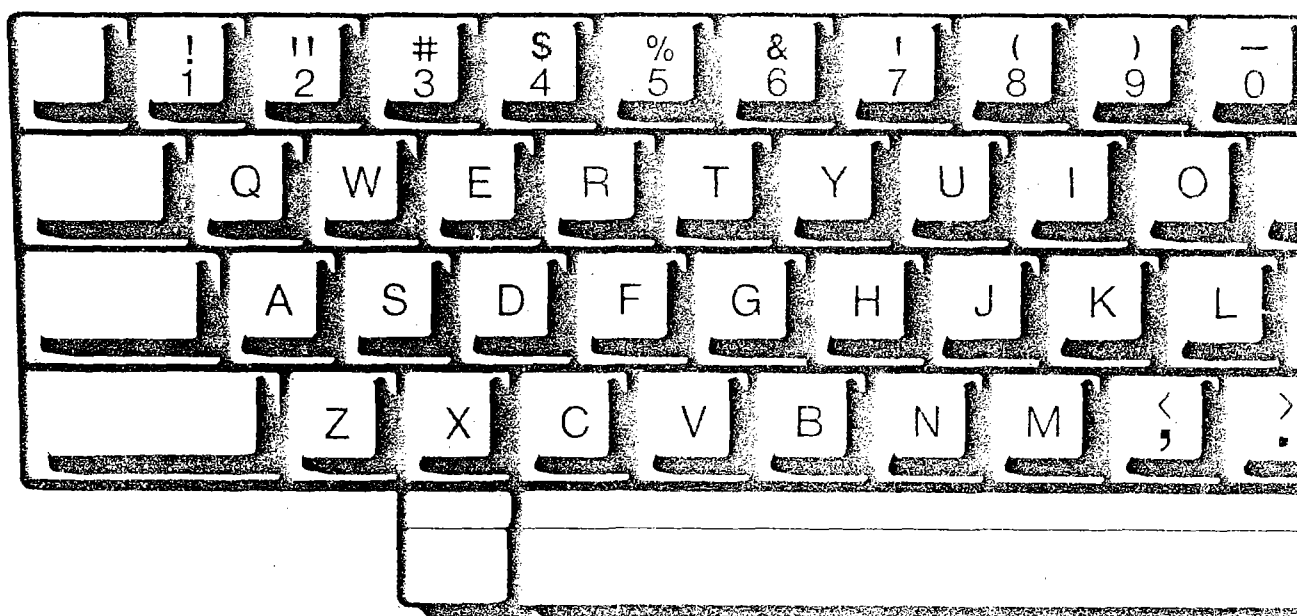
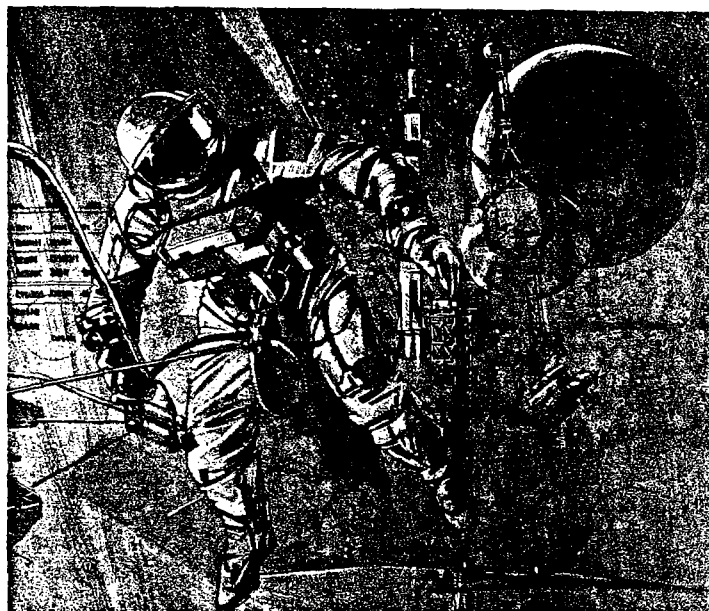


Personal Computer
MZ-800

TECHNICAL REFERENCE
MANUAL



SHARP®

Personal Computer
mz-800

**TECHNICAL REFERENCE
MANUAL**

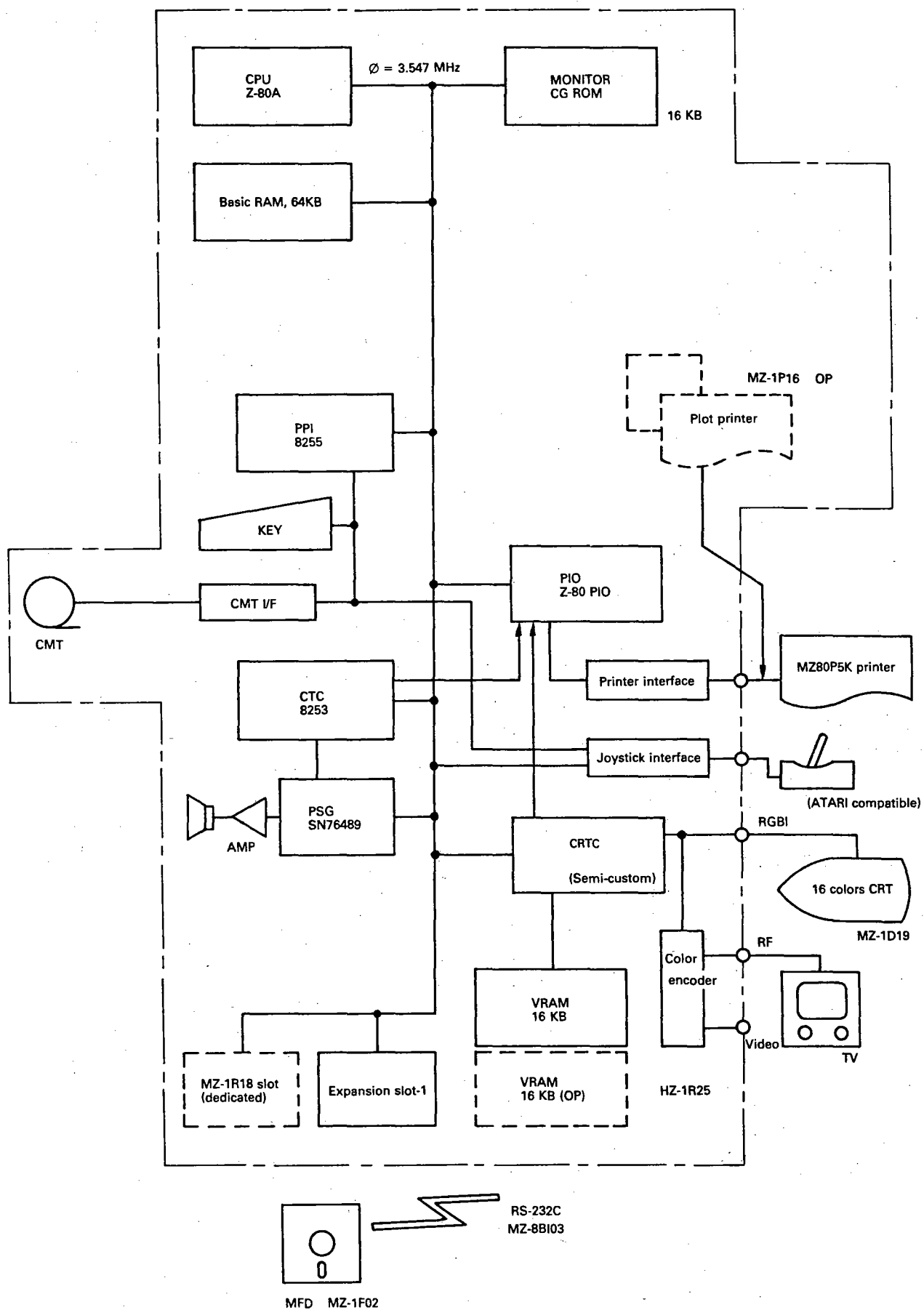
MZ-800 HARDWARE

1	System description	1
2	Block diagram	2
3	Memory map	3
4	Custom LSI	6
4-1	Memory controller	6
4-2	I/O controller	6
4-3	Clock generator and timing generator	9
4-4	Display address generator	9
4-5	Scroll	10
4-6	VRAM data input/output circuit	12
4-7	Register functions	17
4-8	Pallet	22
4-9	CRTC register map	23
4-10	ROM configuration	24
5	8255 Programmable Peripheral Interface	24
6	8253 Programmable Interval Timer	28
7	Printer interface	28
8	Programmable sound generator	31
9	Joystick	32
10	Power supply	33
10-1	Block diagram	33
10-2	Operational description	34
10-3	Maintenance	34
10-4	Problem determination and sequence	34
11	MZ-1P16	35
11-1	Installation	35
11-2	Block diagram	36
11-3	At power on	36
11-4	Colour change operation	36
11-5	Pen change operation	36
11-6	Pen exchange method	37
11-7	Stepping motor driving signal	37
11-8	Colour position detector	37
11-9	Character set	37
11-10	Colour plotter printer control LSI	38
11-11	Interfacing with the MZ-800	39
11-12	Block diagram	39
11-13	Colour plotter printer control codes	40
11-14	Graphic mode commands	41
11-15	Specification	42
11-16	Circuit	42
12	CPU P.W.B. Circuit	43

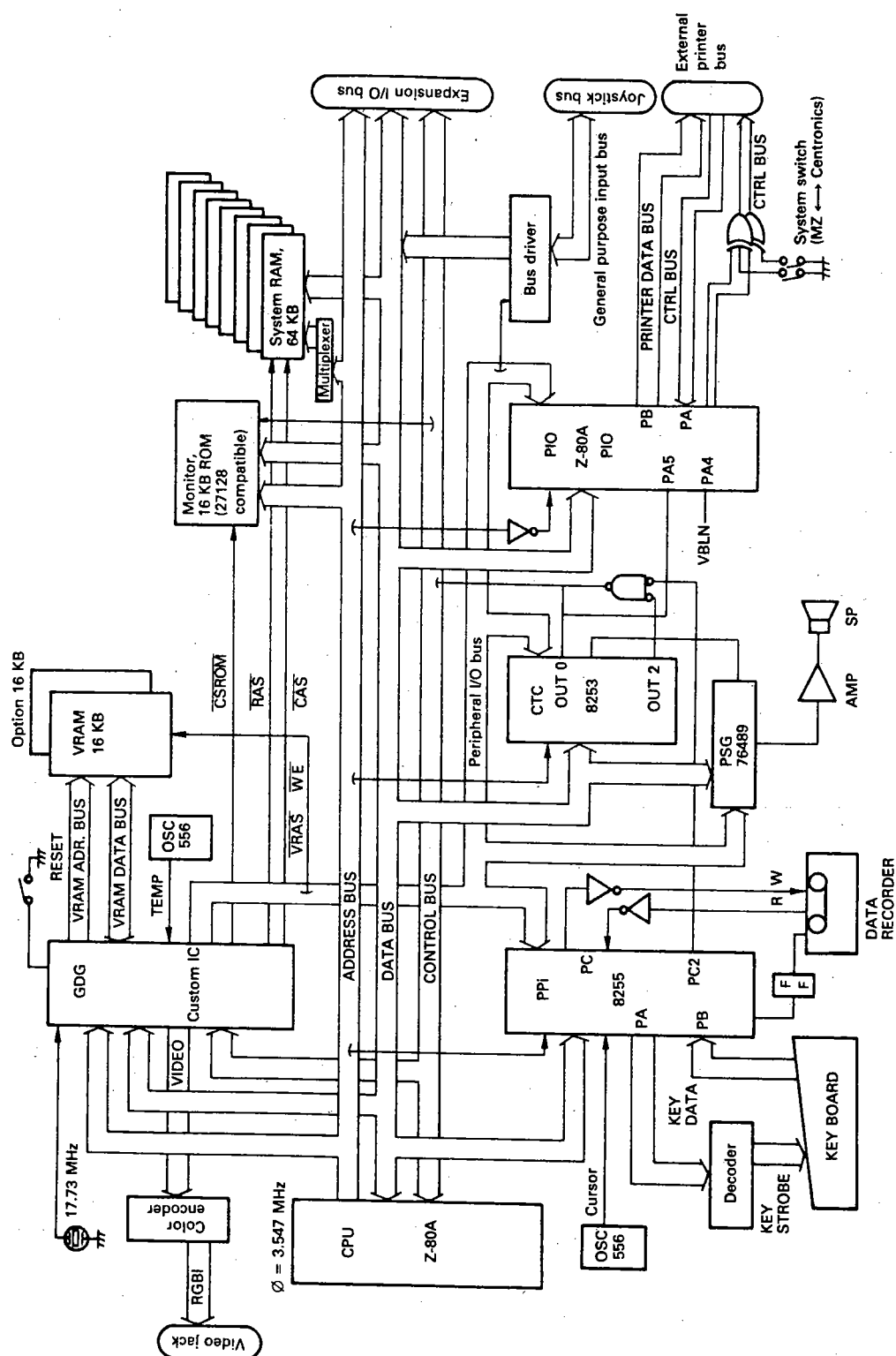
MZ-800 SOFTWARE

1	Character generator (C.G.)	45
1-1	How to modify C.G. ROM	45
1-2	C.G. Table	46
1-3	C.G. ROM dump list	48
2	MZ-800 Monitor subroutines	52
3	Assemble list	67
3-1	Monitor <1Z-013B>	67
3-2	MZ Disk control <9Z-504M-V1.0C>	100
3-3	Mini-floppy disk control	111
3-4	ROM Monitor	117
4	Z-80 Programming reference	138
4-1	Z-80 Status indicators (Flags)	138
4-2	Notation	140
4-3	Z-80 Instruction set	141
	INDEX	173

1 System description



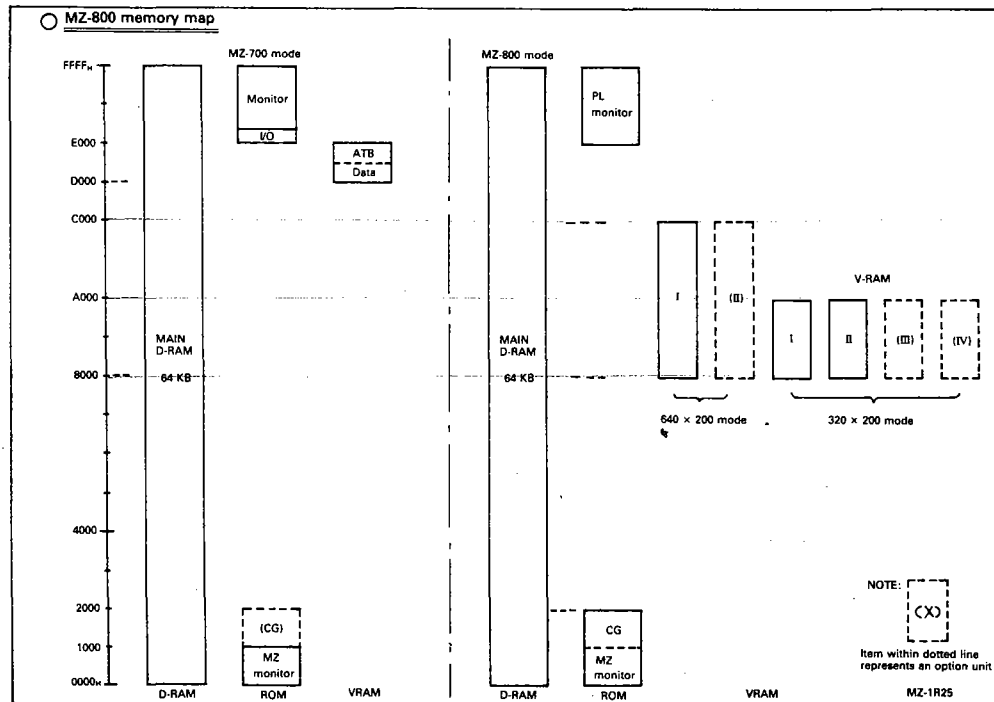
2 Block diagram



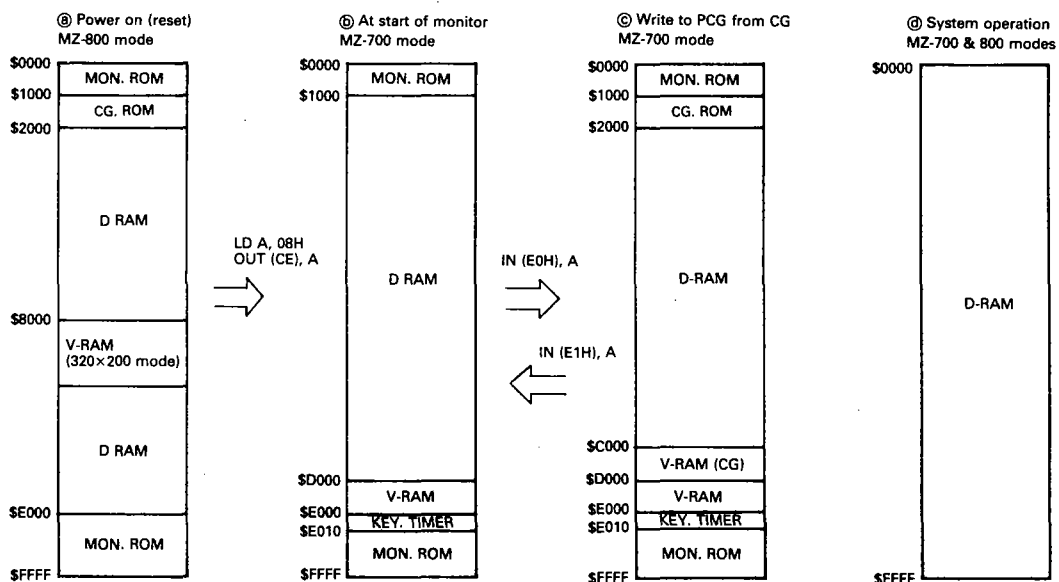
3 Memory map

The MZ-800 has a different memory map depending on

the mode. To have compatibility with the MZ-700, it has two modes of the MZ-700 mode and MZ-800 mode.



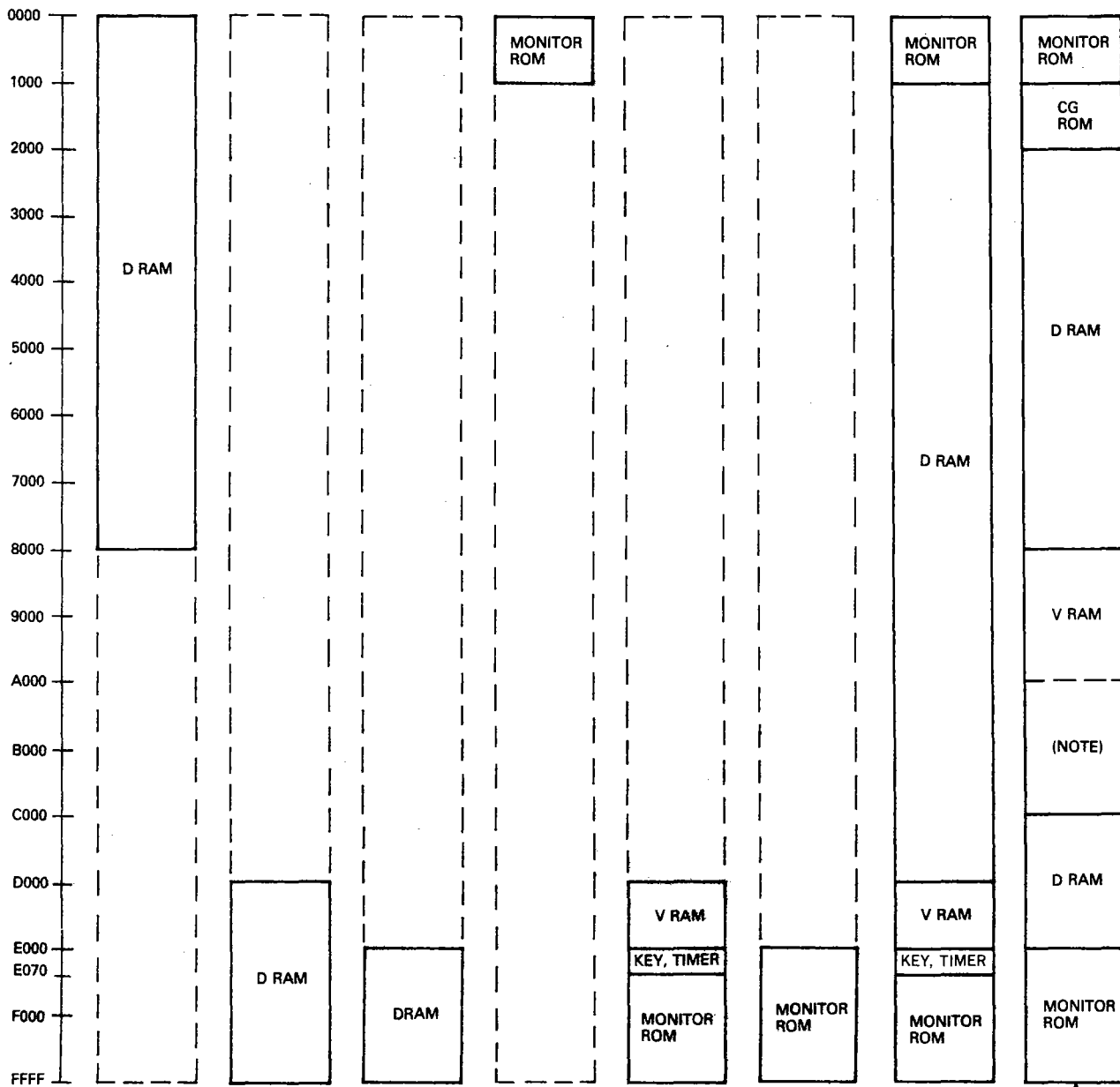
Memory map changes after initial program loading



- Memory map at power on is in the MZ-800 mode as in (a), but it changes to the MZ-700 mode by the monitor ROM when the monitor program starts. After transferring the CG data to the VRAM PCG area from the CG ROM at (c), the memory map then returns to (b).
- When the system program is completed to load, the memory map goes into the MZ-700 mode if the system switch (SW1) is set to ON side. If set to OFF side, it changes to the MZ-800 mode, then the memory map as in (d). During those changes, all memory spaces are composed of RAM and isolated from ROM and VRAM.
- Depression of the manual reset switch assumes memory map transition in order of (a) → (b) → (c) → (b), similar as in the case of power on.
- However, depression of the reset switch in conjunction with the CTRL key assumes the memory map of (d) after being changed once to the MZ-700 or MZ-800 mode depending on the state of the system switch. In the case of the MZ-800 mode, it is set to the plane I, II (4-color mode) of the 320 x 200 mode.

Memory Bank Control

Output port	\$E0	\$E1		\$E2	\$E3		\$E4	
MODE	—	MZ-700 mode	MZ-800 mode	—	MZ-700 mode	MZ-800 mode	MZ-700 mode	MZ-800 mode
Function	◦ \$0000 ~ \$7FFF to DRAM.	◦ \$D000 ~ \$FFFF to DRAM.	◦ \$E000 ~ \$FFFF to DRAM.	◦ \$0000 ~ \$0FFF to monitor ROM.	◦ \$D000 ~ \$FFFF to VRAM, key timer, and monitor ROM.	◦ \$E000 ~ \$FFFF to monitor ROM.	◦ \$0000 ~ \$0FFF to monitor ROM. ◦ \$1000 ~ \$CFFF to DRAM. ◦ \$D000 ~ \$FFFF to VRAM, key timer, and monitor ROM.	◦ \$0000 ~ \$0FFF to monitor ROM. ◦ \$1000 ~ \$1FFF to CG ROM. ◦ \$2000 ~ \$7FFF and \$C000 ~ \$DFFF to DRAM. ◦ \$8000 ~ \$BFFF to VRAM (NOTE). ◦ \$E000 ~ \$FFFF to monitor ROM.

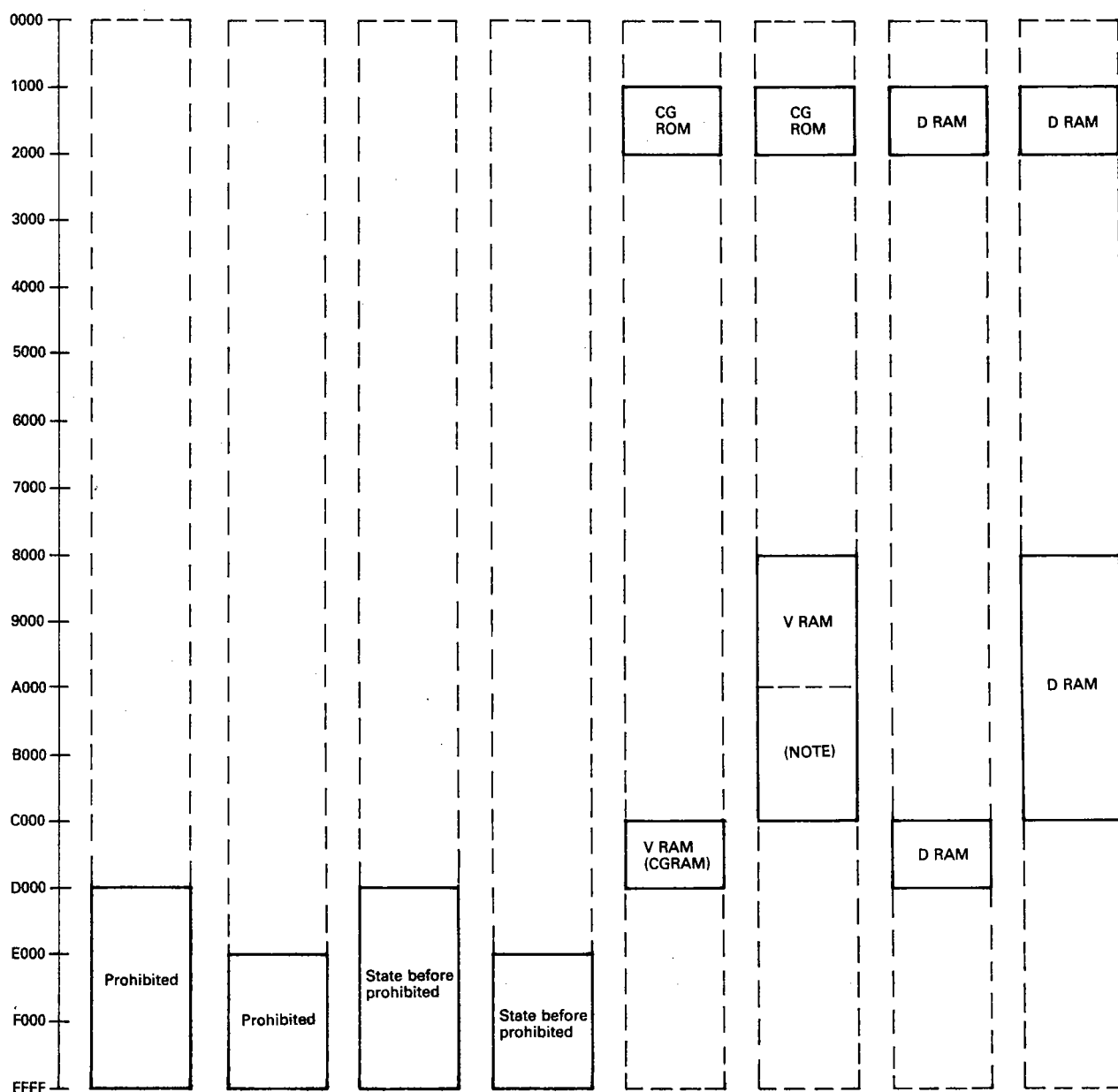


Area within dotted line does not involve change.

(NOTE): In the case of 320 × 200 mode, contents of \$8000 ~ \$9FFF are transferred, instead, and those after \$A000 are transferred to DRAM.

Power on or RESET input

I/O port	OUT (\$E5)		OUT (\$E6)		IN (\$E0)		IN (\$E1)	
MODE	MZ-700 mode	MZ-800 mode	MZ-700 mode	MZ-800 mode	MZ-700 mode	MZ-800 mode	MZ-700 mode	MZ-800 mode
Function	◦ \$D000 ~ \$7FFF prohibited.	◦ \$E000 ~ \$FFFF prohibited.	◦ \$D000 ~ \$FFFF returned to the state before prohibited.	◦ \$E000 ~ \$FFFF returned to the state before prohibited.	◦ \$1000 ~ \$1FFF to CG ROM. ◦ \$C000 ~ \$CFFF to VRAM (PCG RAM).	◦ \$1000 ~ \$1FFF to CG ROM. ◦ \$8000 ~ \$BFFF to VRAM (NOTE).	◦ \$1000 ~ \$1FFF returned to the state before CG was set. ◦ \$C000 ~ \$CFFF to DRAM.	◦ \$1000 ~ \$1FFF returned to the state before CG was set. ◦ \$8000 ~ \$BFFF to DRAM.



HARDWARE

4 Custom LSI

The custom LSI is a 100-pin single chip LSI on which the MZ-800 memory controller (I/O controller) and CRT controller, etc. are contained.

4-1 Memory controller

Used for the control of the memory bank. Addressing of DRAM, ROM, and VRAM is conducted by selection I/O address, \$E0 ~ \$E6, using OUT or IN command.

4-2 I/O controller

In this I/O controller is created the select signal for assignment of MZ-800 internal device.

See Table-2 for relation of internal device vs I/O address.

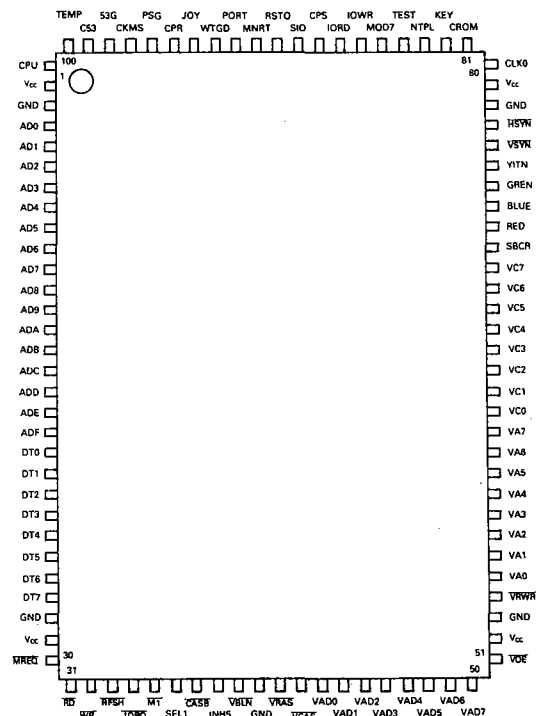
I/O address	Signal name	Device (I/O)	Function
FF FE FD FC	CPR	Z80A PIO (I/O)	Port B, printer data output Port A, printer control and timer interrupt Port B control (Mode 0) Port A control (Mode 3)
F2	PSG	PSG (O)	PSG output port
F1 F0	JOY	JOYSTICK (I)	Joystick-2 input port Joystick-1 input port
F0	—	(O)	Pallet write
E6 E0	—	(I/O)	Memory bank control
D7 D6 D5 D4	C53	8253 (I/O)	Control port output Counter-2 Counter-1 Counter-0 (NOTE): Mapped to E007 ~ E004 in the MZ-700 mode.
D3 D2 D1 D0	KEY	8255 (I/O)	Control Port C, cassette, etc. Port B, key input Port A, key strobe output (NOTE): Mapped to E003 ~ E000 in the MZ-700 mode.
CF CE CD CC	—	O I/O O O	CRTC register
\$E008	—	I/O	TEMP, HBLK input; and 8253 G0 ON/OFF output for the MZ-700 mode only.

* When above I/O address is accessed, it makes IOWR active for OUT or IORD for IN command.

Pin No.	Signal name	I/O	Functional description	Note
1	CPU	O	CPU clock (3.547 MHz)	
2	5 V	—	Power supply	
3	GND	—	Ground	
4	AD0	I	CPU address bus	
19	ADF			
20	DT0	I/O	CPU data bus	
27	DT7			
28	GND			
29	VCC	—	Power supply	
30	MREQ	I	CPU MREQ signal	Negative logic
31	RD	I	CPU RD signal	Negative logic
32	WR	I	CPU WR signal	Negative logic
33	RFSH	I	CPU RFSH signal	Negative logic
34	TORQ	I	CPU TORQ signal	Negative logic
35	M1	I	CPU M1 signal	Negative logic
36	SEL1	O	System RAM address multiplexer select signal	
37	CASB	O	System RAM column address strobe signal	
38	INH5	O	Inhibit bank (OUT \$E5) select signal ("H" = Inhibit).	OPEN
39	VBLN	O	Vertical blanking signal	Negative logic
40	GND	—		
41	VRAS	O	VRAM RAS control signal	Negative logic
42	VCAS	O	VRAM CAS control signal	Negative logic
43	VAD0	O	VRAM address signal (multiplexer output)	
50	VAD7			
51	VOE			
52	VCC	—	Power supply	
53	GND	—	Ground	
54	VRWR	O	VRAM write signal	Negative logic
55	VA0	I/O	VRAM data bus (standard RAM)	
62	VA7			
63	VC0	I/O	VRAM data bus (option RAM)	
70	VC7			
71	SBCR	O	Color sub-carrier wave	
72	RED	O	Video signal, red	
73	BLUE	O	Video signal, blue	
74	GREEN	O	Video signal, green	
75	YITN	O	Brightness control signal	
76	VSYN	O	Vertical sync signal	Negative logic
77	HSYN	O	Horizontal sync signal	Negative logic
78	GND	—		
79	VCC	—		
80	CLK0	I	Clock input (17.7344 MHz)	
81	CROM	O	ROM chip enable	Negative logic
82	KEY	O	8255 chip enable	Negative logic
83	NTPL	I	NTSC/PAL selection (PAL = "L")	GND
84	TEST	I	Test pin ("H" = test mode)	GND
85	MOD7	I	MZ-700/800 mode selection ("L" = MZ-700 mode)	
86	TOWR	O	Sum of CS and WR of I/O controlled by the custom IC	Negative logic
87	TORD	O	Sum of CS and RD of I/O controlled by the custom IC	Negative logic
88	CRS	O	I/O \$B0 ~ \$B3 chip enable	OPEN
89	SIO	O	I/O \$F4 ~ \$F7 chip enable	OPEN
90	RSTO	O	Reset output	Negative logic
91	MNRT	I	Manual reset input	Negative logic
92	PORT	I	Power on reset input	Negative logic
93	WTGD	O	Wait signal to CPU	Open drain
94	JOY	O	Joystick chip enable	Negative logic
95	CPR	O	PIO chip select	Negative logic
96	PSG	O	76489 chip select	Negative logic
97	CKMS	O	8253 musical interval clock	
98	53G	O	8253 musical interval ON/OFF gate signal	
99	C53	O	8253 chip enable	Negative logic
100	TEMP	I	MZ-700 mode, \$E800 tempo input	

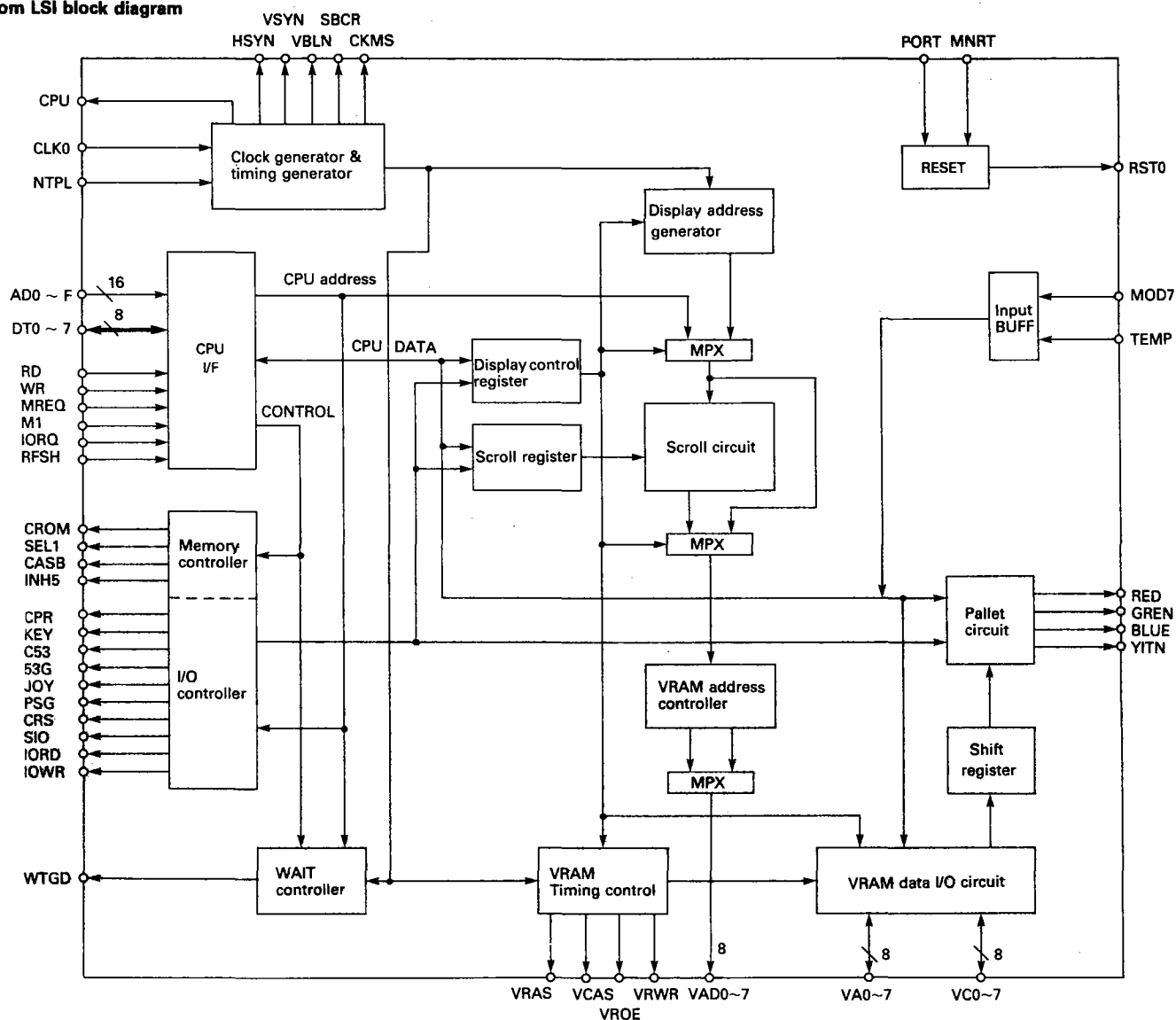
* Term "OPEN" represents the signal not used on the board.

Pin configuration



V_{cc} 2, 29, 52, 79, pin
GND 3, 28, 40, 53, 78 pin

Custom LSI block diagram



Oscillation from the crystal oscillator is divided to create the CPU clock, horizontal sync, vertical sync, and display address control signals.

The diagram illustrates the timing and control logic for the 68000 microprocessor. It shows the internal clock distribution and control signal generation. Key components include:

- CLK Input:** The main clock input, which is divided by 1/4 to produce CKMS (1.10 MHz) and by 1/5 to produce the CPU clock (3.547 MHz).
- NTP1 Input:** The non-trapezoidal pulse input, which is divided by 1/4 and 1/5 to produce CK32 (1/2), DO1 (1/4), DO2 (1/8), and DO3 (1/16).
- Waveform generator (horizontal sync):** Produces HBLN and HSYN signals.
- Dividers:** Produce CK32 (1/2), DO1 (1/4), DO2 (1/8), and DO3 (1/16).
- Pulse generator:** Produces LD64 (840 x 200, shift register load) and LOAD (700 mode, 320 x 200, shift register load) signals.
- Display mode register DM02:** Divided by 1/12 to produce VBLN and VSYN signals.
- Waveform generator (vertical sync):** Produces VBLN and VSYN signals.
- Waveform generator (display address control):** Produces the display address generator signal.

The block diagram illustrates the display control system architecture. Key components and their connections include:

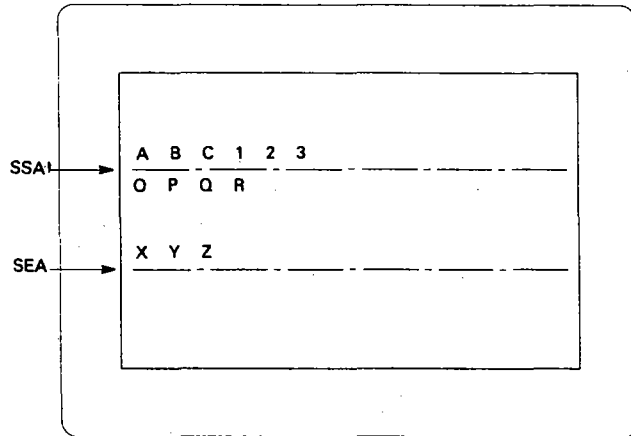
- Display address generator:** Receives **CLK0** and **DMD3** (Display mode register) as inputs. Its output goes to the **Address latch** and the **Horizontal line counter**.
- Address latch:** Receives input from the **Display address generator** and outputs to the **MPX** (Multiplexer).
- Horizontal line counter:** Receives input from the **Display address generator** and outputs to the **MPX** and the **Scroll control circuit**.
- LC0* - LC2:** A logic gate (AND) that receives inputs from the **Horizontal line counter** and the **Address latch**.
- CPU address:** Input to the **Address latch**.
- (Display mode register):** Includes **DMD2** and **DISP***, which provide inputs to the **MPX**.
- CPU data (DT0-7):** Input to the **Scroll data latch (Scroll register)**.
- Scroll data latch (Scroll register):** Receives **DT0-7** and outputs to the **Scroll control circuit**.
- Scroll control circuit:** Receives inputs from the **Horizontal line counter**, **Scroll data latch**, and **VRAM address MPX**. It outputs **SW**, **SOF**, **SSA**, and **SEA** signals.
- VRAM data latch:** Receives **VA0 - 7** (VRAM data) and outputs to the **Address MPX**.
- Address MPX:** Receives inputs from the **Address latch**, **MPX**, and **VRAM data latch**. Its output goes to the **Scroll control circuit** and the **VRAM address MPX**.
- VRAM address MPX:** Receives inputs from the **Address MPX** and the **Scroll control circuit**. Its output goes to the **VRAM address** block.
- VRAM address:** Receives input from the **VRAM address MPX** and outputs **VA00 - 7** (VRAM address).
- DMD3:** An input signal at the bottom of the diagram.

4-5 Scroll

- 1) Scrolling is possible for both horizontal and vertical directions by means of software offset.

The following four registers are used for scroll control.

- a. Scroll start address register: SSA (7-bit)
- b. Scroll end address register: SEA (7-bit)
- c. Scroll width register: SW = SEA-SSA (7-bit)
- d. Scroll offset register: SOF (10-bit)



- 2) Control of scroll starts by the initialization of the scroll control register.

SSA = \$0
SEA = \$7D
SW = \$7D
SOF = \$0

- 3) Way of smooth scrolling

SOF = \$0 → \$5

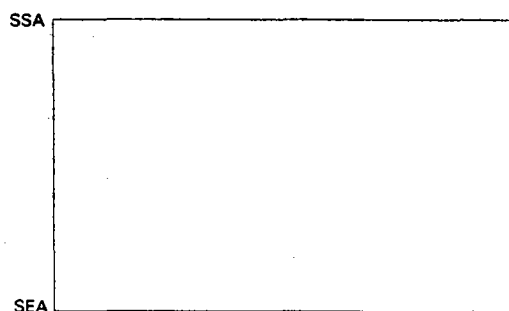
Programming "SOF = \$5" makes the display screen shifted one line up.

The highest line (address: \$0 ~ \$27) is then assigned to the lowest line (\$1F18 ~ \$1F3F).

As normal scroll involves updating of the data for the lowest line, the data of address \$1F18 ~ \$1F3F are updated.

SOF = \$5 → \$0

By reducing the value of SOF by "5", it makes the screen shifted one line down.



- 4) Line scroll

SOF = \$0 → \$28

Programming "SOF = \$28" makes the display screen shifted eight lines up. Data on the highest line therefore shifted to the bottom line.

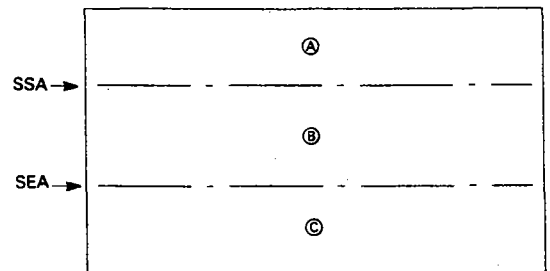
Programming "\$28 → \$0" makes the display screen shifted eight lines down, and the line on the bottom moves to the highest line.

- 5) Screen split

Appropriate deviation of SSA, SEA, and SW permits to divide the screen into three sections of ①, ② and ③.

Though the section ② is permitted to scroll, sections ① and ③ are not permitted to scroll.

See the figure to explain with.



Assume now that the top of the section ② is on the 5th line (40 raster) and the top of the section ③ is on the 18th line (144 raster). Attention must be paid to the fact that values SSA and SEA are used for assigning lines. Scroll registers are set with the following values.

SSA = \$19
SEA = \$5A
SW = \$41
SOF = \$0

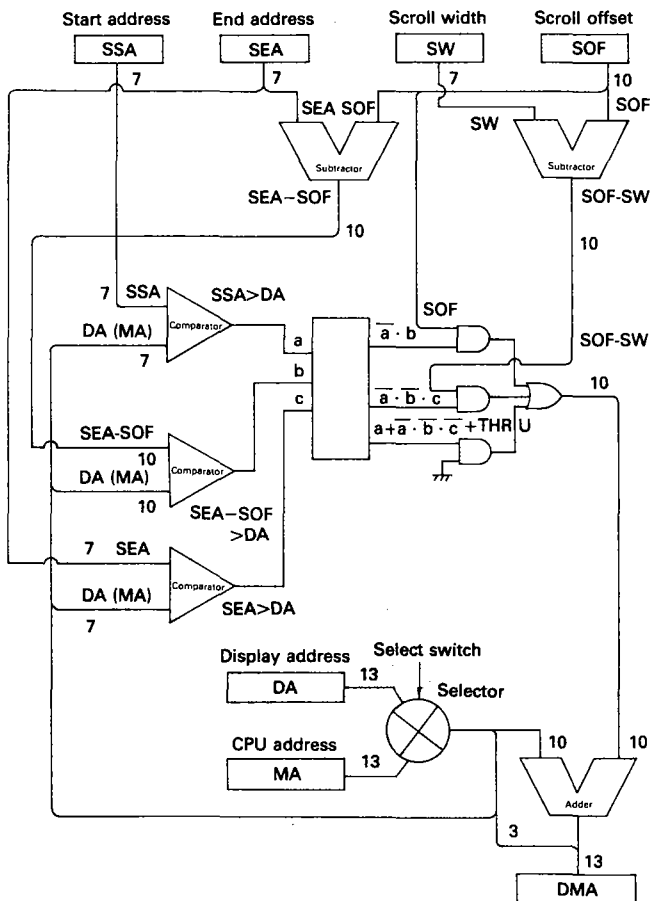
In this occasion, it needs to initialize the screen that has been displayed. "SOF = \$5" must be programmed to scroll ② one line. Then, only the section ② is shifted up, and the highest line of ② moves to the bottom line of ②. Programming "SOF = \$A" makes it scrolled one more line.

SOF ≤ SW

Scroll offset (SOF) should necessarily be within a range of the scroll width. Display is not assured with SOF set greater than SW.

Scroll and control circuit hardware

• Block diagram



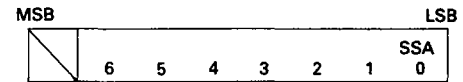
Scroll control register

SSA: Scroll start address

Increment of SSA: \$5

Minimum value of SSA: \$0

Maximum value of SSA: \$78

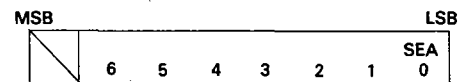


SEA: Scroll end address

Increment of SEA: \$5

Minimum value of SEA: \$5

Maximum value of SEA: \$7D



SW: Scroll width

Increment of SW: \$5

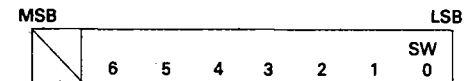
Minimum value of SW: \$5

Maximum value of SW: \$7D

Relation of SW, SEA, vs SSA

$$SW = SEA - SSA$$

$$SW > SSA$$

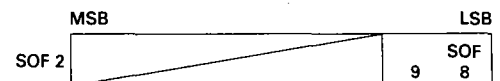
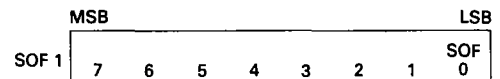


SOF: Scroll offset

Increment of SOF: \$5

Minimum value of SOF: \$0 (without offset)

Maximum value of SOF: \$3E8



Relation of display address, SEA, SSA, vs SOF

Display address	m	l	k	j	i	h	g	f	e	d	c	b	a
SSA	SSA	SSA	SSA	SSA	SSA	SSA	SSA	SSA	SSA	SSA	SSA	SSA	SSA
SEA	SEA	SEA	SEA	SEA	SEA	SEA	SEA	SEA	SEA	SEA	SEA	SEA	SEA
SOF	SOF	SOF	SOF	SOF	SOF	SOF	SOF	SOF	SOF	SOF	SOF	SOF	SOF
Screen left end address													
0 Line	0	0	0	0	0	0	0	0	0	0	0	0	0
1 Line	0	0	0	0	0	0	0	1	0	1	0	0	0
2 Line	0	0	0	0	0	0	1	0	1	0	0	0	0
3 Line	0	0	0	0	0	0	1	1	1	1	0	0	0
8 Line	0	0	0	0	1	0	1	0	0	0	0	0	0
16 Line	0	0	0	1	0	1	0	0	0	0	0	0	0
24 Line	0	0	0	1	1	1	1	0	0	0	0	0	0
192 Line	1	1	1	1	0	0	0	0	0	0	0	0	0
199 Line	1	1	1	1	1	0	0	0	1	1	0	0	0

Relation of SW vs SOF

$$SW > SOF$$

Concept of the scroll control circuit

Scroll method

- Scrolling by means of VRAM address conversion.

Range of scroll

- y-axis programmable.
BASIC console command compatible
- x-axis fixed

Scroll sequence

- The scroll start address is termed "SSA" and end address "SEA".
- Execution of scroll, with offset given from the CPU.
- One line (line S) starting from SSA disappears from the display screen.
- A new line (line S') is added to SEA. Line S' is the same refresh memory as the line S. The contents of the memory was erased (nullified by the CPU) before the execution.

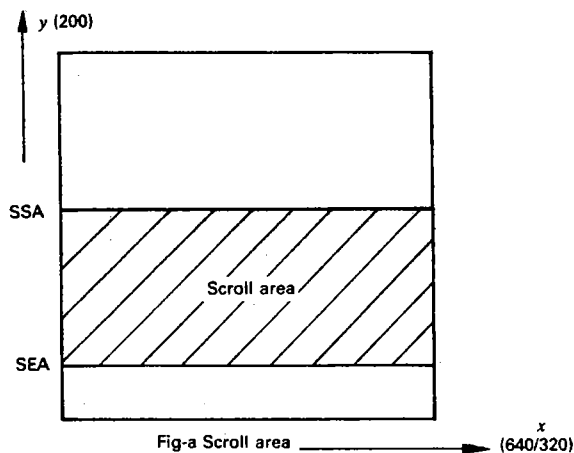


Fig-a Scroll area

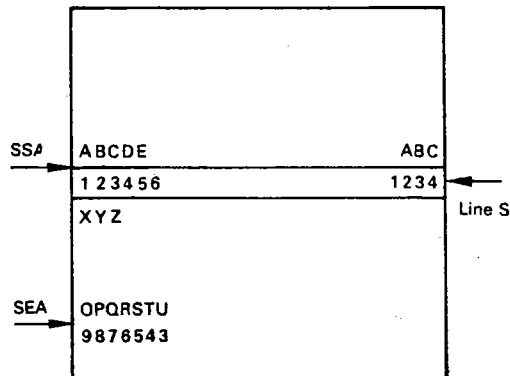


Fig-b Screen before scroll

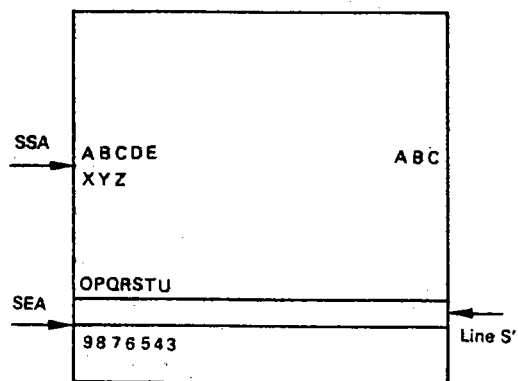


Fig-c Line after scroll

Execution of scrolling by address conversion

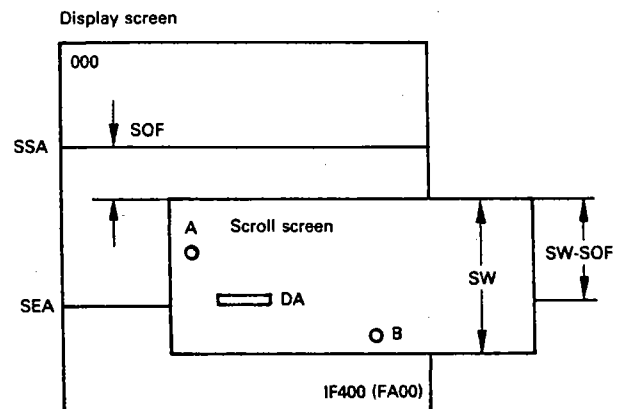
- Scroll offset (SOF) is the count of lines which the CPU gives to the CRTC. For instance, the following must be observed to perform scrolling.

$$3\text{-line scroll: } \text{SOF}_3 = 0F \times 3$$

$$5\text{-line scroll: } \text{SOF}_5 = 0F \times 5$$

And, to scroll one more line after 5-line scroll;

$$5\text{-line scroll: } \text{SOF}_5' = \text{SOF}_5 + 0F = 0F \times 6$$



- Display address DA is the signal created in the CRTC display address generation circuit and arranged in their order from the upper left corner of the screen. The bottom right address is 1F400 in the 640 × 200 mode.
- Display memory address DMA represents the VRAM address corresponding to DA. Since scroll is executed by means of address conversion, the order of DMA may not be the same as DA, necessarily.
- CPU address MA is the VRAM address that obtained from the CPU through the CRTC. To lighten burden on the CPU, a circuit is added to make order of DA identical to order of MA arrangement.

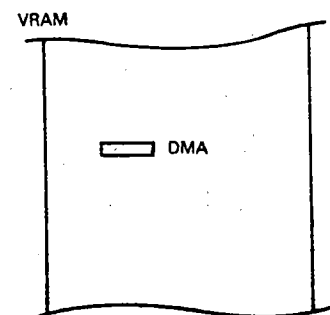


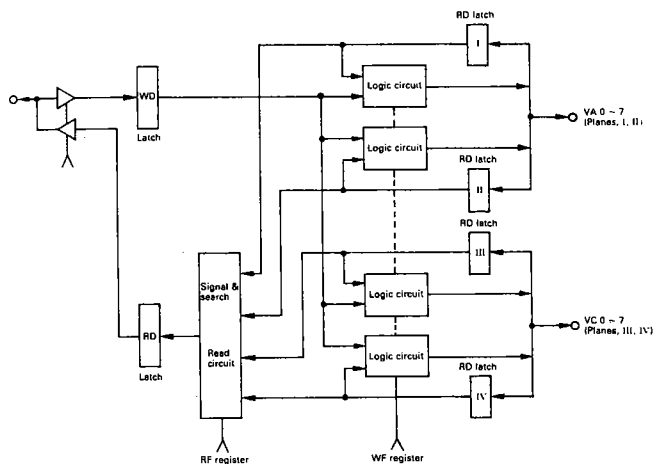
Fig-d Address conversion

4-6 VRAM data input/output circuit

1. Nothing intervenes for input and output of data in the case of the MZ-700 mode.
2. MZ-800 mode
 - Write
Read data (RD) from the VRAM and write data (WD) from the CPU are subjected to logical operation according to the direction from the write format register (WF) and its result is written.

- Read

For plane read data from the VRAM, data to be read by the CPU are arranged in accordance with the direction of the read format register (RF).



* Logic circuit

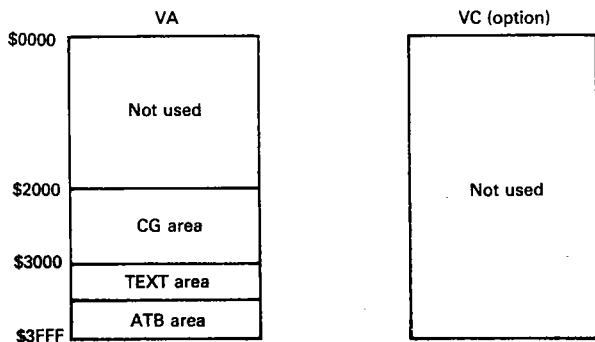
Read data from the VRAM and write data from the CPU are subjected to logical operation (OR, XOR, RESET, etc.) and its result is used for the write data.

VRAM access timing

1) MZ-700 mode

See separate page for display timing chart.

The VRAM is configured in the following manner in this instance.



As the PCG method is adopted for the MZ-700 mode, the text and ATB areas are actually mapped to \$D000 ~ \$DFFF. So, the VRAM address has the following relation with the display character position.

	1	2	3	...	40
1	D000	D001	D002	...	D027
2	D028			...	
...				...	
25	D3C0			...	D3E7

2) MZ-800 mode

As the bit map method is used for the MZ-800 mode, it is possible to four screens of 320×200 dots and two screens (maximum) of 640×200 dots.

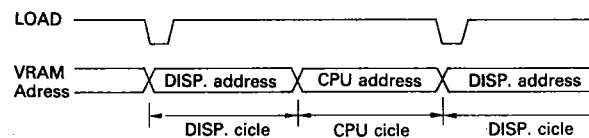
The cycle steal method is used for this mode.

i) 320×200 dots

See separate page for the timing chart during display and CPU read timing.

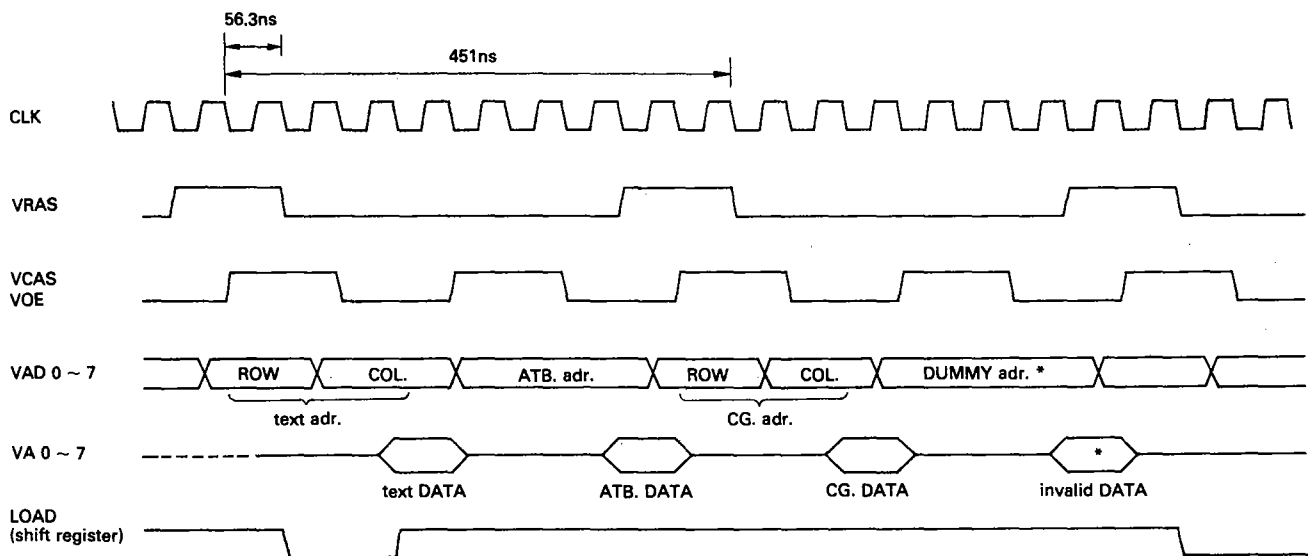
What is pseudo cycle steal

With the MZ-800, the pseudo cycle steal method is adopted for VRAM accessing.

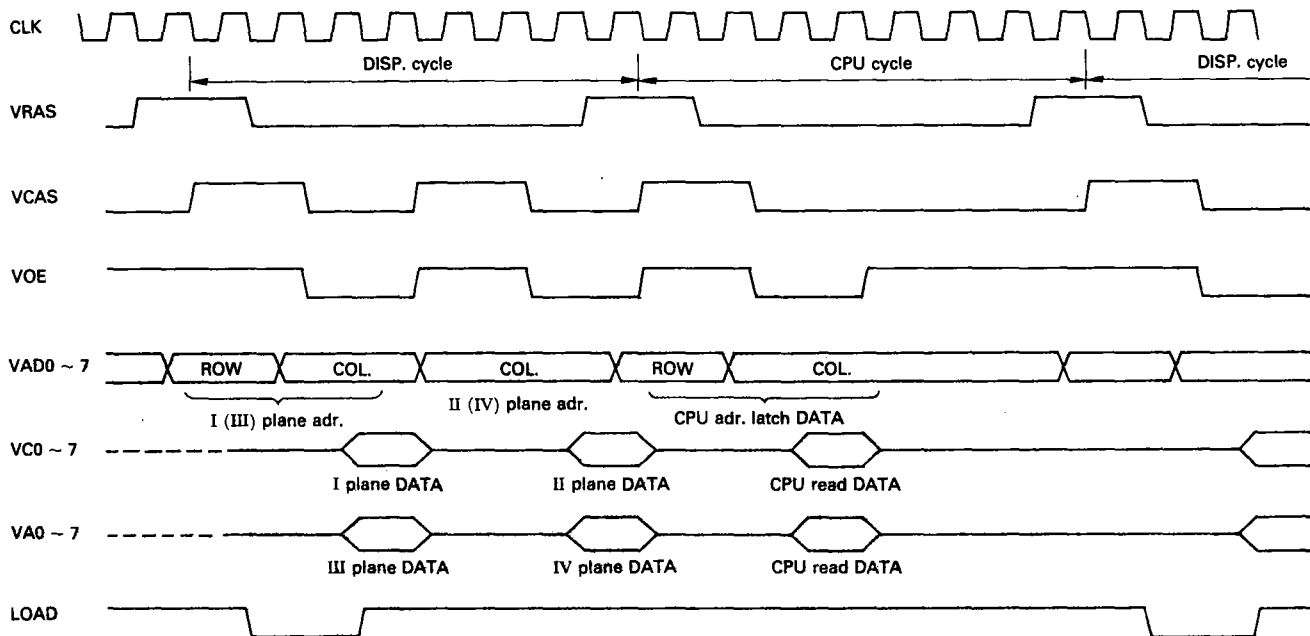


As shown in the figure, a next display data fetch and CPU accessing are multiplexed during a display period. Because accessing of the VRAM while characters are on display causes the screen to blink with the MZ-700 mode, it awaits for blinking to complete before accessing of the VRAM. But, with the cycle steal method it enhances faster screen processing as it enables to access the VRAM during a display period. Because it is not a complete cycle steal with the MZ-800 but timing is taken using a wait in order to synchronize with the CPU cycle for accessing from the CPU, it is therefore called "pseudo cycle steal".

MZ-700 MODE DISPLAY TIMING

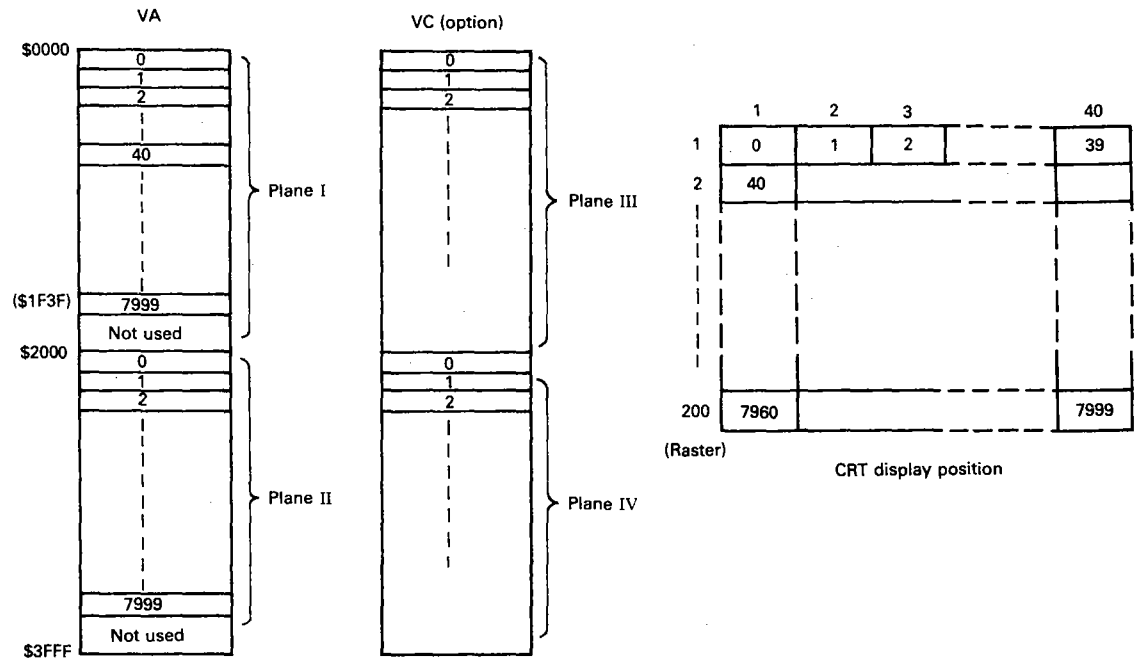


MZ-800 MODE (320 × 200 dot)



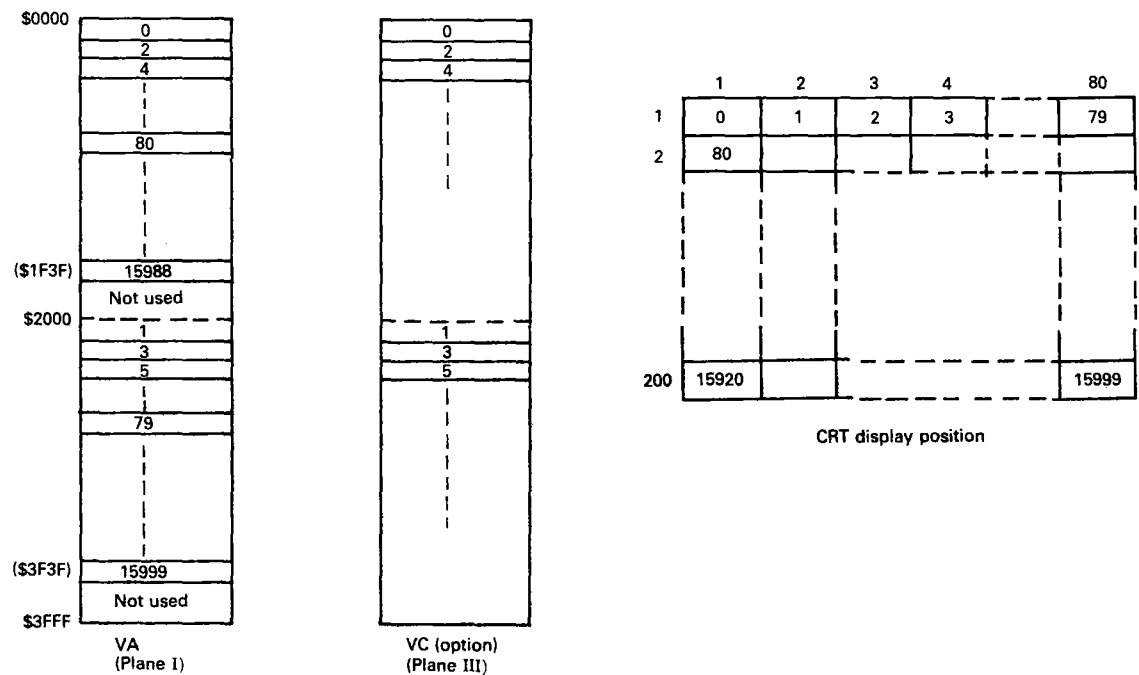
1) 320 × 200 dots

See the figure below for VRAM configuration and CRT character display position.

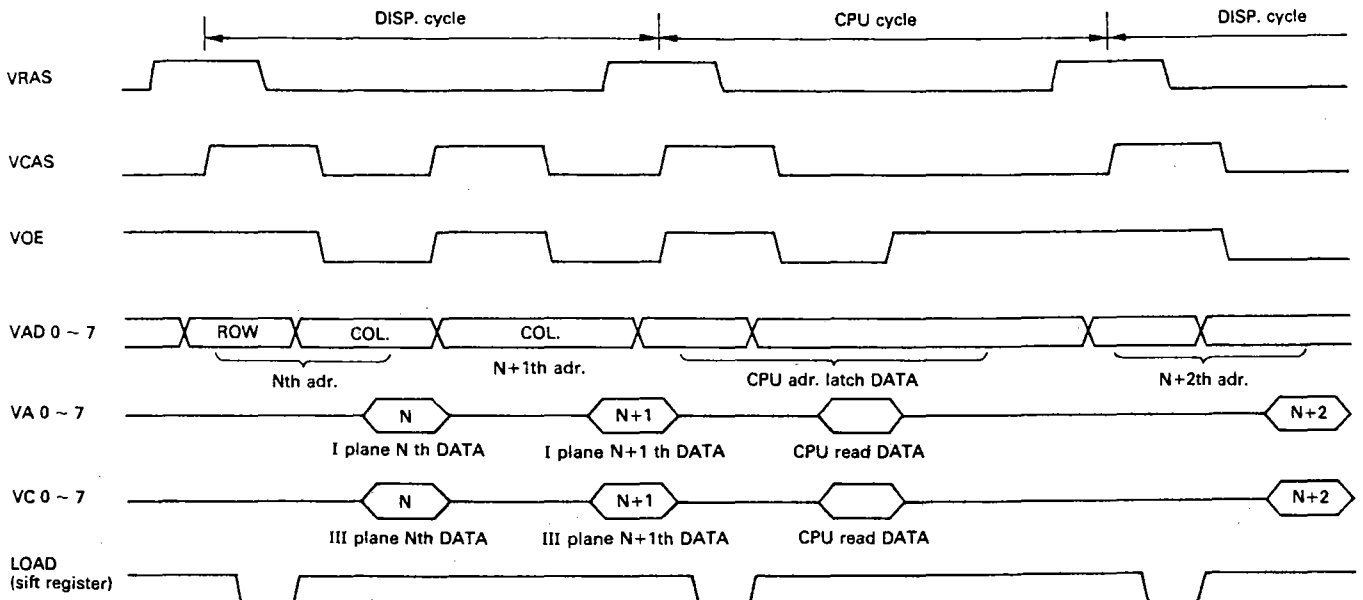


2) 640 × 200 dots

Because it operates in the cycle steal mode, two bytes of display data are fetched during one byte display cycle. (See the chart in separate page.)
See the figure below for VRAM configuration and CRT character display position.



800 MODE (640 × 200 dot)



CPU and VRAM accessing

1. Accessing of the VRAM by the CPU is carried out in the cycle steal mode (MZ-800 mode only) during the flyback period of the display under the control of the CRT controller.
2. Even when there is no accessing from the CPU in the CPU cycle, such as VRAS, VCAS, VOE, etc. are outputted in the timing of the read cycle at all times.
3. Write to the VRAM is carried out after logical operation of the read and write data by means of the read-modify-write method. But, in the case of the 320 × 200, 16-color mode, data are written in two CPU cycles as there is a need of writing to Plane IV. See separate paper for timing chart.

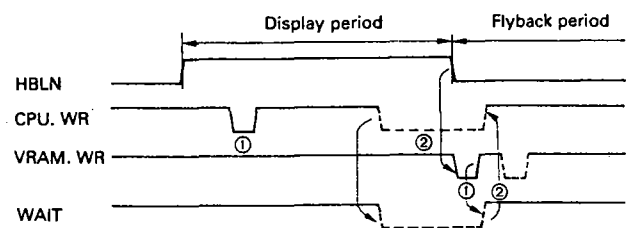
4. CPU wait

1) Write

- As there is a one-byte buffer in the CRT controller, write to the VRAM from the CPU is carried out through the buffer. But, actual write to the VRAM is

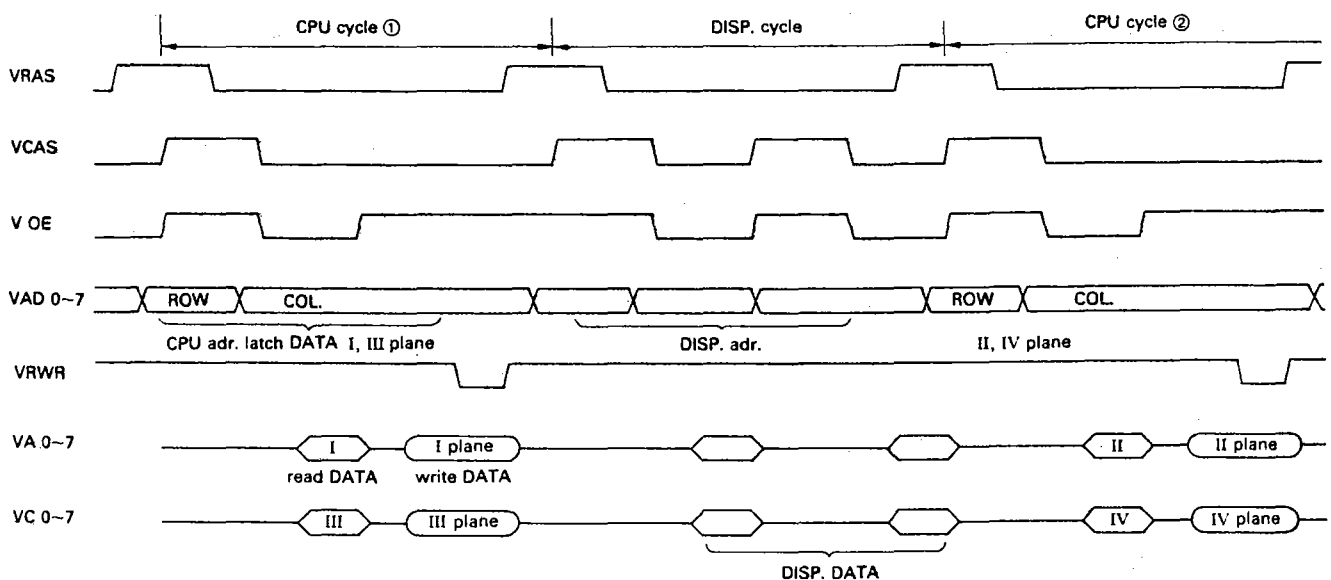
done by the CRT controller. Therefore, there would be no need of wait under almost any condition in the MZ-800 mode.

- Even in the MZ-700 mode, wait is issued when there are more than two writes in a display period.



2) Read

Wait is issued along with the CPU write action both during displaying and flyback periods to perform reading operation in synchronization with the CPU cycle.



4-7 Register functions

VRAM configuration

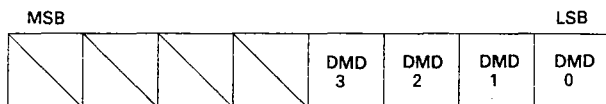
- One or two chips of 16 KB VRAM are used.
- In the case of a single 16 KB VRAM chip, it handles 320 × 200 dots, 4 colors, or 640 × 200 dots 1 color.
- In the case of two 16 KB VRAM chips, it handles 320 × 200 dots, 16 colors, 640 × 200 dots, 4 colors, 320 × 200 dots, 4 colors, 2 frames, or 640 × 200 dots, 1 color, two frames.

* Discussed next are about functions of the custom LSI. There may be some restrictions because the standard version of the MZ-800 incorporates only one 16 KB RAM.

Display mode register (OUT &HCE)

- It consists of four bits which are used to represent display method, resolution, and display screen (color plane) in combined way.

Display mode register (DMD)



• DMD 3, 2: Display method and resolution

DMD 3	2	
0	0	Bit map, 320 × 200
0	1	Bit map, 640 × 200
1	0	MZ-700 mode
1	1	Prohibited

• DMD 1, 0: Display screen designation

DMD 1	DMD 0	320 × 200	640 × 200	MZ-700
0	0	Frame A, Planes I and II	Frame A, Plane I	Normal
0	1	Frame B, Planes III and IV	Frame B, Plane III (NOTE)	Prohibited
1	0	Planes I, II, III, and IV	Planes I, III	Prohibited
1	1	Prohibited		

NOTE: 640 × 200, Plane B is Plane III, not Plane II.

* With the MZ-800, DMD 1 = 0, DMD 0 = 0.

△ 3/3

Table-1 VRAM configuration and display mode

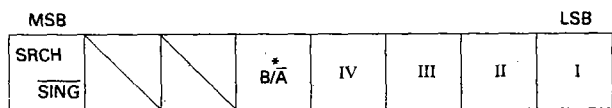
VRAM capacity	VRAM configuration	Resolution	Display color	Display frame	Color combination (NOTE)	DMD			
						3	2	1	0
16 KB		320 × 200	4 colors	Frame A	I, II	0	0	0	0
		640 × 200	1 color	Frame A	I	0	1	0	0
32 KB		320 × 200	4 colors	Frame A	I, II	0	0	0	0
				Frame B	III, IV	0	0	0	1
			16 colors	Frame A	I, II, III, IV	0	0	1	0
		640 × 200	1 color	Frame A	I	0	1	0	0
				Frame B	III	0	1	0	1
			4 colors	Frame A	I, III	0	1	1	0
	MZ-700	40 characters × 25 lines	8 colors	Frame A	RGB	1	0	0	0

(NOTE) Except for the MZ-700 mode, actual display colors are produced by the pallet.

VRAM to CPU interface

- As the CRTC bus is completely separated from the CPU bus, read and write of the VRAM is carried out through the CRTC. Therefore, interfacing with the CPU is done via the read register or write register in the CRTC.
- VRAM access by the CRTC is done under the pseudo cycle steal mode.
- Not only read and write are for the accessing with the CPU, it permits to read multiple number of screen data logical operational results and to write the read-modify-write of the logical operational results for the data already written. So, it has two registers of the read format register and the write format register.
- It permits CPU access to the non-display plane in the display mode according to the $B/\bar{A}$ bit and it enables selection of data buffer and two screens, when the 32 KB VRAM is used.

a) Read format register (RF) (OUT & CD)



* NOTE: Same as the bit $B/\bar{A}$ of the write format register.

• SRCH/ $\bar{SING}$

"0": Single color data read

Reads the data of the color plane, I, II, III, or IV, specified by "1".

NOTE: Only one item should be "1" out of I, II, III, and IV. If it is "1" for more than two or non-existence of the VRAM may not assure the data read.

"1": Specified color search

"1" is returned for the bit of the color specified by 0/1 of I, II, III, and IV.

NOTE: Depending on the display more, color combination is permitted for the bit combination of I, II, III, IV; III, IV; I, II; I; and III. Bit combination otherwise will be disregarded.

(ex. For the 640 × 200, 4-color mode, combination becomes possible for I and III, and II and IV are disregarded.

• $B/\bar{A}$

CPU access plane change

MZ-800 → "0": Frame A access

Accesses the frame A (planes I and II for the 320 × 200, 4-color mode; plane I for the 640 × 200, 1-color mode).

"1": Frame B access

Accesses the (planes III and IV for the 320 × 200, 4-color mode; plane II for the 640 × 200, 1-color mode).

• I, II, III, IV Color plane designation.

Table-2 Display mode vs read format register

	Display mode	SRCH/SING	B/A	IV	III	II	I	Function (NOTE)
Single color data read	320 × 200, 4/16 colors 640 × 200, 1/4 colors	0	(*)	0	0	0	1	Plane I data read
			Frame A: "0"	0	0	1	0	Plane II data read
			Frame B: "1"	0	1	0	0	Plane III data read
				1	0	0	0	Plane IV data read
Specified color search	320 × 200, 4 colors	1	0	×	×	0	0	$\overline{\text{I}}, \overline{\text{II}}$ dot search
				×	×	0	1	$\text{I}, \overline{\text{II}}$ dot search
				×	×	1	0	$\overline{\text{I}}, \text{II}$ dot search
				×	×	1	1	I, II dot search
			1	0	0	×	×	$\overline{\text{III}}, \overline{\text{IV}}$ dot search
				0	1	×	×	$\text{III}, \overline{\text{IV}}$ dot search
				1	0	×	×	$\overline{\text{III}}, \text{IV}$ dot search
				1	1	×	×	III, IV dot search
	320 × 200, 16 colors	1	×	0	0	0	0	$\overline{\text{I}}, \overline{\text{II}}, \overline{\text{III}}, \overline{\text{IV}}$, dot search
				0	0	0	1	$\text{I}, \overline{\text{II}}, \overline{\text{III}}, \overline{\text{IV}}$, dot search
				0	0	1	0	$\overline{\text{I}}, \text{II}, \overline{\text{III}}, \overline{\text{IV}}$, dot search
				0	0	1	1	$\text{I}, \text{II}, \overline{\text{III}}, \overline{\text{IV}}$, dot search
				0	1	0	0	$\overline{\text{I}}, \overline{\text{II}}, \text{III}, \overline{\text{IV}}$, dot search
				⋮	⋮	⋮	⋮	⋮
				1	1	1	1	$\text{I}, \text{II}, \text{III}, \text{IV}$, dot search
	640 × 200, 1 color	1	0	×	×	×	0	$\overline{\text{I}}$, dot search
				×	×	×	1	I , dot search
			1	×	0	×	×	$\overline{\text{III}}$, dot search
				×	1	×	×	III , dot search
	640 × 200, 4 colors	1	×	×	0	×	0	$\overline{\text{I}}, \overline{\text{III}}$, dot search
				×	0	×	1	$\text{I}, \overline{\text{III}}$, dot search
				×	1	×	0	$\overline{\text{I}}, \text{III}$, dot search
				×	1	×	1	I, III , dot search
	MZ-700	0	0	0	0	0	1	Data, ATB, CG area read

(*): Refer to the display frame of Table-1.

NOTES:

- Read for the non-existing VRAM are not assured.
- The above parameter has to be set up for the MZ-700 mode.
- ★ B/A must be set to "0" for the standard MZ-800 (without MZ1R25).

b) Write format register (WR) (OUT & CC)

MSB								LSB
WMD 2	WMD 1	WMD 0	(NOTE) B/A	IV	III	II	I	

NOTE: Same as the bit B/A of the read format register.

- I, II, III, IV
Color plane designation

• WMD 0 ~ 2

Selects the logical operational mode for read-modify-write.

• B/A (NOTE)

Standard MZ-800 → "0": Frame A access

Frame A is accessed for the display mode.

"1": Frame B access

Frame B is accessed for the display mode.

Write mode	WMD			B/A	Color plane				Display mode	Function	WD: Write data VD: VRAM data
	2	1	0		IV	III	II	I			
SINGLE WRITE	0	0	0	(*)	0/1	0/1	0/1	0/1	320 × 200, 4/16 colors 640 × 200, 1/4 colors	Color plane of "1": WD, write Color plane of "0": Fixed	
EXOR	0	0	1	Frame A: 0	0/1	0/1	0/1	0/1		Color plane of "1": $WD \oplus VD$ Color plane of "0": Fixed	
OR	0	1	0	Frame B: 1	0/1	0/1	0/1	0/1		Color plane of "1": $WD + VD$ Color plane of "0": Fixed	
RESET	0	1	1		0/1	0/1	0/1	0/1		Color plane of "1": $WD \cdot VD$ Color plane of "0": Fixed	
REPLACE	1	0	x	0	x	x	0/1	0/1	320 × 200 (A), 4 colors (B)	Writes WD in a specific color (Character write to the graphic plane)	
				1	0/1	0/1	x	x			
				x	0/1	0/1	0/1	0/1	320 × 200, 16 colors	Color plane of "1": WD	
				0	x	x	x	0/1	640 × 200 (A), 1 color (B)	Color plane of "0": Writes "0". Color plane of "X": Fixed	
				1	x	0/1	x	x			
				x	x	0/1	x	0/1	640 × 200, 4 colors		
PSET	1	1	x	0	x	x	0/1	0/1	320 × 200 (A), 4 colors (B)	Writes only bit "1" of WD in a specific color. (Character write to graphic plane)	
				1	0/1	0/1	x	x			
				x	0/1	0/1	0/1	0/1	320 × 200, 16 colors	Color plane of "1": $WD + VD$	
				0	x	x	x	0/1	640 × 200 (A), 1 color, (B)	Color plane of "0": $WD \cdot VD$ Color plane of "X": Fixed	
				1	x	0/1	x	x			
				x	x	0/1	x	0/1	640 × 200, 4 colors		
MZ-700	0	0	0	0	0	0	0	1	MZ-700	Writes WD into the DATA, ATB, and CG area.	

(*) Refer to Table-1 display frame

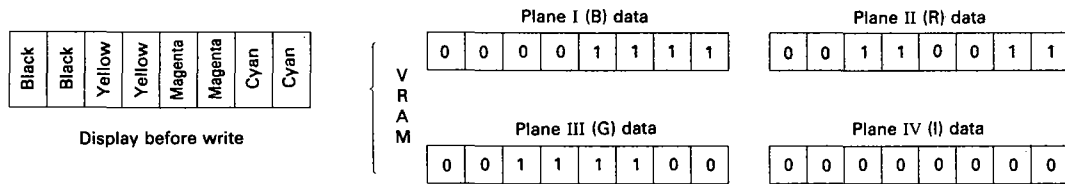
NOTES:

- Write for the non-existing VRAM are not assured.
- The above parameter has to be set up for the MZ-700 mode.
- B/A must be set to "0" for the standard version MZ-800.

c) Example of CPU read/write access

- Shown next are access examples of REPLACE write, PSET write, and SEARCH read in the 320×200 , 16-color mode.

As for display colors, Plane I corresponds to B, II to R, III to G, and IV to I.

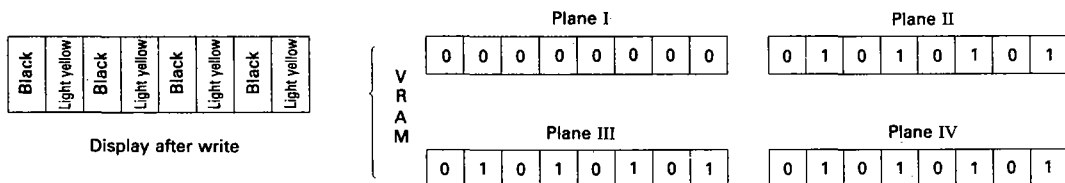


It develops the screen when a next CG patterns are written after setting the REPLACE mode and the light yellow color in the WF register.

So, the bit "1" of the write data becomes the color specified by WF and rest of others become RESET (black).

	Mode	B/ $\bar{A}$	Color designation		Write data (CG pattern)
WF register	1 0 0	0	1 1 1 0	: Light yellow replace mode	0 1 0 1 0 1 0 1

: Light yellow replace mode

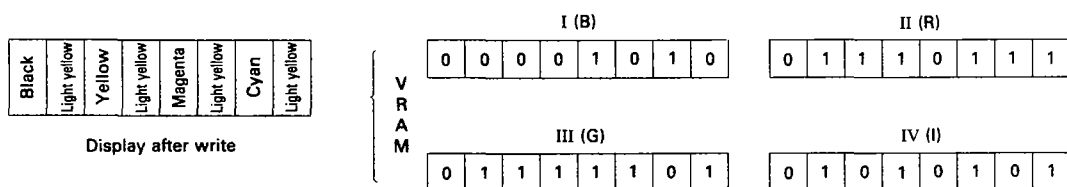


② PSET write

- To overlay a light yellow hatching over the graphic display screen of ①.

	Mode				B/ $\bar{A}$	Color designation				
WF register	1	1	0	0	1	1	1	0	:	Light yellow PSET mode
					I	G	R	B		

Write data	0	1	0	1	0	1	0	1
------------	---	---	---	---	---	---	---	---



So, only the bit "1" of the write data becomes the color specified by WF in this mode, and rest of other colors do not change.

Read data 0 1 0 1 0 1 0 1 : Only the bit of light yellow becomes "1".

③ SEARCH read + PSET write

- To change light yellow in ② above to change to red
- The following data are set when the memory is read after setting the light yellow search mode in the RF register.

	Mode	B/ $\bar{A}$ Color designation						
RF register	1		0	1	1	1	0	: Light yellow search

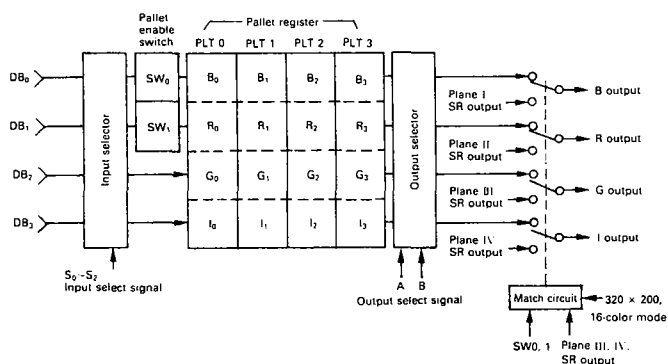
Black	Red	Yellow	Red	Magenta	Red	Cyan	Red
-------	-----	--------	-----	---------	-----	------	-----

Now, a partial color change has been attained. As in above, it enhances fast display change with less of VRAM accessing by using various write modes.

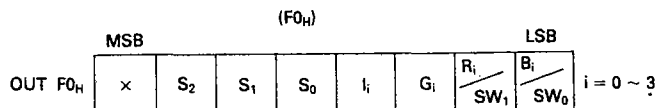
4-8 Pallet

- As there are four 4-bit pallet registers provided inside the unit, it permits choice of R, G, B, and I combinations, and it enables to make choice of any desired two or four colors out of sixteen available colors. However, in the 320 × 200, 16-color mode, choice of colors permitted to four kinds of colors output of sixteen.
- Only the conventional mode is applicable for the MZ-700 mode without using pallet.
- Pallet is not applicable for the border color.

<Configuration>



<Pallet register write> (F0_H)



1) S₀ ~ S₂: Register section

S ₂	S ₁	S ₀	Register No.
0	0	0	PLT 0
0	0	1	PLT 1
0	1	0	PLT 2
0	1	1	PLT 3
1	0	0	SW ₀ , SW ₁

2) Bi, Ri, Gi, Ii: Pallet write data

3) SW₀, SW₁:

With these switches, it is possible to make combination of Planes III and IV data in the 320 × 200, 16-color mode. Switches are used to assign pallets to four groups of colors.

(Plane III data) = SW₀, (Plane IV data) = SW₁

Only for the color information, the color information set by the pallet register are available as B, R, G, and I outputs. For color information other than that, data in Plane I through Plane IV are sent out as the B, R, G, and I outputs.

(See example next.)

<Pallet output and display mode>

- Shown next is the relation of the display mode, color plane data vs R, G, G, I outputs.

Display mode			Display color	Pallet output select		Pallet enable SW ₀ , SW ₁	Output select		Output	Output	Output	Output			
				A	B		A	B							
320 × 200	4 color	Frame A	4 colors out of 16 colors	Plane I data	Plane II data	×	0	0	B ₀	R ₀	G ₀	I ₀			
							1	0	B ₁	R ₁	G ₁	I ₁			
		0	1	B ₂	R ₂		G ₂	I ₂							
		1	1	B ₃	R ₃		G ₃	I ₃							
	Frame B	4 colors out of 16 colors	Plane III data	Plane IV data	×	0	0	B ₀	R ₀	G ₀	I ₀				
						1	0	B ₁	R ₁	G ₁	I ₁				
						0	1	B ₂	R ₂	G ₂	I ₂				
						1	1	B ₃	R ₃	G ₃	I ₃				
	16 colors	—	(Ex.) 16 colors out of 16 colors	Plane I data	Plane II data	$SW_0 = \begin{pmatrix} \text{Plane III} \\ \text{data} \end{pmatrix}$	0	0	B ₀	R ₀	G ₀	I ₀			
							1	0	B ₁	R ₁	G ₁	I ₁			
							0	1	B ₂	R ₂	G ₂	I ₂			
							1	1	B ₃	R ₃	G ₃	I ₃			
										×	×	I	II	III	IV
										0	×	B ₀	R ₀	G ₀	I ₀
										1	×	B ₁	R ₁	G ₁	I ₁
										0	×	B ₀	R ₀	G ₀	I ₀
640 × 200	2 colors	Frame A	2 colors out of 16 colors	Plane I data	×	×	0	×	B ₀	R ₀	G ₀	I ₀			
							1	×	B ₁	R ₁	G ₁	I ₁			
		Frame B	2 colors out of 16 colors	Plane III data	×	×	0	×	B ₀	R ₀	G ₀	I ₀			
							1	×	B ₁	R ₁	G ₁	I ₁			
	4 colors	—	4 colors out of 16 colors	Plane I data	Plane III data	×	0	0	B ₀	R ₀	G ₀	I ₀			
							1	0	B ₁	R ₁	G ₁	I ₁			
							0	1	B ₂	R ₂	G ₂	I ₂			
							1	1	B ₃	R ₃	G ₃	I ₃			

(Ex.)

An example of the pallet in use in the 320 × 200, 16-color mode

- Assume that the pallet register has been set to the following.

PLT0 = Black
PLT1 = Cyan
PLT2 = Red
PLT3 = Magenta

- When SW₀ is set to "0" and SW₁ to "0", the pallet is applied to four colors in group 1 (III = 0, IV = 0) and it results in the color as shown in ① of the table right (yellow to cyan).
- When SW₀ is set to "0" and SW₁ to "1", four colors of group 3 (III = 0, IV = 1) becomes the display color set by the pallet.
- Therefore, any color can be chosen out of 16 colors against four colors of color group selected by SW1 and SW2.
- For group other than selected by SW₀ and SW₁, the color that I ~ IV outputted on B, R, G, I is displayed.

Group	Plane data				Display color of I ~ IV → RGBI	SW ₀ = 0 SW ₁ = 0	SW ₀ = 0 SW ₁ = 1
	I	II	III	IV			
Group 1	0	0	0	0	Black	PLT0 = Black	Black
	1	0	0	0	Blue	PLT1 = Cyan	
	0	1	0	0	Red	PLT2 = Red	
	1	1	0	0	Magenta	PLT3 = Magenta	
Group 2	0	0	1	0	Green	←	←
	1	0	1	0	Cyan	←	←
	0	1	1	0	Yellow	←	←
	1	1	1	0	White	←	←
Group 3	0	0	0	1	Gray	←	PLT0 = Gray
	1	0	0	1	Light blue	←	PLT1 = Light blue
	0	1	0	1	Light red	←	PLT2 = Light red
	1	1	0	1	Light magenta	←	PLT3 = Light magenta
Group 4	0	0	1	1	Light green	←	←
	1	0	1	1	Light cyan	←	←
	0	1	1	1	Light yellow	←	←
	1	1	1	1	Light white	←	←

Border color

- As the CRTC has a 4-bit border color register, it permit to use any border color out of 16 colors.
- Border register (OUT 06CF_H)

	MSB				LSB			
BCOL	x	x	x	x	I	G	R	B

- B, R, G, and I becomes "0" (black) when reset.

4-9 CRTC register map

- VRAM control
- Data display on the video screen

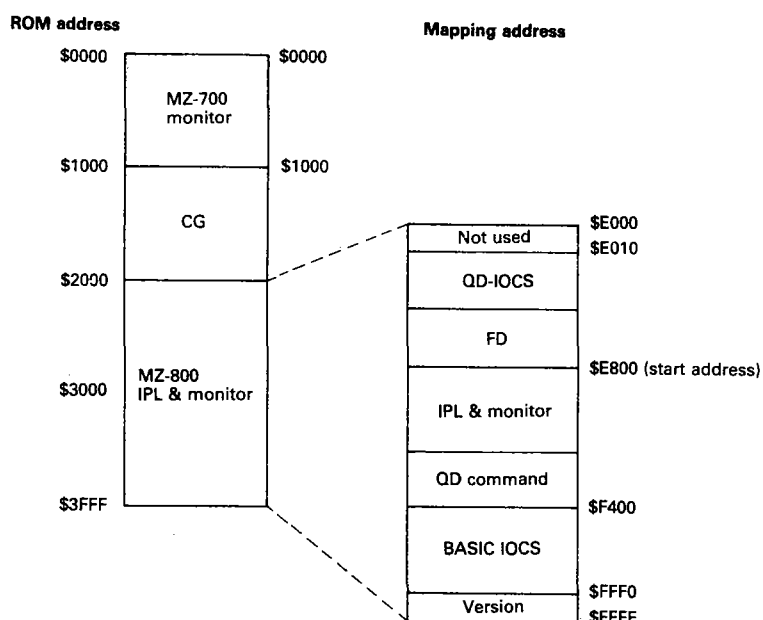
Control I/O address map

I/O address		IN/OUT	
H (B)	L (C, ☆)		
—	CC	O	Write format register (WF)
—	CD	O	Read format register (RF)
—	CE	O	Display mode register (DMD)
—	CE	I	Status read
01	CF	O	Scroll offset register L (SOF1), 8 bits
02	CF	O	Scroll offset register R (SOF2), 2 bits
03	CF	O	Scroll width register (SW), 7 bits
04	CF	O	Scroll start address register (SSA), 7 bits
05	CF	O	Scroll end address register (SEA), 7 bits
06	CF	O	Border color register (BCOL), 4 bits
07	CF	O	Superimpose bit (D7) (CKSW), 1 bit
	FD	O	Pallet register

Written by indirect
OUT command.
B register ← 0~7
OUT(C),A

4-10 ROM configuration

The MZ-700 monitor, character generator (CG), MZ-800 monitor, and IPL are implemented on a single chip of 16k × 8-bit ROM.



5 8255 Programmable Peripheral Interface

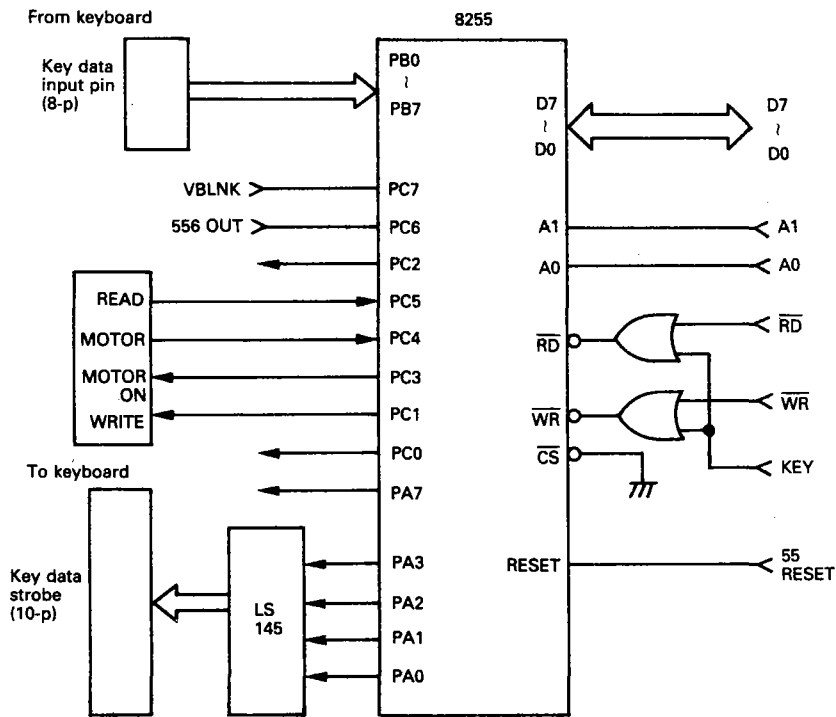
The 8255 has three pairs of 8-bit I/O ports, each one can be assigned to input or output port by means of programming. A different mapping is established de-

pending on the mode. In the MZ-700 mode, it is on memory space, and in the MZ-800 mode, it is on I/O space.

Port name (address)	Pin No.	I/O	Active state	Function
PA (700 \$E000) (800 \$D0)	PA ₀ PA ₁ PA ₂ PA ₃ PA ₄ PA ₅ PA ₇	O	H H H H L L L	Keyboard scan strobe Joystick-1 strobe Joystick-2 strobe CRT cursor blink timer reset
PB (700 \$E001) (800 \$D1)	PB ₀ PB ₁ PB ₂ PB ₃ PB ₄ PB ₅ PB ₆ PB ₇	I	L L L L L L L	Keyboard scan input
PC (NOTE-1) (700 \$E002) (800 \$D2)	PC ₀ PC ₁ PC ₂ PC ₃ PC ₄ PC ₅ PC ₆ PC ₇	O O O O I I I	L — L — H — — —	Prohibits sound output of the 8253 Cassette write data Disables timer interrupt Rotates the cassette motor Checks the cassette motor Cassette read data CRT cursor blink timer input Vertical blink signal
(700 \$E003) (800 \$D3)	—	—	—	Control por

NOTE-1: Output data dependent on the bit set mode.

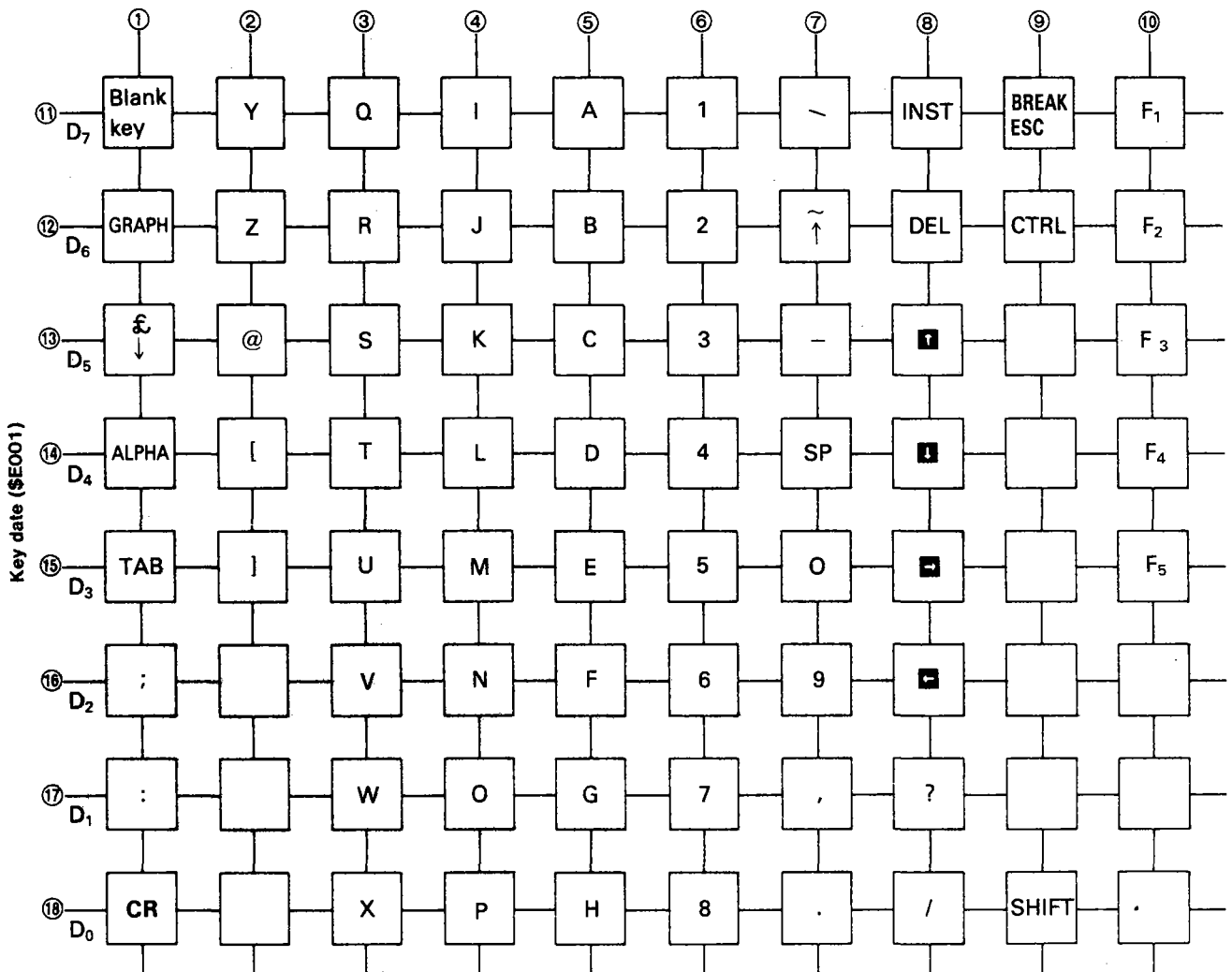
NOTE-2: Motor is controlled on and off by the rising edge of the signal.



a) Key scan

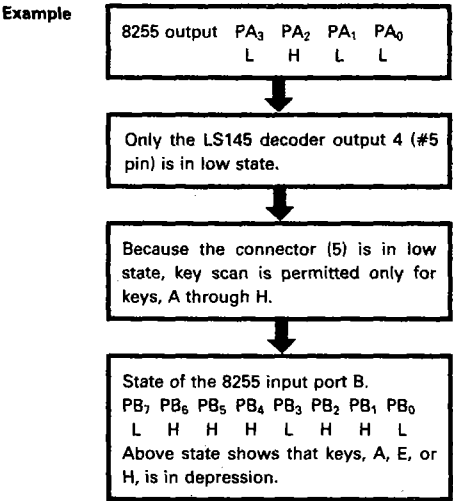
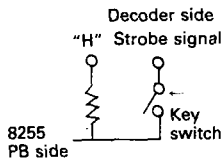
Ports PA₀~PA₃ of the 8255 are connected via the

LS145 decoder, and PB₀~PB₇ are connected to the key matrix directly.



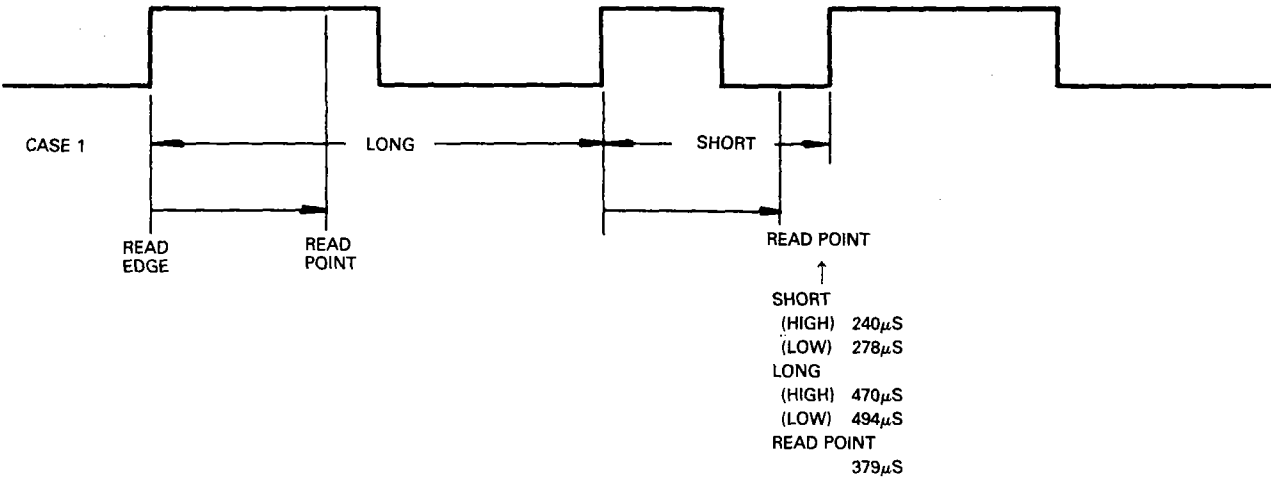
Key strobe is issued through PA₀~PA₃ to scan the key. As it is supplied to the decoder, it makes one of outputs, 0~9, set low. It is then added to the key matrix to scan the line of the key depressed (vertical key matrix scan). The line is in the low state, if it is in depression (horizontal key matrix scan).

NOTE: In the ready for command state, PA₀~PA₃ are normally repeats to be low state and the decoder outputs repeats to be high state. But, since the decoder is of an open collector type, it would not permit to check high and low state.



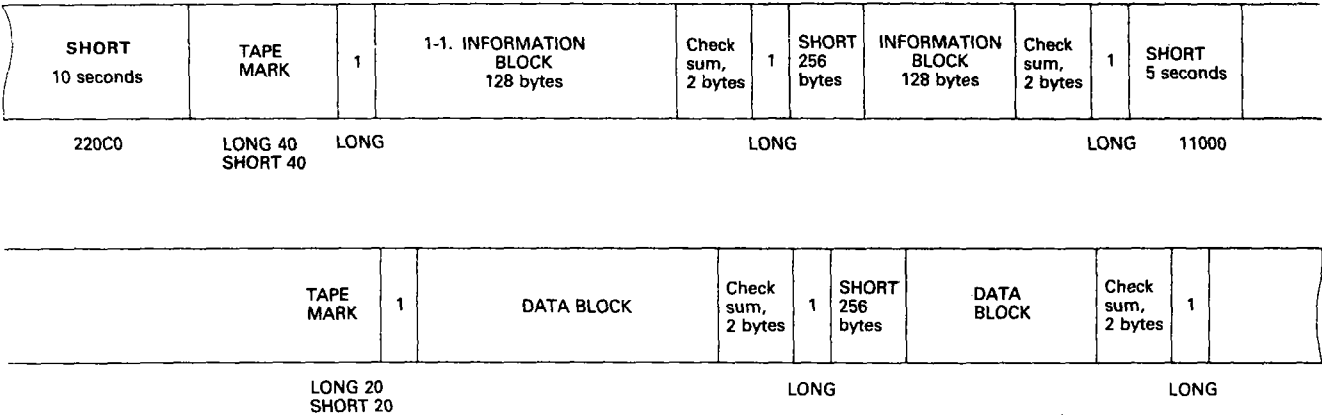
b) Cassette control
The 8255 issues the cassette write data from PC1 and

read signal through PC5. The type of data (input, output) and its format are as follows:



LONG represents the bit value "1" and SHORT the bit value "0". Data will be read at 368 microseconds after

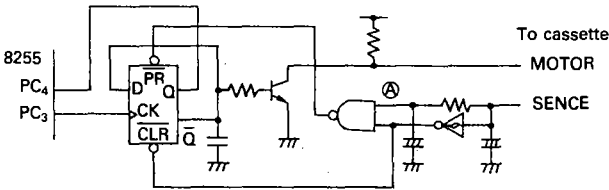
the signal rising edge. Data are recorded in repetition of LONG and SHORT, and the same data are written twice.



See next for the contents of the information block.

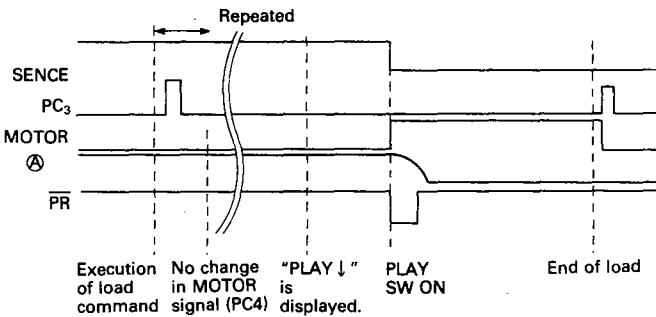
Name	Byte count	Function	Note
ATRB	1	Attribute	
NAME	17	File name (16 characters maximum)	CR (0D) affixed
SIZE	2	File byte size	In order of low to high order
DTADR	2	Loading address	
EXADR	2	Execution address	
COMNT	104	Comment	Not used

Rotation of the cassette (dedicated) is controlled by the 8255 and its peripheral circuits.

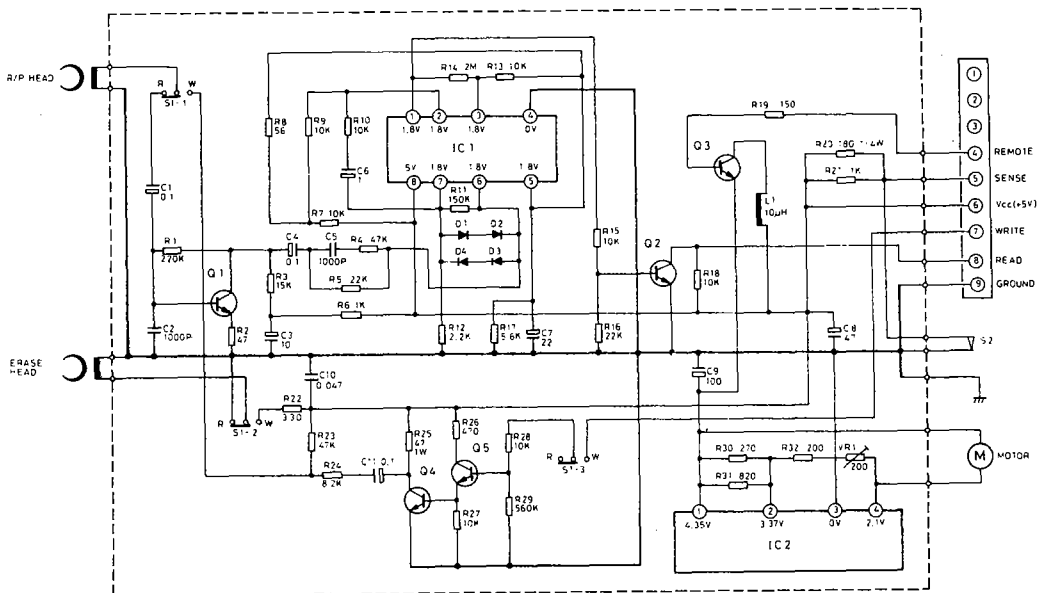


If switch has not been ON on the cassette recorder side, SENCE signal is in high state. When a switch (REW, FF, etc.) is pushed, it makes the signal turned low. It presets the D-FF and the motor starts to rotate with MOTOR in high state. With lock given to the D-FF through PC3, it permits on/off control of the motor. If a switch is pushed on the cassette recorder side, it permits examination of the motor operating state by means of L and PC4.

For use of other than MZ-800 cassette tape recorder type, it needs to short SENCE to GND, READ to EXREAD, and WRITE to EXWRITE of the connector T-5. Use of the cassette recorder of other kind may sometimes not permit proper loading and saving operation. In such an event, adjust the volume and tone controls to find the optimum positions. To meet the opposite polarity of cassette tape recorder, there is a dip switch provided. Changing the switch position makes TPSW signal state changed so as to invert the signal waveform.



Casset Recorder Circuit



IC1 --- 25C1815-GR, AN5562, 10001580P8 or LM3580P
IC2 --- 25C2120-GR, 25D467-C
Q1,2,5 --- 2SC1815-GR, 2SC945-ParQ
Q3,4 --- 2SC2120-GR or 2SD467-C
D1~4 --- 1S1555, 1S1588, DS442 or 1N4148

S1 --- WRITE(REC)/READ(PLAY) SWITCH (READ POSITION)
S2 --- PLAYBACK POSITION

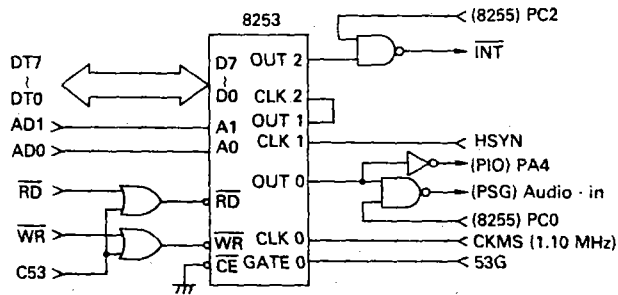
VOLTAGE CHECK POSITION
--- V --- PLAYBACK POSITION

NOTES:
1. ALL RESISTANCE VALUES ARE IN Ω, K, 1000Ω, M, 1000KΩ.
2. ALL CAPACITANCE VALUES ARE IN pF, P, 10P, μF.
3. ALL RESISTORS ARE 1/8 WATT, UNLESS OTHERWISE SPECIFIED.
4. VOLTAGES MEASURED FROM POINT INDICATOR TO CHASSIS GROUND WITH V.T.V.M. AT LINE VOLUME CONTROL MINIMUM AND NO SIGNAL (SPECIFICATIONS AND CIRCUITS SUBJECT TO CHANGE WITHOUT NOTICE FOR IMPROVEMENT).

6 8253 Programmable Interval Timer

The 8253 makes sound generated with the counter #0 and internal timer is operated with the counters #1 and #2.

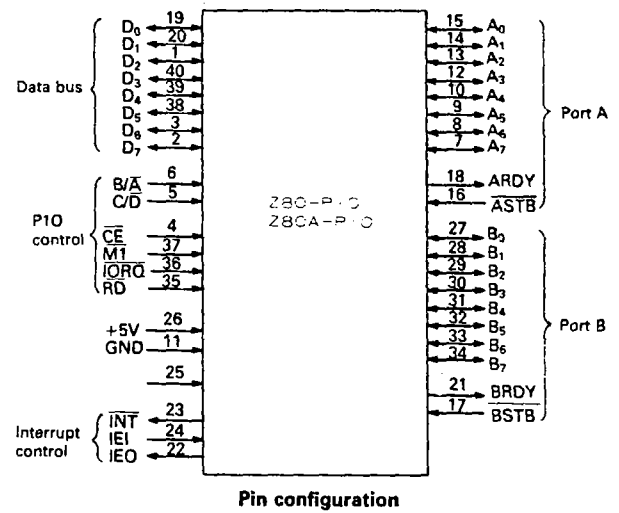
- Counter mode
 - #0 Square waveform generator MODE3
 - #1 Rate generator MODE2
 - #2 Interrupt on terminal counter



- The counter #0 counts input pulse of 1.1 MHz, divided by the predetermined rate (musical score data) to generate sound. It is connected with the mixing audio amplifier through AUDIO-IN of the sound IC (76489AN). This counter output is gated by PC of the 8255 port C, and the counter gate is controlled by D01 of \$E008. The counter #0 output is also used for interrupt control INT0 and connected to A4 of the Z-80A PIO port A.
- The counter #1 counts pulse of 15.6 kHz and generated a pulse on OUT1 at every second. The counter #2 counts pulses and makes OUT2 turned high. OUT2 outputs becomes $\overline{\text{INT}}$ via the gate and is connected to INT of the CPU.

7 Printer interface

The Z-80A PIO is used for the printer interface. It has a pair of 8-bit I/O ports.



Pin name	Pin No.	I/O	Signal name	Description															
D ₀ ~D ₇	19,20,1 40,39,38 3,2	I/O	Z80-CPU Data Bus	Bidirectional, 3-state, Z-80 CPU bus. Data and command transfer between the Z-80 CPU and the PIO is carried out through this data bus. D ₀ is the least significant digit.															
B/ $\overline{A}$	6	I	Port B or A Select	Port select signal. Depending on the state of this signal, the port is specified through which data or command is transferred between the Z-80 CPU and the PIO. } H : Port B } L : Port A															
C/ $\overline{D}$	5	I	Control or Data Select	Control/data select signal. Depending on the state of this signal, control port or data port is selected for the port assigned with B/ $\overline{A}$. <table><tr><th>B/$\overline{A}$</th><th>C/$\overline{D}$</th><th>Selected port</th></tr><tr><td>L</td><td>L</td><td>Port A data</td></tr><tr><td>L</td><td>H</td><td>Port A control</td></tr><tr><td>H</td><td>L</td><td>Port B data</td></tr><tr><td>H</td><td>H</td><td>Port B control</td></tr></table>	B/ $\overline{A}$	C/ $\overline{D}$	Selected port	L	L	Port A data	L	H	Port A control	H	L	Port B data	H	H	Port B control
B/ $\overline{A}$	C/ $\overline{D}$	Selected port																	
L	L	Port A data																	
L	H	Port A control																	
H	L	Port B data																	
H	H	Port B control																	
$\overline{CE}$	4	I	Chip Enable	Chip enable signal. A low on this line enables the PIO. Normally connected with the I/O address decoder output.															
ϕ	25	I	System Clock	System clock CPU clock ϕ is usually used.															
$\overline{M1}$	37	I	Machine Cycle One	Connection with CPU $\overline{M1}$ signal (low active). The PIO attains synchronization with the CPU interrupt control logic by $\overline{M1}$. The PIO will be reset when $\overline{M1}$ is set low at least for a period of two clock cycles after turning $\overline{IORQ}$ and $\overline{RD}$ high state.															
$\overline{IORQ}$	36	I	Input Output Request	Connection with CPU $\overline{IORQ}$ signal (low active). This signal perform data transfer between the CPU and the PIO in connection with B/ $\overline{A}$, C/ $\overline{D}$, $\overline{CE}$, and $\overline{RD}$. If $\overline{CE}$, $\overline{RD}$, and $\overline{IORQ}$ are low, the data on the port selected by B/ $\overline{A}$ are transferred to the CPU. If $\overline{CE}$, $\overline{IORQ}$ are low, data or command is written through the port selected by B/ $\overline{A}$.															
$\overline{RD}$	35	I	Read	Connection with CPU $\overline{RD}$ signal (low active). This signal controls the direction of data transfer between the CPU and the PIO in connection with B/ $\overline{A}$, C/ $\overline{D}$, $\overline{CE}$, and $\overline{IORQ}$.															
IEI	24	I	Interrupt Enable in	Interrupt daisy chain signal. The PIO will respond to the INTA cycle of the CPU only when this signal is high.															
IEO	22	O	Interrupt Enable Out	Interrupt daisy chain signal. This signal is high only when IEI is not high with the PIO having an interrupt request. It goes low when IEI is low or PIO is having an interrupt request.															

Pin name	Pin No.	I/O	Signal name	Description
$\overline{\text{INT}}$	23	O	Interrupt Request	Connection with CPU $\overline{\text{INT}}$ signal. A low on this line causes the PIO to place an interrupt request to the CPU. Because it is of an open drain type, it is possible to make $\overline{\text{INT}}$ of several peripheral LSI wired OR using the pullup resistance.
$\text{A}_0\sim\text{A}_7$	15~12 10~ 7	I/O	Port A Bus	Port A data bus. Data transfer is carried out with the PIO and peripheral device via this bus. A_0 is the least significant digit.
$\overline{\text{ASTB}}$	16	I	Port A Strobe	Port A strobe. Significance of this signal depends on the Port A operational mode. 1) Byte output mode : It indicates that the peripheral device has received data from the PIO at a rising edge of this strobe. 2) Byte input mode : Peripheral device loads data in the PIO port A input data register at a rising edge of this strobe. 3) Bidirectional mode : The contents of the port A output data register are outputted on $\text{A}_0\sim\text{A}_7$ when the strobe is in low state. 4) Bit mode : Not used.
ARDY	18	O	Register A Ready	Register A ready. Significance of this signal depends on the state of the port A operational mode. 1) Byte output mode : Data are loaded in the port A data output register when this signal goes high, makes $\text{A}_0\sim\text{A}_7$ stable, and it indicates that data can be transferred to a peripheral device. 2) Byte input mode : A high on this line indicates that the port A data input register is not occupied so as to be ready for receiving of a next data into the data register. 3) Bidirectional mode : This signal is used to indicate that data has been ready in the port A output data register. Data will not be issued on $\text{A}_0\sim\text{A}_7$ in this mode, unless $\overline{\text{ASTB}}$ turns low. 4) Bit mode : Not used.
$\text{B}_0\sim\text{B}_7$	27~34	I/O	Port B Bus	Port B data bus. Function of this bus is identical to $\text{A}_0\sim\text{A}_7$. But, it permits to drive a Darlington transistor as the bus can supply 1.5 V, 1.5 mA. B_0 is the least significant digit.
$\overline{\text{BSTB}}$	17	I	Port B Strobe	Port B strobe. Function of this signal is identical to $\overline{\text{ASTB}}$, except for the following: This signal is used to load data from a peripheral device into the port A input data register, when the port A is in the bidirectional mode.
BRDY	21	O	Register B Ready	Register B ready. Function of this signal is identical to ARDY, except for the following: This signal indicates that the port A input data register is unoccupied and is ready for receiving of a next data, when the port A is in the bidirectional mode.

For the MZ-800 the PIO located on the I/O space, and address of ports performs the following:

\$FC Port A control

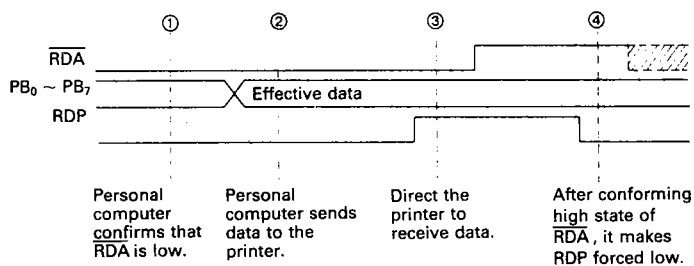
\$FD Port B control

\$FE Port A data

\$FF Port B data

	Pin name	I/O	Active	Signal name	Function
Port A (\$FE)	PA ₀	IN	L	$\overline{RDA}$	A low on this line indicates that the printer data is ready to receive. A low on this line informs the personal a paper depletion during status check.
	PA ₁	IN	L	STA	
	PA ₂	IN	—	GND	
	PA ₃	IN	—	GND	
	PA ₄	IN	H		An 8253 output used for interrupt. Horizontal blanking signal used for interrupt. Used for printer initialization. Indicates the printer to receive data.
	PA ₅	IN	H		
	PA ₆	OUT	H	IRT	
	PA ₇	OUT	H	RDP	
Port B (\$FF)	PB ₀	OUT	—	RD ₀	Printer data or control code to the printer.
	PB ₁	OUT	—	RD ₁	
	PB ₂	OUT	—	RD ₂	
	PB ₃	OUT	—	RD ₃	
	PB ₄	OUT	—	RD ₄	
	PB ₅	OUT	—	RD ₅	
	PB ₆	OUT	—	RD ₆	
	PB ₇	OUT	—	RD ₇	

Interfacing timing



After the personal computer confirms that the printer is ready to receive data at (①), the data is then sent to \$FF port (PIO port B) at step (②).

As reception of data is directed to the printer at step (③), it makes RDP forced low at step (④) upon confirming that the printer received it ($\overline{RDA}=H$). After this, it awaits until $\overline{RDA}$ goes from high to low before transfer of a next data. But, it is possible to transfer successive data by interrupting the CPU at a falling edge of $\overline{RDA}$, since $\overline{RDA}$ is inputted to the $\overline{RSTB}$ input of the PIO, when in the port B mode 0.

It is also possible to interrupt the CPU referring to Port A inputs. Though discussed above is the printer interface method for the MZ compatible printer types, there is the Centronics compatible method for parallel interfacing of the printer. Since this method is basically the same as the MZ mode, except that signal polarity is opposite.

MZ specification		Centronics specification	
Signal name	Active	Signal name	Active
RDA (NOTE)	"L"	BUSY	"H"
RDP	"H"	STB	"L"
IRT	"H"	INPUT PRIME	"L"

NOTE: Though $\overline{RDA}$ is active low, it may be handled the same as high state of BUSY when considered in term of signal significance.

As shown in the figure above, it could be known that RDP and IRT should be inverted in order to make connection with the Centronics compatible printer. It can be attained by changing PRSW to high using the dip switch.

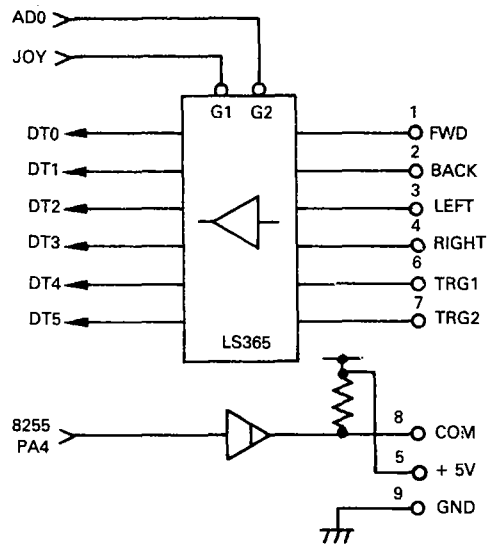
NOTE: When the MZ-800 dedicated printer is used, there may be such a case that proper operation is not attained due to different printing characters and control codes. It must be also noted that all MZ-800 characters can not be printed. Besides, connection with a Centronics compatible printer may not be permitted hardware-wise, sometimes.

8 Programmable sound generator

The SN76489N is used for the programmable sound generator which is controlled by the I/O port \$F2. It is write only. In order to permit smooth sound generation, timer interrupt is applied using the 8253. For the interrupt timer, the count 0 of the 8253 is used. The counter 0 is used for creation of sound steps in the MZ-700 mode, but, it is used for the timer interrupt source of the PSG in the MZ-800 mode. Interrupt is controlled by PA5 of the PIO. It is, however, possible to mask the counter 0 output by PC0 of the 8255, in order to prevent sound generation during interrupt.

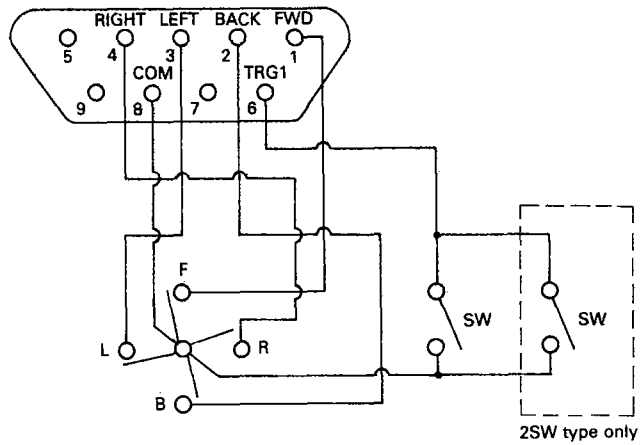
9 Joystick

It permits connection of two ATARI compatible joysticks.



(Configuration of joystick-1)

Low active strobes are issued through PA4 (JOY1) of the 8255 and PA5 (JOY2) interrogate switch activation through inputs to \$F0 (JOY1) and \$F1 (JOY2).



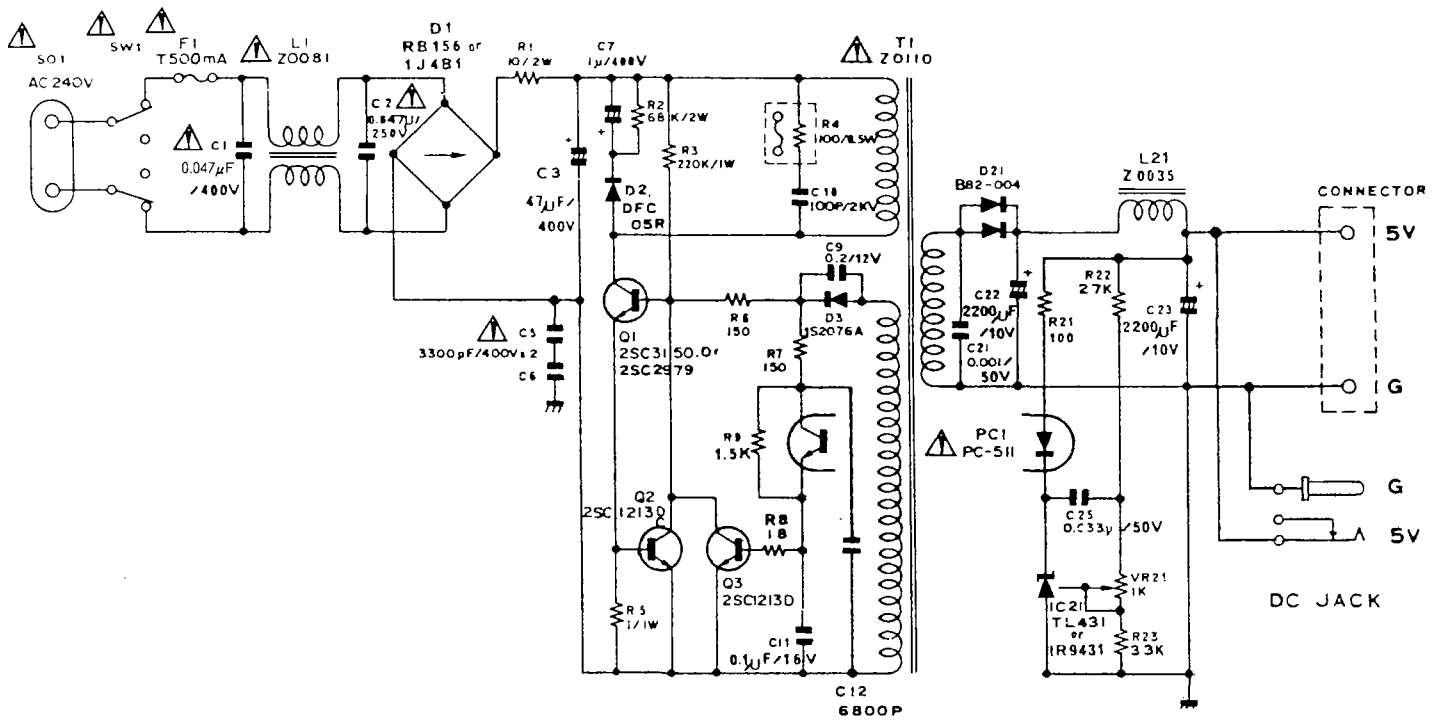
Configuration of ATARI compatible joystick

System switches are assigned as follows:

SW No.	Function	Setup method
1	MZ-700/MZ-800 selection	ON: MZ-700 mode OFF: MZ-800 mode
2 3	MZ/Centronics printer selection	MZ printer with SW2 and SW3 at ON Centronics printer with SW2 and SW3 at OFF
4	External cassette recorder polarity selection	Changed so as to enable read on the external cassette recorder.

* Switch setups at the factory
SW1 OFF (MZ-800 mode)
SW2 ON } (MZ dedicated printer)
SW3 ON }
SW4 ON

10-1 Block diagram



10-2 Operational description

The block diagram of the power supply unit is shown above. It adopts the self-excitation ON/OFF control method. First, the source supply is rectified through the noise filter and converted into direct current. As the dc current is applied to the switching transistor, it causes the transistor to start repeating ON and OFF. After the dc voltage is converted into high frequency pulse, it is added to the primary side of the main transformer which causes to induce voltage on the secondary side. This high frequency pulse is then rectified and filtered to obtain the dc voltage of +5 volts. For control of output voltage, the output voltage is compared with the reference voltage and its error is detected in the control section. While the switching transistor is in the OFF cycle, it makes the photo coupler PC1 active by the detecting signal of the control section for given period. By adjusting control current of the drive circuit, it makes the output stabilized.

10-3 Maintenance

Cleaning

Dust deposit inside the power supply unit may become the cause for overheating as it prevents heat dispersion, which results in damage to components. Stains on the fuse contact and connector contact may lead to contact failure. So, it has to be cleaned using soft cloth dampened with alcohol or dry soft cloth.

10-4 Problem determination and sequence

Follow the next procedure to find the cause of trouble.

- (1) Avoid removing the board to check. But visually observe the board to check for open circuit line, burnt resistor, fuse, and semiconductor chips in the first place.
- (2) If a defective item were found, it has to be replaced with the new one. But, care must also be taken as there are possible defects in multiple numbers of components.

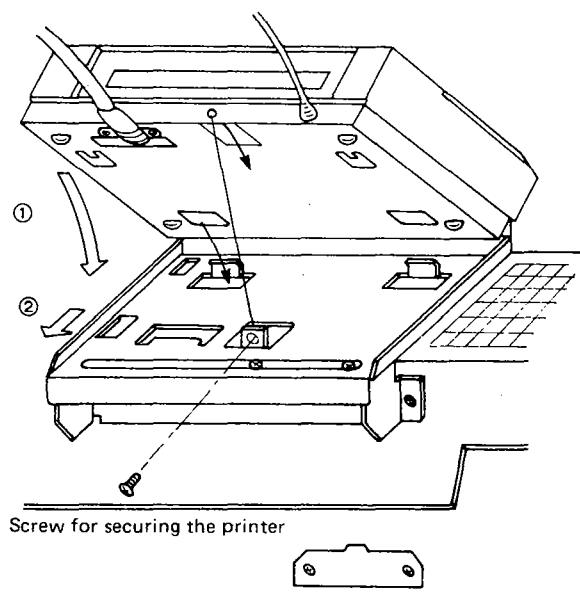
11 MZ-1P16

11-1 Installation

1. Fix the printer unit on the table as shown in the figure.

(Place the printer in the arrow direction ①, lightly move in the arrow direction B, then secure it with screws.)

* It is also possible to use the printer free on the table without securing.

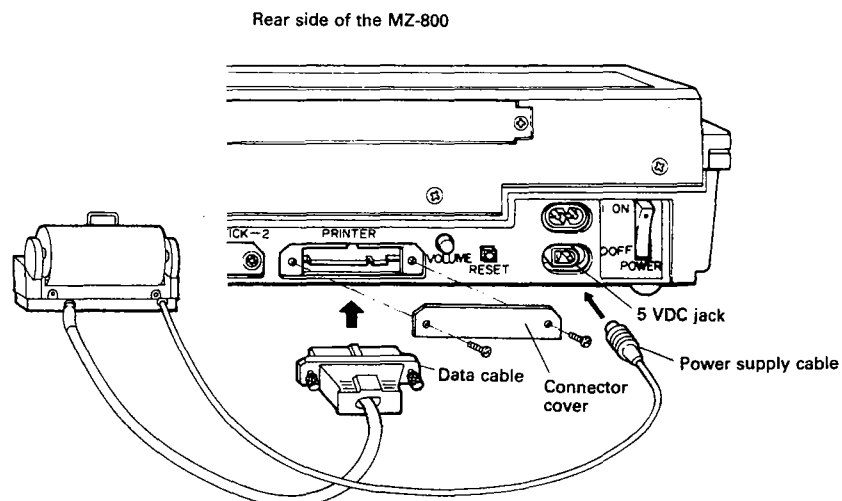


Connection procedure after the installation

- ① Make sure that power is off to the MZ-800 and its peripheral units.
- ② Remove the printer connector cover on the back side of the MZ-800.

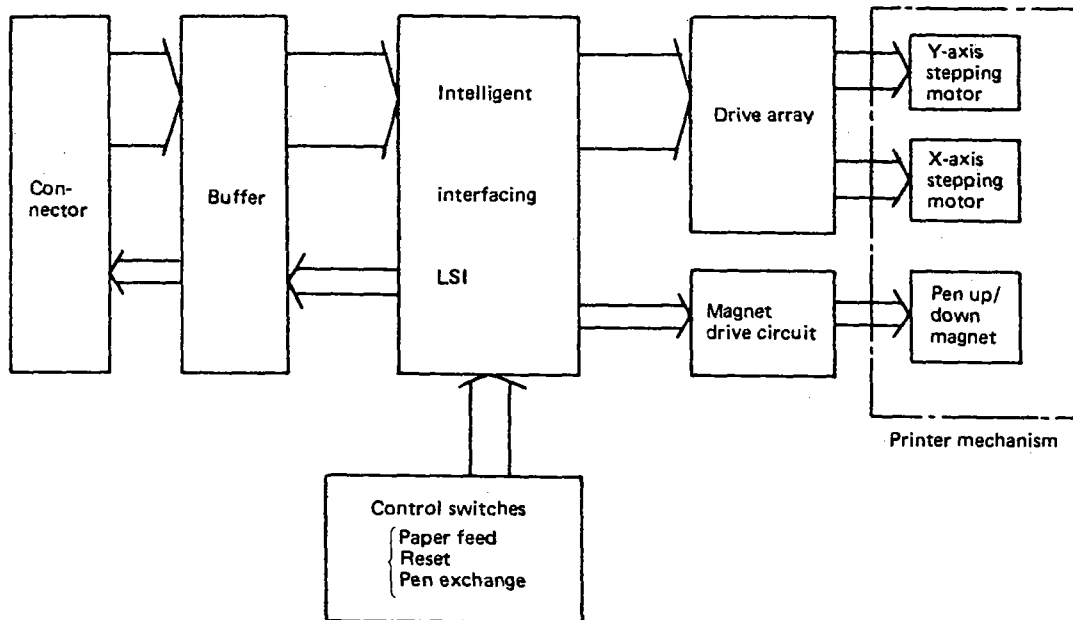
- ③ Connect the data cable and the power supply cable of the unit with the printer connector and the 5 VDC jack of the MZ-800.

- ④ For connection of the printer connector, use the screws that had been at both ends of connector.



* When this unit is in use, set the MZ-800 printer dip switch to the MZ side.

11-2 Block diagram



11-3 At power on

At power on, more than 5V of pen up current is applied for a period of 10ms, plus 5 and minus 0ms, to move the carriage 556 steps backward on the X-axis in order to initialize the colour position. As the carriage is held at the left margin after disengagement of the motor, it is then moved 30 steps forward on the X-axis, then stepped back 30 steps again to check if the colour position detector has been made. If not, it continues to move the carriage 30 steps forward on the X-axis, then return 30 steps to ensure the made condition.

11-4 Colour change operation

To change colour, the slider makes three reciprocating movements of 6mm (30 steps) at the left end of the X-axis to move the pen position one step. When the desired pen position is attained, it then returns to the home position.

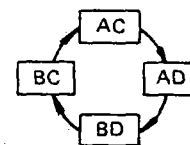
Since the pen rotor makes a unidirectional rotation at the left end of the X-axis, and is locked within printable range, care must be exerted not to touch the rotor and the slider.

11-5 Pen exchange operation

A pen needs to be exchanged with a fresh one when it runs out of ink. In such an event, the pen is moved 485 steps forward on the X-axis from the home position with the used pen located on the top of the rotor, then take out the used pen, by pressing the pen release lever and exchange it with a fresh one.

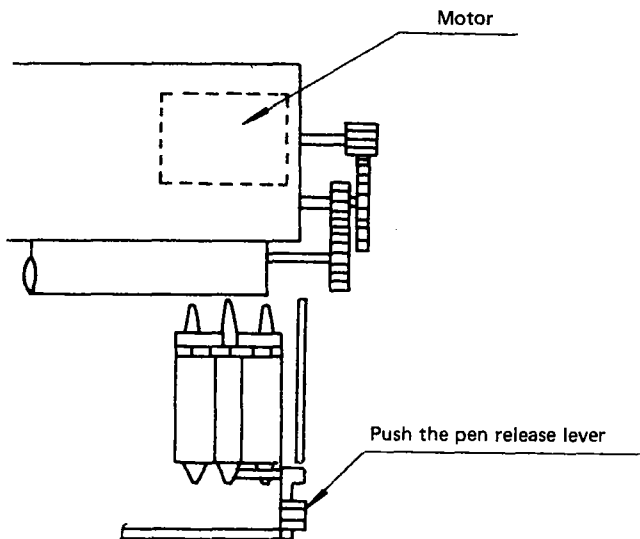
Motor phase and rotating direction

The arrow head indicates the forward direction for both the X-axis and Y-axis.



11-6 Pen exchange method

To remove pen, press the pen exchange button, when the slider is at the right handside, push the pen release lever.



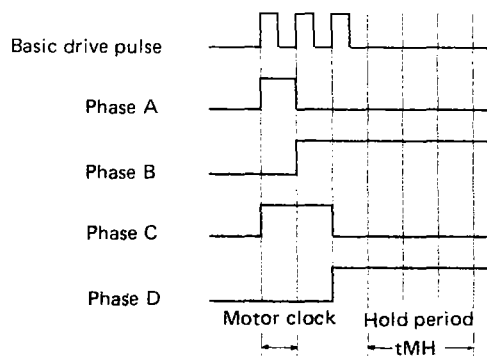
To install the pen, push the tip of the pen through the ring of the return spring in the first place, then push into the holder. Upon completion, ensure that the tip of the pen is engaged with the hole of the pen return spring.

If colour change is done when the pen is disengaged from the hole, it may cause improper rotation of the rotary holder as the slider makes contact with the pen. Do not try to rotate the rotary holder by hand when installing the pen during replacement of the pens.

11-7 Stepping motor driving signal

- The X-axis stepping motor and the Y-axis stepping motor are driven by the two-phase magnet.

Stepping motor driving signal



It is more effective to save power to shut off current while the X and Y axis motors are at a halt. But, there may be a possible malfunction because of unsuppressed vibration, if the current is turned off with a normal pulse width. In order to prevent this, current is applied excessively for more than the given hold time ($t_{MH} = 1\text{ms}$ or more).

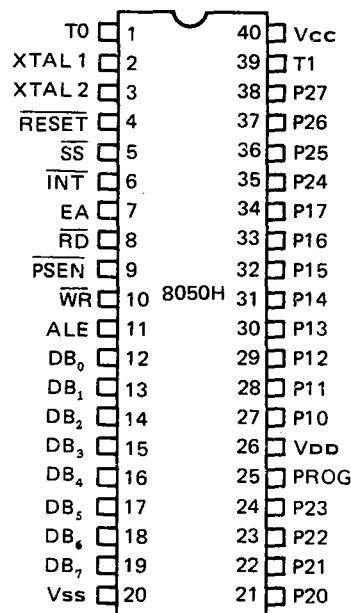
11-8 Colour position detector

The colour position detector consists of a reed switch and a permanent magnet and it may cause malfunction owing to external vibration, and magnetic influence. Especially, when deposit of alien matter or paper fragments is between the left end of the carriage and the frame this may result in a failure of the colour detect performance.

11-9 Character set

Input of an undefined code up to \$20 is ignored. Other undefined codes are represented in hexadecimal notation using the pen in a next colour position.

Pin configuration (top view)



Pin Configuration

11-10 Colour plotter printer control LSI

Pin assignment

Symbol	Name	In/out	Function
V_{SS}	Ground		Connected to 0V.
V_{CC}	Main power		Connected to 5V.
V_{DD}	Power		Connected to 5V.
PROG	Program	Out	Not used.
$P1_0 \sim P1_7$	Port 1		Used as printer control signals.
$P2_0 \sim P2_7$	Port 2		Used for data input port from CPU.
$D_0 \sim D_7$	Data bus		Used for stepper motor control signals.
T_0	Test pin 0	In	Input from pen change switch.
T_1	Test pin 1	In	Input from paper feed switch.
$\overline{INT}$	Interrupt input	In	Data transfer strobe MZ-700 $\rightarrow$ MZ1P01.
$\overline{RD}$	Read signal	In	Not used.
$\overline{WR}$	Write signal	Out	Not used.
$\overline{RESET}$	Reset	In	Used to initialize the processor.
ALE	Address latch enable	Out	Not used.
$\overline{PSEN}$	Program store enable	Out	Not used.
$\overline{SS}$	Single step	In	Not used.
EA	External access	In	Active when EA = 0V.
X_1, X_2	Crystal inputs	In	Pins used to attach the crystal oscillator or RC network to generate internal clock. However, external clock signal may be inputted through these pins.

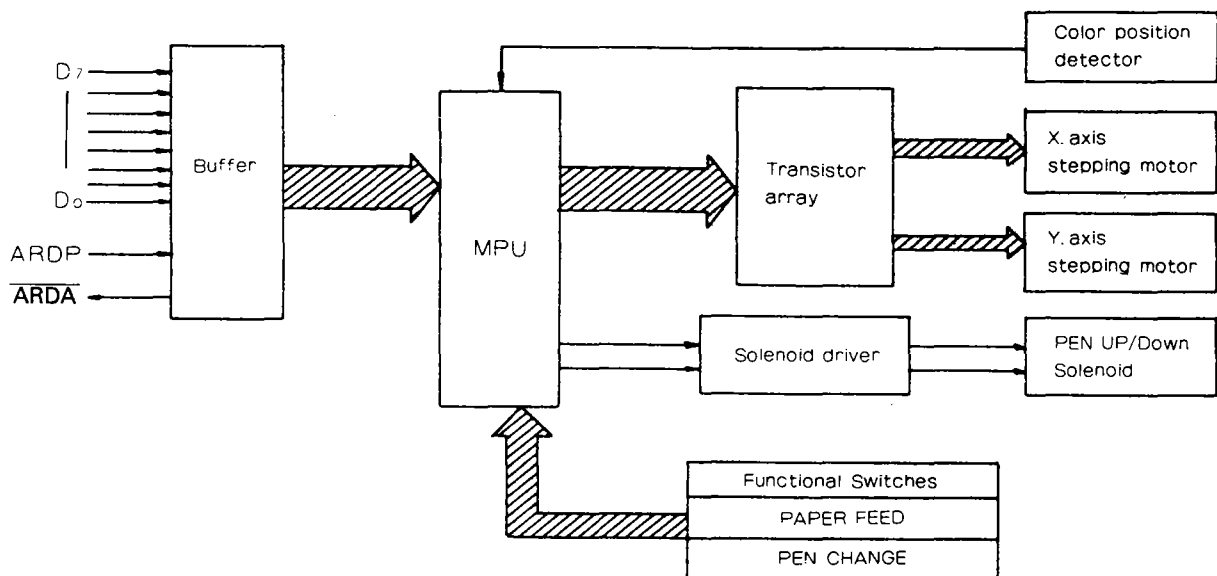
11-11 Interfacing with the MZ-800

Fig.1 shows the block diagram for connection with the printer. Fig.2 shows its circuit description. Fig.3 shows the timing chart.

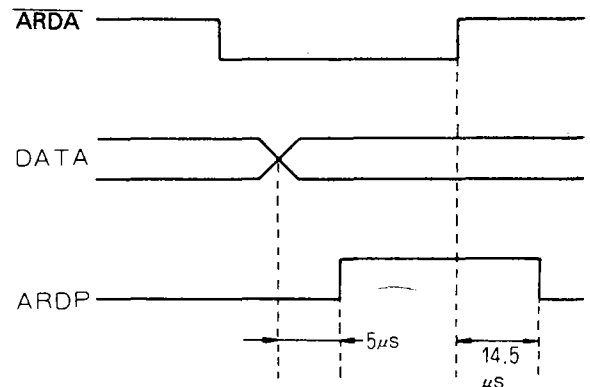
Table of character set

MSD LSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			SP	0	@	P			}		q	n				
1		↓	↑	1	A	Q					a					
2		□	□	2	B	R				e	Z	U				
3		□	□	3	C	S				`	w	m				
4		□	□	4	D	T				~	s					
5		□	□	5	E	U					u					
6		□	□	6	F	V				t	i		→			
7				7	G	W				g		o		=		
8				8	H	X				h	ō	l				
9				9	I	Y					k	ā				
A			*	:	J	Z				b	f	ō				
B			+	:	K	[		°	^	x	v	ā				ℝ
C			g	<	L	\				d						↓
D			-	=	M	]				^	ū	y				
E			.	>	N	↑				p	β	{				
F			/	?	O	←				c	j		=			π

11-12 Block diagram



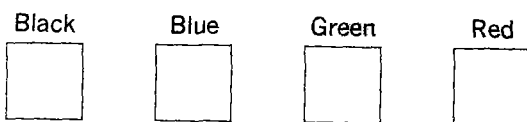
The CPU sends data to the printer after confirming that ARDA is in low state. Five micro seconds later, the strobe signal ARDP goes high. The CPU confirms that ARDA is in high state, ARDP is returned to a low state 14.5 micro seconds later.



11-13 Colour plotter printer control codes

Control codes used in the text mode

- Text code (\$01)
Places the printer in the text mode.
- Graphic code (\$02)
...Same as the BASIC PMODE statement. Places the printer in the graphic mode.
- Line up (\$03)
...Same as the BASIC PSKIP statement. Moves the paper one line in the reverse direction. The line counter is decremented by 1.
- Pen test (\$04)
...Same as the BASIC PTEST statement. Writes the following patterns to start ink following from the pens, then sets scale = 1 (40 chr/line), colour = 0.



- Reduction scale (\$09)+(\$09)+(\$09)
Reduces the scale from 1 to 0 (80 chr/line).
- Reduction cancel (\$09)+(\$09)+(\$0B)
Enlarges the scale from 0 to 1 (40 chr/line).
- Line counter set (\$09)+(\$09)+(ASCII)₂+(ASCII)₁+(ASCII)₀+\$0D
...Same as the BASIC PTEST statement. Specifies the number of lines per page as indicated by the 3 ASCII bytes code. The maximum number of lines per page is 255. Automatically set to 66 when the power is turned on or the system is reset.

- Line feed (\$0A)
...Same as the BASIC PTEST statement. Moves the paper one line in the forward direction. The line counter is incremented by 1.
- Magnify scale (\$0B)
Enlarges the scale from 2 to 1. (26 chr/line)
- Carriage return (\$0D)
Moves the carriage to the left side of the print area.
- Back space (\$0E)
Moves the carriage one column to the left. This code is ignored when the carriage is at the left side of the print area.
- Form feed (\$0F)
Moves the paper to the beginning of the next page and resets the line counter to 0.
- Next colour (\$1D)
Changes the pen to the next colour.

Character scale

- The character scale is automatically set to 1 (40 chr/line) when the power is turned on. Afterwards, it can be changed by the control codes and commands.
- In the graphics mode, the scale can be changed within the range 0 to 63.
- The scale is set to 1 when the mode is switched from graphics to text.

11-14 Graphic mode commands

Command type

In the graphics mode, the computer can control the printer with the following commands.

The words in parentheses are BASIC statements which have the same functions as the graphics mode commands.

Command name	Format	Function
LINE TYPE	Lp (p=0 to 15)	Specifies the type of line (solid or dotted) and the dot pitch. p=0 : solid line, p=1 to 15 : dotted line
ALL INITIALIZE	A	Places the printer in the text mode.
HOME (PHONE)	H	Lifts the pen and returns it to the origin (home position).
INITIALIZE (HSET)	I	Sets the current pen location as the origin (x=0, y=0).
DRAW (LINE)	Dx, y, ..., xn, yn (-999 ≤ x, y ≤ 999)	Draws lines from the current pen location to coordinates (x1, y1), then to coordinates (x2, y2), and so forth.
RELATIVE DRAW (RLINE)	JΔx, Δy, ..., Δxn, Δyn (-999 ≤ Δx, Δy ≤ 999)	Draws lines from the current pen location to relative coordinates (Δx1, Δy1), then to relative coordinates (Δx2, Δy2), and so forth.
MOVE (MOVE)	Mx, y (-999 ≤ x, y ≤ 999)	Lifts the pen and moves it to coordinates (x, y).
RELATIVE MOVE (RMOVE)	RΔx, Δy (-999 ≤ Δx, Δy ≤ 999)	Lifts the pen and moves it to coordinates (Δx, Δy).
COLOR CHANGE (PCOLOR)	Cn (n=0 to 3)	Changes the pen colour to n.
SCALE SET	Sn (n=0 to 63)	Specifies the character scale.
ALPHA ROTATE	Qn (n=0 to 3)	Specifies the direction in which characters are printed.
PRINT	Pc1c2c3 ... cn (n=∞)	Prints characters.
AXIS (AXIS)	Xp,q,r (p= 0 or 1) (q= -999 to 999) (r=1 to 255)	Draws an X axis when p=1 and a Y axis when p=0. q specifies the scale pitch and r specifies the number of scale marks to be drawn.

Command format

There are 5 types of command formats as shown below.

1. Command character only (without parameters)

A, H, I

2. Command character plus one parameter

L, C, S, Q

3. Command character plus pairs of parameters

D, J, M, R

", " is used to separate parameters, and a CR code is used to end the parameter list.

4. Command plus character string

P

The character string is terminated with a CR code.

5. Command plus three parameters

X

", " is used to separate parameters.

《MZ1P16》

11-15 Specification

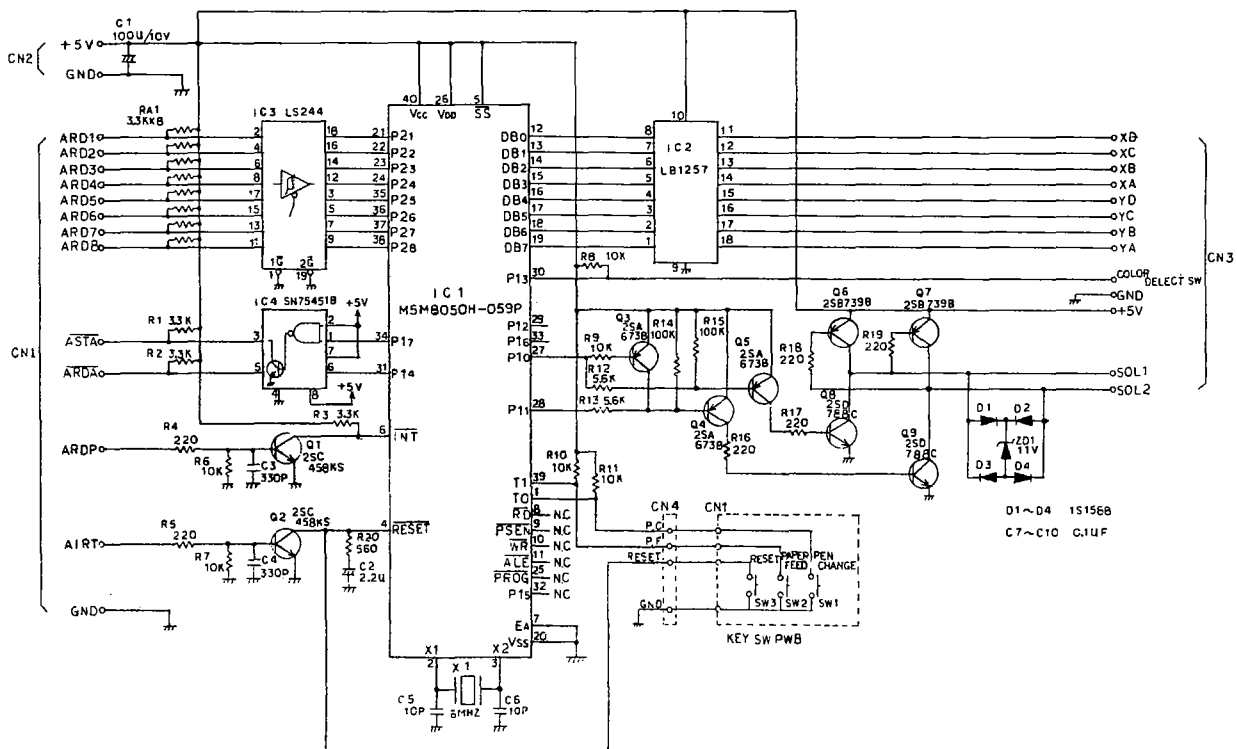
Outline

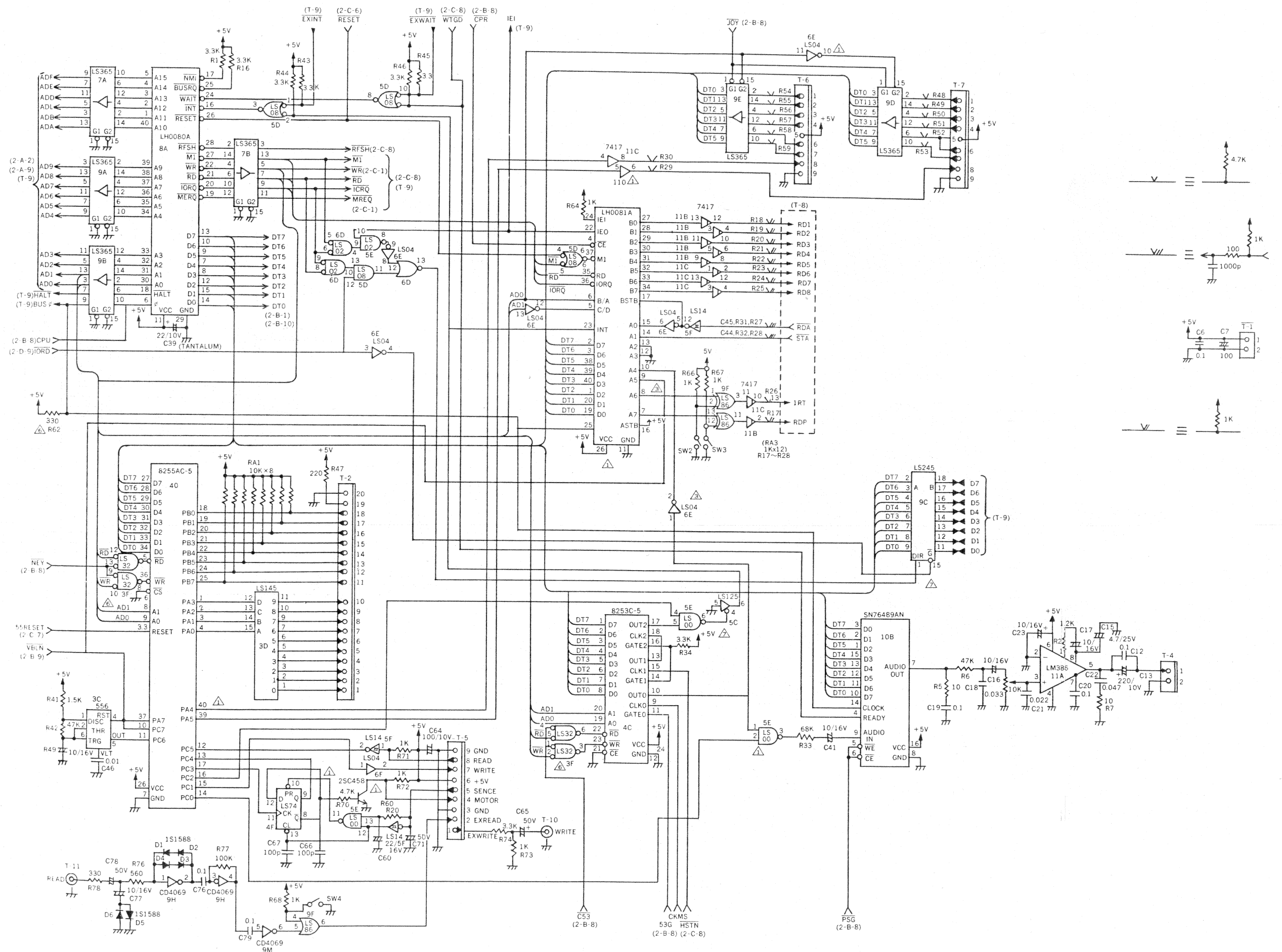
The MZ-1P16 is the external installation 4-color plot printer designed for use with the MZ-800 series personal computer. It can be fixed on the MZ-800 when the table is used.

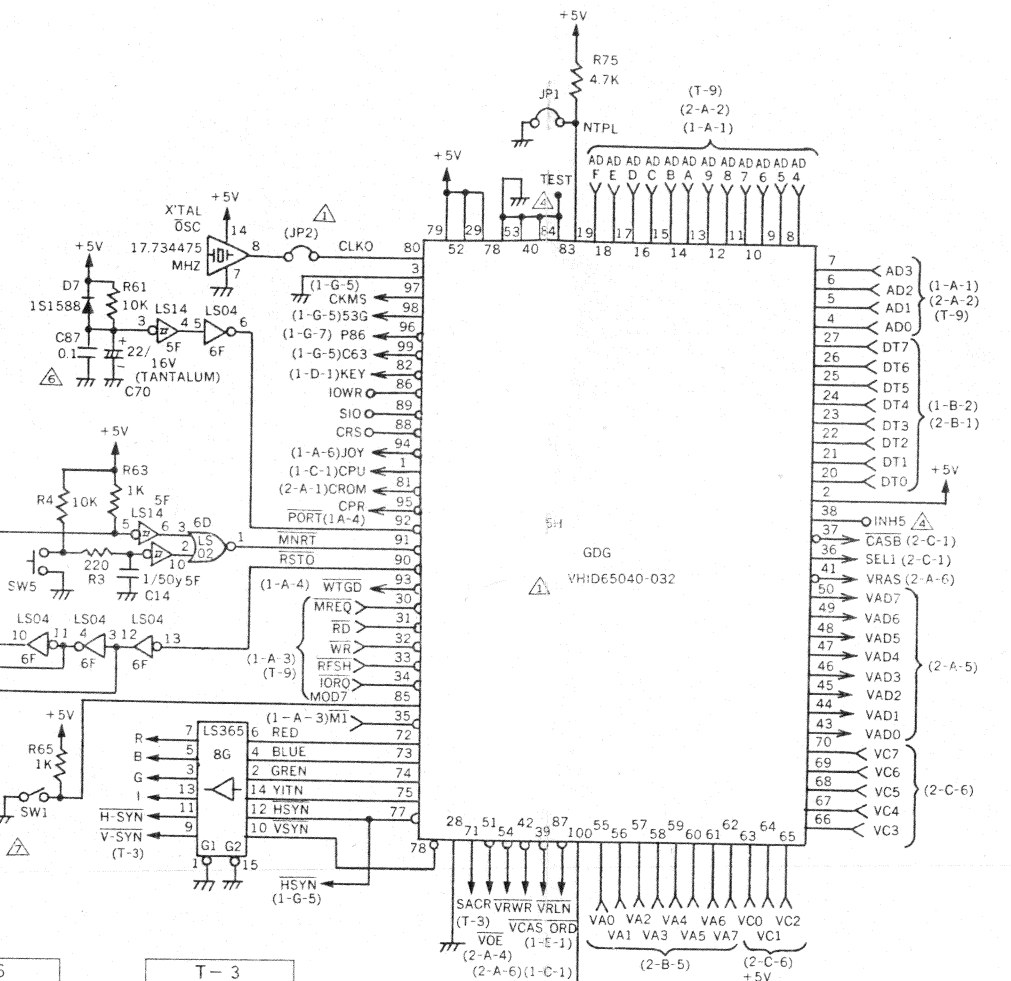
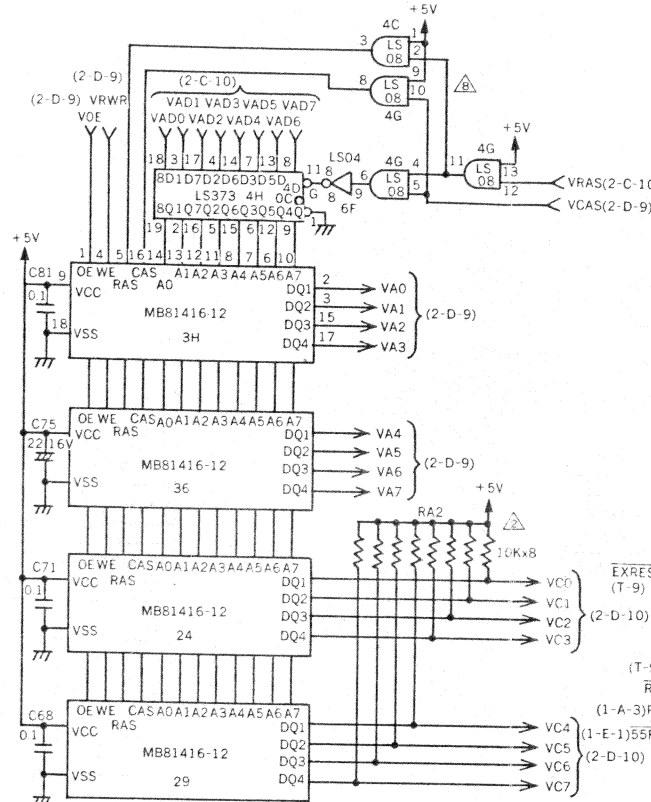
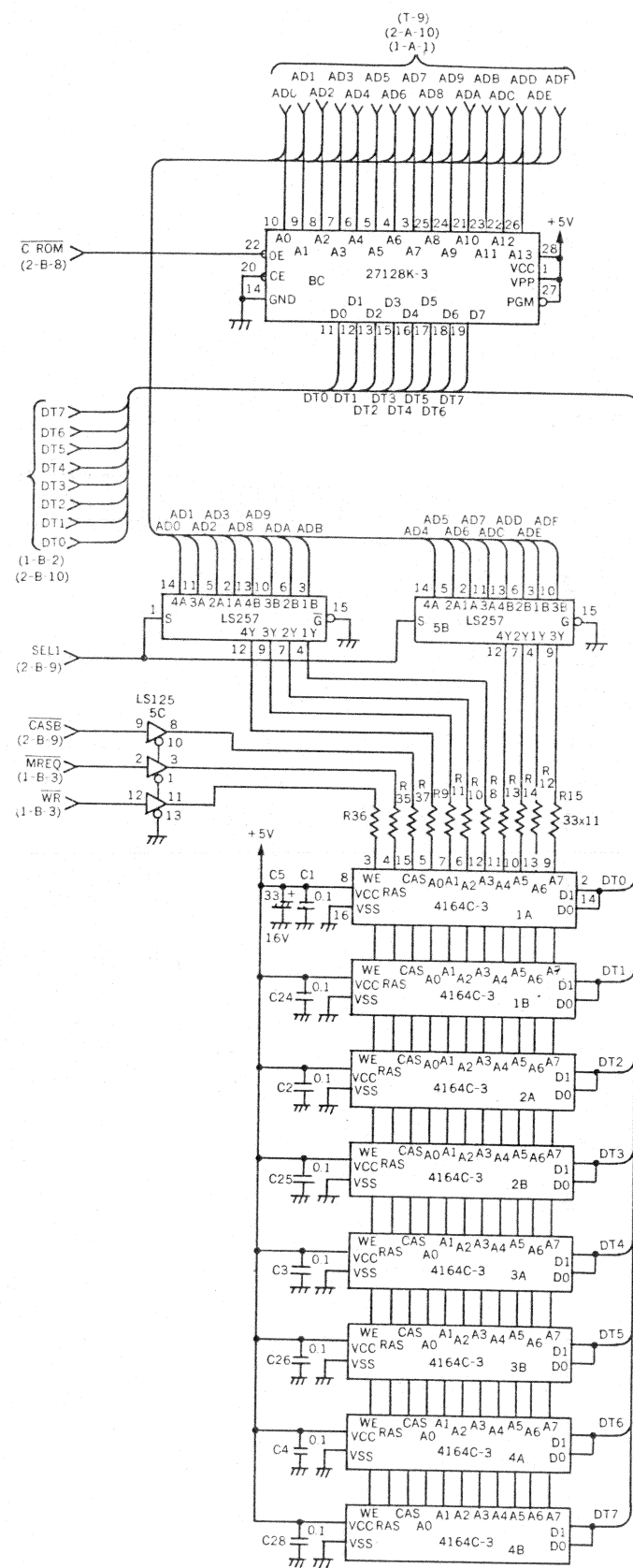
Specification

Type name	: MZ1P16
Print method	: 1, Black; 2, Blue; 3, Green; 4, Red
Print speed	: 10 characters average (smallest letter)
Printing size	: 80/40/26 digits (software assigned)
Character set	: 115
Resolution	: 0.2 mm
Power supply	: +5 V supplied from the MZ-800 via the DC jack
Power consumption	: 11 W
Physical dimensions	: 162(W) × 133(D) × 59(H), excluding accessories.
Weight	: 1 kg (MZ-1P16)
Accessories	: Roll paper (1), ball point pen (one each of black, blue, green, red), paper holder (one each on side), paper shaft (1), paper guide (1)
Operating temperature	: 0 to +35 centigrades
Storage temperature	: -20 to +70 centigrades
Operating humidity	: 80%RH, maximum

11-16 Circuit







T-9	
PARTS SIDE	SOLDER SIDE
1 +5 V	2 +5 V
3 D2	4 D3
5 D1	6 D4
7 D0	8 D5
9 GND	10 D6
11 ADF	12 D7
13 ADE	14 BUS
15 ADD	16 MI
17 ADC	18 WR
19 ADB	20 RD
21 ADA	22 IORQ
23 AD9	24 MREQ
25 AD8	26 GND
27 AD7	28 HALT
29 AD6	30 IEI
31 AD5	32 NC
33 AD4	34 RESET
35 AD3	36 EXRESET
37 AD2	38 EXINT
39 AD1	40 EXWAIT
41 AD0	42 NC
43 GND	44 GND

T-6	
JOY STICK A	
1 FWDA	
2 BACKA	
3 LEFTA	
4 RIGHTA	
5 +5 V	
6 TRG1A	
7 TRG2A	
8 COMA	
9 GND	

T-7	
JOY STICK B	
1 FWDB	
2 BACKB	
3 LEFTB	
4 RIGHTB	
5 +5 V	
6 TRG1B	
7 TRG2B	
8 COMB	
9 GND	
10 GND	
11 SW1	
12 SW2	
13 SW3	
14 SW4	
15 +5 V	

T-3	
COLOR ENCODER	
1 GND	
2 I	
3 GND	
4 H-SYNC	
5 V-SYNC	
6 GND	
7 +5 V	
8 G	
9 B	
10 R	
11 SBCR	
12 GND	

T-8	
PARTS SIDE	SOLDER SIDE
1 RDP	2 GND
3 RD1	4 GND
5 RD2	6 GND
7 RD3	8 GND
9 RD4	10 GND
11 RD5	12 GND
13 RD6	14 GND
15 RD7	16 GND
17 RD8	18 GND
19 IRT	20 GND
21 RDA	22 GND
23 STA	24 GND
25 GND	26 GND

SOFTWARE

1 Character generator (C.G.)

1-1 How to modify C.G. ROM

We will explain here the FONT address in the character generator ROM. The display code corresponding to the character is referenced from the display code table, and multiplied by eight. The result after adding 1000H becomes the address in the ROM where the FONT is stored. When the target character is in the display code table (second 2K-byte half), that display code is multiplied by eight and 1800H is added.

(Example)

The display code for the “☹” character is CFH.

Therefore,

$$CFH \times 8 + 1000H = 1678H$$

The address of the FONT pattern is, therefore, 1678H.

Looking at the character generator dump list we see:

C.G. address	DATA	FONT																																																																
1678H	3CH	<table><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	0	0	1	1	1	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	0	1	0	0	1	0	1	1	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	0	1	0	0	0	0	1	0	0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0																																																											
0	1	0	0	0	0	1	0																																																											
1	0	0	0	0	0	0	1																																																											
1	0	1	0	0	1	0	1																																																											
1	0	0	0	0	0	0	1																																																											
1	0	0	1	1	0	0	1																																																											
0	1	0	0	0	0	1	0																																																											
0	0	1	1	1	1	0	0																																																											
1679H	42H																																																																	
167AH	81H																																																																	
167BH	A5H																																																																	
167CH	81H																																																																	
167DH	99H																																																																	
167EH	42H																																																																	
167FH	3CH																																																																	

↑

LSD

↑

MSD

1-2 C.G.Table

■ Display code table

The display codes are used to address character patterns stored in the character generator. These codes must be transferred to video-RAM to display characters.

Monitor subroutines PRNT(0012H) and MSG(0015H) convert ASCII codes into display codes and transfer them to the V-RAM location indicated for the cursor.

Codes C1H to C6H are for controlling the cursor.

LSD	MSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0 0 0 0	SP	P	0	▬	}	↑	π	□		p	▤	▥	↓	▩	▮	SP
1	0 0 0 1	A	Q	1	▮	♠	<	!	□	a	q	▤	▥	↓	▩	▮	▯
2	0 0 1 0	B	R	2	▮	▴	⌈	"	□	b	r	▤	▥	↑	▩	▮	▯
3	0 0 1 1	C	S	3	▮	■	♥	#	□	c	s	▤	▥	→	▩	▮	▯
4	0 1 0 0	D	T	4	▬	♦	⌋	\$	▬	d	t	▤	▥	←	▩	▮	▯
5	0 1 0 1	E	U	5	▮	←	@	%	▮	e	u	~	▤	▥	▩	▮	▯
6	0 1 1 0	F	V	6	▮	♣	▴	&	▮	f	v	▤	▥	☉	▩	▮	▯
7	0 1 1 1	G	W	7	▮	●	>	'	▮	g	w	▤	▥	☼	▩	▮	▯
8	1 0 0 0	H	X	8	▬	○	↓	(	▬	h	x	▤	▥	H	▩	▮	▯
9	1 0 0 1	I	Y	9	▮	?	▮	)	▮	i	y	▤	▥	I	▩	▮	▯
A	1 0 1 0	J	Z	▬	▬	◐	→	+	▬	j	z	β	▤	♠	▩	▮	▯
B	1 0 1 1	K	£	=	▮	▮	▮	*	▮	k	ä	ü	▤	✱	◊	Y	▯
C	1 1 0 0	L	▮	;	▮	▮	▮	▮	▮	l	▮	ö	▮	✱	▩	▮	▯
D	1 1 0 1	M	▮	▮	▮	▮	▮	▮	▮	m	▮	Ü	▮	✱	▮	▮	▯
E	1 1 1 0	N	H	.	▮	▮	▮	▮	▮	n	▮	Ä	^	●	▮	▮	▯
F	1 1 1 1	O	H	,	▮	:	▮	▮	▮	o	▮	Ö	▮	☺	▮	▮	▯

■MZ-700 Display code table (second 2K-byte half)

MSD LSD		0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
0	0 0 0 0																
1	0 0 0 1																
2	0 0 1 0																
3	0 0 1 1																
4	0 1 0 0																
5	0 1 0 1																
6	0 1 1 0																
7	0 1 1 1																
8	1 0 0 0																
9	1 0 0 1																
A	1 0 1 0																
B	1 0 1 1																
C	1 1 0 0																
D	1 1 0 1																
E	1 1 1 0																
F	1 1 1 1																

1-3 C.G. ROM dump list

(address)	(data)
0000	00 00 00 00 00 00 00 00 18 24 42 7E 42 42 42 00
0010	3E 44 44 3C 44 44 3E 00 38 44 02 02 02 44 38 00
0020	1E 24 44 44 44 24 1E 00 7E 02 02 1E 02 02 7E 00
0030	7E 02 02 1E 02 02 02 00 38 44 02 72 42 44 38 00
0040	42 42 42 7E 42 42 42 00 38 10 10 10 10 10 38 00
0050	70 20 20 20 20 20 20 00 42 22 12 0E 12 22 42 00
0060	02 02 02 02 02 02 7E 00 42 66 5A 5A 42 42 42 00
0070	42 46 4A 52 62 42 42 00 18 24 42 42 42 24 18 00
0080	3E 42 42 3E 02 02 02 00 18 24 42 42 52 24 58 00
0090	3E 42 42 3E 12 22 42 00 3C 42 02 3C 40 42 3C 00
00A0	7C 10 10 10 10 10 10 00 42 42 42 42 42 42 3C 00
00B0	42 42 42 24 24 18 18 00 42 42 42 5A 5A 66 42 00
00C0	42 42 24 18 24 42 42 00 44 44 44 38 10 10 10 00
00D0	7E 40 20 18 04 02 7E 00 30 48 08 1C 08 08 7C 00
00E0	10 10 10 10 10 F0 00 00 00 10 10 10 10 1F 00 00 00
00F0	10 10 10 10 F0 10 10 10 10 10 10 10 10 FF 00 00 00
0100	3C 42 62 5A 46 42 3C 00 10 18 14 10 10 10 7C 00
0110	3C 42 40 3C 0C 02 7E 00 3C 42 40 3C 40 42 3C 00
0120	20 30 28 24 7E 20 20 00 7E 02 1E 20 40 22 1C 00
0130	38 04 02 3E 42 42 3C 00 7E 42 20 10 08 08 08 00
0140	3C 42 42 3C 42 42 3C 00 3C 42 42 7C 40 20 1C 00
0150	00 00 00 7E 00 00 00 00 00 00 7E 00 7E 00 00 00
0160	00 00 10 00 00 10 10 08 00 40 20 10 08 04 02 00
0170	00 00 00 00 00 00 18 18 00 00 00 00 00 10 10 08
0180	00 FF 00 00 00 00 00 00 02 02 02 02 02 02 02 02
0190	01 01 01 01 01 01 01 01 FF 80 80 80 80 80 80 80 FF
01A0	00 00 00 FF 00 00 00 00 08 08 08 08 08 08 08 08
01B0	FF FF 00 00 00 00 00 00 03 03 03 03 03 03 03 03
01C0	00 00 00 00 00 FF 00 00 20 20 20 20 20 20 20 20
01D0	00 00 00 00 FF FF FF FF F0 F0 F0 F0 F0 F0 F0 F0
01E0	00 00 00 00 00 00 00 00 FF 80 80 80 80 80 80 80 80
01F0	00 00 00 00 00 00 FF FF C0 C0 C0 C0 C0 C0 C0 C0
0200	08 10 10 20 10 10 08 00 10 38 7C FE FE 38 7C 00
0210	FF FE FC F8 F0 E0 C0 80 FF FF FF FF FF FF FF FF
0220	10 38 7C FE 7C 38 10 00 00 08 04 FE 04 08 00 00
0230	38 38 D6 FE D6 10 38 00 00 3C 7E 7E 7E 7E 3C 00
0240	00 3C 42 42 42 42 3C 00 3C 42 40 30 08 00 08 00
0250	FF C3 81 81 81 81 C3 FF 00 00 00 00 C0 20 10 10
0260	00 00 00 00 03 04 08 08 01 03 07 0F 1F 3F 7F FF
0270	80 C0 E0 F0 F8 FC FE FF 00 00 10 00 00 10 00 00
0280	00 10 38 54 10 10 10 00 70 18 0C 06 0C 18 70 00
0290	3C 04 04 04 04 04 3C 00 6C FE FE FE 7C 38 10 00
02A0	3C 20 20 20 20 20 3C 00 38 44 52 6A 32 04 78 00
02B0	FF 7F 3F 1F 0F 07 03 01 0E 18 30 60 30 18 0E 00
02C0	00 10 10 10 54 38 10 00 00 02 04 08 10 20 40 00
02D0	00 00 20 40 FE 40 20 00 0F 0F 0F 0F F0 F0 F0 F0
02E0	00 00 00 00 F0 10 10 10 00 00 00 00 1F 10 10 10
02F0	10 10 10 10 1F 10 10 10 00 00 00 00 FF 10 10 10
0300	00 00 80 7C 2A 28 28 00 10 10 10 10 00 00 10 00
0310	24 24 24 00 00 00 00 00 24 24 7E 24 7E 24 24 00
0320	10 78 14 38 50 38 10 00 00 46 26 10 08 64 62 00
0330	0C 12 12 0C 52 22 5C 00 20 10 08 00 00 00 00 00
0340	20 10 08 08 08 10 20 00 04 08 10 10 10 08 04 00
0350	00 10 10 7C 10 10 00 00 10 54 38 7C 38 54 10 00
0360	F0 F0 F0 F0 0F 0F 0F 0F 81 42 24 18 18 24 42 81
0370	08 08 04 03 00 00 00 00 10 10 20 C0 00 00 00 00
0380	FF 00 00 00 00 00 00 00 01 01 01 01 01 01 01 01
0390	FF 01 01 01 01 01 01 01 FF 80 80 80 80 80 80 80
03A0	00 00 FF 00 00 00 00 00 04 04 04 04 04 04 04 04
03B0	80 40 20 10 08 04 02 01 01 02 04 08 10 20 40 80
03C0	00 00 00 00 FF 00 00 00 10 10 10 10 10 10 10 10
03D0	FF FF FF FF 00 00 00 00 0F 0F 0F 0F 0F 0F 0F 0F
03E0	00 00 00 00 00 00 FF 00 40 40 40 40 40 40 40
03F0	00 00 00 00 00 FF FF FF E0 E0 E0 50 E0 E0 E0 E0
0400	18 18 18 18 18 18 18 00 00 00 1C 20 3C 22 5C 00
0410	02 02 3A 46 42 46 3A 00 00 00 3C 42 02 42 3C 00
0420	40 40 5C 62 42 62 5C 00 00 00 3C 42 7E 02 3C 00
0430	30 48 08 3E 08 08 08 00 00 00 5C 62 62 5C 40 3C
0440	02 02 3A 46 42 42 42 00 10 00 18 10 10 10 38 00

1-3 C.G. ROM dump list

(address)	(data)
0000	00 00 00 00 00 00 00 00 18 24 42 7E 42 42 42 00
0010	3E 44 44 3C 44 44 3E 00 38 44 02 02 02 44 38 00
0020	1E 24 44 44 44 24 1E 00 7E 02 02 1E 02 02 7E 00
0030	7E 02 02 1E 02 02 02 00 38 44 02 72 42 44 38 00
0040	42 42 42 7E 42 42 42 00 38 10 10 10 10 10 38 00
0050	70 20 20 20 20 20 20 10 42 22 12 0E 12 22 42 00
0060	02 02 02 02 02 02 7E 00 42 66 5A 5A 42 42 42 00
0070	42 46 4A 52 62 42 42 00 18 24 42 42 42 24 18 00
0080	3E 42 42 3E 02 02 02 00 18 24 42 42 52 24 58 00
0090	3E 42 42 3E 12 22 42 00 3C 42 02 3C 40 42 3C 00
00A0	7C 10 10 10 10 10 10 00 42 42 42 42 42 42 3C 00
00B0	42 42 42 24 24 18 18 00 42 42 42 5A 5A 66 42 00
00C0	42 42 24 18 24 42 42 00 44 44 44 38 10 10 10 00
00D0	7E 40 20 18 04 02 7E 00 30 48 08 1C 08 08 7C 00
00E0	10 10 10 10 10 F0 00 00 00 10 10 10 10 1F 00 00 00
00F0	10 10 10 10 F0 10 10 10 10 10 10 10 10 FF 00 00 00
0100	3C 42 62 5A 46 42 3C 00 10 18 14 10 10 10 7C 00
0110	3C 42 40 3C 0C 02 7E 00 3C 42 40 3C 40 42 3C 00
0120	20 30 28 24 7E 20 20 00 7E 02 1E 20 40 22 1C 00
0130	38 04 02 3E 42 42 3C 00 7E 42 20 10 08 08 08 00
0140	3C 42 42 3C 42 42 3C 00 3C 42 42 7C 40 20 1C 00
0150	00 00 00 7E 00 00 00 00 00 00 7E 00 7E 00 00 00
0160	00 00 10 00 00 10 10 08 00 40 20 10 08 04 02 00
0170	00 00 00 00 00 00 18 18 00 00 00 00 00 10 10 08
0180	00 FF 00 00 00 00 00 00 02 02 02 02 02 02 02 02
0190	01 01 01 01 01 01 01 01 FF 80 80 80 80 80 80 80 FF
01A0	00 00 00 FF 00 00 00 00 08 08 08 08 08 08 08 08
01B0	FF FF 00 00 00 00 00 00 03 03 03 03 03 03 03 03
01C0	00 00 00 00 00 FF 00 00 20 20 20 20 20 20 20 20
01D0	00 00 00 00 FF FF FF FF F0 F0 F0 F0 F0 F0 F0 F0
01E0	00 00 00 00 00 00 00 00 FF 80 80 80 80 80 80 80 80
01F0	00 00 00 00 00 00 FF FF C0 C0 C0 C0 C0 C0 C0 C0
0200	08 10 10 20 10 10 08 00 10 38 7C FE FE 38 7C 00
0210	FF FE FC F8 F0 E0 C0 80 FF FF FF FF FF FF FF FF
0220	10 38 7C FE 7C 38 10 00 00 00 08 04 FE 04 08 00
0230	38 38 D6 FE D6 10 38 00 00 3C 7E 7E 7E 7E 3C 00
0240	00 3C 42 42 42 42 3C 00 3C 42 40 30 08 00 08 00
0250	FF C3 81 81 81 81 C3 FF 00 00 00 00 C0 20 10 10
0260	00 00 00 00 03 04 08 08 01 03 07 0F 1F 3F 7F FF
0270	80 C0 E0 F0 F8 FC FE FF 00 00 10 00 00 10 00 00
0280	00 10 38 54 10 10 10 00 70 18 0C 06 0C 18 70 00
0290	3C 04 04 04 04 04 3C 00 6C FE FE FE 7C 38 10 00
02A0	3C 20 20 20 20 20 3C 00 38 44 52 6A 32 04 78 00
02B0	FF 7F 3F 1F 0F 07 03 01 0E 18 30 60 30 18 0E 00
02C0	00 10 10 10 54 38 10 00 00 02 04 08 10 20 40 00
02D0	00 00 20 40 FE 40 20 00 0F 0F 0F 0F F0 F0 F0 F0
02E0	00 00 00 00 F0 10 10 10 00 00 00 00 1F 10 10 10
02F0	10 10 10 10 1F 10 10 10 00 00 00 00 FF 10 10 10
0300	00 00 80 7C 2A 28 28 00 10 10 10 10 00 00 10 00
0310	24 24 24 00 00 00 00 00 24 24 7E 24 7E 24 24 00
0320	10 78 14 38 50 38 10 00 00 46 26 10 08 64 62 00
0330	0C 12 12 0C 52 22 5C 00 20 10 08 00 00 00 00 00
0340	20 10 08 08 08 10 20 00 04 08 10 10 10 08 04 00
0350	00 10 10 7C 10 10 00 00 10 54 38 7C 38 54 10 00
0360	F0 F0 F0 F0 0F 0F 0F 0F 81 42 24 18 18 24 42 81
0370	08 08 04 03 00 00 00 00 10 10 20 C0 00 00 00 00
0380	FF 00 00 00 00 00 00 00 01 01 01 01 01 01 01 01
0390	FF 01 01 01 01 01 01 01 01 FF 80 80 80 80 80 80 80
03A0	00 00 FF 00 00 00 00 00 04 04 04 04 04 04 04 04
03B0	80 40 20 10 08 04 02 01 01 02 04 08 10 20 40 80
03C0	00 00 00 00 FF 00 00 00 10 10 10 10 10 10 10 10
03D0	FF FF FF FF 00 00 00 00 0F 0F 0F 0F 0F 0F 0F 0F
03E0	00 00 00 00 00 00 FF 00 40 40 40 40 40 40 40 40
03F0	00 00 00 00 00 FF FF FF E0 E0 E0 50 E0 E0 E0 E0
0400	18 18 18 18 18 18 18 00 00 00 1C 20 3C 22 5C 00
0410	02 02 3A 46 42 46 3A 00 00 00 3C 42 02 42 3C 00
0420	40 40 5C 62 42 62 5C 00 00 00 3C 42 7E 02 3C 00
0430	30 48 08 3E 08 08 08 00 00 00 5C 62 62 5C 40 3C
0440	02 02 3A 46 42 42 42 00 10 00 18 10 10 10 38 00

(address)	(data)
08A0	7F 41 77 14 14 14 14 1C 77 55 55 55 55 5D 63 3E
08B0	77 55 55 55 55 2A 14 08 77 55 55 5D 55 55 41 7F
08C0	63 55 2A 14 14 2A 55 63 77 55 49 22 14 14 14 1C
08D0	7F 41 5F 28 14 7A 41 7F 00 02 05 09 FF 7E 00 00
08E0	00 40 A0 90 FF 7E 00 00 00 3E 6B 3E 1C 2A 49 00
08F0	49 2A 1C 7F 1C 2A 49 00 00 00 1C 2A 7F 00 00 00
0900	3E 41 4D 55 55 59 41 3E 1C 12 16 14 14 36 22 3E
0910	3E 41 5D 53 28 74 42 7F 3F 41 5F 44 44 5F 41 3F
0920	30 28 24 2A 6D 41 6F 38 7F 41 7D 21 5E 5F 61 3F
0930	7E 41 7D 3D 41 5D 41 3E 7F 41 5F 28 14 0A 0A 0E
0940	3E 41 5D 3E 41 5D 41 3E 3E 41 5D 41 5E 5F 41 3F
0950	1F 11 7D 55 5F 44 7C 00 F8 88 BE AA FA 22 3E 00
0960	3C 5A FF E7 7E 24 42 81 3C 5A FF E7 7E 24 24 66
0970	10 38 54 FE EE 7C 6C 06 10 38 54 FE EE 7C 6C 28
0980	82 45 3C 5A 7E FF 42 C6 41 A2 3C 5A 7E FF 42 63
0990	00 5A BD 99 24 42 24 00 81 A5 5A 18 18 24 C3 00
09A0	00 24 7E BD 7E 24 24 E7 24 7E BD 7E 24 42 42 C3
09B0	3C 5A FF D5 AB FF BB 91 3C 5A FF D5 AB FF EE 44
09C0	3C 42 A5 81 99 81 AB 55 3C 42 A5 81 99 81 D5 AA
09D0	42 42 66 E7 FF FF 7E 3C 38 7F FC F0 F0 FC 7F 38
09E0	3C 7E FF FF E7 66 42 42 1C FE 3F 0F 0F 3F FE 1C
09F0	3C 7E FF FF FF FF 7E 3C 08 1C 14 14 14 3E 7F 6B
0A00	00 00 E0 7E E3 7E E0 C0 D6 FE 7C 28 28 28 38 10
0A10	00 03 07 7E C7 7E 07 03 3C 30 3C 18 3C 6E 6E 62
0A20	3C 24 3C 18 3C 5A 5A 7E 3C 0C 3C 18 3C 76 76 46
0A30	7E 7E 24 24 24 24 24 36 7E 7E 24 24 24 24 66
0A40	7E 7E 24 24 24 24 24 6C 44 C6 EF ED FF 7E 3C 3C
0A50	1C 36 FF FC F0 FC 3F 1C 3C 3C 7E FF B7 F7 63 22
0A60	38 6C FF 3F 0F FC 7B 3C 7E FF FD FF 7E 3C 3C
0A70	3C 3C 7E FF BF FF 7E 3C 38 6C FF FF FF 7C 38
0A80	1C 36 FF FF FF FF 3E 1C 18 3C 3C 3C 18 3C 3C
0A90	00 00 DE FF FF DE 00 00 3C 3C 18 3C 3C 3C 18
0AA0	00 00 7B FF FF 7B 00 00 04 06 04 04 0C 14 3C 3C
0AB0	00 02 FF D0 E0 C0 00 00 3C 3C 28 30 20 20 60 20
0AC0	00 40 FF 0B 07 03 00 00 08 08 1C 3E 49 08 08 1C
0AD0	00 10 08 8C FF 8C 08 10 1C 08 08 49 3E 1C 08 08
0AE0	00 08 10 31 FF 31 10 08 00 1E 06 0A 12 20 40 00
0AF0	00 40 20 12 0A 06 1E 00 00 02 04 48 50 60 78 00
0B00	00 78 60 50 48 04 02 00 18 7E 7E FF C3 81 81 81
0B10	F8 1E 0E 0F 0F 0E 1E F8 81 81 81 C3 5F 7E 7E 18
0B20	1F 78 70 F0 F0 70 78 1F F0 85 B5 A5 A5 BD 81 FF
0B30	FF 81 BD A5 A1 BF 80 FF FF 81 BD A5 A5 AD A1 BF
0B40	FF 01 FD 85 A5 BD 81 FF 00 18 00 3C 00 7E 00 FF
0B50	80 A0 A8 AA AA A8 A0 80 FF 00 7E 00 3C 00 18 00
0B60	01 05 15 55 55 15 05 01 00 10 38 7C 00 10 38 7C
0B70	00 00 88 CC EE CC 88 00 00 7C 38 10 00 7C 38 10
0B80	00 00 22 66 EE 66 22 00 00 00 E7 A5 E7 00 00 00
0B90	08 1C 2A 08 08 2A 1C 08 00 00 24 42 FF 42 24 00
0BA0	FE 82 44 38 10 10 10 FE AA AA AA AA AA AA AA
0BB0	FF 00 FF 00 FF 00 FF 00 A5 42 A5 00 00 A5 42 A5
0BC0	24 42 81 00 00 81 42 24 FF 01 F9 05 05 05 05 05
0BD0	FF 80 A7 88 A8 88 A8 88 00 00 00 FF 05 F5 05 FF
0BE0	00 00 00 FF 82 82 AA FF 05 F9 01 FF 0C 0C 0C 1E
0BF0	88 87 80 FF 30 30 30 78 01 55 01 A9 01 F1 01 FF
0C00	80 95 80 8A 80 87 80 FF 3C 42 D5 AB 08 08 28 10
0C10	00 00 18 24 24 18 00 00 00 18 24 42 42 24 18 00
0C20	3C 42 81 81 81 81 42 3C 00 00 00 18 18 00 00 00
0C30	00 00 3C 3C 3C 3C 00 00 00 7E 7E 7E 7E 7E 00
0C40	3C 42 B9 85 85 B9 42 3C FF FF FF E7 E7 FF FF FF
0C50	FF FF C3 C3 C3 C3 FF FF FF 81 81 81 81 81 81 FF
0C60	04 0C 04 04 FF 7E 3C 00 3C 42 81 FF FF 81 42 3C
0C70	3C 5A 99 99 99 99 5A 3C 3C 5A 99 FF FF 99 5A 3C
0C80	00 14 7F 55 7F 2A 1C 08 F0 0C 02 72 51 71 01 81
0C90	0F 30 40 4E 8A 8E 80 81 F0 0C 02 71 01 01 81
0CA0	0F 30 40 40 8E 80 80 81 81 01 11 21 C2 02 0C F0
0CB0	81 80 88 84 43 40 30 0F 81 01 01 E1 02 02 0C F0
0CC0	81 80 80 87 40 40 30 0F 81 01 C1 21 C2 02 0C F0
0CD0	81 80 83 84 43 40 30 0F 81 01 E1 11 12 02 0C F0
0CE0	81 80 87 88 48 40 30 0F 10 08 2A 7F 7F 7F 7F 3E

(address)	(data)
0CF0	00 60 10 08 0C 1E 1E 0C 00 4A 20 60 06 34 52 00
0D00	B9 4A 00 C0 03 00 52 91 01 03 07 0F FF FF FF FF
0D10	00 00 80 40 FF C3 C3 FF 00 00 01 02 FF C3 C3 FF
0D20	00 03 04 08 3F 7F FF 3F 80 C0 E0 F0 FF FF FF FF
0D30	40 2B 14 10 2B 2B 10 00 00 7F 42 04 0B 04 42 7F
0D40	00 C0 20 10 FC FE FF FC 00 04 0B 0B 0B 14 12 61
0D50	00 3C 42 42 42 24 A5 E7 00 22 41 41 49 36 00 00
0D60	00 00 36 49 49 36 00 00 00 40 36 09 09 76 00 00
0D70	00 7B 0B 0A 0A 0D 0B 00 00 00 0B 00 3E 00 0B 00
0D80	00 8F DA AA AA 8A 8A 00 FF 91 89 A3 C5 91 89 FF
0D90	FF C3 A5 99 99 A5 C3 FF 00 49 2A 1C 77 1C 2A 49
0DA0	FF 99 99 FF FF 99 99 FF 49 2A 1C 0B 0B 0B 0B
0DB0	1C 0B 1C 0B 1C 0B 1C 0B 00 00 00 55 FF 55 00 00
0DC0	00 0B 0B 3E 0B 0B 00 3E 7E 42 7E 42 7E 42 7E 42
0DD0	00 FF AA AA AA AA FF 00 00 00 00 03 0D 31 C1 FF
0DE0	00 00 00 C0 B0 8C 83 FF 00 00 00 00 3C 7E FF FF
0DF0	FF FF 7E 3C 00 00 00 00 03 07 0F 0F 0F 0F 03
0E00	C0 E0 F0 F0 F0 F0 E0 C0 C0 30 FC FC FF FE EC F8
0E10	03 0C 1D 3B 77 6F DF DF 70 70 50 20 80 80 C0 F0
0E20	5E 2E 2F 2F 2F 5F BF BF 20 72 27 62 F6 FE 06 FC
0E30	04 4E E4 46 6F 7F 60 3F DC 8C D8 F8 0B F8 F0 E0
0E40	3B 31 1B 1F 10 1F 0F 07 8C C0 E0 60 70 7C 0E 0C
0E50	01 03 07 06 0E 3E 70 30 78 70 60 E0 C0 EC FE D1
0E60	1E 0E 06 07 03 37 7F 8B 80 CC DE 9A 31 FB FE FC
0E70	01 33 7B 59 8C 0F 7F 3F FC F8 F8 F0 F0 FE 00 FF
0E80	3F 1F 1F 0F 0F 7F 00 FF 00 80 40 20 40 80 F8 F8
0E90	00 01 02 04 02 01 1F 1F 40 40 40 40 F8 04 FE 00
0EA0	02 02 02 02 1F 20 7F 00 CE CE CE FE FC F8 F0 F0
0EB0	73 73 73 7F 3F 1F 0F 0F F0 F0 F0 1B FE 02 FE FF
0EC0	0F 0F 0F 1B 7F 40 7F FF 1F 22 42 84 84 42 22 1F
0ED0	FF A0 E0 00 00 E0 A0 FF 3F 61 41 81 81 41 61 3F
0EE0	00 00 01 02 FE 01 00 00 00 00 00 00 FF 80 80 80
0EF0	80 80 80 80 FF 00 00 00 FF 01 01 01 01 00 00 00
0F00	00 00 00 00 01 01 01 FF 00 10 30 50 9F 50 30 10
0F10	00 10 30 5C 97 5C 30 10 F8 14 12 7F 11 11 F1 00
0F20	02 03 02 67 90 40 20 F0 02 03 02 47 60 50 F8 40
0F30	02 03 02 F7 80 E0 80 F0 02 05 04 F2 87 E0 80 F0
0F40	03 06 1B 60 1B 06 01 7F 80 60 1B 06 1B 60 80 FE
0F50	00 80 60 B8 54 54 54 F8 D8 F1 A6 8B 93 95 8D CF
0F60	32 EF 0F 1B E0 40 7C 7F FE F9 8C 82 81 81 9F BF
0F70	11 40 02 00 11 82 00 89 02 80 11 00 02 20 01 8B
0F80	00 0C 1A BF FF 9E 0C 00 00 30 5B FD FF 79 30 00
0F90	00 0C 1A BF FC 9F 0C 00 00 30 5B FD 3F F9 30 00
0FA0	0B 14 16 3D 3F 1E 0B 1C 5D 77 55 1C 1C 5D 7F 5D
0FB0	5D 7F 5D 1C 1C 55 77 5D 00 E7 42 FF F9 FF 42 E7
0FC0	00 E7 42 FF 9F FF 42 E7 00 00 3F 3B FE C6 7C 00
0FD0	00 00 FC 1C 7F 63 3E 00 FF 81 A5 81 81 A5 81 FF
0FE0	E7 81 81 00 00 81 81 E7 00 20 10 7F 0B 7F 04 02
0FF0	1B 24 24 04 0B 0B 0B 0B 10 10 10 10 20 24 24 1B

MZ-800 monitor subroutines

The following subroutines are used by the ROM Monitor (9Z-504M). Each subroutine name symbolically represents the function of the corresponding subroutine. These subroutines can be called from user programs.

Registers saved are those whose contents are restored when control is returned to the calling program. The contents of other registers are changed by execution of the subroutine.

Name and entry point (hex.)	Function	Registers saved
CALL LETNL (0006)	Moves the cursor to the beginning of the next line.	All except AF
CALL PRNTS (000C)	Displays a space at the cursor position.	All except AF
CALL PRNTS (0012)	Displays the character corresponding to the ASCII code stored in the ACC at the cursor position. See Appendix J for the ASCII codes. No character is displayed when code 0D (carriage return) or codes 11 to 16 (the cursor control codes) are entered, but the corresponding function is still performed (a carriage return for 0D and cursor movement for 11 to 16).	All except AF
CALL MSG (0015)	Displays a message, starting at the position of the cursor. The starting address of the area in which the message is stored must be loaded into the DE register before calling this subroutine, and the message must end with a carriage return code (0D). The carriage return is not executed. The cursor is moved if any cursor control codes (11 to 16) are included in the message.	All registers
CALL BELL (003E)	Briefly sounds tone of la (about 880 Hz)	All except AF
CALL MELDY (0030)	Plays a tune according to the music data stored in the memory area starting at the address in the DE register. The music data must be in the same format as that for the MUSIC statement of the BASIC, and must end with 0D or C8. When the tune is completed, control is returned to the calling program with the C flag set to 0. When play is interrupted with the BREAK key, control is returned with the C flag set to 1.	All except AF
CALL XTEMP (0041)	Sets the music tempo according to the tempo data stored in the accumulator (ACC). ACC ← 01 Slowest speed ACC ← 04 Middle speed ACC ← 07 Highest speed Note that the data in the accumulator is not the ASCII codes for 1 to 7 but the binary codes.	All registers
CALL MSTA (0044)	Generates a continuous sound of the specified frequency. The frequency is given by the following equation $\text{freq.} = 895 \text{ kHz}/nn'$. Here, nn' is a 2-byte number stored in addresses 11A1 and 11A2 (n in 11A2 and n' in 11A1)	BC and DE

Name and entry point (hex.)	Function	Registers saved																																				
CALL MSTP (0047)	Stops the sound generated with the CALL MSTA subroutine.	All except AF																																				
CALL TIMST (0033)	Sets and starts the built-in clock. The registers must be set as follows before this routine is called. ACC ← 0 (AM), ACC ← 1 (PM) DE ← 4-digit hexadecimal number representing the time in seconds.	All except AF																																				
CALL TIMRD (003B)	Reads the built-in clock and returns the time as follows. ACC ← 0 (AM), ACC ← 1 (PM) DE ← 4-digit hexadecimal number representing the time in seconds.	All except AF and DE																																				
CALL BRKEY (001E)	Checks whether the SHIFT and BREAK keys are both being pressed. The Z flag is set when they are being pressed simultaneously; otherwise, it is reset.	All except AF																																				
CALL GETL (0003)	Reads one line of data from the keyboard and stores it in the memory area starting at the address in the DE register. This routine stops reading data when the CR key is pressed, then appends a carriage return code (0D) to the end of the data read. A maximum of 80 characters (including the carriage return code) can be entered in one line. Characters keyed in are echoed back to the display. Cursor control codes can be included in the line. When the SHIFT and BREAK keys are pressed simultaneously, the BREAK code is stored at the address indicated by the DE register and a carriage return code is stored in the following address.	All registers																																				
CALL GETKY (001B)	Reads a character code (ASCII) from the keyboard. If no key is pressed, control is returned to the calling program with 00 set in ACC. No provision is made to avoid data read errors due to key bounce, and characters entered are not echoed back to the display. When any of the special keys (such as DEL or CR) are pressed, this subroutine returns a code to the ACC which is different to the corresponding ASCII code as shown below. Here, display codes are used to address characters stored in the character generator, and are different from the ASCII codes. <table> <tr> <th>Special key</th><th>Code loaded in ACC</th><th>Display code</th></tr> <tr><td>DEL</td><td>60</td><td>C7</td></tr> <tr><td>INST</td><td>61</td><td>C8</td></tr> <tr><td>ALPHA</td><td>62</td><td>C9</td></tr> <tr><td>BREAK</td><td>64</td><td>CB</td></tr> <tr><td>CR</td><td>66</td><td>CD</td></tr> <tr><td>↓</td><td>11</td><td>C1</td></tr> <tr><td>↑</td><td>12</td><td>C2</td></tr> <tr><td>→</td><td>13</td><td>C3</td></tr> <tr><td>←</td><td>14</td><td>C4</td></tr> <tr><td>HOME</td><td>15</td><td>C5</td></tr> <tr><td>CLR</td><td>16</td><td>C6</td></tr> </table>	Special key	Code loaded in ACC	Display code	DEL	60	C7	INST	61	C8	ALPHA	62	C9	BREAK	64	CB	CR	66	CD	↓	11	C1	↑	12	C2	→	13	C3	←	14	C4	HOME	15	C5	CLR	16	C6	All except AF
Special key	Code loaded in ACC	Display code																																				
DEL	60	C7																																				
INST	61	C8																																				
ALPHA	62	C9																																				
BREAK	64	CB																																				
CR	66	CD																																				
↓	11	C1																																				
↑	12	C2																																				
→	13	C3																																				
←	14	C4																																				
HOME	15	C5																																				
CLR	16	C6																																				

Name and entry point (hex.)	Function	Registers saved																												
CALL ASC (03DA)	Loads the ASCII character corresponding to the hexadecimal number represented by the lower 4 bits of data in ACC.	All except AF																												
CALL HEX (03F9)	Converts the 8 data bits stored in the ACC into a hexadecimal number (assuming that the data is an ASCII character), then loads the hexadecimal number in the lower 4 bits of ACC. The C flag is set to 0 when a hexadecimal number is loaded in ACC; otherwise, it is set to 1.	All except AF																												
CALL HLHEX (0410)	Converts a string of 4 ASCII characters into a hexadecimal number and loads it in the HL register. The call and return conditions are as follows. DE ← Starting address of the memory area which contains the ASCII character string. (e.g., "3" "1" "A" "5") CALL HLHEX CF=0 HL ← hexadecimal number (e.g., HL=31A5 _H) CF=1 The contents of HL are not guaranteed.	All except AF and HL																												
CALL 2HEX (041F)	Converts a string of 2 ASCII characters into a hexadecimal number and loads it into the ACC. The call and return conditions are as follows. DE ← Starting address of the memory area which contains the ASCII character string. (e.g., "3" "A") CALL 2HEX CF=0 ACC ← hexadecimal number (e.g., ACC=3A _H) CF=1 The contents of the ACC are not guaranteed.	All except AF and DE																												
CALL ??KEY (09B3)	Blinks the cursor to prompt for key input. When a key is pressed, the corresponding display code is loaded into the ACC and control is returned to the calling program.	All except AF																												
CALL ?ADCN (0BB9)	Converts ASCII codes into display codes. The call and return conditions are as follows. ACC ← ASCII code CALL ?ADCN ACC ← Display code	All except AF																												
CALL ?DACN (0BCE)	Converts display codes into ASCII codes. The call and return conditions are as follows. ACC ← Display codes CALL ?DACN ACC ← ASCII code	All except AF																												
CALL ?BLNK (0DA6)	Detects the vertical blanking period. Control is returned to the calling program when the vertical blanking period is entered.	All registers																												
CALL ?DPCT (0DDC)	Controls display as follows. <table><tr><th>ACC</th><th>Control</th><th>ACC</th><th>Control</th></tr><tr><td>C0_H</td><td>Scrolling</td><td>C6_H</td><td>Same as the CLR key.</td></tr><tr><td>C1_H</td><td>Same as the ↓ key.</td><td>C7_H</td><td>Same as the DEL key.</td></tr><tr><td>C2_H</td><td>Same as the ↑ key.</td><td>C8_H</td><td>Same as the INST key.</td></tr><tr><td>C3_H</td><td>Same as the → key.</td><td>C9_H</td><td>Same as the</td></tr><tr><td>C4_H</td><td>Same as the ← key.</td><td></td><td>ALPHA key.</td></tr><tr><td>C5_H</td><td>Same as the HOME key.</td><td>CD_H</td><td>Same as the CR key.</td></tr></table>	ACC	Control	ACC	Control	C0 _H	Scrolling	C6 _H	Same as the CLR key.	C1 _H	Same as the ↓ key.	C7 _H	Same as the DEL key.	C2 _H	Same as the ↑ key.	C8 _H	Same as the INST key.	C3 _H	Same as the → key.	C9 _H	Same as the	C4 _H	Same as the ← key.		ALPHA key.	C5 _H	Same as the HOME key.	CD _H	Same as the CR key.	All registers
ACC	Control	ACC	Control																											
C0 _H	Scrolling	C6 _H	Same as the CLR key.																											
C1 _H	Same as the ↓ key.	C7 _H	Same as the DEL key.																											
C2 _H	Same as the ↑ key.	C8 _H	Same as the INST key.																											
C3 _H	Same as the → key.	C9 _H	Same as the																											
C4 _H	Same as the ← key.		ALPHA key.																											
C5 _H	Same as the HOME key.	CD _H	Same as the CR key.																											
CALL ?POINT (0FB1)	Loads the current cursor location into the HL register. The return conditions are as follows. CALL ?POINT HL ← Cursor location (binary)	All except AF and HL																												

MZ-800 monitor call

Functions of this monitor can be called using their function numbers in the same manner as function calls. In the following explanation, two-digit hexadecimal numbers printed in Gothic are the function numbers and the characters at their right are the function names.

The table below lists the main monitor variables related to the monitor calls.

Monitor variable	Address in hexadecimal	Length in bytes	Function
SYSSTA	004D	2	Hot start address of the utilities using this monitor.
ERRORP	004F	2	Address of the error handling routine of the utilities using this monitor
ELMD	1000	1	File mode 1: Object file 2: BASIC text file 3: Source file 4: Relocatable file
ELMD1	1001	17	File name (up to 16 characters) and end mark ODH.
ELMD20	1014	2	File size in bytes
ELMD22	1016	2	Load address
ELMD24	1018	2	Excrution address
ZLOG	1042	1	Logical number
ZRWX	1043	1	File open type 1: Read open 2: Write open
TEXTST	1070	2	Starting address of the text area of utilities using this monitor.
POOL	1072	2	Starting address of the work area of this monitor.
VARST	1074	2	Starting address of the variable area of utilities using this monitor.
TMPEND	107A	2	Ending address of the temporary area of utilities using this monitor.
TEMLMT	107E	2	Ending address of the memory area used by this monitor.
FILOUT	1091	1	Data is output to the CRT if the value at this address is zero and to the printer if it is 1. (The device specification is effective for monitor functions.& CR, .&IC, .&ICX and .&MSG)

00 .MONOP

Function: Returns to the RAM monitor.
Input registers: None
Output registers: None
Registers saved: None

01 .CR1

Function: Starts a new line independent of cursor position on a line
Input registers: None
Output registers: None
Registers saved: Primary registers only

02 .CR2

Function:	Starts a new line if the cursor is not at the beginning of a line.
Input registers:	None
Output registers:	None
Registers saved:	Primary register pairs only

03 .CRT1C

Function:	Outputs a character to the CRT. Control codes are executed.
Input register:	ACC: =Output data
Output registers:	None
Registers saved:	Primary register pairs only

04 .CRT1X

Function:	Outputs a character to the CRT. Control codes are displayed in reverse video.
Input register:	ACC: =Output data
Output registers:	None
Registers saved:	Primary register pairs only

05 .CRTMS

Function:	Outputs a character string. The end code is 00H. Control codes are executed. (Same as .CRT1C)
Input register:	DE: =Pointer position of the character string
Output registers:	None
Registers saved:	Primary register pairs only

06 .LPT0T

Function:	Outputs a character to the printer without code conversion.
Input register:	ACC: =Output data
Output registers:	None
Registers saved:	Primary register pairs only

07 .LPT1C

Function:	Outputs a character to the printer converting its code into that of the printer used with the MZ computer. The PRINT/P statement of BASIC uses this function.
Input register:	ACC: =Output data
Output registers:	None
Registers saved:	Primary register pairs only

08 .&CR

Function: Starts a new line on the CRT or printer according to the value of variable FILOUT(address 1091). Set the value of variable FILOUT to 1 to select the CRT or to 0 to select the printer in advance.

Input registers: None

Output registers: None

Registers saved: Primary register pairs only

09 .&1C

Function: Outputs a character to the CRT or printer according to the value of variable FILOUT. Control codes are executed when they are output to the CRT (same as .CRT1C). When this function outputs a character to the printer, its function is the same as .LPR1C.

Input register: ACC: =Output data

Output registers: None

Registers saved: Primary register pairs only

0A .&1CX

Function: Outputs a character to the CRT or printer according to the value of variable FILOUT. Control codes output to the CRT are displayed in reverse video (same as CRT1X). When the character is output to the printer, it is output in the same manner as with the PRINT/P statement of BASIC.

Input register: ACC: =Output data

Output registers: None

Registers saved: Primary register pairs only

0B .&MSG

Function: Outputs a character string to the CRT or printer. The end code is 00H. The switching condition between the CRT and printer is the same as that of the .&CRT. This function executes control codes when it outputs a character string to the CRT (same as .&CRT1C). When it outputs a character string to the printer, its function is the same as .LPT1C.

Input registers: DE: =Pointer position of the character string

Output registers: None

Registers saved: Primary register pairs only

0C .GETL

Function: Inputs one line of data from the keyboard and adds an end code 00H to the end of the data.

Input register: DE: =Starting address of the buffer in which the input data is stored.

Output register: CF: =1 when **SHIFT** + **BREAK** are depressed

Registers saved: Primary register pairs except AF

0D .INKEY

Function: Inputs a character from the keyboard.
Input registers: A: =0 Real time key scan (same as the GET statement of BASIC)
A: =1 Waits for key input blinking the cursor.
A: =FFH Unlike when A: =0, inputs only once if a key is depressed and held.
Output register: A: =MZ ASCII code
Registers saved: Primary register pairs except AF

0E .BREAK

Function: Detects **SHIFT** + **BREAK**.
Input register: None
Output registers: ZF: =1 when **SHIFT** + **BREAK** are pressed.
Registers saved: Primary register pairs except AF.

0F .HALT

Function: Waits for **SPACE** to be subsequently depressed if it is pressed. If **SHIFT** + **BREAK** are pressed next, this function transfers control to the address identified by .ERRORP.
Input registers: None
Output registers: None
Registers saved: Primary register pairs except AF

10 .DI

Function: Stops spooling or music, and inhibits interrupt.
Input registers: None
Output registers: None
Registers saved: Primary register pairs except AF

11 .EI

Function: Starts spooling or enables interrupt.
Input registers: None
Output registers: None
Registers saved: Primary register pairs except AF

17 .COUNT

Function: Counts the number of characters of the specified character string. The string must end with an end code 00H.
Input register: DE: =Pointer position of the character string
Output registers: ACC: =The length of the string
Registers saved: Primary register pairs except AF

1B .ERRX

Function: Displays an error message.
Input register: ACC: = Error code (the same as the error number listed in the error message table of the OWNER'S MANUAL). When the value of the 7-th bit of the accumulator is 1, the related device name is also displayed.
Output registers: None
Registers saved: Primary register pairs only

2C .DEVNM

Function: Interprets(specifies) the device name
Input registers: DE: = Pointer position of the character string which indicates the device name.
B: = The length of the string
Output registers: HL: = Pointer position next to the end of the device name which has been interpreted.
DE: = The starting address of the device table
ACC: = Device identification number (unit number)
Registers saved: None

2D .DEVFN

Function: Interprets the device name and file name.
Input register: DE: = Pointer position in the device name and file name string.
Output registers: None
Registers saved: None

2E .LUCHK

Function: Checks whether the logical number is defined.
Input registers: ACC: = Logical number
Output registers: ACC: = 1 (read opened)
ACC: = 2 (write opened)
ACC: = 3 (read/write opened)
CF: = 1 (not opened)
Registers saved: Primary registers except AF

2F .LOPEN

Function: Opens files which are not divided into blocks as object files. To execute this function, the device name and filename must be specified with function .DEVFN in advance.
Input registers: None
Output registers: None
Registers saved: None

30 .LOADF

Function: Loads files which are not divided into blocks such as object files. To execute this function, the file must have been opened with functions .DEVFN and .LOPEN in advance.

Input registers: HL: = Loading address

Output registers: None

Registers saved: None

31 .SAVEF

Function: Saves files which are not divided into blocks such as object files. To execute this function, the file name must be specified with function .DEVFN.

Input registers: DE: = Starting address of the memory area to be saved
ELMD20(1014H): = File size in bytes
ELMD22(1916H): = Load address
ELMD24(1018H): = Execution address

Output registers: None

Registers saved: None

32 .VRFYF

Function: Compare the contents of the specified memory area with a file which are not divided into blocks such as object files. To execute this function, the file must be opened with .DEVFN and .LOPEN in advance.

Input registers: None

Output registers: None

Registers saved: None

33 .RWOPN

Function: Opens to read or write a file which are divided into blocks such as source files (files in ASCII code). To execute this function, the device name and file name must be specified with function .DEVFN in advance.
ZRWX(1043H): = 1 for read-open
ZRWX: = 2 for write-open

Input registers: None

Output registers: None

Registers saved: Primary register pairs except AF

35 **.INMSG**

Function: Inputs one line of data of the file opened which has been opened with function call .RWOPN.

Input registers: DE: =Starting address of the input buffer

Output register: B: =Input file size in bytes

 CF: =1 when the file end (EOF) has been detected.

Registers saved: DE,HL

37 **.PRSTR**

Function: Writes the specified bytes of data (max. 255 characters) into the file which has been write opened with .RWOPN.

Input registers: DE: =Starting address of the data to be written.

 B: =Data size in bytes

Output registers: None

Registers saved: Primary register pairs except AF

38 **.CLKL**

Function: Closes or kills the files opened.

Input registers: ACC: =Logical number of the file to be closed or killed
 (When ACC: =0,all files opened are closed or killed.)

 B: =0 for kill and B < > 0 for close

Output registers: None

Registers saved: Primary register pairs except AF

39 **.DIR**

Function: Displays or prints out the information concerning files stored on the disk or the contents of the directory. The device name must be specified with monitor call .DEVNM in advance.

Input registers: ACC: =0 Inputs the contents of directory into the directory buffer in the monitor.

 ACC< >0 Outputs the directory in the directory buffer to the device specified with the value in the ACC.

 ACC: =88H To the CRT

 ACC: =89H To the printer

 Otherwise, the directory is output to the file or device specified by the logical number set.

Output registers: None

Registers saved: Primary register pairs except AF

3A .SETDF

Function: Sets the default device.
Input registers: DE: = Starting address of the device table
ACC: = Device identification number (unit number)
These are output registers set by .DEVNM.
Output registers: None
Registers saved: Primary register pairs only

3C .FINIT

Function: Initializes the I/O handler routine in the monitor (this function is used by the INIT statement of BASIC). The device name must be specified with monitor call .DEVNM in advance.
Input registers: None
Output registers: None
Registers saved: None

43 .ERCVR

Function: Performs the error recovery operation and stops the motor of the MZ disk or floppy disk.
Input registers: None
Output registers: None
Registers saved: None

Examples of use of monitor calls

In following examples, it is assumed that the SVC macro has been defined.

MACRO SVC

RST 3

DEFB @1 → A function number is assigned to parameter @1.

ENDM

When you create programs using this monitor, please add the program below to those programs at the top of them.

```
LD HL,ERADR          ; Sets the address of the error handling routine of the program.
LD(ERRORP),HL
LD HL,hot-start       ; Sets the hot start address of the program.
LD HL,last            ; Sets the last address of the program.
LD(TEXTST),HL
LD(POOL),HL
LD(HL),0
INC(HL),
INC HL
LD(VARST),HL
LD(TMPEND),HL
LD DE,6000            ; Sets 600H for a floppy disk and 800H for the MZ disk.
ADD HL,DE
LD(MEMLMT),HL
LD SP,HL              ; Sets HL to the initial value of the stack pointer.
```

Creates error handling routine ERADR mentioned above as follows

```
ERADR:               OR A
                     JR Z,break-adr; Jumps to SHIFT + BREAK handling routine.
                     CP 80H
                     JR Z,break-adr
                     SVC .ERR;Displays an error message.
```

Loading or verifying an object file

```
LD DE,FILE,X          ; Set name of object file to be loaded in DE.
SVC .COUNT           ; Set length of file name in B and returns.
SVC .DEVFN            ; Interpret (specify) device name and file name.
SVC .LOPEN            ; Open the file.
LD A,(ELDM)           ; Set file mode of file opened in ACC.
CP 1                  ; Object file?
JP NZ,error           ; Jump to error handling routine if not object file.
LD HL,(ELMD22)         ; Set load address in HL
SVC .LOADF(or SVC .VRFYF) ; Load or verify the file.
.
.
FILE. X: DEFM "QD: SAMPLE"
        DEFB0          ; Set the end of the file name to 0.
```

Saving an object file

```
LD DE,FILE.X          ; Set the name of the file to be saved in DE.
SVC.COUNT             ; Set the length of the file name in B and returns.
SVC.DEVFN             ; Interpret (specify) the device name and file name.
LD A,1
LD(ELMD),A            ; Set the file mode to the object file mode.
LD HL,length
LD(ELMD20),HL         ; Set the file length in bytes.
LD HL>Loading-adr
LD(ELMD22),HL         ; Set the load address of the file.
LD HL,exer-adr
LD(ELMD24),HL         ; Set the execution address of the file.
LD DE,save-adr
SVC.SAVEF             ; Save the file.
.
.
FILE. X: DEFM "QD: SAMPLE"
        DEFB0          ; Add 0 to end of filename.
```

Opening a source file (ASCII file)

```
LD A,open-mode           ; Read-open if the open mode is 1 and write-opens if it is 2.
LD(ZRWX),A
LD A,3
LD(ELMD),A               ; Set the file mode to the source file mode.
LD A,1
LD(ZLOG),A               ; Set the logical number to 1.
LD DE,FILE.X             ; Set the name of the file to be opened in DE.
SVC .COUNT              ; Sets the file length in bytes in B.
SVC .DEVFN               ; Interpret (specify) the device name and file name.
SVC .RWOPN               ; Open the file
LD A,(ELMD)              ; Set the file mode opened in A.
CP 3                     ; Source file?
JP NZ,error              ; Jump to the error handling routine if not source file.
:
:
FILE.X: DEFM "QD: SAMPLE"
      DEFB0               ; Set the end of the file name to 0.
                        ; Add 0 to end of filename.
```

Inputting one line of source file(ASCII file)

```
:                           ; The source file is assumed to be read-opened.
LD DE,buffer-adr          ; Set the starting address of the input buffer in DE.
SVC .INMSG                ; Input one line.
JP C,eof                  ; Perform the file end processing if CF: =
:                           ; Set the length of the line read in bytes in B.
:                           (OD at the end of the line is not included.)
```

Output of a source file (ASCII file)

```
:                           ; The source file is assumed to have been opened.
LD DE,save-adr            ; Starting address of the memory area to be saved.
LD B,length               ; File size in bytes (including OD at the end of each line).
SVC .PRSTR
:
```

Closing source files (ASCII files)

```
LD A,logical-number       ; Specified file only if the logical number is not zero and all files if it is
                           ; zero.
LD B,FFH
SVC .CLKL                 ; Closes the file(s).
```

Killing source file

LD A,logical-number ; Specified file only if the logical number is not zero and all files if it is
; zero.
LD B,0
SVC .CLKL ; Kill the file(s).

Setting a default device

LD DE,device-name ; Set the pointer position in the default device name in DE.
?SVC .COUNT ; Sets the length of the device name in B and returns.
SVC .DEVNM ; Interpret (specify) the device name.
SVC .SETDF ; Set the device specified by the value in ACC to the current device.

Display or Print out of directory

LD DE,device name ; Set the pointer position of the device name whose directory is to be
; displayed or printed out in DE.
SVC .COUNT ; Sets the length of the device name in B and returns.
SVC .DEVNM ; Interpret (specify) the device name.
LD B,A
XOR A ; Read the directory into the directory buffer in the monitor.
SVC .DIR
LD A,B
LD A,88H(or 89H)
SVC .DIR ; Output the directory to the CRT if the contents of ACC is 88H or to the
; printer if it is 89H.

Initialization of device (used by INIT statement of BASIC)

LD DE, device-name ; Set the pointer position for the device name to be initialized in DE.
SVC .COUNT ; Set the length of the device name in B and returns.
SVC .DEVNM ; Interpret (specify) the device name.
SVC .FINIT ; Initialize the device.

3-1 Monitor <1Z-013B>

** 780 ASSEMBLER 2Z-004C <1Z-013B> PAGE 01		03.06.85	** 780 ASSEMBLER 2Z-004C <1Z-013B> PAGE 02		03.06.85
01 0000	;		01 004A	START:	ENT
02 0000	;		02 004A	LD	IM
03 0000	;	MONITOR PROGRAM 1Z-013B	03 004A	31F010	CALL
04 0000	;		04 004D	ED56	CALL
05 0000	;	(MZ-800) FOR PAL	05 004F	CD3E07	CALL
06 0000	;		06 0052	CD320A	JR
07 0000	;	REV. 1984.07.03	07 0055	3019	CP
08 0000	;		08 0057	FE20	CP
09 0000	;		09 0059	2015	JR
10 0000	;		10 005B		JR
11 0000	C300E8		11 005B	D3E1	ENT
12 0003	C3E607		12 005D	11F0FF	OUT
13 0003	C3E607		13 0060	216B00	LD
14 0006	C30E09		14 0063	010500	LD
15 0006	C30E09		15 0066	ED00	LDIR
16 0009			16 0068	C3F0FF	JP
17 0009	C31809		17 006B		
18 000C			18 006B		
19 000C	C32009		19 006B	D3E0	ENT
20 000F			20 006D	C300	DEFW
21 000F	C32409		21 006F	00	DEFB
22 0012	C33509		22 0070		
23 0012	C33509		23 0070	06FF	ENT
24 0015			24 0070	21F110	LD
25 0015	C39308		25 0072	21F110	LD
26 0018			26 0075	CD080F	CALL
27 0018	C3A108		27 0078	3E16	LD
28 0018			28 007A	CD1200	CALL
29 001B	C3B008		29 007D	3E71	LD
30 001E	C3320A		30 007F	210008	LD
31 001E			31 0082	CD0509	CALL
32 0021	C33604		32 0085	218D03	LD
33 0021	C33604		33 0088	3EC3	LD
34 0024	C37504		34 008A	323810	LD
35 0024			35 008D	223910	LD
36 0027			36 0090	3E04	LD
37 0027	C30804		37 0092	329E11	LD
38 002A			38 0095	CD0E02	CALL
39 002A	C3F804		39 0098	CD0900	CALL
40 002D			40 009B	11E706	LD
41 002D	C38805		41 009E	DF	RST
42 0030			42 009F	CD7705	CALL
43 0030	C3C701		43 00A2		ENT
44 0033			44 00A2	3E01	LD
45 0033	C30803		45 00A4	329D11	LD
46 0036	00		46 00A7	2100E8	LD
47 0037	00		47 00AA	77	LD
48 0038	C33810		48 00AB	1855	JR
49 003B			49 00AD		ENT
50 003B	C35803		50 00AD	CD0900	CALL
51 003E			51 00B0	3E2A	LD
52 003E	C37705		52 00B2	CD1200	CALL
53 0041			53 00B5	11A311	LD
54 0041	C3E502		54 00B8	CD0300	CALL
55 0044			55 00BB	1A	LD
56 0044	C3A802		56 00BC	13	INC
57 0047			57 00BD	FE0D	CP
58 0047	C38E02		58 00BF	28EC	JR
59 004A			59 00C1	FE4A	CP
60 004A			60 00C3	282E	JR


```

01 01F1 218402
02 01F4 13
03 01F5 C01C02
04 01F8 3807
05 01FA C0C802
06 01FD 3815
07 01FF C0A802
08 0202 41
09 0203 18CC
10 0205 3E03
11 0207 32A011
12 020A 13
13 020B 18C4
14 020D 3E01
15 020F 18F6
16 0211 C0C802
17 0214 F5
18 0215 C0BE02
19 0218 F1
20 0219 C39806
21 021C
22 021C
23 021C
24 021C
25 021C
26 021C
27 021C
28 021C C5
29 021D 0608
30 021F 1A
31 0220 BE
32 0221 2809
33 0223 23
34 0224 23
35 0225 23
36 0226 10F8
37 0228 37
38 0229 13
39 022A C1
40 022B C9
41 022C 23
42 022D D5
43 022E 5E
44 022F 23
45 0230 56
46 0231 EB
47 0232 7C
48 0233 B7
49 0234 2809
50 0236 3AA011
51 0239 30
52 023A 2803
53 023C 29
54 023D 18FA
55 023F 22A111
56 0242 21A011
57 0245 3602
58 0247 2B
59 0248 D1
60 0249 13

```

```

LD INC HL,M#TBL
CALL ONPU
JR C,MLD1
CALL RYTHM
JR C,MLD5
CALL MLDST
LD B,C
JR MLD1
LD A,*3
LD (OCTV),A
LD INC DE
JR MLD1
LD A,1
LD MLD2+2
CALL RYTHM
PUSH AF
CALL MLDSP
POP AF
JP RET3
; ONPU TO RATIO CONV
; EXIT (RATIO)=RATIO VALUE
; C=ONTYO*TEMPO
ONPU: ENT BC
LD B,8
LD A,(DE)
LD (HL)
JP 2,ONP2
INC HL
INC HL
INC HL
INC HL
DJNZ -6
SCF
INC DE
POP B
RET
ONP2: INC HL
PUSH DE
LD E,(HL)
LD HL
LD D,(HL)
EX DE,HL
LD A,H
OR A
JR 2,+11
LD A,(OCTV)
DEC A
JR 2,+5
LD HL,HL
JR -4
LD (RATIO),HL
LD HL,OCTV
LD (HL),2
DEC HL
POP DE
INC

```

```

01 024A 1A
02 024B 47
03 024C E6F0
04 024E FE30
05 0250 2803
06 0252 7E
07 0253 1805
08 0255 13
09 0256 78
10 0257 E60F
11 0259 77
12 025A 219C02
13 025D 85
14 025E 6F
15 025F 4E
16 0260 3A9E11
17 0263 47
18 0264 AF
19 0265 81
20 0266 10FD
21 0268 C1
22 0269 4F
23 026A AF
24 026B C9
25 026C
26 026C
27 026C
28 026C 43
29 026D 4608
30 026F 44
31 0270 5F07
32 0272 45
33 0273 9106
34 0275 46
35 0276 3306
36 0278 47
37 0279 8605
38 027B 41
39 027C EC04
40 027E 42
41 027F 6404
42 0281 52
43 0282 0000
44 0284
45 0284 43
46 0285 CF07
47 0287 44
48 0288 F506
49 028A 45
50 028B 3306
51 028D 46
52 028E DA05
53 0290 47
54 0291 3705
55 0293 41
56 0294 A504
57 0296 42
58 0297 2304
59 0299 52
60 029A

```

```

LD A,(DE)
LD B,A
AND CP
JR 2,+5
LD A,(HL)
JR +7
INC DE
LD A,B
AND 0FH
LD (HL),A
LD HL,OPTBL
ADD A,L
LD L,A
LD C,(HL)
LD A,(TEMPJ)
LD B,A
XOR A
ADD DJNZ -1
POP BC
LD C,A
XOR A
RET
ONP3:
;
; MTBL:
ENT DEFB 43H
DEFW 0846H
DEFB 44H
DEFW 075FH
DEFB 45H
DEFW 0691H
DEFB 46H
DEFW 0633H
DEFB 47H
DEFW 0586H
DEFB 41H
DEFW 04ECH
DEFB 42H
DEFW 0464H
DEFB 52H
DEFW 0
MTBL: ENT
DEFB 43H
DEFW 07CFH
DEFB 44H
DEFW 06F5H
DEFB 45H
DEFW 0633H
DEFB 46H
DEFW 05DAH
DEFB 47H
DEFW 0537H
DEFB 41H
DEFW 04ASH
DEFB 42H
DEFW 0423H
DEFB 52H

```

```

; ONTYO ?
; HL=ONTYO
; HL=ONTYO

```

```

; C
; D
; E
; F
; G
; A
; B
; R
; #C
; #D
; #E
; #F
; #G
; #A
; #B
; #R

```

71

```

01 0308      ; TIME SET
02 0308      ;
03 0308      ; ACC=0 : AM
04 0308      ; =1 : PM
05 0308      ; DE=SEC: BINARY
06 0308      ; ?TMS1: ENT
07 0308      DI
08 0308      F3
09 0308      C5
10 0308      D5
11 0308      E5
12 0308      329B11
13 0308      3EF0
14 0311      329C11
15 0314      21C0A8
16 0317      AF
17 0318      ED52
18 031A      E5
19 031B      00
20 031C      EB
21 031D      2107E0
22 0320      3674
23 0322      36B0
24 0324      2B
25 0325      73
26 0326      72
27 0327      2B
28 0328      360A
29 032A      3600
30 032C      23
31 032D      23
32 032E      3680
33 0330      2B
34 0331      4E
35 0332      7E
36 0333      BA
37 0334      20FB
38 0336      79
39 0337      BB
40 0338      20F7
41 033A      2B
42 033B      00
43 033C      00
44 033D      00
45 033E      36FB
46 0340      363C
47 0342      23
48 0343      D1
49 0344      4E
50 0345      7E
51 0346      BA
52 0347      20FB
53 0349      79
54 034A      BB
55 034B      20F7
56 034D      E1
57 034E      D1
58 034F      C1
59 0350      FB
60 0351      C9

01 0352      ;
02 0352      ; BELL DATA
03 0352      ;
04 0352      ; ?BELD: ENT
05 0352      D7H
06 0352      DEFB
07 0353      4130
08 0355      00
09 0356      ;
10 0356      ;
11 0356      ; DEFS +2
12 0356      ; ORG 0358H
13 0358      ;
14 0358      ; TIME READ
15 0358      ;
16 0358      ; EXIT ACC=0 :AM
17 0358      ; =1 :PM
18 0358      ; DE=SEC. BINARY
19 0358      ;
20 0358      ; ?TMRD: ENT
21 0358      E5
22 0358      PUSH
23 0359      2107E0
24 035C      3680
25 035E      2B
26 035F      F3
27 0360      SE
28 0361      56
29 0362      FB
30 0363      7B
31 0364      B2
32 0365      280E
33 0367      AF
34 0368      21C0A8
35 036B      ED52
36 036D      3810
37 036F      EB
38 0370      3A9B11
39 0373      E1
40 0374      C9
41 0375      11C0A8
42 0378      3A9B11
43 037B      EE01
44 037D      E1
45 037E      C9
46 037F      F3
47 0380      2106E0
48 0383      7E
49 0384      2F
50 0385      5F
51 0386      7E
52 0387      2F
53 0388      57
54 0389      FB
55 038A      13
56 038B      18EB
57 038D      ;
58 038D      ; TIME INTERRUPT
59 038D      ; TIMIN: ENT
60 038D      RET

; AMPM DATA
; TIME FLAG
; 12H
; COUNT DATA = 12H-IN DA

; CONT2
; CONT1
; CONTF
; CONT2
; 1SEC

; TMS1:
; TMS2:

```

```

01 0380 F5      PUSH AF
02 038F C5      PUSH BC
03 038F D5      PUSH DE
04 0390 E5      PUSH HL
05 0391 219B11 LD HL,AMPM
06 0394 7E      LD A,(HL)
07 0395 EE01    XOR I
08 0397 77      LD (HL),A
09 0398 2107E0 LD HL,CONTF
10 0398 3680    LD (HL),80H
11 039D 2B      DEC HL
12 039E E5      PUSH HL
13 039F 5E      LD E,(HL)
14 03A0 56      LD D,(HL)
15 03A1 21C0A8 LD HL,ASC0H
16 03A4 19      ADD HL,DE
17 03A5 2B      DEC HL
18 03A6 2B      DEC HL
19 03A7 EB      EX DE,HL
20 03A8 E1      POP HL
21 03A9 73      LD (HL),E
22 03AA 72      LD (HL),D
23 03AB E1      POP HL
24 03AC 01      POP DE
25 03AD C1      POP BC
26 03AE F1      POP AF
27 03AF FB      EI
28 03B0 C9      RET

; SPACE PRINT AND DISP ACC
;
; INPUT:HL=DISP. ADR.
;
SPHGX: ENT ?PRTS
        LD A,(HL)
        CALL PRTHX
        LD A,(HL)
        RET

;
;
; ORG 03BAH
; (ASCII PRINT) FOR HL
PRTHL: ENT
        LD A,H
        CALL PRTHX
        LD A,L
        JR PRTHX

;
; ORG 03C1
; ORG 03C3H;PRTHX
; (ASCII PRINT) FOR ACC
PRTHX: ENT
        PUSH AF
        RRCA
        ORG 03C4 0F
01 03C5 0F      RRCA
02 03C6 0F      RRCA
03 03C7 0F      RRCA
04 03C8 C0DA03 CALL ASC
05 03CB C01200 CALL PRNT
06 03CE F1      POP AF
07 03CF C0DA03 CALL ASC
08 03D2 C31200 JP PRNT
;
;
; 80 CHA. 1 LINE CODE (DATA)
;
SLPT: ENT
        DEFB 01H
        DEFB 09H
        DEFB 09H
        DEFB 09H
        DEFB 00H
; ORG 03DAH;ASC
;
; HEXADESIMAL TO ASCII
; IN: ACC (D3-D0)=HEXADESIMAL
; EXIT: ACC = ASCII
ASC: ENT
        AND 0FH
        CP 0AH
        JR C,NOADD
        ADD A,7
NOADD: ENT
        ADD A,30H
        RET

; ASCII TO HEXADESIMAL
; IN: ACC = ASCII
; EXIT: ACC = HEXADESIMAL
; CY = 1 ERROR
HEXJ: ENT
        SUB 30H
        RET C
        CP 0AH
        CCF
        RET NC
        SUB 7
        CP 10H
        CCF
        RET C
        CP 0AH
        RET
;
;
; ORG 03F9H;HEX
; HEX: ENT
        JR HEXJ
;

```



```

01 050E      ; IN BC=SIZE
02 050E      ; DE=LOAD ADR.
03 050E      ;
04 050E      ; EXIT ACC=0 : OK CF=0
05 050E      ; =1 : ER =1
06 050E      ; =2 : BREAK=1
07 050E      ;
08 050E      ; RTAPE:
09 050E      ;   PUSH DE
10 050F      ;   PUSH BC
11 0510      ;   PUSH HL
12 0511      ;   LD H,2
13 0513      ; RTPI:
14 0513      ;   ENT BC,KEYPB
15 0516      ;   LD DE,CSTR
16 0519      ; RTP2:
17 0519      ;   CALL
18 051C      ;   JR C,RTP6
19 051E      ;   CALL
20 0521      ;   LD A,(DE)
21 0522      ;   AND 20H
22 0524      ;   JP Z,RTP2
23 0527      ;   LD D,H
24 0528      ;   LD HL,0
25 052B      ;   LD HL,(SUMDT),HL
26 052E      ;   POP HL
27 052F      ;   POP BC
28 0530      ;   PUSH BC
29 0531      ;   PUSH HL
30 0532      ; RTP3:
31 0532      ;   CALL
32 0535      ;   JR C,RTP6
33 0537      ;   LD (HL),A
34 0538      ;   INC HL
35 0539      ;   DEC BC
36 053A      ;   LD A,B
37 053B      ;   OR C
38 053C      ;   JR NZ,RTP3
39 053E      ;   HL,(SUMDT)
40 0541      ;   CALL
41 0544      ;   JR C,RTP6
42 0546      ;   LD E,A
43 0547      ;   CALL
44 054A      ;   JR C,RTP6
45 054C      ;   CP L
46 054D      ;   JR NZ,RTP5
47 054F      ;   LD A,E
48 0550      ;   CP H
49 0551      ;   JR NZ,RTP5
50 0553      ; RTP8:
51 0553      ;   XOR A
52 0554      ; RTP4:
53 0554      ;   ENT
54 0554      ;   POP POP
55 0555      ;   POP POP
56 0556      ;   POP POP
57 0557      ;   CALL MSTOP
58 055A      ;   PUSH AF
59 055B      ;   LD A,(TIMFG)
60 055E      ;   CP F0H

01 0560      ; NZ,+3
02 0562      ; JR EI
03 0563      ; POP AF
04 0564      ; RET
05 0565      ;
06 0565      ; RTPS:
07 0565      ;   ENT
08 0566      ;   DEC
09 0568      ;   JR Z,RTP7
10 0569      ;   LD H,D
11 056C      ;   CALL GAPCK
12 056E      ;   JR RTP1
13 056E      ; RTPT:
14 056E      ;   ENT
15 0570      ;   LD A,1
16 0572      ;   JR RTP9
17 0574      ; RTPT6:
18 0574      ;   ENT
19 0575      ;   LD A,2
20 0577      ;   JR RTP4
21 0577      ;
22 0577      ;   BELL
23 0577      ;
24 0577      ;   ?BEL:
25 0577      ;   ENT
26 0578      ;   PUSH DE,?BELD
27 057B      ;   LD RST 6
28 057C      ;   POP DE
29 057D      ;   RET
30 057E      ;
31 057E      ;   FLASING AND KEYIN
32 057E      ;   EXIT:ACC INPUT KEY DATA(DSP.CODE)
33 057E      ;   H=F0H THEN NO KEYIN(Z FLG.)
34 057E      ;
35 057E      ;   FLKEY:
36 057E      ;   ENT
37 0581      ;   CALL ?FLAS
38 0584      ;   CALL ?KEY
39 0586      ;   CP F0H
40 0587      ;   RET
41 0587      ;
42 0587      ;
43 0587      ;
44 0587      ;
45 0587      ;   DEFS +1
46 0588      ;   ORG 0588H
47 0588      ;
48 0588      ;   VERIFY (FROM $CMT)
49 0588      ;
50 0588      ;   EXIT ACC =0 : OK CF=0
51 0588      ;   =1 : ER CF=1
52 0588      ;   =2 : BREAK CF=1
53 0588      ;
54 0588      ;   ?VRFY:
55 0588      ;   ENT
56 0588      ;   DI
57 0589      ;   PUSH DE
58 058A      ;   BC
59 058B      ;   PUSH HL
60 058C      ;   LD BC,(SIZE)

```

```

** Z80 ASSEMBLER 27-004C <17-013B> PAGE 21                                03.06.85
01 0590 2A0411      HL,(DTADR)
02 0593 16D2        LD D,02H
03 0595 1E53        LD E,53H
04 0597 78         LD A,B
05 0598 B1         OR C
06 0599 28B9       JR Z,RTP4
07 059B CD1A07     CALL CKSUM
08 059E CD9F06     CALL MOTOR
09 05A1 38CF       JR C,RTP6
10 05A3 C05B06     CALL TMRK
11 05A6 38CA       JR C,RTP6
12 05A8 CDAD05     CALL TURFY
13 05AB 18A7       JR RTP4
14 05AD
15 05AD
16 05AD
17 05AD
18 05AD
19 05AD
20 05AD
21 05AD
22 05AD
23 05AD
24 05AD
25 05AD
26 05AD
27 05AD 05        TURFY: ENT PUSH DE
28 05AE C5        PUSH BC
29 05AF E5        PUSH HL
30 05B0 2602      LD H,2
31 05B2           ENT
32 05B2 0101E0    LD BC,KEYPB
33 05B5 1102E0    LD DE,CSTR
34 05B8           ENT
35 05B8 CD0106    CALL EDGE
36 05BB DA7205    JP C,RTP6
37 05BE CD4A0A    CALL DLY3
38 05C1 1A        LD A,(DE)
39 05C2 E620      AND 20H
40 05C4 C8B805    JP Z,TVF2
41 05C7 54        LD D,H
42 05C8 E1        POP HL
43 05C9 C1        POP BC
44 05CA C5        PUSH BC
45 05CB E5        PUSH HL
46 05CC           ENT
47 05CC CD2406    CALL RBYTE
48 05CF 38A1      JR C,RTP6
49 05D1 BE        CP (HL)
50 05D2 209A      JR NZ,RTP7
51 05D4 23        INC HL
52 05D5 0B        DEC BC
53 05D6 78        LD A,B
54 05D7 B1        OR C
55 05D8 20F2      JR NZ,TVF3
56 05DA 2A9911    LD HL,(CSMOT)
57 05DD CD2406    CALL RBYTE
58 05E0 BC        CP H
59 05E1 20B8      JR NZ,RTP7
60 05E3 CD2406    CALL RBYTE

** Z80 ASSEMBLER 27-004C <17-013B> PAGE 22                                03.06.85
01 05E6 8D        CP L
02 05E7 2085      JR NZ,RTP7
03 05E9 15        DEC D
04 05EA CA5305    JP Z,RTP8
05 05ED 62        LD H,D
06 05EE 18C2      JR TVF1
07 05F0           ; FLASHING DATA LOAD
08 05F0           ;
09 05F0           ; ?LOAD: ENT
10 05F0 F5        PUSH AF
11 05F0 F5        LD A,(FLASH)
12 05F1 3A8E11    CALL ?PONT
13 05F4 CDB10F    LD (HL),A
14 05F7 77        POP AF
15 05F8 F1        RET
16 05F9 C9
17 05FA
18 05FA
19 05FA
20 05FA
21 05FA
22 05FA CD0900    NLPHL: ENT NL
23 05FD CDBA03    CALL PRTHL
24 0600 C9        RET
25 0601
26 0601
27 0601           ; ORG 0601H;EDGE
28 0601
29 0601
30 0601           ; EDGE (TAPE DATA EDGE DETECT)
31 0601
32 0601           ; BC=KEYPB ($E001)
33 0601           ; DE=CSTR ($E002)
34 0601           ; EXIT CF=0 OK ; CF=1 BREAK
35 0601
36 0601           EDGE: ENT
37 0601 3EF8      LD A,F8H
38 0603 3200E0    LD (KEYPA),A
39 0606 00        NOP
40 0607           ENT
41 0607 0A        LD A,(BC)
42 0608 E681      AND 81H
43 060A 2002      JR NZ,+4
44 060C 37        SCF
45 060D C9        RET
46 060E 1A        LD A,(DE)
47 060F E620      AND 20H
48 0611 20F4      JR NZ,EDG1
49 0613           ENT
50 0613 0A        LD A,(BC)
51 0614 E681      AND 81H
52 0616 2002      JR NZ,+4
53 0618 37        SCF
54 0619 C9        RET
55 061A 1A        LD A,(DE)
56 061B E620      AND 20H
57 061D 28F4      JR Z,EDG2
58 061F C9        RET
59 0620
60 0620           ; CSTR D5 = 1 :10/14

```



```

01 070A 280B      JR     ENT      Z,MST3
02 070C 3E06      LD     LD      A,06H
03 070E 3203E0    LD     LD      (CSTPT),A
04 0711 3C        INC     INC     A
05 0712 3203E0    LD     LD      (CSTPT),A
06 0715 10EE      DJNZ    DJNZ    MST1
07 0717           ENT     ENT     ?RSTR1
08 0717           JP      JP
09 0717 C3E60E    ;
10 071A           ;
11 071A           ;
12 071A           ;
13 071A           ;
14 071A           ;
15 071A           ;
16 071A           ;
17 071A           ;
18 071A           ;
19 071A           ;
20 071A           ;
21 071A           ;
22 071A C5        ;
23 071B D5        ;
24 071C E5        ;
25 071D 110000    ;
26 0720           ;
27 0720 78        ;
28 0721 B1        ;
29 0722 200B      ;
30 0724 EB        ;
31 0725 229711    ;
32 0728 229911    ;
33 072B E1        ;
34 072C D1        ;
35 072D C1        ;
36 072E C9        ;
37 072F           ;
38 072F 7E        ;
39 0730 C5        ;
40 0731 0608      ;
41 0733           ;
42 0733 07        ;
43 0734 3001      ;
44 0736 13        ;
45 0737 10FA      ;
46 0739 C1        ;
47 073A 23        ;
48 073B 0B        ;
49 073C 18E2      ;
50 073E           ;
51 073E           ;
52 073E           ;
53 073E           ;
54 073E 2103E0    ;
55 0741 368A      ;
56 0743 3607      ;
57 0745 3605      ;
58 0747           ;
59 0747           ;
60 0747 C9        ;

```

```

; 10001010
; PC3=1
; PC2=1

```

```

01 06B4 10F7      DJNZ    -7
02 06B6 AF        XOR     A
03 06B7           MOT7:   RET3
04 06B7 18E2      JR      RET3
05 06B9           MOT4:   LD     A,06H
06 06B9 3E06      LD     LD     HL,CSTPT
07 06B8 2103E0    LD     LD     (HL),A
08 06BE 77        INC     INC     A
09 06BF 3C        LD     LD     (HL),A
10 06C0 77        LD     LD     MOT1
11 06C1 10E1      CALL    CALL    NL
12 06C3 C00900    LD     LD     A,D
13 06C6 7A        CP      CP      07H
14 06C7 FED7      JR      Z,MOT8
15 06C9 2805      LD     LD     DE,MSG#1
16 06CB 11FB03    JR      MOT9
17 06CE 1807      ENT     ENT     DE,MSG#3
18 06D0           RST     RST     3
19 06D0 110204    LD     LD     DE,MSG#2
20 06D3 DF        LD     LD     3
21 06D4 11FD03    ENT     ENT     A,(CSTR)
22 06D7           RST     RST     10H
23 06D7 DF        LD     LD     AND
24 06D8           JR      JR      NZ,MOT2
25 06D8 3A02E0    CALL    CALL    ?BRK
26 06DB E610      JR      NZ,MOT5
27 06DD 20CC      SCF     SCF
28 06DF CD320A    JR      MOT7
29 06E2 20F4      ;
30 06E4 37        ;
31 06E5 1800      ;
32 06E7           ;
33 06E7           ;
34 06E7           ;
35 06E7           ;
36 06E7 2A2A2020  ;
37 06EB 404F4E49  ;
38 06EF 544F5220  ;
39 06F3 315A2D30  ;
40 06F7 31334220  ;
41 06FB 202A2A    ;
42 06FE 0D        ;
43 06FF           ;
44 06FF           ;
45 06FF           ;
46 0700           ;
47 0700           ;
48 0700           ;
49 0700           ;
50 0700           ;
51 0700           ;
52 0700           ;
53 0700 F5        ;
54 0700 C5        ;
55 0701 C5        ;
56 0702 D5        ;
57 0703 060A      ;
58 0705           ;
59 0705 3A02E0    ;
60 0708 E610      ;

```



```

** 780 ASSEMBLER 22-004C <17-013B> PAGE 29 03.06.85
01 07E6 ; (END =CR )
02 07E6 ;
03 07E6 ;
04 07E6 ?GETL: ENT AF
05 07E6 F5 PUSH BC
06 07E7 C5 PUSH HL
07 07E8 E5 PUSH DE
08 07E9 D5 ENT
09 07EA CDB309 GETL1: CALL ??KEY
10 07EB CDB309 AUT03: ENT
11 07ED F5 PUSH AF
12 07EE 47 LD B,A
13 07EF 3A9D11 LD A,(SWRK)
14 07F0 0F RRCA
15 07F2 0F CALL
16 07F3 D47705 LD NC,?BEL
17 07F6 78 LD A,B,KANAF
18 07F7 217011 LD HL,KANAF
19 07FA E6F0 AND F0H
20 07FC FEC0 CP
21 07FE D1 POP
22 07FF 78 LD A,B
23 0800 2016 JR NZ,GETL2
24 0802 FEC0 CP CDH
25 0804 2855 JR Z,GETL3
26 0806 FECB CP CBH
27 0808 CA2208 JP Z,GETLC
28 080B FECF CP CFH
29 080D 2809 JR Z,GETL2
30 080F FEC7 CP C7H
31 0811 300A JR NC,GETL5
32 0813 CB1B RR E
33 0815 78 LD A,B
34 0816 3005 JR NC,GETL5
35 0818 CDB50D GETL2: ENT
36 081B 18CD JR
37 081D CDB50D GETL5: ENT
38 081D CDDC0D CALL
39 0820 18C8 JR
40 0822 ; BREAK IN
41 0822 ;
42 0822 ;
43 0822 ;
44 0822 E1 GETLC: POP HL
45 0823 E5 PUSH HL
46 0824 361B LD (HL),1BH
47 0826 23 INC HL
48 0827 360D LD (HL),0DH
49 0829 1853 JR GETLR
50 082B ; GETLA
51 082B 0F ;
52 082B 0F GETLA: RRCA
53 082C 3037 JR
54 082E 1833 JR
55 0830 ;
56 0830 ;
57 0830 ; DELAY 7M SEC AND SUEP
58 0830 ;
59 0830 ; DSUEP: CALL DLY12
60 0830 CDB609 ;

** 780 ASSEMBLER 22-004C <17-013B> PAGE 30 03.06.85
01 0833 CD500A CALL ?SUEP
02 0836 C9 RET
03 0837 ;
04 0837 ;
05 0837 ; DEFS 36
06 085B ;
07 085B ;
08 085B ;
09 085B ; ORG 085BH;GETL3
10 085B ;
11 085B CDF302 GETL3: CALL .MANG
12 085E 0628 LD B,40
13 0860 30C9 JR NC,GETLA
14 0862 25 DEC H
15 0863 0650 GETLB: LD B,80
16 0865 2E00 GETL6: LD L,0
17 0867 CDB40F CALL ?PNT1
18 086A 01 POP DE
19 086B D5 PUSH DE
20 086C 7E LD A,(HL)
21 086D CDE08 GETLZ: CALL ?DACC
22 0870 12 LD (DE),A
23 0871 23 INC HL
24 0872 13 INC DE
25 0873 10F7 DJNZ GETLZ
26 0875 EB DE,HL
27 0876 360D GETLU: LD (HL),0DH
28 0878 2B HL
29 0879 7E LD A,(HL)
30 087A FE20 CP 20H
31 087C ;
32 087C ; CR AND NEW LINE
33 087C ;
34 087C ; JR Z,GETLU
35 087C 28F8 ;
36 087E ; NEW LINE RETURN
37 087E ;
38 087E CDE09 GETLR: CALL ?LTNL
39 087E 01 POP DE
40 0881 01 POP HL
41 0882 E1 POP BC
42 0883 C1 POP AF
43 0884 F1 RET
44 0885 C9 ;
45 0886 ;
46 0886 ;
47 0886 ;
48 0886 ; DEFS +13
49 0893 ; ORG 0893H
50 0893 ; MESSAGE PRINT
51 0893 ;
52 0893 ; DE PRINT DATA LOW ADR.
53 0893 ; END=CR
54 0893 ;
55 0893 ; ?MSG: ENT
56 0893 F5 PUSH AF
57 0893 F5 PUSH BC
58 0894 C5 PUSH DE
59 0895 D5 MSG1: LD A,(DE)
60 0896 1A ;

```

```

** 780 ASSEMBLER 22-004C <12-013B> PAGE 31 03.06.85
01 0897 FE0D CP 0DH 00H
02 0899 280C JR Z,MSGX2
03 089B C03509 CALL ?PRNT
04 089E 13 INC DE
05 089F 18F5 JR MSG1
;
;
;ORG 08A1H
;
; ALL PRINT MESSAGE
;MSGX: ENT
13 08A1 F5 PUSH AF
14 08A2 C5 PUSH BC
15 08A3 D5 PUSH DE
16 08A4 1A LD A,(DE)
17 08A5 FE0D CP 0DH
18 08A7 CAE60E JP Z,?RSTR1
19 08AA CDB90B CALL ?ADCN
20 08AD C06C09 PRNT3
21 08B0 13 INC DE
22 08B1 18F1 INC DE
;
; TOP OF KEYTBLS
;
;MSGM: LD DE,KTBL
24 08B3 JR ?KY5
;
; BREAK CODE IN
;
;BRK: LD A,CBH
31 08B8 3ECB OR A
32 08BA B7 JR ?KY1
33 08BB 1819
34 08BD
35 08BD
36 08BD
37 08BD
38 08BD
39 08BD
40 08BD
41 08BD
42 08BD
43 08BD
44 08BD
45 08BD CDCAD8
46 08C0 D6F0
47 08C2 C8 RET Z
48 08C3 C6F0 ADD A,F0H
49 08C5 C3CE0B JP ?DACN
;
;
; DEFS +2
;
;
;
;ORG 08CAH;?KEY
;
; 1KEY INPUT
; IN B = KEY MODE(SHIFT,CTRL,BREAK)
60 08CA

** 780 ASSEMBLER 22-004C <12-013B> PAGE 32 03.06.85
01 08CA C = KEY DATA (COLUMN & ROW)
02 08CA ACC=DISPLAY CODE
03 08CA IF NO KEY, ACC=F0H
04 08CA IF CY=1 THEN ATTRIBUTE ON (SMALL,HIRAKANA)
05 08CA
06 08CA
07 08CA ?KEY: ENT
08 08CA C5 PUSH BC
09 08CB D5 PUSH DE
10 08CC E5 PUSH HL
11 08CD CD3008 CALL DSUEP
12 08D0 78 LD A,B
13 08D1 07 RLCA
14 08D2 3806 JR C,?KY2
15 08D4 3EF0 LD A,F0H
16 08D6 ENT
17 08D7 E1 POP HL
18 08D7 D1 POP DE
19 08D8 C1 POP BC
20 08D9 C9 RET
;
; ?KY1:
21 08DA
22 08DA ENT
23 08DA 11EA0B LD DE,KTBL
24 08DD 78 LD A,B
25 08DE FE88 CP 88H
26 08E0 28D6 JR Z,#BRK
27 08E2 2600 LD H,0
28 08E4 69 LD L,C
29 08E5 CB6F BIT 5,A
30 08E7 200E JR NZ,?KY5-3
31 08E9 3A7011 LD A,(KANAF)
32 08EC 0F RRCA
33 08ED DAFE08 JP C,?KYGRP
34 08F0 78 LD A,B
35 08F1 17 RLA
36 08F2 17 RLA
37 08F3 38BE JR C,?KYSM
38 08F5 1803 JR ?KY5
39 08F7 11AA0C LD DE,KTBL
40 08FA ENT
41 08FA 19 ADD HL,DE
42 08FB 7E ENT
43 08FB 7E A,(HL)
44 08FC 18D8 JR ?KY1
45 08FE
46 08FE CB70 BIT 6,B
47 0900 2807 JR Z,?KYGRS
48 0902 11E90C LD DE,KTBLG
49 0905 19 ADD HL,DE
50 0906 37 SCF
51 0907 18F2 JR ?KY55
52 0909
53 0909 116A0C LD DE,KTBLGS
54 090C 18EC JR ?KY5
55 090E
56 090E
57 090E
58 090E
59 090E
60 090E ;ORG 090EH

```

```

01 090E      ; NEWLINE
02 090E      ;
03 090E      ?LTNL: ENT
04 090E      LD
05 090E      XOR
06 090F 329411 A
07 0912 3ECD   LD (DPRNT),A
08 0914 1843   LD A,CDH
09 0916      JR PRNT5
10 0918      DEFS +2
11 0918      ;ORG 0918H
12 0918      ?NL: ENT
13 0918 3A9411 LD A,(DPRNT)
14 091B B7     OR A
15 091C C8     RET Z
16 091D 18EF   JR ?LTNL
17 091F      DEFS +1
18 0920      ;ORG 0920H
19 0920      ;
20 0920      ; PRINT SPACE
21 0920      ;
22 0920      ?PRTS: ENT
23 0920 3E20   LD A,20H
24 0922 1811   JR ?PRT
25 0924      ;
26 0924      ; PRINT TAB
27 0924      ;
28 0924      ?PRTT: ENT
29 0924 C0C000 CALL PRNTS
30 0927 3A9411 LD A,(DPRNT)
31 092A B7     OR A
32 092B C8     RET Z
33 092C D60A   SUB +10
34 092E 38F4   JR C,-10
35 0930 20FA   JR NZ,-4
36 0932      DEFS +3
37 0935      ;ORG 0935H
38 0935      ;
39 0935      ; PRINT
40 0935      ;
41 0935      ; IN ACC = PRINT DATA (ASCII)
42 0935      ;
43 0935      ?PRT: ENT
44 0935 FE0D   CP 0DH
45 0937 28D5   JR Z,?LTNL
46 0939 C5     PUSH BC
47 093A 4F     LD C,A
48 093B 47     LD B,A
49 093C C04609 CALL ?PRT
50 093F 78     LD A,B
51 0940 C1     POP BC
52 0941 C9     RET
53 0942      ;
54 0942      ; MSGOK: ENT
55 0942      DEFS +1
56 0942 4F4B21 DEFH 'OK!'
57 0945 00     DEFB 0DH
58 0946      ;ORG 0946H
59 0946      ;
60 0946      ; PRINT ROUTINE

```

```

01 0946      ; 1 CHA.
02 0946      ; INPUT:C=ASCII DATA (?DSP+?DPCT)
03 0946      ;
04 0946      ?PRT: ENT
05 0946 79     LD A,C
06 0947 C0B90B CALL ?ADCN
07 094A 4F     LD C,A
08 094B FEF0   CP F0H
09 094D C8     RET Z
10 094E E6F0   AND F0H
11 0950 FEC0   CP C0H
12 0952 79     LD A,C
13 0953 2017   JR NZ,PRNT3
14 0955 FEC7   CP C7H
15 0957 3013   JR NC,PRNT3
16 0959      PRNTS: ENT
17 0959 C0C000 CALL ?DPCT
18 095C FEC3   CP C3H
19 095E 280F   JR Z,PRNT4
20 0960 FEC5   CP C5H
21 0962 2803   JR Z,PRNT2
22 0964 FEC6   CP C6H
23 0966 C0     RET NZ
24 0967 AF     PRNT2: XOR A
25 0968 329411 LD (DPRNT),A
26 096B C9     RET
27 096C      PRNT3: ENT
28 096C C0B50D CALL ?DSP
29 096F 3A9411 LD A,(DPRNT)
30 0972 3C     INC A
31 0973 FE50   CP +80
32 0975 38F1   JR C,PRNT2+1
33 0977 D650   SUB +80
34 0979 18ED   JR PRNT2+1
35 097B      ;
36 097B      ;
37 097B      ;
38 097B      ;
39 097B      ; FCLASSING BYPASS 1
40 097B      ;
41 097B      ;
42 097B      FLAS1: ENT
43 097B 3A8E11 LD A,(FLASH)
44 097E 186F   JR FLAS2
45 0980      ;
46 0980      ; BREAK SUBROUTINE BYPASS 1
47 0980      ;
48 0980      ;
49 0980      ?BRK2: ENT
50 0980 C86F   S,A
51 0980 0B6F   BIT S,A
52 0982 2802   JR Z,?BRK3
53 0984 B7     OR A
54 0985 C9     RET
55 0986      ;
56 0986 3E20   ?BRK3: LD A,20H
57 0988 B7     OR A
58 0989 37     SCF
59 098A C9     RET
60 098B      ;

```

```

** Z80 ASSEMBLER 27-004C <17-013B> PAGE 35          03.06.85
01 098B MSGSV: ENT      'FILENAME?',
02 098B 46494C45  DEFN
03 098F 4E414D45
04 0993 3F20      DEFB 0DH
05 0995 0D      ; DLY 7 MSEC
06 0996      ;
07 0996      ; DLY12: ENT
08 0996      ;   LD
09 0996 C5      ;   B,15H
10 0996      ;   CALL DLY3
11 0997 0615      ;   DJNZ -3
12 0999 CD4A0A      ;   POP BC
13 099C 10FB      ;   RET
14 099E C1
15 099F C9
16 09A0
17 09A0
18 09A0
19 09A0
20 09A0
21 09A0 4C4F4144 MSG?2: ENT      'LOADING',
22 09A0 494E4720  DEFN
23 09A4 0D      DEFB 0DH
24 09A8 0D
25 09A9
26 09A9
27 09A9
28 09A9
29 09A9
30 09A9 DLY4: ENT
31 09A9 3E73      LD A,73H
32 09AB 3D      DEC A
33 09AC C2AB09      JP NZ,-1
34 09AF C9      RET
35 09B0
36 09B0
37 09B0
38 09B3
39 09B3
40 09B3
41 09B3
42 09B3
43 09B3
44 09B3
45 09B3
46 09B3
47 09B3
48 09B3
49 09B3
50 09B3 E5      PUSH HL
51 09B4 CD920B      CALL ?SAVE
52 09B7      ENT
53 09B7 C07E05      CALL FLKEY
54 09BA 20FB      JR NZ,KSL1
55 09BC      ENT
56 09BC C07E05      CALL FLKEY
57 09BF 28FB      JR Z,KSL2
58 09C1 67      LD H,A
59 09C2 CD9C09      CALL DLY12
60 09C5 CDCA08      CALL ?KEY

; 7143.9 micro sec
; 458.81 micro sec

; KEY
; KEY IN THEN JUMP
; NOT KEY IN THEN JUMP
; DELAY CHATTER
; DELAY CHATTER

** Z80 ASSEMBLER 27-004C <17-013B> PAGE 36          03.06.85
01 09C8 F5      PUSH AF
02 09C9 BC      CP H
03 09CA E1      POP HL
04 09CB 20EF      JR NZ,KSL2
05 09CD E5      PUSH HL
06 09CE F1      POP AF
07 09CF CDF005      CALL ?LOAD
08 09D2 E1      POP HL
09 09D3 C9      RET
10 09D4
11 09D4
12 09D4
13 09D4
14 09D4
15 09D4 AF      ; CLEAR 2
16 09D5 010008      ; #CLR08: ENT
17 09D8 05      ; #CLR8: ENT
18 09D8 57      ; CLEAR: ENT
19 09D8 05      ; CLEAR1: ENT
20 09D9 57      ; (HL),D
21 09DA 72      LD HL
22 09DA 23      INC HL
23 09DB 08      DEC BC
24 09DC 78      LD A,B
25 09DD B1      OR C
26 09DE 20F9      JR NZ,CLEAR1
27 09E1 01      POP DE
28 09E2 C9      RET
29 09E3
30 09E3
31 09E3
32 09E3
33 09E3
34 09E3
35 09E3
36 09E3
37 09E3 F5      ENT
38 09E4 E5      PUSH AF
39 09E5 3A02E0      LD HL
40 09E8 07      RLCA
41 09E9 07      RLCA
42 09EA 388F      JR C,FLAS1
43 09EC 3A9211      LD A,(FLSDT)
44 09EF      ENT
45 09EF CDB10F      CALL ?PONT
46 09F2 77      LD (HL),A
47 09F3
48 09F3 E1      ENT
49 09F4 F1      POP HL
50 09F5 C9      POP AF
51 09F6      RET
52 09F6
53 09F6
54 09F6
55 09FF
56 09FF
57 09FF
58 09FF
59 09FF
60 09FF 18E2      ; DEFS +9
; ORG 09FF ; ?FLAS
; ?FLAS: ENT
; JR ?FLS

```

```

01 0A01 ;
02 0A01 ;
03 0A01 ;
04 0A01 ; SHORT AND LONG PULSE FOR 1 BIT WRITE
05 0A01 ;
06 0A01 ;
07 0A01 F5 ;
08 0A02 3E03 ; 12
09 0A04 3203E0 LD A,03H ; 9
10 0A07 CD5907 CALL (CSTPT),A ; $E003 PC3=1:16
11 0A0A CD5907 CALL DLY1 ; 20+18*21+20
12 0A0B 3E02 LD A,02H ; 20+18*21+20
13 0A0F 3203E0 LD (CSTPT),A ; 9
14 0A12 CD5907 CALL DLY1 ; $E003 PC3=0:16
15 0A15 CD5907 CALL DLY1 ; 20+18*21+20
16 0A18 F1 POP AF ; 11
17 0A19 C9 RET ; 11
18 0A1A ;
19 0A1A ;
20 0A1A ; LONG:
21 0A1A F5 ;
22 0A1B 3E03 ; 11
23 0A1D 3203E0 LD A,03H ; 9
24 0A20 CD4909 CALL (CSTPT),A ; 16
25 0A23 3E02 LD A,02H ; 20+18*89+20
26 0A25 3203E0 LD (CSTPT),A ; 9
27 0A28 CD4909 CALL DLY4 ; 16
28 0A2B F1 POP AF ; 20+18*89+20
29 0A2C C9 RET ; 11
30 0A2D ;
31 0A2D ;
32 0A2D ;
33 0A32 ;
34 0A32 ;
35 0A32 ;
36 0A32 ;
37 0A32 ;
38 0A32 ;
39 0A32 ;
40 0A32 ;
41 0A32 ;
42 0A32 ;
43 0A32 ;
44 0A32 ;
45 0A32 ;
46 0A32 ;
47 0A32 ;
48 0A32 ;
49 0A32 ;
50 0A32 ;
51 0A32 ;
52 0A32 3EF8 ; LINE 8SWEET
53 0A34 3200E0 LD A,F8H ; (KEYPA),A
54 0A37 00 NOP ;
55 0A38 3A01E0 LD A,(KEYPB)
56 0A3B B7 OR A
57 0A3C 1F RRA
58 0A3D 0A8009 JP C,?BRK2 ; SHIFT ?
59 0A40 17 RLA
60 0A41 17 RLA

```

```

01 0A42 3004 ;
02 0A44 3E40 LD NC,?BRK1 ; BREAK ?
03 0A46 37 SCF ; SHIFT D6=1
04 0A47 C9 RET
05 0A48 ;
06 0A48 AF ;
07 0A48 AF ; ?BRK1: XOR A ; SHIFT ?
08 0A49 C9 RET
09 0A4A ;
10 0A4A ;
11 0A4A ; 320 U SEC DELAY
12 0A4A ;
13 0A4A ; DLY3: ENT
14 0A4A 3E52 LD A,52H ; 331.35 micro sec
15 0A4C C3207 JP 0762H ; JP DLY2+2
16 0A4F ;
17 0A4F ;
18 0A4F ; DEFS +1
19 0A50 ;
20 0A50 ;
21 0A50 ;
22 0A50 ;
23 0A50 ;
24 0A50 ;
25 0A50 ;
26 0A50 ;
27 0A50 ;
28 0A50 ;
29 0A50 ;
30 0A50 ;
31 0A50 ;
32 0A50 ;
33 0A50 ;
34 0A50 ;
35 0A50 ;
36 0A50 ;
37 0A50 ;
38 0A50 ;
39 0A50 ;
40 0A50 D5 ;
41 0A51 E5 ;
42 0A52 AF ;
43 0A53 06F8 ;
44 0A55 57 ;
45 0A56 CD320A ;
46 0A59 2004 ;
47 0A5B 1688 ;
48 0A5D 1814 ;
49 0A5F ;
50 0A5F 3005 ;
51 0A61 57 ;
52 0A62 1802 ;
53 0A64 ;
54 0A64 CBFA ;
55 0A66 05 ;
56 0A66 05 ;
57 0A67 78 ;
58 0A68 3200E0 ;
59 0A6B FEEF ;
60 0A6D 2008 ;

```



```

01 0B1A B2 DEF8 B2H
02 0B1B B6 DEF8
03 0B1C BA DEF8 BAH
04 0B1D BE DEF8 BEH
05 0B1E 9F DEF8 9FH
06 0B1F B3 DEF8 B3H
07 0B20 B7 DEF8 B7H
08 0B21 BB DEF8 BBH
09 0B22 BF ; 90 - 9F ;
10 0B23 A3 DEF8 BFH
11 0B24 A5 DEF8 A3H
12 0B25 A4 DEF8 85H
13 0B26 A5 DEF8 A4H
14 0B27 A6 DEF8 A5H
15 0B28 94 DEF8 A6H
16 0B29 87 DEF8 94H
17 0B2A 88 DEF8 87H
18 0B2B 9C DEF8 88H
19 0B2C 82 DEF8 9CH
20 0B2D 98 DEF8 82H
21 0B2E 84 DEF8 98H
22 0B2F 92 DEF8 84H
23 0B30 90 DEF8 92H
24 0B31 83 DEF8 90H
25 0B32 ; A0 - AF ;
26 0B33 91 DEF8 83H
27 0B34 9A DEF8 91H
28 0B35 97 DEF8 81H
29 0B36 93 DEF8 9AH
30 0B37 95 DEF8 97H
31 0B38 89 DEF8 93H
32 0B39 A1 DEF8 95H
33 0B3A AF DEF8 89H
34 0B3B 8B DEF8 A1H
35 0B3C 86 DEF8 AFH
36 0B3D 96 DEF8 8BH
37 0B3E A2 DEF8 86H
38 0B3F AB DEF8 96H
39 0B40 AA DEF8 A2H
40 0B41 8A DEF8 ABH
41 0B42 8E DEF8 AAH
42 0B43 B0 DEF8 8AH
43 0B44 AD DEF8 8EH
44 0B45 8D DEF8 B0H
45 0B46 A7 DEF8 ADH
46 0B47 A8 DEF8 8DH
47 0B48 A9 DEF8 A7H
48 0B49 8F DEF8 A8H
49 0B4A 8C DEF8 A9H
50 0B4B AE DEF8 8FH
51 0B4C AC DEF8 8CH
52 0B4D 9B DEF8 AEH
53 0B4E A0 DEF8 ACH
54 0B4F 99 DEF8 AC
55 0B50 BC DEF8 9BH
56 0B51 B8 DEF8 A0H
57 0B52 ; C0 - CF ;
58 0B53 ;
59 0B54 ;
60 0B55 ;

```

```

01 0AE2 10 DEF8 10H
02 0AE3 11 DEF8 11H
03 0AE4 12 DEF8 12H
04 0AE5 13 DEF8 13H
05 0AE6 14 DEF8 14H
06 0AE7 15 DEF8 15H
07 0AE8 16 DEF8 16H
08 0AE9 17 DEF8 17H
09 0AEA 18 DEF8 18H
10 0AEB 19 DEF8 19H
11 0AEC 1A DEF8 1AH
12 0AED 1B DEF8 1BH
13 0AEE 1C DEF8 1CH
14 0AEF 1D DEF8 1DH
15 0AEF 1E DEF8 1EH
16 0AEF 1F DEF8 1FH
17 0AF1 45 DEF8 45H
18 0AF2 C7 DEF8 C7H
19 0AF3 C8 DEF8 C8H
20 0AF4 C9 DEF8 C9H
21 0AF5 CA DEF8 CAH
22 0AF6 CB DEF8 CBH
23 0AF7 CC DEF8 CCH
24 0AF8 CD DEF8 CDH
25 0AF9 CE DEF8 CEH
26 0AFA CF DEF8 CFH
27 0AFB DF DEF8 DFH
28 0AFC E7 DEF8 E7H
29 0AFD E8 DEF8 E8H
30 0AFE E9 DEF8 E9H
31 0AFF E5 DEF8 E5H
32 0AFF E9 DEF8 E9H
33 0B00 EC DEF8 ECH
34 0B01 ED DEF8 EDH
35 0B02 D0 DEF8 D0H
36 0B03 D1 DEF8 D1H
37 0B04 D2 DEF8 D2H
38 0B05 D3 DEF8 D3H
39 0B06 D4 DEF8 D4H
40 0B07 D5 DEF8 D5H
41 0B08 D6 DEF8 D6H
42 0B09 D7 DEF8 D7H
43 0B0A D8 DEF8 D8H
44 0B0B D9 DEF8 D9H
45 0B0C DA DEF8 DAH
46 0B0D DB DEF8 DBH
47 0B0E DC DEF8 DCH
48 0B0F DD DEF8 DDH
49 0B10 DE DEF8 DEH
50 0B11 C0 DEF8 C0H
51 0B12 ; 80 - 8F ;
52 0B13 80 DEF8 80H
53 0B14 81 DEF8 81H
54 0B15 82 DEF8 82H
55 0B16 83 DEF8 83H
56 0B17 84 DEF8 84H
57 0B18 85 DEF8 85H
58 0B19 86 DEF8 86H
59 0B1A 87 DEF8 87H
60 0B1B 88 DEF8 88H

```


AGF 46

17-0138

8 22-004C

80 ASSEMBLY

◆◆

03.06.

45

0138> PA

-004C <

ASSEMBLER

82 **

[illegible]

```

01 0C2E F0 DEF8 F0H 60H ; NO
02 0C2F 6A DEF8 6AH ; +
03 0C30 6B DEF8 6BH ; *
04 0C31 CD DEF8 CDH ; CR
05 0C32 99 DEF8 99H ; Y
06 0C33 9A DEF8 9AH ; Z
07 0C34 A4 DEF8 A4H ; \
08 0C35 BC DEF8 BCH ; ^
09 0C36 40 DEF8 40H ; _
10 0C37 F0 DEF8 F0H ; NULL
11 0C38 F0 DEF8 F0H ; NULL
12 0C39 F0 DEF8 F0H ; NULL
13 0C3A 91 DEF8 91H ; q
14 0C3B 92 DEF8 92H ; r
15 0C3C 93 DEF8 93H ; s
16 0C3D 94 DEF8 94H ; t
17 0C3E 95 DEF8 95H ; u
18 0C3F 96 DEF8 96H ; v
19 0C40 97 DEF8 97H ; w
20 0C41 98 DEF8 98H ; x
21 0C42 89 DEF8 89H ; i
22 0C43 8A DEF8 8AH ; j
23 0C44 8B DEF8 8BH ; k
24 0C45 8C DEF8 8CH ; l
25 0C46 8D DEF8 8DH ; m
26 0C47 8E DEF8 8EH ; n
27 0C48 8F DEF8 8FH ; o
28 0C49 90 DEF8 90H ; p
29 0C4A 81 DEF8 81H ; a
30 0C4B 82 DEF8 82H ; b
31 0C4C 83 DEF8 83H ; c
32 0C4D 84 DEF8 84H ; d
33 0C4E 85 DEF8 85H ; e
34 0C4F 86 DEF8 86H ; f
35 0C50 87 DEF8 87H ; g
36 0C51 88 DEF8 88H ; h
37 0C52 61 DEF8 61H ; i
38 0C53 62 DEF8 62H ; j
39 0C54 63 DEF8 63H ; k
40 0C55 64 DEF8 64H ; l
41 0C56 65 DEF8 65H ; m
42 0C57 66 DEF8 66H ; n
43 0C58 67 DEF8 67H ; o
44 0C59 68 DEF8 68H ; p
45 0C5A 80 DEF8 80H ; *
46 0C5B A5 DEF8 A5H ; POND MARK
47 0C5C 2B DEF8 2BH ; YEN
48 0C5D 00 DEF8 00H ; SPACE
49 0C5E 60 DEF8 60H ; )
50 0C5F 69 DEF8 69H ; <
51 0C60 51 DEF8 51H ; >
52 0C61 57 DEF8 57H ; CLR
53 0C62 C6 DEF8 C6H ; CLR
54 0C63 C5 DEF8 C5H ; HOME
55 0C64 C2 DEF8 C2H ; CURSOR UP
56 0C65 C1 DEF8 C1H ; CURSOR DOWN
57 0C66 C3 DEF8 C3H ; CURSOR RIGHT
58 0C67 C4 DEF8 C4H ; CURSOR LEFT
59 0C68 5A DEF8 5AH ;
60 0C69 45 DEF8 45H ;

; GRAPHIC
; KTBLS: ENT
;S0 00-07
DEF8 BFH
DEF8 F0H
DEF8 55H
DEF8 C9H
DEF8 F0H
DEF8 42H
DEF8 B6H
DEF8 CDH
DEF8 75H
DEF8 76H
DEF8 82H
DEF8 D8H
DEF8 4EH
DEF8 F0H
DEF8 F0H
DEF8 F0H
DEF8 3CH
DEF8 30H
DEF8 44H
DEF8 71H
DEF8 79H
DEF8 DAH
DEF8 38H
DEF8 6DH
DEF8 7DH
DEF8 5CH
DEF8 5BH
DEF8 B4H
DEF8 1CH
DEF8 32H
DEF8 B0H
DEF8 D6H
DEF8 53H
DEF8 6FH
DEF8 DEH
DEF8 47H
DEF8 34H
DEF8 44H
DEF8 4BH
DEF8 72H
DEF8 37H
DEF8 3EH
DEF8 7FH
DEF8 7F
DEF8 7E
DEF8 7D
DEF8 7C
DEF8 7B
DEF8 7A
DEF8 79
DEF8 78
DEF8 77
DEF8 76
DEF8 75
DEF8 74
DEF8 73
DEF8 72
DEF8 71
DEF8 70
DEF8 6F
DEF8 6E
DEF8 6D
DEF8 6C
DEF8 6B
DEF8 6A
DEF8 69
DEF8 68
DEF8 67
DEF8 66
DEF8 65
DEF8 64
DEF8 63
DEF8 62
DEF8 61
DEF8 60
DEF8 5F
DEF8 5E
DEF8 5D
DEF8 5C
DEF8 5B
DEF8 5A
DEF8 59
DEF8 58
DEF8 57
DEF8 56
DEF8 55
DEF8 54
DEF8 53
DEF8 52
DEF8 51
DEF8 50
DEF8 4F
DEF8 4E
DEF8 4D
DEF8 4C
DEF8 4B
DEF8 4A
DEF8 49
DEF8 48
DEF8 47
DEF8 46
DEF8 45
DEF8 44
DEF8 43
DEF8 42
DEF8 41
DEF8 40
DEF8 3F
DEF8 3E
DEF8 3D
DEF8 3C
DEF8 3B
DEF8 3A
DEF8 39
DEF8 38
DEF8 37
DEF8 36
DEF8 35
DEF8 34
DEF8 33
DEF8 32
DEF8 31
DEF8 30
DEF8 2F
DEF8 2E
DEF8 2D
DEF8 2C
DEF8 2B
DEF8 2A
DEF8 29
DEF8 28
DEF8 27
DEF8 26
DEF8 25
DEF8 24
DEF8 23
DEF8 22
DEF8 21
DEF8 20
DEF8 1F
DEF8 1E
DEF8 1D
DEF8 1C
DEF8 1B
DEF8 1A
DEF8 19
DEF8 18
DEF8 17
DEF8 16
DEF8 15
DEF8 14
DEF8 13
DEF8 12
DEF8 11
DEF8 10
DEF8 0F
DEF8 0E
DEF8 0D
DEF8 0C
DEF8 0B
DEF8 0A
DEF8 09
DEF8 08
DEF8 07
DEF8 06
DEF8 05
DEF8 04
DEF8 03
DEF8 02
DEF8 01

```



```

** 780 ASSEMBLER 27-004C <17-0138> PAGE 51 03.06.85
01 0CF9 7C DEF8 10-17 7CH ; KA
02 0CF9 70 DEF8 70H ; KE
03 0CFA 41 DEF8 41H ; SHI
04 0CFB 31 DEF8 31H ; KO
05 0CFC 39 DEF8 39H ; HI
06 0CFD A6 DEF8 A6H ; TE
07 0CFE 78 DEF8 78H ; KI
08 0CFF 00 DEF8 00H ; CHI
09 0D00 00 DEF8 00H
10 0D01 3D DEF8 30H 18-1F ; FU
11 0D01 30 DEF8 50H ; M1
12 0D02 50 DEF8 6CH ; MU
13 0D03 6C DEF8 56H ; ME
14 0D04 56 DEF8 1DH ; RHI
15 0D05 10 DEF8 33H ; RA
16 0D06 33 DEF8 05H ; HE
17 0D07 05 DEF8 B1H ; HO
18 0D08 B1 DEF8
19 0D09 46 DEF8 46H 20-27 ; SA
20 0D09 46 DEF8 6EH ; TO
21 0D0A 6E DEF8 09H ; THU
22 0D0B 09 DEF8 48H ; SU
23 0D0C 48 DEF8 74H ; KU
24 0D0D 74 DEF8 43H ; SE
25 0D0E 43 DEF8 4CH ; SO
26 0D0F 4C DEF8 73H ; MA
27 0D10 73 DEF8
28 0D11 3F DEF8 3FH 28-2F ; A
29 0D11 3F DEF8 36H ; I
30 0D12 36 DEF8 7EH ; U
31 0D13 7E DEF8 38H ; E
32 0D14 38 DEF8 7AH ; O
33 0D15 7A DEF8 1EH ; NA
34 0D16 1E DEF8 5FH ; NI
35 0D17 5F DEF8 A2H ; NU
36 0D18 A2 DEF8
37 0D19 D3 DEF8 D3H 30-37 ; YO
38 0D19 D3 DEF8 9FH ; YU
39 0D1A 9F DEF8 01H ; YA
40 0D1B 01 DEF8 00H ; SPACE
41 0D1C 00 DEF8 9DH ; NO
42 0D1D 9D DEF8 A3H ; NE
43 0D1E A3 DEF8 00H ; RU
44 0D1F 00 DEF8 B9H ; RE
45 0D20 B9 DEF8
46 0D21 C6 DEF8 C6H 38-3F ; ?CLR
47 0D21 C6 DEF8 C5H ; ?HOME
48 0D22 C5 DEF8 C2H ; ?CURSOR UP
49 0D23 C2 DEF8 C1H ; ?CURSOR DOWN
50 0D24 C1 DEF8 C3H ; ?CURSOR RIGHT
51 0D25 C3 DEF8 C4H ; ?CURSOR LEFT
52 0D26 C4 DEF8 B8H ; DASH
53 0D27 B8 DEF8 BEH ; RO
54 0D28 BE DEF8
55 0D29 C03001 MEMORY DUMP
56 0D29 ; COMMAND 'D'
57 0D29 ;
58 0D29 DUMP: ENT CALL
59 0D29 ENT CALL
60 0D29 C03001

```

```

** 780 ASSEMBLER 27-004C <17-0138> PAGE 52 03.06.85
01 0D2C CDA602 CALL .40E
02 0D2F E5 PUSH HL
03 0D30 C01004 CALL HLHEX
04 0D33 D110 POP DE
05 0D34 3852 JR C,DUM1
06 0D36 E8 DE,HL
07 0D37 0608 LD B,08H
08 0D39 0E17 LD C,23
09 0D3B CDA605 CALL NPHL
10 0D3E C0B103 CALL SPHEX
11 0D41 23 HL
12 0D42 F5 INC AF
13 0D43 3A7111 LD A,(DSPXY)
14 0D46 81 ADD A,C
15 0D47 327111 LD (DSPXY),A
16 0D4A F1 POP AF
17 0D4B FE20 CP 20H
18 0D4D 3002 JR NC,+4
19 0D4F 3E2E LD A,2EH
20 0D51 C0B90B CALL ?ADCN
21 0D54 C06C09 PRINT3
22 0D57 3A7111 LD A,(DSPXY)
23 0D5A 0C INC C
24 0D5B 91 SUB C
25 0D5C 327111 LD (DSPXY),A
26 0D5F 00 DEC C
27 0D60 00 DEC C
28 0D61 00 DEC C
29 0D62 E5 PUSH HL
30 0D63 ED52 HL,DE
31 0D65 E1 POP HL
32 0D66 281D JR Z,DUM1-3
33 0D68 3EF8 LD A,F8H
34 0D6A 3200E0 LD (KEYPA),A
35 0D6D 00 NOP
36 0D6E 3A01E0 LD A,(KEYPB)
37 0D71 FEFE CP FEH
38 0D73 2003 JR NZ,+5
39 0D75 CDA600 CALL ?BLNK
40 0D78 10C4 DJNZ DUM2
41 0D7A C0CA08 CALL ?KEY
42 0D7D B7 OR A
43 0D7E 28FA JR Z,-4
44 0D80 C0320A CALL ?BRK
45 0D83 20B2 JR NZ,DUM3
46 0D85 C3AD00 JP ST1
47 0D88 21A000 LD HL,160
48 0D8B 19 ADD HL,DE
49 0D8C 18A8 DUM3-1
50 0D8E 51 DEF8 +24
51 0D8E 52 DEF8
52 0D8E 53 DEF8
53 0D8E 54 DEF8
54 0D8E 55 DEF8
55 0DA6 56 DEF8
56 0DA6 57 DEF8
57 0DA6 58 DEF8
58 0DA6 59 DEF8
59 0DA6 60 DEF8

```

```

; END ADR.
; DATA ER. THEN
; DISP 88BYTES
; CHA. PRINT BIAS
; NEWLINE PRINT
; SP. PRT.+ACC PRT.
; DISPLAY POINT
; X AXIS.=X+Creg
; ' '
; ASCII TO DSPAY CODE
; ASCII DSP POSITION
; SHIFT KEY ?
; 64MSEC DELAY
; STOP DISP
; SPACE KEY THEN STOP
; BREAK IN ?
; COMMAND IN !
; 20*8 BYTE

```

```

01 00A6      ; V-BLANK CHECK      ;
;
02 00A6      ?BLNK: ENT
03 00A6      LD A,(KEYPC)
04 00A6      RLC A
05 00A7      JR NC,-4
06 00AA      LD A,(KEYPC)
07 00AB      RLC A
08 00AD      JR C,-4
09 00B0      POP AF
10 00B1      RET
11 00B3      ;
12 00B4      ;
13 00B5      ;ORG 00B5H;?DSP
14 00B5      ;
15 00B5      ;
16 00B5      ;
17 00B5      ; DISPLAY ON POINTER ;
18 00B5      ;
19 00B5      ; ACC = DISPLAY CODE
20 00B5      ; EXCEPT F0H
21 00B5      ;
22 00B5      ;?DSP: ENT
23 00B5      PUSH AF
24 00B5      PUSH BC
25 00B6      PUSH DE
26 00B7      PUSH HL
27 00B8      ;
28 00B9      DSP01: ENT
29 00B9      CALL ?PONT
30 00BC      LD (HL),A
31 00BD      LD HL,(DSPXY)
32 00C0      LD A,L
33 00C1      CP +39
34 00C3      JR NZ,DSP04
35 00C3      CALL .MANG
36 00C8      JR C,DSP04
37 00CA      EX DE,HL
38 00CB      LD HL,(HL),+1
39 00CD      INC HL
40 00CE      LD (HL),0
41 00D0      DSP04: ENT
42 00D0      LD A,C3H
43 00D2      JR ?DPCT+4
44 00D4      ;
45 00D4      ;
46 00D4      ;
47 00D4      ; GRAPHIC STATUS CHECK
48 00D4      ;
49 00D4      ;
50 00D4      GRSTAS: LD A,(KANAF)
51 00D7      CP 01H
52 00D9      LD A,CAH
53 00DB      RET
54 00DC      ;
55 00DC      ;
56 00DC      ;
57 00DC      ;
58 00DC      ;
59 00DC      ;ORG 00DCH;?DPCT
60 00DC      ;

```

```

** Z80 ASSEMBLER 27-004C <17-013B> PAGE 55                                03.06.85
01 0E16 2E00      LD L,0
02 0E18 24       INC H
03 0E19 7C       CP A,H
04 0E1A FE19     JR C,+25
05 0E1C 38E1     JR C,CURS1
06 0E1E 2618     LD L,H,+24
07 0E20 227111  LD L,(DSPXY),HL
08 0E23 1848     ENT SCROL
09 0E25         ;
10 0E25         ; CURSL:
11 0E25 EB       EX DE,HL
12 0E26 7D       LD A,L
13 0E27 B7       OR A
14 0E28 2803     JR Z,+5
15 0E2A 2D       DEC L
16 0E2B 18D2     JR CURS3
17 0E2D 2E27     LD L,+39
18 0E2F 25       DEC H
19 0E30 F2080E   JP P,CURSUI
20 0E33 2600     LD H,0
21 0E35 227111  LD L,(DSPXY),HL
22 0E38 18C8     JR CURS5
23 0E3A         ;
24 0E3A         ; CLRS:
25 0E3A 217311  LD HL,MANG
26 0E3B B,27    LD B,27
27 0E3D 0618     LD L,0
28 0E3F C0D80F   CALL ?CLER
29 0E42 210000   LD L,D000H
30 0E45 C0D409   CALL #CLR08
31 0E48 3E71     LD A,71H
32 0E4A C0D509   CALL #CLR8
33 0E4D         HOME:
34 0E4D 210000   LD HL,0
35 0E50 18AD     JR CURS3
36 0E52         ;
37 0E52         ; DEFS
38 0E5A         ;
39 0E5A         ; CR
40 0E5A         ;
41 0E5A         ; CR:
42 0E5A C0F302   CALL .MANG
43 0E5D 0F       RRCA
44 0E5E 30B6     JR L,0
45 0E60 2E00     LD L,0
46 0E62 24       INC H
47 0E63 FE18     CP +24
48 0E65 2803     JR Z,CR1
49 0E67 24       INC H
50 0E68 1895     JR CURS1
51 0E6A         CR1:
52 0E6A 227111  LD L,(DSPXY),HL
53 0E6D         ;
54 0E6D         ; SCROL
55 0E6D         ;
56 0E6D         ; SCROL:
57 0E6D 01C003   LD BC,03C0H
58 0E70 110000   LD DE,SCRN
59 0E73 2128D0   LD HL,SCRN+40
60 0E76 C5       PUSH BC

** Z80 ASSEMBLER 27-004C <17-013B> PAGE 56                                03.06.85
01 0E77 EDB0     LDIR POP
02 0E79 C1       PUSH BC
03 0E7A D5       LD DE,SCRN+800H
04 0E7B 1100D8   LD HL,SCRN+828H
05 0E7E 2128D8   LDIR
06 0E81 EDB0     LD B,40
07 0E83 0628     EX DE,HL
08 0E85 EB       LD A,71H
09 0E86 3E71     LD A,71H
10 0E88 CDD00F   ?DINT HL
11 0E8B E1       LD B,40
12 0E8C 0628     LD ?CLER
13 0E8E CDD80F   LD BC,26
14 0E91 011A00   LD DE,MANG
15 0E94 117311  LD HL,MANG+1
16 0E97 217411  LDIR
17 0E9A EDB0     LD (HL),0
18 0E9C 3600     LD A,(MANG)
19 0E9E 3A7311   OR A
20 0EA1 B7       JR Z,?RSTR
21 0EA2 2841     LD HL,DSPXY+1
22 0EA4 217211  DEC (HL)
23 0EA7 35       JR SCROL
24 0EA8 18C3     ;
25 0EAA         ; CONTROL CODE TABLE
26 0EAA         ;
27 0EAA         ; CTBL:
28 0EAA 6D0E     ENT SCROL
29 0EAA F80D     DEFJ CURS0
30 0EAC F80D     DEFJ CURSU
31 0EAE 050E     DEFJ CURSR
32 0EB0 000E     DEFJ CURSL
33 0EB2 250E     DEFJ HOME
34 0EB4 4D0E     DEFJ CLRS
35 0EB6 3A0E     DEFJ DEL
36 0EB8 F80E     DEFJ INST
37 0EBA 380F     DEFJ ALPHA
38 0EBC E10E     DEFJ KANA
39 0EBE EE0E     DEFJ ?RSTR
40 0EC0 E50E     DEFJ ?RSTR CR
41 0EC2 E50E     DEFJ ?RSTR
42 0EC4 5A0E     DEFJ ?RSTR
43 0EC6 E50E     DEFJ ?RSTR
44 0EC8 E50E     DEFJ ?RSTR
45 0ECA         ;
46 0ECA         ; INST BYPASS
47 0ECA         ;
48 0ECA         ;
49 0ECA         ; INST2:
50 0ECA CBDC     SET 3,H
51 0ECC 7E       LD A,(HL)
52 0ECD 23       INC HL
53 0ECE 77       LD L,(HL),A
54 0ECF 2B       DEC HL
55 0ED0 CB9C     RES 3,H
56 0ED2 EDA8     LDD A,C
57 0ED4 79       LD B,A
58 0ED5 B0       OR NZ,INST2
59 0ED6 20F2     JR DE,HL
60 0ED8 EB       EX

```

```

** 280 ASSEMBLER 22-004C <12-0138> PAGE 57 03.06.85
01 0ED9 3600 LD (HL),0
02 0ED8 CBDC SET 3,H
03 0EDD 3671 LD (HL),71H
04 0EDF 1804 JR ?RSTR
;
;
;
; ORG 0EE1H:ALPHA
ALPHA: ENT XOR A
ALPH1: ENT LD (KANAF),A
;
;
; RESTORE ;
?RSTR: ENT POP HL
?RSTR1: ENT DE
22 0EE6 D1 POP BC
23 0EE7 C1 POP AF
24 0EE8 F1 POP AF
25 0EE9 C9 RET
;
; MONITOR WORK AREA ;
26 0EEA
27 0EEB
28 0EEC
29 0000 P
30 0003 P
31 0EEA
32 0EEA
33 0EEA
34 0EEA
35 0EEE
36 0EEE
37 0EEE CDD40D
38 0EEE CAB90D
39 0EF1 CAB90D P
40 0EF4 3E01 LD A,+1
41 0EF6 18EA JR ALPH1
42 0EF8
43 0EF8
44 0EF8 ENT
45 0EF8 EX DE,HL
46 0EF9 7C LD A,H
47 0EFA B5 OR L
48 0EFB 28E8 JR Z,?RSTR
49 0EFD 70 LD A,L
50 0EFE B7 OR A
51 0EFF 2000 JR NZ,DEL1
52 0F01 CDF302 CALL .MANG
53 0F04 3808 JR C,DEL1
54 0F06 CDB10F CALL ?PONT
55 0F09 2B DEC HL
56 0F0A 3600 LD (HL),+0
57 0F0C 1825 JR INST-5
58 0F0E ENT .MANG
59 0F11 0F RRCA
60 0F11 0F

01 0F12 3E28 LD A,40
02 0F14 3001 JR NC,+3
03 0F16 07 RLCA
04 0F17 95 SUB L
05 0F18 47 LD B,A
06 0F19 CDB10F CALL ?PONT
07 0F1C 7E LD A,(HL)
08 0F1D 2B DEC HL
09 0F1E 77 LD (HL),A
10 0F1F 23 INC HL
11 0F20 CBDC SET 3,H
12 0F22 7E LD A,(HL)
13 0F23 2B DEC HL
14 0F24 77 LD (HL),A
15 0F25 CB9C RES 3,H
16 0F27 23 INC HL
17 0F28 23 INC HL
18 0F29 10F1 DJNZ DEL2
19 0F2B 2B DEC HL
20 0F2C 3600 LD (HL),0
21 0F2E CBDC SET 3,H
22 0F30 217100 LD HL,71H
23 0F33 3EC4 LD A,C4H
24 0F35 C3E00D JP ?DPCT+4
25 0F38
26 0F38 ENT .MANG
27 0F38 CDF302 CALL RRCA
28 0F3B 0F LD L,+39
29 0F3C 2E27 LD A,L
30 0F3E 7D JR NC,+3
31 0F3F 3001 INC H
32 0F41 24 LD ?PNT1
33 0F42 CDB40F CALL HL
34 0F45 E5 PUSH HL
35 0F46 2A7111 LD HL,(DSPXY)
36 0F49 3002 JR NC,+4
37 0F4B 3E4F LD A,+79
38 0F4D 95 SUB L
39 0F4E 0600 LD B,0
40 0F50 4F LD C,A
41 0F51 01 POP DE
42 0F52 2891 JR Z,?RSTR
43 0F54 1A LD A,(DE)
44 0F55 B7 OR A
45 0F56 288D JR NZ,?RSTR
46 0F58 62 LD H,D
47 0F59 6B LD L,E
48 0F5A 2B DEC HL
49 0F5B C3CA0E JP INST2
50 0F5E
51 0F5E ; PROGRAM SAVE
52 0F5E ;
53 0F5E ; CMD. 'S'
54 0F5E ;
55 0F5E ;
56 0F5E ; SAVE: ENT
57 0F5E C03001 CALL HEX1Y
58 0F61 220411 LD (DIADR),HL
59 0F64 44 LD B,H
60 0F65 4D LD C,L

; START ADR. BUFFER
; DATA ADR. BUFFER

```

96


```

**      Z80 ASSEMBLER 22-004C  <17-013B>  PAGE 65      03.06.85
SP      10F0  SPMX      03B1  SS      00A2  ST0      0070  ST1      00AD
ST2     00BB  START     004A  STRGF   1193  SUNDT   1197  SUNDG   E008
SV0     0BA2  SV1       0BB5  SUEP0   0A66  SUEP01  0A64  SUEP2   0A7F
SUEP3   0A77  SUEP6     0A5F  SUEP9   0A73  SURK    119D  TEMP    E008
TEMPW   119E  TIMFG     119C  TIM1N   038D  TIMR0   003B  TIMST   0033
TM1     0675  TM2       0678  TM3    0688  TM4     069B  TMARK   065B
TMCNT   1195  TVF1      05B2  TVF2   05B8  TVF3    05CC  TVRFY   05AD
VERFY   002D  VGOFF     0747  VRFY    0FCB  VRNS    0BC5  UBY1    076D
UBYTE   0767  WRDAT     0024  WRI1    0444  WRI2    045E  WRI3    0464
URINF   0021  WTAP1     0494  WTAP2   04A5  WTAP3   04D2  UTAPE   048A
XTEMP   0041

```

3-2 MZ Disk control

[illegible]

```

01 E051 200F      JR      NZ,RTY2
02 E053 214511    LD      HL,RTYF
03 E056 35        DEC     (HL)
04 E057 2809      JR      Z,RTY2
05 E059 F1        POP     AF
06 E05A 3A011     LD      A,(FNUPS1)
07 E05D 323F11   LD      A,(FNUPS),A
08 E060 18B3      JR      RTY
09 E062          ;
10 E062 C0DBE1    RTY2:  CALL  URCAN
11 E065 C083E0    RTY2:  CALL  QDHPC
12 E068 F1        POP     AF
13 E069 C9        RTY4:  RET
14 E06A          ;
15 E06A          SKP     H
16 E06A

```

```

01 E051 200F      JR      NZ,RTY2
02 E053 214511    LD      HL,RTYF
03 E056 35        DEC     (HL)
04 E057 2809      JR      Z,RTY2
05 E059 F1        POP     AF
06 E05A 3A011     LD      A,(FNUPS1)
07 E05D 323F11   LD      A,(FNUPS),A
08 E060 18B3      JR      RTY
09 E062          ;
10 E062 C0DBE1    RTY2:  CALL  URCAN
11 E065 C083E0    RTY2:  CALL  QDHPC
12 E068 F1        POP     AF
13 E069 C9        RTY4:  RET
14 E06A          ;
15 E06A          SKP     H
16 E06A

```

```

01 E051 200F      JR      NZ,RTY2
02 E053 214511    LD      HL,RTYF
03 E056 35        DEC     (HL)
04 E057 2809      JR      Z,RTY2
05 E059 F1        POP     AF
06 E05A 3A011     LD      A,(FNUPS1)
07 E05D 323F11   LD      A,(FNUPS),A
08 E060 18B3      JR      RTY
09 E062          ;
10 E062 C0DBE1    RTY2:  CALL  URCAN
11 E065 C083E0    RTY2:  CALL  QDHPC
12 E068 F1        POP     AF
13 E069 C9        RTY4:  RET
14 E06A          ;
15 E06A          SKP     H
16 E06A

```

```

01 E051 200F      JR      NZ,RTY2
02 E053 214511    LD      HL,RTYF
03 E056 35        DEC     (HL)
04 E057 2809      JR      Z,RTY2
05 E059 F1        POP     AF
06 E05A 3A011     LD      A,(FNUPS1)
07 E05D 323F11   LD      A,(FNUPS),A
08 E060 18B3      JR      RTY
09 E062          ;
10 E062 C0DBE1    RTY2:  CALL  URCAN
11 E065 C083E0    RTY2:  CALL  QDHPC
12 E068 F1        POP     AF
13 E069 C9        RTY4:  RET
14 E06A          ;
15 E06A          SKP     H
16 E06A

```

```

01 E051 200F      JR      NZ,RTY2
02 E053 214511    LD      HL,RTYF
03 E056 35        DEC     (HL)
04 E057 2809      JR      Z,RTY2
05 E059 F1        POP     AF
06 E05A 3A011     LD      A,(FNUPS1)
07 E05D 323F11   LD      A,(FNUPS),A
08 E060 18B3      JR      RTY
09 E062          ;
10 E062 C0DBE1    RTY2:  CALL  URCAN
11 E065 C083E0    RTY2:  CALL  QDHPC
12 E068 F1        POP     AF
13 E069 C9        RTY4:  RET
14 E06A          ;
15 E06A          SKP     H
16 E06A

```

QDIO

```

;INIT CHECK SIZE (60KByt
;Retry 4
;write ?
;FNUPF CLR
;RETSP <= SP-2

```

```

ENT A,5
LD (RTYF),A
DI QMEIN
CALL NC
RET AF
PUSH 40
CP 40
JR Z,RTY4
MTOF
CALL A,(QDPA)
LD 4
CP 4
NZ,RTY3
JR A,(FNUPF)
LD A
Z,RTY3
XOR A
LD (FNUPF),A
LD A,(FNA)
PUSH HL
LD (RETSP),SP
POP HL
DI
CALL QDSVFN
JR C,RTY2
CALL MTOF
CP 3
NZ,RTY5
LD HL,HOPT
DEC (HL)
POP AF
PUSH AF
CP 41

```

```

QDIOS:
LD ENT
LD LD
LD LD
DI RTY:
CALL QMEIN
RET NC
PUSH 40
CP 40
JR Z,RTY4
MTOF
CALL A,(QDPA)
LD 4
CP 4
NZ,RTY3
JR A,(FNUPF)
LD A
Z,RTY3
XOR A
LD (FNUPF),A
LD A,(FNA)
PUSH HL
LD (RETSP),SP
POP HL
DI
CALL QDSVFN
JR C,RTY2
CALL MTOF
CP 3
NZ,RTY5
LD HL,HOPT
DEC (HL)
POP AF
PUSH AF
CP 41

```

```

QDIOS:
LD ENT
LD LD
LD LD
DI RTY:
CALL QMEIN
RET NC
PUSH 40
CP 40
JR Z,RTY4
MTOF
CALL A,(QDPA)
LD 4
CP 4
NZ,RTY3
JR A,(FNUPF)
LD A
Z,RTY3
XOR A
LD (FNUPF),A
LD A,(FNA)
PUSH HL
LD (RETSP),SP
POP HL
DI
CALL QDSVFN
JR C,RTY2
CALL MTOF
CP 3
NZ,RTY5
LD HL,HOPT
DEC (HL)
POP AF
PUSH AF
CP 41

```

```

01 E06A E0734811      ;MEIN: LD (RETSP),SP
02 E06E 3A3011      LD A,<QDPA>
03 E071 3D          DEC A
04 E071 3D          JR Z,QDRC
05 E072 2816      DEC A
06 E074 3D          JR Z,QDFM
07 E075 2819      DEC A
08 E077 3D          JR Z,QDRD
09 E078 2860      DEC A
10 E07A 3D          JP Z,QDUR
11 E07B CA4EE1      DEC A
12 E07E 3D          JR Z,QDHPC
13 E07F 2802      JR MTOFX
14 E081 1850      ;
15 E083            ;=====
16 E083            ;
17 E083            ; Head Point Clear
18 E083            ;
19 E083            ;=====
20 E083            ;
21 E083            ; QDHPC: ENT PUSH AF
22 E083 F5          XOR A
23 E083 AF          LD (HDPT),A
24 E084 AF          POP AF
25 E085 323011      RET
26 E088 F1          ;
27 E089 C9          ; Ready Check
28 E08A            ;=====
29 E08A            ;
30 E08A            ;
31 E08A            ;
32 E08A            ; QDRC: ENT LD A,<QDPB>
33 E08A            ; JP QREDY
34 E08A            ;
35 E08A 3A3111      ; SKP H
36 E08A C330E2      ;
37 E080            ;
38 E090            ;
39 E090            ;

01 E090            ;
02 E090            ;=====
03 E090            ;
04 E090            ; Format
05 E090            ;=====
06 E090            ;
07 E090            ; QDFM: ENT XOR A
08 E090 AF          QDSVFN
09 E091 CD68E2      CALL SYNC2
10 E091 CD68E2      CALL BC,FMS
11 E094 CD65E3      LD A,AAH
12 E097 01FFFE      LD A,AAH
13 E09A 3EAA      ;
14 E09C            ; QDFM1: CPL
15 E09C 2F          LD D,A
16 E09D 57          LD TRANS
17 E09E CD0BE3      CALL BC
18 E0A1 0B          LD A,B
19 E0A2 78          LD C
20 E0A3 B1          OR C
21 E0A4 2803      JR Z,QDFM2
22 E0A6 7A          LD A,D
23 E0A7 18F3      JR QDFM1
24 E0A9            ;
25 E0A9 CD82E3      EOM
26 E0AC CDE8E2      CALL MTOF
27 E0AF CD9BE2      CALL MTON
28 E0B2 3A4311      LD A,(FNB)
29 E0B5 3D          DEC A
30 E0B6 201E      JR NZ,FMERR
31 E0B8 CDFDE2      CALL SYNC2
32 E0BB 01FFFE      LD BC,FMS
33 E0BE 1E55      LD E,55H
34 E0C0 BB          CP E
35 E0C1 2013      JR NZ,FMERR
36 E0C3 0B          DEC BC
37 E0C4 78          LD A,B
38 E0C5 B1          OR C
39 E0C6 2808      JR Z,QDFM4
40 E0C8 7B          LD A,E
41 E0C9 2F          CPL
42 E0CA 5F          LD E,A
43 E0CB CDF0E3      RDATA
44 E0CE 18F0      CALL QDFM3
45 E0D0            ;
46 E0D0 CD03E3      QDFM4: CALL
47 E0D3 C3E8E2      MTOFX: JP
48 E0D6            ;
49 E0D6 3E29      FMERR: LD A,41
50 E0D8 37          SCF
51 E0D9 C9          RET
52 E0DA            ;
53 E0DA            ; SKP H

```

;Hard Err

```

01 E00A
02 E00A
03 E00A
04 E00A
05 E00A
06 E00A
07 E00A
08 E00A
09 E00A 3A4411
10 E00D B7
11 E00E CC9BE2
12 E0E1 CD14E1
13 E0E4 D8
14 E0E5 CD35E4
15 E0E8 CDF0E3
16 E0E8 CDF0E3
17 E0EB 4F
18 E0EC CDF0E3
19 E0EF 47
20 E0F0 B1
21 E0F1 CAESE1
22 E0F4 2A3411
23 E0F7 ED42
24 E0F9 DAESE1
25 E0FC 2A3211
26 E0FF
27 E0FF
28 E0FF
29 E0FF CDF0E3
30 E102 77
31 E103 23
32 E104 0B
33 E105 78
34 E106 B1
35 E107 20F6
36 E109 CDC3E3
37 E10C 3A3111
38 E10F CB47
39 E111 20C0
40 E113 C9
41 E114
42 E114

```

;=====
;
; Read
;=====
;
QDR0: ENT A,(MTF)
LD A
OR A
CALL Z,MTON
CALL HPS
RET C
CALL BRKC
;
CALL RDATA
LD C,A
CALL RDATA
LD B,A
OR C
JP Z,QDWE1
LD HL,(QDPE)
SBC HL,BC
JP C,QDWE1
LD HL,(QDPC)
;Block Data Read
;
BDR: CALL RDATA
LD (HL),A
INC HL
DEC BC
LD A,B
OR C
JR NZ,BDR
RDCRC
CALL A,(QDPB)
BIT 0,A
JR NZ,MTOFX
RET
; SKP H

```

** 280 ASSEMBLER 27-004C <97-504M-V1.0C>
01 E114
02 E114
03 E114
04 E114
05 E114 214311
06 E117 35
07 E118 2830
08 E11A CDF0E2
09 E11D 4F
10 E11E 3A3D11
11 E121 213E11
12 E124 BE
13 E125 2020
14 E127 3C
15 E128 323D11
16 E12B 77
17 E12C 3A3111
18 E12F A9
19 E130 1F
20 E131 D0
21 E132
22 E132
23 E132
24 E132 CDF0E3
25 E135 4F
26 E136 CDF0E3
27 E139 47
28 E13A
29 E13A CDF0E3
30 E13D 0B
31 E13E 78
32 E13F B1
33 E140 20F8
34 E142 CDC3E3
35 E145 18CD
36 E147
37 E147 34
38 E148 18E8
39 E14A
40 E14A 3E28
41 E14C 37
42 E14D C9
43 E14E
44 E14E

```

;Head Point Search
;
HPS: ENT HL,FNB
LD (HL)
JR Z,HPNFE
SYNCL2
C,A
A,(HDP1)
HL,HDP10
CP (HL)
JR NZ,HPS1
INC A
(HDP1),A
(HL),A
A,(QDPB)
C
XOR A9
RRA
RET NC
;Dummy read
;
DMR: CALL RDATA
LD C,A
CALL RDATA
LD B,A
;
DMR1: CALL RDATA
DEC BC
LD A,B
OR C
JR NZ,DMR1
RDCRC
HPS
;next
;
HPS1: INC (HL)
JR DMR
;
HPNFE: LD A,40
SCF
RET
; SKP H
;Not Found
;BLKFLG => C reg
;Search ok ?
;HDP1 count up
;HDP10 count up
;=

PC	Op	OpC	OpD	OpE	OpF	OpG	OpH	OpI	OpJ	OpK	OpL	OpM	OpN	OpO	OpP	OpQ	OpR	OpS	OpT	OpU	OpV	OpW	OpX	OpY	OpZ	OpAA	OpAB	OpAC	OpAD	OpAE	OpAF	OpAG	OpAH	OpAI	OpAJ	OpAK	OpAL	OpAM	OpAN	OpAO	OpAP	OpAQ	OpAR	OpAS	OpAT	OpAU	OpAV	OpAW	OpAX	OpAY	OpAZ	OpBA	OpBB	OpBC	OpBD	OpBE	OpBF	OpBG	OpBH	OpBI	OpBJ	OpBK	OpBL	OpBM	OpBN	OpBO	OpBP	OpBQ	OpBR	OpBS	OpBT	OpBU	OpBV	OpBW	OpBX	OpBY	OpBZ	OpCA	OpCB	OpCC	OpCD	OpCE	OpCF	OpCG	OpCH	OpCI	OpCJ	OpCK	OpCL	OpCM	OpCN	OpCO	OpCP	OpCQ	OpCR	OpCS	OpCT	OpCU	OpCV	OpCW	OpCX	OpCY	OpCZ	OpDA	OpDB	OpDC	OpDD	OpDE	OpDF	OpDG	OpDH	OpDI	OpDJ	OpDK	OpDL	OpDM	OpDN	OpDO	OpDP	OpDQ	OpDR	OpDS	OpDT	OpDU	OpDV	OpDW	OpDX	OpDY	OpDZ	OpEA	OpEB	OpEC	OpED	OpEE	OpEF	OpEG	OpEH	OpEI	OpEJ	OpEK	OpEL	OpEM	OpEN	OpEO	OpEP	OpEQ	OpER	OpES	OpET	OpEU	OpEV	OpEW	OpEX	OpEY	OpEZ	OpFA	OpFB	OpFC	OpFD	OpFE	OpFF	OpFG	OpFH	OpFI	OpFJ	OpFK	OpFL	OpFM	OpFN	OpFO	OpFP	OpFQ	OpFR	OpFS	OpFT	OpFU	OpFV	OpFW	OpFX	OpFY	OpFZ	OpGA	OpGB	OpGC	OpGD	OpGE	OpGF	OpGG	OpGH	OpGI	OpGJ	OpGK	OpGL	OpGM	OpGN	OpGO	OpGP	OpGQ	OpGR	OpGS	OpGT	OpGU	OpGV	OpGW	OpGX	OpGY	OpGZ	OpHA	OpHB	OpHC	OpHD	OpHE	OpHF	OpHG	OpHH	OpHI	OpHJ	OpHK	OpHL	OpHM	OpHN	OpHO	OpHP	OpHQ	OpHR	OpHS	OpHT	OpHU	OpHV	OpHW	OpHX	OpHY	OpHZ	OpIA	OpIB	OpIC	OpID	OpIE	OpIF	OpIG	OpIH	OpIJ	OpIK	OpIL	OpIM	OpIN	OpIO	OpIP	OpIQ	OpIR	OpIS	OpIT	OpIU	OpIV	OpIW	OpIX	OpIY	OpIZ	OpJA	OpJB	OpJC	OpJD	OpJE	OpJF	OpJG	OpJH	OpJI	OpJJ	OpJM	OpJN	OpJO	OpJP	OpJQ	OpJR	OpJS	OpJT	OpJU	OpJV	OpJW	OpJX	OpJY	OpJZ	OpKA	OpKB	OpKC	OpKD	OpKE	OpKF	OpKG	OpKH	OpKI	OpKJ	OpKK	OpKL	OpKM	OpKN	OpKO	OpKP	OpKQ	OpKR	OpKS	OpKT	OpKU	OpKV	OpKW	OpKX	OpKY	OpKZ	OpLA	OpLB	OpLC	OpLD	OpLE	OpLF	OpLG	OpLH	OpLI	OpLJ	OpLK	OpLL	OpLM	OpLN	OpLO	OpLP	OpLQ	OpLR	OpLS	OpLT	OpLU	OpLV	OpLW	OpLX	OpLY	OpLZ	OpMA	OpMB	OpMC	OpMD	OpME	OpMF	OpMG	OpMH	OpMI	OpMJ	OpMK	OpML	OpMM	OpMN	OpMO	OpMP	OpMQ	OpMR	OpMS	OpMT	OpMU	OpMV	OpMW	OpMX	OpMY	OpMZ	OpNA	OpNB	OpNC	OpND	OpNE	OpNF	OpNG	OpNH	OpNI	OpNJ	OpNK	OpNL	OpNM	OpNN	OpNO	OpNP	OpNQ	OpNR	OpNS	OpNT	OpNU	OpNV	OpNW	OpNX	OpNY	OpNZ	OpOA	OpOB	OpOC	OpOD	OpOE	OpOF	OpOG	OpOH	OpOI	OpOJ	OpOK	OpOL	OpOM	OpON	OpOO	OpOP	OpOQ	OpOR	OpOS	OpOT	OpOU	OpOV	OpOW	OpOX	OpOY	OpOZ	OpPA	OpPB	OpPC	OpPD	OpPE	OpPF	OpPG	OpPH	OpPI	OpPJ	OpPK	OpPL	OpPM	OpPN	OpPO	OpPP	OpPQ	OpPR	OpPS	OpPT	OpPU	OpPV	OpPW	OpPX	OpPY	OpPZ	OpQA	OpQB	OpQC	OpQD	OpQE	OpQF	OpQG	OpQH	OpQI	OpQJ	OpQK	OpQL	OpQM	OpQN	OpQO	OpQP	OpQQ	OpQR	OpQS	OpQT	OpQU	OpQV	OpQW	OpQX	OpQY	OpQZ	OpRA	OpRB	OpRC	OpRD	OpRE	OpRF	OpRG	OpRH	OpRI	OpRJ	OpRK	OpRL	OpRM	OpRN	OpRO	OpRP	OpRQ	OpRR	OpRS	OpRT	OpRU	OpRV	OpRW	OpRX	OpRY	OpRZ	OpSA	OpSB	OpSC	OpSD	OpSE	OpSF	OpSG	OpSH	OpSI	OpSJ	OpSK	OpSL	OpSM</
----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	--------

```

** 280 ASSEMBLER 27-004C <9Z-504M-V1.0C> PAGE 11
03.06.85
01 E221 C0F0E3      CALL RDATA
02 E224 B9          CP
03 E225 20BE       JR NZ,Q0WE1
04 E227 C0F0E3      CALL RDATA
05 E22A B8         CP B
06 E22B 20B8       JR NZ,Q0WE1
07 E22D           ;
08 E22D C0F0E3      CALL RDATA
09 E230 BE         CP (HL)
10 E231 20B2       JR NZ,Q0WE1
11 E233 23        INC HL
12 E234 08        DEC BC
13 E235 78        LD A,B
14 E236 B1        OR C
15 E237 20F4       JR NZ,B0V1
16 E239 C3C3E3     JP R0CRC
17 E23C           ;
18 E23C           SKP H

01 E23C           ;
02 E23C           ;
03 E23C           ;
04 E23C           ;
05 E23C           ; Quick-Disk
06 E23C           ; Physical iocs (bisync)
07 E23C           ;
08 E23C           ; By MZ-700 IPL-ROM
09 E23C           ; MZ-1500 IPL-ROM
10 E23C           ; MZ-700 Disk-Basic
11 E23C           ; MZ-1500 Basic
12 E23C           ;
13 E23C           ; V2.0A 04.25.1984
14 E23C           ;
15 E23C           ;
16 E23C           ;
17 E23C           ; i/o port adrs
18 E23C           ;
19 E23C           ;
20 E23C           ; SIOAD: EQU F4H; sio A data
21 E23C           ; SIOBD: EQU F5H; sio B data
22 E23C           ; SIOAC: EQU F6H; sio A control
23 E23C           ; SIOBC: EQU F7H; sio B control
24 E23C           ;
25 E23C           ;
26 E23C           ;
27 E23C           ; Ready & Write protect
28 E23C           ; Acc = '0' : Ready check
29 E23C           ; Acc = '1' : & Write Protect
30 E23C           ;
31 E23C           ; QREY: ENT
32 E23C 47         LD B,A
33 E23D 3E02       LD A,02H
34 E23F D3F7       OUT (SIOBC),A
35 E241 3E81       LD A,81H
36 E243 D3F7       OUT (SIOBC),A
37 E245 3E02       LD A,02H
38 E247 D3F7       OUT (SIOBC),A
39 E249 DBF7       IN A,(SIOBC)
40 E24B E681       AND 81H
41 E24D FE81       CP NZ,IOE50
42 E24F C206E4     JP NZ,IOE50
43 E252 3E10       LD A,10H
44 E254 D3F6       OUT (SIOAC),A
45 E256 DBF6       IN A,(SIOAC)
46 E258 4F        LD C,A
47 E259 E608       AND 08H
48 E25B CA06E4     JP Z,IOE50
49 E25E 78        LD A,B
50 E25F B7        OR A
51 E260 C8        RET Z
52 E261 79        LD A,C
53 E262 E620       AND 20H
54 E264 C0        RET NZ
55 E265 C303E4     JP IOE46
56 E268           ;
57 E268           SKP H

```

```

01 E203 AF      XOR      A
02 E204 323E11 LD        (HDPT0),A
03 E207 C9      RET
04 E208         ;
05 E208         SKP      H

```

```

01 E268         ;
02 E268         ;
03 E268         ;
04 E268         ;
05 E268         ;
06 E268 F5      GDSVFN: ENT  PUSH
07 E269 212CE4 LD        HL,SIO5D
08 E26C 0609 LD        B,09H
09 E26E C0D8E2 CALL     LSINT
10 E271         ;
11 E271 3E10 SREDY: LD      A,10H
12 E273 D3F6 OUT      (SIOAC),A
13 E275 DBF6 IN      A,(SIOAC)
14 E277 E608 AND      8
15 E279 CA06E4 JP      Z,10E50
16 E27C 3E10 LD      A,10H
17 E27E D3F7 OUT      (SIOBC),A
18 E280 DBF7 IN      A,(SIOBC)
19 E282 E608 AND      8
20 E284 28EB JR      Z,SREDY
21 E286 01E900 LD      BC,00E9H
22 E289 C014E4 CALL  TIMW
23 E28C C09AE3 CALL  SBRK
24 E28F C07EE3 CALL  SYNC51
25 E292 F1      POP      AF
26 E293 C0DBE3 CALL  TRANS
27 E296 C0B2E3 CALL  EOM
28 E299 1833 JR      FNEND
29 E29B         ;
30 E29B         ;
31 E29B         ;
32 E29B         ;
33 E29B         ;
34 E29B         ;
35 E29B         ;
36 E29B         ;
37 E29B 2121E4 MTON: ENT  LD      HL,SIO1D
38 E29E 0608 LD      B,08H
39 E2A0 C0D8E2 CALL     LSINT
40 E2A3         ;
41 E2A3 3E10 LREDY: LD      A,10H
42 E2A5 D3F6 OUT      (SIOAC),A
43 E2A7 DBF6 IN      A,(SIOAC)
44 E2A9 E608 AND      8
45 E2AB CA06E4 JP      Z,10E50
46 E2AE C035E4 LD      A,10H
47 E2B1 3E10 LD      A,(SIOBC),A
48 E2B3 D3F7 OUT      (SIOBC),A
49 E2B5 DBF7 IN      A,(SIOBC)
50 E2B7 E608 AND      8
51 E2B9 28EB JR      Z,LREDY
52 E2BB 01E900 LD      BC,00E9H
53 E2BE C014E4 CALL  TIMW
54 E2C1 C013E3 CALL  SYNCL1
55 E2C4 324211 LD      (FNA),A
56 E2C7 3C      INC      A
57 E2C8 324311 LD      (FNB),A
58 E2CB C0C3E3 CALL  RDCRC
59 E2CE 214711 LD      HL,SYNCF
60 E2D1 C8DE SET     3,(HL)

```

```

** Z80 ASSEMBLER 27-004C <92-504M-V1.0C> PAGE 15 03.06.85

01 E208 ;
02 E208 ; sio initial
03 E208 ;
04 E208 ;
05 E208 LSIINT: LD C,F6H ; C sioac
06 E20A 0EF6 LD OTIR
07 E20C 3E05 LD (MTF),A
08 E20E 324411 LD (SIOAC),A
09 E2E1 D3F7 OUT
10 E2E3 3E80 LD A,80H
11 E2E5 D3F7 OUT (SIOBC),A
12 E2E7 C9 RET
13 E2E8 ;
14 E2E8 ; Motor off
15 E2E8 ;
16 E2E8 QDOFF: ENT
17 E2E8 MTOF: ENT
18 E2E8 F5 PUSH
19 E2E9 3E05 LD A,05H
20 E2EB D3F6 LD (SIOAC),A
21 E2ED 3E60 LD A,60H
22 E2EF D3F6 LD (SIOAC),A
23 E2F1 3E05 LD A,05H
24 E2F3 D3F7 OUT (SIOBC),A
25 E2F5 AF XOR A
26 E2F6 324411 LD (MTF),A
27 E2F9 D3F7 OUT (SIOBC),A
28 E2FB F1 POP AF
29 E2FC C9 RET
30 E2FD ;
31 E2FD ;
32 E2FD SKP H

01 E2FD ;
02 E2FD ; SYNCL1 -- LOAD F.N SYNC ONLY
03 E2FD ; (SEND BREAK 110ms)
04 E2FD ; SYNCL2 -- LOAD FIRST FILE SYNC
05 E2FD ; (SEND BREAK 110ms)
06 E2FD ; SYNCL3 -- LOAD FILES SYNC
07 E2FD ; (SEND BREAK 002ms)
08 E2FD ;
09 E2FD ; SYNCL2: ENT
10 E2FD LD A,58H
11 E2FF 060B LD B,0BH
12 E301 2121E4 LD HL,S10LD
13 E304 C0A3E3 CALL SYNCA
14 E307 214711 LD HL,SYNCF
15 E30A C85E LD 3,(HL)
16 E30C 010300 LD BC,3
17 E30F 2805 JR 2,TMLPL
18 E311 C89E RES 3,(HL)
19 E313 01A000 SYNCL1: LD BC,00A0H
20 E316 C014E4 ; TMLPL: CALL TIMU
21 E319 3E05 LD A,05H
22 E31B D3F7 OUT (SIOBC),A
23 E31D 3E82 LD A,82H
24 E31F D3F7 OUT (SIOBC),A
25 E321 3E03 LD A,03H
26 E323 D3F6 OUT (SIOAC),A
27 E325 3E03 LD A,D3H
28 E327 D3F6 OUT (SIOAC),A
29 E329 01C02C LD BC,2CC0H
30 E32C ; SYNCL0: LD A,10H
31 E32E D3F6 OUT (SIOAC),A
32 E330 D8F6 AND A,(SIOAC)
33 E332 E610 IN 10H
34 E334 2807 JR Z,SYNCL1
35 E336 0B DEC BC
36 E337 78 LD A,B
37 E338 B1 OR C
38 E339 20F1 JR NZ,SYNCL0
39 E33B 1817 JR SYNCL01
40 E33D ; SYNCL1: LD A,03H
41 E33F D3F6 OUT (SIOAC),A
42 E341 3EC3 LD A,C3H
43 E343 D3F6 OUT (SIOAC),A
44 E345 069F LD B,9FH
45 E347 ; SYNCL2: LD A,10H
46 E349 D3F6 OUT (SIOAC),A
47 E34B D8F6 IN A,(SIOAC)
48 E34D E601 AND 01H
49 E34F 2006 JR NZ,SYNCL3
50 E351 05 DEC B
51 E352 20F3 JR NZ,SYNCL2
52 E354 C30CE4 SYNCL01: JP IOE54
53 E357 3E03 ; SYNCL3: LD A,03H
54 E359 D3F6 OUT (SIOAC),A
55 E35B 3EC9 LD A,C9H

```

SKP H

01 E382

```

01 E35D D3F6      OUT (SIOAC),A
02 E35F CDF0E3    CALL RDATA
03 E362 C3F0E3    JP RDATA
04 E365          ;
05 E365          ;
06 E365          ;
07 E365          ;
08 E365          ;
09 E365          ;
10 E365          ;
11 E365          ;
12 E365          ;
13 E365          ;
14 E365          ;
15 E365 3E98      LD A,98H
16 E367 0609      LD B,09H
17 E369 212CE4    LD HL,SIOSD
18 E36C CDA3E3    CALL SYNCA
19 E36F CD9AE3    CALL SBRK
20 E372          ;
21 E372 214711    LD HL,SYNCF
22 E375 CB5E      BIT 3,(HL)
23 E377 011D00    LD BC,001DH
24 E37A 2805      JR 7,TMLPS
25 E37C CB9E      RES 3,(HL)
26 E37E 014001    SYNCS1: LD BC,0140H
27 E381          ;
28 E381 CD14E4    TMLPS: CALL TIMW
29 E384 3E05      LD A,05H
30 E386 D3F6      OUT (SIOAC),A
31 E388 3EEF      LD A,EFH
32 E38A D3F6      OUT (SIOAC),A
33 E38C 010100    LD BC,1
34 E38F CD14E4    CALL TIMW
35 E392 3EA5      LD A,5H
36 E394 C00BE3    TRANS EOHRS
37 E397 C380F3    JP EOHRS
38 E39A          ;
39 E39A          ;
40 E39A          ;
41 E39A          ;
42 E39A          ;
43 E39A          ;
44 E39A 3E05      ENT L
45 E39C D3F6      LD A,05H
46 E39E 3EFF      OUT (SIOAC),A
47 E3A0 D3F6      LD A,FFH
48 E3A2 C9        OUT (SIOAC),A
49 E3A3          RET
50 E3A3          ;
51 E3A3          ;
52 E3A3 0EF6      SYNCA: LD C,F6H
53 E3A5 ED79      OUT (C),A
54 E3A7 3E05      LD A,5
55 E3A9 D3F7      OUT (SIOBC),A
56 E3AB 3E80      LD A,80H
57 E3AD D3F7      OUT (SIOBC),A
58 E3AF EDB3      OTIR
59 E3B1 C9        RET
60 E3B2          ;

```

```

01 E3B2      ;
02 E3B2      ;
03 E3B2      ;
04 E3B2      ; EOM -- End off message
05 E3B2      ;   Save CRC#1,#2,FLAG
06 E3B2      ;   File space check
07 E3B2      ;
08 E3B2      ; EOM:
09 E3B2      LD BC,1
10 E3B2      LD CALL TIMW
11 E3B5      LD A,10H
12 E3B8      OUT (SIOBC),A
13 E3BA      IN A,(SIOBC)
14 E3BC      AND 8
15 E3BE      RET NZ
16 E3C0      JR IOE53
17 E3C1      LD A,1846
18 E3C3      ;
19 E3C3      ; RDCRC -- READ CRC & CHECK
20 E3C3      ;
21 E3C3      RDCRC: ENT
22 E3C3      LD B,3
23 E3C5      CALL RDATA
24 E3C8      DJNZ RDCR1
25 E3CA      IN A,(SIOAC)
26 E3CC      RRCA
27 E3CD      JR NC,RDCR2
28 E3CF      LD A,01H
29 E3D1      OUT (SIOAC),A
30 E3D3      IN A,(SIOAC)
31 E3D5      AND 40H
32 E3D7      JR NZ,IOE41
33 E3D9      OR A
34 E3DB      RET
35 E3DB      ;
36 E3DB      SKP H

01 E30B      ;
02 E30B      ; Save 1 chr by Acc
03 E30B      ; & ready check
04 E30B      ;
05 E30B      TRANS: ENT
06 E30B      PUSH AF
07 E30C      DBF6
08 E30E      AND 4
09 E310      JR Z,TRA1
10 E312      POP AF
11 E314      OUT (SIOAD),A
12 E316      LD A,10H
13 E318      OUT (SIOAC),A
14 E31A      IN A,(SIOAC)
15 E31C      AND 08H
16 E31E      JR Z,IOE50
17 E320      RET
18 E322      ;
19 E322      ; Read data (1 chr)
20 E324      ;
21 E324      RDATA: ENT
22 E326      CALL NRCK
23 E328      IN A,(SIOAC)
24 E32A      RLCA
25 E32C      JR C,IOE41
26 E32E      RRCA
27 E330      RRCA
28 E332      JR NC,RDATA
29 E334      IN A,(SIOAD)
30 E336      OR A
31 E338      RET
32 E33A      ;
33 E33A      SKP H

;WAIT 0.7ms

;NO file space

```

```

01 E400 ; i/o err
02 E400 ;
03 E400 ;
04 E400 3E29 ;Hard err
05 E402 21 DEF8 21H ;Write protect
06 E403 3E2E LD A,46 ;Not ready
07 E405 21 DEF8 21H ;No file space
08 E406 3E32 LD A,50 ;Un format
09 E408 21 DEF8 21H
10 E409 3E35 LD A,53
11 E40B 21 DEF8 21H
12 E40C 3E36 LD A,54
13 E40E ED7B4811 SP,(RETSP)
14 E412 37 SCF
15 E413 C9 RET
16 E414 ;
17 E414 ;
18 E414 ;
19 E414 ;
20 E414 ; wait timer
21 E414 ;
22 E414 ; BC=001H= 0.7ms( 0.704ms)
23 E414 ; 003H= 2.0ms( 2.107ms)
24 E414 ; 010H= 20.0ms( 19.938ms)
25 E414 ; 0A0H=110.0ms(110.050ms)
26 E414 ; 0E9H=160.0ms(160.140ms)
27 E414 ; 140H=220.0ms(219.940ms)
28 E414 ;
29 E414 ;
30 E414 ;
31 E414 ; TIMW: ENT AF
32 E414 F5 PUSH LD A,96H
33 E415 3E96 TIMW1: DEC A
34 E417 3D NZ,TIMW2
35 E418 20FD JR BC
36 E41A 0B DEC LD A,B
37 E41B 78 LD C
38 E41C B1 OR C
39 E41D 20F6 JR NZ,TIMW1
40 E41F F1 POP AF
41 E420 C9 RET
42 E421 ; SKP H
43 E421 ;

01 E421 ;
02 E421 ;
03 E421 ; SIO CH A COMMAND CHAIN
04 E421 ;
05 E421 ; SIOLD -- LOAD INIT. DATA
06 E421 ; SIOSD -- SAVE INIT. DATA
07 E421 ;
08 E421 ;
09 E421 ;
10 E421 ; SIOLD: DEF8 58H
11 E421 58 DEF8 04H
12 E422 04 DEF8 10H
13 E423 10 DEF8 05H
14 E424 05 DEF8 04H
15 E425 04 DEF8 03H
16 E426 03 DEF8 00H
17 E427 00 ;Rx 8bits
18 E428 06 DEF8 06H
19 E428 06 DEF8 16H
20 E429 16 DEF8 07H
21 E42A 07 DEF8 16H
22 E42B 16
23 E42C ;
24 E42C 98 SIOSD: DEF8 98H
25 E42D 98 ;Tx CRC Generator reset
26 E42D 04 DEF8 04H
27 E42E 10 DEF8 10H
28 E42F 06 DEF8 06H
29 E430 16 DEF8 16H
30 E431 07 DEF8 07H
31 E432 16 DEF8 16H
32 E433 05 DEF8 05H
33 E434 6D DEF8 6DH
34 E435 ;
35 E435 ; BREAK CHECK
36 E435 ;
37 E435 ; BRKC: LD A,E8H
38 E435 3EE8 LD (E000H),A
39 E437 3200E0 LD NOP
40 E43A 00 LD A,(E001H)
41 E43B 3A01E0 LD 81H
42 E43E E681 RET NZ
43 E440 C0 WRCAN
44 E441 C0DBE1 CALL SP,(RETSP)
45 E444 ED7B4811 LD SCF
46 E448 37 RET
47 E449 C9
48 E44A ; SKP H
49 E44A ;

```

** Z80 ASSEMBLER Z2-004C <97-504M-V1.0C> PAGE 23

03.06.85

```

01 E44A      ;-----< MFH Minifloppy control >-----
02 E44A      Subroutine (M8876)
03 E44A      ;
04 E44A      ;
05 E44A      ;
06 E44A      ; Call condition
07 E44A      ;
08 E44A      ; Case of disk initialize
09 E44A      DRIVE N=IX+0 (0-3)
10 E44A      ;
11 E44A      ; Case of sequential read & write
12 E44A      Drive n=IX+0 (0-3)
13 E44A      ;
14 E44A      ; Sector adrs =IX+1,2 (0-045FH)
15 E44A      (0-1119 )
16 E44A      ;
17 E44A      ; Byte size =IX+3,4
18 E44A      Address =IX+5,6
19 E44A      Next track =IX+7
20 E44A      Next sector =IX+8
21 E44A      Start track =IX+9
22 E44A      Start sector =IX+10
23 E44A      ;
24 E44A      ; I/O Port address
25 E44A      ;
26 E44A      CR: EQU D8H
27 00D8      EQU D9H
28 00D9      EQU DAH
29 00DA      EQU DBH
30 00DB      EQU DCH
31 00DC      EQU DDH
32 00DD      EQU D0H
33 E44A      ; Subroutine work
34 E44A      ;
35 E44A      BPRO: EQU CF00H
36 CF00      EQU 11A3H
37 11A3      EQU BPRO-23
38 E44A      BPARA: EQU
39 E44A      ;
40 E44A      ;-----< Erccode map >-----
41 E44A      ;
42 E44A      ; 50 :Not ready
43 E44A      ; 41 :Data error
44 E44A      ;
45 E44A      ; Track 00 err
46 E44A      ; Write protect err
47 E44A      ; Seek err
48 E44A      ; CRC err
49 E44A      ; Lost data
50 E44A      ; 54 :Unformat
51 E44A      ; Recode not found
52 E44A      ; 56 :Invalid data
53 E44A      ;
54 E44A      ; Main routine
55 E44A      ;
56 E44A      ;
57 E44A      ; FDX: ENT (SP),HL
58 E44A      ; EX (BPARA+21),HL
59 E44B      ; LD
60 E44E      ; CALL

```

```

01 E451 C2BAE4
02 E454 11E9DE
03 E457 21D1E4
04 E45A 010800
05 E45D E0B0
06 E45F C030E5
07 E462 D021E9CE
08 E466 C0A7E5
09 E469 2100CF
10 E46C 11CAE4
11 E46F 0607
12 E471 4E
13 E472 1A
14 E473 B9
15 E474 C2AEE4
16 E477 23
17 E478 13
18 E479 10F6
19 E47B 118ED
20 E47E DF
21 E47F 1107CF
22 E482 DF
23 E483 210012
24 E486 D07505
25 E489 D07406
26 E48C 2A14CF
27 E48F D07503
28 E492 D07404
29 E495 2A1ECF
30 E498 D07501
31 E49B D07402
32 E49E
33 E49E C0A7E5
34 E4A1 C030E5
35 E4A4
36 E4A4
37 E4A4
38 E4A4 010002
39 E4A7 D9
40 E4A8 2114CF
41 E4AB C3FCEC
42 E4AE
43 E4AE C030E5
44 E4B1 11F6ED
45 E4B4 180C
46 E4B6
47 E4B6 FE32
48 E4B8 2005
49 E4BA 11B5ED
50 E4BD 1803
51 E4BF 11E5ED
52 E4C2 31EE10
53 E4C5 2AFECE
54 E4C8 E3
55 E4C9 C9
56 E4CA
57 E4CA
58 E4CA 03
59 E4CB 4P504C50
60 E4CB 4P504C50

```

```

JP NZ,ERR10
LD DE,BPARA
LD HL,BOOT
BC,11
LDIR
CALL MOFF
LD IX,BPARA
LD BREAD
CALL HL,BPRO
DE,IPLMC
LD B,7
MCHECK: LD C,(HL)
LD A,(DE)
CP
JP NZ,MASTE
INC HL
INC DE
MCHECK
DJNZ 3
RST 3
DE,BPRO+7
RST 3
HL,1200H
LD (IX+5),L
LD (IX+6),H
LD HL,(BPRO+14H)
LD (IX+3),L
LD (IX+4),H
LD HL,(BPRO+1EH)
LD (IX+1),L
LD (IX+2),H
; CALL BREAD
; CALL MOFF
; Exec load file
LD BC,0200H
EXX
LD HL,BPRO+14H
JP EXF
; MASTE: CALL MOFF
LD DE,ERRM1
JR ERRTR1
; ERRTRT: CP 50
NZ,ERRTR0
DE,IPLM3
ERRTR1
NOTIO: JR
ERRTR0: LD DE,ERRM0
ERRTR1: LD SP,10EEH
LD HL,(BPARA+21)
EX (SP),HL
RET
; PARAMETER SETTING
IPLMC: DEFB 03H
DEFM 'IPLPRO'
; IPL MASTER FLAG

```

; IPL IS LOADING'

;NAME

; Loading adrs

;BYTE SIZE

;START SECTOR

; MAKE READY FD'

; FD:LOADING ERROR'

; IPL MASTER FLAG

```

01 E4CF 524F      BOOT:  DEFB  0
02 E4D1 00      DEFJ  0000H
03 E4D2 0000    DEFJ  0100H
04 E4D4 0001    DEFJ  0PRO
05 E4D6 00CF    DEFJ  0
06 E4D8 0000    DEFJ  0
07 E4DA 0000    DEFJ  0
08 E4DC        ; READY CHECK
09 E4DC        ;
10 E4DC        ;
11 E4DC        ;
12 E4DC 3AF5CE  LD      A,(MTFG)
13 E4DF 0F      RRCA
14 E4E0 D417E5  CALL    NC,FDMTON
15 E4E3 D07E00  LD      A,(IX+0)
16 E4E6 F684    OR      84H
17 E4E8 D30C    OUT    (DM),A
18 E4EA AF      XOR    A
19 E4EB 32F4CE  LD      (CMD),A
20 E4EE CDBFES  LD      DLY60M
21 E4F1 210000  LD      HL,0
22 E4F4 2B      DEC    HL
23 E4F5 7C      LD      A,H
24 E4F6 B5      OR      L
25 E4F7 2819   JR      Z,REDY1
26 E4F9 DB08   IN      A,(CR)
27 E4FB 2F      CPL
28 E4FC 07      RLCA
29 E4FD 38F5   JR      C,REDY0
30 E4FF D04E00  JR      C,(IX+0)
31 E502 21F6CE  LD      HL,CLBF0
32 E505 0600    LD      B,0
33 E507 09      ADD    HL,BC
34 E508 CB46   BIT    0,(HL)
35 E50A 2005   JR      NZ,REDY2
36 E50C CD48E5  RCLB
37 E50F CB06   SET    0,(HL)
38 E511 C9     RET
39 E512 3E32   REDY1: LD      A,50
40 E514 C390E6  ERJMP
41 E517        ; MOTOR ON
42 E517        ;
43 E517        ;
44 E517        ;
45 E517 3E80   FDMTON: ENT    L
46 E519 D3DC   OUT    (DM),A
47 E51B 0610   LD      B,16
48 E51D CDBFE6  LD      DLY60M
49 E520 10FB   DJNZ   MTD1
50 E522 3E01   LD      A,1
51 E524 32F5CE  LD      (MTFG),A
52 E527 C9     RET
53 E528        ; SEEK TREATMENT
54 E528        ;
55 E528        ;
56 E528        ;
57 E528 3E1B   SEEK:  ENT    L
58 E52A C055E5  LD      A,1BH
59 E52D E699   CALL    CHOOT1
60 E52F C9     AND    99H
RET

01 E530        ; DRIVE NO.
02 E530        ;
03 E530        ;
04 E530        ;
05 E530 F5     ;SECTOR ADDR.
06 E531 CDB9E6  ;IFM BYTE SIZE
07 E534 AF      ;IFM LOADING ADDR.
08 E535 D30C    ;IX+7,8
09 E537 32F6CE  ;IX+9,10
10 E53A 32F7CE  LD      D30C
11 E53D 32F8CE  LD      D30C
12 E540 32F9CE  LD      D30C
13 E543 32F5CE  LD      D30C
14 E546 F1     LD      D30C
15 E547 C9     LD      D30C
16 E548        LD      D30C
17 E548        LD      D30C
18 E548        LD      D30C
19 E548 3E08    LD      D30C
20 E54A C055E5  LD      D30C
21 E54D E685    LD      D30C
22 E54F EE04    LD      D30C
23 E551 C8     LD      D30C
24 E552 C361E6  LD      D30C
25 E555        LD      D30C
26 E555        LD      D30C
27 E555        LD      D30C
28 E555 32F4CE  LD      D30C
29 E558 2F     LD      D30C
30 E559 D308    LD      D30C
31 E55B C068E5  LD      D30C
32 E55E CDBFE6  LD      D30C
33 E561 DB08    LD      D30C
34 E563 2F     LD      D30C
35 E564 32F8CE  LD      D30C
36 E567 C9     LD      D30C
37 E568        LD      D30C
38 E568        LD      D30C
39 E568        LD      D30C
40 E568 D5     LD      D30C
41 E569 E5     LD      D30C
42 E56A C0A1E5  LD      D30C
43 E56D 210000  LD      D30C
44 E570 2B     LD      D30C
45 E571 7C     LD      D30C
46 E572 B5     LD      D30C
47 E573 2808   LD      D30C
48 E575 0B08   LD      D30C
49 E577 0F     LD      D30C
50 E578 30F6   LD      D30C
51 E57A E1     LD      D30C
52 E57B D1     LD      D30C
53 E57C C9     LD      D30C
54 E57D        LD      D30C
55 E57D        LD      D30C
56 E57D 1D     LD      D30C
57 E57E 20ED   LD      D30C
58 E580 3E29   LD      D30C
59 E582 E1     LD      D30C
60 E583 D1     LD      D30C

```



```

01 E643 3E01      LD      A,1
02 E645 D07708    LD      (IX+8),A
03 E648 D03407    INC      (IX+7)
04 E64B 7A        LD      A,0
05 E64C B7        OR      A
06 E64D C9        RET
07 E64E          ;
08 E64E          ; COMMAND OUT & WAIT
09 E64E          ;
10 E64E 32F4CE    CMDOT2: LD      (CMD),A
11 E651 2F        CPL
12 E652 D308      OUT      (CR),A
13 E654 CD87E5    CALL     BSYOFF
14 E657 C9        RET
15 E658          ;
16 E658          ; FORCE INTERRUPT
17 E658          ;
18 E658          ;
19 E658 3ED8      INTER: LD      A,D8H
20 E65A 2F        CPL
21 E65B D308      OUT      (CR),A
22 E65D CD88E5    CALL     BSYON
23 E660 C9        RET
24 E661          ;
25 E661          ; STATUS CHECK
26 E661          ;
27 E661          ;
28 E661 3AF4CE    ERROR: LD      A,(CMD)
29 E664 FE0B      CP      0BH
30 E666 281B      JR      Z,ERCK1
31 E668 FE1B      CP      1BH
32 E66A 2817      JR      Z,ERCK1
33 E66C FEF4      CP      F4H
34 E66E 2813      JR      Z,ERCK1
35 E670 3AFBCE    LD      A,(STAFG)
36 E673 CB7F      BIT      7,A
37 E675 2017      JR      NZ,ERRET
38 E677 CB77      BIT      6,A
39 E679 200F      JR      NZ,ERRET1
40 E67B CB67      BIT      4,A
41 E67D 3E36      LD      A,54
42 E67F 200F      JR      NZ,ERJMP
43 E681 1807      JR      ERRET1
44 E683 3AFBCE    ERCK1: LD      A,(STAFG)
45 E686 CB7F      BIT      7,A
46 E688 2004      JR      NZ,ERRET
47 E68A 3E29      LD      A,41
48 E68C 1802      JR      ERJMP
49 E68E 3E32      ERRET: LD      A,50
50 E690 CD30E5    ERJMP: CALL     MOFF
51 E693 C3B6E4    JP      ERTRT
52 E696          ;
53 E696          ;
54 E696          ; SECTOR TO TRACK & SECTOR CONVERT
55 E696          ;
56 E696 0600      CNVRT: LD      B,0
57 E698 111000    LD      DE,16
58 E69B D06E01    LD      L,(IX+1)
59 E69E D06602    LD      H,(IX+2)
60 E6A1 AF        XOR      A

```

```

01 E6A2 ED52      TRANS0: SBC      HL,DE
02 E6A4 3803      JR      C,TRANS1
03 E6A6 04        INC      B
04 E6A7 18F9      JR      TRANS0
05 E6A9 19        TRANS1: ADD     HL,DE
06 E6AA 60        LD      H,B
07 E6AB 2C        INC      L
08 E6AC D07409    LD      (IX+9),H
09 E6AF D0750A    LD      (IX+10),L
10 E6B2 C9        RET
11 E6B3          ;
12 E6B3          ; TIME DELAY ( 1M & 60M & 80U )
13 E6B3          ;
14 E6B3          ;
15 E6B3          ; DLY80U: ENT
16 E6B3 D5        PUSH     DE
17 E6B4 110F00    LD      DE,15
18 E6B7 180A      JR      DLYT
19 E6B9          ;
20 E6B9 D5        PUSH     DE
21 E6BA 11A000    LD      DE,160
22 E6BD 1804      JR      DLYT
23 E6BF          ;
24 E6BF D5        PUSH     DE
25 E6C0 112620    LD      DE,8230
26 E6C3 1B        DEC      DE
27 E6C4 7B        LD      A,E
28 E6C5 B2        OR      D
29 E6C6 20FB      JR      NZ,DLYT
30 E6C8 D1        POP      DE
31 E6C9 C9        RET
32 E6CA          ; FLOPPY WORK AREA
33 E6CA          ;
34 E6CA          ;
35 E6CA P         CMD:      EQU     BPARA+11
36 E6CA P         MTFG:     EQU     CMD+1
37 E6CA P         CLBF0:    EQU     MTFG+1
38 E6CA P         CLBF1:    EQU     CLBF0+1
39 E6CA P         CLBF2:    EQU     CLBF1+1
40 E6CA P         CLBF3:    EQU     CLBF2+1
41 E6CA P         VRFCNT:   EQU     CLBF3+1
42 E6CA P         STAFG:    EQU     VRFCNT+1
43 E6CA          ;
44 E6CA          ; SKP      H

```

```

01 E6CA      ;
02 E6CA      ;=====
03 E6CA      ;
04 E6CA      ; 1R12(Static ram board 32Kbyte)
05 E6CA      ; Read/Write routine
06 E6CA      ;=====
07 E6CA      ;
08 E6CA      ;
09 1108      ; COMNT: EQU 1108H
10 E6CA      ; 1R12 Read sub.
11 E6CA      ;
12 E6CA      ;
13 E6CA      ; 1R12RD: ENT 1R12R0
14 E6CA      ; CALL NZ,PERX
15 E6CD      ; JP
16 E6D0      ;
17 E6D0      ; Exec load file
18 E6D0      ;
19 E6D0      ; EXRAM: ENT BC,0000H ; Default code (SRAM)
20 E6D0      ; LD
21 E6D3      ; EXX
22 E6D4      ; LD HL,SIZE
23 E6D7      ; JP EXF
24 E6DA      ;
25 E6DA      ; 1R12R0: ENT
26 E6DA      ;
27 E6DA      ; Information sum check
28 E6DA      ; CALL CHECK
29 E6DA      ; RET NZ
30 E6D0      ;
31 E6DE      ; Counter reset
32 E6DE      ;
33 E6DE      ; IN A,(C)
34 E6DE      ;
35 E6E0      ; Read information block (9Byte)
36 E6E0      ;
37 E6E0      ; INC C
38 E6E0      ; LD HL,SIZE
39 E6E1      ; LD B,09H
40 E6E4      ; INIR
41 E6E6      ;
42 E6E8      ; Read data block
43 E6E8      ;
44 E6E8      ; LD DE,(SIZE)
45 E6E8      ; LD HL,1200H
46 E6EC      ; LD A,E
47 E6F0      ; OR A
48 E6F0      ; JR Z,+5
49 E6F1      ; LD B,A
50 E6F3      ; INIR
51 E6F4      ; LD B,00H
52 E6F6      ; DEC D
53 E6F8      ; JP P,-5
54 E6F9      ;
55 E6FC      ; Data block's sum check
56 E6FC      ;
57 E6FC      ; LD DE,1200H ; Data adrs
58 E6FC      ; LD BC,(SIZE) ; BCSize
59 E6FF      ; CALL SUM ; HLCheck sum
60 E703      ;

```

```

01 E706      ; 280 ASSEMBLER 27-004C 03.06.85
02 E70A      ; LD DE,(CUMN1)
03 E70B      ; OR A
04 E70D      ; SBC HL,DE
05 E70E      ; RET
06 E70E      ;
07 E70E      ; sum check
08 E70E      ;
09 E70E      ; IN BC=Size
10 E70E      ; DE=Data adr
11 E70E      ; EXIT HL=Check sum
12 E70E      ;
13 E70F      ; SUM: EXX
14 E712      ; LD HL,0
15 E714      ; LD C,8
16 E715      ; EXX
17 E716      ; LD A,B
18 E717      ; OR C
19 E719      ; JR Z,SUMCK2
20 E71A      ; LD A,(DE)
21 E71B      ; LD B,C
22 E71C      ; RLCA
23 E71D      ; JR NC,+3
24 E71F      ; INC HL
25 E720      ; DJNZ SUMCK3
26 E722      ; EXX
27 E723      ; INC DE
28 E724      ; DEC BC
29 E725      ; JR SUMCK1
30 E727      ; EXX
31 E728      ; RET
32 E729      ;
33 E729      ; Information's sum check
34 E729      ;
35 E729      ; CHECK: IN A,(C)
36 E72B      ; LD B,08H
37 E72D      ; LD D,00H
38 E72F      ; INC C
39 E730      ; INC A,(C)
40 E732      ; PUSH BC
41 E733      ; LD B,08H
42 E735      ; RLCA
43 E736      ; JR NC,+3
44 E738      ; INC D
45 E739      ; DJNZ CK2
46 E73B      ; LD A,D
47 E73C      ; POP BC
48 E73D      ; LD D,A
49 E73E      ; DJNZ CK1
50 E740      ; IN A,(C)
51 E742      ; DEC C
52 E743      ; CP D
53 E744      ; RET
54 E745      ;
55 E745      ; =====
56 E745      ; 1R12 Write sub.
57 E745      ;
58 E745      ; =====
59 E745      ;
60 E745      ;

```



```

** Z80 ASSEMBLER 27-004C <92-504M-V1.0C> PAGE 35      03.06.85
01 E7DF AF      12NSET: XOR  A      ; Z flag set
02 E7E0 37      SCF                ; Not set
03 E7E1 C9
04 E7E2
05 E7E2
06 E7E2
07 E7E2
08 E7E2 52      ; Message table
09 E7E3 A1      MSGW01: DEFM  'R'
10 E7E4 B3      DEFDB  A1H
11 E7E5 9F      DEFDB  B3H
12 E7E6 A1      DEFDB  9FH
13 E7E7 9D      DEFDB  A1H
14 E7E8 9C      DEFDB  9DH
15 E7E9 20      DEFDB  9CH
16 E7EA 9E      DEFDB  20H
17 E7EB 9D      DEFDB  9EH
18 E7EC B7      DEFDB  9DH
19 E7ED 97      DEFDB  B7H
20 E7EE 2E      DEFDB  97H
21 E7EF 0D      DEFDB  2EH
22 E7F0
23 E7F0 4D      ; MSGW02: DEFM  'M'
24 E7F1 A1      DEFDB  A1H
25 E7F2 A4      DEFDB  A4H
26 E7F3 96      DEFDB  96H
27 E7F4 92      DEFDB  92H
28 E7F5 9D      DEFDB  9DH
29 E7F6 20      DEFDB  20H
30 E7F7 96      DEFDB  96H
31 E7F8 A1      DEFDB  A1H
32 E7F9 9E      DEFDB  9EH
33 E7FA 92      DEFDB  92H
34 E7FB 20      DEFDB  20H
35 E7FC A4      DEFDB  A4H
36 E7FD 92      DEFDB  92H
37 E7FE 96      DEFDB  96H
38 E7FF 0D      DEFDB  0DH
39 E800
40 E800

; SKP  H

12NSET: XOR  A      ; Z flag set
SCF                ; Not set

; Message table
MSGW01: DEFM  'R'
DEFDB  A1H
DEFDB  B3H
DEFDB  9FH
DEFDB  A1H
DEFDB  9DH
DEFDB  9CH
DEFDB  20H
DEFDB  9EH
DEFDB  9DH
DEFDB  B7H
DEFDB  97H
DEFDB  2EH
DEFDB  0DH

; MSGW02: DEFM  'M'
DEFDB  A1H
DEFDB  A4H
DEFDB  96H
DEFDB  92H
DEFDB  9DH
DEFDB  20H
DEFDB  96H
DEFDB  A1H
DEFDB  9EH
DEFDB  92H
DEFDB  20H
DEFDB  A4H
DEFDB  92H
DEFDB  96H
DEFDB  0DH

; SKP  H

12NSET: XOR  A      ; Z flag set
SCF                ; Not set

; Ramcard prog.'
; A
; M
; C
; A
; R
; D
; P
; R
; O
; G
; .
; Master tape set'
; A
; S
; T
; E
; R
; T
; A
; P
; E
; S
; E
; T

01 E800      ORG  E800H
02 E800
03 E800
04 E800
05 E800
06 E800
07 E800
08 E800
09 E800
10 E800
11 E800
12 E800
13 E800
14 E800
15 E800
16 11A3      EQU  11A3H
17 00CE      EQU  00CEH
18 00CD      EQU  00CDH
19 00CC      EQU  00CCH
20 E800
21 E800
22 E800
23 E800      BEGIN0: ENT  NOP
24 E800      JP  BEGIN
25 E801 C313E8  BGIN00: ENT  ST1X
26 E804      JP  LCM12: ENT  LDCMT
27 E804      C35EEA      JP  SCH12: ENT  SACMT
28 E807      C370EB      JP  VCM12: ENT  VFCMT
29 E807      C370EB      JP  QD10: ENT  QDIOS
30 E80A      C3E8EB
31 E80A      C3E8EB
32 E80D      C309EC
33 E80D      C310E0
34 E810
35 E810
36 E813
37 E813
38 E813
39 E813      F3
40 E814      ED56
41 E816      3E08
42 E818      D3CE
43 E81A      3E01
44 E81C      D3C0
45 E81E      D3CC
46 E820      D3E4
47 E822      31F010
48 E825      CD3E07
49 E828      AF
50 E829      110000
51 E82C      CD0803
52 E82F
53 E82F
54 E82F
55 E82F      012500
56 E832      CD14E4
57 E835
58 E835
59 E835
60 E835      01FC04

; Interrupt mode 1
; '0000 1000'
; 700 Mode
; A,01H
; '0000 0001'
; Read format reg.
; Write format reg.
; Initial Bank
; Stack set (10F0H)
; 8255 MODE SET
; 8253 Mode set

; Wait (25.342ms) for Printer
; LD  BC,25H
; CALL TIMW
; Z80 PIO Initialize
; P10A: LD  BC,04FCH
; Port=A (C;port B;cont)

```

```

01 E838 219AEE LD HL,PIOWDA ; Pio initialize data
02 E83B EDB3 OTIR ;
03 E83D 01FD04 LD BC,04FDH ; Port=B (C:port B:cont)
04 E840 EDB3 OTIR ;
; SIO Initialize
05 E842 LD A,01H
06 E842 XOR A ; Key in silent
07 E842 AND 2 ;
08 E842 3E01 LD A,(E0H) ; CG & Vram
09 E844 D3F7 OUT (SI0BC),A ; Destination
10 E846 AF XOR A ; Sorce
11 E847 D3F7 OUT (SI0BC),A ; Size
12 E849 ;
13 E849 ; (CTRL+Reset) Check
14 E849 ;
15 E849 CD320A CALL ?BRK ; Key check
16 E84C 3014 JR NC,PSG0 ;
17 E84E FE20 CP 20H ;
18 E850 C262E8 JP NZ,PSG0 ; Key is 'CTRL' Key
19 E853 DBCE IN A,(CEH) ; Mode switch
20 E855 E602 AND 2 ; bit 1 check
21 E857 2806 JR Z,CMY00 ; 700 Mode
22 E859 AF XOR A ;
23 E85A D3CE OUT (CEH),A ; 800 Mode
24 E85C DDE1E8 CALL PLTST ; Pallet set
25 E85F C35B00 CMY00: JP CMY0 ; Bank & Jump
; PSG Reset
26 E862 ;
27 E862 ; PSG Reset
28 E862 ;
29 E862 0604 LD B,04H
30 E864 3E9F LD A,9FH
31 E866 D3F2 OUT (F2H),A
32 E868 C620 ADD A,20H
33 E86A 10FA DJNZ PSGSET
; Sound & Interrupt mask reset
34 E86C ;
35 E86C ; Sound & Interrupt mask reset
36 E86C ;
37 E86C 3E01 LD A,01H
38 E86E 3203E0 LD (E003H),A
39 E871 3E05 LD A,05H
40 E873 3203E0 LD (E003H),A
; BG10:
41 E876 ;
42 E876 ;
43 E876 ENT B,FFH
44 E876 06FF LD HL,NAME
45 E878 21F110 LD ?CLER
46 E87B CDD80F CALL A,16H
47 E87E 3E16 LD A,16H
48 E880 CD1200 CALL PRNT
49 E883 3E71 LD A,71H
50 E885 2100D8 LD HL,D800H
51 E888 C00509 CALL #CLR8
52 E88B 218D03 LD HL,TIMIN
53 E88E 3EC3 LD A,C3H
54 E890 323810 LD (1038H),A
55 E893 223910 LD (1039H),HL
56 E896 3E04 LD A,04H
57 E898 329E11 LD (TEMPU),A
58 E89E CDBE02 CALL MLDSP
59 E89E C00900 CALL NL
60 E8A1 C07705 CALL ?BEL

01 E8A4 3E01 LD A,01H
02 E8A6 329D11 LD (S0RK),A
03 E8A9 ;
04 E8A9 ; PCG CG ROM
05 E8A9 ;
06 E8A9 DBE0 IN A,(E0H)
07 E8AB 1100C0 LD DE,C000H
08 E8AE 210010 LD HL,1000H
09 E8B1 010010 LD BC,1000H
10 E8B4 EDB0 LDIR
11 E8B6 DBE1 IN A,(E1H)
12 E8B8 ;
13 E8B8 ; Key check
14 E8B8 ;
15 E8B8 CD1B00 CALL GETKY
16 E8BB FE40 CP 'M' ;Key check
17 E8BD 287C JR Z,MON ; Monitor cmd.
18 E8BF FE51 CP 'g' ;
19 E8C1 2858 JR Z,KSPGB ; Quick disk
20 E8C3 FE43 CP 'C' ;
21 E8C5 287E JR Z,CSS ; Cassette tape
22 E8C7 ;
23 E8C7 ; Floppy check & boot
24 E8C7 ;
25 E8C7 CD05E8 CALL FDC0 ; FD i/o check
26 E8CA 2026 JR NZ,R12C&B
27 E8CC CD59EA CALL DISCLR
28 E8CF C04AE4 CALL FDX
29 E8D2 C34AE4 JP ERRNG
30 E8D5 ENT ;
31 E8D5 3EA5 LD A,A5H
32 E8D7 47 LD B,A
33 E8D8 D3D9 OUT (D9H),A
34 E8DA CDB3E6 CALL DLY80U
35 E8DD DBD9 IN A,(D9H)
36 E8DF B8 CP B
37 E8E0 C9 RET
; Pallet Reg. & Border Reg. set
38 E8E1 ;
39 E8E1 ; PL10-3 Black
40 E8E1 ; Border Black
41 E8E1 ;
42 E8E1 ;
43 E8E1 ;
44 E8E1 ;
45 E8E1 ENT ;
46 E8E2 01F005 PUSH HL
47 E8E5 21A2EE LD BC,05F0H
48 E8E8 EDB3 LD HL,PL1DT
49 E8EA AF OTIR
50 E8EB 01CF06 XOR A
51 E8EE ED79 LD BC,06CFH
52 E8F0 E1 OUT (C),A
53 E8F1 C9 POP HL
54 E8F2 ;
55 E8F2 ; MZ-1R12 check & boot
56 E8F2 ;
57 E8F2 0EF8 LD C,F8H
58 E8F4 CDBAE7 CALL R12CKI
59 E8F7 DAB7E9 JP C,0BT
60 E8FA C0DAE6 CALL IRL2R0 ; Set....cv=0',z='1'

```

```

** Z80 ASSEMBLER 22-004C <92-504H-V1.0C> PAGE 39 03.06.85 ** Z80 ASSEMBLER 22-004C 03.06.85 **
01 E8FD 113AEE LD DE,12ERMG ; 'SRAM:Check sum err'
02 E900 C234EA JP NZ,ERRMG
03 E903 C3D0E6 JP EXRAM
04 E906 ;
05 E906 ; Device selection
06 E906 ;
07 E906 C043EA SMSG: CALL CLLET
08 E909 C006EA SELMSG: CALL SELMSG
09 E90C C091F1 CALL NKEYJT
10 E90F C013EB KSCAN0: CALL SIOCK
11 E912 200A JR NZ,KSCAN1
12 E914 3E02 LD A,02H
13 E916 C015F1 CALL SRCK0
14 E919 FE7F CP 7FH
15 E91B CAB7E9 KSPQB: JP Z,0BT
16 E91E 3E04 KSCAN1: LD A,04H
17 E920 C015F1 CALL SRCK0
18 E923 FE0F CP DFH
19 E925 281E JR Z,CSS
20 E927 4F LD C,A
21 E928 C005E8 CALL FDCC&
22 E92B 2005 JR NZ,KSCAN2
23 E92D 79 LD A,C
24 E92E FEFB CP FBH
25 E930 289A JR Z,FDB
26 E932 3E03 KSCAN2: LD A,03H
27 E934 C015F1 CALL SRCK0
28 E937 FEF7 CP F7H
29 E939 20D4 JR NZ,KSCAN0
30 E93B ;
31 E93B ; Monitor jump
32 E93B ;
33 E93B C059EA MON: CALL DISCLR
34 E93E 114DEE LD DE,MONMSG
35 E941 DF RST 3
36 E942 C35EEA JP STIX
37 E945 ;
38 E945 ; =====
39 E945 ; Cassette boot-up
40 E945 ;
41 E945 ; =====
42 E945 ;
43 E945 ; CSS: ENT
44 E945 ;
45 E945 ; Motor check
46 E945 ;
47 E945 ;
48 E945 2102E0 LD HL,E002H
49 E948 7E LD A,(HL)
50 E949 E610 AND 10H
51 E94B 2025 JR NZ,DSILF
52 E94D 23 INC HL
53 E94E 3E06 LD A,06H
54 E950 77 LD (HL),A
55 E951 3C INC A
56 E952 77 LD (HL),A
57 E953 2B DEC HL
58 E954 7E LD A,(HL)
59 E955 E610 AND 10H
60 E957 2019 JR NZ,DSILF

01 E959 ; Displ 'Make ready CMT'
02 E959 ;
03 E959 C059EA CALL DISCLR
04 E959 C00600 CALL LETNL
05 E95C C00600 CALL LETNL
06 E95F C00600 LD DE,IPLM1
07 E962 1198ED LD DE,IPLM1
08 E965 C04EEA CALL MSGLET
09 E968 ; Wait motor on
10 E968 ;
11 E968 ; WMOD1: CALL BRKEY
12 E968 C01E00 JR Z,TPERX0
13 E96B 283F LD A,(HL)
14 E96D 7E AND 10H
15 E96E E610 LD Z,WMOD1
16 E970 28F6 JR Z,WMOD1
17 E972 ; Displ 'IPL is looking for'
18 E972 ;
19 E972 ; DSILF: CALL DISCLR
20 E972 C00600 CALL LETNL
21 E975 C00600 LD DE,IPLM4
22 E978 11C3ED LD RST 3
23 E97B DF ; Read information block
24 E97C ;
25 E97C ; CALL RDINF
26 E97C JP C,TPERX
27 E97C ; Displ 'IPL is loading...'
28 E97F DAAAE9 ;
29 E982 ;
30 E982 ; CALL DISCLR
31 E982 LD DE,IPLM0
32 E982 RST 3
33 E985 1188ED LD DE,NAME
34 E988 DF LD 3
35 E989 11F110 RST 3
36 E98C DF ; Load adrs set
37 E98D ;
38 E98D ; LD HL,(DTADR)
39 E98D EXX
40 E98D D9 LD HL,1200H
41 E990 D9 LD (DTADR),HL
42 E991 210012 LD (DTADR),HL
43 E994 220411 ; Read data block
44 E997 ;
45 E997 ; ROTB0: CALL RDDAT
46 E997 JP C,TPERX
47 E997 ; Exec load file
48 E99A DAAAE9 ;
49 E99D ; EXCHT: LD BC,0100H
50 E99D ;
51 E99D ; LD (DTADR),HL
52 E99D 010001 LD HL,SIZE
53 E9A0 D9 LD JP EXP
54 E9A1 220411 ; Tape error
55 E9A4 210211 ;
56 E9A7 C3FCEC ;
57 E9AA ;
58 E9AA ; TPERX: CP 02H
59 E9AA ; BREAK?
60 E9AA FE02 ;

```

```

01 E9AC 1198ED TPERX0: LD DE,IPLM1
02 E9AF 2803 JR Z,TPERJP
03 E9B1 1104EE LD DE,ERRM2
04 E9B4 C344EA TPERJP: JP ERRMG
05 E9B7 =====
06 E9B7 ;
07 E9B7 ; Quick disk boot-up
08 E9B7 ;
09 E9B7 ;
10 E9B7 ;
11 E9B7 ;
12 E9B7 ;
13 E9B7 C013EB QBT: ENT
14 E9BA 3E02 CALL SIOCK
15 E9BC 20EC LD A,02H
16 E9BE C0CEEE JR NZ,TPERX
17 E9C1 C027EF CALL IOPRS
18 E9C4 11A7ED CALL QORCK
19 E9C7 3668 LD DE,IPLM2
20 E9C9 C059EA JR C,ERRMG
21 E9CC 3E00 CALL DISCLR
22 E9CE 32A311 LD A,00H
23 E9D1 C05FF2 LD (BUFER),A
24 E9D4 CALL HOPCL
25 E9D4 ;
26 E9D4 ; Error return set
27 E9D4 3E01 LD A,01H
28 E9D6 323A11 LD (QDCPA),A
29 E9D9 2104EA LD HL,QDLOER
30 E9DC 31EE10 LD SP,10EEH
31 E9DF E3 EX (SP),HL
32 E9E0 ;
33 E9E0 ;
34 E9E0 C0F7EE QBFLSH: CALL FILSCH
35 E9E3 D0B2F2 JP C,QERTRT
36 E9E6 3AF010 LD A,(ATRB)
37 E9E9 FE01 CP OBJCD
38 E9EB 1127EE LD DE,ERRH4
39 E9EE 200A JR NZ,QDFMER
40 E9F0 ;
41 E9F0 ; Quick disk boot
42 E9F0 ;
43 E9F0 1188ED LD DE,IPLM0
44 E9F3 DF RST 3
45 E9F4 C3C2EE JP DSFLNA
46 E9F7 11A7ED QDNTR: LD DE,IPLM2
47 E9FA D5 QDFMER: PUSH DE
48 E9FB 3E06 LD A,06H
49 E9FD 323011 LD (QDPA),A
50 EA00 CD10E0 CALL QDIOS
51 EA03 01 POP DE
52 EA04 182E QDLOER: JR ERRMG
53 EA06 ;
54 EA06 ;
55 EA06 ; Disp message
56 EA06 ;
57 EA06 ;
58 EA06 ;
59 EA06 ;
60 EA06 ;

01 EA06 ; Selection message
02 EA06 ;
03 EA06 C00600 SELMSG: CALL LETNL
04 EA09 1143ED LD DE,SELM0
05 EA0C C04EEA CALL MSGLET
06 EA0F C00600 LETNL
07 EA12 C005E8 CALL FDCCK
08 EA15 2006 JR NZ,SELM00
09 EA17 1153ED LD DE,SELM1
10 EA1A C04EEA SELM00: CALL SIOCK
11 EA1D C013EB JR NZ,SELM01
12 EA20 2006 LD DE,SELM2
13 EA22 1161ED CALL MSGLET
14 EA25 C04EEA SELM01: LD DE,SELM3
15 EA28 116EED CALL MSGLET
16 EA2B C04EEA LD DE,SELM4
17 EA2E 117EED JP MSGLET
18 EA31 C30000 ;
19 EA34 ; Error message
20 EA34 ;
21 EA34 ;
22 EA34 ; ERRMG: CALL DISCLR
23 EA34 C059EA CALL LETNL
24 EA37 C00600 CALL LETNL
25 EA3A C00600 CALL MSGLET
26 EA3D C04EEA JP SELMG0
27 EA40 C309E9 ;
28 EA43 ; Disp clear 3-let
29 EA43 ;
30 EA43 ; CLLET: CALL DISCLR
31 EA43 C059EA LD B,03H
32 EA46 0603 LET: CALL LETNL
33 EA48 C00600 LD DJNZ LET
34 EA4B 10FB RET
35 EA4D C9 ;
36 EA4E ; Letnl-message
37 EA4E ;
38 EA4E ; MSGLET: LD B,0CH
39 EA4E 060C MSGPS: CALL PRINTS
40 EA50 C00C00 DJNZ MSGPS
41 EA53 10FB RST 3
42 EA55 DF JP LETNL
43 EA56 C30000 ;
44 EA59 ; Disp clear
45 EA59 ;
46 EA59 ; DISCLR: LD A,C6H
47 EA59 3EC6 JP ?DPCT
48 EA59 3EC6 ;
49 EA5B C3DC0D ;
50 EA5E ;
51 EA5E ;
52 EA5E ;
53 EA5E ; Monitor command
54 EA5E ;
55 EA5E ;
56 EA5E ;
57 EA5E ; Get command routine
58 EA5E ;
59 EA5E ;
60 EA5E ; STIX: ENT

```

```

** Z80 ASSEMBLER 22-004C \92-504M-VI-00\ PAGE 4 0.00.85
01 EA5E 31F010 LD SP,SP
02 EA61 C0D900 CALL NL
03 EA64 3E2A LD A,2AH
04 EA66 C01200 CALL PRNT
05 EA69 11A311 LD DE,BUFER
06 EA6C C00300 CALL GETL
07 EA6F 1A LD A,(DE)
08 EA70 13 INC DE
09 EA71 FE00 CP Z,STIX
10 EA73 28E9 JR J
11 EA75 FE4A CP J
12 EA77 2830 JR Z,GOTOX
13 EA79 FE4C CP Z,LOADX
14 EA7B CA4CEB JP Z,FDCK
15 EA7E FE46 CP 'B'
16 EA80 287E JR Z,SGX
17 EA82 FE42 CP 'M'
18 EA84 CA1EEC JP Z,MCORX
19 EA87 FE4D CP 'S'
20 EA89 CA7BEB JP Z,SAVEX
21 EA8C FE53 CP 'V'
22 EA8E CAEEB JP Z,VRFYX
23 EA91 FE56 CP 'D'
24 EA93 CA00EC JP Z,DUMPX
25 EA96 FE44 CP 'Q'
26 EA98 CA29EC JP Z,QUICK
27 EA9B FE51 CP 'E'
28 EA9D 2816 JR Z,RAMBRD
29 EA9F FE45 CP 'G'
30 EA01 2842 JR Z,CAL0
31 EA03 FE47 CP 'F'
32 EA05 2806 JR ST2X
33 EA07 18C6
34 EA09
35 EA09
36 EAA9
37 EAA9 CD42EB
38 EAAE E9
39 EAAD
40 EAA0
41 EAA0
42 EAA0 CD42EB
43 EAB0 CDACEA
44 EAB3 18A9
45 EAB5
46 EAB5
47 EAB5
48 EAB5 CD13EB
49 EAB8 20A4 JR NZ,STIX
50 EABA CDECEE CALL IOFRS
51 EABD 210000 LD HL,0000H
52 EAC0 223A11 LD (00CPA),HL
53 EAC3 1A LD A,(DE)
54 EAC4 FE4C CP 'L'
55 EAC6 CAA7EE JP Z,QL
56 EAC9 FE53 CP 'S'
57 EACB CA2EEF JP Z,OS
58 EACE FE43 CP 'C'
59 EAD0 CA2CF1 JP Z,OC
60 EAD3 FE46 CP 'F'

; "
; JUMP
; Load cmt
; Floppy boot
; Bell
; Memory correction
; Save CMT
; Verify
; Dump memory
; Quick disk cmd.
; Exit rambrd(1R12) cmd.
; Call cmd.

; Sio hard check
; IOCS frag reset
; Frag reset
; Load QD
; Save QD
; Copy QD
; Format QD

ST2X:
LD SP,SP
CALL NL
LD A,2AH
PRNT
DE,BUFER
GETL
LD A,(DE)
INC DE
CP Z,STIX
JR J
CP J
JR Z,GOTOX
CP Z,LOADX
JP Z,FDCK
CP 'B'
JR Z,SGX
CP 'M'
JP Z,MCORX
CP 'S'
JP Z,SAVEX
CP 'V'
JP Z,VRFYX
CP 'D'
JP Z,DUMPX
CP 'Q'
JP Z,QUICK
CP 'E'
JR Z,RAMBRD
CP 'G'
JR Z,CAL0
CP 'F'
JR ST2X

;
;
; Jump command
GOTOX: CALL HEXIYX
GOTO: JP (HL)
;
; Call command
CAL0: CALL HEXIYX
CALL GOTO
JR STIX
;
; Quick disk CMD.
QUICK: CALL STOCK
JR NZ,STIX
CALL IOFRS
LD HL,0000H
LD (00CPA),HL
LD A,(DE)
CP 'L'
JP Z,QL
CP 'S'
JP Z,OS
CP 'C'
JP Z,OC
CP 'F'

; Xfer (CMTGDU)
; Directry
; 1R12 boot-up
; 1R12 Write
; FD i/o check
; Sio hard check
; SIOCK: LD A,02H
; OUT (SIOBC),A
; LD A,ASH
; OUT (SIOBC),A
; LD A,02H
; OUT (SIOBC),A
; IN A,(SIOBC)
; ASH
; RET
; Error (loading)
?ERX: ENT LD DE,MGBRK
CP 02H
JR Z,?ERX0
LD DE,MSGE1
?ERX0: CALL NL
RST 3
JR STIX
; Getline & break in check
; exit break in then jump (STIX)
; acc=top of line data
; BGETLX: ENT (SP),HL
EX BC
POP DE,BUFER
LD CALL
GETL

```



```

01 EBE8      ENT      CALL      ?URI      ; Write information
02 EBE8      CALL      JR          C,SAVJE
03 EBE8      JR          HL,1200H
04 EBE8      LD          ?JRD      ; Write data
05 EBF0      LD          ?JRD      ; Write data
06 EBF3      CALL      C,SAVJE
07 EBF6      JR          NL
08 EBF8      CALL      DE,MSGOK
09 EBF8      RST      3
10 EBF8      RET
11 EBF8      RET
12 EC00      ;
13 EC00      ; Verifying command
14 EC00      ;
15 EC00      ;
16 EC00      ;
17 EC00      VRFYX: ENT      CALL      ?VRFY      ; 'OK' MSG.
18 EC00      VRFYX: JR          C,?PERX
19 EC00      VRFYX: JR          ?JRD
20 EC00      VRFYX: JR          ?JRD
21 EC00      VRFYX: JR          ?JRD
22 EC00      VRFYX: JR          ?JRD
23 EC00      VRFYX: JR          ?JRD
24 EC00      VRFYX: JR          ?JRD
25 EC00      VRFYX: JR          ?JRD
26 EC00      VRFYX: JR          ?JRD
27 EC00      VRFYX: JR          ?JRD
28 EC00      VRFYX: JR          ?JRD
29 EC00      VRFYX: JR          ?JRD
30 EC00      VRFYX: JR          ?JRD
31 EC00      VRFYX: JR          ?JRD
32 EC00      VRFYX: JR          ?JRD
33 EC00      VRFYX: JR          ?JRD
34 EC00      VRFYX: JR          ?JRD
35 EC00      VRFYX: JR          ?JRD
36 EC00      VRFYX: JR          ?JRD
37 EC00      VRFYX: JR          ?JRD
38 EC00      VRFYX: JR          ?JRD
39 EC00      VRFYX: JR          ?JRD
40 EC00      VRFYX: JR          ?JRD
41 EC00      VRFYX: JR          ?JRD
42 EC00      VRFYX: JR          ?JRD
43 EC00      VRFYX: JR          ?JRD
44 EC00      VRFYX: JR          ?JRD
45 EC00      VRFYX: JR          ?JRD
46 EC00      VRFYX: JR          ?JRD
47 EC00      VRFYX: JR          ?JRD
48 EC00      VRFYX: JR          ?JRD
49 EC00      VRFYX: JR          ?JRD
50 EC00      VRFYX: JR          ?JRD
51 EC00      VRFYX: JR          ?JRD
52 EC00      VRFYX: JR          ?JRD
53 EC00      VRFYX: JR          ?JRD
54 EC00      VRFYX: JR          ?JRD
55 EC00      VRFYX: JR          ?JRD
56 EC00      VRFYX: JR          ?JRD
57 EC00      VRFYX: JR          ?JRD
58 EC00      VRFYX: JR          ?JRD
59 EC00      VRFYX: JR          ?JRD
60 EC00      VRFYX: JR          ?JRD

```

```

** Z80 ASSEMBLER 22-004C <9Z-504M-V1.0C> PAGE 49 03.06.85
01 ECB1 ; Space check
02 ECB1 ; FNNP1: CP ' ',FNNP2
03 ECB1 JR INC DE
04 ECB3 2004 LD A,(DE)
05 ECB5 13 JR FNNP1
06 ECB6 1A ;
07 ECB7 18F8 ; " ?
08 ECB9 ;
09 ECB9 ; FNNP2: CP 22H
10 ECB9 JR Z,FNNP4
11 ECB9 FE22 ; Trans (DE)(HL)
12 ECB8 2808 ;
13 ECB8 ; FNNP3: LD (HL),A
14 ECB8 INC HL
15 ECB8 INC B
16 ECB8 77 LD B,A,11H
17 ECBE 23 LD B,Z,FNNP
18 ECBF 04 LD B,Z,FNNP
19 ECC0 3E11 CP B
20 ECC2 B8 JR Z,FNNP
21 ECC3 28C8 ;
22 ECC5 ; Filename end check
23 ECC5 ;
24 ECC5 ; FNNP4: INC DE
25 ECC5 13 LD A,(DE)
26 ECC6 1A CP 22H
27 ECC7 FE22 JR Z,FNNP5
28 ECC9 2804 CP 00H
29 ECCB FE00 JR NZ,FNNP3
30 ECCD 20EE JR A,00H
31 ECCF 3E0D LD (HL),A
32 ECD1 77 FNNP6: LD RET
33 ECD2 C9 ;
34 ECD3 ; Input save condition
35 ECD3 ;
36 ECD3 ; TAINP: ENT DE,MSGTA
37 ECD3 LD AINP0
38 ECD3 1170EE JR AINP0
39 ECD6 1808 ENT DE,MSGEA
40 ECD8 117BEE LD AINP0
41 ECD8 1803 ENT DE,MSGXA
42 ECD8 1803 LD NL
43 ECD0 1186EE CALL 3
44 ECD0 C00900 RST 3
45 ECE0 C00900 PUSH DE
46 ECE3 DF LD DE,BUFER
47 ECE4 05 LD GETL
48 ECE5 11A311 CALL A,(DE)
49 ECE8 C00300 LD CP 1BH
50 ECEB 1A POP DE
51 ECEC FE1B JR Z,FNNP4
52 ECEE D1 JR
53 ECEF 28AE PUSH DE
54 ECF1 D5 LD DE,BUFER+10
55 ECF2 11AD11 LD HL,HEX
56 ECF5 C01004 CALL DE
57 ECF8 D1 POP DE
58 ECF9 38E5 JR C,AINP0
59 ECFB C9 RET
60 ECFC ;

** Z80 ASSEMBLER 22-004C <9Z-504M-V1.0C> PAGE 50 03.06.85
01 ECFC ; Exec file
02 ECFC ; IN HL ; Data size point
03 ECFC ; BC ; Default code
04 ECFC ;
05 ECFC ; EXF:
06 ECFC DBCE ENT A,(CEH)
07 ECFC CB4F BIT 1,A
08 ED00 2806 JR Z,70MOD
09 EB02 AF XOR A
10 ED03 D3CE OUT (CEH),A
11 ED05 CDE1E8 CALL PLTST
12 ED08 ; 70MOD: LD C,(HL)
13 ED08 4E INC HL
14 ED09 23 LD B,(HL)
15 ED0A 46 INC HL
16 ED0B 23 LD E,(HL)
17 ED0C 5E LD A,(HL)
18 ED0D 7E INC HL
19 ED0E 23 LD D,(HL)
20 ED0F 56 OR (HL)
21 ED10 B6 OR HL
22 ED11 D5 PUSH DE
23 ED12 23 INC HL
24 ED13 5E LD E,(HL)
25 ED14 B6 OR (HL)
26 ED15 23 INC HL
27 ED16 56 LD D,(HL)
28 ED17 B6 OR (HL)
29 ED18 D5 PUSH DE
30 ED19 DDE1 POP IX
31 ED1B D1 POP DE
32 ED1C 2002 JR NZ,BLKTR
33 ED1E D3E0 OUT (E0H),A
34 ED20 ; BLKTR: LD HL,1200H
35 ED20 210012 LD A,12H
36 ED23 3E12 CP D
37 ED25 BA JR C,LDDDEC
38 ED26 3806 JR NZ,LDINC
39 ED28 2014 JR A
40 ED2A AF XOR E
41 ED2B B8 CP NC,LDINC
42 ED2C 3010 JR BC
43 ED2E 08 LDDEC: DEC BC
44 ED2F E5 PUSH HL
45 ED30 210000 LD HL,0000H
46 ED33 19 ADD HL,DE
47 ED34 09 ADD HL,BC
48 ED35 E5 PUSH HL
49 ED36 D1 POP DE
50 ED37 E1 POP HL
51 ED38 09 ADD HL,BC
52 ED39 03 INC BC
53 ED3A ED88 LDDR JR
54 ED3C 1802 LDINC: LDIR
55 ED3E ED80 EXX
56 ED3F 09 JP (IX)
57 ED41 D0E9 ; BCDefault code
58 ED43 ;
59 ED43 ;
60 ED43 ;

```



```

01 EE0C A6      DEFB      A6H      'GD:L'
02 EE0D B0      DEFB      B0H      'GD:Loading error'
03 EE0E 97      DEFB      97H      'O'
04 EE0F 20      DEFB      20H      'A'
05 EE10 92      DEFB      92H      'D'
06 EE11 9D      DEFB      9DH      'I'
07 EE12 90      DEFB      90H      'N'
08 EE13 B7      DEFB      B7H      'G'
09 EE14 9D      DEFB      9DH      'E'
10 EE15 0D      DEFB      0DH      'R'
11 EE16         DEFB         'GD:L'
12 EE16 51443A4C ERRM3:      DEFM      'GD:Loading error'
13 EE1A B7      DEFB      B7H      'O'
14 EE1B A1      DEFB      A1H      'A'
15 EE1C 9C      DEFB      9CH      'D'
16 EE1D A6      DEFB      A6H      'I'
17 EE1E B0      DEFB      B0H      'N'
18 EE1F 97      DEFB      97H      'G'
19 EE20 20      DEFB      20H      'E'
20 EE21 92      DEFB      92H      'R'
21 EE22 9D      DEFB      9DH      'R'
22 EE23 90      DEFB      90H      'O'
23 EE24 B7      DEFB      B7H      'R'
24 EE25 9D      DEFB      9DH      'O'
25 EE26 0D      DEFB      0DH      'R'
26 EE27         DEFM      'GD:File mode error'
27 EE27 51443A46 ERRM4:      DEFM      'GD:File mode error'
28 EE2B A6      DEFB      A6H      'I'
29 EE2C B8      DEFB      B8H      'L'
30 EE2D 92      DEFB      92H      'E'
31 EE2E 20      DEFB      20H      'M'
32 EE2F B3      DEFB      B3H      'O'
33 EE30 B7      DEFB      B7H      'D'
34 EE31 9C      DEFB      9CH      'E'
35 EE32 92      DEFB      92H      'E'
36 EE33 20      DEFB      20H      'R'
37 EE34 92      DEFB      92H      'R'
38 EE35 9D      DEFB      9DH      'O'
39 EE36 9D      DEFB      9DH      'O'
40 EE37 B7      DEFB      B7H      'R'
41 EE38 9D      DEFB      9DH      'O'
42 EE39 0D      DEFB      0DH      'R'
43 EE3A         DEFM      'SRAM:C'
44 EE3A 5352414D 12ERM3: DEFM      'SRAM:C'
45 EE3E 3A43      DEFB      98H      'H'
46 EE40 98      DEFB      92H      'E'
47 EE41 92      DEFB      9FH      'C'
48 EE42 9F      DEFB      A9H      'K'
49 EE43 A9      DEFB      20H      'S'
50 EE44 20      DEFB      A4H      'U'
51 EE45 A4      DEFB      A5H      'M'
52 EE46 A5      DEFB      B3H      'E'
53 EE47 B3      DEFB      20H      'R'
54 EE48 20      DEFB      92H      'R'
55 EE49 92      DEFB      90H      'R'
56 EE4A 9D      DEFB      9DH      'R'
57 EE4B 9D      DEFB      9DH      'R'
58 EE4C 0D      DEFB      0DH      'R'
59 EE4D         DEFM      'MONMSG: 92-504M **'
60 EE4D 2A2A2020 MONMSG: DEFM      'MONMSG: 92-504M **'

```

```

01 EDCB 20      DEFB      A6H      'FD:L'
02 EDCD A6      DEFB      A4H      'FD:Loading error'
03 EDCD A4      DEFB      20H      'O'
04 EDCD B8      DEFB      B8H      'A'
05 EDCF B8      DEFB      B7H      'D'
06 EDD0 B7      DEFB      B7H      'I'
07 EDD1 B7      DEFB      B7H      'N'
08 EDD2 A9      DEFB      A9H      'G'
09 EDD3 A6      DEFB      A6H      'F'
10 EDD4 B0      DEFB      B0H      'O'
11 EDD5 97      DEFB      97H      'R'
12 EDD6 20      DEFB      20H      'A'
13 EDD7 AA      DEFB      AAH      'D'
14 EDD8 B7      DEFB      B7H      'I'
15 EDD9 9D      DEFB      9DH      'N'
16 EDDA 20      DEFB      20H      'G'
17 EDDB A1      DEFB      A1H      'R'
18 EDDC 20      DEFB      20H      'P'
19 EDDD 9E      DEFB      9EH      'R'
20 EDDF 9D      DEFB      9DH      'O'
21 EDDF B7      DEFB      B7H      'O'
22 EDE0 97      DEFB      97H      'G'
23 EDE1 9D      DEFB      9DH      'R'
24 EDE2 A1      DEFB      A1H      'A'
25 EDE3 B3      DEFB      B3H      'M'
26 EDE4 0D      DEFB      0DH      'M'
27 EDE5         DEFM      'FD:Loading error'
28 EDE5 46443A4C ERRM0: DEFM      'FD:Loading error'
29 EDE9 B7      DEFB      B7H      'O'
30 EDEA A1      DEFB      A1H      'A'
31 EDEB 9C      DEFB      9CH      'D'
32 EDEC A6      DEFB      A6H      'I'
33 EDED B0      DEFB      B0H      'N'
34 EDEE 97      DEFB      97H      'G'
35 EDEF 20      DEFB      20H      'E'
36 EDF0 92      DEFB      92H      'R'
37 EDF1 9D      DEFB      9DH      'R'
38 EDF2 9D      DEFB      9DH      'O'
39 EDF3 B7      DEFB      B7H      'R'
40 EDF4 9D      DEFB      9DH      'O'
41 EDF5 0D      DEFB      0DH      'R'
42 EDF6         DEFM      'FD:Not master'
43 EDF6         DEFM      'FD:N'
44 EDF6 46443A4E ERRM1: DEFM      'FD:Not master'
45 EDF6 B7      DEFB      B7H      'O'
46 EDF6 B7      DEFB      96H      'T'
47 EDF6 96      DEFB      20H      'M'
48 EDF6 B3      DEFB      B3H      'A'
49 EDF6 B3      DEFB      A1H      'S'
50 EDF6 A4      DEFB      A4H      'T'
51 EDF6 96      DEFB      96H      'E'
52 EDF6 92      DEFB      92H      'R'
53 EDF6 9D      DEFB      9DH      'R'
54 EDF6 0D      DEFB      0DH      'R'
55 EDF6         DEFM      'CMT:L'
56 EDF6 43D0543A ERRM2: DEFM      'CMT:Loading error'
57 EDF6 4C      DEFB      B7H      'O'
58 EDF6 B7      DEFB      A1H      'A'
59 EDF6 B7      DEFB      9CH      'D'
60 EDF6 9C      DEFB      9CH      'D'

```



```

01 EEA7 ;
02 EEA7 ;
03 EEA7 ; Quick disk control command
04 EEA7 ; QL=Quick load
05 EEA7 ; QS=Quick save
06 EEA7 ; QD=Quick directory
07 EEA7 ; QF=Quick formatting
08 EEA7 ; QX=Quick xfer (CMTDisk)
09 EEA7 ; QC=Quick copy (DiskDisk)
10 EEA7 ;
11 EEA7 ;
12 EEA7 ; BUFF: EQU 11A3H ; Kevin buffer
13 11A3 P
14 EEA7 ;
15 EEA7 ; =====
16 EEA7 ; QUICK DISK LOAD COMMAND
17 EEA7 ;
18 EEA7 ; =====
19 EEA7 ;
20 EEA7 ;
21 EEA7 ; ENT
22 EEA7 ; CALL QDRCK
23 EEA7 ; JR C,GER00
24 EEA7 ; CALL FNINP
25 EEA7 ; CALL HOPCL
26 EEB2 ;
27 EEB2 ; Disp 'Loading...'
28 EEB2 ;
29 EEB2 ; LD DE,MSGLO
30 EEB5 ; RST 3
31 EEB6 ;
32 EEB6 ; File search
33 EEB6 ;
34 EEB6 ; FILSCH: CALL FILSCH
35 EEB9 ; JR C,GER00
36 EEB8 ;
37 EEB8 ; Attribute check
38 EEB8 ;
39 EEB8 ; QDATRC: LD A,(ATRB)
40 EEBE ; CP OBJCO
41 EEC0 ; JR NZ,FILESH
42 EEC2 ;
43 EEC2 ; Disp 'Loading ...'
44 EEC2 ;
45 EEC2 ; DSFLNA: ENT
46 EEC2 ; LD DE,NAME
47 EEC5 ; RST 3
48 EEC6 ;
49 EEC6 ; Iocs parameter set
50 EEC6 ;
51 EEC6 ; LD HL,1200H
52 EEC9 ; JR LPARA1
53 EECB ; LD HL,(QOTADR)
54 EECB ; LD (GDPC),HL
55 EECB ; LD HL,(QSIZE)
56 EECB ; LD HL,(GDPE),HL
57 EECB ; LD HL,0103H
58 EECB ; LD (GDPA),HL
59 EECB ; Read data block
60 EED0 ;

```

```

01 EED0 ;
02 EED0 ; CALL QDIOS
03 EEE0 ; JR C,GER01
04 EEE2 ;
05 EEE2 ; Exec load file
06 EEE2 ;
07 EEE2 ; LD BC,0300H
08 EEE5 ; EXX
09 EEE6 ; LD HL,OSIZE
10 EEE9 ; JP EXF
11 EEE9 ;
12 EEE9 ; Iocs flag reset
13 EEE9 ;
14 EEE9 ; IOFRS: ENT
15 EEE9 ; XOR A
16 EEE9 ; LD (MTE),A
17 EEE9 ; LD (FNUPS),A
18 EEE9 ; LD (FNUPE),A
19 EEE9 ; RET
20 EEE9 ;
21 EEE9 ; File search sub.
22 EEE9 ;
23 EEE9 ;
24 EEE9 ; FILSCH: ENT
25 EEE9 ;
26 EEE9 ; Iocs parameter set
27 EEE9 ;
28 EEE9 ; LD HL,0003H
29 EEE9 ; LD (GDPA),HL
30 EEE9 ; LD HL,10E0H
31 EEE9 ; LD (GDPC),HL
32 EEE9 ; LD HL,0040H
33 EEE9 ; LD (GDPE),HL
34 EEE9 ; LD (GDPE),HL
35 EEE9 ;
36 EEE9 ; Read information block
37 EEE9 ;
38 EEE9 ; QLIN: CALL QDIOS
39 EEE9 ; RET C
40 EEE9 ;
41 EEE9 ; File name check
42 EEE9 ;
43 EEE9 ; LD A,(BUFF)
44 EEE9 ; CP 00H
45 EEE9 ; RET Z
46 EEE9 ; LD HL,BUFF
47 EEE9 ; LD DE,NAME
48 EEE9 ; LD B,NAMSIZ
49 EEE9 ; LD A,(DE)
50 EEE9 ; CP (HL)
51 EEE9 ; JR NZ,QLINF
52 EEE9 ; CP 00H
53 EEE9 ; RET Z
54 EEE9 ; INC DE
55 EEE9 ; INC HL
56 EEE9 ; DJNZ LDFNCK
57 EEE9 ; RET
58 EEE9 ;
59 EEE9 ; Quick disk ready check
60 EEE9 ;

```



```

01 EF07 2804      JR      Z,SFNC0
02 EF09 13        INC     DE
03 EF0A 23        INC     HL
04 EF0B 10F4      DJNZ    SFNC
SFNC0: LD      A,ALEXCD      ; Already exist
06 EF0F 37        SCF
07 EFE0 C9        RET
08 EFE1          ;
09 EFE1          ; Ready & write protect check
10 EFE1          ;
QDRWCK: LD      A,FFH
12 EFE3 323111    LD      (QDPB),A
QDRWCK: LD      A,01H
14 EFE8 323011    LD      (QDPA),A
15 EFEB CD10E0    CALL    QDIOS
16 EFEE C9        RET
17 EFEF          ;
18 EFEF          ; =====
19 EFEF          ; Quick disk directory command
20 EFEF          ;
21 EFEF          ; =====
22 EFEF          ;
23 EFEF          ;
24 EFEF          ; QD:
25 EFEF CD27EF    CALL    QDRCK
26 EFF2 3834      JR      C,QER04
27 EFF4 CD5FF2    CALL    HDPLC
28 EFF7 0600      LD      B,00H
29 EFF9          ; Disp 'Directory of QD:'
30 EFF9          ;
31 EFF9          ;
32 EFF9 C00900    CALL    NL
33 EFFC 115CF3    LD      DE,DIRMSG
34 EFFF DF        RST      3
35 F000          ;
36 F000          ; Iocs parameter set
37 F000          ;
38 F000 2190CD    LD      HL,QDIRBF
DIRIOP: LD      (QDPC),HL
39 F003 223211    LD      HL,0003H
40 F006 210300    LD      (QDPA),HL
41 F009 223011    LD      HL,0040H
42 F00C 214000    LD      (QDPE),HL
43 F00F 223411    LD
44 F012          ; Read information block
45 F012          ;
46 F012          ;
47 F012 C5        PUSH    BC
48 F013 CD10E0    CALL    QDIOS
49 F016 C1        POP      BC
50 F017 380A      JR      C,DIREFC
51 F019 04        INC      B
52 F01A          ; Buffer adrs increment
53 F01A          ;
54 F01A 2A3211    LD      HL,(QDPC)
55 F01A 2A3211    LD      DE,0012H
56 F01D 111200    ADD      HL,DE
57 F020 19        JR      DIRIOP
58 F021 18E0      JR
59 F023          ; End file check
60 F023          ;

01 F023          ;
02 F023 FE28      DIFRFC: CP      NTFECO
03 F025 2804      JR      Z,DIRMTF
04 F027 37        QER04: JP      C,QER05
05 F028 DAC9F0    ; Motor off
06 F02B          ;
07 F02B          ;
08 F02B          ;
09 F02B 3E06      DIRMTF: LD      A,06H
10 F02D 323011    LD      (QDPA),A
11 F030 C5        PUSH    BC
12 F031 CD10E0    CALL    QDIOS
13 F034 C1        POP      BC
14 F035          ; No file check
15 F035          ;
16 F035          ;
17 F035 AF        XOR      A
18 F036 B8        CP      B
19 F037 3079      JR      NC,QDOKM0
20 F039          ; Directory disp
21 F039          ;
22 F039          ;
23 F039 C00900    CALL    NL
24 F03C 2190CD    LD      HL,QDIRBF
25 F03F          ; Disp attribute
26 F03F          ;
27 F03F          ;
28 F03F 7E        DSPATR: LD      A,(HL)
29 F040 116CF2    LD      DE,MSG001
30 F043 3D        DEC      A
31 F044 2836      JR      Z,DRDIS0
32 F046 1170F2    LD      DE,MSG002
33 F049 3D        DEC      A
34 F04A 2830      JR      Z,DRDIS0
35 F04C 1174F2    LD      DE,MSG003
36 F04F 3D        DEC      A
37 F050 282A      JR      Z,DRDIS0
38 F052 1178F2    LD      DE,MSG004
39 F055 3D        DEC      A
40 F056 2824      JR      Z,DRDIS0
41 F058 117CF2    LD      DE,MSG005
42 F05B 3D        DEC      A
43 F05C 281E      JR      Z,DRDIS0
44 F05E 3D        DEC      A
45 F05F 2818      JR      Z,DRDIS1
46 F061 1180F2    LD      DE,MSG007
47 F064 3D        DEC      A
48 F065 2815      JR      Z,DRDIS0
49 F067 3D        DEC      A
50 F068 280F      JR      Z,DRDIS1
51 F06A 3D        DEC      A
52 F06B 280C      JR      Z,DRDIS1
53 F06D 1184F2    LD      DE,MSG010
54 F070 3D        DEC      A
55 F071 2809      JR      Z,DRDIS0
56 F073 1188F2    LD      DE,MSG011
57 F076 3D        DEC      A
58 F077 2803      JR      Z,DRDIS0
59 F079 118CF2    LD      DE,MSG0??
60 F07C C5        PUSH    BC

```

```

01 F07D 0604      LD B,04H
02 F07E C00C00    DROIS2: CALL PRINTS
03 F082 10FB      DJNZ DROIS2
04 F084 C1        POP BC
05 F085 DF        RST 3
06 F086 C00C00    CALL PRINTS
07 F089 C00C00    CALL PRINTS
08 F08C C00C00    CALL PRINTS
09 F08F           ;
10 F08F           ; Disp file name
11 F08F           ;
12 F08F 3E22      LD A,""
13 F091 C01200    CALL PRNT
14 F094 23        INC HL
15 F095 E5        PUSH HL
16 F096 D1        POP DE
17 F097 DF        RST 3
18 F098 3E22      LD A,""
19 F09A C01200    CALL PRNT
20 F09D C00900    CALL NL
21 F0A0           ;
22 F0A0           ; Counter decrement
23 F0A0           ;
24 F0A0 111100    LD DE,011H
25 F0A3 19        ADD HL,DE
26 F0A4 C0CA08    CALL ?KEY
27 F0A7 B7        OR A
28 F0A8 28FA      JR Z,-4
29 F0AA C0320A     CALL ?BRK
30 F0AD C05EEA     JP Z,STIX
31 F0B0 108D      DJNZ DSPATR
32 F0B2           ;
33 F0B2           ; Directory end
34 F0B2           ;
35 F0B2 C395EF    QDOKM0: JP QDOKM
36 F0B5           ;
37 F0B5           ; =====
38 F0B5           ; Quick disk format command
39 F0B5           ;
40 F0B5           ; =====
41 F0B5           ;
42 F0B5           ; QF: ENT
43 F0B5           ;
44 F0B5           ; Disp formatting
45 F0B5           ;
46 F0B5 1130F3    LD DE,QDFMG
47 F0B5           RST 3
48 F0B8 DF        CALL WAITY
49 F0B9 C0CEFO     ; 'QD:FORMATTING'
50 F0BC           ; 'OK?(Y/N)'
51 F0BC           ;
52 F0BC           ; Ready & wrpt check
53 F0BC           ;
54 F0BC C0E1EF     CALL QDRWCK
55 F0BF 3808      JR C,0ER05
56 F0C1           ;
57 F0C1           ; Format
58 F0C1           ;
59 F0C1 3E02      LD A,02H
60 F0C3 323011    LO (QDPA),A

```

```

01 F0C6 C010E0    LD B,06H
02 F0C9 DA60F1    QLEK05: JP C,OLK06
03 F0CC           ;
04 F0CC           ; Format end
05 F0CC           ;
06 F0CC 18E4      JR QDOKM0
07 F0CE           ;
08 F0CE           ; Wait key in 'Y'
09 F0CE           ;
10 F0CE C00900    WAITY: CALL NL
11 F0D1 1153F3    LD DE,0DQCM0
12 F0D4 DF        RST 3
13 F0D5 C091F1    CALL NKYUT
14 F0D8 C0B10F    WAITKY: CALL 0FBIH
15 F0E0 3EEF      LD A,E0H
16 F0E0 77        LD (HL),A
17 F0E0 C0EBF0    WAITK0: LD (HL),A
18 F0E1 3805      CALL WAITM
19 F0E3 7E        JR C,WATK1
20 F0E4 EEEF      LD A,(HL)
21 F0E6 18F5      XOR EFH
22 F0E8 AF        JR WATK0
23 F0E9 77        LD A
24 F0EA C9        WATK1: XOR A
25 F0EB           LD (HL),A
26 F0EB           RET
27 F0EB           ; Wait timer
28 F0EB 010600    WAITM: LD BC,0006H
29 F0EE 08        WATIO: DEC BC
30 F0EF 78        LD A,B
31 F0F0 B1        OR C
32 F0F1 C8        RET Z
33 F0F2           ;
34 F0F2           ; Key search
35 F0F2           ;
36 F0F2 3E01      LD A,01H
37 F0F4 C015F1    CALL SRCK0
38 F0F7 FE7F      CP 7FH
39 F0F9 37        SCF
40 F0FA C8        RET Z
41 F0FB 3E03      LD A,03H
42 F0FD C015F1    CALL SRCK0
43 F100 FEFB      CP FBH
44 F102 2809      JR Z,WAT11
45 F104 3E08      LD A,08H
46 F106 C015F1    CALL SRCK0
47 F109 FE7E      CP 7EH
48 F10B 20E1      JR NZ,WATIO
49 F10D           ;
50 F10D           ; Jump STIX
51 F10D           ;
52 F10D AF        WAT11: XOR A
53 F10E 77        LD (HL),A
54 F10F 31F010    LD SP,10F0H
55 F112 C35EEA    JP STIX
56 F115           ;
57 F115           ; Key scan
58 F115           ;
59 F115 58F115    SRCK0: ENT
60 F115 E5        PUSH HL

```

```

; 'OK?(Y/N)'
; Wait non key
; Call ?PONT
;
; If 'Y' then CY=1
;
; 'Y' ?
;
; 'N' ?
;
; 'Break+Shift' ?

```

```

** Z80 ASSEMBLER 27-004C <97-504M-V1.0C> PAGE 65          03.06.85
01 F116 2100E0      LD HL,E000H
02 F119 77          LD (HL),A
03 F11A 23          INC HL
04 F11B 7E          SRCK2: LD A,(HL)
05 F11C F5          PUSH AF
06 F11D C5          BC
07 F11E 0614        LD B,14H
08 F120 C0B9E6      SRCK1: CALL DLY1M
09 F123 10FB        DJNZ SRCK1
10 F125 C1          POP BC
11 F126 F1          POP AF
12 F127 BE          CP (HL)
13 F128 20F1        JR NZ,SRCK2
14 F12A E1          POP HL
15 F12B C9          RET
16 F12C
17 F12C
18 F12C
19 F12C
20 F12C
21 F12C
22 F12C
23 F12C
24 F12C C027EF      ENT QCRCK
25 F12F 382F        JR C,QER06
26 F131 C08DEC      CALL FNINP
27 F134 3AA311      LD A,(BUFF)
28 F137 FE00        CP 00H
29 F139 28F1        JR Z,QC
30 F13B C05FF2      CALL HOPCL
31 F13E 1191EE      LD DE,MSGLD
32 F141 DF          RST 3
33 F142
34 F142
35 F142
36 F142 C0F7EE      CALL FILSCH
37 F145 3819        JR C,QER06
38 F147
39 F147
40 F147
41 F147 11F110      Disp 'Loading ...'
42 F14A DF          LD DE,NAME
43 F14B
44 F14B
45 F14B
46 F14B 210012      Iocs parameter set
47 F14E 223211      LD HL,1200H
48 F151 2A0411      LD (QDPC),HL
49 F154 223411      LD HL,(QSIZE)
50 F157 210301      LD (QDPE),HL
51 F15A 223011      LD HL,0103H
52 F150
53 F150
54 F150
55 F150 C010E0      Read data block
56 F160 3821        CALL QDIOS
57 F162
58 F162
59 F162
60 F162 C03EE0      QER06: JR C,QER07
                        Bell
                        CALL BELL

** Z80 ASSEMBLER 27-004C <97-504M-V1.0C> PAGE 66          03.06.85
01 F165
02 F165
03 F165
04 F165 C00900      Disp 'Set Disk ...'
05 F168 113EF3      CALL NL
06 F16B DF          LD DE,QDQCM
07 F16C
08 F16C
09 F16C
10 F16C 3E02        Error return set
11 F16E 323A11      QCSPT: LD A,02H
12 F171 216CF1      LD (QDCPA),A
13 F174 31EE10      LD HL,QCSPT
14 F177 E3          LD SP,10EEH
15 F178
16 F178
17 F178
18 F178 C0CEFF0    EX (SP),HL
19 F17B
20 F17B
21 F17B
22 F17B CDE1EF      Wait 'Y'
23 F17E 3803        CALL WAITY
24 F180
25 F180
26 F180
27 F180 C09FEF      Ready & wrpt check
28 F183 DA02F2      QCRWCK: CALL QDRWCK
29 F186 FE28        JR C,QER07
30 F188 C202F2
31 F18B
32 F18B
33 F18B
34 F18B 210012      File end search
35 F18E C374EF      LD HL,1200H
36 F191
37 F191
38 F191
39 F191
40 F191 060A        Save file start
41 F193 2100E0      LD HL,1200H
42 F196 05          JP QSVIOS
43 F197 70          Non key check
44 F198 04          NKYUT: ENT B,0AH
45 F199 23          LD B
46 F19A 7E          NKYUT0: LD HL,E000H
47 F19B FEFF        DEC B
48 F19D 20F2        LD (HL),B
49 F19F 10F2        LD B
50 F1A1 C9          INC B
51 F1A2
52 F1A2
53 F1A2
54 F1A2
55 F1A2
56 F1A2
57 F1A2
58 F1A2
59 F1A2
60 F1A2

; Read data block cmd.
; Read data block
; Quick disk xfer command
; Loading from cmd.

```



```

01 F25C C35EEA      JP      ST1X
02 F25F             ; Header point clear
03 F25F             ;
04 F25F             ;
05 F25F             ;
06 F25F 3E05        HDPCL: ENT      A,05H
07 F261 323011      LD      (Q0PA),A
08 F264 CD10E0      CALL   QDIOS
09 F267 C9          RET
10 F268             ;
11 F268             ;
12 F268             ;
13 F268             ;
14 F268             ;
15 F268             ;
16 F268             ;
17 F268 4F4B21      MSGQ0K: DEFM   'OK!'
18 F26B 00          DEF8   00H
19 F26C 4F424A      MSGQ01: DEFM   'OBJ'
20 F26F 00          DEF8   00H
21 F270 425458      MSGQ02: DEFM   'BTX'
22 F273 00          DEF8   00H
23 F274 425344      MSGQ03: DEFM   'BSD'
24 F277 00          DEF8   00H
25 F278 425244      MSGQ04: DEFM   'BRD'
26 F27B 00          DEF8   00H
27 F27C 524220      MSGQ05: DEFM   'RB '
28 F27F 00          DEF8   00H
29 F280 4C4942      MSGQ07: DEFM   'LIB'
30 F283 00          DEF8   00H
31 F284 535953      MSGQ10: DEFM   'SYS'
32 F287 00          DEF8   00H
33 F288 475220      MSGQ11: DEFM   'GR '
34 F28B 00          DEF8   00H
35 F28C 3F3F3F      MSGQ???: DEFM   '???'
36 F28F 00          DEF8   00H
37 F290 5143A46     MGNFE:  DEFM   'QD:F'
38 F294 A6          DEF8   A6H
39 F295 B8          DEF8   B8H
40 F296 92          DEF8   92H
41 F297 20          DEF8   20H
42 F298 B0          DEF8   B0H
43 F299 B7          DEF8   B7H
44 F29A 96          DEF8   96H
45 F29B 20          DEF8   20H
46 F29C AA          DEF8   AAH
47 F29D B7          DEF8   B7H
48 F29E A5          DEF8   A5H
49 F29F B0          DEF8   B0H
50 F2A0 9C          DEF8   9CH
51 F2A1 00          DEF8   00H
52 F2A2             ;
53 F2A2 5143A54     MGTME:  DEFM   'QD:T'
54 F2A6 B7          DEF8   B7H
55 F2A7 B7          DEF8   B7H
56 F2A8 20          DEF8   20H
57 F2A9 B3          DEF8   B3H
58 F2AA A1          DEF8   A1H
59 F2AB B0          DEF8   B0H
60 F2AC B0          DEF8   B0H

01 F2AD 20          DEF8   20H
02 F2AE AA          DEF8   AAH
03 F2AF A6          DEF8   A6H
04 F2B0 B8          DEF8   B8H
05 F2B1 92          DEF8   92H
06 F2B2 A4          DEF8   A4H
07 F2B3 20          DEF8   20H
08 F2B4 92          DEF8   92H
09 F2B5 90          DEF8   90H
10 F2B6 9D          DEF8   9DH
11 F2B7 00          DEF8   00H
12 F2B8             ;
13 F2B8 51443A48     MGH0E:  DEFM   'QD:H'
14 F2BC A1          DEF8   A1H
15 F2BD 9D          DEF8   9DH
16 F2BE 9C          DEF8   9CH
17 F2BF 20          DEF8   20H
18 F2C0 92          DEF8   92H
19 F2C1 9D          DEF8   9DH
20 F2C2 90          DEF8   90H
21 F2C3 00          DEF8   00H
22 F2C4 41          MGALE:  DEFM   'A'
23 F2C5 B8          DEF8   B8H
24 F2C6 9D          DEF8   9DH
25 F2C7 92          DEF8   92H
26 F2C8 A1          DEF8   A1H
27 F2C9 9C          DEF8   9CH
28 F2CA B0          DEF8   BDH
29 F2CB 20          DEF8   20H
30 F2CC 92          DEF8   92H
31 F2CD 9B          DEF8   9BH
32 F2CE A6          DEF8   A6H
33 F2CF A4          DEF8   A4H
34 F2D0 96          DEF8   96H
35 F2D1 20          DEF8   20H
36 F2D2 92          DEF8   92H
37 F2D3 9D          DEF8   9DH
38 F2D4 9D          DEF8   9DH
39 F2D5 00          DEF8   00H
40 F2D6             ;
41 F2D6 51443A57     MGWPT:  DEFM   'QD:W'
42 F2DA 9D          DEF8   9DH
43 F2DB A6          DEF8   A6H
44 F2DC 96          DEF8   96H
45 F2DD 92          DEF8   92H
46 F2DE 20          DEF8   20H
47 F2DF 9E          DEF8   9EH
48 F2E0 9D          DEF8   9DH
49 F2E1 B7          DEF8   B7H
50 F2E2 96          DEF8   96H
51 F2E3 92          DEF8   92H
52 F2E4 9F          DEF8   9FH
53 F2E5 96          DEF8   96H
54 F2E6 00          DEF8   00H
55 F2E7             ;
56 F2E7 51443A4E     MGNRE:  DEFM   'QD:N'
57 F2EB B7          DEF8   B7H
58 F2EC 96          DEF8   96H
59 F2ED 20          DEF8   20H
60 F2EE 9D          DEF8   9DH

; 'QD:Hard err'
; A
; R
; D
; E
; R
; R
; 'Already exist err'
; L
; R
; E
; A
; D
; Y
; E
; X
; I
; S
; T
; E
; R
; 'QD:Write protect'
; R
; I
; T
; E
; P
; R
; O
; T
; E
; C
; T
; 'QD:Not ready'
; O
; T
; R

```

```

01 F2EF 92      DEFB 92H      ; E
02 F2F0 A1      DEFB A1H      ; A
03 F2F1 9C      DEFB 9CH      ; D
04 F2F2 B0      DEFB B0H      ; Y
05 F2F3 0D      DEFB 0DH
06 F2F4         ; MGNSE:      *QD:N'
07 F2F5 5143A4E DEFB B7H      DEFB *QD:No file space err'
08 F2F8 B7      DEFB B7H      ; Q
09 F2F9 20      DEFB 20H      ; F
10 F2FA AA      DEFB AAH      ; I
11 F2FB A6      DEFB A6H      ; L
12 F2FC B8      DEFB B8H      ; E
13 F2FD 92      DEFB 92H      ; S
14 F2FE 20      DEFB 20H      ; P
15 F2FF A4      DEFB A4H      ; A
16 F300 9E      DEFB 9EH      ; C
17 F301 A1      DEFB A1H      ; E
18 F302 9F      DEFB 9FH      ; E
19 F303 92      DEFB 92H      ; R
20 F304 20      DEFB 20H      ; QD:U'
21 F305 92      DEFB 92H      ; B
22 F306 9D      DEFB 9DH      ; A
23 F307 9D      DEFB 9DH      ; T
24 F308 0D      DEFB 0DH      ; R
25 F309         ; MGNSE:      *QD:U'
26 F309 5143A55 DEFB B0H      DEFB *QD:Bad disk err'
27 F30B B0      DEFB B0H      ; A
28 F30E AA      DEFB AAH      ; D
29 F30F B7      DEFB B7H      ; D
30 F310 9D      DEFB 9DH      ; I
31 F311 B3      DEFB B3H      ; S
32 F312 A1      DEFB A1H      ; K
33 F313 96      DEFB 96H      ; E
34 F314 20      DEFB 20H      ; R
35 F315 92      DEFB 92H      ; B
36 F316 9D      DEFB 9DH      ; R
37 F317 9D      DEFB 9DH      ; QD:B'
38 F318 0D      DEFB 0DH      ; A
39 F319         ; MGBDE:      *QD:B'
40 F319 5143A42 DEFB A1H      DEFB 9CH
41 F31D A1      DEFB A1H      DEFB 20H
42 F31E 9C      DEFB 9CH      DEFB 9CH
43 F31F 20      DEFB 20H      DEFB 9CH
44 F320 9C      DEFB 9CH      DEFB A6H
45 F321 A6      DEFB A6H      DEFB A4H
46 F322 A4      DEFB A4H      DEFB A9H
47 F323 A9      DEFB A9H      DEFB 20H
48 F324 20      DEFB 20H      DEFB 92H
49 F325 92      DEFB 92H      DEFB 9DH
50 F326 9D      DEFB 9DH      DEFB 9DH
51 F327 9D      DEFB 9DH      DEFB 0DH
52 F328 0D      DEFB 0DH      DEFB 0DH
53 F329         ; MGBRK:      *B'
54 F329 ENT      DEFB ENT      DEFB *B'
55 F329 42      DEFB 42H      DEFB 9DH
56 F32A 9D      DEFB 9DH      DEFB 92H
57 F32B 92      DEFB 92H      DEFB A1H
58 F32C A1      DEFB A1H      DEFB A9H
59 F32D A9      DEFB A9H      DEFB 21H
60 F32E 21      DEFB 21H      DEFB 21H

```


137

4 Z-80 Programming reference

4-1 Z-80 Status indicators (Flags)

The flag register (F and F') supplies information to the user regarding the status of the Z80 at any given time. The bit positions for each flag is shown below:

7	6	5	4	3	2	1	0
S	Z	X	H	X	P/V	N	C

where:

- C = Carry Flag
- N = Add/Subtract Flag
- P/V = Parity/Overflow Flag
- H = Half-Carry Flag
- Z = Zero Flag
- S = Sign Flag
- X = Not Used

Each of the two Z-80 Flag Registers contains 6 bits of status information which are set or reset by CPU operations. (Bits 3 and 5 are not used.) Four of these bits are testable (C, P/V, Z and S) for use with conditional jump, call or return instructions. Two flags are not testable (H, N) and are used for BCD arithmetic.

Carry Flag (C)

The carry bit is set or reset depending on the operation being performed. For 'ADD' instructions that generate a carry and 'SUBTRACT' instructions that generate no borrow, the Carry Flag will be set. The Carry Flag is reset by an ADD that does not generate a carry and a 'SUBTRACT' that generates a borrow. This saved carry facilitates software routines for extended precision arithmetic. Also, the 'DAA' instruction will set the Carry Flag if the conditions for making the decimal adjustment are met.

For instructions RLA, RRA, RLS and RRS, the carry bit is used as a link between the LSB and MSB for any register or memory location. During instructions RLCA, RLC s and SLA s, the carry contains the last value shifted out of bit 7 of any register or memory location. During instructions RRCA, RRC s, SRA s and SRL s, the carry contains the last value shifted out of bit 0 of any register or memory location.

For the logical instructions AND s, OR s and XOR s, the carry will be reset.

The Carry Flag can also be set (SCF) and complemented (CCF).

Add/Subtract Flag (N)

This flag is used by the decimal adjust accumulator instruction (DAA) to distinguish between 'ADD' and 'SUBTRACT' instructions. For all 'ADD' instructions, N will be set to a '0'. For all 'SUBTRACT' instructions, N will be set to a '1'.

The Zero Flag (Z)

The Zero Flag (Z) is set or reset if the result generated by the execution of certain instructions is a zero.

For 8-bit arithmetic and logical operations, the Z flag will be set to a '1' if the resulting byte in the Accumulator is zero. If the byte is not zero, the Z flag is reset to '0'.

For compare (search) instructions, the Z flag will be set to a '1' if a comparison is found between the value in the Accumulator and the memory location pointed to by the contents of the register pair HL.

When testing a bit in a register or memory location, the Z flag will contain the complemented state of the indicated bit (see Bit b,s).

When inputting or outputting a byte between a memory location and an I/O device (INI, IND, OUTI and OUTD) if the result of B-1 is zero, the Z flag is set, otherwise it is reset. Also for byte inputs from I/O devices using IN r, (C), the Z flag is set to indicate a zero byte input.

Parity/Overflow Flag

This flag is set to a particular state depending on the operation being performed.

For arithmetic operations, this flag indicates an overflow condition when the result in the Accumulator is greater than the maximum possible number (+127) or is less than the minimum possible number (–128). This overflow condition can be determined by examining the sign bits of the operands.

For addition, operands with different signs will never cause overflow. When adding operands with like signs and the result has a different sign, the overflow flag is set. For example:

+120 = 0111 1000	ADDEND
+105 = 0110 1001	AUGEND
+225 1110 0001	(–95) SUM

The two numbers added together has resulted in a number that exceeds +127 and the two positive operands has resulted in a negative number (–95) which is incorrect. The overflow flag is therefore set.

For subtraction, overflow can occur for operands of unlike signs. Operands of like sign will never cause overflow. For example:

+127 0111 1111	MINUEND
(–) –64 1100 0000	SUBTRAHEND
+191 1011 1111	DIFFERENCE

The minuend sign has changed from a positive to a negative, giving an incorrect difference. Overflow is therefore set.

Another method for predicting an overflow is to observe the carry into and out of the sign bit. If there is a carry in and no carry out, or if there is no carry in and a carry out, then overflow has occurred.

This flag is also used with logical operations and rotate instructions to indicate the parity of the result. The number of '1' bits in a byte are counted. If the total is odd, 'ODD' parity (P=0) is flagged. If the total is even, 'EVEN' parity is flagged (P=1).

During search instructions (CPI, CPIR, CPD, CPDR) and block transfer instructions (LDI, LDIR, LDD, LDDR) the P/V flag monitors the state of the byte count register (BC). When decrementing, the byte counter results in a zero value, the flag is reset to 0, otherwise the flag is a Logic 1.

During LD A, I and LD A, R instructions, the P/V flag will be set with the contents of the interrupt enable flip-flop (1FF2) for storage or testing.

When inputting a byte from an I/O device, IN r, (C), the flag will be adjusted to indicate the parity of the data.

The Half Carry Flag (H)

The Half Carry Flag (H) will be set or reset depending on the carry and borrow status between bits 3 and 4 of an 8-bit arithmetic operation. This flag is used by the decimal adjust accumulator instruction (DAA) to correct the result of a packed BCD add or subtract operation. The H flag will be set (1) or reset (0) according to the following table:

H	ADD	SUBTRACT
1	There is a carry from Bit 3 to Bit 4.	There is no borrow from bit 4.
0	There is no carry from Bit 3 to Bit 4.	There is a borrow from Bit 4.

The Sign Flag (S)

The Sign Flag (S) stores the state of the most significant bit of the Accumulator (Bit 7). When the Z80 performs arithmetic operations on signed numbers, binary two's complement notation is used to represent and process numeric information. A positive number is identified by a '0' in bit 7. A negative number is identified by a '1'. The binary equivalent of the magnitude of a positive number is stored in bits 0 to 6 for a total range of from 0 to 127. A negative number is represented by the two's complement of the equivalent positive number. The total range for negative numbers is from –1 to –128.

When inputting a byte from an I/O device to a register, IN r, (C), the S flag will indicate either positive (S=0) or negative (S=1) data.

4-2 Notation

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

The following notation is used in this table:

SYMBOL

OPERATION

- C Carry/link flag. C=1 if the operation produced a carry from the MSB of the operand or result.
- Z Zero flag. Z=1 if the result of the operation is zero.
- S Sign flag. S=1 if the MSB of the result is one.
- P/V Parity or overflow flag. Parity (P) and overflow (V) share the same flag. Logical operations affect this flag with the parity of the result while arithmetic operations affect this flag with the overflow of the result. If P/V holds parity, P/V=1 if the result of the operation is even, P/V=0 if result is odd. If P/V holds overflow, P/V=1 if the result of the operation produced an overflow.
- H Half-carry flag. H=1 if the add or subtract operation produced a carry into or borrow from bit 4 of the accumulator.
- N Add/Subtract flag. N=1 if the previous operation was a subtract.
- H and N flags are used in conjunction with the decimal adjust instruction (DAA) to properly correct the result into packed BCD format following addition or subtraction using operands with packed BCD format.
- :
- The flag is affected according to the result of the operation.
- The flag is unchanged by the operation.
- 0 The flag is reset by the operation.
- 1 The flag is set by the operation.
- X The flag is a "don't care."
- V P/V flag affected according to the overflow result of the operation.
- P P/V flag affected according to the parity result of the operation.
- r Any one of the CPU registers A, B, C, D, E, H, L
- s Any 8-bit location for all the addressing modes allowed for the particular instruction.
- ss Any 16-bit location for all the addressing modes allowed for that instruction.
- ii Any one of the two index registers IX or IY.
- R Refresh counter.
- n 8-bit value in range <0,255>.
- nn 16-bit value in range <0,65535>.
- m Any 8-bit location for all the addressing modes allowed for the particular instruction.

8-Bit Load Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments	
		C	Z	P	V	S	N	H	76	543	210				
LD r,r'	$r \leftarrow r'$	●	●	●	●	●	●	●	01	r	r'	1	1	4	r,r
LD r,n	$r \leftarrow n$	●	●	●	●	●	●	●	00	r	110	2	2	7	000 B
										$\leftarrow n \rightarrow$					001 C
LD r,(HL)	$r \leftarrow (HL)$	●	●	●	●	●	●	●	01	r	110	1	2	7	010 D
LD r,(IX+d)	$r \leftarrow (IX+d)$	●	●	●	●	●	●	●	11	011	101	3	5	19	011 E
									01	r	110				100 H
										$\leftarrow d \rightarrow$					101 L
LD r,(IY+d)	$r \leftarrow (IY+d)$	●	●	●	●	●	●	●	11	111	101	3	5	19	111 A
									01	r	110				
										$\leftarrow d \rightarrow$					
LD (HL),r	$(HL) \leftarrow r$	●	●	●	●	●	●	●	01	110	r	1	2	7	
LD (IX+d),r	$(IX+d) \leftarrow r$	●	●	●	●	●	●	●	11	011	101	3	5	19	
									01	110	r				
										$\leftarrow d \rightarrow$					
LD (IY+d),r	$(IY+d) \leftarrow r$	●	●	●	●	●	●	●	11	111	101	3	5	19	
									01	110	r				
										$\leftarrow d \rightarrow$					
LD (HL),n	$(HL) \leftarrow n$	●	●	●	●	●	●	●	00	110	110	2	3	10	
										$\leftarrow n \rightarrow$					
LD (IX+d),n	$(IX+d) \leftarrow n$	●	●	●	●	●	●	●	11	011	101	4	5	19	
									00	110	110				
										$\leftarrow d \rightarrow$					
										$\leftarrow n \rightarrow$					
LD (IY+d),n	$(IY+d) \leftarrow n$	●	●	●	●	●	●	●	11	111	101	4	5	19	
									00	110	110				
										$\leftarrow d \rightarrow$					
										$\leftarrow n \rightarrow$					
LD A,(BC)	$A \leftarrow (BC)$	●	●	●	●	●	●	●	00	001	010	1	2	7	
LD A,(DE)	$A \leftarrow (DE)$	●	●	●	●	●	●	●	00	011	010	1	2	7	
LD A,(nn)	$A \leftarrow (nn)$	●	●	●	●	●	●	●	00	111	010	3	4	13	
										$\leftarrow n \rightarrow$					
										$\leftarrow n \rightarrow$					
LD (BC),A	$(BC) \leftarrow A$	●	●	●	●	●	●	●	00	000	010	1	2	7	
LD (DE),A	$(DE) \leftarrow A$	●	●	●	●	●	●	●	00	010	010	1	2	7	
LD (nn),A	$(nn) \leftarrow A$	●	●	●	●	●	●	●	00	110	010	3	4	13	
										$\leftarrow n \rightarrow$					
										$\leftarrow n \rightarrow$					
LD A,I	$A \leftarrow I$	●	↑	IFF2	↑	0	0	0	11	101	111	2	2	9	IFF2: contents of interrupt enable flip-flop 2
									01	010	111				
LD A,R	$A \leftarrow R$	●	↑	IFF2	↑	0	0	0	11	101	101	2	2	9	
									01	011	111				
LD I,A	$I \leftarrow A$	●	●	●	●	●	●	●	11	101	101	2	2	9	
									01	000	111				
LD R,A	$R \leftarrow A$	●	●	●	●	●	●	●	11	101	101	2	2	9	
									01	001	111				

16-Bit Load Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments	
		C	Z	P	V	S	N	H	76	543	210				
LD dd,nn	dd ← nn	•	•	•	•	•	•	•	00	dd0	001	3	3	10	dd register pairs
									←	n	→				00 BC
									←	n	→				01 DE
LD IX,nn	IX ← nn	•	•	•	•	•	•	•	11	011	101	4	4	14	10 HL
									00	100	001				11 SP
									←	n	→				
									←	n	→				
LD IY,nn	IY ← nn	•	•	•	•	•	•	•	11	111	101	4	4	14	
									00	100	001				
									←	n	→				
									←	n	→				
LD HL,(nn)	H ← (nn + 1) L ← (nn)	•	•	•	•	•	•	•	00	101	010	3	5	16	
									←	n	→				
									←	n	→				
LD dd,(nn)	dd _H ← (nn + 1) dd _L ← (nn)	•	•	•	•	•	•	•	11	101	101	4	6	20	
									01	dd1	011				
									←	n	→				
									←	n	→				
LD IX,(nn)	IX _H ← (nn + 1) IX _L ← (nn)	•	•	•	•	•	•	•	11	011	101	4	6	20	
									00	101	010				
									←	n	→				
									←	n	→				
LD IY,(nn)	IY _H ← (nn + 1) IY _L ← (nn)	•	•	•	•	•	•	•	11	111	101	4	6	20	
									00	101	010				
									←	n	→				
									←	n	→				
LD (nn),HL	(nn + 1) ← H (nn) ← L	•	•	•	•	•	•	•	00	100	010	3	5	16	
									←	n	→				
									←	n	→				
LD (nn),dd	(nn + 1) ← dd _H (nn) ← dd _L	•	•	•	•	•	•	•	11	101	101	4	6	20	
									01	dd0	011				
									←	n	→				
									←	n	→				
LD (nn),IX	(nn + 1) ← IX _H (nn) ← IX _L	•	•	•	•	•	•	•	11	011	101	4	6	20	
									00	100	010				
									←	n	→				
									←	n	→				
LD (nn),IY	(nn + 1) ← IY _H (nn) ← IY _L	•	•	•	•	•	•	•	11	111	101	4	6	20	
									00	100	010				
									←	n	→				
									←	n	→				
LD SP,HL	SP ← HL	•	•	•	•	•	•	•	11	111	001	1	1	6	
LD SP,IX	SP ← IX	•	•	•	•	•	•	•	11	011	101	2	2	10	
									11	111	001				
LD SP,IY	SP ← IY	•	•	•	•	•	•	•	11	111	101	2	2	10	
									11	111	001				

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments	
		C	Z	P V	S	N	H	76	543	210					
PUSH qq	$(SP-2) \leftarrow qq_L$	●	●	●	●	●	●	11	qq0	101	1	3	11	qq	register pairs
	$(SP-1) \leftarrow qq_H$													00	BC
PUSH IX	$(SP-2) \leftarrow IX_L$	●	●	●	●	●	●	11	011	101	2	4	15	01	DE
	$(SP-1) \leftarrow IX_H$							11	100	101				10	HL
PUSH IY	$(SP-2) \leftarrow IY_L$	●	●	●	●	●	●	11	111	101	2	4	15	11	AF
	$(SP-1) \leftarrow IY_H$							11	100	101					
POP qq	$qq_H \leftarrow (SP+1)$	●	●	●	●	●	●	11	qq0	001	1	3	10		
	$qq_L \leftarrow (SP)$														
POP IX	$IX_H \leftarrow (SP+1)$	●	●	●	●	●	●	11	011	101	2	4	14		
	$IX_L \leftarrow (SP)$							11	100	001					
POP IY	$IY_H \leftarrow (SP+1)$	●	●	●	●	●	●	11	111	101	2	4	14		
	$IY_L \leftarrow (SP)$							11	100	001					

Exchange, Block Transfer and Search Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments
		C	Z	P/V	S	N	H	76	543	210				
EX DE,HL	DE $\leftrightarrow$ HL	●	●	●	●	●	●	11	101	011	1	1	4	Each value in register pairs is exchanged with the value in aux. register pairs.
EX AF,AF'	AF $\leftrightarrow$ AF'	●	●	●	●	●	●	00	001	000	1	1	4	
EXX	$\begin{pmatrix} BC \\ DE \\ HL \end{pmatrix} \leftrightarrow \begin{pmatrix} BC' \\ DE' \\ HL' \end{pmatrix}$	●	●	●	●	●	●	11	011	001	1	1	4	
EX (SP),HL	H $\leftrightarrow$ (SP+1) L $\leftrightarrow$ (SP)	●	●	●	●	●	●	11	100	011	1	5	19	
EX (SP),IX	IX _H $\leftrightarrow$ (SP+1) IX _L $\leftrightarrow$ (SP)	●	●	●	●	●	●	11	011	101	2	6	23	
EX (SP),IY	IY _H $\leftrightarrow$ (SP+1) IY _L $\leftrightarrow$ (SP)	●	●	●	●	●	●	11	111	101	2	6	23	
LDI	(DE) $\leftarrow$ (HL) DE $\leftarrow$ DE+1 HL $\leftarrow$ HL+1 BC $\leftarrow$ BC-1	●	●	①	●	0	0	11	101	101	2	4	16	for BC $\neq$ 0 for BC=0
LDIR	(DE) $\leftarrow$ (HL) DE $\leftarrow$ DE+1 HL $\leftarrow$ HL+1 BC $\leftarrow$ BC-1 repeats to BC=0	●	●	0	●	0	0	11	101	101	2	5	21	
								10	110	000	2	4	16	
LDD	(DE) $\leftarrow$ (HL) DE $\leftarrow$ DE-1 HL $\leftarrow$ HL-1 BC $\leftarrow$ BC-1	●	●	①	●	0	0	11	101	101	2	4	16	
								10	101	000				
LDDR	(DE) $\leftarrow$ (HL) DE $\leftarrow$ DE-1 HL $\leftarrow$ HL-1 BC $\leftarrow$ BC-1 repeats to BC=0	●	●	0	●	0	0	11	101	101	2	5	21	
								10	111	000	2	4	16	
CPI	A-(HL) HL $\leftarrow$ HL+1 BC $\leftarrow$ BC-1	●	②	①	↑	1	↑	11	101	101	2	4	16	
								10	100	001				

Notes: ① indicates that the P/V flag is set to 0 for BC-1=0 and other cases to 1.

② indicates that the Z flag is set to 1 for A=(HL) and other cases to 0.

Exchange, Block Transfer and Search Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments
		C	Z	P/V	S	N	H	76	543	210				
EX DE,HL	DE $\leftrightarrow$ HL	●	●	●	●	●	●	11	101	011	1	1	4	Each value in register pairs is exchanged with the value in aux. register pairs.
EX AF,AF'	AF $\leftrightarrow$ AF'	●	●	●	●	●	●	00	001	000	1	1	4	
EXX	$\begin{pmatrix} BC \\ DE \\ HL \end{pmatrix} \leftrightarrow \begin{pmatrix} BC' \\ DE' \\ HL' \end{pmatrix}$	●	●	●	●	●	●	11	011	001	1	1	4	
EX (SP),HL	H $\leftrightarrow$ (SP+1) L $\leftrightarrow$ (SP)	●	●	●	●	●	●	11	100	011	1	5	19	
EX (SP),IX	IX _H $\leftrightarrow$ (SP+1) IX _L $\leftrightarrow$ (SP)	●	●	●	●	●	●	11	011	101	2	6	23	
EX (SP),IY	IY _H $\leftrightarrow$ (SP+1) IY _L $\leftrightarrow$ (SP)	●	●	●	●	●	●	11	111	101	2	6	23	
LDI	(DE) $\leftarrow$ (HL) DE $\leftarrow$ DE+1 HL $\leftarrow$ HL+1 BC $\leftarrow$ BC-1	●	●	①	●	0	0	11	101	101	2	4	16	for BC $\neq$ 0 for BC=0
LDIR	(DE) $\leftarrow$ (HL) DE $\leftarrow$ DE+1 HL $\leftarrow$ HL+1 BC $\leftarrow$ BC-1 repeats to BC=0	●	●	0	●	0	0	11	101	101	2	5	21	
								10	110	000	2	4	16	
LDD	(DE) $\leftarrow$ (HL) DE $\leftarrow$ DE-1 HL $\leftarrow$ HL-1 BC $\leftarrow$ BC-1	●	●	①	●	0	0	11	101	101	2	4	16	
								10	101	000				
LDDR	(DE) $\leftarrow$ (HL) DE $\leftarrow$ DE-1 HL $\leftarrow$ HL-1 BC $\leftarrow$ BC-1 repeats to BC=0	●	●	0	●	0	0	11	101	101	2	5	21	
								10	111	000	2	4	16	
CPI	A-(HL) HL $\leftarrow$ HL+1 BC $\leftarrow$ BC-1	●	②	①	↑	1	↑	11	101	101	2	4	16	
								10	100	001				

Notes: ① indicates that the P/V flag is set to 0 for BC-1=0 and other cases to 1.

② indicates that the Z flag is set to 1 for A=(HL) and other cases to 0.

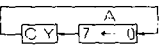
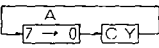
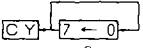
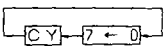
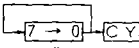
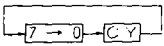
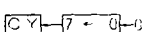
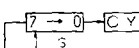
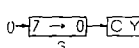
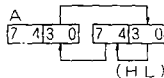
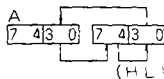
8-Bit Arithmetic and Logical Group

mnemonic	operation	flags							opcode			bytes	No. of M cycles	No. of T cycles	comments	
		C	Z	P	V	S	N	H	76	543	210					
ADD A,r	$A \leftarrow A + r$	↑	↑	V	↑	0	↑	↑	10	000	r	1	1	4	r	register
ADD A,n	$A \leftarrow A + n$	↑	↑	V	↑	0	↑	↑	11	000	110	2	2	7	000	B
										← n →					001	C
ADD A,(HL)	$A \leftarrow A + (HL)$	↑	↑	V	↑	0	↑	↑	10	000	110	1	2	7	010	D
ADD A,(IX+d)	$A \leftarrow A + (IX + d)$	↑	↑	V	↑	0	↑	↑	11	011	101	3	5	19	011	E
									10	000	110				100	H
										← d →					101	L
ADD A,(IY+d)	$A \leftarrow A + (IY + d)$	↑	↑	V	↑	0	↑	↑	11	111	101	3	5	19	111	A
									10	000	110					
										← d →						
ADC A,s	$A \leftarrow A + s + CY$	↑	↑	V	↑	0	↑	↑		001					The s operand is any of r, n, (HL), (IX+d) or (IY+d) as defined for the ADD instruction. Replace the opcode 000 in ADD with the framed bits.	
SUB s	$A \leftarrow A - s$	↑	↑	V	↑	1	↑	↑		010						
SBC A,s	$A \leftarrow A - s - CY$	↑	↑	V	↑	1	↑	↑		011						
AND s	$A \leftarrow A \wedge s$	0	↑	P	↑	0	1	↑		100						
OR s	$A \leftarrow A \vee s$	0	↑	P	↑	0	0	↑		110						
XOR s	$A \leftarrow A \oplus s$	0	↑	P	↑	0	0	↑		101						
CP s	$A - s$	↑	↑	V	↑	1	↑	↑		111						
INC r	$r \leftarrow r + 1$	●	↑	V	↑	0	↑	↑	00	r	100	1	1	4		
INC (HL)	$(HL) \leftarrow (HL) + 1$	●	↑	V	↑	0	↑	↑	00	110	100	1	3	11		
INC (IX+d)	$(IX + d) \leftarrow (IX + d) + 1$	●	↑	V	↑	0	↑	↑	11	011	101	3	6	23		
									00	110	100					
										← d →						
INC (IY+d)	$(IY + d) \leftarrow (IY + d) + 1$	●	↑	V	↑	0	↑	↑	11	111	101	3	6	23		
									00	110	100					
										← d →						
DEC m	$m \leftarrow m - 1$	●	↑	V	↑	1	↑	↑			101				The m operand is any of r, (HL), (IX+d) or (IY+d) as defined for the INC instruction. The opcode is the same as the INC's except 100 changes to 101.	

16-Bit Arithmetic Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments	
		C	Z	P V	S	N	H	76	543	210					
ADD HL,ss	HL←HL+ss	↑	●	●	●	0	X	00	ss1	001	1	3	11	ss	register
ADC HL,ss	HL←HL+ss+CY	↑	↑	V	↑	0	X	11	101	101	2	4	15	00	BC
								01	ss1	010				01	DE
SBC HL,ss	HL←HL-ss-CY	↑	↑	V	↑	1	X	11	101	101	2	4	15	10	HL
								01	ss0	010				11	SP
ADD IX,pp	IX←IX+pp	↑	●	●	●	0	X	11	011	101	2	4	15	pp	register
								00	pp1	001				00	BC
														01	DE
														10	IX
														11	SP
ADD IY,rr	IY←IY+rr	↑	●	●	●	0	X	11	111	101	2	4	15	rr	register
								00	rr1	001				00	BC
														01	DE
														10	IY
														11	SP
INC ss	ss←ss+1	●	●	●	●	●	●	00	ss0	011	1	1	6		
INC IX	IX←IX+1	●	●	●	●	●	●	11	011	101	2	2	10		
								00	100	011					
INC IY	IY←IY+1	●	●	●	●	●	●	11	111	101	2	2	10		
								00	100	011					
DEC ss	ss←ss-1	●	●	●	●	●	●	00	ss1	011	1	1	6		
DEC IX	IX←IX-1	●	●	●	●	●	●	11	011	101	2	2	10		
								00	101	011					
DEC IY	IY←IY-1	●	●	●	●	●	●	11	111	101	2	2	10		
								00	101	011					

Rotate and Shift Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments								
		C	Z	P	V	S	N	H	76	543						210						
RLC A		↑	●	●	●	0	0	00	000	111	1	1	4	The contents of the Accumulator are rotated left.								
RL A		↑	●	●	●	0	0	00	010	111	1	1	4									
RRC A		↑	●	●	●	0	0	00	001	111	1	1	4			The contents of the Accumulator are rotated right.						
RR A		↑	●	●	●	0	0	00	011	111	1	1	4									
RLC r		↑	↑	P	↑	0	0	11	001	011	2	2	8	The contents of register r are rotated left.								
RLC (HL)		↑	↑	P	↑	0	0	11	001	011	2	4	15			r	register					
RLC (IX+d)		↑	↑	P	↑	0	0	00	000	110	4	6	23	000	B							
RLC (IY+d)		↑	↑	P	↑	0	0	11	011	101	4	6	23	001	C							
								11	001	011				010	D							
								← d →						011	E							
								00	000	110				100	H							
								11	111	101				101	L							
RLC (IY+d)								11	001	011	4	6	23	111	A							
								← d →														
								00	000	110												
RL s		↑	↑	P	↑	0	0	010						The s operand is any of r, (HL), (IX+d) or (IY+d)								
RRC s		↑	↑	P	↑	0	0	001														
RR s		↑	↑	P	↑	0	0	011														
SLA s		↑	↑	P	↑	0	0	100														
SRA s		↑	↑	P	↑	0	0	101														
SRL s		↑	↑	P	↑	0	0	111														
RLD		●	↑	P	↑	0	0	11	101	101	2	5	18									
RRD		●	↑	P	↑	0	0	01	101	111		5	18									
								11	101	101												
								01	100	111												

Bit Set, Reset and Test Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments	
		C	Z	P/V	S	N	H	76	543	210					
BIT b,r	$Z \leftarrow \bar{r}_b$	•	↑	X	X	0	1	11	001	011	2	2	8	r	register
BIT b,(HL)	$Z \leftarrow \overline{(HL)}_b$	•	↑	X	X	0	1	01	b	r	2	3	12	000	B
								11	001	011				001	C
								01	b	110				010	D
BIT b,(IX+d)	$Z \leftarrow \overline{(IX+d)}_b$	•	↑	X	X	0	1	11	011	101	4	5	20	011	E
								11	001	011				100	H
								←	d	→				101	L
								01	b	110				111	A
BIT b,(IY+d)	$Z \leftarrow \overline{(IY+d)}_b$	•	↑	X	X	0	1	11	111	101	4	5	20	b	bit tested
								11	001	011				000	0
								←	d	→				001	1
								01	b	110				010	2
SET b,r	$r_b \leftarrow 1$	•	•	•	•	•	•	11	001	011	2	2	8	011	3
								11	b	r				100	4
								11	001	011				101	5
								11	b	110				110	6
SET b,(HL)	$(HL)_b \leftarrow 1$	•	•	•	•	•	•	11	001	011	2	4	15	111	7
								11	b	110					
								11	011	101					
								11	001	011					
SET b,(IX+d)	$(IX+d)_b \leftarrow 1$	•	•	•	•	•	•	←	d	→	4	6	23		
								11	b	110					
								11	111	101					
								11	001	011					
SET b,(IY+d)	$(IY+d)_b \leftarrow 1$	•	•	•	•	•	•	←	d	→	4	6	23		
								11	b	110					
								11	111	101					
								11	001	011					
RES b,s	$s_b \leftarrow 0$ $s \equiv r, (HL),$ $(IX+d),$ $(IY+d)$	•	•	•	•	•	•	11	b	110				Bit b in operand s is reset	
								10							

Jump Group

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments	
		C	Z	P/V	S	N	H	76	543	210					
JP nn	PC←nn	●	●	●	●	●	●	11	000	011	3	3	10		
								←	n	→					
								←	n	→					
JP cc,nn	if cc is true, PC←nn if false, continue	●	●	●	●	●	●	11	cc	010	3	3	10	cc	condition
								←	n	→				000	NZ non zero
								←	n	→				001	Z zero
JR e	PC←PC+e	●	●	●	●	●	●	00	011	000	2	3	12	010	NC non carry
								←	e-2	→				011	C carry
JR C,e	if C=0, continue	●	●	●	●	●	●	00	111	000	2	2	7	100	PO parity odd
								←	e-2	→				101	PE parity even
	if C=1, PC←PC+e										2	3	12	110	P sign positive
JR NC,e	if C=1, continue	●	●	●	●	●	●	00	110	000	2	2	7	111	M sign negative
	if C=0 PC←PC+e							←	e-2	→	2	3	12		
JR Z,e	if Z=0, continue	●	●	●	●	●	●	00	101	000	2	2	7		
								←	e-2	→					
	if Z=1, PC←PC+e										2	3	12		
JR NZ,e	if Z=1, continue	●	●	●	●	●	●	00	100	000	2	2	7		
								←	e-2	→					
	if Z=0 PC←PC+e										2	3	12		
JP (HL)	PC←HL	●	●	●	●	●	●	11	101	001	1	1	4		
JP (IX)	PC←IX	●	●	●	●	●	●	11	011	101	2	2	8		
								11	101	001					
JP (IY)	PC←IY	●	●	●	●	●	●	11	111	101	2	2	8		
								11	101	001					
DJNZ,e	B←B-1 if B=0, continue	●	●	●	●	●	●	00	010	000	2	2	8	for B=0	
								←	e-2	→					
	if B≠0 PC←PC+e										2	3	13	for B≠0	

Note : The value of the displacement e has a range of -126 to +129 bytes. the binnary number equivalent to e-2 must be placed in opcode.

mnemonic	operation	flags						opcode			bytes	No. of M cycles	No. of T cycles	comments																	
		C	Z	P/V	S	N	H	76	543	210																					
CALL nn	(SP-1)←PC _H (SP-2)←PC _L PC←nn	●	●	●	●	●	●	11	001	101	3	5	17																		
CALL cc,nn	if cc is true, equal to CALL nn if false, continue	●	●	●	●	●	●	← n →			3	3	10	for cc is false																	
								← n →																							
								← n →			3	5	17	for cc is true																	
RET	PC _L ←(SP) PC _H ←(SP+1)	●	●	●	●	●	●	11	001	001	1	3	10																		
RET cc	if cc is true, equal to RET if false, continue	●	●	●	●	●	●	11 cc 000			1	1	5	for cc is false																	
											1	3	11	for cc is true																	
RETI	return from interrupt	●	●	●	●	●	●				2	4	14	cc	condition																
								11 101 101						000	NZ non zero																
								01 001 101						001	Z zero																
RETN	return from NMI	●	●	●	●	●	●	11 101 101			2	4	14	010	NC non carry																
								01 000 101						011	C carry																
RST p	(SP-1)←PC _H (SP-2)←PC _L PC _H ←0 PC _L ←P	●	●	●	●	●	●	11 t 111			1	3	11	100	PO parity odd																
														101	PE parity even																
														110	P sign positive																
														111	M sign negative																
														t	P																
														000	00H																
														001	08H																
														010	10H																
														011	18H																

object code	mnemonic
4C	LD C, H
4D	LD C, L
4E	LD C, (HL)
4F	LD C, A
50	LD D, B
51	LD D, C
52	LD D, D
53	LD D, E
54	LD D, H
55	LD D, L
56	LD D, (HL)
57	LD D, A
58	LD E, B
59	LD E, C
5A	LD E, D
5B	LD E, E
5C	LD E, H
5D	LD E, L
5E	LD E, (HL)
5F	LD E, A
60	LD H, B
61	LD H, C
62	LD H, D
63	LD H, E
64	LD H, H
65	LD H, L
66	LD H, (HL)
67	LD H, A
68	LD L, B
69	LD L, C
6A	LD L, D
6B	LD L, E
6C	LD L, H
6D	LD L, L
6E	LD L, (HL)
6F	LD L, A
	LD (HL), B

object code	mnemonic
71	LD (HL), C
72	LD (HL), D
73	LD (HL), E
74	LD (HL), H
75	LD (HL), L
76	HALT
77	LD (HL), A
78	LD A, B
79	LD A, C
7A	LD A, D
7B	LD A, E
7C	LD A, H
7D	LD A, L
7E	LD A, (HL)
7F	LD A, A
80	ADD A, B
81	ADD A, C
82	ADD A, D
83	ADD A, E
84	ADD A, H
85	ADD A, L
86	ADD A, (HL)
87	ADD A, A
88	ADC A, B
89	ADC A, C
8A	ADC A, D
8B	ADC A, E
8C	ADC A, H
8D	ADC A, L
8E	ADC A, (HL)
8F	ADC A, A
90	SUB B
91	SUB C
92	SUB D
93	SUB E
94	SUB H
95	SUB L
96	SUB (HL)

object code	mnemonic
97	SUB A
98	SBC A,B
99	SBC A,C
9A	SBC A,D
9B	SBC A,E
9C	SBC A,H
9D	SBC A,L
9E	SBC A,(HL)
9F	SBC A,A
A0	AND B
A1	AND C
A2	AND D
A3	AND E
A4	AND H
A5	AND L
A6	AND (HL)
A7	AND A
A8	XOR B
A9	XOR C
AA	XOR D
AB	XOR E
AC	XOR H
AD	XOR L
AE	XOR (HL)
AF	XOR A
B0	OR B
B1	OR C
B2	OR D
B3	OR E
B4	OR H
B5	OR L
B6	OR (HL)
B7	OR A
B8	CP B
B9	CP C
BA	CP D
BB	CP E
BC	CP H

object code	mnemonic
BD	CP L
BE	CP (HL)
BF	CP A
C0	RET NZ
C1	POP BC
<u>C28405</u>	JP NZ,nn
<u>C38405</u>	JP nn
<u>C48405</u>	CALL NZ,nn
C5	PUSH BC
<u>C620</u>	ADD A,n
C7	RST 0
C8	RET Z
C9	RET
<u>CA8405</u>	JP Z,nn
<u>CC8405</u>	CALL Z,nn
<u>CD8405</u>	CALL nn
<u>CE20</u>	ADC A,n
CF	RST 1
D0	RET NC
D1	POP DE
<u>D28405</u>	JP NC,nn
<u>D320</u>	OUT (n),A
<u>D48405</u>	CALL NC,nn
D5	PUSH DE
<u>D620</u>	SUB n
D7	RST 2
D8	RET C
D9	EXX
<u>DA8405</u>	JP C,nn
<u>DB20</u>	IN A,(n)
<u>DC8405</u>	CALL C,nn
<u>DE20</u>	SBC A,n
DF	RST 3
E0	RET PO
E1	POP HL
<u>E28405</u>	JP PO,nn
E3	EX (SP),HL

object code	mnemonic
E4 <u>8405</u>	CALL PO,nn
E5	PUSH HL
E6 <u>20</u>	AND n
E7	RST 4
E8	RET PE
E9	JP (HL)
EA <u>8405</u>	JP PE,nn
EB	EX DE,HL
EC <u>8405</u>	CALL PE,nn
EE <u>20</u>	XOR n
EF	RST 5
F0	RET P
F1	POP AF
F2 <u>8405</u>	JP P,nn
F3	DI
F4 <u>8405</u>	CALL P,nn
F5	PUSH AF
F6 <u>20</u>	OR n
F7	RST 6
F8	RET M
F9	LD SP,HL
FA <u>8405</u>	JP M,nn
FB	EI
FC <u>8405</u>	CALL M,nn
FE <u>20</u>	CP n
FF	RST 7
CB00	RLC B
CB01	RLC C
CB02	RLC D
CB03	RLC E
CB04	RLC H
CB05	RLC L
CB06	RLC (HL)
CB07	RLC A
CB08	RRC B
CB09	RRC C
CB0A	RRC D
CB0B	RRC E

object code	mnemonic
CB0C	RRC H
CB0D	RRC L
CB0E	RRC (HL)
CB0F	RRC A
CB10	RL B
CB11	RL C
CB12	RL D
CB13	RL E
CB14	RL H
CB15	RL L
CB16	RL (HL)
CB17	RL A
CB18	RR B
CB19	RR C
CB1A	RR D
CB1B	RR E
CB1C	RR H
CB1D	RR L
CB1E	RR (HL)
CB1F	RR A
CB20	SLA B
CB21	SLA C
CB22	SLA D
CB23	SLA E
CB24	SLA H
CB25	SLA L
CB26	SLA (HL)
CB27	SLA A
CB28	SRA B
CB29	SRA C
CB2A	SRA D
CB2B	SRA E
CB2C	SRA H
CB2D	SRA L
CB2E	SRA (HL)
CB2F	SRA A
CB38	SRL B

object code	mnemonic
CB39	SRL C
CB3A	SRL D
CB3B	SRL E
CB3C	SRL H
CB3D	SRL L
CB3E	SRL (HL)
CB3F	SRL A
CB40	BIT 0,B
CB41	BIT 0,C
CB42	BIT 0,D
CB43	BIT 0,E
CB44	BIT 0,H
CB45	BIT 0,L
CB46	BIT 0,(HL)
CB47	BIT 0,A
CB48	BIT 1,B
CB49	BIT 1,C
CB4A	BIT 1,D
CB4B	BIT 1,E
CB4C	BIT 1,H
CB4D	BIT 1,L
CB4E	BIT 1,(HL)
CB4F	BIT 1,A
CB50	BIT 2,B
CB51	BIT 2,C
CB52	BIT 2,D
CB53	BIT 2,E
CB54	BIT 2,H
CB55	BIT 2,L
CB56	BIT 2,(HL)
CB57	BIT 2,A
CB58	BIT 3,B
CB59	BIT 3,C
CB5A	BIT 3,D
CB5B	BIT 3,E
CB5C	BIT 3,H
CB5D	BIT 3,L
CB5E	BIT 3,(HL)

object code	mnemonic
CB5F	BIT 3,A
CB60	BIT 4,B
CB61	BIT 4,C
CB62	BIT 4,D
CB63	BIT 4,E
CB64	BIT 4,H
CB65	BIT 4,L
CB66	BIT 4,(HL)
CB67	BIT 4,A
CB68	BIT 5,B
CB69	BIT 5,C
CB6A	BIT 5,D
CB6B	BIT 5,E
CB6C	BIT 5,H
CB6D	BIT 5,L
CB6E	BIT 5,(HL)
CB6F	BIT 5,A
CB70	BIT 6,B
CB71	BIT 6,C
CB72	BIT 6,D
CB73	BIT 6,E
CB74	BIT 6,H
CB75	BIT 6,L
CB76	BIT 6,(HL)
CB77	BIT 6,A
CB78	BIT 7,B
CB79	BIT 7,C
CB7A	BIT 7,D
CB7B	BIT 7,E
CB7C	BIT 7,H
CB7D	BIT 7,L
CB7E	BIT 7,(HL)
CB7F	BIT 7,A
CB80	RES 0,B
CB81	RES 0,C
CB82	RES 0,D
CB83	RES 0,E

object code	mnemonic
CB84	RES 0,H
CB85	RES 0,L
CB86	RES 0,(HL)
CB87	RES 0,A
CB88	RES 1,B
CB89	RES 1,C
CB8A	RES 1,D
CB8B	RES 1,E
CB8C	RES 1,H
CB8D	RES 1,L
CB8E	RES 1,(HL)
CB8F	RES 1,A
CB90	RES 2,B
CB91	RES 2,C
CB92	RES 2,D
CB93	RES 2,E
CB94	RES 2,H
CB95	RES 2,L
CB96	RES 2,(HL)
CB97	RES 2,A
CB98	RES 3,B
CB99	RES 3,C
CB9A	RES 3,D
CB9B	RES 3,E
CB9C	RES 3,H
CB9D	RES 3,L
CB9E	RES 3,(HL)
CB9F	RES 3,A
CBA0	RES 4,B
CBA1	RES 4,C
CBA2	RES 4,D
CBA3	RES 4,E
CBA4	RES 4,H
CBA5	RES 4,L
CBA6	RES 4,(HL)
CBA7	RES 4,A
CBA8	RES 5,B
CBA9	RES 5,C

object code	mnemonic
CBAA	RES 5,D
CBAB	RES 5,E
CBAC	RES 5,H
CBAD	RES 5,L
CBAE	RES 5,(HL)
CBAF	RES 5,A
CBB0	RES 6,B
CBB1	RES 6,C
CBB2	RES 6,D
CBB3	RES 6,E
CBB4	RES 6,H
CBB5	RES 6,L
CBB6	RES 6,(HL)
CBB7	RES 6,A
CBB8	RES 7,B
CBB9	RES 7,C
CBBA	RES 7,D
CBBB	RES 7,E
CBBC	RES 7,H
CBBD	RES 7,L
CBBE	RES 7,(HL)
CBBF	RES 7,A
CBC0	SET 0,B
CBC1	SET 0,C
CBC2	SET 0,D
CBC3	SET 0,E
CBC4	SET 0,H
CBC5	SET 0,L
CBC6	SET 0,(HL)
CBC7	SET 0,A
CBC8	SET 1,B
CBC9	SET 1,C
CBCA	SET 1,D
CBCB	SET 1,E
CBCC	SET 1,H
CBCD	SET 1,L
CBCE	SET 1,(HL)
CBCF	SET 1,A

object code	mnemonic
CBD0	SET 2,B
CBD1	SET 2,C
CBD2	SET 2,D
CBD3	SET 2,E
CBD4	SET 2,H
CBD5	SET 2,L
CBD6	SET 2,(HL)
CBD7	SET 2,A
CBD8	SET 3,B
CBD9	SET 3,C
CBDA	SET 3,D
CBDB	SET 3,E
CBDC	SET 3,H
CBDD	SET 3,L
CBDE	SET 3,(HL)
CBDF	SET 3,A
CBE0	SET 4,B
CBE1	SET 4,C
CBE2	SET 4,D
CBE3	SET 4,E
CBE4	SET 4,H
CBE5	SET 4,L
CBE6	SET 4,(HL)
CBE7	SET 4,A
CBE8	SET 5,B
CBE9	SET 5,C
CBEA	SET 5,D
CBEB	SET 5,E
CBEC	SET 5,H
CBED	SET 5,L
CBEE	SET 5,(HL)
CBEF	SET 5,A
CBF0	SET 6,B
CBF1	SET 6,C
CBF2	SET 6,D
CBF3	SET 6,E
CBF4	SET 6,H
CBF5	SET 6,L

object code	mnemonic
CBF6	SET 6,(HL)
CBF7	SET 6,A
CBF8	SET 7,B
CBF9	SET 7,C
CBFA	SET 7,D
CBFB	SET 7,E
CBFC	SET 7,H
CBFD	SET 7,L
CBFE	SET 7,(HL)
CBFF	SET 7,A
DD09	ADD IX,BC
DD19	ADD IX,DE
DD21 ⁸⁴⁰⁵	LD IX,nn
DD22 ⁸⁴⁰⁵	LD (nn),IX
DD23	INC IX
DD29	ADD IX,IX
DD2A ⁸⁴⁰⁵	LD IX,(nn)
DD2B	DEC IX
DD34 ⁰⁵	INC (IX+d)
DD35 ⁰⁵	DEC (IX+d)
DD3605 ⁰⁵²⁰	LD (IX+d),n
DD39	ADD IX,SP
DD46 ⁰⁵	LD B,(IX+d)
DD4E ⁰⁵	LD C,(IX+d)
DD56 ⁰⁵	LD D,(IX+d)
DD5E ⁰⁵	LD E,(IX+d)
DD66 ⁰⁵	LD H,(IX+d)
DD6E ⁰⁵	LD L,(IX+d)
DD70 ⁰⁵	LD (IX+d),B
DD71 ⁰⁵	LD (IX+d),C
DD72 ⁰⁵	LD (IX+d),D
DD73 ⁰⁵	LD (IX+d),E
DD74 ⁰⁵	LD (IX+d),H
DD75 ⁰⁵	LD (IX+d),L
DD77 ⁰⁵	LD (IX+d),A
DD7E ⁰⁵	LD A,(IX+d)
DD86 ⁰⁵	ADD A,(IX+d)
DD8E ⁰⁵	ADC A,(IX+d)
DD96 ⁰⁵	SUB (IX+d)

object code	mnemonic
DD9E05	SBC A,(IX+d)
DDA605	AND (IX+d)
DDAE05	XOR (IX+d)
DDB605	OR (IX+d)
DDBE05	CP (IX+d)
DDE1	POP IX
DDE3	EX (SP),IX
DDE5	PUSH IX
DDE9	JP (IX)
DDF9	LD SP,IX
DDCB0506	RLC (IX+d)
DDCB050E	RRC (IX+d)
DDCB0516	RL (IX+d)
DDCB051E	RR (IX+d)
DDCB0526	SLA (IX+d)
DDCB052E	SRA (IX+d)
DDCB053E	SRL (IX+d)
DDCB0546	BIT 0,(IX+d)
DDCB054E	BIT 1,(IX+d)
DDCB0556	BIT 2,(IX+d)
DDCB055E	BIT 3,(IX+d)
DDCB0566	BIT 4,(IX+d)
DDCB056E	BIT 5,(IX+d)
DDCB0576	BIT 6,(IX+d)
DDCB057E	BIT 7,(IX+d)
DDCB0586	RES 0,(IX+d)
DDCB058E	RES 1,(IX+d)
DDCB0596	RES 2,(IX+d)
DDCB059E	RES 3,(IX+d)
DDCB05A6	RES 4,(IX+d)
DDCB05AE	RES 5,(IX+d)
DDCB05B6	RES 6,(IX+d)
DDCB05BE	RES 7,(IX+d)
DDCB05C6	SET 0,(IX+d)
DDCB05CE	SET 1,(IX+d)
DDCB05D6	SET 2,(IX+d)
DDCB05DE	SET 3,(IX+d)
DDCB05E6	SET 4,(IX+d)
DDCB05EE	SET 5,(IX+d)

object code	mnemonic
DDCB05F6	SET 6,(IX+d)
DDCB05FE	SET 7,(IX+d)
ED40	IN B,(C)
ED41	OUT (C),B
ED42	SBC HL,BC
ED438405	LD (nn),BC
ED44	NEG
ED45	RETN
ED46	IM 0
ED47	LD I,A
ED48	IN C,(C)
ED49	OUT (C),C
ED4A	ADC HL,BC
ED4B8405	LD BC,(nn)
ED4D	RETI
ED50	IN D,(C)
ED51	OUT (C),D
ED52	SBC HL,DE
ED538405	LD (nn),DE
ED56	IM 1
ED57	LD A,I
ED58	IN E,(C)
ED59	OUT (C),E
ED5A	ADC HL,DE
ED5B8405	LD DE,(nn)
ED5E ED5F ED60	IM 2 LD A,R IN H,(C)
ED61	OUT (C),H
ED62	SBC HL,HL
ED67	RRD
ED68	IN L,(C)
ED69	OUT (C),L
ED6A	ADC HL,HL
ED6F	RLD
ED72	SBC HL,SP
ED738405	LD (nn),SP
ED78	IN A,(C)
ED79	OUT (C),A
ED7A	ADC HL,SP

object code	mnemonic
ED7B ⁸⁴⁰⁵	LD SP,(nn)
EDA0	LDI
EDA1	CPI
EDA2	INI
EDA3	OUTI
EDA8	LDD
EDA9	CPD
EDAA	IND
EDAB	OUTD
EDB0	LDIR
EDB1	CPIR
EDB2	INIR
EDB3	OTIR
EDB8	LDDR
EDB9	CPDR
EDBA	INDR
EDBB	OTDR
FD09	ADD IY,BC
FD19	ADD IY,DE
FD21 ⁸⁴⁰⁵	LD IY,nn
FD22 ⁸⁴⁰⁵	LD (nn),IY
FD23	INC IY
FD29	ADD IY,IY
FD2A ⁸⁴⁰⁵	LD IY,(nn)
FD2B	DEC IY
FD34 ⁰⁵	INC (IY+d)
FD35 ⁰⁵	DEC (IY+d)
FD36 ⁰⁵²⁰	LD (IY+d),n
FD39	ADD IY,SP
FD46 ⁰⁵	LD B,(IY+d)
FD4E ⁰⁵	LD C,(IY+d)
FD56 ⁰⁵	LD D,(IY+d)
FD5E ⁰⁵	LD E,(IY+d)
FD66 ⁰⁵	LD H,(IY+d)
FD6E ⁰⁵	LD L,(IY+d)
FD70 ⁰⁵	LD (IY+d),B
FD71 ⁰⁵	LD (IY+d),C
FD72 ⁰⁵	LD (IY+d),D
FD73 ⁰⁵	LD (IY+d),E

object code	mnemonic
FD74 ⁰⁵	LD (IY+d),H
FD75 ⁰⁵	LD (IY+d),L
FD77 ⁰⁵	LD (IY+d),A
FD7E ⁰⁵	LD A,(IY+d)
FD86 ⁰⁵	ADD A,(IY+d)
FD8E ⁰⁵	ADC A,(IY+d)
FD96 ⁰⁵	SUB (IY+d)
FD9E ⁰⁵	SBC A,(IY+d)
FDA6 ⁰⁵	AND (IY+d)
FDAE ⁰⁵	XOR (IY+d)
FDB6 ⁰⁵	OR (IY+d)
FDBE ⁰⁵	CP (IY+d)
FDE1	POP IY
FDE3	EX (SP),IY
FDE5	PUSH IY
FDE9	JP,(IY)
FDF9	LD SP,IY
FDCB ⁰⁵⁰⁶	RLC (IY+d)
FDCB ^{050E}	RRC (IY+d)
FDCB ⁰⁵¹⁶	RL (IY+d)
FDCB ^{051E}	RR (IY+d)
FDCB ⁰⁵²⁶	SLA (IY+d)
FDCB ^{052E}	SRA (IY+d)
FDCB ^{053E}	SRL (IY+d)
FDCB ⁰⁵⁴⁶	BIT 0,(IY+d)
FDCB ^{054E}	BIT 1,(IY+d)
FDCB ⁰⁵⁵⁶	BIT 2,(IY+d)
FDCB ^{055E}	BIT 3,(IY+d)
FDCB ⁰⁵⁶⁶	BIT 4,(IY+d)
FDCB ^{056E}	BIT 5,(IY+d)
FDCB ⁰⁵⁷⁶	BIT 6,(IY+d)
FDCB ^{057E}	BIT 7,(IY+d)
FDCB ⁰⁵⁸⁶	RES 0,(IY+d)
FDCB ^{058E}	RES 1,(IY+d)
FDCB ⁰⁵⁹⁶	RES 2,(IY+d)
FDCB ^{059E}	RES 3,(IY+d)
FDCB ^{05A6}	RES 4,(IY+d)
FDCB ^{05AE}	RES 5,(IY+d)
FDCB ^{05B6}	RES 6,(IY+d)

object code	mnemonic
FDCB <u>05</u> BE	RES 7,(IY+d)
FDCB <u>05</u> C6	SET 0,(IY+d)
FDCB <u>05</u> CE	SET 1,(IY+d)
FDCB <u>05</u> D6	SET 2,(IY+d)
FDCB <u>05</u> DE	SET 3,(IY+d)
FDCB <u>05</u> E6	SET 4,(IY+d)
FDCB <u>05</u> EE	SET 5,(IY+d)
FDCB <u>05</u> F6	SET 6,(IY+d)
FDCB <u>05</u> FE	SET 7,(IY+d)

Value Examples :

nn, n, d and e are the value examples.

where :

nn=584H

n= 20H

d= 5

e= 30H

In the object code column, the code equivalent to this value is indicated in *italics* and by an underline.

INDEX

[A]

access, read/write	21
access timing, VRAM	13
add/subtract flag	138
ALPHA ROTATE	41
ALL INITIALIZE	41
ASC	54
AXIS (AXIS)	41

[B]

back space	40
BELL	52
bit set	149
block transfer	144
border color	23
BRKEY	53

[C]

call (instruction)	151
carriage return	40
carry flag	138
cassette control	26
cassette recorder circuit	27
C.G. ROM, modify	45
C.G. table	46
character generator	45
character scale	40
character set	37
character set, table	39
clock generator	9
COLOR CHANGE (PCOLOR)	41
color position detector	37
CPU control (instruction)	145
custom LSI	6
custom LSI block diagram	8
cycle steal mode	15

[D]

DA (display address)	12
disk control	100
display address generator	9
display mode register (DMD)	17
display period	16
display timing	14
DMA (display memory address)	12
DRAW (LINE)	41

[E]

ELMD	55
ELMD1	55
ELMD20	55
ELMD22	55
ELMD24	55
ERRORP	55
exchange	144

[F]

FILOUT	55
flags	138
flyback period	16
FONT	45
form feed	40

[G]

general purpose arithmetic (instruction)	145
GETKY	53
GETL	53
graphic code	40
graphic mode	41

[H]

half carry flag	139
HLHEX	54
HOME (PHONE)	41

[I]

INITIALIZE (HSET)	41
input (instruction)	152

[J]

joystick	32
jump (instruction)	150

[K]

key scan	25
----------------	----

[L]

LETNL	52
line counter set	40
line feed	40
line scroll	10

LINE TYPE	41
line up	40
logical (instruction)	146

[M]

MA	12
magnify scale	40
MELDY	52
memory bank	4
memory controller	6
memory map	3
monitor <1Z-013B>	67
monitor call	55
monitor subroutines	52
monitor variable	55
MOVE (MOVE)	41
MSG	52
MSTA	52
MSTP	53
MZ1P16	35
MZ1P16, specification	42

[N]

next colour	40
notation	140

[O]

output (instruction)	152
overflow (flag)	139

[P]

pallet	22
parity/overflow flag	139
pen exchange method	37
pen exchange operation	36
pen test	40
plotter printer control codes	40
plotter printer control LSI	38
POOL	55
power supply	33
PRINT	41
PRNTS	52
pseud cycle steal	13

[R]

range of scroll	12
-----------------------	----

read format register	19
read/write access	21
reduction cancel	40
reduction scale	40
RELATIVE DRAW (RLINE)	41
RELATIVE MOVE (RMOVE)	41
reset (instruction)	149
return (instruction)	151
ROM configuration	24
ROM monitor	52
ROM, C.G.	45
rotate (instruction)	148

[S]

SCALE SET	41
screen split	10
scroll	10,11
scroll method	12
scroll sequence	12
scroll, range of	12
scrolling, address conversion	12
SEA	10,11
search (instruction)	144
shift (instruction)	148
sign flag	139
special key (read with GETKEY)	53
specification, MZ1P16	42
SSA	10,11
status indicators	138
stepping motor	37
subtract (flag)	138
SW	10,11
SYSSTA	55
system description	1
system switches	32

[T]

TEMLMT	55
test (instruction)	149
text code	40
TEXTST	55
timing generator	9
TIMRD	53
TIMST	53
TMPEND	55

[V]

VARST	55
VCAS	16
VRAM access timing	13
VRAM accessing	16
VRAM configuration	17
VRAM, I/O circuit	12,13

[W]

write format register (WR)	20
----------------------------------	----

[X]

XTEMP	52
-------------	----

[Z]

Z-80 programming reference	138
zero flag	138
ZLOG	55
ZRWX	55

[Others]

16-bit arithmetic (instruction)	147
16-bit load (instruction)	142
2HEX	54
8-bit arithmetic (instruction)	146
8-bit load (instruction)	141
8050, pin configuration	37
8253	28
8255	24,25
.BREAK	58
.CLKL	61
.COUNT	58
.CRTMS	56
.CRT1C	56
.CRT1X	56
.CR1	55
.CR2	56
.DEVFN	59
.DEVNM	59
.DI	58
.DIR	61
.EI	58
.ERCVR	62
.ERRX	59
.FINFT	62
.GETL	57
.HALT	58
.INKEY	58

.INMSG	61
.LOADF	60
.LOPEN	59
.LPTOT	56
.LPT1C	56
.LUCHK	59
.MONOP	55
.PRSTR	61
.RWOPN	60
.SAVEF	60
.SETDF	62
.VRFYF	60
.&CR	57
.&MSG	57
.&1C	57
.&1CX	57
?ADCN	54
?BLNK	54
?DACN	54
?DPCT	54
?POINT	54
??KEY	54

SHARP CORPORATION
OSAKA, JAPAN

Printed in Japan
Gedruckt in Japan
Imprimé au Japon
Stampato in Giappone

©1985 SHARP CORPORATION
5E 2.25-S(TINSE1318ACZZ)①