INDEX

Z80 CPU PARTS LIST

INTEGRATED CIRCUITS

	QUANTITY	PART#	PLACEMENT
[ ]	1	74LS00	U23
[ ]	2	74S00 (2)	U5,U44
[ ]	1	74LS02	U14
[ ]	4	74LS04	U40,U41,U46,U47
[ ]	1	7406	U16
		OR /16	
[ ]	1	74S08	U7
[ ]	2	74LS08	U27,U43
[ ]	1	74LS14	U17
[ ]	1	74LS21	U13
[ ]	1	74LS30	U19
[ ]	2	74LS32	U8,U10
[ ]	2	74S74 (2)	U6,U42
[ ]	2	74LS74	U3,U26
[ ]	1	74LS85	U25
[ ]	1	74LS123	U4
[ ]	1	74LS132	U15
[ ]	1	74LS139	U12
[ ]	1	74LS240	U34
[ ]	1	74LS241	U22
[ ]	7	74LS244	U20,U21,U28,U31,U32,U33,U45
[ ]	2	74S257 (8)	U51,U52
[ ]	2	74LS266	U11,U24
[ ]	1	74LS373	U30
[ ]	1	74LS374	U18
[ ]	1	1488 (1)	U1
[ ]	1	1489 (1)	U2
[ ]	8	4164 (2)	U48,U49,U50,U53,U54,U55,U56,U57
[ ]	1	8255 (6)	U35
[ ]	1	Z-80A (3)	U29
[ ]	1	8253 (1)	U9
[ ]	1	78H05 (4)	
[ ]	1	78L12 (4)	P
[ ]	1	79L12 (4)	N
[ ]	1	2732 (5)	U39,U38,U37,U36
[ ]	1	6116 (7)	U38

NOTES:

(1) REQUIRED FOR USE WITH OPTIONAL RS-232-C PIGGY BOARD

(2) U42,U44 NOT REQUIRED UNLESS 64K RAM OPTION INSTALLED
REQUIRED FOR 64K RAM OPTION
4164-200 FOR 4 MHz OPERATION
4164-150 FOR 6 MHz OPERATION

(3) Z-80B REQUIRED FOR 6 MHz OPERATION

(4) REQUIRED IF STANDARD S-100 SUPPLY IS USED (+8V,+/-16V)

(5) 2732A REQUIRED FOR 6 MHz OPERATION
STANDARD MULTIFLEX MONITOR IN U39

(6) OPTIONAL PARALLEL PORT

(7) REQUIRED IF 64K OPTION NOT INSTALLED

(8) S258 OR S157 OR S158 MAY BE SUBSTITUTED

QUAN.	PART#	PLACEMENT

RESISTORS

[ ] 12	68 OHM	R24, R25, R26
[ ] 2	330 OHM	R14,R20
[ ] 5	1 kOHM	R7,R27,R28,R30,R31
[ ] 8	2 kOHM	R2,R3,R9,R11,R12,R13,R22,R23
[ ] 1	2.7 kOHM	R29
[ ] 2	10 kOHM	R4,R5
[ ] 1	33 kOHM	R16

SIPS

6 PIN

[ ] 2	10 kOHM [4306-101-103]	R6,R19

8 PIN

[ ] 6	2 kOHM [4308-101-202]	R1,R8,R10,R15,R17,R18

10 PIN

[ ] 1	22 kOHM [4310-101-223]	R21

CAPACITORS

[ ] 1	2.2 pF	C10 **deleted**
[ ] 1	15 pF	C3
[ ] 3	470 pF	C4,C6,C8
[ ] 2	1000 pF	C7,C9
[ ] 1	10000 pF	C2
[ ] 1	22000 pF	C1 **deleted**
[ ] 305	0.1 uF	(UNMARKED)
[ ] 1	68 uF	C5 [TANTALUM]
[ ] 3	3.3 uF	Creg (near pin 1 of S100 connector) [TANTALUM]

SOCKETS

[ ] 2	40 PIN
[ ] 4	28 PIN
[ ] 1	24 PIN
[ ] 11	20 PIN
[ ] 13	16 PIN
[ ] 26	14 PIN

MISCELLANEOUS

[ ]	1	Z80 CPU PC BOARD		
[ ]	1	DUAL MALE HEADER STRIP		J12, J13, J14, J16
[ ]		FEMALE JUMPER PINS		
[ ]	1	4 MHZ CRYSTAL	(9)	Y1
[ ]	1	6 MHZ CRYSTAL	(10)	Y1
[ ]	1	2 MHZ CRYSTAL	(10)	Y2
[ ]	1	1N914 DIODE		D1

(9) USED FOR 4 MHZ OPERATION
(10) USED FOR 6 MHZ OPERATION

MOTHERBOARD PARTS CHECKLIST

		Part #	Placement
[ ]	2	74LS02	U32, U36
[ ]	2	74LS04	U4, U34
[ ]	1	7406	U5
[ ]	2	74LS20	U1, U35
[ ]	2	74LS74	U22,U23
[ ]	1	74LS123	U10
[ ]	1	74LS132	U11
[ ]	1	74LS191	U21
[ ]	2	74LS174	U7, U18
[ ]	2	74128	U24, U25
[ ]	3	74LS374	U8, U26, U27
[ ]	1	74LS367	U9
[ ]	1	74LS240	U29
[ ]	3	74LS244	U28, U38, U40
[ ]	2	74LS138	U2, U3
[ ]	1	ULN2003	U6
[ ]	6	TIL313	U12, U13, U14, U15, U16, U17
[ ]	1	CD4015	U20
[ ]	1	LM3900	U30
[ ]	1	CD4702 (1)	U33
[ ]	1	TL497A (2)	U31
[ ]	1	8255	U37
[ ]	1	8251 (1)	U39
[ ]	1	1488 (1)	U41
[ ]	1	1489 (1)	U42
[ ]	1	7805 (3)	VR1
[ ]	1	7812 (3)	VR3
[ ]	1	7905 (3)	VR2
[ ]	1	7912 (3)	VR4

[ ]	1	1N4001			CR1
[ ]	2	2N4401	or	PN2222A	Q1, Q3
[ ]	1	2N4403	or	2N4402	Q2

[ ]	1	JUMBO RED LED		D1
[ ]	1	2.4576 MHz XTAL	(1)	Y1
[ ]	1	500 uH COIL	(2)	L1

[ ]	1	1 OHM 1/4W	R36
[ ]	2	47 OHM 1/4W	R2, R3
[ ]	1	330 OHM 1/4W	R5
[ ]	1	1.2 kOHM 1/4W	R38
[ ]	7	2.0 kOHM 1/4W	R6, R7, R9, R10, R14, R15, R40
[ ]	2	3.0 kOHM 1/4W	R23, R31
[ ]	1	4.7 kOHM 1/4W	R37
[ ]	5	5.1 kOHM 1/4W	R8, R42, R43, R44
[ ]	1	5.6 kOHM 1/4W	R1
[ ]	2	10 kOHM 1/4W	R4, R11
[ ]	2	15 kOHM 1/4W	R27, R30
[ ]	1	18 kOHM 1/4W	R39

[ ]	2		20 kOHM 1/4W	R19
[ ]	1		24 kOHM 1/4W	R24
[ ]	2		30 kOHM 1/4W	R29, R32
[ ]	2		51 kOHM 1/4W	R18, R20
[ ]	2		82 kOHM 1/4W	R17, R21
[ ]	3		100 kOHM 1/4W	R16, R22, R33
[ ]	2		150 kOHM 1/4W	R12, R26
[ ]	1		300 kOHM 1/4W	R13
[ ]	1		560 kOHM 1/4W	R46
[ ]	1		620 kOHM 1/4W	R47
[ ]	1		5.6 MOHM 1/4W	R41
[ ]	1		10 MOHM 1/4W	R25
[ ]	1		100 kOHM TRIMPOT	R28
[ ]	1	(2)	4.7 kOHM TRIMPOT	R45
[ ]	2		2 kOHM 8 PIN SIP	R34, R35

[ ]	2		68 pf	C9, C10
[ ]	1		390 pF	C11
[ ]	1		1000 pF	C1
[ ]	2		3300 PF [3n3]	C2, C6
[ ]	1		10 nf [10n]	C3, C5
[ ]	28		0.1 uF	C4, BYPASS
[ ]	2		10 uF 10V TANTALUM	
[ ]	1		.47 uF 35V ELECTROLYTIC (RADIAL)	C8
[ ]	1		150 uF 10V " "	C7

[ ]	1	SPDT MOMENTARY OR PUSH BUTTON SWITCH	S1
[ ]	1	4PDT SLIDE SWITCH	S2
[ ]	1	SPDT TOGGLE SWITCH	S3

[ ]	1	40 PIN SOCKET
[ ]	2	28 PIN SOCKET
[ ]	8	20 PIN SOCKET
[ ]	9	16 PIN SOCKET
[ ]	22	14 PIN SOCKET

[ ]	1	DUAL MALE HEADER STRIP	J1, J2, J3
[ ]		FEMALE JUMPER PINS	
[ ]	2	CASSETTE CORDS OR JACKS	J4, J5

[ ]	1	MOTHERBOARD PC BOARD

(1) RS 232 C OPTION
(2) ON BOARD PROGRAMMING SUPPLY OPTION
(3) VOLTAGE REGULATOR OPTION FOR STANDARD S100 POWER SUPPLIES (+8v,+/-16v)

There are 4 versions of the motherboard extant, REV A , REV B, REV C, REV D. Their differences are as follows: REV A was sold with the original Z80A MULTIFLEX Computer kit (It could not program 2764 or 27128; since U19, the EPROM programming socket was a 24 pin socket). REV B was the first production run of the revised motherboard. REV C corrected a few minor mistakes in REV B and a rewiring of the reset circuit. REV D may incorporate several minor design improvements.

REV D is indicated as such in the front right corner of the board next to the notation 'MONITOR BD.'. REV B is identified by examining U11 pins 9 and 10; in the case of REV B, these pins are unconnected. In REV A,U19 is a 24 pin socket.

FOR REVISION D: no changes necessary.

FOR REVISION C: If it will be necessary to program 2708s, check the layout of the PC board around R8. If there is a trace on the component side of the circuit board connecting to the right end of R8, then R8 must be moved. The indication in the silkscreen for the left end of R8 is correct. If this change is needed, then the right end should go to the feedthrough hole about 0.1 in. above U8 pin 20.

Parts C12 (3.3uF) and R48 (10Kohm) may not be indicated in the silkscreen to the right of U11. C12 goes between U11 pins 4,5 and ground, with the + lead connected to pins 4,5. R48 goes just to the right of C12, connecting between pins 4,5 and +5v.

FOR REVISION B: Parts R48 and C12 are not used on this version. The following modifications affect the EPROM programmer only.

[ ] Cut the trace on the component side of the board which, coming from a feedthrough hole to the left of U28, connects to U23 pin 9.

[ ] Connect this trace to pin 8 of U23 instead.

[ ] Cut the trace on the component side of the board which connects U27 pin 1 to a feedthrough hole about an inch to its left.

[ ] Connect pin 1 of U27 to U24 pin 6 instead.

[ ] On the solder side, identify the two parallel traces which connect to J3 pins A3 and A4.

[ ] Cut these traces and interchange them. Thus when this modification is complete, pin A3 of J3 should go to ground, and J3 pin A4 should go to U23 pin 5.

FOR REVISION A: The parts on this version are quite different from the later versions. If it is to be used with the MULTIFLEX Z80B CPU, a special monitor will be required if the cassette port and the EPROM programmer are to be used.

[ ] R2 (330Kohm) connects just to the right of R15, between the hole just to the right of R15 near its top (which connects to pin 20 of U22), and the hole at the bottom right corner of R15.

[ ] R16 (560 ohm) connects between pins 1 and 2 of U22. You will find 2 holes spaced 0.1 in. apart just to the left of the IC, into which it may be inserted.

[ ] D1 connects between the hole just to the left of (and connected to) the bottom end of R2, and the hole just at the right end of R17 (which connects to U22 pin 20). The cathode (indicated by a black band) connects to this latter hole.

[ ] C1 (3.3 uF) connects just below D1. The + end connects to the hole which on the solder side is connected to the junction of R2 and D1. The other end of C1 connects to the hole just above and to the right (component side) which on the solder side is connected to the lower end of R16.

[ ] Looking down at the top of Q1 (MPSA13) with the flat side facing left, the emmitter is the top pin, the base the middle pin, and the collector the bottom pin. The emitter connects to the hole at the end of the trace connecting to the upper end of R16 and to U22 pin 2. The base connects to the hole 0.2 in below this one (which on the solder side is connected to the junction of R2, C1 and D1). The collector connects to a hole 0.2 in. below pin 1 of U22 (which on the solder side connects to U22 pin 20).

FOR VERSION A:

[ ] Cut the trace on the solder side of the board which goes to U4 pin 2.

[ ] Join U4 pin 2 to U4 pin 3.

[ ] Cut the trace which goes to U3 pin 3 on the solder side of the board.

[ ] Join the trace which was just cut to U3 pin 2 instead.

[ ] Join the first trace to be cut to U3 pin 3 instead.

[ ] Cut the trace on the component side of the board which joins pins 2 and 4 of U3.

[ ] Join pin 1 of U3 to pin 4 of U3.

[ ] During construction of the kit, OMIT C1 and R2.

[ ] During construction of the kit, connect a diode D1 in parallel with R16, so that its cathode (indicated by the black band at one end) is closest to the S100 connector.

There are four revisions of the CPU board extant, REV A through REV D. REV C and REV D are specifically identified as such in the silkscreen at the bottom left corner of the voltage regulator. If no such identification is present, it is either REV A or B. These are distinguished by examining the part number 8202C located underneath the voltage regulator on the solder side of the circuit board. In REV B, there is a small B inscribed below this part number. (This B will also appear on REV C boards).

The differences in these boards is as follows: REV A is the first production run. REV B included a correction in the reset circuit around U3, an improved circuit for generating the signal POC (S100 pin 99), and included many of the most common jumpers prewired on the back of the board. REV C is the same as REV B, except that the silkscreen was updated to accord with these changes. REV D includes one further change in which a modification to allow for writing underneath the EPROMs into dynamic RAM is included in the board layout.

FOR VERSION D: No changes required.

FOR VERSIONS A, B, C: To improve your system performance, you may do the following change. It is required if CP/M is to be booted from VER 1.1 or 1.0 of ZMON. Its purpose is to allow the user to write into the portion of the on-board64K dynamic RAM which is normally pre-empted by the EPROM block (thus the user can set up programming in this space before disabling the EPROMs in his software). This change is also of use to those who wish to customize the monitor supplied with the kit.

[ ] On the component side of the board, cut the trace which joins together pins 9 and 12 of U7.

[ ] On the solder side, join U7 pin 12 to U8 pin 3.

[ ] Join U8 pin 2 to U7 pin 10.

[ ] On the solder side, cut the trace between U8 pin 1 and J10 pin 3.

[ ] On the component side, cut the short trace which joins U8 pin 1 to a feedthrough hole just to the left of the IC socket.

[ ] Connect J10 pin 3 directly to this feedthrough hole.

[ ] Connect U8 pin 1 to U7 pin 9.

FOR VERSION C: You may find the description below (for VERSION B) a help in locating the parts R2, R16, D1, Q1, C1. The silkscreen is correct, though a bit unclear.

FOR VERSION B:

[ ] During construction of the kit, OMIT C1 and ignore the indication in the silkscreen for R2.

The silkscreen around R16 and C5 is incorrect. Pay careful attention to the following. Do not solder in the parts until you have them all located.:

SECTION 2

ON-BOARD SOFTWARE:

This section describes the use of the MEVB-1 and lists its functions, it also illustrates the method to affect a change in the default attributes of the board. The on-board software basically performs three major functions: the interpretation of commands sent by the host processor; relaying any data from the keyboard to the host processor via a data port and finally, periodically updating the hardware components (eg. the DMA controller). The details of the software operation for the keyboard and updating the hardware are of no use tho the user and will not be discussed here. If the user is curious about these functions, he can study the software listing at the end of this manual. Suffice it to say that the software works, and only the effects need be noted. The command interpreter has basically two modes called the "command mode" and the Dbug mode". Under normal conditions only the command mode is active, however if the user wishes to change some of the default conditions (eg # of spaces/tab, type of cursor, horizontal and vertical timing), he may enter the Dbug mode and do so. The name Dbug is a bit anachronistic since it was given this name when the MEVB-1 was undergoing development and has stuck because the routines in this mode allow us an easy method to change parameters. (Note: some of the routines the user should not use or never will use are presented here only for completeness.)

Command Mode Instructions:

There are 24 instructions executable by sending the video board one of 24 control characters. These are:

HEX VALUE	CONTROL CHAR	FUNCTION
0	@	no function
1	A	Scroll down 1/2 page (disabled for 1 page operation)
2	B	Scroll up 1/2 page (disabled for 1 page operation)
3	C	no function
4	D	Switch to Dbug mode
5	E	no function
6	F	no function
7	G	no function
8	H	Non-destructive backspace
9	I	Tab
A 10	J	Line Feed (LF)
B 11	K	Move cursor up one row
C 12	L	Non-destructive forward space
D 13	M	Carriage Return (CR)
E 14	N	Blink
F 15	O	Reverse video
10 16	P	Underline
11 17	Q	Highlight

HEX VALUE	CONTROL CHAR	FUNCTION
12	R	Concatenation of field attributes
13	S	Stop all field attributes
14	T	CR, LF, and clear to end of line
15	U	Go to page 0 and home cursor
16	V	Send cursor position (column, row)
17	W	Disable control interpreter for next char.
18	X	Reset control status register
19	Y	Send char. at cursor location
1A	Z	Clear present screen and home
1B,3D	[=	Addressable cursor position
1C	\	Graphic field attribute
1D	]	Initiate graphics operation
1E	^	Home cursor
1F	←	No function

Along with these instructions, the video board can accept any one of the ASCII characters, display it and automatically increment the cursor or issue a line feed, carriage return (if necessary). The most significant bit of any character must not be set since this will indicate a special code to the 8275 CRT controller and strange results may occur. Before discussing the individual instructions of the command mode, the list of instructions for the Dbug mode will be given. Note that most are the same as the command mode.

HEX VALUE	CONTROL CHAR	FUNCTION
0	@	no function
1	A	Same as command mode
2	B	-"- -"-
3	C	Go to command mode
4	D	Reenter Dbug (be careful, this is recursive)
5	E	Set up # spaces per TAB
6	F	Set up cursor type (default = 8)
7	G	Set up horizontal & vertical timing (DO NOT USE!!!)
8	H	Same as command mode
9	I	-"- -"-
A	J	-"- -"-
B	K	-"- -"-
C	L	-"- -"-
D	M	-"- -"-
E	N	Set up row at which scrolling occurs
F	O	Start "one page" operation
10	P	Start "three page" operation
11	Q	Same as command mode
12	R	-"- -"-
13	S	-"- -"-
14	T	-"- -"-
15	U	-"- -"-
16	V	-"- -"-
17	W	-"- -"-
18	X	-"- -"-

HEX VALUE	CONTROL CHAR	FUNCTION
19	Y	Observe an on-board memory location
1A	Z	Same as command mode
1B,3D	[=	-"- -"-
1C	\	-"- -"-
1D	]	-"- -"-
1E	^	-"- -"-
1F	←	No function

Command Mode Function Description:

[1] <CTRL>A: Scroll 1/2 page (disabled for one page operation).
The MEVB-1 has 3 1/2 pages of on-board memory storage (one page being described as a screen full of characters). Thus to allow the user to step forward through the pages of memory 1/2 page at a time, this function was implemented. When the board is set up for "one page" of on-board memory, then this function is disabled.

[2] <CTRL>B: Scroll down 1/2 page
This function is virtually the same as CTRL A, except that is operates in the opposite direction.

[3] <CTRL>D: Enter Dbug Mode
Once this command is issued, the MEVB-1 switches to Dbug mode. (See Dbug Mode Commands).

[4] <CTRL>H: Non-destructive Backspace.
This instruction will move the cursor back one column and will not affect any character that may be present at the cursor position. The cursor will not backspace past the first column.

[5] <CTRL>I: Tab
Tab "marks" are at every 8 columns. When this instruction is issued the cursor will be moved to the next right tab mark. An exception to this rule occurs if the cursor is in the last few columns, then the next tab mark would be off the screen, in which case the cursor moves to the next left tab mark.

[6] <CTRL>J: Line Feed
The line feed command will move the cursor down one row of text, that is either the cursor could be moved down by one row and the text moves up by one row (ie. a 1 row scroll takes place). Under default conditions, the latter normally occurs when the cursor is at row 24 (ie. the last row of the display). If a scroll did not occur the cursor would go off the screen. (Note that the point at which the scroll ocurrs can be changed in the Dbug mode).

[7] <CTRL>K: Upline

This command is the opposite of the line feed command. It moves the cursor up one row. The only difference is that if the cursor is in the top row and the screen is displaying the first page, the command is ignored.

[8] <CTRL>L: Non-destructive Forward Space

This is the opposite of the <CTRL>H command. It moves the cursor forward one space while not affecting any character the cursor may pass over. This commands does wrap arround (ie. if the cursor is in column 80 and a <CTRL>L is received, the cursor will go one row down and be in column 1).

[9] <CTRL>M: Carriage Return

This command causes the cursor to return to column 1 in the current row.

[10] <CTRL>N: Blink

This is a field attribute which causes the video board to put a special character in memory. When the 8275 sees this character, it will interpret it and set up the blinking circuitry. Thus every character after this instruction is issued will blink until another field attribute is encountered.

[11] <CTRL>O: Reverse Video

This is another field attribute which causes the characters to be displayed in reverse video.

[12] <CTRL>P: Underline

This is another field attribute which causes the characters to be displayed with an underline. Note that the underline is on line 8, and if a solid underlining cursor is chosen (in the Dbug mode), it will be impossible to discern the cursor from the underline.

[13] <CTRL>Q: Highlight

This is a field attribute which causes the characters to be displayed in an intensified (highlighted) form.

[14] <CTRL>R: Concatenate Feild Attributes

This control function allows any one of the five field attributes to be concatenated together. The actual interpretaion of this attribute is that it marks the beginning and end of a concatenation sequence. Example: suppose that a blinking, underlined character is desired. This could be accomplished by the following:

<CTRL>R (begin concatenation)
<CTRL>N (set blinking)
<CTRL>P (set underline)
<CTRL>R (end concatenation)

NOTE: Each sequence MUST begin AND end with <CTRL>R. If the second <CTRL>R is forgotten, the video board will misinterpret your intentions, and this will cause many unexpected results in the display.

[15] <CTRL>S: Stop All Attributes
This instruction will cause the effects of all field attributes to stop after this character is recieved.

[16] <CTRL>T: CR, LF and clear to end of screen
This command goes to the beginning of the next line and blanks that line.

[17] <CTRL>U: Go to Page 0 and Home Cursor
This causes the display to show the first page of text, and places the cursor in the upper left hand corner of the display.

[18] <CTRL>V: Send Cursor Position
Upon recieving this command, the MEVB-1 responds by transmitting to the host CPU the column number and then the row number of the cursor. The Home position (the upper left hand corner of the screen) is used as the base location and is called (0,0). The data is sent out through the data I/O port.

[19] <CTRL>W: Disable Command Interpreter
This command allows a control character to be sent to the screen without it being intercepted and interpreted by the command processor on the video board. The control character which follows immediately after the <CTRL>W will be displayed on the screen as a control character. For example if the user wanted a <CTRL>A to be displayed on the screen, then he should send a:

<CTRL>W;<CTRL>A.

[20] <CTRL>X: Reset Control Status Register
In many of the commands more than one character is required to complete the sequence, thus the machine is writen such that the board enters a new state upon the recipit of a command or data (eg. <CTRL>R). For these commands, it is necessary to maintain information pertaining to the state of the machine, hence the Control Status Register. By clearing this register, the effect is to return the board to state 0, that is the beginning of any command sequence. Under normal conditions, it will not be necessary to use this command, however in the event of a program sending garbage to the MEVB-1, the software may enter some unknown state and it is wise to issue this command after control is obtained by the host processor.

[21] <CTRL>Y: Send Character at Cursor Location
This command causes the ASCII representation of the character at the cursor position to sent to the host processor through the data I/O port.

[22] <CTRL>Z: Clear Present Screen and Home Cursor
This causes the 24 lines being presently displayed to be blanked and places the cursor in the upper left hand corner of the display.

[23] <CTRL>[=: Addressable Cursor Postion
When these two characters are sent, and followed by first a row number and then a column number, the cursor will be placed in that location. Note that the row and column number have an offset of 20H (which corresponds to a "blank" in ASCII), so that to send the cursor to loaction (0,0), the following sequence would need to be sent:

<CTRL>[;=;20H;20H

[24] <CTRL>\: Graphics Field Attribute
After recieving this character, the video board switches off the character generator and switches on the graphics generator for each character on the screen until another field attribute is detected.

[25] <CTRL>]: Initate Graphics
The 8275 display controller will, on its own accord, blank the first and last lines comming out of the character generator. This is fine for alphanumerics, but it reeks havoc upon graphics. So to circumvent this problem, this command was created. It reinitalizes the 8275 in such a way that this does not happen. To help the user determine whether or not he is in graphics mode, the cursor type automatically becomes type 1 (a solid, reverse block). A side effect of this command is that the underline is then positioned on line 7 instead of line 8 of the character. To get out of this mode, send another <CTRL>] to the MEVB-1.

[26] <CTRL>^: Home Cursor
This instruction place the cursor at the upper left hand corner of the screen (ie. position (0,0)).

Dbug Mode Command Description:

The Dbug mode will mormally be used only when the user wishes to change some of the default conditions on his board because of software "taste" or due to a piece of software which requires that the board be set up in a particular way. Some of the functions provided should not be used and are listed here only for completeness. Such a function is "Set up horizontal and

vertical timing." If the user really must use this function, then he should consult a data sheet on the 8275 and learn the functions of the "reset parameters", since this is what is altered by this function.

Most of the functions supplied in the Dbug mode are the same as the command mode, so only the command where there is a difference are listed:
NOTE: In this mode the keyboard is in a local mode, that is nothing is sent to the host CPU.

[a] <CTRL>C: Exit Dbug
This returns the user to the command level.

[b] <CTRL>D: Enter Dbug Mode
While this command seems redundant, it is in fact extremely useful. Since every time one of the alter parameter instructions is completed the board returns to the command level, this instruction can be used to call Dbug recursivly so that 2 or 3 changes can be made with out returning to the command mode. Note that since this command uses the stack of the Z80 processor, abuse of this command can cause overwritng of the display memory.

[c] <CTRL>E: Set up Number of Spaces per TAB mark.
This allow the user to set number of spaces between each TAB mark to his own liking. After a <CTRL>E is issued, the MEVB-1 waits for an ASCII character, and subtracts 30H from it. This then becomes the new spacing. Thus if an ASCII "5" is sent from the keyboard, the spacing will be 5, and if a "?" is sent the spacing will be 15.

[d] <CTRL>F: Set up Cursor Type
This allows for different types of cursor formats such as listed below. The sequence is then:

<CTRL>F;cursor type number

TYPE NUMBER	CURSOR FORMAT
0	Blinking reverse video block
1	Solid " " "
2	Blinking underline
3	Solid underline

The default type is type 0. Note that if a solid underline is chosen, and the screen is under the effects of the underline field attribute, that the cursor will be indistinguishable from the underline.

[e] <CTRL>G: Set up Horizontal and Vertical Timing
This command should not be used, it is only presented here for completeness. If the user is curious about what this command does, he should look at the data on

the 8275 (especially on the reset parameters), and the software listing later in this manual.

[f] <CTRL>N: Set up row at which scrolling occurs
Normally, the MVB-1 will automatically scroll when the cursor tries to go past the last line on the screen. It is, however, possible to change the line number at which this occurs. This allows the user to have a few lines at the bottom of the screen permenantly. The row number should be entered as two BCD numbers, the combination of which is a number one less than the desired row number. As an example, to scroll at row 21, the sequence would be:

<CTRL>N;2;0.

[g] <CTRL>O: Set up "one-page" operation
Because many video boards only have 1 page of on-board memory, the screen editors written for them cannot adjust to having more than one page. Thus, this instruction sets up the MEVB-1 to act as if it has only one page of memory.

[h] <CTRL>P: Set up "three-page" operation
This is the complement operation to <CTRL>O. It activates the full 3 1/2 pages of memory on the board.

[i] <CTRL>Y: Observe an on-board memory location
The user has no real use for this instruction. It was implemented for use in the debugging of the MEVB-1. If the user is curious about its operation, he should consult the software listing.

MVEB-1 HARDWARE DESCRIPTION

The basis of operation of this board is three LSI chips: a Z-80A, a 8275, and a 8257. The Z-80A performs all the command interpreting required as well as polling the keyboard for data periodically and updating certain parameters in the 8275 and the 8257. The 8257 is a DMA controller that gets its requests form the 8275, where upon the 8257 "asks" the Z-80A for the data, address and control bus. After the 8257 has supplied the 8257 with 80 more characters, it relinquishes control back to the Z-80A. Because DMA is used to refresh the 8275, then the Z-80A can preform the task of command interpreting and data transfer from the data port very quickly, probably faster than any other board in the same class.

The 8275 is what supplies all the timming, DMA requests, and special field attributes. Basically, the 8275 asks the DMA controller for 80 characters. After it has recieved these characters, the 8275 starts displaying them one by one and at the proper time telling the video circuits when to indicate a cursor or initiate any of the field attributes. A block diagram of the MEVB-1 structure is shown below.

KEYBOARD INTERFACE SPECIFICATIONS

The keybaord data and handshaking signals are available through a standard 16 pin I.C. socket. The signal definition is:

PIN #	DEFINITION
1	Ground
2	"
3	"
4	Data 5
5	Data 7
6	Data 6
7	Vcc (+ 5V)
8	Ready, a positive going signal indicating that valid data is available on the data lines.
9	No connection
10	Vcc (+5V)
11	"
12	Data 4
13	Data 0
14	Data 1
15	Data 2
16	Data 3

STATUS PORT DEFINITION

BIT#	MEANING
0	not used
1	Data available from MEVB-1 (active high)
2	Data NOT taken from CPU (active high)
3	not used
4	not used
5	not used
6	not used
7	not used

PARTS LIST

	QTY	PART	LOCATION(S)
IC'S:			
[ ]	1	8275	U12
[ ]	1	8257	U16
[ ]	1	Z80 CPU	U15
[ ]	2	2732	U17 ("VDB3"), U22 ("KVDBCG")
[ ]	4	6116	U18-U21
[ ]	1	74LS00	U28
[ ]	2	74LS02	U14, U29
[ ]	4	74LS04	U8, U24, U25, U39
[ ]	1	74LS08	U27
[ ]	1	74LS10	U35
[ ]	2	74LS32	U1, U6
[ ]	2	74LS74	U9, U36
[ ]	1	74LS86	U38
[ ]	1	74LS123	U10
[ ]	1	74LS125	U37
[ ]	2	74LS139	U7, U30
[ ]	1	74LS161	U11
[ ]	1	74LS165	U23
[ ]	2	74LS266	U31
[ ]	1	74LS367	U26
[ ]	3	74LS373	U2, U3, U4
[ ]	4	74LS374	U5, U13, U33, U34
CAPACITORS:			
[ ]	18	.1 µF	C1-C3, C7-C11, C14-C19, C21-C24
[ ]	2	3.3 µF	C12, C13
[ ]	1	22 µF	C20
[ ]	1	10 pF	C5
[ ]	1	100 pF	C6
[ ]	1	470 pF	C4
RESISTORS:			
[ ]	1	110 ohm	R10
[ ]	1	330 "	R16
[ ]	2	390 "	R12, R13
[ ]	2	510 "	R4, R5
[ ]	2	1 Kohm	R3, R15
[ ]	1	1.8 "	R7
[ ]	1	2 "	R6
[ ]	2	2.4 "	R8, R9
[ ]	1	5.1 "	R2
[ ]	1	10 "	R1
[ ]	1	20 "	R14
[ ]	1	68 "	R11
[ ]	1	2 Kohm 10 pin SIP	RN2
[ ]	1	10 " " " "	RN1

MISCELANEOUS:

[ ]	1	78H05	U40 (optional)
[ ]	1	14.0000 MHz XTAL	Y1
[ ]	2	2N4400/PN2222	Q1, Q2
[ ]	1	2N4402/2N3906	Q3
[ ]	1	RCA PC mount conn.	COD1

SOCKETS:

[ ]	16	14 pin sockets
[ ]	6	16 pin sockets
[ ]	7	20 pin sockets
[ ]	3	40 pin sockets

MODIFICATIONS

IN THE UPPER LEFT HAND CORNER OF THE COMPONENT SIDE OF THE VIDEO BOARD IS THE NOTATION 'VIDEO BD.'. REV.B IS IDENTIFIED AS SUCH HERE. REV.A WILL HAVE NO REV INDICATED.

TO REV.B - NO CHANGES
TO REV.A - SEE BELOW

WITHOUT THIS MODIFICATION, THE DATA PORT MUST NOT BE LOADED UNTIL AT LEAST 15 USEC AFTER STATUS BIT 2 GOES LOW (INDICATING THAT THE BOARD IS FREE TO ACCEPT DATA). THIS MODIFICATION MAKES SUCH A WAIT UNNECESSARY.

- JUST BELOW THE RECTANGLE WHICH INDICATES U17 IN THE SILKSCREEN, IS A FEEDTHROUGH HOLE WHICH CARRIES A SIGNAL FROM U6 PIN 8 TO U36 PIN 1.
- CUT THE TRACE ON THE SOLDER SIDE OF THE BOARD WHICH CONNECTS THIS FEEDTHROUGH TO U6 PIN 8.
- WITH A SHORT WIRE CONNECT THIS FEEDTHROUGH HOLE TO U30 PIN 9.
- DO A VISUAL CHECK OF THIS CHANGE.

MULTIFLEX Z80 COMPUTER KIT Q.C. AUDIT

The following packages MUST be present in every MULTIFLEX Z80 computer kit.

	QTY	ITEM
[ ]	1	Motherboard
[ ]	1	CPU Card
[ ]	1	S-100 connector
[ ]	1 bag	Bypass capacitors
[ ]	1 bag	Motherboard discrete components
[ ]	1 bag	CPU discrete components
[ ]	1 pack	Motherboard sockets
[ ]	1 pack	CPU sockets
[ ]	1 pack	Keyswitches
[ ]	1 pack	Keycaps and labels
[ ]	1 set	MOS chips
[ ]	1 set	Motherboard TTL chips
[ ]	1 set	CPU TTL chips
[ ]	1 pack	Headers and jumpers
[ ]	1 pack	Displays
[ ]	1 set	Documentation

Initials of Q.C. inspector: _____

Date: _____ _____ , 19____

PREFACE

The purpose of this manual is two-fold. First, it provides complete assembly instructions for the Multiflex Z80A Computer Kit, and second, it is a reference volume for users of the finished computer.

No attempt is made to teach or explain the principles of programming or details of the Z80A-CPU chip. These subjects are most adequately covered in many books and any attempt to summarize them here would only hinder understanding. The following works are recommended for users unfamiliar with these matters:

The Z-80 Microcomputer Handbook
by William Barden, Jr.
Pub. by Howard W. Sams Co., # 21500

Programming the Z80
by Rodnay Zaks
Pub. by Sybex, Inc. # C-280

The Multiflex Z80A Computer, with its monitor, is designed to supplement such works for new users, by making machine-language programming as easy and efficient as possible.

The engineering staff of Multiflex, Inc. are most interested in the suggestions and comments of users of the Z80A system. If you would like to make any suggestions as to possible improvements of the system, please direct your letters to

Multiflex Technology Development, Inc.
319 College Street,
Toronto, Ontario
M5T 1S2

Correspondence is invited in either of Canada's official languages.

Multiflex is planning an international user's group for Z80A system owners, and you will receive information about this within the next few weeks. As of this writing, the exact form of this group is not certain, but monthly bulletins, special events, and a software exchange will certainly be among the benefits available to group members.

The staff of Multiflex thanks you for purchasing the Z80A computer system, and wishes you the best in working with it.

Sincerely,

Eugen F. Hutka
President
Multiflex Technology Develpment, Inc.

INTRODUCTION

The MULTIFLEX Z80B Computer Kit* is the successor to our original Z80A Computer Kit, with expanded hardware features and software capability. This now includes the ability to run all CP/M based software. Both were designed for users unfamiliar with the Z80 CPU chip to learn about the Z80 instructions and architecture, and for experienced users to develop Z80 based systems to meet their product requirements. If used in conjunction with compatible (and of course, we reccommend our own MULTIFLEX products) S100 boards, it will serve as part of an extremely flexible, standalone microcomputer system for that range of users who wish to compute without learning hardware details. The kit can be configured in several ways, with great expansion capability on its S-100 bus.

(For users wishing to run CP/M, they require in addition a disk drive and floppy controller board with appropriate supporting software, and a terminal or video board, keyboard, and monitor. Compatible CP/M and all this hardware is available from MULTIFLEX).

The basic kit is a two board system. The CPU board is inserted into an S100 connector on the Monitor/Motherboard**, which has provisions for 3 further S100 connectors.

The CPU board is a self contained microcomputer with fully bufferred S100 bus interface. Its features are as follows:

- Versatile monitor program simplifies entry of machine code software. It includes a download routine, and CP/M boot for users with the MULTIFLEX floppy board.
- 24 bits of addressing (extended addressing), permits the use of an address space as large as 16 megabytes.
- Provision for resident RAM at the top of each 64K page of the total address space. Size selectable by jumper.
- Up to 64K of RAM on board.
- Up to 32K of EPROM if 2764 devices are used. 2716 and 2732 devices are simultaneously compatible. Size and number of EPROMs and location of EPROM block jumper selectable.
- Software EPROM disable/enable.
- Provision for memory management via optional piggy back board permits reordering of memory by 4K blocks.
- Provision for real-time clock on optional piggy back board.
- Provision for complete interrupt prioritization on optional piggy back board.
- Capable of 6 Mhz operation, it is supplied crystal controlled at 4 Mhz.
- Provision for on board voltage regulators for S100 voltage specifications.
- Optional on-board 8255 parallel port.
- Provision for two RS 232 C serial ports via a small optional piggy back board.
- Optional on-board 3 channel 8253 programmable timer, permitting software controllable baud rates for the serial ports.

The Motherboard provides "front panel" control, and the flexibility to expand the system or develop hardware:

- 4 S100 connectors (3 are optional) to allow expansion of the kit to a full microcomputor system.
- Six digit hex LED display.
- Hex keypad and 16 function keys to implement control functions with the MULTIFLEX Z80 monitor software.
- EPROM programmer which can handle all popular EPROMs on the market.
- 8255 parallel port with 24 I/O bit lines.
- Optional serial RS 232 C port, with crystal controlled, jumper selectable baud rates, RS 232 C driver and receiver chips.
- 2 Kbaud synchronous cassette interface.
- Space for optional power supply regulators to supply all voltages according to S100 specifications.
- Optional on-board voltage converter to supply +21 and 25 volts for programming EPROMs.
- Wirewrap area for customizing the board, and custom hardware development.

* The term "kit" as used here, refers to this two board system, containing motherboard and CPU, whether it is purchased as a kit or assembled and tested.

** Monitor/Motherboard refers to the larger of these two boards containing the I/O systems and S100 backplane. Hereafter we will just call it the "motherboard".

The Monitor program normally resides in a 2732 EPROM in socket U39 on the CPU board. It provides complete control of the I/O systems and aids in the creation, manipulation, and execution of machine language programs. (This monitor is also available in two 2716 EPROMs for users who would use it in this configuration).

With these updated and improved boards, Z.MON. VERSION 1, 2, or 3 is necessary for correct use of the cassette interface and EPROM programmer, and to initialize the extended address. Several variants of Z.MON. are available to suit the differring memory maps required by kits with and without the full 64K of memory on-board. See the memory maps later in this manual for full details.

Version 1.0 resides at 1000 and uses memory between 0000 and 03FF for monitor functions. It is necessary with the basic kit which does not include the dynamic RAM. (It will work as well with the dynamic RAM, but if the extended addressing is to be used, use a 6116 in U36, so that the stack, monitor registers, and NMI vector are not lost when the extended address is changed).

Version 2.0 resides at F000 and uses memory at EF00 to EFFF for the stack, and for other variables used by the monitor. This version is used with the full 64K option when no more than two user EPROMs will be used in addition to the monitor.

Version 3.0 resides at F000 and uses memory between BF00 and BFFF. It is of use to those who have 64K of memory but require 3 user EPROMs.

Z80 CPU BOARD JUMPERS
J3,J4,J6,J7,J8,J9,J10,J11,J12,J13,J14,J15,J16,J17,J18

Note #1: With the exception of J10, jumper pins are labeled by pairs (1st pair, 2nd pair, etc.) according to the option which is implemented when a jumper is installed between a pair. The first pair is indicated by a dot in the silkscreen. It is at the left hand side of a horizontal set on top of a vertical set.

Note #2: To disable any jumpers at J11 or J17 a trace on the wiring side of the circuit board must be cut. Normally the jumpers at J11 are left installed and the one at J17 must be cut.

Note #3: Pin headers are provided for J13,J14, and J16. Headers may be added at the other jumper locations by the user as the need arises.

J3 SERIAL PORT (located on optional piggy-back board)

1 PCICK1 = TQ 1
2 PCICK1 = TQ 0

The baud rate clock for serial port #0 (PCICK0) is generated by TQ 0 from the 8253 PROGRAMMABLE INTERVAL TIMER. The 8251A PROGRAMMABLE COMMUNICATIONS INTERFACE divides its clock input PCICK internally by 1, 16, or 64 to generate its bit rate. J3 selects the source of PCICK1, the clock input for serial port #1.

For the same baud rate on serial ports #1 and #0, use J3-2: PCICK1 = TQ 0 and remember to use the same internal divider for each 8251A P.C.I.

To generate an independant baud rate for serial port #1, use J3-1: and use channel 1 of the 8253 P.I.T. to control the baud rate.

CAUTION: DO NOT ENABLE BOTH JUMPERS J3-1 AND J3-2 AT THE SAME TIME AS THIS CAN CAUSE DAMAGE TO THE CIRCUITRY.

J4 PROGRAMMABLE INTERVAL TIMER (PIT) CLOCK INPUT SELECT

1 TC 2=CLK
2 TC 1=CLK
3 TC 0=CLK

The clock inputs to the 8253 PROGRAMMABLE INTERFACE TIMER may be externally generated or jumpered to CLK, the on-board 2 MHz clock signal, using J4.

1 J4-1 selects the 2 MHz clock for channel 2 of the P.I.T.

2 J4-2 selects the 2 MHz clock for channel 1 of the P.I.T.
3 J4-3 selects the 2 MHz clock for channel 0 of the P.I.T.

J6 PORT BASE ADDRESS SELECT

J6-1 A7 (MSB)
J6-2 A6
J6-3 A5
J6-4 A4

The various ports occupied by the CPU board occupy 16 contiguous port addresses starting at the base address XOH (X is selectable by the user). The upper nibble of the base address is selected by jumpering J6. Inserting a jumper causes a zero in the bit.

J7 EPROM BLOCK BASE ADDRESS SELECT

J7-1 A15 (MSB)
J7-2 A14
J7-3 A13

The EPROM sockets on the CPU board occupy one block of memory address starting at the base address XOOOH (where X is a user selectable even number between OH and EH. The three MSBs of this address are set by the 3 jumpers of J7. For example, with all jumpers in place the EPROM addresses begin at OOOOH.

J8 CLOCK OPTION #1; EXTENDED ADDRESSING

1 CLOCK (S-100 pin 49) = $\underline{\Phi} \div 2$ ($\underline{\Phi}$ = Z80 CPU clock)
2 EXTENDED ADDRESSING ENABLE
3 RESIDENT RAM DISTRIBUTION ENABLE

1 To generate the clock signal on the S-100 bus from the CPU board use either the J8-1(CLOCK = $\underline{\Phi} \div 2$) or J17-1(CLOCK=CLK) jumper. Do not use both. If the clock signal is to originate on a different board on the bus, then use neither jumper.

2 When this jumper is not in place, the extended address register (loaded as a port) is ignored and the upper 8-bit (A23-A16) of the address are not decoded on the CPU card. When it is in place, the memory address range of the CPU card is FF0000H to FFFFFFH and the lower (000000H - FEFFFFH) address area is mostly available to memory cards on the same S-100 bus. Memory in this lower area could also be taken up by resident RAM (see J8-3,J12) or EPROMs.

3 When resident RAM distribution is enabled, the on-board

memory is addressed in a certain amount of address space at the top of any 64K section of memory regardless of the extended address lines. J12 sets the lower limit of the resident RAM. Extended addressing enable must be installed to take advantage of this feature.

J9,J14 EPROM ADDRESS DECODING SELECT

EPROM BLOCK SIZE

		32K (8K)	16K (4K)	8K (2K)
J9	1		X	X
	2			X
J14	1	X		
	2		X	
	3			X
	4	X		
	5		X	
	6			X

X indicates a jumper is installed.

(xK) is the maximum chip size which can be used in any socket. Smaller EPROMs or RAMs may be used. These will have several images in the EPROM space, as one or more address lines are ignored.

To decode for different sizes of EPROM, different address lines must be chosen from A11-A14. Note that the starting address of the EPROM block of memory was set by J7 and that no addresses below that base address are available to EPROMs. If the base address is set too high, some of the EPROM memory may be unusable. First determine the maximum amount of memory that is to be occupied by EPROMs, then refer to the table above. Note that there are 4 EPROM sockets available.

2764 = 8 KBytes each
2732 = 4 KBytes each
2716 = 2 KBytes each

J10 MEMORY CYCLE WAIT SELECT

NOTE: A NUMBER IS GIVEN TO EACH PIN RATHER THAN EACH PAIR OF PINS. PIN #3 SHOULD BE CONNECTED TO ONE OF THE OTHERS.

1 MREQ
2 GROUND
3 COMMON
4 M1

When using normal fast (200ns) RAM, tie J10-3 (COMMON) to J10-2 ground. If using slower memory, tie J10-3 (COMMON) to J10-4 (M1) or J10-1 (MREQ). Tying COMMON to M1 slows down the instruction fetch cycle, which is faster than the other memory cycles. The COMMON to MREQ slows down all memory cycles regardless. This will slow down operation on all memory boards in the system.

J11 S-100 BUS OPTIONS

NOTE: TRACES ON THE WIRING SIDE OF THE BOARD MUST BE CUT TO DISABLE ANY OF THESE OPTIONS.

1 MREQ* ENABLE (not an IEEE 696 siganl) allows a memory request to be sent out on the S-100 bus.

2 RFSH* ENABLE (not an IEEE 696 SIGNAL) allows a refresh signal to be sent out on the S-100 bus.

3 PHANTOM* ENABLE allows the on-board RAM to be disabled, so that data can be read from other memory in the system.

J12 RESIDENT RAM BASE ADDRESS SELECT (in increments of 2K)

NOTE: BOTH J8-2 AND J8-3 MUST BE INSTALLED OR THIS JUMPER SET IS IGNORED.

1 A14 = 0
2 A13 = 0
3 A12 = 0
4 A11 = 0

These jumpers set the base address of the resident RAM. All addresses between this and FFFF in any 64K section of memory will access the on-board RAM. A15 of this base address is set to 1 by hardware. As an example, if all 4 jumpers are installed the resident RAM will be accessed by all addresses in the range 8000H to FFFFH regardless of the extended address lines in every 64K section of memory.

J13 EPROM SOCKET ENABLE

1 U39 ENABLE
2 U38 ENABLE
3 U37 ENABLE
4 U36 ENABLE

These jumpers enable or conceal the EPROMs in the listed sockets from the CPU.

J15 BOOT ADDRESS SELECT

1 A15 = 0
2 A14 = 0
3 A13 = 0
4 A12 = 0
5 A11 = 0
6 A10 = 0

These jumpers set the address from which the CPU loads the first instruction upon start-up. When all 6 jumpers are installed the boot address is 0000H. The lower address lines are set to 0 for start-up.

J16 EPROM SOCKET PERSONALITY SELECT

0 U36
1 U37
2 U38
3 U39

A 2716
B 6116
C 2732 or 2764

These jumpers accomadate the different pinouts for the different types of chips which can be installed in the EPROM sockets. The array of 12 jumpers is set on a per socket basis to select the correct signal for pin #22 of each socket.

J17 CLOCK OPTION #2

NOTE: TO DISABLE THIS OPTION, A TRACE MUST BE CUT ON THE WIRING SIDE OF THE CPU BOARD.

If the clock signal on the S-100 bus is to originate on the CPU board, select EITHER J17 or J8-1 but NOT both. If the clock signal is to originate elsewhere or not at all, select neither of these jumpers. CLK is an on-board 2 MHz signal (see J8-1).

J18 EXTRA MEMORY MANAGEMENT DISABLE

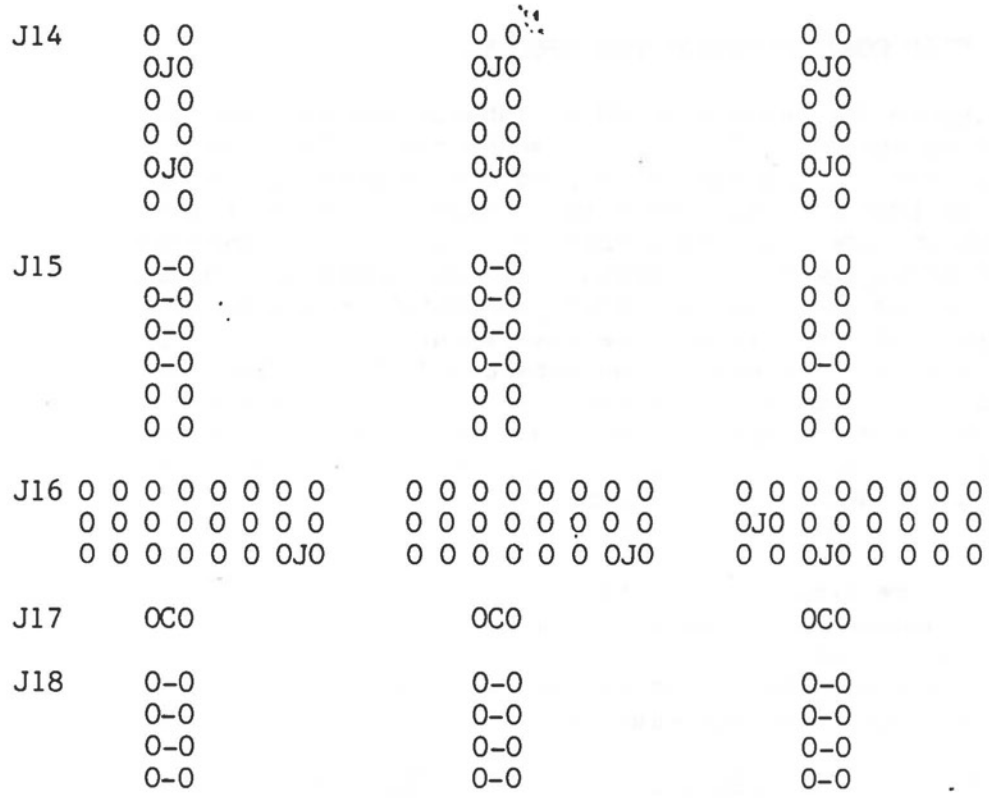
All J18 jumpers are normally in use. They are disabled only for the memory management provided by an optional piggy-back board.

BASIC JUMPER CONFIGUTRATION FOR Z80 CPU

This is a guide to the jumpers which you should install when you are constructing the kit. It is by no means the only possible configuration, but is a guide to help you first getting the kit operational. So that you may change the jumpers later, make them out of a loop of wire (such as a resistor lead) which extends about 1/4 inch above the circuit board. If you expect to change them often, then we would suggest that you install extra headers for these jumpers of the sort that we have supplied for J12, J13, J14, and J15. VER below refers to the version of the monitor you have received in your kit (see the end of the introduction earlier in this manual). Any J's such as J5 which are not in the list below are either immaterial at this point, or are not jumpers, but rather are I/O connectors.

- indicates a wire jumper to be inserted.
J indicates a jumper to be installed on a header.
X indicates doesn't matter.
C indicates a trace to be cut on the solder side.
T indicates a trace on solder side to be left in.

	VER 2.0	VER 3.0	VER 1.0
J6	0 0 0-0 0-0 0-0	0 0 0-0 0-0 0-0	0 0 0-0 0-0 0-0
J7	0 0 0 0 0 0	0 0 0 0 0 0	0-0 0-0 0-0
J8	0-0 0-0 0-0	0-0 0-0 0-0	0-0 0 0 0 0
J9	0-0 0 0	0-0 0 0	0-0 0 0
J10	0 0-0 0	0 0-0 0	0 0-0 0
J11	0 0 0 T T T 0 0 0	0 0 0 T T T 0 0 0	0 0 0 T T T 0 0 0
J12	0 0 0 0 0-0 0 0	0-0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
J13	0J0 0 0 0X0 0X0	0J0 0X0 0X0 0X0	0XO 0XO 0JO 0JO


J14
J15
J16
J17
J18

Before starting assembly, use an ohmmeter to make sure that there are no shorts between between any of the voltages and ground on either the CPU or MOTHERBOARD.

Some of the areas on these board contain dense wiring, and especially there it is important that the soldering be done carefully to avoid solder shorts. The solder should be a fine, rosin core type meant for electronic use, and it should be applied with a fine tip iron and not with a soldering gun. Be careful not to overheat the traces on the PC board for this will destroy their adhesion to the board and leaves them susceptible to breakage. At the same time heat the joint sufficiently for the solder to flow smoothly into the joimt. A dull-looking joint often is a poor one.

Do not hurry the construction, and maintain a high standard of craftmanship. This is essential. It makes the unit far more likely to work when you apply power, and also a lot simpler and less expensive to service.

CPU CARD ASSEMBLY

Before assembling the CPU examine the addendum or modification sheet (if any) and perform that work first.

Start assembly of the CPU card by inserting all the resistors. The silk screen on the board shows the location of each resistor. On the board, pin #1 of a SIP (single inline package) is indicated by a bar at one end of the silkscreen pattern. Pin #1 on the SIP itself is indicated by a notch or indentation at one end; on some types, this may be replaced by a dot or bar printed onto the SIP. If you are not sure which is pin #1 on the SIP a simple test with an ohmmeter will indicate pin #1. In all the SIPs used (except R24,R25,R26) pin #1 is common to all the resistors in the package. Thus a reading from between the first pin and any other pin should give a value equal to the specified value of the SIP.

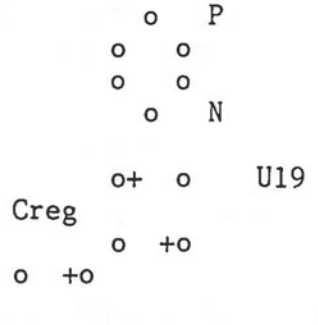
For each of R24, R25, and R26, use 4 resistors, standing up on the board so that the first is soldered in as pins 1 and 2, the second as pins 3 and 4,the 3rd as pins 5 and 6, the 4th as pins 7 and 8.

[ ] R1 8 PIN SIP 2 kOHM (4308-101-202)
[ ] R2 2 kOHM 1/4 W
[ ] R3 2 kOHM 1/4 W
[ ] R4 10 kOHM 1/4 W
[ ] R5 10 kOHM 1/4 W
[ ] R6 6 PIN SIP 10 kOHM (4306-101-103)
[ ] R7 1 kOHM 1/4 W
[ ] R8 8 PIN SIP 2 kOHM (4308-101-202)
[ ] R9 2 kOHM 1/4 W
[ ] R10 8 PIN SIP 2 kOHM (4308-101-202)
[ ] R11 2 kOHM 1/4 W

[ ] R12 2 kOHM 1/4 W
[ ] R13 2 kOHM 1/4 W
[ ] R14 330 OHM 1/4 W
[ ] R15 8 PIN SIP 2 kOHM (4308-101-202)
[ ] R16 33 kOHM 1/4 W
[ ] R17 8 PIN SIP 2 kOHM (4308-101-202)
[ ] R18 8 PIN SIP 2 kOHM (4308-101-202)
[ ] R19 6 PIN SIP 10 kOHM (4306-101-103)
[ ] R20 330 OHM 1/4 W
[ ] R21 10 PIN SIP 27 kOHM (4310R-101-273)
[ ] R22 2 kOHM 1/4 W
[ ] R23 2 kOHM 1/4 W
[ ] R24 4 x 68 OHM 1/4 W
[ ] R25 4 x 68 OHM 1/4 W
[ ] R26 4 x 68 OHM 1/4 W
[ ] R27 1 kOHM 1/4 W
[ ] R28 1 kOHM 1/4 W
[ ] R29 2.7 kOHM 1/4 W (Optional, may improve clock symmetry at 6 Mhz.)
[ ] R30 1 kOHM 1/4 W
[ ] R31 1 kOHM 1/4 W

Next install the capacitors where they are indicated on the silkscreen. Be careful to observe the proper polarity of the tantalum capacitors, they are almost immmediately destroyed by a reverse voltage, becomming a short across the power supply thereafter. The polarity of the 3 tantalums Creg is not clearly marked. These go by the voltage regulators in the bottom left hand side of the component side of the CPU board. (See the diagram below). The letter and numbers in [ ] indicate the likely markings on the capacitors.

[ ] C1 DELETED
[ ] C2 10 nF [10n]
[ ] C3 15 pF [15p]
[ ] C4 470 pF [n47]
[ ] C5 68 uF 6.3V TANTALUM
[ ] C6 470 pF [n47]
[ ] C7 1000 pF [1n0]
[ ] C8 470 pF [n47]
[ ] C9 1000 pF [1n0]
[ ] C10 DELETED
[ ] Creg 3 x 3.3 uF TANTALUM



Next solder in all the bypass capacitors. These are indicated in the silkscreen by a small rectangle, but are not otherwise labelled.

[ ] BYPASS 30 x 0.1 uF

Next solder in the sockets. Pin one of each socket is usually indicated, both on the socket and on the circuit board, by a small indentation at one end. On the CPU circuit board pin #1 of

a socket is always either at the top of a socket which is positioned vertically, or at the left hand side of a horizontal socket. To avoid soldering the wrong sockets in, start with the largest ones and do the smallest ones last.

[ ] 2 40 PIN
[ ] 4 28 PIN
[ ] 1 24 PIN
[ ] 11 20 PIN
[ ] 13 16 PIN
[ ] 26 14 PIN

Now you may solder in some miscellaneous parts.
Y1 is positioned below U47 near the bottom right hand corner of the board. Two crystals are indicated in the silkscreen, Y1 is the leftmost. Y2, to its immediate right, is an option for those operating the board at 6MHz.

[ ] Y1 4 Mhz CRYSTAL or 6 MHz CRYSTAL*
[ ] Y2 2 Mhz CRYSTAL*

Solder in the header strip for J12 and J13, J14, and J16.

[ ] DUAL MALE HEADER STRIP

Next solder in the jumpers which you require for the particular version of the monitor Z.MON which you are using. For the proper configuration, see the beginning of the section entitled "CPU Jumpers".

[ ] JUMPERS

If you plan to use voltage regulators (optional), you may now solder them onto the board.

[ ] P 78L12*
[ ] N 79L12*
[ ] 78H05*

The final step is to insert the IC's into their sockets. If you are using voltage regulators on the board, do not insert the IC's into the board until after you have tested their outputs for the correct voltages, when the card is inserted into the motherboard, and the power supply attached.

Pin #1 of the sockets on the CPU is always at the top of a socket which is positioned vertically, or at the left hand sode of a horizontal socket. On the IC, pin #1 is marked by a dot or indentation at one end. Be very careful not to put the IC's in backwards as this immediately ruins most ICs when the power is applied.

Also check when inserting the IC's to make sure that all the pins go into the socket, and do not bend underneath. This is a very common and needless fault on boards returned to us for servicing.

[ ]	U1	1488*
[ ]	U2	1489*
[ ]	U3	74LS74
[ ]	U4	74LS123
[ ]	U5	74S00
[ ]	U6	74S74
[ ]	U7	74S08
[ ]	U8	74LS32
[ ]	U9	8253*
[ ]	U10	74LS32
[ ]	U11	74LS266
[ ]	U12	74LS139
[ ]	U13	74LS21
[ ]	U14	74LS02
[ ]	U15	74LS132
[ ]	U16	7406/16
[ ]	U17	74LS14
[ ]	U18	74LS374
[ ]	U19	74LS30
[ ]	U20	74LS244
[ ]	U21	74LS244
[ ]	U22	74LS241
[ ]	U23	74LS00
[ ]	U24	74LS266
[ ]	U25	74LS85
[ ]	U26	74LS74
[ ]	U27	74LS08
[ ]	U28	74LS244
[ ]	U29	Z-80
[ ]	U30	74LS373
[ ]	U31	74LS244
[ ]	U32	74LS244
[ ]	U33	74LS244
[ ]	U34	74LS240
[ ]	U35	8255*
[ ]	U36	6116#
[ ]	U37	Z.MON.VER 1.0 #
[ ]	U38	6116#*
[ ]	U39	6116#* or Z.MON. VER 2.0* or 3.0*
[ ]	U40	74LS04
[ ]	U41	74LS04
[ ]	U42	74S74*
[ ]	U43	74LS08
[ ]	U44	74S00*
[ ]	U45	74LS244
[ ]	U46	74LS04
[ ]	U47	74LS04
[ ]	U48	4164*
[ ]	U49	4164*
[ ]	U50	4164*
[ ]	U51	74S257/258/157/158*
[ ]	U52	74S257/258/157/158*
[ ]	U53	4164*

[ ]	U54	4164*
[ ]	U55	4164*
[ ]	U56	4164*
[ ]	U57	4164*

* OPTIONAL # Kit without 64K of on-board RAM only.
Give the board a careful visual inspection before powering it up.

MOTHERBOARD CONSTRUCTION

First solder in all the resistors. Be sure that the two SIPs are soldered in in the proper orientation.

[ ]	5.6 kOHM 1/4W	R1
[ ]	47 OHM 1/4W	R2, R3
[ ]	10 kOHM 1/4W	R4
[ ]	330 OHM 1/4W	R5
[ ]	2.0 kOHM 1/4W	R6, R7
[ ]	5.1 kOHM 1/4W	R8
[ ]	2.0 kOHM 1/4W	R9, R10
[ ]	10 kOHM 1/4W	R11
[ ]	150 kOHM 1/4W	R12
[ ]	300 kOHM 1/4W	R13
[ ]	2.0 kOHM 1/4W	R14, R15
[ ]	100 kOHM 1/4W	R16
[ ]	82 kOHM 1/4W	R17
[ ]	51 kOHM 1/4W	R18
[ ]	20 kOHM 1/4W	R19
[ ]	51 kOHM 1/4W	R20
[ ]	82 kOHM 1/4W	R21
[ ]	100 kOHM 1/4W	R22
[ ]	3.0 kOHM 1/4W	R23
[ ]	24 kOHM 1/4W	R24
[ ]	10 MOHM 1/4W	R25
[ ]	150 kOHM 1/4W	R26
[ ]	15 kOHM 1/4W	R27
[ ]	100 kOHM TRIMPOT	R28
[ ]	30 kOHM 1/4W	R29
[ ]	15 kOHM 1/4W	R30
[ ]	3.0 kOHM 1/4W	R31
[ ]	30 kOHM 1/4W	R32
[ ]	100 kOHM 1/4W	R33
[ ]	2 kOHM 8 PIN SIP	R34, R35
[ ]	1 OHM 1/4W	R36
[ ]	4.7 kOHM 1/4W	R37
[ ]	1.2 kOHM 1/4W	R38
[ ]	18 kOHM 1/4W	R39
[ ]	2.0 kOHM 1/4W	R40
[ ]	5.6 MOHM 1/4W	R41
[ ]	5.1 kOHM 1/4W	R42, R43, R44
[ ]	4.7 kOHM TRIMPOT	R45
[ ]	560 kOHM 1/4W	R46
[ ]	620 kOHM 1/4W	R47

Next solder in the capacitors. Be careful to observe the proper polarity of the tantalums and electrolytics. The two 10 uF tantalums go respectively near the top of U9, and S3 (PGM & RD) where an otherwise unlabelled oval has a + sign at one end to indicate the polarity.

[ ]	1000 pF		C1
[ ]	3300 pF	[3n3]	C2
[ ]	.01 uF	[10n]	C3
[ ]	0.1 uF	[104m]	C4
[ ]	10 nf	[10n]	C5
[ ]	3300 pF	[3n3]	C6
[ ]	150 uF 10V	" "	C7
[ ]	47 uF 35V	ELECTROLYTIC (RADIAL)	C8
[ ]	68 pf		C9, C10
[ ]	390 pF		C11
[ ]	2 x 10 uF 10V	TANTALUM	

Next solder in all the bypass capacitors. These are indicated in the silkscreen by a small oval, but are not otherwise labelled.

[ ] BYPASS 28 x 0.1 uF

Next solder in the sockets. Pin one of each socket is usually indicated, both on the socket and on the circuit board silkscreen, by a small indentation at one end. On the motherboard, Pin #1 of the sockets on the main part of the motherboard (towards the front) all have pin #1 at the far left corner. Those sockets towards the back on the right side have pin 1 in the opposite position, the near right hand corner, as is also indicated on the silkscreen. (Pin #1 of U31 is at the near left hand corner since it is positioned horizontally).

To avoid soldering the wrong sockets in, start with the largest ones and do the smallest ones last.

[ ] 1	40 PIN SOCKET
[ ] 2	28 PIN SOCKET
[ ] 8	20 PIN SOCKET
[ ] 9	16 PIN SOCKET
[ ] 22	14 PIN SOCKET

Now you may solder in some miscellaneous parts. Y1 and L1 are optional, Y1 for RS 232 from the motherboard, L1 for the on-board programming voltage supply.

[ ]	2.4576 MHz XTAL	Y1*
[ ]	500 uH COIL	L1*

Observe the indications on the silkscreen for the following parts. On CR1, the arrow in the silkscreen points to the end on the part which is indicated by a black band. The silkscreen indicates the outline of Q1, Q2, Q3, and D1, when they are inserted properly.

[ ]	1N4001			CR1
[ ]	2N4401	or	PN2222A	Q1, Q3
[ ]	2N4403	or	2N4402	Q2
[ ]	JUMBO RED LED			D1

Solder in the header strip for J1, J2, and J3.

[ ] DUAL MALE HEADER STRIP J1, J2, J3

Next solder in the keys for the two keypads at the front of the motherboard. The keycaps should be labelled in the following pattern:

C	D	E	F	MV	OFT	LD	SV
8	9	A	B	EIO	CMP	PGM	BT
4	5	6	7	EP	ER	BP	DN
0	1	2	3	EM	GO	SI	ESC

(top front of motherboard)

The normally open contacts of the reset switch S1 connect between the center and the right hand contacts indicated on the board. If you are using a two contact pushbutton switch here, solder it into these two holes. Solder in:

[ ]	SPDT MOMENTARY OR PUSH BUTTON SWITCH	S1
[ ]	4PDT SLIDE SWITCH	S2
[ ]	SPDT TOGGLE SWITCH	S3
[ ]	CASSETTE CORDS OR JACKS	J4, J5
[ ]	100 pin S100 CONNECTOR	

If you plan to use voltage regulators (optional), you may now solder them onto the board.

[ ]	7805	VR1*
[ ]	7812	VR3*
[ ]	7905	VR2*
[ ]	7912	VR4*

The final step is to insert the IC's into their sockets. If you are using voltage regulators on the board, do not insert the IC's into the board until after you have tested their outputs for the correct voltages, when the card is inserted into the motherboard, and the power supply attached.

Pin #1 of the sockets on the main part of the motherboard (towards the front) all have pin #1 at the far left corner. Those sockets towards the back on the right side have pin 1 in the opposite position, the near right hand corner, as is also

indicated on the silkscreen. (Pin #1 of U31 is at the near left hand corner since it is positioned horizontally). On the IC, pin #1 is marked by a dot or indentation at one end. Be very careful not to put the IC's in backwards as this immediately ruins most ICs when the power is applied.

Also check when inserting the IC's to make sure that all the pins go into the socket, and do not bend underneath. This is a very common and needless fault on boards returned to us for servicing. Align the TIL313 in their sockets so that the decimal point is closest to the front of the board, and so that they sit right above the socket into which they are inserted. (The pair of pins at each end of the socket are unused).

[ ]	74LS20	U1
[ ]	74LS138	U2, U3
[ ]	74LS04	U4
[ ]	7406	U5
[ ]	ULN2003	U6
[ ]	74LS174	U7
[ ]	74LS374	U8
[ ]	74LS367	U9
[ ]	74LS123	U10
[ ]	74LS132	U11
[ ]	TIL313	U12, U13, U14, U15, U16, U17
[ ]	74LS174	U18
[ ]	empty	U19
[ ]	CD4015	U20
[ ]	74LS191	U21
[ ]	74LS74	U22, U23
[ ]	74128	U24, U25
[ ]	74LS374	U26, U27
[ ]	74LS244	U28
[ ]	74LS240	U29
[ ]	LM3900	U30
[ ]	TL497A	U31*
[ ]	74LS02	U32
[ ]	CD4702	U33*
[ ]	74LS04	U34
[ ]	74LS20	U35
[ ]	74LS02	U36
[ ]	8255	U37
[ ]	74LS244	U38
[ ]	8251	U39*
[ ]	74LS244	U40
[ ]	1488	U41*
[ ]	1489	U42*

Give the board a careful visual inspection before powering it up.

SECTION 2: MONITOR COMMANDS

INTRODUCTION:

The Monitor program provides the user complete control over this computer system, through the use of the Monitor Commands described in this section. The Motherboard contains the following hardware features, which are controlled with and by the Monitor program:

- A 4 hex digit address display.
- A 2 hex digit data display.
- 16 hex numeric keys (0 to F)
- 16 command function keys (14 of which are currently assigned)
- A hardware Reset switch.
- A 24-pin EPROM programming socket and 2 control switches.
- A cassette tape recorder interface.

Applying power to the system or pressing the Reset switch will reset all of the hardware in the system and initialize the Monitor. The system will be in "Command Mode", indicated by an underbar in the extreme leftmost digit of the display. Most commands automatically return the system to this Command mode after their operation is completed. To return to Command mode from other function modes, the following options are available:

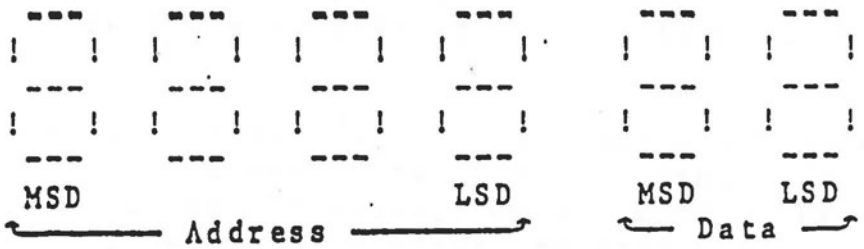
- The [ESC] key may be pressed one or more times. This saves all current mode parameters and Z80A register contents.
- The Reset switch, S1, may be used if [ESC] is not effective, due to a user program error or an error in the use of a device. This re-initializes the entire system and loses all saved information except memory contents.

This is discussed in detail on page 10, under "Recovery From Entry Errors".

The set of Monitor commands forms two groups of 8 independent functions. All Group 1 functions are entered directly by pressing a single command key. All Group 2 functions are entered by entering a 4 hex digit starting address, followed by a single command key. Some of the functions execute immediately while others may request that further information be keyed in before executing.

Some commands take a noticeable time to execute. The display will remain dark until their execution is completed.

DISPLAY LAYOUT:



LIST OF MONITOR COMMANDS:

Group:	Key:	Function: Described on page ( ):
1	EM	Return to examine-memory mode at location previously examined (2.2)
1	GO	Continue executing user program at address last stopped (2.8)
1	ER	Return to examine-register mode, examine register contents as saved at last stop of user program (2.8)
1	SI	Execute next user program instruction and stop in examine-register mode (2.11)
1	BP	Enter Breakpoint setting mode (2.12)
1	EIO	Enter I/O port mode (2.14)
1	LD	Load memory block from cassette tape (2.16)
1	PGM	Read into memory from EPROM programming socket (2.19)
2	EM	Enter examine-memory mode at specified address (2.3)
2	MV	Move memory block starting at specified address (2.5).
2	CMP	Compare memory blocks starting at specified address (2.7)
2	GO	Start execution of a user program from specified address (2.8)
2	SI	Execute one user program instruction at specified address and stop in examine-register mode (2.11)
2	SV	Save memory block, starting at specified address, on cassette tape (2.16)
2	PGM	Program an EPROM with contents of memory block starting at specified address (2.19)
2	OFT	Calculate relative branch offset from specified address (2.17)

RECOVERY FROM ENTRY ERRORS [ESC]:

The [ESC] key is used to recover from errors made in entering data or commands to the system. Some commands have multiple "levels" of entry. That is, several complete and distinct entries of address or data must be made before the command will be executed. At any level of entry, the [ESC] key will:

- After a partial entry, clear the entry and go to the start of the present entry level. It will not be necessary to re-enter the preceding levels of information.
- At the start of a level, go back to the previous level. If pressed repeatedly, it will eventually go back to command mode and the original command will not be executed.

Pressing a group 2 command key, without preceeding it with an address, will cause a triple bar to appear in the address MSD display. This can be cleared by [ESC]. If invalid keys are pressed during an operation, they will be ignored.

Throughout this manual, examples will be given in the following format:

Keystrokes: Display: Comments:

In the "Keystrokes" column, one logical group of keystrokes will be shown on each line, such 0400 [EM], an address and a command. In the "Display" column, the display contents will be represented as well as possible with print. Additional information will be given where necessary in the "Comments" column.

Obviously, an entry such as 0400 [EM] is comprised of the separate keystrokes [0]; [4]; [0]; [0]; [EM].

EXAMINE AND MODIFY MEMORY [EM]:

The contents of any memory location within the Z80A's addressing capability (0000 to FFFF) may be examined and/or modified using the [EM] key. The basic keystroke sequence is as follows:

- Enter the memory address (4 hex digits).
  - Any further digits will be ignored.
- Press [EM] to enter examine-memory mode.
  - The data display will show the 2 hex digits corresponding to the 8-bit pattern stored in the addressed location.
- Press [EM] again (as many times as desired) to step forward through memory and examine the next location.
- Press [EP] (as many times as desired) to step backward through memory and examine the previous location.
- Enter 2 hex digits to store them in the location at the displayed address.
  - After the second digit is entered, the data will be stored in the addressed location and then read back and displayed. Any attempt to write into an EPROM or other read-only device will simply cause the original memory contents to be read back and displayed, unchanged.
  - Every pair of digits entered will be stored in the location at the displayed address. If you enter an incorrect digit pair, simply re-enter the correct pair.
  - Entering an odd number of digits will cause all of the command keys except [ESC] to be ignored until another digit is pressed, or [ESC] is pressed to delete the last digit.
- In Command mode, pressing [EM] without preceding it with an address will cause the most recently examined location to be displayed again. The current address is always remembered by the Monitor. (This is the Group 1 command).
- Pressing [ESC] returns the system to command mode.

Example of Examine and Modify Memory function:

Suppose part of the memory contents are as follows:

```
addr:   0400  0401  0402  0403
data:    32    17    3F    FF
```

Keystrokes:	Display:	Comments:
	_	< underbar
0400 [EM]	0400 32	< contents of location 0400 are read and displayed
[EM]	0401 17	< step forward
[EM]	0402 3F	
[EP]	0401 3F	< step backward
64	0401 64	< 2 digits entered to be stored in location 0401
[ESC]	_	< underbar indicates that system has returned to Command mode
[EM]	0401 64	< examine-memory mode is re-entered. The Monitor will remember the address
4	0401 44	< suppose you want 5A instead
[ESC]	0401 64	< incorrect digit deleted
5A	0401 5A	
[ESC]	_	< back to Command mode

MOVE MEMORY BLOCK [MV]:

Any block of data, of any size, may be copied into any other block of the same size by the Move Data Block command. It is important to note that the data is COPIED into the new block and the contents of the old block are not changed (unless the new block overlaps the old one).

The keystroke sequence is as follows:

- Enter the starting-address of the old, or source block (4 hex digits).
- Press [MV].
  - Address display will become blank and an underbar will appear in the data MSD display.
- Enter the ending-address of the old, or source block (4 hex digits).
- Press [GO].
  - Address display will become blank and the underbar will shift to the data LSD display.
- Enter the starting-address of the new, or destination block (4 hex digits).
- Press [GO].
  - The data in the source block will be written, in order into the destination block, starting at the specified address.
  - The examine-memory mode will be entered automatically with the contents of the first byte of the destination block being displayed. Pressing [EM] or [EP] will allow the destination block to be examined and/or modified.

Memory contents can be moved by any amount in either direction (up or down). You may move part of a program up a few locations to make room for mew instructions, or delete some instructions by moving part of the program down to overlap them.

Example: move 2K of the Monitor program into RAM:

Keystrokes:	Display:	Comments:
	_	< start in Command mode
F000	F000	< enter starting-address of source block
[MV]	_	< underbar in data MSD
F7FF	F7FF _	< enter ending-address of source block
[GO]	_	< underbar in data LSD
0400	0400 _	< enter starting-address of destination block
[GO]	0400 21	< examine-memory mode is entered automatically
[EM]	0401 b0	< step forward, examine
[ESC]	_	< back to Command mode

Example: A block of 120 (hex) bytes, starting in location 0436, is to be moved up 3 locations. The first few bytes are as follows:

```
addr:  0436  0437  0438
data:   2b    A5    7C
```

Keystrokes:	Display:	Comments:
0436	0436	
[MV]	_	
0555	0555 _	< 0436 + 11F = 0555
[GO]	_	
0439	0439 _	< 0436 + 3 = 0439
[GO]	0439 2b	< examine-memory mode is entered
[EM]	043A A5	< step forward
[EM]	043b 7C	
[ESC]	_	< back to command mode

Note that the bytes in the new block are in the same order as they were in, in the old block. The new memory map is:

```
addr:  0436  0437  0438  0439  043A  043b
data:   2b    A5    7C    2b    A5    7C   etc.
```

Note also that the contents of locations 0436, 0437, and 0438 are unchanged. They are never written into in this operation.
Note also that to move 120 (hex) bytes, we specify the ending-address of the source block as 11F (120 - 1 = 11F) bytes above the starting-address, since the starting-address itself contains one of the bytes to be moved.

Example: Previous move operation with entry errors:

Keystrokes:	Display:	Comments:
0436	0436	
[MV]	_	
058	058 _	< entered wrong digit (8)
[ESC]	_	< [ESC] clears entry and goes back to start of level
0555	0555 _	
[GO]	_	
0439	0439 _	< user decides to move 122 bytes instead of 120
[ESC]	_	< [ESC] clears entry and goes back to start of level
[ESC]	_	< [ESC] goes back to start of previous level (awaiting ending-address of source block)
0557	0557 _	< 0436 + 121 = 0557
[GO]	_	
0439	0439 _	
[GO]	0439 2b	< operation is complete. examine-memory mode is entered at location 0439

The result is that 122 bytes, from locations 0436 to 0557 inclusive, have been copied into locations 0439 to 055A.

COMPARE MEMORY BLOCKS [CMP]:

Any block of data in memory, of any size, may be compared with any other block of the same size by the Compare Memory Blocks command. The comparison is byte-for-byte, and the system will automatically indicate where any discrepancies occur. The keystroke sequence is as follows:

- Enter the starting-address of the first block.
- Press [CMP].
- Enter the ending-address of the first block.
- Press [GO].
- Enter the starting-address of the second block.
- Press [GO].
- If all corresponding bytes of both blocks match, two underbars will appear in the two LSDs of the address portion of the display.
    - Return to Command mode by pressing [ESC].
- If the contents of any location in the first block do not match the contents of the corresponding location in the second block, the comparison will stop and the display will show the address and data of this location in the SECOND block.
    - Pressing [EP] displays the address and data of the corresponding location in the FIRST block.
    - Pressing [EM] displays the address and data of the location in the second block.
    - Note: Examine-memory mode is NOT entered. It is not possible to modify memory locations in the middle of a comparison operation.
- Press [GO] to continue comparison.
    - If a further mismatch exists, the comparison stops again as above.
    - If all locations match to the end of the block, display will show two underbars in the two LSDs of the address portion of the display.

Example: Compare locations F000 to F300 with the block starting at address 0400.

Keystrokes:	Display:	Comments:
	_	< start from Command mode
F000	F000	
[CMP]		< underbar shown in data MSD
F300	F300 _	
[GO]		< underbar shown in data LSD
0400	0400 _	
[GO]	045A 16	< assume a mismatch here
[EP]	F05A 06	
[GO]	05E2 37	< assume a mismatch here
[EP]	F1E2 27	
[EM]	05E2 37	
[GO]	--	< underbars indicate rest of bytes are correct to end of block
[ESC]	_	< back to Command mode

EXECUTE PROGRAM [GO], EXAMINE REGISTERS [ER], CONTINUE:

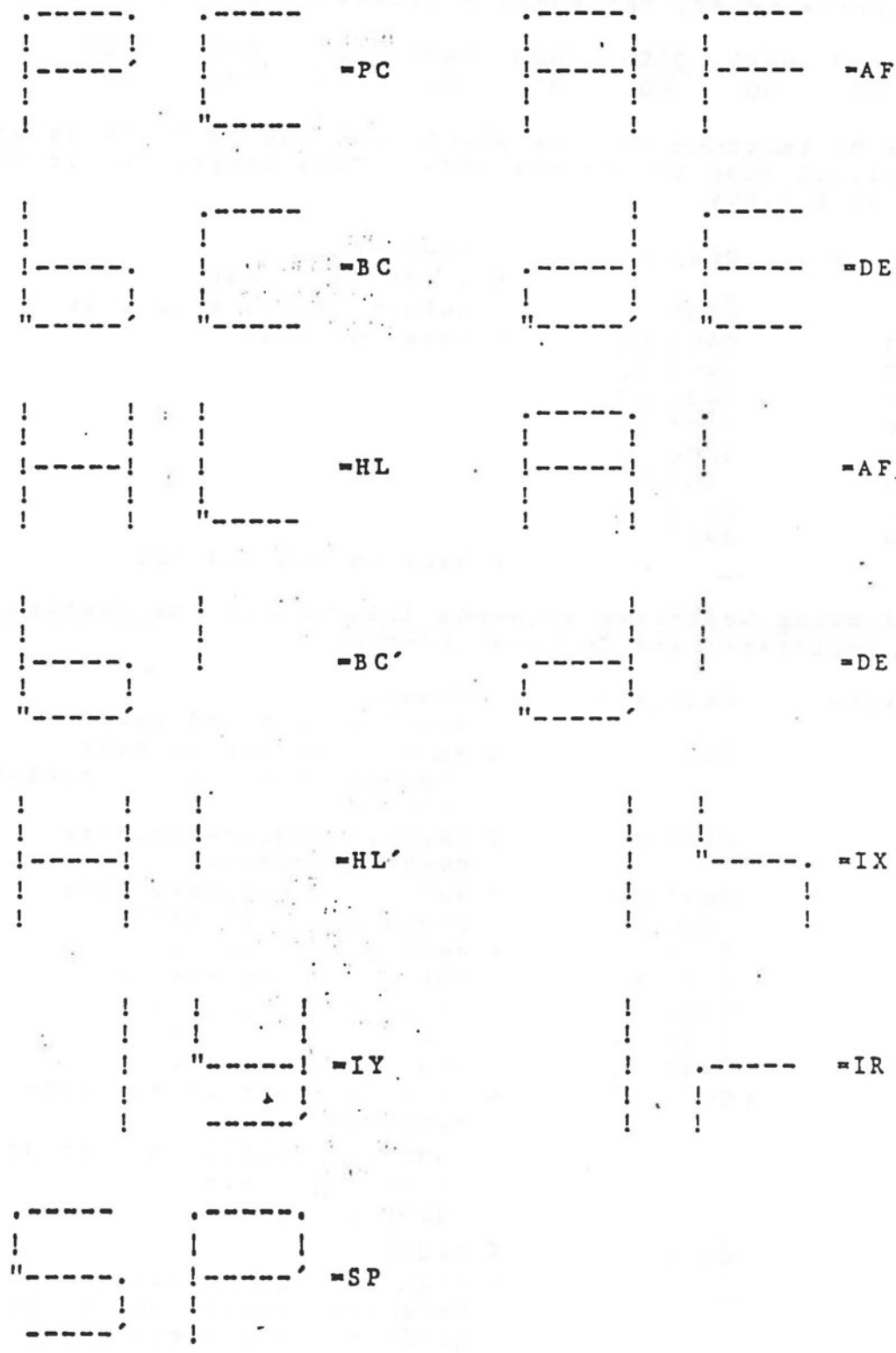
This group of commands allows the user to directly control the execution of machine-language programs. The keystroke sequences are as follows:

- Enter starting-address of program.
- Press [GO].
  - Display will turn off and remain dark while user program executes (unless the program makes use of the display).
- Press [ESC] to interrupt execution (ie: Pause).
- The examine-register mode will be entered and the display will show the contents of the Program Counter (the address of the next instruction to be fetched) in the address portion of the display and "PC" in the data portion.
  - Pressing a digit key will select a register pair as follows:

0	1	2	3	4	5	6	7	8	9	A	B
AF	BC	DE	HL	AF'	BC'	DE'	HL'	IX	IY	IR	SP

    The contents of the register pair will appear in the address portion of the display and the register pair identifier (see next page) will appear in the data portion.
  - This list of registers can be scanned forward or backward with the [EM] and [EP] keys.
- Enter a digit to modify the MSD of the first register of the displayed pair.
  - Pressing [ESC] will clear this digit and display the original contents of the register pair.
- Enter a second digit to store the two digits in the first register and display the updated registers.
  - Pressing [ESC] will skip over the second register and display the register pair with the first register updated.
- Enter a third digit to modify the MSD of the second register.
  - Pressing [ESC] will delete this digit and display the register pair again.
- Enter a fourth digit to store the two most-recently-entered digits into the second register and display the updated registers.
- Pressing [GO] after a register pair has just been displayed will skip over the first register to allow modification of just the second register.
- Pressing [ESC] after a register pair has just been displayed will display PC again and return to the start of examine-register mode.
- Pressing [GO] at the start of examine-register mode will continue program execution with updated registers.
- Pressing [ESC] when PC is displayed will return to Command mode without affecting register contents.
  - In Command mode the user may examine memory or do other operations.
  - Press [ER] to return to examine-register mode.
- Press [GO] in Command mode to continue program execution from last saved address (contents of PC).

* Register pair identifiers: Limitations of the 7-segment displays prevent perfect representations of the letters used to identify register pairs. The approximations used are as follows:



Example:- How to write and execute a simple machine-language program.

Machine-language programs may be written directly into memory using the Examine and Modify Memory command. The following memory map shows a simple looping program:

```
addr:  0400  0401  0402  0403  0404  0405  0406  0407
data:    00    00    00    00    00    C3    00    04
```

The five 00 instructions are NO-OP, and the C3 00 04 is an unconditional jump to address 0400. This program would be entered as follows:

Keystrokes:	Display:	Comments:
	_	< start in Command mode
0400	0400	< select starting-address
[EM] 00	0400 00	< enter program
[EM] 00	0401 00	
[EM] 00	0402 00	
[EM] 00	0403 00	
[EM] 00	0404 00	
[EM] C3	0405 C3	
[EM] 00	0406 00	
[EM] 04	0407 04	
[ESC]	_	< back to Command mode

The following keystroke sequence illustrates the Execute, Examine Registers, and Continue commands:

Keystrokes:	Display:	Comments:
	_	< start in Command mode
0400	0400	< enter starting address
[GO]		< display dark during program execution
[ESC]	0403 PC	< pause: examine-register mode is entered
3	0400 HL	< select HL register pair
[ER]	0144 A'	< advance to AF' pair
81	8144 A'	< modify register A'
[ESC]	8144 A'	< skip over register F'
[EP]	0400 HL	< go back to HL pair
[GO]	0400 HL	< skip over register H
3C	043C HL	< modify register L
[ESC]	0403 PC	< back to start of examine-register mode
[GO]		< continue execution with updated registers (display dark)
[ESC]	0402 PC	< pause
[ESC]	_	< back to Command mode User may examine memory or perform other operations
[GO]		< continue execution (display dark)
[ESC]	0404 PC	< pause
[ESC]	_	< back to Command mode
[ER]	0404 PC	< enter examine-register mode user may modify registers
[GO]		< continue execution

SINGLE-INSTRUCTION EXECUTION [SI]:

The Single-Instruction command allows the user to execute a program one instruction at a time for debugging purposes. The machine will fetch the next instruction from memory and execute it entirely, regardless of whether it is a single- or multi-byte instruction.

The Single-Instruction command may be used to execute one instruction either at the beginning of a program (ie: starting from Command mode) or at any program break. This includes a break caused by the use of [ESC] (page 16) or a break caused by the use of a Breakpoint (page 20).

The basic keystroke sequence for the use of the Single-Instruction command is as follows:

- Enter the starting-address of a program.
- Press [SI].
  - The first instruction of the program will be executed and the examine-register mode will be entered with PC displayed.
- Press [SI] at the start of examine-register mode or from Command mode to execute the next instruction and return to examine-register mode.

Example: Single-step through the program on page 18:

Keystrokes:	Display:	Comments:
	_	< start in Command mode, having entered program
[SI]	0401 PC	< first instruction (00 in location 0400 is executed and examine-register mode is entered
[SI]	0402 PC	< execute next instruction
[SI]	0403 PC	< execute next instruction
2	0400 DE	< select DE register pair user may examine and modify registers if necessary
[ESC]	0403 PC	< return to start of examine-register mode
[SI]	0404 PC	< execute next instruction

BREAKPOINTS [BP]:

The breakpoint facility of the Monitor enables the user to specify up to four points (breakpoints) within a program, at which execution will be halted automatically. Breakpoints are NOT Z80A instructions, but features of the Monitor program and as such, they do not require extra space to be included in a program.

A breakpoint is inserted into a program by specifying the address of an instruction already in the program. When the processor attempts to fetch that instruction, execution will be halted and examine-register mode will be entered. When execution is continued, the instruction which had been "masked" by the breakpoint will be the first to be executed.

Associated with each of the four available breakpoints is an optional delay number. This is a 4 hex digit number specifying how many times that breakpoint must be encountered by the processor, before the break will actually occur. These "delayed" breakpoints may be mixed freely with normal breakpoints within a program.

The keystroke sequence for setting breakpoints is as follows:

- Press [BP].
  - Display shows an underbar in the data MSD and a zero in the data LSD. (Breakpoint 0 has been examined).
- Enter breakpoint address. Note: this address must be that of the FIRST byte of an instruction, and that instruction will be the first to be executed after execution is continued following the break.
- Press [GO] to save this address as breakpoint 0.
- Press [BP] to examine breakpoint 1.
  - Pressing [BP] repeatedly will examine breakpoints 2 and 3, then return to 0.
- Press [EP] to examine the previous breakpoint.
- Press [GO] (without preceding it with an address) to clear the displayed breakpoint and the associated delay number.
- Press [ER] after entering a breakpoint address to examine the delay number associated with this breakpoint.
  - Display shows zeros in the address portion and the symbol

in the data MSD portion, with the breakpoint number in the data LSD. The delay count symbol will be represented here by |-

- Enter a 4 hex digit delay number.
- Press [GO] to save this delay number.
  - Display shows breakpoint address again.
- Press [ESC] to return to Command mode.

...continued next page

- Execute the program in which the breakpoints have been inserted.
- When a breakpoint address is reached the program will be interrupted and examine-register mode will be entered. The user may examine or modify registers, or use SI to step through the program.
  - Press [GO] to continue execution.
- If a non-zero delay number is set for a breakpoint, that breakpoint's address must be encountered the specified number of times before execution will be interrupted.

Notes on Breakpoints:

- Breakpoints may only be set in RAM.
- Every time a delayed breakpoint causes a break, its delay number is re-initialized. The delay number for a given breakpoint will not be affected by breaks caused by other breakpoints.

Example: Using breakpoints with the program on page 18:

Keystrokes:	Display:	Comments:
	_	< start in Command mode, having entered the program
[BP]	_0	< examine breakpoint 0
0402	0402 _0	< enter an address for b.p. 0
[GO]	0402 _0	< store the address
[BP]	_1	< select next breakpoint
0404	0404 _1	< enter an address for b.p. 1
[GO]	0404 _1	< store the address
[ER]	0000|-1	< examine the delay number associated with b.p. 1
0008	0008|-1	< enter a new delay number
[GO]	0404 _1	< store the delay number display shows b.p. address
[ESC]	_	< back to command mode
0400 [GO]	0402 PC	< break in 0402 (b.p. 0)
[GO]	0402 PC	< break in 0402 (b.p. 0)
[GO]	0402 PC	< break in 0402 (b.p. 0)
[GO]	0402 PC	< break in 0402 (b.p. 0)
[GO]	0402 PC	< break in 0402 (b.p. 0)
[GO]	0402 PC	< break in 0402 (b.p. 0)
[GO]	0402 PC	< break in 0402 (b.p. 0)
[GO]	0402 PC	< break in 0402 (b.p. 0)
[GO]	0404 PC	< break in 0404 (b.p. 1)

If you study this example very carefully, you will see that breakpoint 1 did not cause a break until it had been encountered 8 times. Since execution started at address 0400, breakpoint 1 was not encountered after the first pressing of [GO]. Thereafter, both breakpoints were encountered 7 times, and on the final pressing of [GO], breakpoint 1 caused a break, having been encountered 8 times. The pattern of breaks would repeat itself if [GO] were pressed a further nine times.

EXAMINE AND LOAD I/O PORTS [EIO]:

Any of the Z80A's 256 I/O ports may be examined and loaded using this command, which is a little bit like the Examine and Modify Memory command described on page 11. The keystroke sequence for this command is as follows:

- Press [EIO] to enter Examine I/O Ports mode.
  - Display will show underbars in the two MSD positions of the address section.
- Enter a two hex digit I/O port address.
  - Display will show an underbar in the MSD position of the data section.
- Press [ER] to examine (read) the addressed I/O port.
- Enter two hex digits to load the I/O port.
  - The I/O port will be loaded every time the second of two digits is entered.
- Press [EP] to decrement the I/O port address.
  - Display will show an underbar in the MSD data position.
  - A read operation does not occur automatically to prevent device status register side-effects if reading a data register.
  - Press [ER] to read the addressed I/O port (if desired).
- Press [EIO] to increment the I/O port address.
  - If an illegal key (such as [EM]) is pressed, it will be ignored.
- Press [ESC] to return to the start of examine I/O ports mode, to allow entering another I/O port address.
- Press [ESC] when the two underbars are displayed to return to Command mode.

Example of Examine I/O Ports command:

Keystrokes:	Display:	Comments:
	_	< start in Command mode
[EIO]	__	< enter examine ports mode
F5	__F5 _	< select a port (control register of the 8251)
[ER]	__F5 00	< read the port
4E	__F5 4E	< load the port
[ER]	__F5 07	< read the port again contents have changed due to external actions
[EP]	__F4 _	< decrement the port address
[EIO]	__F5 _	< increment the port address
[ESC]	__	< back to start of mode
36	__36 _	< select another port
[ESC]	__	< back to start of mode (assume you did not wish to read port 36)
[ESC]	_	< back to Command mode

CASSETTE TAPE CALIBRATION

The Cassette tape interface of the MULTIFLEX computer is designed for very high speed operation (2000 bits/sec.), and therefore requires careful calibration of both the tape recorder and the interface circuitry, in order that it work reliably. Two short utility routines have been incorporated into the monitor EPROMs to assist in this adjustment. The procedure is as follows:

- Plug the cassette interface OUT cable into the recorder's microphone jack, and the IN cable into the earphone jack.

- Turn on the tape machine and start recording.

- From command mode, enter 1D00 (ZMON VER 1.1) or FD00 (ZMON VER 2.1 or 3.1) (This is 1CA7 in ZMON VER 1.0 only) as a starting address and press [GO].
  A 1 kHz. tone will be recorded onto the tape. If your recorder is equipped with a VU meter and recording level control, adjust it until the meter reads 0 db.
  Continue recording for several minutes.

- Press reset to get back to the command mode.

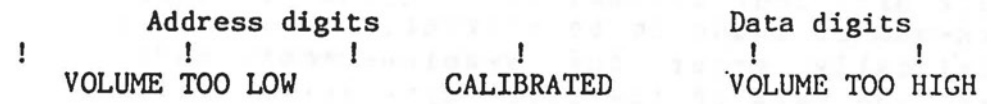
- Stop the tape recorder, rewind, and start playing back the tape.
  If you cannot identify the beginning, remove the computer's IN cable from the machine's earphone jack and listen until you hear a loud, high pitched whine. Then re-insert the cable, and continue playing back.

- Enter 1D20 (ZMON VER 1.1) or FD20 (ZMON VER 2.1 or 3.1) (This is 1CC7 in ZMON VER 1.0 only) and press [GO].
  A vertical bar in the display will indicate the calibration accuracy.

- Set the trimmer potentiometer R28 to midway, the tape recorder volume to about 2/3 of maximum, and the tone control (if your recorder has one) to maximum.

- Adjust R28 to locate the bar in the address LSD of the display.
  Flickering bars may appear in other digits, try to minimize them.

- Adjust the tape recorder volume and tone controls for the most stable display.



- Press reset to return to command mode and stop the tape recorder. Mark the positions of the controls so that you may find them again.

- For best results, recalibrate periodically. R28 should not require a different setting, except for use with a different recorder.

SAVE MEMORY BLOCK [SV]:

A block of data of any size may be saved from memory to cassette tape using this command. The keystroke sequence is as follows:

- From Command mode, enter the starting-address of the block to be saved.
- Press [SV].
- Enter the ending-address of the block.
- Start recording.
- Press [GO].
    - The block of data will be written onto the tape, preceded by an 8-second preamble and a header containing the starting-address in memory and the block size, followed by an check-sum.
    - The display will turn off for the duration of this operation (8 seconds of pre-amble plus four Seconds per K bytes of data).
- The system will automatically return to command mode when finished (signified by an underbar in address MSD position).
- Stop the tape recorder.

LOAD MEMORY BLOCK [LD]:

To put stored data back into memory from the tape, the following simple procedure is done:

- Set up the recorder's tone and volume controls to the positions marked during the calibration procedure (see page 23).
- Find the start of the block pre-amble on the tape by using the tape counter (if your recorder has one), by listening to the tape (the preamble will sound as a loud, high-pitched whine), and/or by watching the LED (D1) on the Motherboard, which will light when pre-amble or data is being received from the tape.
- Press [LD] while the tape is in the pre-amble.
    - Since the Save operation stores the memory starting-address on the tape, the data will load back into the same block in memory as it was stored from.
- After the time required to load the block (remainder of pre-amble plus four seconds per K bytes of data), if the check-sum is found to be correct, the computer will automatically enter the examine-memory mode.
- The address and data of the first byte in the block will be displayed.
- If a check-sum error is detected, due to the tape recorder controls being improperly set or electrical noise on the tape, the display will show three horizontal bars in the data MSD position.
    - It may be necessary to re-calibrate before loading again.
- If a major error occurs (no data on tape, dirty tape, volume control set far too low or high) the read operation will not terminate and must be stopped by the Reset switch.
    - This condition may exist if the read operation takes far more than the required time.

RELATIVE BRANCH OFFSET CALCULATION [OFT]:

This utility function is included to speed up the calculation of offsets for the Z80A's relative branch instructions. It allows the user to quickly calculate both positive and 2's complement negative offset values directly in hexadecimal notation. The procedure is as follows:

- Enter the address of the first byte of the branch instruction.
- Press [OFT].
  - Display will show an underbar in the data MSD position.
- Enter the destination-address (address of the first byte of the instruction to which control is to be transferred).
- Press [GO].
  - Display will show the relative offset in hex. If the offset is out of range, three bars will appear in the data MSD position.
- Press [ESC] to return to Command mode.

Example: Calculate the relative offset required in a JR Z instruction starting in location 05bC, to transfer control to another instruction starting in 05A1:

Keystrokes:	Display:	Comments:
	_	< start in command mode
05bC	05bC	< enter address of first byte of branch instruction
[OFT]		
05A1	05A1 _	< enter address of first byte of destination instruction
[GO]	E3	< relative offset value

Store E3 in location 05bd to complete the JR Z instructio

MOTHERBOARD EPROM PROGRAMMER JUMPERS

Jumper J1 and either of J2 and J3
according to the device being programmed.
Never jumper both of J2 and J3. ALWAYS
change jumpers only WITH S2 = DIS or power off.

```
J1     O-O O                  O O-O
       O-O O                  O O-O
       O-O O                  O O-O
 2764 or 2716     2708, 2716, 2724, 2732,
                      2758, 2532, 2564

J2    O O-O      O-O O      O-O O      X O-O
      O O-O      O-O O      O-O O      O X O
      O O-O      O-O O      O-O O      O O-O
      O O-O      O-O O      O-O O      O O-O
      O O-O      O O-O      O-O O      O O-O
      2708       2532       2564      *TMS2716*

J3    O-O O      O O-O      O O-O      O-O O
      O-O O      O O-O      O O-O      O-O O
      O-O O      O O-O      O O-O      O-O O
      O O-O      O O-O      O-O O      O-O O
      O O-O      O-O O      O-O O      O-O O
      O O-O      O-O O      O-O O      O-O O
  2716,2758   2732,2724    2732A     2764,27128
```

*NOTE: The TMS2516 is equivalent to 2716, BUT
the TMS2716 is not since it, like the 2708,
requires three power supplies. The above
setup (with the two pins indicated by X jumpered
together) will enable one to read this device,
but not to program it.

EPROM PROGRAMMING (PGM):

BEFORE using the on-board programming supply, it must be calibrated. With S2 at DIS, and no jumpers on J1 through J3, do the following:

- Start in command mode
- Type (EIO) (FB) (FF), this will turn on the programmer supply.
- Measure the voltage between U31 pins 6 and 8.
- Adjust R45 until pin 6 is 25 volts positive with respect to pin 8.
- Setup the jumpers for a 2764. The voltage should drop to +21 volts. Tolerance on these voltages should be +/- 1/2 volt.
- Reset the motherboard and CPU via the reset switch S1. Pressing ESC twice would get you back to command mode, BUT would fail to turn off the programming supply.

It is possible to program EPROMs with data stored in memory. The MULTIFLEX Z80 computer can program the following EPROMs: 2708, 2716, 2732, 2532, 2758, 2764, 2564, 2724, 27128. In order to program any EPROM an optional on-board 21 or 25 volt (selected by jumper) power supply can be used (or the user may supply his own), and the following procedure should be obsevered:

-Set S3 at RD and S2 at DIS.
-Set up jumpers (J1-J3) as shown in the table opposite. This is very important since incorrect setup can cause damage to the device being programmed.
-Insert the EPROM.
-Set S2 to EN.
-Follow the procedure below for the particular type of EPROM you are trying to program.

<u>2708</u>:

NOTE: [1] IT IS NOT POSSIBLE TO PARTIALLY PROGRAM A 2708.
[2] VR2 (7905) MUST BE INSTALLED TO PROGRAM OR READ 2708s, OR TO READ A TMS2716. (See note 4 below).

- From command mode, enter the starting address of the data block you wish programmed.
- Press (PGM). (The display will show an under bar in the MSD position.)
- Enter the ending address of the block of memory. In this case this should be starting address + 3FFH. (**)
- Press (GO). (The display will show an underbar in the LSD position).
- Enter the destination address in the EPROM (in this case 0000H). (**)
- Enter the device name, ie. 2708.
- Press (GO). If the 2708 is completely erased the display will completely turn off and the EPROM will be programmed. If it

has some non-FF loactions the dispaly will show the address of the first such location. Press either (GO) to continue programming or (ESC) to return to the command mode.

(**) Even though this information is redundant for a 2708, it must be entered since the Monitor expects it for all programmed devices. It is merely ignored.

Example: Program a 2708 with 1 Kbyte of data in locations 0400H to 07FFH.

KEYSTROKES	DISPLAY	COMMENTS
		Start in command mode
0400	0400	Enter starting address.
(PGM)		
07FF	07FF	Enter ending address.
(GO)		
0000	0000	Enter destination address.
(GO)		
2708	2708	Enter device type.
(GO)		The display will be dark will the EPROM is programming (108 seconds.)
	005C 2d	A mismatch was found during verify.
(EP)	045C 3d	Compare with RAM contents: an error in 1 bit.
(GO)		Proceed.
	—	Rest of locations match.

ALL OTHERS:

This procedure allows the user to program an arbitrarily large block of data (as long as it is less than or equal to the size of the specified EPROM) anywhere in the EPROM to be programmed without disturbing the rest of the contents. There are no programming differences between the above devices, and the basic stucture of the command is the same.

- Set the jumpers J1-J3 with S2 set to DIS.
- Set S2 to EN.
- (If a separate programming supply is to be used, make sure it is on now. Also check in this case that the voltage is correct: +21 volts for 2732A, 2764, or 27128; +25 volts for all other EPROMs).
- Set S3 to PGM.
- Enter starting address of the block to be programmed.
- Press (PGM). The display will show a bar in the MSD data position.
- Enter the location of the ending address of the block. The ending address will also be programmed into the EPROM. NOTE:

For a one byte block: starting address = ending address.

- Press (GO). The display will show a bar in the LSD data position.
- Enter the destinaiton address in the EPROM where the data will start. This must be within the addressing range of the EPROM.
- Press (GO). The bar will remain in the LSD display but the address will be processed.
- Enter the device name.
- Press (GO). If the EPROM is blank (erased), the display will blank and the EPROM will program, otherwise the display will show the first non-FF address. Press (GO) to continue or (ESC) to abort. Programming will take about 1 minute and 45 seconds for every 2K programmed.
- Set S3 to RD. (If you are using a separate power supply for the programming voltage, or on some occasions without it, the LED display will show a message indicating that this is now necessary).
- Correct verification of the data is indicated by two underbars in the two LSD address displays. If a mismatch occurs the address in the EPROM and the incorrect data will be displayed. To continue the verification procedure, press (GO). To examine the correct data for the displayed location, press (EP).
- When verification is completed, press (ESC) to return to the command level.

NOTES:

[1] If an external 21 or 25V supply is used, then S3 = RD will appear on the display before the verification procedure is commenced. Set S3 to RD and press (GO).

[2] 2732A, 2764, and 27128 EPROMs are programmed using 21V. Make sure that the supply is set for 21V when programming these devices since 25V may damage them.

[3] A 2724A can be programmed as the lower half of a 2732, and a 2724B as the upper half. Similarly, a 2758A as the lower half of a 2716, and 2758B as the upper half.

[4] A TMS2516 is equivalent to a 2716, but A TMS2716 CANNOT BE PROGRAMMED, although it may be read.

[5] If using the on-board programming power supply, make sure that the voltage has been calibrated before it is used.

Example #1: Program location 153H in a 2716 from memory location 556H. Only one byte is to be programed.

KEYSTROKES	DISPLAY	COMMENTS
		Start in command mode
0556	0556	Enter starting address
(PGM)		
0556	0556 _	Enter ending address
(GO)	_	

0153	0153 _		Enter destination address
(GO)	_		
2716	2716 _		Enter device type
(GO)	0000	xx	First non-FF location
(GO)			Continue programming
			Verification correct.
(ESC)	_		Return to command mode

Example #2: Program a 2732, with 100H bytes of data located at 0400H.

KEYSTROKES	DISPLAY	COMMENTS
	_	Start in command mode
0400	0400	Enter starting address
(PGM)		
0500	0500 _	Enter ending address
(GO)		
0000	0000 _	Enter destination address
(GO)		
2732	2732 _	Enter device type
(GO)		Display blanks during programming
		All is well
(ESC)	_	Back to command level.

For all devices, certain rudimentary error checking is preformed. These include: specifying a presently unsupported device; specifying a block size larger than the size of the specified EPROM; ending address before starting address; destination address + block size is greater than device size. For all the above error conditions, the machine will display the standard error symbol (≅). Once an error has occured, all of the key sequence preceding the error will be aborted. the only way to recover from an error of this type is to press (ESC) and return to the command level.

READING FROM THE EPROM SOCKET:

To read the contents of an already programmed EPROM into RAM, the following steps must be observed:

-Set S2 to DIS and set up J1-J3 for the kind of EPROM you want to read. Please note that J1-J3 MUST be setup correctly or damage may result to the device being read.

-Insert the EPROM into the programming socket.
-Set S2 to EN and S3 to RD.
-Press (PGM).
-Enter the starting address in memory where you wish the EPROM to be dumped. The computer assumes there is enough RAM at this location to hold the contents of the specified EPROM.
-Press (GO).
-Enter the device name.

-Press (GO).
-The examine-memory mode will be automatically entered with the display showing the address and data of the first address of the block upon complettion of the read.

DOWNLOAD ROUTINE (DN)

The download routine included in the monitor (DN) requires data in both the proper format and at the proper baud rate for the transfer to be successful. The data is received by the serial port on the motherboard, without any hand-shaking, thus this is by no means a true communications routine. If you wish to use a different serial port or to use the DN key for a different use, you may update the monitor to suit your purpose.

Format for the received data:

After the DN key is pressed, the routine looks for an incoming 00H byte from the RS 232 C port. Following this single byte, it expects the following:

- Destination address in memory low byte
- Destination address in memory high byte
- Block size (byte count) low byte
- Block size high byte
- First data byte
- Succeeding data bytes
- Last data byte
- Checksum

The routine will respond to a successful transfer, including correct checksum , with a double underbar appearing in the address LSDs. The checksum is computed by adding all the data bytes in the accumulator, and then taking the two's complement of the result. In other words, the checksum is the least significant 8 bits of the sum of all the data bytes, after it has been inverted and incremented by one.

CP/M BOOT

This routine is used to enter CP/M. It is specifically designed for the configuration we suggest for the MULTIFLEX floppy board in this application. It will handle any CP/M disk, 5 or 8 inch, which we supply. In all versions of the monitor this routine is relocatable, unlike the remainder of the monitor, which must reside within a particular memory block.

In revision 1.1, 2.1, 3.1 of ZMON, this routine is located at C00 (hex) within the EPROM. If you are using the MULTIFLEX Z80B system without the motherboard, you may enter this boot routine directly by jumpering J15 so that the restart address ends in C00 (hex) (Jumper pins 5 and 6 of J15, as well as the appropriate ones among J15 pins 1 to 4). As long as the low address byte for the beginning of this routine remains 00 (hex), it will run in any part of memory, so FOR THIS USE ONLY versions 1, 2, and 3 are interchangeable.

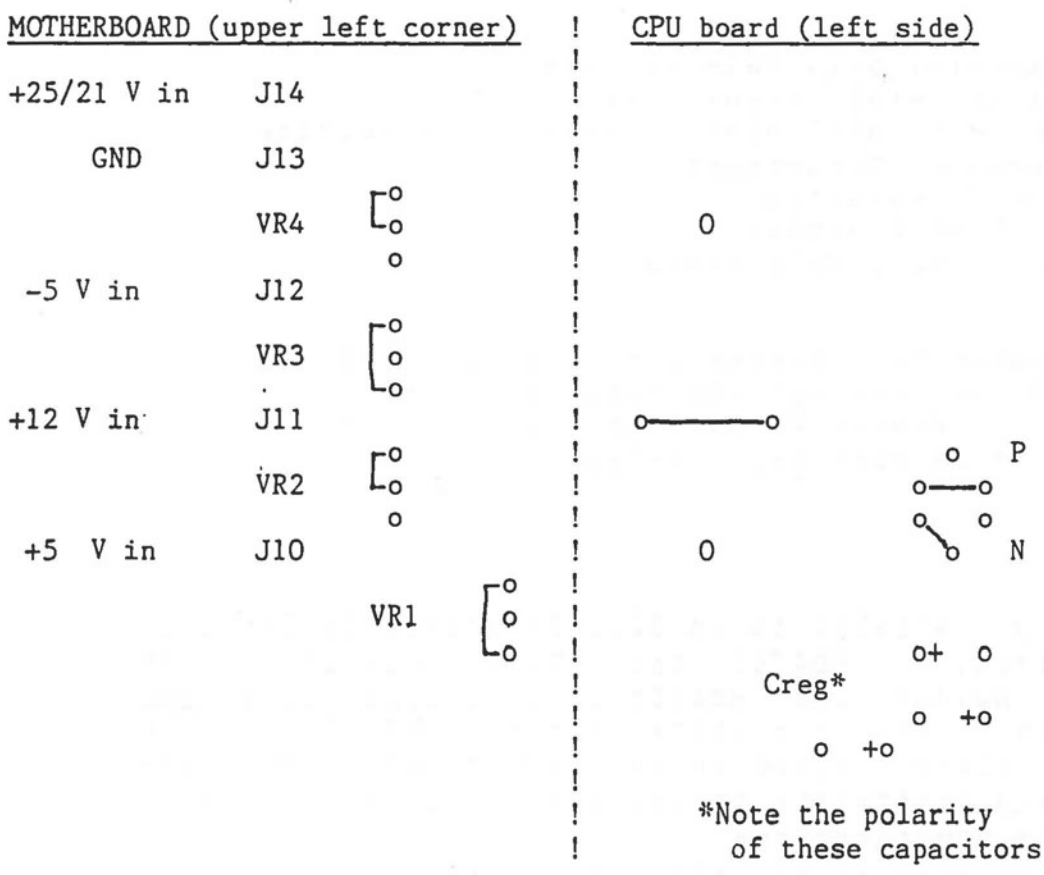
POWER SUPPLY CONNECTIONS:

The MULTIFLEX Z80 computer is designed to operate directly from the standard S-100 power supply voltages (+8, +16, -16 volts) and is equiped with provision for voltage regulators for such supplies. If such a supply is to be used the following regulators, marked as optional on the PARTS LISTS, must be installed:

Motherboard:	VR1	7805	in all cases
	VR2	7905	if 2708 EPROMs are to be used
	VR3	7812	if 2708s or the RS232 interface is to be used
	VR4	7912	if RS232 interface to be used
CPU card:	VR1	78H05	in all cases
	P	78L12	for RS 232 C (on CPU piggyback)
	N	79L12	for RS 232 C (on CPU piggyback)

With the appropriate regulators installed, the S-100 voltages may be fed to the motherboard at the points indicated on the silk-screen.

Alternately, the computer may be powered directly from regulated power supplies by connecting jumpers between the input and output of all the regulators as shown below.



MOTHERBOARD I/O SECTION:

The basic kit includes an 8255 Programmable Peripheral Interface, which provides 24 programmable parallel I/O lines. These are available on J7 on the Motherboard, located to the right of U37 (the 8255 chip). The pin assignment of J7 is as follows:

PC0	13	26	PB0
PC1	12	25	PB1
PC2	11	24	PB2
PC3	10	23	PB3
PC4	9	22	PB4
PC5	8	21	PB5
PC6	7	20	PB6
PC7	6	19	PB7
GND	5	18	GND
PA0	4	17	PA7
PA1	3	16	PA6
PA2	2	15	PA5
PA3	1	14	PA4

No attempt will be made here to explain the programming and use of the 8255, as this is a very complex device and no short description could do justice to its many capabilities. The following reference works are recommended for users need more information:

- Intel Component Data Catalog, 1980
  Published by Intel Corporation.
  Available from Intel distributors or by writing:
  Literature Department
  Intel Corporation
  3065 Bowers Avenue
  Santa Clara, California
  zip 95051

- Microcomputer Interfacing with the 8255 PPI Chip
  by Paul F. Goldsbrough and Peter R. Rony
  Published by Howard W. Sams Co., number 21614.
  Available from Sams book dealers.

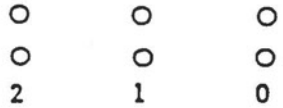
Optionally available is an 8251 Programmable Communication Interface, a CD4702 Baud Rate Generator with crystal, and MC1488 and MC1489 RS-232C line driver and receiver chips. The circuitry for this RS-232C serial interface is already wired on the Motherboard. The components are sold separately to lower the kit cost to users who not require the interface.

Full information on the 8251 PCI chip can be found in the Intel Component Data Catalog for 1980 (see above). No attempt will be made to describe the device here because of space limitations.

The CD4702 provides six baud rate choices, selectable by jumper J6, located to the left of U33 on the Motherboard (the 4702). J8 has three locations for jumpers, numbered 0, 1, and 2, and the configuration of these jumpers selects the baud rate as follows:

0	1	2	baud rate
installed	installed	installed	9600 baud
installed	installed	-	4800 baud
-	installed	installed	2400 baud
installed	-	-	1200 baud
-	installed	-	300 baud
-	-	-	110 baud

J6:



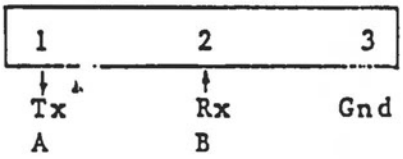
All 8251 I/O signals are available on J9, located to the right of U40 The pin assignments for J9 are as follows:

J9:

1	DSK
2	DTR
3	CTS * If CTS is not being used, it MUST be held low. You may jumper it to SYNDET to do this
4	RTS
5	SYNDET
6	RXRDY
7	RXC
8	RXD
9	TXE
10	TXRDY
11	TXC
12	TXD

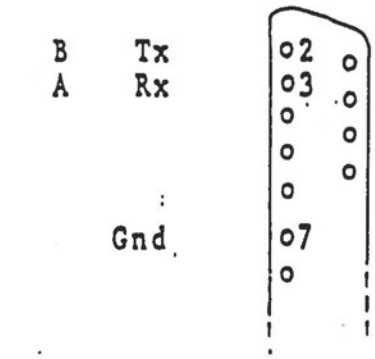
RS-232C signals are available on J8 for connection to a terminal, modem, or printer. J8 is located above U41 on the Motherboard, and has the following pin assignment:

J8:

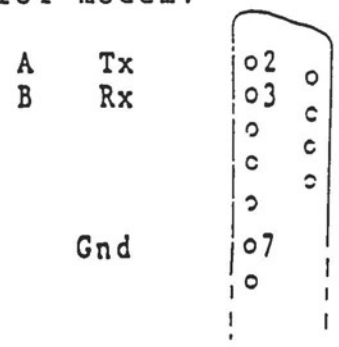


If you are using a standard DB-25 connector, the connections are as shown below:

for terminal or printer:



for modem:



Pin assignment of serial I/O connectors J1,J2
(optional, requires small piggy-back board)

J1 (serial port #0)

1	TxD* 0
2	GND
3	RxD* 0
4	RTS 0
5	GND
6	CTS 0

J2 (serial port #1)

1	TxD* 1
2	GND
3	RxD* 1
4	RTS 1
5	GND
6	CTS 1

Pin assignment of parallel I/O connectors J5,J19

J5 - Programmable Interval Timer Connector
(to/from 8253 Programmable Interval Timer)

1	TC 2	2	GND
3	TQ 2	4	TG 2
5	TC 1	6	GND
7	TQ 1	8	TG 1
9	TC 0	10	GND
11	TQ 0	12	TG 0

J19 - Parallel Interface Connector
(to/from 8255 Programmable Peripheral Interface)

1	GND	2	GND
3	PA 3	4	PA 4
5	PA 2	6	PA 5
7	PA 1	8	PA 6
9	PA 0	10	PA 7
11	GND	12	GND
13	PC 5	14	PC 4
15	PC 7	16	PC 6
17	GND	18	GND
19	PC 1	20	PC 0
21	PC 3	22	PC 2
23	GND	24	GND
25	PB 3	26	PB 4
27	PB 2	28	PB 5
29	PB 1	30	PB 6
31	PB 0	32	PB 7
33	GND	34	GND

Pin assignment of serial handshaking interface connector
(on optional piggyback board)

1	GND	2	CLK 1
3	RxD 1	4	RxRDY 1
5	SYNDET 1	6	TxRDY 1
7	TxD 1	8	TxE 1
9	CTS* 1	10	RSR* 1
11	RTS* 1	12	DTR* 1
13	GND	14	GND
15	CLK 0	16	RxD 0
17	RxRDY 0	18	SYNDET 0
19	TxRDY 0	20	TxD 0
21	TxE 0	22	CTS* 0
23	RSR* 0	24	RTS* 0
25	DTR* 0	26	GND

The S100/IEEE696 Bus

The IEEE standard of the S100 bus appears on the following page. This standard is used on all current Multiflex boards. Note that some signals are not used on Multiflex products. Some exceptions are:

Two of the undefined lines (NDEF) are used for the CPU MREQ* line (pin 65) and the CPU RAM refresh (RFSH*, pin 66) .

pSTVAL* (pin 25) is not used or is tied high on current Multiflex Z80 CPU boards. The 256K Multiflex memory board uses pSTVAL* for its timing.

The ERROR* (pin 98) and PWRFAIL* (pin 13) lines are not used and are left open.

sXTRQ* (pin 58) is tied high by Multiflex 8 bit CPU boards. The Multiflex 256K Memory board responds to this signal for 16 bit transfer.

Note: RFU means Reserved for Future Use. NDEF (Not DEFined) are undedicated lines for manufacturers' use.

A good reference for understanding the S100 bus is 'Interfacing To S-100/IEEE 696 Microcomputers' by Sol Libes and Mark Garetz, published by Osborne/McGraw-Hill. This book is available from Exceltronix Components and Computing.

pin 1	+8V	pin 51	+8V
pin 2	+16V	pin 52	-16V
pin 3	XRDY	pin 53	GND
pin 4	VI0*	pin 54	SLAVE CLR*
pin 5	VI1*	pin 55	DMA0*
pin 6	VI2*	pin 56	DMA1*
pin 7	VI3*	pin 57	DMA2*
pin 8	VI4*	pin 58	sXTRQ*
pin 9	VI5*	pin 59	A19
pin 10	VI6*	pin 60	SIXTN*
pin 11	VI7*	pin 61	A20
pin 12	NMI*	pin 62	A21
pin 13	PWRFAIL*	pin 63	A22
pin 14	DMA3*	pin 64	A23
pin 15	A18	pin 65	NDEF (MERQ*, see notes)
pin 16	A16	pin 66	NDEF (RFSH*, see notes)
pin 17	A17	pin 67	PHANTOM*
pin 18	SDSB*	pin 68	MWRT
pin 19	CDSB*	pin 69	RFU
pin 20	GND	pin 70	GND
pin 21	RFU	pin 71	NDEF
pin 22	ADSB*	pin 72	RDY
pin 23	DODSB*	pin 73	INT*
pin 24	Ø (B)	pin 74	HOLD*
pin 25	pSTVAL*	pin 75	RESET*
pin 26	pHLDA	pin 76	pSYNC
pin 27	RFU	pin 77	pWR*
pin 28	RFU	pin 78	pDBIN
pin 29	A5	pin 79	A0
pin 30	A4	pin 80	A1
pin 31	A3	pin 81	A2
pin 32	A15	pin 82	A6
pin 33	A12	pin 83	A7
pin 34	A9	pin 84	A8
pin 35	DO1	pin 85	A13
pin 36	DO0	pin 86	A14
pin 37	A10	pin 87	A11
pin 38	DO4	pin 88	DO2
pin 39	DO5	pin 89	DO3
pin 40	DO6	pin 90	DO7
pin 41	DI2	pin 91	DI4
pin 42	DI3	pin 92	DI5
pin 43	DI7	pin 93	DI6
pin 44	sM1	pin 94	DI1
pin 45	sOUT	pin 95	DI0
pin 46	sINP	pin 96	sINTA
pin 47	sMEMR	pin 97	sWO*
pin 48	sHLTA	pin 98	ERROR*
pin 49	CLOCK	pin 99	POC*
pin 50	GND	pin 100	GND

A WORD ABOUT . . .

LOGIC LITERALS

The IEEE (Institute of Electrical and Electronic Engineers) has established a new standard for the description of logic symbols. The purpose of this standard is to eliminate ambiguity in the specification of logic literals and to simplify the production of digital documentation.

For the situation where OUT is active when it is LOW, it would appear as OUT* instead of OUT ('OUT barred'). If, on the other hand we wish to talk about the signal when it is low, we would refer to it as -OUT*, indicating that an inversion had taken place.

There are two advantages to this approach. First, suppose you read on a timing diagram a point marked 'OUT'. Now this can be interpreted in two ways. It can be interpreted as saying that OUT is active when it is low. It can also mean that OUT is low at that particular point in time, but would not in that case say anything about the true nature of OUT. This is not usually serious as the correct meaning can be normally determined from the context, however it is preferable to remove all ambiguity.

The new standard also simplifies the production of documentation. First of all, most typesetting equipment and all word processors do not offer the facility of putting a bar over a word. This means that the bar must be added in later on. Murphy's Laws apply here and it is quite easy to leave the odd bar out either by mistake or at some point in the printing process. In this fashion, the new standard greatly reduces the possibility of error in documentation.

The new standard will be used throughout this manual, the sole exception being some schematics, which are hand-drawn.

Z.MON. VER 1.0 Z80 MEMORY MAP (Basic kit)

Socket	Memory Address (Hex)	
U36	0000 0065	- Start of RAM memory (6116 in U36).
	0066 0068	- Non-maskable interrupt (NMI), used by SI, ESC, and BP functions of the monitor.
	0069 03FF	- Variable storage for the monitor, and CPU stack.
	0400 07FF	- Top of memory in most basic kit, (only 1 6116 included).
	0800* 0FFF*	- Image of 0000 through 07FF. (everything reperats)
U37	1000 1FFF	- Z.MON Ver 1.0 in 2732.
U38	2000* 27FF*	- Image of 2800 through 2FFF (6116 or 2716), or 1st optional EPROM (2732).
	2800 2FFF	- 1st optional RAM (6116), or EPROM (2716), or EPROM continued (2732).
U39	3000 37FF	- 2nd optional RAM (6116) or EPROM (2716 or 2732).
	3800* 3FFF*	- Image of 3000 through 37FF (6116 or 2716), or EPROM continued (2732).
	4000 FFFF	- Unused, optional on-board 64K memory, or external memory board.

* If 6116 RAMs are used in the EPROM sockets, then data written into this part of RAM memory will overwrite data stored in the adjacent 2K image of each memory chip. (The cause of this is that address line A11 is ignored by the 6116). Allthough each 6116 will appear throughout its 4K block, there is only 2K of memory, the contents of the chip appearring twice.

With 6116s in U36, U38, and U39 there is 1k of memory available to the user between 0400 and 07FF and a contiguous 8K between 2800 and 37FF. If the on-board dynamic RAM or an external memory board is used, then its memory will appear in these locations if these sockets are not enabled on J13, and no 6116s will be needed.

Z.MON VER 2.0 Z80 MEMORY MAP
(With on-board dynamic RAM and up to 2 user EPROMs)

	0000 0065	- Start of RAM memory
	0066 0068	- Non-maskable interrupt (NMI), used by SI, ESC, and BP functions of the monitor.
	0069 BFFF	- RAM memory
U36	C000 CFFF	- 1st optional EPROM (2716 or 2732). A 2716 will appear twice in this address space.
U37	D000 DFFF	- 2nd optional EPROM.
U38	E000 EEFF	- *THIS EPROM DISABLED*. RAM MUST appear in this space.
	EF00 EFFF	- Variable storage for the monitor, and CPU stack. The stack extends downward in memory from EFA0, and so could if very heavily used, fill up space below EF00 as well.
U39	F000 FFFF	- Z.MON VER 2.0 in 2732.

* Jumper J13 pin 2 must not be connected. RAM will also appear in the address space of any disabled EPROM.

If using the extended address feature with this monitor, then the resident RAM, selected by J12 (See CPU jumpers section), must be selected to begin at E800 or lower in memory if the monitor variables and CPU stack are not to be lost when the extended address is changed. (To set up the resident RAM to begin at B800, connect J8 sections 2 and 3, and J12 pin 3).

Z.MON VER 3.0 Z80 MEMORY MAP
(with on-board dynamic RAM and up to 3 on-board EPROMs).

	0000 0065	- Start of RAM memory.
	0066 0068	- Non-maskable interrupt (NMI), used by SI, ESC, and BP functions of the monitor.
	0069 BEFF	- RAM memory.
	BF00 BFFF	- Variable storage for the monitor, and CPU stack. The stack extends downward in memory from BFA0, and so could if very heavily used, fill up space below BF00 as well.
U36	C000 CFFF	- 1st optional EPROM (2716 or 2732).
U37	D000 DFFF	- 2nd optional EPROM (2716 or 2732).
U38	E000 EFFF	- 3rd optional EPROM (2716 or 2732).
U39	F000 FFFF	- Z.MON VER 3.0 in 2732.

RAM memory will appear in the address space of any disabled EPROM.

If using the extended addressing feature with this monitor, then the resident RAM, selected by J12 (See CPU jumpers section), must be selected to begin at B800 or lower in memory if the CPU stack and monitor variables are not to lost when the extended address is changed. (To set the Resident RAM to begin at B800, jumper J8 sections 2 and 3, and J12 pin 1).

PORTS MAP

The most significant bits of the ports on the CPU are jumper selectable with J6. Our suggested configuration (which is required for running CP/M on our system) is to have the CPU ports appear between port addresses 80 and 8F. If the CPU board is so configured, the 16 ports on the CPU are as follows:

80	- Parallel Port (8255)	- Data register channel A
81		- Data register channel B
82		- Data register channel C
83		- Control register
84	- Interval Timer (8253)	- Counter 0
85		- Counter 1
86		- Counter 2
87		- Control register
88	- Serial Port 0 (8251)	- Data Register
89		- Control Register
8A	- Serial Port 1 (8251)	- Data register
8B		- Control register
8C or 8D	- bits 0 to 4	- Interrupt priority (8214 on optional piggy back board)
	- bits 6 and 7	- EPROM disable register*
8E or 8F	- Extended address register (appears at either port address)	

* A write to either port 8E or 8F which has bit 6 high and bit 7 low will disable the on-board EPROMs, so that the entire address space may be RAM (such as is required for CP/M). A write of anything else to either of these ports will re-enable the EPROMs.

To set up the ports as we suggest here, jumper J6 pins 2, 3, and 4.

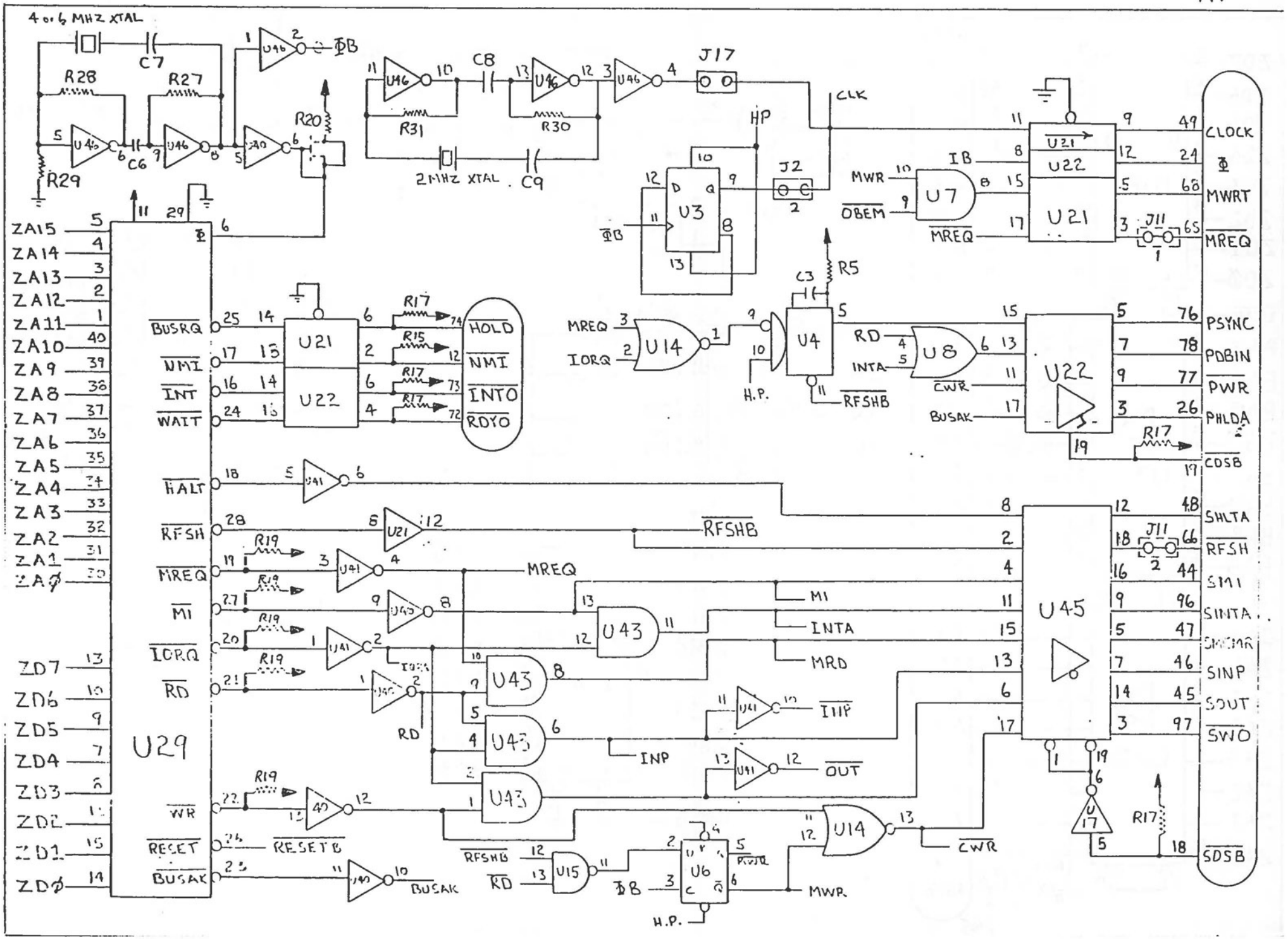
MOTHERBOARD PORT ADDRESSES

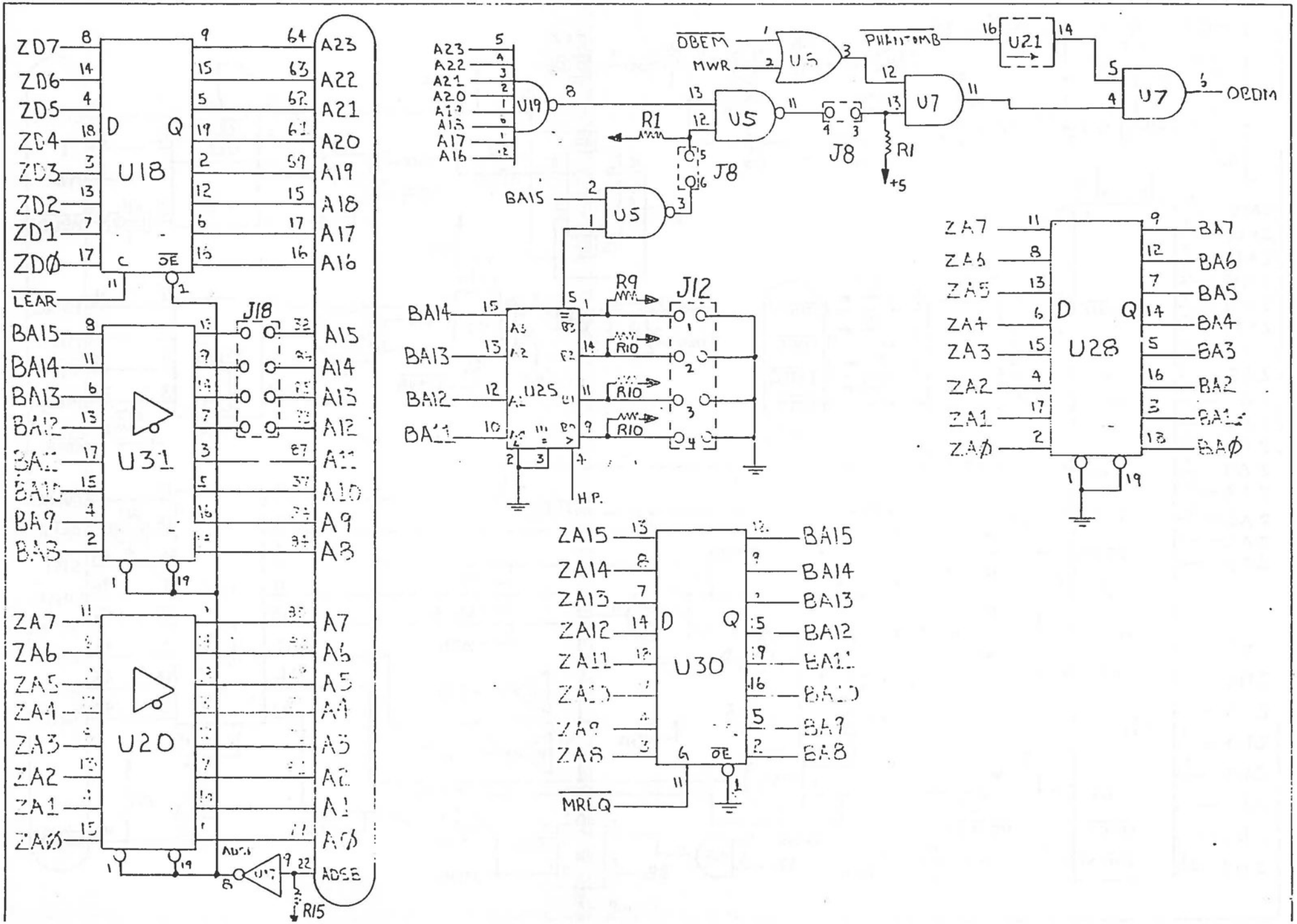
PORT	BIT 7	6	5	4	3	2	1	0	Read/Write
FC:	PGD7	PGD6	PGD5	PGD4	PGD3	PGD2	PGD1	PGDO	R/W PGD
FB:	PPLS	WR	PGA13	PGA12	PGA11	PGA10	PGA9	PGA8	W PGAH
FA:	PGA7	PGA6	PGA5	PGA4	PGA3	PGA2	PGA1	PGAO	W PGAL
	CTTX								
	CTRX	PGV							R PGAL
F9:		SIA	D1	DO	A3	A2	A1	AO	W WLED
	K7	K6	K5	K4	K3	K2	K1	KO	R MKB
F8:		g	f	e	d	c	b	a	W LSEG
	ESC	CTXC							

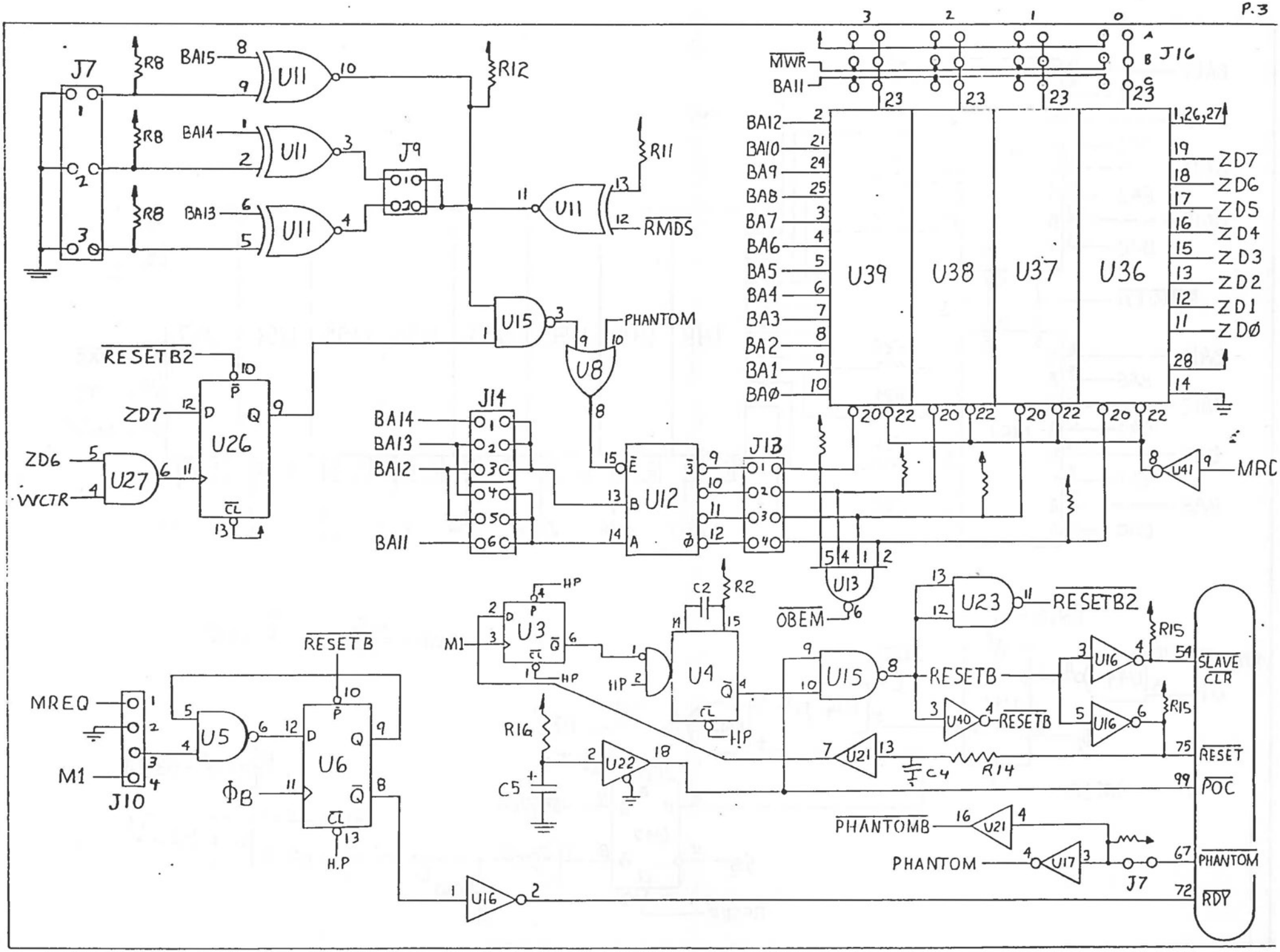
F5:	8251 PCI	- Control
F4:		- Data
F3:	8255 PPI	- Control
F2:		- Port C
F1:		- Port B
FO:		- Port A

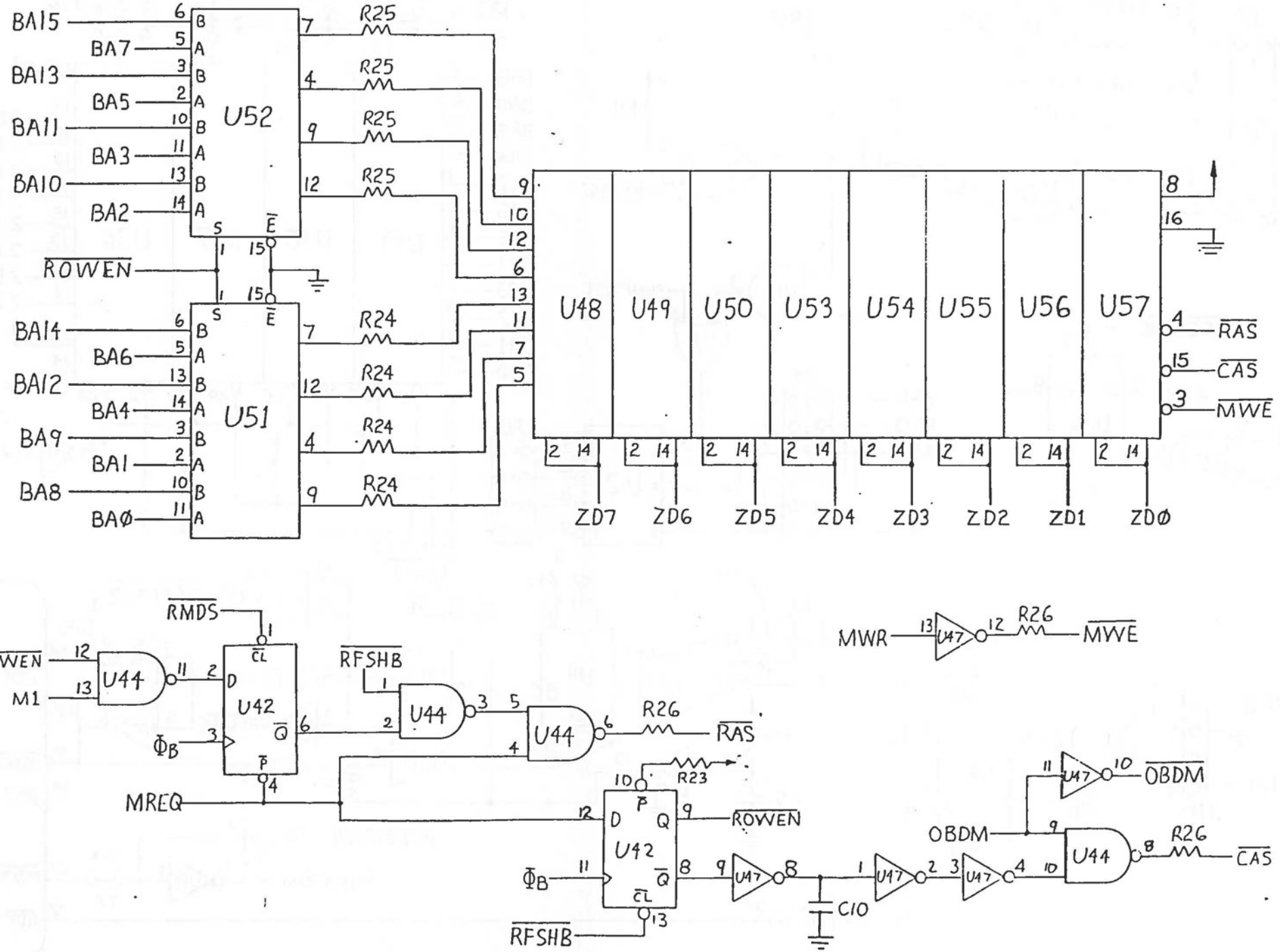
a - g	Individual multiplexed LED segments
AO - A3	Address LED display scan
CTRX	Cassette transition pulse from cassette
CTTX	Cassette transition pulse to cassette
CTXC	Clear cassette transmitter counter
DO - D1	Data LED display scan
ESC	Enables NMI interrupt from ESC key
KO - K7	Keyboard polling
PGA	Progammer address
PGD	Programmer data
PGV	Programmer voltage status
PPLS	Programming pulse
SIA	Single instruction activate
WR	Programmer write

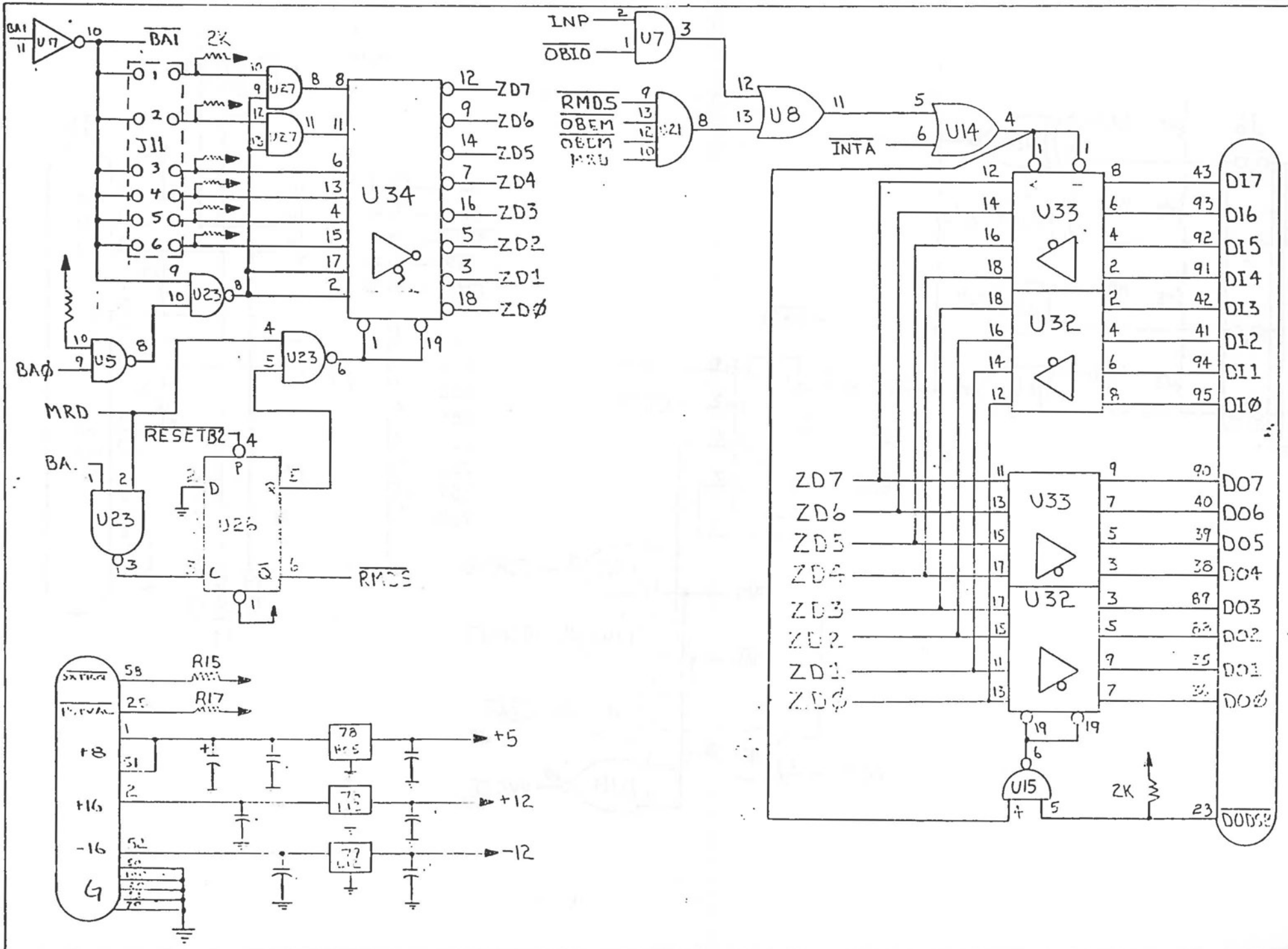
LED segments light up when the corresponding bits are a 0; FF turns display off.

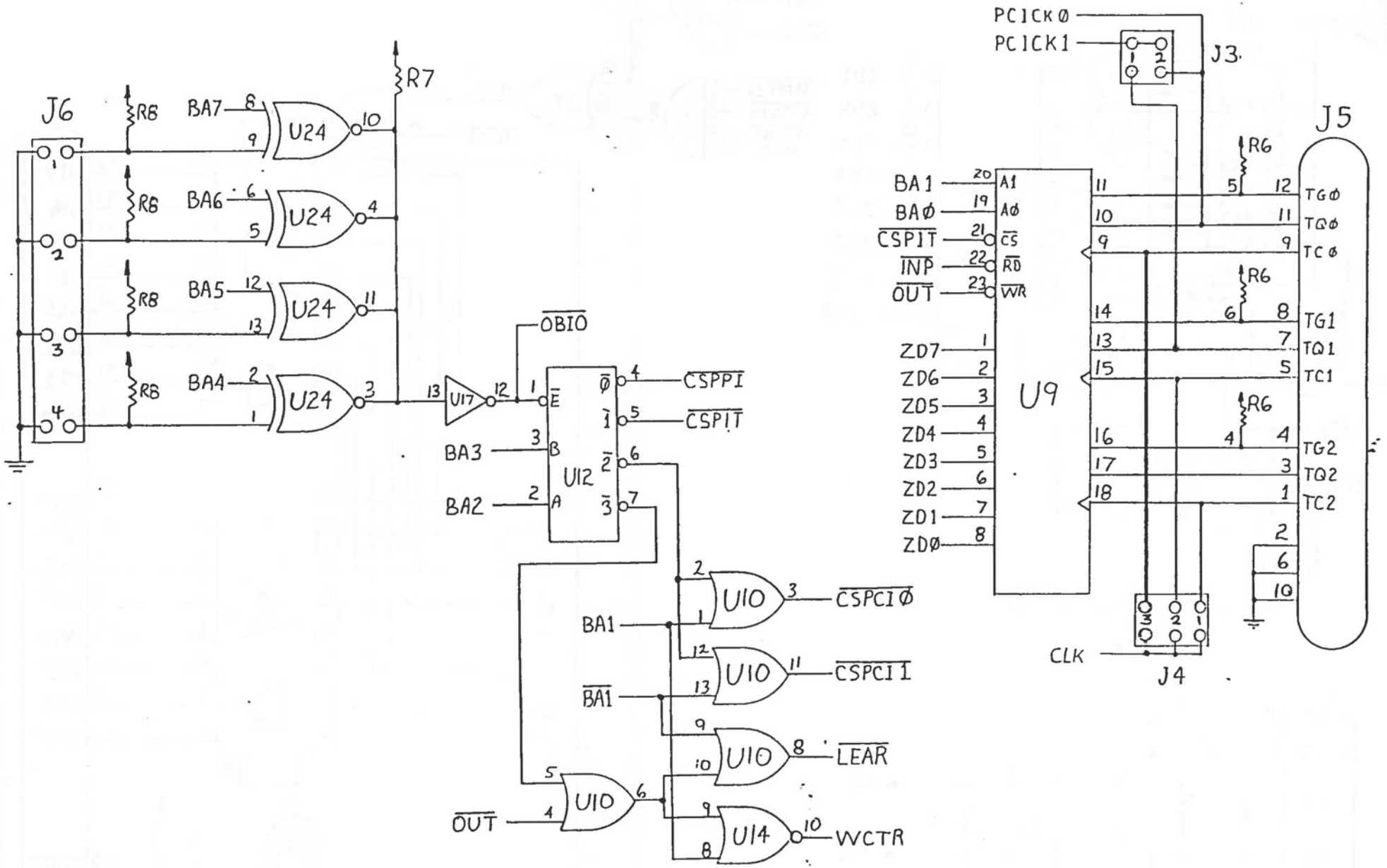

4 or 6 MHZ XTAL
C7
R28
R27
R29
C6
R20
ΦB
C8
R31
R30
2MHZ XTAL
C9
J17
CLK
HP
J2
U3
MWR
OBEM
IB
U7
MREQ
U21
U22
CLOCK
MWRT
J11
MREQ
R5
C3
MREQ
IORQ
U14
U4
H.P.
RFSHB
RD
INTA
U8
CWR
BUSAK
U22
PSYNC
PDBIN
PWR
PHLDA
R17
COSB
ZA15
ZA14
ZA13
ZA12
ZA11
ZA10
ZA9
ZA8
ZA7
ZA6
ZA5
ZA4
ZA3
ZA2
ZA1
ZA0
ZD7
ZD6
ZD5
ZD4
ZD3
ZD2
ZD1
ZD0
U29
BUSRQ
NMI
INT
WAIT
HALT
RFSH
MREQ
M1
IORQ
RD
WR
RESET
BUSAK
R19
R15
HOLD
NMI
INTO
RDYO
RESETB
RFSHB
MREQ
M1
INTA
MRD
INP
IOP
OUT
U43
U41
U45
SHLTA
RFSH
SM1
SINTA
SINP
SOUT
SWO
SDSB
U15
U6
MWR
CWR
H.P.

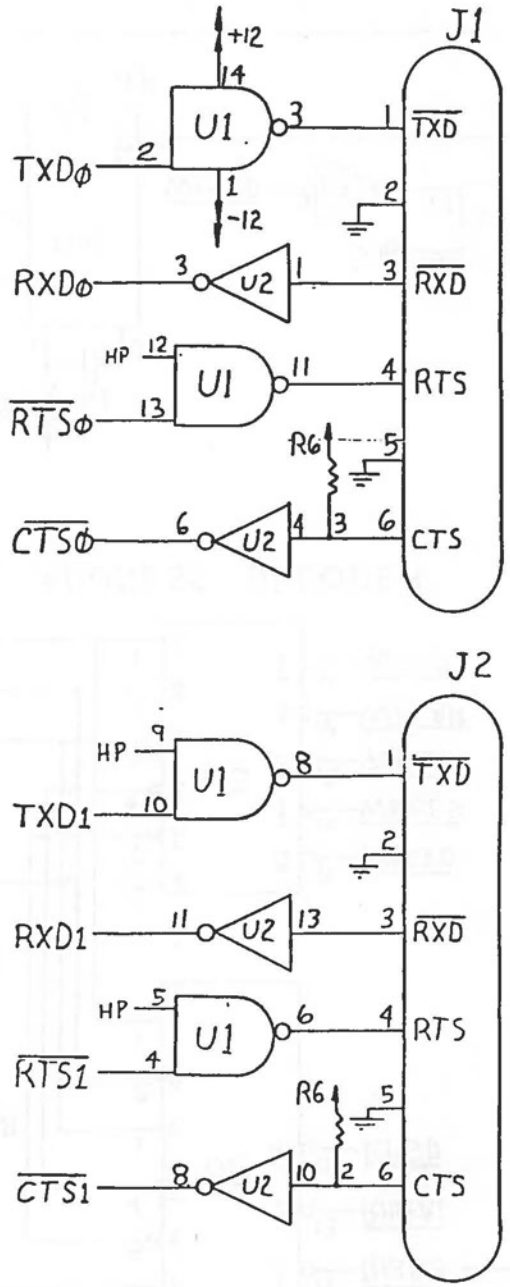

ZD7
ZD6
ZD5
ZD4
ZD3
ZD2
ZD1
ZDØ
U18
D
Q
C
OE
LEAR
A23
A22
A21
A20
A19
A18
A17
A16
J18
BA15
BA14
BA13
BA12
BA11
BA10
BA9
BA8
U31
A15
A14
A13
A12
A11
A10
A9
A8
ZA7
ZA6
ZA5
ZA4
ZA3
ZA2
ZA1
ZAØ
U20
A7
A6
A5
A4
A3
A2
A1
AØ
ADSB
U17
R15
A23
A22
A21
A20
A19
A18
A17
A16
U19
OBEM
MWR
U5
PHANTOMB
U21
U7
OBDIN
R1
J8
+5
BA15
R9
J12
BA14
BA13
BA12
BA11
U25
R10
HP.
ZA7
ZA6
ZA5
ZA4
ZA3
ZA2
ZA1
ZAØ
U28
BA7
BA6
BA5
BA4
BA3
BA2
BA1
BAØ
ZA15
ZA14
ZA13
ZA12
ZA11
ZA10
ZA9
ZA8
U30
BA15
BA14
BA13
BA12
BA11
BA10
BA9
BA8
MRLQ

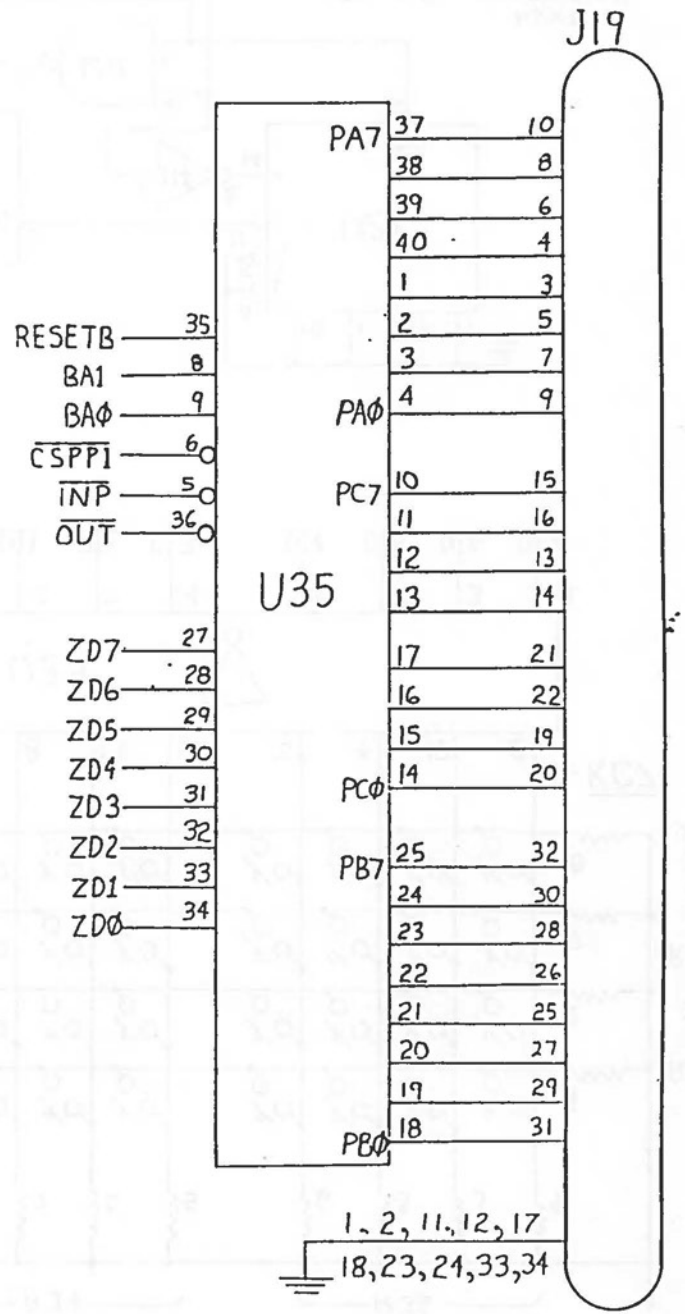

J7
R8
R8
R8
BA15
BA14
BA13
U11
U11
U11
J9
R12
R11
U11
RMDS
U15
PHANTOM
U8
RESETB2
ZD7
ZD6
WCTR
U26
U27
J14
BA14
BA13
BA12
BA11
U12
J13
U13
OBEM
J16
MWR
BA11
BA12
BA10
BA9
BA8
BA7
BA6
BA5
BA4
BA3
BA2
BA1
BA0
U39
U38
U37
U36
ZD7
ZD6
ZD5
ZD4
ZD3
ZD2
ZD1
ZD0
U41
MRD
U3
U4
C2
R2
HP
HP
U15
U23
RESETB2
RESETB
U16
U40
R15
SLAVE CLR
RESET
POC
PHANTOM
RDY
U21
C4
R14
U22
R16
C5
MREQ
M1
J10
U5
U6
ΦB
H.P
U16
PHANTOMB
PHANTOM
U21
U17
J7

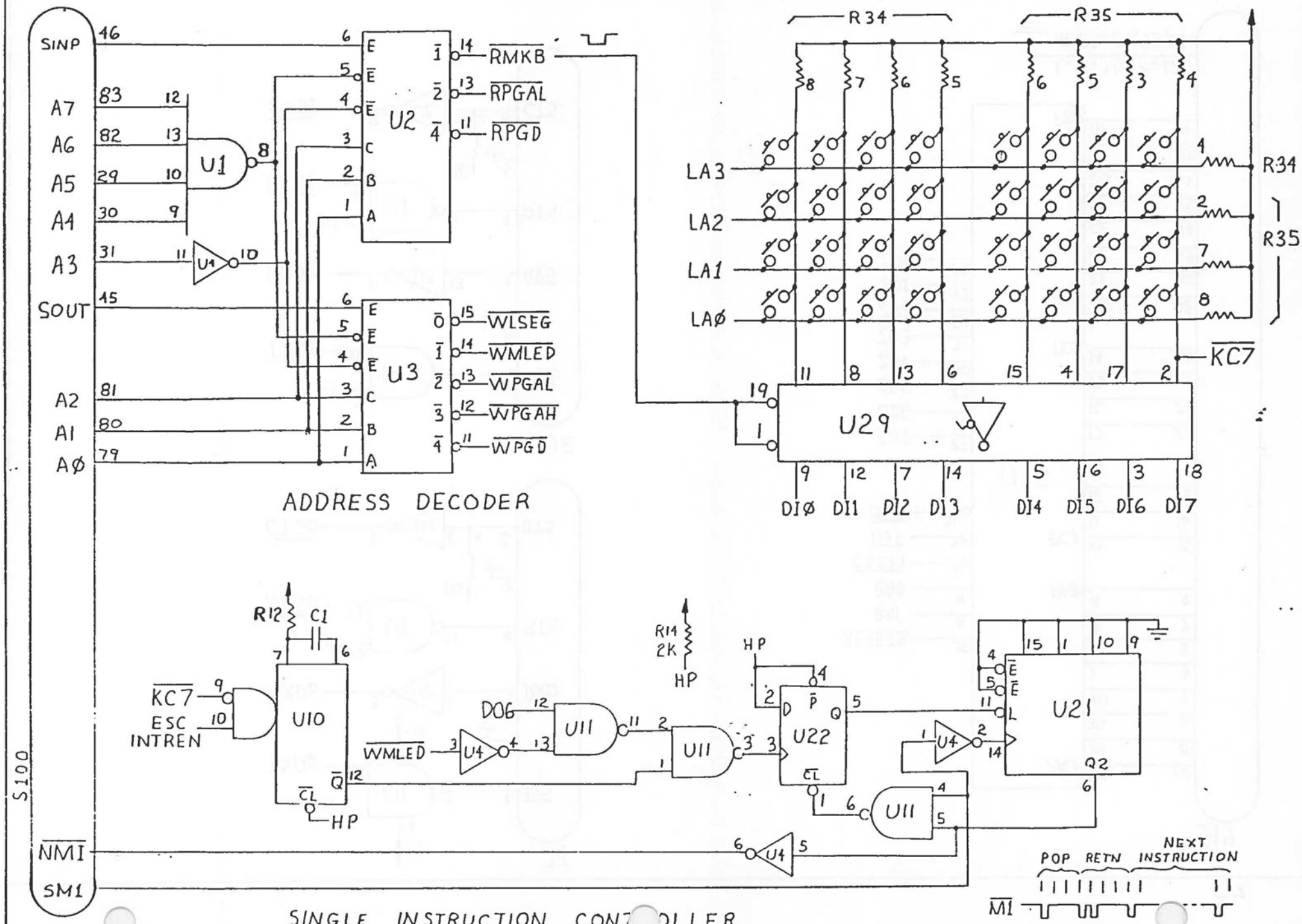

BA15
BA7
BA13
BA5
BA11
BA3
BA10
BA2
U52
ROWEN
U51
BA14
BA6
BA12
BA4
BA9
BA1
BA8
BAØ
R25
R24
U48
U49
U50
U53
U54
U55
U56
U57
RAS
CAS
MWE
ZD7
ZD6
ZD5
ZD4
ZD3
ZD2
ZD1
ZDØ
RMDS
ROWEN
M1
U44
U42
ΦB
MREQ
RFSHB
R26
R23
MWR
U47
OBDM
C10

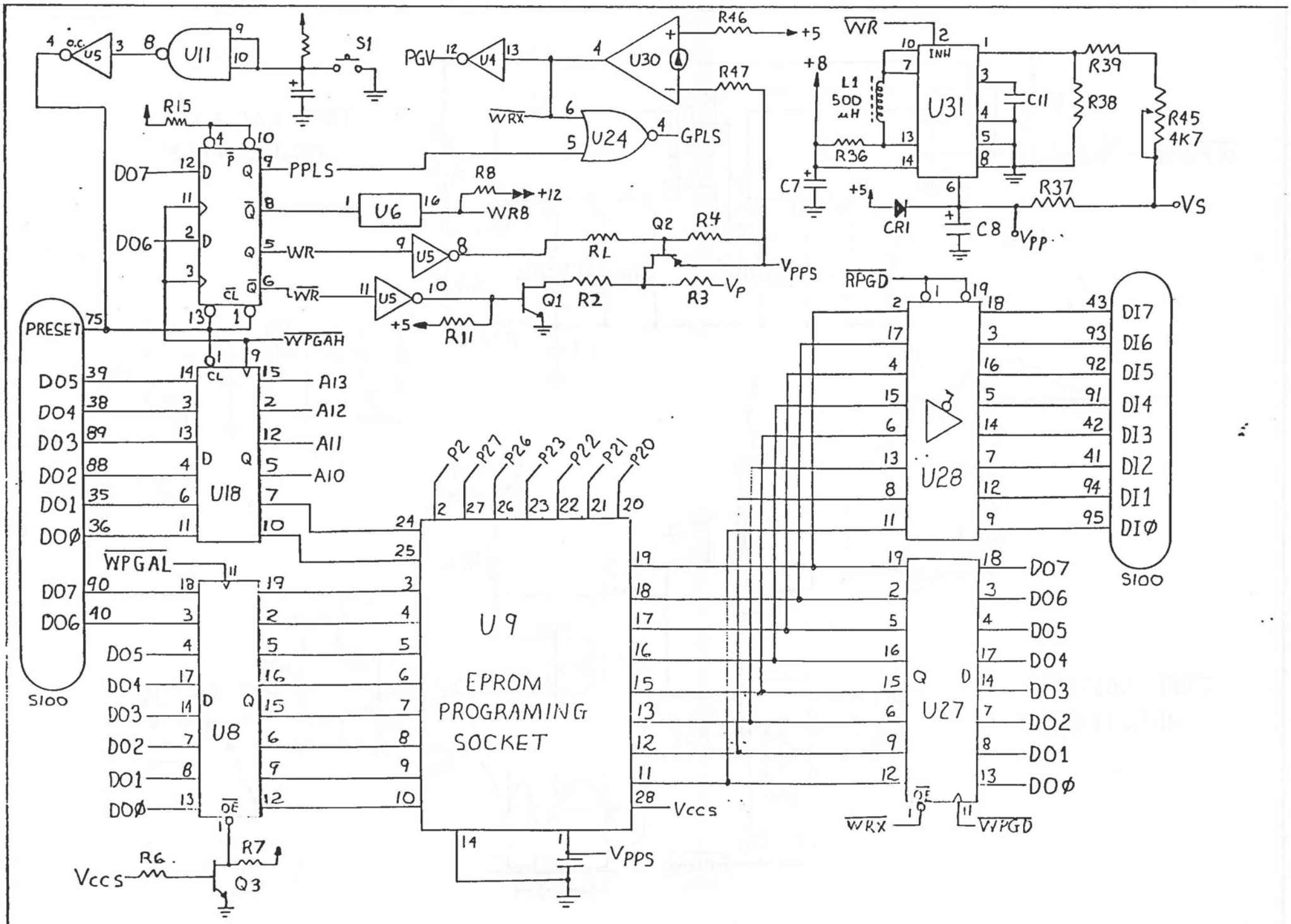

BAI
U17
BAI
2K
J11
U27
U34
U23
U5
BAØ
MRD
RESETB2
BA.
U26
RMDS
ZD7
ZD6
ZD5
ZD4
ZD3
ZD2
ZD1
ZDØ
INP
OBIO
U7
RMDS
OBEM
OBIM
MRD
U21
U8
INTA
U14
U33
U32
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DIØ
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DOØ
U15
2K
DODS2
R15
R17
+8
+16
-16
+5
+12
-12

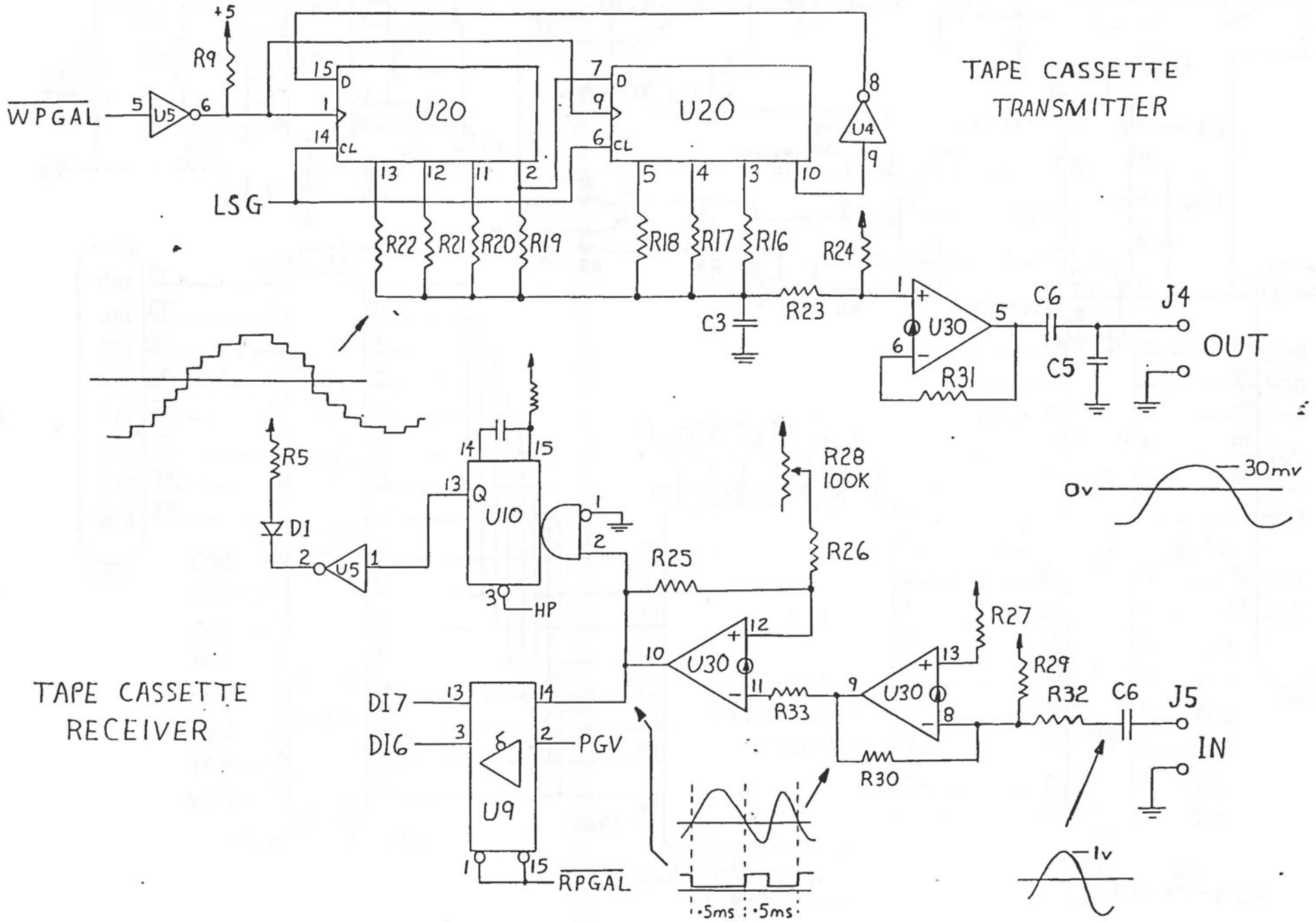

J6
R8
BA7
BA6
BA5
BA4
U24
R7
U17
OBIO
U12
BA3
BA2
CSPPI
CSPIT
U10
BA1
BA1
OUT
U14
WCTR
LEAR
CSPCI0
CSPCI1
PCICK0
PCICK1
J3
J5
R6
U9
BA1
BA0
CSPIT
INP
OUT
ZD7
ZD6
ZD5
ZD4
ZD3
ZD2
ZD1
ZD0
TG0
TQ0
TC0
TG1
TQ1
TC1
TG2
TQ2
TC2
CLK
J4

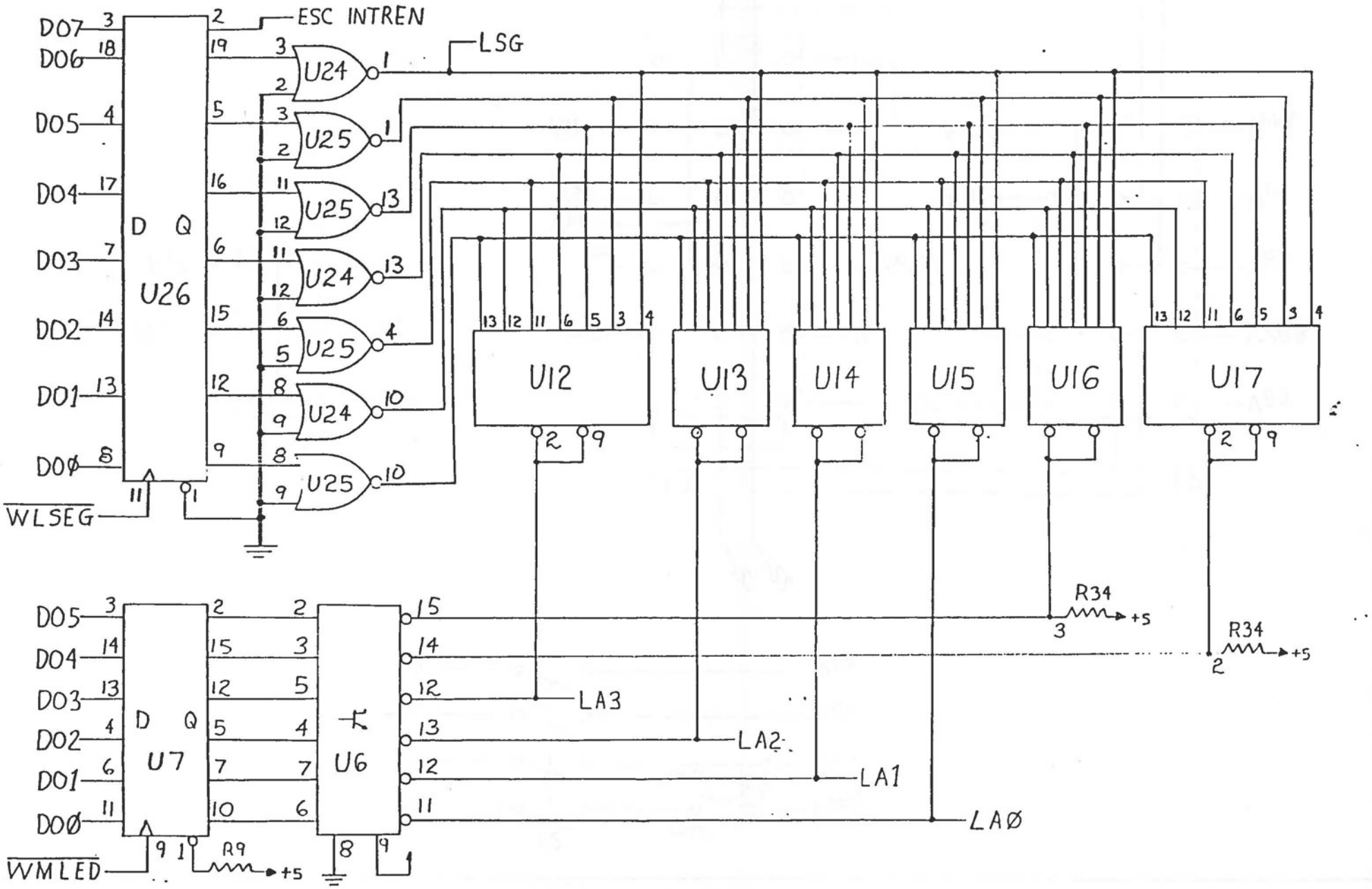

J1
+12
14
U1
3
1
TXD
TXDØ
2
1
-12
2
RXDØ
3
U2
1
3
RXD
HP
12
U1
11
4
RTS
RTSØ
13
R6
5
CTSØ
6
U2
4
3
6
CTS
J2
HP
9
U1
8
1
TXD
TXD1
10
2
RXD1
11
U2
13
3
RXD
HP
5
U1
6
4
RTS
RTS1
4
R6
5
CTS1
8
U2
10
2
6
CTS

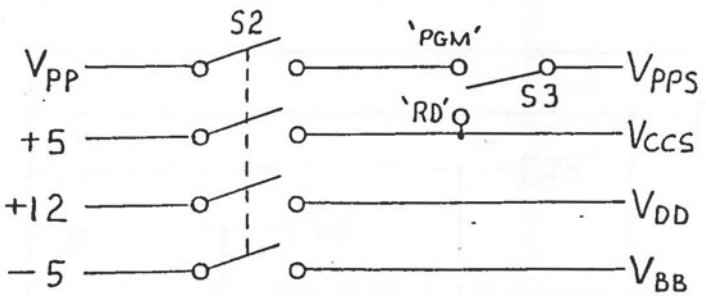

J19
U35
RESETB 35
BA1 8
BAØ 9
CSPPI 6
INP 5
OUT 36
ZD7 27
ZD6 28
ZD5 29
ZD4 30
ZD3 31
ZD2 32
ZD1 33
ZDØ 34
PA7 37 10
38 8
39 6
40 4
1 3
2 5
3 7
PAØ 4 9
PC7 10 15
11 16
12 13
13 14
17 21
16 22
15 19
PCØ 14 20
PB7 25 32
24 30
23 28
22 26
21 25
20 27
19 29
PBØ 18 31
1, 2, 11, 12, 17
18, 23, 24, 33, 34

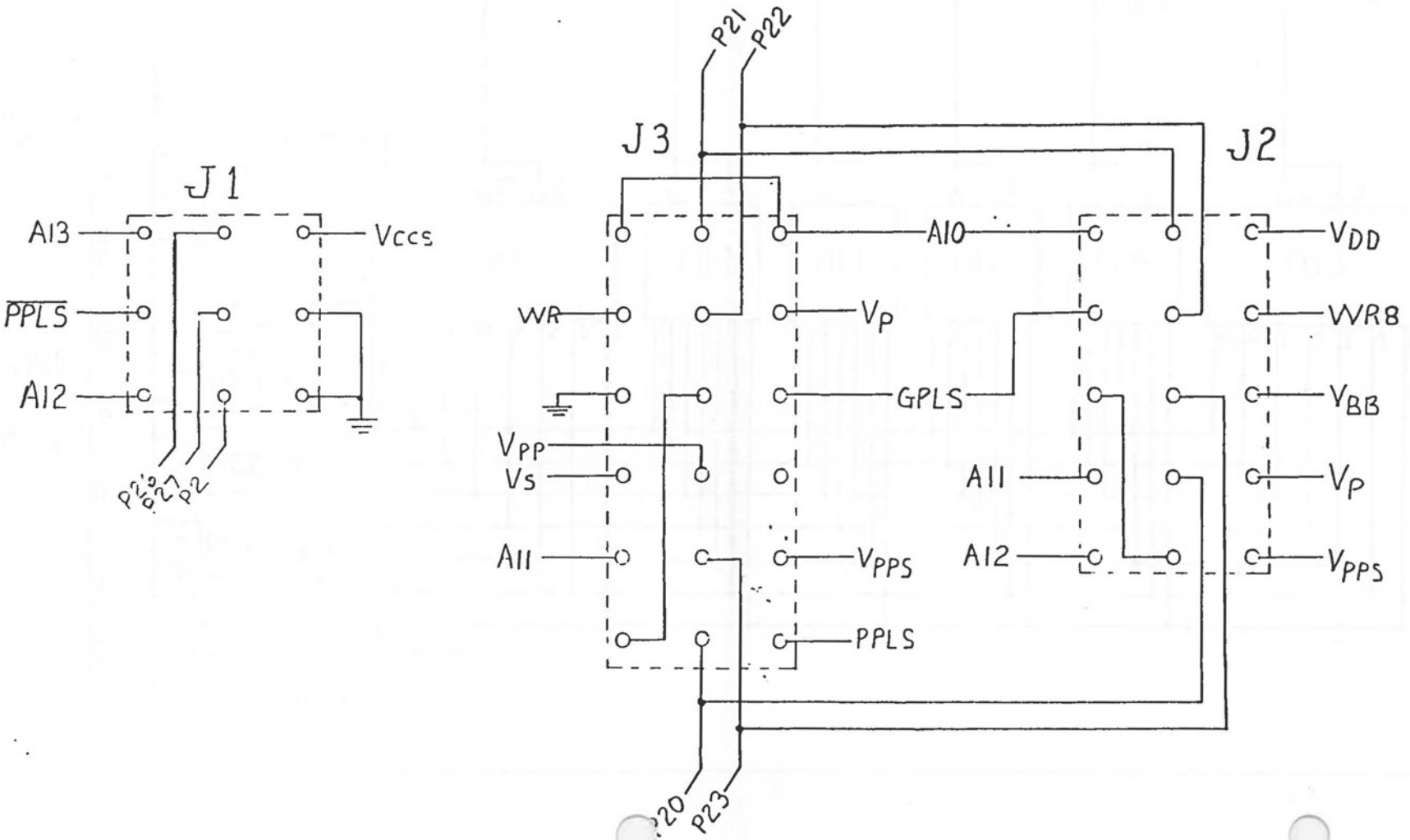

SINP
A7
A6
A5
A4
A3
SOUT
A2
A1
A0
S100
NMI
SM1
U1
U2
U3
U4
RMKB
RPGAL
RPGD
WLSEG
WMLED
WPGAL
WPGAH
WPGD
ADDRESS DECODER
R34
R35
LA3
LA2
LA1
LA0
KC7
U29
DI0
DI1
DI2
DI3
DI4
DI5
DI6
DI7
R12
C1
KC7
ESC
INTREN
U10
HP
WMLED
DOG
U11
R14
2K
U22
U21
Q2
NEXT
POP
RETN
INSTRUCTION
MI
SINGLE INSTRUCTION CONTROLLER

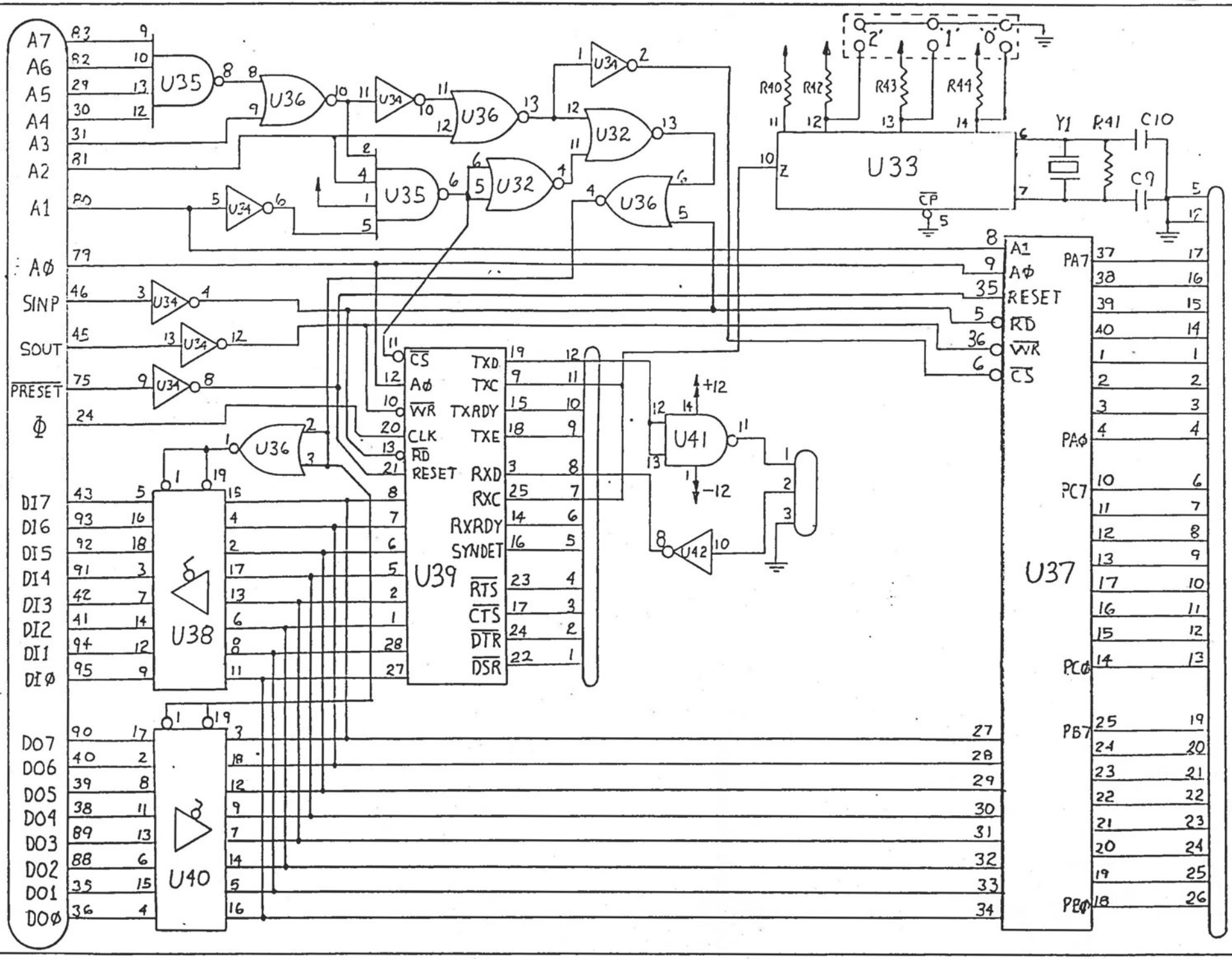

U9
EPROM
PROGRAMING
SOCKET
U8
U18
U28
U27
U31
U30
U24
U11
U6
U4
U5
S100
PRESET
WPGAL
WPGAH
WPGD
RPGD
WRX
WR
PGV
PPLS
GPLS
WR8
VPP
VPPS
VCCS
VS
+5
+8
+12
R15
R46
R47
R39
R38
R45
4K7
R37
R36
R8
RL
R4
R2
R3
R11
R6
R7
C11
C7
C8
CR1
L1
500
μH
Q1
Q2
Q3
P2
P27
P26
P23
P22
P21
P20
A13
A12
A11
A10
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DIØ
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DOØ


TAPE CASSETTE
TRANSMITTER
+5
R9
WPGAL
U5
U20
U20
U4
LSG
R22
R21
R20
R19
R18
R17
R16
R24
R23
C3
U30
R31
C6
C5
J4
OUT
30mv
0v
R5
D1
U10
HP
R28
100K
R26
R25
R27
R29
R32
C6
J5
IN
R33
R30
U30
U30
TAPE CASSETTE
RECEIVER
DI7
DI6
U9
PGV
RPGAL
1v
·5ms
·5ms


DO7
DO6
DO5
DO4
DO3
DO2
DO1
DOØ
D Q
U26
WLSEG
ESC INTREN
LSG
U24
U25
U12
U13
U14
U15
U16
U17
R34
+5
D Q
U7
U6
WMLED
R9
LA3
LA2
LA1
LAØ


S2
'PGM'
S3
'RD'
VPP
+5
+12
−5
VPPS
VCCS
VDD
VBB


J1
A13
PPLS
A12
VCCS
P23
P27
P2
J3
P21
P22
WR
VPP
Vs
A11
VP
GPLS
VPPS
PPLS
A10
A11
A12
J2
VDD
WRB
VBB
VP
VPPS
P20
P23


A7
A6
A5
A4
A3
A2
A1
AØ
SINP
SOUT
PRESET
Ø
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DIØ
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DOØ
U35
U36
U34
U32
U33
U38
U39
U40
U41
U42
U37
R40
R42
R43
R44
Y1
R41
C10
C9
+12
-12
TXD
TXC
TXRDY
TXE
RXD
RXC
RXRDY
SYNDET
RTS
CTS
DTR
DSR
RESET
PA7
PAØ
PC7
PCØ
PB7
PBØ

```
                        TITLE Z.MON VERSION 1.0
                        .Z80

                        ; COPYRIGHT 1981 MULTIFLEX TECH. INC.

0010                    GO      EQU     10H     ; KEY CODES
0011                    EM      EQU     11H
0012                    EP      EQU     12H
0013                    ER      EQU     13H
0014                    MV      EQU     14H
0015                    SI      EQU     15H
0016                    LD      EQU     16H
0017                    BP      EQU     17H
0018                    SV      EQU     18H
0019                    ESC     EQU     19H
001A                    PGM     EQU     1AH
001B                    CMP     EQU     1BH
001C                    OFT     EQU     1CH
001D                    EIO     EQU     1DH
001E                    DWN     EQU     1EH
001F                    CPM     EQU     1FH

                        ; FLOPPY REGISTERS FOR CP/M

0050                    WRST    EQU     50H
0053                    WSEL    EQU     53H
0054                    FCON    EQU     54H
0056                    FSEC    EQU     56H
0057                    FDATA   EQU     57H
0081                    FLOPA   EQU     0081H
0082                    FLOPB   EQU     0082H

                        ; MONITOR REGISTERS

008E                    EXA     EQU     08EH    ; EXTENDED ADDRESS
00F4                    SER1    EQU     0F4H    ; SERIAL DATA REGISTER
00F5                    SER2    EQU     0F5H    ; SERIAL CONTROL REGISTER
00F8                    LSEG    EQU     0F8H    ; LED DISPLAY SEGMENTS
00F9                    MLED    EQU     0F9H    ; DISP DIGIT SELECT
00F9                    MKB     EQU     0F9H    ; KEYBOARD CODE
00FA                    PGAL    EQU     0FAH    ; EPROM PROGRAMMER ADDRESS
00FB                    PGAH    EQU     0FBH
00FC                    PGD     EQU     0FCH    ; PROGRAMMER DATA

03A0                    STKB    EQU     003A0H  ; STACK BASE
03A8                    DEVICE  EQU     003A8H  ; MONITOR RAM LOCATIONS
03AA                    ENDADD  EQU     003AAH
03AC                    STADDR  EQU     003ACH
03AE                    DSTADD  EQU     003AEH
03B0                    BYTCNT  EQU     003B0H
03B4                    CTF     EQU     003B4H  ; CONTIN FLG
03B5                    ERSC    EQU     003B5H  ; REG SEL CODE
03B6                    SSPL    EQU     003B6H  ; SAVED SP
03B8                    SRR     EQU     003B8H
03B9                    SVI     EQU     003B9H
03BA                    ESREG   EQU     003BAH  ; END OF SAVED REG AREA
03CE                    SREG    EQU     003CEH  ; START OF SAVED REG AREA
03CE                    SPCL    EQU     003CEH  ; SAVED PC
03CF                    SPCH    EQU     003CFH
03D0                    BPBA    EQU     003D0H  ; BRK-PT'S TBL BASE ADDR
```

```
03F0                        MDB0    EQU     003F0H  ; A0    LED DISPLAY
03F1                        MDB1    EQU     003F1H  ; A1          BUFFER
03F2                        MDB2    EQU     003F2H  ; A2
03F3                        MDB3    EQU     003F3H  ; A3 (MSD)
03F4                        MDB4    EQU     003F4H  ; D0
03F5                        MDB5    EQU     003F5H  ; D1 (MSD)
03F6                        BPN     EQU     003F6H  ; BRK-PT INTR FLG
03F7                        SCA     EQU     003F7H  ; SAVED CURRENT ADDR
03F9                        BPF     EQU     003F9H  ; BRK-PT'S-ACTIVE FLG
03FA                        CBPNM   EQU     003FAH  ; CUR BRK-PT NUM
03FB                        PRE     EQU     003FBH  ; PRE-AMBLE SYNC BYTE
03FC                        SAL     EQU     003FCH  ; SAVED CASSETTE START ADDR
03FE                        BCL     EQU     003FEH  ; SAVED BYTE CNT

                                    .PHASE  01000H

1000    3E FF               ST:     LD      A, 0FFH         ; SETUP THE EXTENDED ADDRESS
1002    D3 8E                       OUT     (EXA),A
1004    21 03A0                     LD      HL, STKB        ; INITIALIZE THE STACK POINTER
1007    AF                          XOR     A
1008    06 50                       LD      B, 50H
100A    F9                          LD      SP, HL          ; INITIALIZE STACK
100B    77                  ST1:    LD      (HL), A
100C    23                          INC     HL              ; CLR MON DATA LOC'S
100D    10 FC                       DJNZ    ST1
100F    CD 10E0             CTST:   CALL    CLDSP           ; CLR DISP BUF
1012    3E 10                       LD      A, 10H          ; DISP UNDERBAR PROMPT
1014    32 03F3                     LD      (MDB3), A               ; IN A3 DISP
1017    CD 1102             CTST1:  CALL    GETAD           ; GET ADDR & CMD FROM KBD
101A    FE 19                       CP      ESC
101C    28 F1                       JR      Z, CTST         ; BACK TO CMD PROMPT
101E    47                          LD      B, A            ; SAVE CMD, TEMP
101F    79                          LD      A, C
1020    B7                          OR      A               ; CHK IF ADDR PRESENT
1021    78                          LD      A, B
1022    20 39                       JR      NZ, SA1         ; IS ADDR PRESENT ?
1024    FE 17                       CP      BP              ; NO: CHK FOR GP 1 CMDS
1026    CA 16E0                     JP      Z, STBP         ; DEFINE BREAK-POINTS
1029    FE 10                       CP      GO
102B    CA 1AD3                     JP      Z, CTCON        ; CONTIN PROG EXECUTION
102E    FE 1A                       CP      PGM
1030    CA 11E0                     JP      Z, RDPGM        ; READ FROM PGM SOCKET
1033    FE 15                       CP      SI
1035    CA 1AD8                     JP      Z, CTSI         ; EXEC NEXT SINGLE INSTR
1038    FE 13                       CP      ER
103A    CA 1AB4                     JP      Z, STER         ; EXAM REG'S AGAIN
103D    FE 11                       CP      EM
103F    CA 12D8                     JP      Z, CTEM         ; EXAM CURRENT MEM LOC
1042    FE 16                       CP      LD
1044    CA 1B13                     JP      Z, LDCT         ; LOAD INTO MEM FROM CASSETTE
1047    FE 1D                       CP      EIO
1049    CA 15F5                     JP      Z, STEIO        ; EXAM I/O PORTS
104C    FE 1E                       CP      DWN
104E    CA 1D08                     JP      Z, DNLD         ; DOWNLOAD
1051    FE 1F                       CP      CPM
1053    CA 1D4A                     JP      Z, BOOT         ; ENTER CP/M
1056    3E 12               CDER:   LD      A, 12H
1058    32 03F3                     LD      (MDB3), A       ; DISP TRIPLE-BAR FOR ILLEG CMD
105B    18 BA                       JR      CTST1
105D    FE 18               SA1:    CP      SV              ; CHK FOR GP 2 CMDS
105F    CA 1B7A                     JP      Z, STSV         ; SAVE MEM BLOCK ONTO CASSETTE
```

```
1062  FE 15                    CP    SI
1064  CA 1AC4                  JP    Z, STSI          ; EXEC A SPEC'D SINGLE INSTR
1067  FE 14                    CP    MV
1069  CA 1302                  JP    Z, STMV          ; MOVE A BLOCK IN MEM
106C  FE 1A                    CP    PGM
106E  CA 13B3                  JP    Z, STPGM         ; PROGRAM AN EPROM
1071  FE 10                    CP    GO
1073  CA 1A48                  JP    Z, START         ; START EXEC OF A PROG
1076  FE 1B                    CP    CMP
1078  CA 126D                  JP    Z, STCMP         ; COMPARE 2 BLOCKS IN MEM
107B  FE 11                    CP    EM
107D  CA 113F                  JP    Z, STEM          ; EXAM A SPEC'D MEM LOC
1080  FE 1C                    CP    OFT
1082  CA 1355                  JP    Z, STOFT         ; CALC A BR OFFSET
1085  CD 10E0                  CALL  CLDSP
1088  18 CC                    JR    CDER             ; ILLEG CMD

                               ; ROUTINE TO REFRESH LED DISP & SCAN KBD

108A  D9                 DISP: EXX
108B  16 00              DS1:  LD    D, 0
108D  0E 01                    LD    C, 1             ; LED DIGIT & KBD ROW SEL
108F  DD 21 03F0               LD    IX, MDB0         ; LED DISP BUF PTR
1093  79                 DS2:  LD    A, C
1094  D3 F9                    OUT   (MLED), A        ; SEL LED DISP DIGIT
1096  DD 5E 00                 LD    E, (IX+0)        ; GET DIGIT FROM BUF
1099  21 16A4                  LD    HL, LSGTB
109C  19                       ADD   HL, DE           ; INDEX INTO LED SEG TBL
109D  7E                       LD    A, (HL)
109E  D3 F8                    OUT   (LSEG), A        ; DISP THE DIGIT
10A0  3E FF                    LD    A, 0FFH
10A2  3D                 DS3:  DEC   A
10A3  20 FD                    JR    NZ, DS3          ; 1 MS DELAY
10A5  3E 7F                    LD    A, 7FH
10A7  D3 F8                    OUT   (LSEG), A        ; BLANK THE DISP
10A9  DB F9                    IN    A, (MKB)         ; RD KBD COL'S
10AB  B7                       OR    A
10AC  20 0B                    JR    NZ, DS4          ; IS KEY PRESSED ?
10AE  DD 23                    INC   IX               ; NO: SEL NEXT DIGIT IN BUF
10B0  CB 01                    RLC   C                ; SEL NEXT LED DIGIT
10B2  79                       LD    A, C
10B3  E6 40                    AND   40H
10B5  28 DC                    JR    Z, DS2           ; MORE DIGITS ?
10B7  18 D2                    JR    DS1              ; NO: START OVER AGAIN
10B9  CD 10F4            DS4:  CALL  KBDB             ; KBD DEBOUNCE DELAY
10BC  57                       LD    D, A             ; GET KBD COL POSITION
10BD  28 CC                    JR    Z,DS1            ; FALSE ALARM ?
10BF  59                       LD    E, C
10C0  CD 10EC                  CALL  KEYCT            ; NO: CALC ROW NUM
10C3  78                       LD    A, B
10C4  07                       RLCA                   ; MULT BY 8
10C5  07                       RLCA
10C6  07                       RLCA
10C7  5A                       LD    E, D
10C8  CD 10EC                  CALL  KEYCT            ; CALC COL NUM
10CB  80                       ADD   A, B             ; COMBINE TO FORM KEY CODE
10CC  5F                       LD    E, A                     ; TABLE INDEX
10CD  16 00                    LD    D, 0
10CF  21 16C0                  LD    HL, KEYCD        ; KEY CODE TBL BASE
10D2  19                       ADD   HL, DE
10D3  DB F9              DS5:  IN    A, (MKB)         ; RD KBD
```

```
10D5  B7                        OR      A
10D6  20 FB                     JR      NZ, DS5         ; WAIT FOR KEY TO BE REL'D
10D8  CD 10F4           DS6:    CALL    KBDB            ; KBD DEBOUNCE DELAY
10DB  20 FB                     JR      NZ, DS6         ; KEY STILL PRESSED ?
10DD  7E                        LD      A, (HL)         ; NO: GET KEY CODE FROM TBL
10DE  D9                        EXX
10DF  C9                        RET                     ; RET KEY CODE IN A

                                ; ROUTINE TO CLEAR LED DISP

10E0  3E 11             CLDSP:  LD      A, 11H          ; BLANK CHAR
10E2  06 06                     LD      B, 6
10E4  21 03F0                   LD      HL, MDB0        ; START WITH FIRST DIGIT
10E7  77                CLD1:   LD      (HL), A         ; FILL IT WITH BLANK
10E8  23                        INC     HL
10E9  10 FC                     DJNZ    CLD1            ; SAME FOR REST OF DIGITS
10EB  C9                        RET

                                ; ROUTINE TO CALC A NUM FROM A BIT POSITION

10EC  06 00             KEYCT:  LD      B, 0
10EE  CB 3B             KYC1:   SRL     E               ; GET NEXT BIT
10F0  C8                        RET     Z               ; A '1' BIT ?  RET NUM IN B
10F1  04                        INC     B               ; NO: COUNT THIS BIT POSITION
10F2  18 FA                     JR      KYC1

                                ; ROUTINE TO GENERATE KBD DEBOUNCE DELAY

10F4  3E FF             KBDB:   LD      A, OFFH
10F6  16 12             KDB1:   LD      D, 12H
10F8  15                KDB2:   DEC     D
10F9  20 FD                     JR      NZ, KDB2
10FB  3D                        DEC     A
10FC  20 F8                     JR      NZ, KDB1        ; 20 MS DELAY
10FE  DB F9                     IN      A, (MKB)        ; RD KBD
1100  B7                        OR      A
1101  C9                        RET                     ; RET KBD COL POS IN A, & Z FLG

                                ; ROUTINE TO GET ADDR & CMD FROM KBD

1102  21 03F3           GETAD:  LD      HL, MDB3        ; SET BUF PTR TO ADDR MSD
1105  06 04                     LD      B, 4            ; ONLY 4 ADDR DIGITS ALLOWED
1107  0E 00                     LD      C, 0            ; NO-ADDR IND (GP 1 CMDS)
1109  CD 108A                   CALL    DISP            ; REFRESH DISP & SCAN KBD
110C  FE 10                     CP      GO              ; FIRST CMD KEY CODE ABOVE HEX DIG'S
110E  F0                        RET     P               ; KEY CODE > OFH ?  RET CMD IN A
110F  0C                        INC     C               ; NO: GOT ADDR MSD; SET IND FOR GP 2 CMDS
1110  77                GETA3:  LD      (HL), A         ; PUT DIGIT IN BUF
1111  2B                        DEC     HL
1112  05                        DEC     B
1113  28 0D                     JR      Z, GETA1        ; 4TH DIGIT ?  GO LOOK FOR CMD
1115  CD 108A           GETA2:  CALL    DISP            ; NO: DISP IT & LOOK FOR MORE
1118  FE 19                     CP      ESC
111A  C8                        RET     Z               ; RET ESC KEY CODE IN A
111B  FE 10                     CP      GO
111D  F2 1115                   JP      P, GETA2        ; CMD KEY ?  IGNORE & LOOK AGAIN
1120  18 EE                     JR      GETA3           ; NO: DIGIT KEY
1122  CD 108A           GETA1:  CALL    DISP            ; DISP & SCAN KBD
1125  FE 10                     CP      GO
1127  F0                        RET     P               ; CMD KEY ?  RET CMD CODE IN A
1128  18 F8                     JR      GETA1           ; NO: IGNORE & LOOK AGAIN
```

```
                                        ; ROUTINE TO TRANSLATE & ASSEMBLE ADDR FROM DISP BUF INTO A NUM

112A    21 03F3         ADTR:   LD      HL, MDB3        ; SET BUF PTR TO ADDR MSD
112D    7E                      LD      A, (HL)
112E    2B                      DEC     HL
112F    87                      ADD     A, A            ; SHIFT MSD 4 LEFT
1130    87                      ADD     A, A
1131    87                      ADD     A, A
1132    87                      ADD     A, A
1133    B6                      OR      (HL)            ; INSERT 2ND MSD
1134    57                      LD      D, A            ; SAVE ADDR HI-BYTE
1135    2B                      DEC     HL
1136    7E                      LD      A, (HL)
1137    2B                      DEC     HL
1138    87                      ADD     A, A            ; SHIFT 3RD MSD 4 LEFT
1139    87                      ADD     A, A
113A    87                      ADD     A, A
113B    87                      ADD     A, A
113C    B6                      OR      (HL)            ; INSERT LSD
113D    5F                      LD      E, A            ; SAVE ADDR LO-BYTE
113E    C9                      RET                     ; RET ADDR IN DE

                                        ; ROUTINE TO EXAM & MODIFY SPECIFIED MEM LOCATION

113F    CD 112A         STEM:   CALL    ADTR            ; ASSEMBLE THE ADDR
1142    ED 53 03F7              LD      (SCA), DE       ; SAVE IT AS CURRENT ADDR
1146    1A                      LD      A, (DE)         ; GET DATA FROM THIS MEM LOC
1147    47                      LD      B, A
1148    E6 0F                   AND     OFH             ; GET LOW HEX DIGIT
114A    21 03F4                 LD      HL, MDB4        ; SET DISP BUF TO DATA LSD
114D    77                      LD      (HL), A         ; PUT THE DIGIT INTO BUF
114E    23                      INC     HL              ; POINT TO MSD
114F    CB 38                   SRL     B               ; SHIFT 4 RIGHT TO GET MSD
1151    CB 38                   SRL     B
1153    CB 38                   SRL     B
1155    CB 38                   SRL     B
1157    70                      LD      (HL), B         ; PUT IT IN BUF
1158    CD 108A                 CALL    DISP            ; DISP & SCAN KBD
115B    FE 10                   CP      GO
115D    F2 117A                 JP      P, EMOP         ; CMD CODE ?
1160    77                      LD      (HL), A         ; NO: PUT NEW DATA MSD INTO BUF
1161    87                      ADD     A, A            ; SHIFT IT 4 LEFT
1162    87                      ADD     A, A
1163    87                      ADD     A, A
1164    87                      ADD     A, A
1165    2B                      DEC     HL              ; POINT TO LSD
1166    47                      LD      B, A            ; SAVE SHIFTED MSD
1167    CD 108A         EM1:    CALL    DISP            ; DISP & SCAN KBD
116A    FE 10                   CP      GO
116C    F2 1174                 JP      P, EMESC        ; CMD CODE ?
116F    77                      LD      (HL), A         ; NO: PUT NEW DATA LSD INTO BUF
1170    B0                      OR      B               ; COMBINE IT WITH MSD
1171    12                      LD      (DE), A         ; PUT DATA BYTE INTO ADDR'D MEM LOC
1172    18 CB                   JR      STEM            ; GO BACK TO RD & DISP THAT LOC AGAIN
1174    FE 19           EMESC:  CP      ESC
1176    28 C7                   JR      Z, STEM         ; CMD IS ESC ?  IGNORE & REPEAT
1178    18 ED                   JR      EM1             ; NO: IGNORE & LOOK FOR ANOTHER
117A    FE 19           EMOP:   CP      ESC
117C    CA 100F                 JP      Z, CTST         ; CMD IS ESC ?  ABORT & BACK TO CMD PROMPT
117F    FE 11                   CP      EM
```

```
1181  28 06                    JR    Z, EM2          ; NO: CMD IS EM ?  EXAM NEXT MEM LOC
1183  FE 12                    CP    EP
1185  28 11                    JR    Z, EM4          ; NO: CMD IS EP ?  EXAM PREV MEM LOC
1187  18 B6                    JR    STEM            ; NO: IGNORE & REPEAT
1189  21 03F0          EM2:    LD    HL, MDB0        ; POINT TO ADDR LSD
118C  3E 0F                    LD    A, 0FH
118E  34               EM3:    INC   (HL)            ; INCR THE ADDR DIGIT
118F  BE                       CP    (HL)
1190  F2 113F                  JP    P, STEM         ; NO OVFLO ?  REPEAT THE ROUTINE
1193  36 00                    LD    (HL), 0         ; OVFLO: SET DIGIT TO 0
1195  23                       INC   HL
1196  18 F6                    JR    EM3             ; GO INCR MSD
1198  21 03F0          EM4:    LD    HL, MDB0        ; POINT TO ADDR LSD
119B  35               EM5:    DEC   (HL)            ; DECR THE DIGIT
119C  F2 113F                  JP    P, STEM         ; NON-0 ?  REPEAT
119F  36 0F                    LD    (HL), 0FH       ; IS 0: SET DIGIT TO 0FH
11A1  23                       INC   HL
11A2  18 F7                    JR    EM5             ; GO DECR MSD

                               ; ROUTINE TO DIS-ASSEMBLE ADDR FROM NUM INTO DISP BUF

11A4  D5               ADIS:   PUSH  DE              ; ADDR IS IN DE, SAVE IT ON STACK
11A5  E5                       PUSH  HL              ; SAVE REG
11A6  7B                       LD    A, E
11A7  E6 0F                    AND   0FH             ; GET ADDR LSD
11A9  21 03F0                  LD    HL, MDB0        ; SET BUF PTR TO LSD
11AC  77                       LD    (HL), A         ; PUT LSD INTO BUF
11AD  23                       INC   HL
11AE  CB 3B                    SRL   E               ; GET 2ND DIGIT
11B0  CB 3B                    SRL   E
11B2  CB 3B                    SRL   E
11B4  CB 3B                    SRL   E
11B6  73                       LD    (HL), E         ; PUT IT INTO BUF
11B7  23                       INC   HL
11B8  7A                       LD    A, D
11B9  E6 0F                    AND   0FH             ; GET 3RD DIGIT
11BB  77                       LD    (HL), A         ; INTO BUF
11BC  23                       INC   HL
11BD  CB 3A                    SRL   D               ; GET MSD
11BF  CB 3A                    SRL   D
11C1  CB 3A                    SRL   D
11C3  CB 3A                    SRL   D
11C5  72                       LD    (HL), D         ; INTO BUF
11C6  E1                       POP   HL              ; RESTORE REG
11C7  D1                       POP   DE              ; RESTORE ADDR IN DE
11C8  C9                       RET

                               ; ROUTINE TO DIS-ASSEMBLE DATA FROM NUM INTO DISP BUF

11C9  E5               DDIS:   PUSH  HL              ; SAVE REG'S
11CA  C5                       PUSH  BC
11CB  1A                       LD    A, (DE)         ; ADDR IS IN DE, GET CORRESP DATA
11CC  47                       LD    B, A
11CD  E6 0F                    AND   0FH             ; GET DATA LSD
11CF  21 03F4                  LD    HL, MDB4        ; SET BUF PTR TO DATA LSD
11D2  77                       LD    (HL), A         ; PUT LSD INTO BUF
11D3  23                       INC   HL
11D4  CB 38                    SRL   B               ; GET MSD
11D6  CB 38                    SRL   B
11D8  CB 38                    SRL   B
11DA  CB 38                    SRL   B
```

```
11DC  70                        LD      (HL), B         ; PUT IT INTO BUF
11DD  C1                        POP     BC
11DE  E1                        POP     HL              ; RESTORE REG'S
11DF  C9                        RET

                                ; ROUTINE TO READ EPROM SOCKET

11E0  11 03F5           RDPGM:  LD      DE, MDB5        ; TO DISP PROMPT IN DATA MSD
11E3  CD 12E2                   CALL    ADCOM           ; DECODE NEXT KEY CMD
11E6  FE 19                     CP      ESC
11E8  CA 100F                   JP      Z, CTST         ; IF ESC, BACK TO CMD PROMPT
11EB  CD 112A                   CALL    ADTR            ; ASSEMBLE DEST'N ADDR
11EE  ED 53 03AE                LD      (DSTADD), DE    ; SAVE IT
11F2  11 03F4                   LD      DE, MDB4        ; TO DISP PROMPT IN DATA LSD
11F5  CD 12E2                   CALL    ADCOM           ; DECODE NEXT KEY CMD & GET DEV NUM
11F8  FE 19                     CP      ESC             ; IF ESC, GO BACK TO PREV
11FA  CA 11E0                   JP      Z, RDPGM                ; (DEST ADDR) LEVEL
11FD  CD 112A                   CALL    ADTR            ; ASSEMBLE DEST ADDR
1200  ED 53 03A8                LD      (DEVICE), DE    ; SAVE IT
1204  CD 1573                   CALL    EVOLT           ; MAKE SURE PROGRAMMING VOLTAGE IS OFF
1207  CD 1232                   CALL    DEVICK          ; CHECK DEVICE NUMBER AND SIZE
120A  CD 1253                   CALL    READ            ; READ THE EPROM
120D  ED 5B 03AE                LD      DE,(DSTADD)     ; DISPLAY THE FIRST LOCATION
1211  CD 11A4                   CALL    ADIS
1214  C3 113F                   JP      STEM

1217  27 08 04          EPTABL: DB      27H,08H,04H     ; THIS TABLE STORES EPROM NAMES AND SIZES
121A  27 16 08                  DB      27H,16H,08H
121D  27 58 08                  DB      27H,58H,08H
1220  27 32 10                  DB      27H,32H,10H
1223  25 32 10                  DB      25H,32H,10H
1226  27 24 10                  DB      27H,24H,10H
1229  27 64 20                  DB      27H,64H,20H
122C  25 64 20                  DB      25H,64H,20H
122F  27 12 40                  DB      27H,12H,40H

1232  06 09             DEVICK: LD      B,09H           ; DEVICK CHECKS EPROMS AGAINST EPTABL
1234  ED 5B 03A8                LD      DE,(DEVICE)
1238  21 1217                   LD      HL,EPTABL
123B  7E                DEV1:   LD      A,(HL)
123C  23                        INC     HL
123D  BA                        CP      D
123E  20 04                     JR      NZ,DEV2
1240  7E                        LD      A,(HL)
1241  BB                        CP      E
1242  28 0A                     JR      Z,DEV3
1244  23                DEV2:   INC     HL
1245  23                        INC     HL
1246  10 F3                     DJNZ    DEV1
1248  CD 10E0                   CALL    CLDSP
124B  C3 1056                   JP      CDER
124E  23                DEV3:   INC     HL
124F  46                        LD      B,(HL)
1250  0E 00                     LD      C,00H
1252  C9                        RET

1253  2A 03AE           READ:   LD      HL,(DSTADD)     ; THIS ROUTINE READS FROM THE SOCKET
1256  11 0000                   LD      DE,0000H
1259  7B                READ1:  LD      A,E
125A  D3 FA                     OUT     (PGAL),A
125C  7A                        LD      A,D
```

```
125D  D3 FB              OUT     (PGAH),A
125F  DB FC              IN      A,(PGD)
1261  77                 LD      (HL),A
1262  13                 INC     DE
1263  ED A1              CPI
1265  EA 1259            JP      PE,READ1
1268  3E 00              LD      A,00H
126A  D3 FB              OUT     (PGAH),A
126C  C9                 RET

                         ; ROUTINE TO COMPARE 2 BLOCKS OF MEM

126D  CD 112A      STCMP: CALL    ADTR            ; ASSEMBLE STARTING ADDR
1270  D5                 PUSH    DE              ; SAVE IT
1271  11 03F5      CMP1:  LD      DE, MDB5        ; TO DISP PROMPT IN DATA MSD
1274  CD 12E2             CALL    ADCOM           ; DECODE NEXT KEY CMD
1277  FE 19               CP      ESC
1279  20 04              JR      NZ, CMP2
127B  D1                 POP     DE              ; IF ESC, RESTORE STACK
127C  C3 100F            JP      CTST            ; GO BACK TO CMD PROMPT
127F  CD 112A      CMP2:  CALL    ADTR            ; ASSEMBLE ENDING ADDR
1282  D5                 PUSH    DE
1283  11 03F4            LD      DE, MDB4        ; TO DISP PROMPT IN DATA LSD
1286  CD 12E2             CALL    ADCOM           ; DECODE NEXT CMD KEY
1289  FE 19               CP      ESC
128B  20 03              JR      NZ, CMP3
128D  D1                 POP     DE              ; IF ESC, GO BACK TO PREV
128E  18 E1              JR      CMP1                    ; (ST ADDR) LEVEL
1290  CD 112A      CMP3:  CALL    ADTR            ; ASSEMBLE DEST ADDR (2ND BLK)
1293  E1                 POP     HL              ; END ADDR
1294  C1                 POP     BC              ; START ADDR
1295  B7                 OR      A
1296  ED 42              SBC     HL, BC          ; BYTE CNT
1298  44                 LD      B, H
1299  4D                 LD      C, L
129A  3B                 DEC     SP
129B  3B                 DEC     SP
129C  E1                 POP     HL              ; START ADDR
129D  03                 INC     BC
129E  1A           CMP4:  LD      A, (DE)         ; DATA FROM 2ND BLK
129F  BE                 CP      (HL)            ; DATA FROM 1ST BLK
12A0  C4 12B0            CALL    NZ, MISM        ; IF MISMATCH
12A3  23                 INC     HL
12A4  13                 INC     DE              ; ON TO NEXT PAIR OF BYTES
12A5  0B                 DEC     BC
12A6  AF                 XOR     A
12A7  B8                 CP      B
12A8  20 F4              JR      NZ,CMP4
12AA  B9                 CP      C
12AB  20 F1              JR      NZ,CMP4
12AD  C3 15E0            JP      MTD             ; BLK'S MATCH: DISP DOUBLE PROMPT

                         ; ROUTINE TO DISP MIS-MATCH LOC'S

12B0  CD 11A4      MISM:  CALL    ADIS            ; DIS-ASSEMBLE ADDR INTO DISP BUF
12B3  CD 11C9             CALL    DDIS            ; DIS-ASSEMBLE DATA
12B6  CD 108A      MSM1:  CALL    DISP            ; DISP THEM
12B9  FE 11               CP      EM              ; EM KEY ?
12BB  28 F3               JR      Z, MISM         ; YES:  DISP DEST BLK DATA
12BD  FE 12               CP      EP              ; EP KEY ?
12BF  28 0D               JR      Z, MSM3         ; YES:  DISP 1ST BLK DATA
```

```
12C1  FE 10                    CP    GO              ; GO KEY ?
12C3  C8                       RET   Z               ; GO GET NEXT MISMATCH PAIR
12C4  FE 19                    CP    ESC             ; ESC KEY ?
12C6  28 02                    JR    Z, MSM2         ; ABORT
12C8  18 EC                    JR    MSM1            ; IGNORE ALL OTHER KEYS
12CA  C1              MSM2:    POP   BC
12CB  C3 100F                  JP    CTST            ; BACK TO CMD PROMPT
12CE  EB              MSM3:    EX    DE, HL
12CF  CD 11A4                  CALL  ADIS            ; DIS-ASSEMBLE 1ST BLK ADDR
12D2  CD 11C9                  CALL  DDIS                    ; & DATA
12D5  EB                       EX    DE, HL
12D6  18 DE                    JR    MSM1            ; GO DISP THEM

                               ; ROUTINE TO RE-EXAM LAST MEM LOC

12D8  ED 5B 03F7      CTEM:    LD    DE, (SCA)       ; GET SAVED ADDR
12DC  CD 11A4                  CALL  ADIS            ; DIS-ASSEMBLE IT
12DF  C3 113F                  JP    STEM            ; GO EXAM IT

                               ; ROUTINE TO GET ADDR & DECODE CMD FROM KBD

12E2  CD 10E0         ADCOM:   CALL  CLDSP           ; CLR DISP BUF
12E5  3E 10                    LD    A, 10H          ; TO DISP PROMPT IN DIGIT
12E7  12                       LD    (DE), A                 ; SPEC'D BY DE
12E8  CD 1102                  CALL  GETAD           ; GET ADDR & CMD FROM KBD
12EB  47                       LD    B, A            ; SAVE THIS CMD
12EC  79                       LD    A, C            ; GET 'DIGIT-ENTERED' CODE
12ED  B7                       OR    A
12EE  78                       LD    A, B            ; CMD
12EF  20 05                    JR    NZ, ADC1        ; DIGIT ENTERED ?
12F1  FE 19                    CP    ESC             ; NO: IF ESC, RET TO PREV LEVEL
12F3  C8                       RET   Z
12F4  18 EC                    JR    ADCOM           ; IGNORE OTHER CMD'S
12F6  FE 19           ADC1:    CP    ESC
12F8  28 E8                    JR    Z, ADCOM        ; IF ESC, RESTART THIS LEVEL
12FA  FE 10           ADC2:    CP    GO
12FC  C8                       RET   Z               ; IF GO, RET (DIGITS IN BUF)
12FD  CD 1122                  CALL  GETA1           ; IF OTHER CMD, IGNORE & WAIT
1300  18 F8                    JR    ADC2                    ; FOR ANOTHER

                               ; ROUTINE TO MOVE A MEM BLK TO ANOTHER POS

1302  CD 112A          STMV:    CALL  ADTR            ; ASSEMBLE START ADDR
1305  D5                       PUSH  DE              ; SAVE IT
1306  D5                       PUSH  DE
1307  11 03F5           MV1:     LD    DE, MDB5        ; TO DISP PROMPT IN DATA MSD
130A  CD 12E2                  CALL  ADCOM           ; DECODE NEXT KEY CMD
130D  FE 19                    CP    ESC
130F  20 05                    JR    NZ, MV2
1311  D1                       POP   DE              ; IF ESC, RESTORE STACK & GO
1312  D1                       POP   DE                      ; BACK TO CMD PROMPT
1313  C3 100F                  JP    CTST
1316  CD 112A           MV2:     CALL  ADTR            ; ASSEMBLE END ADDR
1319  D5                       PUSH  DE
131A  11 03F4                  LD    DE, MDB4        ; TO DISP PROMPT IN DATA LSD
131D  CD 12E2                  CALL  ADCOM           ; DECODE NEXT KEY CMD
1320  FE 19                    CP    ESC
1322  20 03                    JR    NZ, MV3
1324  D1                       POP   DE              ; IF ESC, BACK TO CMD PROMPT
1325  18 E0                    JR    MV1
1327  CD 112A           MV3:     CALL  ADTR            ; ASSEMBLE DEST ADDR
```

```
132A  E1                        POP   HL          ; END ADDR
132B  C1                        POP   BC          ; START ADDR
132C  33                        INC   SP
132D  33                        INC   SP
132E  D5                        PUSH  DE          ; DEST ADDR
132F  B7                        OR    A
1330  ED 42                     SBC   HL, BC      ; BYTE CNT
1332  44                        LD    B, H
1333  4D                        LD    C, L
1334  3B                        DEC   SP
1335  3B                        DEC   SP
1336  E1                        POP   HL          ; START ADDR
1337  B7                        OR    A
1338  ED 52                     SBC   HL, DE      ; IS START ADDR < DEST ADDR ?
133A  38 08                     JR    C, MV4      ; NO
133C  3B                        DEC   SP
133D  3B                        DEC   SP
133E  E1                        POP   HL          ; START ADDR
133F  03                        INC   BC
1340  ED B0                     LDIR              ; BLK MOVE DOWN, BOTTOM BYTE 1ST
1342  18 0A                     JR    MV5
1344  EB                 MV4:   EX    DE, HL
1345  09                        ADD   HL, BC      ; DEST ADDR + BYTE CNT
1346  EB                        EX    DE, HL
1347  3B                        DEC   SP
1348  3B                        DEC   SP
1349  E1                        POP   HL          ; START ADDR
134A  09                        ADD   HL, BC      ; START ADDR + BYTE CNT
134B  03                        INC   BC                ; IE. TOP OF BLK
134C  ED B8                     LDDR              ; BLK MOVE UP, TOP BYTE 1ST
134E  D1                 MV5:   POP   DE          ; DEST ADDR
134F  CD 11A4                   CALL  ADIS        ; DIS-ASSEMBLE IT
1352  C3 113F                   JP    STEM        ; GO EXAM IT

                                ; ROUTINE TO CALC BR INSTR OFFSETS

1355  CD 112A             STOFT: CALL  ADTR        ; ASSEMBLE START ADDR
1358  D5                        PUSH  DE          ; SAVE IT
1359  11 03F5                   LD    DE, MDB5
135C  CD 12E2                   CALL  ADCOM
135F  FE 19                     CP    ESC
1361  20 04                     JR    NZ, OFT4    ; IF ESC, BACK TO CMD PROMPT
1363  D1                        POP   DE
1364  C3 100F                   JP    CTST
1367  CD 112A             OFT4:  CALL  ADTR        ; ASSEMBLE DEST ADDR
136A  CD 10E0                   CALL  CLDSP
136D  62                        LD    H, D
136E  6B                        LD    L, E
136F  C1                        POP   BC          ; START ADDR
1370  B7                        OR    A
1371  ED 42                     SBC   HL, BC      ; DIFFERENCE BYTE CNT
1373  01 0002                   LD    BC, 2
1376  B7                        OR    A
1377  ED 42                     SBC   HL, BC      ; SUBT 2
1379  7C                        LD    A, H
137A  F2 13A2                   JP    P, OFT1
137D  FE FF                     CP    0FFH
137F  20 2A                     JR    NZ, OFER    ; OUT OF RANGE ?
1381  7D                        LD    A, L
1382  B7                        OR    A
1383  F2 13AB                   JP    P, OFER
```

```
1386  7D                OFT2:  LD    A, L
1387  47                       LD    B, A
1388  E6 OF                    AND   OFH             ; GET OFFSET LSD
138A  21 03F4                  LD    HL, MDB4        ; INTO DATA DISP BUF
138D  77                       LD    (HL), A                 ; LSD
138E  23                       INC   HL
138F  CB 38                    SRL   B               ; GET MSD
1391  CB 38                    SRL   B
1393  CB 38                    SRL   B
1395  CB 38                    SRL   B
1397  70                       LD    (HL), B         ; DATA BUF MSD
1398  CD 108A           OFT3:  CALL  DISP            ; DISP IT
139B  FE 19                    CP    ESC
139D  20 F9                    JR    NZ, OFT3        ; WAIT FOR ESC KEY
139F  C3 100F                  JP    CTST
13A2  FE 00               OFT1:  CP    0
13A4  20 05                    JR    NZ, OFER        ; OUT OF RANGE ?
13A6  7D                       LD    A, L
13A7  B7                       OR    A
13A8  F2 1386                  JP    P, OFT2
13AB  3E 12               OFER:  LD    A, 12H          ; DISP TRIPLE BAR ERR IND
13AD  32 03F5                  LD    (MDB5), A
13B0  C3 1017                  JP    CTST1

                               ; ROUTINE TO PROGRAM EPROMS

13B3  CD 112A             STPGM: CALL  ADTR            ; TRANSLATE THE ADDRESS INTO HEX AND
13B6  ED 53 03AC               LD    (STADDR), DE    ; STORE IT AS THE STARTING ADDRESS
13BA  11 03F5             PGM1:  LD    DE, MDB5        ; MOVE THE UNDERSCORE TO THE MS DATA
13BD  CD 12E2                  CALL  ADCOM           ; DIGIT AND WAIT FOR A CMD KEY
13C0  FE 19                    CP    ESC
13C2  CA 100F                  JP    Z, CTST         ; IF IT WASN°T ESC IT MUST BE GO
13C5  CD 112A                  CALL  ADTR            ; TRANSLATE THE ADDRESS INTO HEX AND
13C8  ED 53 03AA               LD    (ENDADD), DE    ; STORE IT AS THE ENDING ADDRESS
13CC  11 03F4             PGM2:  LD    DE, MDB4        ; MOVE THE UNDERSCORE TO THE LS
13CF  CD 12E2                  CALL  ADCOM           ; DATA DIGIT AND WAIT FOR A CMD KEY
13D2  FE 19                    CP    ESC             ; IF ESC WAS PRESSED GO BACK TO THE
13D4  28 E4                    JR    Z, PGM1         ; PREVIOUS COMMAND LEVEL
13D6  CD 112A                  CALL  ADTR            ; IF ESC WASN'T PRESSED GET THE ADDR
13D9  ED 53 03AE               LD    (DSTADD), DE    ; IN THE DISPLAY AND STORE IT IN
13DD  11 03F4             PGM3:  LD    DE, MDB4        ; DESTINATION ADDRESS
13E0  CD 12E2                  CALL  ADCOM           ; WAIT FOR DEV NUM TO BE ENTERED
13E3  FE 19                    CP    ESC             ; FOLLOWED BY A <GO> OR IF <ESC> IS
13E5  28 E5                    JR    Z, PGM2         ; PRESSED GO BACK TO THE PREVIOUS
13E7  CD 112A                  CALL  ADTR            ; COMMAND LEVEL
13EA  ED 53 03A8               LD    (DEVICE), DE
                               ;
                               ; THIS NEXT SECTION CALCULATES THE TOTAL NUMBER OF BYTES TO
                               ; BE PROGRAMMED AND MAKES SURE IT IS POSITIVE, IE. THE USER
                               ; ENTERED THE STARTING ADDRESS BEFORE ENTERING THE ENDING
                               ; ADDRESS.
                               ;
13EE  2A 03AA                  LD    HL, (ENDADD)
13F1  ED 4B 03AC               LD    BC, (STADDR)
13F5  B7                       OR    A               ; CLEAR CARRY FLAG
13F6  ED 42                    SBC   HL, BC
13F8  23                       INC   HL              ; ADD 1 TO HL SO THAT THE BYTE
13F9  22 03B0                  LD    (BYTCNT), HL    ; COUNT IS THE DIFFERENCE BETWEEN
13FC  3A 03B1                  LD    A, (BYTCNT+1)   ; ENDADDR AND STADDR INCLUSIVE.
13FF  B7                       OR    A               ; SET FLAGS
1400  F5                       PUSH  AF
```

```
1401  CD 10E0                   CALL   CLDSP
1404  F1                        POP    AF
                          ;
                          ; THIS NEXT SECTION IS A CHECKING ROUTINE THAT WILL ENSURE THAT
                          ; THE EPROM IS CLEAN. IF THE EPROM HAS BEEN PREVIOUSLY PROGRAMMED
                          ; OR IT IS BAD, THE CHECKING ROUTINE WILL DISPLAY THOSE LOCATIONS
                          ; IN THE EPROM THAT ARE INCORRECT. NOTE THAT IN A CLEAN EPROM
                          ; ALL LOCATIONS WILL BE 'FF'. CARE MUST BE TAKEN IF THE USER WISHES
                          ; TO PROGRAM A PREVIOUSLY PROGRAMMED LOCATION, IN THAT A BIT MAY
                          ; BE CHANGED FROM A 1 TO A 0 BUT NOT VISE VERSA.
                          ;
1405  FA 1056                   JP     M, CDER          ; BYTE COUNT MUST BE POSITIVE
1408  ED 4B 03B0                LD     BC, (BYTCNT)
140C  2A 03AE                   LD     HL, (DSTADD)
140F  ED 5B 03AA                LD     DE, (ENDADD)
                          ;
                          ; THIS NEXT SECTION IS THE DECODER, ITS PURPOSE IS TO ENSURE
                          ; THAT ONLY THE SPECIFIED EPROMS ARE ENTERED AS THE DEVICE
                          ; TO BE PROGRAMMED. IF IT IS FOUND THAT AN ILLEGAL DEVICE NUMBER
                          ; HAS BEEN ENTERED, THEN IT WILL JUMP TO A ROUTINE (CDER) WHICH
                          ; WILL GIVE AN ERROR PROMPT, OTHERWISE IT WILL BRANCH TO THE
                          ; APPROPRIATE PROGRAMMING ROUTINE.
                          ;
1413  CD 148E                   CALL   CHECK
1416  CD 1232                   CALL   DEVICK
1419  ED 5B 03B0                LD     DE,(BYTCNT)
141D  60                        LD     H,B
141E  69                        LD     L,C
141F  B7                        OR     A
1420  ED 52                     SBC    HL,DE
1422  FA 1056                   JP     M,CDER
1425  3E 04                     LD     A,04H
1427  B8                        CP     B
1428  CA 14E2                   JP     Z,PG2708
142B  CD 1431                   CALL   PROG             ; PROGRAM THE DEVICE
142E  C3 14BE                   JP     VERIFY           ; VERIFY THE DATA


1431  ED 5B 03AE          PROG:  LD     DE,(DSTADD)      ; THIS IS THE MAIN PROGRAMMING ROUTINE
1435  CB F2                     SET    6,D              ; FOR ALL BUT 2708S
1437  CB BA                     RES    7,D
1439  7A                        LD     A,D
143A  D3 FB                     OUT    (PGAH),A         ; TURN ON THE PROGRAMMING VOLTAGE
143C  3E 00                     LD     A,00H
143E  CD 1470                   CALL   DELAYB           ; WAIT FOR IT TO STABILIZE
1441  2A 03AC                   LD     HL,(STADDR)
1444  ED 4B 03B0                LD     BC,(BYTCNT)

1448  7B                  PROG1: LD     A,E
1449  D3 FA                     OUT    (PGAL),A
144B  7E                        LD     A,(HL)
144C  D3 FC                     OUT    (PGD),A
144E  7A                        LD     A,D
144F  D3 FB                     OUT    (PGAH),A
1451  CD 1597                   CALL   DLY10
1454  CB FF                     SET    7,A
1456  D3 FB                     OUT    (PGAH),A
1458  CD 1482                   CALL   DELAY
145B  7A                        LD     A,D
145C  D3 FB                     OUT    (PGAH),A
```

```
145E  13                      INC     DE
145F  ED A1                   CPI
1461  EA 1448                 JP      PE,PROG1
1464  CB B2                   RES     6,D
1466  3E 00                   LD      A,00H
1468  D3 FB                   OUT     (PGAH),A
146A  3E 0C                   LD      A,0CH
146C  CD 1470                 CALL    DELAYB
146F  C9                      RET

1470  21 0000           DELAYB: LD      HL,0000H        ; THIS ROUTINE WAITS FOR THE VOLTAGE TO
1473  06 00             DB1:    LD      B,00H           ; STABILIZE. IF A IS GREATER THAN 0, IT
1475  04                DB2:    INC     B               ; WILL JUMP TO EVOLT AFTER THE WAIT TO
1476  20 FD                     JR      NZ,DB2          ; ACTUALLY CHECK THE VOLTAGE. THE DELAY
1478  23                        INC     HL              ; IS PROPORTIONATE TO A.
1479  BC                        CP      H
147A  20 F7                     JR      NZ,DB1
147C  B7                        OR      A
147D  C8                        RET     Z
147E  CD 1573                   CALL    EVOLT
1481  C9                        RET

                          ;
                          ; THIS PROVIDES APROXIMATELY 50 MS. DELAY BEFORE RETURNING.
                          ;
1482  C5                DELAY:  PUSH    BC
1483  06 32                     LD      B, 32H
1485  0E 00             LOOP:   LD      C, 0
1487  0D                LOOP1:  DEC     C
1488  20 FD                     JR      NZ, LOOP1
148A  10 F9                     DJNZ    LOOP
148C  C1                        POP     BC
148D  C9                        RET
                          ;
                          ; CHECK IS A SUBROUTINE MAKES SURE THE EPROM IS CLEAN, IE.
                          ; EACH LOCATION IS A 'FF'.
                          ;
148E  ED 4B 03B0          CHECK:  LD      BC, (BYTCNT)
1492  7D                  CHECK1: LD      A, L
1493  D3 FA                     OUT     (PGAL), A
1495  7C                        LD      A, H
1496  D3 FB                     OUT     (PGAH), A
1498  DB FC                     IN      A, (PGD)
149A  FE FF                     CP      0FFH
149C  C2 14A5                   JP      NZ, NCLR
149F  ED A1                     CPI
14A1  EA 1492                   JP      PE, CHECK1
14A4  C9                        RET

14A5  E5                  NCLR:   PUSH    HL              ; THIS ROUTINE DISPLAYS ANY NON-FF LOCATIONS
14A6  EB                        EX      DE, HL
14A7  CD 11A4                   CALL    ADIS
14AA  CD 108A             NCLR1:  CALL    DISP
14AD  FE 19                     CP      ESC
14AF  CA 14B8                   JP      Z, NCLR2
14B2  FE 10                     CP      GO
14B4  E1                        POP     HL
14B5  C8                        RET     Z
14B6  18 F2                     JR      NCLR1
14B8  33                  NCLR2:  INC     SP
14B9  33                        INC     SP              ; FIX UP STACK
```

```
14BA   E1                        POP     HL
14BB   C3 100F                   JP      CTST            ; BACK TO CMD PROMPT

14BE   2A 03AC            VERIFY: LD      HL, (STADDR)    ; THIS ROUTINE VERIFIES THAT
14C1   ED 4B 03B0                 LD      BC, (BYTCNT)    ; THE DATA BURNT INTO THE EPROM
14C5   ED 5B 03AE                 LD      DE, (DSTADD)    ; IS CORRECT.
14C9   7A                 PGM14:  LD      A, D
14CA   D3 FB                      OUT     (PGAH), A
14CC   7B                         LD      A, E
14CD   D3 FA                      OUT     (PGAL), A
14CF   DB FC                      IN      A, (PGD)
14D1   BE                         CP      (HL)
14D2   C4 1598                    CALL    NZ, EPMISM
14D5   13                         INC     DE
14D6   ED A1                      CPI
14D8   EA 14C9                    JP      PE, PGM14
14DB   3E 00                      LD      A,00H
14DD   D3 FB                      OUT     (PGAH),A
14DF   C3 15E0                    JP      MTD
                          ;
                          ; THIS IS A SPECIAL ROUTINE WHICH WILL PROGRAM A 2708.
                          ; IT IS COMPLETELY SELF CONTAINED, EXCEPT FOR THE ACCESS
                          ; TO THE GLOBAL PARAMETER STADDR, AND THE GLOBAL ROUTINE
                          ; EPROM MISMATCH (EPMISM).
                          ;
14E2   2A 03AC            PG2708:LD       HL, (STADDR)
14E5   AF                         XOR     A
14E6   5F                         LD      E, A
14E7   57                         LD      D, A

14E8   D3 FB              PGM16:  OUT     (PGAH), A       ; CHECK TO MAKE SURE EPROM IS BLANK
14EA   7B                 PGM17:  LD      A, E
14EB   D3 FA                      OUT     (PGAL), A
14ED   DB FC                      IN      A, (PGD)
14EF   FE FF                      CP      0FFH
14F1   20 6F                      JR      NZ, PGM24       ; DISPLAY ANY NON-FF LOCATIONS
14F3   1C                         INC     E
14F4   20 F4                      JR      NZ, PGM17
14F6   14                         INC     D
14F7   7A                         LD      A, D
14F8   FE 04                      CP      4
14FA   20 EC                      JR      NZ, PGM16

14FC   3E 40                      LD      A,40H           ; TURN ON PROGRAMMING VOLTAGE
14FE   D3 FB                      OUT     (PGAH),A
1500   3E 00                      LD      A,00H
1502   CD 1470                    CALL    DELAYB          ; WAIT FOR VOLTAGE TO STABILIZE

1505   1E 00              PGM26:  LD      E, 0
1507   0E 64                      LD      C, 64H          ; DO 100 PROGRAMMING LOOPS
1509   2A 03AC                    LD      HL, (STADDR)
150C   16 80              PGM18:  LD      D, 80H          ; SET CE = 12V.
150E   7A                 PGM19:  LD      A, D
150F   D3 FB                      OUT     (PGAH), A
1511   7B                 PGM20:  LD      A, E
1512   D3 FA                      OUT     (PGAL), A
1514   7E                         LD      A, (HL)
1515   D3 FC                      OUT     (PGD), A
1517   23                         INC     HL
1518   7A                         LD      A, D
1519   F6 40                      OR      40H             ; SET PROG = 25V.
```

```
151B  CD 1597                    CALL   DLY10
151E  D3 FB                      OUT    (PGAH), A
1520  06 FF                      LD     B, OFFH
1522  05                 PGM21:  DEC    B
1523  20 FD                      JR     NZ, PGM21
1525  E6 83                      AND    83H             ; RESET PROG = 0.
1527  D3 FB                      OUT    (PGAH), A
1529  1C                         INC    E
152A  20 E5                      JR     NZ, PGM20
152C  7A                         LD     A, D
152D  E6 03                      AND    03H
152F  3C                         INC    A
1530  14                         INC    D
1531  FE 04                      CP     04H
1533  20 D9                      JR     NZ, PGM19
1535  2A 03AC                    LD     HL, (STADDR)
1538  0D                         DEC    C
1539  20 D1                      JR     NZ, PGM18

153B  3E 03                      LD     A,3             ; TURN OFF  CE=0V.
153D  D3 FB                      OUT    (PGAH),A
153F  3E 0C                      LD     A, 0CH          ; TURN OF +25V.
1541  CD 1470                    CALL   DELAYB

1544  AF                         XOR    A               ; VERIFY DATA BURNT INTO EPROM
1545  2A 03AC                    LD     HL,(STADDR)
1548  5F                         LD     E, A
1549  57                         LD     D, A
154A  D3 FB              PGM22:  OUT    (PGAH), A
154C  7B                 PGM23:  LD     A, E
154D  D3 FA                      OUT    (PGAL), A
154F  DB FC                      IN     A, (PGD)
1551  BE                         CP     (HL)
1552  C4 1598                    CALL   NZ, EPMISM
1555  23                         INC    HL
1556  1C                         INC    E
1557  20 F3                      JR     NZ, PGM23
1559  14                         INC    D
155A  7A                         LD     A, D
155B  FE 04                      CP     4
155D  20 EB                      JR     NZ, PGM22
155F  C3 15E0                    JP     MTD

1562  CD 11A4            PGM24:  CALL   ADIS            ; DISPLAYS ANY NON-FF LOCATIONS
1565  CD 108A            PGM25:  CALL   DISP
1568  FE 19                      CP     ESC
156A  CA 100F                    JP     Z, CTST
156D  FE 10                      CP     GO
156F  28 94                      JR     Z, PGM26
1571  18 F2                      JR     PGM25

1573  21 03F0            EVOLT:  LD     HL, MDB0        ; THIS ROUTINE PUTS THE
1576  3E 1B                      LD     A, 1BH          ; MESSAGE S3=RD, ON THE DISPLAY
1578  77                         LD     (HL), A         ; SO THE USER CAN DISCONNECT
1579  23                         INC    HL              ; THE HIGH VOLTAGE BEFORE
157A  3E 11                      LD     A, 11H          ; READING THE EPROM
157C  77                         LD     (HL), A
157D  23                         INC    HL
157E  3E 03                      LD     A, 03
1580  77                         LD     (HL), A
1581  23                         INC    HL
```

```
1582  3E 05                    LD      A, 05
1584  77                       LD      (HL), A
1585  23                       INC     HL
1586  3E 0D                    LD      A, ODH
1588  77                       LD      (HL), A
1589  23                       INC     HL
158A  3E 19                    LD      A, 19H
158C  77                       LD      (HL), A
158D  DB FA                    IN      A,(PGAL)
158F  CB 77                    BIT     6,A
1591  C8                       RET     Z
1592  CD 108A                  CALL    DISP
1595  18 DC                    JR      EVOLT

1597  C9               DLY10:  RET
1598  CD 11A4          EPMISM: CALL    ADIS            ; THIS ROUTINE DISPLAYS THE
159B  CD 15C0                  CALL    EPDIS           ; INCORRECT CONTENTS OF THE
159E  CD 108A          EMSM1:  CALL    DISP            ; EPROM OR ITS CORRECT
15A1  FE 11                    CP      EM              ; COUNTER-PART IN MEMORY, THE
15A3  28 F3                    JR      Z, EPMISM       ; ADDR OF THE INCORRECT LOCATION
15A5  FE 12                    CP      EP              ; IN EPROM IS IN DE, THE
15A7  28 0D                    JR      Z, EMSM3        ; CORESPONDING ADDRESS OF THE
15A9  FE 10                    CP      GO              ; CORRECT DATA IN MEMORY IS IN HL
15AB  C8                       RET     Z
15AC  FE 19                    CP      ESC
15AE  28 02                    JR      Z, EMSM2
15B0  18 EC                    JR      EMSM1
15B2  C1               EMSM2:  POP     BC              ; FIX UP STACK
15B3  C3 100F                  JP      CTST            ; BACK TO CMD PROMPT
15B6  EB               EMSM3:  EX      DE, HL
15B7  CD 11A4                  CALL    ADIS
15BA  CD 11C9                  CALL    DDIS
15BD  EB                       EX      DE, HL
15BE  18 DE                    JR      EMSM1

15C0  E5               EPDIS:  PUSH    HL              ; THIS ROUTINE GETS THE CONTENTS
15C1  C5                       PUSH    BC              ; OF THE ADDRESS IN DE FROM THE
15C2  7A                       LD      A, D            ; EPROM AND PUTS IT IN THE
15C3  E6 0F                    AND     0FH             ; DISPLAY BUFFER.
15C5  D3 FB                    OUT     (PGAH), A
15C7  7B                       LD      A, E
15C8  D3 FA                    OUT     (PGAL), A
15CA  DB FC                    IN      A, (PGD)
15CC  47                       LD      B, A            ; SAVE DATA
15CD  E6 0F                    AND     0FH             ; REMOVE UPPER NIBBLE AND STORE
15CF  21 03F4                  LD      HL, MDB4        ; IT IN LS DISPLAY DIGIT
15D2  77                       LD      (HL), A
15D3  23                       INC     HL
15D4  CB 38                    SRL     B               ; GET MS NIBBLE OF DATA AND STORE
15D6  CB 38                    SRL     B               ; IT IN MS DATA DISPLAY DIGIT
15D8  CB 38                    SRL     B
15DA  CB 38                    SRL     B
15DC  70                       LD      (HL), B
15DD  C1                       POP     BC
15DE  E1                       POP     HL
15DF  C9                       RET

                               ; ROUTINE TO DISP DOUBLE PROMPT

15E0  CD 10E0          MTD:    CALL    CLDSP
15E3  3E 10                    LD      A, 10H
```

```
15E5  32 03F0                    LD      (MDB0), A
15E8  32 03F1                    LD      (MDB1), A
15EB  CD 108A             MTD1:  CALL    DISP
15EE  FE 19                      CP      ESC             ; WAIT FOR ESC KEY
15F0  CA 100F                    JP      Z, CTST
15F3  18 F6                      JR      MTD1            ; IGNORE ALL OTHERS

                                 ; ROUTINE TO READ & WRITE I/O PORTS

15F5  CD 10E0             STEIO: CALL    CLDSP
15F8  3E 10                      LD      A, 10H          ; DISP 2 PROMPTS IN
15FA  21 03F3                    LD      HL, MDB3                ; ADDR MSD'S
15FD  77                         LD      (HL), A
15FE  2B                         DEC     HL
15FF  77                         LD      (HL), A
1600  2B                         DEC     HL
1601  CD 108A                    CALL    DISP
1604  FE 19                      CP      ESC
1606  CA 100F                    JP      Z, CTST         ; IF ESC, ABORT
1609  FE 10                      CP      GO
160B  F2 15F5                    JP      P, STEIO        ; IGNORE ALL OTHER CMD'S
160E  77                         LD      (HL), A         ; GOT I/O PORT MSD
160F  2B                         DEC     HL
1610  CD 108A             EIO1:  CALL    DISP            ; DISP IT
1613  FE 19                      CP      ESC
1615  28 DE                      JR      Z, STEIO        ; IF ESC, RESTART THIS
1617  FE 10                      CP      GO
1619  F2 1610                    JP      P, EIO1         ; IGNORE ALL OTHER CMDS
161C  77                         LD      (HL), A         ; GOT I/O PORT LSD
161D  21 03F1             EIO4:  LD      HL, MDB1
1620  7E                         LD      A, (HL)         ; PORT ADDR MSD
1621  87                         ADD     A, A            ; ASSEMBLE THE PORT ADDR
1622  87                         ADD     A, A
1623  87                         ADD     A, A
1624  87                         ADD     A, A
1625  4F                         LD      C, A
1626  2B                         DEC     HL
1627  7E                         LD      A, (HL)         ; LSD
1628  B1                         OR      C
1629  4F                         LD      C, A
162A  21 03F5                    LD      HL, MDB5
162D  3E 10                      LD      A, 10H
162F  77                         LD      (HL), A         ; DISP PROMPT IN DATA MSD
1630  2B                         DEC     HL
1631  3E 11                      LD      A, 11H          ; BLANK DATA LSD
1633  77                         LD      (HL), A
1634  23                 EIO8:   INC     HL
1635  CD 108A            EIO2:   CALL    DISP            ; DISP IT
1638  FE 19                      CP      ESC
163A  28 B9                      JR      Z, STEIO        ; IF ESC, RESTART THIS ROUTINE
163C  FE 1D                      CP      EIO
163E  28 26                      JR      Z, EIO5         ; IF EIO, GO INCR PORT ADDR
1640  FE 12                      CP      EP
1642  28 38                      JR      Z, EIO6         ; IF EP, GO DECR PORT ADDR
1644  FE 13                      CP      ER
1646  28 46                      JR      Z, EIO7         ; IF ER, READ THE PORT
1648  FE 10                      CP      GO
164A  F2 1635                    JP      P, EIO2         ; IGNORE ALL OTHER CMD'S
164D  77                         LD      (HL), A         ; DATA KEY, MSD
164E  87                         ADD     A, A
164F  87                         ADD     A, A            ; ASSEMBLE IT
```

```
1650  87                        ADD   A, A
1651  87                        ADD   A, A
1652  47                        LD    B, A
1653  2B                        DEC   HL
1654  CD 108A           EIO3:   CALL  DISP          ; DISP IT
1657  FE 19                     CP    ESC
1659  28 C2                     JR    Z, EIO4       ; IF ESC, CLR FOR RETRY
165B  FE 10                     CP    GO
165D  F2 1654                   JP    P, EIO3       ; IGNORE OTHER CMD'S
1660  77                        LD    (HL), A       ; DATA KEY, LSD
1661  B0                        OR    B             ; ASSEMBLE IT
1662  ED 79                     OUT   (C), A        ; WRITE IT TO THE I/O PORT
1664  18 CE                     JR    EIO8          ; WAIT FOR NEXT CMD
1666  21 03F0           EIO5:   LD    HL, MDB0
1669  3E 0F                     LD    A, 0FH
166B  34                        INC   (HL)          ; INCR PORT ADDR
166C  BE                        CP    (HL)
166D  F2 161D                   JP    P, EIO4
1670  36 00                     LD    (HL), 0       ; MODULO 256
1672  23                        INC   HL
1673  34                        INC   (HL)
1674  BE                        CP    (HL)
1675  F2 161D                   JP    P, EIO4
1678  36 00                     LD    (HL), 0
167A  18 A1                     JR    EIO4
167C  21 03F0           EIO6:   LD    HL, MDB0
167F  35                        DEC   (HL)          ; DECR PORT ADDR
1680  F2 161D                   JP    P, EIO4
1683  36 0F                     LD    (HL), 0FH     ; MODULO 256
1685  23                        INC   HL
1686  35                        DEC   (HL)
1687  F2 161D                   JP    P, EIO4
168A  36 0F                     LD    (HL), 0FH
168C  18 8F                     JR    EIO4
168E  ED 78             EIO7:   IN    A, (C)        ; READ THE PORT
1690  47                        LD    B, A
1691  E6 0F                     AND   0FH
1693  21 03F4                   LD    HL, MDB4      ; DIS-ASSEMBLE THE DATA
1696  77                        LD    (HL), A             ; INTO DISP BUF
1697  23                        INC   HL
1698  CB 38                     SRL   B
169A  CB 38                     SRL   B
169C  CB 38                     SRL   B
169E  CB 38                     SRL   B
16A0  70                        LD    (HL), B
16A1  C3 1635                   JP    EIO2          ; DISP IT

                                ; LED SEGMENT DEFINITION TABLE FOR CHARACTERS DISP'D

16A4  40                LSGTB:  DEFB  40H     ; 0
16A5  79                        DEFB  79H     ; 1
16A6  24                        DEFB  24H     ; 2
16A7  30                        DEFB  30H     ; 3
16A8  19                        DEFB  19H     ; 4
16A9  12                        DEFB  12H     ; 5
16AA  02                        DEFB  02H     ; 6
16AB  78                        DEFB  78H     ; 7
16AC  00                        DEFB  00H     ; 8
16AD  10                        DEFB  10H     ; 9
16AE  08                        DEFB  08H     ; A
16AF  03                        DEFB  03H     ; B
```

```
16B0  46                        DEFB   46H     ; C
16B1  21                        DEFB   21H     ; D
16B2  06                        DEFB   06H     ; E
16B3  0E                        DEFB   0EH     ; F
16B4  77                        DEFB   77H     ; 10 (_)
16B5  7F                        DEFB   7FH     ; 11 (¯)
16B6  36                        DEFB   36H     ; 12 ( )
16B7  09                        DEFB   09H     ; 13 (H)
16B8  47                        DEFB   47H     ; 14 (L)
16B9  0C                        DEFB   0CH     ; 15 (P)
16BA  5F                        DEFB   5FH     ; 16 (')
16BB  1B                        DEFB   1BH     ; 17 (X)
16BC  11                        DEFB   11H     ; 18 (Y)
16BD  2F                        DEFB   2FH     ; 19 (R)
16BE  0F                        DEFB   0FH     ; 1A ( )
16BF  37                        DEFB   37H     ; IB (=)

                                ; KEY CODE TABLE

16C0  00                KEYCD:  DEFB   00H     ; 0
16C1  01                        DEFB   01H     ; 1
16C2  02                        DEFB   02H     ; 2
16C3  03                        DEFB   03H     ; 3
16C4  11                        DEFB   11H     ; EM
16C5  10                        DEFB   10H     ; GO
16C6  15                        DEFB   15H     ; SI
16C7  19                        DEFB   19H     ; ESC
16C8  04                        DEFB   04H     ; 4
16C9  05                        DEFB   05H     ; 5
16CA  06                        DEFB   06H     ; 6
16CB  07                        DEFB   07H     ; 7
16CC  12                        DEFB   12H     ; EP
16CD  13                        DEFB   13H     ; ER
16CE  17                        DEFB   17H     ; BP
16CF  1E                        DEFB   1EH     ; DOWNLOAD
16D0  08                        DEFB   08H     ; 8
16D1  09                        DEFB   09H     ; 9
16D2  0A                        DEFB   0AH     ; A
16D3  0B                        DEFB   0BH     ; B
16D4  1D                        DEFB   1DH     ; EIO
16D5  1B                        DEFB   1BH     ; CMP
16D6  1A                        DEFB   1AH     ; PGM
16D7  1F                        DEFB   1FH     ; CP/M BOOT
16D8  0C                        DEFB   0CH     ; C
16D9  0D                        DEFB   0DH     ; D
16DA  0E                        DEFB   0EH     ; E
16DB  0F                        DEFB   0FH     ; F
16DC  14                        DEFB   14H     ; MV
16DD  1C                        DEFB   1CH     ; OFT
16DE  16                        DEFB   16H     ; LD
16DF  18                        DEFB   18H     ; SV

                                ; ROUTINE TO CREATE A BREAK-POINT

16E0  AF                STBP:   XOR    A
16E1  32 03FA                   LD     (CBPNM), A      ; CURRENT BRK-PT NUM = 0
16E4  CD 10E0           BP0:    CALL   CLDSP
16E7  3E 10             BP1:    LD     A, 10H          ; DISP PROMPT IN DATA MSD
16E9  32 03F5                   LD     (MDB5), A
16EC  3A 03FA           BPE:    LD     A, (CBPNM)
16EF  32 03F4                   LD     (MDB4), A       ; DISP CUR BRK-PT NUM
```

```
16F2  CB 27                    SLA   A
16F4  CB 27                    SLA   A              ; MULT BY 8
16F6  CB 27                    SLA   A
16F8  FD 21 03D0               LD    IY, BPBA       ; BRK-PT BASE ADDR
16FC  16 00                    LD    D, 0
16FE  5F                       LD    E, A
16FF  FD 19                    ADD   IY, DE         ; PTR TO CUR BRK-PT AREA
1701  FD 7E 03                 LD    A, (IY+3)      ; BRK-PT-DEFN'D FLAG
1704  B7                       OR    A
1705  28 2B                    JR    Z, BP3         ; BRK-PT DEFINED ?
1707  FD 56 01                 LD    D, (IY+1)      ; YES: GET ITS ADDR
170A  FD 5E 00                 LD    E, (IY+0)
170D  CD 11A4                  CALL  ADIS           ; DIS-ASSEMBLE IT
1710  CD 1102          BP2:    CALL  GETAD          ; DISP IT & WAIT FOR CMD
1713  47                       LD    B, A
1714  79                       LD    A, C
1715  B7                       OR    A
1716  78                       LD    A, B
1717  20 1E                    JR    NZ, BP4        ; ANY DIGITS ENTERED ?
1719  FE 19                    CP    ESC            ; YES
171B  CA 100F                  JP    Z, CTST        ; IF ESC, ABORT
171E  FE 17                    CP    BP
1720  CA 17CC                  JP    Z, NBP         ; IF BP, EXAM NEXT BRK-PT
1723  FE 12                    CP    EP
1725  CA 17DF                  JP    Z, PBP         ; IF EP, EXAM PREV BRK-PT
1728  FE 10                    CP    GO
172A  CA 1754                  JP    Z, CLBP        ; IF GO, CLR THIS BRK-PT
172D  FE 13                    CP    ER
172F  CA 177E                  JP    Z, EDLN        ; IF ER, EXAM ITS DELAY NUM
1732  CD 17F3          BP3:    CALL  CLDSA          ; CLR DISP ADDR DIGITS
1735  18 D9                    JR    BP2            ; WAIT FOR CMD
1737  FE 19            BP4:    CP    ESC
1739  28 F7                    JR    Z, BP3         ; IF ESC, RESTART
173B  FE 10            BP5:    CP    GO
173D  28 05                    JR    Z, BP6         ; IF GO, ENTER THE BRK-PT ADDR
173F  CD 1122                  CALL  GETA1                ; INTO TABLE
1742  18 F7                    JR    BP5            ; IGNORE OTHER CMD'S
1744  CD 112A          BP6:    CALL  ADTR           ; ASSEMBLE THE BRK-PT ADDR
1747  FD 72 01                 LD    (IY+1), D      ; STORE IT IN TABLE
174A  FD 73 00                 LD    (IY+0), E
174D  3E FF                    LD    A, OFFH
174F  FD 77 03                 LD    (IY+3), A      ; SET 'DEFINED' FLG
1752  18 BC                    JR    BP2            ; GET NEXT BRK-PT
1754  FD 7E 03          CLBP:   LD    A, (IY+3)      ; BRK-PT DEFINED FLG
1757  B7                       OR    A
1758  28 D8                    JR    Z, BP3         ; IF NOT DEFN'D, CLR DISP
175A  AF                       XOR   A
175B  FD 77 03                 LD    (IY+3), A      ; ELSE, CLR THE BRK-PT
175E  FD 6E 00                 LD    L, (IY+0)      ; BRK-PT ADDR
1761  FD 66 01                 LD    H, (IY+1)
1764  FD 7E 02                 LD    A, (IY+2)      ; SAVED PROG OP-CODE AT ADDR
1767  77                       LD    (HL), A        ; RESTORE IT
1768  AF                       XOR   A
1769  FD 77 00                 LD    (IY+0), A      ; CLR OUT TBL ENTRY
176C  FD 77 01                 LD    (IY+1), A
176F  FD 77 04                 LD    (IY+4), A
1772  FD 77 05                 LD    (IY+5), A
1775  FD 77 06                 LD    (IY+6), A
1778  FD 77 07                 LD    (IY+7), A
177B  C3 1732                  JP    BP3            ; CLR DISP
177E  CD 10E0            EDLN:  CALL  CLDSP
```

```
1781  3E 1A                    LD    A, 1AH          ; BRK-PT DELAY SYMBOL
1783  32 03F5                  LD    (MDB5), A       ; DISP IT
1786  3A 03FA                  LD    A, (CBPNM)      ; CUR BRK-PT NUM
1789  32 03F4                  LD    (MDB4), A       ; DISP IT
178C  FD 56 05                 LD    D, (IY+5)
178F  FD 5E 04                 LD    E, (IY+4)       ; BRK-PT DELAY NUM
1792  CD 11A4                  CALL  ADIS            ; DIS-ASSEMBLE IT
1795  CD 1102          EDL1:   CALL  GETAD           ; DISP IT & GET CMD
1798  47                       LD    B, A
1799  79                       LD    A, C
179A  B7                       OR    A
179B  78                       LD    A, B
179C  20 15                    JR    NZ, EDL3        ; ANY DIGITS ENTERED ?
179E  FE 19                    CP    ESC             ; NO
17A0  20 0C                    JR    NZ, EDL2
17A2  CD 10E0                  CALL  CLDSP           ; IF ESC, RESTART
17A5  3A 03FA                  LD    A, (CBPNM)      ; CUR BRK-PT NUM
17A8  32 03F4                  LD    (MDB4), A       ; DISP IT
17AB  C3 16E7                  JP    BP1
17AE  CD 17F3          EDL2:   CALL  CLDSA           ; CLR DISP ADDR DIGITS
17B1  18 E2                    JR    EDL1            ; RESTART DELAY NUM EXAM
17B3  FE 19            EDL3:   CP    ESC
17B5  28 F7                    JR    Z, EDL2         ; IF ESC, GO BACK & EXAM
17B7  FE 10            EDL4:   CP    GO
17B9  28 05                    JR    Z, EDL5         ; IF GO, ENTER IT
17BB  CD 1122                  CALL  GETA1           ; IGNORE OTHER CMD'S
17BE  18 F7                    JR    EDL4
17C0  CD 112A          EDL5:   CALL  ADTR            ; ASSEMBLE THE DELAY NUM
17C3  FD 72 05                 LD    (IY+5), D
17C6  FD 73 04                 LD    (IY+4), E       ; STORE IT IN TBL
17C9  C3 16E4                  JP    BP0             ; RESTART BRK-PT EXAM
17CC  3A 03FA          NBP:    LD    A, (CBPNM)      ; CUR BRK-PT NUM
17CF  3C                       INC   A
17D0  32 03FA                  LD    (CBPNM), A      ; NEXT BRK-PT NUM
17D3  FE 04                    CP    4
17D5  C2 16EC                  JP    NZ, BPE         ; MODULO 4
17D8  AF                       XOR   A
17D9  32 03FA                  LD    (CBPNM), A
17DC  C3 16EC                  JP    BPE
17DF  3A 03FA          PBP:    LD    A, (CBPNM)      ; CUR BRK-PT NUM
17E2  3D                       DEC   A
17E3  32 03FA                  LD    (CBPNM), A      ; PREV BRK-PT NUM
17E6  FE FF                    CP    0FFH
17E8  C2 16EC                  JP    NZ, BPE         ; MODULO 4
17EB  3E 03                    LD    A, 3
17ED  32 03FA                  LD    (CBPNM), A
17F0  C3 16EC                  JP    BPE

                               ; ROUTINE TO CLEAR DISP ADDR DIGITS

17F3  3E 11            CLDSA:  LD    A, 11H          ; BLANK
17F5  06 04                    LD    B, 4
17F7  21 03F0                  LD    HL, MDB0        ; ADDR LSD
17FA  77               CLDA1:  LD    (HL), A         ; CLR IT
17FB  23                       INC   HL              ; NEXT DIGIT
17FC  10 FC                    DJNZ  CLDA1           ; LAST DIGIT ?
17FE  C9                       RET

                               ; ROUTINE TO SAVE REGISTERS OF PROGRAM AT INTERRUPT PT

17FF  31 03CE          SVREG:  LD    SP, SREG        ; START OF REG SAVE SPACE
```

```
1802  F5                        PUSH    AF
1803  C5                        PUSH    BC
1804  D5                        PUSH    DE
1805  E5                        PUSH    HL
1806  D9                        EXX
1807  08                        EX      AF, AF'
1808  F5                        PUSH    AF
1809  C5                        PUSH    BC
180A  D5                        PUSH    DE
180B  E5                        PUSH    HL
180C  D9                        EXX
180D  08                        EX      AF, AF'
180E  DD E5                     PUSH    IX
1810  FD E5                     PUSH    IY
1812  ED 57                     LD      A, I
1814  32 1BB3                   LD      (SV1), A
1817  ED 5F                     LD      A, R
1819  32 03B8                   LD      (SRR), A
181C  ED 7B 03B6                LD      SP, (SSPL)      ; RESTORE SP
1820  D1                        POP     DE              ; GET SAVED PC FROM STACK
1821  ED 53 03CE                LD      (SPCL), DE      ; SAVE IT IN MEM
1825  3B                        DEC     SP
1826  3B                        DEC     SP              ; FIX UP STACK
1827  3B                        DEC     SP
1828  3B                        DEC     SP
1829  C9                        RET

                                ; ROUTINE TO REMOVE BREAK-POINTS FROM PROGRAM

182A  AF                RMBP:   XOR     A
182B  32 03F9                   LD      (BPF), A        ; CLR BRK-PT'S-ACTIVE FLG
182E  11 0008                   LD      DE, 8           ; 8 BYTES / TBL ENTRY
1831  DD 21 03D0                LD      IX, BPBA        ; TBL BASE ADDR
1835  06 04                     LD      B, 4            ; 4 BRK-PTS MAX
1837  DD 7E 03          RMB1:   LD      A, (IX+3)       ; BRK-PT-DEFN'D FLG
183A  B7                        OR      A
183B  28 0A                     JR      Z, RMB2         ; DEFINED ?
183D  DD 6E 00                  LD      L, (IX+0)       ; YES: GET ITS ADDR
1840  DD 66 01                  LD      H, (IX+1)
1843  DD 7E 02                  LD      A, (IX+2)       ; GET SAVED PROG OP-CODE
1846  77                        LD      (HL), A         ; RESTORE IT
1847  DD 19               RMB2:   ADD     IX, DE          ; CHK NEXT BRK-PT
1849  10 EC                     DJNZ    RMB1
184B  C9                        RET

                                ; BREAK-POINT INTERRUPT ROUTINE

184C  ED 73 03B6          BPINT:  LD      (SSPL), SP      ; SAVE SP
1850  CD 17FF                   CALL    SVREG           ; SAVE REG'S
1853  3E FF                     LD      A, OFFH
1855  32 03F6                   LD      (BPN), A        ; SET BRK-PT-INTR FLG
1858  E1                        POP     HL              ; SAVED PC
1859  2B                        DEC     HL
185A  22 03CE                   LD      (SPCL), HL      ; SAVE PC OF BRK-PT
185D  01 0008                   LD      BC, 8           ; 8 BYTE TBL AREA
1860  DD 21 03D0                LD      IX, BPBA        ; TBL BASE
1864  DD 7E 03          BPI1:   LD      A, (IX+3)       ; BRK-PT-DEFN'D FLG
1867  B7                        OR      A
1868  28 0C                     JR      Z, BPI2         ; DEFINED ?
186A  DD 7E 00                  LD      A, (IX+0)       ; YES
186D  BD                        CP      L
```

```
186E  20 06              JR    NZ, BPI2      ; TBL ADDR MATCHES PC ?
1870  DD 7E 01           LD    A, (IX+1)
1873  BC                 CP    H
1874  28 04              JR    Z, BPI3
1876  DD 09        BPI2: ADD   IX, BC        ; NO: CHK NEXT TBL ENTRY
1878  18 EA              JR    BPI1
187A  DD 7E 04     BPI3: LD    A, (IX+4)     ; YES: GET DELAY NUM
187D  B7                 OR    A
187E  20 06              JR    NZ, BPI4
1880  DD 7E 05           LD    A, (IX+5)     ; DELAY 0 ?
1883  B7                 OR    A
1884  28 27              JR    Z, BPI7
1886  DD 6E 06     BPI4: LD    L, (IX+6)
1889  DD 66 07           LD    H, (IX+7)     ; NO: DECR DELAY CNTR
188C  2B                 DEC   HL
188D  DD 75 06           LD    (IX+6), L
1890  DD 74 07           LD    (IX+7), H
1893  7D                 LD    A, L
1894  B7                 OR    A
1895  20 04              JR    NZ, BPI5      ; CNTR = 0 ?
1897  7C                 LD    A, H
1898  B7                 OR    A
1899  28 06              JR    Z, BPI6
189B  CD 182A      BPI5: CALL  RMBP          ; NO: REMOVE BRK-PT
189E  C3 1AD3            JP    CTCON         ; GO TO CONTIN EXEC
18A1  DD 6E 04     BPI6: LD    L, (IX+4)     ; YES: RE-INITIALIZE CNTR
18A4  DD 66 05           LD    H, (IX+5)
18A7  DD 75 06           LD    (IX+6), L
18AA  DD 74 07           LD    (IX+7), H
18AD  CD 182A      BPI7: CALL  RMBP          ; REMOVE BRK-PTS
18B0  C3 1935            JP    ERPC          ; GO DISP PC & REG'S

                         ; ROUTINE TO SET BREAK-POINT INTERRUPT VECTOR

18B3  3E C3        SBP:  LD    A, 0C3H  ;"JP"
18B5  32 0008            LD    (8), A
18B8  21 184C            LD    HL, BPINT
18BB  22 0009            LD    (9), HL
18BE  C9                 RET

                         ; ROUTINE TO INSERT BREAK-POINTS INTO PROGRAM

18BF  CD 18B3      SUBP: CALL  SBP           ; SET BRK-PT INTR VECT
18C2  3E FF              LD    A, 0FFH
18C4  32 03F9            LD    (BPF), A      ; SET BRK-PT-ACTIVE FLG
18C7  11 0008            LD    DE, 8         ; 8 BYTE TBL AREA
18CA  DD 21 03D0         LD    IX, BPBA      ; TBL BASE
18CE  06 04              LD    B, 4
18D0  DD 7E 03     SUBP1: LD    A, (IX+3)     ; BRK-PT-DEF'D FLG
18D3  B7                 OR    A
18D4  28 0D              JR    Z, SUBP2      ; DEFINED ?
18D6  DD 6E 00           LD    L, (IX+0)     ; YES: GET ITS ADDR
18D9  DD 66 01           LD    H, (IX+1)
18DC  7E                 LD    A, (HL)       ; GET PROG OP-CODE AT BRK-PT
18DD  DD 77 02           LD    (IX+2), A     ; SAVE IT IN TBL
18E0  3E CF              LD    A, 0CFH       ; "RST8" BRK-PT INTR CODE
18E2  77                 LD    (HL), A       ; INSERT BRK-PT INTR INSTR
18E3  DD 19        SUBP2: ADD   IX, DE        ; CHK NEXT BRK-PT
18E5  10 E9              DJNZ  SUBP1
18E7  C9                 RET
```

```
                                ; ROUTINE TO INITIALIZE BREAK-POINT DELAY COUNTERS

18E8  11 0008          SUBD:    LD    DE, 8
18EB  DD 21 03D0               LD    IX, BPBA        ; TBL BASE
18EF  06 04                    LD    B, 4
18F1  DD 7E 03         SUBD1:  LD    A, (IX+3)
18F4  B7                       OR    A
18F5  28 0C                    JR    Z, SUBD2        ; BRK-PT DEFINED ?
18F7  DD 6E 04                 LD    L, (IX+4)       ; YES: GET DELAY NUM
18FA  DD 66 05                 LD    H, (IX+5)
18FD  DD 75 06                 LD    (IX+6), L       ; LOAD IT INTO CNTR
1900  DD 74 07                 LD    (IX+7), H
1903  DD 19            SUBD2:   ADD   IX, DE          ; CHK NEXT BRK-PT
1905  10 EA                     DJNZ  SUBD1
1907  C9                       RET

                                ; PROGRAM INTERRUPT ROUTINE (FROM NMI)

1908  ED 73 03B6       PRGI:    LD    (SSPL), SP      ; SAVE SP
190C  CD 17FF                  CALL  SVREG           ; SAVE REG'S
190F  AF                       XOR   A
1910  32 03F6                  LD    (BPN), A        ; CLR BRK-PT-INTR FLG
1913  3A 03B4                  LD    A, (CTF)        ; CONTIN FLG
1916  B7                       OR    A
1917  3E 00                    LD    A, 0
1919  32 03B4                  LD    (CTF), A        ; CLR IT
191C  C2 1A6D                  JP    NZ, CONTN       ; GO CONTIN EXEC OF PROG
191F  3A 03F9                  LD    A, (BPF)        ; BRK-PT'S-ACTIVE FLG
1922  B7                       OR    A
1923  28 07                    JR    Z, PRGI1        ; BRK-PT'S ACTIVE ?
1925  AF                       XOR   A
1926  32 03F9                  LD    (BPF), A        ; YES: CLR BRK-PT'S-ACTIVE FLG
1929  CD 182A                  CALL  RMBP            ; REMOVE BRK-PT'S
192C  3E 7F            PRGI1:  LD    A, 7FH
192E  D3 F8                    OUT   (LSEG), A       ; CLR ESC-INTR-ENABLE
1930  CD 10F4          PRGI2:  CALL  KBDB            ; KBD DEBOUNCE
1933  20 FB                     JR    NZ, PRGI2       ; WAIT TIL KEY RELEASED
1935  ED 5B 03CE       ERPC:   LD    DE, (SPCL)      ; GET SAVED PC
1939  CD 11A4                  CALL  ADIS            ; DIS-ASSEMBLE IT
193C  3E 15                    LD    A, 15H
193E  32 03F5                  LD    (MDB5), A       ; DISP 'PC'
1941  3E 0C                    LD    A, 0CH
1943  32 03F4                  LD    (MDB4), A
1946  CD 108A          ERDP:   CALL  DISP            ; DISP & GET CMD
1949  FE 10                    CP    GO
194B  F2 1A1E                  JP    P, EROP         ; IF CMD, DECODE IT
194E  FE 0D                    CP    0DH
1950  F2 1935                  JP    P, ERPC         ; DISP PC
1953  FE 0C                    CP    0CH
1955  20 03                    JR    NZ, PRGI3
1957  AF                       XOR   A
1958  18 01                    JR    PRGI4
195A  3C               PRGI3:  INC   A               ; SKIP OVER PC CODE
195B  32 03B5          PRGI4:  LD    (ERSC), A       ; SAVE REG SEL CODE
195E  3A 03B5          ER1:    LD    A, (ERSC)
1961  87                       ADD   A, A            ; MULT BY 2
1962  16 00                    LD    D, 0
1964  5F                       LD    E, A
1965  21 1A2E                  LD    HL, RCDTB       ; REG CODE TBL
1968  19                       ADD   HL, DE          ; PTR TO REG CODE
1969  7E                       LD    A, (HL)         ; GET THE CODE
```

```
196A  32 03F5                  LD    (MDB5), A     ; DISP IT IN DATA DIGITS
196D  23                       INC   HL
196E  7E                       LD    A, (HL)
196F  32 03F4                  LD    (MDB4), A
1972  21 03CE                  LD    HL, SREG      ; REG SAVE AREA BASE
1975  B7                       OR    A
1976  ED 52                    SBC   HL, DE        ; PTR TO SELECTED REG
1978  11 03F0                  LD    DE, MDB0
197B  EB                       EX    DE, HL
197C  1A                       LD    A, (DE)       ; GET REG CONTENTS
197D  47                       LD    B, A
197E  E6 0F                    AND   0FH           ; GET LO-BYTE
1980  77                       LD    (HL), A       ; DISP LSD
1981  23                       INC   HL
1982  CB 38                    SRL   B             ; GET NEXT DIGIT
1984  CB 38                    SRL   B
1986  CB 38                    SRL   B
1988  CB 38                    SRL   B
198A  70                       LD    (HL), B       ; DISP IT
198B  23                       INC   HL
198C  13                       INC   DE
198D  1A                       LD    A, (DE)       ; GET HI-BYTE
198E  47                       LD    B, A
198F  E6 0F                    AND   0FH           ; GET NEXT DIGIT
1991  77                       LD    (HL), A       ; DISP IT
1992  23                       INC   HL
1993  CB 38                    SRL   B             ; GET MSD
1995  CB 38                    SRL   B
1997  CB 38                    SRL   B
1999  CB 38                    SRL   B
199B  70                       LD    (HL), B       ; DISP IT
199C  CD 108A            ER6:  CALL  DISP          ; DISP & GET CMD
199F  FE 10                    CP    GO
19A1  F2 19EE                  JP    P, ER2        ; CMD ?
19A4  21 03F3                  LD    HL, MDB3      ; NO: DIGIT
19A7  77                       LD    (HL), A       ; STORE MSD OF 1ST REG
19A8  87                       ADD   A, A
19A9  87                       ADD   A, A
19AA  87                       ADD   A, A          ; ASSEMBLE MSD
19AB  87                       ADD   A, A
19AC  2B                       DEC   HL
19AD  47                       LD    B, A
19AE  CD 108A            ER7:  CALL  DISP          ; DISP & GET LSD
19B1  FE 10                    CP    GO
19B3  FA 19BC                  JP    M, ER3        ; CMD ?
19B6  FE 19                    CP    ESC           ; YES: IF ESC, EXAM THIS REG
19B8  28 A4                    JR    Z, ER1                ; PAIR AGAIN
19BA  18 F2                    JR    ER7           ; IGNORE OTHER CMD'S
19BC  77                 ER3:  LD    (HL), A       ; DISP LSD
19BD  B0                       OR    B             ; ASSEMBLE BYTE
19BE  12                       LD    (DE), A       ; STORE NEW REG DATA
19BF  CD 108A            ER8:  CALL  DISP          ; DISP IT & GET NEXT KEY
19C2  FE 10                    CP    GO
19C4  FA 19CD                  JP    M, ER4        ; CMD ?
19C7  FE 19                    CP    ESC           ; YES: IF ESC, EXAM THIS REG
19C9  28 93                    JR    Z, ER1                ; PAIR AGAIN
19CB  18 F2                    JR    ER8           ; IGNORE OTHER CMD'S
19CD  21 03F1            ER4:  LD    HL, MDB1      ; MSD OF 2ND DISP'D REG
19D0  77                       LD    (HL), A       ; STORE MSD
19D1  87                       ADD   A, A
19D2  87                       ADD   A, A          ; ASSEMBLE IT
```

```
19D3  87                    ADD   A, A
19D4  87                    ADD   A, A
19D5  2B                    DEC   HL
19D6  47                    LD    B, A
19D7  CD 108A         ER9:  CALL  DISP          ; DISP IT & GET LSD
19DA  FE 10                 CP    GO
19DC  FA 19E6               JP    M, ER5        ; CMD ?
19DF  FE 19                 CP    ESC           ; YES: IF ESC, EXAM THIS REG
19E1  CA 195E               JP    Z, ER1                ; PAIR AGAIN
19E4  18 F1                 JR    ER9           ; IGNORE OTHER CMD'S
19E6  77              ER5:  LD    (HL), A       ; STORE LSD
19E7  B0                    OR    B             ; ASSEMBLE IT
19E8  1B                    DEC   DE
19E9  12                    LD    (DE), A       ; STORE NEW REG DATA
19EA  13                    INC   DE
19EB  C3 195E               JP    ER1           ; DISP AGAIN
19EE  28 CF           ER2:  JR    Z, ER8        ; IF GO, SKIP CHANGING 1ST REG
19F0  FE 19                 CP    ESC
19F2  CA 1935               JP    Z, ERPC       ; IF ESC, GO DISP PC
19F5  FE 13                 CP    ER
19F7  28 06                 JR    Z, ER10       ; IF ER, DISP NEXT REG PAIR
19F9  FE 12                 CP    EP
19FB  28 11                 JR    Z, ER12       ; IF EP, DISP PREV REG PAIR
19FD  18 9D                 JR    ER6           ; IGNORE OTHER CMD'S
19FF  3A 03B5         ER10: LD    A, (ERSC)     ; REG SEL CODE
1A02  3C                    INC   A
1A03  FE 0D                 CP    0DH           ; INCR MODULO 12
1A05  20 01                 JR    NZ, ER11
1A07  AF                    XOR   A
1A08  32 03B5         ER11: LD    (ERSC), A     ; NEXT SEL CODE
1A0B  C3 195E               JP    ER1           ; GO DISP THE REG'S
1A0E  3A 03B5         ER12: LD    A, (ERSC)     ; REG SEL CODE
1A11  3D                    DEC   A
1A12  FE FF                 CP    0FFH          ; DECR MODULO 12
1A14  20 02                 JR    NZ, ER13
1A16  3E 0C                 LD    A, 0CH
1A18  32 03B5         ER13: LD    (ERSC), A     ; NEXT SEL CODE
1A1B  C3 195E               JP    ER1           ; DISP THE REG'S
1A1E  CA 1AD3         EROP: JP    Z, CTCON      ; IF GO, CONTIN EXEC
1A21  FE 19                 CP    ESC
1A23  CA 100F               JP    Z, CTST       ; IF ESC, ABORT
1A26  FE 15                 CP    SI
1A28  CA 1AD8               JP    Z, CTSI       ; IF SI, EXEC SINGLE INSTR
1A2B  C3 1946               JP    ERDP          ; IGNORE OTHER CMD'S

                            ; LED SEGMENT DEFINITION TABLE FOR REG NAMES DISP'D

1A2E  15              RCDTB: DEFB   15H     ; P
1A2F  0C                     DEFB   0CH     ; C
1A30  0A                     DEFB   0AH     ; A
1A31  0F                     DEFB   0FH     ; F
1A32  0B                     DEFB   0BH     ; B
1A33  0C                     DEFB   0CH     ; C
1A34  0D                     DEFB   0DH     ; D
1A35  0E                     DEFB   0EH     ; E
1A36  13                     DEFB   13H     ; H
1A37  14                     DEFB   14H     ; L
1A38  0A                     DEFB   0AH     ; A
1A39  16                     DEFB   16H     ; '
1A3A  0B                     DEFB   0BH     ; B
1A3B  16                     DEFB   16H     ; '
```

```
1A3C  0D                        DEFB  0DH     ; D
1A3D  16                        DEFB  16H     ; '
1A3E  13                        DEFB  13H     ; H
1A3F  16                        DEFB  16H     ; '
1A40  01                        DEFB  01H     ; I
1A41  17                        DEFB  17H     ; X
1A42  01                        DEFB  01H     ; I
1A43  18                        DEFB  18H     ; Y
1A44  01                        DEFB  01H     ; I
1A45  19                        DEFB  19H     ; R
1A46  05                        DEFB  05H     ; S
1A47  15                        DEFB  15H     ; P

                                ; ROUTINE TO START EXECUTION OF A PROGRAM

1A48  CD 18E8           START:  CALL  SUBD            ; INITIALIZE BRK-PT DELAY
1A4B  CD 18BF                   CALL  SUBP            ; INSERT BRK-PT'S
1A4E  CD 112A                   CALL  ADTR            ; ASSEMBLE START ADDR
1A51  CD 1A56                   CALL  SENMI           ; SET NMI INTR VECT
1A54  EB                        EX    DE, HL          ; GET START ADDR
1A55  E9                        JP    (HL)            ; JUMP TO IT

                                ; ROUTINE TO SET NMI INTERRUPT VECTOR

1A56  CD 10E0           SENMI:  CALL  CLDSP
1A59  3E C3                     LD    A, 0C3H         ; "JP"
1A5B  32 0066                   LD    (66H), A
1A5E  21 1908                   LD    HL, PRGI
1A61  22 0067                   LD    (67H), HL
1A64  3E FF                     LD    A, 0FFH
1A66  D3 F8                     OUT   (LSEG), A       ; SET EN-ESC-INTR
1A68  3E 01                     LD    A, 1            ; SEL ESC KEY
1A6A  D3 F9                     OUT   (MLED), A
1A6C  C9                        RET

                                ; ROUTINE TO CONTINUE EXECUTION OF A PROGRAM

1A6D  3A 03CE           CONTN:  LD    A, (SPCL)       ; GET SAVED PC
1A70  B7                        OR    A
1A71  20 07                     JR    NZ, CNTN2       ; IS IT VALID ?
1A73  3A 03CF                   LD    A, (SPCH)
1A76  B7                        OR    A
1A77  CA 100F                   JP    Z, CTST         ; NO: ABORT
1A7A  CD 18BF           CNTN2:  CALL  SUBP            ; INSERT BRK-PT'S
1A7D  CD 1A56                   CALL  SENMI           ; SET NMI INTR VECT
1A80  ED 5B 03CE        CNTN1:  LD    DE, (SPCL)      ; GET SAVED PC
1A84  D5                        PUSH  DE              ; SAVE IT AS RET ADDR ON STACK
1A85  31 03BA                   LD    SP, ESREG       ; END OF SAVED REG AREA
1A88  3A 03B9                   LD    A, (SVI)
1A8B  ED 47                     LD    I, A
1A8D  3A 03B8                   LD    A, (SRR)
1A90  ED 4F                     LD    R, A
1A92  FD E1                     POP   IY              ; RESTORE ALL REG'S
1A94  DD E1                     POP   IX
1A96  D9                        EXX
1A97  08                        EX    AF, AF'
1A98  E1                        POP   HL
1A99  D1                        POP   DE
1A9A  C1                        POP   BC
1A9B  F1                        POP   AF
1A9C  D9                        EXX
```

```
1A9D  08                        EX      AF, AF'
1A9E  E1                        POP     HL
1A9F  D1                        POP     DE
1AA0  C1                        POP     BC
1AA1  3A 03F6                   LD      A, (BPN)        ; BRK-PT INTR ?
1AA4  B7                        OR      A
1AA5  28 06                     JR      Z, CNTN3
1AA7  F1                        POP     AF              ; YES: RESTORE AF
1AA8  ED 7B 03B6                LD      SP, (SSPL)      ; RESTORE SP
1AAC  C9                        RET                     ; CONTIN EXEC OF PROG
1AAD  F1                CNTN3:  POP     AF
1AAE  ED 7B 03B6                LD      SP, (SSPL)
1AB2  ED 45                     RETN                    ; CONTIN EXEC OF INTR'D PROG

                                ; ROUTINE TO EXAM REGISTERS

1AB4  3A 03CE           STER:   LD      A, (SPCL)       ; GET SAVED PC
1AB7  B7                        OR      A
1AB8  20 07                     JR      NZ, STER1       ; IS IT VALID ?
1ABA  3A 03CF                   LD      A, (SPCH)
1ABD  B7                        OR      A
1ABE  CA 100F                   JP      Z, CTST         ; NO: ABORT
1AC1  C3 1935           STER1:  JP      ERPC            ; YES: GO EXAM IT

                                ; ROUTINE TO START EXECUTING A SINGLE INSTRUCTION

1AC4  CD 112A           STSI:   CALL    ADTR            ; ASSEMBLE START ADDR
1AC7  CD 1A56                   CALL    SENMI
1ACA  D5                        PUSH    DE              ; SAVE IT AS RET ADDR ON STACK
1ACB  3E 40                     LD      A, 40H          ; SING INSTR ACTIVATION CODE
1ACD  F5                        PUSH    AF              ; SET UP STACK
1ACE  D3 F9                     OUT     (MLED), A       ; START SING INSTR CNTR
1AD0  F1                        POP     AF              ; RESTORE STACK
1AD1  00                        NOP                     ; ANOTHER CNT
1AD2  C9                        RET                     ; GO EXEC A SING INSTR

                                ; ROUTINE TO CONTINUE EXECUTING NEXT SINGLE INSTRUCTION

1AD3  3E FF             CTCON:  LD      A, OFFH
1AD5  32 03B4                   LD      (CTF), A        ; SET CONTIN FLG

1AD8  3A 03CE           CTSI:   LD      A, (SPCL)       ; GET SAVED PC
1ADB  B7                        OR      A
1ADC  20 07                     JR      NZ, CTSI1       ; IS PC VALID ?
1ADE  3A 03CF                   LD      A, (SPCH)
1AE1  B7                        OR      A
1AE2  CA 100F                   JP      Z, CTST         ; NO: ABORT
1AE5  ED 5B 03CE        CTSI1:  LD      DE, (SPCL)      ; GET SAVED PC
1AE9  D5                        PUSH    DE              ; SAVE IT AS RET ADDR ON STACK
1AEA  31 03BA                   LD      SP, ESREG       ; END OF SAVED REG AREA
1AED  3A 03B9                   LD      A, (SVI)
1AF0  ED 47                     LD      I, A
1AF2  3A 03B8                   LD      A, (SRR)
1AF5  ED 4F                     LD      R, A
1AF7  FD E1                     POP     IY              ; RESTORE ALL REG'S
1AF9  DD E1                     POP     IX
1AFB  D9                        EXX
1AFC  08                        EX      AF, AF'
1AFD  E1                        POP     HL
1AFE  D1                        POP     DE
1AFF  C1                        POP     BC
```

```
1B00  F1                        POP     AF
1B01  D9                        EXX
1B02  08                        EX      AF, AF'
1B03  E1                        POP     HL
1B04  D1                        POP     DE
1B05  C1                        POP     BC
1B06  F1                        POP     AF
1B07  ED 7B 03B6                LD      SP, (SSPL)      ; RESTORE SP
1B0B  F5                        PUSH    AF              ; SAVE AF
1B0C  3E 40                     LD      A, 40H          ; SING INSTR ACTIVATION CODE
1B0E  D3 F9                     OUT     (MLED), A       ; START SING INSTR CNTR
1B10  F1                        POP     AF              ; RESTORE AF
1B11  ED 45                     RETN                    ; GO EXEC A SING INSTR

                                ; ROUTINE TO LOAD INTO MEMORY FROM CASSETTE

1B13  CD 10E0           LDCT:   CALL    CLDSP
1B16  DB FA             LD1:    IN      A, (PGAL)       ; READ CASSETTE SIGNAL
1B18  CB 7F                     BIT     7, A
1B1A  20 FA                     JR      NZ, LD1         ; WAIT FOR FALLING EDGE
1B1C  DB FA             LD2:    IN      A, (PGAL)
1B1E  CB 7F                     BIT     7, A
1B20  28 FA                     JR      Z, LD2          ; WAIT FOR RISING EDGE
1B22  06 60                     LD      B, 60H
1B24  10 FE             LD3:    DJNZ    LD3             ; DELAY TO 3/4 OF BIT CELL (FM)
1B26  DB FA                     IN      A, (PGAL)
1B28  CB 7F                     BIT     7, A
1B2A  28 10                     JR      Z, LD6          ; IS THE BIT A '1' ?
1B2C  DB FA             LD4:    IN      A, (PGAL)       ; NO
1B2E  CB 7F                     BIT     7, A
1B30  20 FA                     JR      NZ, LD4         ; WAIT FOR FALLING EDGE
1B32  06 60                     LD      B, 60H
1B34  10 FE             LD5:    DJNZ    LD5             ; 3/4 BIT CELL DELAY
1B36  DB FA                     IN      A, (PGAL)
1B38  CB 7F                     BIT     7, A
1B3A  28 E0                     JR      Z, LD2          ; IS THE BIT A '0' ?
1B3C  21 03FC           LD6:    LD      HL, SAL         ; NO: FOUND HEADER ON TAPE
1B3F  06 04                     LD      B, 4
1B41  3E 01                     LD      A, 1            ; INITIALIZE CHECK-SUM
1B43  CD 1BF7           LD7:    CALL    BTS             ; ASSEMBLE BYTE FROM TAPE
1B46  77                        LD      (HL), A         ; STORE NEXT HEADER BYTE
1B47  5F                        LD      E, A
1B48  08                        EX      AF, AF'
1B49  83                        ADD     A, E            ; ADD IT TO CHECK-SUM
1B4A  23                        INC     HL
1B4B  10 F6                     DJNZ    LD7             ; END OF HEADER ?
1B4D  2A 03FC                   LD      HL, (SAL)       ; YES: GET SAVED ADDR
1B50  ED 4B 03FE                LD      BC, (BCL)       ; GET SAVED BYTE-CNT
1B54  CD 1BF7           LD8:    CALL    BTS             ; ASSEMBLE NEXT DATA BYTE
1B57  77                        LD      (HL), A         ; STORE IT IN MEM
1B58  5F                        LD      E, A
1B59  08                        EX      AF, AF'
1B5A  83                        ADD     A, E            ; ADD IT TO CHECK-SUM
1B5B  ED A1                     CPI                     ; DECR BYTE-CNT
1B5D  EA 1B54                   JP      PE, LD8         ; END OF DATA ?
1B60  CD 1BF7                   CALL    BTS             ; YES: ASSEMBLE CHECK-SUM
1B63  5F                        LD      E, A                    ; FROM TAPE
1B64  08                        EX      AF, AF'
1B65  83                        ADD     A, E            ; ADD IT TO CALC'D CHECK-SUM
1B66  28 08                     JR      Z, LD9          ; CHECK-SUM ERROR ?
1B68  3E 12                     LD      A, 12H          ; YES
```

```
1B6A  32 03F5                    LD    (MDB5), A      ; DISP TRIPLE BAR ERR IND
1B6D  C3 1017                    JP    CTST1          ; ABORT
1B70  ED 5B 03FC          LD9:   LD    DE, (SAL)      ; GET SAVED ADDR
1B74  CD 11A4                    CALL  ADIS           ; DIS-ASSEMBLE IT INTO DISP BUF
1B77  C3 113F                    JP    STEM           ; GO TO EXAM-MEM ROUTINE

                                 ; ROUTINE TO WRITE MEMORY BLOCK ONTO CASSETTE

1B7A  CD 112A             STSV:  CALL  ADTR           ; ASSEMBLE START ADDR
1B7D  D5                         PUSH  DE             ; SAVE IT
1B7E  11 03F5                    LD    DE, MDB5       ; TO DISP PROMPT IN DATA LSD
1B81  CD 12E2                    CALL  ADCOM          ; DECODE NEXT KEY CMD
1B84  FE 19                      CP    ESC
1B86  20 04                      JR    NZ, SV5        ; ESC CMD ?
1B88  D1                         POP   DE             ; YES: RESTORE STACK
1B89  C3 100F                    JP    CTST           ; GO BACK TO CMD PROMPT
1B8C  CD 112A             SV5:   CALL  ADTR           ; ASSEMBLE END ADDR
1B8F  CD 10E0                    CALL  CLDSP
1B92  62                         LD    H, D
1B93  6B                         LD    L, E           ; END ADDR
1B94  C1                         POP   BC             ; START ADDR
1B95  ED 43 03FC                 LD    (SAL), BC      ; SAVE START ADDR
1B99  B7                         OR    A
1B9A  ED 42                      SBC   HL, BC         ; CALC BYTE CNT
1B9C  23                         INC   HL
1B9D  22 03FE                    LD    (BCL), HL      ; SAVE IT
1BA0  3E 01                      LD    A, 1           ; PRE-AMBLE SYNC BYTE
1BA2  32 03FB                    LD    (PRE), A       ; SAVE IT
1BA5  AF                         XOR   A
1BA6  D3 F9                      OUT   (MLED), A      ; DISABLE DISP DIGITS
1BA8  3E 3F                      LD    A, 3FH
1BAA  D3 F8                      OUT   (LSEG), A      ; CLR CASSETTE SINE-WAVE
1BAC  3E FF                      LD    A, OFFH              ; GEN CNTR
1BAE  D3 F8                      OUT   (LSEG), A
1BB0  06 03                      LD    B, 3
1BB2  AF                         XOR   A
1BB3  D3 FA               SV1:   OUT   (PGAL), A      ; STEP IT 3 TIMES TO REACH
1BB5  10 FC                      DJNZ  SV1                  ; 0-CROSSING
1BB7  08                         EX    AF, AF'
1BB8  AF                         XOR   A
1BB9  08                         EX    AF, AF'
1BBA  AF                         XOR   A              ; CLR CHECK-SUM
1BBB  01 0800                    LD    BC, 800H       ; SEND OUT 2K BYTES OF 0'S
1BBE  1E 00               SV2:   LD    E, 0                ; (8 SEC PRE-AMBLE)
1BC0  CD 1C26                    CALL  BSH
1BC3  ED A1                      CPI
1BC5  EA 1BBE                    JP    PE, SV2        ; END OF PRE-AMBLE ?
1BC8  06 05                      LD    B, 5           ; YES
1BCA  21 03FB                    LD    HL, PRE        ; SEND OUT HEADER
1BCD  5E                  SV3:   LD    E, (HL)
1BCE  CD 1C26                    CALL  BSH
1BD1  23                         INC   HL
1BD2  10 F9                      DJNZ  SV3            ; END OF HEADER ?
1BD4  2A 03FC                    LD    HL, (SAL)      ; YES: GET START ADDR
1BD7  ED 4B 03FE                 LD    BC, (BCL)      ; GET BYTE-CNT
1BDB  5E                  SV4:   LD    E, (HL)        ; GET NEXT DATA BYTE
1BDC  CD 1C26                    CALL  BSH            ; SEND IT OUT
1BDF  ED A1                      CPI                  ; DECR BYTE-CNT
1BE1  EA 1BDB                    JP    PE, SV4        ; END OF DATA ?
1BE4  ED 44                      NEG                  ; YES: COMPL. CHECK-SUM
1BE6  5F                         LD    E, A
```

```
1BE7  CD 1C26                 CALL  BSH          ; SEND IT OUT
1BEA  1E 00                   LD    E, 0
1BEC  CD 1C26                 CALL  BSH          ; SEND OUT 2 0-BYTES
1BEF  1E 00                   LD    E, 0
1BF1  CD 1C26                 CALL  BSH
1BF4  C3 100F                 JP    CTST         ; GO BACK TO CMD PROMPT

                              ; ROUTINE TO DECODE & ASSEMBLE A BYTE FROM CASSETTE

1BF7  D9              BTS:    EXX
1BF8  08                      EX    AF, AF'
1BF9  16 08                   LD    D, 8         ; BIT CNT
1BFB  DB FB           BTS1:   IN    A, (PGAH)    ; READ CASSETTE SIGNAL
1BFD  CB 7F                   BIT   7, A
1BFF  20 0F                   JR    NZ, BTS4
1C01  DB FB           BTS2:   IN    A, (PGAH)
1C03  CB 7F                   BIT   7, A
1C05  28 FA                   JR    Z, BTS2      ; WAIT FOR RISING EDGE
1C07  06 60                   LD    B, 60H
1C09  10 FE           BTS3:   DJNZ  BTS3         ; 3/4 BIT CELL DELAY
1C0B  DB FB                   IN    A, (PGAH)    ; READ THE BIT FROM CASSETTE
1C0D  2F                      CPL                ; IT IS INVERTED HERE
1C0E  18 0C                   JR    BTS6
1C10  DB FB           BTS4:   IN    A, (PGAH)
1C12  CB 7F                   BIT   7, A
1C14  20 FA                   JR    NZ, BTS4     ; WAIT FOR FALLING EDGE
1C16  06 60                   LD    B, 60H
1C18  10 FE           BTS5:   DJNZ  BTS5         ; 3/4 BIT CELL DELAY
1C1A  DB FA                   IN    A, (PGAL)    ; RD THE BIT FROM CASSETTE
1C1C  CB 17           BTS6:   RL    A
1C1E  CB 13                   RL    E            ; SHIFT IT IN (MSB IS 1ST)
1C20  15                      DEC   D
1C21  20 D8                   JR    NZ, BTS1     ; LAST BIT ?
1C23  7B                      LD    A, E         ; YES
1C24  D9                      EXX
1C25  C9                      RET

                              ; ROUTINE TO DIS-ASSEMBLE BYTE, ENCODE IT (FM), AND
                                     ; SHIFT IT OUT TO CASSETTE AS SINE-WAVE

1C26  83              BSH:    ADD   A, E         ; ADD BYTE TO CHECK-SUM
1C27  08                      EX    AF, AF'
1C28  16 09                   LD    D, 9         ; BIT CNTR
1C2A  15              BSH1:   DEC   D
1C2B  20 02                   JR    NZ, BSH2     ; LAST BIT ?
1C2D  08                      EX    AF, AF'      ; YES
1C2E  C9                      RET
1C2F  CB 23           BSH2:   SLA   E            ; SHIFT NEXT DATA BIT (MSD 1ST)
1C31  30 10                   JR    NC, BSH5     ; 0-BIT ?
1C33  D9                      EXX                ; NO
1C34  0E 10                   LD    C, 10H       ; 16 STEPS = 1 FULL SINE-WAVE
1C36  06 06           BSH3:   LD    B, 6                ; CYCLE OF 500 US (2 KHZ)
1C38  10 FE           BSH4:   DJNZ  BSH4         ; 31 US DELAY
1C3A  D3 FA                   OUT   (PGAL), A    ; NEXT CNTR STEP OF SINE-WAVE
1C3C  0D                      DEC   C
1C3D  00                      NOP
1C3E  20 F6                   JR    NZ, BSH3     ; END OF CYCLE ?
1C40  D9                      EXX                ; YES: END OF BIT
1C41  18 E7                   JR    BSH1
1C43  D9              BSH5:   EXX
1C44  0E 08                   LD    C, 8         ; 8 STEPS = 1/2 SINE-WAVE CYCLE
```

```
1C46  06 10          BSH6:  LD    B, 10H          ; OF 500 US (1 KHZ)
1C48  10 FE          BSH7:  DJNZ  BSH7            ; 62.5 US DELAY
1C4A  D3 FA                 OUT   (PGAL), A       ; NEXT CNTR STEP OF SINE-WAVE
1C4C  0D                    DEC   C
1C4D  20 F7                 JR    NZ, BSH6        ; END OF HALF-CYCLE ?
1C4F  D9                    EXX                   ; YES: END OF BIT
1C50  18 D8                 JR    BSH1

1C52                        DS    55H             ; TO REACH ADDR OFDOOH

                            ; CASSETTE CALIBRATION PROGRAM

1CA7  CD 10E0        CALTX: CALL  CLDSP
1CAA  AF                    XOR   A               ; INITIALIZE
1CAB  D3 F9                 OUT   (MLED), A       ; DISABLE DISP DIGITS
1CAD  3E 3F                 LD    A, 3FH
1CAF  D3 F8                 OUT   (LSEG), A       ; CLR CASSETTE SINE-WAVE
1CB1  3E FF                 LD    A, OFFH               ; GEN CNTR
1CB3  D3 F8                 OUT   (LSEG), A
1CB5  06 03                 LD    B, 3
1CB7  AF                    XOR   A
1CB8  D3 FA          CALT1: OUT   (PGAL), A       ; STEP IT 3 TIMES TO REACH
1CBA  10 FC                 DJNZ  CALT1                 ; 0-CROSSING
1CBC  08                    EX    AF, AF'
1CBD  AF                    XOR   A
1CBE  08                    EX    AF, AF'
1CBF  1E 00          CALT2: LD    E, 0
1CC1  CD 1C26               CALL  BSH             ; SEND OUT 0-BYTES
1CC4  18 F9                 JR    CALT2
1CC6  00                    NOP

1CC7  CD 10E0        CALRX: CALL  CLDSP
1CCA  3E 4F                 LD    A, 4FH          ; VERT BAR SYMB
1CCC  D3 F8                 OUT   (LSEG), A
1CCE  06 00          CALR1: LD    B, 0            ; PULSE-WIDTH CNTR
1CD0  DB FA          CALR2: IN    A, (PGAL)       ; READ CASSETTE SIGNAL
1CD2  CB 7F                 BIT   7, A
1CD4  28 FA                 JR    Z, CALR2        ; WAIT FOR RISING EDGE
1CD6  04             CALR3: INC   B               ; COUNT WIDTH OF PULSE
1CD7  DB FA                 IN    A, (PGAL)
1CD9  CB 7F                 BIT   7, A
1CDB  20 F9                 JR    NZ, CALR3
1CDD  78                    LD    A, B            ; GET PULSE-WIDTH CNT
1CDE  FE 32                 CP    32H             ; CALC DEVIATION FROM
1CE0  FA 1CF6               JP    M, DLT                  ; CORRECT WIDTH
1CE3  FE 37                 CP    37H
1CE5  FA 1CFA               JP    M, DLTI
1CE8  FE 41                 CP    41H
1CEA  F2 1CFE               JP    P, DRT
1CED  FE 3C                 CP    3CH
1CEF  F2 1D02               JP    P, DRTI
1CF2  3E 01                 LD    A, 1            ; CORRECT (BAR CENTERED)
1CF4  18 0E                 JR    CALR4
1CF6  3E 08          DLT:   LD    A, 8H
1CF8  18 0A                 JR    CALR4
1CFA  3E 04          DLTI:  LD    A, 4
1CFC  18 06                 JR    CALR4
1CFE  3E 10          DRT:   LD    A, 10H
1D00  18 02                 JR    CALR4
1D02  3E 20          DRTI:  LD    A, 20H
1D04  D3 F9          CALR4: OUT   (MLED), A
```

```
1D06  18 C6                     JR      CALR1
                        ;
                        ;DOWNLOAD ROUTINE TO MULTIFLEX VIA RS 232
                        ;
1D08  AF                DNLD:   XOR     A
1D09  3E 4E                     LD      A, 4EH
1D0B  D3 F5                     OUT     (SER2), A
1D0D  3E 05                     LD      A, 05H
1D0F  D3 F5                     OUT     (SER2), A
1D11  CD 1D3E           DNWT:   CALL    SERD
1D14  7A                        LD      A, D
1D15  FE 00                     CP      00H
1D17  20 F8                     JR      NZ, DNWT
1D19  CD 1D3E                   CALL    SERD
1D1C  6A                        LD      L, D
1D1D  CD 1D3E                   CALL    SERD
1D20  62                        LD      H, D
1D21  CD 1D3E                   CALL    SERD
1D24  4A                        LD      C, D
1D25  CD 1D3E                   CALL    SERD
1D28  42                        LD      B, D
1D29  AF                        XOR     A
1D2A  CD 1D3E           DNCT:   CALL    SERD
1D2D  72                        LD      (HL),D
1D2E  82                        ADD     A, D
1D2F  ED A1                     CPI
1D31  EA 1D2A                   JP      PE, DNCT
1D34  CD 1D3E                   CALL    SERD
1D37  82                        ADD     A, D
1D38  C2 1056                   JP      NZ, CDER
1D3B  C3 15E0                   JP      MTD

1D3E  08                SERD:   EX      AF,AF'
1D3F  DB F5             SERD1:  IN      A,(SER2)
1D41  E6 02                     AND     02H
1D43  28 FA                     JR      Z, SERD1
1D45  DB F4                     IN      A, (SER1)
1D47  57                        LD      D, A
1D48  08                        EX      AF, AF'
1D49  C9                        RET
                        ;
                        ;       CP/M BOOT
                        ;
1D4A  3E FE             BOOT:   LD      A, 0FEH
1D4C  31 0100                   LD      SP, 0100H
1D4F  57                TRY:    LD      D, A
1D50  3E 00                     LD      A, 00H
1D52  D3 50                     OUT     (WRST), A
1D54  06 FF                     LD      B, 0FFH
1D56  10 FE             HERE1:  DJNZ    HERE1
1D58  2F                        CPL
1D59  D3 50                     OUT     (WRST), A
1D5B  7A                        LD      A, D
1D5C  2F                        CPL
1D5D  D3 53                     OUT     (WSEL), A
1D5F  3E D0                     LD      A, 0D0H
1D61  D3 54                     OUT     (FCON), A
1D63  06 FF                     LD      B, 0FFH
1D65  10 FE             HERE2:  DJNZ    HERE2
1D67  3E 0B                     LD      A, 0BH
1D69  D3 54                     OUT     (FCON), A
```

```
1D6B    06 FF                       LD      B, OFFH
1D6D    10 FE               HERE3:  DJNZ    HERE3
1D6F    DB 54               SEE:    IN      A, (FCON)
1D71    32 0081                     LD      (FLOPA), A
1D74    CB 47                       BIT     0, A
1D76    20 F7                       JR      NZ, SEE
1D78    EE 04                       XOR     04H
1D7A    E6 84                       AND     84H
1D7C    20 34                       JR      NZ, NEXT
1D7E    AF                          XOR     A
1D7F    3C                          INC     A
1D80    D3 56                       OUT     (FSEC), A
1D82    7A                          LD      A, D
1D83    EE 80                       XOR     80H
1D85    2F                          CPL
1D86    D3 53                       OUT     (WSEL), A
1D88    01 8057                     LD      BC, 8057H
1D8B    21 0000                     LD      HL, 0000H
1D8E    3E 8C                       LD      A, 8CH
1D90    D3 54                       OUT     (FCON),A
1D92    ED B2                       INIR
1D94    DB 57               LOOK:   IN      A, (FDATA)
1D96    10 FC                       DJNZ    LOOK
1D98    06 80                       LD      B, 80H
1D9A    DB 57               VIEW:   IN      A, (FDATA)
1D9C    10 FC                       DJNZ    VIEW
1D9E    7A                          LD      A, D
1D9F    2F                          CPL
1DA0    D3 53                       OUT     (WSEL), A
1DA2    2F                          CPL
1DA3    77                          LD      (HL), A
1DA4    DB 54               WHEN:   IN      A, (FCON)
1DA6    32 0082                     LD      (FLOPB), A
1DA9    CB 47                       BIT     0, A
1DAB    20 F7                       JR      NZ, WHEN
1DAD    E6 9C                       AND     9CH
1DAF    CA 0000                     JP      Z, 0000H
1DB2    7A                  NEXT:   LD      A, D
1DB3    3E 9E                       LD      A, 9EH
1DB5    18 98                       JR      TRY

                                    .DEPHASE

                                    END
```

Macros:

Symbols:

ADC1	12F6	ADC2	12FA	ADCOM	12E2	ADIS	11A4
ADTR	112A	BCL	03FE	BOOT	1D4A	BP	0017
BP0	16E4	BP1	16E7	BP2	1710	BP3	1732
BP4	1737	BP5	173B	BP6	1744	BPBA	03D0
BPE	16EC	BPF	03F9	BPI1	1864	BPI2	1876
BPI3	187A	BPI4	1886	BPI5	189B	BPI6	18A1
BPI7	18AD	BPINT	184C	BPN	03F6	BSH	1C26
BSH1	1C2A	BSH2	1C2F	BSH3	1C36	BSH4	1C38
BSH5	1C43	BSH6	1C46	BSH7	1C48	BTS	1BF7
BTS1	1BFB	BTS2	1C01	BTS3	1C09	BTS4	1C10
BTS5	1C18	BTS6	1C1C	BYTCNT	03B0	CALR1	1CCE
CALR2	1CD0	CALR3	1CD6	CALR4	1D04	CALRX	1CC7
CALT1	1CB8	CALT2	1CBF	CALTX	1CA7	CBPNM	03FA
CDER	1056	CHECK	148E	CHECK1	1492	CLBP	1754
CLD1	10E7	CLDA1	17FA	CLDSA	17F3	CLDSP	10E0
CMP	001B	CMP1	1271	CMP2	127F	CMP3	1290
CMP4	129E	CNTN1	1A80	CNTN2	1A7A	CNTN3	1AAD
CONTN	1A6D	CPM	001F	CTCON	1AD3	CTEM	12D8
CTF	03B4	CTSI	1AD8	CTSI1	1AE5	CTST	100F
CTST1	1017	DB1	1473	DB2	1475	DDIS	11C9
DELAY	1482	DELAYB	1470	DEV1	123B	DEV2	1244
DEV3	124E	DEVICE	03A8	DEVICK	1232	DISP	108A
DLT	1CF6	DLTI	1CFA	DLY10	1597	DNCT	1D2A
DNLD	1D08	DNWT	1D11	DRT	1CFE	DRTI	1D02
DS1	108B	DS2	1093	DS3	10A2	DS4	10B9
DS5	10D3	DS6	10D8	DSTADD	03AE	DWN	001E
EDL1	1795	EDL2	17AE	EDL3	17B3	EDL4	17B7
EDL5	17C0	EDLN	177E	EI0	001D	EI01	1610
EI02	1635	EI03	1654	EI04	161D	EI05	1666
EI06	167C	EI07	168E	EI08	1634	EM	0011
EM1	1167	EM2	1189	EM3	118E	EM4	1198
EM5	119B	EMESC	1174	EMOP	117A	EMSM1	159E
EMSM2	15B2	EMSM3	15B6	ENDADD	03AA	EP	0012
EPDIS	15C0	EPMISM	1598	EPTABL	1217	ER	0013
ER1	195E	ER10	19FF	ER11	1A08	ER12	1A0E
ER13	1A18	ER2	19EE	ER3	19BC	ER4	19CD
ER5	19E6	ER6	199C	ER7	19AE	ER8	19BF
ER9	19D7	ERDP	1946	EROP	1A1E	ERPC	1935
ERSC	03B5	ESC	0019	ESREG	03BA	EVOLT	1573
EXA	008E	FCON	0054	FDATA	0057	FLOPA	0081
FLOPB	0082	FSEC	0056	GETA1	1122	GETA2	1115
GETA3	1110	GETAD	1102	GO	0010	HERE1	1D56
HERE2	1D65	HERE3	1D6D	KBDB	10F4	KDB1	10F6
KDB2	10F8	KEYCD	16C0	KEYCT	10EC	KYC1	10EE

LD	0016	LD1	1B16	LD2	1B1C	LD3	1B24
LD4	1B2C	LD5	1B34	LD6	1B3C	LD7	1B43
LD8	1B54	LD9	1B70	LDCT	1B13	LOOK	1D94
LOOP	1485	LOOP1	1487	LSEG	00F8	LSGTB	16A4
MDB0	03F0	MDB1	03F1	MDB2	03F2	MDB3	03F3
MDB4	03F4	MDB5	03F5	MISM	12B0	MKB	00F9
MLED	00F9	MSM1	12B6	MSM2	12CA	MSM3	12CE
MTD	15E0	MTD1	15EB	MV	0014	MV1	1307
MV2	1316	MV3	1327	MV4	1344	MV5	134E
NBP	17CC	NCLR	14A5	NCLR1	14AA	NCLR2	14B8
NEXT	1DB2	OFER	13AB	OFT	001C	OFT1	13A2
OFT2	1386	OFT3	1398	OFT4	1367	PBP	17DF
PG2708	14E2	PGAH	00FB	PGAL	00FA	PGD	00FC
PGM	001A	PGM1	13BA	PGM14	14C9	PGM16	14E8
PGM17	14EA	PGM18	150C	PGM19	150E	PGM2	13CC
PGM20	1511	PGM21	1522	PGM22	154A	PGM23	154C
PGM24	1562	PGM25	1565	PGM26	1505	PGM3	13DD
PRE	03FB	PRGI	1908	PRGI1	192C	PRGI2	1930
PRGI3	195A	PRGI4	195B	PROG	1431	PROG1	1448
RCDTB	1A2E	RDPGM	11E0	READ	1253	READ1	1259
RMB1	1837	RMB2	1847	RMBP	182A	SA1	105D
SAL	03FC	SBP	18B3	SCA	03F7	SEE	1D6F
SENMI	1A56	SER1	00F4	SER2	00F5	SERD	1D3E
SERD1	1D3F	SI	0015	SPCH	03CF	SPCL	03CE
SREG	03CE	SRR	03B8	SSPL	03B6	ST	1000
ST1	100B	STADDR	03AC	START	1A48	STBP	16E0
STCMP	126D	STEI0	15F5	STEM	113F	STER	1AB4
STER1	1AC1	STKB	03A0	STMV	1302	STOFT	1355
STPGM	13B3	STSI	1AC4	STSV	1B7A	SUBD	18E8
SUBD1	18F1	SUBD2	1903	SUBP	18BF	SUBP1	18D0
SUBP2	18E3	SV	0018	SV1	1BB3	SV2	1BBE
SV3	1BCD	SV4	1BDB	SV5	1B8C	SVI	03B9
SVREG	17FF	TRY	1D4F	VERIFY	14BE	VIEW	1D9A
WHEN	1DA4	WRST	0050	WSEL	0053		

No Fatal error(s)

MULTIFLEX ECONOMY VIDEO BOARD (MEVB-1)

INTRODUCTION

The MEVB-1 is a high quality display board which provides a means of interfacing a keyboard and monitor to an S-100 system. The monitor should be capable of displaying 80 x 24 characters, since this is the standard format assumed by the MEVB-1. The keyboard interface is through a 16-pin standard I.C. socket. The pin indentification is given later in the manual for those people who own their own keyboards. For the user who own the MULTIFLEX keyboard, the interface is pin-for-pin compatible with the MEVB-1.

The MEVB-1 communicates to the S-100 bus via two sequntial I/O ports whose location can be selected by the user to fit anywhere within the S-100 I/O space. One port is a read-only status port, which indicates when the board is busy and when data is available. The other port is a bidirectional data port, for communication to and from the S-100 bus.

The MEVB-1 provides two sets of monitor siganls. The first is composite video available through a phono plug connector. The other set is separate TTL level horizontal and vertical retrace as well as the analog video signal.

SECTION 1

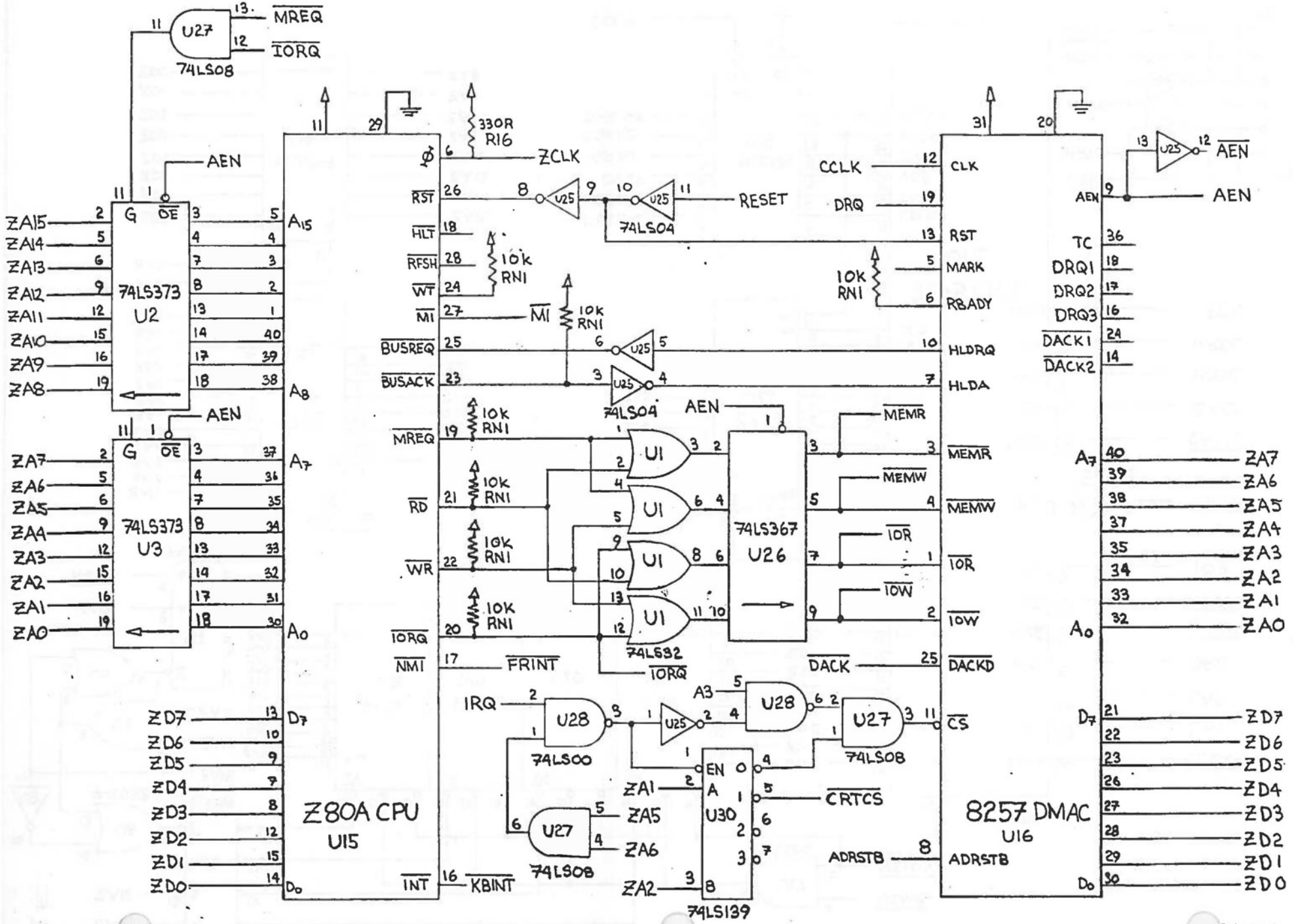
POWER-UP AND TESTING

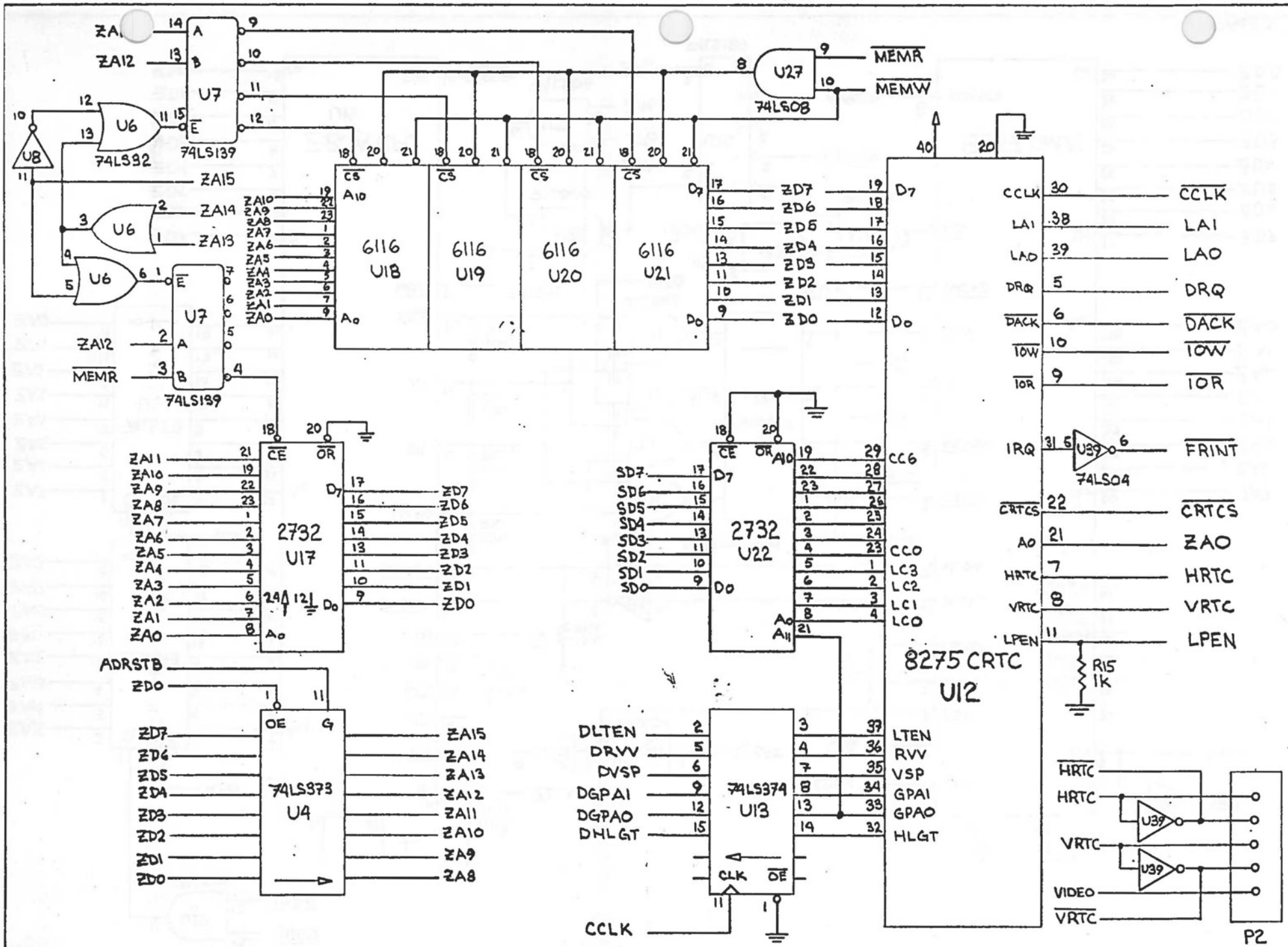
The MEVB-1 is completely self sufficient in the sense that all of the various hardware components are initialized by the on-board software. In fact, upon power-up, the board should be fully functional and the user should not have to make any adjustments what so ever. The only place where adjustments could be made are on the syncronization pulses. Here the user (by changing the value of a capacitor-resistor pair) can alter the time for the front and back portch for the horizontal and vertical syncronization pulses respectively. Another hardware feature that the user can select the location of the data ports in the I/O space. This is acheived by the jumpers at the bottom of the board. There are 7 jumper platforms which allow the data/status port pair to be located in 128 different loactions. (Eg: if the status/data port pair is to be loacted at 00H and 01H respectively, then all the jumpers must be installed.)

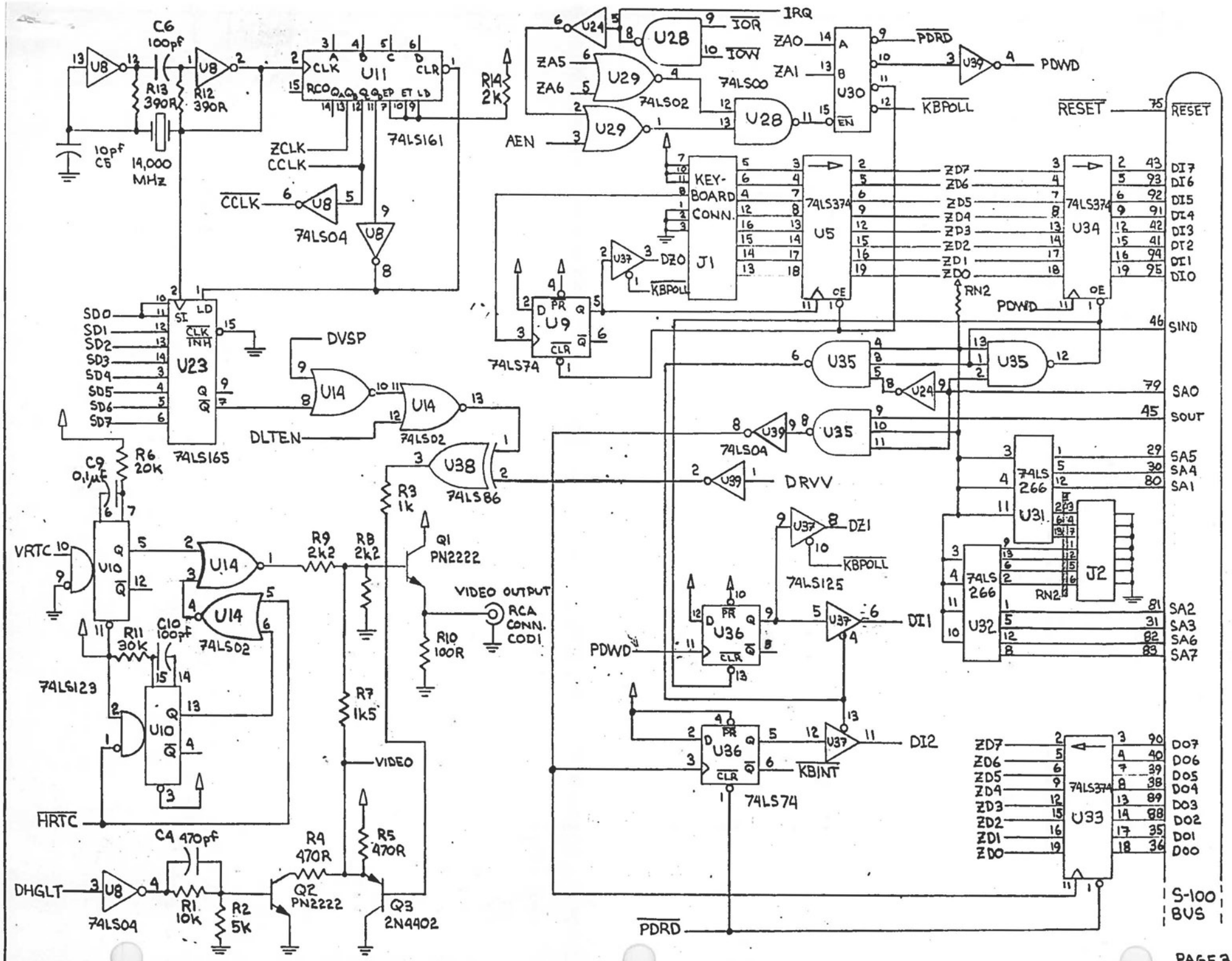
sssssssss

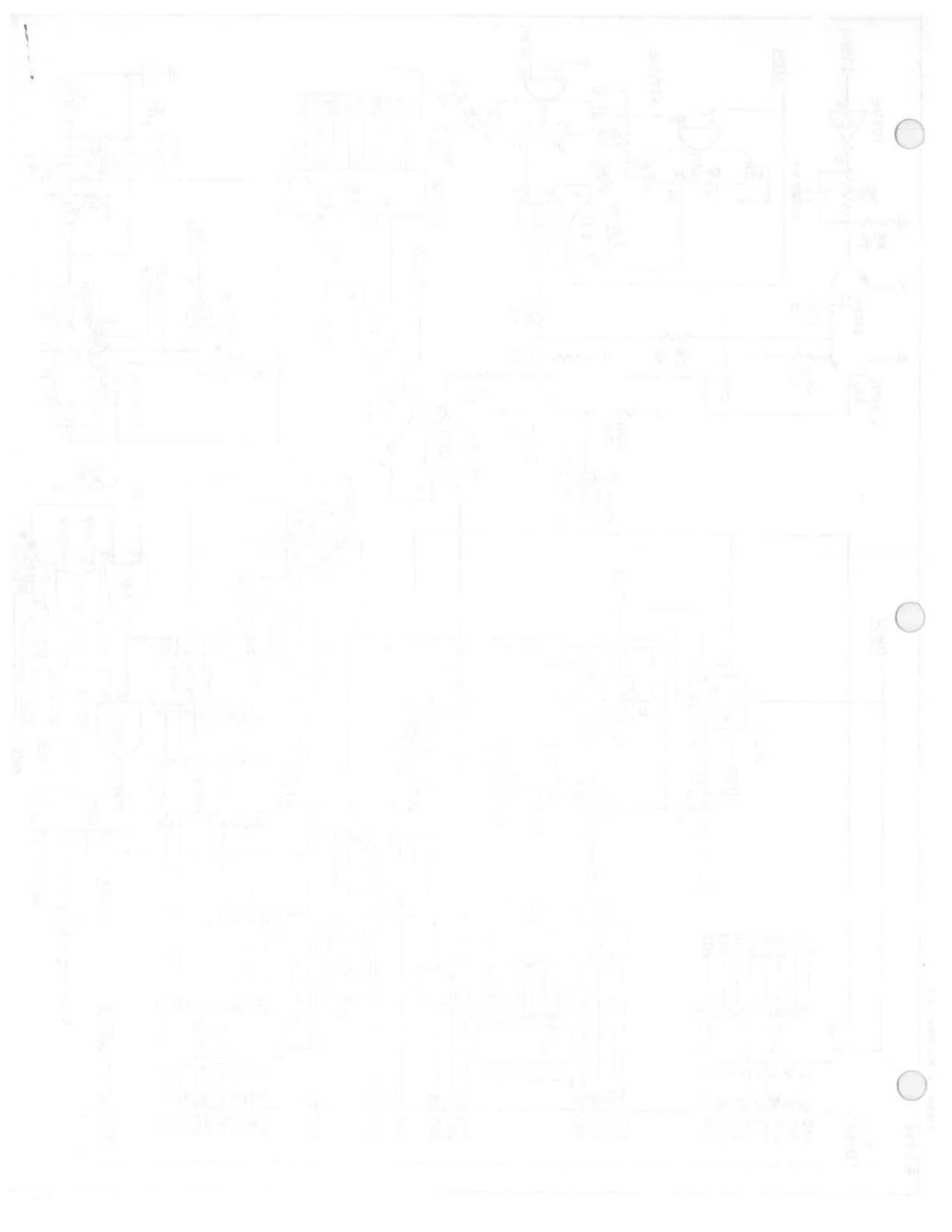
ECONOMY VIDEO BOARD SERIES I

- Rev. A
- Rev. B -




ZA14
ZA12
U7
74LS139
U6
74LS32
U8
ZA15
ZA14
ZA13
U6
U6
U7
ZA12
MEMR
74LS139
U27
74LS08
MEMR
MEMW
6116 U18
6116 U19
6116 U20
6116 U21
ZA10
ZA9
ZA8
ZA7
ZA6
ZA5
ZA4
ZA3
ZA2
ZA1
ZA0
ZD7
ZD6
ZD5
ZD4
ZD3
ZD2
ZD1
ZD0
8275 CRTC U12
CCLK
LA1
LA0
DRQ
DACK
IOW
IOR
IRQ
U39
74LS04
FRINT
CRTCS
ZA0
HRTC
VRTC
LPEN
R15 1K
2732 U17
2732 U22
SD7
SD6
SD5
SD4
SD3
SD2
SD1
SD0
CC6
CC0
LC3
LC2
LC1
LC0
ADRSTB
ZD0
74LS373 U4
ZA15
ZA14
ZA13
ZA12
ZA11
ZA10
ZA9
ZA8
DLTEN
DRVV
DVSP
DGPAI
DGPAO
DHLGT
74LS374 U13
LTEN
RVV
VSP
GPAI
GPAO
HLGT
CCLK
HRTC
VRTC
VIDEO
P2





```
0000
                        .Z80

00F2                    PREG    EQU     0F2H
00F3                    SREG    EQU     0F3H
00F3                    CREG    EQU     0F3H
9FF0                    CRTRS   EQU     9FF0H
9FF1                    RPARM1  EQU     9FF1H
9FF2                    RPARM2  EQU     9FF2H
9FF3                    RPARM3  EQU     9FF3H
9FF4                    RPARM4  EQU     9FF4H
9FF5                    STRTDIS EQU     9FF5H
9FF6                    LDCURS  EQU     9FF6H
9FF7                    CURSCOL EQU     9FF7H
9FF8                    CURSROW EQU     9FF8H
00A0                    ENINT   EQU     0A0H
00C0                    DISINT  EQU     0C0H
0040                    STPDIS  EQU     40H
00E0                    PRESET  EQU     0E0H
9FF9                    MODE    EQU     9FF9H
9FFA                    DADDRL  EQU     9FFAH
9FFB                    DADDRH  EQU     9FFBH
9FFC                    DTCNTL  EQU     9FFCH
9FFD                    DTCNTH  EQU     9FFDH
00F0                    CHOADD  EQU     0F0H
00F1                    CHOTC   EQU     0F1H
00F8                    MODEAD  EQU     0F8H
00F8                    STATUS  EQU     0F8H
9FFE                    KBDATA  EQU     9FFEH
0080                    KBPOLL  EQU     80H
0081                    KBREAD  EQU     81H
0001                    CNTA    EQU     01
0002                    CNTB    EQU     02
0003                    CNTC    EQU     03
0004                    CNTD    EQU     04
0005                    CNTE    EQU     05
0006                    CNTF    EQU     06
0007                    CNTG    EQU     07
0008                    CNTH    EQU     08
0009                    CNTI    EQU     09
000A                    CNTJ    EQU     0AH
000B                    CNTK    EQU     0BH
000C                    CNTL    EQU     0CH
000D                    CNTM    EQU     0DH
000E                    CNTN    EQU     0EH
000F                    CNTO    EQU     0FH
0010                    CNTP    EQU     10H
0011                    CNTQ    EQU     11H
0012                    CNTR    EQU     12H
0013                    CNTS    EQU     13H
0014                    CNTT    EQU     14H
0015                    CNTU    EQU     15H
0016                    CNTV    EQU     16H
0017                    CNTW    EQU     17H
0018                    CNTX    EQU     18H
0019                    CNTY    EQU     19H
001A                    CNTZ    EQU     1AH
001E                    CNTTIL  EQU     1EH
001D                    CNTSQB  EQU     1DH
001C                    CNTBSL  EQU     1CH
```

```
001B                    ESC     EQU     1BH
9FFF                    CPUDATA EQU     9FFFH
9FE0                    TABSP   EQU     9FE0H
9FE1                    CRSLN   EQU     9FE1H
9FE1                    CURSLN  EQU     9FE1H
9FE2                    FRMLN   EQU     9FE2H
9FE3                    CNTRST  EQU     9FE3H
9FE4                    CRTCST  EQU     9FE4H
9FE5                    SCRLPT  EQU     9FE5H
9FE6                    ONEPG   EQU     9FE6H
9FE7                    BUFSIZ  EQU     9FE7H
0020                    SPACE   EQU     20H
00E1                    GREEN   EQU     0E1H
00E2                    RED     EQU     0E2H
00E3                    YELLOW  EQU     0E3H
00E4                    BLUE    EQU     0E4H
00E5                    TURQSE  EQU     0E5H
00E6                    PURPLE  EQU     0E6H


                        ORG 0
                        .PHASE  0
0000    C3 0100         JP      CLDST
                        .DEPHASE
                        ORG     38H
                        .PHASE  38H
0038    C3 0243         JP      MASKINT
                        .DEPHASE
                        ORG     66H
                        .PHASE  66H
0066    C3 0212         JP      NMI
                        .DEPHASE
                        ORG     100H
                        .PHASE  100H
0100    31 9FDF         CLDST:  LD      SP,9FDFH
0103    F3                      DI
0104    3E C0                   LD      A,0C0H
0106    D3 F3                   OUT     (CREG),A        ;DISABLE INTERUPTS FROM CRT
0108    ED 56                   IM      1
010A    3E 00                   LD      A,0
010C    32 9FF0                 LD      (CRTRS),A
010F    21 974F                 LD      HL,974FH
0112    22 9FF1                 LD      (RPARM1),HL
0115    21 4D89                 LD      HL,4D89H
0118    22 9FF3                 LD      (RPARM3),HL
011B    3E 20                   LD      A,20H
011D    32 9FF5                 LD      (STRTDIS),A
0120    3E 80                   LD      A,80H
0122    32 9FF6                 LD      (LDCURS),A
0125    21 0000                 LD      HL,0
0128    22 9FF7                 LD      (CURSCOL),HL
012B    21 8000                 LD      HL,8000H
012E    22 9FFA                 LD      (DADDRL),HL
0131    3E 01                   LD      A,1
0133    32 9FF9                 LD      (MODE),A
0136    3E 00                   LD      A,0
0138    32 9FFC                 LD      (DTCNTL),A
013B    3E A0                   LD      A,0A0H
013D    32 9FFD                 LD      (DTCNTH),A
0140    3E 08                   LD      A,8
```

```
0142  32 9FE0                    LD      (TABSP),A
0145  3E 00                      LD      A,0
0147  32 9FE1                    LD      (CRSLN),A
014A  32 9FE2                    LD      (FRMLN),A
014D  32 9FE3                    LD      (CNTRST),A
0150  32 9FE7                    LD      (BUFSIZ),A
0153  3E 1E                      LD      A,1EH
0155  32 9FE8                    LD      (BUFSIZ +1),A
0158  3E 17                      LD      A,23
015A  32 9FE5                    LD      (SCRLPT),A
015D  3E FF                      LD      A,0FFH
015F  32 9FE6                    LD      (ONEPG),A
0162  CD 0742                    CALL    CLRMEM

0165  3A 9FF0          WRMST:   LD      A,(CRTRS)
0168  D3 F3                      OUT     (CREG),A
016A  3A 9FF1                    LD      A,(RPARM1)
016D  D3 F2                      OUT     (PREG),A
016F  3A 9FF2                    LD      A,(RPARM2)
0172  D3 F2                      OUT     (PREG),A
0174  3A 9FF3                    LD      A,(RPARM3)
0177  D3 F2                      OUT     (PREG),A
0179  3A 9FF4                    LD      A,(RPARM4)
017C  D3 F2                      OUT     (PREG),A
017E  3A 9FF6                    LD      A,(LDCURS)
0181  D3 F3                      OUT     (CREG),A
0183  3A 9FF7                    LD      A,(CURSCOL)
0186  D3 F2                      OUT     (PREG),A
0188  3A 9FF8                    LD      A,(CURSROW)
018B  D3 F2                      OUT     (PREG),A
018D  3E 00                      LD      A,0
018F  D3 F8                      OUT     (MODEAD),A
0191  3A 9FFA                    LD      A,(DADDRL)
0194  D3 F0                      OUT     (CHOADD),A
0196  3A 9FFB                    LD      A,(DADDRH)
0199  D3 F0                      OUT     (CHOADD),A
019B  3A 9FFC                    LD      A,(DTCNTL)
019E  D3 F1                      OUT     (CHOTC),A
01A0  3A 9FFD                    LD      A,(DTCNTH)
01A3  D3 F1                      OUT     (CHOTC),A
01A5  3A 9FF9                    LD      A,(MODE)
01A8  D3 F8                      OUT     (MODEAD),A
01AA  3A 9FF5                    LD      A,(STRTDIS)
01AD  D3 F3                      OUT     (CREG),A
01AF  FB                         EI
01B0  C3 01B3                    JP      MNPRG


01B3  CD 03DA           MNPRG:   CALL    GETCHAR
01B6  3A 9FFE                    LD      A,(KBDATA)
01B9  D3 82                      OUT     (82H),A         ;OUTPUT DATA TO CPU
01BB  18 F6                      JR      MNPRG

01BD  CD 03DA           DBUG:    CALL    GETCHAR
01C0  3A 9FFE                    LD      A,(KBDATA)
01C3  21 9FE3                    LD      HL,CNTRST
01C6  CB 5E                      BIT     3,(HL)
01C8  C4 0425                    CALL    NZ,CHRFNCT
01CB  20 40                      JR      NZ,DRES
01CD  FE 03                      CP      CNTC
01CF  C8                         RET     Z
```

```
01D0    FE 05                           CP      CNTE
01D2    CA 0281                         JP      Z,SETUP
01D5    FE 06                           CP      CNTF
01D7    CA 0281                         JP      Z,SETUP
01DA    FE 07                           CP      CNTG
01DC    CA 0281                         JP      Z,SETUP
01DF    FE 10                           CP      CNTP
01E1    CA 0281                         JP      Z,SETUP
01E4    FE 0E                           CP      CNTN
01E6    CA 0281                         JP      Z,SETUP         ;SET SCROLL POINT
01E9    FE 0F                           CP      CNTO
01EB    CA 0281                         JP      Z,SETUP         ;ONE PAGE OPERATION
01EE    FE 19                           CP      CNTY
01F0    CA 0759                         JP      Z,OBSERVE
01F3    FE 1F                           CP      1FH
01F5    FA 0202                         JP      M,CCALL
01F8    CD 0419                         CALL    CNTCHK
01FB    38 0B                           JR      C,DCNT1
01FD    CD 0425                         CALL    CHRFNCT
0200    18 BB                           JR      DBUG
0202    CD 0434                 CCALL:  CALL    CNTFNCT
0205    C3 01BD                         JP      DBUG
0208    CD 03EB                 DCNT1:  CALL    CNTFNT1
020B    18 B0                           JR      DBUG
020D    CB 9E                   DRES:   RES     3,(HL)
020F    C3 01BD                         JP      DBUG

0212    F5                      NMI:    PUSH    AF
0213    DB F3                           IN      A,(SREG)
0215    32 9FE4                         LD      (CRTCST),A
0218    3A 9FF9                         LD      A,(MODE)
021B    D3 F8                           OUT     (MODEAD),A
021D    3A 9FFA                         LD      A,(DADDRL)
0220    D3 F0                           OUT     (CHOADD),A
0222    3A 9FFB                         LD      A,(DADDRH)
0225    D3 F0                           OUT     (CHOADD),A
0227    3A 9FFC                         LD      A,(DTCNTL)
022A    D3 F1                           OUT     (CHOTC),A
022C    3A 9FFD                         LD      A,(DTCNTH)
022F    D3 F1                           OUT     (CHOTC),A
0231    3A 9FF6                         LD      A,(LDCURS)
0234    D3 F3                           OUT     (CREG),A
0236    3A 9FF7                         LD      A,(CURSCOL)
0239    D3 F2                           OUT     (PREG),A
023B    3A 9FF8                         LD      A,(CURSROW)
023E    D3 F2                           OUT     (PREG),A
0240    F1                              POP     AF
0241    ED 45                           RETN

0243    F3                      MASKINT:DI
0244    F5                              PUSH    AF
0245    C5                              PUSH    BC
0246    E5                              PUSH    HL
0247    DB 83                           IN      A,(83H)
0249    ED 4B 9FFE                      LD      BC,(KBDATA)
024D    C5                              PUSH    BC
024E    32 9FFE                         LD      (KBDATA),A
0251    21 9FE3                         LD      HL,CNTRST
0254    CB 5E                           BIT     3,(HL)
0256    C4 0425                         CALL    NZ,CHRFNCT
```

```
0259    20 22                           JR      NZ,RESET
025B    FE 1F                           CP      1FH
025D    FA 026A                         JP      M,CNTCALL
0260    CD 0419                         CALL    CNTCHK
0263    38 13                           JR      C,CTCALL
0265    CD 0425                         CALL    CHRFNCT
0268    18 03                           JR      RESTORE
026A    CD 0434                 CNTCALL:CALL    CNTFNCT
026D    C1                      RESTORE:POP     BC
026E    ED 43 9FFE                      LD      (KBDATA),BC
0272    E1                              POP     HL
0273    C1                              POP     BC
0274    F1                              POP     AF
0275    FB                              EI
0276    ED 4D                           RETI
0278    CD 03EB                 CTCALL: CALL    CNTFNT1
027B    18 F0                           JR      RESTORE
027D    CB 9E                   RESET:  RES     3,(HL)
027F    18 EC                           JR      RESTORE


0281    3A 9FFE                 SETUP:  LD      A,(KBDATA)
0284    FE 05                           CP      CNTE
0286    CA 02DE                         JP      Z,TABSU
0289    FE 06                           CP      CNTF
028B    CA 02ED                         JP      Z,CURSU
028E    FE 07                           CP      CNTG
0290    CA 030B                         JP      Z,HVSU
0293    FE 0E                           CP      CNTN
0295    CA 02A3                         JP      Z,SETSCR
0298    FE 0F                           CP      CNTO
029A    CA 02C5                         JP      Z,SETOPG
029D    FE 10                           CP      CNTP
029F    CA 02CE                         JP      Z,SET3PG
02A2    C9                              RET

02A3    F5                      SETSCR: PUSH    AF
02A4    C5                              PUSH    BC
02A5    CD 03DA                         CALL    GETCHAR
02A8    3A 9FFE                         LD      A,(KBDATA)
02AB    E6 03                           AND     03H             ;GET MSD OF BCD DIGIT
02AD    CB 27                           SLA     A
02AF    47                              LD      B,A             ;MULTIPLY BY 10
02B0    CB 27                           SLA     A
02B2    CB 27                           SLA     A
02B4    80                              ADD     A,B
02B5    47                              LD      B,A
02B6    CD 03DA                         CALL    GETCHAR         ;GET LSD
02B9    3A 9FFE                         LD      A,(KBDATA)
02BC    E6 0F                           AND     0FH
02BE    80                              ADD     A,B
02BF    32 9FE5                         LD      (SCRLPT),A
02C2    C1                              POP     BC
02C3    F1                              POP     AF
02C4    C9                              RET

02C5    F5                      SETOPG: PUSH    AF
02C6    E5                              PUSH    HL
```

```
02C7    3E FF                       LD      A,0FFH
02C9    21 0780                     LD      HL,780H
02CC    18 07                       JR      SETRPM
02CE    F5                  SET3PG: PUSH    AF
02CF    E5                          PUSH    HL
02D0    3E 00                       LD      A,0
02D2    21 1E00                     LD      HL,1E00H
02D5    32 9FE6             SETRPM: LD      (ONEPG),A
02D8    22 9FE7                     LD      (BUFSIZ),HL
02DB    E1                          POP     HL
02DC    F1                          POP     AF
02DD    C9                          RET


02DE    CD 03DA             TABSU:  CALL    GETCHAR
02E1    3A 9FFE                     LD      A,(KBDATA)
02E4    D6 30                       SUB     30H                 ;CONVERT FROM ASCII
02E6    32 9FE0                     LD      (TABSP),A
02E9    CD 0850                     CALL    RPMADJ
02EC    C9                          RET

02ED    CD 03DA             CURSU:  CALL    GETCHAR
02F0    3A 9FFE                     LD      A,(KBDATA)
02F3    E6 03                       AND     03
02F5    47                          LD      B,A
02F6    CB 20                       SLA     B
02F8    CB 20                       SLA     B
02FA    CB 20                       SLA     B
02FC    CB 20                       SLA     B
02FE    3A 9FF4                     LD      A,(RPARM4)
0301    E6 CF                       AND     0CFH
0303    B0                          OR      B
0304    32 9FF4                     LD      (RPARM4),A
0307    CD 0850                     CALL    RPMADJ
030A    C9                          RET

030B    CD 03DA             HVSU:   CALL    GETCHAR         ;GET LSD FOR HORZ CHAR/ROW
030E    3A 9FFE                     LD      A,(KBDATA)
0311    E6 0F                       AND     0FH
0313    47                          LD      B,A
0314    CD 03DA                     CALL    GETCHAR         ;GET MSD FOR HORZ CHAR/ROW
0317    3A 9FFE                     LD      A,(KBDATA)      ;ALSO MSB=0 FOR NORM ROWS
031A    CB 27                       SLA     A               ;AND =1 FOR SPACE ROWS
031C    CB 27                       SLA     A
031E    CB 27                       SLA     A
0320    CB 27                       SLA     A
0322    B0                          OR      B
0323    32 9FF1                     LD      (RPARM1),A
0326    CD 03DA                     CALL    GETCHAR         ;GET VERT RETRACE ROW COUNT
0329    3A 9FFE                     LD      A,(KBDATA)      ;FROM 0-3
032C    FE 20                       CP      SPACE
```

```
032E  CC 0850                 CALL    Z,RPMADJ
0331  C8                     RET     Z
0332  E6 03                  AND     03
0334  0F                     RRCA
0335  0F                     RRCA
0336  47                     LD      B,A
0337  3A 9FF2                LD      A,(RPARM2)
033A  E6 3F                  AND     3FH
033C  B0                     OR      B
033D  32 9FF2                LD      (RPARM2),A
0340  CD 03DA                CALL    GETCHAR         ;GET LSD FOR VERT ROW/FRAME
0343  3A 9FFE                LD      A,(KBDATA)      ;FROM 0-F
0346  FE 20                  CP      SPACE
0348  CC 0850                CALL    Z,RPMADJ
034B  C8                     RET     Z
034C  E6 0F                  AND     0FH
034E  47                     LD      B,A
034F  CD 03DA                CALL    GETCHAR         ;GET MSD FOR VERT ROW/FRM
0352  3A 9FFE                LD      A,(KBDATA)      ;FROM 1-4
0355  E6 07                  AND     07H
0357  CB 27                  SLA     A
0359  CB 27                  SLA     A
035B  CB 27                  SLA     A
035D  CB 27                  SLA     A
035F  B0                     OR      B
0360  D6 01                  SUB     1
0362  47                     LD      B,A
0363  3A 9FF2                LD      A,(RPARM2)
0366  E6 C0                  AND     0C0H
0368  B0                     OR      B
0369  32 9FF2                LD      (RPARM2),A
036C  CD 03DA                CALL    GETCHAR         ;GET HEX DIG FOR UNDERLINE PLACEMENT
036F  3A 9FFE                LD      A,(KBDATA)      ;FROM 0-F
0372  FE 20                  CP      SPACE
0374  CC 0850                CALL    Z,RPMADJ
0377  C8                     RET     Z
0378  E6 0F                  AND     0FH
037A  47                     LD      B,A
037B  3A 9FF3                LD      A,(RPARM3)
037E  E6 0F                  AND     0FH
0380  CB 20                  SLA     B
0382  CB 20                  SLA     B
0384  CB 20                  SLA     B
0386  CB 20                  SLA     B
0388  B0                     OR      B
0389  32 9FF3                LD      (RPARM3),A
038C  CD 03DA                CALL    GETCHAR         ;GET HEX DIG FOR NUMBER OF LINES/ROW
038F  3A 9FFE                LD      A,(KBDATA)      ;FROM 0-F
0392  FE 20                  CP      SPACE
0394  CC 0850                CALL    Z,RPMADJ
0397  C8                     RET     Z
0398  E6 0F                  AND     0FH
039A  47                     LD      B,A
039B  3A 9FF3                LD      A,(RPARM3)
039E  E6 F0                  AND     0F0H
03A0  B0                     OR      B
03A1  32 9FF3                LD      (RPARM3),A
03A4  CD 03DA                CALL    GETCHAR         ;GET HORIZ RETRACE COUNT
03A7  3A 9FFE                LD      A,(KBDATA)      ;FROM 2,32 IN INCREMENTS OF 2
03AA  FE 20                  CP      SPACE
03AC  CC 0850                CALL    Z,RPMADJ
```

```
03AF   C8                        RET     Z
03B0   E6 0F                     AND     0FH
03B2   47                        LD      B,A
03B3   3A 9FF4                   LD      A,(RPARM4)
03B6   E6 F0                     AND     0F0H
03B8   B0                        OR      B
03B9   32 9FF4                   LD      (RPARM4),A
03BC   CD 03DA                   CALL    GETCHAR         ;GET FIELD ATTRIBUTE AND LINE
03BF   3A 9FFE                   LD      A,(KBDATA)      ;COUNTER MODE:
03C2   FE 20                     CP      SPACE           ;0=NON-OFFSET LINE CNTR MD, TRANSPARENT FLD
03C4   CC 0850                   CALL    Z,RPMADJ        ;1=NON-OFFSET LINE CNTR MD, NON-TRANSPARENT FLD
03C7   C8                        RET     Z
03C8   E6 03                     AND     03              ;2=OFFSET LINE COUNTER MD,  TRANSPARENT FLD
03CA   0F                        RRCA                    ;3=OFFSET LINE COUNTER MD,  NON-TRANSPARENT FLD
03CB   0F                        RRCA
03CC   47                        LD      B,A
03CD   3A 9FF4                   LD      A,(RPARM4)
03D0   E6 3F                     AND     3FH
03D2   B0                        OR      B
03D3   32 9FF4                   LD      (RPARM4),A
03D6   CD 0850                   CALL    RPMADJ
03D9   C9                        RET

03DA   F5                GETCHAR:PUSH    AF
03DB   DB 80             LOOP:   IN      A,(KBPOLL)
03DD   CB 47                     BIT     0,A
03DF   C2 03E4                   JP      NZ,DATAVAL
03E2   18 F7                     JR      LOOP
03E4   DB 81             DATAVAL:IN      A,(KBREAD)
03E6   32 9FFE                   LD      (KBDATA),A
03E9   F1                        POP     AF
03EA   C9                        RET


03EB   F5                CNTFNT1:PUSH    AF
03EC   3A 9FE3                   LD      A,(CNTRST)
03EF   CB 57                     BIT     2,A
03F1   20 18                     JR      NZ,CNT2
03F3   CB 6F                     BIT     5,A
03F5   20 0D                     JR      NZ,CNT1
03F7   CB 4F                     BIT     1,A
03F9   20 02                     JR      NZ,CNT0

03FB   F1                        POP     AF
03FC   C9                        RET

03FD   CB EF             CNT0:   SET     5,A
03FF   CD 06BB                   CALL    ESCAP0
0402   18 10                     JR      CNTREST

0404   CD 06D4           CNT1:   CALL    ESCAP2
0407   CB D7                     SET     2,A
0409   18 09                     JR      CNTREST

040B   CD 06C9           CNT2:   CALL    ESCAP1
040E   CB AF                     RES     5,A
0410   CB 97                     RES     2,A
0412   CB 8F                     RES     1,A
0414   32 9FE3           CNTREST:LD      (CNTRST),A
0417   F1                        POP     AF
0418   C9                        RET
```

```
0419  3A 9FE3          CNTCHK: LD      A,(CNTRST)
041C  CB 4F                   BIT     1,A
041E  20 03                   JR      NZ,BITSET
0420  37                     SCF
0421  3F                     CCF
0422  C9                     RET

0423  37               BITSET: SCF
0424  C9                     RET

0425  F5               CHRFNCT:PUSH    AF
0426  E5                     PUSH    HL
0427  3A 9FFE                LD      A,(KBDATA)
042A  CD 070F                CALL    CURSPOS
042D  77                     LD      (HL),A
042E  CD 072B                CALL    INCCURS
0431  E1                     POP     HL
0432  F1                     POP     AF
0433  C9                     RET

0434  3A 9FFE          CNTFNCT:LD      A,(KBDATA)
0437  FE 01                   CP      CNTA
0439  CA 04BA                JP      Z,SCRL
043C  FE 02                   CP      CNTB
043E  CA 051E                JP      Z,SCRLD
0441  FE 04                   CP      CNTD
0443  CA 01BD                JP      Z,DBUG
0446  FE 08                   CP      CNTH
0448  CA 0544                JP      Z,BS
044B  FE 09                   CP      CNTI
044D  CA 0554                JP      Z,TAB
0450  FE 0A                   CP      CNTJ
0452  CA 0578                JP      Z,LF
0455  FE 0B                   CP      CNTK
0457  CA 05E1                JP      Z,UPLINE
045A  FE 0C                   CP      CNTL
045C  CA 0620                JP      Z,FORSPCE
045F  FE 0D                   CP      CNTM
0461  CA 062F                JP      Z,CR
0464  FE 0E                   CP      CNTN
0466  CA 0637                JP      Z,BLINK
0469  FE 0F                   CP      CNTO
046B  CA 063F                JP      Z,RVV
046E  FE 10                   CP      CNTP
0470  CA 0647                JP      Z,UNDLINE
0473  FE 11                   CP      CNTQ
0475  CA 064F                JP      Z,HGLT
0478  FE 12                   CP      CNTR
047A  CA 065F                JP      Z,CONCAT
047D  FE 13                   CP      CNTS
047F  CA 0657                JP      Z,STOPATT
0482  FE 14                   CP      CNTT
0484  CA 0799                JP      Z,CRLF
0487  FE 15                   CP      CNTU
0489  CA 07BF                JP      Z,PGO
048C  FE 16                   CP      CNTV
048E  CA 07E0                JP      Z,CURREQ
0491  FE 17                   CP      CNTW
```

```
0493  CA 07D8                    JP      Z,DISCNT
0496  FE 18                      CP      CNTX
0498  CA 07F9                    JP      Z,RESSTAT
049B  FE 1A                      CP      CNTZ
049D  CA 067A                    JP      Z,CLEAR
04A0  FE 1E                      CP      CNTTIL
04A2  CA 069F                    JP      Z,HOME
04A5  FE 1B                      CP      ESC
04A7  CA 06B0                    JP      Z,ESCAPE
04AA  FE 19                      CP      CNTY
04AC  CA 0801                    JP      Z,CURCHAR
04AF  FE 1D                      CP      CNTSQB
04B1  CA 0814                    JP      Z,STSPGPH
04B4  FE 1C                      CP      CNTBSL
04B6  CA 080C                    JP      Z,GRAPHIC
04B9  C9                         RET

04BA  F5                 SCRL:   PUSH    AF
04BB  E5                         PUSH    HL
04BC  D5                         PUSH    DE
04BD  3A 9FE6                    LD      A,(ONEPG)       ;DISABLE SCROLLING FOR ONE PAGE OPERATION
04C0  FE FF                      CP      0FFH
04C2  CA 0502                    JP      Z,SCRRST
04C5  21 9500                    LD      HL,9500H
04C8  ED 5B 9FFA                 LD      DE,(DADDRL)
04CC  ED 52                      SBC     HL,DE
04CE  FA 0502                    JP      M,SCRRST
04D1  2A 9FFA                    LD      HL,(DADDRL)
04D4  11 03C0                    LD      DE,12*80
04D7  19                         ADD     HL,DE
04D8  22 9FFA                    LD      (DADDRL),HL
04DB  3A 9FE2                    LD      A,(FRMLN)
04DE  C6 0C                      ADD     A,12
04E0  32 9FE2                    LD      (FRMLN),A
04E3  CD 0506            SCRCOM: CALL    CHKCURS
04E6  30 0F                      JR      NC,CRSONSC
04E8  3A 9FE2                    LD      A,(FRMLN)
04EB  C6 0C                      ADD     A,12
04ED  32 9FE1                    LD      (CURSLN),A
04F0  3E 0C                      LD      A,12
04F2  32 9FF8                    LD      (CURSROW),A
04F5  18 0B                      JR      SCRRST
04F7  3A 9FE2            CRSONSC:LD      A,(FRMLN)
04FA  67                         LD      H,A
04FB  3A 9FE1                    LD      A,(CURSLN)
04FE  94                         SUB     H
04FF  32 9FF8                    LD      (CURSROW),A
0502  D1                 SCRRST: POP     DE
0503  E1                         POP     HL
0504  F1                         POP     AF
0505  C9                         RET

0506  E5                 CHKCURS:PUSH    HL
0507  3A 9FE2                    LD      A,(FRMLN)
050A  67                         LD      H,A
050B  3A 9FE1                    LD      A,(CURSLN)
050E  94                         SUB     H
050F  FA 051B                    JP      M,SETCF
0512  FE 18                      CP      24
0514  F2 051B                    JP      P,SETCF
```

```
0517  37                        SCF
0518  3F                        CCF
0519  18 01                     JR      CHKRST
051B  37              SETCF:    SCF
051C  E1              CHKRST:   POP     HL
051D  C9                        RET


051E  F5              SCRLD:    PUSH    AF
051F  E5                        PUSH    HL
0520  D5                        PUSH    DE
0521  11 8000                   LD      DE,8000H
0524  2A 9FFA                   LD      HL,(DADDRL)
0527  ED 52                     SBC     HL,DE
0529  11 03C0                   LD      DE,12*80
052C  ED 52                     SBC     HL,DE
052E  FA 0502                   JP      M,SCRRST
0531  2A 9FFA                   LD      HL,(DADDRL)
0534  ED 52                     SBC     HL,DE
0536  22 9FFA                   LD      (DADDRL),HL
0539  3A 9FE2                   LD      A,(FRMLN)
053C  D6 0C                     SUB     12
053E  32 9FE2                   LD      (FRMLN),A
0541  C3 04E3                   JP      SCRCOM


0544  F5              BS:       PUSH    AF
0545  3A 9FF7                   LD      A,(CURSCOL)
0548  FE 00                     CP      0
054A  CA 0552                   JP      Z,BSRST
054D  D6 01                     SUB     1
054F  32 9FF7                   LD      (CURSCOL),A
0552  F1              BSRST:    POP     AF
0553  C9                        RET

0554  F5              TAB:      PUSH    AF
0555  E5                        PUSH    HL
0556  C5                        PUSH    BC
0557  3A 9FE0                   LD      A,(TABSP)
055A  47                        LD      B,A
055B  21 9FF7                   LD      HL,CURSCOL
055E  3E 00                     LD      A,0
0560  80              TABLOOP:  ADD     A,B
0561  BE                       CP      (HL)
0562  FA 0560                   JP      M,TABLOOP
0565  CA 0560                   JP      Z,TABLOOP
0568  FE 50                     CP      80
056A  FA 0571                   JP      M,TABOK
056D  21 9FE0                   LD      HL,TABSP
0570  96                        SUB     (HL)
0571  32 9FF7              TABOK:   LD      (CURSCOL),A
0574  C1                        POP     BC
0575  E1                        POP     HL
0576  F1                        POP     AF
0577  C9                        RET

0578  F5              LF:       PUSH    AF
0579  E5                        PUSH    HL
057A  D5                        PUSH    DE
057B  C5                        PUSH    BC
```

```
057C  3A 9FF8                     LD      A,(CURSROW)
057F  21 9FE5                     LD      HL,SCRLPT
0582  BE                          CP      (HL)            ;SHOULD A SCROLL OCCUR
0583  F2 0596                     JP      P,LSCR1
0586  3C                          INC     A
0587  32 9FF8                     LD      (CURSROW),A
058A  3A 9FE1               LUPDATE:LD    A,(CURSLN)
058D  3C                          INC     A
058E  32 9FE1                     LD      (CURSLN),A
0591  C1                          POP     BC
0592  D1                          POP     DE
0593  E1                          POP     HL
0594  F1                          POP     AF
0595  C9                          RET
0596  3A 9FE6               LSCR1:  LD    A,(ONEPG)
0599  FE FF                      CP      0FFH
059B  CA 05B9                     JP      Z,LBLKMV

059E  2A 9FFA                     LD      HL,(DADDRL)
05A1  11 0780                     LD      DE,24*80
05A4  19                          ADD     HL,DE
05A5  11 9E00                     LD      DE,9E00H
05A8  ED 52                       SBC     HL,DE
05AA  F2 05B9                     JP      P,LBLKMV
05AD  11 0050                     LD      DE,80
05B0  2A 9FFA                     LD      HL,(DADDRL)
05B3  19                          ADD     HL,DE
05B4  22 9FFA                     LD      (DADDRL),HL
05B7  18 D1                       JR      LUPDATE
05B9  21 8050               LBLKMV: LD    HL,8000H+50H
05BC  11 8000                     LD      DE,8000H
05BF  ED 4B 9FE7                  LD      BC,(BUFSIZ)
05C3  ED B0                       LDIR
05C5  2A 9FFA                     LD      HL,(DADDRL)     ;CALCULATE ABSOLUTE CURSOR ROW AND
05C8  11 0050                     LD      DE,80           ;CLEAR THAT ROW
05CB  3A 9FF8                     LD      A,(CURSROW)
05CE  19                    LMULT:  ADD     HL,DE
05CF  3D                          DEC     A
05D0  20 FC                       JR      NZ,LMULT
05D2  36 20                       LD      (HL),20H
05D4  E5                          PUSH    HL
05D5  D1                          POP     DE
05D6  13                          INC     DE
05D7  01 004F                     LD      BC,79
05DA  ED B0                       LDIR
05DC  C1                          POP     BC
05DD  D1                          POP     DE
05DE  E1                          POP     HL
05DF  F1                          POP     AF
05E0  C9                          RET

05E1  F5                    UPLINE: PUSH    AF
05E2  E5                          PUSH    HL
05E3  D5                          PUSH    DE
05E4  3A 9FF8                     LD      A,(CURSROW)
05E7  FE 00                       CP      0
05E9  28 0F                       JR      Z,UCHGFRM
05EB  3D                          DEC     A
05EC  32 9FF8                     LD      (CURSROW),A
05EF  3A 9FE1                     LD      A,(CRSLN)
05F2  3D                          DEC     A
```

```
05F3  32 9FE1                     LD      (CRSLN),A
05F6  D1                  URST:   POP     DE
05F7  E1                          POP     HL
05F8  F1                          POP     AF
05F9  C9                          RET
05FA  2A 9FFA             UCHGFRM:LD      HL,(DADDRL)
05FD  11 8001                     LD      DE,8001H
0600  ED 52                       SBC     HL,DE
0602  FA 05F6                     JP      M,URST
0605  2A 9FFA                     LD      HL,(DADDRL)
0608  11 0050                     LD      DE,80
060B  ED 52                       SBC     HL,DE
060D  22 9FFA                     LD      (DADDRL),HL
0610  3A 9FE1                     LD      A,(CURSLN)
0613  3D                         DEC     A
0614  32 9FE1                     LD      (CURSLN),A
0617  3A 9FE2                     LD      A,(FRMLN)
061A  3D                         DEC     A
061B  32 9FE2                     LD      (FRMLN),A
061E  18 D6                       JR      URST

0620  F5                  FORSPCE:PUSH    AF
0621  3A 9FF7                     LD      A,(CURSCOL)
0624  FE 4F                       CP      79
0626  CA 062D                     JP      Z,FORRST
0629  3C                         INC     A
062A  32 9FF7                     LD      (CURSCOL),A
062D  F1                  FORRST: POP     AF
062E  C9                          RET

062F  F5                  CR:     PUSH    AF
0630  3E 00                       LD      A,0
0632  32 9FF7                     LD      (CURSCOL),A
0635  F1                          POP     AF
0636  C9                          RET

0637  F5                  BLINK:  PUSH    AF
0638  3E 82                       LD      A,82H
063A  CD 06E8                     CALL    SETATT
063D  F1                          POP     AF
063E  C9                          RET

063F  F5                  RVV:    PUSH    AF
0640  3E 90                       LD      A,90H
0642  CD 06E8                     CALL    SETATT
0645  F1                          POP     AF
0646  C9                          RET

0647  F5                  UNDLINE:PUSH    AF
0648  3E A0                       LD      A,0A0H
064A  CD 06E8                     CALL    SETATT
064D  F1                          POP     AF
064E  C9                          RET

064F  F5                  HGLT:   PUSH    AF
0650  3E 81                       LD      A,81H
0652  CD 06E8                     CALL    SETATT
0655  F1                          POP     AF
0656  C9                          RET

0657  F5                  STOPATT:PUSH    AF
```

```
0658    3E 7F                       LD      A,7FH
065A    CD 0705                     CALL    RESATT
065D    F1                          POP     AF
065E    C9                          RET

065F    F5                  CONCAT: PUSH    AF
0660    E5                          PUSH    HL
0661    21 9FE3                     LD      HL,CNTRST
0664    CB 46                       BIT     0,(HL)
0666    28 07                       JR      Z,CSET
0668    CB 86                       RES     0,(HL)
066A    CD 072B                     CALL    INCCURS
066D    18 08                       JR      CONRST
066F    CB C6              CSET:    SET     0,(HL)
0671    CD 070F                     CALL    CURSPOS
0674    3E 00                       LD      A,0
0676    77                          LD      (HL),A
0677    E1                  CONRST: POP     HL
0678    F1                          POP     AF
0679    C9                          RET

067A    F5                  CLEAR:  PUSH    AF
067B    E5                          PUSH    HL
067C    D5                          PUSH    DE
067D    C5                          PUSH    BC
067E    3E 00                       LD      A,0
0680    32 9FF7                     LD      (CURSCOL),A
0683    32 9FF8                     LD      (CURSROW),A
0686    3A 9FE2                     LD      A,(FRMLN)
0689    32 9FE1                     LD      (CURSLN),A
068C    2A 9FFA                     LD      HL,(DADDRL)
068F    E5                          PUSH    HL
0690    D1                          POP     DE
0691    13                          INC     DE
0692    3E 20                       LD      A,20H
0694    77                          LD      (HL),A
0695    01 077F                     LD      BC,80*24-1
0698    ED B0                       LDIR
069A    C1                          POP     BC
069B    D1                          POP     DE
069C    E1                          POP     HL
069D    F1                          POP     AF
069E    C9                          RET

069F    F5                  HOME:   PUSH    AF
06A0    3E 00                       LD      A,0
06A2    32 9FF8                     LD      (CURSROW),A
06A5    32 9FF7                     LD      (CURSCOL),A
06A8    3A 9FE2                     LD      A,(FRMLN)
06AB    32 9FE1                     LD      (CURSLN),A
06AE    F1                          POP     AF
06AF    C9                          RET

06B0    F5                  ESCAPE: PUSH    AF
06B1    3A 9FE3                     LD      A,(CNTRST)
06B4    CB CF                       SET     1,A
06B6    32 9FE3                     LD      (CNTRST),A
06B9    F1                          POP     AF
06BA    C9                          RET

06BB    F5                  ESCAPO: PUSH    AF
```

```
06BC  3A 9FFE                     LD      A,(KBDATA)
06BF  FE 3D                       CP      3DH             ;IS THIS CHAR A @=@
06C1  28 04                       JR      Z,ESCOR
06C3  F1                          POP     AF
06C4  CB AF                       RES     5,A             ;INDICATE THAT A = HAS NOT BEEN
06C6  F5                          PUSH    AF              ;FOUND
06C7  F1                ESCOR:    POP     AF
06C8  C9                          RET


06C9  F5                ESCAP1:   PUSH    AF
06CA  3A 9FFE                     LD      A,(KBDATA)
06CD  D6 20                       SUB     20H
06CF  32 9FF7                     LD      (CURSCOL),A
06D2  F1                          POP     AF
06D3  C9                          RET

06D4  F5                ESCAP2:   PUSH    AF
06D5  E5                          PUSH    HL
06D6  3A 9FFE                     LD      A,(KBDATA)
06D9  D6 20                       SUB     20H
06DB  32 9FF8                     LD      (CURSROW),A
06DE  21 9FE2                     LD      HL,FRMLN
06E1  86                          ADD     A,(HL)
06E2  32 9FE1                     LD      (CURSLN),A
06E5  E1                          POP     HL
06E6  F1                          POP     AF
06E7  C9                          RET

06E8  E5                SETATT:   PUSH    HL
06E9  D5                          PUSH    DE
06EA  CD 070F                     CALL    CURSPOS
06ED  57                          LD      D,A
06EE  3A 9FE3                     LD      A,(CNTRST)
06F1  CB 47                       BIT     0,A
06F3  7A                          LD      A,D
06F4  28 01                       JR      Z,ATTCOM
06F6  B6                          OR      (HL)
06F7  77                ATTCOM:   LD      (HL),A
06F8  3A 9FE3                     LD      A,(CNTRST)
06FB  CB 47                       BIT     0,A
06FD  20 03                       JR      NZ,SKIP
06FF  CD 072B                     CALL    INCCURS
0702  D1                SKIP:     POP     DE
0703  E1                          POP     HL
0704  C9                          RET

0705  E5                RESATT:   PUSH    HL
0706  D5                          PUSH    DE
0707  CD 070F                     CALL    CURSPOS
070A  2F                          CPL
070B  77                          LD      (HL),A
070C  C3 06F7                     JP      ATTCOM

070F  F5                CURSPOS:  PUSH    AF
0710  D5                          PUSH    DE
0711  2A 9FFA                     LD      HL,(DADDRL)
0714  3A 9FF8                     LD      A,(CURSROW)
0717  11 0050                     LD      DE,80
071A  3D                CURLOOP:  DEC     A
```

```
071B    FA 0721                         JP      M,CURFINI
071E    19                              ADD     HL,DE
071F    18 F9                           JR      CURLOOP
0721    ED 5B 9FF7          CURFINI:    LD      DE,(CURSCOL)
0725    16 00                           LD      D,0
0727    19                              ADD     HL,DE
0728    D1                              POP     DE
0729    F1                              POP     AF
072A    C9                              RET

072B    F5                  INCCURS:    PUSH    AF
072C    3A 9FF7                         LD      A,(CURSCOL)
072F    FE 4F                           CP      79
0731    F2 073A                         JP      P,NEWLINE
0734    3C                              INC     A
0735    32 9FF7                         LD      (CURSCOL),A
0738    F1                  INCRST:     POP     AF
0739    C9                              RET
073A    CD 062F             NEWLINE:    CALL    CR
073D    CD 0578                         CALL    LF
0740    18 F6                           JR      INCRST


0742    F5                  CLRMEM:     PUSH    AF
0743    E5                              PUSH    HL
0744    D5                              PUSH    DE
0745    C5                              PUSH    BC
0746    21 8000                         LD      HL,8000H
0749    E5                              PUSH    HL
074A    D1                              POP     DE
074B    3E 20                           LD      A,SPACE
074D    77                              LD      (HL),A
074E    13                              INC     DE
074F    01 1E00                         LD      BC,1E00H
0752    ED B0                           LDIR
0754    C1                              POP     BC
0755    D1                              POP     DE
0756    E1                              POP     HL
0757    F1                              POP     AF
0758    C9                              RET

0759    CD 03DA             OBSERVE:    CALL    GETCHAR
075C    3A 9FFE                         LD      A,(KBDATA)
075F    E6 0F                           AND     0FH
0761    CB 27                           SLA     A
0763    CB 27                           SLA     A
0765    CB 27                           SLA     A
0767    CB 27                           SLA     A
0769    26 00                           LD      H,0
076B    67                              LD      H,A
076C    CD 03DA                         CALL    GETCHAR
076F    3A 9FFE                         LD      A,(KBDATA)
0772    E6 0F                           AND     0FH
0774    B4                              OR      H
0775    67                              LD      H,A
0776    CD 03DA                         CALL    GETCHAR
0779    3A 9FFE                         LD      A,(KBDATA)
077C    E6 0F                           AND     0FH
077E    CB 27                           SLA     A
0780    CB 27                           SLA     A
0782    CB 27                           SLA     A
```

```
0784    CB 27                       SLA     A
0786    2E 00                       LD      L,0
0788    6F                          LD      L,A
0789    CD 03DA                     CALL    GETCHAR
078C    3A 9FFE                     LD      A,(KBDATA)
078F    E6 0F                       AND     0FH
0791    B5                          OR      L
0792    6F                          LD      L,A
0793    7E                          LD      A,(HL)
0794    D3 82                       OUT     (82H),A
0796    C3 0165                     JP      WRMST

0799    F5                  CRLF:   PUSH    AF
079A    E5                          PUSH    HL
079B    D5                          PUSH    DE
079C    C5                          PUSH    BC
079D    3A 9FF7                     LD      A,(CURSCOL)
07A0    47                          LD      B,A
07A1    CD 070F                     CALL    CURSPOS
07A4    3E 4F                       LD      A,79
07A6    90                          SUB     B
07A7    28 0B                       JR      Z,CCRLF
07A9    4F                          LD      C,A
07AA    06 00                       LD      B,0
07AC    E5                          PUSH    HL
07AD    D1                          POP     DE
07AE    13                          INC     DE
07AF    3E 20                       LD      A,SPACE
07B1    77                          LD      (HL),A
07B2    ED B0                       LDIR
07B4    CD 062F             CCRLF:  CALL    CR
07B7    CD 0578                     CALL    LF
07BA    C1                          POP     BC
07BB    D1                          POP     DE
07BC    E1                          POP     HL
07BD    F1                          POP     AF
07BE    C9                          RET


07BF    F5                  PGO:    PUSH    AF
07C0    E5                          PUSH    HL
07C1    3E 00                       LD      A,0
07C3    32 9FE1                     LD      (CURSLN),A
07C6    32 9FE2                     LD      (FRMLN),A
07C9    32 9FF8                     LD      (CURSROW),A
07CC    32 9FF7                     LD      (CURSCOL),A
07CF    21 8000                     LD      HL,8000H
07D2    22 9FFA                     LD      (DADDRL),HL
07D5    E1                          POP     HL
07D6    F1                          POP     AF
07D7    C9                          RET


07D8    E5                  DISCNT: PUSH    HL
07D9    21 9FE3                     LD      HL,CNTRST
07DC    CB DE                       SET     3,(HL)
07DE    E1                          POP     HL
07DF    C9                          RET
```

```
07E0  F5                  CURREQ: PUSH    AF
07E1  DB 80                       IN      A,(KBPOLL)
07E3  CB 4F                       BIT     1,A
07E5  20 F9                       JR      NZ,CURREQ
07E7  3A 9FF7                     LD      A,(CURSCOL)
07EA  D3 82                       OUT     (82H),A
07EC  DB 80               CURLP:  IN      A,(KBPOLL)
07EE  CB 4F                       BIT     1,A
07F0  20 FA                       JR      NZ,CURLP
07F2  3A 9FF8                     LD      A,(CURSROW)
07F5  D3 82                       OUT     (82H),A
07F7  F1                          POP     AF
07F8  C9                          RET

07F9  F5                  RESSTAT:PUSH    AF              ;RESETS CONCAT AND CURSOR PLACEMENT TO
07FA  3E 00                       LD      A,0             ;COMMAND MODE
07FC  32 9FE3                     LD      (CNTRST),A
07FF  F1                          POP     AF
0800  C9                          RET

0801  F5                  CURCHAR:PUSH    AF
0802  E5                          PUSH    HL
0803  CD 070F                     CALL    CURSPOS
0806  7E                          LD      A,(HL)
0807  D3 82                       OUT     (82H),A
0809  E1                          POP     HL
080A  F1                          POP     AF
080B  C9                          RET

080C  F5                  GRAPHIC:PUSH    AF
080D  3E 84                       LD      A,84H
080F  CD 06E8                     CALL    SETATT
0812  F1                          POP     AF
0813  C9                          RET

0814  F5                  STSPGPH:PUSH    AF
0815  E5                          PUSH    HL
0816  21 9FE3                     LD      HL,CNTRST
0819  CB 66                       BIT     4,(HL)
081B  C2 0836                     JP      NZ,SPGPH
081E  CB E6                       SET     4,(HL)
0820  3A 9FF3                     LD      A,(RPARM3)
0823  E6 0F                       AND     0FH
0825  C6 70                       ADD     A,70H
0827  32 9FF3                     LD      (RPARM3),A
082A  3A 9FF4                     LD      A,(RPARM4)
082D  E6 CF                       AND     0CFH
082F  C6 20                       ADD     A,20H
0831  32 9FF4                     LD      (RPARM4),A
0834  18 14                       JR      STSPRST
0836  CB A6               SPGPH:  RES     4,(HL)
0838  3A 9FF3                     LD      A,(RPARM3)
083B  E6 0F                       AND     0FH
083D  C6 80                       ADD     A,80H
083F  32 9FF3                     LD      (RPARM3),A
0842  3A 9FF4                     LD      A,(RPARM4)
0845  E6 CF                       AND     0CFH
0847  32 9FF4                     LD      (RPARM4),A
084A  CD 0850             STSPRST:CALL    RPMADJ
084D  E1                          POP     HL
```

```
084E    F1                          POP     AF
084F    C9                          RET


0850    F5                  RPMADJ: PUSH    AF
0851    3A 9FF0                     LD      A,(CRTRS)
0854    D3 F3                       OUT     (CREG),A
0856    3A 9FF1                     LD      A,(RPARM1)
0859    D3 F2                       OUT     (PREG),A
085B    3A 9FF2                     LD      A,(RPARM2)
085E    D3 F2                       OUT     (PREG),A
0860    3A 9FF3                     LD      A,(RPARM3)
0863    D3 F2                       OUT     (PREG),A
0865    3A 9FF4                     LD      A,(RPARM4)
0868    D3 F2                       OUT     (PREG),A
086A    3A 9FF5                     LD      A,(STRTDIS)
086D    D3 F3                       OUT     (CREG),A
086F    F1                          POP     AF
0870    C9                          RET

                                    END
```

MACROS:

SYMBOLS:

ATTCOM	06F7	BITSET	0423	BLINK	0637	BLUE	00E4
BS	0544	BSRST	0552	BUFSIZ	9FE7	CCALL	0202
CCRLF	07B4	CHOADD	00F0	CHOTC	00F1	CHKCUR	0506
CHKRST	051C	CHRFNC	0425	CLDST	0100	CLEAR	067A
CLRMEM	0742	CNT0	03FD	CNT1	0404	CNT2	040B
CNTA	0001	CNTB	0002	CNTBSL	001C	CNTC	0003
CNTCAL	026A	CNTCHK	0419	CNTD	0004	CNTE	0005
CNTF	0006	CNTFNC	0434	CNTFNT	03EB	CNTG	0007
CNTH	0008	CNTI	0009	CNTJ	000A	CNTK	000B
CNTL	000C	CNTM	000D	CNTN	000E	CNTO	000F
CNTP	0010	CNTQ	0011	CNTR	0012	CNTRES	0414
CNTRST	9FE3	CNTS	0013	CNTSQB	001D	CNTT	0014
CNTTIL	001E	CNTU	0015	CNTV	0016	CNTW	0017
CNTX	0018	CNTY	0019	CNTZ	001A	CONCAT	065F
CONRST	0677	CPUDAT	9FFF	CR	062F	CREG	00F3
CRLF	0799	CRSLN	9FE1	CRSONS	04F7	CRTCST	9FE4
CRTRS	9FF0	CSET	066F	CTCALL	0278	CURCHA	0801
CURFIN	0721	CURLOO	071A	CURLP	07EC	CURREQ	07E0
CURSCO	9FF7	CURSLN	9FE1	CURSPO	070F	CURSRO	9FF8
CURSU	02ED	DADDRH	9FFB	DADDRL	9FFA	DATAVA	03E4
DBUG	01BD	DCNT1	0208	DISCNT	07D8	DISINT	00C0
DRES	020D	DTCNTH	9FFD	DTCNTL	9FFC	ENINT	00A0
ESC	001B	ESCOR	06C7	ESCAPO	06BB	ESCAP1	06C9
ESCAP2	06D4	ESCAPE	06B0	FORRST	062D	FORSPC	0620
FRMLN	9FE2	GETCHA	03DA	GRAPHI	080C	GREEN	00E1
HOLT	064F	HOME	069F	HVSU	030B	INCCUR	072B
INCRST	0738	KBDATA	9FFE	KBPOLL	0080	KBREAD	0081
LBLKMV	05B9	LDCURS	9FF6	LF	0578	LMULT	05CE
LOOP	03DB	LSCR1	0596	LUPDAT	058A	MASKIN	0243
MNPRG	01B3	MODE	9FF9	MODEAD	00F8	NEWLIN	073A
NMI	0212	OBSERV	0759	ONEPG	9FE6	PGO	07BF
PREG	00F2	PRESET	00E0	PURPLE	00E6	RED	00E2
RESATT	0705	RESET	027D	RESSTA	07F9	RESTOR	026D
RPARM1	9FF1	RPARM2	9FF2	RPARM3	9FF3	RPARM4	9FF4
RPMADJ	0850	RVV	063F	SCRCOM	04E3	SCRL	04BA
SCRLD	051E	SCRLPT	9FE5	SCRRST	0502	SETOPG	02C5
SET3PG	02CE	SETATT	06E8	SETCF	051B	SETRPM	02D5
SETSCR	02A3	SETUP	0281	SKIP	0702	SPACE	0020
SPGPH	0836	SREG	00F3	STATUS	00F8	STOPAT	0657
STPDIS	0040	STRTDI	9FF5	STSPGP	0814	STSPRS	084A
TAB	0554	TABLOO	0560	TABOK	0571	TABSP	9FE0
TABSU	02DE	TURQSE	00E5	UCHGFR	05FA	UNDLIN	0647
UPLINE	05E1	URST	05F6	WRMST	0165	YELLOW	00E3

NO FATAL ERROR(S)