

AS-100 MS-DOS User's Guide

Canon
COMPUTER SYSTEMS

(i)

(j)

(k)

Preface

This manual summarizes features of the AS-100 series computer and functions of the AS-100 Microsoft Disk Operating System (MS-DOS) which are required to use the AS-100 series under MS-DOS. This manual should be read after reading the standard MS-DOS manual.

CONTENTS

CHAPTER 1 AS-100 MS-DOS

1.1	Outline	1
1.2	Starting the AS-100 MS-DOS	1
1.3	Memory Map	6
1.4	User Interface	7

CHAPTER 2 CHARACTERISTICS OF THE AS-100 SERIES

2.1	Overview	11
2.2	CRT Display Unit	11
2.3	Keyboard	17
2.4	Floppy Disk Unit	24

CHAPTER 3 EXTENDED UTILITY COMMANDS

3.1	Overview	27
3.2	ANSISEQ1 and ANSISEQ2	27
3.3	ASSIGN	33
3.4	CPCOPY	35
3.5	CPDIR	36
3.6	DISKCOPY	37
3.7	FORMAT	39
3.8	GRHND	41
3.9	PAGE	42

3.10	PALETTE	43
3.11	PGADD	46
CHAPTER 4 GRAPHIC FUNCTIONS		
4.1	Outline	48
4.2	Interrupt	49
4.3	Escape Sequence Execution	65
CHAPTER 5 RS232C INTERFACE		
5.1	Outline	68
5.2	RSHND Command	69
5.3	Exeuction	70
CHAPTER 6 PRINTER INTERFACE		
6.1	Overview	76
6.2	Al200	78
6.3	Al210	80
6.4	CNTHND	83
6.5	Messages	83
6.6	Execution	84
CHAPTER 7 EXTENDED INTERRUPTS		
7.1	Overview	86
7.2	Interrupts	88

APPENDIX

APPENDIX A	AS-100 MS-DOS SYSTEM PROGRAMS	104
APPENDIX B	CHARACTER CODE TABLE	105
APPENDIX C	ROM DEBUGGER	106
APPENDIX D	ESCAPE SEQUENCE	111
APPENDIX E	DIP SWITCHES	124

CHAPTER 1

AS-100 MS-DOS

1.1 Outline

The AS-100 MS-DOS is an improved version of the MS-DOS which is developed by Microsoft. The MS-DOS has been modified as follows.

- Powerfull customized BIOS has been developed.
- Bootstrap and secondary bootstrap routines have been developed for loading the AS-100 MS-DOS into the AS-100 memory.
- Expanded utility commands have been developed.

The special features which have been added to the MS-DOS are as follows.

- Extended utility commands (See Chapter 3)
- Graphic functions (See Chapter 4)
- RS232C interface (See Chapter 5)
- Printer interface (See Chapter 6)
- Expanded interrupt processing (See Chapter 7)

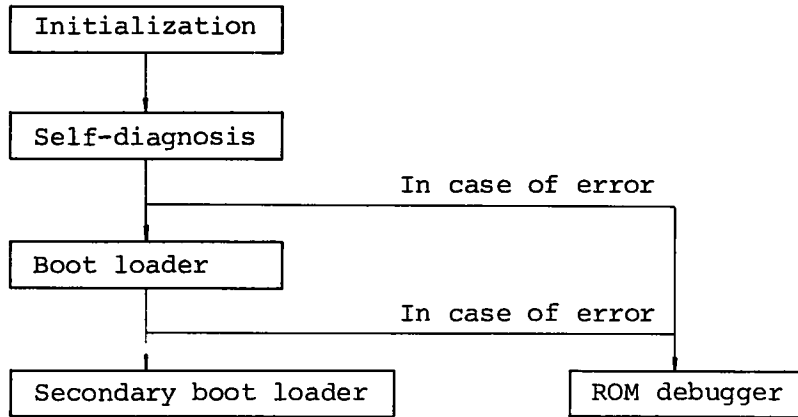
1.2 Starting the AS-100 MS-DOS

The bootstrap program which starts the AS-100 MS-DOS is stored in ROM from FE00:0000 to FE00:1FFF.

This program consists of the following.

- Initialization routine
- Self-diagnostic routine
- Bootstrap loader
- ROM debugger

When the power switch is turned on or when the reset switch is pressed, the above routines are executed in succession as shown below.



(1) Initialization

The system is initialized after control is transferred to the bootstrap program, that is, the registers which are used by the hardware are reset and the bootstrap program work area is cleared.

Control is transferred to the self-diagnostic routine after initialization has been completed.

(2) Self-diagnosis

This routine performs the following.

- RAM check

Writes and reads all the installed RAM area one byte at a time.

- ROM check

Checks the ROM contents by means of checksum.

- Timer check

Checks the software timer.

- Keyboard check

Checks the operation of the keyboard.

If any error is detected during the above checks, an error message is displayed on the screen and control is transferred to the ROM debugger; otherwise, control is transferred to the bootstrap loader.

<Error message>

RAM ERR AT nnnn:mmmm

An error was detected during RAM check.

nnnn: segment address

mmmm: offset address

ERR CODE = nn

An error was detected during self-diagnosis.

nn = 02: RAM checksum error

03: Timer error

04: Keyboard error

(3) Bootstrap loader

This routine loads the secondary bootstrap, which loads the AS-100 MS-DOS, into memory from the system disk and transfers control to it.

If an error occurs during loading the secondary bootstrap, the following error message is displayed and control is transferred to the ROM debugger; otherwise, control is transferred to the secondary bootstrap.

<Error message>

BOOT ERROR

An error was detected during loading the secondary bootstrap.

(4) Secondary bootstrap loader

This program is stored on the first sector in track 0 of a disk when it is initialized with the FORMAT command. (See Chapter 3.)

The secondary bootstrap loads the AS-100 MS-DOS into memory from the disk from which the bootstrap has been loaded and transfers control to the AS-100 MS-DOS.

(5) ROM debugger

The ROM debugger is automatically activated when an error is detected by the self-diagnostic routine or system loading fails. It can also be activated by pressing the STOP switch.

The ROM debugger has the following functions.

- Displays and changes the contents of memory.
- Displays and changes the contents of registers.
- Reads data from an input port.
- Writes data to an output port.
- Sets break points.
- Executes programs step by step.
- Executes the bootstrap.

The ROM debugger is used to display the contents of the AS-100 MS-DOS and check handlers.

The ROM debugger is not suitable for debugging user programs. Use the DEBUG utility for that purpose.

See Appendix C for details of the ROM debugger and refer to the Standard MS-DOS Manual for the DEBUG utility.

The following messages are displayed as the processing of the bootstrap program proceeds.

```
nnnK BYTES SYSTEM
```

```
MS-DOS LOADER Vm.mm
```

```
    nnn = Size of memory installed
```

```
    m.mm = Version number of the secondary bootstrap
```

```
Current data is Sat 1-01-1983
```

```
(This date is stored in the AS-100 MS-DOS.)
```

```
Enter new date:
```

```
The date can be entered in the format shown below.
```

```
MM-DD-YY or MM-DD-YYYY
```

```
MM = month
```

```
DD = date
```

```
YY or YYYY = year
```

```
Ex) 03-05-83, 03-05-1983
```

Current time is 0:00:11.00

(This time is stored in the AS-100 MS-DOS.)

Enter new time:

The time can be entered in the format shown below.

HH, HH:MM or HH:MM:SS

HH = hour

MM = minute

SS = second

Ex) 12, 12:00, 12:30:00

The data stored in the system disk is assumed to be entered when the RETURN key is pressed without entering the date or hour data.

Canon AS-100 MS-DOS Version n.nn

COPYRIGHT (C) by Microsoft 1982, all rights reserved

BIOS (J) Vm.mm by Canon Inc.

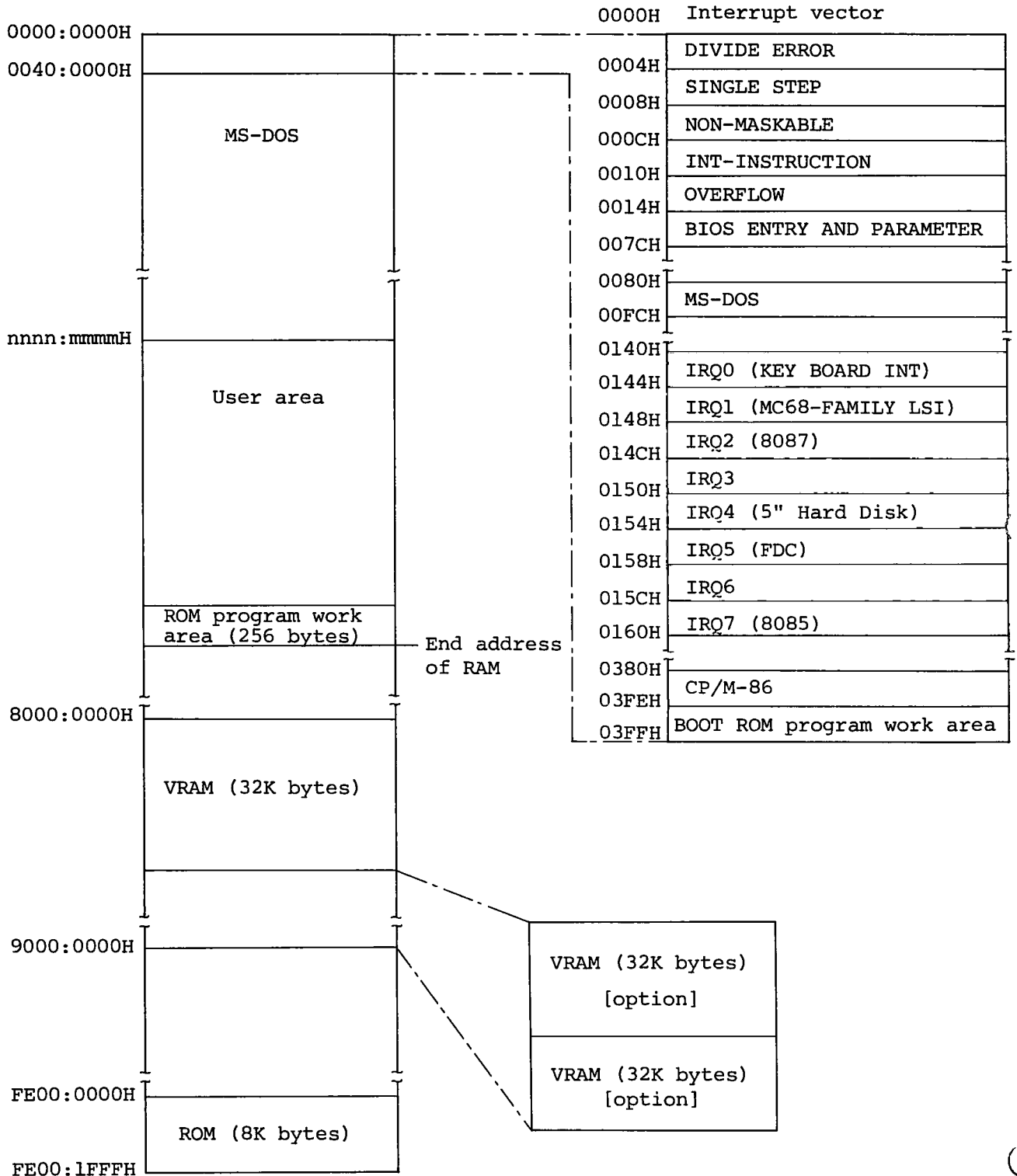
n.nn = version number of AS-100 MS-DOS

m.mm = version number of BIOS

When the system is normally started, the device from which the system has been loaded is assumed as the default device and the device name is displayed as a prompt. Now, the AS-100 MS-DOS can be used.

1.3 Memory Map

The following figure shows the memory map when the AS-100 MS-DOS is operating.



1.4 User Interface

The standard MS-DOS is provided with the special editing function (template function) and control character functions which make entry operation from the keyboard easy.

The special editing function stores the command line which has been entered last in a buffer area called template and makes it possible to edit the command line so that it can be used for the next entry.

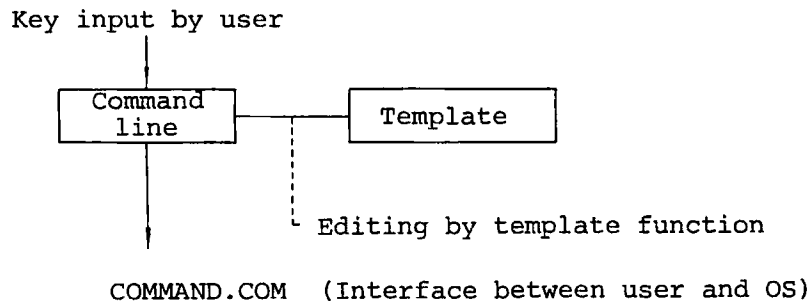
The control character functions can be used by pressing the control key together with another key.

The AS-100 MS-DOS has additional functions as well as the above functions.

(1) Special editing function

The command line which has been entered last is stored in the template buffer.

Relationship between a command line and template buffer is shown below.



With the AS-100 MS-DOS, the contents of the template buffer can be edited by using the following function keys.

Function key	Function
F1	Reads a character from the template buffer and makes its copy in the new command line.
F2	Reads characters from the template buffer until the specified character is encountered and makes a copy of that character string in the new command line.
F3	Reads all the contents of the template buffer and makes a copy of them in the new command line.
DELETE	Skips to read a character from the template buffer.
F4	Skips to read characters from the template buffer until the specified character is encountered.
ESC	Deletes characters entered so far. (The contents of the template is not changed.)
INSERT	Switches the mode between the insert mode and replace mode.
F5	Loads all the characters entered so far into the template buffer.

The above special functions are the same as those provided for the standard MS-DOS. Refer to the Manual for Standard MS-DOS for details of the functions.

Correspondence between the function keys defined for the AS-100 MS-DOS and those defined for the standard MS-DOS is shown below.

Standard MS-DOS

AS-100 MS-DOS

CL	F1
CM	F2
CT	F3
SI	DELETE
SM	F4
QI	ESC
INS	INSERT
REP	INSERT
NT	F5

(2) Control character functions

(i) Standard control character functions

- CTRL** / **C** Interrupts the processing currently being performed.
- CTRL** / **H** Deletes the character at the left of the cursor position. (Back space)
- CTRL** / **J** Moves the cursor to the next line without terminating the current command line.
The command line can be continued until the **RETURN** key is pressed.
- CTRL** / **P** Makes it possible to output a line that is displayed on the screen to the printer.
Pressing these keys together again cancels this function.
- CTRL** / **S** Suspends display. Pressing any key restarts display.

(ii) Extended control character functions

CTRL / **CLEAR SCREEN**

Suspends the current operation. Pressing any key restarts the operation.

CTRL / **SHIFT** / **CLEAR
SCREEN**

Outputs the current screen contents to the printer. If any processing is being processed, that processing is suspended. After printing is completed, the processing is restarted.

CTRL / **CANCEL**

Clears the keyboard buffer and interrupt the current processing.

The keyboard buffer contains some data and **CTRL** / **C** may not be executed if some keys are pressed before **CTRL** / **C** is entered.

CTRL / **CANCEL** is effective in this case.

X	X	CTRL/C	
---	---	--------	--

CTRL / **1** Stops generating a click each time a key is pressed. Pressing these keys again starts generating the click.

CTRL / **2** Changes the screen from the smooth scroll mode to the line scroll mode. Pressing these keys again returns it to the smooth scroll mode.

CTRL / **ALT** / **DELETE**

Reboots the system. The functions is the same as that of the reset switch.

AS-100 MS-DOS has special functions as follows.

ALT / **CANCEL**

Clears the keyboard buffer.

CANCEL

Interrupts the current processing.
(Same as **CTRL** / **C**)

DEL

Erases the character at the left of the cursor position. (Back space)

**DELETE
LINE**

Erases the command line which is being entered. This function is provided instead of the **CTRL** / **X** function which is provided for the standard MS-DOS.

CHAPTER 2

CHARACTERISTICS OF THE AS-100 SERIES

2.1 Overview

The AS-100 MS-DOS utilizes the powerful AS-100 series features in addition to the standard MS-DOS functions for more effective processing.

This chapter describes the characteristics of the AS-100 series CRT display unit, keyboard, and floppy disk drive unit.

2.2 CRT Display Unit

The AS-100 series CRT display unit employs the bit map method with 640 x 400 dots to provide various facilities such as graphic capabilities and attribute/mode modification.

(1) V-RAM configuration

The display unit is equipped with video RAM (plane) to store screen control data. One screen displays a bit map of 640 x 400 dots, which is equivalent to approximately 32K bytes.

There are two models in the AS-100 series: color and monochrome. The color model uses three planes with the RGB method. The monochrome model is further classified into two types: one-plane and two-plane. For the two-plane monochrome model, the high intensity and blink attributes can be specified. See (2) below for details on the attributes.

Model	Color	Monochrome	
Number of planes	3	2	1

(2) Palette registers

When displaying a character in a desired color or with a desired attribute, the user specifies the number of a register called a palette.

Up to eight palette registers are provided, numbered from 0 to 7. The number of palette registers actually available depends on the CRT display unit type (AS-100 model) as follows:

Model	Palettes available
Color	No. 0 ~ 7
Monochrome two-plane	No. 0 ~ 3
Monochrome one-plane	No. 0 and 1

Each palette register is given an initial value, and it can be modified by interrupt INT 10H (see Chapter 8). When a palette value is modified, it corresponds to the newly specified color or attribute.

Palette register initial values

(Color model)

Palette number	value	Color
0	0	No display
1	1	Blue
2	3	Green
3	4	Cyan
4	9	Red
5	12	Magenta
6	10	Yellow
7	13	White

(Monochrome one-plane model)

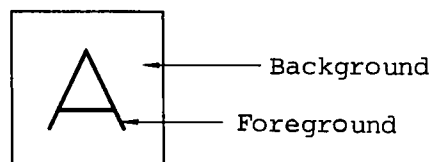
Palette number	value	Attribute
0	0	No display
1	1	Standard intensity

(Monochrome two-plane model)

Palette number	value	Attribute
0	0	No display
1	2	High intensity
2	27	Standard intensity blink
3	1	Standard intensity

Each character box on the screen is divided into two fields, foreground and background, which are assigned different palette registers. That is, each field appears in the color or with the attribute represented by the corresponding palette register.

Character box



The table below shows the initial palette registers assigned to the foreground and background. This assignment can be modified by interrupt INT 10H explained in Chapter 8.

CRT display model	Foreground	Background
Color	Palette 7 (White)	Palette 0 (No display)
Monochrome two-plane	Palette 3 (Standard intensity)	Palette 0 (No display)
Monochrome one-plane	Palette 1 (Standard intensity)	Palette 0 (No display)

The following explains the values that can be set in palette registers to specify colors or attributes.

(i) Color model

Up to 27 colors can be specified by combining red, green, and blue in standard intensity and red, green, and blue in half intensity.

Value	r R g G b B	Remarks
0	0 0 0 0 0 0	No display
1	0 0 0 0 0 1	Blue
2	0 0 0 0 1 1	
3	0 0 0 1 0 0	Green
4	0 0 0 1 0 1	Cyan
5	0 0 0 1 1 1	
6	0 0 1 1 0 0	
7	0 0 1 1 0 1	
8	0 0 1 1 1 1	
9	0 1 0 0 0 0	Red
10	0 1 0 0 0 1	Yellow
11	0 1 0 0 1 1	
12	0 1 0 1 0 0	Magenta
13	0 1 0 1 0 1	White
14	0 1 0 1 1 1	
15	0 1 1 1 0 0	
16	0 1 1 1 0 1	
17	0 1 1 1 1 1	
18	1 1 0 0 0 0	
19	1 1 0 0 0 1	
20	1 1 0 0 1 1	
21	1 1 0 1 0 0	
22	1 1 0 1 0 1	

Value	r R g G b B	Remarks
23	1 1 0 1 1 1	
24	1 1 1 1 0 0	
25	1 1 1 1 0 1	
26	1 1 1 1 1 1	
27, 28	1 1 1 1 1 1	

R: Red in standard intensity
G: Green in standard intensity
B: Blue in standard intensity
r: Red in half intensity
g: Green in half intensity
b: Blue in half intensity

(ii) Monochrome model

Value	B H S	Remarks
0	0 0 0	No display
1	0 0 1	Standard intensity
2	0 1 0	High intensity
3 ~ 26	0 0 1	Standard intensity
27	1 0 1	Standard intensity blink
28	1 1 0	High intensity blink

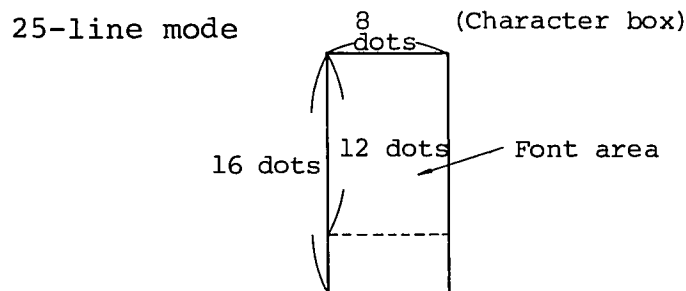
B: Blink
H: High intensity
S: Standard intensity

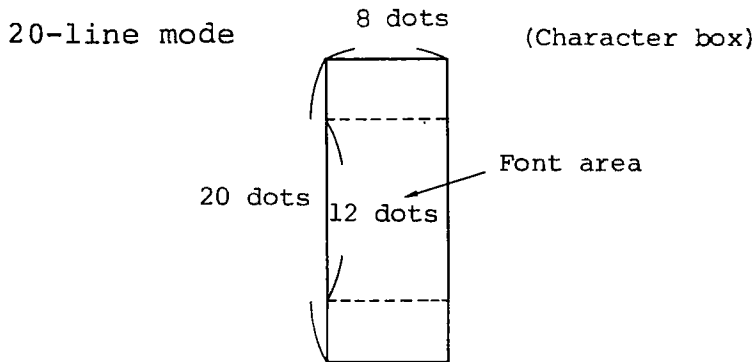
(3) Modes

The CRT display unit can operate in many different ways depending on the specified modes. Interrupt INT 10H, explained in Chapter 8, is used to modify the display modes.

(i) Line count and character box size

The display unit operates in 25-line or 20-line mode, signifying the number of lines on screen. The character box size depends on the line count mode, but the character font size does not change. The initial value is the 25-line mode.





(ii) Line length

The line length can be 80 or 40 characters. If the 40-character mode is specified, each character is automatically expanded horizontally in BIOS to be twice as wide as in 80-character mode, before it is displayed.

(iii) Scrolling

There are two modes for scrolling: smooth scroll and line scroll. The scrolling operation is faster in line scroll mode than in smooth scroll mode. Note that the line scroll mode can be specified only if the entire screen is defined to be the scroll area.

The initial value is the smooth scroll mode. The user may specify part of the screen to be the scroll area.

(iv) Cursors

A character cursor and a graphic cursor can be displayed on screen. The character cursor represents the location of a character box, while the graphic cursor represents the location of a dot.

Both cursors can appear at the same time on screen. In the initial state, only the character cursor appears.

(4) Character buffer

The AS-100 MS-DOS BIOS includes a 4K-byte character buffer which stores the codes and attributes of the characters being displayed. The buffer contents are updated whenever data is displayed on screen.

The PGADD command (see Chapter 3) can be used to implement additional character buffers, corresponding to up to three screens, in memory. Interrupt INT 10H (see Chapter 8) is used to switch the additional buffer.

(5) ANK character display

The AS-100 series CRT display unit can display characters indicated in the character code table given in Appendix B.

The underlining and reverse display operations are also allowed, through the use of interrupt INT 10H (see Chapter 8).

There are four codes that are treated as a control character or an ordinary character depending on the output method, as explained below.

Treated as control characters

When any of the following codes is output by the BIOS CONOUT feature, it is treated as a control character and the corresponding function is executed:

0AH: LF (Move the cursor to the same column on the next line.)

0DH: CR (Move the cursor to the leftmost column of the current line.)

07H: BELL (Generate buzzer sound.)

08H: BS (Move the cursor one column to the left.)

Treated as ordinary characters

When any of these four codes is output by interrupt INT 10H (see Chapter 8), the corresponding symbol appears on the screen.

0AH: ┘

0DH: └

07H: ♪

08H: ≡

2.3 Keyboard

The AS-100 series keyboard consists of four types of keys as follows:

- ASCII keys
- Numeric keypad
- Function keys
- Special keys

Some of the keys are provided with the repeat capability; the same character is entered successively by holding down the key.

An optional pointing device can be used with the keyboard.

The following describes the four types of keys.

(1) ASCII keys

The ASCII keys are used to enter characters from the keyboard. When an ASCII key is pressed, the character to be entered depends on the on/off conditions of the CTRL, ALPHA-LOCK, SHIFT, and ALT keys. These four keys are on when held down, and are off when released. The following tables show the codes to be generated and characters to be displayed by the ASCII keys when other keys are on or off.

All ASCII keys have the repeat capability.

KEY	Base	Upper
0	0	
1	1	
2 "	2	"
3 #	3	#
4 \$	4	\$
5 %	5	%
6 &	6	&
7 '	7	'
8 (8	(
9)	9)
: *	:	*
; +	;	+
, <	,	<
- =	-	=
. >	.	>
/ ?	/	?
@ \	@	\
A	a	A
B	b	B
C	c	C
D	d	D
E	e	E
F	f	F
G	g	G

KEY	Base	Upper
H	h	H
I	i	I
J	j	J
K	k	K
L	l	L
M	m	M
N	n	N
O	o	O
P	p	P
Q	q	Q
R	r	R
S	s	S
T	t	T
U	u	U
V	v	V
W	w	W
X	x	X
Y	y	Y
Z	z	Z
[{	[{
\	\	
] }]	}
^ ~	^	~
- -	-	-

Base: When the ALPHA-LOCK and SHIFT keys are both on or both off

Upper: When either one of the ALPHA-LOCK and SHIFT keys is on.

KEY	CTRL	ALT	KEY	CTRL	ALT
0		<00,81>	H	<08>	<00,23>
1 !		<00,78>	I	<09>	<00,17>
2 "		<00,79>	J	<0A>	<00,24>
3 #		<00,7A>	K	<0B>	<00,25>
4 \$		<00,7B>	L	<0C>	<00,26>
5 %		<00,7C>	M	<0D>	<00,32>
6 &		<00,7D>	N	<0E>	<00,31>
7 '		<00,7E>	O	<0F>	<00,18>
8 (<00,7F>	P	<10>	<00,19>
9)		<00,80>	Q	<11>	<00,10>
: *			R	<12>	<00,13>
; +			S	<13>	<00,1F>
, <			T	<14>	<00,14>
- =		<00,83>	U	<15>	<00,16>
. >			V	<16>	<00,2F>
/ ?			W	<17>	<00,11>
@ ,	<00,29>		X	<18>	<00,2D>
A	<01>	<00,1E>	Y	<19>	<00,15>
B	<02>	<00,30>	Z	<1A>	<00,2C>
C	<03>	<00,2E>	[{	<1B>	
D	<04>	<00,20>	\	<1C>	
E	<05>	<00,12>] }	<1D>	
F	<06>	<00,21>	^ ~	<1E>	
G	<07>	<00,22>	- -	<1F>	<00,82>

CTRL: When the CTRL key is held down

ALT: When the ALT key is held down

< >: <XX> indicates that a one-byte hexadecimal code XX is generated. <XX,YY> indicates that a two-byte hexadecimal code XXYY is generated.

Blank: No code is generated.

(2) Numeric keypad

The functions of the keys on the numeric keypad depend on whether the CURSOR-LOCK key is on or off. The on/off status of the CURSOR-LOCK key can be switched by pressing the key.

- While CURSOR-LOCK is off, the numeric keypad is used for numeric value entry.
- While CURSOR-LOCK is on, the numeric keypad is used to move the cursor on screen.

The keys on the numeric keypad have the repeat capability.

If a character code is entered in decimal notation from the numeric keypad while the ALT key is held down, the corresponding character appears on the screen when the ALT key is released. This enables the user to enter character codes directly from the keyboard.

Example: If 65 is entered while ALT is held down, and then ALT is released, character code 41H (65 in decimal) is generated and letter A is displayed on screen.

If the 00 key is pressed while the ALT key is held down, the current numeric value entry is canceled.

The following table shows the codes to be generated and characters to be displayed when the keys on the numeric keypad are pressed.

KEY	Base	Upper	CTRL
0	0	0	
1	1	<00,4F>	<00,75>
2	2	<00,50>	<00,A5>
3	3	<00,51>	<00,76>
4	4	<00,4B>	<00,73>
5	5	<00,4C>	<00,A6>
6	6	<00,4D>	<00,74>
7	7	<00,47>	<00,77>
8	8	<00,48>	<00,A7>
9	9	<00,49>	<00,84>
00	00	00	
.	.	.	
-	-	-	

Base: When the CURSOR-LOCK and SHIFT keys are both on or both off

Upper: When either one of the CURSOR-LOCK and SHIFT keys is on

The meanings of CTRL, < >, and blank in this table are the same as those used in the tables in "(1) ASCII keys" above.

(3) Function keys

The function keys generate codes which depend on the on/off status of the SHIFT key. The SHIFT key is on when it is held down, and is off when released. The function keys only generate codes. They do not cause any character to appear on screen.

Function keys F1 to F5 are used to execute special edit functions. The table below shows the codes to be generated by the function keys under different on/off conditions of other keys.

KEY	Base	Upper	CTRL	ALT
F 1	<00,3B>	<00,54>	<00,5E>	<00,68>
F 2	<00,3C>	<00,55>	<00,5F>	<00,69>
F 3	<00,3D>	<00,56>	<00,60>	<00,6A>
F 4	<00,3E>	<00,57>	<00,61>	<00,6B>
F 5	<00,3F>	<00,58>	<00,62>	<00,6C>
F 6	<00,40>	<00,59>	<00,63>	<00,6D>
F 7	<00,41>	<00,5A>	<00,64>	<00,6E>
F 8	<00,42>	<00,5B>	<00,65>	<00,6F>
F 9	<00,43>	<00,5C>	<00,66>	<00,70>
F10	<00,44>	<00,5D>	<00,67>	<00,71>
F11	<00,94>	<00,96>	<00,98>	<00,9A>
F12	<00,95>	<00,97>	<00,99>	<00,9B>

Base: When the SHIFT key is off

Upper: When the SHIFT key is on

The meanings of CTRL, < >, and blank in this table are the same as those used in the tables in "(1) ASCII keys" above.

(4) Special keys

The special keys refer to all keys except the ASCII keys, numeric keypad, and function keys.

When a special key is pressed, a code is generated and the corresponding function is executed (see Chapter 1). For example, the control character function is executed by using a special key with the CTRL or ALT key.

The SPACE and DEL keys have the repeat capabilities.

KEY	Base	Upper
SPACE	<20>	<20>
ESC	<1B>	<1B>
DEL	<7F>	<7F>
LINE FEED	<0A>	<0A>
←	<0D>	<0D>
CLEAR SCREEN	<00,77>	<00,77>
DELETE LINE	<00,90>	<00,90>
CANCEL	<03>	<03>
COPY	<00,91>	<00,A1>
MOVE	<00,92>	<00,A2>
DELETE	<00,53>	<00,A3>
INSERT	<00,52>	<00,A4>
↔	<09>	<00,0F>
START	<0D>	<0D>

Base: When the SHIFT key is off

Upper: When the SHIFT key is on

(5) Pointing device

The pointing device is used to move the cursor on screen. There are two cursor control modes: character cursor mode and graphic cursor mode. The mode can be switched by generating interrupt INT 10H (see Chapter 8).

In character cursor mode, operating the slider generates a two-byte code as if a numeric key is pressed while CURSOR-LOCK is on.

Slider direction	Code
↑	<00,48>
↓	<00,50>
→	<00,4D>
←	<00,4B>

In graphic cursor mode, the pointing device is used to move the graphic cursor on screen. In this mode, graphic cursor movement is always done in BIOS. The user can only find where the graphic cursor is when a function key is pressed. This means that, in graphic cursor mode, no key code is generated by operating the slider.

The pointing device is supplied with three function keys A to C. Pressing a function key generates a specific key code as shown below.

Key	Code
A	
B	<00,9D>
C	<00,9E>

Function key A does not generate a code. This key is used to find the graphic cursor location in graphic cursor mode.

In AS-100 MS-DOS, the pointing device is always controlled by interrupt INT 10H. See Chapter 8 for further details.

2.4 Floppy Disk Unit

The following two types of floppy disk unit can be connected to the AS-100 series:

- Minifloppy disk unit (A-1300) for 5-inch diskettes
- Standard floppy disk unit (A-1330) for 8-inch diskettes

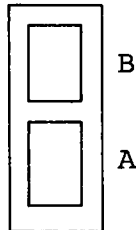
The following describes the floppy disk unit configuration and usage.

(1) Configuration

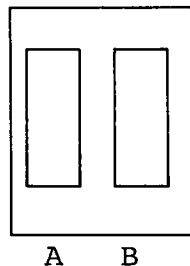
Each disk unit contains two disk drives. In the AS-100 series, the five types of disk unit configuration shown below are allowed. Logical device names A to D are assigned to the drives. When issuing a command to access a specific floppy disk drive, the user specifies the logical device name of the drive.

The logical device name assignment can be modified by operating the dip switches in the disk unit (see Appendix E).

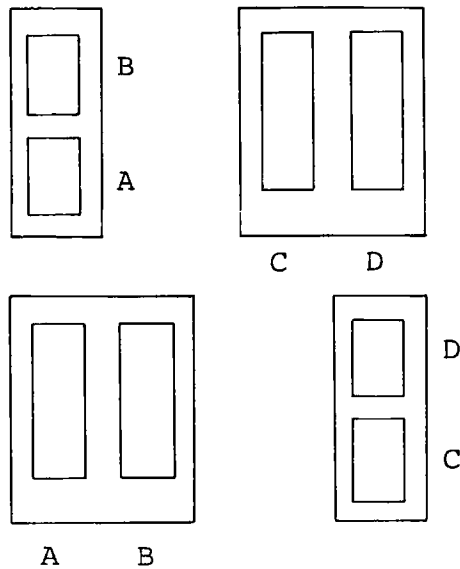
(i) One minifloppy disk unit



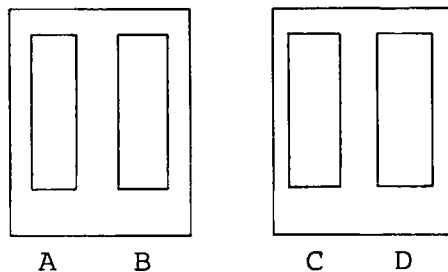
(ii) One standard floppy disk unit



(iii) One minifloppy disk unit and one standard floppy disk unit



(iv) Two standard floppy disk units



(2) Usage

The following floppy diskettes can be used with these floppy disk units in the AS-100 series:

- 5-inch diskettes: Double-sided, double-density, double-track, with total capacity 640K bytes
- 8-inch diskettes: Double-sided, double-density, with total capacity 1M bytes

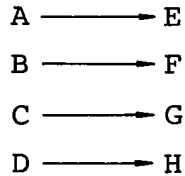
The FORMAT command must be used to format diskettes before they can be used under AS-100 MS-DOS control. See Chapter 3 for an explanation on the FORMAT command.

(3) Special disk access

The following diskettes can also be used with the floppy disk units by specifying logical device names E to H (instead of A to D) in the command:

- (1) 5-inch diskettes: Single-sided, double-density
(512 bytes per sector)
- (2) 8-inch diskettes: Single-sided, single-density
(128 bytes per sector)

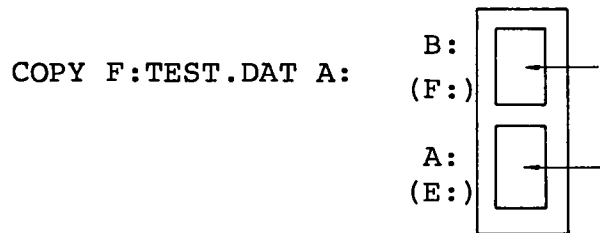
Logical device names E to H correspond to names A to D as shown below.



Example: Suppose that a file must be copied from one drive to the other in a minifloppy disk unit under the following conditions:

- Input diskette: Single-sided, double-density
- Output diskette: Double-sided, double-density, double-track
- File name: TEST.DAT

In this case, the following command is used:



CHAPTER 3

EXTENDED UTILITY COMMANDS

3.1 Overview

The AS-100 MS-DOS is provided with the following extended utility commands in addition to the standard MS-DOS commands:

ANSISEQ1: These two commands enable escape sequence execution
ANSISEQ2: under AS-100 MS-DOS control.

ASSIGN: Assigns logical devices to physical devices.

CPCOPY: Copies AS-100 CP/M-86 diskfiles to an AS-100 MS-DOS disk.

CPDIR: Displays AS-100 CP/M-86 disk directory information.

DISKCOPY: Copies physically the contents of a disk to another disk of the same type, in high-speed mode.

FORMAT: Formats disks for use by AS-100 MS-DOS.

GRHND: Enables the graphic capabilities to be used under AS-100 MS-DOS control.

PAGE: Switches the screen using an additional CRT character buffer installed in memory.

PALETTE: Modifies the color or attribute defined for a palette register.

PGADD: Reserves a memory area for a CRT character buffer.

The following details each of these commands.

3.2 ANSISEQ1 and ANSISEQ2

Type: External commands

Format:

```
ANSISEQ1 { ON  
          OFF }  
ANSISEQ2 { ON  
          OFF }
```

Function: These two commands enable escape sequence execution under AS-100 MS-DOS control.

Description:

Generally, the AS-100 MS-DOS cannot execute escape sequences. However, the user can enable escape sequence execution by issuing the ANSISEQ1 or ANSISEQ2 command with the ON parameter. When ON is specified, the CONIN and CONOUT vector tables in BIOS are updated and the program for escape sequence execution is loaded into the addresses indicated by the updated tables.

The difference between ANSISEQ1 and ANSISEQ2 is that ANSISEQ2 can also execute escape sequences for the graphic feature.

The GRHND command (see Section 3.8) must be issued before ANSISEQ2 ON can be used.

Examples: ANSISEQ1 ON

This enables escape sequence execution.

ANSISEQ2 ON

This enables escape sequence execution for the graphic feature as well.

When the ANSISEQ1 or ANSISEQ2 command is issued with the OFF parameter, the original vector table contents are restored and the AS-100 MS-DOS also returns to its original state.

Example: ANSISEQ1 OFF

However, the program for escape sequence execution remains in memory even after this command is executed. This program can be used immediately when ON is specified.

The tables below list the escape sequences that can be executed by issuing the ANSISEQ1 and ANSISEQ2 commands with the ON parameter. For details on the escape sequences, see Chapter 4 and Appendix D.

Escape sequences enabled by ANSISEQ1 ON

No.	Escape sequence	Function
1	ESC D (1B44h)	Index
2	ESC E (1B45h)	Line feed
3	ESC M (1B4Dh)	Reverse index
4	ESC c (1B63h)	Initialization
5	ESC [Pn A	Moves the cursor upward.
6	ESC [Pn B	Moves the cursor downward.

No.	Escape sequence	Function
7	ESC [Pn C	Moves the cursor to the right.
8	ESC [Pn D	Moves the cursor to the left.
9	ESC [Pℓ;PcH	Moves the cursor to the specified position.
10	ESC [OK	Clears the current cursor position through the end of the line.
11	ESC [1K	Clears the head of the current cursor line through the cursor position.
12	ESC [2K	Clears the current cursor line entirely.
13	ESC [0J	Clears the current cursor position through the bottom line of the scroll area.
14	ESC [1J	Clears the first line of the scroll area through the current cursor position.
15	ESC [2J	Clears the scroll area entirely.
16	ESC [0m	Cancels the character display attribute.
17	ESC [1m	Specifies the highlight attribute.
18	ESC [4m	Specifies underlining.
19	ESC [5m	Specifies the blink attribute.
20	ESC [7m	Specifies reverse display.
21	ESC [Pnm	Specifies the foreground and background colors.
22	ESC [Pf;Ptr	Specifies the scroll area.
23	ESC [>0h	Displays the character cursor.
24	ESC [>1h	Displays the graphic cursor.
25	ESC [>2h	Specifies the 25-line mode.
26	ESC [>3h	Specifies the smooth scroll mode.
27	ESC [>4h	Specifies the graphic cursor mode for the pointing device.
28	ESC [>5h	Specifies the character box size (16- or 20-dot mode).
29	ESC [>0ℓ	Deletes the character cursor from the screen.
30	ESC [>1ℓ	Deletes the graphic cursor from the screen.

No.	Escape sequence	Function
31	ESC [>2ℓ	Specifies the 20-line mode.
32	ESC [>3ℓ	Specifies the line scroll mode.
33	ESC [>4ℓ	Specifies the character cursor mode for the pointing device.
34	ESC [>5ℓ	Specifies the character box size (16- or 20-dot mode).
35	ESC [>6n	Obtains the current graphic cursor position.
36	ESC [6n	Obtains the current cursor position.
37	ESC [>0;Pnc	Specifies a color for the character cursor.
38	ESC [>1;Pnc	Specifies a color for the graphic cursor.

Escape sequences enabled by ANSISEQ2 ON

No.	Command following ESC P	Parameters	Function
1	M	x, y	Moves the current point.
2	D	None	Plots a dot.
3	L	x, y	Plots a line.
4	R	x, y[, p]	Plots and paints a rectangle.
5	C	r[, p]	Plots and paints a circle.
6	E	r _x ; r _y , θ[,p]	Plots and paints an ellipse.
7	F	r, θ ₁ , θ ₂ [,p]	Plots and paints a fan shape.
8	A	r, θ ₁ , θ ₂	Plots an arc.
9	S	p	Plots a mark.
10	Q	None	Paints inside the closed surface.
11	P	p	Paints inside the closed curve.
12	T	"character string"	Display text (expanded or slanted)
13	W	p ₁ , p ₂	Specifies a color or attribute.

No.	Command following ESC P	Parameters	Function
14	X	p	Specifies the line type.
15	Y	p	Specifies the paint pattern.
16	Z	p ₁ , p ₂ , p ₃	Specifies the character size and angle.

x, y: Coordinates

r : Radius

θ : Angle in degrees

p : Numeric string defined by the function

The BIOS CONIN feature varies when an ANSISEQ1 or ANSISEQ2 command is issued with the ON parameter. When a key is pressed, CONIN generates an escape sequence code as shown in the table below. This means that, after ON is specified, some functions (such as the template function) cannot be executed because different key codes are used.

KEY	SHIFT KEY OFF	SHIFT KEY ON
F1	ESC O	ESC o
F2	ESC P	ESC p
F3	ESC Q	ESC q
F4	ESC R	ESC r
F5	ESC S	ESC s
F6	ESC T	ESC t
F7	ESC U	ESC u
F8	ESC V	ESC v
F9	ESC W	ESC w
F10	ESC X	ESC x
F11	ESC Y	ESC y
F12	ESC Z	ESC z
←→	(09)	ESC 7
CLEAR SCREEN	ESC [2J	ESC [2J
DELETE LINE	(18)	(18)
COPY	ESC 3	ESC 3
MOVE	ESC 4	ESC 4
DELETE	ESC 5	ESC 5
INSERT	ESC 6	ESC 6
PD A	ESC 0	ESC 0
PD B	ESC 1	ESC 1
PD C	ESC 2	ESC 2

KEY	CURSOR-LOCK KEY ON	CURSOR-LOCK KEY OFF
0	ESC [N	0
1	ESC [I	1
2	ESC [B	2
3	ESC [F	3
4	ESC [D	4
5	ESC [G	5
6	ESC [C	6
7	ESC [H	7
8	ESC [A	8
9	ESC [E	9

Numbers in parentheses are hexadecimal.

3.3 ASSIGN

Type: External command

Format: ASSIGN [<logical device>=<physical device>]

Function: This command assigns a logical device to a physical device.

Description:

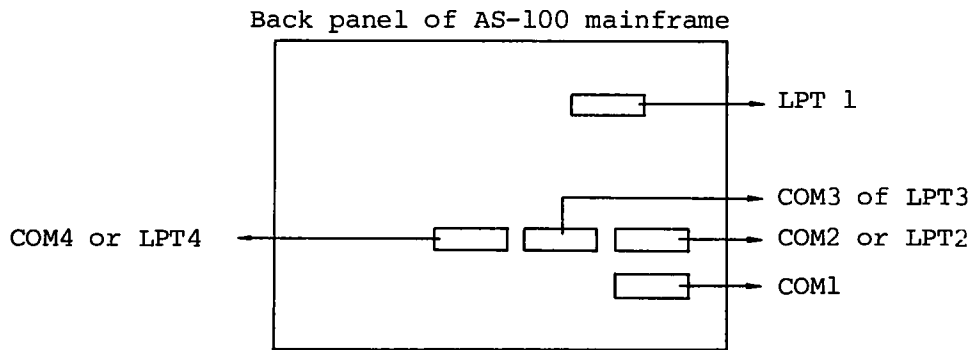
The following three logical devices are used in the AS-100 MS-DOS system:

Logical device	Meaning
CON:	Console
AUX:	Auxiliary I/O port
PRN:	Printer output port

The physical devices used in AS-100 MS-DOS are as follows:

Physical device	Meaning
CRT	Keyboard and display
COM1 ~ COM4	Ports for RS232C
LPT1 ~ LPT4	Parallel ports for printer
DUM	Dummy device

COM1 to COM 4 and LPT 1 to LPT 4 correspond to ports as shown below.



The ASSIGN command is used to modify the assignment of physical devices to the physical and logical devices defined in BIOS. After a new logical device is assigned to a physical device, the physical device can be accessed through the new logical device name.

The following shows the physical devices to which each logical device can be assigned:

```
CON: = CRT or COM1 to COM4
AUX: = Any physical device
PRN: = Any physical device
```

Example: ASSIGN AUX: = COM3

The following message appears on screen when the specified assignment is done successfully:

```
"Normally Assigned"
```

The ASSIGN command with no parameters has a different function. It displays information on physical devices to which logical devices are currently assigned.

Example: ASSIGN

```
COM: = CRT
AUX: = DUM
PRN: = LPT1
COM-Port = None    (Nothing is mounted)
LPT-Port = LPT1
```

COM-Port and LPT-Port indicate the physical device names of all currently-mounted RS232C ports and parallel printer ports, respectively. For the physical device name of an RS232C port, in addition, a minus sign (-) is displayed if the RSHND command (see Chapter 5) has not been executed.

Examples: COM2- ... The handler has not been loaded by the RSHND command for specific conditions.

COM2 ... The handler has been loaded.

<Error messages>

Invalid Parameter

An invalid parameter was specified.

Invalid Assignment

An attempt was made to assign a logical device to an invalid physical device.

Port does not exist

The specified physical device did not exist in the system.

3.4 CPCOPY

Type: External command

Format: CPCOPY <file-1> [<file-2>]

Function: This command copies AS-100 CP/M-86 disk files indicated by <file-1> into an AS-100 MS-DOS disk indicated by <file-2>.

Description:

The <file-1> and <file-2> parameters are coded in the same way as the parameters for the MS-DOS COPY command. For details, refer to the COPY command explanation given in the Standard MS-DOS manual.

Example: CPCOPY B:*.CMD A:

This command copies all files of type CMD stored in the AS-100 CP/M-86 disk mounted in drive B into the AS-100 MS-DOS disk mounted in drive A without changing the file names.

If the <file-2> parameter is omitted, the current drive is assumed.

Example: CPCOPY B:ABC.SRC

This command copies the ABC.SRC file from the AS-100 CP/M-86 disk mounted in drive B into the AS-100 MS-DOS disk mounted in the current drive without changing the file name.

The following message appears on screen when copying is completed:

```
B:ABC.CMD --> A:ABC.CMD
```

```
1 file(s) copied
```

The CPCOPY command copies data in 128-byte units due to an AS-100 CP/M-86 restriction. This means that the last record copied to the AS-100 MS-DOS disk may contain unexpected data.

<Error messages>

Parameter error

An invalid parameter was specified.

Can't read directory

The AS-100 CP/M-86 disk directory could not be read.

Not found file

The specified file was not found in the AS-100 CP/M-86 disk.

Error when physical reading

A read error occurred while a file was being read from the AS-100 CP/M-86 disk.

File can't be copied onto itself

The same disk drive was specified for both input and output.

3.5 CPDIR

Type: External command

Format: CPDIR [<d:>][<file-name>]

Function: This command displays AS-100 CP/M-86 disk directory information on screen.

Description:

If the drive name is omitted, the current drive is assumed. The wild card specification is allowed for the <file-name> parameter. That is, the parameters for the CPDIR command are identical to those for the AS-100 CP/M-86 DIR command.

As an exception, unlike the AS-100 CP/M-86 command, the CPDIR command cannot discriminate files by the system attribute. This means that information on files with the system attribute is also displayed by the CPDIR command.

<Error message>

No file

The specified file did not exist.

3.6 DISKCOPY

Type: External command

Format: DISKCOPY [<d1:>] [<d2:>]

Function: This command physically copies the contents of the disk on device d1: into the disk on device d2: in units of sectors.

Description:

The DISKCOPY command can handle the following types of diskettes:

- 5-inch diskettes: 512 bytes per sector, double-sided, double-density, double-track
- 8-inch diskettes: 1024 bytes per sector, double-sided, double-density

In addition, it can handle the following types when pseudo-device codes (E:, F:, G:, and H:) are specified (see Section 2.4):

- 5-inch diskettes: 512 bytes per sector, single-sided, double-density
- 8-inch diskettes: 128 bytes per sector, single-sided, single-density

When a DISKCOPY command is issued, the system displays a confirmation message as shown in the example below.

Example: DISKCOPY A: B:

Buffer size: nnn k bytes*2

This shows the buffer area size
required for copying in decimal.

Insert source diskette for drive A:

& insert target diskette for drive B:

and strike any key when ready

The user then inserts the input and output diskettes into the drives specified in the command, and presses any key to start copying.

When copying is completed normally, the following message appears on screen:

Diskette copy complete

Copy another? (Y/N)

Enter "Y" to continue copying operation with another pair of diskettes. Enter "N" to terminate copying.

If the drive name (<d2:>) is omitted, the current default drive (indicated by the prompt message) is assumed.

Example: DISKCOPY B:

This command copies the contents of the diskette mounted on drive B into the diskette on the default drive.

Example: DISKCOPY

If the drive names (<d1:> and <d2:>) are both omitted, copying is done using the default drive only. In this case, the following two messages are output alternately until the input disk contents are copied entirely to the output disk:

Insert source diskette for drive X:

and strike any key when ready

Insert the input disk into the default drive and press any key. The disk contents are then read into the buffer until the buffer is full.

Insert target diskette for drive X:

and strike any key when ready

Remove the input disk from the default drive, insert the output disk into the drive, and press any key. The buffer contents are then output to the disk.

<Error messages>

Diskette copy failure

An error occurred during copying.

Invalid parameter

An invalid parameter was specified.

Diskette type unmatched

The input and output diskettes were not of the same type.

Lack of memory

There was not enough space in memory for the buffer required for copying.

Drive X: Not attached

The specified disk drive did not exist.

3.7 FORMAT

Type: External command

Format: FORMAT [<d:>][/S][/C]

Function: This command formats the diskette mounted on the specified drive so that the diskette can be accessed by AS-100 MS-DOS.

Description:

The FORMAT command can format the following two types of diskettes:

- 5-inch diskettes: Double-sided, double-density, double-track
- 8-inch diskettes: Double-sided, double-density

The following two operations are carried out during disk formatting:

- Copying the second boot program into track 0, sector 1 of the diskette
- Initializing the directory and the file allocation table

If the /S parameter is specified, the following system files are copied from the disk on the default drive into the formatted disk:

- ASIO.SYS
- ASMSDOS.SYS
- COMMAND.COM

This means that the AS-100 MS-DOS FORMAT command covers all the functions of the standard MS-DOS FORMAT command.

In addition, the AS-100 MS-DOS FORMAT command is provided with more functions as described below.

If the FORMAT command is issued for an 8-inch floppy disk, physical formatting is done automatically for sectors. That is, any double-sided double-density 8-inch diskette can be handled by the FORMAT command regardless of the number of sectors in the diskette. After being formatted, the diskette has the following attributes:

- 8-inch double-sided double-density
- 8 sectors per track
- 1024 bytes per sector

This physical sector formatting operation is suppressed if the /C parameter is specified.

Physical sector formatting only applies to 8-inch floppy disks. When issuing the FORMAT command for a 5-inch diskette, be sure that the diskette has the following attributes:

- 5-inch double-sided double-density double-track
- 8 sectors per track
- 512 bytes per sector

The following shows the maximum number of files that can be registered in the directory of a formatted diskette:

- 5-inch diskette: 112 files
- 8-inch diskette: 192 files

Example: `FORMAT B:/S`

This command formats the diskette on drive B: and copies system files into the diskette.

Example: `FORMAT /S`

This command formats the diskette on the default drive because the drive name is omitted, and copies system files into the diskette.

When a FORMAT command is issued, the system outputs the following message:

- Insert new diskette for drive d:
- and strike any key when ready

Insert the diskette into the specified drive and press any key to start formatting the diskette.

When formatting is completed normally, the following messages appear:

- System transferred --- This appears only if /S was specified.

1250304 bytes total disk space
33792 bytes used by system
1216512 bytes available on disk

Format another (Y/N)?

Enter "Y" if there are more disks to be formatted, or enter "N" if not.

<Error messages>

Invalid drive specification

An invalid drive was specified.

Invalid parameter

An invalid parameter was specified.

Format failure

An error occurred during the disk formatting procedure.

Fatal format failure

A fatal error occurred during the disk formatting procedure. The system terminates the FORMAT command and displays the system prompt.

3.8 GRHND

Type: External command

Format: GRHND

Function: This command enables the graphic functions to be used under AS-100 MS-DOS control.

Description:

The GRHND command loads the graphic handler program into memory for graphic processing under AS-100 MS-DOS control.

Example: GRHND

When the graphic handler program is loaded successfully into memory, the following message appears on screen:

-- Graphic Handler Vn.m --

If the GRHND command is issued while the graphic handler program is already in memory, the system ignores the command and displays the following message:

**** Handler Already Loaded ****

See Chapter 4 for details on graphic function execution.

3.9 PAGE

Type: External command

Format: PAGE [<page-number>]

Function: This command switches the screen using an additional character buffer.

Description:

The AS-100 MS-DOS system allows installation of additional character buffers equivalent to up to three screens. The PGADD command, explained in Section 3.11, is used for such installation.

A character buffer corresponding to one screen of data is prepared in AS-100 MS-DOS BIOS in the standard configuration. Therefore, up to four screens of data can be stored in buffers in total. In 40-column-per-line mode, this number is doubled and up to eight screens can be handled.

The PAGE command replaces the current screen with the contents of the character buffer specified by the <page-number> parameter. The <page-number> must be in the following range:

80-column-per-line mode: 0 to 3

40-column-per-line mode: 0 to 7

Example: PAGE 2

This command replaces the current screen with the contents of the character buffer to which page number 2 is assigned.

Note that only character codes and attributes are recorded in a character buffer. This means that symbols and characters generated by the graphic functions do not appear on the screen displayed by the PAGE command.

If the <page-number> parameter is omitted, the system displays messages to indicate the total number of character buffers

(pages) prepared and the page number of the currently displayed screen. See the example below.

Example: PAGE

```
-- 4 Page(s) in 80 column Mode (8 Page in 40) --  
* Now on Page 1
```

<Error message>

** Page Does Not Exist **

There was no character buffer in memory to which the specified <page-number> was assigned.

3.10 PALETTE

Type: External command

Format: PALETTE [<color>],[<color>],...,<color>]

Function: This command modifies the color or attribute codes set in palette registers.

Description:

The maximum number of parameters that can be specified in the PALETTE command depends on the display model as follows:

- 8 for color model
- 4 for monochrome two-plane model
- 2 for monochrome one-plane model

If more parameters are specified than the maximum, the excess parameters are ignored. The parameters must be specified in the order corresponding to the palette numbers as shown below.

```
PALETTE [<color>],[<color>],...,<color>]  
          ↑           ↑           ↑  
Palette # 0           1     ....  7
```

If the parameter for a specific palette is omitted, the current color/attribute definition of the palette remains unchanged. When the PALETTE command is completed normally, the new color/attribute definitions of the palettes are displayed on screen.

See Chapter 2 for details on palettes.

The meanings of the <color> parameters depend on the display model as described below.

For color model

Color patterns are specified in the parameters.

Up to 27 different colors can be used on the display screen by combining the following basic colors:

- R: Red in standard intensity
- G: Green in standard intensity
- B: Blue in standard intensity
- N: No display
- r: Red in half intensity
- g: Green in half intensity
- b: Blue in half intensity

The table below lists the color pattern codes that can be specified in the PALETTE command parameters and their code numbers.

Code	Pattern
0	N
1	B
2	b
3	G
4	GB
5	Gb
6	g
7	gB
8	gb
9	R
10	RB
11	Rb
12	RG
13	RGB
14	RGb
15	Rg

Code	Pattern
16	RgB
17	Rgb
18	r
19	rB
20	rb
21	rG
22	rGB
23	rGb
24	rg
25	rgB
26	rgb

Example: PALETTE ,R,r,,rGB,N

This command modifies the current palette definitions as follows:

<u>Palette #</u>	<u>Definition</u>
0	Unchanged
1	Modified to color R
2	Modified to color r
3	Unchanged
4	Modified to color rGB
5	Modified
6	Unchanged
7	Unchanged

For monochrome two-plane model

The parameters indicate the attributes through the following codes:

S: Standard intensity
H: High intensity
SB: Standard intensity blink
HB: High intensity blink
N: No display

For monochrome one-plane model

The parameters indicate the attributes through the following codes:

D: Display (in standard intensity)
N: No display

If a PALETTE command is issued with no parameters, the system displays the meanings of the PALETTE command parameters that can be specified, as explained below.

Example: PALETTE

◦ For color model

```
-- PALETTE SET VX.XX --  
** Color Model --- Attributes **  
* Any combination of R or r, G or g, B or b
```

```

* R:RED           G:GREEN           B:BLUE
* r:Half RED     g:Half GREEN     B:Half BLUE
* <ex> Rgb, B, rB, g, ... etc.
* Or N for Non Display

```

◦ For monochrome two-plane model

```

-- PALETTE SET VX.XX --
** Monochrome 2 Plane Model -- Attributes **
* S -- Standard Intensity
* H -- High Intensity
* SB -- Standard Intensity Blink
* HB -- High Intensity Blink
* N -- Non Display

```

◦ For monochrome one-plane model

```

-- PALETTE SET VX.XX --
** Monochrome 1 Plane Model -- Attributes **
* D -- Display
* N -- Non Display

```

3.11 PGADD

Type: External command

Format: PGADD [<screen-count>]

Function: This command generates additional character buffers in memory.

Description:

A character buffer corresponding to one screen is prepared in AS-100 MS-DOS BIOS. It stores the codes and attributes of characters displayed on screen (see Chapter 2 for details).

The PGADD command is used to implement additional character buffers in memory. Buffers equivalent to up to three screens can be additionally installed.

The <screen-count> parameter indicates the number of screens to which the buffers to be added correspond.

Example: PGADD 2

This command implements additional character buffers corresponding to two screens in memory.

When character buffer allocation is completed normally, the following message appears on screen:

```
  ** n Page(s) Added (Total m Pages) **
    ↑                               ↑
Number of buffers      Total number of
newly added           additional buffers
```

If the <screen-count> parameter is omitted, 1 is assumed. Additional character buffers can be used to switch the screen. The PAGE command, explained in Section 3.9, is used for screen switching.

The additional character buffers must reside in the same segment as BIOS. This means that the user must issue the PGADD command for additional buffer allocation before loading any handler program into memory.

<Error messages>

**** Invalid Parameter ****

An invalid parameter was specified.

**** Maximum Page Over (now n Page(s): maximum 4) ****

An attempt was made to add more buffers than the maximum.

**** Not Enough Memory In System Segment ****

The additional character buffer area address was outside the range that can be accessed by BIOS.

CHAPTER 4
GRAPHIC FUNCTIONS

4.1 Outline

A wide variety of graphic functions are provided: these can be used by loading a program to execute graphic functions into the memory with GRHND command (refer to Chapter 3).

These functions can be used in either of two manners below.

(1) Access by interrupt (INT 15H)

See the function code and parameters in registers, then execute interrupt INT 15H.

(2) Escape sequence

Use the ANSISEQ2 command (refer to Chapter 3) to modify BIOS so that escape sequences can be executed by CONOUT, then use the escape sequences.

Coordinate system

The range used for executing graphic functions are as follows.

$$-32768 \leq x \leq 32767$$

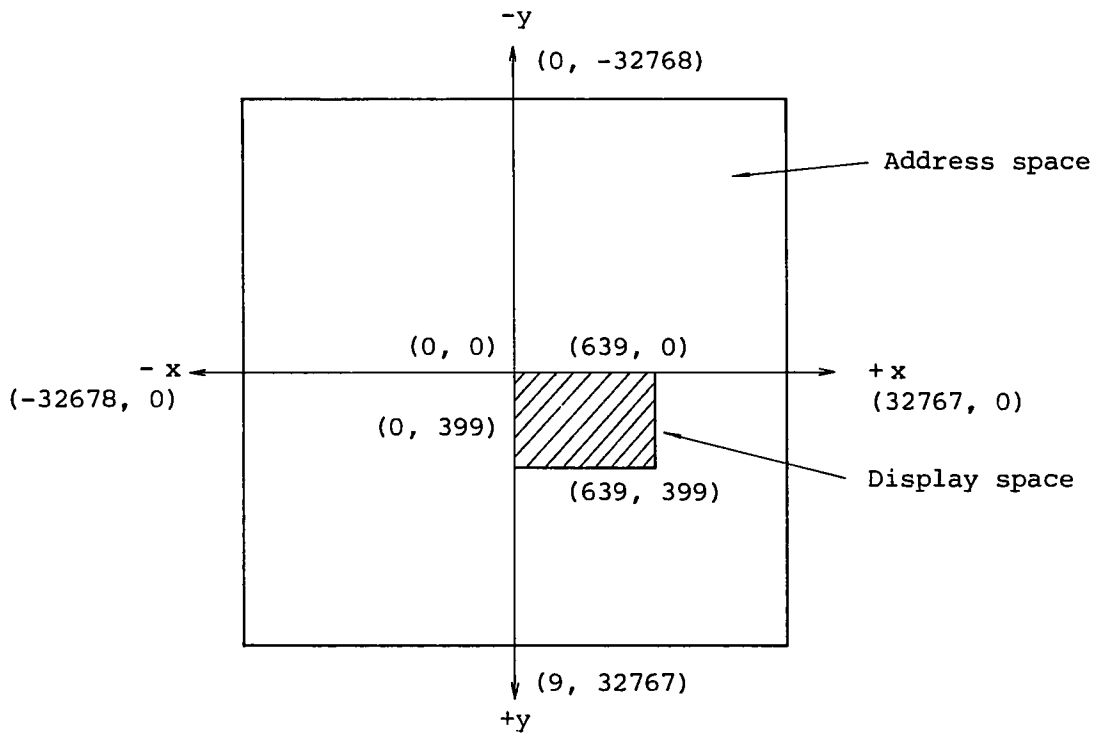
$$-32768 \leq y \leq 32767$$

However, only points in the following range are displayed on the screen.

$$0 \leq x \leq 639$$

$$0 \leq y \leq 399$$

The VRAM read and write functions and the paint functions allow only this area to be specified. The other functions allow the entire area to be specified but only the display area is displayed.



Current point (CP)

The current point is used as the point of reference for drawing figures on the screen. It is initially set to the origin of the coordinate system (0, 0).

Color and attribute

Characters and figures are displayed using the colors and attributes of the current palette.

It is possible to assign different colors to the foreground and background respectively and is also possible to mix colors with the paint function by specifying the mesh, half-tone and other paint patterns.

4.2 Interrupt

Set a function code in register CL and parameters in other registers, then interrupt with the INT 15H instruction. The following graphic functions can be used.

No.	Function code	Parameter	Function
1	0	AX: x BX: y	Changes the position of the current point.
2	1	None	Displays a dot at the position indicated for the current point.
3	2	AX: x BX: y	Draws a line between the current point and the specified point.
4	3	AX: x BX: y BP: Paint specification	Draws a rectangle.
5	4	AX: Radius BP: Paint specification	Draws a circle.
6	5	AX: Radius BX: Starting angle CX: Internal angle BP: Paint specification	Draws a fan shape.
7	6	AX: Radius BX: Starting angle DX: Internal angle	Draws an arc.
8	7	AL: Symbol pattern	Draws a symbol.
9	8	AL: Boundary color number	Paints an area enclosed with the specified color.
10	9	None	Paints a contiguous area.
11	10	AX: Character code AH=None:AL 8 bit AH≠None:AX 16 bit	Displays one character on the screen.
12	11	AX: x BX: y DI: Transfer destination offset address ES: Transfer destination segment address	Reads V-RAM data from the rectangular area defined by a diagonal line between the current point and the point specified and stores it in a specified area in memory.
13	12	AX: x BX: y DI: Transfer source offset address ES: Transfer source segment address	Writes data from an area in memory specified in the parameters to a rectangular area in V-RAM defined by a diagonal line connecting the current point and the point specified.

No.	Function code	Parameter	Function
14	13	AX: Major axis BX: Minor axis DX: Angle of major axis to positive X-axis	Draws an ellips.
15	14	BP: Paint specification AH: Foreground color or attribute number AL: Background color or attribute number	Specifies colors or attributes.
16	15	AL: Line type	Specifies the line type.
17	16	AL: Paint pattern	Specifies the paint pattern.
18	17	AH: Vertical scale AL: Horizontal scale BL: Angle of rotation	Specifies the size and rotation angle of characters.

These functions are described in detail below.

(1) Move current point

Function code CL: 0

Parameters AX: x
BX: y

Functions Moves the current point to the position specified by x and y.

(2) Dot

Function code CL: 1

Parameters None

Function Displays a dot at the coordinates of the current point. The color and attribute of the dot are determined by the color or attribute of the foreground for the current palette.

(3) Line

Function code CL: 2

Parameters AX: x
BX: y

Function Draws a line from the current point to the point specified with the parameters. The position of the current point is updated to the position specified. The line type used is that which is currently specified. The color or attribute used are those assigned to the foreground for the current palette.

(4) Rectangle

Function code CL: 3

Parameters AX: x
 BX: y
 BP: Paint specification

 { 0: Not painted.
 1: Painted.
 If this parameter is omitted, the
 rectangle is not painted.

Function Draws a rectangle around the diagonal line connecting the current point and the point specified. The position of the current point is not changed. Unless "1" is specified for BP, the rectangle is drawn using the line type currently specified. Otherwise, the rectangle is drawn with solid line and the area surrounded by the rectangle is painted using the currently specified paint pattern. The color or attribute used are those of the foreground for the current palette.

(5) Circle

Function code CL: 4

Parameters AX: Radius (positive value)
 BP: Paint specification

 { 0: Not painted
 1: Painted
 If omitted, the circle is not painted.

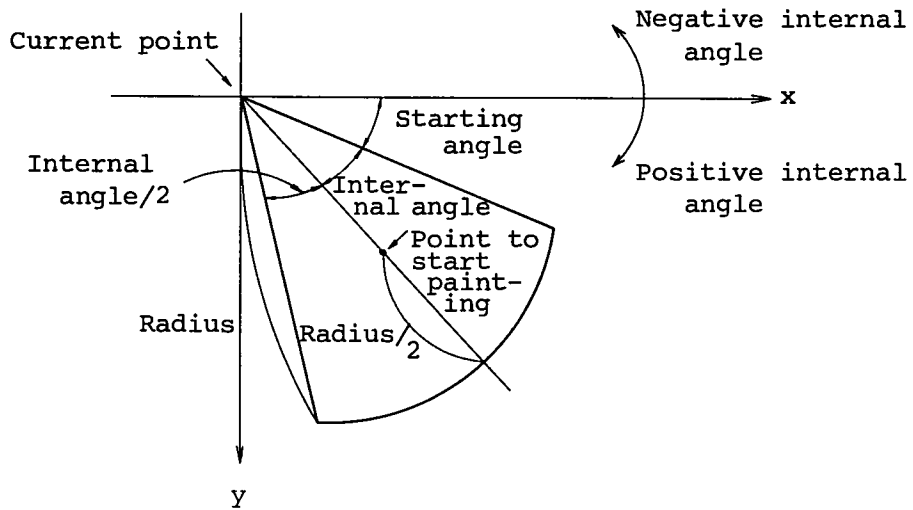
Function Draws a circle of the specified radius centered on the current point. The position of the current point is not changed. Unless "1" is specified for BP, the circle is drawn using the currently specified line type. Otherwise, the circle is drawn with solid line and painted according to the currently specified paint pattern. The color or attribute are those of the foreground for the current palette.

(6) Fan shape

Function Code CL: 5

Parameters AX: Radius (positive number)
 BX: Starting angle (degrees)
 { Positive: Clockwise
 { Negative: Counter-clockwise
 DX: Internal angle (degrees)
 BP: Paint specification
 { 0: Not painted.
 { 1: Painted.
 { If omitted, the fan shape is not painted.

Function Draws a fan shape of the specified radius and internal angle centering it on the current point with the specified starting angle to the positive X-axis. The position of the current point is not changed. Unless "1" is specified for BP, the current line type is used. Otherwise, the fan shape is drawn with solid line and painted using the current paint pattern. Since the area is gradually painted from a starting point, the entire inner area will not be painted if it is divided into subareas. The color and attribute are those of the foreground for the current palette.

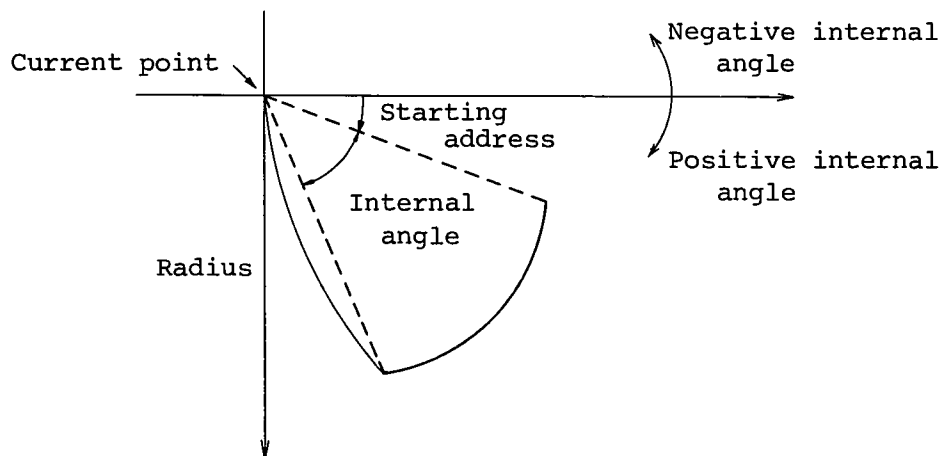


(7) Arc

Function code CL: 6

Parameters AX: Radius
 BX: Starting angle (degrees)
 { Positive value: Clockwise
 { Negative value: Counter-clockwise
 DX: Internal angle (degrees)

Function Draws an arc of the specified radius and internal angle centering it on the current point with the specified starting angle to the positive X-axis. The position of the current point is not changed. The current line type is used and the color or attribute are those of the foreground for the current palette.



(8) Symbols

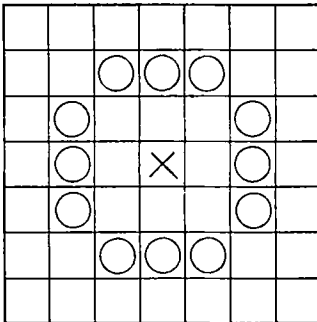
Function code CL: 7

Parameter AL: Symbol pattern number

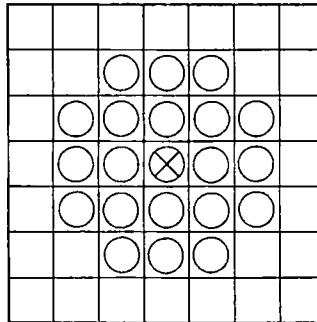
- 0: ○
- 1: ●
- 2: □
- 3: ■
- 4: ×
- 5: △
- 6: ▲

Function Draws the symbol specified in AL around the current point with the currently specified color or attribute. The dot patterns of the symbols are shown below. x in each pattern indicates the current point.

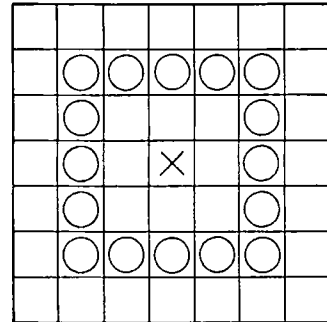
Symbol 0



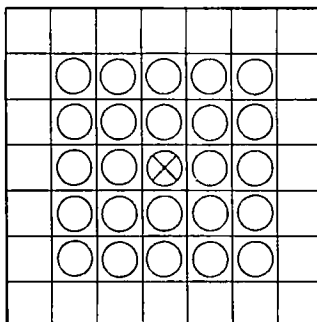
Symbol pattern 1



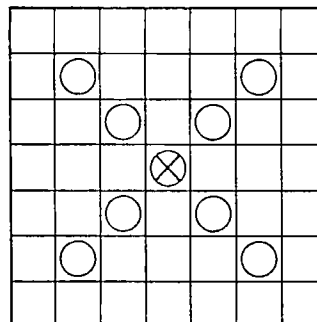
Symbol pattern 2



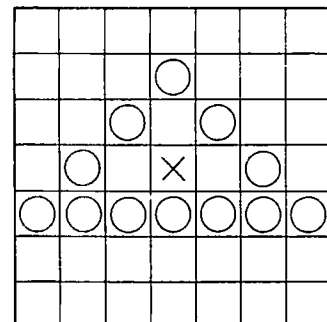
Symbol pattern 3



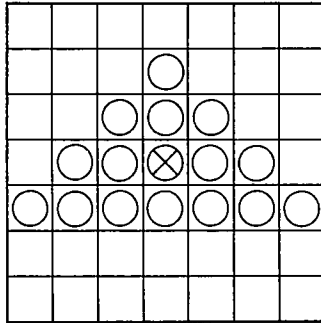
Symbol pattern 4



Symbol pattern 5



Symbol pattern 6



(9) Painting an enclosed area

Function code CL: 8

Parameter AL: Palette number for boundary color

Function Paints the area from the current point through the position at which the color or attribute specified by the parameter is displayed, with the color or attribute assigned to the foreground palette. If the screen does not include a point where the specified color or attribute is displayed, the entire screen is painted with the color or attribute assigned to the foreground palette. If the screen already includes a shape drawn by the specified palette number, the entire screen excluding that area is painted with the foreground color or attribute. The area is painted with the currently effective paint pattern. The current point position remains unchanged.

(10) Painting a contiguous area

Function code CL: 9

Parameters None

Function Paints the contiguous area that starts with the current point and contains the color or attribute of the current point, with the color or attribute assigned to the foreground palette. The area is painted with the currently effective paint pattern. The current point position remains unchanged.

(11) Displaying a character

Function code CL: 10

Parameter AX: Character code

Function Displays the character corresponding to the specified character code in the character box whose upper left corner is the current point position. The dots making up the character to be displayed conform to the color or attribute assigned to the foreground palette. The size and rotation angle of the character are identical to those that are effective when the interrupt is generated. The current point position moves to the upper left corner of the next character box.

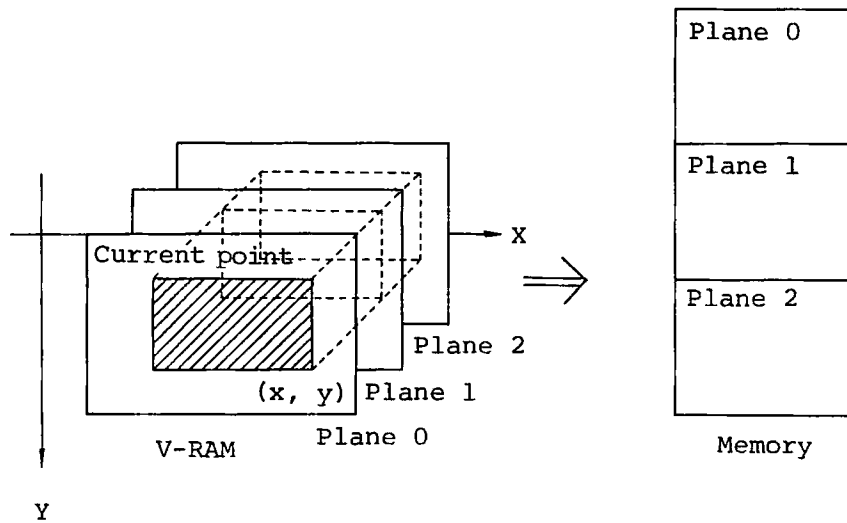
(12) Reading from V-RAM

Function code CL: 11

Parameters AX: x coordinate value
BX: y coordinate value
DI: Destination offset address
ES: Destination segment address

Function Transfers V-RAM information from the rectangular area whose diagonal line connects the current point to the specified position, to the specified memory area. Transfer is done in bit units. If V-RAM consists of several planes, bit map transfer is done for plane 0, then for plane 1, and so on. During bit map transfer, the plane area excluding the rectangular area is padded with null codes (00H). Transfer is suppressed if the current point and the specified x,y position do not both reside on the plane.

(For color model)



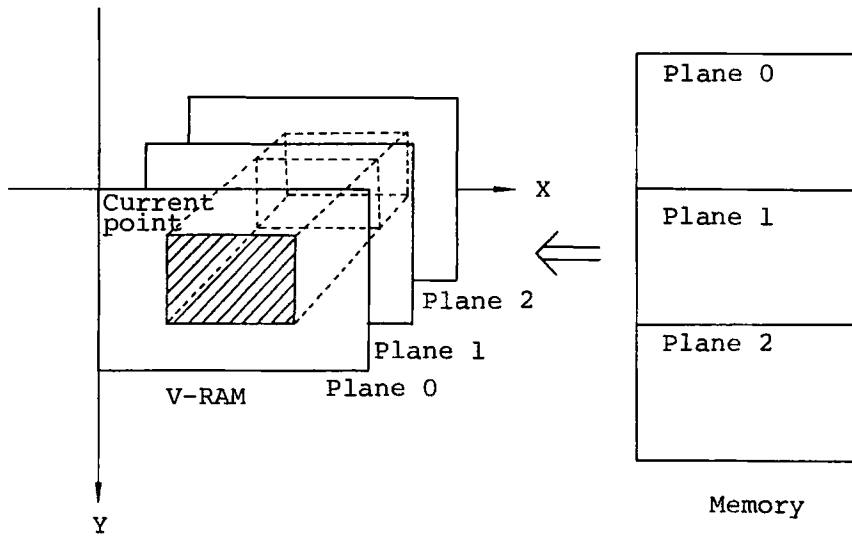
(13) Writing to V-RAM

Function code CL:12

Parameters AX: x coordinate value
BX: y coordinate value
DI: Destination offset address
ES: Destination segment address

Function Transfers V-RAM information from the memory area specified by the DI and ES parameters to the rectangular area whose diagonal line connects the current point and the specified x,y position.
If V-RAM consists of several planes, transfer is done successively for all the planes.
Transfer is suppressed if the current point and the specified x,y position do not both reside in display area.

(For color model)



(14) Drawing an ellipse

Function code CL: 13

Parameters

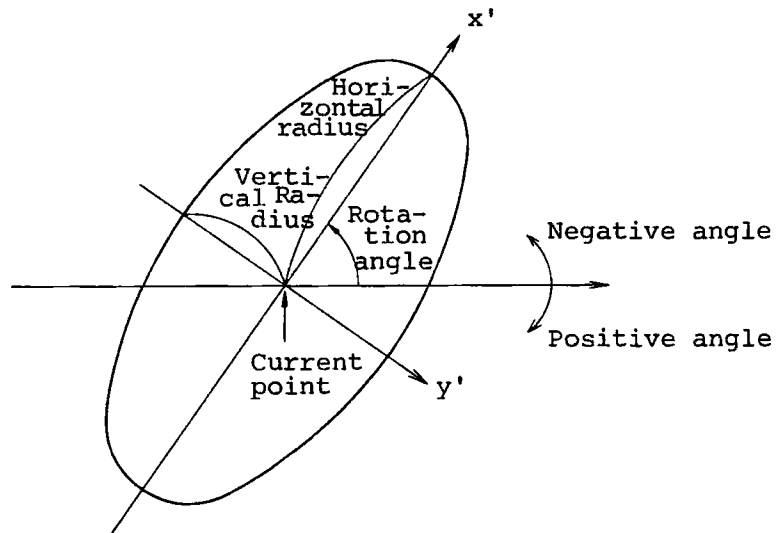
- AX: Horizontal radius
- BX: Vertical radius
- DX: Rotation angle in degrees
 - { Positive: Clockwise
 - { Negative: Counterclockwise
- BP: Painting specification
 - { 0: Painted
 - { 1: Not painted (default)

Function

Draws an ellipse centered on the current position, using the specified horizontal radius, vertical radius, and rotation angle.

If 1 (painted) is not specified for BP, the ellipse is drawn with the currently effective type of line.

If 1 is specified for BP, it is drawn with a solid line and painted with the currently effective paint pattern. Note that the painting feature handles the ellipse painting operation. This means that, if shapes or lines are included in the ellipse, only the contiguous area containing the current point is painted. The color or attribute assigned to the foreground palette is used for drawing the ellipse.



(15) Specifying a colors or attributes

Function code CL: 14

Parameters AH: Foreground palette number
AL: Background palette number

Function Specifies the colors or attributes to be assigned to the foreground and background palette. When AS-100 MS-DOS is loaded again, the palettes are initialized to their initial values shown below.

Color model

Palette number	Color
0	Black (no display)
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Yellow
7	White

Monochrome two-plane model

Palette number	Attribute
0	No display
1	High intensity
2	Standard intensity blink
3	Standard intensity





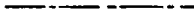
Monochrome one-plane model

Palette number	Attribute
0	No display
1	Standard intensity

(16) Specifying the line type

Function code CL: 15

Parameters AL: Line type

- 0: Solid line (default) 
- 1: Short chain line 
- 2: Long chain line 
- 3: Alternate long and short dash line 
- 4: Alternate long and two short dashes line 

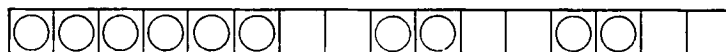
Function Specifies the type of line to be displayed. The specified line type remains effective until this interrupt is issued again.

The following shows the dot patterns of the five line types:

0 Solid line



1 Short chain line



2 Long chain line



3 Alternate long and short dash line



4 Alternate long and two short dashes line



(17) Specifying the paint pattern

Function code CL: 16

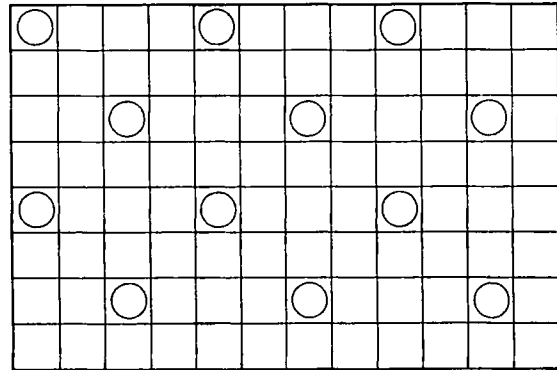
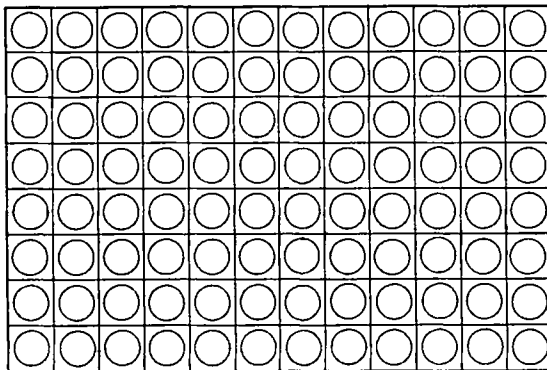
Parameter AL: Paint pattern

- 0: Entire painting (default)
- 1: Half tone
- 2: Left slanting stripes
- 3: Right slanting stripes
- 4: Vertical stripes
- 5: Horizontal stripes
- 6: Slanting meshes
- 7: Meshes
- 8: Intense half tone

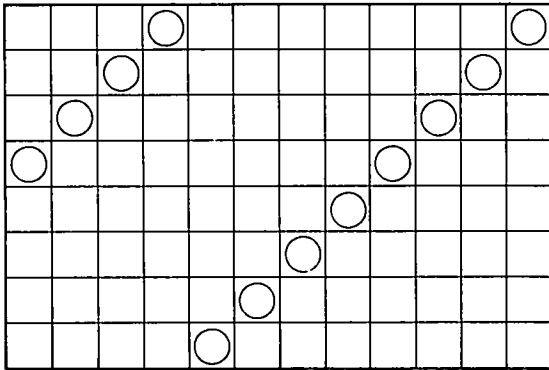
Function Specifies the paint pattern to be used, which is one of the following:

0. Entire painting (default)

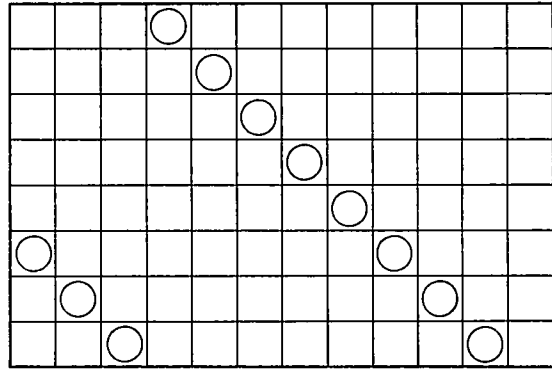
1. Half tone



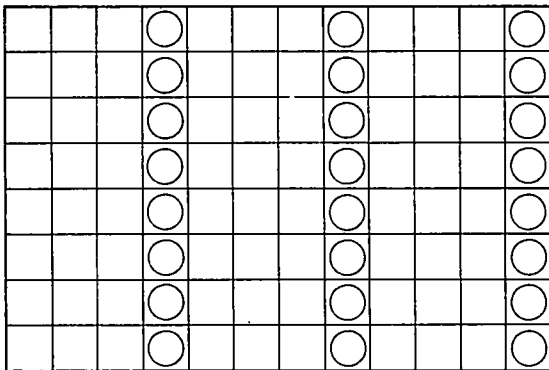
2. Left slanting stripes



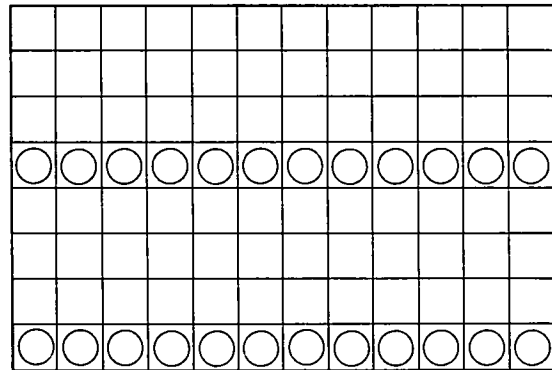
3. Right slanting stripes



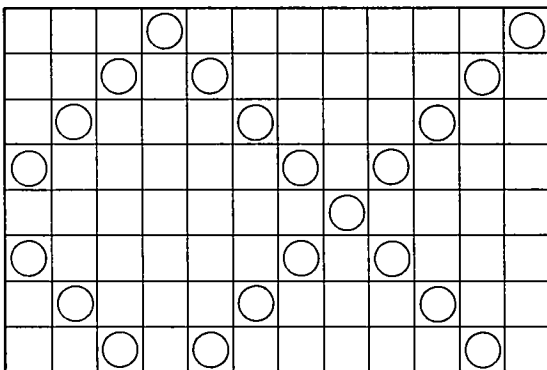
4. Vertical stripes



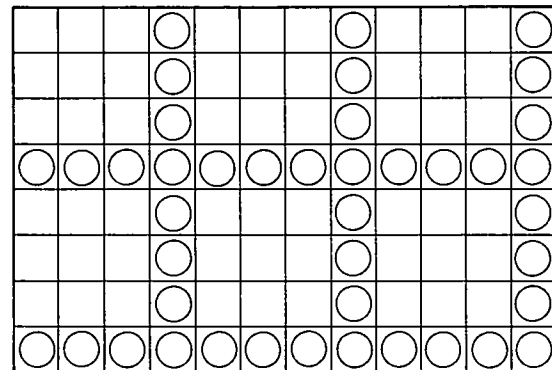
5. Horizontal stripes



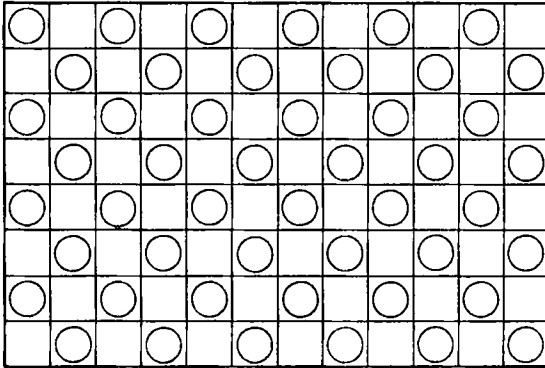
6. Slanting meshes



7. Meshes



8. Intense half tone

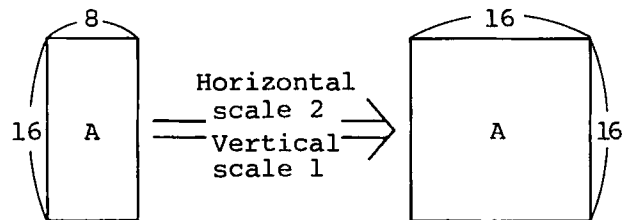


(18) Specifying the character size and angle

Function code CL: 17

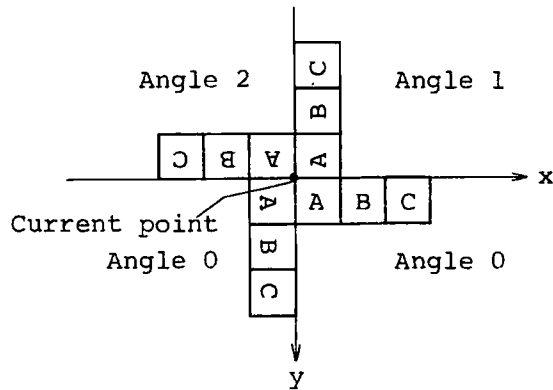
Parameters AH: Vertical scale (1 to 16)
 AL: Horizontal scale (1 to 16)
 BL: Angle of rotation (0 to 3)

Function Specifies the size and rotation angle of each character to be displayed by function code 10. The size is indicated by vertical and horizontal scale factors based on the standard character box size (8x16 dots). The following shows an example.



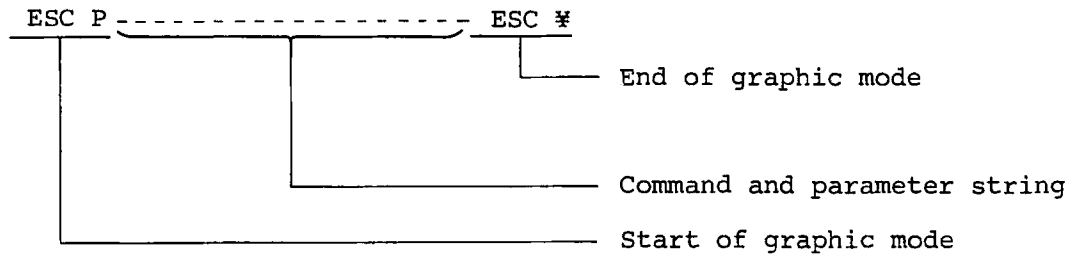
Standard box

The rotation angle indicates the direction of the character string in 90-degree units, based on the upper left corner (current point) of the character box. The following shows an example.



4.3 Escape Sequence Execution

After ANSISEQ2 command execution, escape sequences for graphic functions can be executed by BIOS CONOUT. Graphic escape sequences are specified in the following format:



ESC P (1B50H) places the CONOUT feature into graphic mode which remains effective until ESC ¥ (1B5CH) is encountered.

In graphic mode, CONOUT executes graphic functions according to the specified command and parameter string.

When CONOUT enters graphic mode due to the ESC P specification, the following items are set to the indicated initial values:

Line type	Solid line
Paint pattern	Entire painting
Character size	1 (not expanded)
Rotation angle	0 (not rotated)

The command and parameter string consists of one-letter command codes each of which may be followed by one or more parameters. A command is executed when the next command or ESC $\%$ is encountered. Command codes are written in uppercase letters; if written in lowercase, they are converted to uppercase. Undefined command codes are ignored. The following 15 characters can be used in parameters:

0 1 2 3 4 5 6 7 8 9 , " ' + -

Parameters indicated by any other characters are ignored. The table below lists the graphic functions that can be executed by escape sequences.

No.	Command	Parameters	Function
1	M	x, y	Moves the current point.
2	D	None	Draws a dot.
3	L	x, y	Draws a line.
4	R	x, y[,p]	Draws a rectangle.
5	C	r[,p]	Draws a circle.
6	E	rx, ry, θ [,p]	Draws an ellipse.
7	F	r, θ_1 , θ_2 [,p]	Draws a fan shape.
8	A	r, θ_1 , θ_2	Draws an arc.
9	S	k	Draws a symbol.
10	Q	None	Paints an enclosed area.
11	P	k	Paints a contiguous area.
12	T	"character string" or 'character string'	Displays a character.
13	W	k ₁ , k ₂	Specifies colors or attributes.
14	X	Pk	Specifies the line type.
15	Y	Pk	Specifies the paint pattern.
16	Z	Pk ₁ , Pk ₂ , Pk ₃	Specifies the character size and angle.

x, y	x and y coordinate values	p	Paint specification
r	Radius	Pk ₁	Vertical scale
rx	Horizontal radius	Pk ₂	Horizontal scale
ry	Vertical radius	Pk ₃	Angle of rotation
θ	Rotation angle in degrees	k	Pattern specification
θ_1	Starting angle in degrees	k ₁ , k ₂	Color/attribute specification
θ_2	Internal angle in degrees	Pk	Pattern specification

Note that escape sequence execution cannot handle reading from V-RAM or writing to V-RAM.

The parameters for the commands have the same meanings as the parameters for interrupts. See Section 4.1 for details.

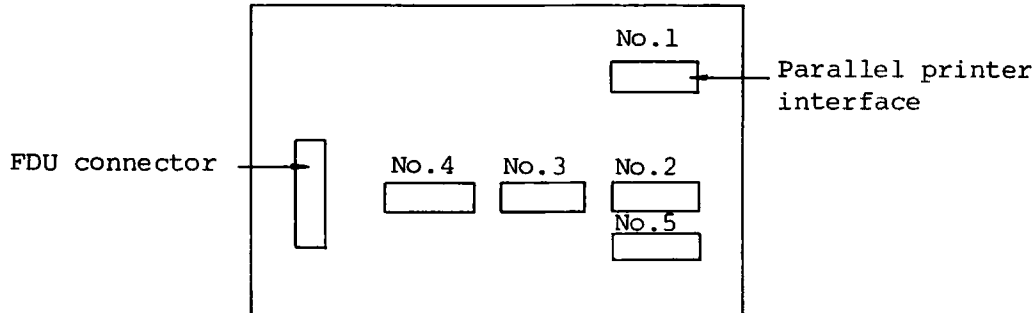
CHAPTER 5

RS232C INTERFACE

5.1 Outline

The AS-100 computer is provided with five I/O slots as shown in the figure below. Of these, slots 2 to 5 can be used to install RS232C interface cards.

Bottom of the AS-100 CPU board



Physical device names assigned to the RS232C interface I/O ports are as shown in the table below.

Port No.	Physical device
2	COM2
3	COM3
4	COM4
5	COM1

MS-DOS BIOS for the AS-100 is equipped with a handler which controls the RS232C interfaces according to the following specifications.

- AUTO XON/XOFF function is not provided. (See the description of the RSHND command.)
- Physical device: COM2
- Baud rate: 1200 bps
- Bit configuration: 8S (8 bits, no parity check bit, and one stop bit)

When using the RS232C interface with the above specifications, I/O operations can be performed by issuing function call INT 21H after physical device COM2 has been assigned to logical device AUX: with the ASSIGN command.

When the RS232C interface is used with other specifications, an I/O handler is required to control the interface. AS-100 MS-DOS provides the RSHND command for installing the RS232C handler in BIOS.

5.2 RSHND Command

Type: External command

Format: RSHND[/AUTOX]_┘<physical device>_┘<baud rate>_┘
<bit configuration>

Function: Installs the handler which controls RS232C interfaces according to specified parameters.

Remarks:

/AUTOX

Specifies that the AUTO XON/XOFF function is to be used. This function stops data transmission temporarily when an XOFF code (13H) is received from the output device, then restarts transmission when an XON code (11H) is received.

<physical device>

Specify the physical device name of the slot in which the RS232C interface is installed. The following physical device names can be specified.

COM1, COM2, COM3, COM4

<baud rate>

Specify the baud rate for data transmission as one of the following values.

110, 150, 300, 600, 1200, 2400, 4800, and 9600

<bit configuration>

Specify the bit configuration for data transmission as one of the following values.

7ESS, 7OSS, 7ES, 7OS, 8SS, 8ES, 8OS, and 8S

E: Even parity O: Odd parity

S: One stop bit SS: Two stop bits

Example: RSHND COM2 4800 7ES

If all parameters are omitted, the following specifications are assumed.

- AUTO XON/OFF function disabled.
- Physical device: COM2
- Baud rate: 1200 bps
- Bit configuration: 8S

When the RSHND command is executed, the following message is displayed on the screen if the handler for the port has been already installed in BIOS.

This port is already initialized, over write? (Y/N)

If Y is entered, the handler is reloaded into BIOS from the system diskette. Note that this increases the size of BIOS because the handler previously loaded is left in memory.

<Error messages>

Invalid param

An invalid parameter was specified.

Too long param

The parameter is too long.

There is no empty slot for IRQ routine

No space is available in the vector table for installing the handler in BIOS.

Sorry, I don't have this option.

The RS232C interface is not installed in the slot with the specified device name.

5.3 Execution

The RS232C interface is activated by one of the following function calls:

- (1) INT 14H
- (2) INT 21H

(1) Function call INT 14H

The RS232C interface can be activated by INT 14H.

Before issuing this call, load register AH with the function code. Since function call INT 14H specifies the port address directly, it is not necessary to assign the logical device name AUX: to the physical device.

Since function call INT 14H can not use the handler that is installed in BIOS, be sure to install a handler by means of the RSHND command to enable data transmission.

Function code	Parameter	Function
0	DX: Port No. AL: Initialization parameter	Initializes the port specified in register DX according to the condition specified in register AL.
1	DX: Port No. AL: Output data code	Outputs the data in register AL to the port specified in register DX.
2	DX: Port No. AL: Input data code	Inputs data from the port specified in register DX and stores it in register AL. When no data is input, enters the wait state.
3	DX: Port No. AX: Port status	Stores the bit pattern in register AX which indicates the status of the port specified in register DX.

Port initialization

Function code AH: 0

Parameter DX: Port No. to be initialized

0, 4 --- Port No. 5 (COM1)

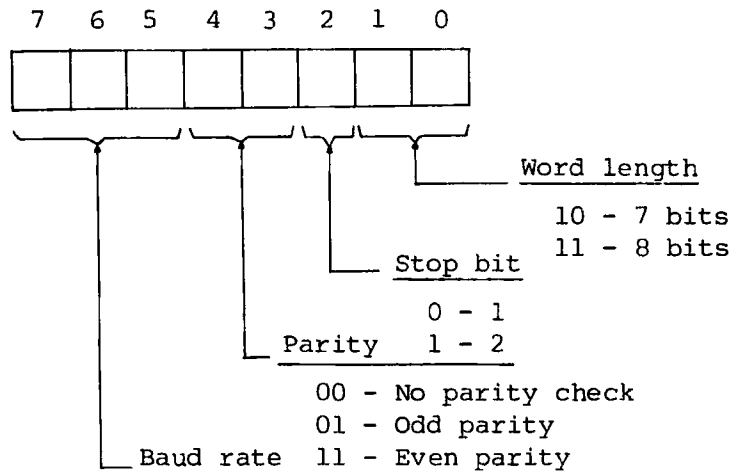
1, 5 --- Port No. 2 (COM2)

2, 6 --- Port No. 3 (COM3)

3, 7 --- Port No. 4 (COM4)

When a number from 4 to 7 is stored in register DX, the XON/XOFF function is enabled for the corresponding port. This can be specified only when the port is initialized.

AL: The bit pattern indicating the initial status



000 - 110 Baud

001 - 150

010 - 300

011 - 600

100 - 1200

101 - 2400

110 - 4800

111 - 9600

AX: The status of the complete execution
(See function code AH=3.)

Function: Initializes the port specified in register DX according to the initial status specified in register AL.

Data output

Function code: AH = 1

Parameter DX: Port No. to which data is to be output

0 - Port No. 5 (COM1)

1 - Port No. 2 (COM2)

2 - Port No. 3 (COM3)

3 - Port No. 4 (COM4)

AL: Character code to be output

AH: Result status
(See function code AH=3.)

Function: Outputs the character code specified in register AL to the port specified in register DX. The status is stored in register AH upon completion.

Data input

Function code: AH = 2

Parameter DX: Port No. from which data is to be input

0 - Port No. 5 (COM1)

1 - Port No. 2 (COM2)

2 - Port No. 3 (COM3)

3 - Port No. 4 (COM4)

AL: Input character code

AH: Result status
(See function code AH=3.)

Function: Stores the character code input from the port specified in register DX, in register AL. The status is stored in register AH upon completion.

Port status check

Function code: AH = 3

Parameter DX: Port No.

0 - Port No. 5 (COM1)

1 - Port No. 2 (COM2)

2 - Port No. 3 (COM3)

3 - Port No. 4 (COM4)

AH: Line control status

Bit 7 - timeout

Bits 6 to 0 - not used

AL: Modem status

Bit 7 - received data available

Bit 6 - not used

Bit 5 - data set ready

Bit 4 - clear to send

Bits 3 to 0 - not used

Function: Stores the status of the port, specified in register DX, in registers AH and AL.

Each bit set to 1 indicates the corresponding state.

(2) Function call INT 21H

The RS232C interface can be activated by INT21H. Before issuing this call, load register AH with the function code. It is necessary to assign logical device AUX: to the physical device in which the RS232C interface used by ASSIGN command is installed.

Receiving a character

AH: 3H

AC: Character received

When a character is read from the RS232C interface port installed in the physical device assigned to logical device AUX:, that character is stored in register AL. When no character is read from the port, the wait state is entered.

Sending a character

AH: 4H

AL: Character to be sent

Outputs the character in register AL to the port whose physical device address is assigned to logical device AUX.

CHAPTER 6

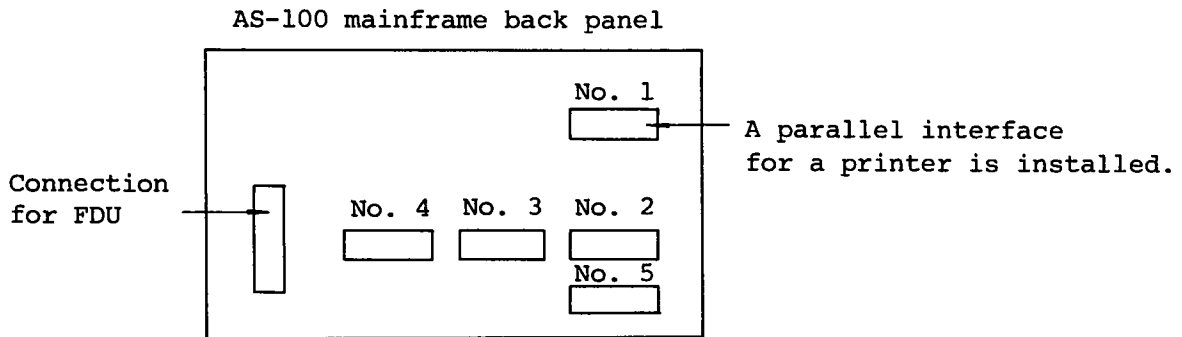
PRINTER INTERFACE

6.1 Overview

In the standard AS-100 series system configuration, a parallel data interface for a printer is installed in the AS-100 mainframe. This means that any printer matching this interface, conforming to the Centronix specifications, can be connected to the AS-100 series system.

This parallel interface is provided at port #1 as shown in the figure below. It is assigned physical device name LPT1 and, in initial status, is associated with logical device PRN:.

In addition, AS-100 MS-DOS BIOS is equipped with a handler with the standard specifications. It enables the printer connected to port #1 to be used through logical device PRN: without any special operation.



The user may install additional printer interfaces in ports #2 to #5. These ports are assigned the following physical device names:

Port #	Physical device
2	LPT2
3	LPT3
4	LPT4
5	LPT5

The printer connected to any of these ports can be accessed by issuing an ASSIGN command (see Chapter 3) to assign logical device name PRN: or AUX: to the physical device name of the port.

To use such an additional printer, however, the user must implement the interface handler in BIOS. In Addition, another special handler is required for producing hardcopy of screen data. AS-100 MS-DOS provides the following commands for printer handler implementation and execution:

Command name	Printer model	Additional features
A1200	A-1200	Screen hardcopy
A1210	A-1210	Screen hardcopy on color printer
CNTHND	Printers conforming to the standard Centronix specifications	None

6.2 A1200

Type: External command

Format: A1200[/HCOPY] [<physical device>]

Function: This command installs a handler for the A-1200 wire dot printer in BIOS.

Description:

The A1200 command with the /HCOPY parameter specifies that a handler with the screen hardcopy capabilities is to be installed in BIOS. Before issuing an A1200 command with /HCOPY, the user must execute the GRHND command explained in Chapter 3.

In an A1200 command is issued without the /HCOPY parameter, the following confirmation message is displayed:

```
Graphic system not ready OK? (y/n)
```

If "Y" is entered in response, a handler without the hardcopy capabilities is loaded into BIOS. If "N" is entered in response, A1200 command execution is terminated.

The <physical device> parameter indicates the physical device name assigned to the port to be used. The default physical device name is LPT1.

Example: A1200/HCOPY LPT1

After an A-1200 printer handler is implemented in BIOS, one-byte data is output under the following conditions:

Character output

(1) Control codes (00-1FH)

Only the following control codes are output:

```
LF (0AH), FF (0CH), CR (0DH), DC1 (11H), DC3 (13H),  
CAN (18H), ESC (1BH), US (1FH)
```

(2) Alphanumeric characters (20-7FH)

All codes except DEL (7FH) are output.

(3) Special codes (E0-EFH)

All codes are output.

(4) Other codes (80-DFH, F0-FFH)

Not output.

The handler implemented by this command can execute escape sequences, each of which is a character string headed by the ESC code (1BH), as explained below. This escape sequence handling feature is provided by the handler itself. For an explanation on the escape sequence handling feature of the printer, refer to the A-1200 printer manual.

Escape sequence feature

(1) Large character mode -- ESC l (1B31H)

This makes the large character mode effective. All following characters are expanded when displayed. On the screen hardcopy, however, the characters are printed in standard size.

(2) Normal mode -- ESCc (1B63H)

This cancels the large character mode and returns to the normal mode.

(3) Screen hardcopy -- ESC#7 (1B2337H)

This outputs the currently displayed data to the printer. Note that, after printing a screen hardcopy, the printer is placed in 6-line-per-inch (6LPI) mode.

(4) Partial screen hardcopy -- ESC#8n₁n₂
(1B2338nn₁nn₂H)

This outputs the contents of the specified series of lines on the screen to the printer. Parameter n₁ indicates the first line in the series, and n₂ indicates the number of lines to be output. nn₁ and nn₂ are hexadecimal indications of parameters n₁ and n₂, respectively.

The values of parameters n₁ and n₂ are the actual line number or line count plus 20H.

$$n_1 = L_1 + 20H \quad (1 \leq L_1 \leq 25)$$

$$n_2 = L_2 + 20H \quad (1 \leq L_2 \leq 25)$$

If an invalid value is specified for n₁ or n₂, the following default value is assumed:

$$\text{For } n_1: \quad 25 + 20H$$

$$\text{For } n_2: \quad (25 - X) + 20H$$

Example: The following specifies that the contents of lines 10 to 14 are to be printed out:

```
ESC#8 10 + 20H    5 + 20H
      ↓
      1B23382A25H
```

The printer is in 6LPI mode after printing the partial screen hardcopy, which is similar to the entire hardcopy operation described in (3) above.

6.3 A1210

Type: External command

Format: A1210[/HCOPY] [<physical device>]

Function: This command installs a handler for the A-1210 color inkjet printer into BIOS.

Description:

The meanings of the /HCOPY and <physical device> parameters are the same as those for the A1200 command explained in Section 6.2 above.

After an A-1210 printer handler is implemented in BIOS, one-byte data is output under the following conditions:

Character output

(1) Control codes (00-1FH)

Only the following control codes are output:

LF (0AH), VT (0BH), FF (0CH), CR (0DH), DC1 (11H),
DC3 (13H), CAN (18H), ESC (1BH), US (1FH)

(2) Alphanumeric characters (20-7FH)

All codes except DEL (7FH) are output.

(3) Special codes (E0-EFH)

All codes are output.

(4) Other codes (80-DFH, F0-FFH)

Not output.

The handler implemented by this command can execute escape sequences, each of which is a character string headed by the ESC code (1BH), as explained below. This escape sequence handling feature is provided by the handler itself. For an explanation on the escape sequence handling feature of the printer, refer to the A-1210 printer manual.

Escape sequence feature

(1) Large character mode -- ESC1 (1B31H)

This makes the large character mode effective. All following characters are expanded when displayed. On the screen hardcopy, however, the characters are printed in standard size.

(2) Normal mode -- ESCc (1B63H)

This cancels the large character mode and returns to the normal mode.

(3) Screen hardcopy -- ESC#7 (1B2337H)

This outputs the currently displayed screen contents to the printer.

On the hardcopy of a color monitor, the colors closest to the actually displayed colors are selected. This means that data appearing in different colors on screen may be printed in the same color on the hardcopy. Since the A-1210 can handle all the eight basic colors (black, white, magenta, green, yellow, cyan, red, and blue), the user can obtain a hardcopy in the correct colors by using only basic colors on screen.

On the hardcopy of a monochrome monitor, different colors are used to indicate different attributes as shown below.

Attribute	Color
No display	No print
Standard intensity	White
High intensity	Magenta
Standard intensity blink	Cyan
High intensity blink	Red

The line feed pitch of the A-1210 printer does not change after printing a screen hardcopy.

(4) Partial screen hardcopy -- ESC#8n₁n₂

(1B2338nn₁nn₂H)

This outputs the contents of the specified series of lines on the screen to the printer. Parameter n₁ indicates the first line in the series, and n₂ indicates the number of lines to be output. nn₁ and nn₂ are hexadecimal indications of parameters n₁ and n₂, respectively.

See (4) in Section 6.2 for complete explanations on parameters n₁ and n₂.

6.4 CNTHND

Type: External command

Format: CNTHND [<physical device>]

Function: This command installs, in BIOS, a handler for a printer with the parallel interface.

Description:

The <physical device> parameter specifies the physical device name assigned to the port to be used. The default is LPT1.

This handler does not have special additional features provided. It outputs all one-byte codes to the printer as is.

6.5 Messages

The following message appears at normal end of a command for implementing a printer handler in BIOS.

```
XXXXX HANDLER Vn.nn
```

where XXXXX is the handler name and n.nn is the version number of the handler.

If the command ends abnormally, one of the following error messages is output to indicate the cause of the failure:

```
&& Illegal parameter
```

```
An invalid physical device name was specified.
```

```
&& Undefine switch
```

```
/HCOPY was specified in a command that does not allow  
/HCOPY.
```

```
&& This handler has not hardcopy function
```

```
/HCOPY was specified for a handler that does not have  
the hardcopy function.
```

```
&& Too many parameter
```

```
The command contained too many parameters.
```

```
&& There is no physical port n
```

```
The AS-100 mainframe does not include the port  
corresponding to the specified physical device name.  
"n" in the message indicates the port number.
```

n=1: LPT1
n=2: LPT2
n=3: LPT3
n=4: LPT4

&& The same driver is already attached

The specified handler was already installed in BIOS.

&& Hardcopy function already attached

An attempt was made to install a handler with the /HCOPY specification, but the hardcopy function was already being used for another handler.

&& Irq entry full

The IRQ table used for handler installation had no entries available.

&& Graphic system not ready

The subsystem for hardcopy output had not been loaded.

XXXXX is already attached

Overwrite? (y/n)

A handler using the specified physical device was already installed in BIOS.

XXXXX indicates the name of the already installed handler.

If "Y" is entered in response to this message, the specified new handler replaces the currently installed handler indicated by XXXXX. If "N" is entered in response, the command is terminated.

6.6 Execution

After installing a printer handler in BIOS, the user can access the connected printer by issuing an ASSIGN command (see Chapter 3) to assign a logical device name to the physical device name of the port.

Programs written in the Assembler language can access the printer via the handler, by storing a function code and parameters in registers and generating interrupt INT 17H.

Initialization

Function code AH: 1

Parameters DX: Port to be initialized
 0 for port #1 (LPT1)
 1 for port #2 (LPT2)
 2 for port #3 (LPT3)
 3 for port #4 (LPT4)

Function Loads register DX with a port number and initializes the port. Execution result status is given in register AH. See the explanation of AH=2 for details.

One-character output

Function code AH: 0

Parameters AL: Character code to be output
 DX: Port number (0 to 3)

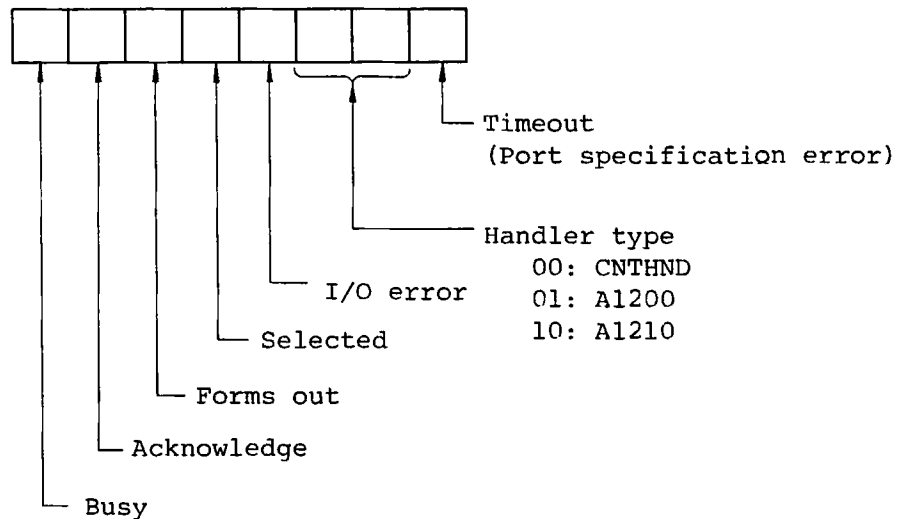
Function Outputs the character code stored in register AL to the port indicated by register DX. Execution result status is given in register AH. See the explanation of AH=2 for details.

Status check

Function code AH: 2

Parameter DX: Port number (0 to 3)

Function Checks status of the port indicated by register DX, and loads register AH with status information in the following format:



CHAPTER 7

EXTENDED INTERRUPTS

7.1 Overview

MS-DOS reserves interrupts types 20H to 3FH for the system.

The following interrupt codes (hex) are defined for the standard MS-DOS:

INT	Function
20	Program terminate
21	Function request
22	Terminate address
23	$\overline{\text{CTR}}$ / $\overline{\text{LC}}$ exit address
24	Fatal error abort vector
25	Absolute disk read
26	Absolute disk write
27	Program termination, with the program remaining resident

Refer to the Standard MS-DOS manual for explanations on these interrupts.

In addition to these standard interrupts, the extended interrupts listed below are provided for AS-100 MS-DOS for more effective utilization of the AS-100 series capabilities.

INT	Function
5	Screen hardcopy output
10	Screen control
11	Checking connected devices
12	Checking memory size
13	Floppy disk I/O control
14	RS232C I/O control
15	Graphic functions
16	Keyboard control
17	Printer control
19	Bootstrap loader
1B	Keyboard break
1C	Timer interrupt
1D	BIOS parameter (Key code table)
1E	BIOS parameter (Font addresses for codes 00H-7FH)
1F	BIOS parameter (Font addresses for codes 80H-FFH)

This chapter describes these extended interrupts, except INT 14H, INT 15H, AND INT 17H which have already been discussed in previous chapters as follows:

INT 14H: Chapter 5

INT 15H: Chapter 4

INT 17H: Chapter 6

7.2 Interrupts

(1) INT 5H -- Screen hardcopy output

This interrupt is used to output the currently displayed screen contents to the printer assigned physical device name LPT1.

This interrupt handling routine can be called by pressing

CTRL / SHIFT / CLEAR
SCREEN .

If the printer is not ready when the interrupt is generated, the routine waits until the printer becomes ready.

Note that the interrupt is ignored if the printer with physical device name LPT1 is not connected or the printer is out of forms.

(2) INT 10H -- Screen control

This interrupt is used for any special screen control operation done in the AS-100 MS-DOS system.

The interrupt is processed by setting a function code in register AH and storing other necessary information in other registers. The following describes the meanings of the function codes.

(AH)=0 --- Setting the screen mode
Determines the screen mode according to the value stored in register AL.

(AL)=0:	40 characters/line, 25 lines, monochrome
1:	40 characters/line, 25 lines, color
2:	80 characters/line, 25 lines, monochrome
3:	80 characters/line, 25 lines, color
4:	40 characters/line, 25 lines, monochrome (G)
5:	40 characters/line, 25 lines, color (G)
6:	80 characters/line, 25 lines, monochrome (G)
7:	80 characters/line, 25 lines, color (G)
8:	40 characters/line, 20 lines, monochrome
9:	40 characters/line, 20 lines, color
10:	80 characters/line, 20 lines, monochrome
11:	80 characters/line, 20 lines, color
12:	40 characters/line, 20 lines, monochrome
13:	40 characters/line, 20 lines, color (G)
14:	80 characters/line, 20 lines, monochrome (G)
15:	80 characters/line, 20 lines, color (G)

(G) represents the graphic mode.

In the AS-100 series, since V-RAM is used to control screen data in bit units, all of the 16 modes listed above can be used for graphic handling. If (G) is specified, however, the character cursor does not appear on screen in the initial conditions.

The color and monochrome modes are discriminated by the hardware. For example, if (AL)=3 is specified on a monochrome model, (AL)=2 is assumed. The initial value is (AL)=2 for monochrome models and (AL)=3 for color models.

(AH)=1 --- Specifying whether to display the character cursor
Specifies whether or not to display the character cursor on screen, depending on bit 5 of register CH as follows:

(CH) bit 5=1: The character cursor is displayed.
 5=0: The character cursor is not displayed.
(CL): Dummy

(AH)=2 --- Specifying the character cursor position

Moves the character cursor to the position indicated by registers DH and DL. If additional character buffers have been installed by the PGADD command explained in Chapter 3, register BH must be loaded with a page number representing the page in which the cursor is to be set.

(DH)=Line number (0 to 24)
(DL)=Column number (0 to 79)
(BH)=Page number (0 to 3 in 80-character mode,
 0 to 7 in 40-character mode)

(AH)=3 --- Obtaining the current character cursor position

Finds the location of the cursor in the page (character buffer) indicated by register BH, and records it in registers DH and DL. Character cursor mode information is also given in register CH.

(BH)=Page number
(DH)=Set to the line number of the cursor position
(DL)=Set to the column number of the cursor position
(CH)=Set to the character cursor type (see the explanation of AH=1)
(CL)=Dummy

(AH)=4 --- Obtaining the current pointing device position

Obtains the current pointing device status in register AH, and obtains the pointing device location in the following registers:

In character cursor mode: Registers DH and DL
In graphic cursor mode: Registers BX and CX

(AH) = 0

Indicates that the pointing device was closed and the character cursor mode was made active before this interrupt was executed, or that function key A was not pressed. In this case, the current pointing device position is given in the following registers:

(DH) = Set to the line number of the character cursor position (0 to 24)

(DL) = Set to the column number of the character cursor position (0 to 79)

(AH) = 1

Indicates that the pointing device was opened and the graphic cursor mode was made active before this interrupt was executed, or that function key A was pressed. In this case, the position at which the pointing device was located when function key A was pressed is given in the following registers:

(BX) = Set to the X coordinate value of the graphic cursor position (0 to 639)

(CX) = Set to the Y coordinate value of the graphic cursor position (0 to 399)

(AH) = 5 --- Specifying the page to be displayed

Specifies that the contents of the character buffer corresponding to the page number stored in register AL are to be displayed. The buffer contents then become ready for output.

(AL) = Page number in the following range:

(0 to 3 in 80-character mode)
(0 to 7 in 40-character mode)

(AH) 6 --- Specifying upward scrolling

Scrolls the screen upward, according to the values stored in registers AL, CH, CL, DH, DL, and BH.

(AL) = Number of lines to be scrolled. This is identical to the number of blank lines to be entered at the bottom of the scroll area. If (AL) = 0 is specified, the entire screen is used as the scroll area.

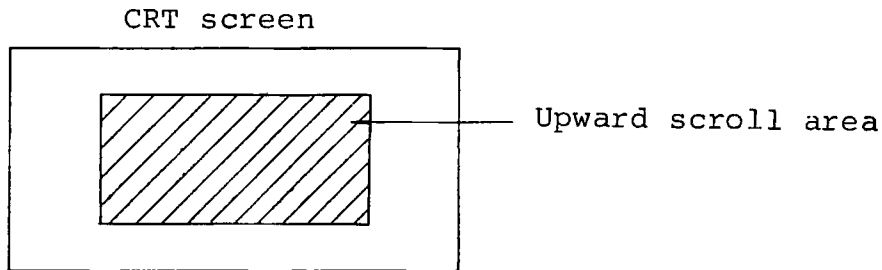
(CH) = Line number of the upper left corner of the scroll area

(CL) = Column number of the upper left corner of the scroll area

(DH) = Line number of the lower right corner of the scroll area

(DL) = Column number of the lower right corner of the scroll area

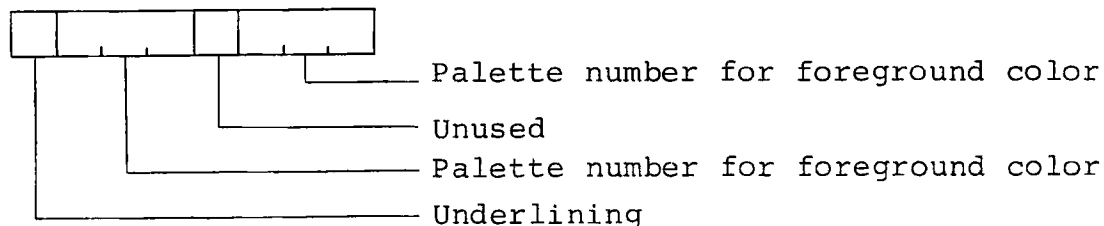
(DH) = Line number of the lower right corner of the scroll area
 (DL) = Column number of the lower right corner of the scroll area



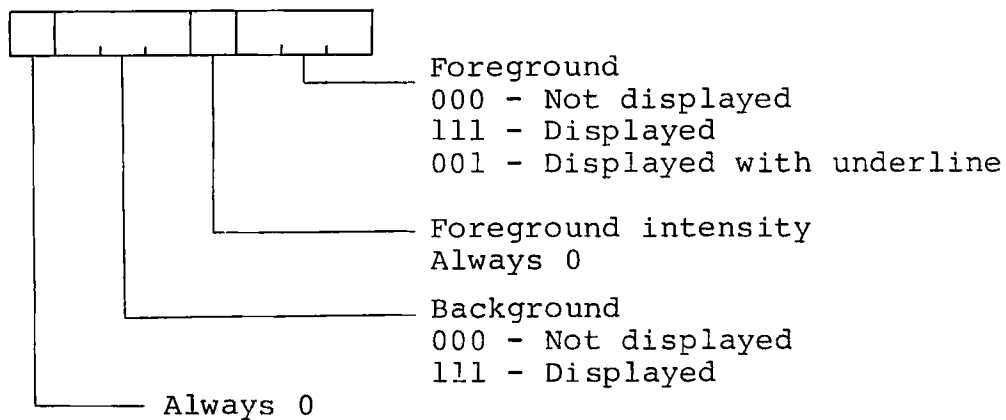
If the DH and DL values represent a position outside the effective screen area, the maximum values are assumed.

(BH) = Attributes of the blank lines to appear at the bottom of the scroll area during scroll-up operation. These attributes are indicated by a bit pattern as shown below.

(Color model)



(Monochrome model)



(AH)=7 --- Specifying downward scrolling

Scrolls the screen downward according to the values stored in registers AL, CH, CL, DH, DL, and BH.

The meanings of these register values are identical to those for (AH)=6 explained above.

(AH)=8 --- Obtaining the character code and attributes of the current cursor position

Loads registers AL and AH with the character code and attributes of the current cursor position in the page (character buffer) indicated by register BH.

(BH) = Page number of the page (character buffer) currently displayed. The page number must be in the following range:

(0 to 3 in 80-character mode,
 0 to 7 in 40-character mode)

(AL) = Set to the character code displayed in the cursor position

(AH) = Set to the attributes of the cursor position. The attributes are indicated by a bit pattern explained in (AH)=6.

(AH)=9 --- Outputting character codes with specified attributes to the current cursor position

Displays a character code the specified number of times with the specified attributes, starting from the current cursor position in the page (character buffer) indicated by register BH.

(BH) = Page number of the page (character buffer) currently displayed. The page number must be in the following range:

(0 to 3 in 80-character mode,
 0 to 7 in 40-character mode)

(CX) = Number of times the character is to be output

(BX) = Character code to be output

(BL) = Attributes for character display. The attributes are indicated by a bit pattern explained in (AH)=6.

(AH)=10 --- Outputting character codes to the current cursor position

Displays a character code the specified number of times, starting from the current cursor position in the page (character buffer) indicated by register BH.

(BH) = Page number of the page (character buffer) currently displayed. The page number must be in the following range:

(0 to 3 in 80-character mode)
 (0 to 7 in 40-character mode)

(CX) = Number of times the character is to be output

(AL) = Character code to be output

(AH) = 11 --- Assigning a color to a palette

Assigns a color to a palette register.

(BH) = Color number representing the color to be assigned to the palette (0 to 28)

(BL) = Palette number (0 to 7)

The following table shows the colors indicated by color numbers in register BH:

Value							Remarks
	r	R	g	G	b	B	
0	0	0	0	0	0	0	Black (no display)
1	0	0	0	0	0	1	Blue
2	0	0	0	0	1	1	
3	0	0	0	1	0	0	Green
4	0	0	0	1	0	1	Cyan
5	0	0	0	1	1	1	
6	0	0	1	1	0	0	
7	0	0	1	1	0	1	
8	0	0	1	1	1	1	
9	0	1	0	0	0	0	Red
10	0	1	0	0	0	1	Yellow
11	0	1	0	0	1	1	
12	0	1	0	1	0	0	Magenta
13	0	1	0	1	0	1	White
14	0	1	0	1	1	1	
15	0	1	1	1	0	0	
16	0	1	1	1	0	1	
17	0	1	1	1	1	1	
18	1	1	0	0	0	0	
19	1	1	0	0	0	1	
20	1	1	0	0	1	1	
21	1	1	0	1	0	0	
22	1	1	0	1	0	1	
23	1	1	0	1	1	1	
24	1	1	1	1	0	0	
25	1	1	1	1	0	1	
26	1	1	1	1	1	1	
27. 28							

R: Red r: Red in half intensity
 G: Green g: Green in half intensity
 B: Blue b: Blue in half intensity

(AH)=12 --- Displaying a dot at the specified position

Displays a dot at the specified coordinate position with the color or attribute indicated by the specified palette.

(CX)=X coordinate value (0 to 639)

(DX)=Y coordinate value (0 to 399)

(AL)=Palette number corresponding to the color or attribute to be used

(AH)=13 --- Finding the palette number assigned to the dot at the specified position

Loads register AL with the palette number assigned to the dot at the specified coordinate position.

(CX)=X coordinate value (0 to 639)

(DX)=Y coordinate value (0 to 399)

(AL)=Set to the palette number assigned to the dot at the specified position

(AH)=14 --- Displaying a character

Displays a character on screen according to the character code and attributes indicated by registers AL and BL. This interrupt handling routine is used by the BIOS CONOUT function. If any of the four codes listed below is specified for output, the corresponding function is executed instead of displaying a character:

0AH (LF)

0DH (CR)

07H (BELL)

08H (BS)

(AL)=Character code to be output

(BL)=Attributes (see the explanation of AH=6)

(AH)=15 --- Obtaining screen mode information

Obtains the current screen mode information in registers as follows:

(AL)=Set to the screen mode value (see the explanation of AH=0)

(BL)=Set to the number of columns on screen

(RH)=Set to the number representing the page (character buffer) currently displayed. The page number is in the following range:

(0 to 3 in 80-character mode)

(0 to 7 in 40-character mode)

(AH)=16 --- Outputting specified attributes to the current cursor position

Outputs the specified attributes in the specified number of character positions, starting from the current cursor position in the page (character buffer) indicated by register BH.

(BH)=Page number of the page (character buffer) currently displayed.

(CX)=Number of character positions in which the attributes are to be output

(AL)=Attributes to be output. See the explanation on (AH)=6 for details.

(AH)=17 --- Pointing device control

Controls the pointing device according to the values stored in registers AL, BX, and CX.

(AL)=0: Closes the pointing device and activates the character cursor mode.

1: Opens the pointing device and activates the graphic cursor mode.

2: Displays the graphic cursor (+).

3: Deletes the graphic cursor from the screen.

4: Moves the graphic cursor the specified number of coordinate positions. Registers BX and CX indicate the number of positions as follows:

(BX)=Number of X coordinate positions

(CX)=Number of Y coordinate positions

5: Moves the cursor to the specified coordinate position. Registers BX and CX indicate the position as follows:

(BX)=X coordinate value (0 to 639)

(CX)=Y coordinate value (0 to 399)

(AH)=18 --- Determining the attributes used for screen control

Determines the attributes to be used for screen control, from the values stored in registers AL, BL, and BH.

(AL)=0: Determines the attributes to be used for screen display by CONOUT.

(BL)=Attributes (see the explanation on AH=6)

(AL)=1: Determines the screen scroll mode.

(BL)=0 for smooth scroll mode

1 for line scroll mode

- (AL)=2:Determines the character cursor attributes.
(BL)=Attributes (see the explanation on AH=6)
- (AL)=3: Determines the graphic cursor attributes.
(BL)=Attributes (see the explanation on AH=6)
- (AL)=4: Determines the bank select register value
(BL)=Value to be set in the bank select register
- (AL)=5:Determines the attributes for underlines.
(BL)=Attributes (see the explanation on AH=6)
- (AL)=6:Copies the currently displayed page (character buffer) contents to another page.
(BL)=Page number indicating the destination page (character buffer). The page number must be in the following range.
 (0 to 3 in 80-character mode,
 (0 to 7 in 40-character mode)
- (AL)=7: Generates a sound.
(BH)=Sound tone code (0 to 60)
(BL)=Sound length code (0 to 255)
The tables below show the meanings of the sound tone and length codes.

Tone codes and scales

Scale	Octave				
	0	1	2	3	4
C	1	13	25	37	49
C*	2	14	26	38	50
D	3	15	27	39	51
D*	4	16	28	40	52
E	5	17	29	41	53
F	6	18	30	42	54
F*	7	19	31	43	55
G	8	20	32	44	56
G*	9	21	33	45	57
A	10	22	34	46	58
A"	11	23	35	47	59
B	12	24	36	48	60

Example: Tone code 22 (A1) --- 440 Hz
If 0 is specified for the tone code, no sound is output.

Sound length codes

Code	Length in seconds
1	0.016
10	0.16
50	0.8
100	1.6
150	2.4
200	3.2
255	4.08

(AL)=8: Specifies the data select register attributes.
The data select register determines the foreground attributes.
(BL)=Attributes (see the explanation on AH=6)

(AL)=9: Loads register AL with the tone code value stored in the music buffer.

(AH)=19 --- Outputting a character display

Reads a character string from the memory address indicated by registers DS and DX, and displays it on screen from the current cursor position.

(DS:DX)=These two registers indicate the memory address at which the character string to be displayed is defined.

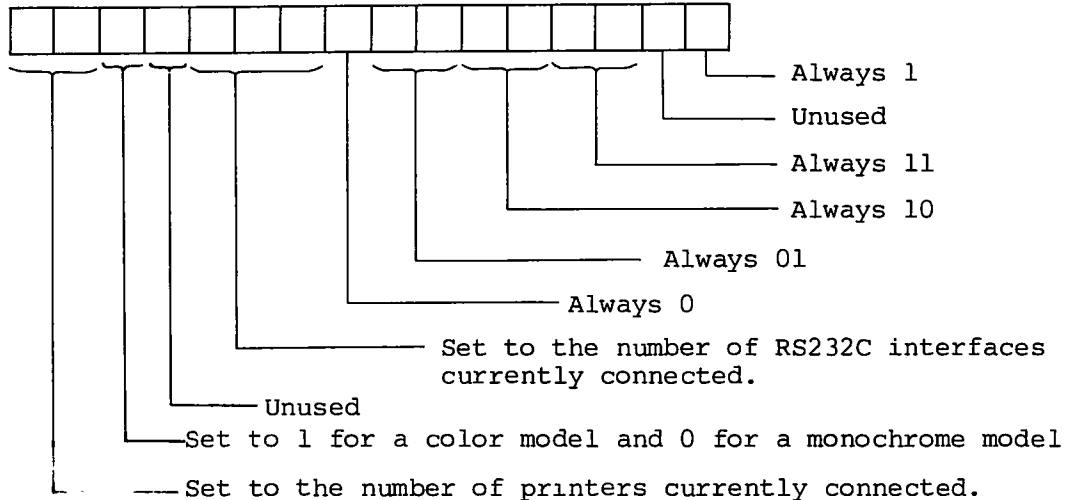
(DS: Segment address
DX: Offset address)

(BL)=Attributes (see the explanation on AH=6)

(CX)=Number of characters to be output

(3) INT 11H -- Checking the connected devices

This interrupt is used to check the devices currently connected to the AS-100 mainframe, and load register AX with a bit pattern indicating device status.



(4) INT 12H -- Checking the memory size

This interrupt is used to check the system memory size and load register AX with the size. The memory size here refers to the entire memory size excluding the ROM area.

(AX)=Set to the number of 1K-byte blocks in memory.

(5) INT 13H -- Floppy disk I/O control

This interrupt is used to control the floppy disk drives. It is processed after loading register AH with a function code and other registers with parameters required for processing.

(AH)=0 --- Resets the disk system and initializes FDC software.

(AH)=1 --- Sets register AL to the sector number of the last sector processed.

The following explains the parameters that must be set in registers for reading, writing, checking, or formatting disks:

(DL) = Drive number

0 (4) -- Logical device A (E)
1 (5) -- Logical device B (F)
2 (6) -- Logical device C (G)
3 (7) -- Logical device D (H)

Drive numbers 0 to 3 are used to process the following types of disks:

5-inch disks -- Double-sided, double-density,
double-track, 512 bytes/sector
8-inch disks -- Double-sided, double-density,
1024 bytes/sector

Drive numbers 4 to 7 are used to process the following types of disks:

5-inch disks -- Single-sided, double-density,
512 bytes/sector
8-inch disks -- Single-sided, single-density,
128 bytes/sector

(DH) = Head number

(CH) = Track number

(CL) = Sector number

(AL) = Number of sectors

(ES:BX) = Buffer address

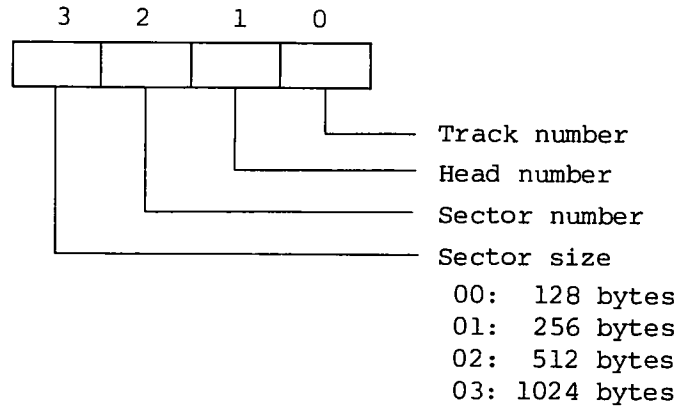
(AH) = 2 --- Transfers the contents of the specified sector to memory.

(AH) = 3 --- Transfers the memory contents to the specified sector.

(AH) = 4 --- Checks the specified sector contents.

(AH) = 5 --- Formats the specified track.

For the disk formatting procedure, the buffer pointer (ES:BX) must be set to the next address field to be used. An address field is a four-byte area containing the following information:



One address field is provided for each sector. It is accessed to find the next sector to be read or written.

After interrupt INT 13H is processed, registers AH and AL are set to the processing result status as follows:

(AH)=Set to termination status. If the Carry flag (CY) is 0, disk I/O processing ended normally. If the flag is 1, an error occurred during processing.

(AL)=Set to the sector number of the last sector processed during read, write, or check processing. The AL value does not change if an error occurred during processing.

(6) INT 16H -- Keyboard control

This interrupt is used to control the keyboard. It is processed after a function code is stored in register AH.

(AH)=0 --- Loads register AL with the next code entered from keyboard.

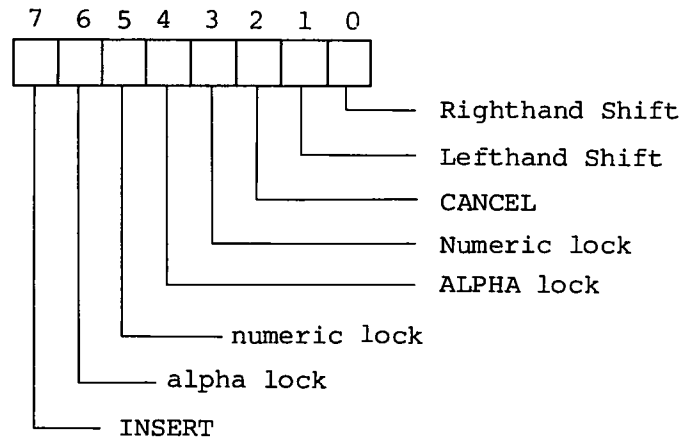
(AL)=Set to the ASCII code (0 for an extended code)
(AH)=Set to the scan code

(AH)=1 --- Checks whether the entered ASCII character can be read, and sets ZF to the result.

(ZF)=1: Cannot be read.
0: Can be read.

If the character can be read (ZF=0), the next character stored in the keyboard buffer is recorded in register AX. The characters remain in the keyboard buffer even after this interrupt is processed.

(AH)=2 --- Loads register AL with a bit pattern indicating the current shift status as shown below.



(7) INT 19H -- Bootstrap loader

This interrupt is used to jump to the reset address in ROM monitor.

(8) INT 1BH -- Keyboard break

This interrupt handling routine is called by pressing CTRL / CANCEL. It clears the keyboard buffer and suspends the current operation. This routine can also be called from a program.

(9) INT 1CH -- Timer interrupt

This interrupt handling routine is called by a timer interrupt once every second. At present, however, a dummy routine is used.

(10) INT 1DH -- BIOS parameter (Key code table)

This vector indicates the key code table. Its initial value is 0:0. When the vector value is not 0:0, if a numeric or character key code is generated from BIOS, the corresponding code in the key code table is given in return.

The key code table consists of 48 entries (30H-5FH), each consisting of five bytes as follows:

- Byte 1: Code generated while the CTRL key is held down.
- Byte 2: Basic code
- Byte 3: Basic code generated while the SHIFT key is held down.
- Byte 4: Code generated while the ALPHA-LOCK key is on.
- Byte 5: Code generated while the ALPHA-LOCK key is on and the SHIFT key is held down

Note: If a null code (00H) is defined in the table entry, BIOS does not return any code. This means that the key code table cannot be used to generate a null code.

(11) INT 1EH -- BIOS parameter (Font addresses for codes 00H to 7FH)

This vector indicates the address of a font corresponding to a code from 00H to 7FH.

Each font consists of 16 bytes (8×16 dots). The font of a character can be changed by setting this vector to another font area or modifying the contents of the area indicated by this vector.

(12) INT 1FH -- BIOS parameter (Font addresses for codes 80H to FFH)

This vector indicates the address of a font corresponding to a code from 80H to FFH.

APPENDIX A

AS-100 MS-DOS SYSTEM PROGRAMS

The AS-100 MS-DOS system disk contains the following system programs:

- ASMSDOS. SYS
- ASIO. SYS
- COMMAND. COM
- CHKDSK. COM
- CREF. EXE
- DEBUG. COM
- EDLIN. COM
- EXE2BIN. COM
- FILCOM. COM
- LIB. EXE
- LINK. EXE
- MASM. EXE

These are standard MS-DOS programs and are explained in the Standard MS-DOS manual. ASMSDOS.SYS and ASIO.SYS correspond to MSDOS.SYS and IO.SYS in standard MS-DOS, respectively. In standard MS-DOS, the SYS.COM program is provided for copying these two programs. In AS-100 MS-DOS, the copying function is included in the FORMAT command. Therefore, AS-100 MS-DOS does not provide SYS.COM as an independent system program.

In addition to the programs listed above, AS-100 MS-DOS supplies the system programs given below. These additional programs are explained in this manual.

- A1200.COM
- A1210.COM
- ANSISEQ1.COM
- ANSISEQ2.COM
- ASSIGN.COM
- CNTHND.COM
- CPDIR.EXE
- CPCOPY.EXE
- DISKCOPY.COM
- FORMAT.COM
- GRHND.COM
- PAGE.COM
- PALETTE.COM
- PGADD.COM
- RSHND.COM

APPENDIX B
CHARACTER CODE TABLE

APPENDIX C
ROM DEBUGGER

The ROM debugger is supplied in boot ROM. It can handle the vector tables in system memory area, except the single-step, NMI, and breakpoint tables. It also controls the address space at addresses 400H to FFFFFH.

There are two ways to activate the ROM debugger as follows:

- 1 Turn on the mainframe power without setting the system disk in a floppy disk drive. The debugger start message is displayed and the debugger waits for command entry.
- 2 While the system is running, press the STOP switch located to the lower left of the screen. Symbol @ and the CS and IP contents are displayed, and the debugger waits for command entry.

An asterisk (*) appears for debugger prompt. Any of the commands listed below can be entered in response to the prompt.

Command	Meaning	Format
B	Boot	B[D]<CR>
D	Display memory contents	D[W]<ADDR>[,<ADDR2>]<CR>
G	Execute	G<CS: IP><OLD>[<ADDR>[,<BRK>]]<CR>
I	Input data from a port	I[W]<PORT><CR/,>
N	Execute steps	N<CS:IP><OLD>[<ADDR>]<CR/,>
O	Output data to a port	O[W]<PORT>,<NEW><CR/,>
R	Read from Intel-Hexa format file	R[<BIAS>]<CR>
S	Modify data in memory	S[W]<ADDR>,<OLD>[<NEW>]<CR/,>
X	Display or modify register values	X<CR> X<REG><OLD>[<NEW>]<CR/,>
T	Change the debugger console	T<CR>

[] indicates an optional parameter.

— indicates debugger output.

< > indicates an acronym.

Acronyms

- W Indicates that the command is to handle data in word units. If this parameter is omitted, data is handled in byte units.
- <ADDR> Specifies an address, which is represented by a segment address and an offset address. Each of the two addresses is either a hexadecimal number with four or fewer digits or indicated by a register name (see the explanation on REG below). If the segment address is omitted, the CS value is assumed.
- Example: SP:1234
- <BIAS> Specifies the bias for program loading, which is either a hexadecimal number with four or fewer digits or indicated by a register name.
- <ERK> Specifies a breakpoint in the same way as <ADDR>.
- <CR> Indicates Carriage Return (CR) key entry.
- <CR/,> Indicates entry of the Carriage Return (CR) key or a comma. If CR is input, the debugger waits for another command entry. If a comma is input, the debugger executes the same command for the next address. The next address here refers to the following:
- ° For byte access: Current address + 1 (except for the I and O commands)
 - ° For word access: Current address + 2
 - ° For step execution: Current address + Instruction length in bytes
- <CS:IP> Indicates a message issued by the debugger. The message contains a code segment and the instruction pointer value, each in a hexadecimal four-digit number.
- <NEW> Specifies a hexadecimal two- or four-digit number which is to be newly stored in a register or memory.
- <OLD> Indicates a hexadecimal two- or four-digit number currently stored in a register or memory.
- <PORT> Specifies a hexadecimal number with four or fewer digits representing a port address.

<REG> Specifies one of the following two-letter codes representing a register:

AX	Accumulator
BX	Base
CX	Count
DX	Data
SP	Stack Pointer
BP	Base Pointer
SI	Source Index
DI	Destination Index
CS	Code Segment
DS	Data Segment
SS	Stack Segment
ES	Extra Segment
IP	Instruction Pointer
FL	Flag

Commands

B --- Boot

B[D]<CR>

Carries out initial boot operation from drive A. If parameter D is specified, the debugger receives control after initial boot.

D --- Display memory contents

D[W]<ADDR>[,<ADDR2>]<CR>

Displays the contents of the memory area, from <ADDR> to <ADDR2>, in word or byte units. If parameter <ADDR2> is omitted, only the value at address ADDR is displayed. If any key is pressed during the display operation, the operation ends and the debugger is ready to accept the next command.

G --- Execute

G<CS:IP><OLD>[<ADDR>[,<BRK>]]<CR>

When G is entered, the debugger displays the CS and IP values of the next instruction to be executed, and also displays one digit of the value of the instruction address.

If execution must begin with that address, press the <CR> key. If it must start with another address, enter the desired address for <ADDR> and press <CR>. A breakpoint may be specified for <BRK> after <ADDR>.

When the breakpoint is reached, the following message appears and the debugger is ready to accept the next command.

BR@<CS:IP>

I --- Input from port

I[W]<PORT><CR/,>

If the <CR> key or a comma is entered with a port address, the contents of the specified port are displayed. If only a comma is entered, the contents of the same port are displayed. Commas are therefore used to find how the port contents have changed.

N --- Execute steps

N<CS:IP><OLD>[<ADDR>] <CR/,>

When N is entered, the debugger displays the CS and IP values of the next instruction to be executed, and two digits of the value of the instruction address. Specify parameter <ADDR> to start execution from another address. Enter a comma to carry out step execution.

O --- Output to port

O[W]<PORT>,<NEW><CR/,>

Outputs data to the specified port. If a comma is entered, a hyphen appears and the user can enter different data for the same port.

R --- Input from Intel-Hexa file

R[<BIAS>] <CR>

Loads an Intel-Hexa format file from the RS232C interface in I/O port COM2 into memory. If parameter <BIAS> is specified, the load address is assumed to be the record load address plus the BIAS value.

The RS232C initial values are 2400 baud and 8S. If they have been modified by the handler, the new values are used for file loading.

If any key is pressed while the debugger is waiting for data entry, the debugger switches to the command entry wait state.

S --- Modify memory contents

S[W]<ADDR>,<OLD>[<NEW>] <CR/,>

Displays the value of the memory address indicated by parameter <ADDR>. If the value need not be modified, enter <CR> or a comma only. If modification is needed, enter a new value for parameter <NEW> and enter <CR> or a comma. Read-after-write check is done during modification.

X --- Display or modify register values

X<CR>

X<REG><OLD>[<NEW>] <CR/,>

If X<CR> is entered, the debugger displays the names and contents of all registers. If a specific register is specified for parameter <REG>, the debugger displays the contents of that register.

If the register value need be modified, enter a new value for parameter <NEW> and enter <CR> or a comma. If a comma is used, the contents of the next register are displayed. The register order is listed in the explanation of <REG> given under the heading "Acronyms" in this appendix.

T --- Change debugger console

T<CR>

Switches the debugger console from the AS-100 mainframe to the RS232C interface in I/O port COM2, or vice versa. In response to the T<CR> command, the debugger issues message "ARE YOU SURE?" on screen. Console switching is done only if "Y" is entered in response to this message.

Calculation feature

When issuing a command, the user may specify addition or subtraction in parameters <ADDR>, <BRK>, <CS:IP>, <NEW>, and <PORT> using symbol + or -.

Examples:

D AX+100-BX<CR>

S DS:BX-100<, >

Error handling

Upon detecting an error in a command or the operand, the debugger displays symbol #, feeds a line, and waits for next command entry. If a specified numeric value exceeds the maximum length, the debugger ignores the first digits but does not generate an error.

Only alphanumeric keys can be entered from the keyboard for a command. If the user notices that wrong characters have been input, there is no way to correct the command even before the end-of-command key is entered. In this case, the user must enter an invalid key (such as @) intentionally to cause an error, and reenter the command from the beginning. The debugger also displays # if an error occurs after command execution.

Remarks

If the STOP switch is used to activate the debugger, the screen data may appear in an unexpected form. This is not an error; it occurs because the operating system and the ROM debugger control the screen independently. The screen format can be corrected by entering ESC/Z to reset the screen. To get around this problem, use the T command to specify RS232C for the debugger console.

APPENDIX D
ESCAPE SEQUENCE

The ANSISEQ1 or ANSISEQ2 command with the <ON> parameter specification enables escape sequence execution. This appendix explains the escape sequences that can be executed in the AS-100 MS-DOS system.

See Chapter 4 for explanations on graphic escape sequences.

(1) Index

Format: $E_{S_C D}$

Function: Same as Line Feed (LF)

(2) Line feed

Format: $E_{S_C E}$

Function: Same as Carriage Return (CR) plus Line Feed (LF)

(3) Reverse index

Format: $E_{S_C M}$

Function: Moves the cursor one line upward. If this causes the cursor to move to the top line in the scroll area, downward scrolling occurs in the scroll area.

(4) Initialization

Format: $E_{S_C C}$

Function: Initializes the screen conditions by resetting all modes, clearing the screen, and moving the cursor to the home position.

The following shows the initial values of the modes:

Mode	Initial value
Character cursor display toggle	ON
Graphic cursor display toggle	OFF
Number of lines on screen	25
Scroll mode	Smooth scroll
Scroll area	Entire screen
Character box size	8×16 dots (25-line mode)
Character cursor position	(1, 1)
Graphic cursor position	(320, 200)
Graphic current point position	(0, 0)
Pointing device	Character cursor mode

The palette registers are set to the initial values which depend on the CRT display model as shown below.

Palette number	Color model	Monochrome two-plane model	Monochrome one-plane model
0	0: Black (no display)	0: Black (no display)	0: Black (no display)
1	1: Blue	2: High intensity	1: Standard intensity
2	3: Green	27: Standard intensity	/
3	4: Cyan	blink	
4	9: Red	1: Standard intensity	
5	10: Magenta		
6	12: Yellow		
7	13: White		

In the initial conditions, the following palettes are used for character, cursor, and underline display:

	Color model	Monochrome two-plane model	Monochrome one-plane model
Character display	7: White	3: Standard intensity	1: Standard intensity
Character cursor	6: Yellow	2: Standard blink	1: Standard intensity
Graphic cursor	6: Yellow	2: Standard blink	1: Standard intensity
Underline	7: White	3: Standard intensity	1: Standard intensity

(5) Moving the cursor up

Format: $E_{SC} \{ PnA$

Function: Moves the cursor upward the number of lines indicated by Pn. If Pn is omitted or 0 is specified for Pn, 1 is assumed for Pn. The cursor does not move if it is already located on the top line.

(6) Moving the cursor down

Format: $E_{SC} \{ PnB$

Function: Moves the cursor downward the number of lines indicated by Pn. If Pn is omitted or 0 is specified for Pn, 1 is assumed for Pn. The cursor does not move if it is already located on the bottom line.

(7) Moving the cursor to the right

Format: $E_{SC} \{ PnC$

Function: Moves the cursor to the right the number of columns indicated by Pn. If Pn is omitted or 0 is specified by Pn, 1 is assumed for Pn. If the current cursor position is the last column (column 80) of a line, the cursor moves to the first column of the next line. If the current position is the last column of the bottom line, the cursor moves to the first column of that line and no scrolling occurs.

(8) Moving the cursor to the left

Format: $E_{SC} \{ PnD$

Function: Moves the cursor to the left the number of columns indicated by Pn. If Pn is omitted or 0 is specified for Pn, 1 is assumed for Pn. If the current cursor position is the first column of a line, the cursor moves to the last column of the preceding line. If the current position is the first column of the top line, the screen is not scrolled down.

(9) Moving the cursor to the specified position

Format: $E_{S_C} (P_l; P_c H$ or $E_{S_C} (P_l; P_c f$

Function: Moves the cursor to the position specified by P_l (line number) and P_c (column number). If the P_l or P_c value is outside the screen range, the nearest value in the range is assumed.

Examples:

$E_{S_C} (0; 90H \rightarrow E_{S_C} (1; 80H$

$E_{S_C} (H \rightarrow E_{S_C} (1; 1H$

(10) Clearing columns to the right of the cursor

Format: $E_{S_C} (OK$ or $E_{S_C} (K$

Function: Clears the character positions from the current cursor position to the last column (column 80) of that line. The cursor does not move.

(11) Clearing columns to the left of the cursor

Format: $E_{S_C} (lK$

Function: Clears the character positions on the cursor line, from the first column to the current cursor position. The cursor moves to the first column of that line.

(12) Clearing the cursor line

Format: $E_{S_C} (2K$

Function: Clears entirely the line on which the cursor is currently located. The cursor moves the first column of that line.

(13) Clearing the cursor position through the last scroll line

Format: $E_{S_C} (0J$ or $E_{S_C} (J$

Function: Clears the current cursor position through the bottom line of the scroll area. The cursor does not move.

Note: If a partial scroll area is defined, the area to be cleared depends on the current cursor position. See the following example:

A	
B	Partial scroll area
C	

In this case, if the cursor is located within the scroll area (B), the current cursor position through the bottom line in B are cleared. If the cursor is within area A, the cursor position through the bottom line in A are cleared.

(14) Clearing the first scroll line through the cursor position

Format: $E_{SC} \{ 1J$

Function: Clears the top line in the scroll area through the current cursor position. The cursor moves to the head of the scroll area. The "Note" for $E_{SC} \{ 0J$ given above also applies to this escape sequence.

(15) Clearing the scroll area

Format: $E_{SC} \{ 2J$

Function: Clears the scroll area entirely. The cursor moves to the head of the scroll area. The "Note" for $E_{SC} \{ 0J$ given above also applies to this escape sequence.

(16) Canceling the character display attribute

Format: $E_{SC} \{ 0m$

Function: Cancels the underlining and reverse display attributes previously specified for character display.

(17) Specifying the high intensity attribute

Format: $E_{SC} \{ 1m$

Function: Displays subsequent characters in high intensity. That is, this escape sequence changes the foreground palette number to 1. This is only effective for monochrome two-plane models.

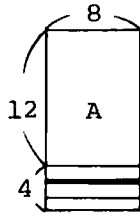
(18) Specifying the underlining attribute

Format: $E_{SC} \{ 4m$

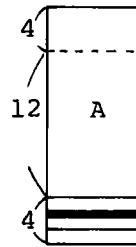
Function: Displays subsequence characters with underlines.

25-line mode

20-line mode



ANK character box



ANK character box

(19) Specifying the blink attribute

Format: $E_{S_C} 5m$

Function: Displays subsequent characters with the blink attribute. That is, this escape sequence changes the foreground palette number to 2. This is only effective for monochrome two-plane models.

(20) Specifying the reverse display attribute

Format: $E_{S_C} 7m$

Function: Displays subsequent characters by reversing the colors or attributes assigned to the foreground and background palettes.

(21) Specifying the foreground or background color

Format: $E_{S_C} Pnm$

Function: Specifies the foreground or background color by assigning a palette register to the foreground or background. This is only effective for color models.

Pn value		Palette	
Foreground	Background	Number	Default color
30	40	0	Black
31	41	4	Red
32	42	2	Green
33	43	6	Yellow
34	44	1	Blue
35	45	5	Magenta
36	46	3	Cyan
37	47	7	White

(Remarks on m-type sequences)

Two or more m-type escape sequences
(E_{S_C} (0m to E_{S_C} (47m) may be specified in the same
sequence format.

Example: E_{S_C} (4;3l;4lm

The table below shows which of the attribute
values 0 to 47

Attribute value	Function	Color model	Monochrome one-plane mode	Monochrome two-plane model
0	Canceling the attributes	o	o	o
1	High intensity attribute		o	
4	Underlining attribute	o	o	o
5	Blink attribute		o	
7	Reverse display attribute	o	o	o
30 - 37, 40 - 47	Foreground/background color	o		

(22) Specifying the scroll area

Format: E_{S_C} (Pf;Ptr

Function: Defines the scroll area. Pf and Pt indicate the top
and last lines in the scroll area, respectively.
These two values must be in the following range:

In 25-line mode: $1 \leq Pf \leq Pt \leq 25$

In 20-line mode: $1 \leq Pf \leq Pt \leq 20$

If the Pf or Pt value is outside the range, the nearest
value in the range is assumed.

Example:

E_{S_C} (0;28r → E_{S_C} (1;25r

If not the entire screen is specified to be the scroll
area, the partial scroll mode is active and a special
scrolling method is used. The processing speed is
faster in partial scroll mode than in smooth or line
scroll mode.

(23) Displaying the character cursor

Format: E_S{ >0h
C

Function: Causes the character cursor to be on; i.e.,
displays the character cursor on screen.

(24) Displaying the graphic cursor

Format: E_S{ >1h
C

Function: Causes the graphic cursor to be on; i.e.,
displays the graphic cursor on screen.

(25) Specifying the 25-line mode

Format: $E_{S_C} [>2h$

Function: Places the screen into 25-line mode, clears the entire screen, specifies the entire screen to be the scroll area, and moves the cursor to the home position.

(26) Specifying the smooth scroll mode

Format: $E_{S_C} [>3h$

Function: Makes the smooth scroll mode effective. If partial scrolling is active, however, the smooth scroll mode does not become effective immediately because a special method is used for partial scrolling. It becomes effective after the entire screen is specified for the scroll area.

(27) Specifying the pointing device graphic cursor mode

Format: $E_{S_C} (>4h$

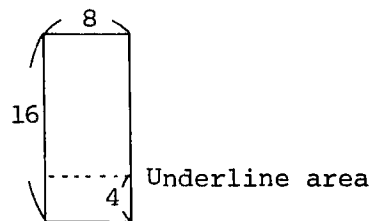
Function: Places the pointing device into the mode in which it can be used to move the graphic cursor, and causes the graphic cursor to be on.

(28) Specifying the character box size in 16/20 mode

Format: $E_{S_C} (>5h$

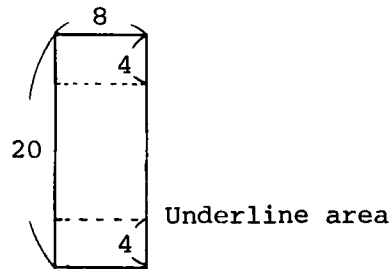
Function: Changes the height of each character box according to the line mode as shown below.

25-line mode



ANK character box

20-line mode



ANK character box

In this mode, the lower four-dot underline/cursor area is also taken into account when displaying a character. This means that, if a character is displayed in a column where an underline is currently displayed, the underline disappears.

(Remarks on h-type sequences)

Two or more h-type escape sequences
($E_{S_C} \{ >0h$ to $E_{S_C} \{ >0h$ to $E_{S_C} \{ >5h$)

can be included in the same escape sequence format.

Example: $E_{S_C} \{ > 0;1;4h$

(29) Deleting the character cursor

Format: $E_{S_C} \{ > 0l$

Function: Causes the character cursor to be off (to disappear from the screen). Screen processing is faster when the character cursor is off.

(30) Deleting the graphic cursor

Format: $E_{S_C} \{ > 1l$

Function: Causes the graphic cursor to be off (to disappear from the screen). Screen processing is faster when the graphic cursor is off.

(31) Specifying the 20-line mode

Format: $E_{S_C} \{ >2\ell$

Function: Places the screen into 20-line mode, clears the entire screen, specifies the entire screen to be the scroll area, and moves the cursor to the home position.

(32) Specifying the line scroll mode

Format: $E_{S_C} \{ 3\ell$

Function: Makes the line scroll mode effective. If partial scrolling is active, however, the line scroll mode does not become effective immediately because a special method is used for partial scrolling. It becomes effective after the entire screen is specified for the scroll area.

(33) Specifying the pointing device character cursor mode

Format: $E_{S_C} \{ >4\ell$

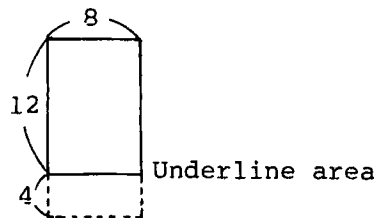
Function: Places the pointing device into the mode in which it can be used to move the character cursor, and causes the graphic cursor to be off.

(34) Specifying the character box size in 12/16 mode

Format: $E_{S_C} \{ >5\ell$

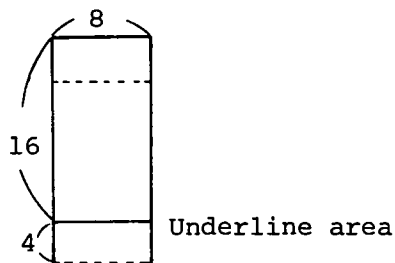
Function: Changes the height of each character box according to the line mode as shown below.

25-line mode



ANK character box

20-line mode



ANK character box

In this mode, the lower four-dot underline/cursor area is not taken into account when displaying a character. This means that, if a character is displayed in a column where an underline or ruler line is currently displayed, the underline or ruler line remains there. As an exception, if a pattern is displayed in the entire 8×16-dot character box, such as a semi-graphic character pattern, the lower four dots are not displayed in 25-line mode.

(Remarks on ℓ -type sequences)

Two or more ℓ -type escape sequences ($E_{S_C} \{ >0\ell$ to $E_{S_C} \{ >5\ell$) can be included in the same escape sequence format.

Example: $E_{S_C} \{ 1;2;3\ell$

(35) Finding the graphic cursor position

Format: $E_{S_C} \{ >6n$

Function: Requests for the current graphic cursor position. The response indicating the position is input from CONIN in the format shown below. The user must fetch the response from CONIN.

Response format: $E_{S_C} \{ >Px;PyR$

Px: X coordinate value
Py: Y coordinate value

(36) Finding the character cursor position

Format: $E_{SC} \{ 6n$

Function: Requests for the current character cursor position. The response indicating the position is input from CONIN in the format shown below. The user must fetch the response from CONIN.

Response format: $E_{SC} \{ P\ell;PcR$

$P\ell$: Line number

Pc : Column number

(37) Specifying the character cursor color/attribute

Format: $E_{SC} \{ >0;Pnc$

Function: Specifies the palette number corresponding to the color or attribute in which the character cursor is to be displayed. The allowable range of Pn depends on the model as follows:

Color model	$0 \leq Pn \leq 7$
Monochrome two-plane model	$0 \leq Pn \leq 3$
Monochrome one-plane model	$0 \leq Pn \leq 1$

(38) Specifying the graphic cursor color/attribute

Format: $E_{SC} \{ >1;Pnc$

Function: Specifies the palette number corresponding to the color or attribute in which the graphic cursor is to be displayed. See the table above for the allowable range of Pn .

APPENDIX E
DIP SWITCHES

Dip switches are provided on the AS-100 series CPU card, I/O card, and floppy disk unit interface card. The user can define the system operation conditions by setting these switches to ON or OFF.

The following explains the functions of the dip switches.

(1) CPU card

SW1	SW2	SW3	SW4
-----	-----	-----	-----

SW1: ON
SW2: OFF if the 8087 chip is installed; ON if not.
SW3: OFF
SW4: OFF if the clock option is installed; ON if not.

(2) I/O card

SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
-----	-----	-----	-----	-----	-----	-----	-----

SW1: ON if the ROM debugger console is provided through RS232C; OFF if it is provided on the AS-100 mainframe.
SW2: OFF
SW3: OFF
SW4: ON if the CRT screen mode is ASCII; OFF if it is JIS.
SW5: ON if the CRT model is monochrome one-plane; OFF if it is color or monochrome two-plane.
SW6: ON if the CRT model is monochrome; OFF if it is color.
SW7: OFF
SW8: OFF

(3) Minifloppy disk drive unit

SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
-----	-----	-----	-----	-----	-----	-----	-----

SW1: OFF
SW2: OFF
SW3: OFF
SW4: OFF

If logical device names A: and B: are assigned to the drives:

SW5: OFF
SW6: ON
SW7: OFF
SW8: ON

If logical device names C: and D: are assigned to the drives:

SW5: ON
SW6: OFF
SW7: ON
SW8: OFF

(4) Standard floppy disk drive unit

SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
-----	-----	-----	-----	-----	-----	-----	-----

SW1: OFF
SW2: OFF
SW3: OFF
SW4: ON

If logical device names A: and B: are assigned to the drives:

SW5: ON
SW6: OFF
SW7: ON
SW8: OFF

If logical device names C: and D: are assigned to the drives:

SW5: OFF
SW6: ON
SW7: OFF
SW8: ON

(

(

(