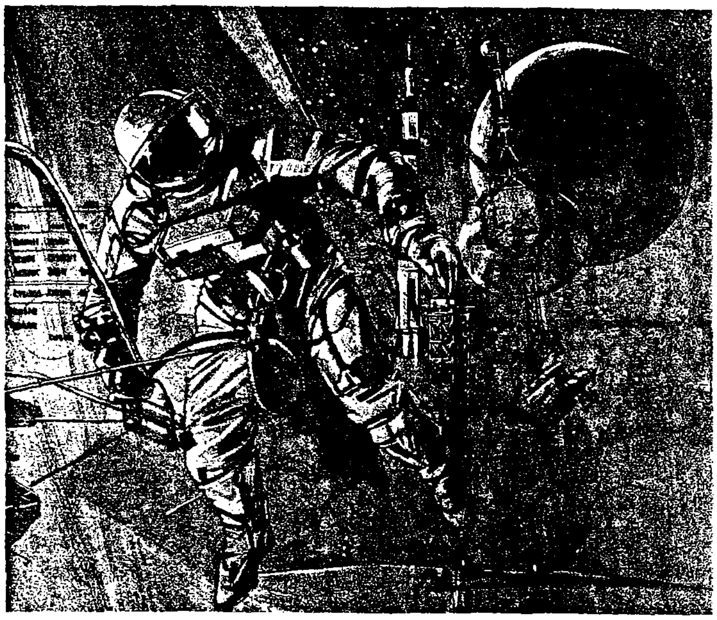
Personal Computer

MZ-800

TECHNICAL REFERENCE MANUAL



SHARP

Personal Computer

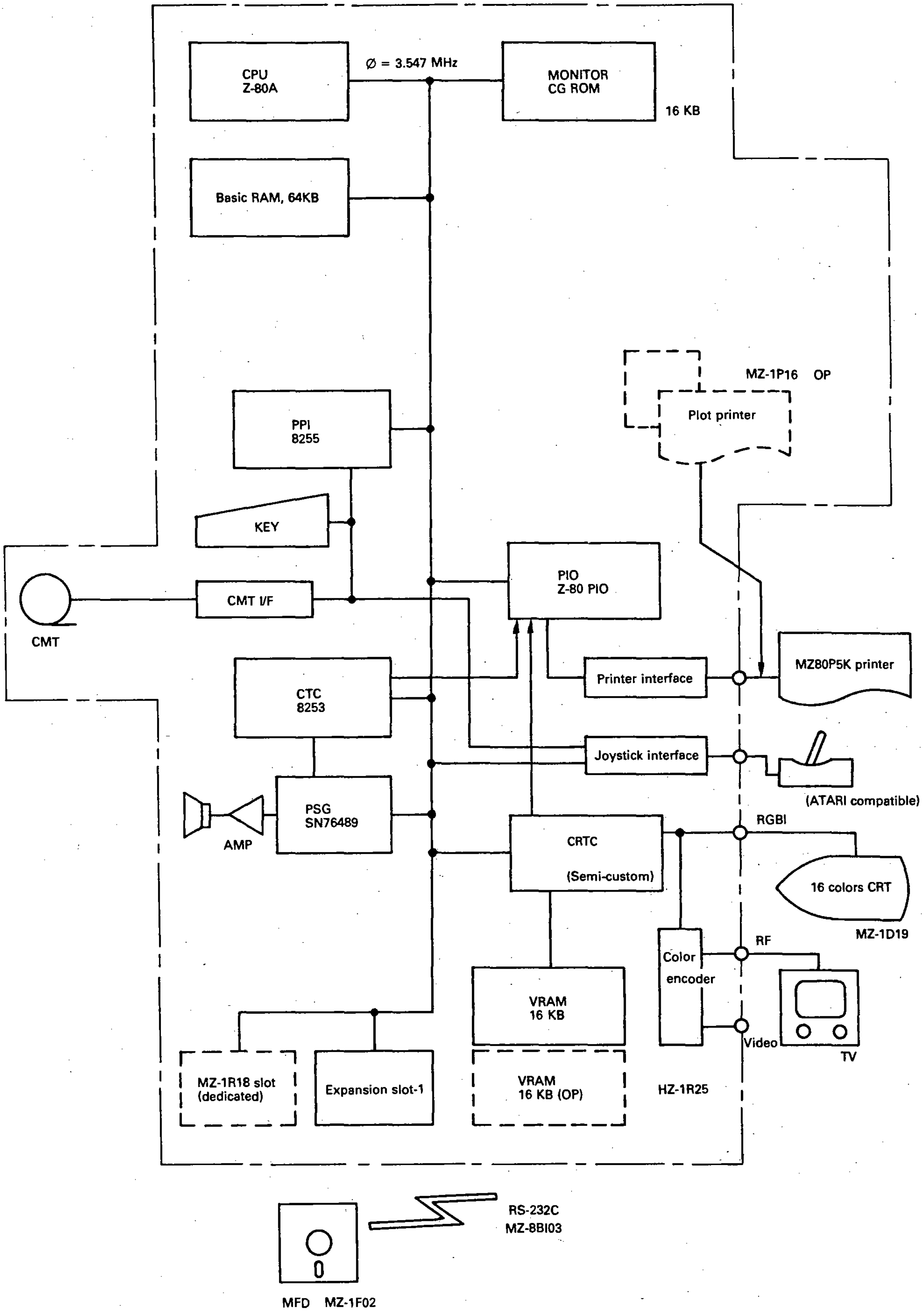
MZ-800

TECHNICAL REFERENCE MANUAL

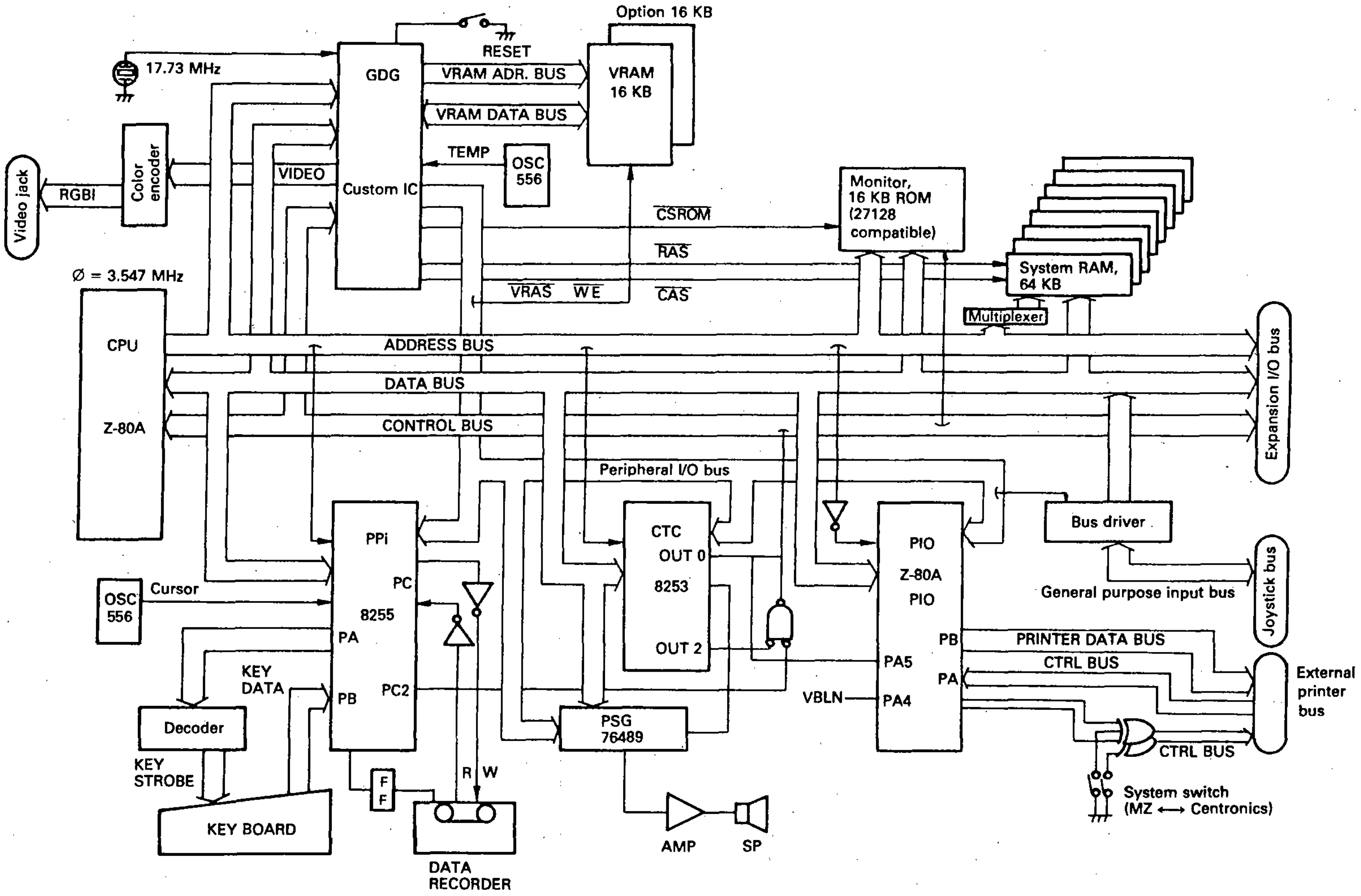
MZ-800 HARDWARE

MZ-800 SOFTWARE

1 System description

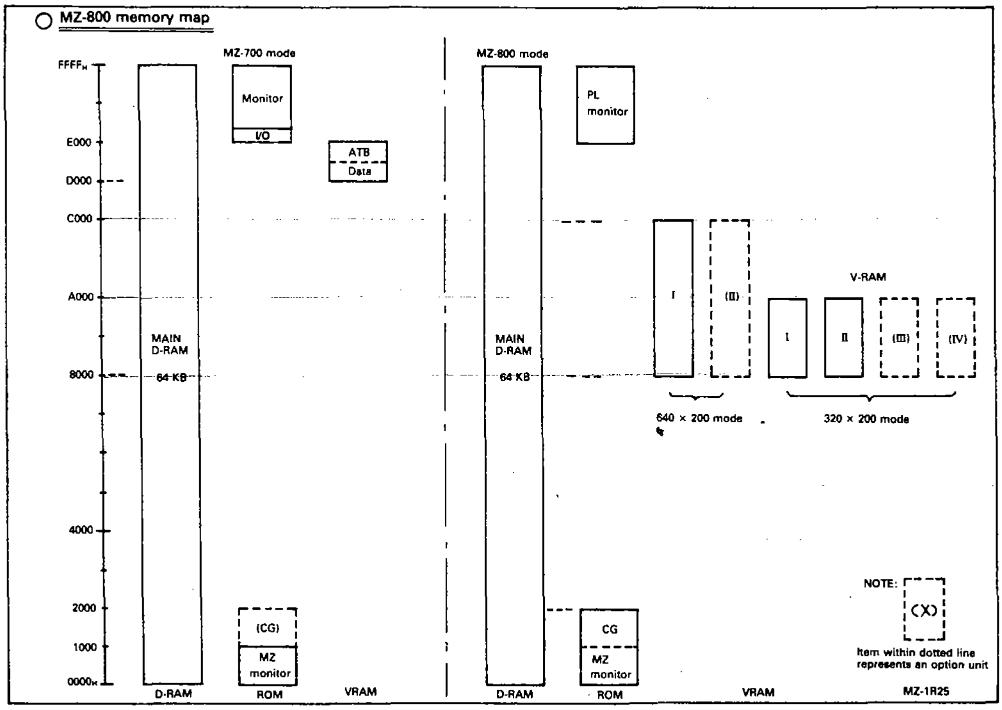


2 Block diagram

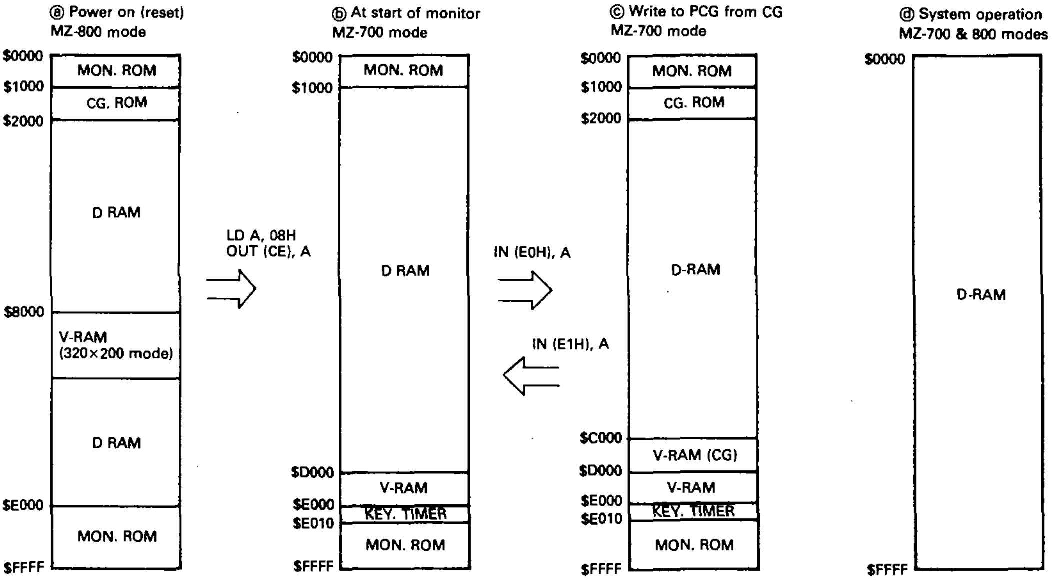


3 Memory map

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.



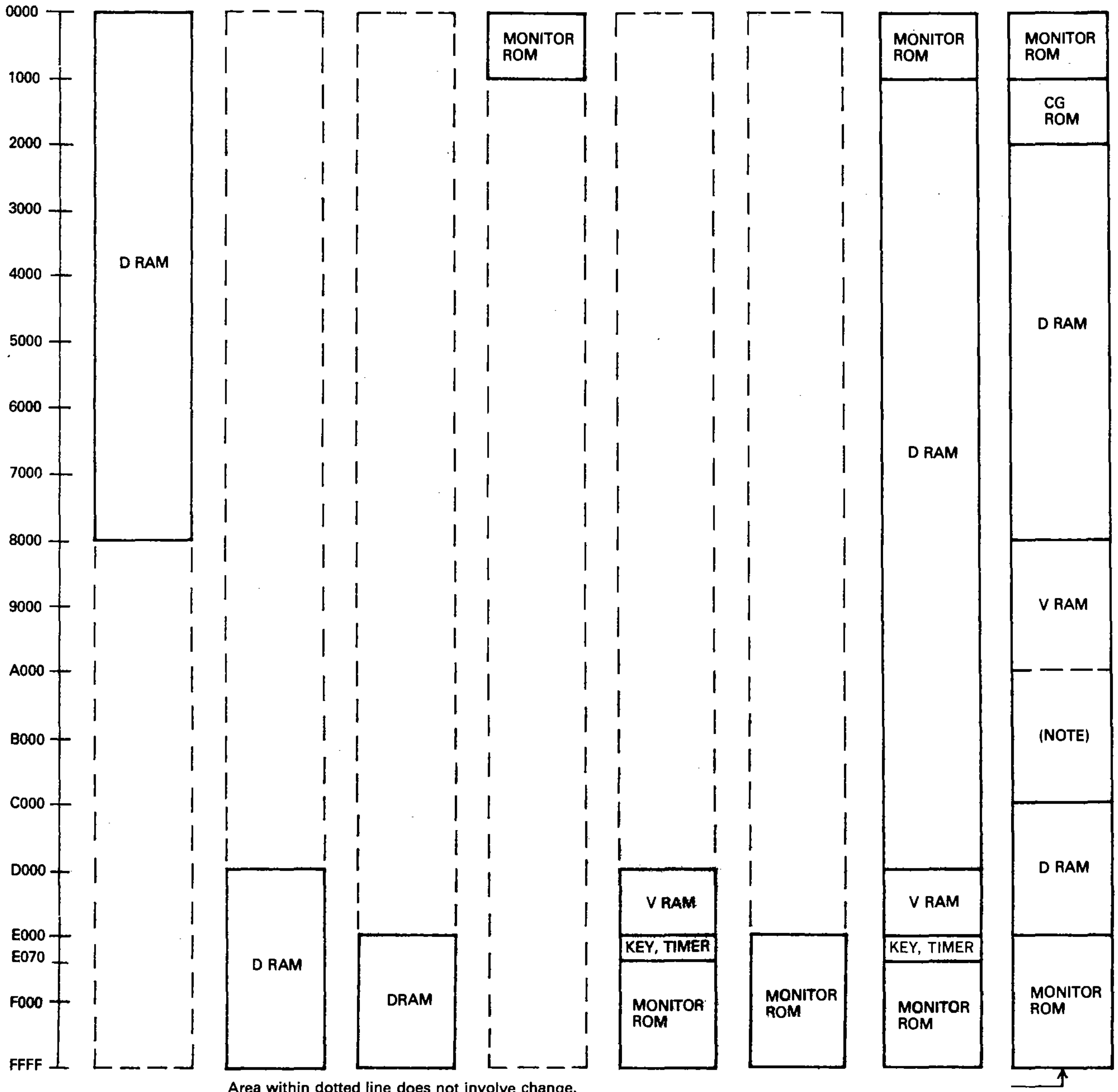
Memory map changes after initial program loading



- Memory map at power on is in the MZ-800 mode as in ⓐ, 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 ⓒ, the memory map then returns to ⓑ.
- 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 ⓓ. 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 ⓐ → ⓑ → ⓒ → ⓑ, 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 ⓓ 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 × 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	o $0000 ~ $7FFF to DRAM.	o $D000 ~ $FFFF to DRAM.	o $E000 ~ $FFFF to DRAM.	o $0000 ~ $0FFF to monitor ROM.	o $D000 ~ $FFFF to VRAM, key timer, and monitor ROM.	o $E000 ~ $FFFF to monitor ROM.	o $0000 ~ $0FFF to monitor ROM. o $1000 ~ $CFFF to DRAM o $D000 ~ $FFFF to VRAM, key timer, and monitor ROM.	o $0000 ~ $0FFF to monitor ROM. o $1000 ~ $1FFF to CG ROM. o $2000 ~ $7FFF and $C000 ~ $DFFF to DRAM. o $8000 ~ $BFFF to VRAM (NOTE). o $E000 ~ $FFFF to monitor ROM.

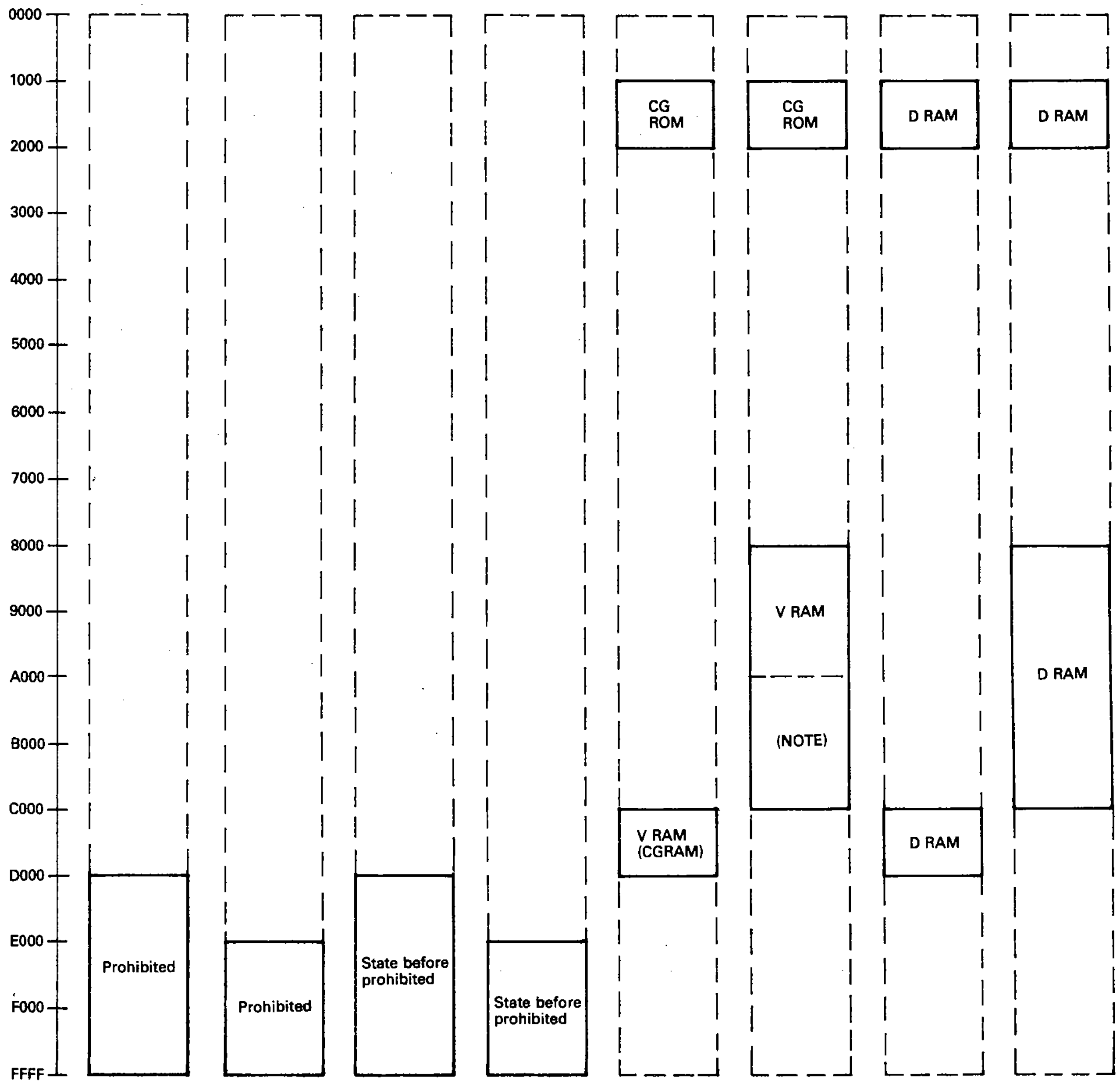


Area within dotted line does not involve change.

(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	o $D000 ~ $7FFF prohibited.	o $E000 ~ $FFFF prohibited.	o $D000 ~ $FFFF returned to the state before prohibited.	o $E000 ~ $FFFF returned to the state before prohibited.	o $1000 ~ $1FFF to CG ROM. o $C000 ~ $CFFF to VRAM (PCG RAM).	o $1000 ~ $1FFF to CG ROM. o $8000 ~ $BFFF to VRAM (NOTE).	o $1000 ~ $1FFF returned to the state before CG was set. o $C000 ~ $CFFF to DRAM.	o $1000 ~ $1FFF returned to the state before CG was set. o $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.

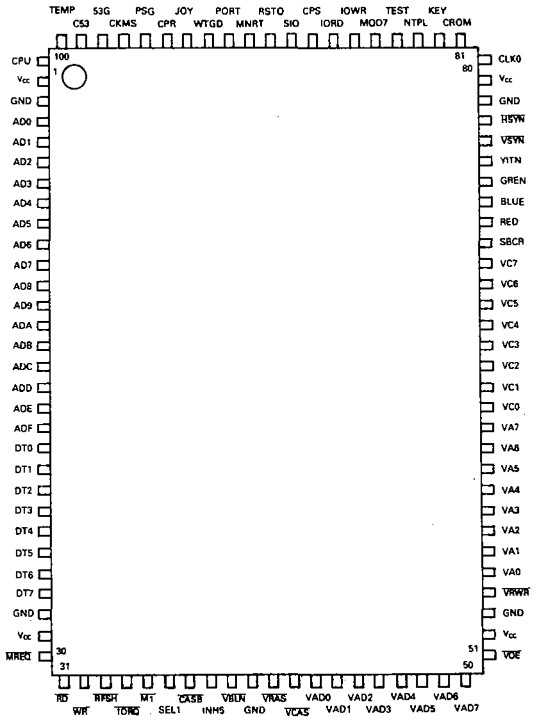
I/O address	Signal name	Device (I/O)	Function
FF FE FD FC	CPR	Z80A PIO (I/O)	Port B, printer data output Port A, printer control and timer interrupt Port B control (Mode 0) Port A control (Mode 3)
F2	PSG	PSG (O)	PSG output port
F1 F0	JOY	JOYSTICK (I)	Joystick-2 input port Joystick-1 input port
F0	——	(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 ≀ 19	AD0 ≀ ADF	I	CPU address bus	
20 ≀ 27	DT0 ≀ DT7	I/O	CPU data bus	
28	GND	—	Ground	
29	VCC	—	Power supply	
30	$\overline{MREQ}$	I	CPU $\overline{MREQ}$ signal	Negative logic
31	$\overline{RD}$	I	CPU $\overline{RD}$ signal	Negative logic
32	$\overline{WR}$	I	CPU $\overline{WR}$ signal	Negative logic
33	$\overline{RFSH}$	I	CPU $\overline{RFSH}$ signal	Negative logic
34	$\overline{IORQ}$	I	CPU $\overline{IORQ}$ signal	Negative logic
35	$\overline{M1}$	I	CPU $\overline{M1}$ 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	$\overline{VBLN}$	O	Vertical blanking signal	Negative logic
40	GND	—		
41	$\overline{VRAS}$	O	VRAM RAS control signal	Negative logic
42	$\overline{VCAS}$	O	VRAM CAS control signal	Negative logic
43 ≀ 50	VAD0 ≀ VAD7	O	VRAM address signal (multiplexer output)	
51	$\overline{VOE}$	O	VRAM output enable	Negative logic
52	VCC	—	Power supply	
53	GND	—	Ground	
54	$\overline{VRWR}$	O	VRAM write signal	Negative logic
55 ≀ 62	VA0 ≀ VA7	I/O	VRAM data bus (standard RAM)	
63 ≀ 70	VC0 ≀ VC7	I/O	VRAM data bus (option RAM)	
71	SBCR	O	Color sub-carrier wave	
72	RED	O	Video signal, red	
73	BLUE	O	Video signal, blue	
74	GREN	O	Video signal, green	
75	YITN	O	Brightness control signal	
76	$\overline{VSYN}$	O	Vertical sync signal	Negative logic
77	$\overline{HSYN}$	O	Horizontal sync signal	Negative logic
78	GND	—		
79	VCC	—		
80	CLK0	I	Clock input (17.7344 MHz)	
81	$\overline{CROM}$	O	ROM chip enable	Negative logic
82	$\overline{KEY}$	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	$\overline{IOWR}$	O	Sum of CS and WR of I/O controlled by the custom IC	Negative logic
87	$\overline{IORD}$	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	$\overline{RSTO}$	O	Reset output	Negative logic
91	$\overline{MNRT}$	I	Manual reset input	Negative logic
92	$\overline{PORT}$	I	Power on reset input	Negative logic
93	WTGD	O	Wait signal to CPU	Open drain
94	$\overline{JOY}$	O	Joystick chip enable	Negative logic
95	$\overline{CPR}$	O	PIO chip select	Negative logic
96	$\overline{PSG}$	O	76489 chip select	Negative logic
97	CKMS	O	8253 musical interval clock	
98	53G	O	8253 musical interval ON/OFF gate signal	
99	$\overline{C53}$	O	8253 chip enable	Negative logic
100	TEMP	I	MZ-700 mode, $E800 tempo input	

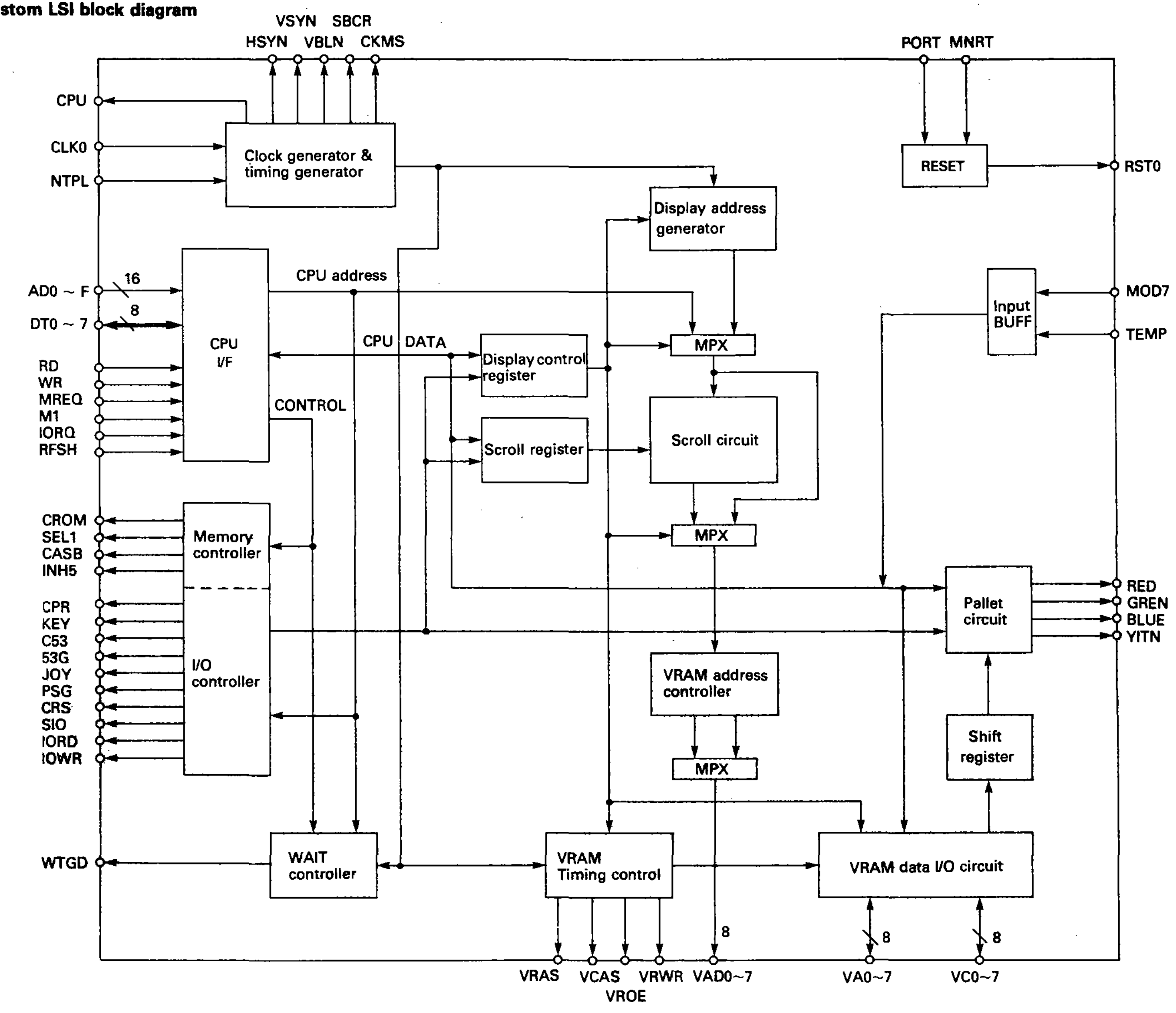
* Term "OPEN" represents the signal not used on the board.

Pin configuration



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

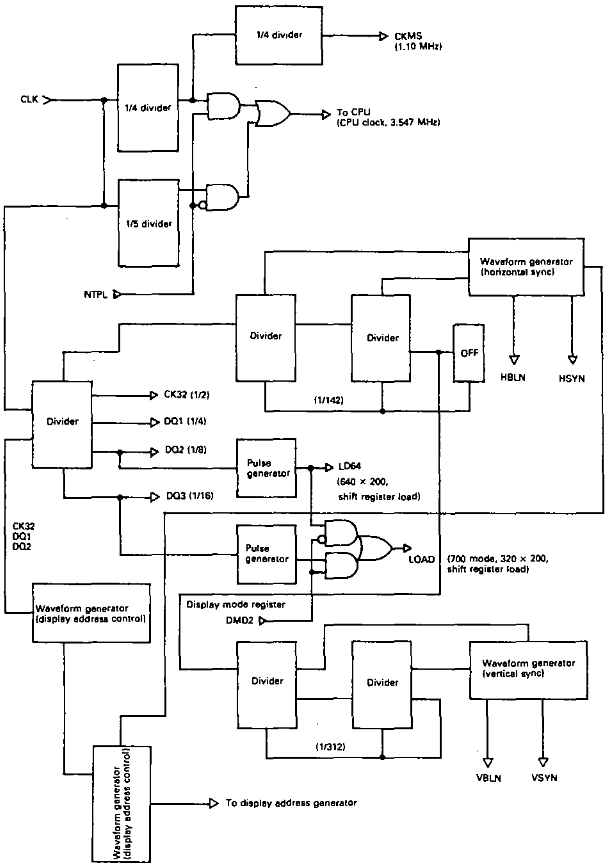
Custom LSI block diagram



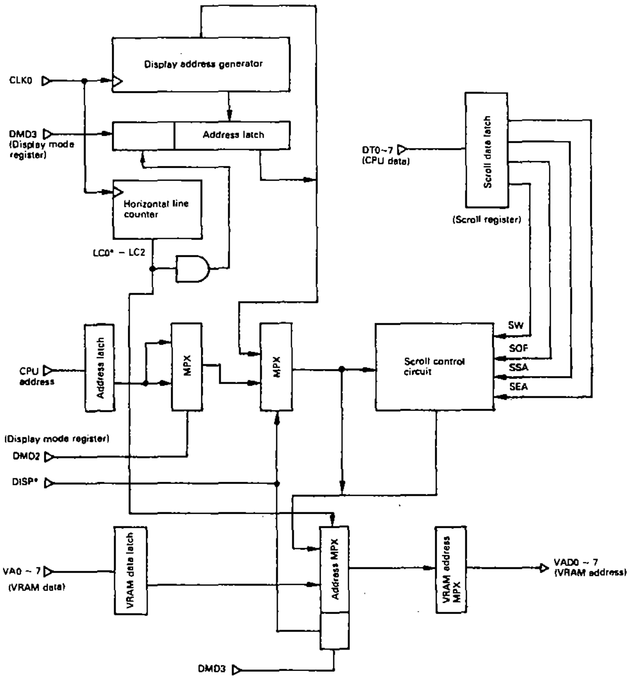
4-3 Clock generator and timing generator

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

Since the low state of signal is used for NTPL (NTSC/PAL selection) with the MZ-800, the CPU clock of 3.547 MHz is derived from the crystal frequency of 17.734 MHz by dividing it 1/5.


1/4 divider
CKMS (1.10 MHz)
CLK
1/4 divider
To CPU (CPU clock, 3.547 MHz)
1/5 divider
NTPL
Waveform generator (horizontal sync)
Divider
Divider
OFF
HBLN
HSYN
(1/142)
Divider
CK32 (1/2)
DQ1 (1/4)
DQ2 (1/8)
DQ3 (1/16)
Pulse generator
LD64 (640 × 200, shift register load)
CK32
DQ1
DQ2
Pulse generator
LOAD
(700 mode, 320 × 200, shift register load)
Waveform generator (display address control)
Display mode register
DMD2
Divider
Divider
Waveform generator (vertical sync)
(1/312)
VBLN
VSYN
Waveform generator (display address control)
To display address generator

Clock generator and timing generator circuits



Display address generator block diagram

4-4 Display address generator

1) Display address generation

- Display address increments from left to right as beginning from the home position at the upper left corner of the CRT screen (address $000). The first display line dominates address $000 through $027. Because a screen frame consists of 200 rasters, the address at the right side of the bottom corner is as follows:

  (200 × 40) − 1 = 7999 = $1F3F
- The address counter stops counting for a horizontal flyback line and stored in the address latch circuit. When the horizontal flyback line terminates, the address latch output is preset in the address counter (display address generator).
- Address is generated even while the vertical flyback line is active and it makes the counter reset before termination of the vertical flyback line.

2) Display address generation in the MZ-700 mode

- Because characters are displayed under the PCG method in the MZ-700 mode, address is generated for each character and the same address is used for displaying of one character. The 3-bit horizontal line counter is provided to count horizontal lines to generate the address (LC0 ~ LC2) for selection of the character front.

  Display address increments from left to right having the uppermost left corner of the screen for the home position.

  Since 25 lines are used to develop displaying of characters composed of 8 × 8 dots, the address at the right of the bottom lines becomes $3EF.

3) Display address multiplexed with CPU address

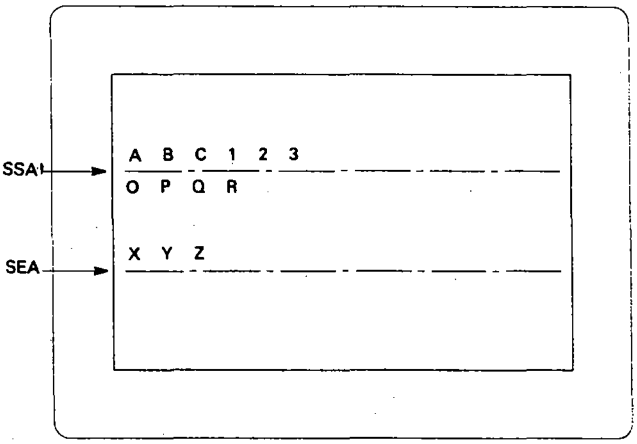
- Address used to write data to the VRAM is latched in order to avoid CPU wait. Display modes of 640 dots and 320 dots are assigned by the mode switch (DMD2).
- Display address is multiplexed with the VRAM write address in the timing of DISP which has the timing that the display address and CPU address may become a pseudo cycle steal.

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

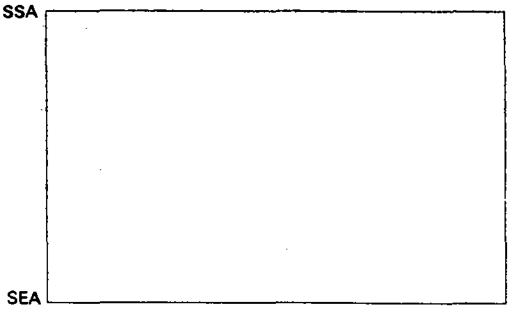
3) Way of smooth scrolling

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.



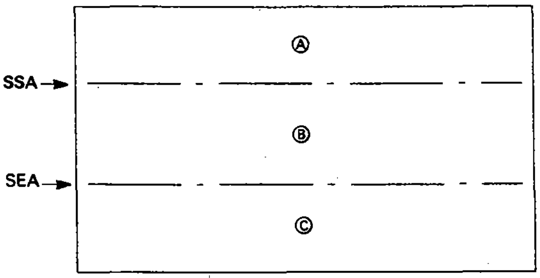
4) Line scroll

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 Ⓐ, Ⓑ and Ⓒ.
Though the section Ⓑ is permitted to scroll, sections Ⓐ and Ⓒ are not permitted to scroll.
See the figure to explain with.



Assume now that the top of the section Ⓑ is on the 5th line (40 raster) and the top of the section Ⓒ 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

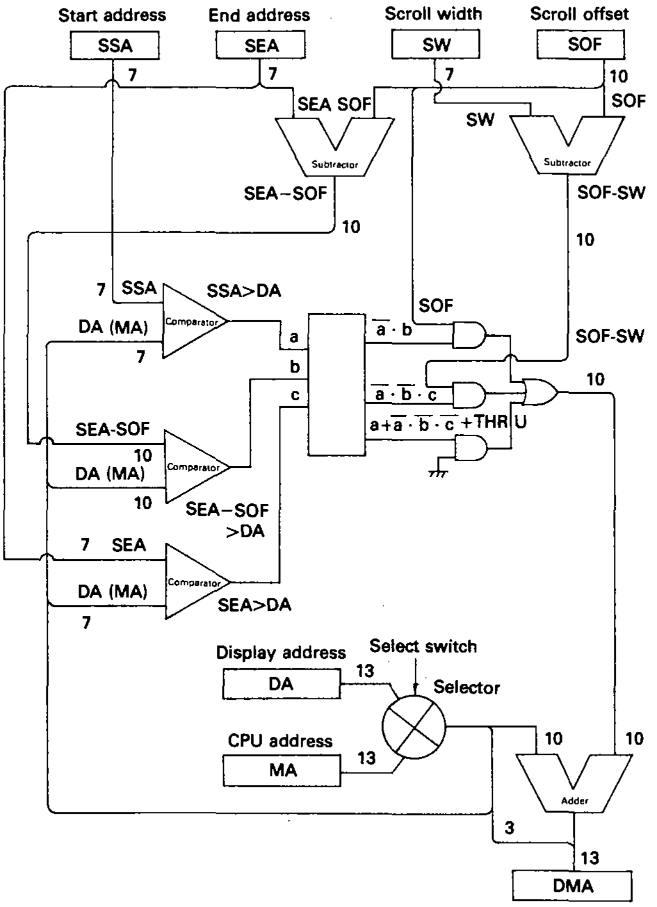
In this occasion, it needs to initialize the screen that has been displayed. "SOF = $5" must be programmed to scroll Ⓑ one line. Then, only the section Ⓑ is shifted up, and the highest line of Ⓑ moves to the bottom line of Ⓑ. Programming "SOF = $A" makes it scrolled one more line.

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

MSB						LSB
╲	6	5	4	3	2	1	SSA 0

SEA: Scroll end address
Increment of SEA: $5
Minimum value of SEA: $5
Maximum value of SEA: $7D

MSB						LSB
╲	6	5	4	3	2	1	SEA 0

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

MSB						LSB
╲	6	5	4	3	2	1	SW 0

SOF: Scroll offset
Increment of SOF: $5
Minimum value of SOF: $0 (without offset)
Maximum value of SOF: $3E8

	MSB							LSB
SOF 1	7	6	5	4	3	2	1	SOF 0

	MSB		LSB
SOF 2	╱	9	SOF 8

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 6	SSA 5	SSA 4	SSA 3	SSA 2	SSA 1	SSA 0	–	–	–	–	–	–	
SEA	SEA 6	SEA 5	SEA 4	SEA 3	SEA 2	SEA 1	SEA 0	–	–	–	–	–	–	
SOF	SOF 9	SOF 8	SOF 7	SOF 6	SOF 5	SOF 4	SOF 3	SOF 2	SOF 1	SOF 0				
Screen left end address														
0 Line	0	0	0	0	0	0	0	0	0	0	0	0	0	First line
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	Second line
16 Line	0	0	0	1	0	1	0	0	0	0	0	0	0	Second line
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	Twenty fifth line
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.

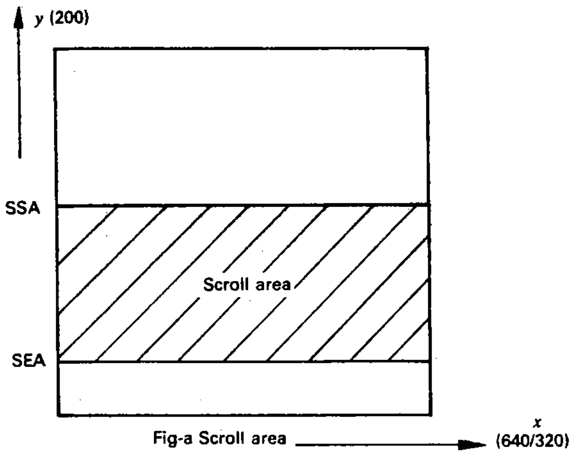


Fig-a Scroll area

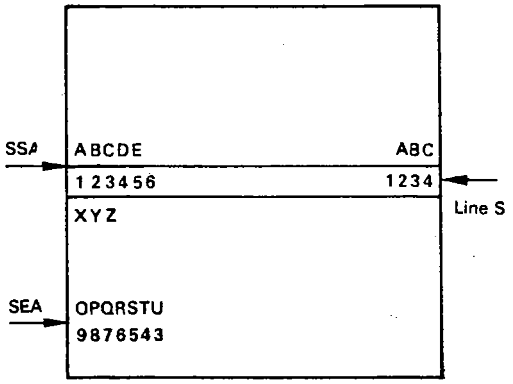


Fig-b Screen before scroll

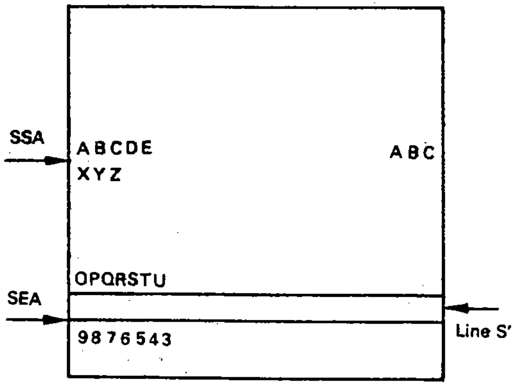
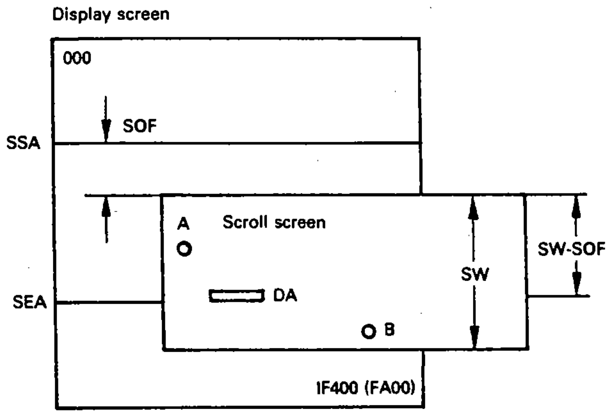


Fig-c Line after scroll

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-line scroll: $SOF_3 = 0F \times 3$
  5-line scroll: $SOF_5 = 0F \times 5$
  And, to scroll one more line after 5-line scroll;
  5-line scroll: $SOF_5' = SOF5 + 0F = OF \times 6$



- 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.

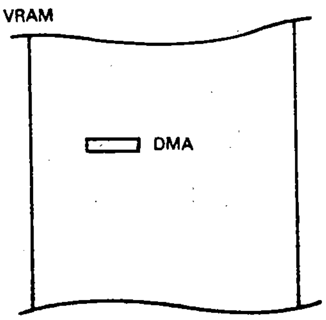


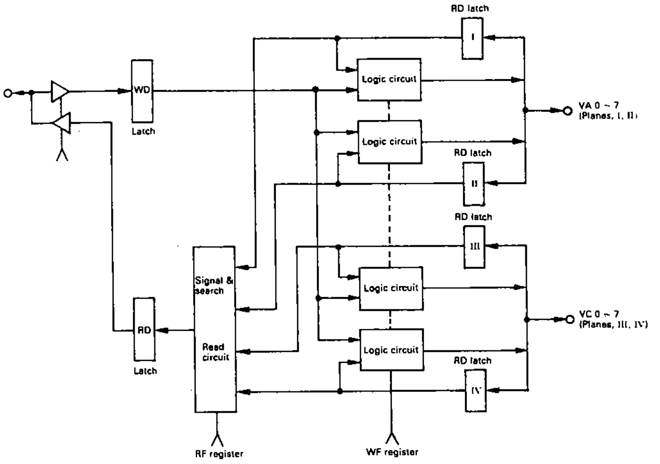
Fig-d Address conversion

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.

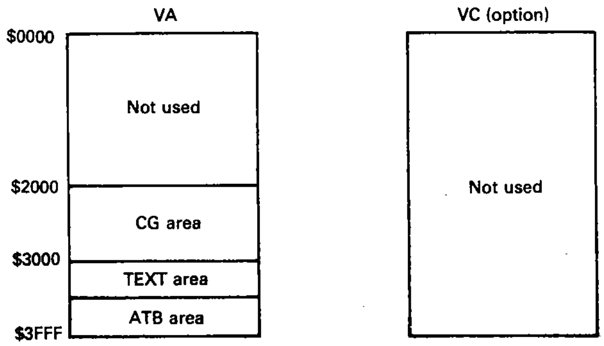
- Read
  For plane read data from the VRAM, data to be read by the CPU are arranged in accordance with the direction of the read format register (RF).



* 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.



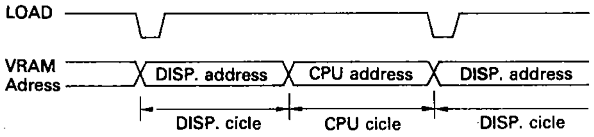
As the PCG method is adopted for the MZ-700 mode, the text and ATB areas are actually mapped to $D000 ~ $DFFF. So, the VRAM address has the following relation with the display character position.

	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.

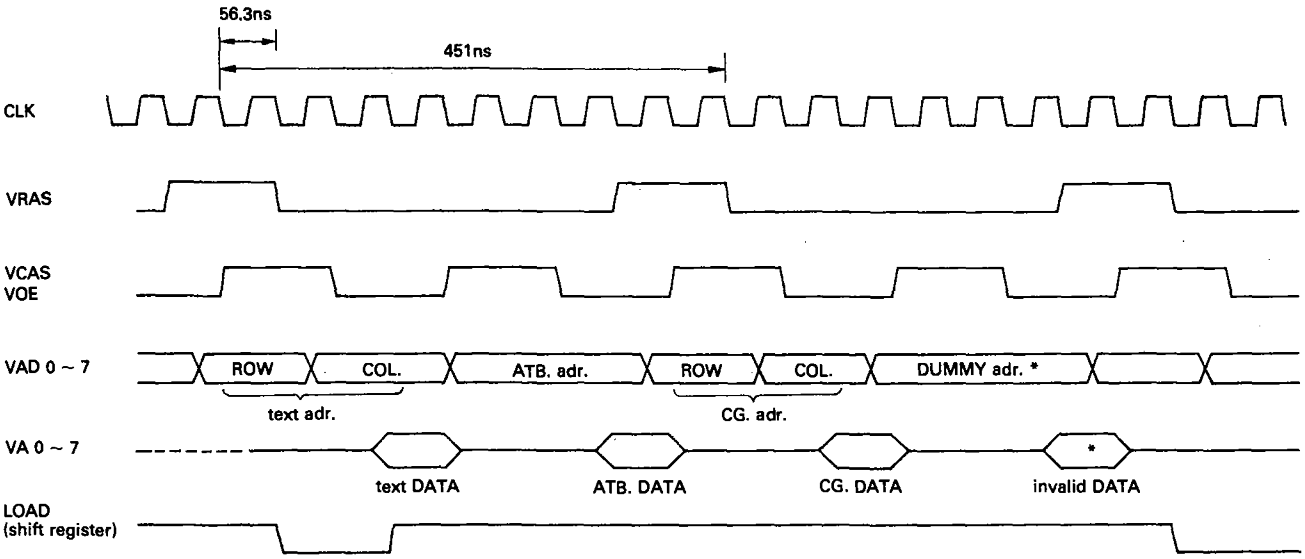
What is pseudo cycle steal

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

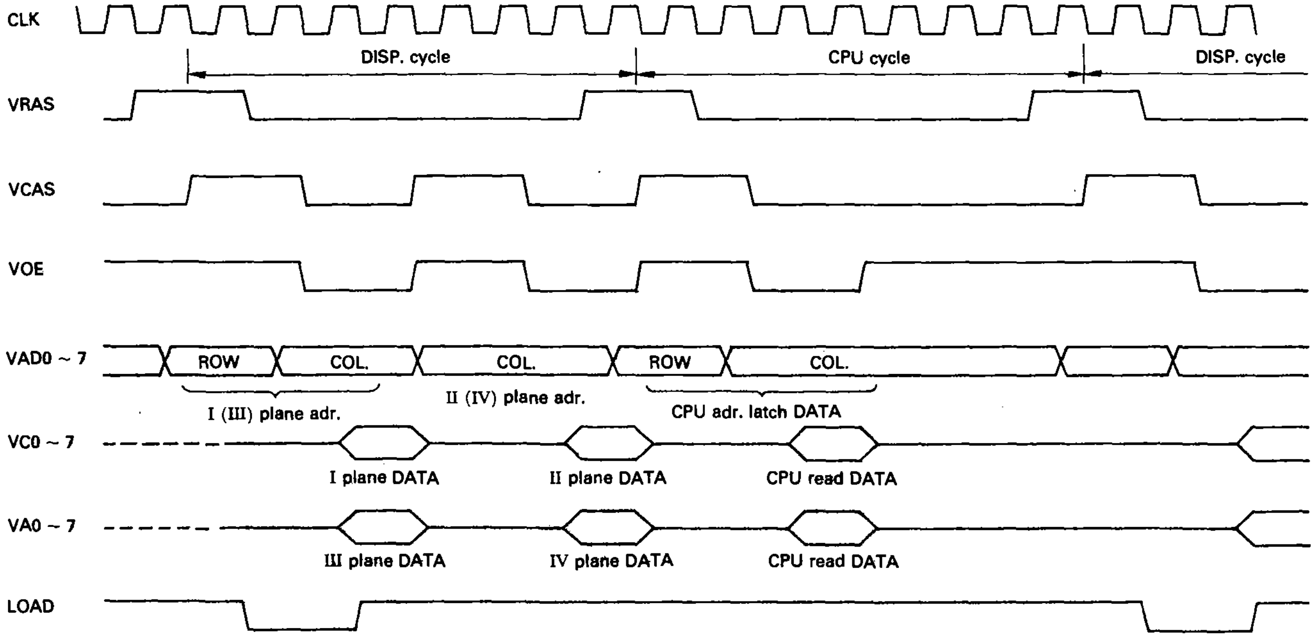


As shown in the figure, a next display data fetch and CPU accessing are multiplexed during a display period. Because accessing of the VRAM while characters are on display causes the screen to blink with the MZ-700 mode, it awaits for blinking to complete before accessing of the VRAM. But, with the cycle 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

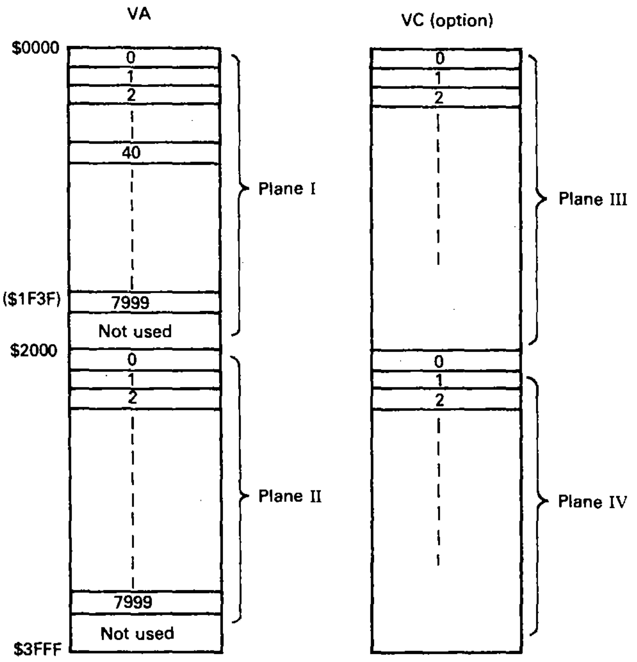


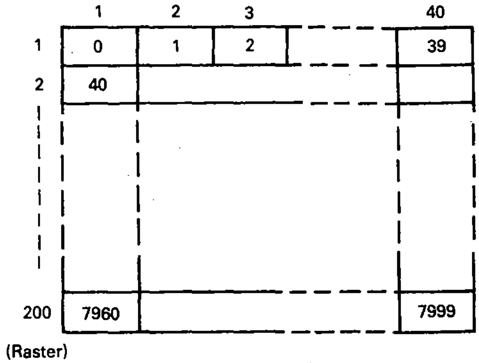
MZ-800 MODE (320 × 200 dot)



1) 320 × 200 dots

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



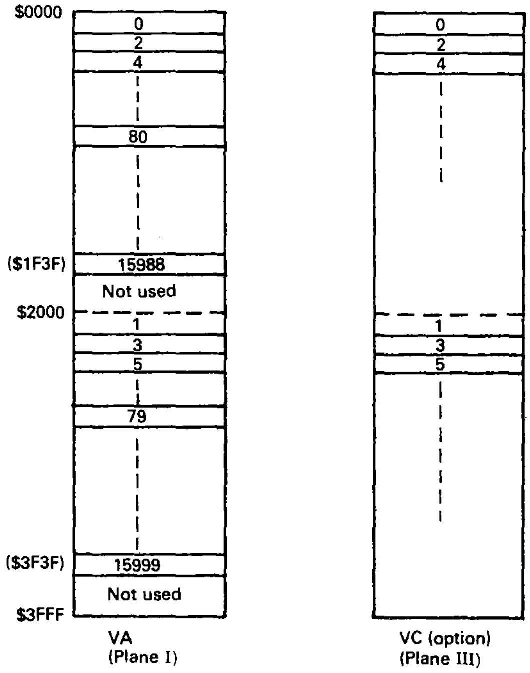


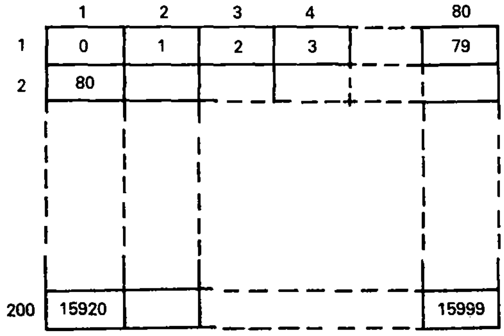
CRT display position

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.)

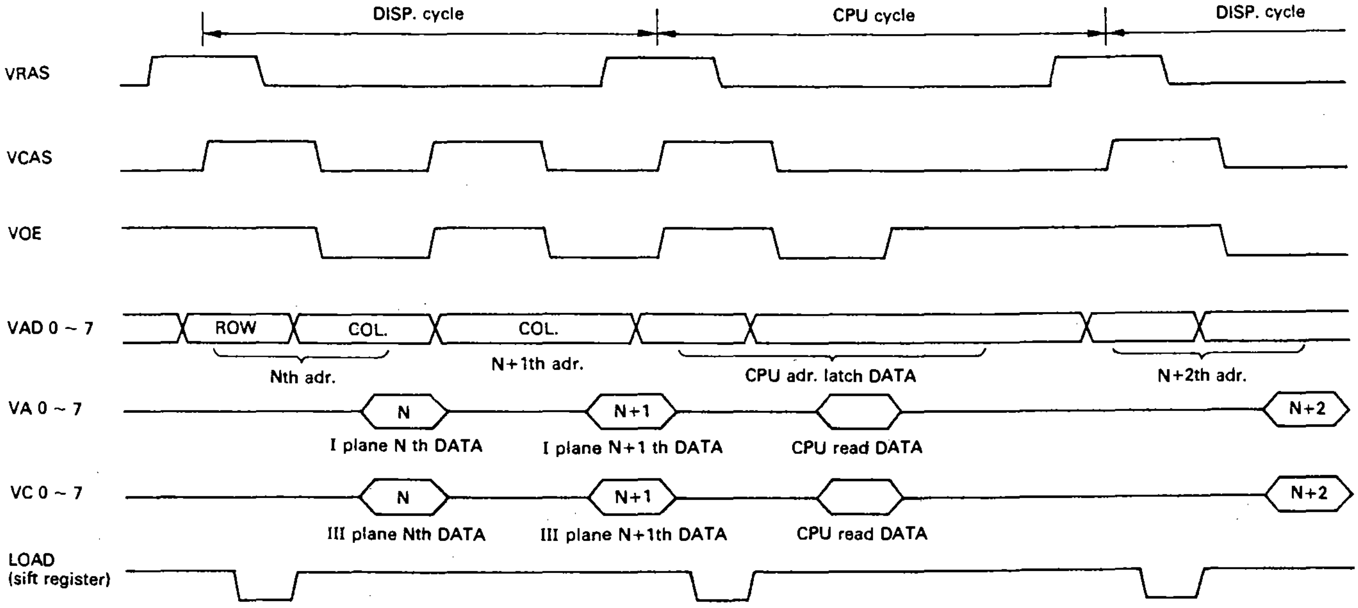
See the´figure below for VRAM configuration and CRT character display position.





CRT display position

800 MODE (640 × 200 dot)

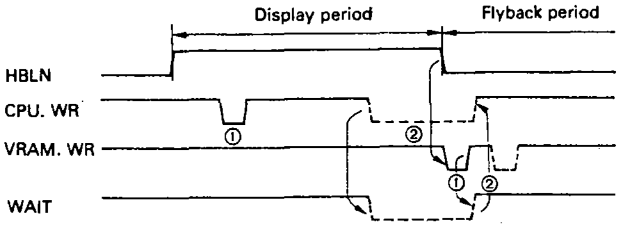


CPU and VRAM accessing

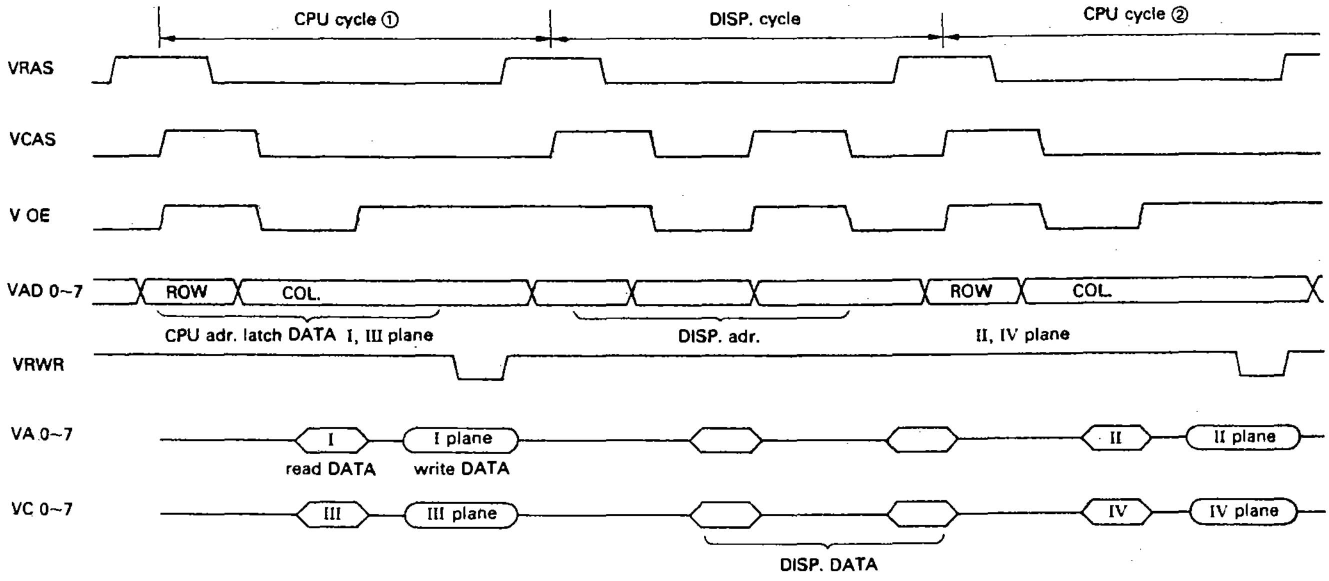
1. Accessing of the VRAM by the CPU is carried out in the cycle steal mode (MZ-800 mode only) during the flyback period of the display under the control of the CRT controller.
2. Even when there is no accessing from the CPU in the CPU cycle, such as VRAS, VCAS, VOE, etc. are outputted in the timing of 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)

MSB							LSB
╲	╲	╲	╲	DMD 3	DMD 2	DMD 1	DMD 0

- 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

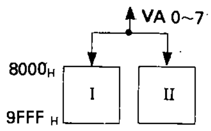
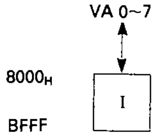
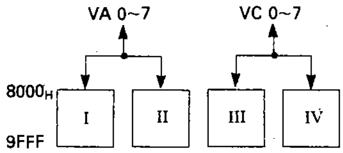
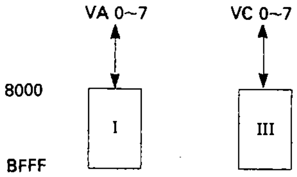
- DMD 1, 0: Display screen designation

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. △ 3/3

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

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	0
		640 × 200	1 color	Frame A	I	0	1	0	0	4
32 KB		320 × 200	4 colors	Frame A	I, II	0	0	0	0	0
				Frame B	III, IV	0	0	0	1	1
			16 colors	Frame A	I, II, III, IV	0	0	1	0	2
		640 × 200	1 color	Frame A	I	0	1	0	0	4
				Frame B	III	0	1	0	1	5
			4 colors	Frame A	I, III	0	1	1	0	6
	MZ-700	40 characters × 25 lines	8 colors	Frame A	RGB	1	0	0	0	8

(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/$\overline{\text{A}}$ 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)

MSB LSB

SRCH $\overline{\text{SING}}$			* B/$\overline{\text{A}}$	IV	III	II	I

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

- SRCH/$\overline{\text{SI NG}}$

  "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/$\overline{\text{A}}$

  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/$\overline{\mathrm{SING}}$	B/$\overline{\mathrm{A}}$	IV	III	II	I	Function (NOTE)
Single color data read	320 × 200, 4/16 colors 640 × 200, 1/4 colors	0	(*) Frame A: "0" Frame B: "1"	0	0	0	1	Plane I data read
				0	0	1	0	Plane II data read
				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	$\overline{\mathrm{I}}$, $\overline{\mathrm{II}}$ dot search
				×	×	0	1	I, $\overline{\mathrm{II}}$ dot search
				×	×	1	0	$\overline{\mathrm{I}}$, II dot search
				×	×	1	1	I, II dot search
			1	0	0	×	×	$\overline{\mathrm{III}}$, $\overline{\mathrm{IV}}$ dot search
				0	1	×	×	III, $\overline{\mathrm{IV}}$ dot search
				1	0	×	×	$\overline{\mathrm{III}}$, IV dot search
				1	1	×	×	III, IV dot search
	320 × 200, 16 colors	1	×	0	0	0	0	$\overline{\mathrm{I}}$, $\overline{\mathrm{II}}$, $\overline{\mathrm{III}}$, $\overline{\mathrm{IV}}$(, dot search
				0	0	0	1	I, $\overline{\mathrm{II}}$, $\overline{\mathrm{III}}$, $\overline{\mathrm{IV}}$(, dot search
				0	0	1	0	$\overline{\mathrm{I}}$, II, $\overline{\mathrm{III}}$, $\overline{\mathrm{IV}}$, dot search
				0	0	1	1	I, II, $\overline{\mathrm{III}}$, $\overline{\mathrm{IV}}$, dot search
				0	1	0	0	$\overline{\mathrm{I}}$, $\overline{\mathrm{II}}$, III, $\overline{\mathrm{IV}}$, dot search
				⋮	⋮	⋮	⋮	⋮
				1	1	1	1	I, II, III, IV, dot search
	640 × 200, 1 color	1	0	×	×	×	0	$\overline{\mathrm{I}}$, dot search
				×	×	×	1	I, dot search
			1	×	0	×	×	$\overline{\mathrm{III}}$, dot search
				×	1	×	×	III, dot search
	640 × 200, 4 colors	1	×	×	0	×	0	$\overline{\mathrm{I}}$, $\overline{\mathrm{III}}$, dot search
				×	0	×	1	I, $\overline{\mathrm{III}}$, dot search
				×	1	×	0	$\overline{\mathrm{I}}$, 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/$\overline{\mathrm{A}}$ 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) $B/\overline{A}$	IV	III	II	I

NOTE: Same as the bit $B/\overline{A}$ of the read format register.

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

- WMD 0 ~ 2 .....
  Selects the logical operational mode for read-modify-write.

- $B/\overline{A}$ (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 2	1	0	$B/\overline{A}$	Color plane IV	III	II	I	Display mode	Function [WD: Write data VD : VRAM data]
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 ⊕ 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": $\overline{WD}\cdot VD$ Color plane of "0": Fixed
REPLACE	1	0	×	0	×	×	0/1	0/1	320 × 200 Ⓐ, 4 colors Ⓑ	Writes WD in a specific color (Character write to the graphic plane)
				1	0/1	0/1	×	×		
				×	0/1	0/1	0/1	0/1	320 × 200, 16 colors	Color plane of "1": WD
				0	×	×	×	0/1	640 × 200 Ⓐ, 1 color Ⓑ	Color plane of "0": Writes "0". Color plane of "X": Fixed
				1	×	0/1	×	×		
				×	×	0/1	×	0/1	640 × 200, 4 colors	
PSET	1	1	×	0	×	×	0/1	0/1	320 × 200 Ⓐ, 4 colors Ⓑ	Writes only bit "1" of WD in a specific color. (Character write to graphic plane)
				1	0/1	0/1	×	×		
				×	0/1	0/1	0/1	0/1	320 × 200, 16 colors	Color plane of "1": WD + VD
				0	×	×	×	0/1	640 × 200 Ⓐ, 1 color, Ⓑ	Color plane of "0": $\overline{WD}\cdot VD$ Color plane of "X": Fixed
				1	×	0/1	×	×		
				×	×	0/1	×	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.
- $B/\overline{A}$ 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.

① REPLACE write

- To develop light yellow characters on the graphic screen.

Black	Black	Yellow	Yellow	Magenta	Magenta	Cyan	Cyan

Display before write

VRAM

Plane I (B) data

0	0	0	0	1	1	1	1

Plane II (R) data

0	0	1	1	0	0	1	1

Plane III (G) data

0	0	1	1	1	1	0	0

Plane IV (I) data

0	0	0	0	0	0	0	0

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/Ā	Color designation				
WF register	1	0	0	0	1	1	1	0	: Light yellow replace mode

Write data (CG pattern)	0	1	0	1	0	1	0	1

Black	Light yellow	Black	Light yellow	Black	Light yellow	Black	Light yellow

Display after write

VRAM

Plane I

0	0	0	0	0	0	0	0

Plane II

0	1	0	1	0	1	0	1

Plane III

0	1	0	1	0	1	0	1

Plane IV

0	1	0	1	0	1	0	1

② PSET write

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

	Mode			B/Ā	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

Black	Light yellow	Yellow	Light yellow	Magenta	Light yellow	Cyan	Light yellow

Display after write

VRAM

I (B)

0	0	0	0	1	0	1	0

II (R)

0	1	1	1	0	1	1	1

III (G)

0	1	1	1	1	1	0	1

IV (I)

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/Ā	Color designation				
RF register	1		0	1	1	1	0	: Light yellow search

- When the above read data are read after setting the red PSET mode in the WR register.

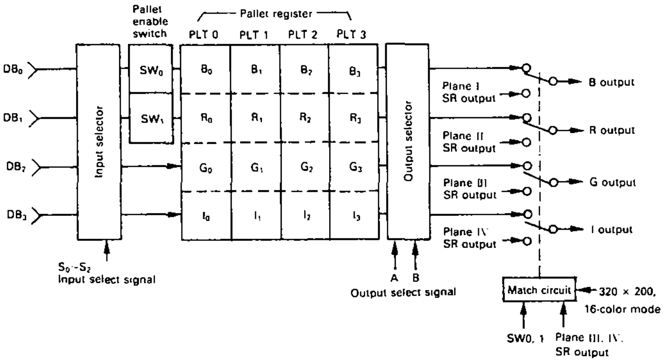
Black	Red	Yellow	Red	Magenta	Red	Cyan	Red

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 output and display mode>

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

<Pallet register write> ($F0_H$)

($F0_H$)

	MSB							LSB	
OUT $F0_H$	×	S_2	S_1	S_0	I_i	G_i	R_i / SW_1	B_i / SW_0	i = 0 ~ 3

1) $S_0 \sim S_2$: 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) SW_0, SW_1:

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) = SW_0, (Plane Ⅳ data) = SW_1

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.)

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 = (Plane Ⅲ data)	0	0	B_0	R_0	G_0	I_0
							1	0	B_1	R_1	G_1	I_1
						SW_1 = (Plane Ⅳ data)	0	1	B_2	R_2	G_2	I_2
							1	1	B_3	R_3	G_3	I_3
							×	×	I	Ⅱ	Ⅲ	Ⅳ
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 SW_0 is set to "0" and SW_1 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 SW_0 is set to "0" and SW_1 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 SW_0 and SW_1, 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	×	×	×	×	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

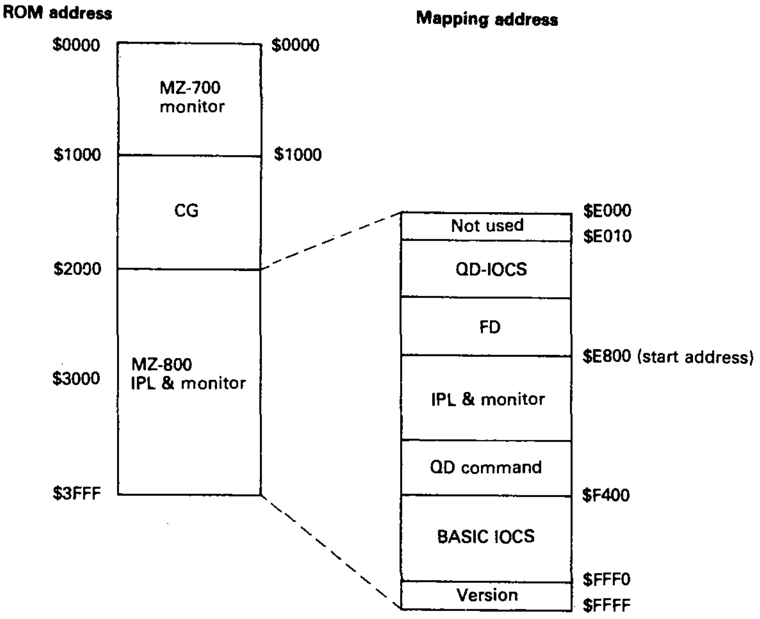
Control I/O address map

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.

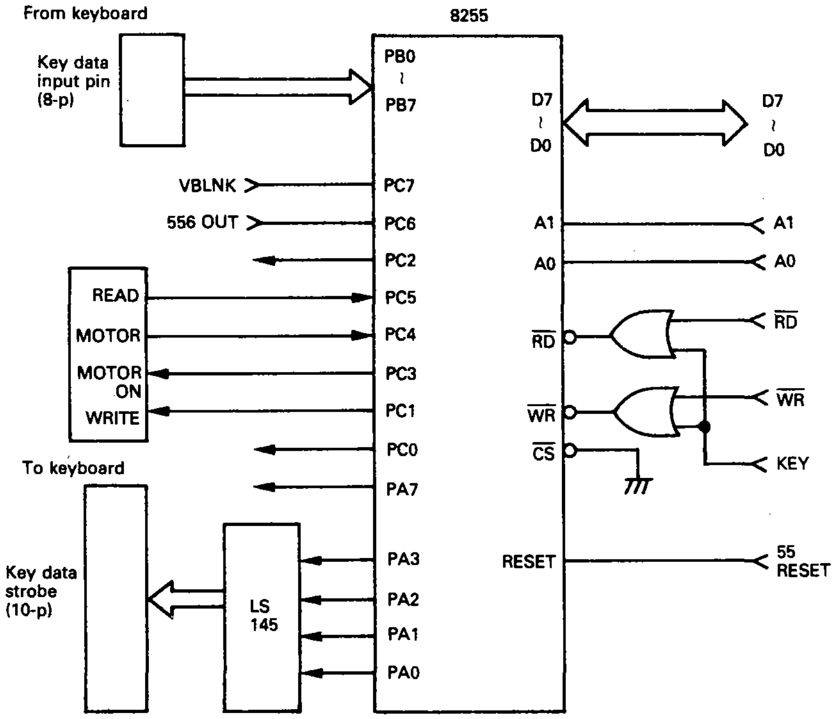


5 8255 Programmable Peripheral Interface

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 depending 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.

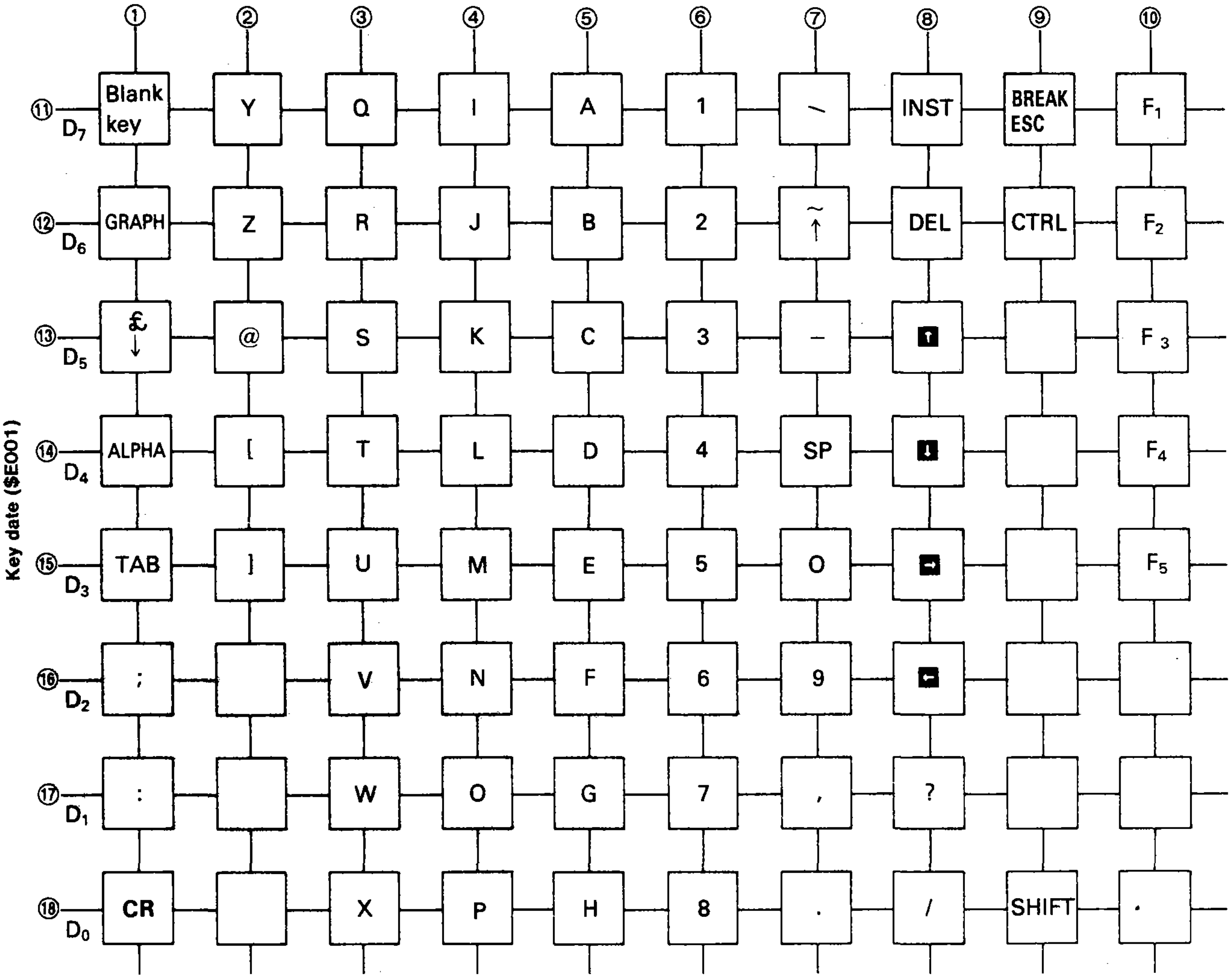
Port name (address)	Pin No.	I/O	Active state	Function
PA (700 $E000 800 $D0)	PA_0	O	H	Keyboard scan strobe
	PA_1		H	
	PA_2		H	
	PA_3		H	
	PA_4		L	Joystick-1 strobe
	PA_5		L	Joystick-2 strobe
	PA_7		L	CRT cursor blink timer reset
PB (700 $E001 800 $D1)	PB_0	I	L	Keyboard scan input
	PB_1			
	PB_2			
	PB_3			
	PB_4			
	PB_5			
	PB_6			
	PB_7		L	
PC (NOTE-1) (700 $E002 800 $D2)	PC_0	O	L	Prohibits sound output of the 8253
	PC_1	O	—	Cassette write data
	PC_2	O	L	Disables timer interrupt
	PC_3	O	_↑‾|_ (NOTE-2)	Rotates the cassette motor
	PC_4	I	H	Checks the cassette motor
	PC_5	I	—	Cassette read data
	PC_6	I	—	CRT cursor blink timer input
	PC_7	I	—	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 PA_0~PA_3 of the 8255 are connected via the LS145 decoder, and PB_0~PB_7 are connected to the key matrix directly.

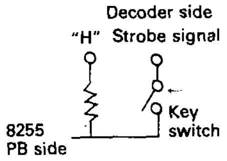


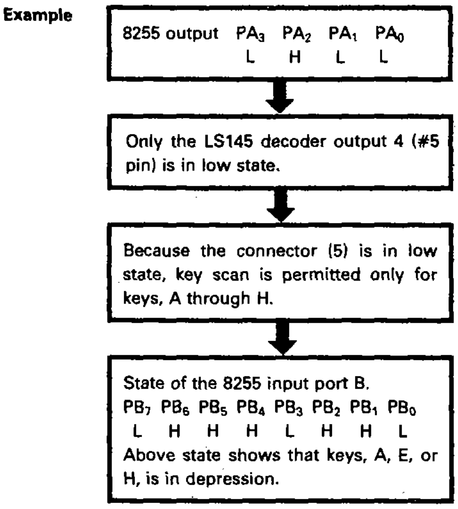
Key date ($E001)

	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩
⑪ D_7	Blank key	Y	Q	I	A	1	\	INST	BREAK ESC	F_1
⑫ D_6	GRAPH	Z	R	J	B	2	~ ↑	DEL	CTRL	F_2
⑬ D_5	£ ↓	@	S	K	C	3	–	↑		F_3
⑭ D_4	ALPHA	[	T	L	D	4	SP	↓		F_4
⑮ D_3	TAB	]	U	M	E	5	O	→		F_5
⑯ D_2	;		V	N	F	6	9	←		
⑰ D_1	:		W	O	G	7	,	?		
⑱ D_0	CR		X	P	H	8	.	/	SHIFT	·

Key strobe is issued through $PA_0 \sim PA_3$ 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, $PA_0 \sim PA_3$ 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.




Example

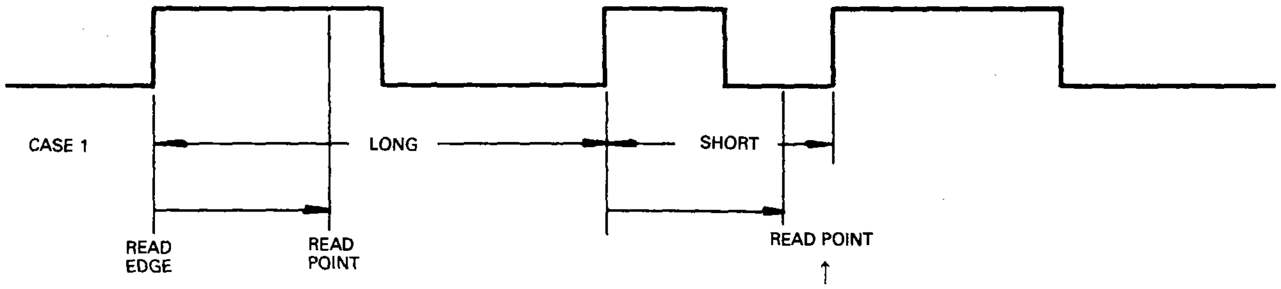
8255 output PA_3 PA_2 PA_1 PA_0
L H L L

Only the LS145 decoder output 4 (#5 pin) is in low state.

Because the connector (5) is in low state, key scan is permitted only for keys, A through H.

State of the 8255 input port B.
PB_7 PB_6 PB_5 PB_4 PB_3 PB_2 PB_1 PB_0
L H H H L H H L
Above state shows that keys, A, E, or H, is in depression.

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:



SHORT
(HIGH) 240μS
(LOW) 278μS
LONG
(HIGH) 470μS
(LOW) 494μS
READ POINT
379μS

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.

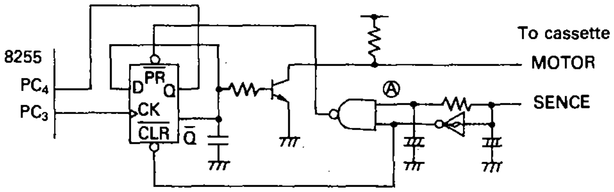
SHORT 10 seconds	TAPE MARK	1	1-1. INFORMATION BLOCK 128 bytes	Check sum, 2 bytes	1	SHORT 256 bytes	INFORMATION BLOCK 128 bytes	Check sum, 2 bytes	1	SHORT 5 seconds	
220C0	LONG 40 SHORT 40	LONG			LONG				LONG	11000	

TAPE MARK	1	DATA BLOCK	Check sum, 2 bytes	1	SHORT 256 bytes	DATA BLOCK	Check sum, 2 bytes	1	
LONG 20 SHORT 20				LONG				LONG	

See next for the contents of the information block.

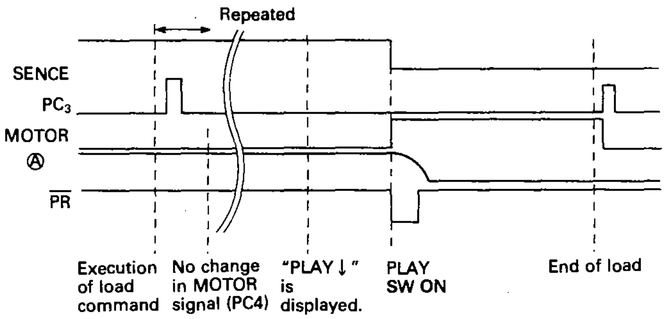
Name	Byte count	Function	Note
ATRB	1	Attribute	
NAME	17	File name (16 characters maximum)	CR (0D) affixed
SIZE	2	File byte size	In order of low to high order
DTADR	2	Loading address	
EXADR	2	Execution address	
COMNT	104	Comment	Not used

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

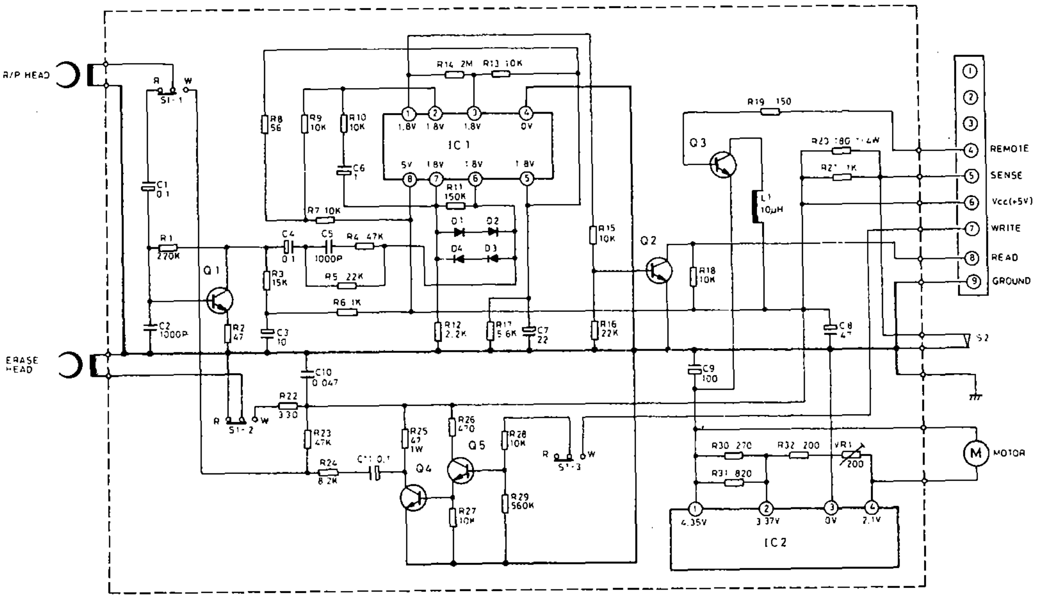


If switch has not been ON on the cassette recorder side, SENCE signal is in high state. When a switch (REW, FF, etc.) is pushed, it makes the signal turned low. It presets the D-FF and the motor starts to rotate with MOTOR in high state. With lock given to the D-FF through PC3, it permits on/off control 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 — — — — μPC358C, AN6562,
TDB0158DP8 or LM358DP
IC2 — — — — μPC1470H

Q1,2,5 — — — — 2SC1815-GR, 2SC945-PorQ
Q3,4 — — — — 2SC2120-Y or 2SD467-C

D1~4 — — — — 1S1555, 1S1588, DS442 or 1N4148

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

VOLTAGE CHECK POSITION
__V ---PLAYBACK POSITION

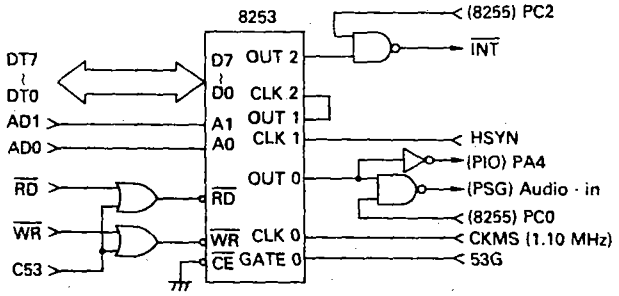
NOTES:
1 ALL RESISTANCE VALUES ARE IN Ω K 1000Ω M 1000KΩ
2 ALL CAPACITANCE VALUES ARE IN μF P 10^{-6}μF
3 ALL RESISTORS ARE 1/6 WATT UNLESS OTHERWISE SPECIFIED
4 VOLTAGES MEASURED FROM POINT INDICATOR TO CHASSIS GROUND WITH V.T.V.M AT LINE VOLUME CONTROL MINIMUM AND NO SIGNAL

(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.

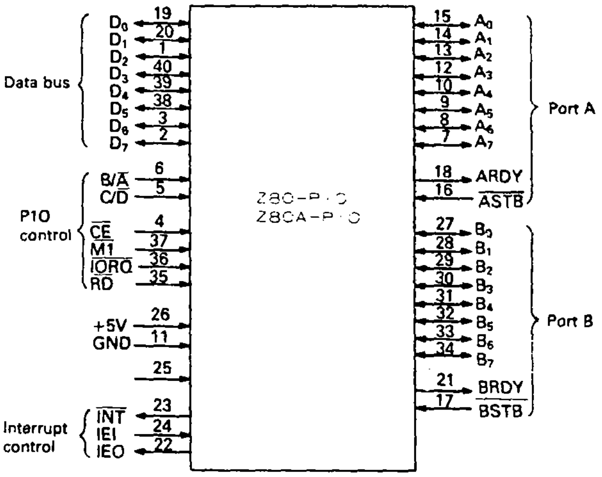
- Counter mode
  #0 ..... Square waveform generator MODE3
  #1 ..... Rate generator MODE2
  #2 ..... Interrupt on terminal counter



- 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 configuration

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 Date 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/$\overline{A}$. 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.
ø	25	I	System Clock	System clock CPU clock ø 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.

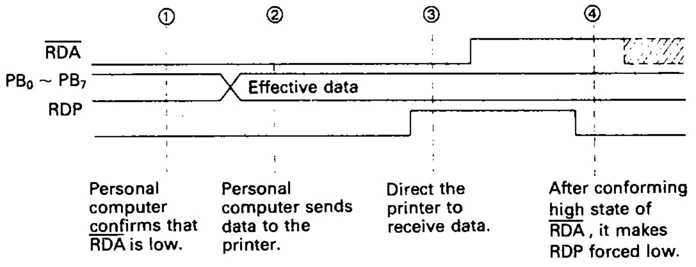
Pin name	Pin No.	I/O	Signal name	Description
$\overline{\text{INT}}$	23	O	Interrupt Request	Connection with CPU $\overline{\text{INT}}$ signal. A low on this line causes the PIO to place an interrupt request to the CPU. Because it is of an open drain type, it is possible to make $\overline{\text{INT}}$ of several peripheral LSI wired OR using the pullup resistance.
A_0~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. A0 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 riding 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 A0~A7 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 A0~A7 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 A0~A7 in this mode, unless $\overline{\text{ASTB}}$ turns low. 4) Bit mode : Not used.
B_0~B_7	27~34	I/O	Port B Bus	Port B data bus. Function of this bus is identical to A0~A7. But, it permits to drive a Darlington transistor as the bus can supply 1.5 V, 1.5 mA. B0 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	$\overline{\text{RDA}}$	A low on this line indicates that the printer data is ready to receive.
	PA$_1$	IN	L	$\overline{\text{STA}}$	A low on this line informs the personal a paper depletion during status check.
	PA$_2$	IN	—	GND	
	PA$_3$	IN	—	GND	
	PA$_4$	IN	H		An 8253 output used for interrupt.
	PA$_5$	IN	H		Horizontal blanking signal used for interrupt.
	PA$_6$	OUT	H	IRT	Used for printer initialization.
	PA$_7$	OUT	H	RDP	Indicates the printer to receive data.
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


①
②
③
④
$\overline{\text{RDA}}$
PB$_0$ ~ PB$_7$
Effective data
RDP
Personal computer confirms that $\overline{\text{RDA}}$ is low.
Personal computer sends data to the printer.
Direct the printer to receive data.
After conforming high state of $\overline{\text{RDA}}$, it makes RDP forced low.

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 ($\overline{\text{RDA}}$=H). After this, it awaits until $\overline{\text{RDA}}$ 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 $\overline{\text{RDA}}$, since RDA is inputted to the $\overline{\text{RSTB}}$ 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 $\overline{\text{RDA}}$ 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.

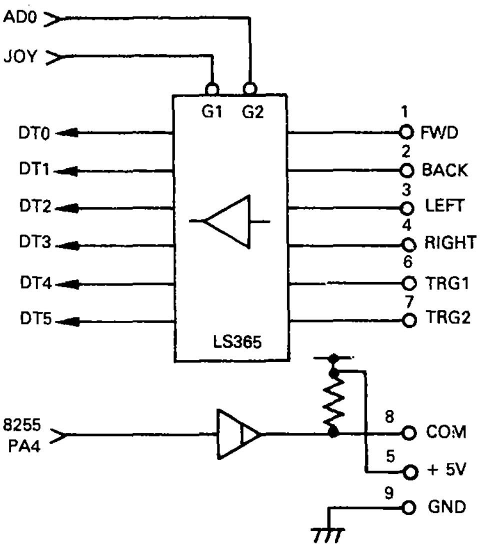
NOTE: When the MZ-800 dedicated printer is used, there may be such a case that proper operation is not attained due to different printing characters and control codes. It must be also noted that all MZ-800 characters can not be printed. Besides, connection with a Centronics compatible printer may not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

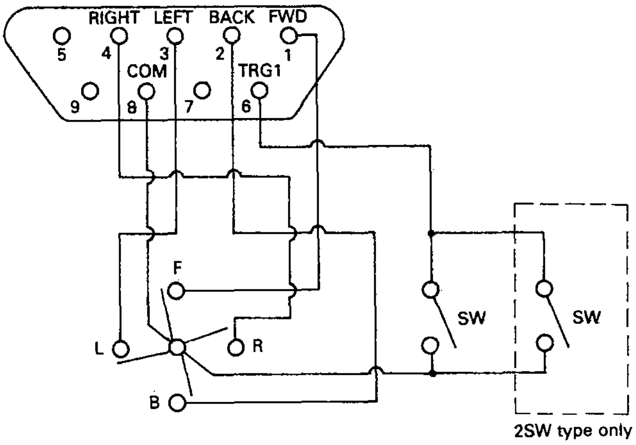
9 Joystick

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:

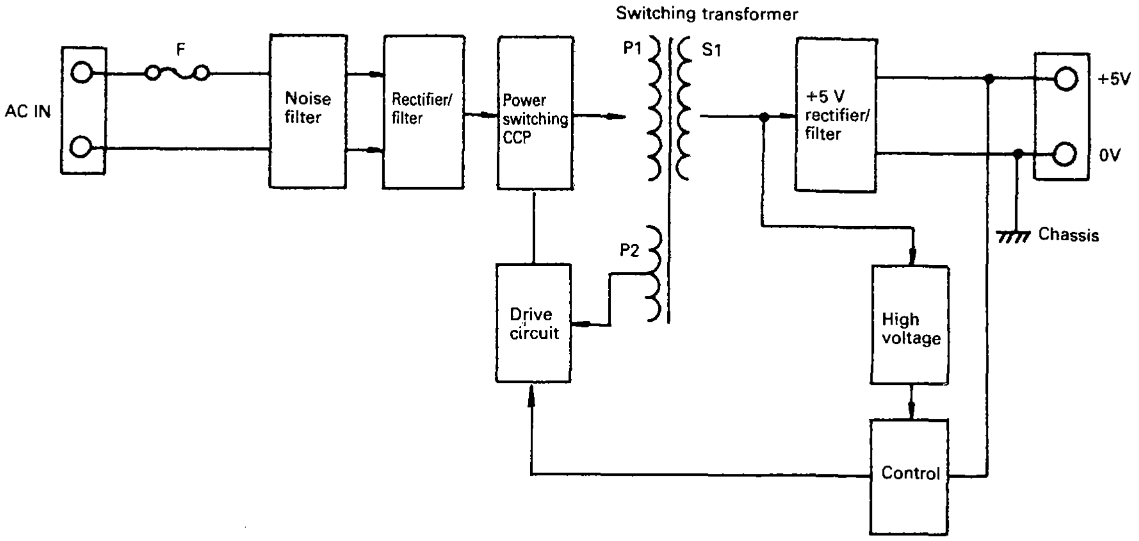
SW No.	Function	Setup method
1	MZ-700/MZ-800 selection	ON: MZ-700 mode OFF: MZ-800 mode
2 3	MZ/Centronics printer selection	MZ printer with SW2 and SW3 at ON Centronics printer with SW2 and SW3 at OFF
4	External cassette recorder polarity selection	Changed so as to enable read on the external cassette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

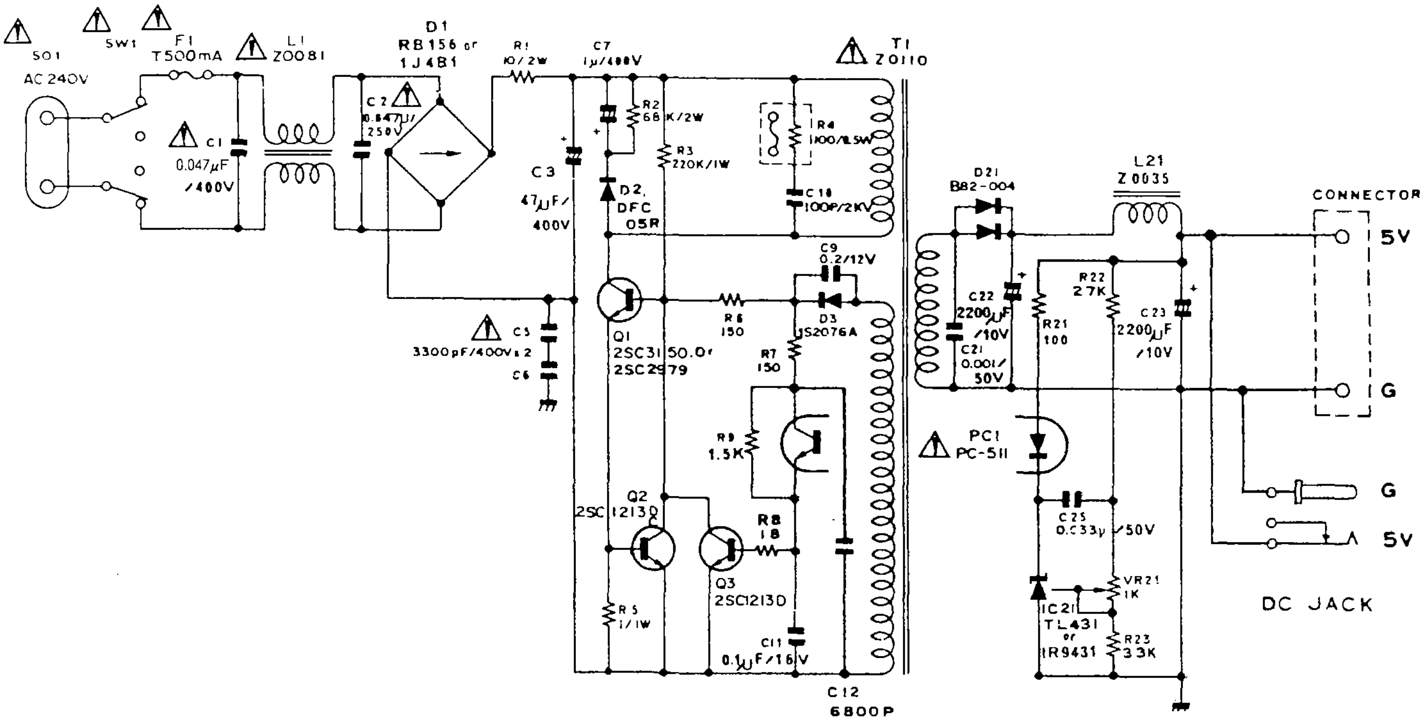
* Switch setups at the factory

SW1 ...... OFF (MZ-800 mode)
SW2 ...... ON } (MZ dedicated printer)
SW3 ...... ON }
SW4 ...... ON ——

10 Power supply

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/ON 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 cuppler PC1 active by the detecting signal of the control section for given period. By adjusting control current of the drive circuit, it makes the out put stabilized.

10-3 Maintenance

Cleaning

Dust deposit inside the power supply unit may becomes the cause for overheat as it prevent heat dispersion, which results in damage in component. 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.

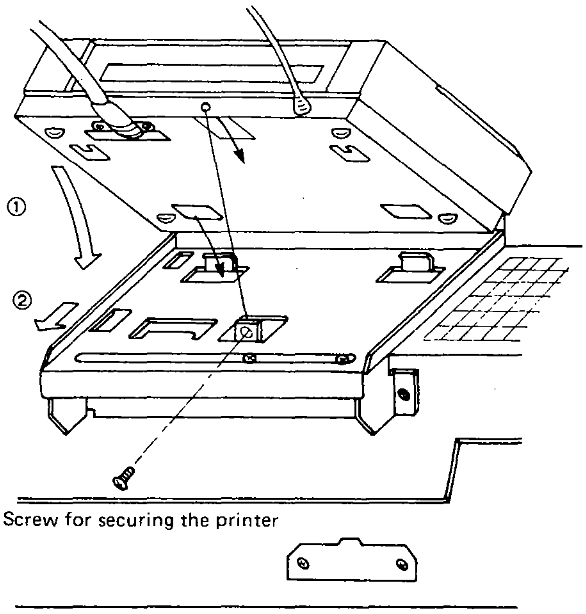
(2) If a defective item were found, it has to be replaced with the new one. But, care must also taken as there are possible defects in multiple number of components.

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

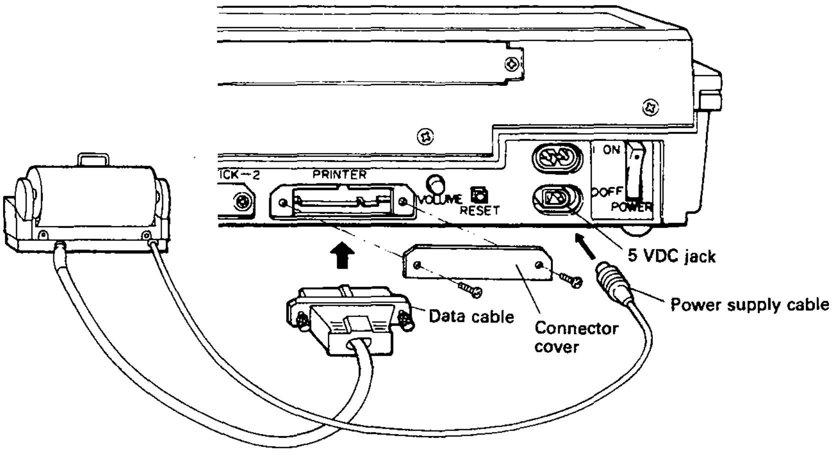
① Make sure that power is off to the MZ-800 and its peripheral units.

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.

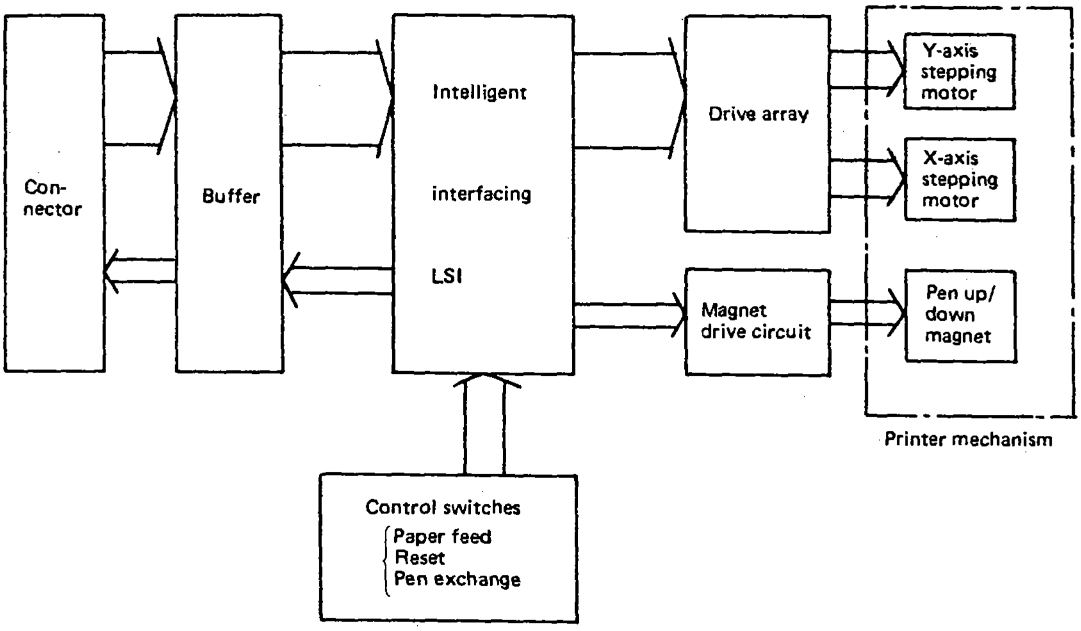
④ For connection of the printer connector, use the screws that had been at both ends of connector.

Rear side of the MZ-800



* 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.

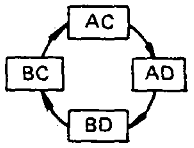
Since the pen rotor makes a undirectionai rotation at the left end of the X-axis, and is locked within prointable range, care must be exerted not to touch the rotor and the slider.

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.

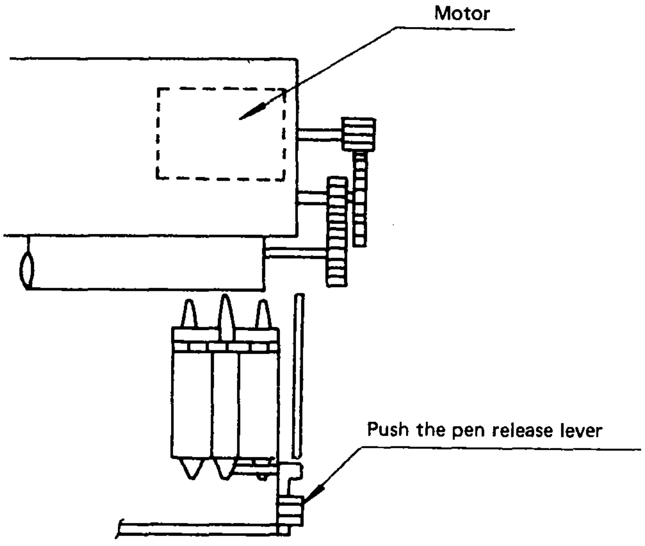
Motor phase and rotating direction

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.



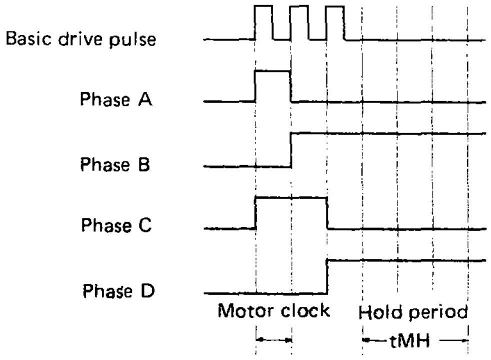
To install the pen, push the tip of the pen through the ring of the return spring in the first place, then push into the holder. Upon completion, ensure that the tip of the pen is engaged with the hole of the pen return spring.

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 (tMH = lms or more).

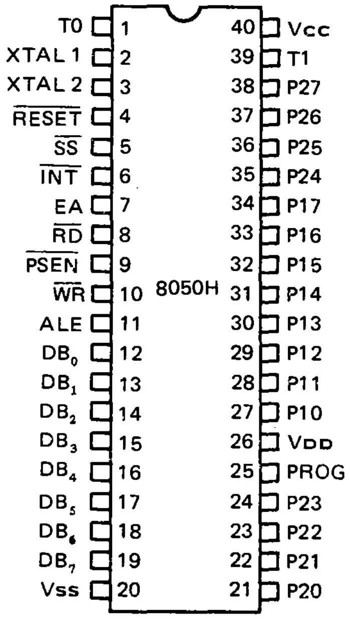
11-8 Colour position detector

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 → 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.

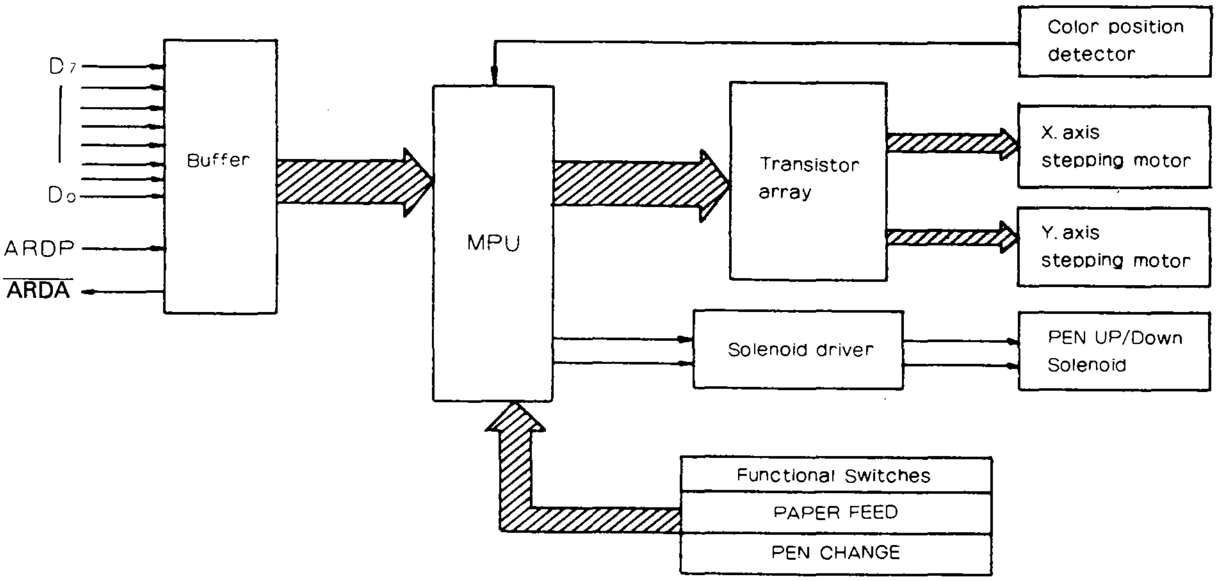
11-11 Interfacing with the MZ-800

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

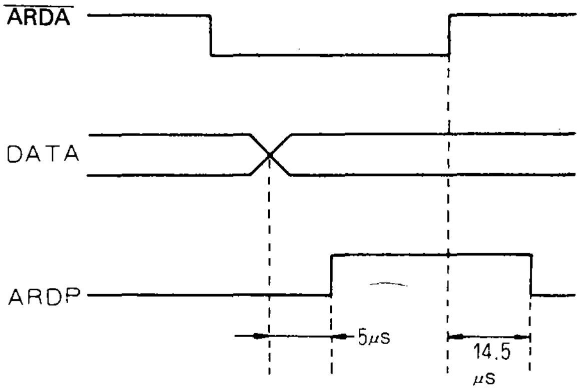
Table of character set

LSD \ MSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			SP	Ø	@	P			}		q	n				
1		⍗	!	1	A	Q					a					
2		⍐	"	2	B	R				e	Z	Ü				
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	Ö	¦				
9			)	9	I	Y					k	Ä				
A			*	:	J	Ƶ				b	f	ö				
B			+	;	K	[		°	^	x	v	ä				£
C			,	<	L	\				d						↓
D			—	=	M	]				r	ü	y				
E			.	>	N	↑				p	ß	{				
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

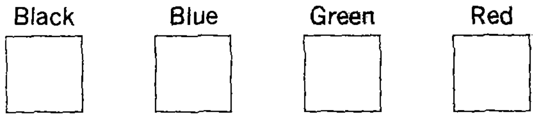
Control codes used in the text mode

- 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)	J△x, △y, .., △xn, △yn (−999≦△x, △y≦999)	Draws lines from the current pen location to relative coordinates (△x1, △y1), then to relative coordinates (△x2, △y2), and so forth.
MOVE (MOVE)	Mx, y (−999≦x, y≦999)	Lifts the pen and moves it to coordinates (x, y).
RELATIVE MOVE (RMOVE)	R△x,△y (−999≦△x,△y≦999)	Lifts the pen and moves it to coordinates(△x, △y)
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 separater 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

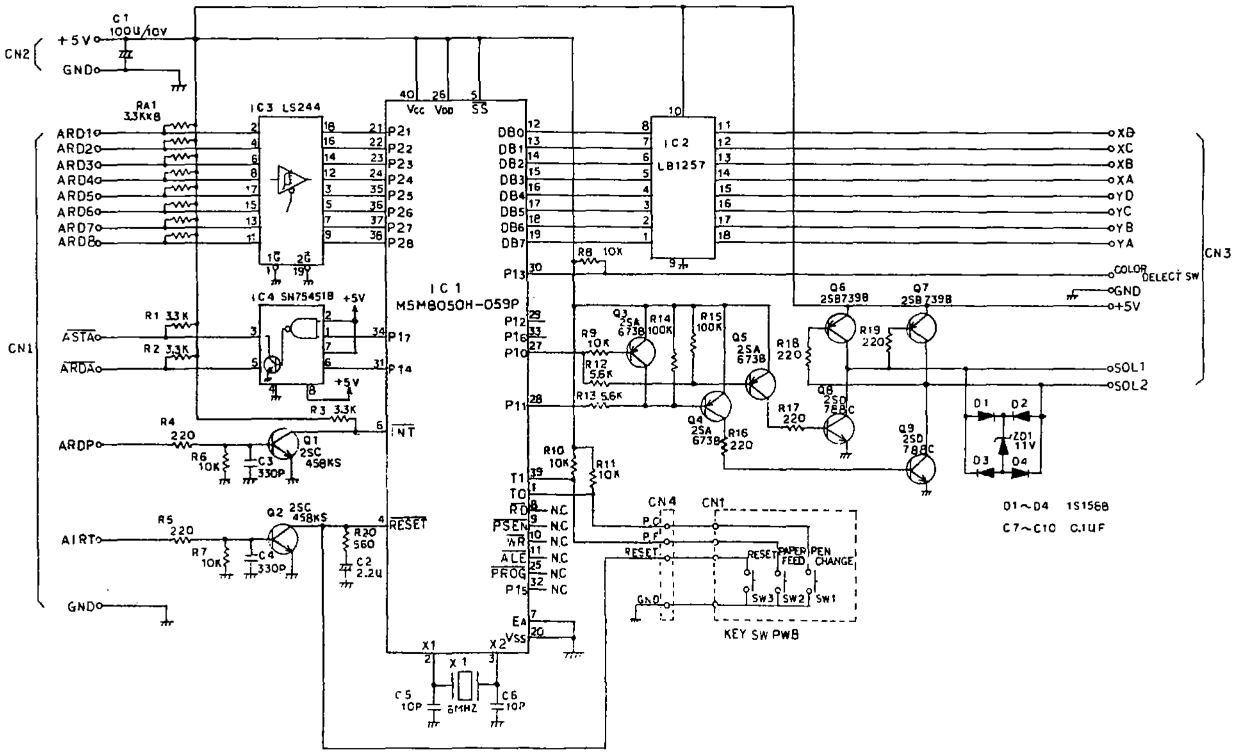
Outline

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.

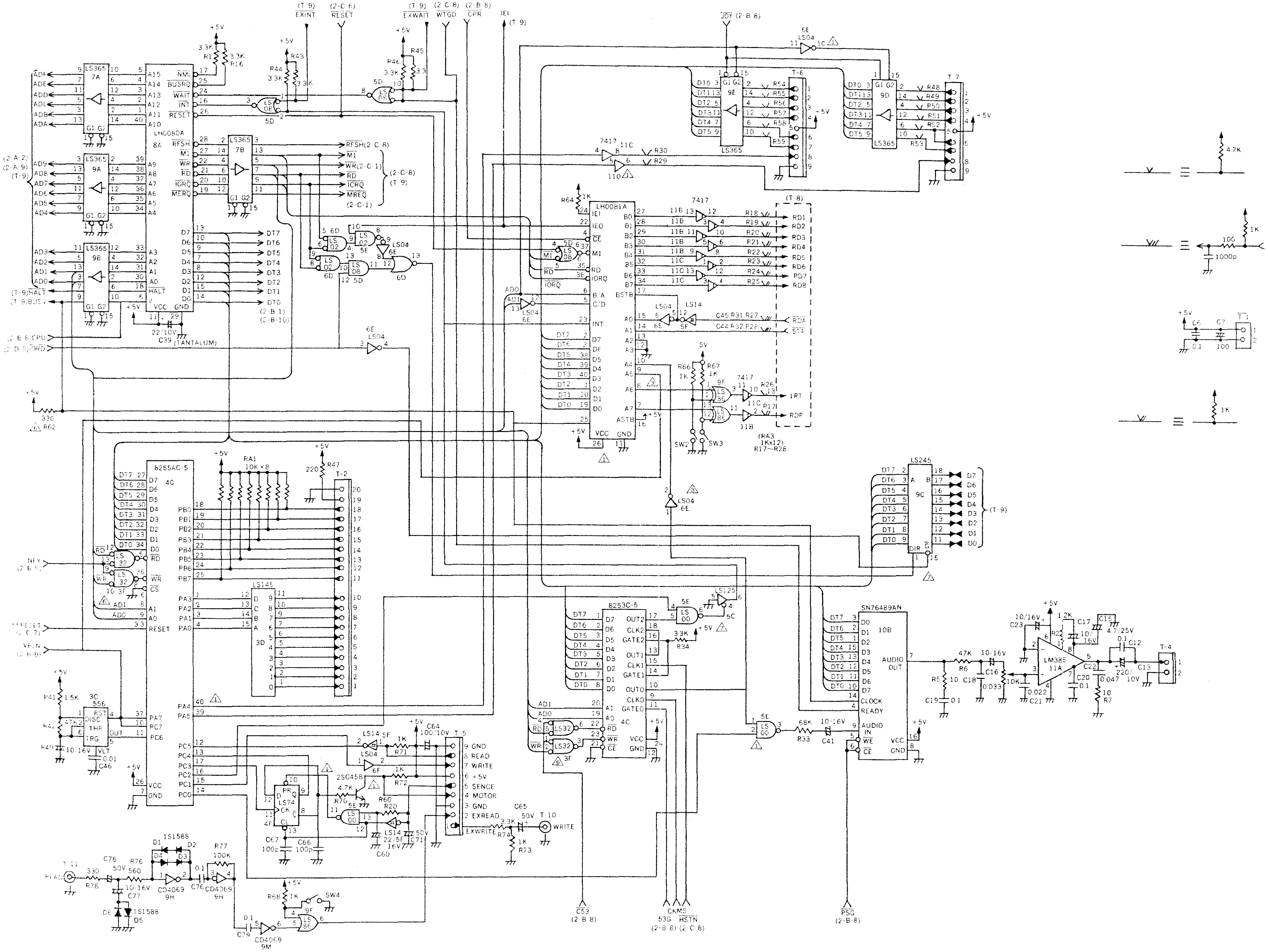
Specification

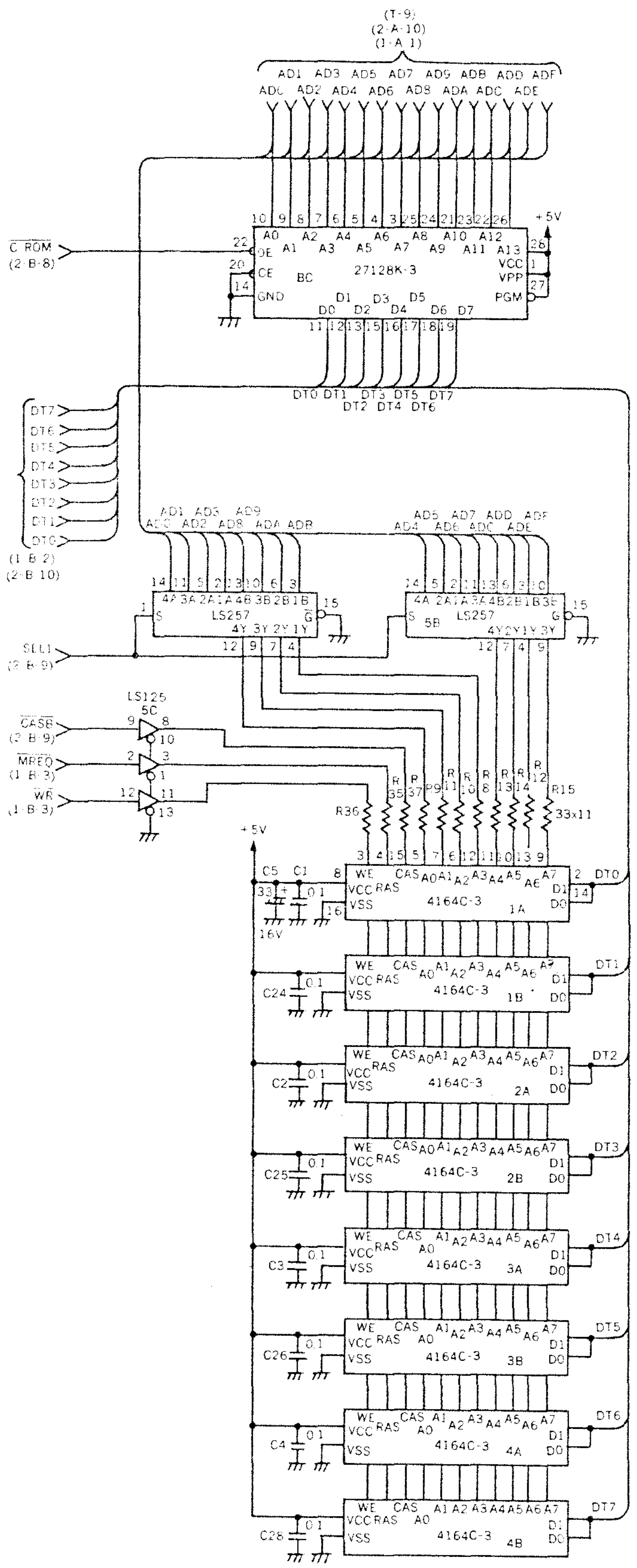
Type name	: MZ1P16
Print method	: 1, Black; 2, Blue; 3, Green; 4, Red
Print speed	: 10 characters average (smallest letter)
Printing size	: 80/40/26 digits (software assigned)
Character set	: 115
Resolution	: 0.2 mm
Power supply	: +5 V supplied from the MZ-800 via the DC jack
Power consumption	: 11 W
Physical dimensions	: 162(W) x 133(D) x 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

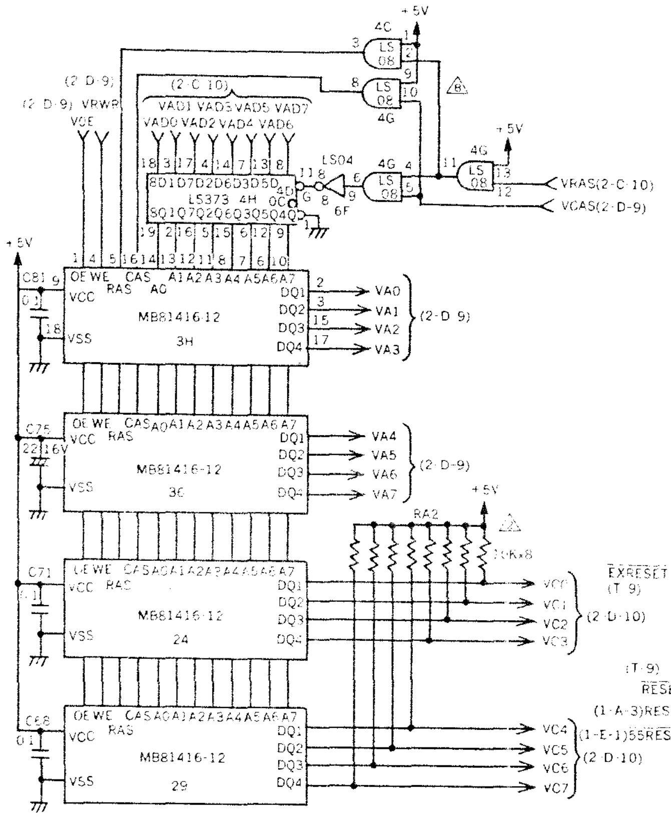
11-16 Circuit

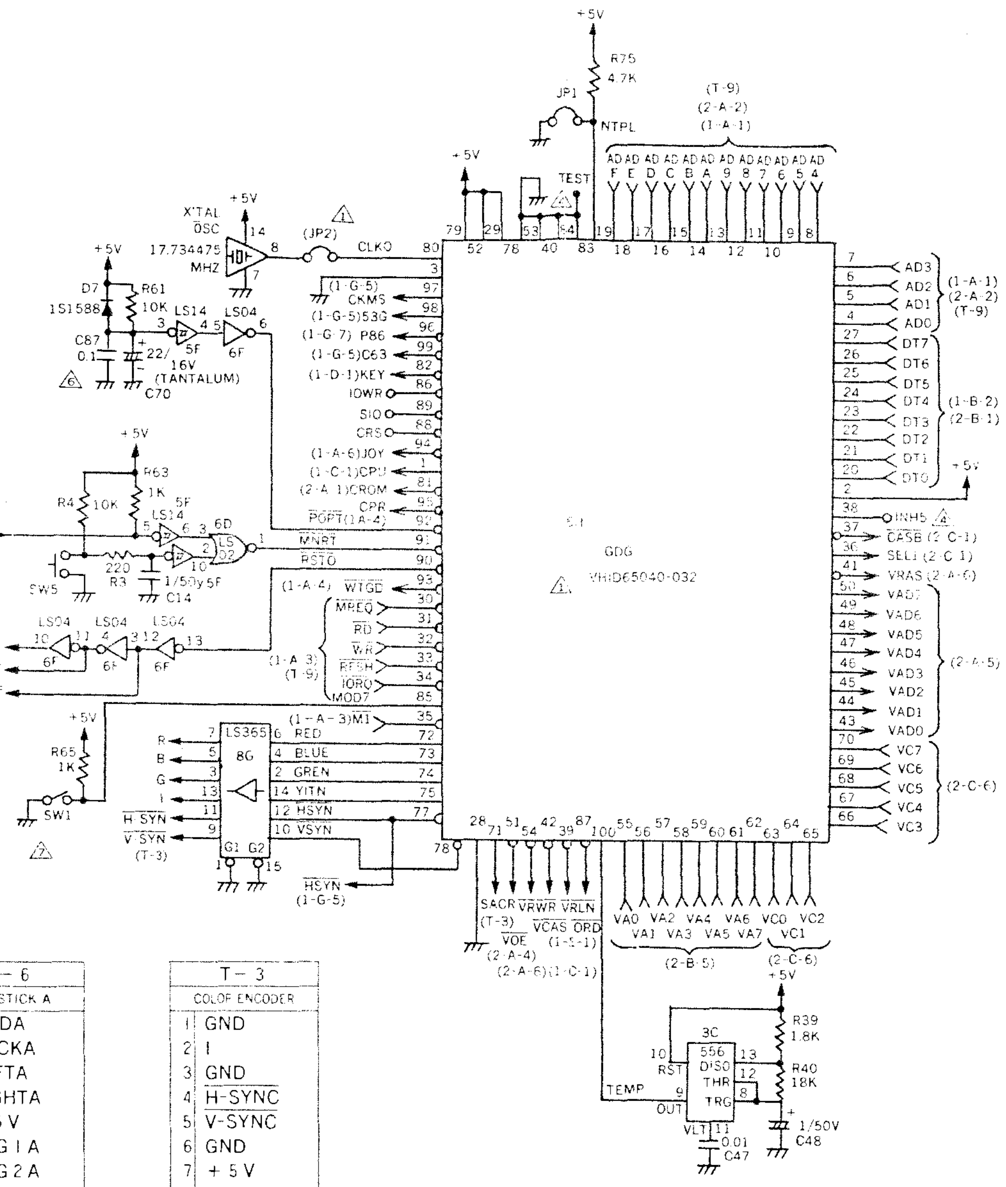


12 CPU P.W.B. Circuit









T−9			
PARTS SIDE		SOLDER SIDE	
1	+5V	2	+5V
3	D 2	4	D 3
5	D 1	6	D 4
7	D 0	8	D 5
9	GND	10	D 6
11	ADF	12	D 7
13	ADE	14	$\overline{\text{BUS}}$ ⚡
15	ADD	16	$\overline{\text{M 1}}$
17	ADC	18	$\overline{\text{WR}}$
19	ADB	20	$\overline{\text{RD}}$
21	ADA	22	$\overline{\text{IORQ}}$
23	AD 9	24	$\overline{\text{MREQ}}$
25	AD 8	26	GND
27	AD 7	28	$\overline{\text{HALT}}$
29	AD 6	30	IEI △
31	AD 5	32	NC.
33	AD 4	34	RESET
35	AD 3	36	$\overline{\text{EXRESET}}$
37	AD 2	38	$\overline{\text{EXINT}}$
39	AD 1	40	$\overline{\text{EXWAIT}}$
41	AD 0	42	NC.
43	GND	44	GND

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

T−7	
JOY STICK B	
1	FWDB
2	BACKB
3	LEFTB
4	RIGHTB
5	+5V
6	TRG 1 B
7	TRG 2 B
8	COMB
9	GND
10	GND
11	SW 1
12	SW 2
13	SW 3
14	SW 4
15	+5V

T−3	
COLOF ENCODER	
1	GND
2	I
3	GND
4	$\overline{\text{H-SYNC}}$
5	$\overline{\text{V-SYNC}}$
6	GND
7	+5V
8	G
9	B
10	R
11	SBCR
12	GND

T−8			
PARTS SIDE		SOLDER SIDE	
1	RDP	2	GND
3	RD 1	4	GND
5	RD 2	6	GND
7	RD 3	8	GND
9	RD 4	10	GND
11	RD 5	12	GND
13	RD 6	14	GND
15	RD 7	16	GND
17	RD 8	18	GND
19	IRT	20	GND
21	$\overline{\text{RDA}}$	22	GND
23	$\overline{\text{STA}}$	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 X 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	0 0 1 1 1 1 0 0
1679H	42H	0 1 0 0 0 0 1 0
167AH	81H	1 0 0 0 0 0 0 1
167BH	A5H	1 0 1 0 0 1 0 1
167CH	81H	1 0 0 0 0 0 0 1
167DH	99H	1 0 0 1 1 0 0 1
167EH	42H	0 1 0 0 0 0 1 0
167FH	3CH	0 0 1 1 1 1 0 0

↑ LSD ↑ MSD

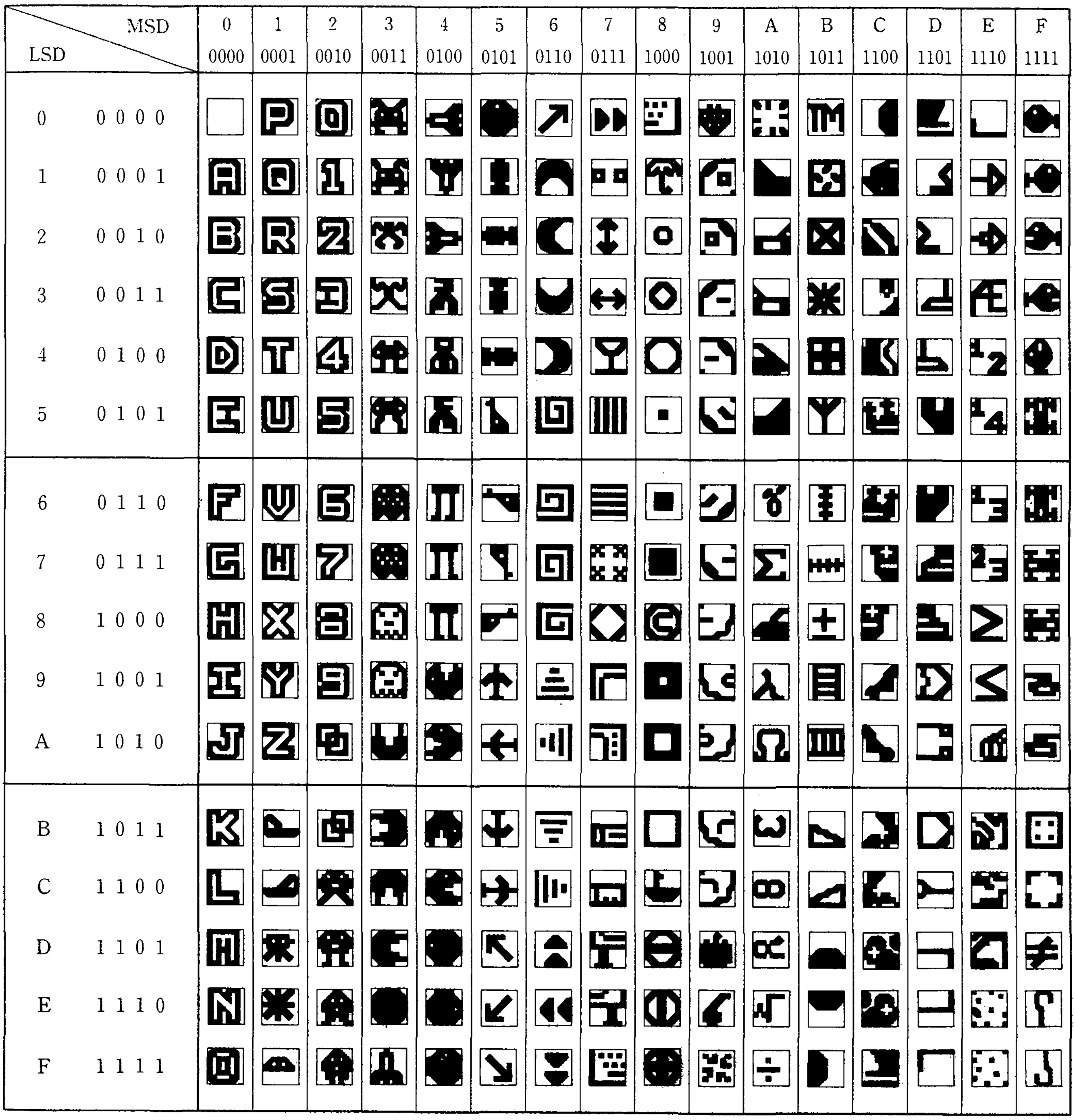
1-2 C.G.Table

■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 CIH to C6H are for controlling the cursor.

MSD LSD		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	Ø	▔	}	↑	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	▩	◿	◙	▚	Σ	▗
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	╲	K	▚
A	1 0 1 0	J	Z	−	▄	◘	→	+	▀	j	z	β	◡	웃	╲	K	▐
B	1 0 1 1	K	£	=	▐	◜	▚	*	▌	k	ä	ü	◸	⊀	°	⅄	▜
C	1 1 0 0	L	┗	;	▁	◣	┏	▞	▂	l	◿	ö	{	⊁	⁙	╢╟	▄
D	1 1 0 1	M	┛	/	□	◣	┓	☒	▐	m	◜	Ü	⊞	¥	╱	÷	▙
E	1 1 1 0	N	┣	.	▂	◢	╫	◸	▄	n	◝	Ä	^	☻	╱	Σ	▟
F	1 1 1 1	O	┻	,	▐	:	┳	◝	▐	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 22 1C 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 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 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 30 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 18 18 00 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 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 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 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 E0 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 22 1C 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 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 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 30 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 18 18 00 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 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 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 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 E0 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 C6 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 C0 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 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 3F FC 78 3C 7E FF FD FF 7E 3C 3C
0A70        3C 3C 7E FF BF FF 7E 3C 38 6C FF FF FF FF 7C 38
0A80        1C 36 FF FF FF FF 3E 1C 18 3C 3C 3C 3C 18 3C 3C
0A90        00 00 DE FF FF DE 00 00 3C 3C 18 3C 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 FF 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 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 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 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 2C 60 06 34 52 00
0D00       89 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 28 14 10 28 28 10 00 00 7F 42 04 08 04 42 7F
0D40       00 C0 20 10 FC FE FF FC 00 04 08 08 08 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 78 08 0A 0A 0D 08 00 00 00 08 00 3E 00 08 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 08 08 08 08 08
0DB0       1C 08 1C 08 1C 08 1C 08 00 00 00 55 FF 55 00 00
0DC0       00 08 08 3E 08 08 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 07 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 08 F8 F0 E0
0E40       3B 31 1B 1F 10 1F 0F 07 80 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 18 FE 02 FE FF
0EC0       0F 0F 0F 18 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 18 60 18 06 01 7F 80 60 18 06 18 60 80 FE
0F50       00 80 60 B8 54 54 54 F8 D8 F1 A6 88 93 95 8D CF
0F60       32 EF 0F 18 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 88
0F80       00 0C 1A BF FF 9E 0C 00 00 30 58 FD FF 79 30 00
0F90       00 0C 1A BF FC 9F 0C 00 00 30 58 FD 3F F9 30 00
0FA0       08 14 16 3D 3F 1E 08 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 38 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 08 7F 04 02
0FF0       18 24 24 04 08 08 08 08 10 10 10 10 20 24 24 18
```

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 freq. = 895 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.	All except AF

	Special key	Code loaded in ACC	Display code
Special key read with GETKY	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

<table>
<tr><th>Name and entry point (hex.)</th><th>Function</th><th>Registers saved</th></tr>
<tr><td>CALL ASC
(03DA)</td><td>Loads the ASCII character corresponding to the hexadecimal number represented by the lower 4 bits of data in ACC.</td><td>All except AF</td></tr>
<tr><td>CALL HEX
(03F9)</td><td>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.</td><td>All except AF</td></tr>
<tr><td>CALL HLHEX
(0410)</td><td>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 └─DE
CF=0 HL ← hexadecimal number (e.g., HL=31A5$_H$)
CF=1 The contents of HL are not guaranteed.</td><td>All except AF and HL</td></tr>
<tr><td>CALL 2HEX
(041F)</td><td>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 └─DE
CF=0 ACC ← hexadecimal number (e.g., ACC=3A$_H$)
CF=1 The contents of the ACC are not guaranteed.</td><td>All except AF and DE</td></tr>
<tr><td>CALL ??KEY
(09B3)</td><td>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.</td><td>All except AF</td></tr>
<tr><td>CALL ?ADCN
(0BB9)</td><td>Converts ASCII codes into display codes. The call and return conditions are as follows.
ACC ← ASCII code
CALL ?ADCN
ACC ← Display code</td><td>All except AF</td></tr>
<tr><td>CALL ?DACN
(0BCE)</td><td>Converts display codes into ASCII codes. The call and return conditions are as follows.
ACC ← Display codes
CALL ?DACN
ACC ← ASCII code</td><td>All except AF</td></tr>
<tr><td>CALL ?BLNK
(0DA6)</td><td>Detects the vertical blanking period. Control is returned to the calling program when the vertical blanking period is entered.</td><td>All registers</td></tr>
<tr><td>CALL ?DPCT
(0DDC)</td><td>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 $\boxed{\text{CLR}}$ key.</td></tr>
<tr><td>C1$_H$</td><td>Same as the $\boxed{\downarrow}$ key.</td><td>C7$_H$</td><td>Same as the $\boxed{\text{DEL}}$ key.</td></tr>
<tr><td>C2$_H$</td><td>Same as the $\boxed{\uparrow}$ key.</td><td>C8$_H$</td><td>Same as the $\boxed{\text{INST}}$ key.</td></tr>
<tr><td>C3$_H$</td><td>Same as the $\boxed{\rightarrow}$ key.</td><td>C9$_H$</td><td>Same as the $\boxed{\text{ALPHA}}$ key.</td></tr>
<tr><td>C4$_H$</td><td>Same as the $\boxed{\leftarrow}$ key.</td><td></td><td></td></tr>
<tr><td>C5$_H$</td><td>Same as the $\boxed{\text{HOME}}$ key.</td><td>CD$_H$</td><td>Same as the $\boxed{\text{CR}}$ key.</td></tr>
</table></td><td>All registers</td></tr>
<tr><td>CALL ?POINT
(0FB1)</td><td>Loads the current cursor location into the HL register. The return conditions are as follows.
CALL ?POINT
HL ← Cursor location (binary)</td><td>All except AF and HL</td></tr>
</table>

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 0DH.
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 .LPTOT

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 divice 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 dufault 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”
        DEFBO             ; 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>

```
   **  Z80 ASSEMBLER 2Z-004C  <1Z-013B>  PAGE 01           03.06.85

0000                ;
0000                ;
0000                ;    MONITOR PROGRAM 1Z-013B
0000                ;
0000                ;            (MZ-800) FOR PAL
0000                ;
0000                ;           REV. 1984.07.03
0000                ;
0000                ;
0000                MONIT:   ENT
0000 C300E8                  JP      E800H           ; MONITOR ON
0003                GETL:    ENT
0003 C3E607                  JP      ?GETL           ; GET LINE (END'CR')
0006                LETNL:   ENT
0006 C30E09                  JP      ?LTNL           ; NEW LINE
0009                NL:      ENT
0009 C31809                  JP      ?NL             ;
000C                PRNTS:   ENT
000C C32009                  JP      ?PRTS           ; PRINT SPACE
000F                PRNTT:   ENT
000F C32409                  JP      ?PRTT           ; PRINT TAB
0012                PRNT:    ENT
0012 C33509                  JP      ?PRNT           ; 1 CHARACTER PRINT
0015                MSG:     ENT
0015 C39308                  JP      ?MSG            ; 1 LINE PRINT (END'0DH'
)
0018                MSGX:    ENT
0018 C3A108                  JP      ?MSGX           ; RST 3
001B                GETKY:   ENT
001B C3BD08                  JP      ?GET            ; GET KEY
001E                BRKEY:   ENT
001E C3320A                  JP      ?BRK            ; GET BREAK
0021                WRINF:   ENT
0021 C33604                  JP      ?WRI            ; WRITE INFORMATION
0024                WRDAT:   ENT
0024 C37504                  JP      ?WRD            ; WRITE DATA
0027                RDINF:   ENT
0027 C3D804                  JP      ?RDI            ; READ INFORMATION
002A                RDDAT:   ENT
002A C3F804                  JP      ?RDD            ; READ DATA
002D                VERFY:   ENT
002D C38805                  JP      ?VRFY           ; VERIFING CMT
0030                MELDY:   ENT
0030 C3C701                  JP      ?MLDY           ; RST 6
0033                TIMST:   ENT
0033 C30803                  JP      ?TMST           ; TIME SET
0036 00                      NOP
0037 00                      NOP
0038 C33810                  JP      1038H           ; INTERRUPT ROUTINE
003B                TIMRD:   ENT
003B C35803                  JP      ?TMRD           ; TIME READ
003E                BELL:    ENT
003E C37705                  JP      ?BEL            ; BELL ON
0041                XTEMP:   ENT
0041 C3E502                  JP      ?TEMP           ; TEMPO SET (17)
0044                MSTA:    ENT
0044 C3AB02                  JP      MLDST           ; MELODY START
0047                MSTP:    ENT
0047 C3BE02                  JP      MLDSP           ; MELODY STOP
004A                ;
004A                ;
```

```
   **  Z80 ASSEMBLER 2Z-004C  <1Z-013B>  PAGE 02           03.06.85

004A                ;
004A                START:   ENT
004A 31F010                  LD      SP,SP           ; STACK SET (10F0H)
004D ED56                    IM      1               ; IM 1 SET
004F CD3E07                  CALL    ?MODE           ; 8255,8253 MODE SET
0052 CD320A                  CALL    ?BRK            ; CTRL ?
0055 3019                    JR      NC,ST0
0057 FE20                    CP      20H             ; KEY IS CTRL KEY
0059 2015                    JR      NZ,ST0
005B                CMY0:    ENT
005B D3E1                    OUT     (E1H),A         ; D000H-FFFFH IS DRAM
005D 11F0FF                  LD      DE,FFF0H        ; TRANS. ADR.
0060 216B00                  LD      HL,$MCP         ; MEMORY CHANG PROGRAM
0063 010500                  LD      BC,05           ; BYTE SIZE
0066 EDB0                    LDIR
0068 C3F0FF                  JP      FFF0H           ; JUMP $FFF0
006B                ;
006B                $MCP:    ENT                     ; 0000H-0FFFH IS DRAM
006B D3E0                    DEFW    E0D3H           ; OUT (E0H),A
006D C300                    DEFW    00C3H           ; JP 0000H
006F 00                      DEFB    00H
0070                ;
0070                ST0:     ENT
0070 06FF                    LD      B,FFH           ; BUFFER CLEAR
0072 21F110                  LD      HL,NAME         ; 10F1H-11F0H CLEAR
0075 CDD80F                  CALL    ?CLER
0078 3E16                    LD      A,16H           ; LASTER CLR.
007A CD1200                  CALL    PRNT
007D 3E71                    LD      A,71H           ; BACK:BLUE CHA.:WRITE
007F 2100D8                  LD      HL,D800H        ; COLOR ADDRESS
0082 CD0509                  CALL    #CLR8
0085 218D03                  LD      HL,TIMIN        ; INTERRUPT JUMP ROUTINE
0088 3EC3                    LD      A,C3H           ;
008A 323810                  LD      (1038H),A
008D 223910                  LD      (1039H),HL
0090 3E04                    LD      A,04            ; NORMAL TEMPO
0092 329E11                  LD      (TEMPW),A
0095 CDBE02                  CALL    MLDSP           ; MELODY STOP
0098 CD0900                  CALL    NL
009B 11E706                  LD      DE,MSG?3        ; ** MONITOR 1Z-013B **
009E DF                      RST     3               ; CALL MGX
009F CD7705                  CALL    ?BEL
00A2                SS:      ENT
00A2 3E01                    LD      A,01H
00A4 329D11                  LD      (SWRK),A        ; KEY IN SILENT
00A7 2100E8                  LD      HL,E800H        ; USR ROM ?
00AA 77                      LD      (HL),A          ; ROM CHECK
00AB 1855                    JR      FD2
00AD                ST1:     ENT
00AD CD0900                  CALL    NL
00B0 3E2A                    LD      A,2AH           ; '*' PRINT
00B2 CD1200                  CALL    PRNT
00B5 11A311                  LD      DE,BUFER        ; GET LINE WORK (11A3H)
00B8 CD0300                  CALL    GETL
00BB 1A             ST2:     LD      A,(DE)
00BC 13                      INC     DE
00BD FE0D                    CP      0DH
00BF 28EC                    JR      Z,ST1
00C1 FE4A                    CP      'J'             ; JUMP
00C3 282E                    JR      Z,GOTO
```

```
**  Z80 ASSEMBLER 2Z-004C  <1Z-013B>  PAGE 03              03.06.85

00C5 FE4C                     CP      'L'             ; LOAD PROGRAM
00C7 2848                     JR      Z,LOAD
00C9 FE46                     CP      'F'             ; FLOPPY ACCESS
00CB 2832                     JR      Z,FD
00CD FE42                     CP      'B'             ; KEY IN BELL
00CF 2826                     JR      Z,SG
00D1 FE23                     CP      '#'             ; CHANG MEMORY
00D3 2886                     JR      Z,CMY0
00D5 FE50                     CP      'P'             ; PRINTER TEST
00D7 287C                     JR      Z,PTEST
00D9 FE4D                     CP      'M'             ; MEMORY CORRECTION
00DB CAA807                   JP      Z,MCOR
00DE FE53                     CP      'S'             ; SAVED DATA
00E0 CA5E0F                   JP      Z,SAVE
00E3 FE56                     CP      'V'             ; VERIFYING DATA
00E5 CACB0F                   JP      Z,VRFY
00E8 FE44                     CP      'D'             ; DUMP DATA
00EA CA290D                   JP      Z,DUMP
00ED                 ;
00ED                 ;
00ED                          DEFS    +4
00F1                 ;
00F1 18C8                     JR      ST2             ; NOT COMMAND
00F3                 ;
00F3                 ;   JUMP COMMAND
00F3                 ;
00F3 CD3D01          GOTO:    CALL    HEXIY
00F6 E9                       JP      (HL)
00F7                 ;
00F7                 ;   KEY SOUND ON OFF
00F7                 ;
00F7 3A9D11          SG:      LD      A,(SWRK)        ; D0 = SOUND WORK
00FA 1F                       RRA
00FB 3F                       CCF                     ; CHENGE MODE
00FC 17                       RLA
00FD 18A5                     JR      SS+2
00FF                 ;
00FF                 ;   FLOPPY
00FF                 ;
00FF 2100F0          FD:      LD      HL,F000H        ; FLOPPY I/O CHECK
0102 7E              FD2:     LD      A,(HL)
0103 B7                       OR      A
0104 20A7                     JR      NZ,ST1
0106 E9              FD1:     JP      (HL)
0107                 ;
0107                 ;
0107                 ;   ERROR (LOADING)
0107                 ;
0107                 ?ER:     ENT                     ;
0107 FE02                     CP      02H             ; A=02H : BREAK IN
0109 28A2                     JR      Z,ST1           ;
010B 114701                   LD      DE,MSGE1        ; CHECK SUM ERROR
010E DF                       RST     3               ; CALL MSGX
010F 189C                     JR      ST1
0111                 ;
0111                 ;
0111                 ;   LOAD COMMAND
0111                 ;
0111 CDD804          LOAD:    CALL    ?RDI
0114 38F1                     JR      C,?ER
```

```
**  Z80 ASSEMBLER 2Z-004C  <1Z-013B>  PAGE 04              03.06.85

0116 CD0900          LOA0:    CALL    NL
0119 11A009                   LD      DE,MSG?2        ; LOADING
011C DF                       RST     3               ; CALL MSGX
011D 11F110                   LD      DE,NAME         ; FILE NAME
0120 DF                       RST     3               ; CALL MSGX
0121 CDF804                   CALL    ?RDD
0124 38E1                     JR      C,?ER
0126 2A0611                   LD      HL,(EXADR)      ; EXECUTE ADDRESS
0129 7C                       LD      A,H
012A FE12                     CP      12H             ; EXECUTE CHECK
012C 38E1                     JR      C,LOAD-2
012E E9                       JP      (HL)
012F                 ;
012F                 ;
012F                 ;
012F                 ;       GETLINE AND BREAK IN CHECK
012F                 ;
012F                 ;     EXIT BREAK IN THEN JUMP (ST1)
012F                 ;         ACC=TOP OF LINE DATA
012F                 ;
012F                 BGETL:   ENT
012F E3                       EX      (SP),HL
0130 C1                       POP     BC              ; STACK LOAD
0131 11A311                   LD      DE,BUFER        ; MONITOR GETLINE BUFF
0134 CD0300                   CALL    GETL
0137 1A                       LD      A,(DE)
0138 FE1B                     CP      1BH             ; BREAK CODE
013A 28D3                     JR      Z,LOAD-2        ; JP Z,ST1
013C E9                       JP      (HL)
013D                 ;
013D                 ;    ASCII TO HEX CONVERT
013D                 ;      INPUT (DE)=ASCII
013D                 ;      CY=1 THEN JUMP (ST1)
013D                 ;
013D                 HEXIY:   ENT
013D FDE3                     EX      (SP),IY
013F F1                       POP     AF
0140 CD1004                   CALL    HLHEX
0143 38CA                     JR      C,LOAD-2        ; JP C,ST1
0145 FDE9                     JP      (IY)
0147                 ;
0147                 ;
0147                 ;
0147                 MSGE1:   ENT
0147 43484543                 DEFM    'CHECK SUM ER.'
014B 4B205355
014F 4D204552
0153 2E
0154 0D                       DEFB    0DH
0155                 ;
0155                 ;
0155                 ;    PLOTTER PRINTER TEST COMMAND
0155                 ;                (DPG23)
0155                 ;       &=CONTROL COMMANDS GROUP
0155                 ;           C=PEN CHENGE
0155                 ;           G=GRAPH MODE
0155                 ;           S=80 CHA. IN 1 LINE
0155                 ;           L=40 CHA. IN 1 LINE
0155                 ;           T=PLOTTER TEST
0155                 ;     IN  (DE)=PRINT DATA
```

```
0155                  ;
0155                  PTEST:  ENT
0155 1A                       LD      A,(DE)
0156 FE26                     CP      '&'
0158 2016                     JR      NZ,PTST1
015A 13               PTST0:  INC     DE
015B 1A                       LD      A,(DE)
015C FE4C                     CP      'L'             ; 80 IN 1 LINE
015E 2816                     JR      Z,.LPT
0160 FE53                     CP      'S'             ; 80 IN 1LINE
0162 2817                     JR      Z,..LPT
0164 FE43                     CP      'C'             ; PEN CHENGE
0166 2823                     JR      Z,PEN
0168 FE47                     CP      'G'             ; GRAPH MODE
016A 2818                     JR      Z,PLOT
016C FE54                     CP      'T'             ; TEST
016E 2810                     JR      Z,PTRN
0170                  ;
0170 CDA501           PTST1:  CALL    PMSG            ; PLOT MESSAGE
0173 C3AD00                   JP      ST1
0176                  ;
0176 117004           .LPT:   LD      DE,LLPT         ; 01-09-09-0B-0D
0179 18F5                     JR      PTST1
017B                  ;
017B 11D503           ..LPT:  LD      DE,SLPT         ; 01-09-09-09-0D
017E 18F0                     JR      PTST1
0180                  ;
0180 3E04             PTRN:   LD      A,04H           ; TEST PATTERN
0182 1802                     JR      PLOT+2
0184                  ;
0184 3E02             PLOT:   LD      A,02H           ; GRAPH CODE
0186 CD8F01                   CALL    LPRNT
0189 18CF                     JR      PTST0
018B                  ;
018B 3E1D             PEN:    LD      A,1DH           ; 1 CHENGE CODE (TEXT MO
DE)
018D 18F7                     JR      PLOT+2
018F                  ;
018F                  ;
018F                  ;     1CHA. PRINT TO $LPT
018F                  ;
018F                  ;        IN: ACC PRINT DATA
018F                  ;
018F                  ;
018F 0E00             LPRNT:  LD      C,0             ; RDA TEST
0191 47                       LD      B,A             ; PRINT DATA STORE
0192 CDB601                   CALL    RDA
0195 78                       LD      A,B
0196 D3FF                     OUT     (FFH),A         ; DATA OUT
0198 3E80                     LD      A,80H           ; RDP HIGH
019A D3FE                     OUT     (FEH),A
019C 0E01                     LD      C,01H           ; RDA TEST
019E CDB601                   CALL    RDA
01A1 AF                       XOR     A               ; RDP LOW
01A2 D3FE                     OUT     (FEH),A
01A4 C9                       RET
01A5                  ;
01A5                  ;     $LPT  MSG.
01A5                  ;        IN: DE DATA LOW ADR.
01A5                  ;          0DH  MSG. END
01A5                  ;
```

```
01A5 D5               PMSG:   PUSH    DE
01A6 C5                       PUSH    BC
01A7 F5                       PUSH    AF
01A8 1A               PMSG1:  LD      A,(DE)          ; ACC=DATA
01A9 CD8F01                   CALL    LPRNT
01AC 1A                       LD      A,(DE)
01AD 13                       INC     DE
01AE FE0D                     CP      0DH             ; END ?
01B0 20F6                     JR      NZ,PMSG1
01B2 F1                       POP     AF
01B3 C1                       POP     BC
01B4 D1                       POP     DE
01B5 C9                       RET
01B6                  ;
01B6                  ;    RDA CHECK
01B6                  ;
01B6                  ;        BRKEY IN TO MONITOR RETURN
01B6                  ;      IN: C RDA CODE
01B6                  ;
01B6 DBFE             RDA:    IN      A,(FEH)
01B8 E60D                     AND     0DH
01BA B9                       CP      C
01BB C8                       RET     Z
01BC CD1E00                   CALL    BRKEY
01BF 20F5                     JR      NZ,RDA
01C1 31F010                   LD      SP,SP
01C4 C3AD00                   JP      ST1
01C7                  ;
01C7                  ;
01C7                  ;ORG 01C7H
01C7                  ;
01C7                  ;    MELODY
01C7                  ;
01C7                  ;      DE=DATA LOW ADR.
01C7                  ;    EXIT  CF=1 BREAK
01C7                  ;          CF=0 OK
01C7                  ;
01C7                  ?MLDY:  ENT
01C7 C5                       PUSH    BC
01C8 D5                       PUSH    DE
01C9 E5                       PUSH    HL
01CA 3E02                     LD      A,02H
01CC 32A011                   LD      (OCTV),A
01CF 0601                     LD      B,01
01D1 1A               MLD1:   LD      A,(DE)
01D2 FE0D                     CP      0DH             ; CR
01D4 283B                     JR      Z,MLD4
01D6 FEC8                     CP      C8H             ; END MARK
01D8 2837                     JR      Z,MLD4
01DA FECF                     CP      CFH             ; UNDER OCTAVE
01DC 2827                     JR      Z,MLD2
01DE FE2D                     CP      2DH             ; '-'
01E0 2823                     JR      Z,MLD2
01E2 FE2B                     CP      2BH             ; '+'
01E4 2827                     JR      Z,MLD3
01E6 FED7                     CP      D7H             ; UPPER OCTAVE
01E8 2823                     JR      Z,MLD3
01EA FE23                     CP      23H             ; "#" HANON
01EC 216C02                   LD      HL,MTBL
01EF 2004                     JR      NZ,+6
```

```
01F1 218402             LD      HL,M#TBL
01F4 13                 INC     DE
01F5 CD1C02             CALL    ONPU            ; ONTYO SET
01F8 38D7               JR      C,MLD1
01FA CDC802             CALL    RYTHM           ;
01FD 3815               JR      C,MLD5
01FF CDAB02             CALL    MLDST           ; MELODY START
0202 41                 LD      B,C
0203 18CC               JR      MLD1
0205 3E03       MLD2:   LD      A,+3
0207 32A011             LD      (OCTV),A
020A 13                 INC     DE
020B 18C4               JR      MLD1
020D 3E01       MLD3:   LD      A,1
020F 18F6               JR      MLD2+2
0211 CDC802     MLD4:   CALL    RYTHM
0214 F5         MLD5:   PUSH    AF
0215 CDBE02             CALL    MLDSP
0218 F1                 POP     AF
0219 C39B06             JP      RET3
021C            ;
021C            ;   ONPU TO RATIO CONV
021C            ;
021C            ;    EXIT (RATIO)=RATIO VALUE
021C            ;         C=ONTYO*TEMPO
021C            ;
021C            ONPU:   ENT
021C C5                 PUSH    BC
021D 0608               LD      B,8
021F 1A         ONP1:   LD      A,(DE)
0220 BE                 CP      (HL)
0221 2809               JR      Z,ONP2
0223 23                 INC     HL
0224 23                 INC     HL
0225 23                 INC     HL
0226 10F8               DJNZ    -6
0228 37                 SCF
0229 13                 INC     DE
022A C1                 POP     BC
022B C9                 RET
022C 23         ONP2:   INC     HL
022D D5                 PUSH    DE
022E 5E                 LD      E,(HL)
022F 23                 INC     HL
0230 56                 LD      D,(HL)
0231 EB                 EX      DE,HL
0232 7C                 LD      A,H
0233 B7                 OR      A
0234 2809               JR      Z,+11
0236 3AA011             LD      A,(OCTV)        ; 11A0H OCTAVE WORK
0239 3D                 DEC     A
023A 2803               JR      Z,+5
023C 29                 ADD     HL,HL
023D 18FA               JR      -4
023F 22A111             LD      (RATIO),HL      ; 11A1H ONPU RATIO
0242 21A011             LD      HL,OCTV
0245 3602               LD      (HL),2
0247 2B                 DEC     HL
0248 D1                 POP     DE
0249 13                 INC     DE
```

```
024A 1A                 LD      A,(DE)
024B 47                 LD      B,A
024C E6F0               AND     F0H             ; ONTYO ?
024E FE30               CP      30H
0250 2803               JR      Z,+5
0252 7E                 LD      A,(HL)          ; HL=ONTYO
0253 1805               JR      +7
0255 13                 INC     DE
0256 78                 LD      A,B
0257 E60F               AND     0FH
0259 77                 LD      (HL),A          ; HL=ONTYO
025A 219C02             LD      HL,OPTBL
025D 85                 ADD     A,L
025E 6F                 LD      L,A
025F 4E                 LD      C,(HL)
0260 3A9E11             LD      A,(TEMPW)
0263 47                 LD      B,A
0264 AF                 XOR     A
0265 81         ONP3:   ADD     A,C
0266 10FD               DJNZ    -1
0268 C1                 POP     BC
0269 4F                 LD      C,A
026A AF                 XOR     A
026B C9                 RET
026C            ;
026C            ;
026C            MTBL:   ENT
026C 43                 DEFB    43H             ; C
026D 4608               DEFW    0846H
026F 44                 DEFB    44H             ; D
0270 5F07               DEFW    075FH
0272 45                 DEFB    45H             ; E
0273 9106               DEFW    0691H
0275 46                 DEFB    46H             ; F
0276 3306               DEFW    0633H
0278 47                 DEFB    47H             ; G
0279 8605               DEFW    0586H
027B 41                 DEFB    41H             ; A
027C EC04               DEFW    04ECH
027E 42                 DEFB    42H             ; B
027F 6404               DEFW    0464H
0281 52                 DEFB    52H             ; R
0282 0000               DEFW    0
0284            M#TBL:  ENT
0284 43                 DEFB    43H             ; #C
0285 CF07               DEFW    07CFH
0287 44                 DEFB    44H             ; #D
0288 F506               DEFW    06F5H
028A 45                 DEFB    45H             ; #E
028B 3306               DEFW    0633H
028D 46                 DEFB    46H             ; #F
028E DA05               DEFW    05DAH
0290 47                 DEFB    47H             ; #G
0291 3705               DEFW    0537H
0293 41                 DEFB    41H             ; #A
0294 A504               DEFW    04A5H
0296 42                 DEFB    42H             ; #B
0297 2304               DEFW    0423H
0299 52                 DEFB    52H             ; #R
029A
```

```
029A 0000              DEFW   0
029C          OPTBL:   ENT
029C 01                DEFB   1
029D 02                DEFB   2
029E 03                DEFB   3
029F 04                DEFB   4
02A0 06                DEFB   6
02A1 08                DEFB   8
02A2 0C                DEFB   0CH
02A3 10                DEFB   10H
02A4 18                DEFB   18H
02A5 20                DEFB   20H
02A6          ;
02A6          ;
02A6          ;
02A6          ;       INCREMENT DE REG.
02A6          ;
02A6          .4DE:    ENT
02A6 13                INC    DE
02A7 13                INC    DE
02A8 13                INC    DE
02A9 13                INC    DE
02AA C9                RET
02AB          ;
02AB          ;
02AB          ;
02AB          ;ORG 02ABH ; MLDST
02AB          ;
02AB          ;    MELODY START & STOP
02AB          ;
02AB          MLDST:   ENT
02AB 2AA111            LD     HL,(RATIO)
02AE 7C                LD     A,H
02AF B7                OR     A
02B0 280C              JR     Z,MLDSP
02B2 D5                PUSH   DE
02B3 EB                EX     DE,HL
02B4 2104E0            LD     HL,CONT0
02B7 73                LD     (HL),E
02B8 72                LD     (HL),D
02B9 3E01              LD     A,1
02BB D1                POP    DE
02BC 1806              JR     MLDS1
02BE          ;
02BE          MLDSP:   ENT
02BE 3E36              LD     A,36H          ; MODE SET (8253 C0)
02C0 3207E0            LD     (CONTF),A      ; E007H
02C3 AF                XOR    A
02C4 3208E0   MLDS1:   LD     (SUNDG),A      ; E008H
02C7 C9                RET                   ; TEHRO RESET
02C8          ;
02C8          ;    RHYTHM
02C8          ;
02C8          ;      B=COUNT DATA
02C8          ;          IN
02C8          ;     EXIT CF=1 BREAK
02C8          ;          CF=0 OK
02C8          ;
02C8          RYTHM:   ENT
02C8 2100E0            LD     HL,KEYPA       ; E000H
```

```
02CB 36F8              LD     (HL),F8H
02CD 23                INC    HL
02CE 7E                LD     A,(HL)
02CF E681              AND    81H            ; BREAK IN CHECK
02D1 2002              JR     NZ,+4
02D3 37                SCF
02D4 C9                RET
02D5 3A08E0            LD     A,(TEMP)       ; E008H
02D8 0F                RRCA                  ; TEMPO OUT
02D9 38FA              JR     C,-4
02DB 3A08E0            LD     A,(TEMP)
02DE 0F                RRCA
02DF 30FA              JR     NC,-4
02E1 10F2              DJNZ   -12
02E3 AF                XOR    A
02E4 C9                RET
02E5          ;
02E5          ;
02E5          ;   TEMPO SET
02E5          ;
02E5          ;     ACC=VALUE (1-7)
02E5          ;
02E5          ?TEMP:   ENT
02E5 F5                PUSH   AF
02E6 C5                PUSH   BC
02E7 E60F              AND    0FH
02E9 47                LD     B,A
02EA 3E08              LD     A,8
02EC 90                SUB    B
02ED 329E11            LD     (TEMPW),A
02F0 C1                POP    BC
02F1 F1                POP    AF
02F2 C9                RET
02F3          ;
02F3          ;         CRT MANAGMENT
02F3          ;
02F3          ;     EXIT  HL:DSPXY H=Y,L=X
02F3          ;           DE:MANG ADR. (ON DSPXY)
02F3          ;           A :MANG DATA
02F3          ;           CY:MANG=1
02F3          ;
02F3          .MANG:   ENT
02F3 217311            LD     HL,MANG        ; CRT MANG. POINTER
02F6 3A7211            LD     A,(1172H)      ; DSPXY+1
02F9 85                ADD    A,L
02FA 6F                LD     L,A
02FB 7E                LD     A,(HL)
02FC 23                INC    HL
02FD CB16              RL     (HL)
02FF B6                OR     (HL)
0300 CB1E              RR     (HL)
0302 0F                RRCA
0303 EB                EX     DE,HL
0304 2A7111            LD     HL,(DSPXY)
0307 C9                RET
0308          ;
0308          ;
0308          ;
0308          ;ORG 0308H
0308          ;
```

```
0308                   ;  TIME SET
0308                   ;
0308                   ;    ACC=0 : AM
0308                   ;       =1 : PM
0308                   ;    DE=SEC: BINARY
0308                   ;
0308                   ?TMST:  ENT
0308 F3                        DI
0309 C5                        PUSH   BC
030A D5                        PUSH   DE
030B E5                        PUSH   HL
030C 329B11                    LD     (AMPM),A       ; AMPM DATA
030F 3EF0                      LD     A,F0H
0311 329C11                    LD     (TIMFG),A      ; TIME FLAG
0314 21C0A8                    LD     HL,A8C0H       ; 12H
0317 AF                        XOR    A
0318 ED52                      SBC    HL,DE          ; COUNT DATA = 12H-IN DA
TA
031A E5                        PUSH   HL
031B 00                        NOP
031C EB                        EX     DE,HL
031D 2107E0                    LD     HL,CONTF       ; E007H
0320 3674                      LD     (HL),74H
0322 36B0                      LD     (HL),B0H
0324 2B                        DEC    HL             ; CONT2
0325 73                        LD     (HL),E
0326 72                        LD     (HL),D
0327 2B                        DEC    HL             ; CONT1
0328 360A                      LD     (HL),0AH
032A 3600                      LD     (HL),0
032C 23                        INC    HL
032D 23                        INC    HL             ; CONTF
032E 3680                      LD     (HL),80H
0330 2B                        DEC    HL             ; CONT2
0331 4E                ?TMS1:  LD     C,(HL)
0332 7E                        LD     A,(HL)
0333 BA                        CP     D
0334 20FB                      JR     NZ,?TMS1
0336 79                        LD     A,C
0337 BB                        CP     E
0338 20F7                      JR     NZ,?TMS1
033A 2B                        DEC    HL
033B 00                        NOP
033C 00                        NOP
033D 00                        NOP
033E 36FB                      LD     (HL),FBH       ; 1SEC
0340 363C                      LD     (HL),3CH
0342 23                        INC    HL
0343 D1                        POP    DE
0344 4E                ?TMS2:  LD     C,(HL)
0345 7E                        LD     A,(HL)
0346 BA                        CP     D
0347 20FB                      JR     NZ,?TMS2
0349 79                        LD     A,C
034A BB                        CP     E
034B 20F7                      JR     NZ,?TMS2
034D E1                        POP    HL
034E D1                        POP    DE
034F C1                        POP    BC
0350 FB                        EI
0351 C9                        RET
```

```
0352                   ;
0352                   ;
0352                   ;   BELL DATA
0352                   ;
0352                   ?BELD:  ENT
0352 D7                        DEFB   D7H
0353 4130                      DEFM   'A0'
0355 0D                        DEFB   0DH
0356                   ;
0356                   ;
0356                   ;
0356                           DEFS   +2
0358                   ;ORG 0358H
0358                   ;
0358                   ;   TIME READ
0358                   ;
0358                   ;     EXIT  ACC=0 :AM
0358                   ;                =1 :PM
0358                   ;           DE=SEC. BINARY
0358                   ;
0358                   ?TMRD:  ENT
0358 E5                        PUSH   HL
0359 2107E0                    LD     HL,CONTF
035C 3680                      LD     (HL),80H
035E 2B                        DEC    HL             ; CONT2
035F F3                        DI
0360 5E                        LD     E,(HL)
0361 56                        LD     D,(HL)
0362 FB                        EI
0363 7B                        LD     A,E
0364 B2                        OR     D
0365 280E                      JR     Z,?TMR1
0367 AF                        XOR    A
0368 21C0A8                    LD     HL,A8C0H
036B ED52                      SBC    HL,DE
036D 3810                      JR     C,?TMR2
036F EB                        EX     DE,HL
0370 3A9B11                    LD     A,(AMPM)
0373 E1                        POP    HL
0374 C9                        RET
0375 11C0A8            ?TMR1:  LD     DE,A8C0H       ; 12H
0378 3A9B11                    LD     A,(AMPM)
037B EE01                      XOR    1
037D E1                        POP    HL
037E C9                        RET
037F F3                ?TMR2:  DI
0380 2106E0                    LD     HL,CONT2
0383 7E                        LD     A,(HL)
0384 2F                        CPL
0385 5F                        LD     E,A
0386 7E                        LD     A,(HL)
0387 2F                        CPL
0388 57                        LD     D,A
0389 FB                        EI
038A 13                        INC    DE
038B 18EB                      JR     ?TMR1+3
038D                   ;
038D                   ;   TIME INTERRUPT
038D                   ;
038D                   TIMIN:  ENT
```

```
038D F5               PUSH   AF
038E C5               PUSH   BC
038F D5               PUSH   DE
0390 E5               PUSH   HL
0391 219B11           LD     HL,AMPM
0394 7E               LD     A,(HL)
0395 EE01             XOR    1
0397 77               LD     (HL),A
0398 2107E0           LD     HL,CONTF
039B 3680             LD     (HL),80H
039D 2B               DEC    HL              ; CONT2
039E E5               PUSH   HL
039F 5E               LD     E,(HL)
03A0 56               LD     D,(HL)
03A1 21C0A8           LD     HL,A8C0H
03A4 19               ADD    HL,DE
03A5 2B               DEC    HL
03A6 2B               DEC    HL
03A7 EB               EX     DE,HL
03A8 E1               POP    HL
03A9 73               LD     (HL),E
03AA 72               LD     (HL),D
03AB E1               POP    HL
03AC D1               POP    DE
03AD C1               POP    BC
03AE F1               POP    AF
03AF FB               EI
03B0 C9               RET
03B1          ;
03B1          ;       SPACE PRINT AND DISP ACC
03B1          ;
03B1          ;          INPUT:HL=DISP. ADR.
03B1          ;
03B1          SPHEX:  ENT
03B1 CD2009           CALL   ?PRTS           ; SP.PRINT
03B4 7E               LD     A,(HL)
03B5 CDC303           CALL   PRTHX           ; DSP OF ACC (ASCII)
03B8 7E               LD     A,(HL)
03B9 C9               RET
03BA          ;
03BA          ;
03BA          ;
03BA          ;ORG 03BAH
03BA          ;
03BA          ;  (ASCII PRINT) FOR HL
03BA          ;
03BA          PRTHL:  ENT
03BA 7C               LD     A,H
03BB CDC303           CALL   PRTHX
03BE 7D               LD     A,L
03BF 1802             JR     PRTHX
03C1          ;
03C1                  DEFS   +2
03C3          ;ORG 03C3H;PRTHX
03C3          ;
03C3          ;   (ASCII PRINT) FOR ACC
03C3          ;
03C3          PRTHX:  ENT
03C3 F5               PUSH   AF
03C4 0F               RRCA
```

```
03C5 0F               RRCA
03C6 0F               RRCA
03C7 0F               RRCA
03C8 CDDA03           CALL   ASC
03CB CD1200           CALL   PRNT
03CE F1               POP    AF
03CF CDDA03           CALL   ASC
03D2 C31200           JP     PRNT
03D5          ;
03D5          ;
03D5          ;
03D5          ;
03D5          ;    80 CHA. 1 LINE CODE (DATA)
03D5          ;
03D5          SLPT:   ENT
03D5 01               DEFB   01H             ; TEXT MODE
03D6 09               DEFB   09H
03D7 09               DEFB   09H
03D8 09               DEFB   09H
03D9 0D               DEFB   0DH
03DA          ;
03DA          ;ORG 03DAH;ASC
03DA          ;
03DA          ;  HEXADECIMAL TO ASCII
03DA          ;    IN : ACC (D3-D0)=HEXADECIMAL
03DA          ;   EXIT: ACC = ASCII
03DA          ;
03DA          ASC:    ENT
03DA E60F             AND    0FH
03DC FE0A             CP     0AH
03DE 3802             JR     C,NOADD
03E0 C607             ADD    A,7
03E2          NOADD:  ENT
03E2 C630             ADD    A,30H
03E4 C9               RET
03E5          ;
03E5          ;  ASCII TO HEXADECIMAL
03E5          ;  IN  : ACC = ASCII
03E5          ; EXIT : ACC = HEXADECIMAL
03E5          ;         CY  = 1 ERROR
03E5          ;
03E5          HEXJ:   ENT
03E5 D630             SUB    30H
03E7 D8               RET    C
03E8 FE0A             CP     0AH
03EA 3F               CCF
03EB D0               RET    NC
03EC D607             SUB    7
03EE FE10             CP     10H
03F0 3F               CCF
03F1 D8               RET    C
03F2 FE0A             CP     0AH
03F4 C9               RET
03F5          ;
03F5          ;
03F5                  DEFS   +4
03F9          ;ORG 03F9H;HEX
03F9          HEX:    ENT
03F9 18EA             JR     HEXJ
03FB          ;
```

```
03FB                    ;     PRASS PLAY MESSAGE
03FB                    ;
03FB                    MSG#1:  ENT
03FB 7F20                       DEFW   207FH
03FD                    MSG#2:  ENT
03FD 504C4159                   DEFM   'PLAY'
0401 0D                         DEFB   0DH
0402                    MSG#3:  ENT
0402 7F20                       DEFW   207FH
0404 5245434F                   DEFM   'RECORD.'        ; PRESS RECORD
0408 52442E
040B 0D                         DEFB   0DH
040C                    ;
040C                    ;
040C                            DEFS   +4
0410                    ;ORG 0410H;HLHEX
0410                    ;
0410                    ;
0410                    ;   4 ASCII TO (HL)
0410                    ;
0410                    ;   IN  DE=DATA LOW ADR.
0410                    ;   EXIT  CF=0 : OK
0410                    ;           =1 : OUT
0410                    ;
0410                    HLHEX:  ENT
0410 D5                         PUSH   DE
0411 CD1F04                     CALL   2HEX
0414 3807                       JR     C,+9
0416 67                         LD     H,A
0417 CD1F04                     CALL   2HEX
041A 3801                       JR     C,+3
041C 6F                         LD     L,A
041D D1                 HL1:    POP    DE
041E C9                         RET
041F                    ;
041F                    ;ORG 041FH;2HEX
041F                    ;
041F                    ;
041F                    ;    2 ASCII TO (ACC)
041F                    ;
041F                    ;    IN  DE=DATA  LOW ADR.
041F                    ;
041F                    ;    EXIT  CF=0 : OK
041F                    ;            =1 : OUT
041F                    ;
041F                    2HEX:   ENT
041F C5                         PUSH   BC
0420 1A                         LD     A,(DE)
0421 13                         INC    DE
0422 CDF903                     CALL   HEX
0425 380D                       JR     C,+15
0427 0F                         RRCA
0428 0F                         RRCA
0429 0F                         RRCA
042A 0F                         RRCA
042B 4F                         LD     C,A
042C 1A                         LD     A,(DE)
042D 13                         INC    DE
042E CDF903                     CALL   HEX
0431 3801                       JR     C,+3
```

```
0433 B1                         OR     C
0434 C1                 2HE1:   POP    BC
0435 C9                         RET
0436                    ;
0436                    ;
0436                    ;   WRITE INFORMATION
0436                    ;
0436                    ?WRI:   ENT
0436 F3                         DI
0437 D5                         PUSH   DE
0438 C5                         PUSH   BC
0439 E5                         PUSH   HL
043A 16D7                       LD     D,D7H            ; 'W'
043C 1ECC                       LD     E,CCH            ; 'L'
043E 21F010                     LD     HL,IBUFE         ; 10F0H
0441 018000                     LD     BC,80H           ; WRITE BYTE SIZE
0444 CD1A07             WRI1:   CALL   CKSUM            ; CHECK SUM
0447 CD9F06                     CALL   MOTOR            ; MOTOR ON
044A 3818                       JR     C,WRI3
044C 7B                         LD     A,E
044D FECC                       CP     CCH              ; 'L'
044F 200D                       JR     NZ,WRI2
0451 CD0900                     CALL   NL
0454 D5                         PUSH   DE
0455 116704                     LD     DE,MSG#7         ; WRITING
0458 DF                         RST    3                ; CALL MSGX
0459 11F110                     LD     DE,NAME          ; FILE NAME
045C DF                         RST    3                ; CALL MSGX
045D D1                         POP    DE
045E CD7A07             WRI2:   CALL   GAP
0461 CD8A04                     CALL   WTAPE
0464 C35405             WRI3:   JP     RET2
0467                    ;
0467                    MSG#7:  ENT
0467 57524954                   DEFM   'WRITING '
046B 494E4720
046F 0D                         DEFB   0DH
0470                    ;
0470                    ;
0470                    ;
0470                    ;     40 CHA. IN 1 LINE CODE (DATA)
0470                    ;
0470                    LLPT:   ENT
0470 01                         DEFB   01H              ; TEXT MODE
0471 09                         DEFB   09H
0472 09                         DEFB   09H
0473 0B                         DEFB   0BH
0474 0D                         DEFB   0DH
0475                    ;
0475                    ;ORG 0475H
0475                    ;
0475                    ;
0475                    ;   WRITE DATA
0475                    ;
0475                    ;     EXIT CF=0 : OK
0475                    ;            =1 : BREAK
0475                    ;
0475                    ?WRD:   ENT
0475 F3                         DI
0476 D5                         PUSH   DE
```

```
0477 C5                     PUSH   BC
0478 E5                     PUSH   HL
0479 16D7                   LD     D,D7H         ; 'W'
047B 1E53                   LD     E,53H         ; 'S'
047D ED4B0211               LD     BC,(SIZE)     ; WRITE DATA BYTE SIZE
0481 2A0411                 LD     HL,(DTADR)    ; WRITE DATA ADDRESS
0484 78                     LD     A,B
0485 B1                     OR     C
0486 284A                   JR     Z,RET1
0488 18BA                   JR     WRI1
048A                ;
048A                ;
048A                ;   TAPE WRITE
048A                ;
048A                ;     BC=BYTE SIZE
048A                ;     HL=DATA LOW ADR.
048A                ;
048A                ;    EXIT  CF=0 : OK
048A                ;            =1 : BREAK
048A                ;
048A D5             WTAPE:  PUSH   DE
048B C5                     PUSH   BC
048C E5                     PUSH   HL
048D 1602                   LD     D,2
048F 3EF8                   LD     A,F8H
0491 3200E0                 LD     (KEYPA),A     ; E000H
0494 7E             WTAP1:  LD     A,(HL)
0495 CD6707                 CALL   WBYTE         ; 1 BYTE WRITE
0498 3A01E0                 LD     A,(KEYPB)     ; E001H
049B E681                   AND    81H           ; SHIFT & BREAK
049D C2A504                 JP     NZ,WTAP2
04A0 3E02                   LD     A,02H         ; BREAK IN CODE
04A2 37                     SCF
04A3 182D                   JR     WTAP3
04A5 23             WTAP2:  INC    HL
04A6 0B                     DEC    BC
04A7 78                     LD     A,B
04A8 B1                     OR     C
04A9 C29404                 JP     NZ,WTAP1
04AC 2A9711                 LD     HL,(SUMDT)    ; SUM DATA SET
04AF 7C                     LD     A,H
04B0 CD6707                 CALL   WBYTE
04B3 7D                     LD     A,L
04B4 CD6707                 CALL   WBYTE
04B7 CD1A0A                 CALL   LONG
04BA 15                     DEC    D
04BB C2C204                 JP     NZ,+7
04BE B7                     OR     A
04BF C3D204                 JP     WTAP3
04C2 0600                   LD     B,0
04C4 CD010A                 CALL   SHORT
04C7 05                     DEC    B
04C8 C2C404                 JP     NZ,-4
04CB E1                     POP    HL
04CC C1                     POP    BC
04CD C5                     PUSH   BC
04CE E5                     PUSH   HL
04CF C39404                 JP     WTAP1
04D2                WTAP3:
04D2 E1             RET1:   POP    HL
```

```
04D3 C1                     POP    BC
04D4 D1                     POP    DE
04D5 C9                     RET
04D6                ;
04D6                ;
04D6                ;
04D6                ;
04D6                ;
04D8                        ORG    04D8H
04D8                ;
04D8                ;
04D8                ;   READ INFORMATION (FROM $CMT)
04D8                ;
04D8                ;    EXIT ACC=0 : OK CF=0
04D8                ;            =1 : ER CF=1
04D8                ;            =2 : BREAK CF=1
04D8                ;
04D8                ?RDI:   ENT
04D8 F3                     DI
04D9 D5                     PUSH   DE
04DA C5                     PUSH   BC
04DB E5                     PUSH   HL
04DC 16D2                   LD     D,D2H         ; 'R'
04DE 1ECC                   LD     E,CCH         ; 'L'
04E0 018000                 LD     BC,80H
04E3 21F010                 LD     HL,IBUFE
04E6                RD1:    ENT
04E6 CD9F06                 CALL   MOTOR
04E9 DA7205                 JP     C,RTP6
04EC CD5B06                 CALL   TMARK
04EF DA7205                 JP     C,RTP6
04F2 CD0E05                 CALL   RTAPE
04F5 C35405                 JP     RTP4
04F8                ;
04F8                ;
04F8                ;
04F8                ;ORG 04F8H
04F8                ;
04F8                ;
04F8                ;   READ DATA (FROM $CMT)
04F8                ;
04F8                ;      EXIT SAME UP
04F8                ;
04F8                ?RDD:   ENT
04F8 F3                     DI
04F9 D5                     PUSH   DE
04FA C5                     PUSH   BC
04FB E5                     PUSH   HL
04FC 16D2                   LD     D,D2H         ; 'R'
04FE 1E53                   LD     E,53H         ; 'S'
0500 ED4B0211               LD     BC,(SIZE)
0504 2A0411                 LD     HL,(DTADR)
0507 78                     LD     A,B
0508 B1                     OR     C
0509 CA5405                 JP     Z,RTP4
050C 18D8                   JR     RD1
050E                ;
050E                ;
050E                ;   READ TAPE
050E                ;
```

```
050E                   ;     IN  BC=SIZE
050E                   ;           DE=LOAD ADR.
050E                   ;
050E                   ;     EXIT  ACC=0 : OK CF=0
050E                   ;              =1 : ER    =1
050E                   ;              =2 : BREAK=1
050E                   ;
050E                   RTAPE:  ENT
050E D5                        PUSH    DE
050F C5                        PUSH    BC
0510 E5                        PUSH    HL
0511 2602                      LD      H,2             ; TWICE WRITE
0513                   RTP1:   ENT
0513 0101E0                    LD      BC,KEYPB
0516 1102E0                    LD      DE,CSTR
0519                   RTP2:   ENT
0519 CD0106                    CALL    EDGE            ; 10 EDGE DETECT
051C 3854                      JR      C,RTP6
051E CD4A0A                    CALL    DLY3            ; CALL DLY2*3
0521 1A                        LD      A,(DE)          ; DATA (1BIT) READ
0522 E620                      AND     20H
0524 CA1905                    JP      Z,RTP2
0527 54                        LD      D,H
0528 210000                    LD      HL,0
052B 229711                    LD      (SUMDT),HL
052E E1                        POP     HL              ;
052F C1                        POP     BC
0530 C5                        PUSH    BC
0531 E5                        PUSH    HL
0532                   RTP3:   ENT
0532 CD2406                    CALL    RBYTE           ; 1BYTE READ
0535 383B                      JR      C,RTP6
0537 77                        LD      (HL),A
0538 23                        INC     HL
0539 0B                        DEC     BC
053A 78                        LD      A,B
053B B1                        OR      C
053C 20F4                      JR      NZ,RTP3
053E 2A9711                    LD      HL,(SUMDT)      ; CHECK SUM
0541 CD2406                    CALL    RBYTE           ; CHECK SUM DATA
0544 382C                      JR      C,RTP6
0546 5F                        LD      E,A
0547 CD2406                    CALL    RBYTE           ; CHECK SUM DATA
054A 3826                      JR      C,RTP6
054C BD                        CP      L
054D 2016                      JR      NZ,RTP5
054F 7B                        LD      A,E
0550 BC                        CP      H
0551 2012                      JR      NZ,RTP5
0553                   RTP8:   ENT
0553 AF                        XOR     A
0554                   RTP4:   ENT
0554                   RET2:   ENT
0554 E1                        POP     HL
0555 C1                        POP     BC
0556 D1                        POP     DE
0557 CD0007                    CALL    MSTOP
055A F5                        PUSH    AF
055B 3A9C11                    LD      A,(TIMFG)       ; INT. CHECK
055E FEF0                      CP      F0H
```

```
0560 2001                      JR      NZ,+3
0562 FB                        EI
0563 F1                        POP     AF
0564 C9                        RET
0565                   ;
0565                   RTP5:   ENT
0565 15                        DEC     D
0566 2806                      JR      Z,RTP7
0568 62                        LD      H,D
0569 CDE20F                    CALL    GAPCK
056C 18A5                      JR      RTP1
056E                   RTP7:   ENT
056E 3E01                      LD      A,1
0570 1802                      JR      RTP9
0572                   RTP6:   ENT
0572 3E02                      LD      A,2
0574                   RTP9:   ENT
0574 37                        SCF
0575 18DD                      JR      RTP4
0577                   ;
0577                   ;
0577                   ;    BELL
0577                   ;
0577                   ?BEL:   ENT
0577 D5                        PUSH    DE
0578 115203                    LD      DE,?BELD
057B F7                        RST     6               ; CALL MELDY
057C D1                        POP     DE
057D C9                        RET
057E                   ;
057E                   ;     FLASING AND KEYIN
057E                   ;        EXIT:ACC INPUT KEY DATA(DSP.CODE)
057E                   ;           H=F0H THEN NO KEYIN(Z FLG.)
057E                   ;
057E                   FLKEY:  ENT
057E CDFF09                    CALL    ?FLAS
0581 CDCA08                    CALL    ?KEY
0584 FEF0                      CP      F0H
0586 C9                        RET
0587                   ;
0587                   ;
0587                   ;
0587                   ;
0587                   ;
0587                           DEFS    +1
0588                   ;ORG 0588H
0588                   ;
0588                   ;
0588                   ;    VERIFY (FROM $CMT)
0588                   ;
0588                   ;     EXIT ACC =0 : OK CF=0
0588                   ;              =1 : ER CF=1
0588                   ;              =2 : BREAK CF=1
0588                   ;
0588                   ?VRFY:  ENT
0588 F3                        DI
0589 D5                        PUSH    DE
058A C5                        PUSH    BC
058B E5                        PUSH    HL
058C ED4B0211                  LD      BC,(SIZE)
```

```
0590 2A0411             LD      HL,(DTADR)
0593 16D2               LD      D,D2H            ; 'R'
0595 1E53               LD      E,53H            ; 'S'
0597 78                 LD      A,B
0598 B1                 OR      C
0599 28B9               JR      Z,RTP4
059B CD1A07             CALL    CKSUM
059E CD9F06             CALL    MOTOR
05A1 38CF               JR      C,RTP6
05A3 CD5B06             CALL    TMARK            ; TAPE MARK DETECT
05A6 38CA               JR      C,RTP6
05A8 CDAD05             CALL    TVRFY
05AB 18A7               JR      RTP4
05AD          ;
05AD          ;
05AD          ;   DATA VERIFY
05AD          ;
05AD          ;     BC=SIZE
05AD          ;     HL=DATA LOW ADR
05AD          ;     CSMDT=CHECK SUM
05AD          ;   EXIT ACC=0 : OK CF=0
05AD          ;           =1 : ER    =1
05AD          ;           =2 : BREAK=1
05AD          ;
05AD          ;
05AD          TVRFY:    ENT
05AD D5                 PUSH    DE
05AE C5                 PUSH    BC
05AF E5                 PUSH    HL
05B0 2602               LD      H,2
05B2          TVF1:     ENT
05B2 0101E0             LD      BC,KEYPB
05B5 1102E0             LD      DE,CSTR
05B8          TVF2:     ENT
05B8 CD0106             CALL    EDGE
05BB DA7205             JP      C,RTP6
05BE CD4A0A             CALL    DLY3             ; CALL DLY2*3
05C1 1A                 LD      A,(DE)
05C2 E620               AND     20H
05C4 CAB805             JP      Z,TVF2
05C7 54                 LD      D,H
05C8 E1                 POP     HL
05C9 C1                 POP     BC
05CA C5                 PUSH    BC
05CB E5                 PUSH    HL
05CC          TVF3:     ENT
05CC CD2406             CALL    RBYTE
05CF 38A1               JR      C,RTP6
05D1 BE                 CP      (HL)
05D2 209A               JR      NZ,RTP7
05D4 23                 INC     HL
05D5 0B                 DEC     BC
05D6 78                 LD      A,B
05D7 B1                 OR      C
05D8 20F2               JR      NZ,TVF3
05DA 2A9911             LD      HL,(CSMDT)
05DD CD2406             CALL    RBYTE
05E0 BC                 CP      H
05E1 208B               JR      NZ,RTP7
05E3 CD2406             CALL    RBYTE
```

```
05E6 BD                 CP      L
05E7 2085               JR      NZ,RTP7
05E9 15                 DEC     D
05EA CA5305             JP      Z,RTP8
05ED 62                 LD      H,D
05EE 18C2               JR      TVF1
05F0          ;
05F0          ;   FLASHING DATA LOAD
05F0          ;
05F0          ?LOAD:    ENT
05F0 F5                 PUSH    AF
05F1 3A8E11             LD      A,(FLASH)
05F4 CDB10F             CALL    ?PONT
05F7 77                 LD      (HL),A
05F8 F1                 POP     AF
05F9 C9                 RET
05FA          ;
05FA          ;
05FA          ;   NEW LINE AND PRINT HL REG.(ASCII)
05FA          ;
05FA          NLPHL:    ENT
05FA CD0900             CALL    NL
05FD CDBA03             CALL    PRTHL
0600 C9                 RET
0601          ;
0601          ;
0601          ;ORG 0601H;EDGE
0601          ;
0601          ;
0601          ;   EDGE       (TAPE DATA EDGE DETECT)
0601          ;
0601          ;   BC=KEYPB  ($E001)
0601          ;   DE=CSTR   ($E002)
0601          ;   EXIT CF=0 OK :     CF=1 BREAK
0601          ;
0601          EDGE:     ENT
0601 3EF8               LD      A,F8H            ; BREAK KEY IN
0603 3200E0             LD      (KEYPA),A
0606 00                 NOP
0607          EDG1:     ENT
0607 0A                 LD      A,(BC)
0608 E681               AND     81H              ; SHIFT & BREAK
060A 2002               JR      NZ,+4
060C 37                 SCF
060D C9                 RET
060E 1A                 LD      A,(DE)
060F E620               AND     20H
0611 20F4               JR      NZ,EDG1          ; CSTR D5 = 0
0613          EDG2:     ENT
0613 0A                 LD      A,(BC)           ; 8
0614 E681               AND     81H              ; 9
0616 2002               JR      NZ,+4            ; 10/14
0618 37                 SCF
0619 C9                 RET
061A 1A                 LD      A,(DE)           ; 8
061B E620               AND     20H              ; 9
061D 28F4               JR      Z,EDG2           ; CSTR D5 = 1   :10/14
061F C9                 RET                      ; 11
0620          ;
0620          ;
```

```
0620                    DEFS   +4
0624          ;ORG 0624H;RBYTE
0624          ;
0624          ;
0624          ;   1 BYTE READ
0624          ;
0624          ;     EXIT  SUMDT=STORE
0624          ;      CF=1 : BREAK
0624          ;
0624          ;       CF=0 : DATA=ACC
0624          ;
0624          RBYTE:    ENT
0624 C5                 PUSH   BC
0625 D5                 PUSH   DE
0626 E5                 PUSH   HL
0627 210008             LD     HL,0800H
062A 0101E0             LD     BC,KEYPB         ; KEY DATA $E001
062D 1102E0             LD     DE,CSTR          ; $TAPE DATA $E002
0630          RBY1:     ENT
0630 CD0106             CALL   EDGE             ; 41 OR 101
0633 DA5406             JP     C,RBY3           ; 13
0636 CD4A0A             CALL   DLY3             ; 20+18*63+33
0639 1A                 LD     A,(DE)           ; DATA READ :8
063A E620               AND    20H
063C CA4906             JP     Z,RBY2
063F E5                 PUSH   HL
0640 2A9711             LD     HL,(SUMDT)
0643 23                 INC    HL
0644 229711             LD     (SUMDT),HL
0647 E1                 POP    HL
0648 37                 SCF
0649          RBY2:     ENT
0649 7D                 LD     A,L
064A 17                 RLA
064B 6F                 LD     L,A
064C 25                 DEC    H
064D C23006             JP     NZ,RBY1
0650 CD0106             CALL   EDGE
0653 7D                 LD     A,L
0654          RBY3:     ENT
0654 E1                 POP    HL
0655 D1                 POP    DE
0656 C1                 POP    BC
0657 C9                 RET
0658          ;
0658          ;
0658          ;   TAPE MARK DETECT
0658          ;
0658          ;     E=@L@ :INFORMATION
0658          ;      =@S@ :DATA
0658          ;    EXIT CF=0 ;OK
0658          ;           =1 :BREAK
0658          ;
0658                    DEFS   +3
065B          ;
065B          TMARK:    ENT
065B          ;
065B          ;ORG 065BH
065B          ;
065B CDE20F             CALL   GAPCK
```

```
065E C5                 PUSH   BC               ; ORG 065EH
065F D5                 PUSH   DE
0660 E5                 PUSH   HL
0661 212828             LD     HL,2828H
0664 7B                 LD     A,E
0665 FECC               CP     CCH              ;'L'
0667 2803               JR     Z,+5
0669 211414             LD     HL,1414H
066C 229511             LD     (TMCNT),HL
066F 0101E0             LD     BC,KEYPB
0672 1102E0             LD     DE,CSTR
0675          TM1:      ENT
0675 2A9511             LD     HL,(TMCNT)
0678          TM2:      ENT
0678 CD0106             CALL   EDGE
067B 381E               JR     C,TM4
067D CD4A0A             CALL   DLY3             ; CALL DLY2*3
0680 1A                 LD     A,(DE)
0681 E620               AND    20H
0683 28F0               JR     Z,TM1
0685 25                 DEC    H
0686 20F0               JR     NZ,TM2
0688          TM3:      ENT
0688 CD0106             CALL   EDGE
068B 380E               JR     C,TM4
068D CD4A0A             CALL   DLY3             ; CALL DLY2*3
0690 1A                 LD     A,(DE)
0691 E620               AND    20H
0693 20E0               JR     NZ,TM1
0695 2D                 DEC    L
0696 20F0               JR     NZ,TM3
0698 CD0106             CALL   EDGE
069B          RET3:     ENT
069B          TM4:      ENT
069B E1                 POP    HL
069C D1                 POP    DE
069D C1                 POP    BC
069E C9                 RET
069F          ;
069F          ;
069F          ;   MOTOR ON
069F          ;
069F          ;    IN  D=@W@ :WRITE
069F          ;          =@R@ :READ
069F          ;   EXIT CF=0 :OK
069F          ;           =1 :BREAK
069F          MOTOR:    ENT
069F C5                 PUSH   BC
06A0 D5                 PUSH   DE
06A1 E5                 PUSH   HL
06A2 060A               LD     B,10
06A4          MOT1:     ENT
06A4 3A02E0             LD     A,(CSTR)
06A7 E610               AND    10H
06A9 280E               JR     Z,MOT4
06AB          MOT2:     ENT
06AB 06FF               LD     B,FFH            ; 2 SEC DELAY
06AD CD9609             CALL   DLY12            ; 7 MSEC DELAY
06B0 1802               JR     +4               ;MOTOR ENTRY ADJUST
06B2 18EB               JR     MOTOR            ; ORG 06B2H
```

```
06B4 10F7                DJNZ   -7
06B6 AF                  XOR    A
06B7            MOT7:    ENT
06B7 18E2                JR     RET3
06B9            MOT4:    ENT
06B9 3E06                LD     A,06H
06BB 2103E0              LD     HL,CSTPT
06BE 77                  LD     (HL),A
06BF 3C                  INC    A
06C0 77                  LD     (HL),A
06C1 10E1                DJNZ   MOT1
06C3 CD0900              CALL   NL
06C6 7A                  LD     A,D
06C7 FED7                CP     D7H          ; 'W'
06C9 2805                JR     Z,MOT8
06CB 11FB03              LD     DE,MSG#1     ; PLAY MARK
06CE 1807                JR     MOT9
06D0            MOT8:    ENT
06D0 110204              LD     DE,MSG#3     ; "RECORD."
06D3 DF                  RST    3            ; CALL MSGX
06D4 11FD03              LD     DE,MSG#2     ; "PLAY"
06D7            MOT9:    ENT
06D7 DF                  RST    3            ; CALL MSGX
06D8            MOT5:    ENT
06D8 3A02E0              LD     A,(CSTR)
06DB E610                AND    10H
06DD 20CC                JR     NZ,MOT2
06DF CD320A              CALL   ?BRK
06E2 20F4                JR     NZ,MOT5
06E4 37                  SCF
06E5 18D0                JR     MOT7
06E7            ;
06E7            ;    INITIAL MESSAGE
06E7            ;
06E7            MSG?3:   ENT
06E7 2A2A2020            DEFM   '**  MONITOR 1Z-013B  **'
06EB 4D4F4E49
06EF 544F5220
06F3 315A2D30
06F7 31334220
06FB 202A2A
06FE 0D                  DEFB   0DH
06FF            ;
06FF            ;
06FF                     DEFS   +1
0700            ;
0700            ;
0700            ;ORG 0700H;MSTOP
0700            ;
0700            ;
0700            ;  MOTOR STOP
0700            ;
0700            MSTOP:   ENT
0700 F5                  PUSH   AF
0701 C5                  PUSH   BC
0702 D5                  PUSH   DE
0703 060A                LD     B,10
0705            MST1:    ENT
0705 3A02E0              LD     A,(CSTR)
0708 E610                AND    10H
```

```
070A 280B                JR     Z,MST3
070C            MST2:    ENT
070C 3E06                LD     A,06H
070E 3203E0              LD     (CSTPT),A
0711 3C                  INC    A
0712 3203E0              LD     (CSTPT),A
0715 10EE                DJNZ   MST1
0717            MST3:    ENT
0717 C3E60E              JP     ?RSTR1
071A            ;
071A            ;
071A            ;
071A            ;
071A            ;    CHECK SUM
071A            ;
071A            ;      IN    BC=SIZE
071A            ;            HL=DATA ADR.
071A            ;     EXIT  SUMDT=STORE
071A            ;            CSMDT=STORE
071A            ;
071A            CKSUM:   ENT
071A C5                  PUSH   BC
071B D5                  PUSH   DE
071C E5                  PUSH   HL
071D 110000              LD     DE,0
0720            CKS1:    ENT
0720 78                  LD     A,B
0721 B1                  OR     C
0722 200B                JR     NZ,CKS2
0724 EB                  EX     DE,HL
0725 229711              LD     (SUMDT),HL
0728 229911              LD     (CSMDT),HL
072B E1                  POP    HL
072C D1                  POP    DE
072D C1                  POP    BC
072E C9                  RET
072F            CKS2:    ENT
072F 7E                  LD     A,(HL)
0730 C5                  PUSH   BC
0731 0608                LD     B,+8
0733            CKS3:    ENT
0733 07                  RLCA
0734 3001                JR     NC,+3
0736 13                  INC    DE
0737 10FA                DJNZ   CKS3
0739 C1                  POP    BC
073A 23                  INC    HL
073B 0B                  DEC    BC
073C 18E2                JR     CKS1
073E            ;
073E            ;    MODE SET OF KEYPORT
073E            ;
073E            ?MODE:   ENT
073E 2103E0              LD     HL,KEYPF
0741 368A                LD     (HL),8AH     ; 10001010
0743 3607                LD     (HL),07H     ; PC3=1
0745 3605                LD     (HL),05H     ; PC2=1
0747            VGOFF:   ENT
0747            ;
0747 C9                  RET
```

```
0748                 ;
0748                 ;
0748                         DEFS    +17
0759                 ;
0759                 ;ORG 0759H;DLY1
0759                 ;
0759                 ;   107 MICRO SEC DELY
0759                 ;
0759                 DLY1:   ENT
0759 3E1B                    LD      A,1BH          ; 111.39 micro sec
075B 3D                      DEC     A
075C C25B07                  JP      NZ,-1
075F C9                      RET
0760                 ;
0760                 ;ORG 0760H;DLY2
0760                 ;
0760                 DLY2:   ENT
0760 3E19                    LD      A,19H          ; 103.45 micro sec
0762 3D                      DEC     A
0763 C26207                  JP      NZ,-1
0766 C9                      RET
0767                 ;
0767                 ;
0767                 ;
0767                 ;
0767                 ;
0767                 ;  1 BYTE WRITE
0767                 ;
0767                 WBYTE:  ENT
0767 C5                      PUSH    BC
0768 0608                    LD      B,+8
076A CD1A0A                  CALL    LONG
076D                 WBY1:   ENT
076D 07                      RLCA
076E DC1A0A                  CALL    C,LONG
0771 D4010A                  CALL    NC,SHORT
0774 05                      DEC     B
0775 C26D07                  JP      NZ,WBY1
0778 C1                      POP     BC
0779 C9                      RET
077A                 ;
077A                 ;
077A                 ;   GAP + TAPEMARK
077A                 ;
077A                 ;    E=@L@ LONG GAP
077A                 ;     =@S@ SHORT GAP
077A                 ;
077A                 GAP:    ENT
077A C5                      PUSH    BC
077B D5                      PUSH    DE
077C 7B                      LD      A,E
077D 01F055                  LD      BC,55F0H
0780 112828                  LD      DE,2828H
0783 FECC                    CP      CCH            ;'L'
0785 CA8E07                  JP      Z,GAP1
0788 01F82A                  LD      BC,2AF8H
078B 111414                  LD      DE,1414H
078E                 GAP1:   ENT
078E CD010A                  CALL    SHORT
0791 0B                      DEC     BC
```

```
0792 78                      LD      A,B
0793 B1                      OR      C
0794 20F8                    JR      NZ,-6
0796                 GAP2:   ENT
0796 CD1A0A                  CALL    LONG
0799 15                      DEC     D
079A 20FA                    JR      NZ,-4
079C                 GAP3:   ENT
079C CD010A                  CALL    SHORT
079F 1D                      DEC     E
07A0 20FA                    JR      NZ,-4
07A2 CD1A0A                  CALL    LONG
07A5 D1                      POP     DE
07A6 C1                      POP     BC
07A7 C9                      RET
07A8                 ;
07A8                 ;    MEMORY CORRECTION
07A8                 ;        COMMAND 'M'
07A8                 ;
07A8                 MCOR:   ENT
07A8 CD3D01                  CALL    HEXIY           ; CRRECTION ADR.
07AB                 MCR1:   ENT
07AB CDFA05                  CALL    NLPHL           ; COR. ADR. PRINT
07AE CDB103                  CALL    SPHEX           ; ACC  ASCII DISP.
07B1 CD2009                  CALL    ?PRTS           ; SPACE PRINT
07B4 CD2F01                  CALL    BGETL           ; GET DATA & CHECK DATA
07B7 CD1004                  CALL    HLHEX           ; HLASCII(DE)
07BA 381B                    JR      C,MCR3
07BC CDA602                  CALL    .4DE            ; (INC DE)*4
07BF 13                      INC     DE
07C0 CD1F04                  CALL    2HEX            ; DATA CHECK
07C3 38E6                    JR      C,MCR1
07C5 BE                      CP      (HL)
07C6 20E3                    JR      NZ,MCR1
07C8 13                      INC     DE
07C9 1A                      LD      A,(DE)
07CA FE0D                    CP      0DH             ; NOT CORRECTION ?
07CC 2806                    JR      Z,MCR2
07CE CD1F04                  CALL    2HEX            ; ACCHL(ASCII)
07D1 38D8                    JR      C,MCR1
07D3 77                      LD      (HL),A          ; DATA CORRECT
07D4                 MCR2:   ENT
07D4 23                      INC     HL
07D5 18D4                    JR      MCR1
07D7                 ;
07D7 60              MCR3:   LD      H,B             ; MEMORY ADR.
07D8 69                      LD      L,C
07D9 18D0                    JR      MCR1
07DB                 ;
07DB                 ;
07DB                 ;
07DB                 ;
07DB                 ;
07E6                         ORG     07E6H
07E6                 ;
07E6                 ;
07E6                 ;
07E6                 ;   GET 1 LINE STATEMENT   *
07E6                 ;
07E6                 ;      DE = DATA STORE LOW ADR.
```

```
07E6             ;           (END =CR )
07E6             ;
07E6             ;
07E6             ?GETL:  ENT
07E6 F5                  PUSH  AF
07E7 C5                  PUSH  BC
07E8 E5                  PUSH  HL
07E9 D5                  PUSH  DE
07EA             GETL1:  ENT
07EA CDB309              CALL  ??KEY          ; ENTRY KEY
07ED             AUTO3:  ENT
07ED F5                  PUSH  AF             ; IN KEY DATA SAVE
07EE 47                  LD    B,A
07EF 3A9D11              LD    A,(SWRK)       ; BELL WORK
07F2 0F                  RRCA
07F3 D47705              CALL  NC,?BEL        ; ENTRY BELL
07F6 78                  LD    A,B
07F7 217011              LD    HL,KANAF       ; KANA & GRAPH FLAG
07FA E6F0                AND   F0H
07FC FEC0                CP    C0H
07FE D1                  POP   DE             ; Ereg=FLAGreg
07FF 78                  LD    A,B
0800 2016                JR    NZ,GETL2
0802 FECD                CP    CDH            ; CR
0804 2855                JR    Z,GETL3
0806 FECB                CP    CBH            ; BREAK
0808 CA2208              JP    Z,GETLC
080B FECF                CP    CFH            ; NIKO MARK WH.
080D 2809                JR    Z,GETL2
080F FEC7                CP    C7H            ; CRT EDITION
0811 300A                JR    NC,GETL5
0813 CB1B                RR    E              ; CY ?
0815 78                  LD    A,B
0816 3005                JR    NC,GETL5
0818             GETL2:  ENT
0818 CDB50D              CALL  ?DSP
081B 18CD                JR    GETL1
081D             GETL5:  ENT
081D CDDC0D              CALL  ?DPCT          ; CRT CONTROL
0820 18C8                JR    GETL1
0822             ;
0822             ;    BREAK IN
0822             ;
0822 E1          GETLC:  POP   HL
0823 E5                  PUSH  HL
0824 361B                LD    (HL),1BH       ; BREAK CODE
0826 23                  INC   HL
0827 360D                LD    (HL),0DH
0829 1853                JR    GETLR
082B             ;  GETLA
082B             ;
082B 0F          GETLA:  RRCA                 ; CYD7
082C 3037                JR    NC,GETL6
082E 1833                JR    GETLB
0830             ;
0830             ;
0830             ;
0830             ;   DELAY 7M SEC AND SWEP
0830             ;
0830 CD9609      DSWEP:  CALL  DLY12
```

```
0833 CD500A              CALL  ?SWEP
0836 C9                  RET
0837             ;
0837             ;
0837                     DEFS  36
085B             ;
085B             ;
085B             ;
085B             ; ORG 085BH;GETL3
085B             ;
085B CDF302      GETL3:  CALL  .MANG          ; CR
085E 0628                LD    B,40           ; 1LINE
0860 30C9                JR    NC,GETLA
0862 25                  DEC   H              ; BEFORE LINE
0863 0650        GETLB:  LD    B,80           ; 2 LINE
0865 2E00        GETL6:  LD    L,0
0867 CDB40F              CALL  ?PNT1
086A D1                  POP   DE             ; STORE TOP ADR.
086B D5                  PUSH  DE
086C 7E          GETLZ:  LD    A,(HL)
086D CDCE0B              CALL  ?DACN
0870 12                  LD    (DE),A
0871 23                  INC   HL
0872 13                  INC   DE
0873 10F7                DJNZ  GETLZ
0875 EB                  EX    DE,HL
0876 360D        GETLU:  LD    (HL),0DH
0878 2B                  DEC   HL
0879 7E                  LD    A,(HL)
087A FE20                CP    20H            ; SPACE THEN CR
087C             ;
087C             ;
087C             ;   CR AND NEW LINE
087C             ;
087C 28F8                JR    Z,GETLU
087E             ;
087E             ;   NEW LINE RETURN
087E             ;
087E CD0E09      GETLR:  CALL  ?LTNL
0881 D1                  POP   DE
0882 E1                  POP   HL
0883 C1                  POP   BC
0884 F1                  POP   AF
0885 C9                  RET
0886             ;
0886             ;
0886             ;
0886                     DEFS  +13
0893             ;ORG 0893H
0893             ;
0893             ;   MESSAGE PRINT
0893             ;
0893             ;     DE PRINT DATA LOW ADR.
0893             ;         END=CR
0893             ;
0893             ?MSG:   ENT
0893 F5                  PUSH  AF
0894 C5                  PUSH  BC
0895 D5                  PUSH  DE
0896 1A          MSG1:   LD    A,(DE)
```

```
0897 FE0D            CP    0DH             ; CR
0899 280C            JR    Z,MSGX2
089B CD3509          CALL  ?PRNT
089E 13              INC   DE
089F 18F5            JR    MSG1
08A1         ;
08A1         ;
08A1         ;ORG 08A1H
08A1         ;
08A1         ;  ALL PRINT MESSAGE
08A1         ;
08A1         ?MSGX:  ENT
08A1 F5              PUSH  AF
08A2 C5              PUSH  BC
08A3 D5              PUSH  DE
08A4 1A      MSGX1:  LD    A,(DE)
08A5 FE0D            CP    0DH
08A7 CAE60E  MSGX2:  JP    Z,?RSTR1
08AA CDB90B          CALL  ?ADCN
08AD CD6C09          CALL  PRNT3
08B0 13              INC   DE
08B1 18F1            JR    MSGX1
08B3         ;
08B3         ;   TOP OF KEYTBLS
08B3         ;
08B3 112A0C  ?KYSM:  LD    DE,KTBLS
08B6 1842            JR    ?KY5
08B8         ;
08B8         ;   BREAK CODE IN
08B8         ;
08B8 3ECB    #BRK:   LD    A,CBH           ; BREAK CODE
08BA B7              OR    A
08BB 1819            JR    ?KY1
08BD         ;
08BD         ;
08BD         ;ORG 08BDH
08BD         ;
08BD         ;   GETKEY
08BD         ;
08BD         ;       NOT ECHO BACK
08BD         ;
08BD         ;     EXIT:ACC=ASCII CODE
08BD         ;
08BD         ?GET:   ENT
08BD CDCA08          CALL  ?KEY            ; KEY IN (DISPLAY CODE)
08C0 D6F0            SUB   F0H             ; NOT KEYIN CODE
08C2 C8              RET   Z
08C3 C6F0            ADD   A,F0H
08C5 C3CE0B          JP    ?DACN           ; DIAPLAY TO ASCII CODE
08C8         ;
08C8         ;
08C8                 DEFS  +2
08CA         ;
08CA         ;
08CA         ;
08CA         ;
08CA         ;ORG 08CAH;?KEY
08CA         ;
08CA         ;   1KEY INPUT
08CA         ;  IN      B = KEY MODE(SHIFT,CTRL,BREAK)
```

```
08CA         ;          C = KEY DATA (COLUMN & ROW)
08CA         ;  EXIT    ACC=DISPLAY CODE
08CA         ;          IF NO KEY   ACC=F0H
08CA         ;        IF CY=1 THEN ATTRIBUTE ON
08CA         ;                     (SMALL,HIRAKANA)
08CA         ;
08CA         ?KEY:   ENT
08CA C5              PUSH  BC
08CB D5              PUSH  DE
08CC E5              PUSH  HL
08CD CD3008          CALL  DSWEP           ; DELAY AND KEY SWEP
08D0 78              LD    A,B
08D1 07              RLCA
08D2 3806            JR    C,?KY2
08D4 3EF0            LD    A,F0H
08D6         ?KY1:   ENT
08D6 E1              POP   HL
08D7 D1              POP   DE
08D8 C1              POP   BC
08D9 C9              RET
08DA         ;
08DA         ?KY2:   ENT
08DA 11EA0B          LD    DE,KTBL         ; NORMAL KEY TABLE
08DD 78              LD    A,B
08DE FE88            CP    88H             ; BREAK IN
08E0 28D6            JR    Z,#BRK
08E2 2600            LD    H,0             ; HL=ROW & COLUMN
08E4 69              LD    L,C
08E5 CB6F            BIT   5,A             ; CTRL CHECK
08E7 200E            JR    NZ,?KY5-3
08E9 3A7011          LD    A,(KANAF)       ; 0=NR.,1=GRAPH
08EC 0F              RRCA
08ED DAFE08          JP    C,?KYGRP        ; GRAPH MODE
08F0 78              LD    A,B             ; CTRL KEY CHECK
08F1 17              RLA
08F2 17              RLA
08F3 38BE            JR    C,?KYSM
08F5 1803            JR    ?KY5
08F7 11AA0C          LD    DE,KTBLC        ; CONTROL KEY TABLE
08FA         ?KY5:   ENT
08FA 19              ADD   HL,DE           ; TABLE
08FB         ?KY55:  ENT
08FB 7E              LD    A,(HL)
08FC 18D8            JR    ?KY1
08FE         ?KYGRP: ENT
08FE CB70            BIT   6,B
0900 2807            JR    Z,?KYGRS
0902 11E90C          LD    DE,KTBLG
0905 19              ADD   HL,DE
0906 37              SCF
0907 18F2            JR    ?KY55
0909         ;
0909 116A0C  ?KYGRS: LD    DE,KTBLGS
090C 18EC            JR    ?KY5
090E         ;
090E         ;
090E         ;
090E         ;
090E         ;
090E         ;ORG 090EH
```

```
0 90E                   ;
090E                    ;   NEWLINE
090E                    ;
090E                    ?LTNL:  ENT
090E AF                         XOR     A
090F 329411                     LD      (DPRNT),A       ; ROW POINTER
0912 3ECD                       LD      A,CDH           ; CR
0914 1843                       JR      PRNT5
0916                            DEFS    +2
0918                    ;ORG 0918H
0918                    ;
0918                    ?NL:    ENT
0918 3A9411                     LD      A,(DPRNT)
091B B7                         OR      A
091C C8                         RET     Z
091D 18EF                       JR      ?LTNL
091F                            DEFS    +1
0920                    ;ORG 0920H
0920                    ;
0920                    ;   PRINT SPACE
0920                    ;
0920                    ?PRTS:  ENT
0920 3E20                       LD      A,20H
0922 1811                       JR      ?PRNT
0924                    ;
0924                    ;   PRINT TAB
0924                    ;
0924                    ?PRTT:  ENT
0924 CD0C00                     CALL    PRNTS
0927 3A9411                     LD      A,(DPRNT)
092A B7                         OR      A
092B C8                         RET     Z
092C D60A                       SUB     +10
092E 38F4                       JR      C,-10
0930 20FA                       JR      NZ,-4
0932                            DEFS    +3
0935                    ;ORG 0935H
0935                    ;
0935                    ;   PRINT
0935                    ;
0935                    ;     IN ACC = PRINT DATA (ASCII)
0935                    ;
0935                    ?PRNT:  ENT
0935 FE0D                       CP      0DH             ; CR
0937 28D5                       JR      Z,?LTNL
0939 C5                         PUSH    BC
093A 4F                         LD      C,A
093B 47                         LD      B,A
093C CD4609                     CALL    ?PRT
093F 78                         LD      A,B
0940 C1                         POP     BC
0941 C9                         RET
0942                    ;
0942                    ;
0942                    MSGOK:  ENT
0942 4F4B21                     DEFM    'OK!'
0945 0D                         DEFB    0DH
0946                    ;ORG 0946H
0946                    ;
0946                    ;  PRINT ROUTINE
```

```
0946                    ;       1 CHA.
0946                    ;   INPUT:C=ASCII DATA (?DSP+?DPCT)
0946                    ;
0946                    ?PRT:   ENT
0946 79                         LD      A,C
0947 CDB90B                     CALL    ?ADCN           ; ASCII TO DSPLAY
094A 4F                         LD      C,A
094B FEF0                       CP      F0H
094D C8                         RET     Z               ; ZERO=ILLEGAL DATA
094E E6F0                       AND     F0H             ; MSD CHECK
0950 FEC0                       CP      C0H
0952 79                         LD      A,C
0953 2017                       JR      NZ,PRNT3
0955 FEC7                       CP      C7H
0957 3013                       JR      NC,PRNT3        ; CRT EDITOR
0959                    PRNT5:  ENT
0959 CDDC0D                     CALL    ?DPCT
095C FEC3                       CP      C3H
095E 280F                       JR      Z,PRNT4
0960 FEC5                       CP      C5H             ; HOME
0962 2803                       JR      Z,PRNT2
0964 FEC6                       CP      C6H             ; CLR
0966 C0                         RET     NZ
0967 AF                 PRNT2:  XOR     A
0968 329411                     LD      (DPRNT),A
096B C9                         RET
096C                    PRNT3:  ENT
096C CDB50D                     CALL    ?DSP
096F 3A9411             PRNT4:  LD      A,(DPRNT)       ; TAB POINT+1
0972 3C                         INC     A
0973 FE50                       CP      +80
0975 38F1                       JR      C,PRNT2+1
0977 D650                       SUB     +80
0979 18ED                       JR      PRNT2+1
097B                    ;
097B                    ;
097B                    ;
097B                    ;
097B                    ;
097B                    ;  FLASSING BYPASS 1
097B                    ;
097B                    FLAS1:  ENT
097B 3A8E11                     LD      A,(FLASH)
097E 186F                       JR      FLAS2
0980                    ;
0980                    ;  BREAK SUBROUTINE BYPASS 1
0980                    ;
0980                    ;       CTRL OR NOT KEY
0980                    ;
0980                    ?BRK2:  ENT
0980 CB6F                       BIT     5,A             ; NOT OR CTRL
0982 2802                       JR      Z,?BRK3         ; CTRL
0984 B7                         OR      A               ; NOTKEY A=7FH
0985 C9                         RET
0986                    ;
0986 3E20               ?BRK3:  LD      A,20H           ; CTRL D5=1
0988 B7                         OR      A               ; ZERO FLG. CLR
0989 37                         SCF
098A C9                         RET
098B                    ;
```

```
098B                 MSGSV:  ENT
098B 46494C45                DEFM   'FILENAME? '
098F 4E414D45
0993 3F20
0995 0D                      DEFB   0DH
0996                 ;
0996                 ;   DLY 7  MSEC
0996                 ;
0996                 DLY12:  ENT
0996 C5                      PUSH   BC
0997 0615                    LD     B,15H          ; 7143.9 micro sec
0999 CD4A0A                  CALL   DLY3
099C 10FB                    DJNZ   -3
099E C1                      POP    BC
099F C9                      RET
09A0                 ;
09A0                 ;
09A0                 ;
09A0                 ;   LOADING MESSAGE
09A0                 ;
09A0                 MSG?2:  ENT
09A0 4C4F4144                DEFM   'LOADING '
09A4 494E4720
09A8 0D                      DEFB   0DH
09A9                 ;
09A9                 ;
09A9                 ;
09A9                 ;   DELAY FOR LONG PULSE
09A9                 ;
09A9                 DLY4:   ENT
09A9 3E73                    LD     A,73H          ; 458.81 micro sec
09AB 3D                      DEC    A
09AC C2AB09                  JP     NZ,-1
09AF C9                      RET
09B0                 ;
09B0                 ;
09B0                         DEFS   +3
09B3                 ;
09B3                 ;
09B3                 ;ORG 09B3H;??KEY
09B3                 ;
09B3                 ;   KEY BOAD SEARCH
09B3                 ;     & DISPLAY CODE CONV.
09B3                 ;
09B3                 ;     EXIT A = DISPLAY CODE
09B3                 ;          CY= GRAPH MODE
09B3                 ;     WITH CURSOR DISPLAY
09B3                 ;
09B3                 ??KEY:  ENT
09B3 E5                      PUSH   HL
09B4 CD920B                  CALL   ?SAVE
09B7                 KSL1:   ENT
09B7 CD7E05                  CALL   FLKEY          ; KEY
09BA 20FB                    JR     NZ,KSL1        ; KEY IN THEN JUMP
09BC                 KSL2:   ENT
09BC CD7E05                  CALL   FLKEY
09BF 28FB                    JR     Z,KSL2         ; NOT KEY IN THEN JUMP
09C1 67                      LD     H,A
09C2 CD9609                  CALL   DLY12          ; DELAY CHATTER
09C5 CDCA08                  CALL   ?KEY
```

```
09C8 F5                      PUSH   AF
09C9 BC                      CP     H              ; CHATER CHECK
09CA E1                      POP    HL
09CB 20EF                    JR     NZ,KSL2
09CD E5                      PUSH   HL
09CE F1                      POP    AF             ; IN KEY DATA
09CF CDF005                  CALL   ?LOAD          ; FLSHING DATA LOAD
09D2 E1                      POP    HL
09D3 C9                      RET
09D4                 ;
09D4                 ;
09D4                 ;   CLEAR 2
09D4                 ;
09D4                 #CLR08: ENT                   ; CY FLG.
09D4 AF                      XOR    A
09D5                 #CLR8:  ENT
09D5 010008                  LD     BC,0800H
09D8                 CLEAR:  ENT                   ; BC = CLR BYTE SIZE
09D8 D5                      PUSH   DE             ; A = CLR DATA
09D9 57                      LD     D,A
09DA                 CLEAR1: ENT
09DA 72                      LD     (HL),D
09DB 23                      INC    HL
09DC 0B                      DEC    BC
09DD 78                      LD     A,B
09DE B1                      OR     C
09DF 20F9                    JR     NZ,CLEAR1
09E1 D1                      POP    DE
09E2 C9                      RET
09E3                 ;
09E3                 ;
09E3                 ;
09E3                 ;
09E3                 ;   FLASHING 2
09E3                 ;
09E3                 ?FLS:   ENT
09E3 F5                      PUSH   AF
09E4 E5                      PUSH   HL
09E5 3A02E0                  LD     A,(KEYPC)
09E8 07                      RLCA
09E9 07                      RLCA
09EA 388F                    JR     C,FLAS1
09EC 3A9211                  LD     A,(FLSDT)
09EF                 FLAS2:  ENT
09EF CDB10F                  CALL   ?PONT          ; DISPLAY POSITION
09F2 77                      LD     (HL),A
09F3                 FLAS3:  ENT
09F3 E1                      POP    HL
09F4 F1                      POP    AF
09F5 C9                      RET
09F6                 ;
09F6                 ;
09F6                 ;
09F6                         DEFS   +9
09FF                 ;
09FF                 ;
09FF                 ;ORG 09FF ; ?FLAS
09FF                 ;
09FF                 ?FLAS:  ENT
09FF 18E2                    JR     ?FLS
```

```
0A01                    ;
0A01                    ;
0A01                    ;
0A01                    ;  SHORT AND LONG PULSE FOR 1 BIT WRITE
0A01                    ;
0A01                    SHORT:  ENT
0A01 F5                         PUSH   AF              ; 12
0A02 3E03                       LD     A,03H           ; 9
0A04 3203E0                     LD     (CSTPT),A       ; $E003 PC3=1:16
0A07 CD5907                     CALL   DLY1            ; 20+18*21+20
0A0A CD5907                     CALL   DLY1            ; 20+18*21+20
0A0D 3E02                       LD     A,02H           ; 9
0A0F 3203E0                     LD     (CSTPT),A       ; $E003 PC3=0:16
0A12 CD5907                     CALL   DLY1            ; 20+18*21+20
0A15 CD5907                     CALL   DLY1            ; 20+18*21+20
0A18 F1                         POP    AF              ; 11
0A19 C9                         RET                    ; 11
0A1A                    ;
0A1A                    ;
0A1A                    LONG:   ENT
0A1A F5                         PUSH   AF              ; 11
0A1B 3E03                       LD     A,03H           ; 9
0A1D 3203E0                     LD     (CSTPT),A       ; 16
0A20 CDA909                     CALL   DLY4            ; 20+18*89+20
0A23 3E02                       LD     A,02H           ; 9
0A25 3203E0                     LD     (CSTPT),A       ; 16
0A28 CDA909                     CALL   DLY4            ; 20+18*89+20
0A2B F1                         POP    AF              ; 11
0A2C C9                         RET                    ; 11
0A2D                    ;
0A2D                    ;
0A2D                            DEFS   +5
0A32                    ;
0A32                    ;
0A32                    ;ORG 0A32H
0A32                    ;
0A32                    ;   BREAK KEY CHECK
0A32                    ;     AND SHIFT,CTNL KEY CHECK
0A32                    ;
0A32                    ;     EXIT BREAK ON : ZERO=1
0A32                    ;                OFF: ZERO=0
0A32                    ;          NO KEY   : CY  =0
0A32                    ;          KEY IN   : CY  =1
0A32                    ;           A D6=1  : SHIFT ON
0A32                    ;               =0  :      OFF
0A32                    ;             D5=1  : CTRL ON
0A32                    ;               =0  :      OFF
0A32                    ;             D4=1  : SFT+CNT ON
0A32                    ;               =0            OFF
0A32                    ;
0A32                    ?BRK:   ENT
0A32 3EF8                       LD     A,F8H           ; LINE 8SWEEP
0A34 3200E0                     LD     (KEYPA),A
0A37 00                         NOP
0A38 3A01E0                     LD     A,(KEYPB)
0A3B B7                         OR     A
0A3C 1F                         RRA
0A3D DA8009                     JP     C,?BRK2         ; SHIFT ?
0A40 17                         RLA
0A41 17                         RLA
```

```
0A42 3004                       JR     NC,?BRK1        ; BREAK ?
0A44 3E40                       LD     A,40H           ; SHIFT D6=1
0A46 37                         SCF
0A47 C9                         RET
0A48                    ;
0A48                    ;
0A48 AF                 ?BRK1:  XOR    A               ; SHIFT ?
0A49 C9                         RET
0A4A                    ;
0A4A                    ;
0A4A                    ;   320 U SEC DELAY
0A4A                    ;
0A4A                    DLY3:   ENT
0A4A 3E52                       LD     A,52H           ; 331.35 micro sec
0A4C C36207                     JP     0762H           ; JP DLY2+2
0A4F                    ;
0A4F                    ;
0A4F                            DEFS   +1
0A50                    ;
0A50                    ;
0A50                    ;
0A50                    ;ORG 0A50H ; ?SWEP
0A50                    ;
0A50                    ;
0A50                    ;   KEY BOAD SWEEP
0A50                    ;
0A50                    ;   EXIT B,D7=0  NO DATA
0A50                    ;            =1  DATA
0A50                    ;          D6=0  SHIFT OFF
0A50                    ;            =1  SHIFT ON
0A50                    ;          D5=0  CTRL OFF
0A50                    ;            =1  CTRL ON
0A50                    ;          D4=0  SHIFT+CTRL OFF
0A50                    ;            =1  SHIFT+CTRL ON
0A50                    ;        C   = ROW & COLOUMN
0A50                    ;           7 6 5 4 3 2 1 0
0A50                    ;           * *
0A50                    ;
0A50                    ;
0A50                    ?SWEP:  ENT
0A50 D5                         PUSH   DE
0A51 E5                         PUSH   HL
0A52 AF                         XOR    A
0A53 06F8                       LD     B,F8H
0A55 57                         LD     D,A
0A56 CD320A                     CALL   ?BRK
0A59 2004                       JR     NZ,SWEP6
0A5B 1688                       LD     D,88H           ; BREAK ON
0A5D 1814                       JR     SWEP9
0A5F                    SWEP6:  ENT
0A5F 3005                       JR     NC,SWEP0
0A61 57                         LD     D,A
0A62 1802                       JR     SWEP0
0A64                    SWEP01: ENT
0A64 CBFA                       SET    7,D
0A66                    SWEP0:  ENT
0A66 05                         DEC    B
0A67 78                         LD     A,B
0A68 3200E0                     LD     (KEYPA),A
0A6B FEEF                       CP     EFH             ; MAP SWEEP END ?
0A6D 2008                       JR     NZ,SWEP3
```

```
0A6F FEF8                CP    F8H              ; BREAK KEY ROW
0A71 28F3                JR    Z,SWEP0
0A73             SWEP9:  ENT
0A73 42                  LD    B,D
0A74 E1                  POP   HL
0A75 D1                  POP   DE
0A76 C9                  RET
0A77             ;
0A77             SWEP3:  ENT
0A77 3A01E0              LD    A,(KEYPB)
0A7A 2F                  CPL
0A7B B7                  OR    A
0A7C 28E8                JR    Z,SWEP0
0A7E 5F                  LD    E,A
0A7F             SWEP2:  ENT
0A7F 2608                LD    H,8
0A81 78                  LD    A,B
0A82 E60F                AND   0FH
0A84 07                  RLCA
0A85 07                  RLCA
0A86 07                  RLCA
0A87 4F                  LD    C,A
0A88 7B                  LD    A,E
0A89 25                  DEC   H
0A8A 0F                  RRCA
0A8B 30FC                JR    NC,-2
0A8D 7C                  LD    A,H
0A8E 81                  ADD   A,C
0A8F 4F                  LD    C,A
0A90 18D2                JR    SWEP01
0A92             ;
0A92             ;
0A92             ;   ASCII TO DISPLAY CODE TABL   ;
0A92             ;
0A92             ATBL:
0A92             ;  00 - 0F  ;
0A92 F0                  DEFB  F0H              ; @
0A93 F0                  DEFB  F0H              ; A
0A94 F0                  DEFB  F0H              ; B
0A95 F3                  DEFB  F3H              ; C
0A96 F0                  DEFB  F0H              ; D
0A97 F5                  DEFB  F5H              ; E
0A98 F0                  DEFB  F0H              ; F
0A99 F0                  DEFB  F0H              ; G
0A9A F0                  DEFB  F0H              ; H
0A9B F0                  DEFB  F0H              ; I
0A9C F0                  DEFB  F0H              ; J
0A9D F0                  DEFB  F0H              ; K
0A9E F0                  DEFB  F0H              ; L
0A9F F0                  DEFB  F0H              ; M
0AA0 F0                  DEFB  F0H              ; N
0AA1 F0                  DEFB  F0H              ; O
0AA2             ;  10 - 1F
0AA2 F0                  DEFB  F0H              ; P
0AA3 C1                  DEFB  C1H              ; Q CUR. DOWN
0AA4 C2                  DEFB  C2H              ; R CUR. UP
0AA5 C3                  DEFB  C3H              ; S CUR. RIGHT
0AA6 C4                  DEFB  C4H              ; T CUR. LEFT
0AA7 C5                  DEFB  C5H              ; U HOME
0AA8 C6                  DEFB  C6H              ; V CLEAR
```

```
0AA9 F0                  DEFB  F0H              ; W
0AAA F0                  DEFB  F0H              ; X
0AAB F0                  DEFB  F0H              ; Y
0AAC F0                  DEFB  F0H              ; Z SEP.
0AAD F0                  DEFB  F0H              ; [
0AAE F0                  DEFB  F0H              ; ¥
0AAF F0                  DEFB  F0H              ; ]
0AB0 F0                  DEFB  F0H              ; ^
0AB1 F0                  DEFB  F0H              ; _
0AB2             ;  20 ~ 2F  ;
0AB2 00                  DEFB  00H              ; SPACE
0AB3 61                  DEFB  61H              ; !
0AB4 62                  DEFB  62H              ; "
0AB5 63                  DEFB  63H              ; #
0AB6 64                  DEFB  64H              ; $
0AB7 65                  DEFB  65H              ; %
0AB8 66                  DEFB  66H              ; &
0AB9 67                  DEFB  67H              ; '
0ABA 68                  DEFB  68H              ; (
0ABB 69                  DEFB  69H              ; )
0ABC 6B                  DEFB  6BH              ; *
0ABD 6A                  DEFB  6AH              ; +
0ABE 2F                  DEFB  2FH              ; ,
0ABF 2A                  DEFB  2AH              ; -
0AC0 2E                  DEFB  2EH              ; .
0AC1 2D                  DEFB  2DH
0AC2             ;  30 -3F  ;
0AC2 20                  DEFB  20H              ; 0
0AC3 21                  DEFB  21H              ; 1
0AC4 22                  DEFB  22H              ; 2
0AC5 23                  DEFB  23H              ; 3
0AC6 24                  DEFB  24H              ; 4
0AC7 25                  DEFB  25H              ; 5
0AC8 26                  DEFB  26H              ; 6
0AC9 27                  DEFB  27H              ; 7
0ACA 28                  DEFB  28H              ; 8
0ACB 29                  DEFB  29H              ; 9
0ACC 4F                  DEFB  4FH              ; :
0ACD 2C                  DEFB  2CH              ; ;
0ACE 51                  DEFB  51H              ; <
0ACF 2B                  DEFB  2BH              ; =
0AD0 57                  DEFB  57H              ; >
0AD1 49                  DEFB  49H              ; ?
0AD2             ;  40 - 4F  ;
0AD2 55                  DEFB  55H              ; @
0AD3 01                  DEFB  01H              ; A
0AD4 02                  DEFB  02H              ; B
0AD5 03                  DEFB  03H              ; C
0AD6 04                  DEFB  04H              ; D
0AD7 05                  DEFB  05H              ; E
0AD8 06                  DEFB  06H              ; F
0AD9 07                  DEFB  07H              ; G
0ADA 08                  DEFB  08H              ; H
0ADB 09                  DEFB  09H              ; I
0ADC 0A                  DEFB  0AH              ; J
0ADD 0B                  DEFB  0BH              ; K
0ADE 0C                  DEFB  0CH              ; L
0ADF 0D                  DEFB  0DH              ; M
0AE0 0E                  DEFB  0EH              ; U
0AE1 0F                  DEFB  0FH              ; O
```

```
0AE2                ;  50 - 5F  ;
0AE2 10                     DEFB    10H             ; P
0AE3 11                     DEFB    11H             ; Q
0AE4 12                     DEFB    12H             ; R
0AE5 13                     DEFB    13H             ; S
0AE6 14                     DEFB    14H             ; T
0AE7 15                     DEFB    15H             ; U
0AE8 16                     DEFB    16H             ; V
0AE9 17                     DEFB    17H             ; W
0AEA 18                     DEFB    18H             ; X
0AEB 19                     DEFB    19H             ; Y
0AEC 1A                     DEFB    1AH             ; Z
0AED 52                     DEFB    52H             ; [
0AEE 59                     DEFB    59H             ; ¥
0AEF 54                     DEFB    54H             ; ]
0AF0 50                     DEFB    50H             ; ^
0AF1 45                     DEFB    45H             ; _
0AF2                ;  60 - 6F  ;
0AF2 C7                     DEFB    C7H             ; UFO
0AF3 C8                     DEFB    C8H
0AF4 C9                     DEFB    C9H
0AF5 CA                     DEFB    CAH
0AF6 CB                     DEFB    CBH
0AF7 CC                     DEFB    CCH
0AF8 CD                     DEFB    CDH
0AF9 CE                     DEFB    CEH
0AFA CF                     DEFB    CFH
0AFB DF                     DEFB    DFH
0AFC E7                     DEFB    E7H
0AFD E8                     DEFB    E8H
0AFE E5                     DEFB    E5H
0AFF E9                     DEFB    E9H
0B00 EC                     DEFB    ECH
0B01 ED                     DEFB    EDH
0B02                ;  70 - 7F  ;
0B02 D0                     DEFB    D0H
0B03 D1                     DEFB    D1H
0B04 D2                     DEFB    D2H
0B05 D3                     DEFB    D3H
0B06 D4                     DEFB    D4H
0B07 D5                     DEFB    D5H
0B08 D6                     DEFB    D6H
0B09 D7                     DEFB    D7H
0B0A D8                     DEFB    D8H
0B0B D9                     DEFB    D9H
0B0C DA                     DEFB    DAH
0B0D DB                     DEFB    DBH
0B0E DC                     DEFB    DCH
0B0F DD                     DEFB    DDH
0B10 DE                     DEFB    DEH
0B11 C0                     DEFB    C0H
0B12                ;  80 - 8F  ;
0B12 80                     DEFB    80H             ; }
0B13 BD                     DEFB    BDH             ;
0B14 9D                     DEFB    9DH
0B15 B1                     DEFB    B1H
0B16 B5                     DEFB    B5H
0B17 B9                     DEFB    B9H
0B18 B4                     DEFB    B4H             ;
0B19 9E                     DEFB    9EH             ;
```

```
0B1A B2                     DEFB    B2H             ;
0B1B B6                     DEFB    B6H             ;
0B1C BA                     DEFB    BAH             ;
0B1D BE                     DEFB    BEH             ;
0B1E 9F                     DEFB    9FH             ;
0B1F B3                     DEFB    B3H             ;
0B20 B7                     DEFB    B7H             ;
0B21 BB                     DEFB    BBH             ;
0B22                ;  90 - 9F  ;
0B22 BF                     DEFB    BFH             ; -
0B23 A3                     DEFB    A3H             ;
0B24 85                     DEFB    85H             ;
0B25 A4                     DEFB    A4H             ; `
0B26 A5                     DEFB    A5H             ; ~
0B27 A6                     DEFB    A6H             ;
0B28 94                     DEFB    94H             ;
0B29 87                     DEFB    87H             ;
0B2A 88                     DEFB    88H             ;
0B2B 9C                     DEFB    9CH             ;
0B2C 82                     DEFB    82H             ;
0B2D 98                     DEFB    98H             ;
0B2E 84                     DEFB    84H             ;
0B2F 92                     DEFB    92H             ;
0B30 90                     DEFB    90H             ;
0B31 83                     DEFB    83H             ;
0B32                ;  A0 - AF  ;
0B32 91                     DEFB    91H             ;
0B33 81                     DEFB    81H             ;
0B34 9A                     DEFB    9AH             ;
0B35 97                     DEFB    97H             ;
0B36 93                     DEFB    93H             ;
0B37 95                     DEFB    95H             ;
0B38 89                     DEFB    89H             ;
0B39 A1                     DEFB    A1H             ;
0B3A AF                     DEFB    AFH             ;
0B3B 8B                     DEFB    8BH             ;
0B3C 86                     DEFB    86H             ;
0B3D 96                     DEFB    96H             ;
0B3E A2                     DEFB    A2H             ;
0B3F AB                     DEFB    ABH             ;
0B40 AA                     DEFB    AAH             ;
0B41 8A                     DEFB    8AH             ;
0B42                ;  B0 - BF  ;
0B42 8E                     DEFB    8EH             ;
0B43 B0                     DEFB    B0H             ;
0B44 AD                     DEFB    ADH             ;
0B45 8D                     DEFB    8DH             ;
0B46 A7                     DEFB    A7H             ;
0B47 A8                     DEFB    A8H             ;
0B48 A9                     DEFB    A9H             ;
0B49 8F                     DEFB    8FH             ;
0B4A 8C                     DEFB    8CH             ;
0B4B AE                     DEFB    AEH             ;
0B4C AC                     DEFB    ACH             ;
0B4D 9B                     DEFB    9BH             ;
0B4E A0                     DEFB    A0H             ;
0B4F 99                     DEFB    99H             ;
0B50 BC                     DEFB    BCH             ; {
0B51 B8                     DEFB    B8H
0B52                ;  C0 - CF  ;
```

```
0B52 40                        DEFB    40H              ; ¦
0B53 3B                        DEFB    3BH
0B54 3A                        DEFB    3AH
0B55 70                        DEFB    70H
0B56 3C                        DEFB    3CH
0B57 71                        DEFB    71H
0B58 5A                        DEFB    5AH
0B59 3D                        DEFB    3DH
0B5A 43                        DEFB    43H
0B5B 56                        DEFB    56H
0B5C 3F                        DEFB    3FH
0B5D 1E                        DEFB    1EH
0B5E 4A                        DEFB    4AH
0B5F 1C                        DEFB    1CH
0B60 5D                        DEFB    5DH
0B61 3E                        DEFB    3EH
0B62                ;  D0 - DF  ;
0B62 5C                        DEFB    5CH
0B63 1F                        DEFB    1FH
0B64 5F                        DEFB    5FH
0B65 5E                        DEFB    5EH
0B66 37                        DEFB    37H
0B67 7B                        DEFB    7BH
0B68 7F                        DEFB    7FH
0B69 36                        DEFB    36H
0B6A 7A                        DEFB    7AH
0B6B 7E                        DEFB    7EH
0B6C 33                        DEFB    33H
0B6D 4B                        DEFB    4BH
0B6E 4C                        DEFB    4CH
0B6F 1D                        DEFB    1DH
0B70 6C                        DEFB    6CH
0B71 5B                        DEFB    5BH
0B72                ;  E0 - EF  ;
0B72 78                        DEFB    78H
0B73 41                        DEFB    41H
0B74 35                        DEFB    35H
0B75 34                        DEFB    34H
0B76 74                        DEFB    74H
0B77 30                        DEFB    30H
0B78 38                        DEFB    38H
0B79 75                        DEFB    75H
0B7A 39                        DEFB    39H
0B7B 4D                        DEFB    4DH
0B7C 6F                        DEFB    6FH
0B7D 6E                        DEFB    6EH
0B7E 32                        DEFB    32H
0B7F 77                        DEFB    77H
0B80 76                        DEFB    76H
0B81 72                        DEFB    72H
0B82                ;  F0 - FF  ;
0B82 73                        DEFB    73H
0B83 47                        DEFB    47H
0B84 7C                        DEFB    7CH
0B85 53                        DEFB    53H
0B86 31                        DEFB    31H
0B87 4E                        DEFB    4EH
0B88 6D                        DEFB    6DH
0B89 48                        DEFB    48H
0B8A 46                        DEFB    46H
```

```
0B8B 7D                       DEFB    7DH
0B8C 44                       DEFB    44H
0B8D 1B                       DEFB    1BH
0B8E 58                       DEFB    58H
0B8F 79                       DEFB    79H
0B90 42                       DEFB    42H
0B91 60                       DEFB    60H
0B92                 ;
0B92                 ;
0B92                 ;   FLASHING DATA SAVE
0B92                 ;
0B92                 ?SAVE:   ENT
0B92 219211                   LD      HL,FLSDT          ; NOMAL CURSOR
0B95 36EF                     LD      (HL),EFH
0B97 3A7011                   LD      A,(KANAF)
0B9A 0F                       RRCA
0B9B 3803                     JR      C,SV0-2           ; GRAPH MODE
0B9D 0F                       RRCA
0B9E 3002                     JR      NC,SV0            ; NORMAL MODE
0BA0 36FF                     LD      (HL),FFH          ; GRAPH CURSOR
0BA2                 SV0:     ENT
0BA2 7E                       LD      A,(HL)
0BA3 F5                       PUSH    AF
0BA4 CDB10F                   CALL    ?PONT             ; FLASING POSITION
0BA7 7E                       LD      A,(HL)
0BA8 328E11                   LD      (FLASH),A
0BAB F1                       POP     AF
0BAC 77                       LD      (HL),A
0BAD AF                       XOR     A
0BAE 2100E0                   LD      HL,KEYPA
0BB1 77                       LD      (HL),A
0BB2 2F                       CPL
0BB3 77                       LD      (HL),A
0BB4 C9                       RET
0BB5                 SV1:     ENT
0BB5 3643                     LD      (HL),43H          ; KANA CURSOR
0BB7 18E9                     JR      SV0
0BB9                 ;
0BB9                 ;ORG 0BB9H;?ADCN
0BB9                 ;
0BB9                 ;
0BB9                 ;   ASCII TO DISPLAY CODE CONVERTE
0BB9                 ;
0BB9                 ;        IN   ACC:ASCII
0BB9                 ;        EXIT ACC:DISPLAY CODE
0BB9                 ;
0BB9                 ?ADCN:   ENT
0BB9 C5                       PUSH    BC
0BBA E5                       PUSH    HL
0BBB 21920A                   LD      HL,ATBL
0BBE 4F                       LD      C,A
0BBF 0600                     LD      B,0
0BC1 09                       ADD     HL,BC
0BC2 7E                       LD      A,(HL)
0BC3 181B                     JR      DACN3
0BC5                 ;
0BC5 56312E30        VRNS:    DEFM    "V1.0A"           ; VERSION MANAGEMENT
0BC9 41
0BCA 0D                       DEFB    0DH
0BCB                          DEFS    +3
```

```
0BCE                ;
0BCE                ;
0BCE                ;ORG 0BCEH;?DACN
0BCE                ;
0BCE                ;   DISPLAY CODE TO ASCII CONV.  ;
0BCE                ;
0BCE                ;     IN   ACC = DISPLAY CODE
0BCE                ;     EXIT ACC = ASCII
0BCE                ;
0BCE                ?DACN:  ENT
0BCE C5                     PUSH    BC
0BCF E5                     PUSH    HL
0BD0 D5                     PUSH    DE
0BD1 21920A                 LD      HL,ATBL
0BD4 54                     LD      D,H
0BD5 5D                     LD      E,L
0BD6 010001                 LD      BC,0100H
0BD9 EDB1                   CPIR
0BDB 2806                   JR      Z,DACN1
0BDD 3EF0                   LD      A,F0H
0BDF                DACN2:  ENT
0BDF D1                     POP     DE
0BE0                DACN3:  ENT
0BE0 E1                     POP     HL
0BE1 C1                     POP     BC
0BE2 C9                     RET
0BE3                ;
0BE3                DACN1:  ENT
0BE3 B7                     OR      A
0BE4 2B                     DEC     HL
0BE5 ED52                   SBC     HL,DE
0BE7 7D                     LD      A,L
0BE8 18F5                   JR      DACN2
0BEA                ;
0BEA                ;
0BEA                ;
0BEA                ;   KEY MATRIX TO DISPLAY CODE TABL
0BEA                ;
0BEA                KTBL:   ENT
0BEA                ;S0   00 - 07  ;
0BEA BF                     DEFB    BFH             ; SPARE
0BEB CA                     DEFB    CAH             ; GRAPH
0BEC 58                     DEFB    58H             ;
0BED C9                     DEFB    C9H             ; ALPHA
0BEE F0                     DEFB    F0H             ; NO
0BEF 2C                     DEFB    2CH             ; ;
0BF0 4F                     DEFB    4FH             ; :
0BF1 CD                     DEFB    CDH             ; CR
0BF2                ;S1   08 - 0F  ;
0BF2 19                     DEFB    19H             ; Y
0BF3 1A                     DEFB    1AH             ; Z
0BF4 55                     DEFB    55H             ; @
0BF5 52                     DEFB    52H             ; [
0BF6 54                     DEFB    54H             ; ]
0BF7 F0                     DEFB    F0H             ; NULL
0BF8 F0                     DEFB    F0H             ; NULL
0BF9 F0                     DEFB    F0H             ; NULL
0BFA                ;S2   0 - 17  ;
0BFA 11                     DEFB    11H             ; Q
0BFB 12                     DEFB    12H             ; R
```

```
0BFC 13                     DEFB    13H             ; S
0BFD 14                     DEFB    14H             ; T
0BFE 15                     DEFB    15H             ; U
0BFF 16                     DEFB    16H             ; V
0C00 17                     DEFB    17H             ; W
0C01 18                     DEFB    18H             ; X
0C02                ;S3   18 - 1F  ;
0C02 09                     DEFB    09H             ; I
0C03 0A                     DEFB    0AH             ; J
0C04 0B                     DEFB    0BH             ; K
0C05 0C                     DEFB    0CH             ; L
0C06 0D                     DEFB    0DH             ; M
0C07 0E                     DEFB    0EH             ; N
0C08 0F                     DEFB    0FH             ; O
0C09 10                     DEFB    10H             ; P
0C0A                ;S4  20 - 27  ;
0C0A 01                     DEFB    01H             ; A
0C0B 02                     DEFB    02H             ; B
0C0C 03                     DEFB    03H             ; C
0C0D 04                     DEFB    04H             ; D
0C0E 05                     DEFB    05H             ; E
0C0F 06                     DEFB    06H             ; F
0C10 07                     DEFB    07H             ; G
0C11 08                     DEFB    08H             ; H
0C12                ;S5  28 - 2F  ;
0C12 21                     DEFB    21H             ; 1
0C13 22                     DEFB    22H             ; 2
0C14 23                     DEFB    23H             ; 3
0C15 24                     DEFB    24H             ; 4
0C16 25                     DEFB    25H             ; 5
0C17 26                     DEFB    26H             ; 6
0C18 27                     DEFB    27H             ; 7
0C19 28                     DEFB    28H             ; 8
0C1A                ;S6  30 - 37  ;
0C1A 59                     DEFB    59H             ; ¥
0C1B 50                     DEFB    50H             ;
0C1C 2A                     DEFB    2AH             ; -
0C1D 00                     DEFB    00H             ; SPACE
0C1E 20                     DEFB    20H             ; 0
0C1F 29                     DEFB    29H             ; 9
0C20 2F                     DEFB    2FH             ; ,
0C21 2E                     DEFB    2EH             ; .
0C22                ;S7  38 - 3F  ;
0C22 C8                     DEFB    C8H             ; INST.
0C23 C7                     DEFB    C7H             ; DEL.
0C24 C2                     DEFB    C2H             ; CURSOR UP
0C25 C1                     DEFB    C1H             ; CURSOR DOWN
0C26 C3                     DEFB    C3H             ; CURSOR RIGHT
0C27 C4                     DEFB    C4H             ; CURSOR LEFT
0C28 49                     DEFB    49H             ; ?
0C29 2D                     DEFB    2DH             ; /
0C2A                ;
0C2A                ;   KTBL SHIFT ON
0C2A                ;
0C2A                KTBLS:  ENT
0C2A                ;S0   00-07
0C2A BF                     DEFB    BFH             ; SPARE
0C2B CA                     DEFB    CAH             ; GRAPH
0C2C 1B                     DEFB    1BH             ; POND
0C2D C9                     DEFB    C9H             ; ALPHA
```

```
0C2E F0                  DEFB   F0H          ; NO
0C2F 6A                  DEFB   6AH          ; +
0C30 6B                  DEFB   6BH          ; *
0C31 CD                  DEFB   CDH          ; CR
0C32             ;S1   08-0F
0C32 99                  DEFB   99H          ; y
0C33 9A                  DEFB   9AH          ; z
0C34 A4                  DEFB   A4H          ; `
0C35 BC                  DEFB   BCH          ; {
0C36 40                  DEFB   40H          ; }
0C37 F0                  DEFB   F0H          ; NULL
0C38 F0                  DEFB   F0H          ; NULL
0C39 F0                  DEFB   F0H          ; NULL
0C3A             ;S2   10-17
0C3A 91                  DEFB   91H          ; q
0C3B 92                  DEFB   92H          ; r
0C3C 93                  DEFB   93H          ; s
0C3D 94                  DEFB   94H          ; t
0C3E 95                  DEFB   95H          ; u
0C3F 96                  DEFB   96H          ; v
0C40 97                  DEFB   97H          ; w
0C41 98                  DEFB   98H          ; x
0C42             ;S3   18-1F
0C42 89                  DEFB   89H          ; i
0C43 8A                  DEFB   8AH          ; j
0C44 8B                  DEFB   8BH          ; k
0C45 8C                  DEFB   8CH          ; l
0C46 8D                  DEFB   8DH          ; m
0C47 8E                  DEFB   8EH          ; n
0C48 8F                  DEFB   8FH          ; o
0C49 90                  DEFB   90H          ; p
0C4A             ;S4   20-27
0C4A 81                  DEFB   81H          ; a
0C4B 82                  DEFB   82H          ; b
0C4C 83                  DEFB   83H          ; c
0C4D 84                  DEFB   84H          ; d
0C4E 85                  DEFB   85H          ; e
0C4F 86                  DEFB   86H          ; f
0C50 87                  DEFB   87H          ; g
0C51 88                  DEFB   88H          ; h
0C52             ;S5   28-2F
0C52 61                  DEFB   61H          ; !
0C53 62                  DEFB   62H          ; "
0C54 63                  DEFB   63H          ; #
0C55 64                  DEFB   64H          ; $
0C56 65                  DEFB   65H          ; %
0C57 66                  DEFB   66H          ; &
0C58 67                  DEFB   67H          ; '
0C59 68                  DEFB   68H          ; (
0C5A             ;S6   30-37
0C5A 80                  DEFB   80H          ; ¥
0C5B A5                  DEFB   A5H          ; POND MARK
0C5C 2B                  DEFB   2BH          ; YEN
0C5D 00                  DEFB   00H          ; SPACE
0C5E 60                  DEFB   60H          ;
0C5F 69                  DEFB   69H          ; )
0C60 51                  DEFB   51H          ; <
0C61 57                  DEFB   57H          ; >
0C62             ;S7      38-3F
0C62 C6                  DEFB   C6H          ; CLR
```

```
0C63 C5                  DEFB   C5H          ; HOME
0C64 C2                  DEFB   C2H          ; CURSOR UP
0C65 C1                  DEFB   C1H          ; CURSOR DOWN
0C66 C3                  DEFB   C3H          ; CURSOR RIGHT
0C67 C4                  DEFB   C4H          ; CURSOR LEFT
0C68 5A                  DEFB   5AH          ;
0C69 45                  DEFB   45H          ;
0C6A             ;
0C6A             ;   GRAPHIC
0C6A             ;
0C6A             KTBLGS: ENT
0C6A             ;S0   00-07
0C6A BF                  DEFB   BFH          ; SPARE
0C6B F0                  DEFB   F0H          ; GRAPH BUT NULL
0C6C E5                  DEFB   E5H          ; #
0C6D C9                  DEFB   C9H          ; ALPHA
0C6E F0                  DEFB   F0H          ; NO
0C6F 42                  DEFB   42H          ; #;
0C70 B6                  DEFB   B6H          ; #:
0C71 CD                  DEFB   CDH          ; CR
0C72             ;S1   08-0F
0C72 75                  DEFB   75H          ; #Y
0C73 76                  DEFB   76H          ; #Z
0C74 B2                  DEFB   B2H          ; #@
0C75 D8                  DEFB   D8H          ; #[
0C76 4E                  DEFB   4EH          ; #]
0C77 F0                  DEFB   F0H          ; #NULL
0C78 F0                  DEFB   F0H          ; #NULL
0C79 F0                  DEFB   F0H          ; #NULL
0C7A             ;S2   10-17
0C7A 3C                  DEFB   3CH          ; #Q
0C7B 30                  DEFB   30H          ; #R
0C7C 44                  DEFB   44H          ; #S
0C7D 71                  DEFB   71H          ; #T
0C7E 79                  DEFB   79H          ; #U
0C7F DA                  DEFB   DAH          ; #V
0C80 38                  DEFB   38H          ; #W
0C81 6D                  DEFB   6DH          ; #X
0C82             ;S3   18-1F
0C82 7D                  DEFB   7DH          ; #I
0C83 5C                  DEFB   5CH          ; #J
0C84 5B                  DEFB   5BH          ; #K
0C85 B4                  DEFB   B4H          ; #L
0C86 1C                  DEFB   1CH          ; #M
0C87 32                  DEFB   32H          ; #N
0C88 B0                  DEFB   B0H          ; #O
0C89 D6                  DEFB   D6H          ; #P
0C8A             ;S4   20-27
0C8A 53                  DEFB   53H          ; #A
0C8B 6F                  DEFB   6FH          ; #B
0C8C DE                  DEFB   DEH          ; #C
0C8D 47                  DEFB   47H          ; #D
0C8E 34                  DEFB   34H          ; #E
0C8F 4A                  DEFB   4AH          ; #F
0C90 4B                  DEFB   4BH          ; #G
0C91 72                  DEFB   72H          ; #H
0C92             ;S5   28-2F
0C92 37                  DEFB   37H          ; #1
0C93 3E                  DEFB   3EH          ; #2
0C94 7F                  DEFB   7FH          ; #3
```

```
0C95 7B                 DEFB   7BH          ; #4
0C96 3A                 DEFB   3AH          ; #5
0C97 5E                 DEFB   5EH          ; #6
0C98 1F                 DEFB   1FH          ; #7
0C99 BD                 DEFB   BDH          ; #8
0C9A          ;S6   30-3F
0C9A D4                 DEFB   D4H          ; #YEN
0C9B 9E                 DEFB   9EH          ; #+
0C9C D2                 DEFB   D2H          ; #-
0C9D 00                 DEFB   00H          ; SPACE
0C9E 9C                 DEFB   9CH          ; #0
0C9F A1                 DEFB   A1H          ; #9
0CA0 CA                 DEFB   CAH          ; #,
0CA1 B8                 DEFB   B8H          ; #.
0CA2          ;S7    38-3F
0CA2 C8                 DEFB   C8H          ; INST
0CA3 C7                 DEFB   C7H          ; DEL.
0CA4 C2                 DEFB   C2H          ; CURSOR UP
0CA5 C1                 DEFB   C1H          ; CURSOR DOWN
0CA6 C3                 DEFB   C3H          ; CURSOR RIGHT
0CA7 C4                 DEFB   C4H          ; CURSOR LEFT
0CA8 BA                 DEFB   BAH          ; #?
0CA9 DB                 DEFB   DBH          ; #/
0CAA          ;
0CAA          ;    CONTROL CODE
0CAA          ;
0CAA          KTBLC:    ENT
0CAA          ;S0       00-07N
0CAA F0                 DEFB   F0H          ;
0CAB F0                 DEFB   F0H
0CAC F0                 DEFB   F0H          ;
0CAD F0                 DEFB   F0H
0CAE F0                 DEFB   F0H
0CAF F0                 DEFB   F0H
0CB0 F0                 DEFB   F0H
0CB1 F0                 DEFB   F0H
0CB2          ;S1       08-0F
0CB2 F0                 DEFB   F0H          ; Y  E3
0CB3 5A                 DEFB   5AH          ; Z  E4 (CHECKER)
0CB4 F0                 DEFB   F0H          ; @
0CB5 F0                 DEFB   F0H          ; [  E5
0CB6 F0                 DEFB   F0H          ; ]  E7
0CB7 F0                 DEFB   F0H
0CB8 F0                 DEFB   F0H
0CB9 F0                 DEFB   F0H
0CBA          ;S2       10-17
0CBA C1                 DEFB   C1H          ; Q
0CBB C2                 DEFB   C2H          ; R
0CBC C3                 DEFB   C3H          ; S
0CBD C4                 DEFB   C4H          ; T
0CBE C5                 DEFB   C5H          ; U
0CBF C6                 DEFB   C6H          ; V
0CC0 F0                 DEFB   F0H          ; W  E1
0CC1 F0                 DEFB   F0H          ; X  E2
0CC2          ;S3       18-1F
0CC2 F0                 DEFB   F0H          ; I  F9
0CC3 F0                 DEFB   F0H          ; J  FA
0CC4 F0                 DEFB   F0H          ; K  FB
0CC5 F0                 DEFB   F0H          ; L  FC
0CC6 F0                 DEFB   F0H          ; M  FD
```

```
0CC7 F0                 DEFB   F0H          ; N  FE
0CC8 F0                 DEFB   F0H          ; O  FF
0CC9 F0                 DEFB   F0H          ; P  E0
0CCA          ;S4       20-27
0CCA F0                 DEFB   F0H          ; A  F1
0CCB F0                 DEFB   F0H          ; B  F2
0CCC F0                 DEFB   F0H          ; C  F3
0CCD F0                 DEFB   F0H          ; D  F4
0CCE F0                 DEFB   F0H          ; E  F5
0CCF F0                 DEFB   F0H          ; F  F6
0CD0 F0                 DEFB   F0H          ; G  F7
0CD1 F0                 DEFB   F0H          ; H  F8
0CD2          ;S5       28-2F
0CD2 F0                 DEFB   F0H
0CD3 F0                 DEFB   F0H
0CD4 F0                 DEFB   F0H
0CD5 F0                 DEFB   F0H
0CD6 F0                 DEFB   F0H
0CD7 F0                 DEFB   F0H
0CD8 F0                 DEFB   F0H
0CD9 F0                 DEFB   F0H
0CDA          ;S6       30-37
0CDA F0                 DEFB   F0H          ; YEN E6
0CDB F0                 DEFB   F0H
0CDC F0                 DEFB   F0H
0CDD F0                 DEFB   F0H
0CDE F0                 DEFB   F0H
0CDF F0                 DEFB   F0H          ; , EF
0CE0 F0                 DEFB   F0H
0CE1          ;S7          38-3F
0CE1 F0                 DEFB   F0H
0CE2 F0                 DEFB   F0H
0CE3 F0                 DEFB   F0H
0CE4 F0                 DEFB   F0H
0CE5 F0                 DEFB   F0H
0CE6 F0                 DEFB   F0H
0CE7 F0                 DEFB   F0H
0CE8 F0                 DEFB   F0H          ; / EE
0CE9          ;
0CE9          ;    KANA
0CE9          ;
0CE9          KTBLG:    ENT
0CE9          ;S0          00-07
0CE9 BF                 DEFB   BFH          ; SPARE
0CEA F0                 DEFB   F0H          ; GRAPH BUT NULL
0CEB CF                 DEFB   CFH          ; NIKO WH.
0CEC C9                 DEFB   C9H          ; ALPHA
0CED F0                 DEFB   F0H          ; NO
0CEE B5                 DEFB   B5H          ; MO
0CEF 4D                 DEFB   4DH          ; DAKU TEN
0CF0 CD                 DEFB   CDH          ; CR
0CF1          ;S1          08-0F
0CF1 35                 DEFB   35H          ; HA
0CF2 77                 DEFB   77H          ; TA
0CF3 D7                 DEFB   D7H          ; WA
0CF4 B3                 DEFB   B3H          ; YO
0CF5 B7                 DEFB   B7H          ; HANDAKU
0CF6 F0                 DEFB   F0H
0CF7 F0                 DEFB   F0H
0CF8 F0                 DEFB   F0H
```

```
0CF9                  ;S2      10-17
0CF9 7C                       DEFB    7CH              ; KA
0CFA 70                       DEFB    70H              ; KE
0CFB 41                       DEFB    41H              ; SHI
0CFC 31                       DEFB    31H              ; KO
0CFD 39                       DEFB    39H              ; HI
0CFE A6                       DEFB    A6H              ; TE
0CFF 78                       DEFB    78H              ; KI
0D00 DD                       DEFB    DDH              ; CHI
0D01                  ;S3      18-1F
0D01 3D                       DEFB    3DH              ; FU
0D02 5D                       DEFB    5DH              ; MI
0D03 6C                       DEFB    6CH              ; MU
0D04 56                       DEFB    56H              ; ME
0D05 1D                       DEFB    1DH              ; RHI
0D06 33                       DEFB    33H              ; RA
0D07 D5                       DEFB    D5H              ; HE
0D08 B1                       DEFB    B1H              ; HO
0D09                  ;S4      20-27
0D09 46                       DEFB    46H              ; SA
0D0A 6E                       DEFB    6EH              ; TO
0D0B D9                       DEFB    D9H              ; THU
0D0C 48                       DEFB    48H              ; SU
0D0D 74                       DEFB    74H              ; KU
0D0E 43                       DEFB    43H              ; SE
0D0F 4C                       DEFB    4CH              ; SO
0D10 73                       DEFB    73H              ; MA
0D11                  ;S5      28-2F
0D11 3F                       DEFB    3FH              ; A
0D12 36                       DEFB    36H              ; I
0D13 7E                       DEFB    7EH              ; U
0D14 3B                       DEFB    3BH              ; E
0D15 7A                       DEFB    7AH              ; O
0D16 1E                       DEFB    1EH              ; NA
0D17 5F                       DEFB    5FH              ; NI
0D18 A2                       DEFB    A2H              ; NU
0D19                  ;S6      30-37
0D19 D3                       DEFB    D3H              ; YO
0D1A 9F                       DEFB    9FH              ; YU
0D1B D1                       DEFB    D1H              ; YA
0D1C 00                       DEFB    00H              ; SPACE
0D1D 9D                       DEFB    9DH              ; NO
0D1E A3                       DEFB    A3H              ; NE
0D1F D0                       DEFB    D0H              ; RU
0D20 B9                       DEFB    B9H              ; RE
0D21                  ;S7      38-3F
0D21 C6                       DEFB    C6H              ; ?CLR
0D22 C5                       DEFB    C5H              ; ?HOME
0D23 C2                       DEFB    C2H              ; ?CURSOR UP
0D24 C1                       DEFB    C1H              ; ?CURSOR DOWN
0D25 C3                       DEFB    C3H              ; ?CURSOR RIGHT
0D26 C4                       DEFB    C4H              ; ?CURSOR LEFT
0D27 BB                       DEFB    BBH              ; DASH
0D28 BE                       DEFB    BEH              ; RO
0D29                  ;
0D29                  ;     MEMORY DUMP
0D29                  ;        COMMAND 'D'
0D29                  ;
0D29                  DUMP:   ENT
0D29 CD3D01                   CALL    HEX1Y            ; START ADR.
```

```
0D2C CDA602                   CALL    .4DE
0D2F E5                       PUSH    HL
0D30 CD1004                   CALL    HLHEX            ; END ADR.
0D33 D1                       POP     DE
0D34 3852                     JR      C,DUM1           ; DATA ER. THEN
0D36 EB                       EX      DE,HL
0D37 0608             DUM3:   LD      B,08H            ; DISP 8BYTES
0D39 0E17                     LD      C,23             ; CHA. PRINT BIAS
0D3B CDFA05                   CALL    NLPHL            ; NEWLINE PRINT
0D3E CDB103           DUM2:   CALL    SPHEX            ; SP. PRT.+ACC PRT.
0D41 23                       INC     HL
0D42 F5                       PUSH    AF
0D43 3A7111                   LD      A,(DSPXY)        ; DISPLAY POINT
0D46 81                       ADD     A,C
0D47 327111                   LD      (DSPXY),A        ; X AXIS.=X+Creg
0D4A F1                       POP     AF
0D4B FE20                     CP      20H
0D4D 3002                     JR      NC,+4
0D4F 3E2E                     LD      A,2EH            ; '.'
0D51 CDB90B                   CALL    ?ADCN            ; ASCII TO DSPLAY CODE
0D54 CD6C09                   CALL    PRNT3
0D57 3A7111                   LD      A,(DSPXY)
0D5A 0C                       INC     C
0D5B 91                       SUB     C                ; ASCII DSP POSITION
0D5C 327111                   LD      (DSPXY),A
0D5F 0D                       DEC     C
0D60 0D                       DEC     C
0D61 0D                       DEC     C
0D62 E5                       PUSH    HL
0D63 ED52                     SBC     HL,DE
0D65 E1                       POP     HL
0D66 281D                     JR      Z,DUM1-3
0D68 3EF8                     LD      A,F8H
0D6A 3200E0                   LD      (KEYPA),A
0D6D 00                       NOP
0D6E 3A01E0                   LD      A,(KEYPB)
0D71 FEFE                     CP      FEH              ; SHIFT KEY ?
0D73 2003                     JR      NZ,+5
0D75 CDA60D                   CALL    ?BLNK            ; 64MSEC DELAY
0D78 10C4                     DJNZ    DUM2
0D7A CDCA08                   CALL    ?KEY             ; STOP DISP
0D7D B7                       OR      A
0D7E 28FA                     JR      Z,-4             ; SPACE KEY THEN STOP
0D80 CD320A                   CALL    ?BRK             ; BREAK IN ?
0D83 20B2                     JR      NZ,DUM3
0D85 C3AD00                   JP      ST1              ; COMMAND IN !
0D88 21A000           DUM1:   LD      HL,160           ; 20*8 BYTE
0D8B 19                       ADD     HL,DE
0D8C 18A8                     JR      DUM3-1
0D8E                  ;
0D8E                  ;
0D8E                  ;
0D8E                  ;
0D8E                          DEFS    +24
0DA6                  ;
0DA6                  ;
0DA6                  ;
0DA6                  ;ORG 0DA6H;?BLNK
0DA6                  ;
0DA6                  ;
```

```
0DA6                 ;  V-BLANK CHECK  ;
0DA6                 ;
0DA6                 ?BLNK:  ENT
0DA6 F5                      PUSH   AF
0DA7 3A02E0                  LD     A,(KEYPC)        ; V-BLNK
0DAA 07                      RLCA
0DAB 30FA                    JR     NC,-4
0DAD 3A02E0                  LD     A,(KEYPC)
0DB0 07                      RLCA
0DB1 38FA                    JR     C,-4
0DB3 F1                      POP    AF
0DB4 C9                      RET
0DB5                 ;
0DB5                 ;ORG 0DB5H;?DSP
0DB5                 ;
0DB5                 ;
0DB5                 ;
0DB5                 ;  DISPLAY ON POINTER  ;
0DB5                 ;
0DB5                 ;    ACC = DISPLAY CODE
0DB5                 ;    EXCEPT F0H
0DB5                 ;
0DB5                 ?DSP:   ENT
0DB5 F5                      PUSH   AF
0DB6 C5                      PUSH   BC
0DB7 D5                      PUSH   DE
0DB8 E5                      PUSH   HL
0DB9                 DSP01:  ENT
0DB9 CDB10F                  CALL   ?PONT            ; DSPLAY POSITION
0DBC 77                      LD     (HL),A
0DBD 2A7111                  LD     HL,(DSPXY)
0DC0 7D                      LD     A,L
0DC1 FE27                    CP     +39
0DC3 200B                    JR     NZ,DSP04
0DC5 CDF302                  CALL   .MANG
0DC8 3806                    JR     C,DSP04
0DCA EB                      EX     DE,HL
0DCB 3601                    LD     (HL),+1          ; LOGICAL 1ST COLUMN
0DCD 23                      INC    HL
0DCE 3600                    LD     (HL),0           ; LOGICAL 2ND COLUMN
0DD0                 DSP04:  ENT
0DD0 3EC3                    LD     A,C3H            ; CURSL
0DD2 180C                    JR     ?DPCT+4
0DD4                 ;
0DD4                 ;
0DD4                 ;
0DD4                 ;
0DD4                 ;  GRAPHIC STATUS CHECK
0DD4                 ;
0DD4 3A7011          GRSTAS: LD     A,(KANAF)
0DD7 FE01                    CP     01H
0DD9 3ECA                    LD     A,CAH
0DDB C9                      RET
0DDC                 ;
0DDC                 ;
0DDC                 ;
0DDC                 ;
0DDC                 ;
0DDC                 ;ORG 0DDCH;?DPCT
```

```
0DDC                 ;
0DDC                 ;
0DDC                 ;   DISPLAY CONTROL   ;
0DDC                 ;
0DDC                 ;     ACC = CONTROL CODE
0DDC                 ;
0DDC                 ?DPCT:  ENT
0DDC F5                      PUSH   AF
0DDD C5                      PUSH   BC
0DDE D5                      PUSH   DE
0DDF E5                      PUSH   HL
0DE0 47                      LD     B,A
0DE1 E6F0                    AND    F0H
0DE3 FEC0                    CP     C0H
0DE5 201B                    JR     NZ,CURS5
0DE7 A8                      XOR    B
0DE8 07                      RLCA
0DE9 4F                      LD     C,A
0DEA 0600                    LD     B,+0
0DEC 21AA0E                  LD     HL,CTBL          ; PAGE MODE1
0DEF 09                      ADD    HL,BC
0DF0 5E                      LD     E,(HL)
0DF1 23                      INC    HL
0DF2 56                      LD     D,(HL)
0DF3 2A7111                  LD     HL,(DSPXY)
0DF6 EB                      EX     DE,HL
0DF7 E9                      JP     (HL)
0DF8                 ;
0DF8                 ;
0DF8                 ;
0DF8                 CURSD:  ENT
0DF8 EB                      EX     DE,HL            ; LD HL,(DSPXY)
0DF9 7C                      LD     A,H
0DFA FE18                    CP     +24
0DFC 2825                    JR     Z,CURS4
0DFE 24                      INC    H
0DFF                 CURS1:  ENT
0DFF
0DFF
0DFF
0DFF                 CURS3:  ENT
0DFF 227111                  LD     (DSPXY),HL
0E02 C3E50E          CURS5:  JP     ?RSTR
0E05                 ;
0E05                 CURSU:  ENT
0E05 EB                      EX     DE,HL            ; LD HL,(DSPXY)
0E06 7C                      LD     A,H
0E07 B7                      OR     A
0E08 28F8                    JR     Z,CURS5
0E0A 25                      DEC    H
0E0B                 CURSU1: ENT
0E0B 18F2                    JR     CURS3
0E0D                 CURSR:  ENT
0E0D EB                      EX     DE,HL            ; LD HL,(DSPXY)
0E0E 7D                      LD     A,L
0E0F FE27                    CP     +39
0E11 3003                    JR     NC,CURS2
0E13 2C                      INC    L
0E14 18E9                    JR     CURS3
0E16                 CURS2:  ENT
```

```
**  Z80 ASSEMBLER 2Z-004C  <1Z-013B>  PAGE 55          03.06.85

0E16 2E00                LD      L,+0
0E18 24                  INC     H
0E19 7C                  LD      A,H
0E1A FE19                CP      +25
0E1C 38E1                JR      C,CURS1
0E1E 2618                LD      H,+24
0E20 227111              LD      (DSPXY),HL
0E23             CURS4:  ENT
0E23 1848                JR      SCROL
0E25             ;
0E25             CURSL:  ENT
0E25 EB                  EX      DE,HL          ; LD HL,(DSPXY)
0E26 7D                  LD      A,L
0E27 B7                  OR      A
0E28 2803                JR      Z,+5
0E2A 2D                  DEC     L
0E2B 1802                JR      CURS3
0E2D 2E27                LD      L,+39
0E2F 25                  DEC     H
0E30 F20B0E              JP      P,CURSU1
0E33 2600                LD      H,0
0E35 227111              LD      (DSPXY),HL
0E38 18C8                JR      CURS5
0E3A             ;
0E3A             CLRS:   ENT
0E3A 217311              LD      HL,MANG
0E3D 061B                LD      B,27
0E3F CDD80F              CALL    ?CLER
0E42 2100D0              LD      HL,D000H       ; SCRN TOP
0E45 CDD409              CALL    #CLR08
0E48 3E71                LD      A,71H          ; COLOR DATA
0E4A CDD509              CALL    #CLR8          ; D800H-DFFFH CLR.
0E4D             HOME:   ENT
0E4D 210000              LD      HL,0           ; DSPXY:0 X=0,Y=0
0E50 18AD                JR      CURS3
0E52             ;
0E52                     DEFS    +8
0E5A             ;
0E5A             ;      CR
0E5A             ;
0E5A             CR;     ENT
0E5A CDF302              CALL    .MANG
0E5D 0F                  RRCA
0E5E 30B6                JR      NC,CURS2
0E60 2E00                LD      L,0
0E62 24                  INC     H
0E63 FE18                CP      +24
0E65 2803                JR      Z,CR1
0E67 24                  INC     H
0E68 1895                JR      CURS1
0E6A             CR1:    ENT
0E6A 227111              LD      (DSPXY),HL
0E6D             ;
0E6D             ;    SCROL
0E6D             ;
0E6D             SCROL:  ENT
0E6D 01C003              LD      BC,03C0H
0E70 1100D0              LD      DE,SCRN        ; TOP OF $CRT ADR.
0E73 2128D0              LD      HL,SCRN+40     ; 1 COLUMN
0E76 C5                  PUSH    BC             ; 1000 STORE
```

```
**  Z80 ASSEMBLER 2Z-004C  <1Z-013B>  PAGE 56          03.06.85

0E77 EDB0                LDIR
0E79 C1                  POP     BC
0E7A D5                  PUSH    DE
0E7B 1100D8              LD      DE,SCRN+800H   ; COLOR RAM SCROLL
0E7E 2128D8              LD      HL,SCRN+828H   ; SCROLL TOP + 40
0E81 EDB0                LDIR
0E83 0628                LD      B,40           ; ONE LINE
0E85 EB                  EX      DE,HL
0E86 3E71                LD      A,71H          ; COLOR RAM INITIAL DATA
0E88 CDDD0F              CALL    ?DINT
0E8B E1                  POP     HL
0E8C 0628                LD      B,40
0E8E CDD80F              CALL    ?CLER          ; LAST LINE CLEAR
0E91 011A00              LD      BC,26          ; ROW NUMBER +1
0E94 117311              LD      DE,MANG        ; LOGICAL MANAGEMENT
0E97 217411              LD      HL,MANG+1
0E9A EDB0                LDIR
0E9C 3600                LD      (HL),0
0E9E 3A7311              LD      A,(MANG)
0EA1 B7                  OR      A
0EA2 2841                JR      Z,?RSTR
0EA4 217211              LD      HL,DSPXY+1
0EA7 35                  DEC     (HL)
0EA8 18C3                JR      SCROL
0EAA             ;
0EAA             ;    CONTROL CODE TABLE
0EAA             ;
0EAA             CTBL:   ENT
0EAA 6D0E                DEFW    SCROL          ; SCROLLING
0EAC F80D                DEFW    CURSD          ; CURSOR
0EAE 050E                DEFW    CURSU
0EB0 0D0E                DEFW    CURSR
0EB2 250E                DEFW    CURSL
0EB4 400E                DEFW    HOME
0EB6 3A0E                DEFW    CLRS
0EB8 F80E                DEFW    DEL
0EBA 380F                DEFW    INST
0EBC E10E                DEFW    ALPHA
0EBE EE0E                DEFW    KANA
0EC0 E50E                DEFW    ?RSTR
0EC2 E50E                DEFW    ?RSTR
0EC4 5A0E                DEFW    CR
0EC6 E50E                DEFW    ?RSTR
0EC8 E50E                DEFW    ?RSTR
0ECA             ;
0ECA             ;
0ECA             ;
0ECA             ;    INST BYPASS
0ECA             ;
0ECA CBDC        INST2:  SET     3,H            ; COLOR RAM
0ECC 7E                  LD      A,(HL)         ; FROM
0ECD 23                  INC     HL
0ECE 77                  LD      (HL),A         ; TO
0ECF 2B                  DEC     HL             ; ADR ADJ.
0ED0 CB9C                RES     3,H
0ED2 EDA8                LDD                    ; CHA. TRNS.
0ED4 79                  LD      A,C
0ED5 B0                  OR      B              ; BC=0 ?
0ED6 20F2                JR      NZ,INST2
0ED8 EB                  EX      DE,HL
```

```
0ED9 3600                   LD      (HL),0
0EDB CBDC                   SET     3,H              ; COLOR RAM
0EDD 3671                   LD      (HL),71H
0EDF 1804                   JR      ?RSTR
0EE1                ;
0EE1                ;
0EE1                ;
0EE1                ;
0EE1                ;ORG 0EE1H;ALPHA
0EE1                ;
0EE1                ALPHA:  ENT
0EE1 AF                     XOR     A
0EE2                ALPH1:  ENT
0EE2 327011                 LD      (KANAF),A
0EE5                ;
0EE5                ;
0EE5                ;  RESTORE  ;
0EE5                ;
0EE5                ?RSTR:  ENT
0EE5 E1                     POP     HL
0EE6                ?RSTR1: ENT
0EE6 D1                     POP     DE
0EE7 C1                     POP     BC
0EE8 F1                     POP     AF
0EE9 C9                     RET
0EEA                ;
0EEA                ;  MONITOR WORK AREA  ;
0EEA                ;
D000  P             SCRN:   EQU     D000H
E003  P             KANST:  EQU     E003H            ; KANA STATUS PORT
0EEA                ;
0EEA                ;
0EEA                ;
0EEA                        DEFS    +4
0EEE                ;ORG 0EEEH;KANA
0EEE                ;
0EEE                KANA:   ENT
0EEE CDD40D                 CALL    GRSTAS
0EF1 CAB90D                 JP      Z,DSP01          ; NOT GRAPH KEY THEN JUM
P
0EF4 3E01                   LD      A,+1
0EF6 18EA                   JR      ALPH1
0EF8                ;
0EF8                ;
0EF8                DEL:    ENT
0EF8 EB                     EX      DE,HL            ; LD HL,(DSPXY)
0EF9 7C                     LD      A,H              ; HOME ?
0EFA B5                     OR      L
0EFB 28E8                   JR      Z,?RSTR
0EFD 7D                     LD      A,L
0EFE B7                     OR      A
0EFF 200D                   JR      NZ,DEL1          ; LEFT SIDE ?
0F01 CDF302                 CALL    .MANG
0F04 3808                   JR      C,DEL1
0F06 CDB10F                 CALL    ?PONT
0F09 2B                     DEC     HL
0F0A 3600                   LD      (HL),+0
0F0C 1825                   JR      INST-5           ; JP CURSL
0F0E                DEL1:   ENT
0F0E CDF302                 CALL    .MANG
0F11 0F                     RRCA
```

```
0F12 3E28                   LD      A,40
0F14 3001                   JR      NC,+3
0F16 07                     RLCA                     ; ACC=80
0F17 95                     SUB     L
0F18 47                     LD      B,A              ; TRNS. BYTE
0F19 CDB10F                 CALL    ?PONT
0F1C 7E             DEL2:   LD      A,(HL)           ; CHA. FROM ADR
0F1D 2B                     DEC     HL
0F1E 77                     LD      (HL),A           ; TO
0F1F 23                     INC     HL
0F20 CBDC                   SET     3,H              ; COLOR RAM
0F22 7E                     LD      A,(HL)
0F23 2B                     DEC     HL
0F24 77                     LD      (HL),A
0F25 CB9C                   RES     3,H              ; CHA.
0F27 23                     INC     HL
0F28 23                     INC     HL               ; NEXT
0F29 10F1                   DJNZ    DEL2
0F2B 2B                     DEC     HL               ; ADR.ADJUST
0F2C 3600                   LD      (HL),0
0F2E CBDC                   SET     3,H
0F30 217100                 LD      HL,71H           ; BLUE + WHITE
0F33 3EC4                   LD      A,C4H            ; JP CURSL
0F35 C3E00D                 JP      ?DPCT+4
0F38                ;
0F38                INST:   ENT
0F38 CDF302                 CALL    .MANG
0F3B 0F                     RRCA
0F3C 2E27                   LD      L,+39
0F3E 7D                     LD      A,L
0F3F 3001                   JR      NC,+3
0F41 24                     INC     H
0F42 CDB40F                 CALL    ?PNT1
0F45 E5                     PUSH    HL
0F46 2A7111                 LD      HL,(DSPXY)
0F49 3002                   JR      NC,+4
0F4B 3E4F                   LD      A,+79
0F4D 95                     SUB     L
0F4E 0600                   LD      B,0
0F50 4F                     LD      C,A
0F51 D1                     POP     DE
0F52 2891                   JR      Z,?RSTR
0F54 1A                     LD      A,(DE)
0F55 B7                     OR      A
0F56 208D                   JR      NZ,?RSTR
0F58 62                     LD      H,D              ; HLDE
0F59 6B                     LD      L,E
0F5A 2B                     DEC     HL
0F5B C3CA0E                 JP      INST2            ; JUMP NEXT (BYPASS)
0F5E                ;
0F5E                ;
0F5E                ;   PROGRAM SAVE
0F5E                ;
0F5E                ;     CMD. 'S'
0F5E                ;
0F5E                SAVE:   ENT
0F5E CD3D01                 CALL    HEXIY            ; START ADR.
0F61 220411                 LD      (DTADR),HL       ; DATA ADR. BUFFER
0F64 44                     LD      B,H
0F65 4D                     LD      C,L
```

```
0F66 CDA602           CALL   .4DE
0F69 CD3D01           CALL   HEXIY          ; END ADR.
0F6C ED42             SBC    HL,BC          ; BYTE SIZE
0F6E 23               INC    HL
0F6F 220211           LD     (SIZE),HL      ; BYTE SIZE BUFFER
0F72 CDA602           CALL   .4DE
0F75 CD3D01           CALL   HEXIY          ; EXECUTE ADR.
0F78 220611           LD     (EXADR),HL     ; BUFFER
0F7B CD0900           CALL   NL
0F7E 118B09           LD     DE,MSGSV       ; SAVED FILENAME
0F81 DF               RST    3              ; CALL MSGX
0F82 CD2F01           CALL   BGETL          ; FILENAME INPUT
0F85 CDA602           CALL   .4DE
0F88 CDA602           CALL   .4DE
0F8B 21F110           LD     HL,NAME        ; NAME BUFFER
0F8E          SAV1:   ENT
0F8E 13               INC    DE
0F8F 1A               LD     A,(DE)
0F90 77               LD     (HL),A         ; FILENAME TRANS.
0F91 23               INC    HL
0F92 FE0D             CP     0DH            ; END CODE
0F94 20F8             JR     NZ,SAV1
0F96 3E01             LD     A,01H          ; ATTRIBUE:OBJ.
0F98 32F010           LD     (ATRB),A
0F9B CD3604           CALL   ?WRI
0F9E DA0701           JP     C,?ER          ; WRITE ERROR
0FA1 CD7504           CALL   ?WRD           ; DATA
0FA4 DA0701           JP     C,?ER
0FA7 CD0900           CALL   NL
0FAA 114209           LD     DE,MSGOK       ; OK MESSAGE
0FAD DF               RST    3              ; CALL MSGX
0FAE C3AD00           JP     ST1
0FB1          ;
0FB1          ;
0FB1          ;ORG 0FB1H;?PONT
0FB1          ;
0FB1          ;
0FB1          ;  COMPUTE POINT ADR .  ;
0FB1          ;
0FB1          ;    HL = SCREEN CORDINATE
0FB1          ;    EXIT
0FB1          ;    HL = POINT ADR. ON SCREEN
0FB1          ;
0FB1          ?PONT:  ENT
0FB1 2A7111           LD     HL,(DSPXY)
0FB4          ;
0FB4          ;ORG 0FB4H;?PNT1
0FB4          ;
0FB4          ?PNT1:  ENT
0FB4 F5               PUSH   AF
0FB5 C5               PUSH   BC
0FB6 D5               PUSH   DE
0FB7 E5               PUSH   HL
0FB8 C1               POP    BC
0FB9 112800           LD     DE,0028H       ; 40
0FBC 21D8CF           LD     HL,SCRN-40
0FBF          ?PNT2:  ENT
0FBF 19               ADD    HL,DE
0FC0 05               DEC    B
0FC1 F2BF0F           JP     P,-2
```

```
0FC4 0600             LD     B,+0
0FC6 09               ADD    HL,BC
0FC7 D1               POP    DE
0FC8 C1               POP    BC
0FC9 F1               POP    AF
0FCA C9               RET
0FCB          ;
0FCB          ;   VERIFYING
0FCB          ;
0FCB          ;     COMMAND  'V'
0FCB          ;
0FCB          VRFY:   ENT
0FCB CD8805           CALL   ?VRFY
0FCE DA0701           JP     C,?ER
0FD1 114209           LD     DE,MSGOK
0FD4 DF               RST    3
0FD5 C3AD00           JP     ST1
0FD8          ;
0FD8          ;
0FD8          ;
0FD8          ;ORG 0FD8H;?CLER
0FD8          ;
0FD8          ;
0FD8          ;  CLER  ;
0FD8          ;    B=SIZE
0FD8          ;    HL=LOW ADR.
0FD8          ;
0FD8          ?CLER:  ENT
0FD8 AF               XOR    A
0FD9 1802             JR     +4
0FDB          ?CLRFF: ENT
0FDB 3EFF             LD     A,FFH
0FDD          ?DINT:  ENT
0FDD 77               LD     (HL),A
0FDE 23               INC    HL
0FDF 10FC             DJNZ   -2
0FE1 C9               RET
0FE2          ;
0FE2          ;
0FE2          ;   GAP CHECK
0FE2          ;
0FE2          GAPCK:  ENT
0FE2 C5               PUSH   BC
0FE3 D5               PUSH   DE
0FE4 E5               PUSH   HL
0FE5 0101E0           LD     BC,KEYPB
0FE8 1102E0           LD     DE,CSTR
0FEB          GAPCK1: ENT
0FEB 2664             LD     H,100
0FED          GAPCK2: ENT
0FED CD0106           CALL   EDGE
0FF0 380B             JR     C,GAPCK3
0FF2 CD4A0A           CALL   DLY3           ; CALL DLY2*3
0FF5 1A               LD     A,(DE)
0FF6 E620             AND    20H
0FF8 20F1             JR     NZ,GAPCK1
0FFA 25               DEC    H
0FFB 20F0             JR     NZ,GAPCK2
0FFD          GAPCK3: ENT
0FFD C39B06           JP     RET3
```

```
0F66 CDA602             CALL    .4DE
0F69 CD3D01             CALL    HEXIY           ; END ADR.
0F6C ED42               SBC     HL,BC           ; BYTE SIZE
0F6E 23                 INC     HL
0F6F 220211             LD      (SIZE),HL       ; BYTE SIZE BUFFER
0F72 CDA602             CALL    .4DE
0F75 CD3D01             CALL    HEXIY           ; EXECUTE ADR.
0F78 220611             LD      (EXADR),HL      ; BUFFER
0F7B CD0900             CALL    NL
0F7E 118B09             LD      DE,MSGSV        ; SAVED FILENAME
0F81 DF                 RST     3               ; CALL MSGX
0F82 CD2F01             CALL    BGETL           ; FILENAME INPUT
0F85 CDA602             CALL    .4DE
0F88 CDA602             CALL    .4DE
0F8B 21F110             LD      HL,NAME         ; NAME BUFFER
0F8E            SAV1:   ENT
0F8E 13                 INC     DE
0F8F 1A                 LD      A,(DE)
0F90 77                 LD      (HL),A          ; FILENAME TRANS.
0F91 23                 INC     HL
0F92 FE0D               CP      0DH             ; END CODE
0F94 20F8               JR      NZ,SAV1
0F96 3E01               LD      A,01H           ; ATTRIBUE:OBJ.
0F98 32F010             LD      (ATRB),A
0F9B CD3604             CALL    ?WRI
0F9E DA0701             JP      C,?ER           ; WRITE ERROR
0FA1 CD7504             CALL    ?WRD            ; DATA
0FA4 DA0701             JP      C,?ER
0FA7 CD0900             CALL    NL
0FAA 114209             LD      DE,MSGOK        ; OK MESSAGE
0FAD DF                 RST     3               ; CALL MSGX
0FAE C3AD00             JP      ST1
0FB1            ;
0FB1            ;
0FB1            ;ORG 0FB1H;?PONT
0FB1            ;
0FB1            ;
0FB1            ;  COMPUTE POINT ADR .  ;
0FB1            ;
0FB1            ;    HL = SCREEN CORDINATE
0FB1            ;    EXIT
0FB1            ;    HL = POINT ADR. ON SCREEN
0FB1            ;
0FB1            ?PONT:  ENT
0FB1 2A7111             LD      HL,(DSPXY)
0FB4            ;
0FB4            ;ORG 0FB4H;?PNT1
0FB4            ;
0FB4            ?PNT1:  ENT
0FB4 F5                 PUSH    AF
0FB5 C5                 PUSH    BC
0FB6 D5                 PUSH    DE
0FB7 E5                 PUSH    HL
0FB8 C1                 POP     BC
0FB9 112800             LD      DE,0028H        ; 40
0FBC 21D8CF             LD      HL,SCRN-40
0FBF            ?PNT2:  ENT
0FBF 19                 ADD     HL,DE
0FC0 05                 DEC     B
0FC1 F2BF0F             JP      P,-2
```

```
0FC4 0600               LD      B,+0
0FC6 09                 ADD     HL,BC
0FC7 D1                 POP     DE
0FC8 C1                 POP     BC
0FC9 F1                 POP     AF
0FCA C9                 RET
0FCB            ;
0FCB            ;   VERIFYING
0FCB            ;
0FCB            ;     COMMAND  'V'
0FCB            ;
0FCB            VRFY:   ENT
0FCB CD8805             CALL    ?VRFY
0FCE DA0701             JP      C,?ER
0FD1 114209             LD      DE,MSGOK
0FD4 DF                 RST     3
0FD5 C3AD00             JP      ST1
0FD8            ;
0FD8            ;
0FD8            ;
0FD8            ;ORG 0FD8H;?CLER
0FD8            ;
0FD8            ;
0FD8            ;  CLER  ;
0FD8            ;    B=SIZE
0FD8            ;    HL=LOW ADR.
0FD8            ;
0FD8            ?CLER:  ENT
0FD8 AF                 XOR     A
0FD9 1802               JR      +4
0FDB            ?CLRFF: ENT
0FDB 3EFF               LD      A,FFH
0FDD            ?DINT:  ENT
0FDD 77                 LD      (HL),A
0FDE 23                 INC     HL
0FDF 10FC               DJNZ    -2
0FE1 C9                 RET
0FE2            ;
0FE2            ;
0FE2            ;   GAP CHECK
0FE2            ;
0FE2            GAPCK:  ENT
0FE2 C5                 PUSH    BC
0FE3 D5                 PUSH    DE
0FE4 E5                 PUSH    HL
0FE5 0101E0             LD      BC,KEYPB
0FE8 1102E0             LD      DE,CSTR
0FEB            GAPCK1: ENT
0FEB 2664               LD      H,100
0FED            GAPCK2: ENT
0FED CD0106             CALL    EDGE
0FF0 380B               JR      C,GAPCK3
0FF2 CD4A0A             CALL    DLY3                    ; CALL DLY2*3
0FF5 1A                 LD      A,(DE)
0FF6 E620               AND     20H
0FF8 20F1               JR      NZ,GAPCK1
0FFA 25                 DEC     H
0FFB 20F0               JR      NZ,GAPCK2
0FFD            GAPCK3: ENT
0FFD C39B06             JP      RET3
```

```
  **  Z80 ASSEMBLER 2Z-004C  <1Z-013B>  PAGE 63          03.06.85

01 11A3                    BUFER:  ENT                   ; GET LINE BUFFER
02 11A3                            DEFS    +81
03 11F4                    ;
04 11F4                    ;
05 11F4                    ;       EQU TABLE I/O PORT
06 11F4                    ;
07 11F4                    ;
08 E000   P                KEYPA:  EQU     E000H
09 E001   P                KEYPB:  EQU     E001H
10 E002   P                KEYPC:  EQU     E002H
11 E003   P                KEYPF:  EQU     E003H
12 E002   P                CSTR:   EQU     E002H
13 E003   P                CSTPT:  EQU     E003H
14 E004   P                CONT0:  EQU     E004H
15 E005   P                CONT1:  EQU     E005H
16 E006   P                CONT2:  EQU     E006H
17 E007   P                CONTF:  EQU     E007H
18 E008   P                SUNDG:  EQU     E008H
19 E008   P                TEMP:   EQU     E008H
20 11F4                    ;
21 11F4                            END
```

```
  **  Z80 ASSEMBLER 2Z-004C  <1Z-013B>  PAGE 64          03.06.85

#BRK    08B8  #CLR08 09D4  #CLR8   09D5  $MCP    006B  ..LPT   017B
.4DE    02A6  .LPT    0176  .MANG   02F3  2HE1    0434  2HEX    041F
??KEY   09B3  ?ADCN   0BB9  ?BEL    0577  ?BELD   0352  ?BLNK   0DA6
?BRK    0A32  ?BRK1   0A48  ?BRK2   0980  ?BRK3   0986  ?CLER   0FD8
?CLRFF  0FDB  ?DACN   0BCE  ?DINT   0FDD  ?DPCT   0DDC  ?DSP    0DB5
?ER     0107  ?FLAS   09FF  ?FLS    09E3  ?GET    08BD  ?GETL   07E6
?KEY    08CA  ?KY1    08D6  ?KY2    08DA  ?KY5    08FA  ?KY55   08FB
?KYGRP  08FE  ?KYGRS  0909  ?KYSM   08B3  ?LOAD   05F0  ?LTNL   090E
?MLDY   01C7  ?MODE   073E  ?MSG    0893  ?MSGX   08A1  ?NL     0918
?PNT1   0FB4  ?PNT2   0FBF  ?PONT   0FB1  ?PRNT   0935  ?PRT    0946
?PRTS   0920  ?PRTT   0924  ?RDD    04F8  ?RDI    04D8  ?RSTR   0EE5
?RSTR1  0EE6  ?SAVE   0B92  ?SWEP   0A50  ?TEMP   02E5  ?TMR1   0375
?TMR2   037F  ?TMRD   0358  ?TMS1   0331  ?TMS2   0344  ?TMST   0308
?VRFY   0588  ?WRD    0475  ?WRI    0436  ALPH1   0EE2  ALPHA   0EE1
AMPM    119B  ASC     03DA  ATBL    0A92  ATRB    10F0  AUTO3   07ED
BELL    003E  BGETL   012F  BRKEY   001E  BUFER   11A3  CKS1    0720
CKS2    072F  CKS3    0733  CKSUM   071A  CLEAR   09D8  CLEAR1  09DA
CLRS    0E3A  CMY0    005B  COMNT   1108  CONT0   E004  CONT1   E005
CONT2   E006  CONTF   E007  CR      0E5A  CR1     0E6A  CSMDT   1199
CSTPT   E003  CSTR    E002  CTBL    0EAA  CURS1   0DFF  CURS2   0E16
CURS3   0DFF  CURS4   0E23  CURS5   0E02  CURSD   0DF8  CURSL   0E25
CURSR   0E0D  CURSU   0E05  CURSU1  0E0B  DACN1   0BE3  DACN2   0BDF
DACN3   0BE0  DEL     0EF8  DEL1    0F0E  DEL2    0F1C  DLY1    0759
DLY12   0996  DLY2    0760  DLY3    0A4A  DLY4    09A9  DPRNT   1194
DSP01   0DB9  DSP04   0DD0  DSPXY   1171  DSWEP   0830  DTADR   1104
DUM1    0D88  DUM2    0D3E  DUM3    0D37  DUMP    0D29  EDG1    0607
EDG2    0613  EDGE    0601  EXADR   1106  FD      00FF  FD1     0106
FD2     0102  FLAS1   097B  FLAS2   09EF  FLAS3   09F3  FLASH   118E
FLKEY   057E  FLPST   118F  FLSDT   1192  FLSST   1191  GAP     077A
GAP1    078E  GAP2    0796  GAP3    079C  GAPCK   0FE2  GAPCK1  0FEB
GAPCK2  0FED  GAPCK3  0FFD  GETKY   001B  GETL    0003  GETL1   07EA
GETL2   0818  GETL3   085B  GETL5   081D  GETL6   0865  GETLA   082B
GETLB   0863  GETLC   0822  GETLR   087E  GETLU   0876  GETLZ   086C
GOTO    00F3  GRSTAS  0DD4  HEX     03F9  HEXIY   013D  HEXJ    03E5
HL1     041D  HLHEX   0410  HOME    0E4D  IBUFE   10F0  INST    0F38
INST2   0ECA  KANA    0EEE  KANAF   1170  KANST   E003  KEYPA   E000
KEYPB   E001  KEYPC   E002  KEYPF   E003  KSL1    09B7  KSL2    09BC
KTBL    0BEA  KTBLC   0CAA  KTBLG   0CE9  KTBLGS  0C6A  KTBLS   0C2A
LETNL   0006  LLPT    0470  LOA0    0116  LOAD    0111  LONG    0A1A
LPRNT   018F  M#TBL   0284  MANG    1173  MCOR    07A8  MCR1    07AB
MCR2    07D4  MCR3    07D7  MELDY   0030  MLD1    01D1  MLD2    0205
MLD3    020D  MLD4    0211  MLD5    0214  MLDS1   02C4  MLDSP   02BE
MLDST   02AB  MONIT   0000  MOT1    06A4  MOT2    06AB  MOT4    06B9
MOT5    06D8  MOT7    06B7  MOT8    06D0  MOT9    06D7  MOTOR   069F
MSG     0015  MSG#1   03FB  MSG#2   03FD  MSG#3   0402  MSG#7   0467
MSG1    0896  MSG?2   09A0  MSG?3   06E7  MSGE1   0147  MSGOK   0942
MSGSV   098B  MSGX    0018  MSGX1   08A4  MSGX2   08A7  MST1    0705
MST2    070C  MST3    0717  MSTA    0044  MSTOP   0700  MSTP    0047
MTBL    026C  NAME    10F1  NL      0009  NLPHL   05FA  NOADD   03E2
OCTV    11A0  ONP1    021F  ONP2    022C  ONP3    0265  ONPU    021C
ONTYO   119F  OPTBL   029C  PEN     018B  PLOT    0184  PMSG    01A5
PMSG1   01A8  PRNT    0012  PRNT2   0967  PRNT3   096C  PRNT4   096F
PRNT5   0959  PRNTS   000C  PRNTT   000F  PRTHL   03BA  PRTHX   03C3
PTEST   0155  PTRN    0180  PTST0   015A  PTST1   0170  RATIO   11A1
RBY1    0630  RBY2    0649  RBY3    0654  RBYTE   0624  RD1     04E6
RDA     01B6  RDDAT   002A  RDINF   0027  RET1    04D2  RET2    0554
RET3    069B  RTAPE   050E  RTP1    0513  RTP2    0519  RTP3    0532
RTP4    0554  RTP5    0565  RTP6    0572  RTP7    056E  RTP8    0553
RTP9    0574  RYTHM   02C8  SAV1    0F8E  SAVE    0F5E  SCRN    D000
SCROL   0E6D  SG      00F7  SHORT   0A01  SIZE    1102  SLPT    03D5
```

```
    **  Z80 ASSEMBLER 2Z-004C  <1Z-013B>  PAGE 65           03.06.85

SP      10F0  SPHEX  03B1  SS     00A2  ST0    0070  ST1    00AD
ST2     00BB  START  004A  STRGF  1193  SUMDT  1197  SUNDG  E008
SV0     0BA2  SV1    0BB5  SWEP0  0A66  SWEP01 0A64  SWEP2  0A7F
SWEP3   0A77  SWEP6  0A5F  SWEP9  0A73  SWRK   119D  TEMP   E008
TEMPW   119E  TIMFG  119C  TIMIN  038D  TIMRD  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  WBY1   076D
WBYTE   0767  WRDAT  0024  WRI1   0444  WRI2   045E  WRI3   0464
WRINF   0021  WTAP1  0494  WTAP2  04A5  WTAP3  04D2  WTAPE  048A
XTEMP   0041
```

3-2 MZ Disk control

```
  **  Z80 ASSEMBLER 2Z-004C  <9Z-504M-V1.0C>  PAGE 01              03.06.85

01 E010                         ORG     E010H
02 E010                  ;
03 E010                  ;==============================
04 E010                  ;
05 E010                  ;       Equate file
06 E010                  ;          for MZ-800
07 E010                  ;          1984.09.01
08 E010                  ;
09 E010                  ;==============================
10 E010                  ;
11 E010                  ;
12 E010                  ;
13 10F0   P              SP:     EQU     10F0H
14 003E   P              BELL:   EQU     003EH
15 0006   P              LETNL:  EQU     0006H
16 000C   P              PRNTS:  EQU     000CH
17 0012   P              PRNT:   EQU     0012H
18 0015   P              MSG:    EQU     0015H
19 001E   P              BRKEY:  EQU     001EH
20 0003   P              GETL:   EQU     0003H
21 001B   P              GETKY:  EQU     001BH
22 0DDC   P              ?DPCT:  EQU     0DDCH
23 0BB9   P              ?ADCN:  EQU     0BB9H
24 0009   P              NL:     EQU     0009H
25 03BA   P              PRTHL:  EQU     03BAH
26 03DA   P              ASC:    EQU     03DAH
27 0410   P              HLHEX:  EQU     0410H
28 041F   P              2HEX:   EQU     041FH
29 04D8   P              ?RDI:   EQU     04D8H
30 10F1   P              NAME:   EQU     10F1H
31 04F8   P              ?RDD:   EQU     04F8H
32 10F0   P              ATRB:   EQU     10F0H
33 1102   P              SIZE:   EQU     1102H
34 1104   P              DTADR:  EQU     1104H
35 1106   P              EXADR:  EQU     1106H
36 0027   P              RDINF:  EQU     0027H
37 002A   P              RDDAT:  EQU     002AH
38 0470   P              LLPT:   EQU     0470H
39 03D5   P              SLPT:   EQU     03D5H
40 05FA   P              NLPHL:  EQU     05FAH
41 03B1   P              SPHEX:  EQU     03B1H
42 0920   P              ?PRTS:  EQU     0920H
43 012F   P              BGETL:  EQU     012FH
44 02A6   P              .4DE:   EQU     02A6H
45 00F3   P              GOTO:   EQU     00F3H
46 0436   P              ?WRI:   EQU     0436H
47 0475   P              ?WRD:   EQU     0475H
48 0942   P              MSGOK:  EQU     0942H
49 0588   P              ?VRFY:  EQU     0588H
50 005B   P              CMY0:   EQU     005BH
51 119D   P              SWRK:   EQU     119DH
52 1171   P              DSPXY:  EQU     1171H
53 E000   P              KEYPA:  EQU     E000H
54 E001   P              KEYPB:  EQU     E001H
55 08CA   P              ?KEY:   EQU     08CAH
56 0A32   P              ?BRK:   EQU     0A32H
57 0147   P              MSGE1:  EQU     0147H
58 096C   P              PRNT3:  EQU     096CH
59 0DA6   P              ?BLNK:  EQU     0DA6H
60 06E7   P              MSG?3:  EQU     06E7H
```

```
  **  Z80 ASSEMBLER 2Z-004C  <9Z-504M-V1.0C>  PAGE 02              03.06.85

01 001E   P              BRKCK:  EQU     001EH
02 0FD8   P              ?CLER:  EQU     0FD8H
03 038D   P              TIMIN:  EQU     038DH
04 119E   P              TEMPW:  EQU     119EH
05 02BE   P              MLDSP:  EQU     02BEH
06 0577   P              ?BEL:   EQU     0577H
07 073E   P              ?MODE:  EQU     073EH
08 09D5   P              #CLR8:  EQU     09D5H
09 0308   P              ?TMST:  EQU     0308H
10 E010                  ;
11 E010                  ;
12 E010                  ; Quick disk equ table
13 E010                  ;
14 E010                  ;
15 00F4   P              SIOAD:  EQU     F4H            ; Sio A Data
16 00F5   P              SIOBD:  EQU     F5H            ; Sio B Data
17 00F6   P              SIOAC:  EQU     F6H            ; Sio A Control
18 00F7   P              SIOBC:  EQU     F7H            ; Sio B Control
19 1130   P              QDTBL:  EQU     1130H          ; Iocs parameter adrs
20 1130   P              QDPA:   EQU     QDTBL          ; Iocs command
21 1131   P              QDPB:   EQU     QDPA+1         ; Iocs parameter
22 1132   P              QDPC:   EQU     QDPB+1         ; Data head adrs(1)
23 1134   P              QDPE:   EQU     QDPC+2         ; Data byte size(1)
24 1136   P              QDPG:   EQU     QDPE+2         ; Data head adrs(2)
25 1138   P              QDPI:   EQU     QDPG+2         ; Data byte size(2)
26 113A   P              QDCPA:  EQU     QDPI+2         ; QD Flag byte (A)
27 113B   P              QDCPB:  EQU     QDCPA+1        ; QD Flag byte (B)
28 113C   P              QDCPC:  EQU     QDCPB+1        ; QD Flag byte (C)
29 113D   P              HDPT:   EQU     QDCPC+1        ; Iocs work
30 113E   P              HDPT0:  EQU     HDPT+1
31 113F   P              FNUPS:  EQU     HDPT0+1
32 1140   P              FNUPS1: EQU     FNUPS+1
33 1141   P              FNUPF:  EQU     FNUPS1+1
34 1142   P              FNA:    EQU     FNUPF+1
35 1143   P              FNB:    EQU     FNA+1
36 1144   P              MTF:    EQU     FNB+1
37 1145   P              RTYF:   EQU     MTF+1
38 1146   P              BFRF:   EQU     RTYF+1
39 1147   P              SYNCF:  EQU     BFRF+1
40 1148   P              RETSP:  EQU     SYNCF+1
41 1104   P              QSIZE:  EQU     1104H          ; QD File size
42 1106   P              QDTADR: EQU     1106H          ; QD Data adrs
43 1108   P              QEXADR: EQU     1108H          ; QD Exec adrs
44 0011   P              NAMSIZ: EQU     0011H          ; Name size
45 CD90   P              QDIRBF: EQU     CD90H          ; QD dir buffer
46 0001   P              OBJCD:  EQU     01H            ; Obj code
47 0002   P              BTXCD:  EQU     02H            ; Btx code
48 0003   P              BSDCD:  EQU     03H            ; Bsd code
49 0000   P              BRKCD:  EQU     00H            ; Break code
50 0028   P              NTFECD: EQU     40             ; Not found err code
51 0029   P              HDERCD: EQU     41             ; Hard err code
52 002A   P              ALEXCD: EQU     42             ; Already exist code
53 002E   P              WPRTCD: EQU     46             ; Write protect err code
54 0032   P              QNTRCD: EQU     50             ; Not ready
55 0033   P              TMFECD: EQU     51             ; Too many files err
56 0035   P              NFSECD: EQU     53             ; No file space err code
57 0036   P              UNFMCD: EQU     54             ; Unformat err code
58 0039   P              BDSKCD: EQU     57             ; Bad disk err code
59 E010                  ;
60 E010                          SKP     H
```

```
E010                  ;
E010                  ;
E010                  ;
E010                  ;    Quick-Disk(BISYNC)
E010                  ;         Control Program
E010                  ;
E010                  ;         V1.0A 04.25.1984
E010                  ;
E010                  ;
E010                  ;
E010                  ;
E010                  ;
E010                  ;
EFFF  P               FMS:    EQU    EFFFH            ;INIT CHECK SIZE (60KByt
e)
E010                  ;
E010                  ;
E010                  ;
E010                  ;
E010                  ;
E010                  ;          QDIO
E010                  ;
E010                  ;
E010                  ;
E010                  ;
E010                  QDIOS:  ENT
E010 3E05                     LD     A,5              ;Retry 4
E012 324511                   LD     (RTYF),A
E015                  ;
E015 F3               RTY:    DI
E016 CD6AE0                   CALL   QMEIN
E019 D0                       RET    NC
E01A F5                       PUSH   AF
E01B FE28                     CP     40
E01D 2849                     JR     Z,RTY4
E01F CDE8E2                   CALL   MTOF
E022 3A3011                   LD     A,(QDPA)
E025 FE04                     CP     4                ;Write ?
E027 201C                     JR     NZ,RTY3
E029 3A4111                   LD     A,(FNUPF)
E02C B7                       OR     A
E02D 2816                     JR     Z,RTY3
E02F AF                       XOR    A                ;FNUPF CLR
E030 324111                   LD     (FNUPF),A
E033 3A4211                   LD     A,(FNA)
E036 E5                       PUSH   HL               ;RETSP <= SP-2
E037 ED734811                 LD     (RETSP),SP
E03B E1                       POP    HL
E03C                  ;
E03C F3                       DI
E03D CD68E2                   CALL   QDSVFN
E040 3820                     JR     C,RTY2
E042 CDE8E2                   CALL   MTOF
E045                  ;
E045 FE03             RTY3:   CP     3
E047 2004                     JR     NZ,RTY5
E049 213D11                   LD     HL,HDPT
E04C 35                       DEC    (HL)
E04D F1               RTY5:   POP    AF
E04E F5                       PUSH   AF
E04F FE29                     CP     41
```

```
E051 200F                     JR     NZ,RTY2
E053                  ;
E053 214511                   LD     HL,RTYF
E056 35                       DEC    (HL)
E057 2809                     JR     Z,RTY2
E059 F1                       POP    AF
E05A 3A4011                   LD     A,(FNUPS1)
E05D 323F11                   LD     (FNUPS),A
E060 18B3                     JR     RTY
E062                  ;
E062 CDDBE1           RTY2:   CALL   WRCAN
E065 CD83E0                   CALL   QDHPC
E068 F1               RTY4:   POP    AF
E069 C9                       RET
E06A                  ;
E06A                          SKP    H
```

```
E06A                ;
E06A ED734811       QMEIN:  LD      (RETSP),SP
E06E 3A3011                 LD      A,(QDPA)
E071 3D                     DEC     A
E072 2816                   JR      Z,QDRC          ;Ready Check
E074 3D                     DEC     A
E075 2819                   JR      Z,QDFM          ;Format
E077 3D                     DEC     A
E078 2860                   JR      Z,QDRD          ;Read
E07A 3D                     DEC     A
E07B CA4EE1                 JP      Z,QDWR          ;Write
E07E 3D                     DEC     A
E07F 2802                   JR      Z,QDHPC         ;Head Point Clear
E081 1850                   JR      MTOFX           ;Motor Off
E083                ;
E083                ;==========================
E083                ;
E083                ;       Head Point Clear
E083                ;
E083                ;==========================
E083                ;
E083                QDHPC:  ENT
E083 F5                     PUSH    AF
E084 AF                     XOR     A
E085 323D11                 LD      (HDPT),A
E088 F1                     POP     AF
E089 C9                     RET
E08A                ;
E08A                ;==========================
E08A                ;
E08A                ;       Ready Check
E08A                ;
E08A                ;==========================
E08A                ;
E08A                QDRC:   ENT
E08A 3A3111                 LD      A,(QDPB)
E08D C33CE2                 JP      QREDY
E090                ;
E090                        SKP     H
```

```
E090                ;
E090                ;==========================
E090                ;
E090                ;       Format
E090                ;
E090                ;==========================
E090                ;
E090                QDFM:   ENT
E090 AF                     XOR     A
E091 CD68E2                 CALL    QDSVFN
E094 CD65E3                 CALL    SYNCS2
E097 01FFEF                 LD      BC,FMS
E09A 3EAA                   LD      A,AAH
E09C                ;
E09C 2F             QDFM1:  CPL
E09D 57                     LD      D,A
E09E CDDBE3                 CALL    TRANS
E0A1 0B                     DEC     BC
E0A2 78                     LD      A,B
E0A3 B1                     OR      C
E0A4 2803                   JR      Z,QDFM2
E0A6 7A                     LD      A,D
E0A7 18F3                   JR      QDFM1
E0A9                ;
E0A9 CDB2E3         QDFM2:  CALL    EOM
E0AC CDE8E2                 CALL    MTOF
E0AF CD9BE2                 CALL    MTON
E0B2 3A4311                 LD      A,(FNB)
E0B5 3D                     DEC     A
E0B6 201E                   JR      NZ,FMERR
E0B8 CDFDE2                 CALL    SYNCL2
E0BB 01FFEF                 LD      BC,FMS
E0BE 1E55                   LD      E,55H
E0C0 BB             QDFM3:  CP      E
E0C1 2013                   JR      NZ,FMERR
E0C3 0B                     DEC     BC
E0C4 78                     LD      A,B
E0C5 B1                     OR      C
E0C6 2808                   JR      Z,QDFM4
E0C8 7B                     LD      A,E
E0C9 2F                     CPL
E0CA 5F                     LD      E,A
E0CB CDF0E3                 CALL    RDATA
E0CE 18F0                   JR      QDFM3
E0D0                ;
E0D0 CDC3E3         QDFM4:  CALL    RDCRC
E0D3 C3E8E2         MTOFX:  JP      MTOF
E0D6                ;
E0D6 3E29           FMERR:  LD      A,41            ;Hard Err
E0D8 37                     SCF
E0D9 C9                     RET
E0DA                ;
E0DA                        SKP     H
```

```
E0DA                    ;
E0DA                    ;=========================
E0DA                    ;
E0DA                    ;         Read
E0DA                    ;
E0DA                    ;=========================
E0DA                    ;
E0DA                    QDRD:   ENT
E0DA 3A4411                     LD      A,(MTF)
E0DD B7                         OR      A
E0DE CC9BE2                     CALL    Z,MTON
E0E1 CD14E1                     CALL    HPS
E0E4 D8                         RET     C
E0E5 CD35E4                     CALL    BRKC
E0E8                    ;
E0E8 CDF0E3                     CALL    RDATA
E0EB 4F                         LD      C,A
E0EC CDF0E3                     CALL    RDATA
E0EF 47                         LD      B,A
E0F0 B1                         OR      C
E0F1 CAE5E1                     JP      Z,QDWE1
E0F4 2A3411                     LD      HL,(QDPE)       ;Byte size check
E0F7 ED42                       SBC     HL,BC
E0F9 DAE5E1                     JP      C,QDWE1
E0FC 2A3211                     LD      HL,(QDPC)
E0FF                    ;
E0FF                    ;Block Data Read
E0FF                    ;
E0FF CDF0E3             BDR:    CALL    RDATA
E102 77                         LD      (HL),A
E103 23                         INC     HL
E104 0B                         DEC     BC
E105 78                         LD      A,B
E106 B1                         OR      C
E107 20F6                       JR      NZ,BDR
E109 CDC3E3                     CALL    RDCRC
E10C 3A3111                     LD      A,(QDPB)
E10F CB47                       BIT     0,A
E111 20C0                       JR      NZ,MTOFX
E113 C9                         RET
E114                    ;
E114                            SKP     H
```

```
E114                    ;
E114                    ;Head Point Search
E114                    ;
E114                    HPS:    ENT
E114 214311                     LD      HL,FNB
E117 35                         DEC     (HL)
E118 2830                       JR      Z,HPNFE         ;Not Found
E11A CDFDE2                     CALL    SYNCL2
E11D 4F                         LD      C,A             ;BLKFLG => C reg
E11E 3A3D11                     LD      A,(HDPT)
E121 213E11                     LD      HL,HDPT0
E124 BE                         CP      (HL)            ;Search ok ?
E125 2020                       JR      NZ,HPS1
E127 3C                         INC     A               ;HDPT count up
E128 323D11                     LD      (HDPT),A
E12B 77                         LD      (HL),A          ;HDPT0 count up
E12C 3A3111                     LD      A,(QDPB)
E12F A9                         XOR     C
E130 1F                         RRA
E131 D0                         RET     NC              ;=
E132                    ;
E132                    ;Dummy read
E132                    ;
E132 CDF0E3             DMR:    CALL    RDATA
E135 4F                         LD      C,A
E136 CDF0E3                     CALL    RDATA
E139 47                         LD      B,A
E13A                    ;
E13A CDF0E3             DMR1:   CALL    RDATA
E13D 0B                         DEC     BC
E13E 78                         LD      A,B
E13F B1                         OR      C
E140 20F8                       JR      NZ,DMR1
E142 CDC3E3                     CALL    RDCRC
E145 18CD                       JR      HPS             ;next
E147                    ;
E147 34                 HPS1:   INC     (HL)
E148 18E8                       JR      DMR
E14A                    ;
E14A 3E28               HPNFE:  LD      A,40            ;Not Found
E14C 37                         SCF
E14D C9                         RET
E14E                    ;
E14E                            SKP     H
```

```
E14E                ;
E14E                ;==========================
E14E                ;
E14E                ;          Write
E14E                ;
E14E                ;==========================
E14E                ;
E14E                QDWR:   ENT
E14E 3A3F11                 LD      A,(FNUPS)
E151 324011                 LD      (FNUPS1),A
E154 3A4411                 LD      A,(MTF)
E157 B7                     OR      A
E158 2017                   JR      NZ,QDWR1
E15A CD9BE2                 CALL    MTON
E15D 3A3F11                 LD      A,(FNUPS)
E160 214311                 LD      HL,FNB
E163 86                     ADD     A,(HL)
E164 324311                 LD      (FNB),A
E167 3C                     INC     A
E168 323D11                 LD      (HDPT),A
E16B CD14E1                 CALL    HPS
E16E D2E5E1                 JP      NC,QDWE1          ;Hard err
E171 3A3111         QDWR1:  LD      A,(QDPB)
E174 47                     LD      B,A
E175 E601                   AND     1
E177 200C                   JR      NZ,QDWR2
E179 113211                 LD      DE,QDPC
E17C 78                     LD      A,B
E17D CB97                   RES     2,A
E17F CDEDE1                 CALL    BDW
E182 CD35E4                 CALL    BRKC
E185                ;
E185 113611         QDWR2:  LD      DE,QDPG
E188 3A3111                 LD      A,(QDPB)
E18B CBC7                   SET     0,A
E18D CDEDE1                 CALL    BDW
E190 CDE8E2                 CALL    MTOF
E193 CD35E4                 CALL    BRKC
E196                ;
E196 3A3111                 LD      A,(QDPB)
E199 E604                   AND     4
E19B 284B                   JR      Z,NFNUP
E19D 3A4211                 LD      A,(FNA)
E1A0 213F11                 LD      HL,FNUPS
E1A3 86                     ADD     A,(HL)
E1A4 CD68E2                 CALL    QDSVFN
E1A7 3E01                   LD      A,1
E1A9 324111                 LD      (FNUPF),A
E1AC 3A4211         QDWR4:  LD      A,(FNA)
E1AF 214011                 LD      HL,FNUPS1
E1B2 86                     ADD     A,(HL)
E1B3 3C                     INC     A
E1B4 324311                 LD      (FNB),A
E1B7 3C                     INC     A
E1B8 323D11                 LD      (HDPT),A
E1BB CD14E1                 CALL    HPS
E1BE 3025                   JR      NC,QDWE1
E1C0 3A3111                 LD      A,(QDPB)
E1C3 E601                   AND     1
E1C5 2007                   JR      NZ,QDWR3
```

```
E1C7 113211                 LD      DE,QDPC
E1CA CD1BE2                 CALL    BDV
E1CD D8                     RET     C
E1CE 113611         QDWR3:  LD      DE,QDPG
E1D1 CD1BE2                 CALL    BDV
E1D4 D8                     RET     C
E1D5 3A4111                 LD      A,(FNUPF)
E1D8 B7                     OR      A
E1D9 2807                   JR      Z,QDWR5
E1DB AF             WRCAN:  XOR     A
E1DC 323F11                 LD      (FNUPS),A
E1DF 324111                 LD      (FNUPF),A
E1E2                ;
E1E2 C3E8E2         QDWR5:  JP      MTOF
E1E5                ;
E1E5 C3D6E0         QDWE1:  JP      FMERR
E1E8                ;
E1E8 CD9BE2         NFNUP:  CALL    MTON
E1EB 18BF                   JR      QDWR4
E1ED                ;
E1ED                ;Block Data Write
E1ED                ;
E1ED F5             BDW:    PUSH    AF
E1EE 213F11                 LD      HL,FNUPS
E1F1 34                     INC     (HL)
E1F2 CD65E3                 CALL    SYNCS2
E1F5 F1                     POP     AF
E1F6 CDDBE3                 CALL    TRANS
E1F9 CD11E2                 CALL    RSET
E1FC 79                     LD      A,C
E1FD CDDBE3                 CALL    TRANS
E200 78                     LD      A,B
E201 CDDBE3                 CALL    TRANS
E204                ;
E204 7E             BDW1:   LD      A,(HL)
E205 CDDBE3                 CALL    TRANS
E208 23                     INC     HL
E209 0B                     DEC     BC
E20A 78                     LD      A,B
E20B B1                     OR      C
E20C 20F6                   JR      NZ,BDW1
E20E C3B2E3                 JP      EOM
E211                ;
E211                ;HL,BC SET
E211                ;
E211 EB             RSET:   EX      DE,HL
E212 5E                     LD      E,(HL)
E213 23                     INC     HL
E214 56                     LD      D,(HL)
E215 23                     INC     HL
E216 4E                     LD      C,(HL)
E217 23                     INC     HL
E218 46                     LD      B,(HL)
E219 EB                     EX      DE,HL
E21A C9                     RET
E21B                ;
E21B                ;Block Data Verify
E21B                ;
E21B CDFDE2         BDV:    CALL    SYNCL2
E21E CD11E2                 CALL    RSET
```

```
E221 CDF0E3            CALL    RDATA
E224 B9                CP      C
E225 20BE              JR      NZ,QDWE1
E227 CDF0E3            CALL    RDATA
E22A B8                CP      B
E22B 20B8              JR      NZ,QDWE1
E22D           ;
E22D CDF0E3    BDV1:   CALL    RDATA
E230 BE                CP      (HL)
E231 20B2              JR      NZ,QDWE1
E233 23                INC     HL
E234 0B                DEC     BC
E235 78                LD      A,B
E236 B1                OR      C
E237 20F4              JR      NZ,BDV1
E239 C3C3E3            JP      RDCRC
E23C           ;
E23C                   SKP     H
```

```
E23C            ;
E23C            ;
E23C            ;
E23C            ;
E23C            ; Quick-Disk
E23C            ;      Physical iocs (bisync)
E23C            ;
E23C            ;          By MZ- 700 IPL-ROM
E23C            ;             MZ-1500 IPL-ROM
E23C            ;             MZ- 700 Disk-Basic
E23C            ;             MZ-1500 Basic
E23C            ;
E23C            ;      V2.0A 04.25.1984
E23C            ;
E23C            ;
E23C            ;
E23C            ;
E23C            ; i/o port adrs
E23C            ;
E23C            ;SIOAD:EQU F4H; sio A data
E23C            ;SIOBD:EQU F5H; sio B data
E23C            ;SIOAC:EQU F6H; sio A control
E23C            ;SIOBC:EQU F7H; sio B control
E23C            ;
E23C            ;
E23C            ;
E23C            ; Ready & Write protect
E23C            ;    Acc = '0' : Ready check
E23C            ;    Acc = '1' : & Write Protect
E23C            ;
E23C            QREDY:  ENT
E23C 47                 LD      B,A
E23D 3E02               LD      A,02H           ;SIO hard check
E23F D3F7               OUT     (SIOBC),A
E241 3E81               LD      A,81H
E243 D3F7               OUT     (SIOBC),A
E245 3E02               LD      A,02H
E247 D3F7               OUT     (SIOBC),A
E249 DBF7               IN      A,(SIOBC)
E24B E681               AND     81H
E24D FE81               CP      81H
E24F C206E4             JP      NZ,IOE50        ;Not ready
E252 3E10               LD      A,10H
E254 D3F6               OUT     (SIOAC),A
E256 DBF6               IN      A,(SIOAC)
E258 4F                 LD      C,A
E259 E608               AND     08H
E25B CA06E4             JP      Z,IOE50         ;Not ready
E25E 78                 LD      A,B
E25F B7                 OR      A
E260 C8                 RET     Z               ;No err
E261 79                 LD      A,C
E262 E620               AND     20H
E264 C0                 RET     NZ              ;No err
E265 C303E4             JP      IOE46           ;Write protect
E268            ;
E268                    SKP     H
```

```
E268                     ;
E268                     ;
E268                     ; Write FN
E268                     ;
E268                     QDSVFN: ENT
E268 F5                          PUSH    AF
E269 212CE4                      LD      HL,SIOSD
E26C 0609                        LD      B,09H
E26E CDD8E2                      CALL    LSINT           ;save init
E271                     ;
E271 3E10                SREDY:  LD      A,10H
E273 D3F6                        OUT     (SIOAC),A
E275 DBF6                        IN      A,(SIOAC)
E277 E608                        AND     8
E279 CA06E4                      JP      Z,IOE50         ;Not ready
E27C 3E10                        LD      A,10H
E27E D3F7                        OUT     (SIOBC),A
E280 DBF7                        IN      A,(SIOBC)
E282 E608                        AND     8
E284 28EB                        JR      Z,SREDY
E286 01E900                      LD      BC,00E9H        ;Wait 160ms
E289 CD14E4                      CALL    TIMW
E28C CD9AE3                      CALL    SBRK            ;Send Break
E28F CD7EE3                      CALL    SYNCS1          ;FN Only SYNC
E292 F1                          POP     AF
E293 CDDBE3                      CALL    TRANS           ;FN=A
E296 CDB2E3                      CALL    EOM             ;CRC FLAG(7EH)
E299 1833                        JR      FNEND
E29B                     ;
E29B                     ;
E29B                     ;
E29B                     ; MTON -- QD MOTOR ON
E29B                     ;          READ FILE NUMBER
E29B                     ;          READ &CHECK CRC,FLAG
E29B                     ;
E29B                     MTON:   ENT
E29B 2121E4                      LD      HL,SIOLD
E29E 060B                        LD      B,0BH
E2A0 CDD8E2                      CALL    LSINT           ;load init
E2A3                     ;
E2A3 3E10                LREDY:  LD      A,10H
E2A5 D3F6                        OUT     (SIOAC),A
E2A7 DBF6                        IN      A,(SIOAC)
E2A9 E608                        AND     8
E2AB CA06E4                      JP      Z,IOE50         ;Not ready
E2AE CD35E4                      CALL    BRKC
E2B1 3E10                        LD      A,10H
E2B3 D3F7                        OUT     (SIOBC),A
E2B5 DBF7                        IN      A,(SIOBC)
E2B7 E608                        AND     8
E2B9 28E8                        JR      Z,LREDY
E2BB 01E900                      LD      BC,00E9H        ;Wait 160ms
E2BE CD14E4                      CALL    TIMW
E2C1 CD13E3                      CALL    SYNCL1          ;LOAD SYNC
E2C4 324211                      LD      (FNA),A
E2C7 3C                          INC     A
E2C8 324311                      LD      (FNB),A
E2CB CDC3E3                      CALL    RDCRC
E2CE 214711              FNEND:  LD      HL,SYNCF
E2D1 CBDE                        SET     3,(HL)
```

```
E2D3 AF                          XOR     A
E2D4 323E11                      LD      (HDPT0),A
E2D7 C9                          RET
E2D8                     ;
E2D8                             SKP     H
```

```
E2D8                  ;
E2D8                  ;    sio initial
E2D8                  ;
E2D8                  ;
E2D8 0EF6             LSINT:  LD     C,F6H             ; C sioac
E2DA EDB3                     OTIR
E2DC 3E05                     LD     A,05H
E2DE 324411                   LD     (MTF),A
E2E1 D3F7                     OUT    (SIOBC),A
E2E3 3E80                     LD     A,80H
E2E5 D3F7                     OUT    (SIOBC),A
E2E7 C9                       RET
E2E8                  ;
E2E8                  ; Motor off
E2E8                  ;
E2E8                  QDOFF:  ENT                      ;Basic call
E2E8                  MTOF:   ENT
E2E8 F5                       PUSH   AF
E2E9 3E05                     LD     A,05H
E2EB D3F6                     OUT    (SIOAC),A
E2ED 3E60                     LD     A,60H
E2EF D3F6                     OUT    (SIOAC),A         ;WRGT OFF,TRANS DISABLE
E2F1 3E05                     LD     A,05H
E2F3 D3F7                     OUT    (SIOBC),A
E2F5 AF                       XOR    A
E2F6 324411                   LD     (MTF),A
E2F9 D3F7                     OUT    (SIOBC),A
E2FB F1                       POP    AF
E2FC C9                       RET
E2FD                  ;
E2FD                  ;
E2FD                          SKP    H
```

```
E2FD                  ;
E2FD                  ; SYNCL1 -- LOAD F.N SYNC ONLY
E2FD                  ;                   (SEND BREAK 110ms)
E2FD                  ; SYNCL2 -- LOAD FIRST FILE SYNC
E2FD                  ;                   (SEND BREAK 110ms)
E2FD                  ; SYNCL3 -- LOAD FILES SYNC
E2FD                  ;                   (SEND BREAK 002ms)
E2FD                  ;
E2FD                  SYNCL2: ENT
E2FD 3E58                     LD     A,58H
E2FF 060B                     LD     B,0BH
E301 2121E4                   LD     HL,SIOLD
E304 CDA3E3                   CALL   SYNCA
E307 214711                   LD     HL,SYNCF
E30A CB5E                     BIT    3,(HL)
E30C 010300                   LD     BC,3              ;WAIT 2ms
E30F 2805                     JR     Z,TMLPL
E311 CB9E                     RES    3,(HL)
E313 01A000           SYNCL1: LD     BC,00A0H          ;WAIT 110ms
E316                  ;
E316 CD14E4           TMLPL:  CALL   TIMW
E319 3E05                     LD     A,05H
E31B D3F7                     OUT    (SIOBC),A
E31D 3E82                     LD     A,82H
E31F D3F7                     OUT    (SIOBC),A
E321 3E03                     LD     A,03H
E323 D3F6                     OUT    (SIOAC),A
E325 3ED3                     LD     A,D3H
E327 D3F6                     OUT    (SIOAC),A
E329 01C02C                   LD     BC,2CC0H          ;loop 220ms
E32C                  ;
E32C 3E10             SYNCW0: LD     A,10H
E32E D3F6                     OUT    (SIOAC),A
E330 DBF6                     IN     A,(SIOAC)
E332 E610                     AND    10H
E334 2807                     JR     Z,SYNCW1
E336 0B                       DEC    BC
E337 78                       LD     A,B
E338 B1                       OR     C
E339 20F1                     JR     NZ,SYNCW0
E33B 1817                     JR     SYNW01            ;Un format
E33D                  ;
E33D 3E03             SYNCW1: LD     A,03H
E33F D3F6                     OUT    (SIOAC),A
E341 3EC3                     LD     A,C3H
E343 D3F6                     OUT    (SIOAC),A
E345 069F                     LD     B,9FH             ;loop 3ms
E347                  ;
E347 3E10             SYNCW2: LD     A,10H
E349 D3F6                     OUT    (SIOAC),A
E34B DBF6                     IN     A,(SIOAC)
E34D E601                     AND    01H
E34F 2006                     JR     NZ,SYNCW3
E351 05                       DEC    B
E352 20F3                     JR     NZ,SYNCW2
E354 C30CE4           SYNW01: JP     IOE54             ;Un format
E357                  ;
E357 3E03             SYNCW3: LD     A,03H
E359 D3F6                     OUT    (SIOAC),A
E35B 3EC9                     LD     A,C9H
```

```
E35D D3F6                  OUT    (SIOAC),A
E35F CDF0E3                CALL   RDATA
E362 C3F0E3                JP     RDATA
E365             ;
E365             ;
E365             ;
E365             ; SYNCS1 -- SAVE F.N SYNC
E365             ;               (SEND BREAK 220ms)
E365             ; SYNCS2 -- SAVE FIRST FILE SYNC
E365             ;               (SEND BREAK 220ms)
E365             ; SYNCS3 -- SAVE FILES SYNC
E365             ;               (SEND BREAK 020ms)
E365             ;
E365             SYNCS2:   ENT
E365 3E98                  LD     A,98H
E367 0609                  LD     B,09H
E369 212CE4                LD     HL,SIOSD
E36C CDA3E3                CALL   SYNCA
E36F CD9AE3                CALL   SBRK
E372             ;
E372 214711                LD     HL,SYNCF
E375 CB5E                  BIT    3,(HL)
E377 011D00                LD     BC,001DH          ;WAIT 20ms
E37A 2805                  JR     Z,TMLPS
E37C CB9E                  RES    3,(HL)
E37E 014001      SYNCS1:   LD     BC,0140H          ;WAIT 220ms
E381             ;
E381 CD14E4      TMLPS:    CALL   TIMW
E384 3E05                  LD     A,05H
E386 D3F6                  OUT    (SIOAC),A
E388 3EEF                  LD     A,EFH
E38A D3F6                  OUT    (SIOAC),A
E38C 010100                LD     BC,1              ;WAIT 0.7ms
E38F CD14E4                CALL   TIMW
E392 3EA5                  LD     A,A5H             ; Patch (1984.10.11)
E394 CDDBE3                CALL   TRANS             ;
E397 C380F3                JP     EOMRS             ; Eom reset & return
E39A             ;
E39A             ;
E39A             ;
E39A             ; SBRK -- SEND BREAK (00H)
E39A             ;
E39A             SBRK:     ENT
E39A 3E05                  LD     A,05H
E39C D3F6                  OUT    (SIOAC),A
E39E 3EFF                  LD     A,FFH
E3A0 D3F6                  OUT    (SIOAC),A
E3A2 C9                    RET
E3A3             ;
E3A3             ;
E3A3             ;
E3A3 0EF6        SYNCA:    LD     C,F6H             ; C sioac
E3A5 ED79                  OUT    (C),A
E3A7 3E05                  LD     A,5
E3A9 D3F7                  OUT    (SIOBC),A
E3AB 3E80                  LD     A,80H
E3AD D3F7                  OUT    (SIOBC),A
E3AF EDB3                  OTIR
E3B1 C9                    RET
E3B2             ;
```

```
E3B2                       SKP    H
```

```
E3B2                    ;
E3B2                    ;
E3B2                    ;
E3B2                    ; EOM -- End off message
E3B2                    ;        Save CRC#1,#2,FLAG
E3B2                    ;        File space check
E3B2                    ;
E3B2                    ;
E3B2                    EOM:    ENT
E3B2 010100                     LD      BC,1            ;WAIT 0.7ms
E3B5 CD14E4                     CALL    TIMW
E3B8 3E10                       LD      A,10H
E3BA D3F7                       OUT     (SIOBC),A
E3BC DBF7                       IN      A,(SIOBC)
E3BE E608                       AND     8
E3C0 C0                         RET     NZ
E3C1 1846                       JR      IOE53           ;NO file space
E3C3                    ;
E3C3                    ; RDCRC -- READ CRC & CHECK
E3C3                    ;
E3C3                    RDCRC:  ENT
E3C3 0603                       LD      B,3
E3C5 CDF0E3             RDCR1:  CALL    RDATA
E3C8 10FB                       DJNZ    RDCR1
E3CA DBF6               RDCR2:  IN      A,(SIOAC)
E3CC 0F                         RRCA
E3CD 30FB                       JR      NC,RDCR2        ; Rx Available
E3CF 3E01                       LD      A,01H
E3D1 D3F6                       OUT     (SIOAC),A
E3D3 DBF6                       IN      A,(SIOAC)
E3D5 E640                       AND     40H
E3D7 2027                       JR      NZ,IOE41        ;Hard err
E3D9 B7                         OR      A
E3DA C9                         RET
E3DB                    ;
E3DB                            SKP     H
```

```
E3DB                    ;
E3DB                    ; Save 1 chr by Acc
E3DB                    ;     & ready check
E3DB                    ;
E3DB                    TRANS:  ENT
E3DB F5                         PUSH    AF
E3DC DBF6               TRA1:   IN      A,(SIOAC)
E3DE E604                       AND     4               ;TRANS buf null
E3E0 28FA                       JR      Z,TRA1
E3E2 F1                         POP     AF
E3E3 D3F4                       OUT     (SIOAD),A
E3E5 3E10               NRCK:   LD      A,10H
E3E7 D3F6                       OUT     (SIOAC),A
E3E9 DBF6                       IN      A,(SIOAC)
E3EB E608                       AND     08H
E3ED 2817                       JR      Z,IOE50         ;Not ready
E3EF C9                         RET
E3F0                    ;
E3F0                    ; Read data (1 chr)
E3F0                    ;
E3F0                    RDATA:  ENT
E3F0 CDE5E3                     CALL    NRCK
E3F3 DBF6                       IN      A,(SIOAC)       ;RR0
E3F5 07                         RLCA
E3F6 3808                       JR      C,IOE41         ;Hard err
E3F8 0F                         RRCA
E3F9 0F                         RRCA
E3FA 30F4                       JR      NC,RDATA
E3FC DBF4                       IN      A,(SIOAD)
E3FE B7                         OR      A
E3FF C9                         RET
E400                    ;
E400                            SKP     H
```

```
E400                    ;
E400                    ; i/o err
E400                    ;
E400 3E29               IOE41:  LD      A,41            ;Hard err
E402 21                         DEFB    21H
E403 3E2E               IOE46:  LD      A,46            ;Write protect
E405 21                         DEFB    21H
E406 3E32               IOE50:  LD      A,50            ;Not ready
E408 21                         DEFB    21H
E409 3E35               IOE53:  LD      A,53            ;No file space
E40B 21                         DEFB    21H
E40C 3E36               IOE54:  LD      A,54            ;Un format
E40E ED7B4811                   LD      SP,(RETSP)
E412 37                         SCF
E413 C9                         RET
E414                    ;
E414                    ;
E414                    ;
E414                    ;
E414                    ;   wait  timer
E414                    ;
E414                    ; BC=001H=  0.7ms(  0.704ms)
E414                    ;    003H=  2.0ms(  2.107ms)
E414                    ;    01DH= 20.0ms( 19.938ms)
E414                    ;    0A0H=110.0ms(110.050ms)
E414                    ;    0E9H=160.0ms(160.140ms)
E414                    ;    140H=220.0ms(219.940ms)
E414                    ;
E414                    ;
E414                    ;
E414                    TIMW:   ENT
E414 F5                         PUSH    AF
E415 3E96               TIMW1:  LD      A,96H           ; for ex
E417 3D                 TIMW2:  DEC     A
E418 20FD                       JR      NZ,TIMW2
E41A 0B                         DEC     BC
E41B 78                         LD      A,B
E41C B1                         OR      C
E41D 20F6                       JR      NZ,TIMW1
E41F F1                         POP     AF
E420 C9                         RET
E421                    ;
E421                            SKP     H
```

```
E421                    ;
E421                    ;
E421                    ;
E421                    ; SIO CH A COMMAND CHAIN
E421                    ;
E421                    ; SIOLD -- LOAD INIT. DATA
E421                    ; SIOSD -- SAVE INIT. DATA
E421                    ;
E421                    ;
E421                    ;
E421 58                 SIOLD:  DEFB    58H             ;CHANNEL RESET
E422 04                         DEFB    04H             ;POINT WR4
E423 10                         DEFB    10H             ;X1 CLOCK
E424 05                         DEFB    05H             ;POINT WR1
E425 04                         DEFB    04H             ;CRC-16
E426 03                         DEFB    03H             ;POINT WR3
E427 D0                         DEFB    D0H             ;ENTER HUNT PHASE
E428                    ;Rx 8bits
E428 06                         DEFB    06H             ;POINT WR6
E429 16                         DEFB    16H             ;SYNC CHR(1)
E42A 07                         DEFB    07H             ;POINT WR7
E42B 16                         DEFB    16H             ;SYNC CHR(2)
E42C                    ;
E42C 98                 SIOSD:  DEFB    98H             ;CHANNEL RESET
E42D                    ;Tx CRC Generator reset
E42D 04                         DEFB    04H             ;POINT WR4
E42E 10                         DEFB    10H             ;X1 CLOCK
E42F 06                         DEFB    06H             ;POINT WR6
E430 16                         DEFB    16H             ;SYNC CHR(1)
E431 07                         DEFB    07H             ;POINT WR7
E432 16                         DEFB    16H             ;SYNC CHR(2)
E433 05                         DEFB    05H             ;POINT WR5
E434 6D                         DEFB    6DH             ;Tx CRC ENABLE
E435                    ;
E435                    ;
E435                    ; BREAK CHECK
E435                    ;
E435 3EE8               BRKC:   LD      A,E8H
E437 3200E0                     LD      (E000H),A
E43A 00                         NOP
E43B 3A01E0                     LD      A,(E001H)
E43E E681                       AND     81H
E440 C0                         RET     NZ
E441 CDDBE1                     CALL    WRCAN
E444 ED7B4811                   LD      SP,(RETSP)
E448 37                         SCF
E449 C9                         RET
E44A                    ;
E44A                            SKP     H
```

```
 **  Z80 ASSEMBLER 2Z-004C  <9Z-504M-V1.0C>  PAGE 23            03.06.85
E44A                 ;
E44A                 ;----( MFM Minifloppy control )----
E44A                 ;      Subroutine  (MB8876)
E44A                 ;
E44A                 ;
E44A                 ;
E44A                 ;  Call condition
E44A                 ;
E44A                 ;  Case of disk initialize
E44A                 ;     DRIVE N=IX+0  (0~3)
E44A                 ;
E44A                 ;  Case of sequential read & write
E44A                 ;     Drive n=IX+0  (0~3)
E44A                 ;
E44A                 ;     Sector adrs  =IX+1,2 (0-045FH)
E44A                 ;                          (0-1119 )
E44A                 ;     Byte size    =IX+3,4
E44A                 ;     Address      =IX+5,6
E44A                 ;     Next track   =IX+7
E44A                 ;     Next sector  =IX+8
E44A                 ;     Start track  =IX+9
E44A                 ;     Start sector =IX+10
E44A                 ;
E44A                 ;
E44A                 ;  I/O Port address
E44A                 ;
00D8         P       CR:     EQU    D8H
00D9         P       TR:     EQU    D9H
00DA         P       SCR:    EQU    DAH
00DB         P       DR:     EQU    DBH
00DC         P       DM:     EQU    DCH
00DD         P       HS:     EQU    DDH
E44A                 ;
E44A                 ;  Subroutine work
E44A                 ;
CF00         P       BPRO:   EQU    CF00H            ;IFM loading adrs.
11A3         P       BUF:    EQU    11A3H
CEE9         P       BPARA:  EQU    BPRO-23          ;FD work
E44A                 ;
E44A                 ;
E44A                 ;--------( Ercode map )--------
E44A                 ;
E44A                 ; 50 :Not ready
E44A                 ; 41 :Data error
E44A                 ;               Track 00 err
E44A                 ;               Write protect err
E44A                 ;               Seek err
E44A                 ;               CRC err
E44A                 ;               Lost data
E44A                 ; 54 :Unformat
E44A                 ;         Recode not found
E44A                 ; 56 :Invalid data
E44A                 ;
E44A                 ;
E44A                 ;  Main routine
E44A                 ;
E44A                 FDX:    ENT
E44A E3                      EX     (SP),HL
E44B 22FECE                  LD     (BPARA+21),HL
E44E CDD5E8                  CALL   FDCC&            ;FD i/o check
```

```
 **  Z80 ASSEMBLER 2Z-004C
E451 C2BAE4                  JP     NZ,NOTIO
E454 11E9CE                  LD     DE,BPARA
E457 21D1E4                  LD     HL,BOOT
E45A 010B00                  LD     BC,11
E45D EDB0                    LDIR
E45F CD30E5                  CALL   MOFF
E462 DD21E9CE        SJP:    LD     IX,BPARA
E466 CDA7E5                  CALL   BREAD
E469 2100CF                  LD     HL,BPRO
E46C 11CAE4                  LD     DE,IPLMC
E46F 0607                    LD     B,7
E471 4E              MCHECK: LD     C,(HL)
E472 1A                      LD     A,(DE)
E473 B9                      CP     C
E474 C2AEE4                  JP     NZ,MASTE
E477 23                      INC    HL
E478 13                      INC    DE
E479 10F6                    DJNZ   MCHECK
E47B 1188ED                  LD     DE,IPLM0         ;'IPL IS LOADING'
E47E DF                      RST    3
E47F 1107CF                  LD     DE,BPRO+7        ;NAME
E482 DF                      RST    3
E483 210012                  LD     HL,1200H         ; Loading adrs
E486 DD7505                  LD     (IX+5),L
E489 DD7406                  LD     (IX+6),H
E48C 2A14CF                  LD     HL,(BPRO+14H)    ;BYTE SIZE
E48F DD7503                  LD     (IX+3),L
E492 DD7404                  LD     (IX+4),H
E495 2A1ECF                  LD     HL,(BPRO+1EH)    ;START SECTOR
E498 DD7501                  LD     (IX+1),L
E49B DD7402                  LD     (IX+2),H
E49E                 ;
E49E CDA7E5                  CALL   BREAD
E4A1 CD30E5                  CALL   MOFF
E4A4                 ;
E4A4                 ;    Exec load file
E4A4                 ;
E4A4 010002                  LD     BC,0200H         ; Default code
E4A7 D9                      EXX
E4A8 2114CF                  LD     HL,BPRO+14H      ; Size point
E4AB C3FCEC                  JP     EXF
E4AE                 ;
E4AE CD30E5          MASTE:  CALL   MOFF
E4B1 11F6ED                  LD     DE,ERRM1         ;'NOT MASTER'
E4B4 180C                    JR     ERRTR1
E4B6                 ;
E4B6 FE32            ERRTRT: CP     50
E4B8 2005                    JR     NZ,ERRTR0
E4BA 11B5ED          NOTIO:  LD     DE,IPLM3         ;'MAKE READY FD'
E4BD 1803                    JR     ERRTR1
E4BF 11E5ED          ERRTR0: LD     DE,ERRM0         ;'FD:LOADING ERROR'
E4C2 31EE10          ERRTR1: LD     SP,10EEH
E4C5 2AFECE                  LD     HL,(BPARA+21)
E4C8 E3                      EX     (SP),HL
E4C9 C9                      RET
E4CA                 ;
E4CA                 ;   PARAMETER SETTING
E4CA                 ;
E4CA 03              IPLMC:  DEFB   03H              ;IPL MASTER FLAG
E4CB 49504C50                DEFM   'IPLPRO'
```

```
E4CF 524F
E4D1 00      BOOT:   DEFB    0             ;DRIVE NO.
E4D2 0000            DEFW    0000H         ;SECTOR ADDR.
E4D4 0001            DEFW    0100H         ;IFM BYTE SIZE
E4D6 00CF            DEFW    BPRO          ;IFM LOADING ADDR.
E4D8 0000            DEFW    0             ;IX+7,8
E4DA 0000            DEFW    0             ;IX+9,10
E4DC         ;
E4DC         ; READY CHECK
E4DC         ;
E4DC         READY:  ENT
E4DC 3AF5CE          LD      A,(MTFG)
E4DF 0F              RRCA
E4E0 D417E5          CALL    NC,FDMTON
E4E3 DD7E00          LD      A,(IX+0)      ;DRIVE NO GET
E4E6 F684            OR      84H
E4E8 D3DC            OUT     (DM),A        ;DRIVE SELECT MOTON
E4EA AF              XOR     A
E4EB 32F4CE          LD      (CMD),A
E4EE CDBFE6          CALL    DLY60M
E4F1 210000          LD      HL,0
E4F4 2B      REDY0:  DEC     HL
E4F5 7C              LD      A,H
E4F6 B5              OR      L
E4F7 2819            JR      Z,REDY1
E4F9 DBD8            IN      A,(CR)        ;STATUS GET
E4FB 2F              CPL
E4FC 07              RLCA
E4FD 38F5            JR      C,REDY0
E4FF DD4E00          LD      C,(IX+0)
E502 21F6CE          LD      HL,CLBF0
E505 0600            LD      B,0
E507 09              ADD     HL,BC
E508 CB46            BIT     0,(HL)
E50A 2005            JR      NZ,REDY2
E50C CD48E5          CALL    RCLB
E50F CBC6            SET     0,(HL)
E511 C9      REDY2:  RET
E512 3E32    REDY1:  LD      A,50
E514 C390E6          JP      ERJMP
E517         ;
E517         ; MOTOR ON
E517         ;
E517         FDMTON: ENT
E517 3E80            LD      A,80H
E519 D3DC            OUT     (DM),A
E51B 0610            LD      B,16
E51D CDBFE6  MTD1:   CALL    DLY60M
E520 10FB            DJNZ    MTD1
E522 3E01            LD      A,1
E524 32F5CE          LD      (MTFG),A
E527 C9              RET
E528         ;
E528         ; SEEK TREATMENT
E528         ;
E528         SEEK:   ENT
E528 3E1B            LD      A,1BH
E52A CD55E5          CALL    CMDOT1
E52D E699            AND     99H
E52F C9              RET
```

```
E530         ;
E530         ;  MOTER OFF
E530         ;
E530         MOFF:   ENT
E530 F5              PUSH    AF
E531 CDB9E6          CALL    DLY1M         ;1000 US DELAY
E534 AF              XOR     A
E535 D3DC            OUT     (DM),A
E537 32F6CE          LD      (CLBF0),A
E53A 32F7CE          LD      (CLBF1),A
E53D 32F8CE          LD      (CLBF2),A
E540 32F9CE          LD      (CLBF3),A
E543 32F5CE          LD      (MTFG),A
E546 F1              POP     AF
E547 C9              RET
E548         ;
E548         ; RECALIBLATION
E548         ;
E548         RCLB:   ENT
E548 3E0B            LD      A,0BH
E54A CD55E5          CALL    CMDOT1
E54D E685            AND     85H
E54F EE04            XOR     04
E551 C8              RET     Z
E552 C361E6          JP      ERROR         ;
E555         ;
E555         ;     COMAND OUT ROUTINE
E555         ;
E555 32F4CE  CMDOT1: LD      (CMD),A
E558 2F              CPL
E559 D3D8            OUT     (CR),A
E55B CD68E5          CALL    BSYON
E55E CDBFE6          CALL    DLY60M
E561 DBD8            IN      A,(CR)
E563 2F              CPL
E564 32FBCE          LD      (STAFG),A
E567 C9              RET
E568         ;
E568         ; BUSY AND WAIT
E568         ;
E568 D5      BSYON:  PUSH    DE
E569 E5              PUSH    HL
E56A CDA1E5          CALL    BSY0
E56D 210000  BSYON2: LD      HL,0000H
E570 2B      BSYON0: DEC     HL
E571 7C              LD      A,H
E572 B5              OR      L
E573 2808            JR      Z,BSYON1
E575 DBD8            IN      A,(CR)
E577 0F              RRCA
E578 30F6    BSYON3: JR      NC,BSYON0
E57A E1              POP     HL
E57B D1              POP     DE
E57C C9              RET
E57D         ;
E57D 1D      BSYON1: DEC     E
E57E 20ED            JR      NZ,BSYON2
E580 3E29    BSYONE: LD      A,41
E582 E1              POP     HL
E583 D1              POP     DE
```

```
E584 C390E6             JP     ERJMP
E587           ;
E587 D5        BSYOFF:  PUSH   DE
E588 E5                 PUSH   HL
E589 CDA1E5             CALL   BSY0
E58C 210000    BSYOF2:  LD     HL,0000H
E58F 2B        BSYOF0:  DEC    HL
E590 7C                 LD     A,H
E591 B5                 OR     L
E592 2808               JR     Z,BSYOF1
E594 DBD8               IN     A,(CR)
E596 0F                 RRCA
E597 38F6      BSYOF3:  JR     C,BSYOF0
E599 E1                 POP    HL
E59A D1                 POP    DE
E59B C9                 RET
E59C           ;
E59C 1D        BSYOF1:  DEC    E
E59D 20ED               JR     NZ,BSYOF2
E59F 18DF      BSYOFE:  JR     BSYONE
E5A1           ;
E5A1 CDB3E6    BSY0:    CALL   DLY80U
E5A4 1E07               LD     E,07H
E5A6 C9                 RET
E5A7           ;
E5A7           ;
E5A7           ; SEQENTIAL READ
E5A7           ;
E5A7           BREAD:   ENT
E5A7 CD96E6             CALL   CNVRT
E5AA CDFBE5    READ1:   CALL   PARST1
E5AD CD1BE6    RE8:     CALL   SIDST
E5B0 CD28E5             CALL   SEEK
E5B3 C290E6             JP     NZ,ERJMP
E5B6 CD2BE6             CALL   PARST2
E5B9 F3                 DI
E5BA 3E94               LD     A,94H                   ;READ & CMD
E5BC CD4EE6             CALL   CMDOT2
E5BF 0600      RE6:     LD     B,00
E5C1 DBD8      RE4:     IN     A,(CR)
E5C3 0F                 RRCA
E5C4 381A               JR     C,RE3
E5C6 0F                 RRCA
E5C7 38F8               JR     C,-6
E5C9 EDA2               INI
E5CB 20F4               JR     NZ,RE4
E5CD DD3408             INC    (IX+8)
E5D0 DD7E08             LD     A,(IX+8)
E5D3 FE11               CP     17
E5D5 2805               JR     Z,+7
E5D7 15                 DEC    D
E5D8 20E5               JR     NZ,RE6
E5DA 1801               JR     +3
E5DC 15                 DEC    D
E5DD CD58E6             CALL   INTER
E5E0 00        RE3:     NOP
E5E1 DBD8               IN     A,(CR)
E5E3 2F                 CPL
E5E4 32FBCE             LD     (STAFG),A
E5E7 E6FF               AND    FFH
```

```
E5E9 2076               JR     NZ,[illegible]
E5EB CD3CE6             CALL   ADJ
E5EE CAF6E5             JP     Z,REND
E5F1 DD7E07             LD     A,(IX+7)
E5F4 18B7               JR     RE8
E5F6 3E80      REND:    LD     A,80H
E5F8 D3DC               OUT    (DM),A
E5FA C9                 RET
E5FB           ;
E5FB           ;
E5FB           ;
E5FB           ; PARAMETER SET
E5FB           ;
E5FB           ;
E5FB CDDCE4    PARST1:  CALL   READY
E5FE DD5604             LD     D,(IX+4)
E601 DD7E03             LD     A,(IX+3)
E604 B7                 OR     A
E605 2801               JR     Z,+3
E607 14                 INC    D
E608 DD7E0A             LD     A,(IX+10)
E60B DD7708             LD     (IX+8),A
E60E DD7E09             LD     A,(IX+9)
E611 DD7707             LD     (IX+7),A
E614 DD6E05             LD     L,(IX+5)
E617 DD6606             LD     H,(IX+6)
E61A C9                 RET
E61B           ;
E61B           ;  SIZE SEEK SET
E61B           ;
E61B CB3F      SIDST:   SRL    A
E61D 2F                 CPL
E61E D3DB               OUT    (DR),A
E620 3004               JR     NC,+6
E622 3E01               LD     A,1
E624 1801               JR     +3
E626 AF                 XOR    A
E627 2F                 CPL
E628 D3DD               OUT    (HS),A
E62A C9                 RET
E62B           ;
E62B           ;
E62B           ; TRACK & SECTOR SET
E62B           ;
E62B 0EDB      PARST2:  LD     C,DBH
E62D DD7E07             LD     A,(IX+7)
E630 CB3F               SRL    A
E632 2F                 CPL
E633 D3D9               OUT    (TR),A
E635 DD7E08             LD     A,(IX+8)
E638 2F                 CPL
E639 D3DA               OUT    (SCR),A
E63B C9                 RET
E63C           ;
E63C           ;
E63C           ; ADJUST SECT & TRACK
E63C           ;
E63C DD7E08    ADJ:     LD     A,(IX+8)
E63F FE11               CP     17
E641 2008               JR     NZ,+10
```

```
    **  Z80 ASSEMBLER 2Z-004C  <9Z-504M-V1.0C>  PAGE 29          03.06.85

E643 3E01                  LD      A,1
E645 DD7708                LD      (IX+8),A
E648 DD3407                INC     (IX+7)
E64B 7A                    LD      A,D
E64C B7                    OR      A
E64D C9                    RET
E64E               ;
E64E               ;  COMAMD OUT & WAIT
E64E               ;
E64E 32F4CE        CMDOT2: LD      (CMD),A
E651 2F                    CPL
E652 D3D8                  OUT     (CR),A
E654 CD87E5                CALL    BSYOFF
E657 C9                    RET
E658               ;
E658               ;
E658               ;  FORCE INTERRUPT
E658               ;
E658 3ED8          INTER:  LD      A,D8H
E65A 2F                    CPL
E65B D3D8                  OUT     (CR),A
E65D CD68E5                CALL    BSYON
E660 C9                    RET
E661               ;
E661               ;
E661               ; STATUS CHECK
E661               ;
E661 3AF4CE        ERROR:  LD      A,(CMD)
E664 FE0B                  CP      0BH
E666 281B                  JR      Z,ERCK1
E668 FE1B                  CP      1BH
E66A 2817                  JR      Z,ERCK1
E66C FEF4                  CP      F4H
E66E 2813                  JR      Z,ERCK1
E670 3AFBCE                LD      A,(STAFG)
E673 CB7F                  BIT     7,A
E675 2017                  JR      NZ,ERRET
E677 CB77                  BIT     6,A
E679 200F                  JR      NZ,ERRET1
E67B CB67                  BIT     4,A
E67D 3E36                  LD      A,54
E67F 200F                  JR      NZ,ERJMP
E681 1807                  JR      ERRET1
E683 3AFBCE        ERCK1:  LD      A,(STAFG)
E686 CB7F                  BIT     7,A
E688 2004                  JR      NZ,ERRET
E68A 3E29          ERRET1: LD      A,41
E68C 1802                  JR      ERJMP
E68E 3E32          ERRET:  LD      A,50
E690 CD30E5        ERJMP:  CALL    MOFF
E693 C3B6E4                JP      ERRTRT
E696               ;
E696               ;
E696               ; SECTOR TO TRACK & SECTOR CONVERT
E696               ;
E696 0600          CNVRT:  LD      B,0
E698 111000                LD      DE,16
E69B DD6E01                LD      L,(IX+1)
E69E DD6602                LD      H,(IX+2)
E6A1 AF                    XOR     A
```

```
    **  Z80 ASSEMBLER 2Z-004C  <9Z-504M-V1.0C>  PAGE 30          03.06.85

E6A2 ED52          TRANS0: SBC     HL,DE
E6A4 3803                  JR      C,TRANS1
E6A6 04                    INC     B
E6A7 18F9                  JR      TRANS0
E6A9 19            TRANS1: ADD     HL,DE
E6AA 60                    LD      H,B
E6AB 2C                    INC     L
E6AC DD7409                LD      (IX+9),H
E6AF DD750A                LD      (IX+10),L
E6B2 C9                    RET
E6B3               ;
E6B3               ;
E6B3               ;  TIME DELAY ( 1M & 60M & 80U )
E6B3               ;
E6B3               DLY80U: ENT
E6B3 D5                    PUSH    DE
E6B4 110F00                LD      DE,15
E6B7 180A                  JR      DLYT
E6B9               DLY1M:  ENT
E6B9 D5                    PUSH    DE
E6BA 11A000                LD      DE,160
E6BD 1804                  JR      DLYT
E6BF               DLY60M: ENT
E6BF D5                    PUSH    DE
E6C0 112620                LD      DE,8230
E6C3 1B            DLYT:   DEC     DE
E6C4 7B                    LD      A,E
E6C5 B2                    OR      D
E6C6 20FB                  JR      NZ,DLYT
E6C8 D1                    POP     DE
E6C9 C9                    RET
E6CA               ;
E6CA               ;  FLOPPY WORK AREA
E6CA               ;
CEF4  P            CMD:    EQU     BPARA+11
CEF5  P            MTFG:   EQU     CMD+1
CEF6  P            CLBF0:  EQU     MTFG+1
CEF7  P            CLBF1:  EQU     CLBF0+1
CEF8  P            CLBF2:  EQU     CLBF1+1
CEF9  P            CLBF3:  EQU     CLBF2+1
CEFA  P            VRFCNT: EQU     CLBF3+1
CEFB  P            STAFG:  EQU     VRFCNT+1
E6CA               ;
E6CA                       SKP     H
```

```
E6CA                        ;
E6CA                        ;========================================
E6CA                        ;
E6CA                        ;     1R12(Static ram board 32Kbyte)
E6CA                        ;                     Read/Write routine
E6CA                        ;
E6CA                        ;========================================
E6CA                        ;
1108  P                     COMNT:  EQU    1108H
E6CA                        ;
E6CA                        ; 1R12 Read sub.
E6CA                        ;
E6CA                        1R12RD: ENT
E6CA CDDAE6                         CALL   1R12R0
E6CD C224EB                         JP     NZ,?ERX
E6D0                        ;
E6D0                        ;     Exec load file
E6D0                        ;
E6D0                        EXRAM:  ENT
E6D0 010000                         LD     BC,0000H       ; Default code (SRAM)
E6D3 D9                             EXX
E6D4 210211                         LD     HL,SIZE
E6D7 C3FCEC                         JP     EXF
E6DA                        ;
E6DA                        1R12R0: ENT
E6DA                        ;
E6DA                        ;     Information sum check
E6DA                        ;
E6DA CD29E7                         CALL   CHECK
E6DD C0                             RET    NZ
E6DE                        ;
E6DE                        ;     Counter reset
E6DE                        ;
E6DE ED78                          ,IN     A,(C)
E6E0                        ;
E6E0                        ;     Read information block (9Byte)
E6E0                        ;
E6E0 0C                             INC    C
E6E1 210211                         LD     HL,SIZE
E6E4 0609                           LD     B,09H
E6E6 EDB2                           INIR
E6E8                        ;
E6E8                        ;     Read data block
E6E8                        ;
E6E8 ED5B0211                       LD     DE,(SIZE)
E6EC 210012                         LD     HL,1200H       ; Loading adrs
E6EF 7B                             LD     A,E
E6F0 B7                             OR     A
E6F1 2803                           JR     Z,+5
E6F3 47                             LD     B,A
E6F4 EDB2                           INIR
E6F6 0600                           LD     B,00H
E6F8 15                             DEC    D
E6F9 F2F4E6                         JP     P,-5
E6FC                        ;
E6FC                        ;     Data block's sum check
E6FC                        ;
E6FC 110012                         LD     DE,1200H       ; Data adrs
E6FF ED4B0211                       LD     BC,(SIZE)      ; BCSize
E703 CD0EE7                         CALL   SUM            ; HLCheck sum
```

```
E706 ED5B0811                       LD     DE,(COMNT)     ; DECheck sum data
E70A B7                             OR     A
E70B ED52                           SBC    HL,DE          ; OK?
E70D C9                             RET
E70E                        ;
E70E                        ;     sum check
E70E                        ;
E70E                        ; IN BC=Size
E70E                        ;    DE=Data adr
E70E                        ; EXIT HL=Check sum
E70E                        ;
E70E D9                     SUM:    EXX
E70F 210000                         LD     HL,0           ; HL'= Check sum clr
E712 0E08                           LD     C,8            ; C' = Loop count
E714 D9                             EXX
E715 78                     SUMCK1: LD     A,B            ; BC = Size
E716 B1                             OR     C
E717 280E                           JR     Z,SUMCK2
E719 1A                             LD     A,(DE)         ; DE = Data adrs
E71A D9                             EXX
E71B 41                             LD     B,C            ; BC'
E71C 07                     SUMCK3: RLCA
E71D 3001                           JR     NC,+3
E71F 23                             INC    HL             ; HL'= Check sum data
E720 10FA                           DJNZ   SUMCK3
E722 D9                             EXX
E723 13                             INC    DE             ; DE
E724 0B                             DEC    BC             ; BC
E725 18EE                           JR     SUMCK1
E727 D9                     SUMCK2: EXX
E728 C9                             RET
E729                        ;
E729                        ;     Information's sum check
E729                        ;
E729 ED78                   CHECK:  IN     A,(C)          ; Counter reset
E72B 0608                           LD     B,08H          ; B=Byte Counter
E72D 1600                           LD     D,00H          ; Sum Counter
E72F 0C                             INC    C              ; C=C+1
E730 ED78                   CK1:    IN     A,(C)          ; C=C+1
E732 C5                             PUSH   BC
E733 0608                           LD     B,08H          ; Bit Counter
E735 07                     CK2:    RLCA
E736 3001                           JR     NC,+3
E738 14                             INC    D
E739 10FA                           DJNZ   CK2
E73B 7A                             LD     A,D
E73C C1                             POP    BC
E73D 57                             LD     D,A
E73E 10F0                           DJNZ   CK1
E740 ED78                           IN     A,(C)          ; C=C+1
E742 0D                             DEC    C              ; C=C+0
E743 BA                             CP     D
E744 C9                             RET
E745                        ;
E745                        ;============================
E745                        ;
E745                        ;     1R12 Write sub.
E745                        ;
E745                        ;============================
E745                        ;
```

```
E745                1R12WR: ENT
E745                ;
E745                ;      Disp 'Writing.....'
E745                ;
E745 11E2E7                 LD     DE,MSGW01       ; 'Writing now'
E748 CD1809                 CALL   0918H
E74B DF                     RST    3
E74C CD1809                 CALL   0918H
E74F 11F0E7                 LD     DE,MSGW02       ; 'Master tape set'
E752 DF                     RST    3
E753 CD1809                 CALL   0918H
E756                ;
E756                ;     Read information block
E756                ;
E756 CD2700                 CALL   RDINF
E759 380E                   JR     C,RDB0
E75B                ;
E75B                ;     Disp 'Loading.....'
E75B                ;
E75B 1191EE                 LD     DE,MSGLD        ; 'Loading'
E75E CD1809                 CALL   0918H
E761 DF                     RST    3
E762 11F110                 LD     DE,NAME
E765 DF                     RST    3
E766                ;
E766                ;      Read data block
E766                ;
E766 CD2A00                 CALL   RDDAT
E769 DA24EB         RDB0:   JP     C,?ERX
E76C                ;
E76C                ;      Counter reset
E76C                ;
E76C ED78                   IN     A,(C)           ; C=C+0
E76E                ;
E76E                ;      Sum check for data
E76E                ;
E76E ED430A11               LD     (COMNT+2),BC    ; C=Port adrs,C=C+0
E772 ED5B0411               LD     DE,(DTADR)
E776 ED4B0211               LD     BC,(SIZE)
E77A D5                     PUSH   DE
E77B C5                     PUSH   BC
E77C CD0EE7                 CALL   SUM             ; HLCheck sum
E77F 220811                 LD     (COMNT),HL
E782                ;
E782                ;     Write information (8Byte)
E782                ;
E782 210211                 LD     HL,SIZE
E785 ED4B0A11               LD     BC,(COMNT+2)    ; C=Port adrs,C=C+0
E789 0608                   LD     B,08H           ; Byte Counter
E78B 0C                     INC    C
E78C 0C                     INC    C
E78D E5                     PUSH   HL
E78E C5                     PUSH   BC
E78F EDB3                   OTIR                   ; C=C+2
E791 C1                     POP    BC
E792 E1                     POP    HL
E793                ;
E793                ;     Sum check for information block
E793                ;         AccCheck sum data
E793                ;
```

```
E793 D5                     PUSH   DE               DESize
E794 1600                   LD     D,00H            Sum Counter
E796 C5             WCK1:   PUSH   BC
E797 0608                   LD     B,08H
E799 7E                     LD     A,(HL)
E79A 07             WCK2:   RLCA
E79B 3001                   JR     NC,WCK3
E79D 14                     INC    D
E79E 10FA           WCK3:   DJNZ   WCK2
E7A0 23                     INC    HL
E7A1 C1                     POP    BC
E7A2 10F2                   DJNZ   WCK1
E7A4 7A                     LD     A,D
E7A5 D1                     POP    DE
E7A6 ED79                   OUT    (C),A          ; C=C+2
E7A8                ;
E7A8                ;      Write data block
E7A8                ;
E7A8 D1                     POP    DE             ; DESize
E7A9 E1                     POP    HL             ; HLData adrs
E7AA 7B                     LD     A,E
E7AB B7                     OR     A
E7AC 2803                   JR     Z,+5
E7AE 43                     LD     B,E
E7AF EDB3                   OTIR                  ; C=C+2
E7B1 0600                   LD     B,00
E7B3 15                     DEC    D
E7B4 F2AFE7                 JP     P,-5
E7B7 C35EEA                 JP     ST1X
E7BA                ;
E7BA                ;      1R12 set check
E7BA                ;
E7BA                R12CK1: ENT
E7BA AF                     XOR    A              ; carry flag reset
E7BB ED78                   IN     A,(C)          ; Counter reset,C=C+0
E7BD 0C                     INC    C
E7BE ED78                   IN     A,(C)          ; C=C+1
E7C0 08                     EX     AF,AF'
E7C1 0D                     DEC    C
E7C2 ED78                   IN     A,(C)          ; Counter reset
E7C4 06A5                   LD     B,A5H
E7C6 0C                     INC    C
E7C7 0C                     INC    C
E7C8 ED41                   OUT    (C),B          ; C=C+2
E7CA 0D                     DEC    C
E7CB 0D                     DEC    C
E7CC ED78                   IN     A,(C)          ; Counter reset
E7CE 0C                     INC    C
E7CF ED78                   IN     A,(C)          ; C=C+1
E7D1 B8                     CP     B
E7D2 200B                   JR     NZ,12NSET
E7D4 0D                     DEC    C
E7D5 ED78                   IN     A,(C)          ; Counter reset
E7D7 08                     EX     AF,AF'
E7D8 0C                     INC    C
E7D9 0C                     INC    C
E7DA ED79                   OUT    (C),A          ; C=C+2
E7DC 0D                     DEC    C
E7DD 0D                     DEC    C              ; C=C+0
E7DE C9                     RET
```

```
  **  Z80 ASSEMBLER 2Z-004C  <9Z-504M-V1.0C>  PAGE 35          03.06.85

E7DF AF            12NSET: XOR     A              ; Z flag set
E7E0 37                    SCF                    ; Not set
E7E1 C9                    RET
E7E2               ;
E7E2               ;
E7E2               ;     Message table
E7E2               ;
E7E2 52            MSGW01: DEFM    'R'            ; 'Ramcard prog.'
E7E3 A1                    DEFB    A1H            ; A
E7E4 B3                    DEFB    B3H            ; M
E7E5 9F                    DEFB    9FH            ; C
E7E6 A1                    DEFB    A1H            ; A
E7E7 9D                    DEFB    9DH            ; R
E7E8 9C                    DEFB    9CH            ; D
E7E9 20                    DEFB    20H            ;
E7EA 9E                    DEFB    9EH            ; P
E7EB 9D                    DEFB    9DH            ; R
E7EC B7                    DEFB    B7H            ; O
E7ED 97                    DEFB    97H            ; G
E7EE 2E                    DEFB    2EH            ; .
E7EF 0D                    DEFB    0DH
E7F0               ;
E7F0 4D            MSGW02: DEFM    'M'            ; 'Master tape set'
E7F1 A1                    DEFB    A1H            ; A
E7F2 A4                    DEFB    A4H            ; S
E7F3 96                    DEFB    96H            ; T
E7F4 92                    DEFB    92H            ; E
E7F5 9D                    DEFB    9DH            ; R
E7F6 20                    DEFB    20H            ;
E7F7 96                    DEFB    96H            ; T
E7F8 A1                    DEFB    A1H            ; A
E7F9 9E                    DEFB    9EH            ; P
E7FA 92                    DEFB    92H            ; E
E7FB 20                    DEFB    20H            ;
E7FC A4                    DEFB    A4H            ; S
E7FD 92                    DEFB    92H            ; E
E7FE 96                    DEFB    96H            ; T
E7FF 0D                    DEFB    0DH
E800               ;
E800                       SKP     H
```

```
  **  Z80 ASSEMBLER 2Z-004C  <9Z-504M-V1.0C>  PAGE 36          03.06.85

E800                       ORG     E800H
E800               ;
E800               ;================================================
E800               ;
E800               ;
E800               ;                   MZ-800
E800               ;
E800               ;     Rom Monitor & Initial program loader
E800               ;
E800               ;            V1.0C   1984.10.11
E800               ;
E800               ;
E800               ;================================================
E800               ;
E800               ;
11A3   P           BUFER:  EQU     11A3H          ; Key buffer
00CE   P           GDGMD:  EQU     00CEH          ; CRTC Mode reg.
00CD   P           GDGRF:  EQU     00CDH          ;      Read format reg.
00CC   P           GDGWF:  EQU     00CCH          ;      Write format reg.
E800               ;
E800               ;     Jump table
E800               ;
E800               BEGIN0: ENT                    ; MZ-800 Monitor start
E800 00                    NOP
E801 C313E8                JP      BEGIN
E804               BGIN00: ENT                    ; Monitor command
E804 C35EEA                JP      ST1X
E807               LCM12:  ENT                    ; CMT Load $1200~
E807 C370EB                JP      LDCMT
E80A               SCM12:  ENT                    ; CMT Save $1200~
E80A C3E8EB                JP      SACMT
E80D               VCM12:  ENT                    ; CMT Verify $1200~
E80D C309EC                JP      VFCMT
E810               QDIO:   ENT                    ; Quick disk iocs
E810 C310E0                JP      QDIOS
E813               ;
E813               ;
E813               BEGIN:  ENT                    ; 800 Monitor start
E813 F3                    DI
E814 ED56                  IM      1              ; Interrupt mode 1
E816 3E08                  LD      A,08H          ; '0000 1000'
E818 D3CE                  OUT     (GDGMD),A      ; 700 Mode
E81A 3E01                  LD      A,01H          ; '0000 0001'
E81C D3CD                  OUT     (GDGRF),A      ; Read format reg.
E81E D3CC                  OUT     (GDGWF),A      ; Write format reg.
E820 D3E4                  OUT     (E4H),A        ; Initial Bank
E822 31F010                LD      SP,SP          ; Stack set (10F0H)
E825 CD3E07                CALL    ?MODE          ; 8255 MODE SET
E828 AF                    XOR     A
E829 110000                LD      DE,0000H
E82C CD0803                CALL    ?TMST          ; 8253 Mode set
E82F               ;
E82F               ;     Wait (25.342ms) for Printer
E82F               ;
E82F 012500                LD      BC,25H
E832 CD14E4                CALL    TIMW
E835               ;
E835               ;     Z80 PIO Initialize
E835               ;
E835 01FC04        PIOA:   LD      BC,04FCH       ; Port=A (C;port B;cont)
```

```
E838 219AEE             LD     HL,PIOIDA       ; Pio initialize data
E83B EDB3               OTIR
E83D 01FD04             LD     BC,04FDH        ; Port=B (C;port B;cont)
E840 EDB3               OTIR
E842           ;
E842           ;    SIO Initialize
E842           ;
E842 3E01               LD     A,01H
E844 D3F7               OUT    (SIOBC),A
E846 AF                 XOR    A
E847 D3F7               OUT    (SIOBC),A       ; WR1 Reg. Clear
E849           ;
E849           ;    (CTRL+Reset) Check
E849           ;
E849 CD320A             CALL   ?BRK            ; CTRL ?
E84C 3014               JR     NC,PSG0
E84E FE20               CP     20H             ; Key is 'CTRL' Key
E850 C262E8             JP     NZ,PSG0
E853 DBCE               IN     A,(CEH)         ; Mode switch
E855 E602               AND    2               ; bit 1 check
E857 2806               JR     Z,CMY00         ; 700 Mode
E859 AF                 XOR    A
E85A D3CE               OUT    (CEH),A         ; 800 Mode
E85C CDE1E8             CALL   PLTST           ; Pallet set
E85F C35B00    CMY00:   JP     CMY0            ; Bank & Jump
E862           ;
E862           ;    PSG Reset
E862           ;
E862 0604      PSG0:    LD     B,04H
E864 3E9F               LD     A,9FH
E866 D3F2      PSGSET:  OUT    (F2H),A
E868 C620               ADD    A,20H
E86A 10FA               DJNZ   PSGSET
E86C           ;
E86C           ;    Sound & Interrupt mask reset
E86C           ;
E86C 3E01      SORES:   LD     A,01H
E86E 3203E0             LD     (E003H),A       ; Sound mask reset
E871 3E05               LD     A,05H
E873 3203E0             LD     (E003H),A       ; Interrupt mask reset
E876           ;
E876           ;
E876           BGI0:    ENT
E876 06FF               LD     B,FFH           ; Buffer Clear
E878 21F110             LD     HL,NAME         ; 10F1-11F0H Clear
E87B CDD80F             CALL   ?CLER
E87E 3E16               LD     A,16H           ; Laster Clear
E880 CD1200             CALL   PRNT
E883 3E71               LD     A,71H           ; Back:Blue  Chara:White
E885 2100D8             LD     HL,D800H        ; Color Adrs
E888 CDD509             CALL   #CLR8
E88B 218D03             LD     HL,TIMIN        ; Interrupt jump routin
E88E 3EC3               LD     A,C3H
E890 323810             LD     (1038H),A
E893 223910             LD     (1039H),HL
E896 3E04               LD     A,04H           ; Normal tempo
E898 329E11             LD     (TEMPW),A
E89B CDBE02             CALL   MLDSP           ; Melody stop
E89E CD0900             CALL   NL
E8A1 CD7705             CALL   ?BEL
```

```
E8A4 3E01               LD     A,01H           ; Key in silent
E8A6 329D11             LD     (SWRK),A
E8A9           ;
E8A9           ;    PCG  CG ROM
E8A9           ;
E8A9 DBE0               IN     A,(E0H)         ; CG & Vram
E8AB 1100C0             LD     DE,C000H        ; Destination
E8AE 210010             LD     HL,1000H        ; Sorce
E8B1 010010             LD     BC,1000H        ; Size
E8B4 EDB0               LDIR
E8B6 DBE1               IN     A,(E1H)         ; 1000~1FFFH,C000~CFFFH
E8B8           ;
E8B8           ;    Key check
E8B8           ;
E8B8 CD1B00    KEYCK:   CALL   GETKY           ;Key check
E8BB FE4D               CP     'M'
E8BD 287C               JR     Z,MON           ; Monitor cmd.
E8BF FE51               CP     'Q'
E8C1 2858               JR     Z,KSJPQB        ; Quick disk
E8C3 FE43               CP     'C'
E8C5 287E               JR     Z,CSS           ; Cassette tape
E8C7           ;
E8C7           ;     Floppy check & boot
E8C7           ;
E8C7 CDD5E8             CALL   FDCC&           ; FD i/o check
E8CA 2026               JR     NZ,R12C&B
E8CC CD59EA    FDB:     CALL   DISCLR
E8CF CD4AE4             CALL   FDX
E8D2 C334EA             JP     ERRMG
E8D5           FDCC&:   ENT
E8D5 3EA5               LD     A,A5H
E8D7 47                 LD     B,A
E8D8 D3D9               OUT    (D9H),A
E8DA CDB3E6             CALL   DLY80U
E8DD DBD9               IN     A,(D9H)
E8DF B8                 CP     B
E8E0 C9                 RET
E8E1           ;
E8E1           ;
E8E1           ;        Pallet Reg. & Border Reg. set
E8E1           ;             PLT0~3  Black
E8E1           ;             Border  Black
E8E1           ;
E8E1           PLTST:   ENT
E8E1 E5                 PUSH   HL
E8E2 01F005             LD     BC,05F0H        ; C=port , B=counts
E8E5 21A2EE             LD     HL,PLTDT        ; Data
E8E8 EDB3               OTIR
E8EA AF                 XOR    A               ; Border  Black
E8EB 01CF06             LD     BC,06CFH        ; Port
E8EE ED79               OUT    (C),A
E8F0 E1                 POP    HL
E8F1 C9                 RET
E8F2           ;
E8F2           ;     MZ-1R12 check & boot
E8F2           ;
E8F2 0EF8      R12C&B:  LD     C,F8H
E8F4 CDBAE7             CALL   R12CK1          ; Set...cy='0',z='1'
E8F7 DAB7E9             JP     C,QBT
E8FA CDDAE6             CALL   1R12R0
```

```
E8FD 113AEE             LD    DE,12ERMG         ; 'SRAM:Check sum err'
E900 C234EA             JP    NZ,ERRMG
E903 C3D0E6             JP    EXRAM
E906          ;
E906          ;    Device selection
E906          ;
E906 CD43EA   SLMSG:    CALL  CLLET
E909 CD06EA   SELMG0:   CALL  SELMSG            ; Selection message
E90C CD91F1             CALL  NKYWT             ; Wait non key
E90F CD13EB   KSCAN0:   CALL  SIOCK             ; Sio hard check
E912 200A               JR    NZ,KSCAN1
E914 3E02               LD    A,02H
E916 CD15F1             CALL  SRCK0             ; Key scan
E919 FE7F               CP    7FH               ; 'Q'
E91B CAB7E9   KSJPQB:   JP    Z,QBT
E91E 3E04     KSCAN1:   LD    A,04H
E920 CD15F1             CALL  SRCK0
E923 FEDF               CP    DFH               ; 'C'
E925 281E               JR    Z,CSS
E927 4F                 LD    C,A
E928 CDD5E8             CALL  FDCC&             ; Fdc check
E92B 2005               JR    NZ,KSCAN2
E92D 79                 LD    A,C
E92E FEFB               CP    FBH               ; 'F'
E930 289A               JR    Z,FDB
E932 3E03     KSCAN2:   LD    A,03H
E934 CD15F1             CALL  SRCK0
E937 FEF7               CP    F7H               ; 'M'
E939 20D4               JR    NZ,KSCAN0
E93B          ;
E93B          ;    Monitor jump
E93B          ;
E93B CD59EA   MON:      CALL  DISCLR
E93E 114DEE             LD    DE,MONMSG         ; '**  MONITOR  **'
E941 DF                 RST   3
E942 C35EEA             JP    ST1X
E945          ;
E945          ;=========================
E945          ;
E945          ;  Cassette boot-up
E945          ;
E945          ;=========================
E945          ;
E945          CSS:      ENT
E945          ;
E945          ;    Motor check
E945          ;
E945 2102E0             LD    HL,E002H
E948 7E       WMOD0:    LD    A,(HL)
E949 E610               AND   10H
E94B 2025               JR    NZ,DSILF
E94D 23                 INC   HL                ; PC3:"L""H"
E94E 3E06               LD    A,06H             ;
E950 77                 LD    (HL),A            ;
E951 3C                 INC   A                 ; Acc07
E952 77                 LD    (HL),A            ;
E953 2B                 DEC   HL                ;
E954 7E                 LD    A,(HL)
E955 E610               AND   10H
E957 2019               JR    NZ,DSILF
```

```
E959          ;
E959          ; Disp 'Make ready CMT'
E959          ;
E959 CD59EA             CALL  DISCLR
E95C CD0600             CALL  LETNL
E95F CD0600             CALL  LETNL
E962 1198ED             LD    DE,IPLM1          ; 'Make ready CMT'
E965 CD4EEA             CALL  MSGLET
E968          ;
E968          ; Wait motor on
E968          ;
E968 CD1E00   WMOD1:    CALL  BRKEY             ; Break?
E96B 283F               JR    Z,TPERX0
E96D 7E                 LD    A,(HL)
E96E E610               AND   10H
E970 28F6               JR    Z,WMOD1
E972          ;
E972          ;    Disp 'IPL is looking for'
E972          ;
E972 CD59EA   DSILF:    CALL  DISCLR
E975 CD0600             CALL  LETNL
E978 11C3ED             LD    DE,IPLM4          ; 'IPL is looking ...'
E97B DF                 RST   3
E97C          ;
E97C          ;    Read information block
E97C          ;
E97C CD2700             CALL  RDINF
E97F DAAAE9             JP    C,TPERX
E982          ;
E982          ;    Disp 'IPL is loading...'
E982          ;
E982 CD59EA             CALL  DISCLR
E985 1188ED             LD    DE,IPLM0          ; 'IPL is loading '
E988 DF                 RST   3
E989 11F110             LD    DE,NAME
E98C DF                 RST   3
E98D          ;
E98D          ;    Load adrs set
E98D          ;
E98D 2A0411             LD    HL,(DTADR)
E990 D9                 EXX
E991 210012             LD    HL,1200H
E994 220411             LD    (DTADR),HL
E997          ;
E997          ;    Read data block
E997          ;
E997 CD2A00   RDTB0:    CALL  RDDAT
E99A DAAAE9             JP    C,TPERX
E99D          ;
E99D          ;    Exec load file
E99D          ;
E99D 010001   EXCMT:    LD    BC,0100H          ; Default code (CMT)
E9A0 D9                 EXX
E9A1 220411             LD    (DTADR),HL
E9A4 210211             LD    HL,SIZE
E9A7 C3FCEC             JP    EXF
E9AA          ;
E9AA          ;    Tape error
E9AA          ;
E9AA FE02     TPERX:    CP    02H               ; BREAK?
```

```
E9AC 1198ED     TPERX0: LD      DE,IPLM1        ; 'Make ready CMT'
E9AF 2803               JR      Z,TPERJP
E9B1 1104EE             LD      DE,ERRM2        ; 'CMT:Loading error'
E9B4 C334EA     TPERJP: JP      ERRMG
E9B7            ;
E9B7            ;==========================
E9B7            ;
E9B7            ;       Quick disk boot-up
E9B7            ;
E9B7            ;==========================
E9B7            ;
E9B7            QBT:    ENT
E9B7 CD13EB             CALL    SIOCK           ; Sio hard check
E9BA 3E02               LD      A,02H           ; Break code
E9BC 20EC               JR      NZ,TPERX
E9BE CDECEE             CALL    IOFRS           ; Iocs flag reset
E9C1 CD27EF             CALL    QDRCK           ; Media ready check
E9C4 11A7ED             LD      DE,IPLM2        ; 'Make ready QD'
E9C7 386B               JR      C,ERRMG
E9C9 CD59EA             CALL    DISCLR
E9CC 3E0D               LD      A,0DH
E9CE 32A311             LD      (BUFER),A
E9D1 CD5FF2             CALL    HDPCL           ; Head point clear
E9D4            ;
E9D4            ;       Error return set
E9D4            ;
E9D4 3E01               LD      A,01H
E9D6 323A11             LD      (QDCPA),A
E9D9 2104EA             LD      HL,QDLDER
E9DC 31EE10             LD      SP,10EEH
E9DF E3                 EX      (SP),HL
E9E0            ;
E9E0            ;
E9E0 CDF7EE     QBFLSH: CALL    FILSCH          ; File search
E9E3 DA02F2             JP      C,QERTRT
E9E6 3AF010             LD      A,(ATRB)
E9E9 FE01               CP      OBJCD
E9EB 1127EE             LD      DE,ERRM4        ; 'QD:File mode error'
E9EE 200A               JR      NZ,QDFMER
E9F0            ;
E9F0            ;       Quick disk boot
E9F0            ;
E9F0 1188ED             LD      DE,IPLM0        ; 'Ipl is loading'
E9F3 DF                 RST     3
E9F4 C3C2EE             JP      DSFLNA
E9F7 11A7ED     QDNTR:  LD      DE,IPLM2        ; 'Make ready QD'
E9FA D5         QDFMER: PUSH    DE
E9FB 3E06               LD      A,06H           ; Motor off
E9FD 323011             LD      (QDPA),A
EA00 CD10E0             CALL    QDIOS
EA03 D1                 POP     DE
EA04 182E       QDLDER: JR      ERRMG
EA06            ;
EA06            ;======================
EA06            ;
EA06            ;       Disp message
EA06            ;
EA06            ;======================
EA06            ;
EA06            ;
```

```
EA06                    ;       Selection message
EA06                    ;
EA06 CD0600             SELMSG: CALL    LETNL
EA09 1143ED                     LD      DE,SELM0        ; 'Please push key'
EA0C CD4EEA                     CALL    MSGLET
EA0F CD0600                     CALL    LETNL
EA12 CDD5E8                     CALL    FDCC&           ; Fdc check
EA15 2006                       JR      NZ,SELM00
EA17 1153ED                     LD      DE,SELM1        ; 'F:Floppy disk'
EA1A CD4EEA                     CALL    MSGLET
EA1D CD13EB             SELM00: CALL    SIOCK           ; Sio check
EA20 2006                       JR      NZ,SELM01
EA22 1161ED                     LD      DE,SELM2        ; 'Q:Quick disk'
EA25 CD4EEA                     CALL    MSGLET
EA28 116EED             SELM01: LD      DE,SELM3        ; 'C:Cassette tape'
EA2B CD4EEA                     CALL    MSGLET
EA2E 117EED                     LD      DE,SELM4        ; 'M:Rom monitor'
EA31 C30000     F               JP       MSGLET
EA34                    ;
EA34                    ;
EA34                    ;       Error message
EA34                    ;
EA34 CD59EA             ERRMG:  CALL    DISCLR
EA37 CD0600                     CALL    LETNL
EA3A CD0600                     CALL    LETNL
EA3D CD4EEA                     CALL    MSGLET
EA40 C309E9                     JP      SELMG0
EA43                    ;
EA43                    ;       Disp clear 3-let
EA43                    ;
EA43 CD59EA             CLLET:  CALL    DISCLR
EA46 0603                       LD      B,03H
EA48 CD0600             LET:    CALL    LETNL
EA4B 10FB                       DJNZ    LET
EA4D C9                         RET
EA4E                    ;
EA4E                    ;       Letnl-message
EA4E                    ;
EA4E 060C               MSGLET: LD      B,0CH
EA50 CD0C00             MSGPS:  CALL    PRNTS
EA53 10FB                       DJNZ    MSGPS
EA55 DF                         RST     3
EA56 C30000     F               JP       LETNL
EA59                    ;
EA59                    ;
EA59                    ;       Disp clear
EA59                    ;
EA59 3EC6               DISCLR: LD      A,C6H
EA5B C3DC0D                     JP      ?DPCT
EA5E                    ;
EA5E                    ;==========================
EA5E                    ;
EA5E                    ;       Monitor command
EA5E                    ;
EA5E                    ;==========================
EA5E                    ;
EA5E                    ;       Get command routine
EA5E                    ;
EA5E                    STix:   ENT
```

```
EA5E 31F010                    LD      SP,SP
EA61 CD0900                    CALL    NL
EA64 3E2A                      LD      A,2AH          ;"*"
EA66 CD1200                    CALL    PRNT
EA69 11A311                    LD      DE,BUFER
EA6C CD0300                    CALL    GETL
EA6F 1A                ST2X:   LD      A,(DE)
EA70 13                        INC     DE
EA71 FE0D                      CP      0DH
EA73 28E9                      JR      Z,ST1X
EA75 FE4A                      CP      'J'            ; JUMP
EA77 2830                      JR      Z,GOTOX
EA79 FE4C                      CP      'L'            ; Load cmt
EA7B CA4CEB                    JP      Z,LOADX
EA7E FE46                      CP      'F'            ; Floppy boot
EA80 287E                      JR      Z,FDCK
EA82 FE42                      CP      'B'            ; Bell
EA84 CA1EEC                    JP      Z,SGX
EA87 FE4D                      CP      'M'            ; Memory correction
EA89 CA7BEB                    JP      Z,MCORX
EA8C FE53                      CP      'S'            ; Save CMT
EA8E CAAEEB                    JP      Z,SAVEX
EA91 FE56                      CP      'V'            ; Verify
EA93 CA00EC                    JP      Z,VRFYX
EA96 FE44                      CP      'D'            ; Dump memory
EA98 CA29EC                    JP      Z,DUMPX
EA9B FE51                      CP      'Q'            ; Quick disk cmd.
EA9D 2816                      JR      Z,QUICK
EA9F FE45                      CP      'E'            ; Exit rambrd(1R12) cmd.
EAA1 2842                      JR      Z,RAMBRD
EAA3 FE47                      CP      'G'            ; Call cmd.
EAA5 2806                      JR      Z,CAL0
EAA7 18C6                      JR      ST2X
EAA9                   ;
EAA9                   ;       Jump command
EAA9                   ;
EAA9 CD42EB            GOTOX:  CALL    HEXIYX
EAAC E9                GOT0:   JP      (HL)
EAAD                   ;
EAAD                   ;       Call command
EAAD                   ;
EAAD CD42EB            CAL0:   CALL    HEXIYX
EAB0 CDACEA                    CALL    GOT0
EAB3 18A9                      JR      ST1X
EAB5                   ;
EAB5                   ;       Quick disk CMD.
EAB5                   ;
EAB5 CD13EB            QUICK:  CALL    SIOCK          ; Sio hard check
EAB8 20A4                      JR      NZ,ST1X
EABA CDECEE                    CALL    IOFRS          ; IOCS frag reset
EABD 210000                    LD      HL,0000H       ; Frag reset
EAC0 223A11                    LD      (QDCPA),HL
EAC3 1A                        LD      A,(DE)
EAC4 FE4C                      CP      'L'            ; Load QD
EAC6 CAA7EE                    JP      Z,QL
EAC9 FE53                      CP      'S'            ; Save QD
EACB CA2EEF                    JP      Z,QS
EACE FE43                      CP      'C'            ; Copy QD
EAD0 CA2CF1                    JP      Z,QC
EAD3 FE46                      CP      'F'            ; Format QD
```

```
EAD5 CAB5F0                    JP      Z,QF
EAD8 FE58                      CP      'X'            ; Xfer (CMTQD)
EADA CAA2F1                    JP      Z,QX
EADD FE44                      CP      'D'            ; Directry
EADF CAEFEF                    JP      Z,QD
EAE2 C35EEA            JST1X:  JP      ST1X
EAE5 0EF8              RAMBRD: LD      C,F8H
EAE7 CDBAE7                    CALL    R12CK1
EAEA 3007                      JR      NC,RMBD1
EAEC 0EA8                      LD      C,A8H
EAEE CDBAE7                    CALL    R12CK1
EAF1 38EF                      JR      C,JST1X
EAF3 1A                RMBD1:  LD      A,(DE)
EAF4 FE42                      CP      'B'
EAF6 CACAE6                    JP      Z,1R12RD       ; 1R12 boot-up
EAF9 FE53                      CP      'S'
EAFB CA45E7                    JP      Z,1R12WR       ; 1R12 Write
EAFE 18E2                      JR      JST1X
EB00 1A                FDCK:   LD      A,(DE)
EB01 FE0D                      CP      0DH
EB03 20DD                      JR      NZ,JST1X
EB05 CDD5E8                    CALL    FDCC&          ; FD i/o check
EB08 20D8                      JR      NZ,JST1X
EB0A CD4AE4                    CALL    FDX
EB0D CD0900                    CALL    NL
EB10 DF                        RST     3
EB11 18CF                      JR      JST1X
EB13                   ;
EB13                   ;    Sio hard check
EB13                   ;
EB13 3E02              SIOCK:  LD      A,02H
EB15 D3F7                      OUT     (SIOBC),A
EB17 3EA5                      LD      A,A5H
EB19 D3F7                      OUT     (SIOBC),A
EB1B 3E02                      LD      A,02H
EB1D D3F7                      OUT     (SIOBC),A
EB1F DBF7                      IN      A,(SIOBC)
EB21 FEA5                      CP      A5H
EB23 C9                        RET
EB24                   ;
EB24                   ;    Error (loading)
EB24                   ;
EB24                   ?ERX:   ENT
EB24 1129F3                    LD      DE,MGBRK       ; 'Break!'
EB27 FE02                      CP      02H
EB29 2803                      JR      Z,?ERX0
EB2B 114701                    LD      DE,MSGE1       ; 'Check sum err.'
EB2E CD0900            ?ERX0:  CALL    NL
EB31 DF                        RST     3
EB32 18AE                      JR      JST1X
EB34                   ;
EB34                   ; Getline & break in check
EB34                   ;       exit break in then jump (ST1X)
EB34                   ;       acc=top of line data
EB34                   ;
EB34                   BGETLX: ENT
EB34 E3                        EX      (SP),HL
EB35 C1                        POP     BC
EB36 11A311                    LD      DE,BUFER
EB39 CD0300                    CALL    GETL
```

```
EB3C 1A                    LD      A,(DE)
EB3D FE1B                  CP      1BH
EB3F 28A1                  JR      Z,JST1X
EB41 E9                    JP      (HL)
EB42               ;
EB42               ; Ascii to hex convert
EB42               ;       input (DE)=Ascii
EB42               ;       CY=1 then jump (ST1X)
EB42               ;
EB42               HEXIYX: ENT
EB42 FDE3                  EX      (SP),IY
EB44 F1                    POP     AF
EB45 CD1004                CALL    HLHEX
EB48 3898                  JR      C,JST1X
EB4A FDE9                  JP      (IY)
EB4C               ;
EB4C               ;      load command
EB4C               ;
EB4C               LOADX:  ENT
EB4C CD54EB                CALL    LD00                ; Read inf & data
EB4F 38D3                  JR      C,?ERX
EB51 C39DE9                JP      EXCMT               ; Exec load file
EB54               ;
EB54               ;
EB54               LD00:   ENT
EB54               ;
EB54               ;  Read information block
EB54               ;
EB54 CDD804                CALL    ?RDI
EB57 D8                    RET     C
EB58               ;
EB58               ; Disp 'Loading...'
EB58               ;
EB58 CD0900        LOA0X:  CALL    NL
EB5B 1191EE                LD      DE,MSGLD            ; 'loading '
EB5E DF                    RST     3
EB5F 11F110                LD      DE,NAME
EB62 DF                    RST     3
EB63               ;
EB63               ;      Load adrs set
EB63               ;
EB63 2A0411                LD      HL,(DTADR)
EB66 D9                    EXX
EB67 210012                LD      HL,1200H
EB6A 220411                LD      (DTADR),HL
EB6D               ;
EB6D               ;      Read data block
EB6D               ;
EB6D C3F804        RDDTBL: JP      ?RDD
EB70               ;
EB70               ;
EB70               ;   Loading from CMT ($1200~)
EB70               ;
EB70               LDCMT:  ENT
EB70 CD54EB                CALL    LD00
EB73 38AF                  JR      C,?ERX
EB75 D9                    EXX
EB76 220411                LD      (DTADR),HL
EB79 187D                  JR      SAC01
EB7B               ;
```

```
EB7B               ;
EB7B               ;      Memory correction
EB7B               ;
EB7B               ;
EB7B               MCORX:  ENT
EB7B CD42EB                CALL    HEXIYX
EB7E               MCR1X:  ENT
EB7E CDFA05                CALL    NLPHL
EB81 CDB103                CALL    SPHEX
EB84 CD2009                CALL    ?PRTS
EB87 CD34EB                CALL    BGETLX
EB8A CD1004                CALL    HLHEX
EB8D 381B                  JR      C,MCR3X
EB8F CDA602                CALL    .4DE
EB92 13                    INC     DE
EB93 CD1F04                CALL    2HEX
EB96 38E6                  JR      C,MCR1X
EB98 BE                    CP      (HL)
EB99 20E3                  JR      NZ,MCR1X
EB9B 13                    INC     DE
EB9C 1A                    LD      A,(DE)
EB9D FE0D                  CP      0DH
EB9F 2806                  JR      Z,MCR2X
EBA1 CD1F04                CALL    2HEX
EBA4 38D8                  JR      C,MCR1X
EBA6 77                    LD      (HL),A
EBA7               MCR2X:  ENT
EBA7 23                    INC     HL
EBA8 18D4                  JR      MCR1X
EBAA 60            MCR3X:  LD      H,B
EBAB 69                    LD      L,C
EBAC 18D0                  JR      MCR1X
EBAE               ;
EBAE               ;      Save command
EBAE               ;
EBAE CD8DEC        SAVEX:  CALL    FNINP
EBB1 21A311                LD      HL,BUFER
EBB4 11F110                LD      DE,NAME
EBB7 011100                LD      BC,0011H
EBBA EDB0                  LDIR
EBBC CDD3EC                CALL    TAINP
EBBF 220411                LD      (DTADR),HL          ; Data adrs
EBC2 CDD8EC                CALL    EAINP
EBC5 ED4B0411              LD      BC,(DTADR)
EBC9 37                    SCF
EBCA ED42                  SBC     HL,BC
EBCC 23                    INC     HL
EBCD 23                    INC     HL
EBCE 220211                LD      (SIZE),HL           ; Byte size
EBD1 CDDDEC                CALL    XAINP
EBD4 220611                LD      (EXADR),HL          ; Exec adrs
EBD7 3E01                  LD      A,01H
EBD9 32F010                LD      (ATRB),A
EBDC CD3604                CALL    ?WRI
EBDF DA24EB        SAVJE:  JP      C,?ERX
EBE2 CDF3EB                CALL    SAC00
EBE5 C35EEA        SAVJP:  JP      ST1X
EBE8               ;
EBE8               ;      Save CMT ($1200~)
EBE8               ;
```

```
EBE8                SACMT:   ENT
EBE8 CD3604                  CALL   ?WRI              ; Write infomation
EBEB 38F2                    JR     C,SAVJE
EBED 210012                  LD     HL,1200H
EBF0 220411                  LD     (DTADR),HL
EBF3 CD7504         SAC00:   CALL   ?WRD              ; Write data
EBF6 38E7                    JR     C,SAVJE
EBF8 CD0900         SAC01:   CALL   NL
EBFB 114209                  LD     DE,MSGOK          ; 'OK'
EBFE DF                      RST    3
EBFF C9                      RET
EC00                ;
EC00                ;
EC00                ;     Verifying command
EC00                ;
EC00                ;
EC00                VRFYX:   ENT
EC00 CD8805                  CALL   ?VRFY
EC03 DA24EB         VRFYE:   JP     C,?ERX
EC06 C395EF                  JP     QDOKM             ; 'OK' MSG.
EC09                ;
EC09                VFCMT:   ENT
EC09 CD10EC                  CALL   VRF120
EC0C 38F5                    JR     C,VRFYE
EC0E 18E8                    JR     SAC01             ; 'OK' MSG.
0593  P             ?VRFY0:  EQU    0593H
EC10 F3             VRF120:  DI
EC11 D5                      PUSH   DE
EC12 C5                      PUSH   BC
EC13 E5                      PUSH   HL
EC14 ED4B0211                LD     BC,(SIZE)
EC18 210012                  LD     HL,1200H
EC1B C39305                  JP     ?VRFY0
EC1E                ;
EC1E                ;  Keyin bell command 'B'
EC1E                ;
EC1E 3A9D11         SGX:     LD     A,(SWRK)
EC21 1F                      RRA
EC22 3F                      CCF
EC23 17                      RLA
EC24 329D11                  LD     (SWRK),A
EC27 18BC           SGJP:    JR     SAVJP             ; JP ST1X
EC29                ;
EC29                ;  Memory dump command
EC29                ;
EC29                DUMPX:   ENT
EC29 CD42EB                  CALL   HEXIYX
EC2C CDA602                  CALL   .4DE
EC2F E5                      PUSH   HL
EC30 CD1004                  CALL   HLHEX
EC33 D1                      POP    DE
EC34 3851                    JR     C,DUM1X
EC36 EB                      EX     DE,HL
EC37 0608           DUM3X:   LD     B,08H
EC39 0E17                    LD     C,17H
EC3B CDFA05                  CALL   NLPHL
EC3E CDB103         DUM2X:   CALL   SPHEX
EC41 23                      INC    HL
EC42 F5                      PUSH   AF
EC43 3A7111                  LD     A,(DSPXY)
```

```
EC46 81                      ADD    A,C
EC47 327111                  LD     (DSPXY),A
EC4A F1                      POP    AF
EC4B FE20                    CP     20H
EC4D 3002                    JR     NC,+4
EC4F 3E2E                    LD     A,2EH
EC51 CDB90B                  CALL   ?ADCN
EC54 CD6C09                  CALL   PRNT3
EC57 3A7111                  LD     A,(DSPXY)
EC5A 0C                      INC    C
EC5B 91                      SUB    C
EC5C 327111                  LD     (DSPXY),A
EC5F 0D                      DEC    C
EC60 0D                      DEC    C
EC61 0D                      DEC    C
EC62 E5                      PUSH   HL
EC63 ED52                    SBC    HL,DE
EC65 E1                      POP    HL
EC66 281C                    JR     Z,DUM1X-3
EC68 3EF8                    LD     A,F8H
EC6A 3200E0                  LD     (KEYPA),A
EC6D 00                      NOP
EC6E 3A01E0                  LD     A,(KEYPB)
EC71 FEFE                    CP     FEH
EC73 2003                    JR     NZ,+5
EC75 CDA60D                  CALL   ?BLNK
EC78 10C4                    DJNZ   DUM2X
EC7A CDCA08                  CALL   ?KEY
EC7D B7                      OR     A
EC7E 28FA                    JR     Z,-4
EC80 CD320A                  CALL   ?BRK
EC83 20B2                    JR     NZ,DUM3X
EC85 18A0                    JR     SGJP              ; JP ST1X
EC87 21A000         DUM1X:   LD     HL,00A0H
EC8A 19                      ADD    HL,DE
EC8B 18A9                    JR     DUM3X-1
EC8D                ;
EC8D                ;    Input file name
EC8D                ;
EC8D                FNINP:   ENT
EC8D CD0900                  CALL   NL
EC90 1165EE                  LD     DE,MSGSV0         ; 'Filename? '
EC93 DF                      RST    3
EC94 11A311                  LD     DE,BUFER
EC97 CD0300                  CALL   GETL
EC9A 1A                      LD     A,(DE)
EC9B FE1B                    CP     1BH               ; Break ?
EC9D 2005                    JR     NZ,FNINP0
EC9F 215EEA         FNINPR:  LD     HL,ST1X
ECA2 E3                      EX     (SP),HL
ECA3 C9                      RET
ECA4                ;
ECA4                ;
ECA4 0600           FNINP0:  LD     B,00H
ECA6 11AD11                  LD     DE,BUFER+10
ECA9 21A311                  LD     HL,BUFER
ECAC 1A                      LD     A,(DE)
ECAD FE0D                    CP     0DH
ECAF 2820                    JR     Z,FNINP6
ECB1                ;
```

```
ECB1                  ;     Space check
ECB1                  ;
ECB1 FE20             FNINP1: CP     ' '              ; Space ?
ECB3 2004                     JR     NZ,FNINP2
ECB5 13                       INC    DE
ECB6 1A                       LD     A,(DE)
ECB7 18F8                     JR     FNINP1
ECB9                  ;
ECB9                  ;     " ?
ECB9                  ;
ECB9 FE22             FNINP2: CP     22H
ECBB 2808                     JR     Z,FNINP4
ECBD                  ;
ECBD                  ;    Trans (DE)(HL)
ECBD                  ;
ECBD 77               FNINP3: LD     (HL),A
ECBE 23                       INC    HL
ECBF 04                       INC    B
ECC0 3E11                     LD     A,11H
ECC2 B8                       CP     B
ECC3 28C8                     JR     Z,FNINP
ECC5                  ;
ECC5                  ;   Filename end check
ECC5                  ;
ECC5 13               FNINP4: INC    DE
ECC6 1A                       LD     A,(DE)
ECC7 FE22                     CP     22H              ; " ?
ECC9 2804                     JR     Z,FNINP5
ECCB FE0D                     CP     0DH
ECCD 20EE                     JR     NZ,FNINP3
ECCF 3E0D             FNINP5: LD     A,0DH
ECD1 77               FNINP6: LD     (HL),A
ECD2 C9                       RET
ECD3                  ;
ECD3                  ;   Input save condition
ECD3                  ;
ECD3                  TAINP:  ENT
ECD3 1170EE                   LD     DE,MSGTA         ; 'Top adrs?'
ECD6 1808                     JR     AINP0
ECD8                  EAINP:  ENT
ECD8 117BEE                   LD     DE,MSGEA         ; 'End adrs?'
ECDB 1803                     JR     AINP0
ECDD                  XAINP:  ENT
ECDD 1186EE                   LD     DE,MSGXA         ; 'Exc adrs?'
ECE0 CD0900           AINP0:  CALL   NL
ECE3 DF                       RST    3
ECE4 D5                       PUSH   DE
ECE5 11A311                   LD     DE,BUFER
ECE8 CD0300                   CALL   GETL
ECEB 1A                       LD     A,(DE)
ECEC FE1B                     CP     1BH              ; Break ?
ECEE D1                       POP    DE
ECEF 28AE                     JR     Z,FNINPR
ECF1 D5                       PUSH   DE
ECF2 11AD11                   LD     DE,BUFER+10
ECF5 CD1004                   CALL   HLHEX
ECF8 D1                       POP    DE
ECF9 38E5                     JR     C,AINP0
ECFB C9                       RET
ECFC                  ;
```

```
ECFC                  ;     Exec file
ECFC                  ;        IN HL ; Data size point
ECFC                  ;           BC'; Default code
ECFC                  ;
ECFC                  EXF:    ENT
ECFC DBCE                     IN     A,(CEH)          ; Mode sw.
ECFE CB4F                     BIT    1,A
ED00 2806                     JR     Z,70MOD
ED02 AF                       XOR    A
ED03 D3CE                     OUT    (CEH),A          ; 800 Mode set
ED05 CDE1E8                   CALL   PLTST            ; Pallet set
ED08                  ;
ED08 4E               70MOD:  LD     C,(HL)           ; BCData adrs
ED09 23                       INC    HL
ED0A 46                       LD     B,(HL)
ED0B 23                       INC    HL
ED0C 5E                       LD     E,(HL)           ; DEData adrs
ED0D 7E                       LD     A,(HL)
ED0E 23                       INC    HL
ED0F 56                       LD     D,(HL)
ED10 B6                       OR     (HL)
ED11 D5                       PUSH   DE               ; HLExec adrs
ED12 23                       INC    HL
ED13 5E                       LD     E,(HL)
ED14 B6                       OR     (HL)
ED15 23                       INC    HL
ED16 56                       LD     D,(HL)
ED17 B6                       OR     (HL)
ED18 D5                       PUSH   DE
ED19 DDE1                     POP    IX
ED1B D1                       POP    DE
ED1C 2002                     JR     NZ,BLKTR         ; Data adrs & Exec adrs
= 0000H ?
ED1E D3E0                     OUT    (E0H),A          ; Bank $0000~8000 Dram
ED20                  ;
ED20 210012           BLKTR:  LD     HL,1200H
ED23 3E12                     LD     A,12H
ED25 BA                       CP     D
ED26 3806                     JR     C,LDDEC
ED28 2014                     JR     NZ,LDINC
ED2A AF                       XOR    A
ED2B BB                       CP     E
ED2C 3010                     JR     NC,LDINC
ED2E 0B               LDDEC:  DEC    BC
ED2F E5                       PUSH   HL
ED30 210000                   LD     HL,0000H
ED33 19                       ADD    HL,DE
ED34 09                       ADD    HL,BC
ED35 E5                       PUSH   HL
ED36 D1                       POP    DE
ED37 E1                       POP    HL
ED38 09                       ADD    HL,BC
ED39 03                       INC    BC
ED3A EDB8                     LDDR
ED3C 1802                     JR     EXF0
ED3E                  ;
ED3E EDB0             LDINC:  LDIR
ED40 D9               EXF0:   EXX                     ; BCDefault code
ED41 DDE9                     JP     (IX)
ED43                  ;
ED43                  ;
```

```
ED43                    ;
ED43 50                 SELM0:  DEFB   50H          ; 'Please push key'
ED44 B8                         DEFB   B8H          ; L
ED45 92                         DEFB   92H          ; E
ED46 A1                         DEFB   A1H          ; A
ED47 A4                         DEFB   A4H          ; S
ED48 92                         DEFB   92H          ; E
ED49 20                         DEFB   20H          ;
ED4A 9E                         DEFB   9EH          ; P
ED4B A5                         DEFB   A5H          ; U
ED4C A4                         DEFB   A4H          ; S
ED4D 98                         DEFB   98H          ; H
ED4E 20                         DEFB   20H
ED4F A9                         DEFB   A9H          ; K
ED50 92                         DEFB   92H          ; E
ED51 BD                         DEFB   BDH          ; Y
ED52 0D                         DEFB   0DH
ED53                    ;
ED53 463A46             SELM1:  DEFM   'F:F'        ; 'F:Floppy disk'
ED56 B8                         DEFB   B8H          ; L
ED57 B7                         DEFB   B7H          ; O
ED58 9E                         DEFB   9EH          ; P
ED59 9E                         DEFB   9EH          ; P
ED5A BD                         DEFB   BDH          ; Y
ED5B 20                         DEFB   20H          ;
ED5C 9C                         DEFB   9CH          ; D
ED5D A6                         DEFB   A6H          ; I
ED5E A4                         DEFB   A4H          ; S
ED5F A9                         DEFB   A9H          ; K
ED60 0D                         DEFB   0DH
ED61                    ;
ED61 513A51             SELM2:  DEFM   'Q:Q'        ; 'Q:Quick disk'
ED64 A5                         DEFB   A5H          ; U
ED65 A6                         DEFB   A6H          ; I
ED66 9F                         DEFB   9FH          ; C
ED67 A9                         DEFB   A9H          ; K
ED68 20                         DEFB   20H          ;
ED69 9C                         DEFB   9CH          ; D
ED6A A6                         DEFB   A6H          ; I
ED6B A4                         DEFB   A4H          ; S
ED6C A9                         DEFB   A9H          ; K
ED6D 0D                         DEFB   0DH
ED6E                    ;
ED6E 433A43             SELM3:  DEFM   'C:C'        ; 'C:Cassette tape'
ED71 A1                         DEFB   A1H          ; A
ED72 A4                         DEFB   A4H          ; S
ED73 A4                         DEFB   A4H          ; S
ED74 92                         DEFB   92H          ; E
ED75 96                         DEFB   96H          ; T
ED76 96                         DEFB   96H          ; T
ED77 92                         DEFB   92H          ; E
ED78 20                         DEFB   20H          ;
ED79 96                         DEFB   96H          ; T
ED7A A1                         DEFB   A1H          ; A
ED7B 9E                         DEFB   9EH          ; P
ED7C 92                         DEFB   92H          ; E
ED7D 0D                         DEFB   0DH
ED7E                    ;
ED7E 4D3A4D             SELM4:  DEFM   'M:M'        ; 'M:Monitor'
ED81 B7                         DEFB   B7H          ; O
```

```
ED82 B0                         DEFB   B0H          ; N
ED83 A6                         DEFB   A6H          ; I
ED84 96                         DEFB   96H          ; T
ED85 B7                         DEFB   B7H          ; O
ED86 9D                         DEFB   9DH          ; R
ED87 0D                         DEFB   0DH
ED88                    ;
ED88                    IPLM0:  ENT
ED88 49504C20                   DEFM   'IPL '       ; 'IPL is loading'
ED8C A6                         DEFB   A6H          ; I
ED8D A4                         DEFB   A4H          ; S
ED8E 20                         DEFB   20H          ;
ED8F B8                         DEFB   B8H          ; L
ED90 B7                         DEFB   B7H          ; O
ED91 A1                         DEFB   A1H          ; A
ED92 9C                         DEFB   9CH          ; D
ED93 A6                         DEFB   A6H          ; I
ED94 B0                         DEFB   B0H          ; N
ED95 97                         DEFB   97H          ; G
ED96 20                         DEFB   20H          ;
ED97 0D                         DEFB   0DH
ED98 4D                 IPLM1:  DEFM   'M'          ; 'Make ready CMT'
ED99 A1                         DEFB   A1H          ; A
ED9A A9                         DEFB   A9H          ; K
ED9B 92                         DEFB   92H          ; E
ED9C 20                         DEFB   20H          ;
ED9D 9D                         DEFB   9DH          ; R
ED9E 92                         DEFB   92H          ; E
ED9F A1                         DEFB   A1H          ; A
EDA0 9C                         DEFB   9CH          ; D
EDA1 BD                         DEFB   BDH          ; Y
EDA2 20434D54                   DEFM   ' CMT'
EDA6 0D                         DEFB   0DH
EDA7 4D                 IPLM2:  DEFM   'M'          ; 'Make ready QD'
EDA8 A1                         DEFB   A1H          ; A
EDA9 A9                         DEFB   A9H          ; K
EDAA 92                         DEFB   92H          ; E
EDAB 20                         DEFB   20H          ;
EDAC 9D                         DEFB   9DH          ; R
EDAD 92                         DEFB   92H          ; E
EDAE A1                         DEFB   A1H          ; A
EDAF 9C                         DEFB   9CH          ; D
EDB0 BD                         DEFB   BDH          ; Y
EDB1 205144                     DEFM   ' QD'
EDB4 0D                         DEFB   0DH
EDB5                    IPLM3:  ENT
EDB5 4D                         DEFM   'M'          ; 'Make ready FD'
EDB6 A1                         DEFB   A1H          ; A
EDB7 A9                         DEFB   A9H          ; K
EDB8 92                         DEFB   92H          ; E
EDB9 20                         DEFB   20H          ;
EDBA 9D                         DEFB   9DH          ; R
EDBB 92                         DEFB   92H          ; E
EDBC A1                         DEFB   A1H          ; A
EDBD 9C                         DEFB   9CH          ; D
EDBE BD                         DEFB   BDH          ; Y
EDBF 204644                     DEFM   ' FD'
EDC2 0D                         DEFB   0DH
EDC3 20202020           IPLM4:  DEFM   '      IPL ' ; 'IPL is looking for ..
.'
EDC7 2049504C
```

```
EDCB 20
EDCC A6                    DEFB   A6H        ; I
EDCD A4                    DEFB   A4H        ; S
EDCE 20                    DEFB   20H        ;
EDCF B8                    DEFB   B8H        ; L
EDD0 B7                    DEFB   B7H        ; O
EDD1 B7                    DEFB   B7H        ; O
EDD2 A9                    DEFB   A9H        ; K
EDD3 A6                    DEFB   A6H        ; I
EDD4 B0                    DEFB   B0H        ; N
EDD5 97                    DEFB   97H        ; G
EDD6 20                    DEFB   20H        ;
EDD7 AA                    DEFB   AAH        ; F
EDD8 B7                    DEFB   B7H        ; O
EDD9 9D                    DEFB   9DH        ; R
EDDA 20                    DEFB   20H        ;
EDDB A1                    DEFB   A1H        ; A
EDDC 20                    DEFB   20H        ;
EDDD 9E                    DEFB   9EH        ; P
EDDE 9D                    DEFB   9DH        ; R
EDDF B7                    DEFB   B7H        ; O
EDE0 97                    DEFB   97H        ; G
EDE1 9D                    DEFB   9DH        ; R
EDE2 A1                    DEFB   A1H        ; A
EDE3 B3                    DEFB   B3H        ; M
EDE4 0D                    DEFB   0DH
EDE5              ERRM0:   ENT
EDE5 46443A4C              DEFM   'FD:L'     ; 'FD:Loadig error'
EDE9 B7                    DEFB   B7H        ; O
EDEA A1                    DEFB   A1H        ; A
EDEB 9C                    DEFB   9CH        ; D
EDEC A6                    DEFB   A6H        ; I
EDED B0                    DEFB   B0H        ; N
EDEE 97                    DEFB   97H        ; G
EDEF 20                    DEFB   20H        ;
EDF0 92                    DEFB   92H        ; E
EDF1 9D                    DEFB   9DH        ; R
EDF2 9D                    DEFB   9DH        ; R
EDF3 B7                    DEFB   B7H        ; O
EDF4 9D                    DEFB   9DH        ; R
EDF5 0D                    DEFB   0DH
EDF6              ;
EDF6              ERRM1:   ENT
EDF6 46443A4E              DEFM   'FD:N'     ; 'FD:Not master'
EDFA B7                    DEFB   B7H        ; O
EDFB 96                    DEFB   96H        ; T
EDFC 20                    DEFB   20H        ;
EDFD B3                    DEFB   B3H        ; M
EDFE A1                    DEFB   A1H        ; A
EDFF A4                    DEFB   A4H        ; S
EE00 96                    DEFB   96H        ; T
EE01 92                    DEFB   92H        ; E
EE02 9D                    DEFB   9DH        ; R
EE03 0D                    DEFB   0DH
EE04              ;
EE04 434D543A     ERRM2:   DEFM   'CMT:L'    ; 'CMT:Loading error'
EE08 4C
EE09 B7                    DEFB   B7H        ; O
EE0A A1                    DEFB   A1H        ; A
EE0B 9C                    DEFB   9CH        ; D
```

```
EE0C A6                    DEFB   A6H        ; I
EE0D B0                    DEFB   B0H        ; N
EE0E 97                    DEFB   97H        ; G
EE0F 20                    DEFB   20H        ;
EE10 92                    DEFB   92H        ; E
EE11 9D                    DEFB   9DH        ; R
EE12 9D                    DEFB   9DH        ; R
EE13 B7                    DEFB   B7H        ; O
EE14 9D                    DEFB   9DH        ; R
EE15 0D                    DEFB   0DH
EE16              ;
EE16 51443A4C     ERRM3:   DEFM   'QD:L'     ; 'QD:Loading error'
EE1A B7                    DEFB   B7H        ; O
EE1B A1                    DEFB   A1H        ; A
EE1C 9C                    DEFB   9CH        ; D
EE1D A6                    DEFB   A6H        ; I
EE1E B0                    DEFB   B0H        ; N
EE1F 97                    DEFB   97H        ; G
EE20 20                    DEFB   20H        ;
EE21 92                    DEFB   92H        ; E
EE22 9D                    DEFB   9DH        ; R
EE23 9D                    DEFB   9DH        ; R
EE24 B7                    DEFB   B7H        ; O
EE25 9D                    DEFB   9DH        ; R
EE26 0D                    DEFB   0DH
EE27              ;
EE27 51443A46     ERRM4:   DEFM   'QD:F'     ; 'QD:File mode error'
EE2B A6                    DEFB   A6H        ; I
EE2C B8                    DEFB   B8H        ; L
EE2D 92                    DEFB   92H        ; E
EE2E 20                    DEFB   20H        ;
EE2F B3                    DEFB   B3H        ; M
EE30 B7                    DEFB   B7H        ; O
EE31 9C                    DEFB   9CH        ; D
EE32 92                    DEFB   92H        ; E
EE33 20                    DEFB   20H        ;
EE34 92                    DEFB   92H        ; E
EE35 9D                    DEFB   9DH        ; R
EE36 9D                    DEFB   9DH        ; R
EE37 B7                    DEFB   B7H        ; O
EE38 9D                    DEFB   9DH        ; R
EE39 0D                    DEFB   0DH
EE3A              ;
EE3A 5352414D     12ERMG:  DEFM   'SRAM:C'   ; 'SRAM:Check sum error'
EE3E 3A43
EE40 98                    DEFB   98H        ; H
EE41 92                    DEFB   92H        ; E
EE42 9F                    DEFB   9FH        ; C
EE43 A9                    DEFB   A9H        ; K
EE44 20                    DEFB   20H        ;
EE45 A4                    DEFB   A4H        ; S
EE46 A5                    DEFB   A5H        ; U
EE47 B3                    DEFB   B3H        ; M
EE48 20                    DEFB   20H        ;
EE49 92                    DEFB   92H        ; E
EE4A 9D                    DEFB   9DH        ; R
EE4B 9D                    DEFB   9DH        ; R
EE4C 0D                    DEFB   0DH
EE4D              ;
EE4D 2A2A2020     MONMSG:  DEFM   '** MONITOR 9Z-504M **'
```

```
EE51 4D4F4E49
EE55 544F5220
EE59 395A2D35
EE5D 30344D20
EE61 202A2A
EE64 0D                        DEFB   0DH
EE65 46               MSGSV0:  DEFM   'F'           ; 'Filename? '
EE66 A6                        DEFB   A6H           ; I
EE67 B8                        DEFB   B8H           ; L
EE68 92                        DEFB   92H           ; E
EE69 B0                        DEFB   B0H           ; N
EE6A A1                        DEFB   A1H           ; A
EE6B B3                        DEFB   B3H           ; M
EE6C 92                        DEFB   92H           ; E
EE6D 3F                        DEFB   3FH           ; ?
EE6E 20                        DEFB   20H           ;
EE6F 0D                        DEFB   0DH
EE70 54               MSGTA:   DEFM   'T'           ; 'Top adrs? '
EE71 B7                        DEFB   B7H           ; O
EE72 9E                        DEFB   9EH           ; P
EE73 20                        DEFB   20H           ;
EE74 A1                        DEFB   A1H           ; A
EE75 9C                        DEFB   9CH           ; D
EE76 9D                        DEFB   9DH           ; R
EE77 A4                        DEFB   A4H           ; S
EE78 3F                        DEFB   3FH           ; ?
EE79 20                        DEFB   20H           ;
EE7A 0D                        DEFB   0DH
EE7B 45               MSGEA:   DEFM   'E'           ; 'End adrs? '
EE7C B0                        DEFB   B0H           ; N
EE7D 9C                        DEFB   9CH           ; D
EE7E 20                        DEFB   20H           ;
EE7F A1                        DEFB   A1H           ; A
EE80 9C                        DEFB   9CH           ; D
EE81 9D                        DEFB   9DH           ; R
EE82 A4                        DEFB   A4H           ; S
EE83 3F                        DEFB   3FH           ; ?
EE84 20                        DEFB   20H           ;
EE85 0D                        DEFB   0DH
EE86 45               MSGXA:   DEFM   'E'           ; 'Exc adrs? '
EE87 9B                        DEFB   9BH           ; X
EE88 9F                        DEFB   9FH           ; C
EE89 20                        DEFB   20H           ;
EE8A A1                        DEFB   A1H           ; A
EE8B 9C                        DEFB   9CH           ; D
EE8C 9D                        DEFB   9DH           ; R
EE8D A4                        DEFB   A4H           ; S
EE8E 3F                        DEFB   3FH           ; ?
EE8F 20                        DEFB   20H           ;
EE90 0D                        DEFB   0DH
EE91                  MSGLD:   ENT
EE91 4C                        DEFM   'L'           ; 'Loading '
EE92 B7                        DEFB   B7H           ; O
EE93 A1                        DEFB   A1H           ; A
EE94 9C                        DEFB   9CH           ; D
EE95 A6                        DEFB   A6H           ; I
EE96 B0                        DEFB   B0H           ; N
EE97 97                        DEFB   97H           ; G
EE98 20                        DEFB   20H           ;
EE99 0D                        DEFB   0DH
```

```
EE9A                  ;
EE9A                  ;     Pio initialize data
EE9A                  ;
EE9A 00               PIOIDA: DEFB   00H           ; Int vector
EE9B CF                       DEFB   CFH           ; Mode 3 (bit)
EE9C 3F                       DEFB   3FH           ; I/O Reg. set
EE9D 07                       DEFB   07H           ; Int seqence
EE9E                  ;
EE9E 00               PIOIDB: DEFB   00H           ; Int vector
EE9F CF                       DEFB   CFH           ; Mode 3 (bit)
EEA0 00                       DEFB   00H           ; I/O Reg. set
EEA1 07                       DEFB   07H           ; Int seqence
EEA2                  ;
EEA2                  ;
EEA2                  ;     Pallet set data
EEA2                  ;
EEA2 00               PLTDT:  DEFB   00H           ; all black
EEA3 10                       DEFB   10H
EEA4 20                       DEFB   20H
EEA5 30                       DEFB   30H
EEA6 40                       DEFB   40H           ; Sw 0,1=0
EEA7                  ;
EEA7                          SKP    H
```

```
EEA7                 ;
EEA7                 ;
EEA7                 ;
EEA7                 ; Quick disk control command
EEA7                 ;                QL=Quick load
EEA7                 ;                QS=Quick save
EEA7                 ;                QD=Quick directory
EEA7                 ;                QF=Quick formatting
EEA7                 ;                QX=Quick xfer (CMTDisk)
EEA7                 ;                QC=Quick copy (DiskDisk)
EEA7                 ;
EEA7                 ;
11A3  P              BUFF:   EQU    11A3H           ; Keyin buffer
EEA7                 ;
EEA7                 ;==================================
EEA7                 ;
EEA7                 ;      QUICK DISK LOAD COMMAND
EEA7                 ;
EEA7                 ;==================================
EEA7                 ;
EEA7                 QL:     ENT
EEA7 CD27EF                  CALL   QDRCK           ; Ready check
EEAA 3834                    JR     C,QER00
EEAC CD8DEC                  CALL   FNINP           ; Input filename
EEAF CD5FF2                  CALL   HDPCL           ; Head point clear
EEB2                 ;
EEB2                 ;      Disp 'Loading...'
EEB2                 ;
EEB2 1191EE                  LD     DE,MSGLD        ; 'Loading'
EEB5 DF                      RST    3
EEB6                 ;
EEB6                 ;     File search
EEB6                 ;
EEB6 CDF7EE          FILESH: CALL   FILSCH
EEB9 3825                    JR     C,QER00
EEBB                 ;
EEBB                 ;     Atribute check
EEBB                 ;
EEBB 3AF010          QDATRC: LD     A,(ATRB)
EEBE FE01                    CP     OBJCD
EEC0 20F4                    JR     NZ,FILESH
EEC2                 ;
EEC2                 ;     Disp 'Loading ...'
EEC2                 ;
EEC2                 DSFLNA: ENT
EEC2 11F110                  LD     DE,NAME
EEC5 DF                      RST    3
EEC6                 ;
EEC6                 ;     Iocs parameter set
EEC6                 ;
EEC6 210012                  LD     HL,1200H
EEC9 1803                    JR     LPARA1
EECB 2A0611          LPARA0: LD     HL,(QDTADR)
EECE 223211          LPARA1: LD     (QDPC),HL       ; Data adrs set
EED1 2A0411                  LD     HL,(QSIZE)
EED4 223411                  LD     (QDPE),HL
EED7 210301                  LD     HL,0103H        ; Read data block cmd.
EEDA 223011                  LD     (QDPA),HL
EEDD                 ;
EEDD                 ;     Read data block
```

```
EEDD                 ;
EEDD CD10E0                  CALL   QDIOS           ; QD iocs
EEE0 384F            QER00:  JR     C,QER01
EEE2                 ;
EEE2                 ;     Exec load file
EEE2                 ;
EEE2 010003                  LD     BC,0300H        ; Default code (QD)
EEE5 D9                      EXX
EEE6 210411                  LD     HL,QSIZE
EEE9 C3FCEC                  JP     EXF
EEEC                 ;
EEEC                 ;     Iocs flag reset
EEEC                 ;
EEEC                 IOFRS:  ENT
EEEC AF                      XOR    A               ;Acc00H
EEED 324411                  LD     (MTF),A         ; Motor flag
EEF0 323F11                  LD     (FNUPS),A       ; File number flag
EEF3 324111                  LD     (FNUPF),A       ; File number up flag
EEF6 C9                      RET
EEF7                 ;
EEF7                 ;
EEF7                 ;     File search sub.
EEF7                 ;
EEF7                 ;
EEF7                 FILSCH: ENT
EEF7                 ;
EEF7                 ;     Iocs parameter set
EEF7                 ;
EEF7 210300                  LD     HL,0003H        ; Read information cmd.
EEFA 223011                  LD     (QDPA),HL
EEFD 21F010                  LD     HL,10E0H        ; Head adrs
EF00 223211                  LD     (QDPC),HL
EF03 214000                  LD     HL,0040H        ; Read size
EF06 223411                  LD     (QDPE),HL
EF09                 ;
EF09                 ;     Read information block
EF09                 ;
EF09 CD10E0          QLINF:  CALL   QDIOS
EF0C D8                      RET    C
EF0D                 ;
EF0D                 ;     File name check
EF0D                 ;
EF0D 3AA311                  LD     A,(BUFF)
EF10 FE0D                    CP     0DH
EF12 C8                      RET    Z               ;CY=0
EF13 21A311                  LD     HL,BUFF
EF16 11F110                  LD     DE,NAME
EF19 0611                    LD     B,NAMSIZ
EF1B 1A              LDFNCK: LD     A,(DE)
EF1C BE                      CP     (HL)
EF1D 20EA                    JR     NZ,QLINF
EF1F FE0D                    CP     0DH
EF21 C8                      RET    Z               ;CY=0
EF22 13                      INC    DE
EF23 23                      INC    HL
EF24 10F5                    DJNZ   LDFNCK
EF26 C9                      RET
EF27                 ;
EF27                 ;     Quick disk ready check
EF27                 ;
```

```
EF27                QDRCK:  ENT
EF27 AF                     XOR     A
EF28 323111                 LD      (QDPB),A
EF2B C3E6EF                 JP      QDRWC0
EF2E                ;
EF2E                ;=================================
EF2E                ;
EF2E                ;       Quick disk save command
EF2E                ;
EF2E                ;=================================
EF2E                ;
EF2E                QS:     ENT
EF2E CDE1EF                 CALL    QDRWCK          ; Ready & wprt check
EF31 385F           QER01:  JR      C,QER03
EF33                ;
EF33                ;      Input file name
EF33                ;
EF33 CD8DEC                 CALL    FNINP           ; Input file name
EF36 3AA311                 LD      A,(BUFF)
EF39 FE0D                   CP      0DH
EF3B 28F1                   JR      Z,QS
EF3D 21A311                 LD      HL,BUFF
EF40 11F110                 LD      DE,NAME
EF43 011100                 LD      BC,0011H
EF46 EDB0                   LDIR
EF48                ;
EF48                ;      Input top & end & exc adrs
EF48                ;
EF48 CDD3EC                 CALL    TAINP           ; Top adrs
EF4B 220611                 LD      (QDTADR),HL
EF4E CDD8EC                 CALL    EAINP           ; End adrs
EF51 B7                     OR      A               ; Carry reset
EF52 ED4B0611               LD      BC,(QDTADR)
EF56 ED42                   SBC     HL,BC
EF58 23                     INC     HL
EF59 220411                 LD      (QSIZE),HL
EF5C CDDDEC                 CALL    XAINP           ; Exc adrs
EF5F 220811                 LD      (QEXADR),HL
EF62                ;
EF62                ;      Atribute set
EF62                ;
EF62 3E01                   LD      A,OBJCD         ; OBJ Code
EF64 32F010                 LD      (ATRB),A
EF67                ;
EF67                ;     File end search
EF67                ;
EF67 CD9FEF                 CALL    FEDSCH
EF6A 3826                   JR      C,QER03
EF6C                ;
EF6C                ;     Save file start
EF6C                ;
EF6C FE28           SVFLST: CP      NTFECD          ; Not found ?
EF6E C202F2                 JP      NZ,QERTRT
EF71                ;
EF71                ;     Iocs parameter set
EF71                ;
EF71 2A0611         QSIOST: LD      HL,(QDTADR)     ; Data top adrs
EF74 223611         QSVIOS: LD      (QDPG),HL       ; Copy jump point
EF77 210404                 LD      HL,0404H        ; Write Inf+Data blk cmd
EF7A 223011                 LD      (QDPA),HL
```

```
EF7D 21F010                 LD      HL,ATRB         ; Infm top adrs.
EF80 223211                 LD      (QDPC),HL
EF83 214000                 LD      HL,0040H        ; Infm byte size
EF86 223411                 LD      (QDPE),HL
EF89 2A0411                 LD      HL,(QSIZE)      ; Dada byte size
EF8C 223811                 LD      (QDPI),HL
EF8F                ;
EF8F                ;     Write inf & data block
EF8F                ;
EF8F CD10E0                 CALL    QDIOS
EF92 DA28F0         QER03:  JP      C,QER04
EF95                ;
EF95                ;     Quick disk 'OK' message
EF95                ;
EF95                QDOKM:  ENT
EF95 CD0900                 CALL    NL
EF98 1168F2                 LD      DE,MSGQOK       ; 'OK!'
EF9B DF                     RST     3
EF9C C35EEA                 JP      ST1X
EF9F                ;
EF9F                ;     File end search sub.
EF9F                ;
EF9F                FEDSCH: ENT
EF9F CD5FF2                 CALL    HDPCL           ; Hard point clear
EFA2                ;
EFA2                ;     Iocs parameter set
EFA2                ;
EFA2 210300                 LD      HL,0003H        ; Read information cmd.
EFA5 223011                 LD      (QDPA),HL
EFA8 21A311                 LD      HL,BUFF         ; Head adrs
EFAB 223211                 LD      (QDPC),HL
EFAE 214000                 LD      HL,0040H        ; Read size
EFB1 223411                 LD      (QDPE),HL
EFB4                ;
EFB4                ;     Too many files ?
EFB4                ;
EFB4 AF                     XOR     A
EFB5 323C11                 LD      (QDCPC),A
EFB8 3A3C11         EDFLCK: LD      A,(QDCPC)
EFBB 3C                     INC     A
EFBC FE21                   CP      33              ; Files
EFBE 323C11                 LD      (QDCPC),A
EFC1 3E33                   LD      A,TMFECD        ; Too many files err
EFC3 D0                     RET     NC
EFC4                ;
EFC4                ;     End file check
EFC4                ;
EFC4 CD10E0                 CALL    QDIOS
EFC7 3F                     CCF                     ; Comp cy
EFC8 D0                     RET     NC
EFC9                ;
EFC9                ;     Same file name check
EFC9                ;
EFC9 11A411                 LD      DE,BUFF+1
EFCC 21F110                 LD      HL,NAME
EFCF 0611                   LD      B,11H
EFD1 1A             SFNC:   LD      A,(DE)
EFD2 BE                     CP      (HL)
EFD3 20E3                   JR      NZ,EDFLCK
EFD5 FE0D                   CP      0DH
```

```
EFD7 2804                 JR     Z,SFNC0
EFD9 13                   INC    DE
EFDA 23                   INC    HL
EFDB 10F4                 DJNZ   SFNC
EFDD 3E2A       SFNC0:    LD     A,ALEXCD           ; Already exist
EFDF 37                   SCF                       ; cy=1
EFE0 C9                   RET
EFE1            ;
EFE1            ;    Ready & write protect check
EFE1            ;
EFE1 3EFF       QDRWCK:   LD     A,FFH
EFE3 323111               LD     (QDPB),A
EFE6 3E01       QDRWC0:   LD     A,01H
EFE8 323011               LD     (QDPA),A
EFEB CD10E0               CALL   QDIOS
EFEE C9                   RET
EFEF            ;
EFEF            ;=================================
EFEF            ;
EFEF            ;   Quick disk directory command
EFEF            ;
EFEF            ;=================================
EFEF            ;
EFEF            QD:       ENT
EFEF CD27EF               CALL   QDRCK              ; Ready check
EFF2 3834                 JR     C,QER04
EFF4 CD5FF2               CALL   HDPCL              ; Hard point clear
EFF7 0600                 LD     B,00H              ; Counter reset
EFF9            ;
EFF9            ;    Disp 'Directory of QD:'
EFF9            ;
EFF9 CD0900               CALL   NL
EFFC 115CF3               LD     DE,DIRMSG
EFFF DF                   RST    3
F000            ;
F000            ;    Iocs parameter set
F000            ;
F000 2190CD               LD     HL,QDIRBF          ; QD dir buffer
F003 223211     DIRIOP:   LD     (QDPC),HL
F006 210300               LD     HL,0003H           ; Read information cmd.
F009 223011               LD     (QDPA),HL
F00C 214000               LD     HL,0040H           ; Read size
F00F 223411               LD     (QDPE),HL
F012            ;
F012            ;    Read information block
F012            ;
F012 C5                   PUSH   BC
F013 CD10E0               CALL   QDIOS
F016 C1                   POP    BC
F017 380A                 JR     C,DIREFC
F019 04                   INC    B
F01A            ;
F01A            ;    Buffer adrs increment
F01A            ;
F01A 2A3211               LD     HL,(QDPC)
F01D 111200               LD     DE,0012H
F020 19                   ADD    HL,DE
F021 18E0                 JR     DIRIOP
F023            ;
F023            ;    End file check
```

```
F023            ;
F023 FE28       DIREFC:   CP     NTFECD             ; Not found ?
F025 2804                 JR     Z,DIRMTF
F027 37                   SCF
F028 DAC9F0     QER04:    JP     C,QER05
F02B            ;
F02B            ;    Motor off
F02B            ;
F02B 3E06       DIRMTF:   LD     A,06H              ; Motor off cmd.
F02D 323011               LD     (QDPA),A
F030 C5                   PUSH   BC
F031 CD10E0               CALL   QDIOS
F034 C1                   POP    BC
F035            ;
F035            ;    No file check
F035            ;
F035 AF                   XOR    A
F036 B8                   CP     B
F037 3079                 JR     NC,QDOKM0
F039            ;
F039            ;    Directory disp
F039            ;
F039 CD0900               CALL   NL
F03C 2190CD               LD     HL,QDIRBF          ; QD dir buffer
F03F            ;
F03F            ;    Disp atribute
F03F            ;
F03F 7E         DSPATR:   LD     A,(HL)
F040 116CF2               LD     DE,MSGQ01          ; 'OBJ'
F043 3D                   DEC    A
F044 2836                 JR     Z,DRDIS0           ; Object file
F046 1170F2               LD     DE,MSGQ02          ; 'BTX'
F049 3D                   DEC    A
F04A 2830                 JR     Z,DRDIS0           ; Btx file
F04C 1174F2               LD     DE,MSGQ03          ;'BSD'
F04F 3D                   DEC    A
F050 282A                 JR     Z,DRDIS0           ; Bsd file
F052 1178F2               LD     DE,MSGQ04          ;'BRD'
F055 3D                   DEC    A
F056 2824                 JR     Z,DRDIS0           ; Brd file
F058 117CF2               LD     DE,MSGQ05          ;'RB '
F05B 3D                   DEC    A
F05C 281E                 JR     Z,DRDIS0           ; Rb  file
F05E 3D                   DEC    A
F05F 2818                 JR     Z,DRDIS1           ; ??? file
F061 1180F2               LD     DE,MSGQ07          ; 'LIB'
F064 3D                   DEC    A
F065 2815                 JR     Z,DRDIS0           ; Lib file
F067 3D                   DEC    A
F068 280F                 JR     Z,DRDIS1           ; ??? file
F06A 3D                   DEC    A
F06B 280C                 JR     Z,DRDIS1           ; ??? file
F06D 1184F2               LD     DE,MSGQ10          ; 'SYS'
F070 3D                   DEC    A
F071 2809                 JR     Z,DRDIS0           ; Sys file
F073 1188F2               LD     DE,MSGQ11          ; 'GR '
F076 3D                   DEC    A
F077 2803                 JR     Z,DRDIS0           ; GR  file
F079 118CF2     DRDIS1:   LD     DE,MSGQ??          ; '???'
F07C C5         DRDIS0:   PUSH   BC
```

```
F07D 0604                    LD      B,04H
F07F CD0C00         DRDIS2: CALL    PRNTS
F082 10FB                    DJNZ    DRDIS2
F084 C1                      POP     BC
F085 DF                      RST     3
F086 CD0C00                  CALL    PRNTS
F089 CD0C00                  CALL    PRNTS
F08C CD0C00                  CALL    PRNTS
F08F                ;
F08F                ;     Disp file name
F08F                ;
F08F 3E22                    LD      A,'"'
F091 CD1200                  CALL    PRNT
F094 23                      INC     HL
F095 E5                      PUSH    HL
F096 D1                      POP     DE
F097 DF                      RST     3
F098 3E22                    LD      A,'"'
F09A CD1200                  CALL    PRNT
F09D CD0900                  CALL    NL
F0A0                ;
F0A0                ;    Counter decrement
F0A0                ;
F0A0 111100                  LD      DE,0011H
F0A3 19                      ADD     HL,DE
F0A4 CDCA08                  CALL    ?KEY
F0A7 B7                      OR      A
F0A8 28FA                    JR      Z,-4
F0AA CD320A                  CALL    ?BRK
F0AD CA5EEA                  JP      Z,ST1X
F0B0 108D                    DJNZ    DSPATR           ; B=File counter
F0B2                ;
F0B2                ;    Directory end
F0B2                ;
F0B2 C395EF         QDOKM0: JP      QDOKM
F0B5                ;
F0B5                ;==============================
F0B5                ;
F0B5                ;   Quick disk format command
F0B5                ;
F0B5                ;==============================
F0B5                ;
F0B5                QF:      ENT
F0B5                ;
F0B5                ;
F0B5                ;     Disp formatting
F0B5                ;
F0B5 1130F3                  LD      DE,QDFMG         ; 'QD:FORMATTING'
F0B8 DF                      RST     3
F0B9 CDCEF0                  CALL    WAITY            ; 'OK?(Y/N)'
F0BC                ;
F0BC                ;     Ready & wrpt check
F0BC                ;
F0BC CDE1EF                  CALL    QDRWCK
F0BF 3808                    JR      C,QER05
F0C1                ;
F0C1                ;     Format
F0C1                ;
F0C1 3E02                    LD      A,02H            ; Init cmd set
F0C3 323011                  LD      (QDPA),A
```

```
F0C6 CD10E0                  CALL    QDIOS
F0C9 DA60F1         QER05:  JP      C,QER06
F0CC                ;
F0CC                ;     Format end
F0CC                ;
F0CC 18E4                    JR      QDOKM0
F0CE                ;
F0CE                ;     Wait key in 'Y'
F0CE                ;
F0CE CD0900         WAITY:  CALL    NL
F0D1 1153F3                  LD      DE,QDQCM0        ; 'OK?(Y/N)'
F0D4 DF                      RST     3
F0D5 CD91F1                  CALL    NKYWT            ; Wait non key
F0D8 CDB10F         WATKY:  CALL    0FB1H            ; Call ?PONT

F0DB 3EEF                    LD      A,EFH            ;'
F0DD 77             WATK0:  LD      (HL),A
F0DE CDEBF0                  CALL    WATIM
F0E1 3805                    JR      C,WATK1          ; If 'Y' then CY=1
F0E3 7E                      LD      A,(HL)
F0E4 EEEF                    XOR     EFH
F0E6 18F5                    JR      WATK0
F0E8 AF             WATK1:  XOR     A
F0E9 77                      LD      (HL),A
F0EA C9                      RET
F0EB                ;
F0EB                ;     Wait timer
F0EB                ;
F0EB 010600         WATIM:  LD      BC,0006H
F0EE 0B             WATI0:  DEC     BC
F0EF 78                      LD      A,B
F0F0 B1                      OR      C
F0F1 C8                      RET     Z
F0F2                ;
F0F2                ;     Key search
F0F2                ;
F0F2 3E01                    LD      A,01H
F0F4 CD15F1                  CALL    SRCK0
F0F7 FE7F                    CP      7FH              ; 'Y' ?
F0F9 37                      SCF
F0FA C8                      RET     Z
F0FB 3E03                    LD      A,03H
F0FD CD15F1                  CALL    SRCK0
F100 FEFB                    CP      FBH              ; 'N' ?
F102 2809                    JR      Z,WATI1
F104 3E08                    LD      A,08H
F106 CD15F1                  CALL    SRCK0
F109 FE7E                    CP      7EH              ; 'Break+Shift' ?
F10B 20E1                    JR      NZ,WATI0
F10D                ;
F10D                ;     Jump ST1X
F10D                ;
F10D AF             WATI1:  XOR     A
F10E 77                      LD      (HL),A
F10F 31F010                  LD      SP,10F0H
F112 C35EEA                  JP      ST1X
F115                ;
F115                ;     Key scan
F115                ;
F115                SRCK0:  ENT
F115 E5                      PUSH    HL
```

```
F116 2100E0              LD      HL,E000H
F119 77                  LD      (HL),A
F11A 23                  INC     HL
F11B 7E          SRCK2:  LD      A,(HL)
F11C F5                  PUSH    AF
F11D C5                  PUSH    BC
F11E 0614                LD      B,14H           ; 1ms*20=20ms
F120 CDB9E6      SRCK1:  CALL    DLY1M           ; 1ms Delay
F123 10FB                DJNZ    SRCK1
F125 C1                  POP     BC
F126 F1                  POP     AF
F127 BE                  CP      (HL)
F128 20F1                JR      NZ,SRCK2
F12A E1                  POP     HL
F12B C9                  RET
F12C             ;
F12C             ;============================
F12C             ;
F12C             ;    Quick disk copy command
F12C             ;
F12C             ;============================
F12C             ;
F12C             QC:     ENT
F12C CD27EF              CALL    QDRCK           ; Ready check
F12F 382F                JR      C,QER06
F131 CD8DEC              CALL    FNINP           ; Input file name
F134 3AA311              LD      A,(BUFF)
F137 FE0D                CP      0DH
F139 28F1                JR      Z,QC
F13B CD5FF2              CALL    HDPCL           ; Hard point clear
F13E 1191EE              LD      DE,MSGLD        ; 'Loading'
F141 DF                  RST     3
F142             ;
F142             ;     File search
F142             ;
F142 CDF7EE              CALL    FILSCH
F145 3819                JR      C,QER06
F147             ;
F147             ;     Disp 'Loading ...'
F147             ;
F147 11F110              LD      DE,NAME
F14A DF                  RST     3
F14B             ;
F14B             ;    Iocs parameter set
F14B             ;
F14B 210012              LD      HL,1200H
F14E 223211              LD      (QDPC),HL
F151 2A0411              LD      HL,(QSIZE)
F154 223411              LD      (QDPE),HL
F157 210301              LD      HL,0103H        ; Read data block cmd.
F15A 223011              LD      (QDPA),HL
F15D             ;
F15D             ;    Read data block
F15D             ;
F15D CD10E0              CALL    QDIOS
F160 3821        QER06:  JR      C,QER07
F162             ;
F162             ;    Bell
F162             ;
F162 CD3E00              CALL    BELL
```

```
F165             ;
F165             ;    Disp 'Set Disk ...'
F165             ;
F165 CD0900              CALL    NL
F168 113EF3              LD      DE,QDQCM        ; 'Set distination disk'
F16B DF                  RST     3
F16C             ;
F16C             ;    Error return set
F16C             ;
F16C 3E02        QCSPST: LD      A,02H
F16E 323A11              LD      (QDCPA),A
F171 216CF1              LD      HL,QCSPST
F174 31EE10              LD      SP,10EEH
F177 E3                  EX      (SP),HL
F178             ;
F178             ;    Wait 'Y'
F178             ;
F178 CDCEF0              CALL    WAITY           ; 'OK?(Y/N)'
F17B             ;
F17B             ;    Ready & wprt check
F17B             ;
F17B CDE1EF      QCRWCK: CALL    QDRWCK
F17E 3803                JR      C,QER07
F180             ;
F180             ;    File end search
F180             ;
F180 CD9FEF              CALL    FEDSCH          ; File end search
F183 DA02F2      QER07:  JP      C,QERTRT        ; Already exist err
F186 FE28                CP      40              ; Not found ?
F188 C202F2              JP      NZ,QERTRT
F18B             ;
F18B             ;    Save file start
F18B             ;
F18B 210012              LD      HL,1200H        ; Data adrs
F18E C374EF              JP      QSVIOS          ; Jump save routin
F191             ;
F191             ;    Non key check
F191             ;
F191             NKYWT:  ENT
F191 060A                LD      B,0AH
F193 2100E0      NKYWT0: LD      HL,E000H
F196 05                  DEC     B
F197 70                  LD      (HL),B
F198 04                  INC     B
F199 23                  INC     HL
F19A 7E                  LD      A,(HL)
F19B FEFF                CP      FFH
F19D 20F2                JR      NZ,NKYWT
F19F 10F2                DJNZ    NKYWT0
F1A1 C9                  RET
F1A2             ;
F1A2             ;============================
F1A2             ;
F1A2             ;   Quick disk xfer command
F1A2             ;
F1A2             ;============================
F1A2             ;
F1A2             QX:     ENT
F1A2             ;
F1A2             ;    Loading from cmt.
```

```
F1A2                 ;
F1A2 CD54EB                 CALL   LD00          ; Read inf & data
F1A5 DA24EB                 JP     C,?ERX
F1A8 D9                     EXX
F1A9 220411                 LD     (DTADR),HL
F1AC                 ;
F1AC                 ;    Bell
F1AC                 ;
F1AC CD3E00                 CALL   BELL
F1AF                 ;
F1AF                 ;    Input file name
F1AF                 ;
F1AF CD8DEC          QXFN0: CALL   FNINP         ; Input file name
F1B2 3AA311                 LD     A,(BUFF)
F1B5 FE0D                   CP     0DH
F1B7 28F6                   JR     Z,QXFN0
F1B9 21A311                 LD     HL,BUFF
F1BC 11F110                 LD     DE,NAME
F1BF 011100                 LD     BC,0011H
F1C2 EDB0                   LDIR
F1C4                 ;
F1C4                 ;    Disp 'Set Disk ...'
F1C4                 ;
F1C4 CD0900                 CALL   NL
F1C7 113EF3                 LD     DE,QDQCM      ; 'Set distination disk'
F1CA DF                     RST    3
F1CB                 ;
F1CB                 ;    Atrb change (700800)
F1CB                 ;
F1CB 3AF010                 LD     A,(ATRB)
F1CE FE04                   CP     04H           ; 700 BSD file ?
F1D0 2806                   JR     Z,ATRC0
F1D2 FE05                   CP     05H           ; 700 BTX file ?
F1D4 2006                   JR     NZ,QXPRST
F1D6 3D                     DEC    A
F1D7 3D                     DEC    A
F1D8 3D              ATRC0: DEC    A
F1D9 32F010                 LD     (ATRB),A      ; Atrb set
F1DC                 ;
F1DC                 ;    Parameter trans
F1DC                 ;
F1DC 212E11          QXPRST: LD    HL,112EH
F1DF 113011                 LD     DE,1130H
F1E2 012D00                 LD     BC,002DH
F1E5 EDB8                   LDDR
F1E7 210000                 LD     HL,0000H      ; Lock & Secret
F1EA 220211                 LD     (1102H),HL
F1ED                 ;
F1ED                 ;    Error return set
F1ED                 ;
F1ED 3E02            QXSPST: LD    A,02H
F1EF 323A11                 LD     (QDCPA),A
F1F2 21EDF1                 LD     HL,QXSPST
F1F5 31EE10                 LD     SP,10EEH
F1F8 E3                     EX     (SP),HL
F1F9                 ;
F1F9                 ;    Wait 'Y'
F1F9                 ;
F1F9 CDCEF0                 CALL   WAITY         ; 'OK?(Y/N)'
F1FC                 ;
```

```
F1FC                 ;     Iocs flag reset
F1FC                 ;
F1FC CDECEE                 CALL   IOFRS
F1FF                 ;
F1FF                 ;    Ready & wprt check
F1FF                 ;
F1FF C37BF1                 JP     QCRWCK
F202                 ;
F202                 ;=====================
F202                 ;
F202                 ;    Error treatment
F202                 ;
F202                 ;=====================
F202                 ;
F202                 QERTRT: ENT
F202 1190F2                 LD     DE,MGNFE      ; 'Not Found err'
F205 FE28                   CP     NTFECD        ; Not found err
F207 283B                   JR     Z,QERMF
F209 1119F3          QERT0: LD     DE,MGBDE      ; 'Bad disk error'
F20C FE39                   CP     BDSKCD        ; Bad disk err
F20E 2834                   JR     Z,QERMF
F210 11D6F2          QER1:  LD     DE,MGWPT      ; 'Write protect'
F213 FE2E                   CP     WPRTCD        ; Write protect
F215 282D                   JR     Z,QERMF
F217 11E7F2          QER2:  LD     DE,MGNRE      ; 'Not ready'
F21A FE32                   CP     QNTRCD        ; Not ready
F21C 2826                   JR     Z,QERMF
F21E 11F4F2          QER3:  LD     DE,MGNSE      ; 'No file space error'
F221 FE35                   CP     NFSECD        ; No file space err
F223 281F                   JR     Z,QERMF
F225 1109F3          QER4:  LD     DE,MGUFE      ; 'Unformat'
F228 FE36                   CP     UNFMCD        ; Unformat err
F22A 2818                   JR     Z,QERMF
F22C 11C4F2          QER5:  LD     DE,MGALE      ; 'Already exist'
F22F FE2A                   CP     ALEXCD        ; Already exist
F231 2811                   JR     Z,QERMF
F233 11A2F2          QER6:  LD     DE,MGTME      ; 'Too many files'
F236 FE33                   CP     TMFECD        ; Too many files err
F238 280A                   JR     Z,QERMF
F23A 1129F3          QER7:  LD     DE,MGBRK      ; 'Break!'
F23D FE00                   CP     BRKCD         ; Break
F23F 2803                   JR     Z,QERMF
F241 11B8F2          QER8:  LD     DE,MGHDE      ; 'Haard error'
F244                 ;
F244                 ;    Motor off
F244                 ;
F244 3E06            QERMF: LD     A,06H         ; Motor off cmd.
F246 323011                 LD     (QDPA),A
F249 CD10E0                 CALL   QDIOS
F24C CD5FF2                 CALL   HDPCL
F24F                 ;
F24F 3A3A11                 LD     A,(QDCPA)
F252 1F                     RRA
F253 D8                     RET    C             ; Boot err
F254 F5                     PUSH   AF
F255 CD0900                 CALL   NL
F258 DF                     RST    3
F259 F1                     POP    AF
F25A 1F                     RRA
F25B D8                     RET    C             ; CMD QC,QX err
```

```
F25C C35EEA             JP      ST1X
F25F           ;
F25F           ;    Header point clear
F25F           ;
F25F           HDPCL:   ENT
F25F 3E05               LD      A,05H          ; Head point clear cmd.
F261 323011             LD      (QDPA),A
F264 CD10E0             CALL    QDIOS
F267 C9                 RET
F268           ;
F268           ;==================
F268           ;
F268           ;   Message table
F268           ;
F268           ;==================
F268           ;
F268 4F4B21    MSGQOK:  DEFM    'OK!'
F26B 0D                 DEFB    0DH
F26C 4F424A    MSGQ01:  DEFM    'OBJ'
F26F 0D                 DEFB    0DH
F270 425458    MSGQ02:  DEFM    'BTX'
F273 0D                 DEFB    0DH
F274 425344    MSGQ03:  DEFM    'BSD'
F277 0D                 DEFB    0DH
F278 425244    MSGQ04:  DEFM    'BRD'
F27B 0D                 DEFB    0DH
F27C 524220    MSGQ05:  DEFM    'RB '
F27F 0D                 DEFB    0DH
F280 4C4942    MSGQ07:  DEFM    'LIB'
F283 0D                 DEFB    0DH
F284 535953    MSGQ10:  DEFM    'SYS'
F287 0D                 DEFB    0DH
F288 475220    MSGQ11:  DEFM    'GR '
F28B 0D                 DEFB    0DH
F28C 3F3F3F    MSGQ??:  DEFM    '???'
F28F 0D                 DEFB    0DH
F290 51443A46  MGNFE:   DEFM    'QD:F'         ; 'QD:File not found'
F294 A6                 DEFB    A6H            ; I
F295 B8                 DEFB    B8H            ; L
F296 92                 DEFB    92H            ; E
F297 20                 DEFB    20H            ;
F298 B0                 DEFB    B0H            ; N
F299 B7                 DEFB    B7H            ; O
F29A 96                 DEFB    96H            ; T
F29B 20                 DEFB    20H            ;
F29C AA                 DEFB    AAH            ; F
F29D B7                 DEFB    B7H            ; O
F29E A5                 DEFB    A5H            ; U
F29F B0                 DEFB    B0H            ; N
F2A0 9C                 DEFB    9CH            ; D
F2A1 0D                 DEFB    0DH
F2A2           ;
F2A2 51443A54  MGTME:   DEFM    'QD:T'         ; 'Too many files err'
F2A6 B7                 DEFB    B7H            ; O
F2A7 B7                 DEFB    B7H            ; O
F2A8 20                 DEFB    20H            ;
F2A9 B3                 DEFB    B3H            ; M
F2AA A1                 DEFB    A1H            ; A
F2AB B0                 DEFB    B0H            ; N
F2AC BD                 DEFB    BDH            ; Y
```

```
F2AD 20                 DEFB    20H            ;
F2AE AA                 DEFB    AAH            ; F
F2AF A6                 DEFB    A6H            ; I
F2B0 B8                 DEFB    B8H            ; L
F2B1 92                 DEFB    92H            ; E
F2B2 A4                 DEFB    A4H            ; S
F2B3 20                 DEFB    20H            ;
F2B4 92                 DEFB    92H            ; E
F2B5 9D                 DEFB    9DH            ; R
F2B6 9D                 DEFB    9DH            ; R
F2B7 0D                 DEFB    0DH
F2B8           ;
F2B8 51443A48  MGHDE:   DEFM    'QD:H'         ; 'QD:Hard err'
F2BC A1                 DEFB    A1H            ; A
F2BD 9D                 DEFB    9DH            ; R
F2BE 9C                 DEFB    9CH            ; D
F2BF 20                 DEFB    20H            ;
F2C0 92                 DEFB    92H            ; E
F2C1 9D                 DEFB    9DH            ; R
F2C2 9D                 DEFB    9DH            ; R
F2C3 0D                 DEFB    0DH
F2C4 41        MGALE:   DEFM    'A'            ; 'Already exxist err'
F2C5 B8                 DEFB    B8H            ; L
F2C6 9D                 DEFB    9DH            ; R
F2C7 92                 DEFB    92H            ; E
F2C8 A1                 DEFB    A1H            ; A
F2C9 9C                 DEFB    9CH            ; D
F2CA BD                 DEFB    BDH            ; Y
F2CB 20                 DEFB    20H            ;
F2CC 92                 DEFB    92H            ; E
F2CD 9B                 DEFB    9BH            ; X
F2CE A6                 DEFB    A6H            ; I
F2CF A4                 DEFB    A4H            ; S
F2D0 96                 DEFB    96H            ; T
F2D1 20                 DEFB    20H            ;
F2D2 92                 DEFB    92H            ; E
F2D3 9D                 DEFB    9DH            ; R
F2D4 9D                 DEFB    9DH            ; R
F2D5 0D                 DEFB    0DH
F2D6           ;
F2D6 51443A57  MGWPT:   DEFM    'QD:W'         ; 'QD:Write protect'
F2DA 9D                 DEFB    9DH            ; R
F2DB A6                 DEFB    A6H            ; I
F2DC 96                 DEFB    96H            ; T
F2DD 92                 DEFB    92H            ; E
F2DE 20                 DEFB    20H            ;
F2DF 9E                 DEFB    9EH            ; P
F2E0 9D                 DEFB    9DH            ; R
F2E1 B7                 DEFB    B7H            ; O
F2E2 96                 DEFB    96H            ; T
F2E3 92                 DEFB    92H            ; E
F2E4 9F                 DEFB    9FH            ; C
F2E5 96                 DEFB    96H            ; T
F2E6 0D                 DEFB    0DH
F2E7           ;
F2E7 51443A4E  MGNRE:   DEFM    'QD:N'         ; 'QD:Not ready'
F2EB B7                 DEFB    B7H            ; O
F2EC 96                 DEFB    96H            ; T
F2ED 20                 DEFB    20H            ;
F2EE 9D                 DEFB    9DH            ; R
```

```
F2EF 92                    DEFB    92H            ; E
F2F0 A1                    DEFB    A1H            ; A
F2F1 9C                    DEFB    9CH            ; D
F2F2 BD                    DEFB    BDH            ; Y
F2F3 0D                    DEFB    0DH
F2F4                ;
F2F4 51443A4E       MGNSE: DEFM    'QD:N'         ; 'QD:No file space err'
F2F8 B7                    DEFB    B7H            ; O
F2F9 20                    DEFB    20H            ;
F2FA AA                    DEFB    AAH            ; F
F2FB A6                    DEFB    A6H            ; I
F2FC B8                    DEFB    B8H            ; L
F2FD 92                    DEFB    92H            ; E
F2FE 20                    DEFB    20H            ;
F2FF A4                    DEFB    A4H            ; S
F300 9E                    DEFB    9EH            ; P
F301 A1                    DEFB    A1H            ; A
F302 9F                    DEFB    9FH            ; C
F303 92                    DEFB    92H            ; E
F304 20                    DEFB    20H            ;
F305 92                    DEFB    92H            ; E
F306 9D                    DEFB    9DH            ; R
F307 9D                    DEFB    9DH            ; R
F308 0D                    DEFB    0DH
F309                ;
F309 51443A55       MGUFE: DEFM    'QD:U'         ; 'QD:Unformat err'
F30D B0                    DEFB    B0H            ; N
F30E AA                    DEFB    AAH            ; F
F30F B7                    DEFB    B7H            ; O
F310 9D                    DEFB    9DH            ; R
F311 B3                    DEFB    B3H            ; M
F312 A1                    DEFB    A1H            ; A
F313 96                    DEFB    96H            ; T
F314 20                    DEFB    20H            ;
F315 92                    DEFB    92H            ; E
F316 9D                    DEFB    9DH            ; R
F317 9D                    DEFB    9DH            ; R
F318 0D                    DEFB    0DH
F319                ;
F319 51443A42       MGBDE: DEFM    'QD:B'         ; 'QD:Bad disk err'
F31D A1                    DEFB    A1H            ; A
F31E 9C                    DEFB    9CH            ; D
F31F 20                    DEFB    20H            ;
F320 9C                    DEFB    9CH            ; D
F321 A6                    DEFB    A6H            ; I
F322 A4                    DEFB    A4H            ; S
F323 A9                    DEFB    A9H            ; K
F324 20                    DEFB    20H            ;
F325 92                    DEFB    92H            ; E
F326 9D                    DEFB    9DH            ; R
F327 9D                    DEFB    9DH            ; R
F328 0D                    DEFB    0DH
F329                ;
F329                MGBRK: ENT
F329 42                    DEFM    'B'            ; 'Break!'
F32A 9D                    DEFB    9DH            ; R
F32B 92                    DEFB    92H            ; E
F32C A1                    DEFB    A1H            ; A
F32D A9                    DEFB    A9H            ; K
F32E 21                    DEFB    21H            ; !
```

```
F32F 0D                    DEFB    0DH
F330 51443A46       QDFMG: DEFM    'QD:F'         ; 'QD:Formatting'
F334 B7                    DEFB    B7H            ; O
F335 9D                    DEFB    9DH            ; R
F336 B3                    DEFB    B3H            ; M
F337 A1                    DEFB    A1H            ; A
F338 96                    DEFB    96H            ; T
F339 96                    DEFB    96H            ; T
F33A A6                    DEFB    A6H            ; I
F33B B0                    DEFB    B0H            ; N
F33C 97                    DEFB    97H            ; G
F33D 0D                    DEFB    0DH
F33E                ;
F33E 53             QDQCM: DEFM    'S'            ; 'Set distination disk'
F33F 92                    DEFB    92H            ; E
F340 96                    DEFB    96H            ; T
F341 20                    DEFB    20H            ;
F342 9C                    DEFB    9CH            ; D
F343 92                    DEFB    92H            ; E
F344 A4                    DEFB    A4H            ; S
F345 96                    DEFB    96H            ; T
F346 A6                    DEFB    A6H            ; I
F347 B0                    DEFB    B0H            ; N
F348 A1                    DEFB    A1H            ; A
F349 96                    DEFB    96H            ; T
F34A A6                    DEFB    A6H            ; I
F34B B7                    DEFB    B7H            ; O
F34C B0                    DEFB    B0H            ; N
F34D 20                    DEFB    20H            ;
F34E 9C                    DEFB    9CH            ; D
F34F A6                    DEFB    A6H            ; I
F350 A4                    DEFB    A4H            ; S
F351 A9                    DEFB    A9H            ; K
F352 0D                    DEFB    0DH
F353                ;
F353 4F4B3F28       QDQCM0: DEFM   'OK?(Y/N)'
F357 592F4E29
F35B 0D                    DEFB    0DH
F35C 44             DIRMSG: DEFM   'D'            ; 'Directory of QD:'
F35D A6                    DEFB    A6H            ; I
F35E 9D                    DEFB    9DH            ; R
F35F 92                    DEFB    92H            ; E
F360 9F                    DEFB    9FH            ; C
F361 96                    DEFB    96H            ; T
F362 B7                    DEFB    B7H            ; O
F363 9D                    DEFB    9DH            ; R
F364 BD                    DEFB    BDH            ; Y
F365 20                    DEFB    20H            ;
F366 B7                    DEFB    B7H            ; O
F367 AA                    DEFB    AAH            ; F
F368 20                    DEFB    20H            ;
F369 51                    DEFB    51H            ; Q
F36A 44                    DEFB    44H            ; D
F36B 3A                    DEFB    3AH            ; :
F36C 0D                    DEFB    0DH
F36D                ;
F36D                       SKP     H
```

```
F36D                   ;
F36D                   ;   Quick disk Iocs (Syncs)
F36D                   ;        Patch
F36D                   ;      1984.10.11
F36D                   ;
F380                          ORG     F380H
F380                   EOMRS: ENT
F380 3EC0                     LD      A,C0H            ; Eom Reset
F382 D3F6                     OUT     (SIOAC),A
F384 C9                       RET
F385                   ;
F385                          END
```

```
#CLR8  09D5  .4DE   02A6  12ERMG EE3A  12NSET E7DF  1R12R0 E6DA
1R12RD E6CA  1R12WR E745  2HEX   041F  70MOD  ED08  ?ADCN  0BB9
?BEL   0577  ?BLNK  0DA6  ?BRK   0A32  ?CLER  0FD8  ?DPCT  0DDC
?ERX   EB24  ?ERX0  EB2E  ?KEY   08CA  ?MODE  073E  ?PRTS  0920
?RDD   04F8  ?RDI   04D8  ?TMST  0308  ?VRFY  0588  ?VRFY0 0593
?WRD   0475  ?WRI   0436  ADJ    E63C  AINP0  ECE0  ALEXCD 002A
ASC    03DA  ATRB   10F0  ATRC0  F1D8  BDR    E0FF  BDSKCD 0039
BDV    E21B  BDV1   E22D  BDW    E1ED  BDW1   E204  BEGIN  E813
BEGIN0 E800  BELL   003E  BFRF   1146  BGETL  012F  BGETLX EB34
BGI0   E876  BGIN00 E804  BLKTR  ED20  BOOT   E4D1  BPARA  CEE9
BPRO   CF00  BREAD  E5A7  BRKC   E435  BRKCD  0000  BRKCK  001E
BRKEY  001E  BSDCD  0003  BSY0   E5A1  BSYOF0 E58F  BSYOF1 E59C
BSYOF2 E58C  BSYOF3 E597  BSYOFE E59F  BSYOFF E587  BSYON  E568
BSYON0 E570  BSYON1 E57D  BSYON2 E56D  BSYON3 E578  BSYONE E580
BTXCD  0002  BUF    11A3  BUFER  11A3  BUFF   11A3  CAL0   EAAD
CHECK  E729  CK1    E730  CK2    E735  CLBF0  CEF6  CLBF1  CEF7
CLBF2  CEF8  CLBF3  CEF9  CLLET  EA43  CMD    CEF4  CMDOT1 E555
CMDOT2 E64E  CMY0   005B  CMY00  E85F  CNVRT  E696  COMNT  1108
CR     00D8  CSS    E945  DIREFC F023  DIRIOP F003  DIRMSG F35C
DIRMTF F02B  DISCLR EA59  DLY1M  E6B9  DLY60M E6BF  DLY80U E6B3
DLYT   E6C3  DM     00DC  DMR    E132  DMRI   E13A  DR     00DB
DRDIS0 F07C  DRDIS1 F079  DRDIS2 F07F  DSFLNA EEC2  DSILF  E972
DSPATR F03F  DSPXY  1171  DTADR  1104  DUM1X  EC87  DUM2X  EC3E
DUM3X  EC37  DUMPX  EC29  EAINP  ECD8  EDFLCK EFB8  EOM    E3B2
EOMRS  F380  ERCK1  E683  ERJMP  E690  ERRET  E68E  ERRET1 E68A
ERRM0  EDE5  ERRM1  EDF6  ERRM2  EE04  ERRM3  EE16  ERRM4  EE27
ERRMG  EA34  ERROR  E661  ERRTR0 E4BF  ERRTR1 E4C2  ERRTRT E4B6
EXADR  1106  EXCMT  E99D  EXF    ECFC  EXF0   ED40  EXRAM  E6D0
FDB    E8CC  FDCC&  E8D5  FDCK   EB00  FDMTON E517  FDX    E44A
FEDSCH EF9F  FILESH EEB6  FILSCH EEF7  FMERR  E0D6  FMS    EFFF
FNA    1142  FNB    1143  FNEND  E2CE  FNINP  EC8D  FNINP0 ECA4
FNINP1 ECB1  FNINP2 ECB9  FNINP3 ECBD  FNINP4 ECC5  FNINP5 ECCF
FNINP6 ECD1  FNINPR EC9F  FNUPF  1141  FNUPS  113F  FNUPS1 1140
GDGMD  00CE  GDGRF  00CD  GDGWF  00CC  GETKY  001B  GETL   0003
GOT0   EAAC  GOTO   00F3  GOTOX  EAA9  HDERCD 0029  HDPCL  F25F
HDPT   113D  HDPT0  113E  HEXIYX EB42  HLHEX  0410  HPNFE  E14A
HPS    E114  HPS1   E147  HS     00DD  INTER  E658  IOE41  E400
IOE46  E403  IOE50  E406  IOE53  E409  IOE54  E40C  IOFRS  EEEC
IPLM0  ED88  IPLM1  ED98  IPLM2  EDA7  IPLM3  EDB5  IPLM4  EDC3
IPLMC  E4CA  JST1X  EAE2  KEYCK  E8B8  KEYPA  E000  KEYPB  E001
KSCAN0 E90F  KSCAN1 E91E  KSCAN2 E932  KSJPQB E91B  LCM12  E807
LD00   EB54  LDCMT  EB70  LDDEC  ED2E  LDFNCK EF1B  LDINC  ED3E
LET    EA48  LETNL  0006  LLPT   0470  LOA0X  EB58  LOADX  EB4C
LPARA0 EECB  LPARA1 EECE  LREDY  E2A3  LSINT  E2D8  MASTE  E4AE
MCHECK E471  MCORX  EB7B  MCR1X  EB7E  MCR2X  EBA7  MCR3X  EBAA
MGALE  F2C4  MGBDE  F319  MGBRK  F329  MGHDE  F2B8  MGNFE  F290
MGNRE  F2E7  MGNSE  F2F4  MGTME  F2A2  MGUFE  F309  MGWPT  F2D6
MLDSP  02BE  MOFF   E530  MON    E93B  MONMSG EE4D  MSG    0015
MSG?3  06E7  MSGE1  0147  MSGEA  EE7B  MSGLD  EE91  MSGLET EA4E
MSGOK  0942  MSGPS  EA50  MSGQ01 F26C  MSGQ02 F270  MSGQ03 F274
MSGQ04 F278  MSGQ05 F27C  MSGQ07 F280  MSGQ10 F284  MSGQ11 F288
MSGQ?? F28C  MSGQOK F268  MSGSV0 EE65  MSGTA  EE70  MSGW01 E7E2
MSGW02 E7F0  MSGXA  EE86  MTD1   E51D  MTF    1144  MTFG   CEF5
MTOF   E2E8  MTOFX  E0D3  MTON   E29B  NAME   10F1  NAMSIZ 0011
NFNUP  E1E8  NFSECD 0035  NKYWT  F191  NKYWT0 F193  NL     0009
NLPHL  05FA  NOTIO  E4BA  NRCK   E3E5  NTFECD 0028  OBJCD  0001
PARST1 E5FB  PARST2 E62B  PIOA   E835  PIOIDA EE9A  PIOIDB EE9E
PLTDT  EEA2  PLTST  E8E1  PRNT   0012  PRNT3  096C  PRNTS  000C
PRTHL  03BA  PSG0   E862  PSGSET E866  QBFLSH E9E0  QBT    E9B7
QC     F12C  QCRWCK F17B  QCSPST F16C  QD     EFEF  QDATRC EEBB
```

```
QDCPA  113A  QDCPB  113B  QDCPC  113C  QDFM   E090  QDFM1  E09C
QDFM2  E0A9  QDFM3  E0C0  QDFM4  E0D0  QDFMER E9FA  QDFMG  F330
QDHPC  E083  QDIO   E810  QDIOS  E010  QDIRBF CD90  QDLDER EA04
QDNTR  E9F7  QDOFF  E2E8  QDOKM  EF95  QDOKM0 F0B2  QDPA   1130
QDPB   1131  QDPC   1132  QDPE   1134  QDPG   1136  QDPI   1138
QDQCM  F33E  QDQCM0 F353  QDRC   E08A  QDRCK  EF27  QDRD   E0DA
QDRWC0 EFE6  QDRWCK EFE1  QDSVFN E268  QDTADR 1106  QDTBL  1130
QDWE1  E1E5  QDWR   E14E  QDWR1  E171  QDWR2  E185  QDWR3  E1CE
QDWR4  E1AC  QDWR5  E1E2  QER00  EEE0  QER01  EF31  QER03  EF92
QER04  F028  QER05  F0C9  QER06  F160  QER07  F183  QER1   F210
QER2   F217  QER3   F21E  QER4   F225  QER5   F22C  QER6   F233
QER7   F23A  QER8   F241  QERMF  F244  QERT0  F209  QERTRT F202
QEXADR 1108  QF     F0B5  QL     EEA7  QLINF  EF09  QMEIN  E06A
QNTRCD 0032  QREDY  E23C  QS     EF2E  QSIOST EF71  QSIZE  1104
QSVIOS EF74  QUICK  EAB5  QX     F1A2  QXFN0  F1AF  QXPRST F1DC
QXSPST F1ED  R12C&B E8F2  R12CK1 E7BA  RAMBRD EAE5  RCLB   E548
RDATA  E3F0  RDB0   E769  RDCR1  E3C5  RDCR2  E3CA  RDCRC  E3C3
RDDAT  002A  RDDTBL EB6D  RDINF  0027  RDTB0  E997  RE3    E5E0
RE4    E5C1  RE6    E5BF  RE8    E5AD  READ1  E5AA  READY  E4DC
REDY0  E4F4  REDY1  E512  REDY2  E511  REND   E5F6  RETSP  1148
RMBD1  EAF3  RSET   E211  RTY    E015  RTY2   E062  RTY3   E045
RTY4   E068  RTY5   E04D  RTYF   1145  SAC00  EBF3  SAC01  EBF8
SACMT  EBE8  SAVEX  EBAE  SAVJE  EBDF  SAVJP  EBE5  SBRK   E39A
SCM12  E80A  SCR    00DA  SEEK   E528  SELM0  ED43  SELM00 EA1D
SELM01 EA28  SELM1  ED53  SELM2  ED61  SELM3  ED6E  SELM4  ED7E
SELMG0 E909  SELMSG EA06  SFNC   EFD1  SFNC0  EFDD  SGJP   EC27
SGX    EC1E  SIDST  E61B  SIOAC  00F6  SIOAD  00F4  SIOBC  00F7
SIOBD  00F5  SIOCK  EB13  SIOLD  E421  SIOSD  E42C  SIZE   1102
SJP    E462  SLMSG  E906  SLPT   03D5  SORES  E86C  SP     10F0
SPHEX  03B1  SRCK0  F115  SRCK1  F120  SRCK2  F11B  SREDY  E271
ST1X   EA5E  ST2X   EA6F  STAFG  CEFB  SUM    E70E  SUMCK1 E715
SUMCK2 E727  SUMCK3 E71C  SVFLST EF6C  SWRK   119D  SYNCA  E3A3
SYNCF  1147  SYNCL1 E313  SYNCL2 E2FD  SYNCS1 E37E  SYNCS2 E365
SYNCW0 E32C  SYNCW1 E33D  SYNCW2 E347  SYNCW3 E357  SYNW01 E354
TAINP  ECD3  TEMPW  119E  TIMIN  038D  TIMW   E414  TIMW1  E415
TIMW2  E417  TMFECD 0033  TMLPL  E316  TMLPS  E381  TPERJP E9B4
TPERX  E9AA  TPERX0 E9AC  TR     00D9  TRA1   E3DC  TRANS  E3DB
TRANS0 E6A2  TRANS1 E6A9  UNFMCD 0036  VCM12  E80D  VFCMT  EC09
VRF120 EC10  VRFCNT CEFA  VRFYE  EC03  VRFYX  EC00  WAITY  F0CE
WATI0  F0EE  WATI1  F10D  WATIM  F0EB  WATK0  F0DD  WATK1  F0E8
WATKY  F0D8  WCK1   E796  WCK2   E79A  WCK3   E79E  WMOD0  E948
WMOD1  E968  WPRTCD 002E  WRCAN  E1DB  XAINP  ECDD
```

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 an '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 nagative 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 occured.

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
INIR; INDR; OTIR; OTDR	●	1	X	X	1	X	Z=0 if B≠0 otherwise Z=1
LDI, LDD	●	X	:	X	0	0	Block transfer instructions
LDIR, LDDR	●	X	0	X	0	0	P/V=1 if BC≠0, otherwise P/V=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←r'	●	●	●	●	●	●	01 r r'	1	1	4	
LD r,n	r←n	●	●	●	●	●	●	00 r 110 ← n →	2	2	7	
LD r,(HL)	r←(HL)	●	●	●	●	●	●	01 r 110	1	2	7	
LD r,(IX+d)	r←(IX+d)	●	●	●	●	●	●	11 011 101 01 r 110 ← d →	3	5	19	
LD r,(IY+d)	r←(IY+d)	●	●	●	●	●	●	11 111 101 01 r 110 ← d →	3	5	19	
LD (HL),r	(HL)←r	●	●	●	●	●	●	01 110 r	1	2	7	
LD (IX+d),r	(IX+d)←r	●	●	●	●	●	●	11 011 101 01 110 r ← d →	3	5	19	
LD (IY+d),r	(IY+d)←r	●	●	●	●	●	●	11 111 101 01 110 r ← d →	3	5	19	
LD (HL),n	(HL)←n	●	●	●	●	●	●	00 110 110 ← n →	2	3	10	
LD (IX+d),n	(IX+d)←n	●	●	●	●	●	●	11 011 101 00 110 110 ← d → ← n →	4	5	19	
LD (IY+d),n	(IY+d)←n	●	●	●	●	●	●	11 111 101 00 110 110 ← d → ← n →	4	5	19	
LD A,(BC)	A←(BC)	●	●	●	●	●	●	00 001 010	1	2	7	
LD A,(DE)	A←(DE)	●	●	●	●	●	●	00 011 010	1	2	7	
LD A,(nn)	A←(nn)	●	●	●	●	●	●	00 111 010 ← n → ← n →	3	4	13	
LD (BC),A	(BC)←A	●	●	●	●	●	●	00 000 010	1	2	7	
LD (DE),A	(DE)←A	●	●	●	●	●	●	00 010 010	1	2	7	
LD (nn),A	(nn)←A	●	●	●	●	●	●	00 110 010 ← n → ← n →	3	4	13	
LD A,I	A←I	●	↕	IFF2	↕	0	0	11 101 111 01 010 111	2	2	9	IFF2: contents of interrupt enable flip-flop 2
LD A,R	A←R	●	↕	IFF2	↕	0	0	11 101 101 01 011 111	2	2	9	
LD I,A	I←A	●	●	●	●	●	●	11 101 101 01 000 111	2	2	9	
LD R,A	R←A	●	●	●	●	●	●	11 101 101 01 001 111	2	2	9	

r,r	register
000	B
001	C
010	D
011	E
100	H
101	L
111	A

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 ← n → ← n →	3	3	10	dd register pairs 00 BC 01 DE 10 HL 11 SP
LD IX,nn	IX←nn	●	●	●	●	●	●	11 011 101 00 100 001 ← n → ← n →	4	4	14	
LD IY,nn	IY←nn	●	●	●	●	●	●	11 111 101 00 100 001 ← n → ← n →	4	4	14	
LD HL,(nn)	H←(nn+1) L←(nn)	●	●	●	●	●	●	00 101 010 ← n → ← n →	3	5	16	
LD dd,(nn)	dd_H←(nn+1) dd_L←(nn)	●	●	●	●	●	●	11 101 101 01 dd1 011 ← n → ← n →	4	6	20	
LD IX,(nn)	IX_H←(nn+1) IX_L←(nn)	●	●	●	●	●	●	11 011 101 00 101 010 ← n → ← n →	4	6	20	
LD IY,(nn)	IY_H←(nn+1) IY_L←(nn)	●	●	●	●	●	●	11 111 101 00 101 010 ← n → ← n →	4	6	20	
LD (nn),HL	(nn+1)←H (nn)←L	●	●	●	●	●	●	00 100 010 ← n → ← n →	3	5	16	
LD (nn),dd	(nn+1)←dd_H (nn)←dd_L	●	●	●	●	●	●	11 101 101 01 dd0 011 ← n → ← n →	4	6	20	
LD (nn),IX	(nn+1)←IX_H (nn)←IX_L	●	●	●	●	●	●	11 011 101 00 100 010 ← n → ← n →	4	6	20	
LD (nn),IY	(nn+1)←IY_H (nn)←IY_L	●	●	●	●	●	●	11 111 101 00 100 010 ← n → ← n →	4	6	20	
LD SP,HL	SP←HL	●	●	●	●	●	●	11 111 001	1	1	6	
LD SP,IX	SP←IX	●	●	●	●	●	●	11 011 101 11 111 001	2	2	10	
LD SP,IY	SP←IY	●	●	●	●	●	●	11 111 101 11 111 001	2	2	10	

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	$(SP-2) \leftarrow qq_L$ $(SP-1) \leftarrow qq_H$	●	●	●	●	●	●	11 qq0 101	1	3	11	qq register pairs 00 BC 01 DE 10 HL 11 AF
PUSH IX	$(SP-2) \leftarrow IX_L$ $(SP-1) \leftarrow IX_H$	●	●	●	●	●	●	11 011 101 11 100 101	2	4	15	
PUSH IY	$(SP-2) \leftarrow IY_L$ $(SP-1) \leftarrow IY_H$	●	●	●	●	●	●	11 111 101 11 100 101	2	4	15	
POP qq	$qq_H \leftarrow (SP+1)$ $qq_L \leftarrow (SP)$	●	●	●	●	●	●	11 qq0 001	1	3	10	
POP IX	$IX_H \leftarrow (SP+1)$ $IX_L \leftarrow (SP)$	●	●	●	●	●	●	11 011 101 11 100 001	2	4	14	
POP IY	$IY_H \leftarrow (SP+1)$ $IY_L \leftarrow (SP)$	●	●	●	●	●	●	11 111 101 11 100 001	2	4	14	

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↔HL	●	●	●	●	●	●	00 101 011	1	1	4	
EX AF, AF'	AF↔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	Each value in register pairs is exchanged with the value in aux. register pairs.
EX (SP), HL	H↔(SP+1) L↔(SP)	●	●	●	●	●	●	11 100 011	1	5	19	
EX (SP), IX	IX_H↔(SP+1) IX_L↔(SP)	●	●	●	●	●	●	11 011 101 11 100 011	2	6	23	
EX (SP), IY	IY_H↔(SP+1) IY_L↔(SP)	●	●	●	●	●	●	11 111 101 11 100 011	2	6	23	
LDI	(DE)←(HL) DE←DE+1 HL←HL+1 BC←BC−1	●	●	① ↕	●	0	0	11 101 101 10 100 000	2	4	16	
LDIR	(DE)←(HL) DE←DE+1 HL←HL+1 BC←BC−1 repeats to BC=0	●	●	0	●	0	0	11 101 101 10 110 000	2 2	5 4	21 16	for BC≠0 for BC=0
LDD	(DE)←(HL) DE←DE−1 HL←HL−1 BC←BC−1	●	●	① ↕	●	0	0	11 101 101 10 101 000	2	4	16	
LDDR	(DE)←(HL) DE←DE-1 HL←HL-1 BC←BC-1 repeats to BC=0	●	●	0	●	0	0	11 101 101 10 111 000	2 2	5 4	21 16	for BC≠0 for BC=0
CPI	A−(HL) HL←HL+1 BC←BC−1	●	② ↕	① ↕	↕	1	↕	11 101 101 10 100 001	2	4	16	

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↔HL	●	●	●	●	●	●	11 101 011	1	1	4	
EX AF,AF'	AF↔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	Each value in register pairs is exchanged with the value in aux. register pairs.
EX (SP),HL	H↔(SP+1) L↔(SP)	●	●	●	●	●	●	11 100 011	1	5	19	
EX (SP),IX	IX_H↔(SP+1) IX_L↔(SP)	●	●	●	●	●	●	11 011 101 11 100 011	2	6	23	
EX (SP),IY	IY_H↔(SP+1) IY_L↔(SP)	●	●	●	●	●	●	11 111 101 11 100 011	2	6	23	
LDI	(DE)←(HL) DE←DE+1 HL←HL+1 BC←BC−1	●	●	① ↕	●	0	0	11 101 101 10 100 000	2	4	16	
LDIR	(DE)←(HL) DE←DE+1 HL←HL+1 BC←BC−1 repeats to BC=0	●	●	0	●	0	0	11 101 101 10 110 000	2 2	5 4	21 16	for BC≠0 for BC=0
LDD	(DE)←(HL) DE←DE−1 HL←HL−1 BC←BC−1	●	●	① ↕	●	0	0	11 101 101 10 101 000	2	4	16	
LDDR	(DE)←(HL) DE←DE-1 HL←HL-1 BC←BC-1 repeats to BC=0	●	●	0	●	0	0	11 101 101 10 111 000	2 2	5 4	21 16	for BC≠0 for BC=0
CPI	A−(HL) HL←HL+1 BC←BC−1	●	② ↕	① ↕	↕	1	↕	11 101 101 10 100 001	2	4	16	

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.

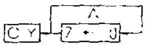
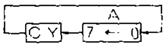
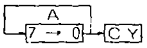
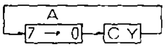
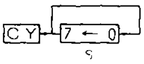
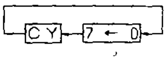
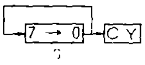
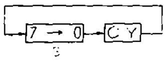
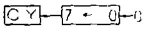
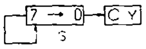
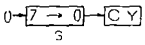
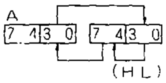
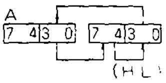
8-Bit Arithmetic and Logical Group

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←A+r	↕	↕	V	↕	0	↕	10 $\boxed{000}$ r	1	1	4	r	register
ADD A,n	A←A+n	↕	↕	V	↕	0	↕	11 $\boxed{000}$ 110	2	2	7	000	B
								← n →				001	C
ADD A,(HL)	A←A+(HL)	↕	↕	V	↕	0	↕	10 $\boxed{000}$ 110	1	2	7	010	D
ADD A,(IX+d)	A←A+(IX+d)	↕	↕	V	↕	0	↕	11 011 101	3	5	19	011	E
								10 $\boxed{000}$ 110				100	H
								← d →				101	L
ADD A,(IY+d)	A←A+(IY+d)	↕	↕	V	↕	0	↕	11 111 101	3	5	19	111	A
								10 $\boxed{000}$ 110					
								← d →					
ADC A,s	A←A+s+CY	↕	↕	V	↕	0	↕	$\boxed{001}$				The s operand is any of r, n, (HL), (IX+d) or (IY+d) as defined for the ADD instruction. Replace the opcode $\boxed{000}$ in ADD with the framed bits.	
SUB s	A←A−s	↕	↕	V	↕	1	↕	$\boxed{010}$					
SBC A,s	A←A−s−CY	↕	↕	V	↕	1	↕	$\boxed{011}$					
AND s	A←A∧s	0	↕	P	↕	0	1	$\boxed{100}$					
OR s	A←A∨s	0	↕	P	↕	0	0	$\boxed{110}$					
XOR s	A←A⊕s	0	↕	P	↕	0	0	$\boxed{101}$					
CP s	A—s	↕	↕	V	↕	1	↕	$\boxed{111}$					
INC r	r←r+1	●	↕	V	↕	0	↕	00 r $\boxed{100}$	1	1	4		
INC (HL)	(HL)←(HL)+1	●	↕	V	↕	0	↕	00 110 $\boxed{100}$	1	3	11		
INC (IX+d)	(IX+d)←	●	↕	V	↕	0	↕	11 011 101	3	6	23		
	(IX+d)+1							00 110 $\boxed{100}$					
								← d →					
INC (IY+d)	(IY+d)←	●	↕	V	↕	0	↕	11 111 101	3	6	23		
	(IY+d)+1							00 110 $\boxed{100}$					
								← d →					
DEC m	m←m−1	●	↕	V	↕	1	↕	$\boxed{101}$				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 $\boxed{100}$ changes to $\boxed{101}$.	

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+I	●	●	●	●	●	●	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 00 $\boxed{000}$ r	2	2	8	The contents of register r are rotated left.
RLC (HL)		↕	↕	P	↕	0	0	11 001 011 00 $\boxed{000}$ 110	2	4	15	r register 000 B 001 C 010 D 011 E 100 H 101 L 111 A
RLC (IX+d)		↕	↕	P	↕	0	0	11 011 101 11 001 011 ← d → 00 $\boxed{000}$ 110	4	6	23	
RLC (IY+d)		↕	↕	P	↕	0	0	11 111 101 11 001 011 ← d → 00 $\boxed{000}$ 110	4	6	23	
RL s		↕	↕	P	↕	0	0	$\boxed{010}$				The s operand is any of r, (HL), (IX+d) or (IY+d)
RRC s		↕	↕	P	↕	0	0	$\boxed{001}$				
RR s		↕	↕	P	↕	0	0	$\boxed{011}$				
SLA s		↕	↕	P	↕	0	0	$\boxed{100}$				
SRA s		↕	↕	P	↕	0	0	$\boxed{101}$				
SRL s		↕	↕	P	↕	0	0	$\boxed{111}$				
RLD		●	↕	P	↕	0	0	11 101 101 01 101 111	2	5	18	
RRD		●	↕	P	↕	0	0	11 101 101 01 100 111	2	5	18	

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	$Z \leftarrow \overline{r}_b$	●	↕	X	X	0	1	11 001 011	2	2	8	r	register
								01 b r				000	B
BIT b,(HL)	$Z \leftarrow \overline{(HL)}_b$	●	↕	X	X	0	1	11 001 011	2	3	12	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	$\bar{r}_b \leftarrow 1$	●	●	●	●	●	●	11 001 011	2	2	8	011	3
								$\boxed{11}$ b r				100	4
SET b,(HL)	$(HL)_b \leftarrow 1$	●	●	●	●	●	●	11 001 011	2	4	15	101	5
								$\boxed{11}$ b 110				110	6
SET b,(IX+d)	$(IX+d)_b \leftarrow 1$	●	●	●	●	●	●	11 011 101	4	6	23	111	7
								11 001 011					
								← d →					
								$\boxed{11}$ b 110					
SET b,(IY+d)	$(IY+d)_b \leftarrow 1$	●	●	●	●	●	●	11 111 101	4	6	23		
								11 001 011					
								← d →					
								$\boxed{11}$ b 110					
RES b,s	$s_b \leftarrow 0$ $s \equiv r$, (HL), (IX+d), (IY+d)							$\boxed{10}$				Bit b in operand s is reset	

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 ← n → ← n →	3	3	10	
JP cc,nn	if cc is true, PC←nn if false, continue	●	●	●	●	●	●	11 cc 010 ← n → ← n →	3	3	10	
JR e	PC←PC+e	●	●	●	●	●	●	00 011 000 ← e-2 →	2	3	12	
JR C,e	if C=0, continue	●	●	●	●	●	●	00 111 000 ← e-2 →	2	2	7	
	if C=1, PC←PC+e								2	3	12	
JR NC,e	if C=1, continue	●	●	●	●	●	●	00 110 000 ← e-2 →	2	2	7	
	if C=0 PC←PC+e								2	3	12	
JR Z,e	if Z=0, continue	●	●	●	●	●	●	00 101 000 ← e-2 →	2	2	7	
	if Z=1, PC←PC+e								2	3	12	
JR NZ,e	if Z=1, continue	●	●	●	●	●	●	00 100 000 ← e-2 →	2	2	7	
	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 11 101 001	2	2	8	
JP (IY)	PC←IY	●	●	●	●	●	●	11 111 101 11 101 001	2	2	8	
DJNZ,e	B←B−1 if B=0, continue	●	●	●	●	●	●	00 010 000 ← e-2 →	2	2	8	for B=0
	if B≠0 PC←PC+e								2	3	13	for B≠0

cc	condition
000	NZ non zero
001	Z zero
010	NC non carry
011	C carry
100	PO parity odd
101	PE parity even
110	P sign positive
111	M sign negative

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.

Call and Return 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				
CALL nn	$(SP-1) \leftarrow PC_H$ $(SP-2) \leftarrow PC_L$ $PC \leftarrow nn$	●	●	●	●	●	●	11 001 101 ← n → ← n →	3	5	17	
CALL cc,nn	if cc is true, equal to CALL nn if false, continue	●	●	●	●	●	●	11 cc 100 ← n → ← n →	3 3	3 5	10 17	for cc is false for cc is true
RET	$PC_L \leftarrow (SP)$ $PC_H \leftarrow (SP+1)$	●	●	●	●	●	●	11 001 001	1	3	10	
RET cc	if cc is true, equal to RET if false, continue	●	●	●	●	●	●	11 cc 000	1 1	1 3	5 11	for cc is false for cc is true
RETI	return from interrupt	●	●	●	●	●	●	11 101 101 01 001 101	2	4	14	
RETN	return from NMI	●	●	●	●	●	●	11 101 101 01 000 101	2	4	14	
RST p	$(SP-1) \leftarrow PC_H$ $(SP-2) \leftarrow PC_L$ $PC_H \leftarrow 0$ $PC_L \leftarrow P$	●	●	●	●	●	●	11 t 111	1	3	11	

cc	condition
000	NZ non zero
001	Z zero
010	NC non carry
011	C carry
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
100	20H
101	28H
110	30H
111	38H

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
C2*8405*	JP NZ,nn
C3*8405*	JP nn
C4*8405*	CALL NZ,nn
C5	PUSH BC
C6*20*	ADD A,n
C7	RST 0
C8	RET Z
C9	RET
CA*8405*	JP Z,nn
CC*8405*	CALL Z,nn
CD*8405*	CALL nn
CE*20*	ADC A,n
CF	RST 1
D0	RET NC
D1	POP DE
D2*8405*	JP NC,nn
D3*20*	OUT (n),A
D4*8405*	CALL NC,nn
D5	PUSH DE
D6*20*	SUB n
D7	RST 2
D8	RET C
D9	EXX
DA*8405*	JP C,nn
DB*20*	IN A,(n)
DC*8405*	CALL C,nn
DE*20*	SBC A,n
DF	RST 3
E0	RET PO
E1	POP HL
E2*8405*	JP PO,nn
E3	EX (SP),HL

object code	mnemonic
E48405	CALL PO,nn
E5	PUSH HL
E620	AND n
E7	RST 4
E8	RET PE
E9	JP (HL)
EA8405	JP PE,nn
EB	EX DE,HL
EC8405	CALL PE,nn
EE20	XOR n
EF	RST 5
F0	RET P
F1	POP AF
F28405	JP P,nn
F3	DI
F48405	CALL P,nn
F5	PUSH AF
F620	OR n
F7	RST 6
F8	RET M
F9	LD SP,HL
FA8405	JP M,nn
FB	EI
FC8405	CALL M,nn
FE20	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
DD9E*05*	SBC A,(IX+d)
DDA6*05*	AND (IX+d)
DDAE*05*	XOR (IX+d)
DDB6*05*	OR (IX+d)
DDBE*05*	CP (IX+d)
DDE1	POP IX
DDE3	EX (SP),IX
DDE5	PUSH IX
DDE9	JP (IX)
DDF9	LD SP,IX
DDCB*05*06	RLC (IX+d)
DDCB*05*0E	RRC (IX+d)
DDCB*05*16	RL (IX+d)
DDCB*05*1E	RR (IX+d)
DDCB*05*26	SLA (IX+d)
DDCB*05*2E	SRA (IX+d)
DDCB*05*3E	SRL (IX+d)
DDCB*05*46	BIT 0,(IX+d)
DDCB*05*4E	BIT 1,(IX+d)
DDCB*05*56	BIT 2,(IX+d)
DDCB*05*5E	BIT 3,(IX+d)
DDCB*05*66	BIT 4,(IX+d)
DDCB*05*6E	BIT 5,(IX+d)
DDCB*05*76	BIT 6,(IX+d)
DDCB*05*7E	BIT 7,(IX+d)
DDCB*05*86	RES 0,(IX+d)
DDCB*05*8E	RES 1,(IX+d)
DDCB*05*96	RES 2,(IX+d)
DDCB*05*9E	RES 3,(IX+d)
DDCB*05*A6	RES 4,(IX+d)
DDCB*05*AE	RES 5,(IX+d)
DDCB*05*B6	RES 6,(IX+d)
DDCB*05*BE	RES 7,(IX+d)
DDCB*05*C6	SET 0,(IX+d)
DDCB*05*CE	SET 1,(IX+d)
DDCB*05*D6	SET 2,(IX+d)
DDCB*05*DE	SET 3,(IX+d)
DDCB*05*E6	SET 4,(IX+d)
DDCB*05*EE	SET 5,(IX+d)

object code	mnemonic
DDCB*05*F6	SET 6,(IX+d)
DDCB*05*FE	SET 7,(IX+d)
ED40	IN B,(C)
ED41	OUT (C),B
ED42	SBC HL,BC
ED43*8405*	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
ED4B*8405*	LD BC,(nn)
ED4D	RETI
ED50	IN D,(C)
ED51	OUT (C),D
ED52	SBC HL,DE
ED53*8405*	LD (nn),DE
ED56	IM 1
ED57	LD A,I
ED58	IN E,(C)
ED59	OUT (C),E
ED5A	ADC HL,DE
ED5B*8405*	LD DE,(nn)
ED5E	IM 2
ED5F	LD A,R
ED60	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
ED73*8405*	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*05*06	RLC (IY+d)
FDCB*05*0E	RRC (IY+d)
FDCB*05*16	RL (IY+d)
FDCB*05*1E	RR (IY+d)
FDCB*05*26	SLA (IY+d)
FDCB*05*2E	SRA (IY+d)
FDCB*05*3E	SRL (IY+d)
FDCB*05*46	BIT 0,(IY+d)
FDCB*05*4E	BIT 1,(IY+d)
FDCB*05*56	BIT 2,(IY+d)
FDCB*05*5E	BIT 3,(IY+d)
FDCB*05*66	BIT 4,(IY+d)
FDCB*05*6E	BIT 5,(IY+d)
FDCB*05*76	BIT 6,(IY+d)
FDCB*05*7E	BIT 7,(IY+d)
FDCB*05*86	RES 0,(IY+d)
FDCB*05*8E	RES 1,(IY+d)
FDCB*05*96	RES 2,(IY+d)
FDCB*05*9E	RES 3,(IY+d)
FDCB*05*A6	RES 4,(IY+d)
FDCB*05*AE	RES 5,(IY+d)
FDCB*05*B6	RES 6,(IY+d)

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

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

5E 2.25-S (TINSE1318ACZZ) ①

SHARP CORPORATION
OSAKA, JAPAN