ; POSITIONS. THERE ARE 4
001F 19      0094      ADD     HL,DE                      ; OF THEM.
0020 10FC    0095      DJNZ   ISZFIL
0022 0604    0096      LD      B,4                      ; REPEAT FOR OUTPUT
0024 210A10 0097      LD      HL,OSIZAD                ; BUFFER POINTER LOCATIONS
0027 3E80    0098      LD      A,OBUFF                    ;
002A 19      0099 OSZFIL: LD      (HL),A          ;
002B 10FC    0100      ADD     HL,DE                      ;
002D 0604    0101      DJNZ   OSZFIL
002F 210610 0102      LD      B,4                      ; WRITE PORT NUMBERS TO
0032 3E00    0103      LD      HL,PORT1                    ; PORT POINTER LOCATIONS
0034 110800 0104      LD      A,PTNUM1                  ; THERE ARE 8 OF THESE
0037 77      0105      LD      DE,8                      ; TWO FOR EACH USER. ONE
0038 19      0106 PTLD:  LD      (HL),A          ; OUTPUT AND ONE INPUT.
0039 77      0107      ADD     HL,DE                      ; THE OUTPUT AND INPUT FOR
003A 19      0108      LD      (HL),A          ; THE SAME USER ARE EQUAL
003B C602    0109      ADD     HL,DE
003D 10F8    0110      ADD     A,2
003F 115000 0111      DJNZ   PTLD
0042 0604    0112      LD      DE,50H                    ; LOAD THE MEMORY POINTERS
0044 214010 0113      LD      B,4                      ; FOR EACH USER. FIRST THE
0047 220010 0114      LD      HL,BUFAD1                  ; BEGINNING LOCATION IN EACH
004A 19      0115      ADD     HL,DE                      ; INPUT BUFFER IS LOADED.
004B 221010 0116      LD      (MPCIM2),HL          ; THERE ARE 4 OF THEM.
004E 19      0117      ADD     HL,DE                      ; USER 2 INPUT BUFFER
004F 222010 0118      LD      (MPCIM3),HL          ; USER 3 INPUT BUFFER
0052 19      0119      ADD     HL,DE
0053 223010 0120      LD      (MPCIM4),HL          ; USER 4 INPUT BUFFER
0056 19      0121      ADD     HL,DE                      ; BEGINNING OF OUTPUT
0057 220810 0122      LD      (MPCOM1),HL          ; BUFFERS; USER 1
005E 118000 0123      LD      DE,80H
0061 19      0124      ADD     HL,DE
0062 1849    0125      LD      (MPCOM2),HL          ; USER 2 OUTPUT BUFFER
0063 19      0126      ADD     HL,DE
0064 1849    0127      JR      AROUND                    ; SKIRT INTERRUPT ROUTINE
0065 1849    0128

```

```

0130 ;
0131 ;
0132 ;
0133 ;
>0066 0134 ORG 66H
'0066 08 0135 EX AF,AF'
'0067 D9 0136 EXX ; SAVE REGISTERS
'0068 111000 0137 LD DE,10H ; GET USER BYTE
'006B 3AAC13 0138 LD A,(UBYTE) ; GET USER BYTE
'006E 2F 0139 CPL ; SAVE IT
'006F 47 0140 LD B,A ; SAVE IT
'0070 C5 0141 PUSH BC
'0071 2F 0142 CPL
'0072 FE09 0143 CP 9 ; SEE IF INPUT OR
'0074 3008 0144 JR NC,OUTPT ; OUTPUT DATA
'0076 210010 0145 LD HL,MP CIM1 ; IF BORROW, INPUT
'0079 010500 0146 LD BC,5
'007C 180A 0147 JR PUTFF
'007E 210810 0148 OUTPT: LD HL,MP COM1 ; IF NO BORROW, OUTPUT
'0081 010D00 0149 LD BC,0DH
'0084 0F 0150 RRCA ; IF OUTPUT DATA, UBYTE
'0085 0F 0151 RRCA ; NEEDS TO BE SHIFTED
'0086 0F 0152 RRCA ; RIGHT FOUR TIMES.
'0087 0F 0153 RRCA
'0088 0F 0154 PUTFF: RRCA ; LOOP TO DECIDE WHICH
'0089 3803 0155 JR C,LOAD ; USER INTERRUPTED
'008B 19 0156 ADD HL,DE ; NEXT USER
'008C 18FA 0157 JR PUTFF
'008E 09 0158 LOAD: ADD HL,BC
'008F C1 0159 POP BC
'0090 AF 0160 XOR A
'0091 BE 0161 CP (HL) ; SEE IF 0 IN DATA BYTE
'0092 2002 0162 JR NZ,STAT
'0094 3D 0163 DEC A
'0095 77 0164 LD (HL),A ; IF SO CHANGE TO 0FFH
'0096 3AAF13 0165 STAT: LD A,(PTBYTE) ; GET CURRENT STATUS
'0099 A0 0166 AND B ; CLEAR CURRENT USER BIT
'009A D30F 0167 OUT (STATUS),A ; OUTPUT NEW STATUS
'009C 32AF13 0168 LD (PTBYTE),A ; SAVE NEW STATUS
'009F 3AAB13 0169 LD A,(OUTFLG) ; GET OUTPUT FLAG
'00A2 B7 0170 OR A
'00A3 2804 0171 JR Z,RTRN ; IF FLAG = 0 NORMAL RET
'00A5 219801' 0172 LD HL,SET0
'00A8 E3 0173 EX (SP),HL ; NEW RETURN ADDRESS
'00A9 D9 0174 RTRN: EXX
'00AA 08 0175 EX AF,AF' ; RETRIEVE REGISTERS
'00AB ED45 0176 RETN
  
```

```

0178 ;
0179 ;
0180 ;
'00AD 222810 0181 AROUND: LD (MPCOM3),HL ; USER 3 OUTPUT BUFFER
'00B0 19 0182 ADD HL,DE
'00B1 223810 0183 LD (MPCOM4),HL ; USER 4 OUTPUT BUFFER
'00B4 215D03' 0184 LD HL,VCTRS ; FILL VECTOR TABLE
'00B7 11C013 0185 LD DE,VTABL ; IN MEMORY
'00BA 012000 0186 LD BC,20H
'00BD EDB0 0187 LDIR
'00BF 21A601' 0188 LD HL,TIME ; LOAD REAL TIME ADDR
'00C2 22BE13 0189 LD (RTINT),HL ; INTO VECTOR ADDR
'00C5 3EBE 0190 LD A,0BEH
'00C7 D308 0191 OUT (CTC0),A ; CTC INTERRUPT VECTOR
'00C9 3EA5 0192 LD A,0A5H
'00CB D30B 0193 OUT (CTC3),A ; CTC CONTROL WORD
'00CD 3E80 0194 LD A,128
'00CF D30B 0195 OUT (CTC3),A ; TIME CONSTANT
'00D1 ED5E 0196 IM 2
'00D3 FB 0197 EI
'00D4 DD210810 0198 LD IX,MPCOM1
'00D8 DD7E05 0199 LOOP: LD A,(IX+DATA) ; SYSTEM OUTPUTS 0FFH
'00DB FEFF 0200 CP 0FFH ; WHEN BAUD RATE CON-
'00DD 20F9 0201 JR NZ,LOOP ; STANTS READY. BAUD
'00DF AF 0202 XOR A ; RATE CONSTANTS OUTPUT
'00E0 DD7705 0203 LD (IX+DATA),A ; TO PROGRAMMABLE
'00E3 3AAD13 0204 LD A,(BR1) ; BAUD RATE GENE-
'00E6 D30D 0205 OUT (BAUD1),A ; RATORS. USERS 1,2
'00E8 3AAE13 0206 LD A,(BR2)
'00EB D30E 0207 OUT (BAUD2),A ; USERS 3,4
'00ED DD210010 0208 LD IX,MPCIM1
'00F1 CD7102' 0209 CALL TSTRT ; START USER 1
'00F4 DD211010 0210 LD IX,MPCIM2
'00F8 CD7102' 0211 CALL TSTRT ; START USER 2
'00FB DD212010 0212 LD IX,MPCIM3
'00FF CD7102' 0213 CALL TSTRT ; START USER 3
'0102 DD213010 0214 LD IX,MPCIM4
'0106 CD7102' 0215 CALL TSTRT ; START USER 4
'0109 3EF0 0216 LD A,0F0H
'010B 32AF13 0217 LD (PTBYTE),A ; OUTPUT STATUS
'010E D30F 0218 OUT (STATUS),A ; TO STATUS PORT

```



```

0220 ;
0221 ;
0222 ;
0223 ;
0224 ;           USER POLLING ROUTINE
0225 ;
0226 ;           THIS PORTION OF THE PROGRAM IS CONCERNED WITH CHECKING
0227 ;           USER'S MEMORY TO SEE IF THERE IS DATA TO INPUT TO THE
0228 ;           OR IF THE SYSTEM HAS DATA TO BE OUTPUT.
0229 ;
0230 ;
'0110 DD210010 0231 START: LD      IX, MPCIM1      ; SET UP INSERT MEMORY
'0114 DD7E05   0232 START1: LD     A, (IX+DATA)    ; SEE IF SYSTEM HAS TAKEN
'0117 FEFF    0233          CP      0FFH        ; PREVIOUS BYTE OF DATA.
'0119 2006    0234          JR      NZ, ENDI
'011B CD5A02' 0235          CALL   DELTA          ; CHECK IOFF-EOFF
'011E C44201' 0236          CALL   NZ, INPUT
'0121 CD0502' 0237 ENDI:   CALL   NXTUSR
'0124 28EE    0238          JR      Z, START1      ; IF <4 USERS, NEXT USER
'0126 DD210810 0239          LD     IX, MPCOM1     ; ELSE CHECK OUTPUT
'012A DD7E05   0240 START2: LD     A, (IX+DATA)    ; SEE IF SYSTEM HAS DATA
'012D A7      0241          AND     A              ; TO OUTPUT TO USER
'012E 2805    0242          JR      Z, DAT
'0130 CD6101' 0243          CALL   OTDATA
'0133 1806    0244          JR      ENDO
'0135 CD5A02' 0245 DAT:   CALL   DELTA          ; ELSE CHECK IOFF-EOFF
'0138 C42C02' 0246          CALL   NZ, CONOUT      ; IF NOT=0 OUTPUT DATA
'013B CD0502' 0247 ENDO:   CALL   NXTUSR
'013E 28EA    0248          JR      Z, START2      ; IF <4 USERS NEXT USER
'0140 18CE    0249          JR      START        ; ELSE START OVER
0250 ;
0251 ;

```

0253 ;
 0254 ;
 0255 ;
 0256 ;
 0257 ;
 0258 ;
 0259 ;
 0260 ;
 0261 ;
 0262 ;
 0263 ;
 0264 ;
 0265 ;
 0266 ;

DATA INPUT ROUTINE

THIS ROUTINE CHECKS EACH USER'S MEMORY TO SEE WHO HAS DATA TO INPUT TO THE SYSTEM. IF A USER HAS DATA TO INPUT, THE DATA IS PUT IN THE DATA BYTE AND THE APPROPRIATE BIT IN THE STATUS PORT BYTE IS SET AND OUTPUT TO THE STATUS PORT.

```

'0142 DD6E00 0267 INPUT: LD L,(IX+0) ; GET EXTRACT BUFFER
'0145 DD6601 0268 LD H,(IX+1) ; POINTER INTO HL REG
'0148 0600 0269 LD B,0 ; AND GET THE OFFSET
'014A DD4E04 0270 LD C,(IX+EOFF) ; INTO BC REG
'014D 09 0271 ADD HL,BC ; ADD OFFSET TO POINTER
'014E 7E 0272 LD A,(HL) ; PUT DATA IN A
'014F DD7705 0273 LD (IX+DATA),A ; PUT DATA IN DATA
'0152 79 0274 LD A,C ; BYTE AND INCREMENT
'0153 3C 0275 INC A ; EXTRACT OFFSET
'0154 DDBE02 0276 CP (IX+SIZE) ; MAKE SURE NOT TOO BIG
'0157 2001 0277 JR NZ,INCEX ; IF NOT JUMP
'0159 AF 0278 XOR A ; ELSE ZERO OFFSET
'015A DD7704 0279 INCEX: LD (IX+EOFF),A ; LOAD NEW OFFSET
'015D 1E01 0280 LD E,1 ; SET E FOR INPUT
'015F 182E 0281 JR SET ; SET STATUS BIT
0282 ;
0283 ;
  
```

ADDR	CODE	STMT	SOURCE	STATEMENT
------	------	------	--------	-----------

0285 ;
 0286 ;
 0287 ;
 0288 ;
 0289 ;
 0290 ;
 0291 ;
 0292 ;
 0293 ;
 0294 ;
 0295 ;
 0296 ;
 0297 ;
 0298 ;
 0299 ;

DATA OUTPUT ROUTINE

THIS ROUTINE IS RESPONSIBLE FOR TAKING DATA FROM THE SYSTEM AND OUTPUTTING IT TO THE APPROPRIATE USER. THE SYSTEM CAUSES AN NMI FOR EACH BYTE OF DATA SENT. IF NO INTERRUPTS ARE BEING GENERATED THE OFFSETS MUST BE CHECKED TO SEE IF THEY ARE EQUAL. IF THEY ARE THERE IS NO DATA LEFT IN THE DATA BUFFER.

'0161	CD5A02'	0300	OTDATA:	CALL	DELTA	
'0164	CD6202'	0301		CALL	DELTA1	; SEE IF OFFSET > LIMIT
'0167	A7	0302		AND	A	; OF BUFFER
'0168	20D1	0303		JR	NZ,ENDO	; IF NOT, READY FOR OUTPUT
'016A	DD6E00	0304	OTBYTE:	LD	L,(IX+0)	; GET INSERT POINTER
'016D	DD6601	0305		LD	H,(IX+1)	; INTO HL REG
'0170	DD7E05	0306		LD	A,(IX+DATA)	
'0173	0600	0307		LD	B,0	; AND LOAD THE OFFSET
'0175	DD4E03	0308		LD	C,(IX+IOFF)	; INTO THE BC REGISTER
'0178	09	0309		ADD	HL,BC	; ADD OFFSET TO INDEX
'0179	77	0310		LD	(HL),A	; PUT DATA IN INSERT BUFFER
'017A	AF	0311		XOR	A	
'017B	DD7705	0312		LD	(IX+DATA),A	; CLEAR DATA BUFFER
'017E	DD3403	0313		INC	(IX+IOFF)	; INC INSERT BUFFER OFFSET
'0181	DD7E02	0314		LD	A,(IX+SIZE)	; GET BUFFER SIZE
'0184	DDBE03	0315		CP	(IX+IOFF)	; MAKE SURE NOT TOO LARGE
'0187	2004	0316		JR	NZ,INCIN	; IF NOT, SKIP
'0189	AF	0317		XOR	A	
'018A	DD7703	0318		LD	(IX+IOFF),A	; LOAD NEW OFFSET
'018D	1E10	0319	INCIN:	LD	E,10H	; SET E FOR OUTPUT
		0320				
		0321				

```

    0323 ;
    0324 ;
    0325 ;
    0326 ;          SET AND RESET ROUTINES
    0327 ;
    0328 ;
'018F CD1902' 0329 SET:  CALL  SETUP
'0192 F3      0330      DI
'0193 3E01    0331      LD   A,1
'0195 32AB13 0332      LD   (OUTFLG),A      ; SET OUTPUT FLAG
'0198 3AAF13 0333 SET0:  LD   A,(PTBYTE)      ; GET CURRENT STATUS
'019B B3      0334      OR   E          ; SET USER BIT
'019C 32AF13 0335      LD   (PTBYTE),A
'019F D30F    0336      OUT  (STATUS),A      ; OUTPUT NEW STATUS
'01A1 AF      0337      XOR  A
'01A2 32AB13 0338      LD   (OUTFLG),A      ; CLEAR OUTPUT FLAG
'01A5 C9      0339      RET
  
```

```

0341 ;
0342 ;
0343 ;
0344 ; REAL TIME INTERRUPT ROUTINE
0345 ;
0346 ;
'01A6 E5 0347 TIME: PUSH HL
'01A7 F5 0348 PUSH AF
'01A8 21A413 0349 LD HL,TIMER ; ADDR FOR # OF COUNTS
'01AB 7E 0350 LD A,(HL) ; INCREMENT FOR EACH
'01AC 3C 0351 INC A ; INTERRUPT
'01AD FE4B 0352 CP 75 ; 75 COUNTS = 1 SEC
'01AF 2803 0353 JR Z,SEC
'01B1 77 0354 LD (HL),A ; IF < 75 THEN
'01B2 184C 0355 JR RTURN ; RETURN
'01B4 AF 0356 SEC: XOR A ; ELSE CLEAR COUNT
'01B5 77 0357 LD (HL),A
'01B6 21AA13 0358 LD HL,SECOND ; AND INCREMENT
'01B9 7E 0359 LD A,(HL)
'01BA 3C 0360 INC A ; SECONDS
'01BB 27 0361 DAA ; STORE IN BCD
'01BC FE60 0362 CP 60H ; 60 SECS = 1 MIN
'01BE 2803 0363 JR Z,MIN
'01C0 77 0364 LD (HL),A ; IF < 60 THEN
'01C1 183D 0365 JR RTURN ; RETURN
'01C3 AF 0366 MIN: XOR A ; ELSE CLEAR SECONDS
'01C4 77 0367 LD (HL),A
'01C5 2B 0368 DEC HL
'01C6 7E 0369 LD A,(HL) ; AND INCREMENT
'01C7 3C 0370 INC A ; MINUTES
'01C8 27 0371 DAA ; BCD
'01C9 FE60 0372 CP 60H ; 60 MIN = 1 HR
'01CB 2803 0373 JR Z,HR
'01CD 77 0374 LD (HL),A ; IF < 60 THEN
'01CE 1830 0375 JR RTURN ; RETURN
'01D0 AF 0376 HR: XOR A ; ELSE CLEAR MINUTES
'01D1 77 0377 LD (HL),A
'01D2 2B 0378 DEC HL
'01D3 7E 0379 LD A,(HL) ; AND INCREMENT
'01D4 3C 0380 INC A ; HOURS
'01D5 27 0381 DAA ; BCD
'01D6 FE24 0382 CP 24H ; 24 HOURS = 1 DAY
'01D8 2803 0383 JR Z,DY
'01DA 77 0384 LD (HL),A ; IF < 24 THEN
'01DB 1823 0385 JR RTURN ; RETURN
'01DD AF 0386 DY: XOR A ; ELSE CLEAR HOURS
'01DE 77 0387 LD (HL),A
'01DF 2B 0388 DEC HL
'01E0 7E 0389 LD A,(HL) ; AND INCREMENT
'01E1 3C 0390 INC A ; DAYS
'01E2 27 0391 DAA ; BCD
'01E3 FE30 0392 CP 30H ; 30 DAYS = 1 MONTH
'01E5 2803 0393 JR Z,MNTH
'01E7 77 0394 LD (HL),A ; IF < 30 THEN
'01E8 1816 0395 JR RTURN ; RETURN
'01EA 3E01 0396 MNTH: LD A,1 ; ELSE SET DAY = 1
'01EC 77 0397 LD (HL),A

```

MPC ADDR	MPC CODE	CONTROL STMT	SOFTWARE SOURCE	STATEMENT	
'01ED	2B	0398		DEC	HL
'01EE	7E	0399		LD	A, (HL) ; AND INCREMENT
'01EF	3C	0400		INC	A ; MONTH
' '0	27	0401		DAA	; BCD
'01F1	FE12	0402		CP	12H ; 12 MONTHS = 1 YEAR
'01F3	2803	0403		JR	Z, YR
'01F5	77	0404		LD	(HL), A ; IF < 12 THEN
'01F6	1808	0405		JR	RTURN ; RETURN
'01F8	3E01	0406	YR:	LD	A, 1 ; ELSE SET MONTH = 1
'01FA	77	0407		LD	(HL), A
'01FB	2B	0408		DEC	HL
'01FC	7E	0409		LD	A, (HL) ; AND INCREMENT
'01FD	3C	0410		INC	A ; YEAR
'01FE	27	0411		DAA	; BCD
'01FF	77	0412		LD	(HL), A
'0200	F1	0413	RTURN:	POP	AF
'0201	E1	0414		POP	HL
'0202	FB	0415		EI	
'0203	ED4D	0416		RETI	

```

ADDR CODE STMT SOURCE STATEMENT
0418 ;
0419 ;
0420 ; STATUS PORT ROUTINES
0421 ;
0422 ; THE NXTUSR ROUTINE READIES THE IX REGISTER FOR THE
0423 ; USER
0424 ; THE SETUP ROUTINE RETURNS TO THE CALLER WITH THE ST
0425 ; BIT SET IN THE ACCUMULATOR FOR THE CURRENT USER. ALL
0426 ; BITS IN THE ACCUMULATOR ARE RESET. THE STATUS BYTE IS
0427 ; ORGANIZED AS FOLLOWS:
0428 ;
0429 ; *****
0430 ; BIT 0 1 2 3 4 5 6
0431 ; *****
0432 ; FUNC-*USR 1 * USR 2 * USR 3 * USR 4 * USR 1 * USR 2 * USR 3 :
0433 ; TION *INPUT * INPUT * INPUT * INPUT * OUTPT * OUTPT * OUTPT :
0434 ; *****
0435 ;
0436 ; BIT DEFINITIONS
0437 ;
0438 ; INPUT 0 - NO DATA AVAILABLE
0439 ; 1 - DATA AVAILABLE
0440 ; OUTPUT 0 - NOT READY FOR DATA
0441 ; 1 - READY FOR DATA
0442 ;
'0205 FB 0443 NXTUSR: EI
'0206 DDE5 0444 PUSH IX ; SET INDEX POINTER
'0208 E1 0445 POP HL ; FOR NEXT USER
'0209 CB5D 0446 BIT 3,L ; IF OUTPUT LOOP
'020B C42C02' 0447 CALL NZ,CONOUT ; OUTPUT BYTE
'020E 7D 0448 LD A,L
'020F C610 0449 ADD A,10H ; NEXT USER POINTER
'0211 CB77 0450 BIT 6,A ; SEE IF MORE THAN 4
'0213 C0 0451 RET NZ ; USERS. IF SO, RETURN
'0214 6F 0452 CONT: LD L,A ; HL HAS NEW INDEX
'0215 E5 0453 PUSH HL ; POINTER. LOAD INTO
'0216 DDE1 0454 POP IX ; IX REGISTER AND
'0218 C9 0455 RET ; RETURN
0456 ;
'0219 DDE5 0457 SETUP: PUSH IX
'021B E1 0458 POP HL ; GET MEMORY POINTER
'021C 7D 0459 LD A,L
'021D 0F 0460 RRCA
'021E 0F 0461 RRCA
'021F 0F 0462 RRCA
'0220 0F 0463 RRCA ; A= 00H - 03H
'0221 E60F 0464 AND 0FH
'0223 47 0465 LD B,A
'0224 04 0466 INC B
'0225 1001 0467 SHIFT: DJNZ NOMORE
'0227 C9 0468 RET ; E HAS BIT SET FOR USER
'0228 CB23 0469 NOMORE: SLA E
'022A 18F9 0470 JR SHIFT

```

```

0472 ;
0473 ;
0474 ;
'022C CD5A02' 0475 CONOUT: CALL DELTA ; CHECK IOFF-EOFF
'022F C8 0476 RET Z ; IF ZERO NO DATA
'0230 DD4E06 0477 LD C,(IX+PORT) ; LOAD C STATUS PORT
'0233 0C 0478 INC C
'0234 ED78 0479 CHKST: IN A,(C) ; CHECK STATUS
'0236 E604 0480 AND PTSTAT
'0238 28FA 0481 JR Z,CHKST
'023A DD6E00 0482 LD L,(IX+0)
'023D DD6601 0483 LD H,(IX+1)
'0240 DD5E04 0484 LD E,(IX+EOFF)
'0243 1600 0485 LD D,0 ; GET EXTRACT ADDRESS
'0245 19 0486 ADD HL,DE ; IN HL REGISTER
'0246 7E 0487 LD A,(HL) ; GET DATA IN A
'0247 0D 0488 DEC C ; C HAS DATA PORT
'0248 ED79 0489 OUT (C),A ; OUTPUT DATA
'024A 1C 0490 INC E ; INCREMENT EXTRACT
'024B DD7E02 0491 LD A,(IX+SIZE) ; OFFSET AND
'024E BB 0492 CP E ; CHECK SIZE
'024F 2002 0493 JR NZ,OK ; IF OK SAVE EXTRACT
'0251 1E00 0494 LD E,0 ; ELSE ZERO EXTRACT
'0253 DD7304 0495 OK: LD (IX+EOFF),E
'0256 DDE5 0496 PUSH IX
'0258 E1 0497 POP HL
'0259 C9 0498 RET ; AND RETURN
0499 ;
0500 ;

```



```

0502 ;
0503 ;
0504 ;
'025A DD4604 0505 DELTA: LD B,(IX+EOFF) ; GET DIFFERENCE BETWEEN
'025D DD7E03 0506 LD A,(IX+IOFF) ; INSERT AND EXTRACT
'0260 90 0507 SUB B ; A CONTAINS DIFFERENCE
'0261 C9 0508 RET
'0262 FEFF 0509 DELTA1: CP -1 ; CHECK FOR DELTA
'0264 2808 0510 JR Z,SET1 ; LIMITS
'0266 3C 0511 INC A
'0267 DDBE02 0512 CP (IX+SIZE)
'026A 2802 0513 JR Z,SET1
'026C AF 0514 XOR A ; IF NOT AT LIMIT CLEAR
'026D C9 0515 RET ; A AND RETURN
'026E 3E01 0516 SET1: LD A,1 ; ELSE, RETURN WITH
'0270 C9 0517 RET ; A = 1
0518 ;

```

0520 ;
 0521 ;
 0522 ; START ROUTINE FOR TERMINALS
 0523 ;

THIS ROUTINE INITIALIZES THE DART FOR THE APPROPRIATE USER AND ENABLES THE INTERRUPTS FOR THAT CHANNEL.

```

0271 DD7E06 0528 TSTRT: LD A,(IX+PORT)
0274 FE00 0529 CP 0
0276 280A 0530 JR Z,STRT1 ; USER 1 PORT 0
0278 CB57 0531 BIT 2,A
027A 280F 0532 JR Z,STRT2 ; USER 2 PORT 2
027C CB4F 0533 BIT 1,A
027E 2814 0534 JR Z,STRT3 ; USER 3 PORT 4
0280 181B 0535 JR STRT4 ; USER 4 PORT 6
0282 0E01 0536 STRT1: LD C,1
0284 CDAC02' 0537 CALL IDART
0287 3ED0 0538 LD A,0D0H ; LOAD INTERRUPT VECTOR
0289 1819 0539 JR RTN
028B 0E03 0540 STRT2: LD C,3
028D CDAC02' 0541 CALL IDART
0290 3ED0 0542 LD A,0D0H ; LOAD INTERRUPT VECTOR
0292 1810 0543 JR RTN
0294 0E05 0544 STRT3: LD C,5
0296 CDAC02' 0545 CALL IDART
0299 3EC0 0546 LD A,0C0H ; LOAD INTERRUPT VECTOR
029D 1807 0547 JR RTN
029F CDAC02' 0548 STRT4: LD C,7
02A2 3EC0 0549 CALL IDART
02A4 ED79 0550 LD A,0C0H ; LOAD INTERRUPT VECTOR
02A6 3EFF 0551 RTN: OUT (C),A
02A8 DD7705 0552 LD A,0FFH ; LOAD A 0FFH IN
02AB C9 0553 LD (IX+DATA),A ; THE DATA BYTE LOCATION
0554 RET
  
```

DART INITIALIZATION ROUTINE

```

02AC 0608 0558 IDART LD B,8
02AE 21BA02' 0559 LD HL,IDART1 ; OUTPUT THE SETUP
02B1 EDB3 0560 OTIR ; CONSTANTS TO THE
02B3 CBC9 0561 SET 1,C ; DARTS
02B5 0603 0562 LD B,3
02B7 EDB3 0563 OTIR
02B9 C9 0564 RET
0565 ;
02BA 0444011C 0566 IDART1 DEFB 4,44H,1,1CH,3,0C1H,5,0EAH,1,1CH,2
03C105EA
011C02
  
```

```

ADDR CODE STMT SOURCE STATEMENT
0568 ;
0569 ;
0570 ;
0571 ; INTERRUPT ROUTINES
0572 ;
0573 ; THESE ROUTINES ARE THE INTERRUPT RESPONSE ROUTINES.
0574 ; DEPENDING UPON WHICH USER INTERRUPTS, THE MPC-4 WILL
0575 ; ENTER AT ONE OF THE USRXIN ENTRIES.
0576 ;
0577 ;
0578 ;
'>02C5 0579 TRINT1:
'>02C5 0580 EXINT1:
'>02C5 0581 TRINT2:
'>02C5 0582 EXINT2:
'>02C5 0583 TRINT3:
'>02C5 0584 EXINT3:
'>02C5 0585 TRINT4:
'02C5 FB 0586 EXINT4: EI
'02C6 ED4D 0587 RETI
0588 ;
0589 ;
0590 ;
'02C8 DDE5 0591 RCINT1: PUSH IX ; USER 1 ENTRY
'02CA C5 0592 PUSH BC
'02CB DD210010 0593 LD IX,MPCIM1
'02CF 0E00 0594 LD C,0
'02D1 1855 0595 JR DATAIN
'02D3 DDE5 0596 RCINT2: PUSH IX ; USER 2 ENTRY
'02D5 C5 0597 PUSH BC
'02D6 DD211010 0598 LD IX,MPCIM2
'02DA 0E02 0599 LD C,2
'02DC 184A 0600 JR DATAIN
'02DE DDE5 0601 RCINT3: PUSH IX ; USER 3 ENTRY
'02E0 C5 0602 PUSH BC
'02E1 DD212010 0603 LD IX,MPCIM3
'02E5 0E04 0604 LD C,4
'02E7 183F 0605 JR DATAIN
'02E9 DDE5 0606 RCINT4: PUSH IX ; USER 4 ENTRY
'02EB C5 0607 PUSH BC
'02EC DD213010 0608 LD IX,MPCIM4
'02F0 0E06 0609 LD C,6
'02F2 1834 0610 JR DATAIN
0611 ;
0612 ;

```

```

0614 ;
0615 ;
0616 ;          SPECIAL CONDITION INTERRUPT ROUTINES
0617 ;
0618 ;
'02F4 C5      0619 SCINT1: PUSH    BC
'02F5 E5      0620          PUSH    HL
'02F6 210710  0621          LD      HL, MPCIM1+ERROR ; ERROR BYTE USER 1
'02F9 0E01    0622          LD      C, 1
'02FB 1819    0623          JR      ERRST
'02FD C5      0624 SCINT2: PUSH    BC
'02FE E5      0625          PUSH    HL
'02FF 211710  0626          LD      HL, MPCIM2+ERROR ; ERROR BYTE USER 2
'0302 0E03    0627          LD      C, 3
'0304 1810    0628          JR      ERRST
'0306 C5      0629 SCINT3: PUSH    BC
'0307 E5      0630          PUSH    HL
'0308 212710  0631          LD      HL, MPCIM3+ERROR ; ERROR BYTE USER 3
'030B 0E05    0632          LD      C, 5
'030D 1807    0633          JR      ERRST
'030F C5      0634 SCINT4: PUSH    BC
'0310 E5      0635          PUSH    HL
'0311 213710  0636          LD      HL, MPCIM4+ERROR ; ERROR BYTE USER 4
'0314 0E07    0637          LD      C, 7
'0316 F5      0638 ERRST:  PUSH    AF
'0317 3E01    0639          LD      A, 1
'0319 ED79    0640          OUT    (C), A          ; READ STATUS AND
'031B ED78    0641          IN     A, (C)          ; STORE IN ERROR
'031D 77      0642          LD      (HL), A        ; BYTE
'031E 3E30    0643          LD      A, 30H
'0320 ED79    0644          OUT    (C), A
'0322 F1      0645          POP    AF
'0323 E1      0646          POP    HL
'0324 C1      0647          POP    BC
'0325 FB      0648          EI
'0326 ED4D    0649          RETI
0650 ;
0651 ;

```

```

MPC          MPC CONTROL SOFTWARE          SD SYSTEMS Z80 ASSEMBLER PAGE 0018
ADDR CODE      STMT SOURCE STATEMENT

                0653 ;
                0654 ;
                0655 ;
'0328 F5        0656 DATAIN: PUSH    AF
'0329 E5        0657          PUSH    HL                ; SAVE REGISTERS
'032A CD5A02'   0658          CALL    DELTA          ; SEE IF BUFFER FULL
'032D CD6202'   0659          CALL    DELTAI
'0330 A7        0660          AND     A
'0331 F5        0661          PUSH    AF
'0332 ED78      0662          IN     A,(C)          ; INPUT DATA
'0334 47        0663          LD     B,A
'0335 F1        0664          POP    AF
'0336 201D      0665          JR     NZ,INC          ; IF BUFFER FULL RETURN
'0338 78        0666          LD     A,B
'0339 DD4E03    0667          LD     C,(IX+IOFF)    ; ELSE PUT DATA
'033C 0600      0668          LD     B,0            ; IN INSERT ADDRESS
'033E DD6E00    0669          LD     L,(IX+0)
'0341 DD6601    0670          LD     H,(IX+1)
'0344 09        0671          ADD    HL,BC
'0345 77        0672          LD     (HL),A
'0346 DD3403    0673          INC    (IX+IOFF)
'0349 DD7E02    0674          LD     A,(IX+SIZE)
'034C DDBE03    0675          CP     (IX+IOFF)
'034F 2004      0676          JR     NZ,INC
'0351 AF        0677          XOR    A
'0352 DD7703    0678          LD     (IX+IOFF),A
'0355 E1        0679 INC:    POP    HL                ; RESTOR REGISTERS
'0356 F1        0680          POP    AF            ; AND RETURN
'0357 C1        0681          POP    BC
'0358 DDE1      0682          POP    IX
'035A FB        0683          EI
'035B ED4D      0684          RETI

```

0686 ;
0687 ;
0688 ;
0689 ;
0690 ;

INTERRUPT VECTORS

'035D C502C502' 0691 VCTRS: DEFW TRINT4,EXINT4,RCINT4,SCINT4,TRINT3,EXINT3
E9020F03
C502C502
'0369 DE020603' 0692 DEFW RCINT3,SCINT3,TRINT2,EXINT2,RCINT2,SCINT2
C502C502
D302FD02
'0375 C502C502' 0693 DEFW TRINT1,EXINT1,RCINT1,SCINT1
C802F402

ADDR CODE STMT SOURCE STATEMENT

CROSS REFERENCE LISTING

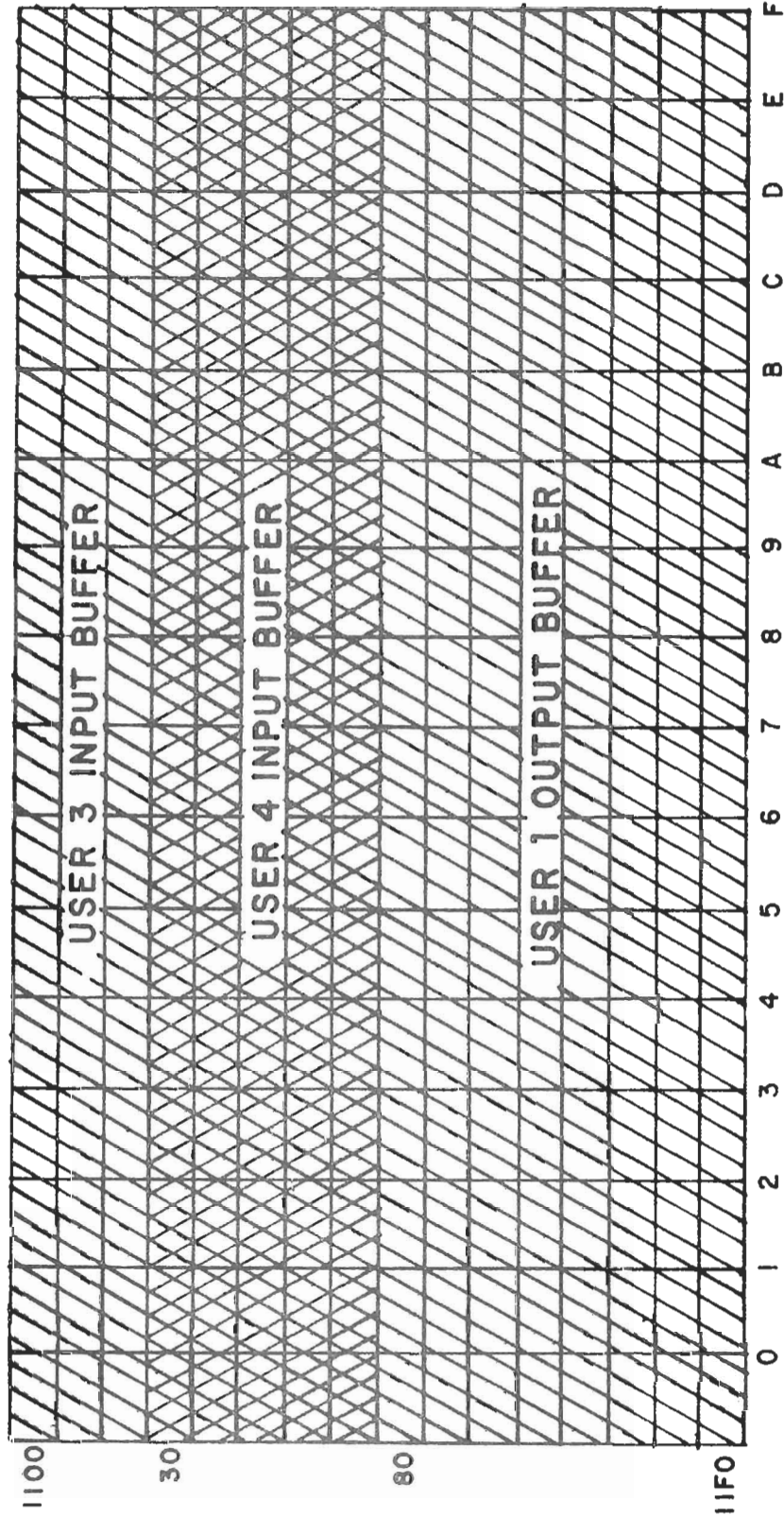
SYMBOL	VALUE	TYPE	STMT	STATEMENT	REFERENCES						
-----	-----	-----	-----	-----	-----						
AROUND	00AD		0181	0128							
BAUD1	000D		0055	0205							
BAUD2	000E		0056	0207							
BR1	13AD		0053	0204							
BR2	13AE		0054	0206							
BUFAD1	1040		0029	0114							
CHKST	0234		0479	0481							
CONOUT	022C		0475	0447	0246						
CONT	0214		0452								
CTC0	0008		0065	0191							
CTC1	0009		0066								
CTC2	000A		0067								
CTC3	000B		0068	0195	0193						
CTCV0	13B8		0030	0031							
CTCV1	13BA		0031	0032							
CTCV2	13BC		0032	0033							
CTCV3	13BE		0033	0034							
DAT	0135		0245	0242							
DATA	0005		0062	0553	0312	0306	0273	0240	0232	0203	0199
DATAIN	0328		0656	0610	0605	0600	0595				
DAY	13A7		0038	0039							
DELTA	025A		0505	0658	0475	0300	0245	0235			
DELTA1	0262		0509	0659	0301						
DY	01DD		0386	0383							
ENDI	0121		0237	0234							
ENDO	013B		0247	0303	0244						
E0FF	0004		0061	0505	0495	0484	0279	0270			
ERROR	0007		0064	0636	0631	0626	0621				
ERRST	0316		0638	0633	0628	0623					
EXINT1	02C5		0580	0693							
EXINT2	02C5		0582	0692							
EXINT3	02C5		0584	0691							
EXINT4	02C5		0586	0691							
FILL	0009		0083	0085							
HOUR	13A8		0039	0040							
HR	01D0		0376	0373							
IBUFF	0050		0022	0092							
IDART	02AC		0558	0549	0545	0541	0537				
IDART1	02BA		0566	0559							
INC	0355		0679	0676	0665						
INCEX	015A		0279	0277							
INCIN	018D		0319	0316							
INIT	0000		0079								
INPUT	0142		0267	0236							
IOFF	0003		0060	0678	0675	0673	0667	0506	0318	0315	0313
				0308							
ISIZAD	1002		0024	0091							
ISZFIL	001E		0093	0095							
LOAD	008E		0158	0155							
LOOP	00D8		0199	0201							
MIN	01C3		0366	0363							
MINUTE	13A9		0040	0041							
MNTH	01EA		0396	0393							

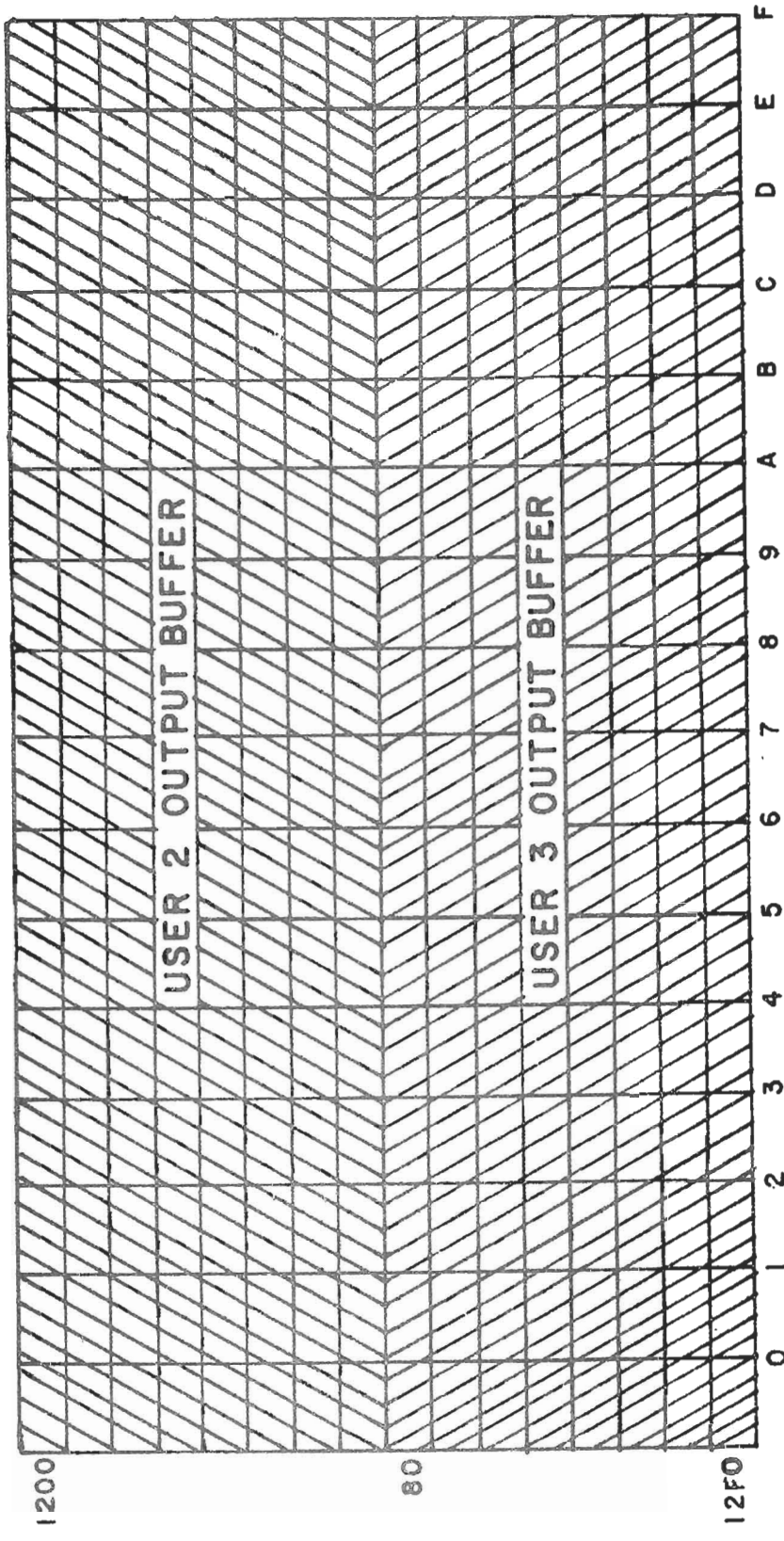
MPC ADDR	MPC CODE	CONTROL	SOFTWARE	SD	SYSTEMS	Z80	ASSEMBLER	PAGE	0021
ADDR	CODE	STMT	SOURCE	STATEMENT					
MONTH	13A6	0037	0038						
MPCIM	1000	0021	0058	0057	0054	0053	0052	0051	0043 0042
			0035	0030	0029	0026	0025	0024	
MPCIM1	1000	0043	0621	0593	0231	0208	0145	0115	0080 0047
			0044						
MPCIM2	1010	0044	0626	0598	0210	0117	0045		
MPCIM3	1020	0045	0631	0603	0212	0119	0046		
MPCIM4	1030	0046	0636	0608	0214	0121			
MPCOM1	1008	0047	0239	0198	0148	0123	0048		
MPCOM2	1018	0048	0126	0049					
MPCOM3	1028	0049	0181	0050					
MPCOM4	1038	0050	0183						
NOMORE	0228	0469	0467						
NXTUSR	0205	0443	0247	0237					
OBUFF	0080	0023	0098						
OK	0253	0495	0493						
OSIZAD	100A	0025	0097						
OSZFIL	0029	0099	0101						
OTBYTE	016A	0304							
OTDATA	0161	0300	0243						
OUTFLG	13AB	0051	0338	0332	0169				
OUTPT	007E	0148	0144						
PORT	0006	0063	0528	0477					
PORT1	1006	0026	0103						
PTBYTE	13AF	0057	0335	0333	0217	0168	0165	0086	
PTLD	0037	0106	0111						
PTNUM1	0000	0027	0104						
STAT	0004	0028	0480						
TFF	0088	0154	0157	0147					
RCINT1	02C8	0591	0693						
RCINT2	02D3	0596	0692						
RCINT3	02DE	0601	0692						
RCINT4	02E9	0606	0691						
RTINT	13BE	0034	0189						
RTN	02A4	0551	0547	0543	0539				
RTRN	00A9	0174	0171						
RTURN	0200	0413	0405	0395	0385	0375	0365	0355	
SCINT1	02F4	0619	0693						
SCINT2	02FD	0624	0692						
SCINT3	0306	0629	0692						
SCINT4	030F	0634	0691						
SEC	01B4	0356	0353						
SECOND	13AA	0041	0358						
SET	018F	0329	0281						
SET0	0198	0333	0172						
SET1	026E	0516	0513	0510					
SETUP	0219	0457	0329						
SHIFT	0225	0467	0470						
SIZE	0002	0059	0674	0512	0491	0314	0276		
STACK	1400	0058	0079						
START	0110	0231	0249						
START1	0114	0232	0238						
START2	012A	0240	0248						
AT	0096	0165	0162						
STATUS	000F	0069	0336	0218	0167				
STR1	0282	0536	0530						
STR2	028B	0540	0532						
STR3	0294	0544	0534						

APPENDIX E
MEMORY MAP

* NOTE 1020-13FF mapped to 1400-17FF !

INPUT										OUTPUT																						
BUFFER ADDRESS	BUFFER SIZE	INSERT OFFSET	EXTRACT OFFSET	DATA BYTE	PORT NUMBER	ERROR STATUS	BUFFER ADDRESS	BUFFER SIZE	INSERT OFFSET	EXTRACT OFFSET	DATA BYTE	PORT NUMBER	ERROR STATUS	USER 1	USER 2	USER 3	USER 4															
1000	"	"	"	"	"	"	"	"	"	"	"	"	"																			
10	"	"	"	"	"	"	"	"	"	"	"	"	"																			
20	"	"	"	"	"	"	"	"	"	"	"	"	"																			
30	"	"	"	"	"	"	"	"	"	"	"	"	"																			
40	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px;">USER 1 INPUT BUFFER</div> <div style="border: 1px solid black; padding: 5px;">USER 2 INPUT BUFFER</div> </div>																															
50																																
60																																
70																																
80																																
90																																
A0																																
B0																																
C0																																
D0																																
E0																																
10F0																																
0																		1	2	3	4	5	6	7	8	9	A	B	C	D	E	F





1300	USER 4 OUTPUT BUFFER																									70																														
80																																																								
90																																																								
A0	TIMER YEAR MONTH DAY HOUR MINUTE SECOND																																																							
B0																										CTC 0												CTC 1																		
C0	USER 4 TRANSMIT BUFFER EMPTY VECTOR												USER 4 SPECIAL RECEIVE VECTOR												CTC 2		CTC 3																													
D0	USER 2 TRANSMIT BUFFER EMPTY VECTOR												USER 2 SPECIAL RECEIVE VECTOR												USER 3		USER 3																													
E0	USER 1 TRANSMIT BUFFER EMPTY VECTOR												USER 1 SPECIAL RECEIVE VECTOR												USER 1		USER 1																													
13F0	STACK																																																							

0 1 2 3 4 5 6 7 8 9 A B C D E F

APPENDIX F
MPC-4 PORT DESIGNATIONS

PORT DESIGNATIONS

S100 BUS PORT ADDRESSES

FFH - When the data output to this port matches the MPC-4 board number, that MPC-4 is selected.

FXH - Where X represents the board number as selected by X10. When this port is output to, an NMI is generated on the MPC-4 which causes the status port to be updated. An input gives status of all four MPC-4 channels.

MPC-4 PORT ADDRESSES

00 - USER 1 DATA PORT
01 - USER 1 COMMAND PORT
02 - uUSER 2 DATA
03 - USER 2 COMMAND
04 - USER 3 DATA
05 - USER 3 COMMAND
06 - USER 4 DATA
07 - USER 4 COMMAND
08 - CTC CHANNEL 0
09 - CTC CHANNEL 1
0AH - CTC CHANNEL 2
0BH - CTC CHANNEL 3
0CH - NO DESIGNATION
0DH - BAUD RATE PORT FOR USERS 1 AND 2
0EH - BAUD RATE PORT FOR USERS 3 AND 4
0FH - STATUS PORT

APPENDIX G
CTC PROGRAMMING

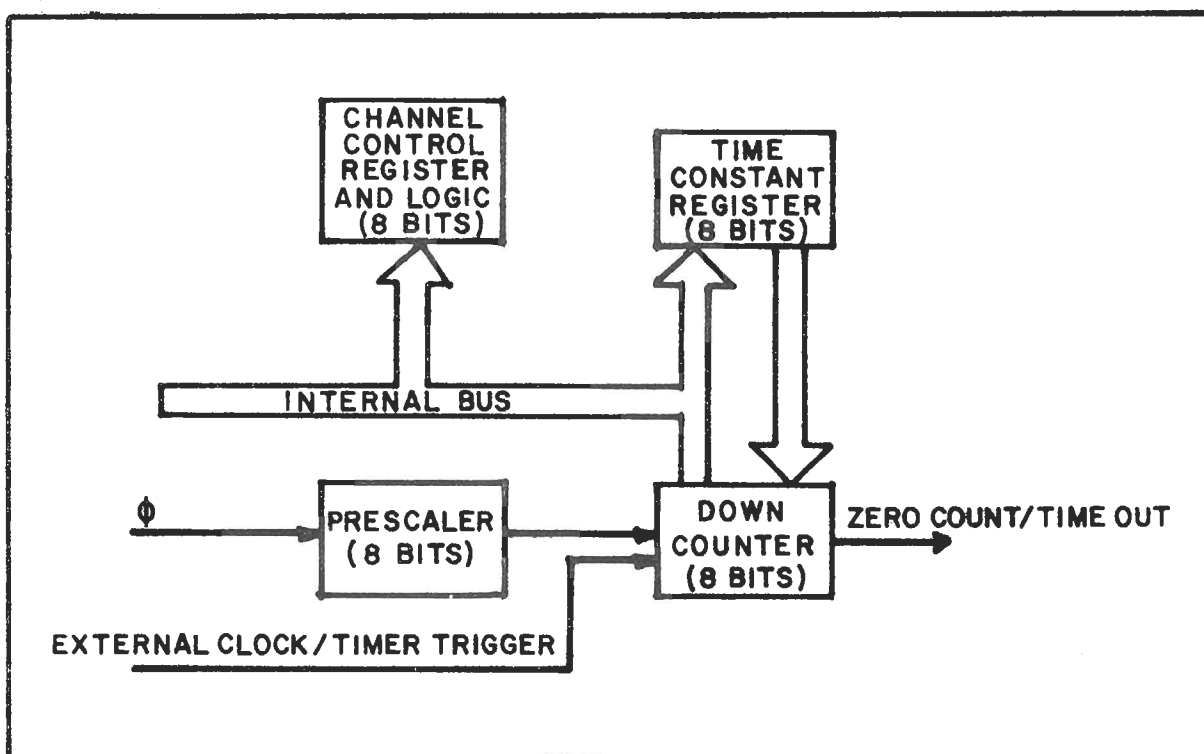
("Reproduced by permission c 1979 Zilog, Inc. This material shall not be reproduced without the written consent of Zilog, Inc.")

CTC PROGRAMMING

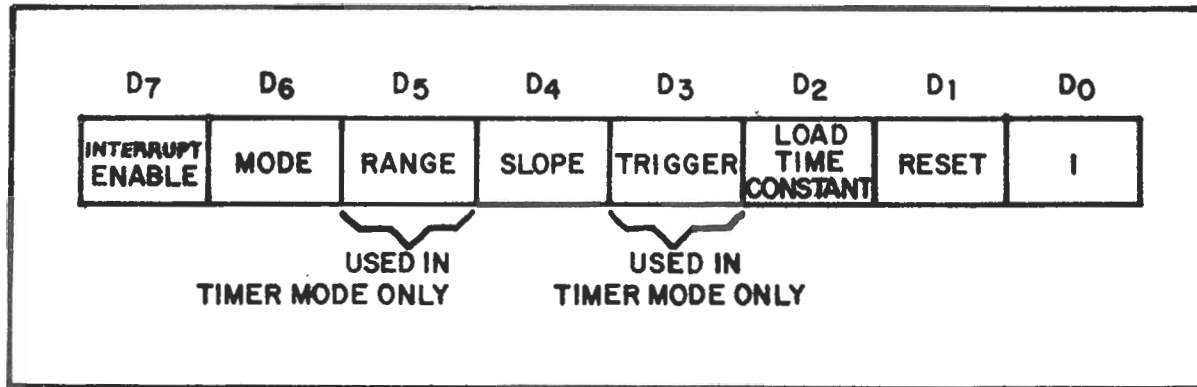
Before a Z80-CTC channel can begin counting or timing operations, a Channel Control Word and a Time Constant data word must be written to it by the CPU. These words will be stored in the Channel Control Register and the Time Constant Register of that channel. In addition, if any of the four channels have been programmed with bit 7 of their Channel Control Words to enable interrupts, an Interrupt Vector must be written to the appropriate register in the CTC. Due to automatic features in the Interrupt Control Logic, one pre-programmed Interrupt Vector suffices for all four channels.

LOADING THE CHANNEL CONTROL REGISTER

To load a Channel Control Word, the CPU performs a normal I/O Write sequence to the port address corresponding to the desired CTC channel. Two CTC input pins, namely CS0 and CS1, are used to form a 2-bit binary address to select one of four channels within the device. In many system architectures, these two input pins are connected to Address Bus lines A0 and A1, respectively, so that the four channels in a CTC device will occupy contiguous I/O port addresses. A word written to a CTC channel will be interpreted as a Channel Control Word, and loaded into the Channel Control Register, its bit 0 is a logic 1. The other seven bits of this word select operating modes and conditions as indicated in the diagram below. Following the diagram the meaning of each bit will be discussed in detail.



LOADING THE CHANNEL CONTROL REGISTER (cont'd)



Bit 7=1

The channel is enabled to generate an interrupt request sequence every time the Down Counter reaches a zero-count condition. To set this bit to 1 in any of the four Channel Control Registers necessitates that an Interrupt Vector also be written to the CTC before operation begins. Channel interrupts may be programmed in either Counter Mode or Timer mode. If an updated Channel Control Word is written to a channel already in operation, with bit 7 set, the interrupt enable selection will not be retroactive to a preceding zero-count condition.

Bit 7=0

Channel interrupts disabled.

Bit 6=1

Counter Mode selected. The Down Counter is decremented by each triggering edge of the External Clock (CLK/TRG) input. The Prescaler is not used.

Bit 6=0

Time Mode selected. The Prescaler is clocked by the System Clock and the output of the Prescaler in turn clocks the Down Counter. The output of the Down Counter (the channel's ZC/TO output) is a uniform pulse train of period given by the product.

$$t * P * TC$$

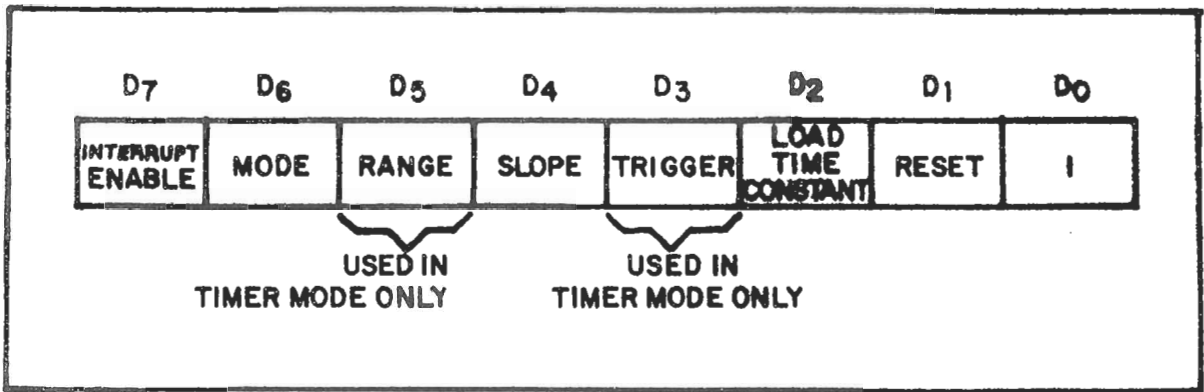
where t is the period of System Clock, P is the Prescaler factor of 16 or 256, and TC is the time constant data word.

Bit 5=1

(Defined for Timer Mode only.) Prescaler factor is 256.

Bit 5=0

(Defined for Timer Mode only.) Prescaler factor is 16.



Bit 4=1

TIMER MODE - positive edge trigger starts timer operation.
 COUNTER MODE - positive edge decrements the down counter.

Bit 4=0

TIMER MODE - negative edge trigger starts timer operation.
 COUNTER MODE - negative edge decrements the down counter.

Bit 3=1-

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

Bit 3=0

Timer Mode Only - Timer begins operation on the rising edge of T of the machine cycle following the one that loads the time constant.

Bit 2=1

The time constant data word for the Time Constant Register will be the next word written to this channel. If an updated Channel Control Word and time constant data word are written to a channel while it is already in operation, the Down Counter will continue decrementing to zero before the new time constant is loaded into it.

Bit 2=0

No time constant data word for the Time Constant Register should be expected to follow. To program bit 2 to this state implies that this Channel Control Word is intended to update the status of a channel already in operation, since a channel will not operate without a correctly programmed data word in the Time Constant Register, and a set bit 2 in this Channel Control Word provides the only way of writing to the Time Constant Register.

Bit 1=1

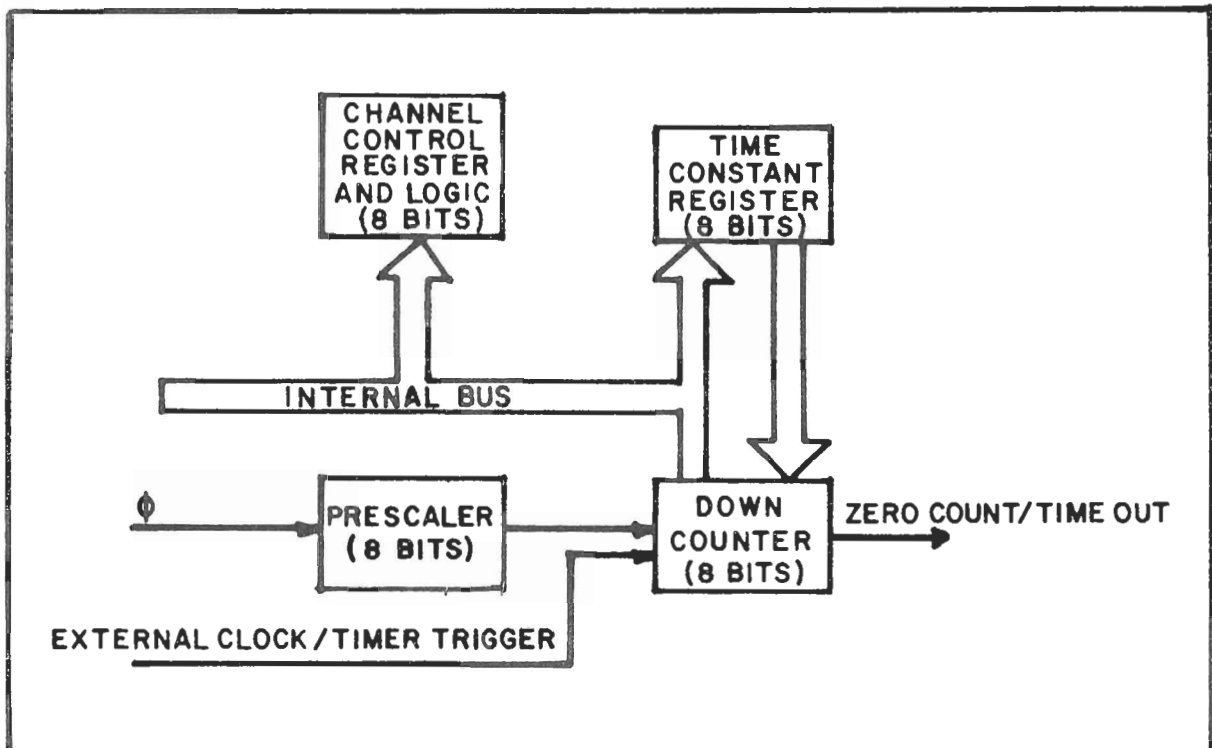
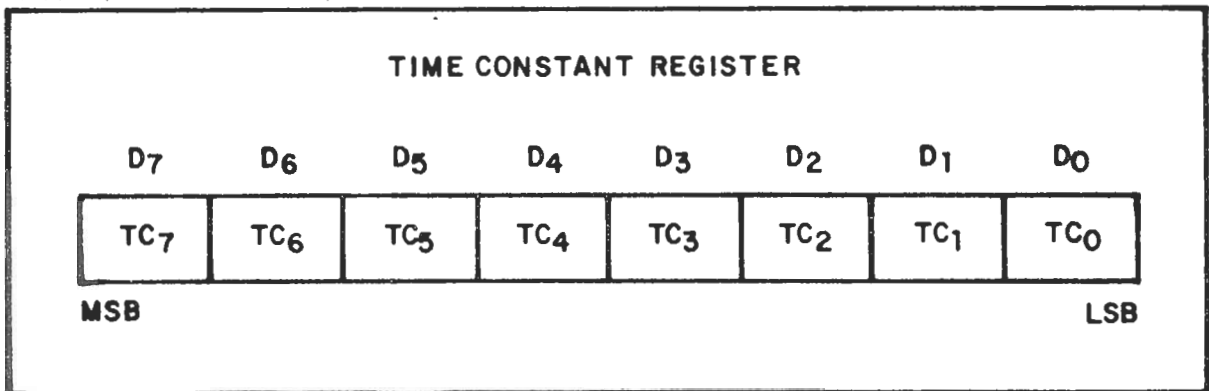
Reset channel. Channel stops counting or timing. This is not a stored condition. Upon writing into this bit a reset pulse discontinues current channel operation, however, none of the bits in the channel control register are changed. If both bit 2=1 and bit 1=1 the channel will resume operation upon loading a time constant.

Bit 1=0

Channel continues current operation.

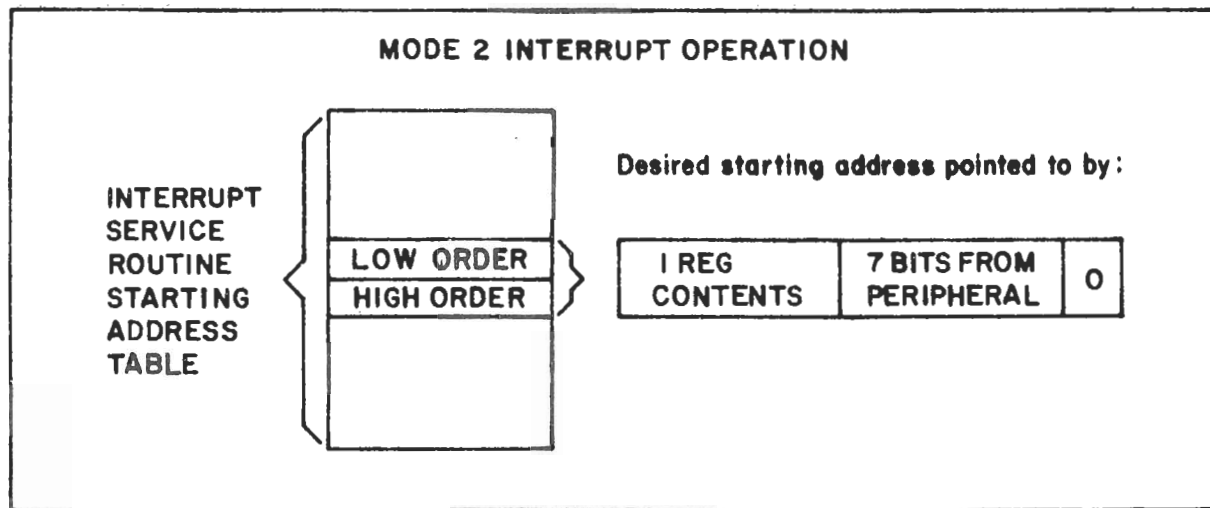
LOADING THE TIME CONSTANT REGISTER

A channel may not begin operation in either Timer Mode or Counter Mode unless a time constant data word is written into the Time Constant Register by the CPU. This data word will be expected on the next I/O Write to this channel following the I/O Write of the Channel Control Word, provided that bit 2 of the Channel Control Word is set. The time constant data word may be any integer value in the range 1-256. If all eight bits in this word are zero, it is interpreted as 256. If a time constant data word is loaded to a channel already in operation, the Down Counter will continue decrementing to zero before the new time constant is loaded from the Time Constant Register to the Down Counter.



LOADING THE INTERRUPT VECTOR REGISTER

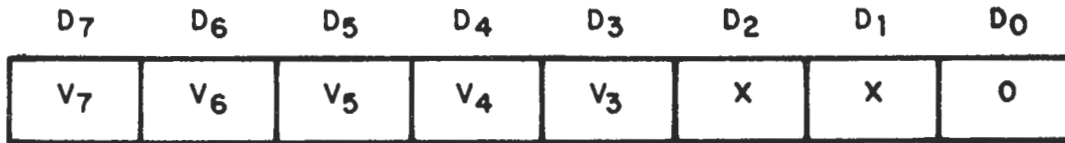
The Z80-CTC has been designed to operate with the Z80-CPU programmed for mode 2 interrupt response. Under the requirements of this mode, when a CTC channel requests an interrupt and is acknowledged, a 16-bit pointer must be formed to obtain a corresponding interrupt service routine starting address from a table in memory. The upper 8 bits of this pointer are provided by the CPU's I register, and the lower 8 bits of the pointer are provided by the CTC in the form of an Interrupt Vector unique to the particular channel that requested the interrupt.



The high order 5 bits of this Interrupt Vector must be written to the CTC in advance as part of the initial programming sequence. To do so, the CPU must write to the I/O port address corresponding to the CTC channel 0, just as it would if a Channel Control Word were being written to that channel, except that bit 0 of the word being written must contain a 0. (As explained before, if bit 0 of a word written to a channel were set to 1, the word would be interpreted as a Channel Control Word, so a 0 in bit 0 signals the CTC to load the incoming word into the Interrupt Vector Register). Bits 1 and 2, however, are not used when loading this vector. At the time when the interrupting channel must place the Interrupt Vector on the Z80 Data Bus, the Interrupt Control Logic of the CTC automatically supplies a binary code in bits 1 and 2 identifying which of the four CTC channels is to be serviced.

INTERRUPT VECTOR REGISTER

INTERRUPT VECTOR REGISTER



SUPPLIED BY
USER

0	0	CHANNEL 0 (Highest Priority)
0	1	CHANNEL 1
1	0	CHANNEL 2
1	1	CHANNEL 3 (Lowest Priority)

AUTOMATICALLY INSERTED
BY Z80-CTC

APPENDIX H
Z80 DART PROGRAMMING

-
("Reproduced by permission c 1979 Zilog, Inc. This
material shall not be reproduced without the
written consent of Zilog, Inc.")

Z80 DART PROGRAMMING

To program the Z80 DART, the system program first issues a series of commands that initialize the basic mode and then other commands that qualify conditions within the selected mode. For example, the character length, clock rate, number of stop bits, even or odd parity are first set, then the interrupt mode and, finally, receiver or transmitter enable.

Both channels contain command registers that must be programmed via the system program prior to operation. The Channel Select input (B/A) and the Control/Data input (C/D) are the command structure addressing controls, and are normally controlled by the CPU address bus.

WRITE REGISTERS

The Z80 DART contains six registers (WR0-WR5) in each channel that are programmed separately by the system program to configure the functional personality of the channels. With the exception of WR0, programming the write registers requires two bytes. The first byte contains three bits (D - D) that point to the selected register; the second byte is the actual control word that is written into the register to configure the Z80 DART.

WR0 is a special case in that all the basic commands (CMD -CMD) can be accessed with a single byte. Reset (internal or external) initializes the pointer bits D - D2 to point to WR0. This means that a register cannot be pointed to in the same operation as a channel reset.

WRITE REGISTER FUNCTIONS

- WR0 Registers pointers, initialization commands for the various modes, etc.
- WR1 Transmit/Receive interrupt and data transfer mode definition.
- WR2 Interrupt Vector (Channel B only).
- WR3 Receive parameters and control.
- WR4 Transmit/Receive miscellaneous parameters and modes.
- WR5 Transmit parameters and controls.

READ REGISTERS

The Z80 DART contains three registers (RR0-RR2) that can be read to obtain the status information for each channel (except for RR2, which applies to Channel B only). The status information includes error conditions, interrupt vector and standard communications-interface signals.

To read the contents of a selected read register other than RR0, the system program must first write the pointer byte to WR0 in exactly the same way as a write register operation. Then, by executing an input instruction, the contents of the addresses read register can be read by the CPU.

The status bits of RR0 and RR1 are carefully grouped to

APPENDIX I
MPC HOST PROGRAM

ADDR	CODE	STMT	SOURCE	STATEMENT
		0002	;	
		0003	;	
		0004	;	
		0005	MPCMEM EQU	1000H
>1000		0006	MPCBR EQU	MPCMEM+3ADH ; BAUD RATE BYTES IN MPC
>13AD		0007	UMASK EQU	MPCMEM+3ACH ; MASK FOR STATUS
>13AC		0008	MPCN EQU	7
>0007		0009	MPORT EQU	-1
>FFFF		0010	IDATA EQU	5
>0005		0011	ODATA EQU	13
>000D		0012	BRK EQU	0EEEEH ; BAUD RATE CONSTANTS
>EEEE		0013	;	
		0014	;	
		0015	GLOBAL SWITCH	
		0016	GLOBAL ACTIVE	
		0017	;	
		0018	;	
		0019	;	
		0020	;	
		0021	;	
		0022	MINIT LD	HL, BRK ; CONSTANTS
'0000	21EEEE	0023	LD	A, MPCN ; MPC PAGE #
'0003	3E07	0024	DI	
'0005	F3	0025	OUT	(MPORT), A ; SELECT MPC
'0006	D3FF	0026	LD	(MPCBR), HL ; STUFF BAUD RATES
'0008	22AD13	0027	LD	A, -1
'0009	3EFF	0028	LD	(MPCMEM+ODATA), A ; START MPC-4
'000D	320D10	0029	LD	A, (ACTIVE)
'0010	3AFFFF	0030	OUT	(MPORT), A ; SELECT ACTIVE MEMORY
'0013	D3FF	0031	EI	
'0015	FB	0032	RET	
'0016	C9	0033	;	
		0034	;	
		0035	;	
		0036	MSTAT LD	B, 1 ; SET UP FOR SMASK
'0017	0601	0037	JR	MRDY1
'0019	1802	0038	MRDY LD	B, 10H ; SET UP FOR SMASK
'001B	0610	0039	MRDY1 CALL	SETUP ; CREATE RDY MASK AND SET IX
'001D	CD2900'	0040	RDY IN	A, (0F0H+MPCN)
'0020	DBF7	0041	AND	B
'0022	A0	0042	RET	Z
'0023	C8	0043	LD	A, -1
'0024	3EFF	0044	RET	
'0026	C9	0045	SETUP0 SLA	B ; SHIFT MASK
'0027	CB20	0046	SETUP DEC	A
'0029	3D	0047	JR	NZ, SETUP0
'002A	20FB	0048	LD	IX, MPCMEM ; NOW SET UP IX
'002C	DD210010	0049	LD	A, C ; ACTV USER #
'0030	79	0050	DEC	A
'0031	3D	0051	RLCA	;
'0032	07	0052	RLCA	;
'0033	07	0053	RLCA	;
'0034	07	0054	RLCA	;
'0035	07	0055	LD	E, A
'0036	5F	0056	LD	D, 0
'0037	1600	0057	ADD	IX, DE
'0039	DD19	0058	RET	
'003B	C9			

ADDR	CODE	STMT	SOURCE	STATEMENT	
'003C	0601	0059	MIN	LD B,1	
'003E	CD2900'	0060		CALL SETUP	; CREATE RDY MASK AND SE=IX
'0041	CD2000'	0061	MINO	CALL RDY	; WAIT FOR DATA READY
'0044	CCFFFF	0062		CALL Z,SWITCH	
'0047	28F8	0063		JR Z,MINO	
'0049	3E07	0064		LD A,MPCN	; SELECT MPC MEMORY
'004B	F3	0065		DI	
'004C	D3FF	0066		OUT (MPORT),A	
'004E	DD5605	0067		LD D,(IX+IDATA)	
'0051	DD360500	0068		LD (IX+IDATA),0	; SET IDATA=0
'0055	1819	0069		JR MOUT1	
'0057	F1	0070	MOUT	POP AF	
'0058	FE00	0071		CP 0	
'005A	C8	0072		RET Z	; FILTER NULLS
'005B	F5	0073		PUSH AF	
'005C	79	0074		LD A,C	
'005D	0610	0075		LD B,10H	
'005F	CD2900'	0076		CALL SETUP	; CREATE RDY MASK AND SET IX
'0062	CD2000'	0077	MOUT0	CALL RDY	; WAIT FOR XMIT READY
'0065	28FB	0078		JR Z,MOUT0	
'0067	D1	0079		POP DE	; GET DATA
'0068	3E07	0080		LD A,MPCN	
'006A	F3	0081		DI	
'006B	D3FF	0082		OUT (MPORT),A	
'006D	DD720D	0083		LD (IX+ODATA),D	; DATA
'0070	78	0084	MOUT1	LD A,B	; RDY MASK
'0071	32AC13	0085		LD (UMASK),A	; MPC MEMORY
'0074	79	0086		LD A,C	; CURRENT USER
'0075	D3FF	0087		OUT (MPORT),A	
'0077	D3F7	0088		OUT (0F0H+MPCN),A	; FORCE MPC STATUS UPDATE
'0079	DBF7	0089	MOUT2	IN A,(0F0H+MPCN)	; STATUS
'007B	A0	0090		AND B	
'007C	20FB	0091		JR NZ,MOUT2	; WAIT FOR BUSY AGAIN
'007E	7A	0092		LD A,D	; DATA TO ACC
'007F	FB	0093		EI	
'0080	C9	0094		RET	

ERRORS=0000