

Personal Computer
11Z-80B

Appendix

SHARP

NOTICE

The MZ-80 series of sophisticated personal computers is manufactured by the SHARP CORPORATION. Hardware and software specifications are subject to change without prior notice; therefore, you are requested to pay special attention to version numbers of the monitor and the system software (supplied in the form of cassette tape or mini-floppy disk files).

This manual is for reference only and the SHARP CORPORATION will not be responsible for difficulties arising out of inconsistencies caused by version changes, typographical errors or omissions in the descriptions.

This manual is based on the SB-1500 series monitor and the SB-7000 series Floppy DOS.

— CONTENTS —

LINKING ASSEMBLY PROGRAM WITH Floppy DOS	1
Sample Program (Command)	2
USER CODED I/O ROUTINES	4
User I/O Routine	4
Relocating User I/O Routines	6
Linking User I/O Routines with Floppy DOS	6
Sample Program (I/O Driver)	8
CONVERSION OF CASSETTE BASED SYSTEM PROGRAMS	11
MEMORY EXPANSION	12
I/O MAP	12
PAPER TAPE PUNCH AND READER INTERFACE	13
Single Name	13
I/O Ports	13
Timing Chart	14
Preparing a Paper Tape Punch/Reader I/O Card	15
COMMAND TABLES & ERROR MESSAGES	17
DOS Built-in Commands	17
DOS Transient Commands	20
System Error Messages	23
Editor Commands	25
Editor Error Messages	26
Assembler Messages	27
Symbolic Debugger Commands	28
Symbolic Debugger Error Messages	29
PROM Formatter Commands	30
File Mode	31
I/O Devices Handled by Floppy DOS	31
File Attributes	32
ASCII CODE TABLE	33
MZ-80B CIRCUIT DIAGRAMS	34
UNIVERSAL I/O CARD CIRCUIT DIAGRAM	45

1	LINKING ASSEMBLY PROGRAM WITH Floppy DOS
2	Sample Program (Command)
4	USER CODED I/O ROUTINES
4	User I/O Routine
6	Relocating User I/O Routines
6	Linking User I/O Routines with Floppy DOS
8	Sample Program (I/O Driver)
11	CONVERSION OF CASSETTE BASED SYSTEM PROGRAMS
12	MEMORY EXPANSION
12	I/O MAP
13	PAPER TAPE PUNCH AND READER INTERFACE
13	Single Name
13	I/O Ports
14	Timing Chart
15	Preparing a Paper Tape Punch/Reader I/O Card
17	COMMAND TABLES & ERROR MESSAGES
17	DOS Built-in Commands
20	DOS Transient Commands
23	System Error Messages
25	Editor Commands
26	Editor Error Messages
27	Assembler Messages
28	Symbolic Debugger Commands
29	Symbolic Debugger Error Messages
30	FROM Formatter Commands
31	File Mode
31	I/O Devices Handled by Floppy DOS
32	File Attributes
33	ASCII CODE TABLE
34	MZ-808 CIRCUIT DIAGRAMS
42	UNIVERSAL I/O CARD CIRCUIT DIAGRAM

LINKING ASSEMBLY PROGRAM WITH Floppy DOS

An object program generated with the Floppy DOS editor, assembler and linker can be executed with the RUN command.

Example 1: 1>RUN GALAXY CR

This command loads GALAXY.OBJ into memory from the floppy disk and executes it. Execution of a RET statement in the object program returns control to Floppy DOS. The contents of the stack pointer must be restored to the value contained when the object program was called before the RET statement is executed. CF must be reset before control is returned because an error message will be output on the assumption that the ACC contains a system error code if CF is set.

Global switches and/or arguments can be assigned after the file name in the RUN command as shown below.

Example 2: 2>RUN ASMZ8/P CONTROL-ACR

↑ ↑
Global switch Argument

In this case Floppy DOS converts the entire command line into intermediate codes (refer to the LIBRARY/PACKAGE Manual), and loads ASMZ8.OBJ into memory from the floppy disk, then executes it. At this time, the HL register points to the intermediate code corresponding to /P (88H). The RJOB area in Floppy DOS has the same value as the HL register.

Switches and arguments following the file name (ASMZ8) must be decoded by user object program. They can be decoded using Floppy DOS subroutines. When the last character (":" or 0DH) is decoded, the HL register contents must be stored in RJOB. To return control to Floppy DOS, execute a RET statement in the object program.

The sample program listed on the following pages illustrates command line decoding. It outputs an ASCII file to the CRT display or printer. This program operates in a manner similar to the DOS TYPE command. The file name of this program is TYPE'. Thus, executing

Example 3: 2>RUN TYPE' ABC CR

outputs file ABC.ASC to the CRT display and executing

Example 4: 2>RUN TYPE'/P ABC CR

outputs ABC.ASC to the printer.

All external labels (indicated by the E message) in this program list are defined in DOSEQU.LIB. See page SYS- 51 is System Command for the RUN command.

—Sample Program (Command)—

** Z80 ASSEMBLER SB-7201 <TYPE> PAGE 01

??/??/??

```

01 0000 ; TYPE COMMAND
02 0000 ;
03 0000 .TYPE: ENT
04 0000 116200 LD DE,SWTBL ; DE:=SWITCH TABLE
05 0003 CD0000 E CALL ?GSW ; CHECK GLOBAL SWITCH
06 0006 D8 RET C
07 0007 CD0000 E CALL C&L1 ; SELECT CRT OR LPT
08 000A CD0000 E CALL ?SEP ; CHECK SEPARATER
09 000D D8 RET C
10 000E FE2C CP 2CH ; SEPARATER="," ?
11 0010 3E03 LD A,3 ; 3 IS ERR CODE
12 0012 37 SCF
13 0013 C0 RET NZ ; NO, ERR RETURN
14 0014 CD0000 E TYPE0: CALL ?LSW ; CHECK LOCAL SWITCH
15 0017 D8 RET C
16 0018 3E08 LD A,8 ; 8 IS ERR CODE
17 001A 37 SCF
18 001B C0 RET NZ ; ERROR, LSW EXIST
19 001C 0E80 LD C,128 ; LU#:=128
20 001E D9 EXX
21 001F 0604 LD B,4 ; DEFAULT MODE=ASC
22 0021 D9 EXX
23 0022 CD0000 E CALL ROPEN ; READ-OPEN
24 0025 D8 RET C
25 0026 CD0000 E CALL &NL
26 0029 382D JR C,TYPEER
27 002B CD0000 E CALL TESW ; TEST GLOBAL SWITCH
28 002E 88 DEFB 88H ; /P
29 002F 3F CCF
30 0030 DC0000 E CALL C,PPAGE ; LPT PAGING
31 0033 3823 JR C,TYPEER
32 0035 CD0000 E CALL MODECK ; FILEMODE CHECK
33 0038 04 DEFB 4 ; .ASC ?
34 0039 116400 LD DE,BUFFER
35 003C D40000 E TYPE10: CALL NC,GET1L ; GET 1 LINE
36 003F D40000 E CALL NC,&1L ; DISP OR PRINT 1 LINE
37 0042 30F8 JR NC,TYPE10 ; NO ERROR
38 0044 A7 AND A
39 0045 C25800 JP NZ,TYPEER ; ERROR
40 0048 ; ; END-OF-FILE
41 0048 CD0000 E TYPE20: CALL CLOSE ; CLOSE FILE
42 004B CD0000 E CALL ?SEP ; CHECK SEPARATER
43 004E D8 RET C
44 004F FE2C CP 2CH ; SEPARATER="," ?
45 0051 28C1 JR Z,TYPE0 ; YES, TYPE NEXT FILE
46 0053 220000 E LD (RJOB),HL ; SAVE CLI POINTER
47 0056 AF XOR A
48 0057 C9 RET
49 0058 CD0000 E TYPEER: CALL ERR ; ERROR OCCUR
50 005B CD0000 E CALL KILL ; KILL FILE (C=128)
51 005E 37 SCF ; SET ERROR FLAG
52 005F 3EFF LD A,FFH ; ALREADY DISP ERR MSG
53 0061 C9 RET
54 0062 88 SWTBL: DEFB 88H ; /P
55 0063 FF DEFB FFH ; END OF SWTBL
56 0064 BUFFER: DEFS 128 ; 128 BYTE BUFFER
57 00E4 END

```

```
.TYPE 0000  BUFFER 0064  SWTBL  0062  TYPE0  0014  TYPE10 003C
TYPE20 0048  TYPEER 0058
```

USER CODED I/O ROUTINES

Floppy DOS supports control programs not only for the floppy disk drive but for the printer (\$LPT) and the paper tape reader (\$PTR), etc. Other I/O devices can be operated under the control of Floppy DOS by means of user coded control programs.

—User I/O Routine—

A user I/O routine consists of the following sections.

- A. Device table (57 bytes)
- B. ROPEN or WOPEN procedure
- C. Data transfer program
- D. CLOSE and KILL procedure

These sections are explained below using the Floppy DOS paper tape reader control program (\$PTR) as an example.

A. Device Table (line 7 through 20, bytes 0 through 56)

- * FDOS uses bytes 0 through 2 (FLAG 0 ~ FLAG 2).

This area must be written exactly as it is shown.

- * Byte 3 (FLAG 3) represents the attribute of the I/O device.

Bit 7 : 0

Bit 6 : 1 indicates that tabulation is possible. (This bit is set to 1 for the printer. See Note 1 on page 7.)

Bit 5 : 1 indicates that parity specification (\$PTR/PE, etc.) can be made.

Bit 4 : 1 indicates that only .ASC mode files can be transferred.

Bit 3 : 0

Bit 2 : 0

Bit 1 : 1 indicates that WOPEN is possible. (See note 2 on page 7.)

Bit 0 : 1 indicates that ROPEN is possible. (See note 2 on page 7.)

- * Byte 4 indicates the data transfer format. (described later)
- * Bytes 5 and 6 are the starting address of the subroutine to be called during ROPEN execution.
- * Bytes 7 and 8 are the starting address of the subroutine to be called during WOPEN execution. (WOPEN is not executed for \$PTR so DEFW 0 is specified in this program.)
- * Bytes 9 and 10 are loaded with data by the Floppy DOS STATUS command. (Not used for \$PTR.)
- * Bytes 11 through 14 are the starting addresses of the subroutines for CLOSE and KILL processing.
- * Bytes 15 through 22 (Procedures 1 through 4) are loaded with data transfer routine addresses. The data transfer procedure differs according to the transfer format.

ROPEN

Transfer format	1	4
Procedure 1	Input 1 character (ACC ← data)	Input 1 line (From the address indicated by DE to a CR code.)
Procedure 2 ~ 4	Unused	Unused

WOPEN

Transfer format	1	5
Procedure 1	Unused	Carriage return
Procedure 2	Output 1 character (ACC : data) [†]	Output 1 character (ACC : data) [†]
Procedure 3	Unused	Output 1 line (Corresponds to monitor subroutine MSG)
Procedure 4	Unused	Output line (Corresponds to monitor subroutine MSGX)

[†] On .ASC mode, 0DH means carriage-return,
and 0CH means form-feed.

- * Bytes 23 and 24 are used by Floppy DOS.
- * Byte 25 is used only when bit 6 of FLAG 3 is 1, in which case it must be loaded with the number of characters of the line which have been output by I/O routine.
- * Byte 26 is loaded with the file mode by Floppy DOS.
- * Bytes 27 through 56 are the device name (up to 16 characters); the rest area must be reserved with DEFS.
- * When the transfer format is 4, a buffer area for 1 line is reserved after the byte 56 with DEFS.

B. ROPEN or WOPEN procedure (lines 39 through 50)

Only ROPEN is needed for the paper tape reader (\$PTR). The tape-feeder is skipped by this procedure. WOPEN is also used to start a new page during output of an assembly listing.

C. Data transfer program (lines 51 through 83)

Program which performs actual transfer of data.

D. CLOSE and KILL procedures (lines 49 through 50)

No function with \$PTR.

To return control to Floppy DOS from the ROPEN, WOPEN, Procedure 1 ~ 4, CLOSE or KILL routines, set registers as follows before executing the RET statement.

Normal : CR ← 0

Error : CF ← 1, ACC ← error code (refer to the System Error Messages in the System Command Manual.)

File end : CF ← 1, ACC ← 0

The contents of the IY, BC, DE and HL registers must be saved in any case.

—Relocating User I/O Routines—

First, assemble the program coded (the program name DVM is used below).

Example 1: 2> ASM DVM, \$LPT/L CR

Next, relocate the file to generate the object program. A higher loading address must be specified at this time because of factors related to the LIMIT command described later. Take care to ensure that addresses do not overlap when two or more user I/O programs are used. If necessary, link MONEQU.LIB or DOSEQU.LIB with the user I/O programs.

Example 2: 2> LINK \$C000, DVM CR

Example 2': 2> LINK \$C400, CDISP, \$FD1 ; DOSEQU.LIB CR

—Linking User I/O Routines with Floppy DOS—

User I/O routines must be linked with Floppy DOS I/O controller every time Floppy DOS is activated.

First, use the LIMIT command to reserve an area in memory for loading the object program (DVM.OBJ).

Example 3: 2> LIMIT \$F000 CR

Next, load the object program.

Example 4: 2> LOAD DVM CR

Finally, link the routine to the Floppy DOS I/O controller. \$USR1 through 4 are provided in Floppy DOS as device names for user I/O routines; assign the user I/O control routine to one of these device names.

Example 5: 2> ASSIGN \$USR1, \$F000 CR

Now the user program is linked with Floppy DOS and can be called by specifying \$USR1 (~4). It is convenient to prepare EXEC files which include LIMIT, LOAD and ASSIGN commands such as those shown above. (Refer to the System Command Manual).

User I/O programs are called as shown below.

Example of use by DOS commands

2> TYPE \$USR1 CR

2> XFER DATA4, \$USR2 CR

Example of use by BASIC compiler

```

10 ROPEN #2, "$USR1"
20 INPUT #2, A$
30 IF EOF (#2) THEN 100
40 PRINT A$
.....
999 CLOSE #2

```

Note 1: Bit 6 determines the functions of BASIC statements PRINT # and INPUT #.

When bit 6 = 1, data is treated in the same manner as with the PRINT and INPUT statements. When bit 6 = 0, separators (" , " and " ; ") in the PRINT # statement are replaced with CR and commas included in the input character string for the INPUT # statement are treated as data; only CR is regarded as a data separator. (This is the same as with the PRINT # statement supported by SB-6510 and the PRINT/T statement supported by SB-5510.)

Note 2: Both ROPEN and WOPEN are possible, when both bit 1 and bit 0 are set, but they cannot be executed simultaneously.

—Sample Program (I/O Driver)—

** Z80 ASSEMBLER SB-7201 <PTRP> PAGE 01

07/06/81

```

01 0000      ;
02 0000      ; PTR/PTP DRIVER for MZ-80B
03 0000      ; Copyright 1981 by SHARP Corp.
04 0000      ;
05 0000      ;
06 0000      $PTR: ENT
07 0000 0000      DEFW 0
08 0002 00      DEFB 0
09 0003 21      DEFB 21H      ;PAR,READ-ENABLE
10 0004 01      DEFB 1
11 0005 7900      DEFW $PTR0      ;ROPEN
12 0007 0000      DEFW 0
13 0009 0000      DEFW 0      ;STATUS
14 000B 8600      DEFW CLC      ;CLOSE
15 000D 8600      DEFW CLC      ;KILL
16 000F 8800      DEFW $PTR1
17 0011      DEFS 10
18 001B 24505452  $PTRNM: DEFM '$PTR'
19 001F 0D      DEFB 0DH
20 0020      DEFS 25
21 0039      ;
22 0039      $PTP: ENT
23 0039 0000      DEFW 0H
24 003B 00      DEFB 0H
25 003C 22      DEFB 22H      ;PAR,WOPEN-ENABLE
26 003D 01      DEFB 1
27 003E      DEFS 2
28 0040 2901      DEFW $PTPFD      ;WOPEN
29 0042 0000      DEFW 0H      ;STATUS
30 0044 2901      DEFW $PTPFD      ;CLOSE
31 0046 8600      DEFW CLC
32 0048      DEFS 2
33 004A BE00      DEFW $PTP1C
34 004C      DEFS 8
35 0054 24505450  $PTPNM: DEFM '$PTP'
36 0058 0D      DEFB 0DH
37 0059      DEFS 25
38 0072      ;
39 0072 111B00      $PTRNR: LD DE,$PTRNM
40 0075 CD0000      CALL IOWAIT
41 0078 D8      RET C
42 0079 CD9500      $PTR0: CALL $PTRIN      ;ROPEN
43 007C 38F4      JR C,$PTRNR
44 007E 78      LD A,B
45 007F A7      AND A
46 0080 28F7      JR Z,$PTR0
47 0082 78      LD A,B
48 0083 32BD00      LD ($PTRD),A
49 0086 AF      CLC: XOR A
50 0087 C9      RET
51 0088 CD9500      $PTR1: CALL $PTRIN      ;GET1C
52 008B D8      RET C
53 008C 21BD00      LD HL,$PTRD
54 008F 7E      LD A,M
55 0090 70      LD M,B
56 0091 A7      AND A
57 0092 C0      RET NZ
58 0093 37      SCF      ;EOF
59 0094 C9      RET
60 0095      ;

```


01 0095 3EEF	\$PTRIN: LD	A, EFH	
02 0097 D3FD	OUT	(FDH), A	
03 0099 CDB000	\$PTR2: CALL	\$PTRCK	
04 009C CB67	BIT	4, A	
05 009E 28F9	JR	Z, \$PTR2	
06 00A0 CDB000	\$PTR3: CALL	\$PTRCK	
07 00A3 CB67	BIT	4, A	
08 00A5 20F9	JR	NZ, \$PTR3	
09 00A7 DBFC	IN	A, (FCH)	
10 00A9 2F	CPL		
11 00AA 47	LD	B, A	
12 00AB 3EFF	\$PTR5: LD	A, FFH	
13 00AD D3FD	OUT	(FDH), A	
14 00AF C9	RET		
15 00B0 DBFD	\$PTRCK: IN	A, (FDH)	
16 00B2 CB6F	BIT	5, A	
17 00B4 C8	RET	Z	
18 00B5 F1	POP	AF	
19 00B6 CDAB00	CALL	\$PTR5	
20 00B9 37	SCF		
21 00BA 3E3C	LD	A, 60	: NOT READY
22 00BC C9	RET		
23 00BD	\$PTRD: DEFS	1	
24 00BE	;		
25 00BE	;		
26 00BE F5	\$PTP1C: PUSH	AF	
27 00BF DBFD	IN	A, (FDH)	
28 00C1 E601	AND	1	
29 00C3 2818	JR	Z, \$PTP20	
30 00C5 3EFE	LD	A, FEH	
31 00C7 D3FD	OUT	(FDH), A	
32 00C9 210000	LD	HL, 0H	
33 00CC 2B	\$PTP10: DEC	HL	
34 00CD DBFD	IN	A, (FDH)	
35 00CF E601	AND	1	
36 00D1 280A	JR	Z, \$PTP20	
37 00D3 7C	LD	A, H	
38 00D4 B5	OR	L	
39 00D5 00	NOP		
40 00D6 00	NOP		
41 00D7 3E3C	LD	A, 60	: NOT READY
42 00D9 2837	JR	Z, \$PTP60	
43 00DB 18EF	JR	\$PTP10	
44 00DD	;		
45 00DD 3EFE	\$PTP20: LD	A, FEH	
46 00DF D3FD	OUT	(FDH), A	
47 00E1 DBFD	\$PTP30: IN	A, (FDH)	
48 00E3 CB47	BIT	0, A	
49 00E5 20FA	JR	NZ, \$PTP30	
50 00E7 F1	POP	AF	
51 00E8 F5	PUSH	AF	
52 00E9 2F	CPL		
53 00EA D3FC	OUT	(FCH), A	
54 00EC 3EFC	LD	A, FCH	
55 00EE D3FD	OUT	(FDH), A	
56 00F0 210000	LD	HL, 0H	
57 00F3 2B	\$PTP40: DEC	HL	
58 00F4 7C	LD	A, H	
59 00F5 B5	OR	L	
60 00F6 3E4E	LD	A, 78	: TIME OUT

```

01 00F8 2818      JR    Z,$PTP60
02 00FA DBFD      IN     A,(FDH)
03 00FC CB4F      BIT    1,A
04 00FE 20F3      JR    NZ,$PTP40
05 0100 DBFD      IN     A,(FDH)
06 0102 CB4F      BIT    1,A
07 0104 20ED      JR    NZ,$PTP40
08 0106 DBFD      $PTP50: IN    A,(FDH)
09 0108 CB4F      BIT    1,A
10 010A 28FA      JR    Z,$PTP50
11 010C DBFD      IN     A,(FDH)
12 010E CB4F      BIT    1,A
13 0110 28F4      JR    Z,$PTP50
14 0112 E1        $PTP60: POP    HL
15 0113 F5        PUSH   AF
16 0114 3EFE      LD     A,FEH
17 0116 D3FD      OUT    (FDH),A
18 0118 CD3601     CALL   DLY80U
19 011B 3D        DEC     A
20 011C D3FC      OUT    (FCH),A
21 011E F1 TON:   POP    AF
22 011F C0        RET     NZ
23 0120 37        SCF
24 0121 C9        RET
25 0122          ;
26 0122 115400     $PTPNR: LD    DE,$PTPNM
27 0125 CD0000     E      CALL   IOWAIT
28 0128 D8        RET     C
29 0129 0696      $PTPFD: LD    B,150
30 012B C5        PUSH   BC
31 012C AF        XOR    A
32 012D CDBE00     CALL   $PTP1C
33 0130 C1        POP    BC
34 0131 38EF      JR    C,$PTPNR
35 0133 10F6      DJNZ   $PTPFD+2
36 0135 C9        RET
37 0136          ;
38 0136 111000     DLY80U: LD    DE,16
39 0139 1B        DEC    DE
40 013A 7A        LD     A,D
41 013B B3 TON:   OR     E
42 013C 20FB      JR    NZ,-3
43 013E C9        RET
44 013F          END

```

```

$PTP  0039  $PTP10 00CC  $PTP1C 00BE  $PTP20 00DD  $PTP30 00E1
$PTP40 00F3  $PTP50 0106  $PTP60 0112  $PTPFD 0129  $PTPNM 0054
$PTPNR 0122  $PTR   0000  $PTR1   0088  $PTR2   0099  $PTR3   00A0
$PTR5   00AB  $PTRCK 00B0  $PTRD   00BD  $PTRIN 0095  $PTRNM 001B
$PTRNR 0072  $PTR0   0079  CLC      0086  DLY80U 0136

```

CONVERSION OF CASSETTE BASED SYSTEM PROGRAMS

The following cassette based system programs (for MZ-80K) have thus far been released.

- MACHINE LANGUAGE SP-2001
- EDITOR-ASSEMBLER SP-2201, SP-2102
- RELOCATABLE LOADER SP-2301
- SYMBOLIC DEBUGGER SP-2401

These system programs (and the MZ-80K Floppy DOS) generate source files (with file mode .ASC), relocatable files (with file mode .RB), object files (with file mode .OBJ) and debug mode save files (i.e., object files with symbol tables). Of these, source files and object files can be transferred to Floppy DOS disks.

The procedure for transferring a cassette file to an Floppy DOS file is as follows.

When the file name consists of characters which are usable with Floppy DOS:

XFER \$CMT1, \$FDn (n = 1 - 4)

When the file name includes characters which are not allowed by Floppy DOS, a new file name must be assigned as follows:

XFER \$CMT1, \$FDn ; filename (n = 1 - 4)

When an assembly source file is to be transferred, use the following procedures to determine whether or not pseudo instruction REL is used: load the file with the Floppy DOS text editor and search for REL with the S command. Delete all REL instructions; this is because Floppy DOS system programs do not require REL.

Next, assemble the file from which REL instructions have been deleted to generate a relocatable file with the Floppy DOS assembler. The object file is obtained by relocating it.

When object files generated by cassette based system programs are transferred to an Floppy DOS file, they can be executed with the following command.

RUN \$FDn ; filename

The following message is displayed on the CRT screen when the specified object file has a loading address which results in destruction of the Floppy DOS area.

destroy Floppy DOS?

Pressing the **[Y]** key at this time performs the transfer operation, destroying the Floppy DOS area; pressing the **[N]** key stops the operation and returns the system to the DOS command wait state.

MEMORY EXPANSION

Floppy DOS requires 64K bytes of RAM

The standard memory size of the MZ-80B is 32K bytes; this is expandable to 64K bytes.

The optional 32K byte expansion RAM card, MZ-80RM, is inserted into the 20-pin connector which is located on the right rear side of the CPU board (as viewed from the rear). The standard 32K byte RAM card is located next to the expansion RAM connector. The connector pins on the bottom of the expansion RAM card are inserted into the 20 pin connector on the CPU board.

Visually check orientation of the expansion RAM card before inserting it.

I/O MAP

I/O ports with addresses equal to or higher than B0 are reserved by the manufacturer for control of external devices; those used by Floppy DOS are assigned device names such as \$LPT.

00	User ports
B0	(RS-232C)
C0	
D0	
D8	Floppy disk (\$FD1 ~ \$FD4)
E0	8255, 8253, PIO
EC	
EE	
F0	Graphic display
F8	EX-ROM
FA	
FC	Paper tape punch and reader (\$PTP, \$PTR)
FE	Printer (\$LPT)

PAPER TAPE PUNCH AND READER INTERFACE

Floppy DOS has built-in paper tape punch and reader control programs. These are assigned the device names \$PTP and \$PTR, respectively. In actuality, however an interface circuit must be established with a universal interface I/O card to connect the paper tape punch and reader with the MZ-80 series micro-computer. The circuit diagram is shown on page 45.

The method for controlling the paper tape punch and reader is not standardized. A paper tape punch and reader which can be controlled by Floppy DOS must have the following signal timing system. The signal names and timing charts shown below are based on the RP-600 paper tape punch and reader manufactured by Nada Electronics Laboratory. (For details, refer to the manual included with the paper tape punch and reader.)

—Signal Name—

< Puncher >

- $\overline{DT}_1 \sim \overline{DT}_8$: Data (PTP \leftarrow CPU)
- \overline{MI}^* : Motor ON/OFF control signal (PTP \leftarrow CPU)
- \overline{ST} : START/STOP control signal (PTP \leftarrow CPU)
- \overline{TO} : Timing signal (PTP \rightarrow CPU)
- $(\overline{RDY})^{**}$: Ready state signal (PTP \rightarrow CPU)

(This signal is not output from the RP-600 since it can be used in remote operation. Ground it when the RP-600 is used.)

< Reader >

- $\overline{RD}_1 \sim \overline{RD}_8$: Data (PTR \rightarrow CPU)
- \overline{STA} : START/STOP control signal (PTR \leftarrow CPU)
- \overline{SPR} : Sprocket signal (PTR \rightarrow CPU)
- \overline{RB} : Tape end signal (abnormal stop signal) (PTR \rightarrow CPU)

* Do not connect when the motor is not remotely controlled.

** The DPT26A manufactured by the Anritsu Electric Co. outputs this signal, but the RP-600 does not.

—I/O Ports—

Port FC_H is used for data by both the punch and the reader. Port FD_H is used for control signals. See Table 1.

< Punch >				< Reader >			
O ₁₀							I ₁₇
[Data]							
O ₂₀							O ₂₇
[Control signals]							
I ₂₀							I ₂₇

Table 1. Port allocation

—Timing Chart—

Punch

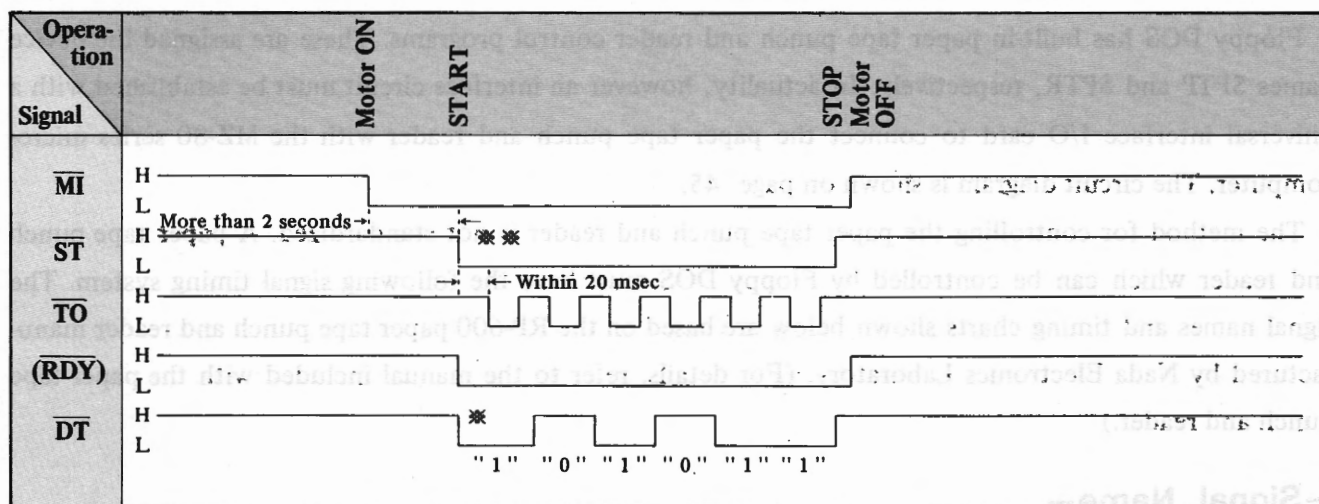


Figure 1. Punch timing chart

- * The next data to be punched is readied while \overline{TO} is H and maintained while \overline{TO} is L.
- ** \overline{ST} is set to L 2 or more seconds after the motor has been started, and is set to H after \overline{TO} has risen from L to H for the last data.

Reader

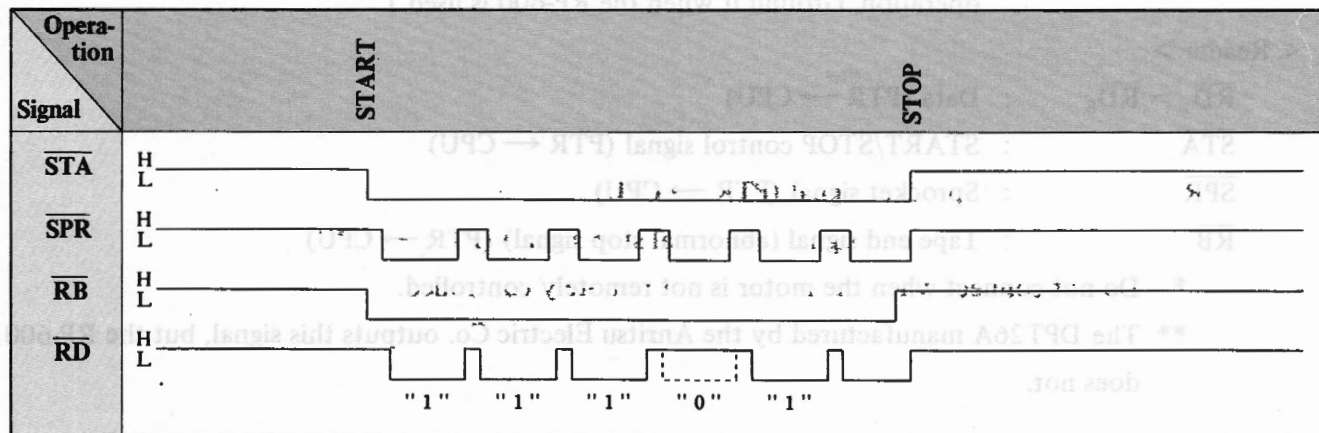


Figure 2. Reader timing chart

—Preparing a Paper Tape Punch/Reader I/O Card—

It is convenient to use a universal I/O card (MZ-80IO2) for preparing a paper tape punch and reader I/O interface circuit. Markings such as O₁₀ or O₁₇ in the port allocation table on page 13 match those on the universal I/O card. See page 16 for setting the universal I/O card switches to select port addresses FC and FD.

The RP-600 internal interface circuit and input and output pin connections are shown below for reference. (For details, refer to the manual included with the RP-600).

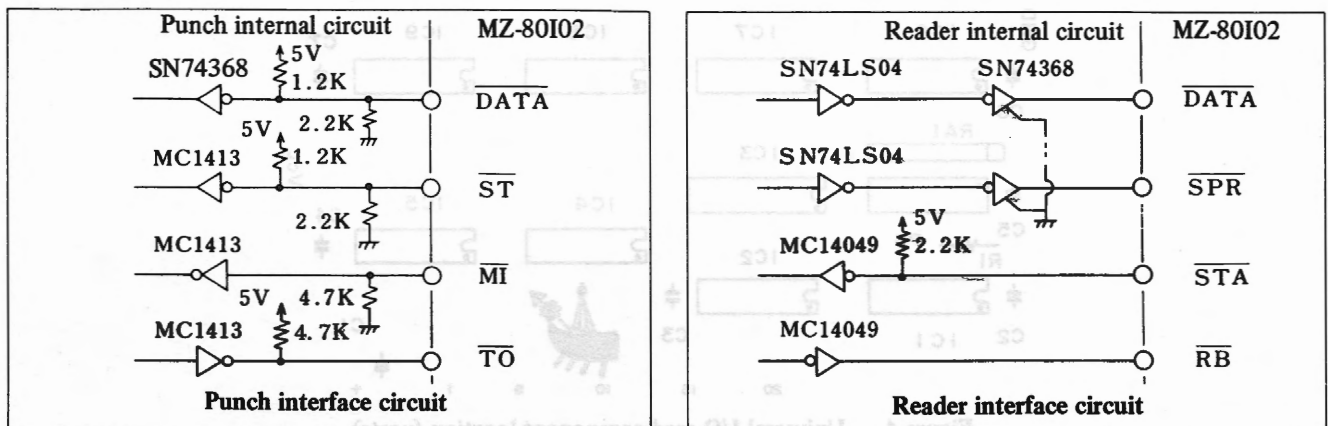


Figure 3. Interface circuit (RP-600)

Punch I/O connector

Pin	Signal	Pin	Signal
1	DT ₁	14	
2	DT ₂	15	
3	DT ₃	16	
4	DT ₄	17	
5	DT ₅	18	
6	DT ₆	19	
7	DT ₇	20	
8		20	
9	DT ₈	21	
10		22	MI Motor ON/OFF signal
11	TO Timing signal	23	ST START/STOP signal
12	GND	24	FG Frame ground
13			

Reader I/O connector

Pin	Signal	Pin	Signal
1	RD ₁	20	
2	RD ₂	21	
3	RD ₃	22	
4	RD ₄	23	Data
5	RD ₅	24	
6	RD ₆	25	
7	RD ₇	26	
8	SPR Sprocket signal	27	
9	RD ₈ Data	28	STA START/STOP signal
10		29	RB Operating state
11		30	FG Frame ground
12	GND	31	
13		32	
14		33	
15		34	
16		34	
17		34	
18		35	
19		36	

Table 2. Connector pin connections

Switch No.	Address bit
A ₁	A ₁
A ₂	A ₂
A ₃	A ₃
A ₄	A ₄
A ₅	A ₅
A ₆	A ₆
A ₇	A ₇

COMMAND TABLES & ERROR MESSAGES

—DOS Built-in Commands—

BOOT	
Terminates the Floppy DOS and activates system IPL. Example: BOOT ↵	
CHATTR sign, filename1, attribute [, ...filenameN, attribute]	
Matches the password's sign and changes the file attribute(s) of the matching file(s) identified by filename to attribute(s). P: Permanent file R: Read inhibit O: No protection W: Write inhibit Examples: CHATTR KEY, ABC, O, XYZ, P ↵ : Deletes the file attribute of file ABC and changes the file attribute of file XYZ to PERMANENT if matches occur with the password KEY. CHATTR KEY, \$FD2 ; UVW, R ↵ : Changes the file attribute of file UVW in FD2 to READ INHIBIT if a match occurs with the password KEY. CHATTR ↵ : This allows the programmer to interactively specify the password, file name and attribute.	
CONSOLE Sscrolling-start-line, end-line [, Ccharacter-number, R, N]	
Sets the scrolling area on the CRT screen, sets the character display mode and/or reverses the picture on the screen. Examples: CONSOLE C80 ↵ : Sets the number of characters per line to 80. CONSOLE R ↵ : Reverses the picture on the screen.	
DATE [MM.DD.YY]	
Displays the current date or sets the specified date in month, date, year format. The set information is used as file information when new files are created. Global switch /P : Specifies that the date is to be printed on the LPT. Examples: DATE /P ↵ : Lists the current date on the LPT. DATE 12.25.80 ↵ : Sets the current date to December 25, 1980.	
DELETE filename1 [, ..., filenameN] (?, *)	
Deletes the file(s) specified by filename(s) Global switch /C : Specifies that each file name is to be displayed on the screen for verification. The programmer must enter Y to delete it or N to suppress deletion. Examples: DELETE ABC . * ↵ : Deletes all files identified by ABC . * . DELETE /C A * . * ↵ : Displays files identified by A * . * on the screen for verification before deletion. filename : ABC.ASC deleted ← Indicates that the file is deleted since "Y" is entered. filename : ABC.RB ← Indicates that the file is not deleted "N" is entered. filename : AX.Y.OBJ permanent ← Indicates that the file is not deleted because it is assigned the PERMANENT file attribute.	

DIR [\$FDn] or [filename] (? ,*)	
<p>Displays file information in the directory specified by \$FDn or of the file specified by filename on the screen.</p> <p>Global switch /P : Specifies that the file information is to be output to LPT. The file information is displayed on the screen when this switch is not specified.</p> <p>Examples: DIR ↵ : Displays all file information in the current directory on the screen.</p> <p>DIR/P \$FD2 ↵ : Outputs all FD2 file names to LPT and switches the currently logged disk to FD2.</p> <p>DIR \$FD2 ; ABC. * ↵ : Displays the file information of files in FD2 identified by ABC. * .</p>	
EXEC filename	
<p>Executes the contents of the file identified by filename as DOS commands.</p> <p>Example: EXEC ABC . ASC ↵ : Sequentially executes the DOS commands in file ABC.</p>	
FAST	
<p>Fast forwards the cassette tape.</p> <p>Example: FAST ↵</p>	
FREE [\$FDn]	
<p>Lists statistical information about the disk identified by \$FDn on the screen or on the LPT.</p> <p>Example: FREE \$FD2 ↵</p> <p>\$FD2 master left : XXXX used : YYYY</p> <p>Indicates that the disk on FD2 is a master disk, that the number of unused sectors is XXXX and that the number of used sectors is YYYY.</p>	
KEY keynumber = "S"	
<p>Assigns a function to the definable function key indicated by a keynumber from 1 through 20. The function is specified by writing a string or command name enclosed in double quotation marks.</p> <p>Example: KEY 1 = "RUN" ↵ : Assigns the function of the RUN command to key 1.</p>	
KLIST	
<p>Lists the definition status of the definable function keys on the screen.</p> <p>Example: KLIST ↵</p>	
MON	
<p>Terminates FDOS processing and returns control to the monitor.</p> <p>Example: MON ↵</p>	
POKE \$nnnn, data [, ..., \$uuuu, dataN]	
<p>Stores data in the specified addresses in memory.</p> <p>Example: POKE \$000D, 2010, \$000F, 40 ↵</p>	
RENAME oldname1, newname1 [, ..., oldnameN, newnameN]	
<p>Renames the file specified by oldname to newname.</p> <p>Examples: RENAME ABC, XYZ ↵ : Renames file ABC to XYZ.</p> <p>RENAME ABC, DEF, UVW, XYZ ↵ : Renames file ABC to DEF and UVW to XYZ.</p>	

REW	
Rewinds the cassette tape. Example: REW ↵	
RUN filename	
Executes the program in the object file identified by filename. Example: RUN ABC ↵ : Executes the program in file ABC, assuming it to be ABC.OBJ.	
TIME [HH : MM : SS]	
Displays the current time or sets specified time in hour, minute, second format. The current time is set to 00 : 00 : 00 upon system start. Global switch /P : Specifies that the current time is to be listed on the LPT. Examples: TIME/P ↵ : Lists the current time on the LPT. TIME 16 : 30 : 30 ↵ : Sets the current time to 16 : 30 : 30	
TYPE filename1 [, ..., filenameN]	(?, *)
Lists the contents of the file(s) identified by filename(s) on the screen or on LPT. Global switch /P : Lists the file contents on LPT. Examples: TYPE ABC, DEF ↵ : Displays the contents of files ABC and DEF on the screen. TYPE/P \$FD3 ; XYZ ↵ : Lists the contents of file XYZ in FD3 on LPT. TYPE \$PTR ↵ : Reads paper tape data from PTR and displays it on the screen.	
XFER sourcefile1, destinationfile2 [, ..., sourcefileN, destinationfileN]	(sourcefile only ?, *)
Transfers the source file(s) to the destination file(s). Examples: XFER ABC, XYZ ↵ : Copies file ABC to XYZ. XFER \$PTR, DEF ↵ : Transfers the file at the PTR to file DEF. XFER XYZ, \$PTP/PE ↵ : Transfers file XYZ to the PTP with even parity in ASCII code.	
COPY	
Copies the files on the disk in drive 1 to the disk in drive 2. The system matches the passwords in these disks before carrying out a copy operation. Example: COPY ↵	
DEBUG filename [, ... filenameN]	
Invokes the symbolic debugger and links and loads relocatable file(s). Global switch \T : Specifies that the symbol table information is to be output. Global switch \P : Specifies that the listing is to be directed to the LPT (the listing is displayed on the screen if omitted). Local switch \O : Specifies that the object file is to be generated with the specified file name. Example: DEBUG ABC, DEF ↵ : Invokes the symbolic debugger, links and loads relocatable files ABC and DEF and waits for a symbolic debugger command.	
EDIT filename	
Loads the text editor and reads in the file (if specified). The file must be an ASCII mode file. Examples: EDIT ↵ : Loads the text editor and waits for an editor command. EDIT \$PD3 ; ABC ↵ : Loads the text editor and reads in the ABC from FD3.	

—DOS Transient Commands—

ASM filename	
Assembles the source file identified by filename and produces a relocatable file and an assembly listing.	
Global switch (none)	: Specifies that the relocatable file is to be output.
Global switch/N	: Suppresses generation of the relocatable file.
Local switch/O	: Specifies that the relocatable file is to be output with the specified file name.
Local switch/E	: Specifies that error statements are to be output to the specified file.
Local switch/L	: Specifies that the listing is to be directed to the specified file.
Examples: ASM ABC ↵	: Assembles source file ABC and generates relocatable file ABC.RB.
ASM/N ABC, \$CRT/E ↵	: Assembles source file ABC and displays error statements on the screen (no relocatable file is created).
ASM ABC, XYZ/O, \$LPT/L ↵	: Assembles source file ABC and generates relocatable file XYZ.RB and an assembly listing on the LPT.
ASM ABC, \$FD2 ; XYZ/L, \$LPT/E ↵	: Assembles source file ABC outputs the assembly listing to file XYZ.ASC in FD2 and outputs error statements on the LPT.
ASSIGN devicename, address	
Sets the address of a user device drive routine.	
Examples: ASSIGN \$USR1, \$B000 ↵	: Sets the drive routine address of user device \$USR1 to B000 (hexadecimal).
BASIC filename	
Invokes the BASIC compiler to compile the source program identified by filename.	
Example: BASIC XYZ ↵	: Invokes the BASIC compiler, compiles source file XYZ.ASC and generates relocatable file XYZ.RB.
CONVERT	
Converts a file generated with the SB-5000 series BASIC interpreter or the D-BASIC SB-6000 series into a file which can be used under Floppy DOS, or converts a file generated with Floppy DOS into a file which can be used under the SB-5000 series BASIC interpreter or the D-BASIC SB-6000 series.	
Example: CONVERT ↵	
COPY	
Copies the files on the disk in drive 1 to the disk in drive 2. The system matches the passwords in these disks before carrying out a copy operation.	
Example: COPY ↵	
DEBUG filename [, ..., filenameN]	
Invokes the symbolic debugger and links and loads relocatable file(s).	
Global switch /T	: Specifies that the symbol table information is to be output.
Global switch /P	: Specifies that the listing is to be directed to the LPT (the listing is displayed on the screen if omitted).
Local switch /O	: Specifies that the object file is to be generated with the specified file name.
Example: DEBUG ABC, DEF ↵	: Invokes the symbolic debugger, links and loads relocatable files ABC and DEF and waits for a symbolic debugger command.
EDIT [filename]	
Loads the text editor and reads in the file (if specified). The file must be an ASC mode file.	
Examples: EDIT ↵	: Loads the text editor and waits for an editor command.
EDIT \$FD2 ; ABC ↵	: Loads the text editor and reads in file ABC from FD2.

FORMAT [\$FDn]	
Initializes the disk in \$FDn in the system format. The password set by the SIGN command is checked before execution.	
Examples: FORMAT ↵	: Initializes the currently logged-on disk.
FORMAT \$FD2 ↵	: Initializes the disk in FD2.
HCOPY n	
Copies a data frame from the CRT screen to the LPT.	
Examples: HCOPY 4 ↵	: Copies a data frame from the CRT where the contents of graphic areas 1 and 2 are displayed simultaneously.
LIBRARY filename1 [, ..., filenameN]	
Links specified file(s) into a library file.	
Global switch (none)	: Specifies that the link information is to be displayed on the screen.
Global switch /P	: Specifies that the link information is to be printed on the LPT.
Examples: LIBRARY ABC, DEF, ↵	: Links relocatable files ABC and DEF and stores their contents into library file ABC.LIB
LIBRARY ABC, DEF, XYZ/O ↵	: Links relocatable files ABC and DEF and stores their contents into library file XYZ.LIB.
LIMIT address	
Sets or changes the end address of the memory area managed by Floppy DOS.	
Examples: LIMIT \$B000 ↵	: Sets the Floppy DOS area to B000 (hexadecimal).
LIMIT MAX ↵	: Sets the Floppy DOS area to the maximum available address.
LINK filename1 [, ..., filenameN]	
Links relocatable files identified by filename1 through filenameN and outputs an object file with a link table listing.	
Global switch /T	: Specifies that the symbol table information is to be listed.
Global switch /P	: Specifies that the listing is to be directed to the LPT (the listing is displayed on the screen if the switch is omitted).
Global switch /S	: Specifies that a system file is to be generated.
Examples: LINK ABC, DEF ↵	: Links relocatable files ABC and DEF and outputs object file ABC.OBJ
LINK/T/P ABC, DEF, XYZ/O ↵	: Links relocatable files ABC and DEF and outputs object file XYZ.OBJ with the link table information on the LPT.
LOAD filename	
Loads the object file identified by filename into the area immediately following the area established by the LIMIT command.	
Example: LOAD ABC.OBJ ↵	: Loads object file ABC.OBJ into memory.
MLINK filename1 [, ..., filenameN]	
Links relocatable files identified by filename1 through filenameN and outputs an object file with a link table listing. This command can link files to generate an object file of up to 46K bytes, although the LINK command can only deal with up to 36K bytes.	
Global switch /T	: Specifies that the symbol table information is to be listed.
Global switch /P	: Specifies that the listing is to be output on the LPT (the listing is displayed on the screen if this switch is omitted).
Example: MLINK ABC, DEF ↵	: Links relocatable files ABC and DEF and outputs object file ABC.OBJ.

PAGE [output-device] or nn	
Performs a form feed operation on the output device identified by output-device, or sets the number of lines per page on the LPT.	
Examples:	PAGE ↵ : Moves the print position to the home position of the printer form. PAGE 22 ↵ : Sets the number of lines per page on the LPT to 22. The print form is fed to the top of the next page when a page feed code is issued or the TOP OF FORM button is pressed.
PROM	
Generates formatted code on the paper tape punch from an object file. Applicable PROM writers are those which are supplied by Britronics, Intel, Takeda and Minato Electronics.	
Example:	PROM ↵
SIGN [\$FDn]	
Changes the password of the disk in \$FDn.	
During a disk copy or formatting operation, the system checks the programmer-specified password with that stored in the disk directory for a match and carries out the specified operation only when a match occurs.	
Example:	SIGN ↵ : Changes the password of the disk currently logged on.
STATUS devicename, status	
Sets the status of the I/O device identified by devicename to status.	
Example:	STATUS \$SIA, \$1234 ↵ : Sets the control status of serial input port A to 1234 (hexadecimal).
VERIFY filename1, filename2 [, ..., filenameN-1, filenameN] (?, * only for filename1, ..., filenameN-1)	
Compares the contents of files filename1 through filenameN.	
Global switch /P	: Specifies that the results of the comparison are to be listed on the LPT.
Example:	VERIFY \$CMT, \$FD2 ; ABC ↵ : Compares the first file on the cassette tape with source file ABC in FD2.

—System Error Messages—

There are four system error message formats.

— ERR: error message

Pertains mainly to coding errors. Issued when invalid commands are detected.

— ERR filename (device name) : error message

Indicates errors pertaining to file or device specifications.

— ERR logical number: error message

Indicates errors pertaining to logical number specifications.

— ERR logical number file name (device name): error message

Indicates errors pertaining to logical number specifications and file (or device) specifications.

The system error messages are listed below. The error numbers are not output.

ERR- 1	syntax	
2	il command	
3	il argument	
4	il global switch	
5	il data	
6	il attribute	; Illegal file attribute found
7	different file mode	
8	il local switch	
9	il device switch	
10		
11	no usable device	; Device unavailable
12	double device	
13	directory in use	
14		
15		
16	not enough arguments	
17	too many argument	
18		
19		
20	no memory space	
21	memory protection	
22	END ?	
37	Break	
38	system id	; Disk not conforming to Floppy DOS format.
39	System error	; System malfunction, user program error, disk replaced improperly, etc.

50	not found	
51	too long file	; File size exceeds 65535 bytes
52	already exist	
53	already opened	; The file has been already opened or
54	not opened	the logical number is already used.
55	read protected	
56	write protected	
57	permanent	
58	end of file	
59	no byte file	
60	not ready	
61	too many files	; Number of files exceeds 96
62	disk volume	; Disk replaced improperly
63	no file space	
64	unformat	; Disk unformatted
65	FD hard error	; Hardware related disk error
66	il data	
67	no usable disk	
68	(sub)master disk	
69	mismatch sign	
70	il file name	; Invalid file name
71	il file attribute	; Invalid file attribute
72	il file type	; Invalid file type
73	il file mode	; Invalid file mode
74	il lu#	; Invalid logical number
75	not ready	} ; Printer error
76	alarm	
77	paper empty	
78	time out	} ; Paper tape reader or punch error
79	parity	
80	check sum	
81	flaming	} ; Serial I/O errors (to be implemented later)
82	over run	
83	interconnect	
84	full buffer	
85	uncontrollable	} ; IEEE-488 related errors (to be implemented later)
86	interface	
87	less data	
88	much data	
89	lu table overflow	; Attempt made to open too many files
90	source ?	
91	destination ?	
92	can't xopen	
93	too long line	; Line exceeding 128 bytes
94	end of record	
95	diff record length	

—Editor Commands—

Command type	Command name	Function
Input command	R	Clears the edit buffer and loads it with the input file indicated by the filename. The CP is positioned at the beginning of the edit buffer after execution of this command.
	A	Appends the input file indicated by the filename to the contents of the edit buffer. The CP position is not changed.
Output command	W	Writes the edit buffer contents to the output file specified by the filename in ASCII code.
Page processing command	PR	Loads the remainder of a file whose beginning has been loaded with the R or A command. The PR command clears the edit buffer before the data is loaded and the PA command adds the data to the current contents of the edit buffer.
	PA	
	PW	Same as the W command, except that the maximum amount of data output is 1 page.
	PC	Terminates execution of the processing command. This command is required whenever a PR, PA or PW command is executed.
	PK	Kills the file opened by a page processing command.
Type command	T	Displays the entire contents of the edit buffer. The CP position is not changed.
	nT	Displays n lines starting at the CP position.
CP positioning command	B	Positions the CP at the beginning of the edit buffer.
	nJ	Positions the CP at the beginning of the line indicated by n.
	nL	Moves the CP to the beginning of the line n lines after the current CP position.
	L	Moves the CP to the beginning of the current line. This is the same as when n = 0 in the nL command.
	nM	Changes the CP position by n characters.
	M	Does not move the CP. This is the same as when n = 0 in the nM command.
Correction command	Z	Moves the CP to the end of the text in the edit buffer.
	C	Searches for the specified character string and replaces it with another character string; the search starts at the current CP position and proceeds to the end of the edit buffer. The CP is repositioned to the end of the character string replaced.
	Q	Repeats the C command each time the specified character string is found until the end of the edit buffer is reached. The CP is repositioned to the end of the character string last replaced.
	I	Inserts the specified character string at the position of the CP. The CP is repositioned to the end of the character string inserted. Line numbers are updated when a line is inserted with this command.
	nK	Deletes the n lines following the CP. The CP position is not changed.
	K	Deletes all characters preceding the CP position until a [CR] code is detected. The [CR] code is not deleted.
	nD	Deletes the n characters following the CP.
	D	No operation
Search command	S	Searches for the specified character string, starting at the CP position and proceeding to the end of the buffer. The CP is repositioned to the end of the character string when it is found.
Special command	\	Executes the specified built-in command.
	=	Displays the number of characters stored in the edit buffer (including spaces and CRs).
	.	Displays the number of the line at which the CP is located.
	&	Deletes the entire contents of the edit buffer.
	#	Changes the list mode for listing to the printer.
	!	Transfers control to the Floppy DOS.

Most of the above commands are compatible with those used in the NOVA editor program manufactured by the Data General Corporation.



—Assembler Messages—

Definition status message	Meaning	Example
E (External)	Indicates that a label symbol is being referenced externally; that is, the label is not defined in the current source program unit.	<pre> E LD B, CONST0 ↑ The data byte "CONST0" is undefined. E CALL SORT ↑ The address "SORT" is undefined. EE BIT TOP, (IY+FLAG) ↑↑ The data byte "FLAG" is undefined. The data byte "TOP" is undefined. </pre>
P (Phase)	Defines a label symbol with a constant assigned. This message is also output when a label symbol is encountered during pass 2 which was not encountered during pass 1.	<pre> P LETNL : EQU 0762H P DATA1 : EQU 3 ↑ LETNL and DATA1 are defined by EQU. The P message is displayed in the relocatable binary code column rather than in the assembler message column. </pre>

Error message	Meaning	Example
C (illegal Character error)	Indicates that an illegal character is used in the operand.	<pre> C JP +1000-3 </pre>
F (Format error)	Indicates that the instruction format is incorrect.	
N (Non label error)	Indicates that no label symbol is specified for ENT or EQU.	<pre> N EQU 0012H ↑ No label symbol </pre>
L (erroneous Label error)	Indicates that an illegal label symbol is used.	<pre> L JR XYZ ↑ XYZ is not defined in the current program. No externally defined global symbol can be used as the operand of a JR or DJNZ command. If such a label symbol is specified, the L message is displayed. </pre>
M (Multiple label error)	Indicates that a label symbol is defined two or more times.	<pre> M ABC : LD DE, BUFFER ? M ABC : ENT ↑ ABC is defined twice. </pre>
O (erroneous Operand)	Indicates that an illegal operand is specified.	
Q (Questionable mnemonic)	Indicates that the mnemonic code is incorrect.	<pre> Q CAL XYZ CALL XYZ is correct. </pre>
S (String error)	Indicates that single or double quotation mark(s) are omitted.	<pre> S DEFM GAME OVER DEFM 'GAME OVER' is correct. </pre>
V (Value over)	Indicates that the value of the operand is out of the prescribed range.	<pre> V LD A, FF8H V SET 8, A V JR -130 </pre>
END?	Indicates that the END directive is missing from the source program.	

Note: Refer to the System Error Messages in the System Command manual for other system errors.

—Symbolic Debugger Commands—

Command type	Command name	Function
Symbol table command	T	Displays the contents of the symbol table; i.e., the label symbol name, its absolute address and the definition status for each table entry. (Table Dump)
Debugging commands	B[†]	Displays, sets or alters a breakpoint. (Breakpoint)
	&	Clears all breakpoints set. (Clear Breakpoints)
	M[†]	Displays the contents of the specified block in the link area in hexadecimal representation or alters them. (Memory Dump)
	D[†]	Displays the contents of the specified block in the link area in hexadecimal representation with one instruction on a line. (Memory List Dump)
	W[†]	Writes hexadecimal data, starting at the specified address in the link area. (Write)
	G[†]	Executes the program at the specified address. (GOTO)
	I	Executes the program at the address designated by PC with the register buffer data set to the CPU internal registers. (Indicative Start)
	A	Displays the contents of registers A, F, B, C, D, E, H and L in hexadecimal representation or alters them. (Accumulator)
	C	Displays the contents of complementary registers A', F', B', C', D', E', H' and L' in hexadecimal representation or alters them. (Complementary)
	P	Displays the contents of registers PC, SP, IX, IY and I in hexadecimal representation or alters them. (Program Counter)
File I/O commands	R	Displays the contents of all registers in hexadecimal representation. (Register)
	X	Transfers the specified memory block to the specified address. (Transfer)
	S	Saves the object program in the link area in an output file with the specified name. (Save)
Special commands	Y	Reads the object program from the object file with the specified file name into memory. (Yank)
	\	Executes the specified DOS built-in command.
	# !	Switches the printer list mode for listing printout. Transfers control to Floppy DOS.

Note: Commands marked by a dagger permit symbolic operations.

—Symbolic Debugger Error Messages—

Error message	Description	Related commands
???	<ul style="list-style-type: none"> The command operand fields does not match the 4-digit hexadecimal format. A symbolic label is missing. A data defining symbol is used as a label. 	M, D, W, B, G
Error	<ul style="list-style-type: none"> An invalid number of digits was entered when altering register or memory contents, or a key other than 0 through 9 or A through F was pressed. 	A, C, P, M
DJNZ?	A breakpoint was set for a DJNZ instruction.	B
CALL?	A breakpoint was set for a CALL instruction.	B
RST 6?	A breakpoint was set for a RST 6 instruction.	B
Over	An attempt was made to set more than 9 breakpoints.	B
?	<ul style="list-style-type: none"> An attempt was made to access outside the link area. The starting address is greater than the ending address. An attempt was made to clear an undefined breakpoint. The breakpoint counter was set to F (the maximum permissible value is E in hexadecimal). 	M, D, W, B, G, X M, D B B

Note: Refer to the System Error Messages in the System Command manual for other system error messages.

—PROM Formatter Commands—

COMMAND		OPERATION
File Input/ Output commands	Y (Yank)	Loads a program (data) from the disk into the free area.
	S (Save)	Saves the program (data) in the free area on disk.
	CY (Yank disk)	Loads data in 256-byte units from the specified sector(s) of the specified track on the disk into RAM.
	CS (Save disk)	Saves data in 256-byte units from RAM memory in the specified sector(s) of the specified track of the disk.
Format commands	P (Punch)	Punches the specified contents of the free area in the specified format.
	R (Read)	Reads in a paper tape punched in the format specified.
Other commands	M (Memory)	Displays and modifies data in the free area.
	V (Verify)	Reads data from the paper tape reader and compares it with the contents of the RAM free area.
	\ (FDOS)	Executes the specified built-in DOS command.
	#	Switches the list mode for listing on a printer.
	& (Clear)	Buries all data in the free area in hexadecimal code FFH.
	?	Displays the starting and ending addresses of the free area.
	! (Return)	Returns control to Floppy DOS.

Error message	Error content	Related command
memory protection	An address outside of the free area was specified.	Y, S, P, R, M, V
il command	The command was not entered correctly.	
il data	The format specified does not match the format read.	R, V
check sum	Check sum error.	R, V
\$ LPT : not ready	The printer is not ready.	#
\$ PTP : not ready	The paper tape punch is not ready.	P
\$ PTR : not ready	The paper tape reader is not ready.	R, V

See the "System Error Messages" in System Command for other error messages.

Caution:

Entry of characters other than S, Y, CS, CY, P, R, M, V, \, &, #, ? or ! will cause a return to the command wait state after the command table is displayed.

If a character other than A~H is input while "format?" is displayed and format entry awaited, the format table will be displayed, after which the format entry wait state will be reentered. A return can be made to the command wait state at this time by pressing **BREAK**.

—FILE Mode—

File mode	Meanings
.ASC	ASCII file. A source file generated by the text editor or a file containing ASCII character strings generated by a BASIC interpreter.
.RB	Relocatable file. A file containing pseudo-machine language code (relocatable binary code) which can be loaded into any location in memory. It is generated by the assembler or the compiler.
.OBJ	Object file. A file containing Z-80 machine language codes.
.LIB	Library file. A file into which Floppy DOS links multiple relocatable files.
.SYS	System file. A file containing a system program runs under Floppy DOS and which contains relocatable binary codes (such as the text editor and the assembler).

—I/O Devices Handled by Floppy DOS—

\$KB : MZ-80B system keyboard
 \$CRT : MZ-80B system display unit
 \$FD1 :
 \$FD2 : } Floppy disk drives (MZ-80FB or MZ-80FBK)
 \$FD3 : }
 \$FD4 :
 \$CMT : System cassette tape deck
 \$LPT : System printer (MZ-80P4 or MZ-80P5)
 \$MEM : A part of MZ-80 main memory
 \$PTR : Paper tape reader
 \$PTP : Paper tape punch
 \$SIA : Serial input port A
 \$SIB : Serial input port B
 \$SOA : Serial output port A
 \$SOB : Serial output port B
 \$USR1 :
 \$USR2 : } User devices 1 ~ 4
 \$USR3 : }
 \$USR4 :
 \$CMT1: Cassette tape deck for MZ-80K

—File Attributes—

File attributes are information pertaining to file protection. There are four types of file attribute: 0, R, W and P. File attribute 0 indicates that a file is not protected. The other attributes inhibit the use of specific commands as indicated below.

File attribute	R	W	P	
Inhibited FDOS commands	TYPE XFER EDIT ASM LINK DEBUG PROM BASIC	DELETE RENAME	TYPE XFER EDIT ASM LINK DEBUG PROM BASIC DELETE RENAME	0: No file protection R: Read-inhibited file W: Write-inhibited file P: Permanent file
Inhibited BASIC statements	ROPEN# INPUT#()	PRINT#()	ROPEN# INPUT#() PRINT#()	

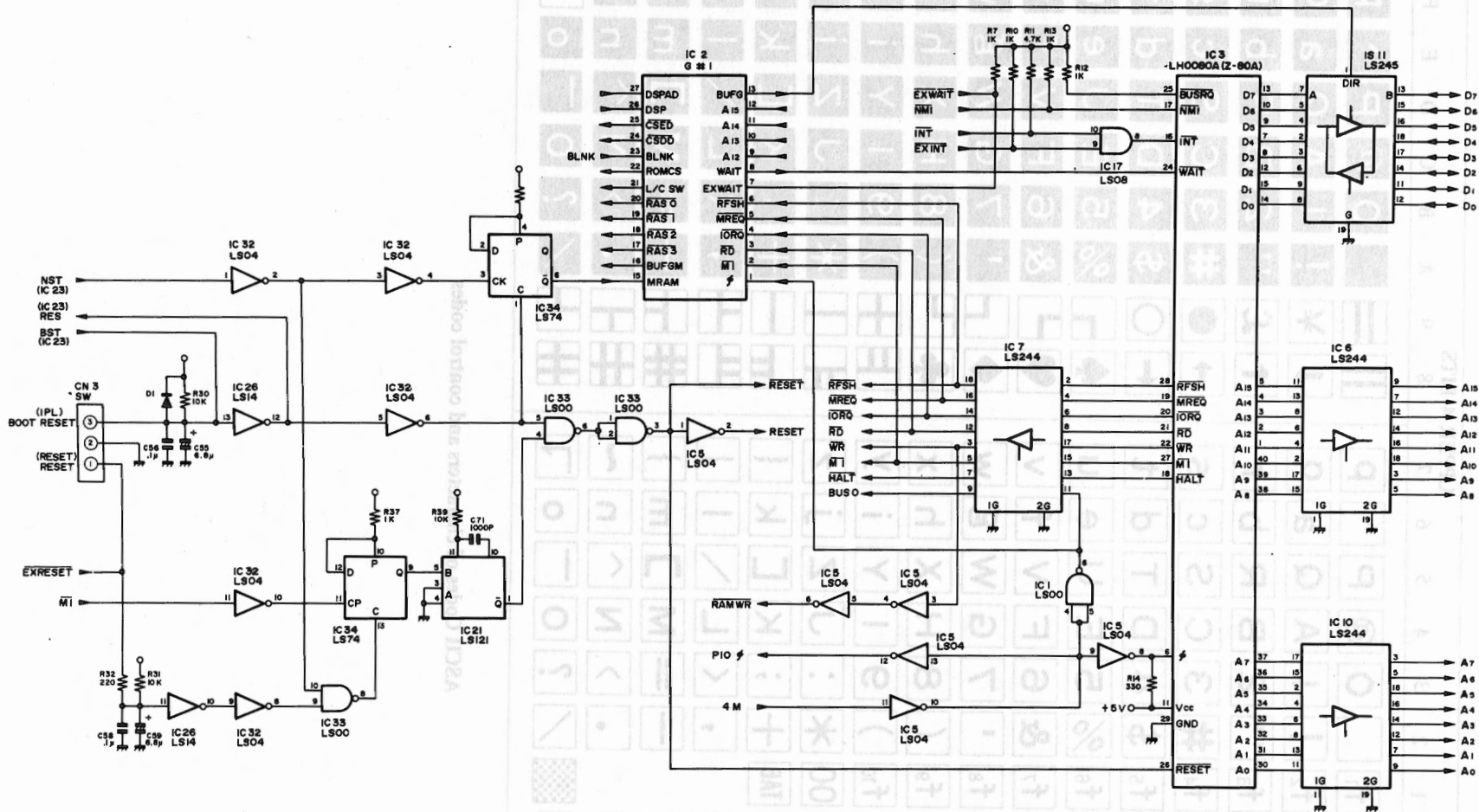
ASCII CODE TABLE

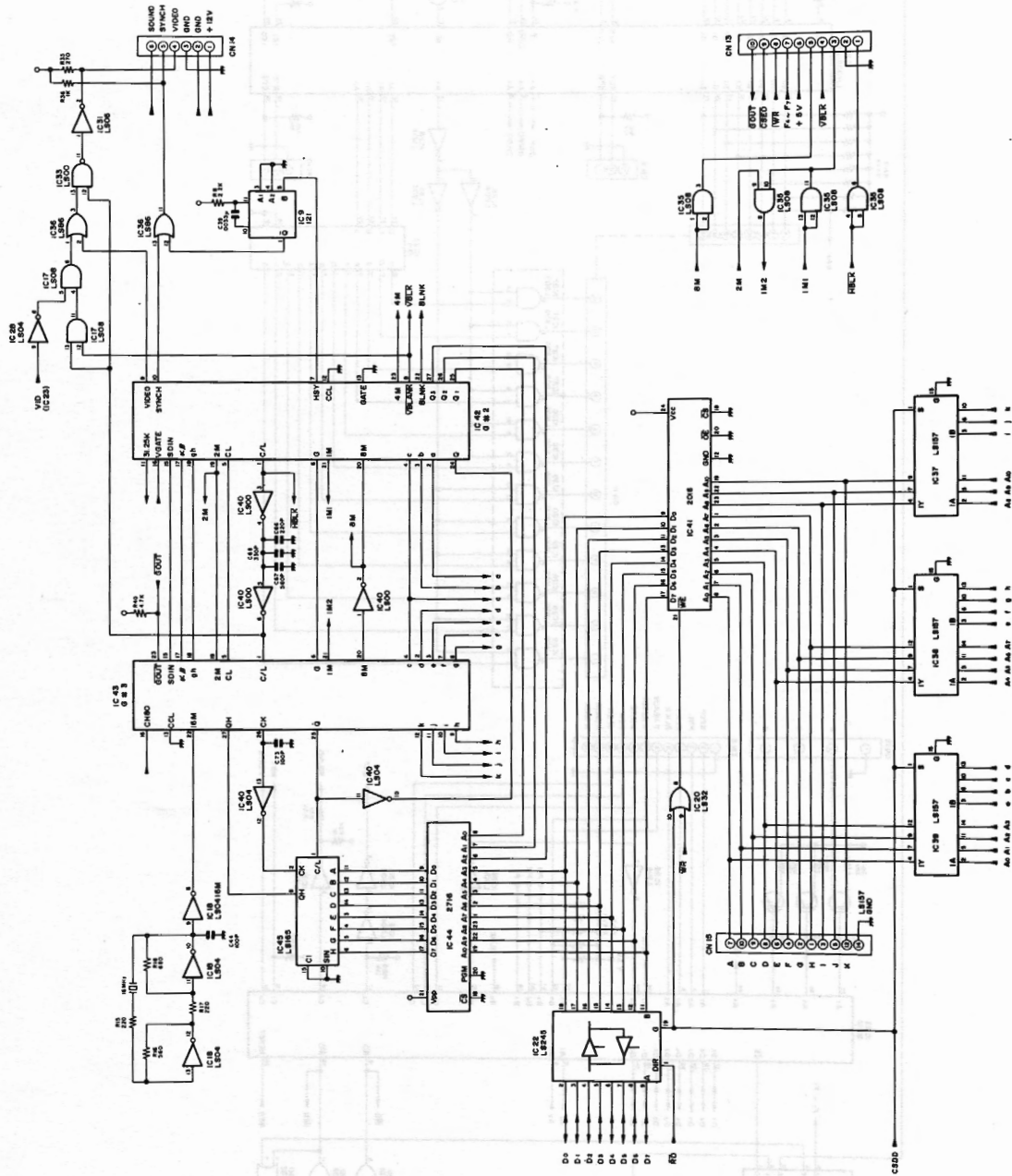
		UPPER 4 BITS															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
LOWER 4 BITS	0	NULL	f ₁		O	@	P	`	p		=		O	@	P	`	p
	1	↓	f ₂	!	I	A	Q	a	q	↓	¥	!	I	A	Q	a	q
	2	↑	f ₃	"	2	B	R	b	r	↑	£	"	2	B	R	b	r
	3	→	f ₄	#	3	C	S	c	s	→	●	#	3	C	S	c	s
	4	←	f ₅	\$	4	D	T	d	t	←	○	\$	4	D	T	d	t
	5	HOME	f ₆	%	5	E	U	e	u	♠	☐	%	5	E	U	e	u
	6	CLR	f ₇	&	6	F	V	f	v	♥	☐	&	6	F	V	f	v
	7	DEL	f ₈	'	7	G	W	g	w	♦	☐	'	7	G	W	g	w
	8	INST	f ₉	(8	H	X	h	x	♣	☐	(8	H	X	h	x
	9	GRPH	f ₁₀)	9	I	Y	i	y	⚡	⚡)	9	I	Y	i	y
	A	SFT LOCK	00	*	:	J	Z	j	z	⚡		*	:	J	Z	j	z
	B	BREAK	TAB	+	;	K	[k	{	⚡	—	+	;	K	[k	{
	C	RVS		,	<	L	\	l		⚡	⚡	,	<	L	\	l	
	D	CR		—	=	M]	m	}	⚡	⚡	—	=	M]	m	}
	E	SCRIPT		.	>	N	^	n	~	⚡	⚡	.	>	N	^	n	~
	F	RVS CANCEL	☐	/	?	O	_	o	↓	⚡	⚡	/	?	O	_	o	π

ASCII Codes of characters and control codes

MZ-80B CIRCUIT DIAGRAMS

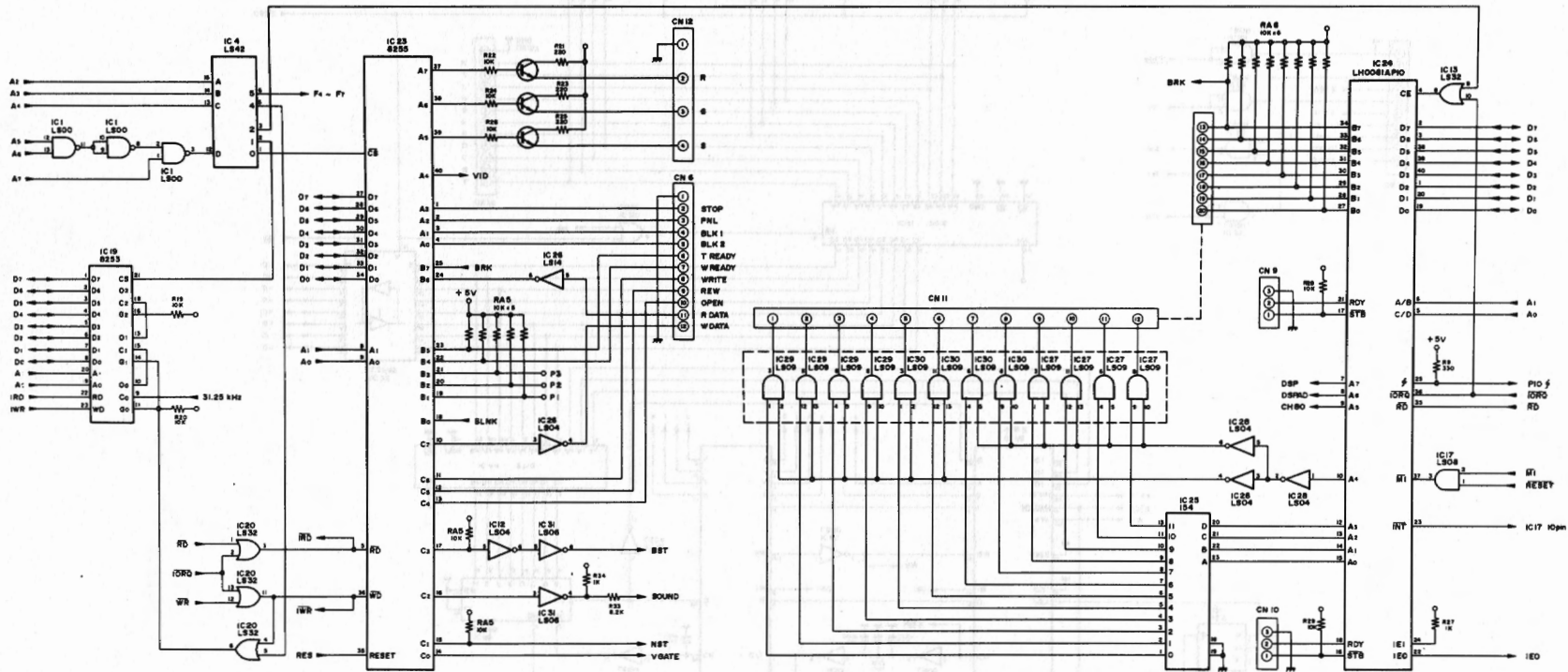
CPU board, block 1 : CPU signal system





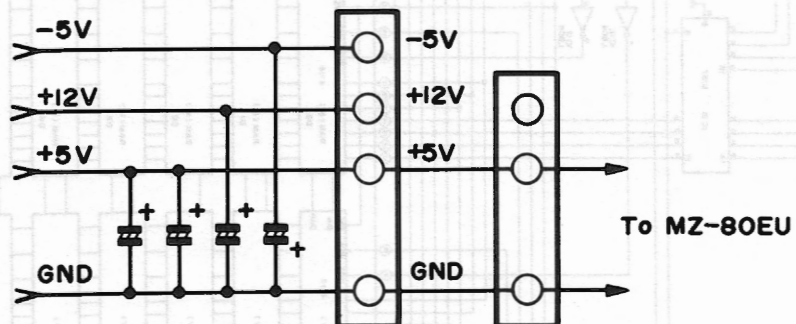
CPU board, block 3 : 8255 signal system

APPENDIX-36



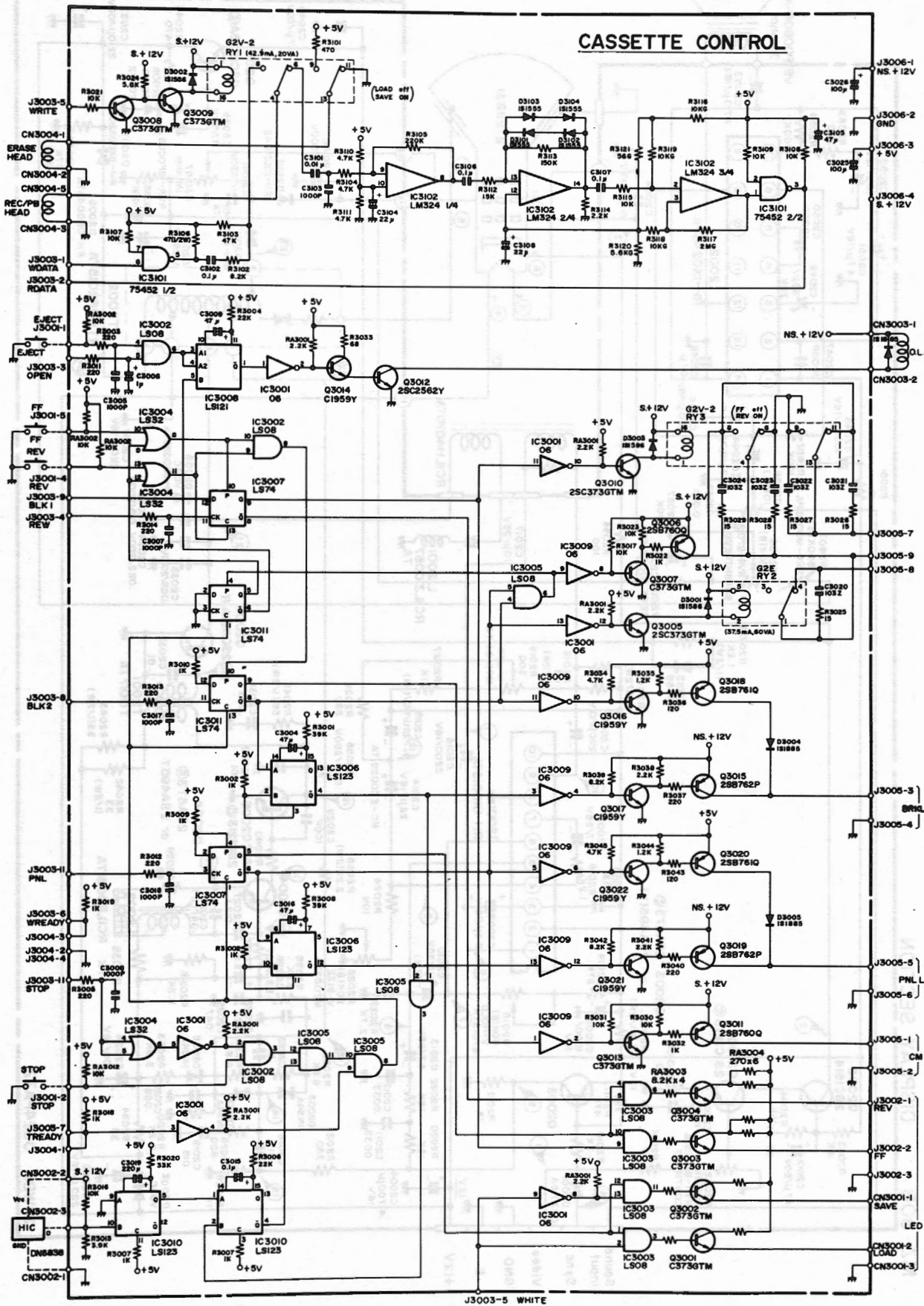
CN5 → MZ-80GM

CN4, 5		40P	
1	A15	2	A14
3	A13	4	A12
5	A11	6	A10
7	A9	8	A8
9	GND	10	A7
11	A6	12	A5
13	A4	14	A3
15	A2	16	A1
17	A0	18	GND
19	D7	20	D6
21	D5	22	D4
23	D3	24	D2
25	D1	26	D0
27	GND	28	$\overline{\text{NMI}}$
29	$\overline{\text{EX WAIT}}$	30	$\overline{\text{EX INT}}$
31	$\overline{\text{EX RESET}}$	32	RESET
33	$\overline{\text{IEO}}$	34	$\overline{\text{HALT}}$
35	$\overline{\text{MREQ}}$	36	$\overline{\text{IOREQ}}$
37	$\overline{\text{RD}}$	38	$\overline{\text{WR}}$
39	$\overline{\text{MT}}$	40	BUS0

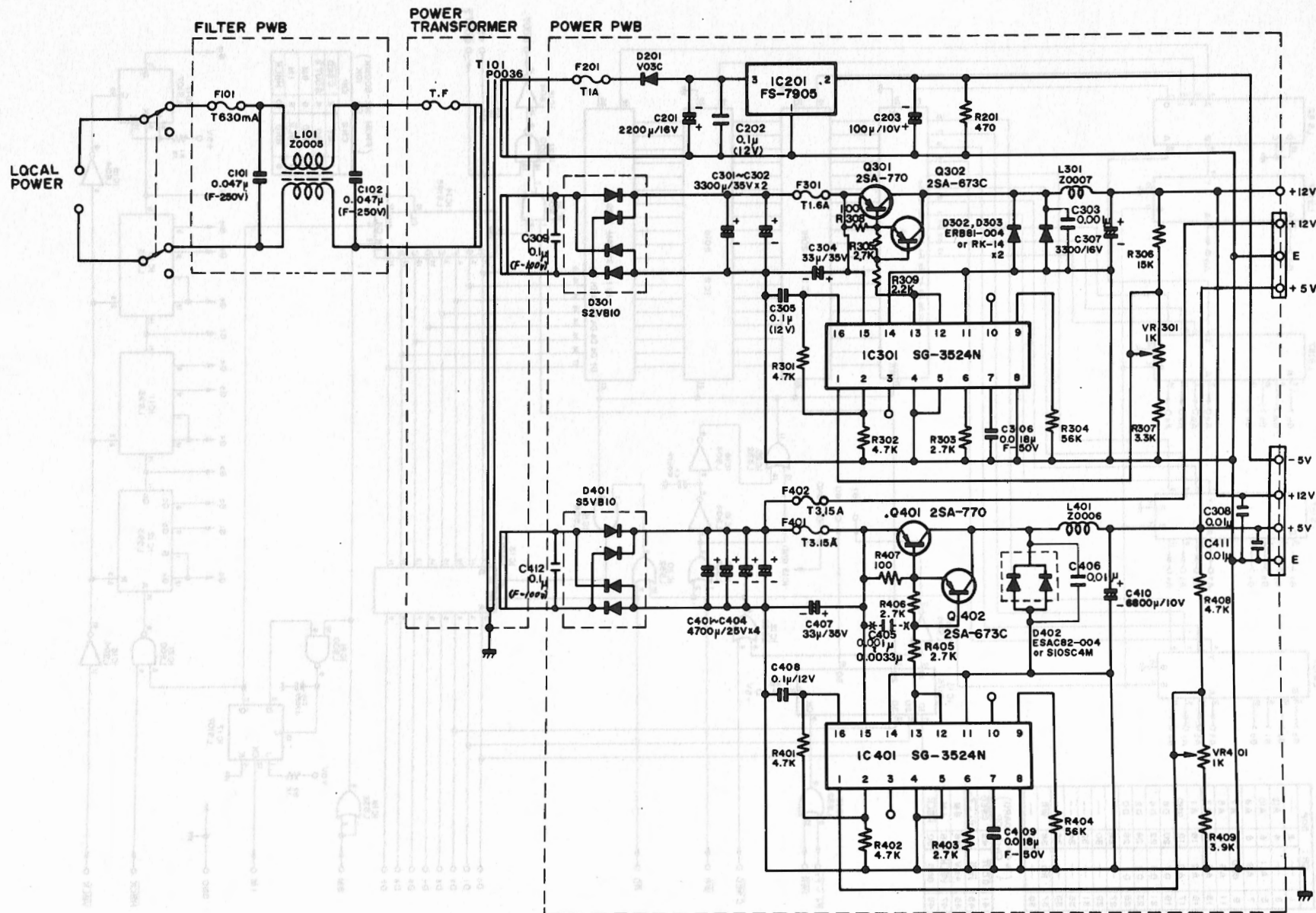


CRT display control





Cassette tape deck control

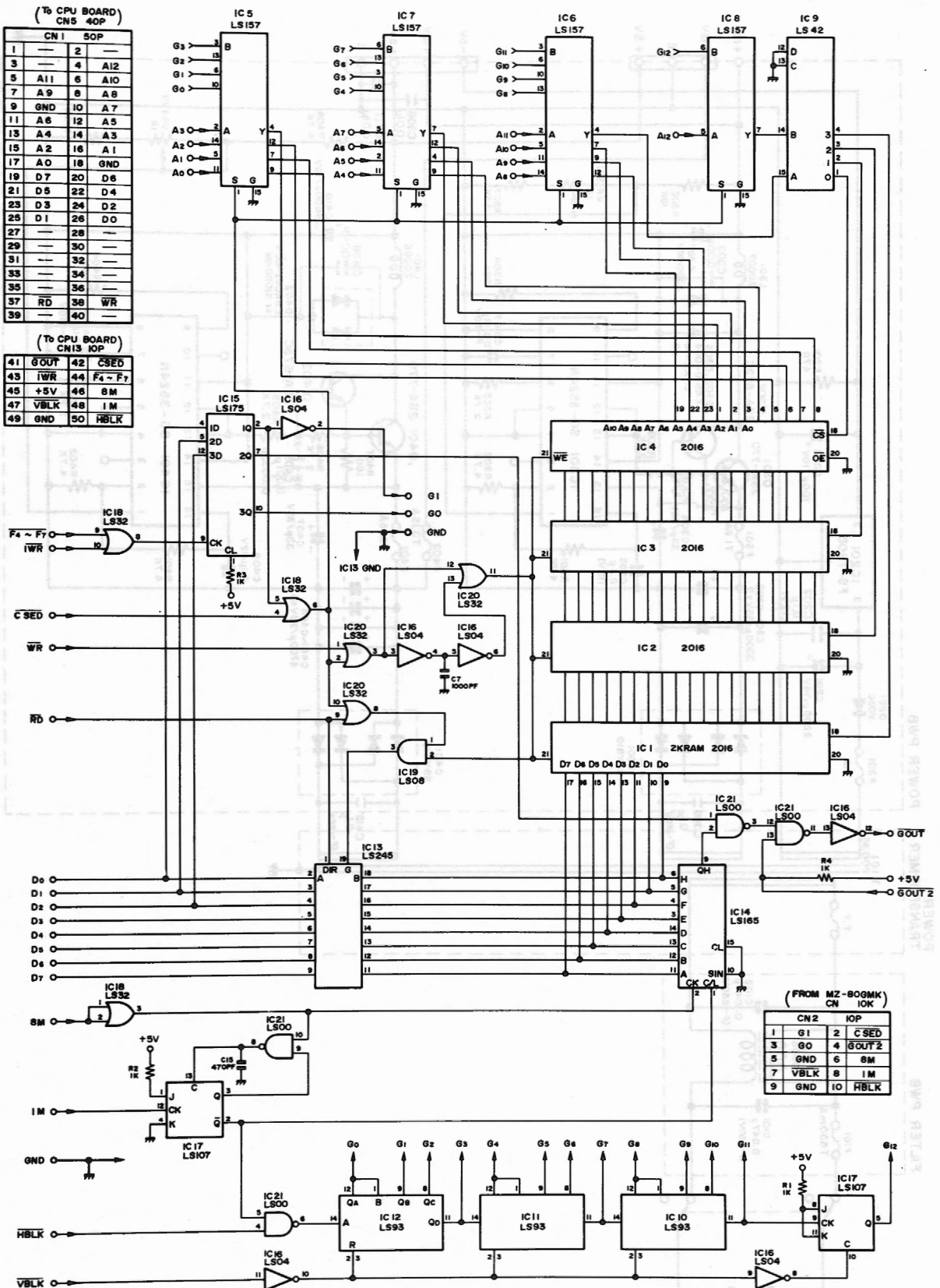


(To CPU BOARD)
CN5 40P

CN1 50P	
1	2
3	4
5	A11
6	A10
7	A9
8	A8
9	GND
10	A7
11	A6
12	A5
13	A4
14	A3
15	A2
16	A1
17	A0
18	GND
19	D7
20	D6
21	D5
22	D4
23	D3
24	D2
25	D1
26	D0
27	28
29	30
31	32
33	34
35	36
37	RD
38	WR
39	40

(To CPU BOARD)
CN13 10P

41	GOUT
42	CSED
43	IWR
44	F4 ~ F7
45	+5V
46	8M
47	VBLK
48	IM
49	GND
50	HBLK

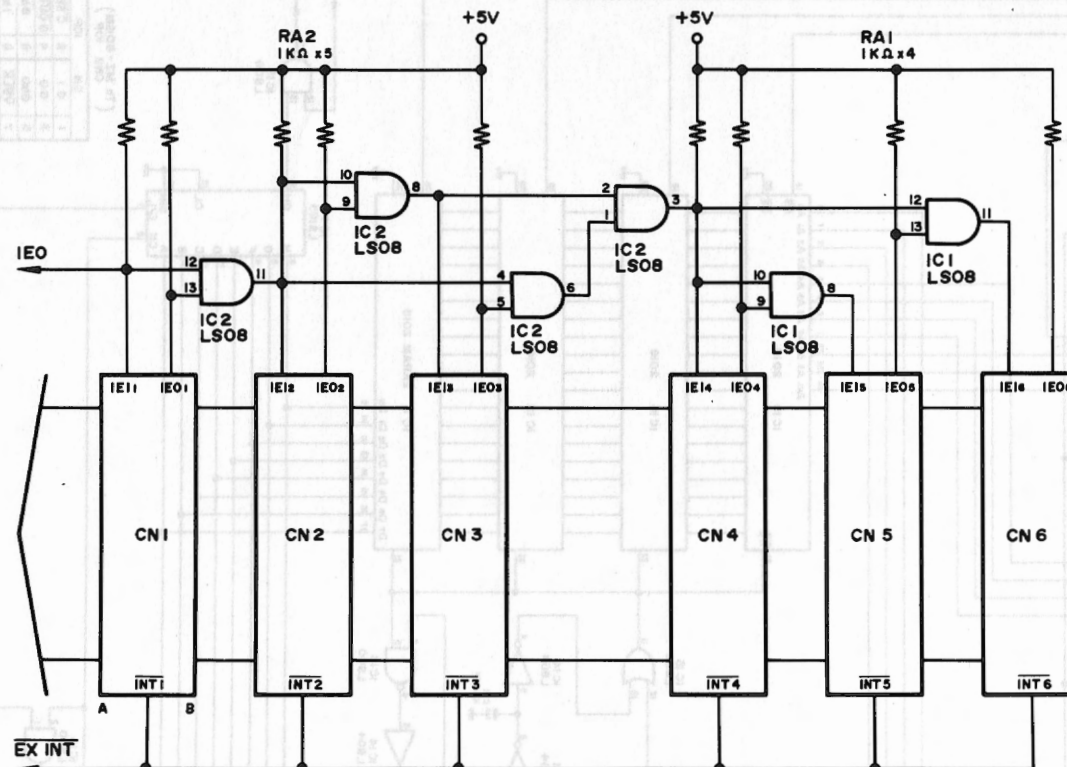


Graphic Memory 1 card (optional) MZ-80GM

Expansion I/O port (optional) MZ-80EU

(To CPU BOARD)
CN4 40P

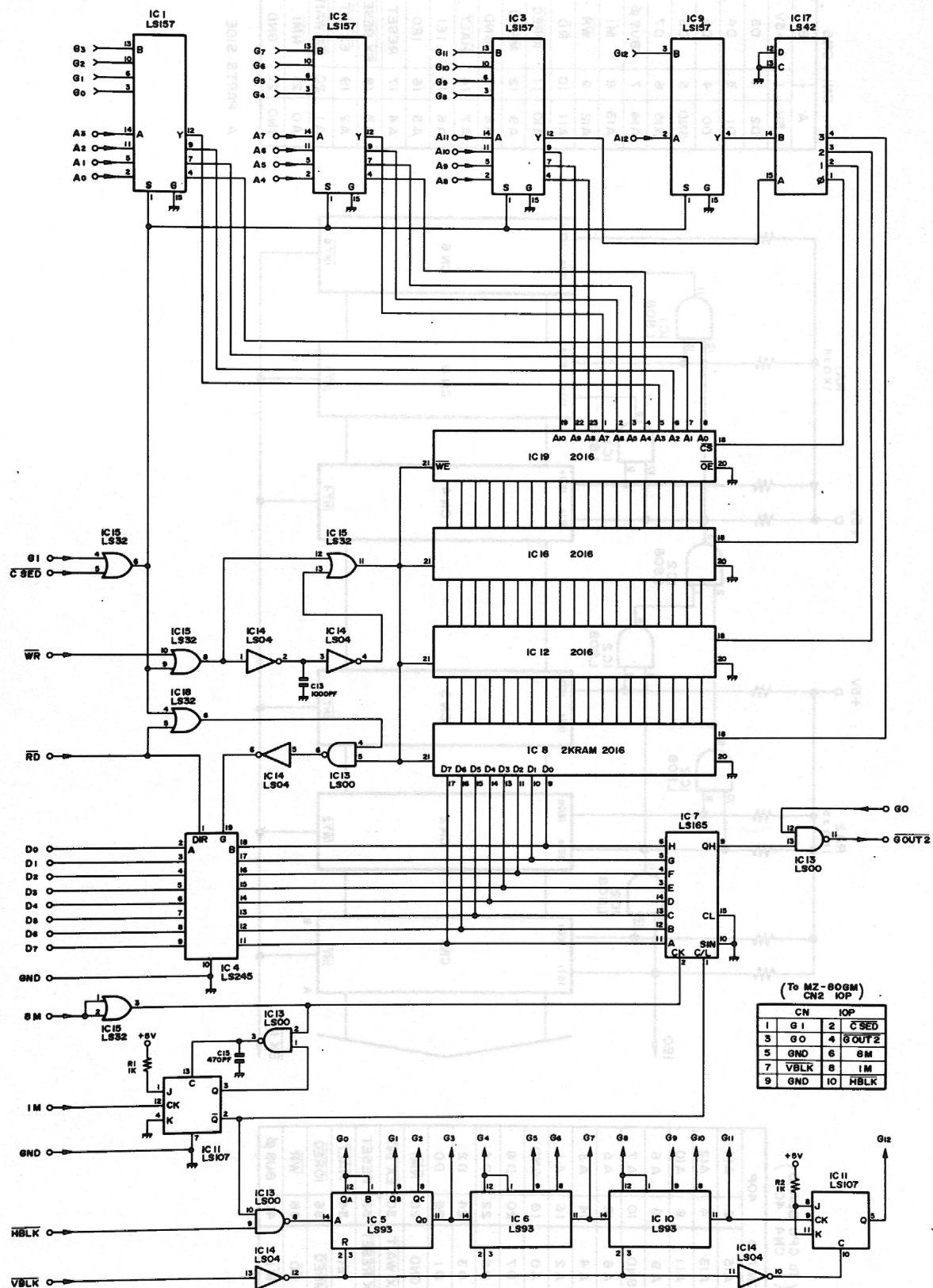
CN 7, 8 40P			
1	A15	2	A14
3	A13	4	A12
5	A11	6	A10
7	A9	8	A8
9	GND	10	A7
11	A6	12	A5
13	A4	14	A3
15	A2	16	A1
17	A0	18	GND
19	D7	20	D6
21	D5	22	D4
23	D3	24	D2
25	D1	26	D0
27	GND	28	NMI
29	EX WAIT	30	EX INT
31	EX RESET	32	RESET
33	IEO	34	HALT
35	MREQ	36	IOREQ
37	RD	38	WR
39	MI	40	BUS ϕ



CN1 ~ CN6

A	B
+5V	1
D2	2
D1	3
D0	4
GND	5
A15	6
A14	7
A13	8
A12	9
A11	10
A10	11
A9	12
A8	13
A7	14
A6	15
A5	16
A4	17
A3	18
A2	19
A1	20
A0	21
GND	22

A: PARTS SIDE



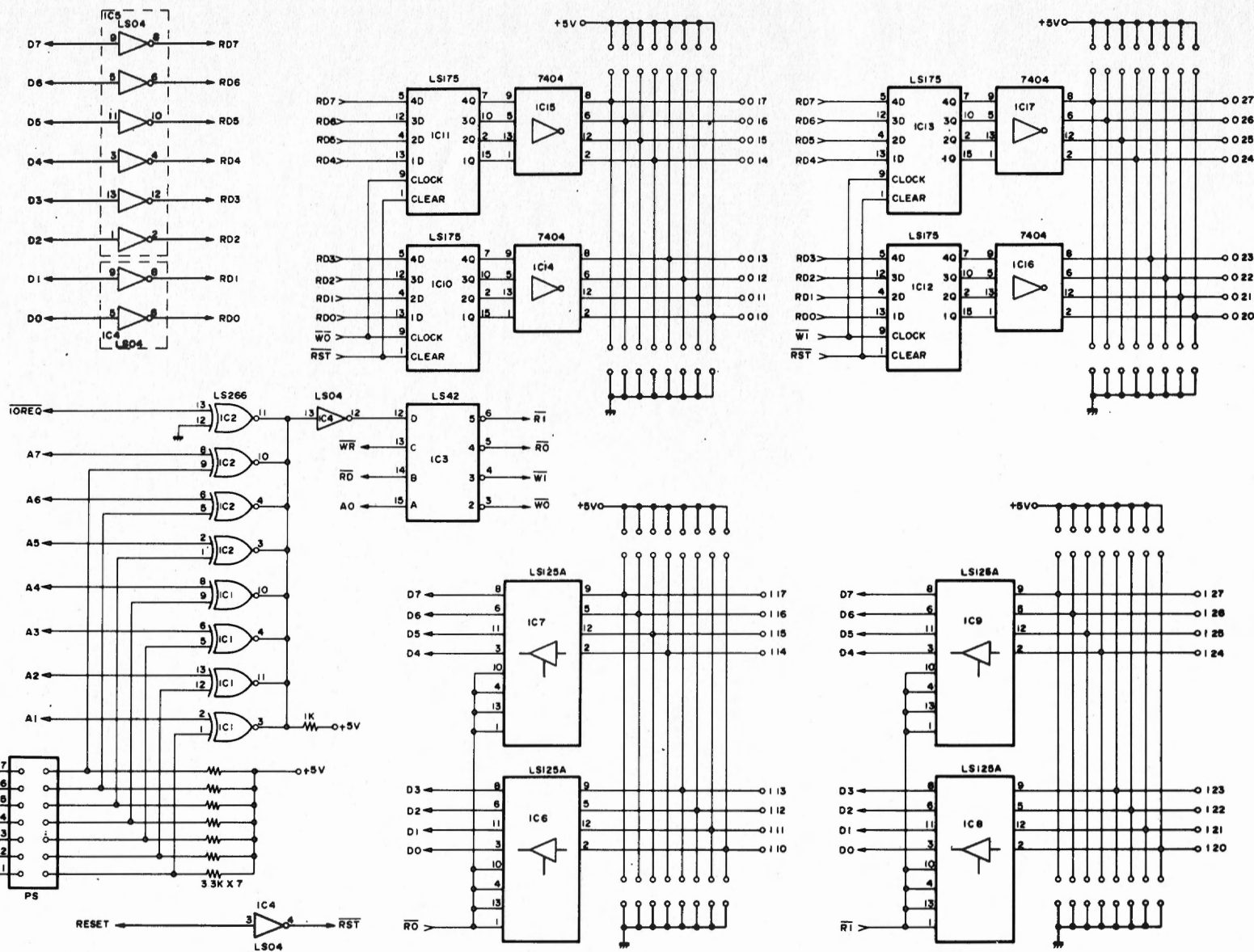
Graphic Memory 2 card (optional) MZ-80GMK

I/O CONNECTOR

A		B
+5V	1	+5V
D2	2	D3
D1	3	D4
D0	4	D5
GND	5	D6
A15	6	D7
A14	7	BUS ϕ
A13	8	WT
A12	9	WR
A11	10	RD
A10	11	IOREQ
A9	12	MREQ
A8	13	GND
A7	14	HALT
A6	15	IEI
A5	16	IEO
A4	17	RESET
A3	18	EX RESET
A2	19	EXINT
A1	20	EX WAIT
A0	21	NMT
GND	22	GND

A PARTS SIDE

SIGNAL TERMINAL			
GND	1	20	GND
O10	2	21	O11
O12	3	22	O13
O14	4	23	O15
O16	5	24	O17
GND	6	25	I10
I11	7	26	I12
I13	8	27	I14
I15	9	28	I16
I17	10	29	GND
O20	11	30	O21
O22	12	31	O23
O24	13	32	O25
O26	14	33	O27
GND	15	34	I20
I21	16	35	I22
I23	17	36	I24
I25	18	37	I26
I27	19		



UNIVERSAL I/O CARD

