

READER'S COMMENTS

Your feedback about this document is important to us: only in this way can we ascertain your needs and fulfill them in the future. Please take the time to fill out this questionnaire and return it to us. This information will be helpful to us, and, in time, to the future users of Zilog systems. Thank you.

Your Name: _____

Company Name: _____

Address: _____

Title of this document: _____

What software products do you have? _____

What is your hardware configuration (including memory size)? _____

Does this publication meet your needs? Yes No

If not, why not? _____

How do you use this publication? (Check all that apply)

- As an introduction to the subject?
- As a reference manual?
- As an instructor or student?

How do you find the material?

	Excellent	Good	Poor
Technicality	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completeness	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

What would have improved the material? _____

Other comments, suggestions or corrections: _____

If you found any mistakes in this document, please let us know what and where they were:



First Class

Permit No. 475
Cupertino
California
95014

Business Reply Mail

No Postage Necessary if Mailed in the United States

Postage Will Be Paid By



Zilog
Software Department Librarian
10460 Bubb Road
Cupertino, California 95014

Z80-MCB HARDWARE USER'S MANUAL

TABLE OF CONTENTS

	PAGE
1.0 INTRODUCTION.....	2
2.0 MCB DESIGN FEATURES.....	4
3.0 DYNAMIC RAM INTERFACE.....	6
4.0 MCB NON-VOLATILE MEMORY CAPABILITY.....	10
5.0 MEMORY PAGE SELECTION.....	11
6.0 I/O PORT DECODER.....	15
7.0 DMA TRANSFERS AND DATA BUS CONTROL.....	18
8.0 MCB PARALLEL INTERFACE.....	21
9.0 MCB SERIAL INTERFACE.....	25
10.0 MCB INTERRUPT STRUCTURE.....	29
11.0 A MINIMAL PROTOTYPE CONFIGURATION.....	35
12.0 MEMORY EXPANSION.....	43
13.0 SYSTEM EXPANSION.....	45
14.0 HOOKING UP THE MCB.....	48
15.0 MCB EDGE CONNECTOR SIGNAL LIST.....	52
16.0 MCB SCHEMATICS.....	54
17.0 MCB SPECIFICATIONS.....	64
18.0 Z80-CPU INSTRUCTION SET.....	65
19.0 PARTS LIST.....	70

Z80-MCB HARDWARE USER'S MANUAL

This document describes the structure and operation of the Zilog Z80 Microcomputer Board (MCB). It is not intended to give full descriptions on individual logic elements. The user is referred to selected documentation on the various components and to the following Zilog literature:

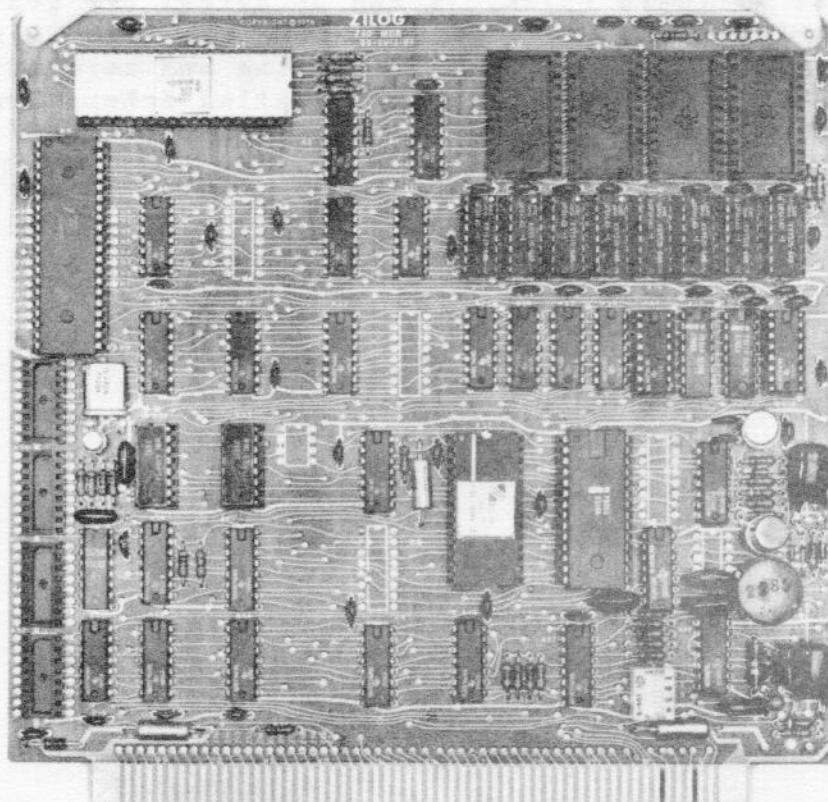
Z80-MCB Software Users Manual

Z80-CPU Technical Manual

Z80-Assembly Language Programming Manual

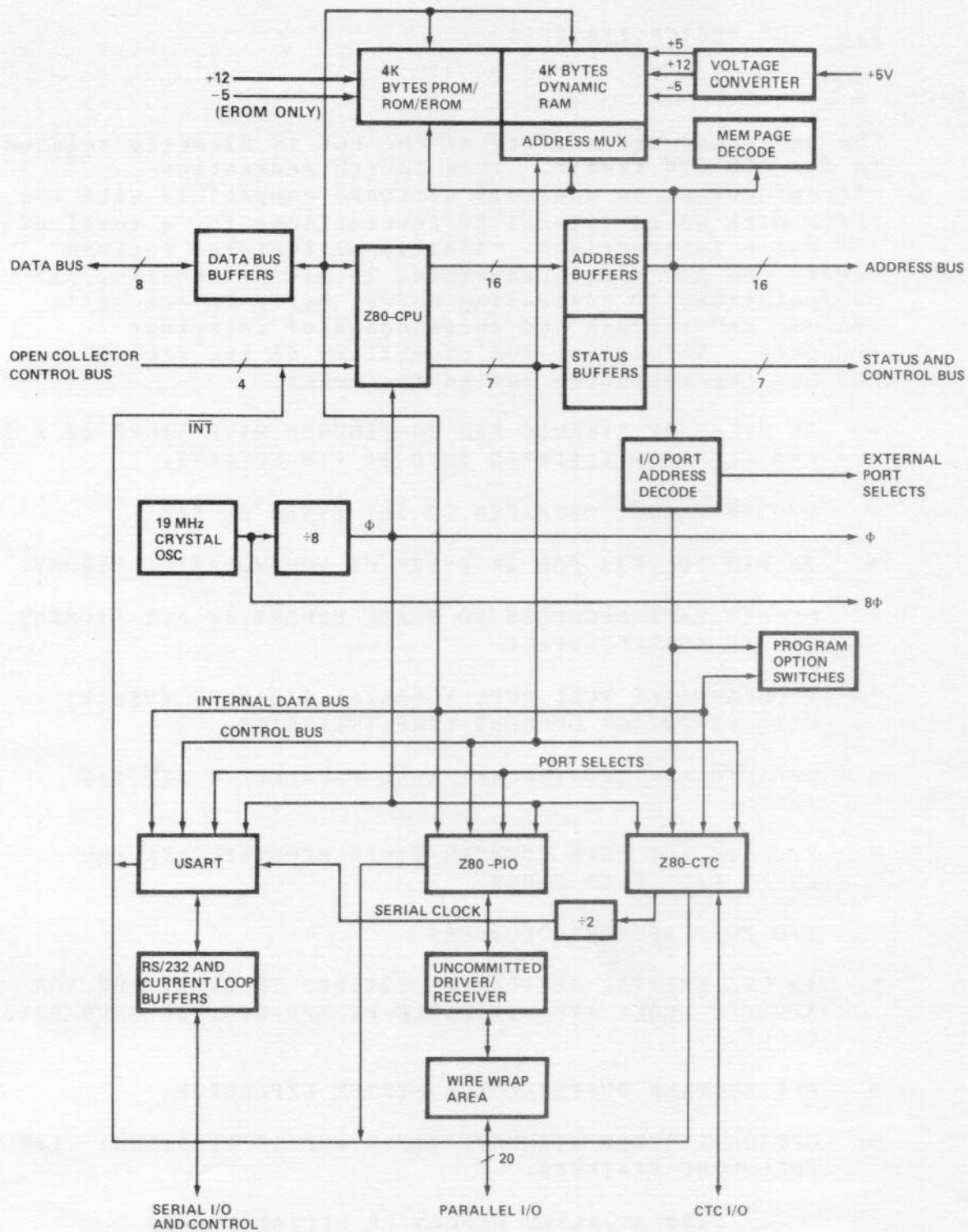
Z80-PIO Technical Manual

Z80-CTC Technical Manual



1.0 INTRODUCTION

The Zilog Z80-MCB is a single board microcomputer on a 7.7 X 7.5 in. PC card, designed to be adaptable to a wide range of OEM applications. The Block Diagram on the following page identifies the major components on the card, the heart of which is the Z80 Microprocessor. Associated logic includes 4K bytes of dynamic RAM, provision for up to 4K bytes of PROM, ROM or EPROM, both parallel and serial I/O ports, I/O port decoders and a crystal controlled clock. The parallel port is implemented with the Z80-PIO with area reserved for user applied driver and/or receiver logic. The Z80-CTC is used as a baud rate generator for the serial interface implemented with an 8251 USART. A voltage converter is included to provide the +12 and -5 volts needed by the dynamic RAMS, thus allowing the card to be powered by a single +5 volt supply. Strapping options are available for selecting several memory and I/O port configurations, terminal interface schemes and operating modes. Expansion of the card is made possible by feeding all buffered address, data and control lines to a 122 pin edge connector (compatible with Augut PN 14005-19P1 and Garry PN 4000-2). Three versions of monitor software (1/2K, 1K and 3K bytes) are available in bipolar PROMS for insertion into the four 24 pin PROM sockets, allowing software debugging and terminal interface (TTY, CRT or Disk). Detailed schematics on the MCB appear at the end of this Hardware Manual.



Z80-MCB BLOCK DIAGRAM

2.0 MCB DESIGN FEATURES

The power and versatility of the MCB is directly related to the Z80 CPU itself. This fourth generation microprocessor is upwardly software compatible with the 8080A with an additional 80 instructions for a total of 158 basic instructions. Additional features include memory and I/O block transfers, 16 bit arithmetic, bit manipulation, 10 addressing modes, built-in automatic dynamic RAM refresh and three modes of interrupt response. To utilize the capability of the Z80-CPU the MCB has the following design features:

- * 4K BYTES OF DYNAMIC RAM CONFIGURED WITH EIGHT 4K X 1 RAM ELEMENTS INSERTED INTO 16 PIN SOCKETS.
- * SYSTEM EASILY UPGRDED TO 16K BYTES OF RAM.
- * 24 PIN SOCKETS FOR 4K BYTES OF NON-VOLATILE MEMORY.
- * MEMORY PAGE DECODERS TO PLACE MEMORY IN ANY SEGMENT OF THE ADDRESS SPACE
- * PROGRAMMABLE FULL DUPLEX SERIAL I/O PORT (USART) WITH RS-232 OR CURRENT LOOP INTERFACE
- * Z80-PIO FOR TWO PROGRAMMABLE PARALLEL 8 BIT I/O PORTS.
- * Z80-CTC FOR USER COUNTER/TIMER REQUIREMENTS AND USART BAUD RATE CLOCK
- * I/O PORT ADDRESS DECODERS
- * 19 MHZ CRYSTAL OSCILLATOR DIVIDED TO 2.457 MHZ FOR Z80-CPU CLOCK AND DIVISIBLE BY Z80-CTC FOR BAUD RATE CLOCK.
- * ALL SIGNALS BUFFERED FOR SYSTEM EXPANSION.
- * OPTIONAL DEBUG FIRMWARE (1/2K AND 1K VERSIONS) WITH FOLLOWING FEATURES:
 - DISPLAY/ALTER MEMORY OR REGISTER DATA
 - GENERATE/READ PAPER TAPE
 - JUMP TO ADDRESS

- SET BREAKPOINTS (1K VERSION ONLY)
- * OPTIONAL 3K MONITOR FOR INTERFACE TO A FLOPPY DISK.

3.0 DYNAMIC RAM INTERFACE

Dynamic RAM Interface is greatly simplified using the Z80 microprocessor. During each memory opcode fetch cycle a dedicated line from the CPU (\overline{RFSH}) is used to indicate that a refresh read of all dynamic memories should be performed. With \overline{RFSH} in the true state (LOW), the lower 7 bits of the address bus identify one ROW address to be refreshed. Before the next opcode fetch, this address will have been incremented to point to the next ROW address. Since it is only necessary to refresh the 'ROWS', a total of 64 refresh cycles will refresh the entire 4K RAM. This 'invisible refresh' approach keeps the system running at full speed avoiding the complexity of trying to stop the CPU for a refresh operation.

The 4K RAM element will require 12 address lines to select a unique bit location. To allow for a 16 pin package, these 12 lines are divided into two groups; 6 row addresses and 6 column addresses. Each group is applied to the RAM on the same 6 input lines through an external multiplexer and latched into the RAM by applying two clock strobes (\overline{RAS} and \overline{CAS}) in succession. Figures 1 and 2 indicate the relationship between the CPU and memory control signals for opcode fetch and read/write cycles.

The first clock (\overline{RAS}) is generated from \overline{MREQ} and latches the 6 row addresses into the chip. After a delay through a low power inverter chain (A26), the second clock (\overline{CAS}) latches the 6 column addresses. The two to one multiplexer (A10 and A11) switches the addresses from row to column after the row address hold time but before \overline{CAS} . This switching strobe is taken from the delay chain at midpoint between \overline{RAS} and \overline{CAS} strobes (A26 pin 8). During \overline{RAS} only refresh, the generation of \overline{CAS} and the multiplexer switching signal will be disabled allowing only \overline{RAS} to be applied to the RAM elements. The PROM will also be disabled during memory refresh.

Considerable tolerance is allowed for the occurrence of the \overline{RAS} and \overline{CAS} signals within a memory cycle, allowing for propagation differences between the low power gates from system to system. For a read cycle the CPU does not require data for a minimum of 590ns after the addresses have stabilized. For write cycles input data is latched by \overline{CAS} or 'WRITE', whichever occurs last. Thus if \overline{CAS} is delayed with respect to 'WRITE', data

will not be latched until \overline{CAS} goes active.

The output from the RAMS are buffered onto the internal data bus through A27 and A30. These gates are only enabled for memory read operations.

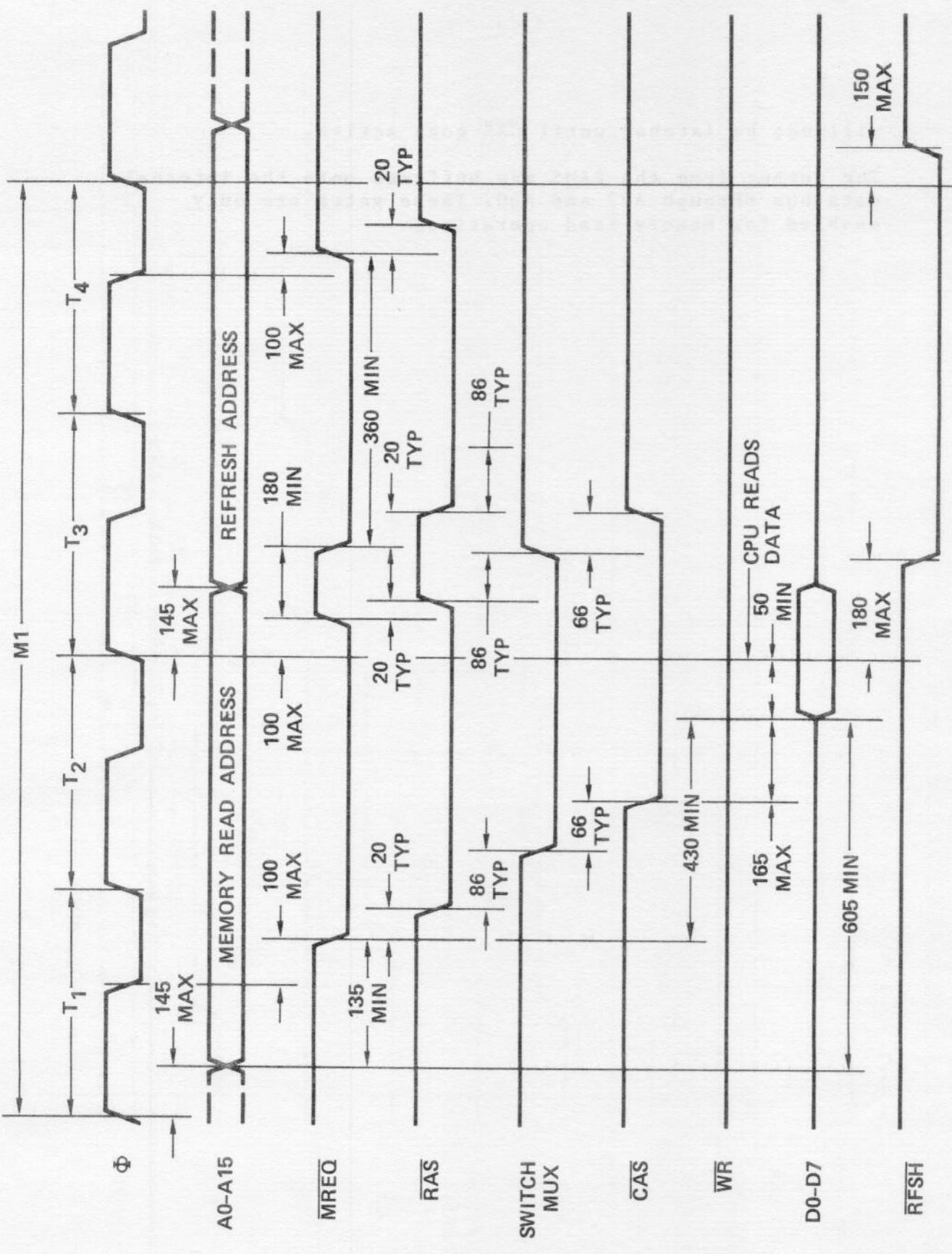


Figure 1 Instruction Op Code Fetch Cycle

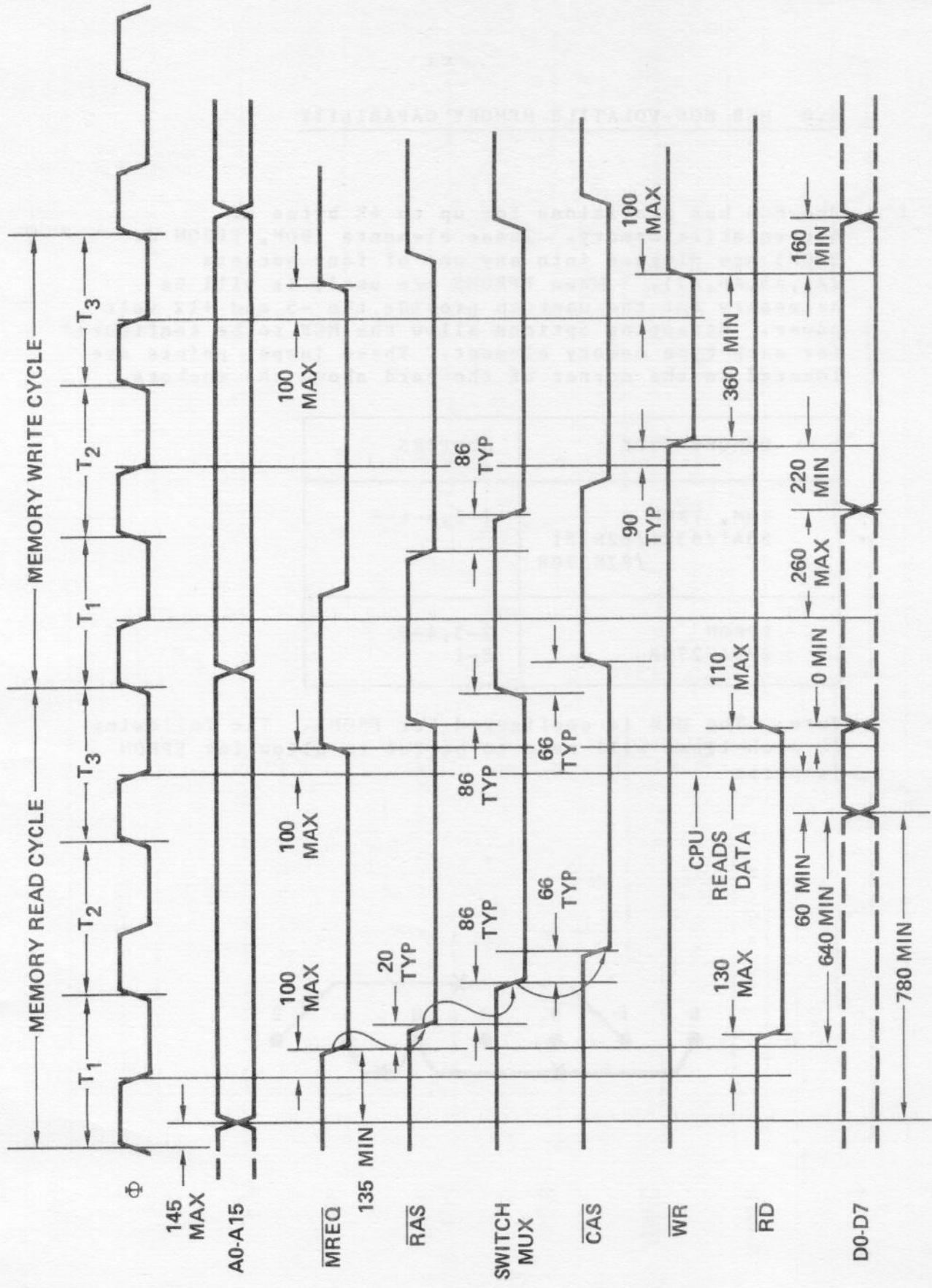


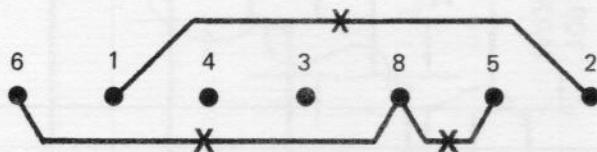
Figure 2 Memory Read And Write Cycles

4.0 MCB NON-VOLATILE MEMORY CAPABILITY

The MCB has provisions for up to 4K bytes of non-volatile memory. These elements (ROM, EPROM or PROM) are plugged into any one of four sockets (A4,A5,A6,A7). When EPROMS are used, it will be necessary for the user to provide the -5 and +12 volt power. Strapping options allow the MCB to be configured for each type memory element. These jumper points are located in the corner of the card above the sockets.

MEMORY TYPE	JUMPERS
ROM, PROM 6341/6381/82S181 /82S2708	1-2,5-6-8
EPROM 2704/2708	2-3,4-5 8-1

Note: The MCB is configured for PROMS. The following PC etch lines will need to be cut to allow for EPROM jumpers:



5.0 MEMORY PAGE SELECTION

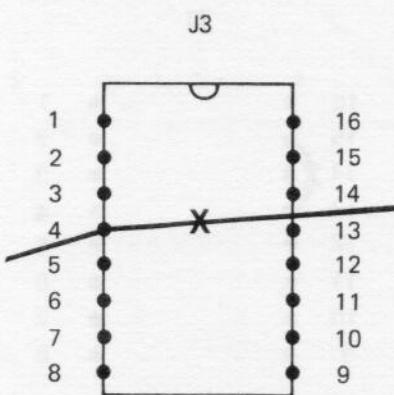
Normally, the 4K bytes of non-volatile memory are located in page 0 while the 4K bytes of dynamic RAM are located in page 1. (A page is 4K bytes of contiguous memory.) However, jumper programmable options allow the RAM and ROM to be placed anywhere in the 64K byte address space. Table 1 identifies a memory page with the appropriate output from the 74LS139 page decoder (Sheet 2 of schematic) and the jumper points on J3.

5.1 RAM PLACEMENT

Pins 9 and 11 on J3 are tied to appropriate pins indicated in table 1 on Page 13 to place the RAM in the address space. For example, if the RAM is to be placed in page 4 (Hex 4000), pins 9 and 11 are tied to pins 3 and 12. The placement of these jumpers (3 to 9 or 11, or 12 to 9 or 11) is immaterial.

The RAM enable line (sheet 2 of schematic) is routed to edge connector pins (21 and 16) and can be used to enable (jumper across 21 and 16) or disable (jumpers removed) on card RAM with off card logic.

Note: It will be necessary to cut PC etch in order to reconfigure RAM placement. To free J3-9 cut etch on component side of board at point shown below:



To free J3-11 cut etch on solder side at point shown below.

J3

16	•	•	1
15	•	•	2
14	•	•	3
13		•	4
12	X	•	5
11		•	6
10	•	•	7
9	•	•	8

5.2 NON-VOLATILE MEMORY PLACEMENT

Pins 15 and 16 on J3 are tied to appropriate pins indicated in table 1 on Page 14 to place the Non-Volatile memory bank in the address space. For example, for the Non-Volatile memory to reside in page 5 (Hex 5000), pins 15 and 16 are tied to pins 4 and 12.

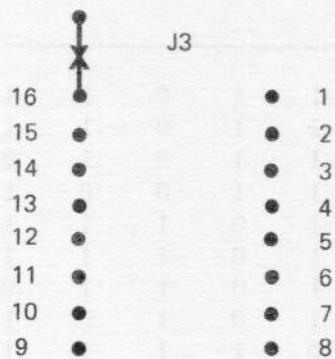
The Non-Volatile memory enable line (sheet 2 and 3 of schematics) is routed to edge connector pins (83 and 92) and can be used to enable (jumper across 83 and 92) or disable (jumper removed) on card memory.

Note: It will be necessary to cut PC etch in order to reconfigure Non-Volatile memory. To free J3-15 cut etch on solder side at point shown below:

J3

16	•	•	1
15		•	2
14	X	•	3
13		•	4
12	•	•	5
11	•	•	6
10	•	•	7
9	•	•	8

To free J3-16 cut etch on solder side at point shown below:

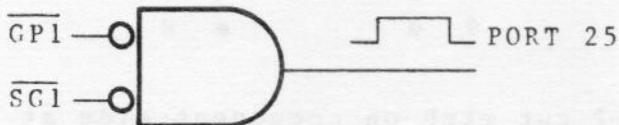


PAGE ADDRESS				A22 DECODER OUTPUT PIN (74LS139)								JUMPERS ON J3	
15	BINARY 14	13	12	PEX	9	10	11	12	7	6	5	4	
0	0	0	0	0000	1	1	1	0	1	1	1	0	3,13
0	0	0	1	1000	1	1	1	0	1	1	0	1	4,13
0	0	1	0	2000	1	1	1	0	1	0	1	1	5,13
0	0	1	1	3000	1	1	1	0	0	1	1	1	6,13
0	1	0	0	4000	1	1	0	1	1	1	1	0	3,12
0	1	0	1	5000	1	1	0	1	1	1	0	1	4,12
0	1	1	0	6000	1	1	0	1	1	0	1	1	5,12
0	1	1	1	7000	1	1	0	1	0	1	1	1	6,12
1	0	0	0	8000	1	0	1	1	1	1	1	0	3,7
1	0	0	1	9000	1	0	1	1	1	1	0	1	4,7
1	0	1	0	A000	1	0	1	1	1	0	1	1	5,7
1	0	1	1	B000	1	0	1	1	0	1	1	1	6,7
1	1	0	0	C000	0	1	1	1	1	1	1	0	3,8
1	1	0	1	D000	0	1	1	1	1	1	0	1	4,8
1	1	1	0	E000	0	1	1	1	1	0	1	1	5,8
1	1	1	1	F000	0	1	1	1	0	1	1	1	6,8

TABLE 1 PAGE SELECT DECODER

6.0 I/O POPT DECODER

I/O Port Decoders are provided on the card. Three of the decoded outputs are used to select the MCB USART, CTC and PIO. The other outputs are uncommitted and can be used for external I/O device selection. The decoding scheme consists of first decoding the two least significant address bits (AB0 and AB1) into four unique signals (SG0, SG1, SG2 and SG3). (A45 on Sheet 4.) Another decoder (A40) transfers the next 6 address lines (AB2-AB7) into 8 signals (GP0-GP7). This decoder is user programmable using jumper pins J2. Table 2 and sheet 4 of the schematic detail the decoding procedure. The positions of the jumpers on J2 will determine the grouping of the I/O devices within the 256 port address space. For any given jumper pattern, 32 contiguous port addresses can be decoded by logically combining the GPX and SGX terms. For example, if it is desired to place the port logic in the address space from 20 to 3F, the jumpers corresponding to Y=2 and Z=3 are installed (J2-5 to J2-15, J2-1 to J2-7 and J2-3 to J2-6). To decode address 25, GP1 and SG1 are combined as follows:

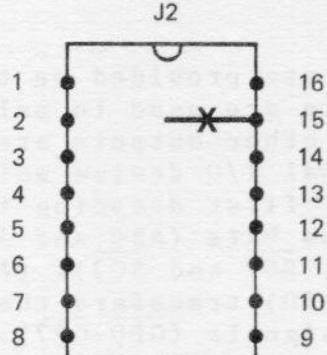


Notice that the CTC, PIO and USART are in GP5, GP6 and GP7 respectively. If, for example, Z=1, the CTC will respond to addresses 14H, 15H, 16H and 17H. The PIO will respond to 18H, 19H, 1AH and 1BH, while the USART will respond only to 1EH and 1FH.

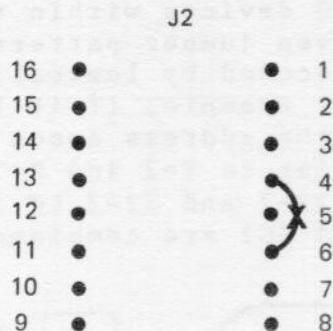
Note: The factory shipped card will have the following points jumpered to configure I/O decode from COH to DFH: J2-5 to J2-15, J2-2 to J2-7 and J2-4 to J2-6. The CTC address is, therefore, D4H-D7H, the PIO address is D8H-DBH and the 8251 address is DEH-DFH.

In order to reconfigure I/O decode logic it will be necessary to cut PC etch.

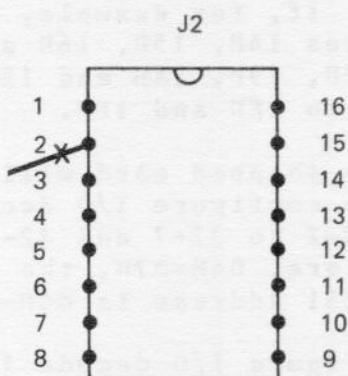
To free J2-5 cut etch on component side at point shown below:



To free J2-6 cut etch on solder side at point shown below:



To free J2-7 cut etch on component side at point shown below:



PORT GROUP DECODE		
Y	Z	JUMPER PINS OF J2
0	1	5-15, 1-7, 3-6
2	3	5-16, 1-7, 3-6
4	5	5-15, 2-7, 3-6
6	7	5-16, 2-7, 3-6
8	9	5-15, 1-7, 4-6
A	B	5-16, 1-7, 4-6
C	D	5-15, 2-7, 4-6
E	F	5-16, 2-7, 4-6

I/O PORT DECODE				
	SG0	SG1	SG2	SG3
GP0	Y0	Y1	Y2	Y3
GP1	Y4	Y5	Y6	Y7
GP2	Y8	Y9	YA	YB
GP3	YC	YD	YE	YF
GP4	Z0	Z1	Z2	Z3
GP5	Z4	Z5	Z6	Z7
GP6	Z8	Z9	ZA	ZB
GP7	ZC	ZD	ZE	ZF

Note: MDC uses GP4 and GP3/S63 (CF,D0,D1,D2,D3)

TABLE 2 I/O PORT DECODE MATPIX

7.0 DMA TRANSFERS AND DATA BUS CONTROL

It is possible to transfer data directly to and from MCB memory or external memory without the need for CPU control. This process (DMA - direct memory access) is initiated by a bus request (EUSRQ pulled low.) The Z80-CPU will complete the current instruction and issue a bus acknowledge (BUSAK). The drivers on the address lines (A3, A21, A9), on the data lines (A28, A29) and on output control lines (A20) will go to a high impedance state. The external circuitry making the request is now allowed full control over the memory. Notice that output control lines BUSAK and HALT are always enabled and never enter the high impedance state (See sheet 1 of schematic).

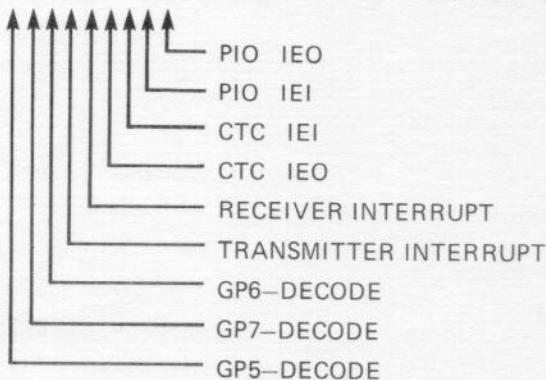
The MCP will control the data bus drivers A28 and A29 to allow the DMA Controller access to the internal data bus (1DB0-7). (During DMA transfer the Z80-CPU data bus lines are in the high impedance state.) This data bus control is obtained from two decoding PROMS (6306's), the outputs of which are programmed function of the inputs. The 'DB CNTL' and 'DB ENA' lines from the PROM (A32) control the 3216 driver/receivers (A28, A29) between the Z80-CPU, the internal data bus and the external data bus. Tables 3 and 4 identify the state of each PROM output for the associated input. The five most significant input address lines are shown in binary format along the vertical axis. The four least significant address inputs (XXXX on the vertical axis) are identified in hexadecimal notation along the horizontal axis. For each unique address input, the corresponding hex code for the output is found at the crosspoints between the two axes. For example, the PROM in location A32 will produce a low on the 'DB CNTL' output for an input address of 05H which is found at the intersection of 00000XXXX and 5.

Consideration needs to be given to dynamic RAM refresh during DMA transfers. Since refresh depends only on 64 RAS cycles occurring every 2ms, the user has considerable time in which to perform a DMA transfer without being concerned with external refresh logic. And if the user can guarantee a memory cycle at each of the 64 row address within this 2ms time frame, no refresh is required.

TABLE 3

ADDRESS

TNPUTS	0	1	2	3	4	5	6	7	R	9	4	H	C	D	F	F
00000XXXX	1	1	3	1	3	3	3	3	1	1	3	1	1	1	3	1
00001XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
00010XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
00011XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
00100XXXX	1	1	3	1	3	3	3	3	1	1	3	1	1	1	3	1
00101XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
00110XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
00111XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
01000XXXX	1	1	3	1	3	3	3	3	1	1	3	1	1	1	3	1
01001XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
01010XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
01011XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
01100XXXX	1	1	3	1	3	3	3	3	1	1	3	1	1	1	3	1
01101XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
01110XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
01111XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
10000XXXX	1	1	3	1	3	3	3	3	1	1	3	1	1	1	3	1
10001XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
10010XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
10011XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
10100XXXX	1	1	3	1	3	3	3	3	1	1	3	1	1	1	3	1
10101XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
10110XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
10111XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
11000XXXX	1	1	3	1	3	3	3	3	1	1	3	1	1	1	3	1
11001XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
11010XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
11011XXXX	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
11100XXXX	0	0	2	0	2	2	2	2	0	0	2	0	0	0	2	0
11101XXXX	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
11110XXXX	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
11111XXXX	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2



The table above lists the contents of the MCB bus control PROM. This PROM is located at A33 on the MCB. Of the four PROM outputs, only O1 and O2 are used. O1 is the least significant bit of the PROM.

TABLE 4

ADDRESS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
INPUTS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000XXXX	0	0	0	0	0	-0-	0	0	0	0	0	0	0	0	0	0
00001XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00010XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00011XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00100XXXX	0	-0-	0	0	0	0	-0-	0	0	0	0	0	0	0	0	0
00101XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00110XXXX	0	-0-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00111XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01000XXXX	0	0	0	0	0	0	0	0	-0-	0	0	0	0	0	0	0
01001XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01010XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01011XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01100XXXX	0	0	0	0	0	0	0	0	-0-	0	0	0	0	0	0	0
01101XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01110XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01111XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10000XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10001XXXX	0	0	0	0	0	0	0	0	-0-	0	0	0	0	0	0	0
10010XXXX	0	0	0	0	0	0	0	0	-0-	0	0	0	0	0	0	0
10011XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10100XXXX	0	-0-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10101XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10110XXXX	0	-0-	0	0	0	0	-0-	0	0	0	0	0	0	0	0	0
10111XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11000XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11001XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11010XXXX	0	0	0	0	0	0	-0-	0	0	0	0	0	0	0	0	0
11011XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11100XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11101XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11110XXXX	0	0	0	0	0	0	-0-	0	0	0	0	0	0	0	0	0
11111XXXX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

M1-

IORQ-

RD-

BUSAK

RAM SEL-

ROM SEL-

02 OF PROM A33 (33-0053-14)

01 OF PROM A33

NOT USED (GROUNDED)

The table above lists the contents of the MCB bus control PROM. This PROM is located at A32 on the MCB. Of the four PROM outputs, only 03 and 04 are used. 04 is the most significant bit of the PROM output.

8.0 MCR PARALLEL INTERFACE

The MCR provides a software-programmable, two port parallel I/O device (PIO) for standard hardware interface between peripheral devices and the Z80 CPU. The PIO contains two independent 8 bit ports with full handshake control that can be configured by the CPU to operate in any of 4 major modes.

In the output mode (Mode 0), data is written to the ports from the Z80 and onto the port data bus. In the input mode (Mode 1), the peripheral device supplies data to the port. The bidirectional mode (Mode 2) allows one port (Port A) to be bidirectional using the handshake signals from the other port. The control mode (Mode 3) allows for direct bit set and reset capability. This mode also allows any bit in either port to be individually programmed to be either an input bit or an output bit. Vectored Interrupt communication with the CPU is included to facilitate data transfer between the peripheral device. A unique feature of the PIO is that it can be programmed to interrupt the Z80-CPU on the occurrence of specified status conditions in the peripheral device. One port has the ability to source a minimum of 1.5ma of current at 1.5 volts allowing darlington transistors to be directly driven (for printer and high voltage displays, for example).

The I/O Ports are bussed to a set of wire wrap pins that can then be wired in any desired configuration to four uncommitted 16 pin sockets (A49, A50, A51, A52). The user can provide termination for input pins or drivers for outputs pins and can also wire from the 16 pin sockets over to the edge connector.

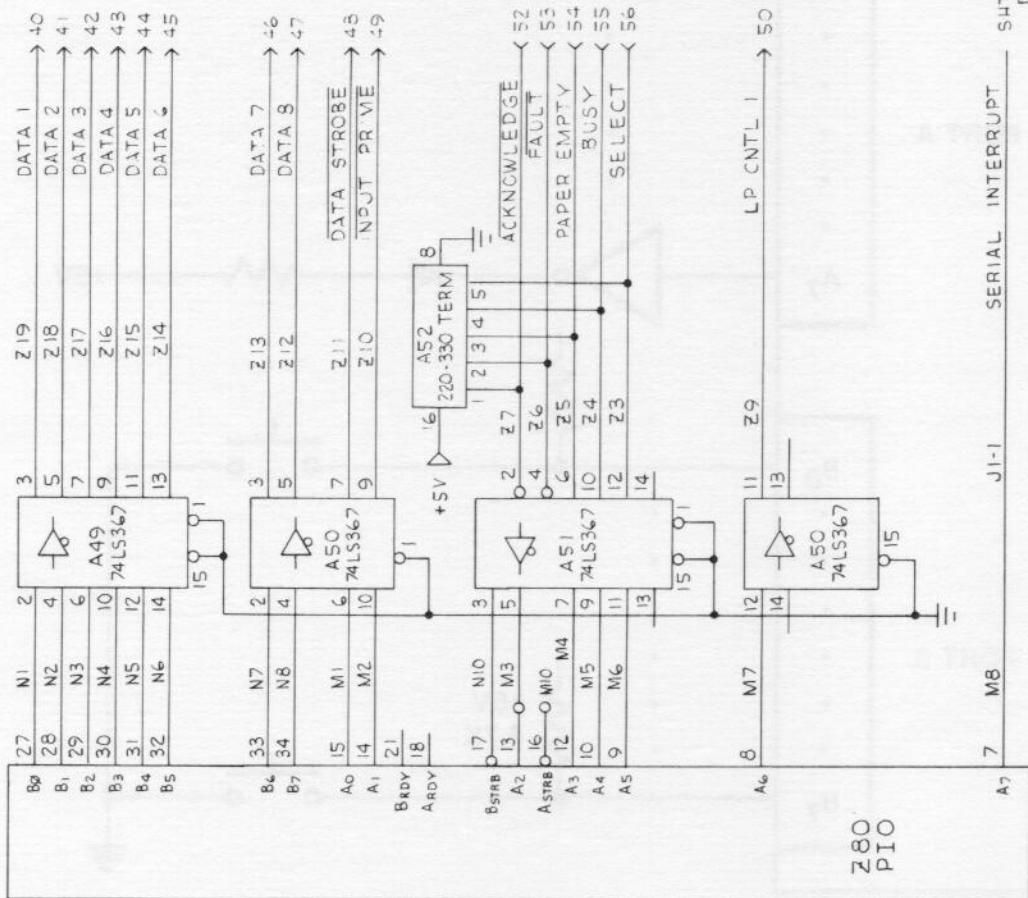
In the following examples, the PIO is used to interface the Z80-CPU to a Centronics line printer (model 306) (Figure 3) and to configure a simple switch input and display output system (Figure 4).

In the second example, the PIO can be programmed to interrupt the CPU whenever a push button is activated. The following statements are used to program Port A:

```
3E12      1      LD A,12H ;EXAMPLE VECTOR
          2
D340      3      OUT (PIO),A ;LOAD INTERRUPT VECTOR
          4
3EFF      5      LD A,0FFH ;SET PIO MODE
          6
D340      7      OUT (PIO),A ;LOAD BIT CONTROL MODE 3
          8
D340      9      OUT (PIO),A ;SET I/O, ALL LINES INPUT
         10
3E87     11      LD A,87H ;SET INTERRUPT CONTROL WORD
         12
D340     13      OUT (PIO),A ;LOAD CONTROL WORD
         14
         15  PIO EQU 40H
```

CONFIDENTIAL
This document is the property of Zilog Inc. or its corporate affiliates
and is confidential and is loaned to you by Zilog Inc. No part of this document may
be copied, reproduced or disclosed to third parties without the prior written
consent of Zilog Incorporated.

LTR	DESCRIPTION	DATE	APPROVED
REVISIONS			



UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES		SIGNATURE AND DATE	
ANGLES $\pm 1^\circ$ TOLERANCES DRAWN <i>John D. H. Jones</i> 7/27/76 DRAFTED <i>J. D. H. Jones</i> 7/27/76 APPROVED <i>Zilog Inc.</i> 7/27/76 2 PLCE DEC 2.0% 2 PLC DEC 2.0%		TITLE LOGIC DIAGRAM MCB - LINE PRINTER INTERFACE (CENTRONICS) SIZE DRAWING NO C SCALE NONE ISSUE SHEET 1 OF 1	
NEXT ASSY	USED ON	FINISH	MATL
APPLICATION			

Figure 3

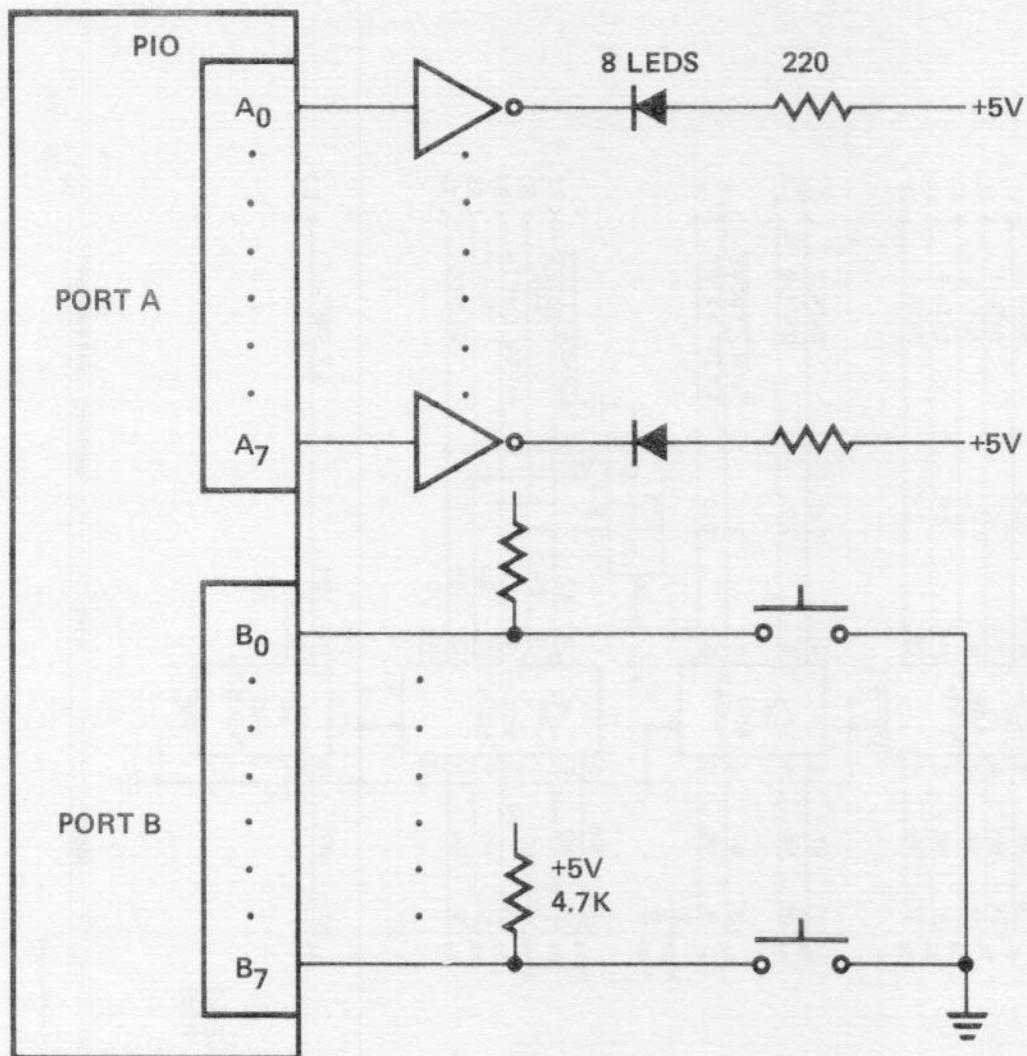


Figure 4 Push Button Switch/Display/PIO Interface

9.0 MCB SERIAL INTERFACE

The MCB supports both synchronous and asynchronous serial communication formats. The serial I/O interface uses the 8251 USART, which is programmable by means of a control byte loaded by the Z80 CPU. Controllable parameters include bits per character (5,6,7 or 8), number of stop bits for asynchronous operation (1.0, 1.5, 2.0), parity insertion/checking (odd, even or none) and clocking of transmitted or received data, at frequencies of 1, 16 or 64 times the data rates. The USART accepts data characters from the Z80 in a parallel format and converts them into a continuous serial data stream for transmission. It can simultaneously receive serial data streams and convert them into parallel data characters for the Z80. The user has the option, through jumper connections, of configuring the interface as RS232 compatible, or as a teletype compatible current loop interface. Additional information can be obtained from selected 8251 documentation.

One of the four channels of the CTC (Channel 1) is used in conjunction with the USART to generate a baud rate clock. The software configures the CTC to provide a clock signal at 32 times the desired communication serial frequency. A divide by 2 circuit (A2) is used to convert the pulse signal from the CTC (A35, pin 8) into a 50% duty cycle signal required by the USART.

Note: Channel 0 of the CTC has wiring options that allow it to be used with the Memory/Disk Controller Card (Z80-MDC) for disk timing functions. Channels 2 and 3 are unused and are available for user operations.

Switches are available on the board that can be read by the CPU software to set the communication frequency to any of 14 common rates. (The switches can also be used for other functions at the User's discretion.)

Table 3 indicates the available baud rates and the switch settings for each that is compatible with the monitor software.

$$\phi = 2.4576 \text{MHz} \quad f_s = \phi / 2N$$

TABLE 5 BAUD RATE SELECTION

BAUD RATE	f_s 16XPATE	SWITCHES				N (DIVISOR)
		S1	S2	S3	S4	
50	800HZ	ON	ON	ON	ON	96X16
75	1.2KHZ	OFF	ON	ON	ON	64X16
(TTY)110	1.745KHZ	ON	OFF	ON	ON	44X16
134.5	2.133KHZ	OFF	OFF	ON	ON	36X16
150	2.4KHZ	ON	ON	OFF	ON	32X16
200	3.2KHZ	OFF	ON	OFF	ON	24X16
300	4.8KHZ	ON	OFF	OFF	ON	16X16
600	9.6KHZ	OFF	OFF	OFF	ON	8X16
1200	19.2KHZ	ON	ON	ON	OFF	4X16
2400	38.4KHZ	OFF	ON	ON	OFF	2X16
4800	76.8KHZ	ON	OFF	ON	OFF	1X16
9600	153.6KHZ	OFF	OFF	ON	OFF	4*
19200	307.2KHZ	ON	ON	OFF	OFF	2*
38400	614.4KHZ	OFF	ON	OFF	OFF	1*
AUTO*	AUTO	OFF	OFF	OFF	OFF	AUTO**

f_s (16X baud rate) is the serial clock presented to the transmit clock (TXC) and receive clock (RXC) of the 8251. (Jumpers J4-7 to J4-4 to J4-16. See sheet 6 of schematic.)

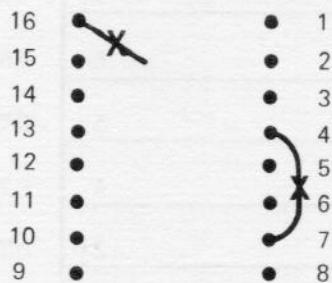
* Jumper 18 to 118 for high speed serial.

** Special software support required.

It is possible to allow external clocking for the TXC and RXC lines instead of clocking from the CTC, by jumpering J4-9 to J4-7 for the TXC clock and jumpering J4-8 to J4-4 for the RXC clock. Note: The card as shipped from the factory has the CTC clock (Serial clock) tied to TXC and RXC.

To free J4-7 from J4-4 and from J4-16, cut etch on solder side at the following point shown below:

J4



The USART can assume the role of a modem or a data processing terminal in the User's system. Pin definition for the two configurations are given in Figures 5 and 6.

MCB		TERMINAL
XMITED DATA	15	2
DATA TFPN RDY	76	20
REQ TO SEND	14	4
CLE TO SEND	11	5
DATA SET RDY	74	6
RECV DATA	7	3
LINE SIG DET	80	8
XMITED DATA		
PTP		
RTS		
CTS		
PSR		
PFRCV DATA		
LSD		

FIGURE 5 MCB TO TERMINAL (RS-232)

Note: Figure 5 represents the 'as shipped' configuration with 'Clear to Send' always enabled and 'Request to Send' always ignored.

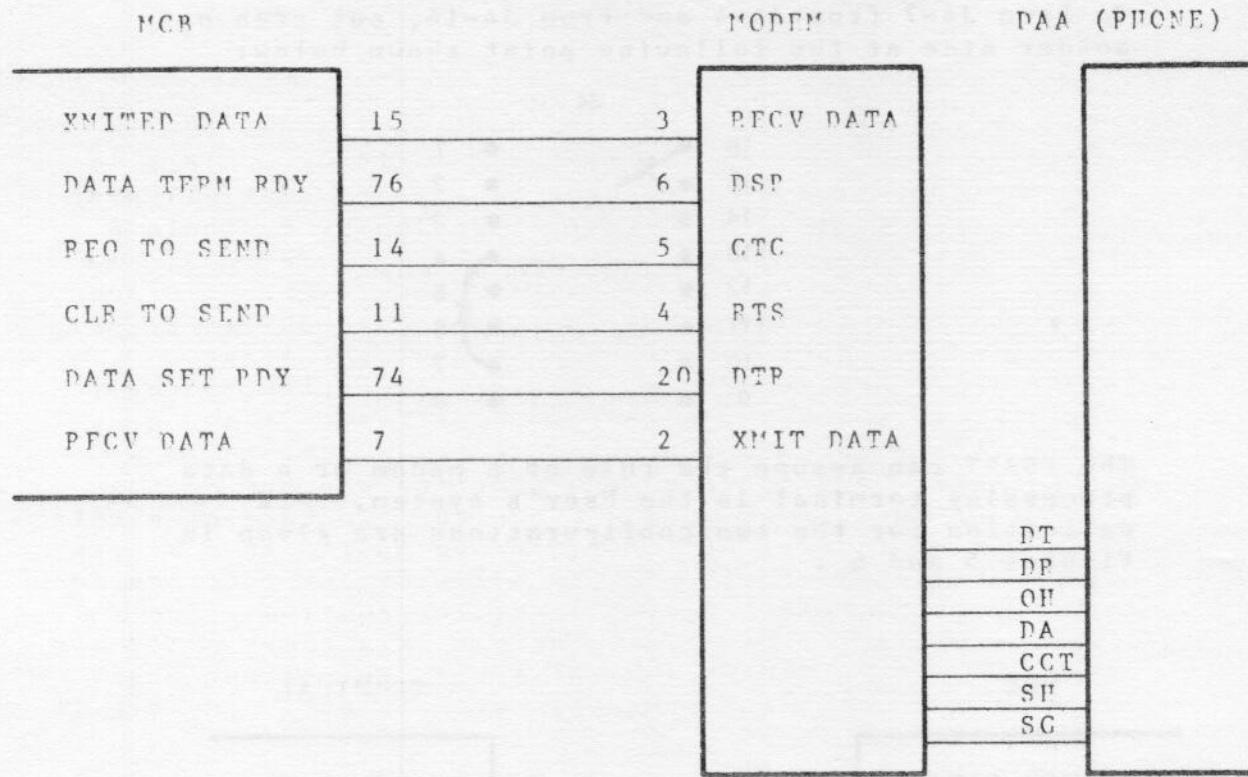


FIGURE 6 MCR TO MODEM

Note: To configure the system in Figure 6, the following points on solder side of card will need to be cut on J4:

J4

16	•	•	1
15	*	•	2
14	*	•	3
13	•	•	4
12	•	*	5
11	•	*	6
10	•	•	7
9	•	•	8

The following jumpers must then be applied:

J4-13 to J4-6
J4-5 to J4-14

10.0 MCB INTERRUPT STRUCTURE

The CTC and PIO on the MCB are designed to use the vectored interrupt capability of the Z80-CPU. This is the most powerful of the three interrupt modes allowing an indirect call to any memory location by a single 8 bit vector supplied from the peripheral. In this mode (mode 2) the peripheral generating the interrupt places the vector on the data bus in response to an interrupt acknowledge. This vector then becomes the least significant 8 bits of the indirect pointer while the I register in the CPU provides the most significant 8 bits. This pointer in turn points to an address in a jump table which contains the starting address of the interrupt routine. Figure 7 shows the sequence of events for vector processing. Interrupt processing thus starts at an arbitrary 16 bit address allowing any location in memory to be the start of the service routine. The least significant bit of the vector is automatically set to a '0' within the peripheral since the pointer must point to two adjacent memory locations for a complete 16 bit address.

Priority of interrupts is determined by serially connecting the devices. Two lines (Interrupt Enable In [IEI] and Interrupt Enable Out [IEO]) are provided in each peripheral to form a daisy chain configuration between devices. Figure 8 shows the CTC and PIO in such a configuration. The IEI of the CTC is tied to +5 to indicate that it has the highest priority. The PIO is the second highest priority device with its IEI tied to the IEO of the CTC. It would be possible to connect external peripherals to this chain by tieing the IEI of the next priority device to the IEO of the PIO (pin 111 of the MCB. Both IEI and IFO lines of the CTC and PIO are routed to edge connector pins). For a device to generate interrupts its IEI line must be high.

For example, if the CTC in Figure 8 needs service, (any of the four channels) it will generate an interrupt. It pulls low on the IEO line blocking any interrupts from downstream devices until the CTC service routine has been completed. When it receives an interrupt acknowledge ($\overline{M1}$ low and \overline{IOFO} low), it places the service vector address on the bus. Note that interrupt enable status must have been resolved before the fall of \overline{IOFO} (with $\overline{M1}$ low) and during the interrupt acknowledge time frame, no new interrupt requests can be generated.

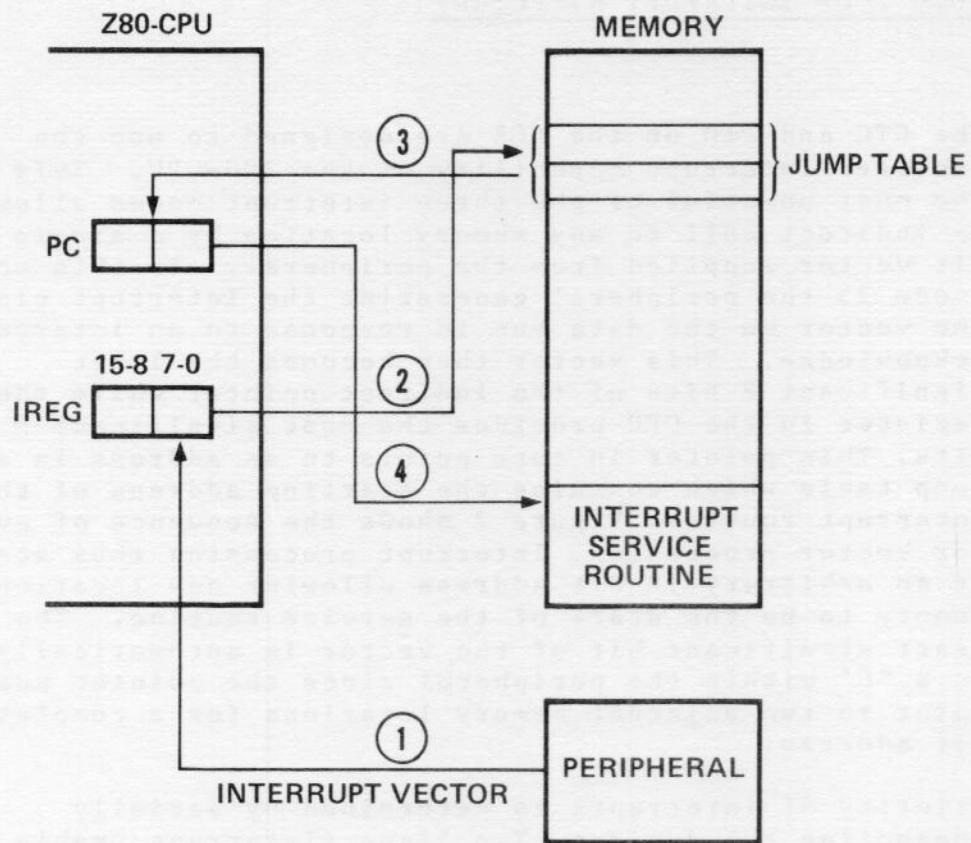


Figure 7 Vector Processing Sequence

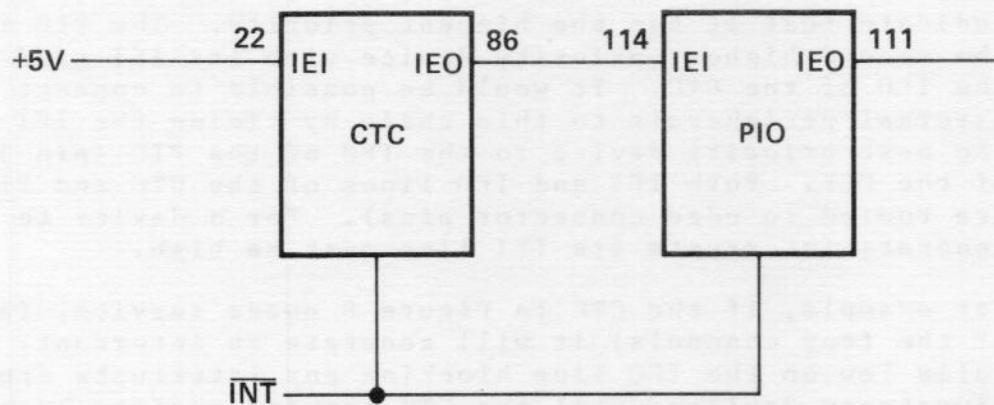


Figure 8 CTC/PIO Priority Chain

The IFO line of any given peripheral will satisfy the following equation:

$$IEO = IFI + \overline{HELP}$$

Where HELP is an internal peripheral signal indicating that the device needs service. Each device will propagate to the next downstream device with a low on its IFO when IFI goes low.

Interrupt nesting to any level is possible with this interrupt structure. The return from interrupt instruction (PFTI) facilitates this nesting, allowing higher priority devices to temporarily suspend service of lower priority service routines. After the interrupt service routine has been completed, the PFTI (Return from Interrupt) instruction is used to restore the contents of the Program Counter (PC) (by popping the stack) and to reset the HELP logic of the highest priority device just having been serviced.

10.1 NESTING WITH INTERRUPTS ENABLED

To illustrate the vectored interrupt and nesting mechanism, a typical interrupt processing routine will be detailed on the configuration in Figure 8 . The sequence of events is as follows:

1. With interrupts enabled, the PIO (either port) generates an interrupt (INT, HELP and IEO all go to a logic zero).
2. The Z80-CPU finishes the current instruction and responds with an interrupt acknowledge (M1 and IORQ low), and reads in the interrupt vector. (See Figure 9). The contents of the Program Counter (PC) are stored in the external stack and the enable flip flop (IFF1 and IFF2) is reset inhibiting further interrupts. (Until re-enabled by an EI instruction.)
3. While in the service routine for the PIO (with interrupts having been enabled (IFF1=1)) the CTC generates an interrupt (INT, HELP and IEO all go to a logic zero; see Figure 9). The IFI of the PIO is now low.
4. The Z80-CPU finishes the current instruction of the PIO service routine, responds with an interrupt

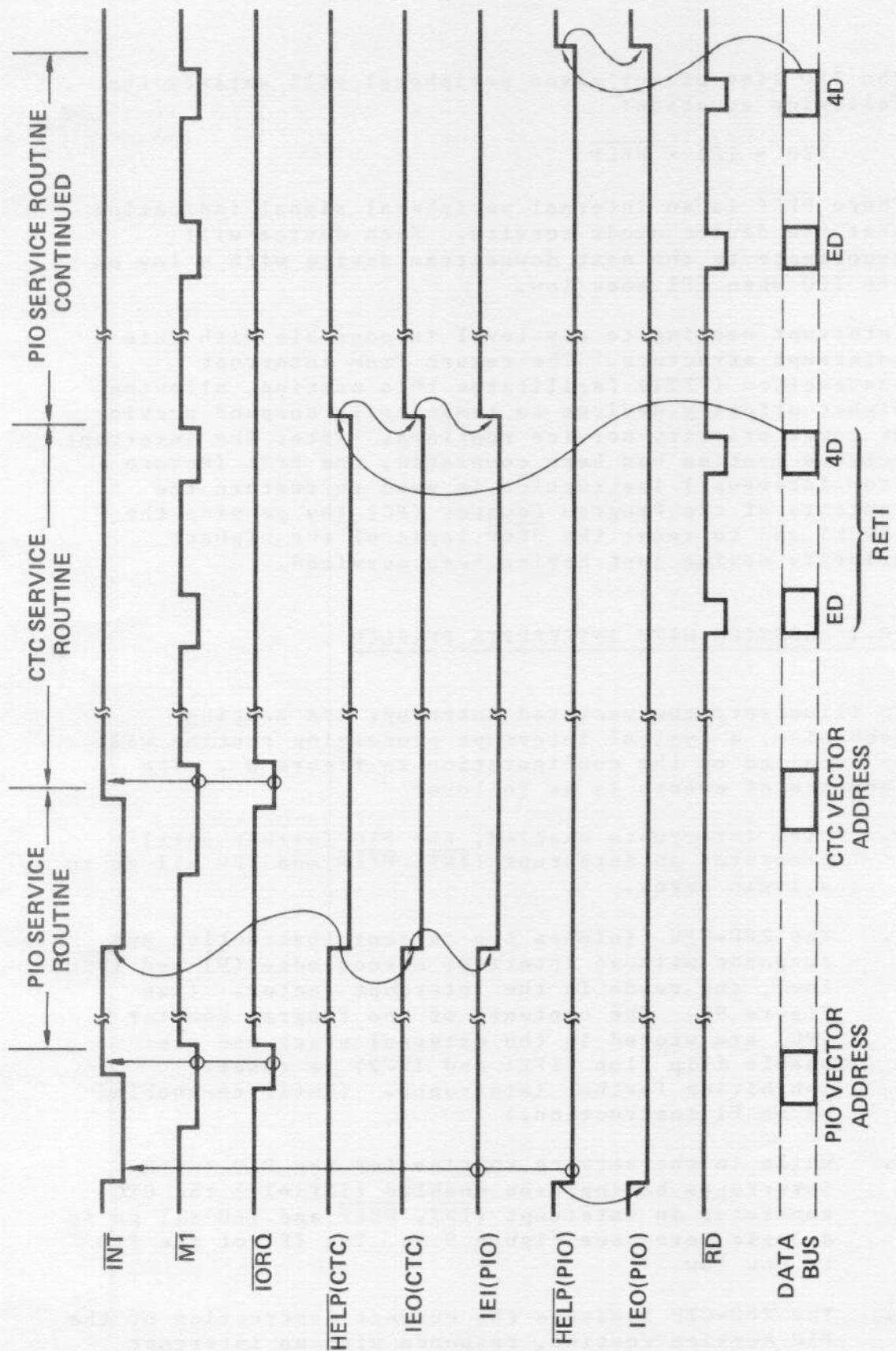


Figure 9 CTC/PIO Interrupt Timing With Interrupts Enabled

NOTE: $IEO = IEI \cdot \overline{HELP}$

acknowledge and reads in the interrupt vector. The contents of the PC are stored in the external stack and IFF1 is once again reset to logic 0. The stack now looks like:



5. The CTC service routine is completed and an RETI instruction (ED 4D) resets the CTC HELP logic. The CTC IEO and the PIO IEI go high. The stack is popped restoring the contents of the PC to the PIO service routine. Note that prior to the PETI instruction, an EI instruction can be executed to enable interrupts for the remaining segment of the PIO service routine.
6. The PIO service routine is completed and another RETI instruction (ED 4D) resets the HELP logic and the IEO line. The stack is popped again restoring the PC to the main line program.

10.2 PRIORITY STRUCTURE WITH INTERRUPTS DISABLED

A variation to the above example results when interrupts are disabled during the PIO service routine. Now if the CTC needs service (HELP logic low with PIO low), the INT line will be pulled low but the CPU will not respond. (IFF1=0) However, when an RETI is issued for the PIO routine, the CTC HELP logic must 'not' be reset. Therefore, the CTC will allow its IEO line to go high for one M1 cycle during the RETI instruction (see Figure 10). In other words, if an interrupt acknowledge is not given to a device requesting service, its IEO line will be forced high for one M1 cycle after decoding ED (first byte of RFTI) to allow down stream devices to decode PFTI. Note again that prior to the RETI, an EI should be executed to allow the CTC interrupt access after the PIO service routine.

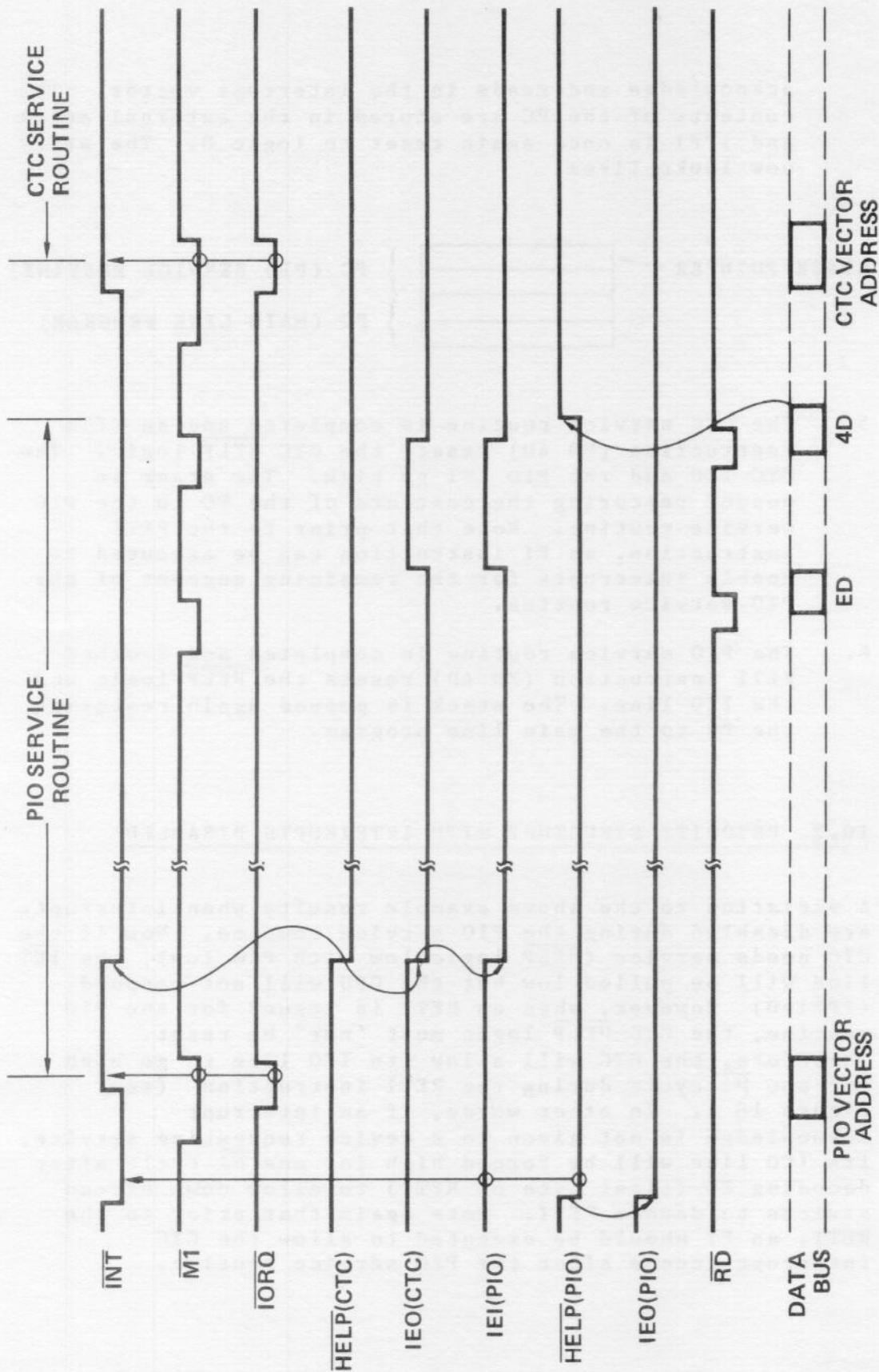


Figure 10 CTC/PIO Interrupt Timing with Interrupts Disabled

11.0 A MINIMAL PROTOTYPE CONFIGURATION

The MCB can be used with a minimal amount of additional hardware to configure a basic prototyping or evaluation system. Monitor software for system development and debugging is available in 1/2K and 1K versions. This software resides in bipolar PPOMS and can be inserted into the 24 pin sockets above the RAM on the MCB. The 1/2K version requires a single PROM (MCB1) placed in socket A4. The 1K (2PROM) version requires MCB2 be placed in A4 and MCB3 be placed in A5. The 1K (1PROM) version requires MCB4 be placed in A4. This debug software is strapped to operate in the first page of memory in the following configuration:

MONITOR VERSION	PROM(S)	ADDRESS SPACE
1/2K	1-512 X 8	0-1FFFH
1K	2-512 X 8	0-1FFFH 4000H-5FFFH
1K	1-1K X 8	0-3FF

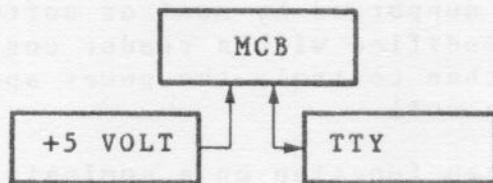
Since each PROM socket is wired for 1K X 8 devices, the 512 X 8 elements will ignore address line A9. When the 1/2K byte version is used (MCB1), the high order 1/2K byte of memory space duplicates the low order half in the first 1K segment. Likewise, two 512 X 8 PROMS for the 1K version (MCB2,3) will occupy the memory space from 0 to 7FF as indicated in Table 4. The 1K, 1 PROM version (MCB4) will reside in the memory space from 0 to 3FF.

ADDR BIT A11	ADDR BIT A10	ADDR BIT A9	ADDR BIT HEX	1/2K 1PROM 1MCB1	1K 2PROMS MCB2, 3	1K 1PROM MCB4
			7FF		MCB3	
0	1	1	600		MCB3	
			5FF		MCB3	
0	1	0	400		MCB2	
			3FF	MCB1	MCB2	
0	0	1	200			MCB4
			1FF		MCB1	MCB2
0	0	0	000			

TABLE 6 MONITOR SOFTWARE PAGE 0 MEMORY MAP

(A15, A14, A13, A12=0)

The following configuration shows the MCB in a typical prototyping configuration.



To configure the MCB for teletype terminals it will be necessary to cut two PC traces. The following points must be cut:

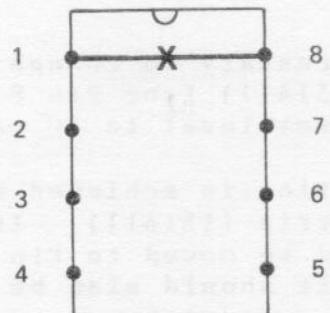
On the solder side of J4:

J4

16	•	*	1
15	•	•	2
14	•	•	3
13	•	•	4
12	•	•	5
11	•	•	6
10	•	•	7
9	•	•	8

On the component side of J5:

J5



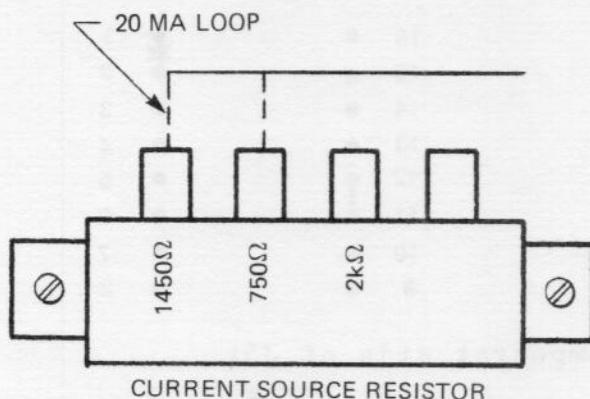
Jumpers are now applied between: J4-8 to J4-2 and J5-1 to J5-6.

It may be necessary to modify the TTY for operation with the MCB. The standard ASR TTY must be configured to operate in the full duplex mode (echoing the transmitted character) utilizing two 20 ma current loops. The first loop is used to transmit TTY keyboard data to the MCB receive input. The second loop is used to print data from the MCB transmit output. Data transmission is performed at a rate of 10 characters per second (110

baud).

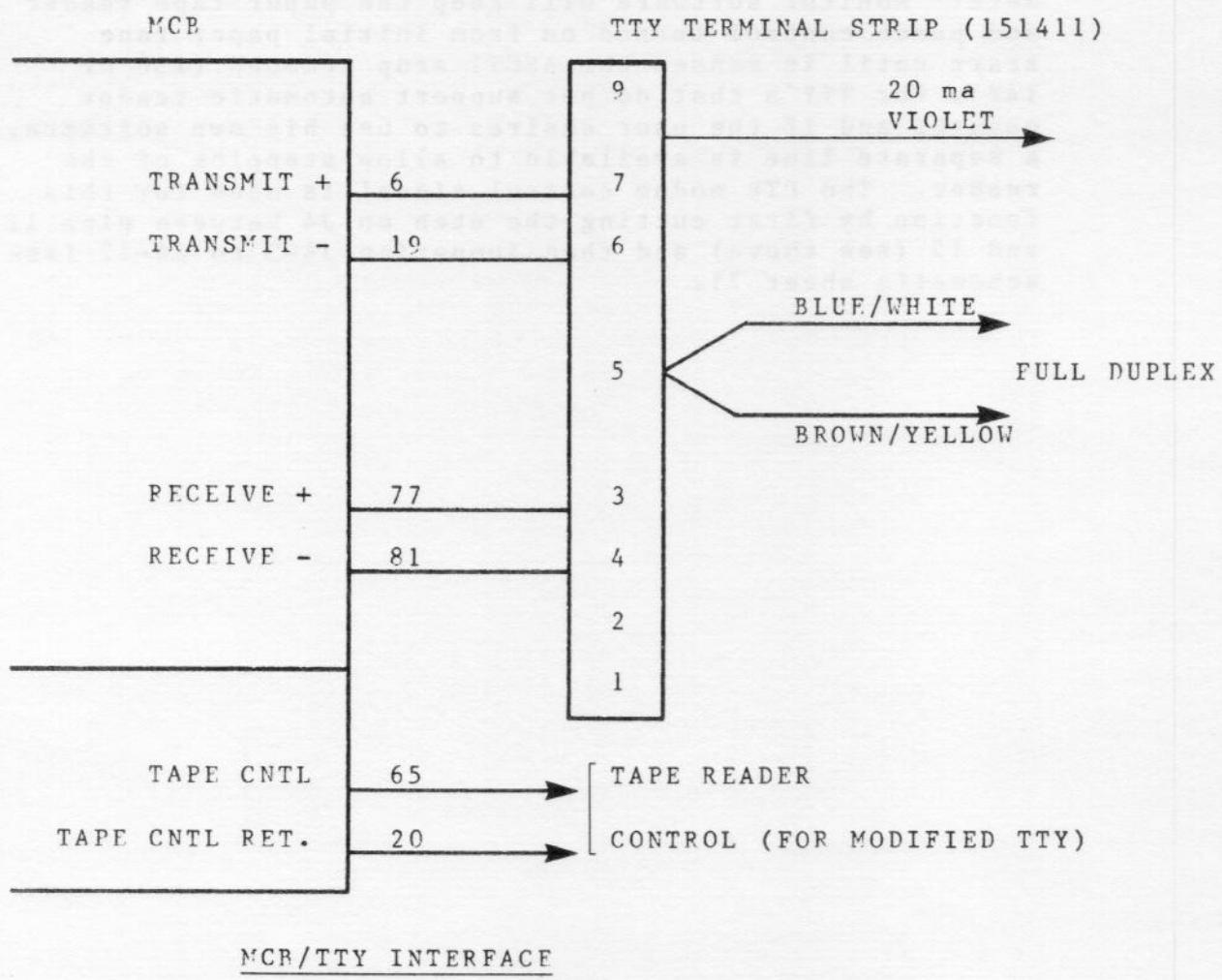
If the tape reader is to be externally controlled by the MCR (not supported by monitor software, however) the TTY must be modified with a reader control relay circuit. The CPU then controls the power applied to the relay's switching coil.

The TTY can function on a nominal loop current of either 60 ma or 20 ma. To make the selection of loop current, a wire is shifted in the TTY's electrical service unit. This will change the current resistor from 750 OHMS (60 ma mode) to 1450 OHMS (20 ma mode). This resistor is tapped and conversion between the two modes is accomplished by shifting the push-on connector between positions as indicated below:



It may also be necessary to change a violet wire on the terminal strip (151411) from Pin 8 to Pin 9 to condition the receiver current level to 20 ma.

Full-duplex operation is achieved by shifting two wires on the terminal strip (151411). If a blue/white wire is on Pin 4 it should be moved to Pin 5. If a brown/yellow wire is on Pin 3 it should also be moved to Pin 5.



Note: Monitor software will keep the paper tape reader and punch control turned on from initial paper tape start until it senses the ASCII stop command (13H or 14H). For TTY's that do not support automatic reader control and if the user desires to use his own software, a separate line is available to allow stepping of the reader. The DTR modem control signal is used for this function by first cutting the etch on J4 between pins 11 and 12 (see above) and then jumpering J4-3 to J4-12 (see schematic sheet 7).

11.1 PROGRAMMING THE CTC AND USART

The monitor software uses D5H for the CTC device address, DFF for the USART address and DD to address TTY speed control logic.

At power up or as a result of a reset, the software will program both the CTC and the USART, to establish the rate and format of data transfer.

The software will read the state of the switches (via A47) to select the proper time constant needed by the CTC to operate as a baud rate generator. The software statement is:

```
DB DD    IN A, (SPEED) ;GET TTY SPEED SETTING
```

For a 110 baud TTY, the switches are set as follows:

S1, S3, S4	ON (GND)
S2	OFF (+)

The CTC is programmed first with two data transfers. The first transfer sets the operating mode: (timer mode)

```
D3 D5    OUT    (CTC1),A    A=07H
```

The second transfer will load the time constant:

```
D3 D5    OUT    (CTC1),A    A=0DH
```

The software configures the CTC to provide a clock signal at 32 times the desired communications serial frequency. A divide by 2 circuit (A2) will convert the pulse signal from the CTC into a 16X, 50% duty cycle signal required by the USART.

For transfer speeds above 4800 baud, (9600 to 38,400 baud), the software will program the CTC to operate in the counter mode with an external clock:

```
D3 D5    OUT    (CTC1),A    A=47H
```

The clock is obtained by jumpering pin 18 (CTC, CT2) to pin 118 (1/2 ϕ clock).

The USART is also programmed with a two byte transfer. The first transfer sets the mode:

```
D3 DF OUT (TTYCNT),A A=CEH
```

For TTY operation the USART is configured to operate asynchronously with a character length of 8 bits at a baud rate factor of 16X with two stop bits inserted after character transfer.

The second transfer sets the command:

```
D3 DF OUT (TTYCNT),A A=27H
```

The USART will now do a normal operation with the receiver and transmitter enabled and RTS and DTR forced to zero.

Note that monitor software on the MCR can be switched out of the memory address space to facilitate software development. A jumper between pin 92 and pin 83 will be required to enable on card ROM. Logic can be inserted between these two points to allow software generation starting at location 0. For example, the user can generate code into external memory (on card RAM must be in page 1 to accomodate monitor storage requirements) starting at location 0, then switch out the monitor software, reset the MCR and start program execution from user location 0.

12.0 MEMORY EXPANSION

Since the 16K X 1 RAM element is pin compatible with the 4K X 1 element, the MCB is easily upgraded to a 16K byte RAM system. This compatibility between the two memory elements depends on making the 4K chip select input the equivalent of the seventh multiplexed address on the 16K chip. (For 16K RAMS, 14 address bits needed to address 1 of 16,384 bits are multiplexed into 7 address inputs.) Strapping options are provided on the MCR to facilitate this expansion. For 4K systems the chip select (CS-PIN 13) is grounded (J1-11 and J1-9 to J1-7). For 16K systems, pin 13 becomes the seventh multiplexed address (A6) and the two extra address lines (AP12 and AB13) are applied to the multiplexer with jumpers between J1-10 and J1-12, J1-9 and J1-16, J1-8 and J1-11.

To configure a 16K system, it will be necessary to cut etch on the card. To free J1-12, cut etch on solder side at point shown below:

J1

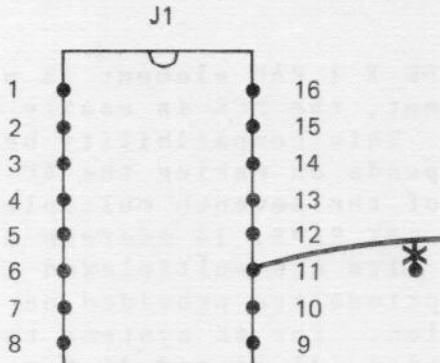
16	●	● 1
15	●	● 2
14	●	● 3
13	●	● 4
12	●	● 5
11	●	● 6
10	●	● 7
9	●	● 8

To free J1-9 from ground, cut etch on solder side at:

J1

16	●	● 1
15	●	● 2
14	●	● 3
13	●	● 4
12	●	● 5
11	●	● 6
10	●	● 7
9	●	● 8

To free J1-11 from ground, cut etch on component side
at:



Notice that a 16K memory bank must start at a 16K memory boundary (decoder dependent). These boundaries are 0000, 4000H, 8000H and C000H. Also, only one line from the Page Select Decoder need be connected. For example, to place the RAM at a starting address of 4000H, a jumper is placed between J3-9,11 and J3-12 (see Table 1).

13.0 SYSTEM EXPANSION

The flexibility of the MCB allows this card to be the basic building block for a complete line of microcomputer related products. Included in this line is a Memory Disk Controller (MDC), a video interface card (VDB), a 16K RAM/8K PPROM Memory Card (RMP), both serial and parallel I/O Boards (SIP and IOB), and analog boards (AIO, AI^R).

The MDC formats and adjusts the Data rate between a standard Shugart 800 floppy disk and the CPU. The 3K monitor firmware used with the MCB provides the basic debugging commands, input/output control and bootstrap portions of a floppy disk based operating system. Figure 11 details the interface signals between the MCB, MDC and floppy disk.

The VDB allows the MCB to be interfaced to a video monitor, for the display of either printed or graphic data. 2K bytes of memory on the MCB is used as the video refresh buffer while 256 bytes of RAM on the VDB is used as a line buffer. Figure 12 details the interface signals between the MCB, VDB, TV monitor and an ASCII keyboard.

Complete microcomputer systems (Z80-MCS and Z80-PDS) are available from Zilog which incorporates the MCB and MDC with a floppy disk and power supply into a single chassis.

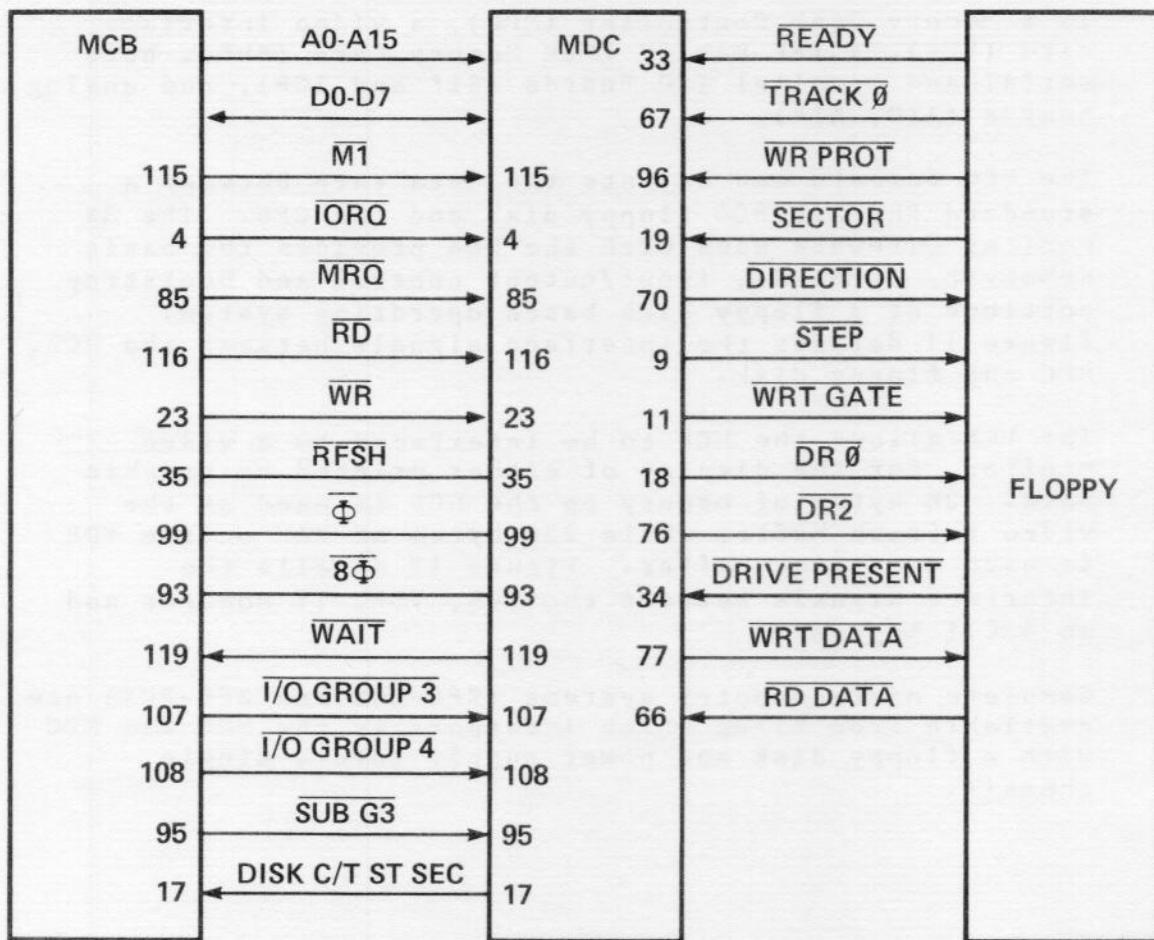


Figure 11 MCB/MDC/Floppy Interface

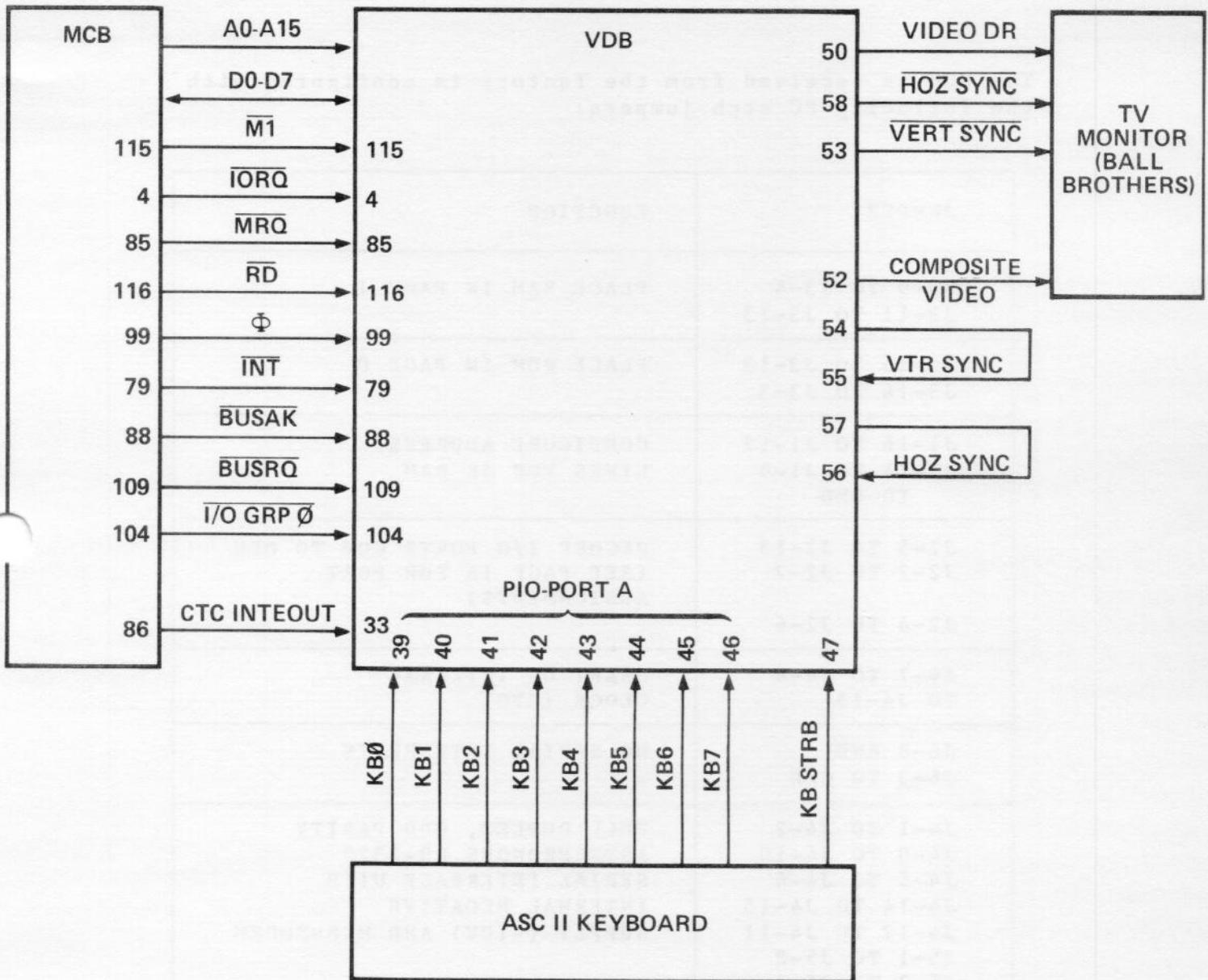


Figure 12 MCB/VDB/TV Monitor/Keyboard Interface

14.0 HOOKING UP THE MCB

The MCB as received from the factory is configured with the following PC etch jumpers:

JUMPER	FUNCTION
J3-9 TO J3-4 J3-11 TO J3-13	PLACE RAM IN PAGE 1
J3-15 TO J3-13 J3-16 TO J3-3	PLACE ROM IN PAGE 0
J1-16 TO J1-12 J1-11 TO J1-9 TO GND	CONFIGURE ADDRESS LINES FOR 4K RAM
J2-5 TO J2-15 J2-2 TO J2-7 J2-4 TO J2-6	DECODE I/O PORTS COH TO DFH (SEE PAGE 16 FOR PORT ASSIGNMENTS)
J4-7 TO J4-4 TO J4-16	USART OR INTERNAL CLOCK (CTC)
J6-8 AND J6-3 TO GND	NO SERIAL INTERRUPTS
J4-1 TO J4-2 J4-9 TO J4-10 J4-5 TO J4-6 J4-14 TO J4-15 J4-12 TO J4-11 J5-1 TO J5-8 J5-2 TO J5-4	FULL DUPLEX, ODD PARITY ASYNCHRONOUS RS-232C SERIAL INTERFACE WITH INTERNAL NEGATIVE SUPPLY (-10V) AND MCB=MODEM
1 TO 2 5 TO 6 TO 8	CONFIGURE PROM NON-VOLATILE MEMORY

Note: When using the MCR with a single +5 volt supply, the RS232 levels will be from +12 to -10 volts. These levels will generally work with any RS-232 receiver. Only when external -12 volt power is supplied will full RS-232 level specifications be met.

14.1 MCP EDGE CONNECTOR INTERFACE

Certain lines are purposely left open to allow for maximum user flexibility. Depending upon the users requirements, the following jumpers may need to be applied between edge connector pins:

JUMPER	FUNCTION
PIN 18 TO PIN 118	TO APPLY 1/2 Φ CLOCK (SHEET 1) TO CTC EXTERNAL CLOCK (SHEET 4) FOR HIGH SPEED SERIAL
PIN 16 TO PIN 21	SELECTS ON CARD RAM (SHEET 2)
PIN 83 TO PIN 92	SELECTS ON CARD PROM/EPROM (SHEETS 2 AND 3)
PIN 87 TO PIN 112	CONNECTS 2X SERIAL CLOCK OUT FROM CTC (SHEET 4) TO SERIAL CLOCK IN (2X) (SHEET 4) FOR SERIAL CLK OF 8251 COMMUNICATION INTERFACE (SHEET 6)
PIN 117 TO 120	DISABLE SERIAL INTERRUPT FROM 8251 (SHEET 6) TO Z80 (SHEET 1)
PIN 22 TO PIN 38	ENABLES INTERRUPT CONTROL FOR CTC (SHEET 4) FROM PIO (SHEET 5) IF JUMPER IS APPLIED BETWEEN Z21 (SHEET 5) AND APPROPRIATE PIO OUTPUT
PIN 114 TO PIN 121	DISABLE INTERRUPTS FOR PIO (SHEET 5)

14.2 TERMINAL INTERFACE

The PS-232 terminal is connected to the MCP according to the following interface connection:

PS-232 STANDARD	MCB PIN NUMBER	DESCRIPTION
1	N.C.	PROTECTIVE GND
2	15	RS-232 DATA IN
3	7	RS-232 DATA OUT
4	14	REQUEST TO SEND
5	11	CLEAR TO SEND
6	74	DATA READY
7	120	SIGNAL GND
8	80	RECEIVED LINE SIG DET
20	76	DATA TER RDY

Note: The signals on Pins 5,6,8 and 20 are used between data terminals and communication modems. For interface to most standard terminals, these lines can be tied together on the MCB connector.

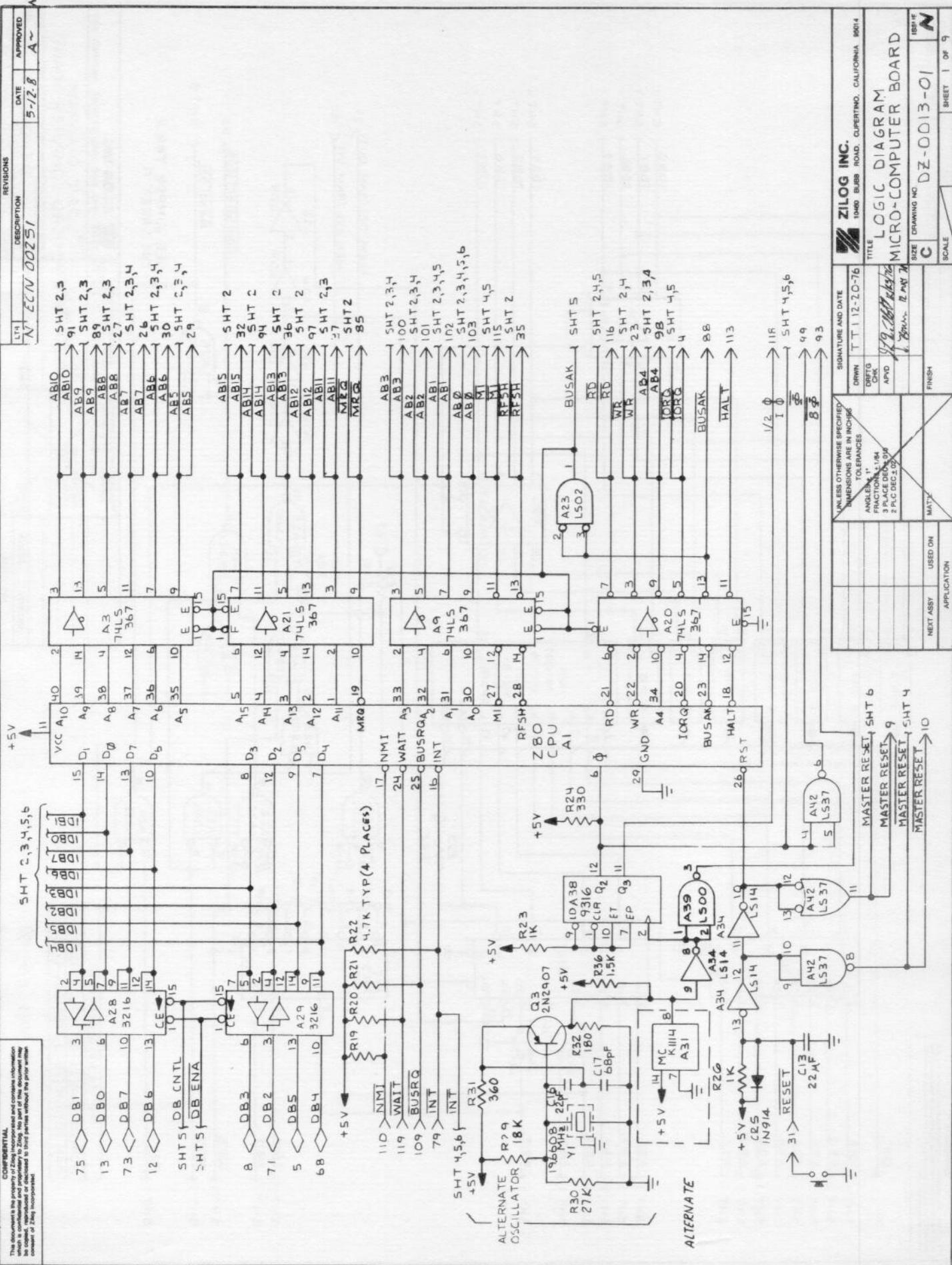
COMPONENT SIDE PIN NUMBER	SIGNAL NAME	FUNCTION
1	+5V	Vcc SUPPLY
2	+5V	
3	+5V	
4	IOR0-	
5	DB5	
6	20mA.DATA	TTY CURRENT LOOP DATA
7	RFCV.DATA	RS-232 DATA OUTPUT
8	DR3	
9	MASTER.RESET	RESET OUTPUT TO SYSTEM
10	MASTER.RESET-	RESET OUTPUT TO SYSTEM
11	CLEAR.TO.SEND	
12	DB6	
13	DB0	
14	REQ.TO.SEND	
15	XMITED.DATA	RS-232 DATA INPUT
16	MEMORYSEL.(RAM IN)	MCB MEMORY ENABLE
17	DISK,(C/T.0)	CTC CHANNEL 0 CLK/trigger INPUT
18	1/2.PHI.(C/T.1)	CTC CHANNEL 1 CLK/trigger INPUT
19	20mA.DATA.RTN	TTY CURRENT LOOP DATA RETURN
20	TTY.TAPE.CNTL.RET	
21	MEMORYSEL.(RAM OUT)	MCB MEMORY SELECTED
22	IEI.MCB.CTC	CTC INTERRUPT ENABLE INPUT
23	WR-	
24	USER STRB.2	CTC CHANNEL 2 ZERO COUNT OUTPUT
25	DTSK.STRB	CTC CHANNEL 0 ZFRO COUNT OUTPUT
26	AB7	
27	AB8	
28	IOWR-	
29	AB5	
30	AB6	
31	RESET-	RESET INPUT TO MCB
32	AB15	
33	SG2-	I/O SUBGROUP 2 DECODE
34	SG0-	I/O SUBGROUP 0 DECODE
35	RFSH-	MEMORY REFRESH SIGNAL FROM CPU
36	AB13	
37	AB11	
38	(Z21 PULL UP : USER DEFINABLE)	
39	(Z20 : USER DEFINABLE)	
40	(Z19 : USER DEFINABLE)	
41	(Z18 : USER DEFINABLE)	
42	(Z17 : USER DEFINABLE)	
43	(Z16 : USER DEFINABLE)	
44	(Z15 : USER DEFINABLE)	
45	(Z14 : USER DEFINABLE)	
46	(Z13 : USER DEFINABLE)	
47	(Z12 : USER DEFINABLE)	
48	(Z11 : USER DEFINABLE)	
49	(Z10 : USER DEFINABLE)	
50	(Z9 : USER DEFINABLE)	
51	(Z8 : USER DEFINABLE)	
52	(Z7 : USER DEFINABLE)	
53	(Z6 : USER DEFINABLE)	
54	(Z5 : USER DEFINABLE)	
55	(Z4 : USER DEFINABLE)	
56	(Z3 : USER DEFINABLE)	
57	(Z2 : USER DEFINABLE)	
58	(Z1 : USER DEFINABLE)	
59	+5V	Vcc SUPPLY
60	+5V	
61	+5V	

SOLDER SIDE PIN NUMBER	SIGNAL NAME	FUNCTION
62	GND	
63	GND	
64	GND	
65	TTY.TAPE.CNTL	
66	-5V.EXTERNAL	EXTERNAL Vbb SUPPLY
67	-5V.EXTERNAL	
68	DB4	
69	+12V.EXTERNAL	EXTERNAL Vdd SUPPLY
70	+12V.EXTERNAL	
71	DR2	
72	-12V.EXTERNAL	EXTERNAL -12V INPUT
73	DB7	
74	DATA.SET.RDY	
75	DB1	
76	DATA.TERM.RDY/XMITED.CLK	
77	20mA.RECV.RETN/REC.CLK	
78	SYNC.DET	SYNC CHARACTER DETECT OUTPUT
79	INT-	INTERRUPT REQUEST
80	LINE.SIGNAL.DET	+12V PULL UP
81	20mA.RECV	
82	USER.C/T.2	CTC CHANNEL 2 CLK/TRIGGER INPUT
83	ROM.SELECT-.(OUT)	MCB ROM SELECTED
84	RTC.(USER.C/T.3)	CTC CHANNEL 3 CLK/TRIGGER INPUT
85	MRQ-	
86	IEO.MCB.CTC	CTC INTERRUPT ENABLE OUTPUT
87	2X.SERIAL.CLOCK.(OUT)	2X UART XMIT/REC CLOCK
88	BUSAK-	
89	AB9	
90	IORD-	
91	AB10	
92	ROM.SELECT-.(IN)	MCB ROM ENABLE
93	8.PHI-.(8X SYSTEM CLOCK-)	
94	AB14	
95	SG3-	I/O SUBGROUP 3 DECODE
96	SG1-	I/O SUBGROUP 1 DECODE
97	AB12	
98	AB4	
99	PHI-.(SYSTEM CLOCK-)	
100	AB3	
101	AB2	
102	AB1	
103	AB0	
104	GP0-	I/O GROUP 0 DECODE
105	GP1-	I/O GROUP 1 DECODE
106	GP2-	I/O GROUP 2 DECODE
107	GP3-	I/O GROUP 3 DECODE
108	GP4-	I/O GROUP 4 DECODE
109	BUSRQ-	
110	NMI-	NON-MASKABLE INTERRUPT INPUT
111	IEO.MCB.PIO	PIO INTERRUPT ENABLE OUTPUT
112	2X.SERIAL.CLOCK.(IN)	UART XMIT/REC CLOCK INPUT
113	HALT-	CPU HALT STATUS OUTPUT
114	IEI.MCB.PIO	PIO INTERRUPT ENABLE INPUT
115	M1-	
116	RD-	
117	SFRIAL.TNTE.IN.(MCB)	UART INTERRUPT ENABLE INPUT
118	1/2.PHI.(HALF SYSTEM CLOCK)	
119	WAIT-	WAIT REQUEST INPUT TO CPU
120	GND	
121	GND	
122	GND	

16.0 MCB SCHEMATICS

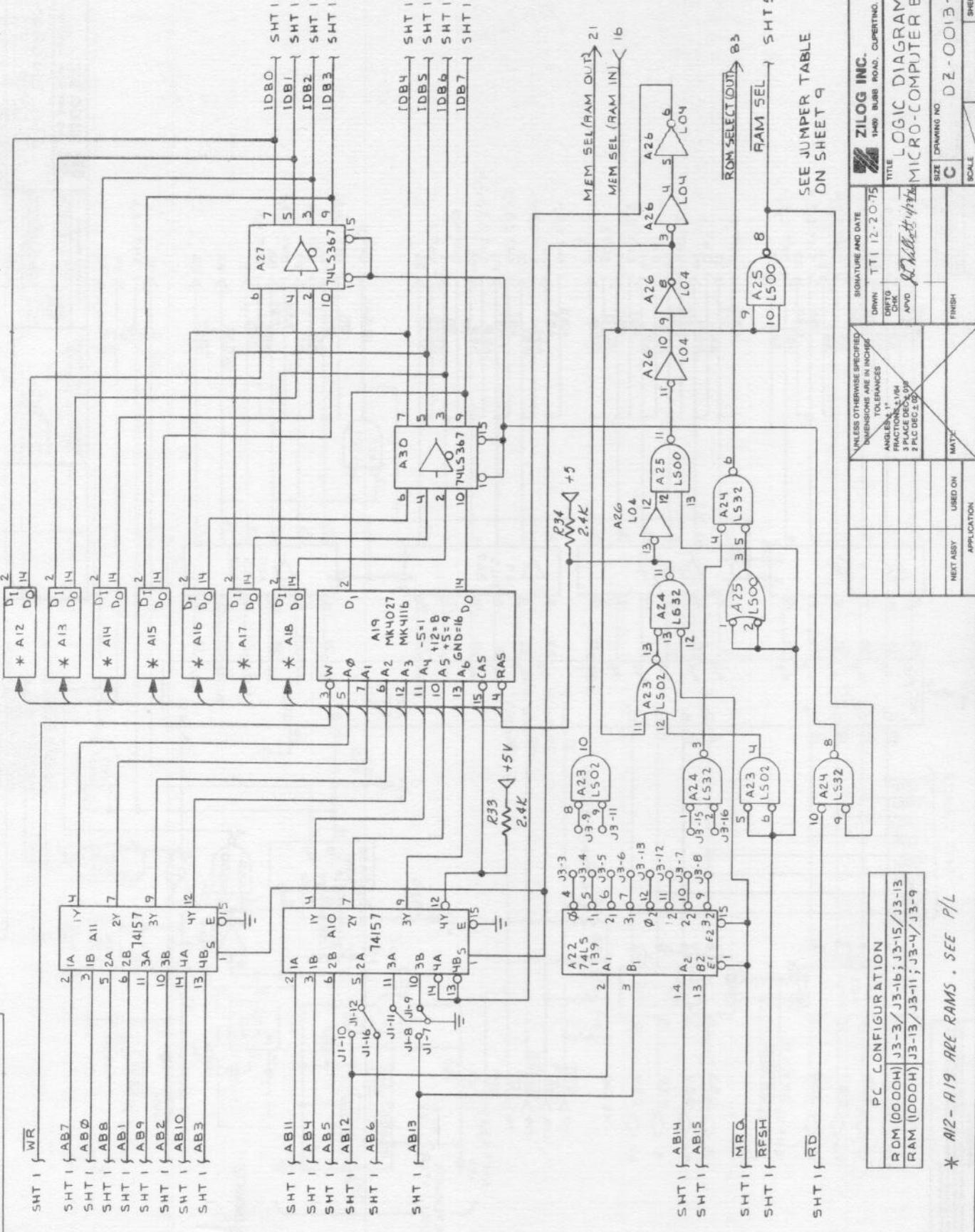
CONFIDENTIAL
This document is the property of Zilog Incorporated and contains information which is confidential and proprietary to Zilog. No part of this document may be copied, reproduced or disclosed to third parties without the prior written consent of Zilog Incorporated.

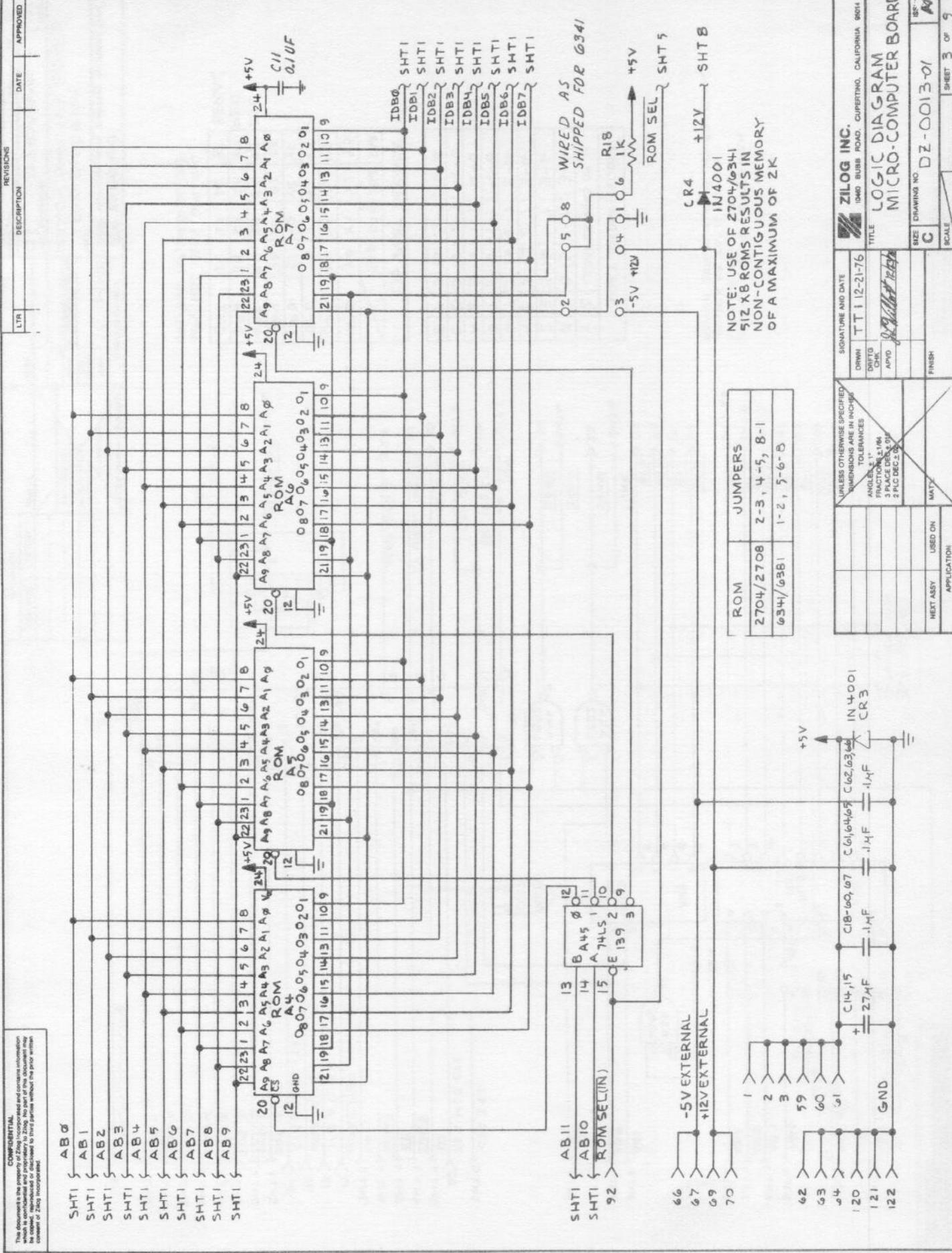
CONFIDENTIAL
Document is the property of Zilog Incorporated and contains information
confidential and proprietary to Zilog. No part of this document may
be reproduced or disclosed to third parties without the prior written
consent of Zilog Incorporated.

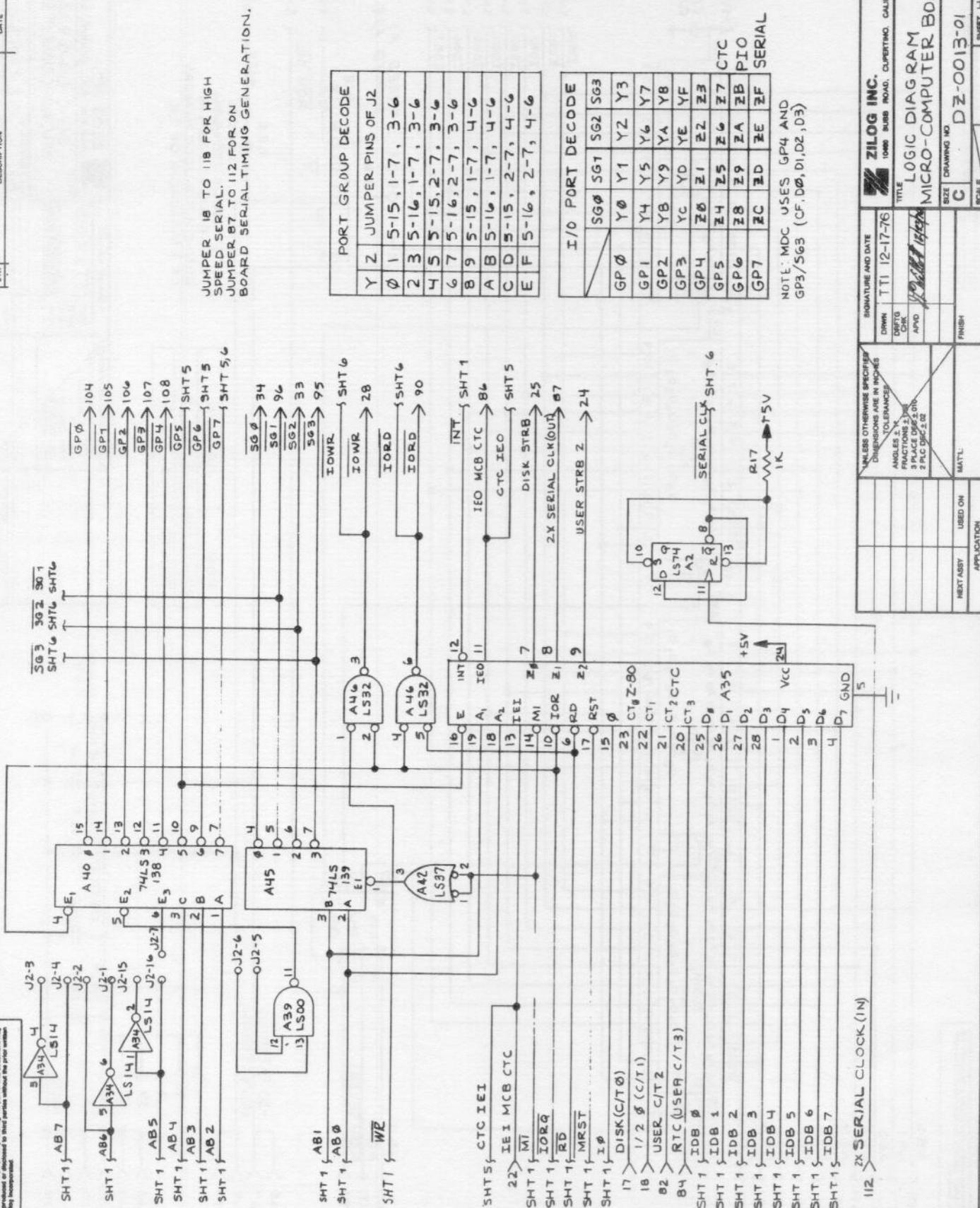


CONFIDENTIAL

This document is the property of Zilog Incorporated and contains information which is confidential and proprietary to Zilog. No part of this document may be copied, reproduced or disclosed to third parties without the prior written consent of Zilog.

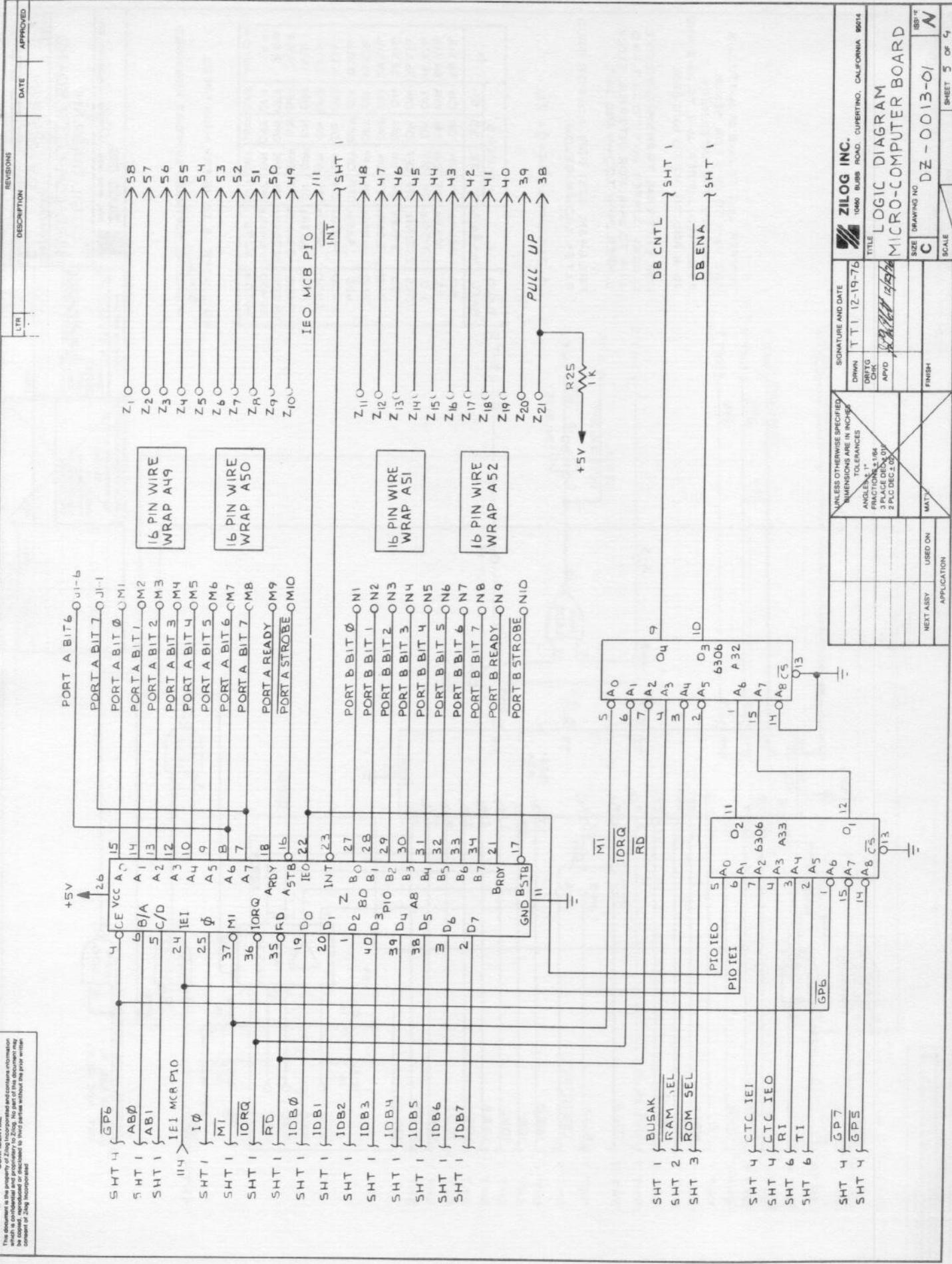




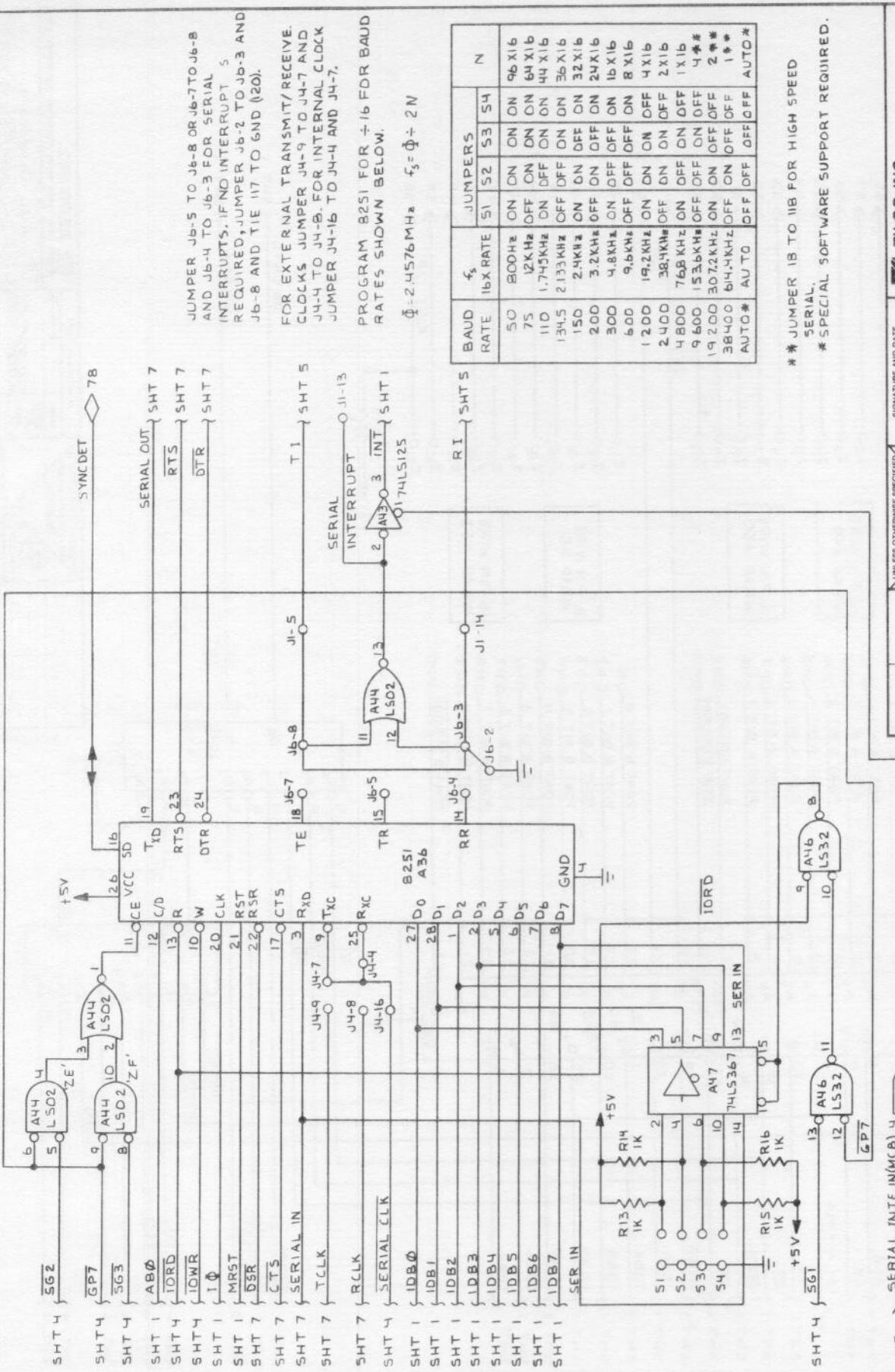


CONFIDENTIAL.
This document is the property of ZTE Incorporated and contains information which is confidential and proprietary to ZTE. No part of this document may be copied, reproduced or disclosed to third parties without the prior written consent of ZTE.

CONFIDENTIAL
This document is the property of Zilog Incorporated
which reserves the right to copy, loan, rent or otherwise to Zilog. No part of this document may
be reproduced or communicated to anyone by any means without the prior written
consent of Zilog Incorporated.



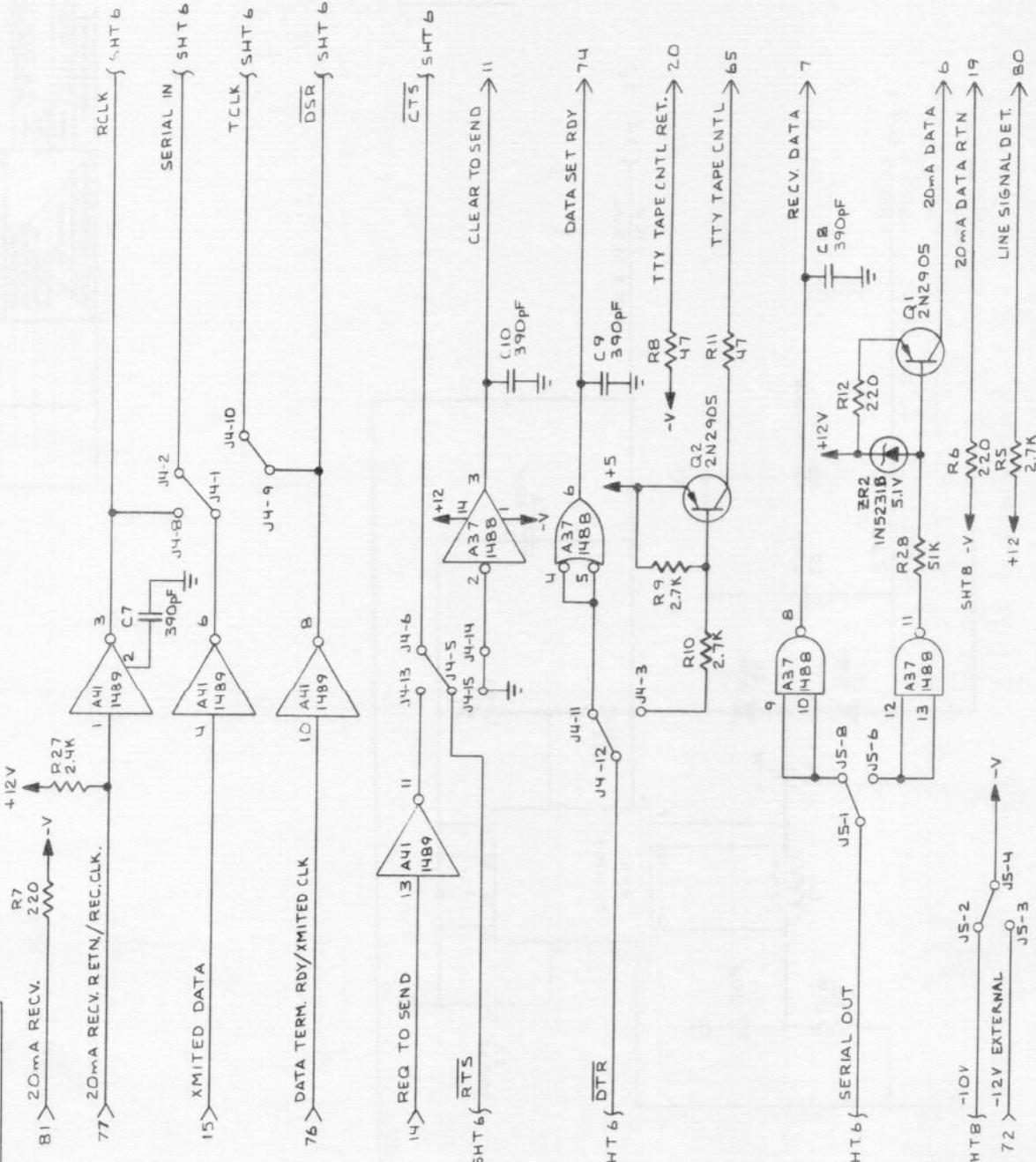
CONFIDENTIAL.
This document is the property of Zilog Incorporated and contains information which is confidential and proprietary to Zilog. No part of this document may be copied, reproduced or disclosed to third parties without the prior written consent of Zilog Incorporated.



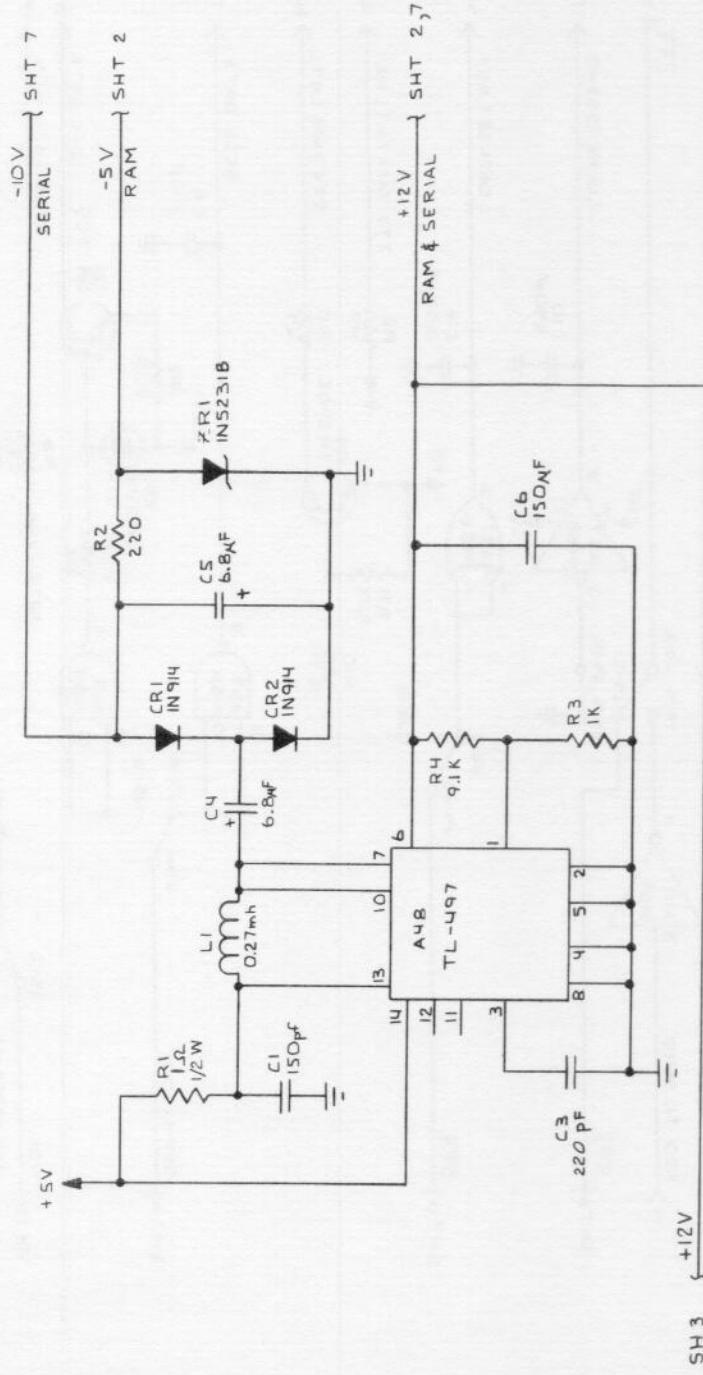
117		(GND IF SERIAL INT NOT REQ)	
		A39 LS00	
		b	
UNLESS OTHERWISE SPECIFIED TOLERANCES ARE IN INCHES		STANDARD AND DRAWING ANGLES: 1° FRACTION: 1/164 3 PLACE DEC. 0.001 2 PLACE DEC. 0.01	
		DRWNS: T T 12-20-76 DRAFTING CHART APVD	
		TITLE LOGIC DIAGRAM MICRO-COMPUTER BOARD	
		SIZE DRAWING NO. DZ-0013-01	
		SCALE	
		ISSUE A	
		SHEET 6 OF 9	
NEXT ASSY/ APPLICATN		USED ON MATE	

CONFIDENTIAL
This document is the property of Zilog Incorporated and contains information
which is confidential and proprietary to Zilog Incorporated.
It is to be controlled and handled in accordance with Zilog's security procedures.
No part of this document may be reproduced or given to anyone without the prior written
consent of Zilog Incorporated.

REVISIONS
DESCRIPTION
DATE APPROVED



Zilog Inc.		Signature and Date	
		Drawn	T/T 1 12-17-76
		Design	
		Approved	
Title: LOGIC DIAGRAM			
Micro-Computer Board			
Size	Drawing No.	Issue	
C	DZ-0013-01	4	
Sheet 7 of 9			



NOT USED	LAST USED
C11,C12,C25	A52,C67,CR3, J6,L,M10,NiO, Q3,R32,Z21, ZR2

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES		SIGNAL AND DATE DRAWN BY TITLE APPROVED REVIEWED SIZE C DRAWING NO. DZ-0013-01	
TOLERANCES ANGLES: 1° FRACTIONS: 1/164 3 PLACE DEC: 0.001 2 PL DEC: 0.01		LOGIC DIAGRAM MICRO-C COMPUTER BOARD	
MATERIAL		FINISH	
NEXT ASSY:		USED ON	
APPLICATION			

CONFIDENTIAL
This document is the property of Zilog Incorporated and contains information which is confidential and proprietary to Zilog. No part of this document may be reproduced or transmitted in whole or in part without the prior written consent of Zilog Incorporated.

REVISIONS		DESCRIPTION		DATE	APPROVED
LTR					
RAM MEMORY JUMPERS	SERIAL INTERFACE JUMPERS				
4K '4027P-4' J1-12/J1-16,J1-7/J1-9/J1-11 16K (4116D) J1-10/J1-12,J1-9/J1-16,J1-8/J1-11	NEGATIVE SUPPLY -10V (ON BOARD SOURCE) J5-2 TO J5-4 -12V (EXTERNAL SOURCE) J5-3 TO J5-4				
MEMORY PAGE DECODING RAM-J3-9 & J3-11 ROM-J3-15 & J3-16	20mA CURRENT LOOP RS-232-C MCB = 'MODEM' ALWAYS 'CLEAR TO SEND' IGNORE 'REQUEST TO SEND' SIGNAL SWINGS FROM +12V TO NEGATIVE SUPPLY. (AS SHIPPED)	J4-8/J4-2, J4-9/J4-10, J4-5/J4-6, J4-14/J4-15, J4-12/J4-3, J5-1/J5-6, J4-4/J4-7/J4-16 DATA TO MCB SOURCE-BI, RETURN-77 DATA FROM MCB SOURCE-b, RETURN-19 READER CONTROL SOURCE-65, RETURN-20			
4K PAGES HEX ADDRESSES 16K PAGES	RS-232-C MCB = 'TERMINAL' SIGNAL SWINGS FROM +12V TO NEGATIVE SUPPLY.	J4-1/J4-2, J4-9/J4-10, J4-13/J4-6, J4-5/J4-14, J4-11/J4-12, J5-1/J5-8, J4-4/J4-7/J4-16. DATA FROM MODEM-15 DATA TO MODEM-7 CLEAR TO SEND -14 REQUEST TO MODEM-11 DATA SET READY -76 DATA TERMINAL READY -74			
SYNCHRONOUS EXAMPLE: FOR RAM ON 4K PAGE 1 JUMPER J3-4 TO J3-9 & J3-11 TO J3-13 FOR ROM ON 4K PAGE 0 JUMPER J3-3 TO J3-15 & J3-13 TO J3-16	SYNCHRONOUS WITH CONTROL SIGNAL SWINGS FROM +12V TO NEGATIVE SUPPLY.	J4-1/J4-2, J4-9/J4-7, J4-5/J4-6, J4-14/J4-8, J5-1/J5-8 DATA TO MCB (RECV) -15 DATA FROM MCB (XMIT) -17 RECEIVE DATA CLOCK -77 TRANSMIT DATA CLOCK (INPUT)-76 BI-DIRECTIONAL SYNC DETECT -78			
SERIAL INTERFACE INTERRUPT JUMPER NO INTERRUPTS-GROUND I/O PIN 117 E JUMPER Jb-2/Jb-3/Jb-8	ROM (6341/6381) ON 4K PAGE 0 USART ON INTERNAL CLOCK MCB = 'MODEM'	J4-1/J4-2, J4-9/J4-7, J4-5/J4-14, J4-11/J4-12, J5-1/J5-8. DATA TO MCB (RECV) -15 DATA FROM MCB (XMIT) -17 RECEIVE DATA CLOCK -77 TRANSMIT DATA CLOCK (INPUT)-76 STATUS/CONTROL (OUT)-11 TRANSMITTER ENABLE -14 STATUS/CONTROL 2(OUT)-74 BI-DIRECTIONAL SYNC DETECT -78			
INTERRUPT ON RECEIVER READY- Jb-3 TO Jb-4 TRANSMITTER READY- Jb-5 TO Jb-8 TRANSMITTER EMPTY- Jb-7 TO Jb-8 INTERRUPT ENABLE CONTROL I/O PIN 117	AS SHIPPED CONFIGURATION TIE I/O PIN 18 TO 118, PIN 16 TO 21, PIN 87 TO 112, PIN 117 TO 120, TIE I/O PIN 22 TO 38, PIN 119 TO 121.	RAM (4K) ON 4K PAGE 1 NO SERIAL INTERRUPTS. SERIAL INTERFADE RS-232-C WITH INTERNAL NEG. SUPPLY (10V)			
VECTORED INTERRUPT (MODE 2) JUMPER J1-13 TO J1-1 (PORT A BIT7) PROGRAM PIO PORT A TO OPERATE IN MODE 3 (BIT MODE). PROGRAM INTERRUPTS, OR, CONTROL TO ENABLE INTERRUPTS, OR, LOW, & MASK FOLLOWS. SET MASK TO ALLOW BIT 7 TO INTERRUPT (7FH).	UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES ANGLES IN DEGREES FRACTIONAL: 1/16, 3 PLACE DECIMAL: 0.0625 2 PLIC DEC 2 OF	ZILOG INC. 10400 BULL ROAD, CUPERTINO, CALIFORNIA 95014 TITLE LOGIC DIAGRAM DRAWING NO. DZ-DO13-01 SIZE C SCALE 1:1 MATERIAL APPLICATION			

17.0 MCB SPECIFICATIONS

POWER SUPPLY:	+5 VDC +5%, MAX CURRENT 2 AMPS (WITH 3 PROMS)	
CONNECTOR:	122 PIN EDGE (100 MIL SPACING) AVAILABLE FROM:	
	VENDOR	PART NO.
	CARRY MFG CO.	4000-2
	AUGAT	14005-19P1
CARD CHARACTERISTICS:	LENGTH: 7.7 IN. (19.6cm) DEPTH: 7.5 IN. (19.1cm) THICKNESS: .062 IN. (1.6mm) SPACING BETWEEN CARDS: 0.5 IN. CENTERS (1.3cm) MAX COMPONENT HEIGHT: 0.4 IN. (1.0cm) ETCH LAYERS: TWO	
ENVIRONMENTAL:	0 -50 C TEMP. RANGE UP TO 90% HUMIDITY WITHOUT CONDENSATION	
CLOCK:	19.6608 MHZ	

ALPHABETICAL
ASSEMBLY MNEMONIC

OPERATION

ADC HL,ss	Add with Carry Reg. pair ss to HL
ADC A,s	Add with carry operand s to Acc.
ADD A,n	Add value n to Acc.
ADD A,r	Add Reg. r to Acc.
ADD A,(HL)	Add location (HL) to Acc.
ADD A,(IX+d)	Add location (IX+d) to Acc.
ADD A,(IY+d)	Add location (IY+d) to Acc.
ADD HL,ss	Add Reg. pair ss to HL
ADD IX,pp	Add Reg. pair pp to IX
ADD IY,rr	Add Reg. pair rr to IY
AND s	Logical 'AND' of operand s and Acc.
BIT b,(HL)	Test BIT b of location (HL)
BIT b,(IX+d)	Test BIT b of location (IX+d)
BIT b,(IY+d)	Test BIT b of location (IY+d)
BIT b,r	Test BIT b of Reg. r
CALL cc,nn	Call subroutine at location nn if condition cc is true
CALL nn	Unconditional call subroutine at location nn
CCF	Complement carry flag
CP s	Compare operand s with Acc.
CPD	Compare location (HL) and Acc.
CPDR	decrement HL and BC Compare location (HL) and Acc. decrement HL and BC, repeat until BC=0
CPI	Compare location (HL) and Acc. increment HL and decrement BC
CPIR	Compare location (HL) and Acc. increment HL, decrement BC repeat until BC=0
CPL	Complement Acc. (1's comp)
DAA	Decimal adjust Acc.
DEC m	Decrement operand m
DEC IX	Decrement IX
DEC IY	Decrement IY
DEC ss	Decrement Reg. pair ss
DI	Disable interrupts
DJNZ e	Decrement B and Jump relative if B≠0
EI	Enable interrupts
EX (SP),HL	Exchange the location (SP) and HL

EX (SP),IX	Exchange the location (SP) and IX
EX (SP),IY	Exchange the location (SP) and IY
EX AF,AF'	Exchange the contents of AF and AF'
EX DE,HL	Exchange the contents of DE and HL
EXX	Exchange the contents of BC,DE,HL with contents of BC',DE',HL' respectively
HALT	HALT (wait for interrupt or reset)
IM 0	Set interrupt mode 0
IM 1	Set interrupt mode 1
IM 2	Set interrupt mode 2
IN A,(n)	Load the Acc. with input from device n
IN r,(C)	Load the Reg. r with input from device (C)
INC (HL)	Increment location (HL)
INC IX	Increment IX
INC (IX+d)	Increment location (IX+d)
INC IY	Increment IY
INC (IY+d)	Increment location (IY+d)
INC r	Increment Reg. r
INC ss	Increment Reg. pair ss
IND	Load location (HL) with input from port (C), decrement HL and B
INDR	Load location (HL) with input from port (C), decrement HL and decrement B, repeat until B=0
INI	Load location (HL) with input from port (C); and increment HL and decrement B
INIR	Load location (HL) with input from port (C), increment HL and decrement B, repeat until B=0
JP (HL)	Unconditional Jump to (HL)
JP (IX)	Unconditional Jump to (IX)
JP (IY)	Unconditional Jump to (IY)
JP cc,nn	Jump to location nn if condition cc is true
JP nn	Unconditional jump to location nn
JR C,e	Jump relative to PC+e if carry=1
JR e	Unconditional Jump relative to PC+e
JR NC,e	Jump relative to PC+e if carry=0

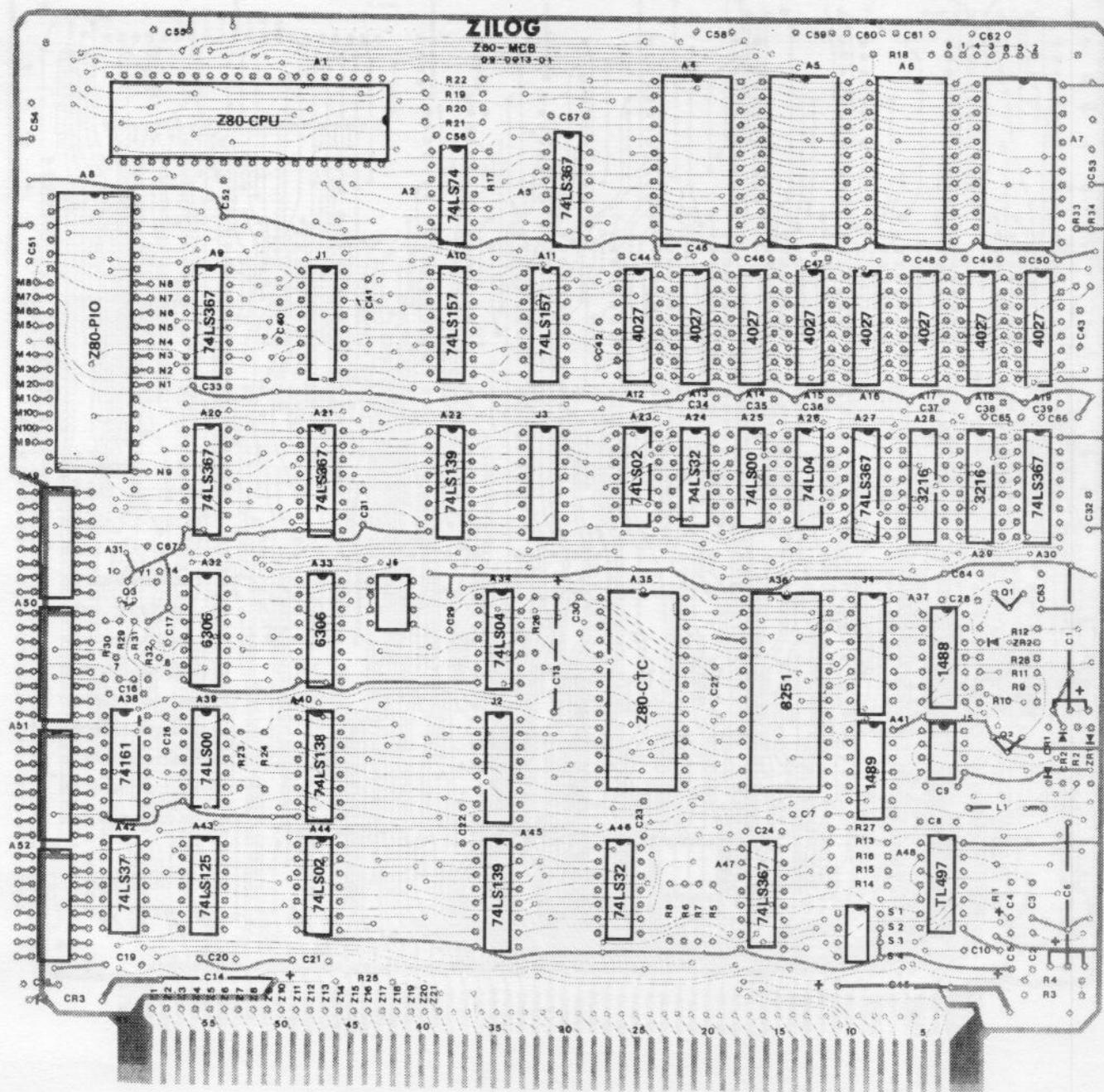
JR NZ,e	Jump relative to PC+e if non zero (Z=0)
JR Z,e	Jump relative to PC+e if zero (Z=1)
LD A,(BC)	Load Acc. with location (BC)
LD A,(DE)	Load Acc. with location (DE)
LD A,I	Load Acc. with I
LD A,(nn)	Load Acc. with location nn
LD A,R	Load Acc. with Reg. R
LD (BC),A	Load location (BC) with Acc.
LD (DE),A	Load location (DE) with Acc.
LD (HL),n	Load location (HL) with value n
LD dd,nn	Load Reg. pair dd with value nn
LD dd,(nn)	Load Reg. pair dd with location (nn)
LD HL,(nn)	Load HL with location (nn)
LD (HL),r	Load location (HL) with Reg. r
LD I,A	Load I with Acc.
LF IX,nn	Load IX with value nn
LD IX,(nn)	Load IX with location (nn)
LD (IX+d),n	Load location (IX+d) with value n
LD (IX+d),r	Load location (IX+d) with Reg. r
LD IY,nn	Load IY with value nn
LD IY,(nn)	Load IY with location (nn)
LD (IY+d),n	Load location (IY+d) with value n
LD (IY+d),r	Load location (IY+d) with Reg. r
LD (nn),A	Load location (nn) with Acc.
LD (nn),dd	Load location (nn) with Reg. pair dd
LD (nn),HL	Load location (nn) with HL
LD (nn),IX	Load location (nn) with IX
LD (nn),IY	Load location (nn) with IY
LD R,A	Load R with Acc.
LD r,(HL)	Load Reg. r with location (HL)
LD r,(IX+d)	Load Reg. r with location (IX+d)
LD r,(IY+d)	Load Reg. r with location (IY+d)
LD r,n	Load Reg. r with value n
LD r,r'	Load Reg. r with Reg. r'
LD SP,HL	Load SP with HL
LD SP,IX	Load SP with IX
LD SP,IY	Load SP with IY
LDD	Load location (DE) with location (HL), decrement DE,HL and BC
LDDR	Load location (DE) with location (HL), decrement DE,HL and BC; repeat until BC=0

LDI	Load location (DE) with location (HL), increment DE, HL, decrement BC
LDIR	Load location (DE) with location (HL), increment DE, HL, decrement BC and repeat until BC=0
NEG	Negate Acc. (2's complement)
NOP	No operation
OR s	Logical 'OR' of operand s and Acc.
OTDR	Load output port (C) with location (HL) decrement HL and B, repeat until B=0
OTIR	Load output port (C) with location (HL), increment HL, decrement B, repeat until B=0
OUT (C),r	Load output port (C) with Reg. r
OUT (n),A	Load output port (n) with Acc.
OUTD	Load output port (C) with location (HL), decrement HL and B
OUTI	Load output port (C) with location (HL), increment HL and decrement B
POP IX	Load IX with top of stack
POP IY	Load IY with top of stack
POP qq	Load Reg. pair qq with top of stack
PUSH IX	Load IX onto stack
PUSH IY	Load IY onto stack
PUSH qq	Load Reg. pair qq onto stack
RES b,m	Reset Bit b of operand m
RET	Return from subroutine
RET cc	Return from subroutine if condition cc is true
RETI	Return from interrupt
RETN	Return from non maskable interrupt
RL m	Rotate left through carry operand m
RLA	Rotate left Acc. through carry
RLC (HL)	Rotate location (HL) left circular
RLC (IX+d)	Rotate location (IX+d) left circular
RLC (IY+d)	Rotate location (IY+d) left circular
RLC r	Rotate Reg. r left circular
RLCA	Rotate left circular Acc.
RLD	Rotate digit left and right between Acc. and location (HL)
RR m	Rotate right through carry operand m
RRA	Rotate right Acc. through carry
RCR m	Rotate operand m right circular

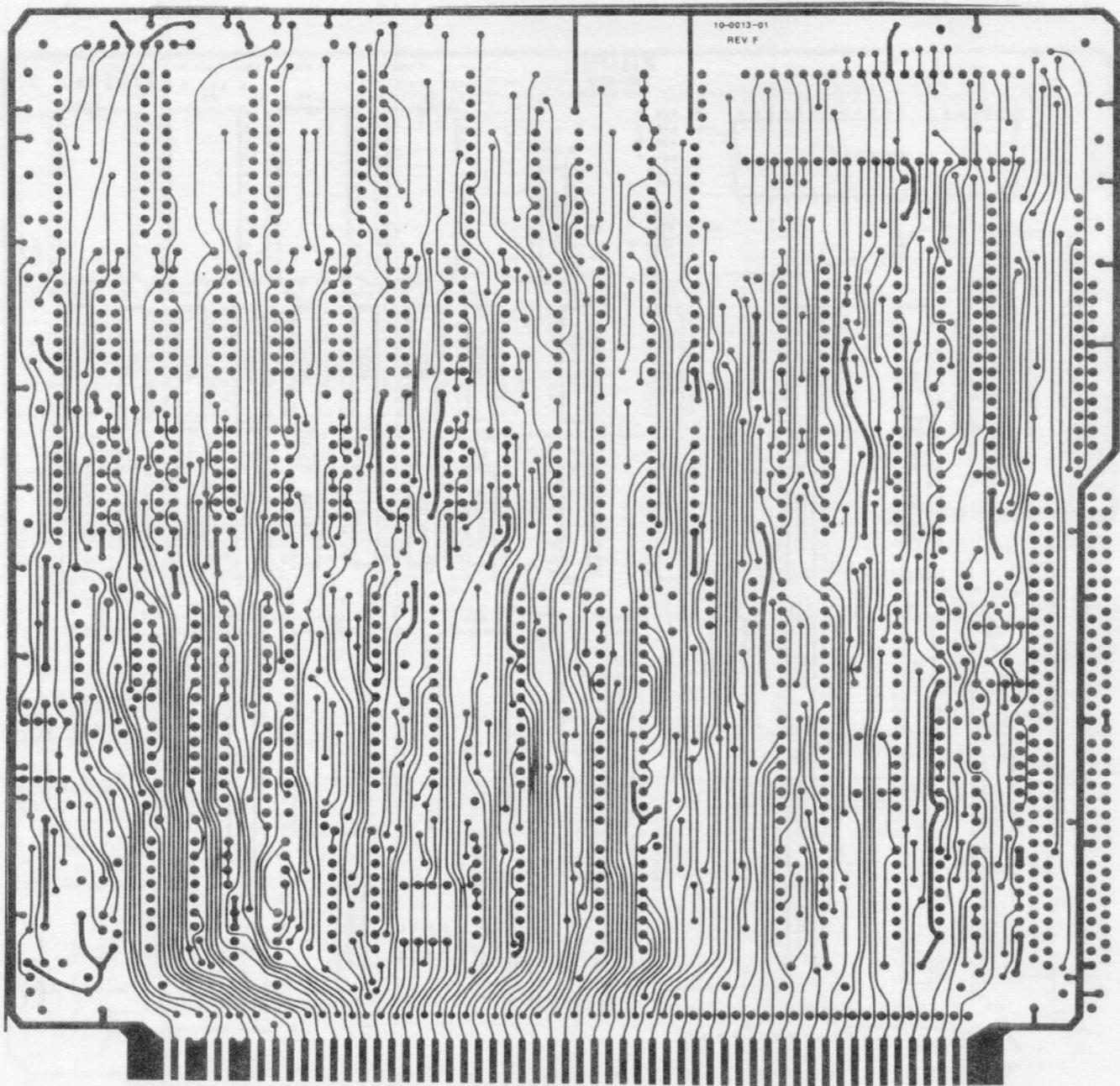
RRCA	Rotate right circular Acc.
RRD	Rotate digit right and left between Acc. and location (HL)
RST p	Restart to location p
SBC A,s	Subtract operand s from Acc. with carry
SBC HL,ss	Subtract Reg. pair ss from HL with carry
SCF	Set carry flag (C=1)
SET b,(HL)	Set Bit b of location (HL)
SET b,(IX+d)	Set Bit b of location (IX+d)
SET b,(IY+d)	Set Bit b of location (IY+d)
SET b,r	Set Bit b of Reg. r
SLA m	Shift operand m left arithmetic
SRA m	Shift operand m right arithmetic
SRL m	Shift operand m right logical
SUB s	Subtract operand s from Acc.
XOR s	Exclusive 'OR' operand s and Acc.

19.0 PARTS LIST

ITEM	DESCRIPTION	QUANTITY	REFERENCE
1	PCB, BLANK, FEVR, C, F, E	1	
2	I.C., Z-80, CPU	1	A1
3	74LS74	1	A2
4	74LS367	7	A3, 9, 20, 21, 27, 30, 47
5			
6	7-80, PTO	1	A8
7	74157	2	A10, 11
8	NFC-LPD-414D	8	A12-19
9	74LS139	2	A22, 45
10	74LS02	2	A23, 44
11	74LS32	2	A24, A46
12	74LS00	2	A25, 39
13	74L04	1	A26
14	3216	2	A28, 29
15	6306	1	A32
16	6306	1	A33
17	74LS14	1	A34
18			
19	8251	1	A36
20	1488	1	A37
21	9316	1	A38
22	74LS138	1	A40
23	1489	1	A41
24	74LS125	1	A43
25			
26	TL-497	1	A48
27	74LS37	1	A42
28	RES, 1/2W, 1-OHM	1	R1
29	RES, 1/4W, 220-OHM	4	R2, 6, 7, 12
30	1-K	10	R3, 13-18, 23, 25, 26
31	9.1-K	1	R4
32	2.7-K	3	R5, 9, 10
33	47-OHM	2	R8, 11
34	4.7-K	4	R19-22
35	330-OHM	1	R24
36	2.4-K	3	R27, 33, 34
37	51-K	1	R28
38	27-K	1	R30
39	360-OHM	1	R31
40	180-OHM	1	R32
41	SOCKET, 40PIN	2	A1, 8
42	SOCKET, 28PIN	2	A35, 36
43	SOCKET, 24PIN	4	A4-7
44	SOCKET, 16PIN	12	A12-19, 32, 33, J1, J3
45	RFS, 1/4W, 18-K	1	R29
46	CRYSTAL, 19.6608MHZ	1	Y1
47	DIODE, 1N914	2	CR1, 2
48	1N4001	2	CR3, 4
49	RFG, 1N5231B	2	ZR1, 2
50	TRANSISTOR, 2N2905	2	Q1, 2
	PAD, TRANSISTOR	2	Q1, 2
	PAD, TRANSISTOR	1	Q3
51	TRANSISTOR, 2N2907	1	Q3
52	INDUCTOR, 0.27MH	1	L1
53	SWITCH, AMP7634	1	S1
54	CAP, 150-UF	2	C1, 6
56	220PF	1	C3
55			
57	6.8UF	2	C4, 5
58	390PF	4	C7-10
59			
60	22-UF	3	C13, 14, 15
61	22PF	1	C16
62	68PF	1	C17
63	0.1-UF	49	C11, 18-24, 26-66
64	EJECTOR, ENGRAVED	1	
65	EJECTOR, BLANK	1	
66	PIN, EJECTOR	2	



Z80-MCB COMPONENT SIDE



Z80-MCB SOLDER SIDE