Chapter 1

BIOS Routine Operation

Chapter 1 Operation of the BIOS Subroutines

This Chapter describes operation of the various subroutines of the QX-10's basic input/output system.

§1 BOOT (Cold Boot), WBOOT (Warm Boot)

1.1 Loading BIOS

When the QX-10's power is turned on or its reset button is pressed, the initial program loader (IPL) reads a program called the boot loader into resident RAM (starting at address 0F000H), then control is passed to the boot loader.

The boot loader transfers itself to the area beginning at address 0E000H, starts execution from there and loads the console command processor (CCP) starting at address 0200H, the basic disk operating system (BDOS) starting at address 0E800H, BIOS1 starting at address 0F600H, and BIOS2, 3, 4, and 5 starting at address 0H of the system bank.

Next, the various code tables used by BIOS are loaded.

After loading has been completed, CCP is moved from the area beginning at 0200H to the area from 0E000H to 0E7FFH and control is passed to the BOOT entry.

A list of the BIOS entry points is shown on the next page.

BIOS ENTRY POINTS

ADDRESS	ENTRY NAME
0F600H	BOOT
0F603H	WBOOT
0F606H	CONST
0F609H	CONIN
0F60CH	CONOUT
0F60FH	LIST
0F612H	PUNCH
0F615H	READER
0F618H	HOME
0F61BH	SELDSK
0F61EH	SETTRK
0F621H	SETSEC
0F624H	SETDMA
0F627H	READ
0F62AH	WRITE
0F62DH	LISTST
0F630H	SECTRN
0F633H	PSET
0F636H	HCOPY
0F639H	BEEP
0F63CH	RSOPEN
0F63FH	RSCLOSE
0F642H	RSINST
0F645H	RSOUTST
0F648H	RSIN
0F64BH	RSOUT
0F64EH	TIMDAT
0F651H	MEMORY
0F654H	RSIOX
0F657H	LIGHTPEN
0F65AH	MASKI
0F65DH	LOADX
0F660H	STORX
0F663H	LDIRX
0F666H	JUMPX
0F669H	CALLX
0F66CH	GETPFK
0F66FH	PUTPFK

1.2 BOOT (Address: 0F600H)

1.2.1 General

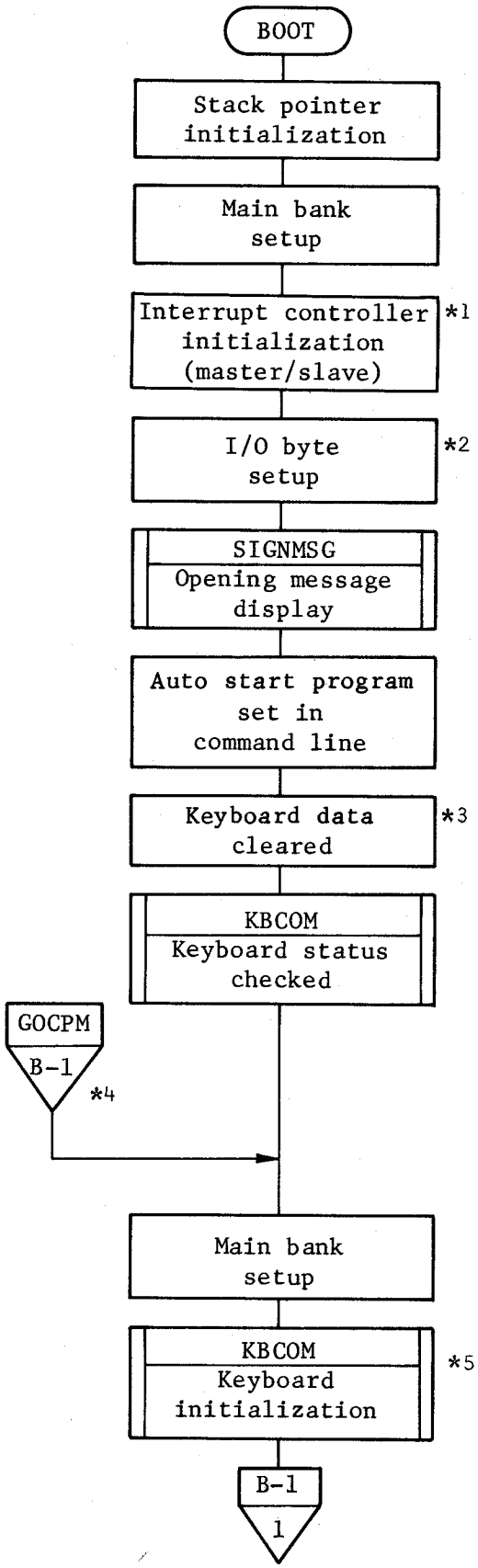
This is the routine to which control is passed by the boot loader for system loading. Therefore, this routine is never used except when a cold start is made. Cold start processing performed by this routine is as follows.

- o The interrupt controller (an NEC μPD8259) is initialized.
- o The I/O byte is set for the system area (main bank addresses 00H to 0FFH).
- o The CP/M opening message is displayed.
- o The auto start command (if any) is set.
- o The keyboard is initialized.†
- o The date and time are read.†
- o All tables are initialized.†
- o A check is made to determine whether a MultiFonts CG ROM card is installed.†
- o Jump vectors are set to BIOS and BDOS in the system area.†

After this processing has been completed, control is passed to CCP. If an auto start command has been set, CCP then executes it; if not, CCP goes to command standby (in other words, it goes into an idle loop).

Note 1: Daggers (†) indicate processing which is performed by the same routines as with WBOOT.

1.2.2 General flowchart



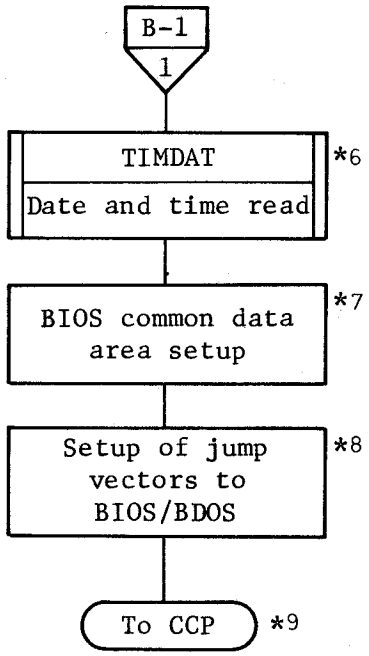
*1 Interrupt vector address (master): 0FD80H
Interrupt vector address (slave): 0FDA9H
All interrupts are masked except for the flexible disk controller, keyboard, RS-232C, and Power-down interrupts, interrupts from the calendar clock, and interrupts from the slave CPU.

*2 See Appendix B for details.

*3 Clears the buffer in the keyboard interface controller (NEC μPD7201).

*4 Following routines are the same as those used by WBOOT from GOCPM.

*5 Repeat start time and repeat interval setup.



*6 See the explanation of the TIMDAT routine starting on page 1-90.

*7 Table setup for subroutine GOCPM (BIOS2), international character pattern setup, MF CG ROM card check (flag set when MF CG ROM card is installed).

*8 BIOS warm boot address is 0F603H, and BDOS address is 0E806H.

*9 Control passed to CCP.

1.3 WBOOT (Address 0F603H)

1.3.1 General

The WBOOT routine reloads CCP and BDOS and performs the following processing.

- o Initializes the keyboard.
- o Reads the date and time.
- o Initializes all tables.
- o Makes a check to determine whether the MF CG ROM card is installed.
- o Sets jump vectors to BIOS and BDOS in the system area.

After this processing has been completed, control is passed to CCP. If a flexible disk read error occurs during reloading of CCP and BDOS (including when the system disk is not mounted), the following message is displayed and the buzzer sounds.

```
CANNOT WARM BOOT !!
```

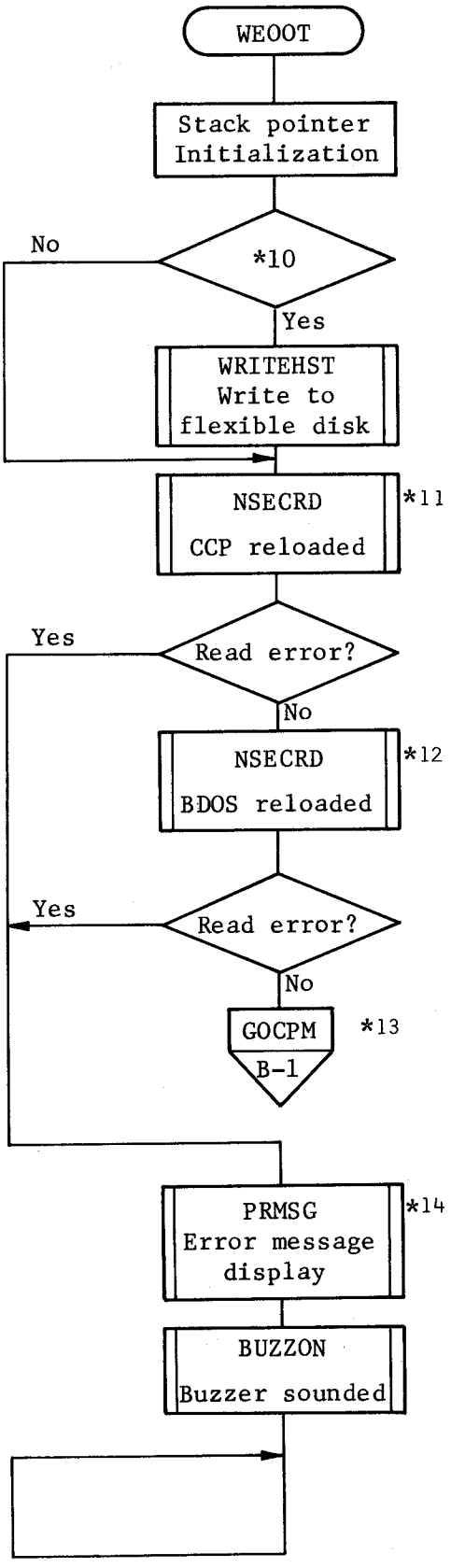
When this occurs, insert the CP/M system disk and press the reset button.

1.3.2 Call procedure

```
Example) JP      0
```

```
Example) WBOOT   EQU     0F603H
         JP      WBOOT
```

1.3.3 General flowchart



*10 Is there any data remaining to be written to a flexible disk ?

*11 Loaded into the area beginning at address 0E000H.

*12 Loaded into the area beginning at address 0E800H.

*13 Subsequent processing is the same as with GOCPM of BOOT.

*14 "CANNOT WARM BOOT" displayed.

§2 CONST (Console Status), CONIN (Console Input)

2.1 CONST (Address 0F606H)

2.1.1 General

This routine checks the status of the device currently assigned to "CON:" and returns the result in register A. If any data is pending input, 0FFH is set in register A; otherwise, 00H is set. The contents of all other registers are changed by execution of this routine.

The device assigned to "CON:" is determined by the setting of the I/O byte.

1.2 Call procedure

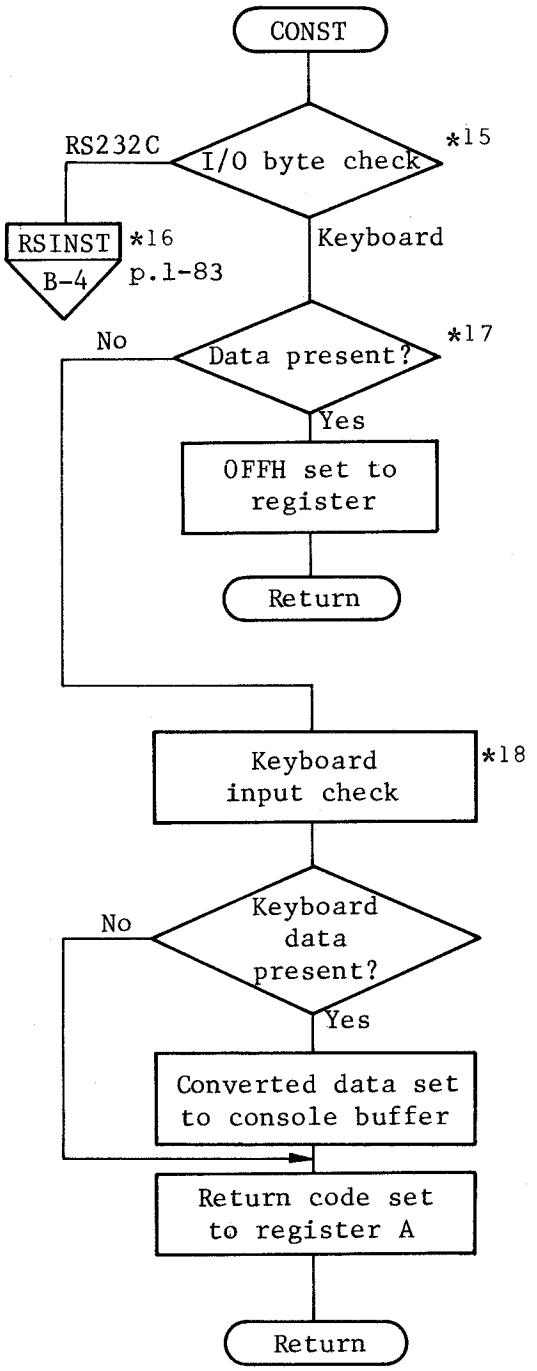
Entry parameters: None

Example:

```
            .
            .
CONST       EQU     0F606H
            .
            .
            CALL    CONST
            .
            .
```

Return information: Register A=0FFH Data present.
Register A=00H No data present.

2.1.3 General flowchart



*15 Checks the CON: logical device assignment in the I/O byte.

*16 Checks whether any data is being input from the RS-232C interface.

*17 Checks whether the the console buffer contains any character codes from converted keyboard input data.

*18 The CHRSTAT subroutine checks for keyboard data and converts any found to ASCII code, MF code, or a function key code.

2.2 CONIN (Address 0F609H)

2.2.1 General

This routine inputs one character from the device currently assigned to "CON:" and returns the corresponding character code in register A. The device assigned to "CON:" is determined by the setting of the I/O byte.

If the console is assigned to "CON:" and operation is in the function key check mode (when FUNCFLG at address 0FED1H in the BIOS common data area is set to 0FFH), 0FFH is returned in register C; this differentiates the PFKs (programmable function keys) and ten-keys from other keys.

When no data is present, this routine loops until data is input.

The contents of all registers are changed by execution of this routine.

2.2.2 Call procedure

Entry parameters: None

Example a) Normal mode (PFKs and ten-keys not checked)

```
CONIN    EQU    0F609H        ;CONSOLE INPUT.
FUNCFLG  EQU    0FED1H        ;FUNCTION KEY CHECK FLAG.
;
EXCONIN: JP     CONIN         ;EXECUTE CONSOLE INPUT.
         .
         .
         LD     A,(FUNCFLG)
         PUSH   AF            ;SAVE CHECK MODE.
         XOR    A             ;
         LD     (FUNCFLG),A   ;NON PFK CHECK MODE.
         CALL   EXCONIN       ;GO BIOS.
         LD     B,A           ;B= INPUT DATA.
         POP    AF            ;
         LD     (FUNCFLG),A   ;RECOVER CHECK FLAG.
         .
         .
```

Return information: Register A=Input data

Example b) Function key check mode

```
BIOSJMP  EQU    1             ;BIOS JUMP ADDRESS.
CONIN    EQU    2*3           ;CONSOLE INPUT
FUNCFLG  EQU    0FED1H        ;FUNCTION KEY CHECK FLAG.
;
EXBIOS:  LD     HL,(BIOSJMP);EXECUTE BIOS FUNCTION.
         ADD    HL,DE
         JP     (HL)
         .
```

```
.
LD      A,(FUNCFLG) ;
PUSH    AF          ;SAVE INPUT CHECK MODE.
LD      A,0FFH      ;FUNCTION KEY CHECK MODE.
LD      (FUNCFLG),A ;
LD      DE,CONIN    ;EXECUTE CONIN.
CALL    EXBIOS      ;
LD      B,A         ;B= INPUT DATA
POP     AF          ;
LD      (FUNCFLG),A ;RECOVER CHECK MODE.
.
.
```

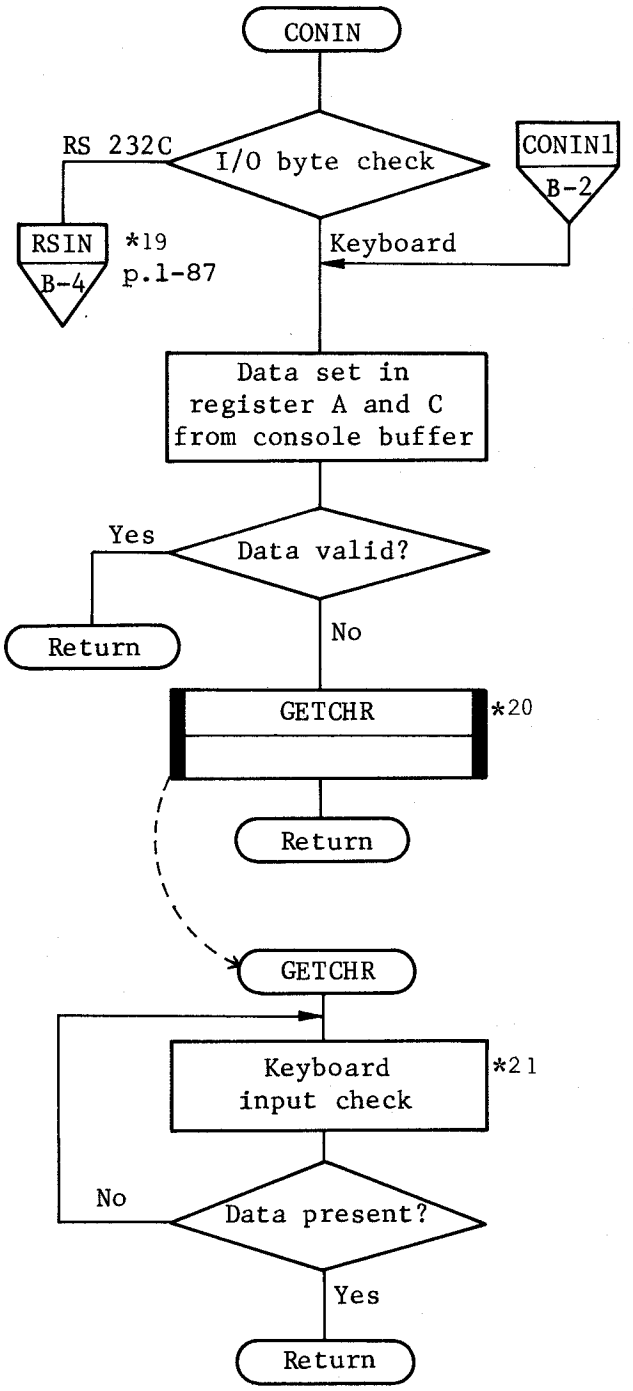
Note: Return parameters in the function key check mode are as follows.

(i) When register C=0, input is from other than the PFKs or ten keys. The character code (including MultiFonts code) is set in register A.

(ii) When register C=0FFH, input is from a PFK or a key on the ten-key pad.

Value set in register A	Key
0E0H	PFK1
0E1H	PFK2
0E2H	PFK3
0E3H	PFK4
0E4H	PFK5
0E5H	PFK6
0E6H	PFK7
0E7H	PFK8
0E8H	PFK9
0E9H	PFK10
0F5H	Ten-key 000 (for other than the German keyboard)
0F4H	Ten-key 00 (for the German keyboard only)
30H	Ten-key 0
31H	Ten-key 1
32H	Ten-key 2
33H	Ten-key 3
34H	Ten-key 4
35H	Ten-key 5
36H	Ten-key 6
37H	Ten-key 7
38H	Ten-key 8
39H	Ten-key 9
0DH	Ten-key ↵
2AH	Ten-key *
2BH	Ten-key +
2CH	Ten-key , (for other than the German keyboard)
5EH	Ten-key ^ (for the German keyboard only)
2DH	Ten-key -
2EH	Ten-key .
2FH	Ten-key /
3DH	Ten-key =

2.2.3 General flowchart



*19 Data input from the RS-232C interface.

*20 Waits until data is input from the keyboard.

*21 CHRSTAT subroutine; checks for input from the keyboard and converts any found into ASCII code or other character code.

§3 CONOUT (Console Output - Address: 0F60CH)

3.1 General

The CONOUT routine outputs the character code set in register C to the device currently assigned to "CON:". The contents of all registers are changed by execution of this routine.

The current "CON:" assignment is determined by the I/O byte. Devices which can be accessed as "CON:" include the display, RS-232C interface, and printer. The initial assignment (that effective when a cold start is made) is the display.
Display modes available include the alphanumeric mode, the Non-MFBASIC mode, and the MFBASIC mode. In order to display MultiFonts characters, display must be placed in the Non-MFBASIC MF mode or MFBASIC mode by executing the MFONT transient command, or by setting the MF mode flag (address 0FE82H) and MFBASIC mode flag (address 0FE50H) in the BIOS common area as follows.

Non-MFBASIC normal mode	0FE82H <-- 00H and 0FE50H <-- 00H
Non-MFBASIC MF mode	0FE82H <-- 0FFH and 0FE50H <--00H
MFBASIC mode	0FE50H <-- 0FFH and 0FE82H <--00H

Internal processing by BIOS in these three modes is as follows.

		Number of characters displayed (columns x lines)	Internal character code
Non-MF BASIC normal mode		80 x 25 characters	1-byte ASCII code
MF mode		40 x 20 characters	1-byte ASCII code
MFBASIC mode	Width 80	Max. 80 x 20 characters	1-byte ASCII code or 2-byte MultiFonts code
	Width 40	Max. 40 x 20 characters	1-byte ASCII code or 2-byte MultiFonts code

The display is cleared when the first data is displayed after switching from any of these modes to another. Therefore, the display must be used in the MFBASIC mode when alphanumeric characters are to be displayed in the same screen with Multiple Font characters. Further, the "WIDTH 40" specification can only be made in the MFBASIC mode.
When a cold start is made, display is set in the Non-MFBASIC normal mode. If a warm boot is made after switching modes, the display is returned to the mode it was in prior to switching; the display mode in use when the warm boot is made can be maintained by setting the following flags.

Non-MFBASIC normal mode	0FE82H <-- 00H
	0FE99H <-- 00H
	0FE40H <-- 00H
	0FE50H <-- 00H
MF mode	0FE82H <-- 0FFH
	0FE99H <-- 0FFH
	0FE40H <-- 00H
	0FE50H <-- 00H
MFBASIC mode	0FE82H <-- 00H
	0FE99H <-- 00H
	0FE40H <-- 0FFH
	0FE50H <-- 0FFH

3.2 Call procedure

Entry parameters: Register C=Data to be output

Return information: None

Example a) Alphanumeric character display in the Non-MFBASIC normal mode
Display by the following routine terminates when "0" is input.

```
CONOUT    EQU     0F60CH
          .
          .
          LD      HL,OUTDATA1 ;
OUTLOOP1:LD       A,(HL)      ;
          CP      0           ;
          JR      Z,OUTEND1   ;
          LD      C,A         ;
          PUSH    HL          ;
          CALL    CONOUT
          POP     HL
          INC     HL
          JR      OUTLOOP1
;
OUTEND1:  .
          .
          .
OUTDATA:  DB      'ABC',020H,0B1H,0B2H,0B3H,0
```

Result of execution: ABC_123

Example b) Alphanumeric and Multiple Font character output in the MFBASIC mode

```
CONOUT    EQU     0F06CH
MFBFLG    EQU     0FE50H
MFBMOD    EQU     0FFH
          .
          .
          LD      A,MFBMOD     ;SET MFBASIC MODE.
          LD      (MFBFLG),A
          .
          .
          LD      HL,OUTDATA2
OUTLOOP2: LD      A,(HL)
          CP      0
          OR      Z,OUTEND2
          LD      C,A
          PUSH    HL
          CALL    CONOUT
          POP     HL
          INC     HL
          JR      OUTLOOP2
;
OUTEND2:  .
          .
          .
OUTDATA2: DB      'ABC ',0B1H,0B2H,0B3H,020H
          DB      0A4H,0C1H,0A4H,0C2H,0A4H,0C3H,0
```

Result of execution: ABC 123 ｱｲｳ

Example c) This exmaple displays characters in arbitrary display screen locations in arbitrary colors (with a green display, characters are always displayed in green). (This sequence can only be used in the MFBASIC mode.)

```
DB      01BH,'='     ;CURSOR POSITION (ESC "=")
DB      10+32,20+32 ;DISPLAY START COL21, LINE11
DB      01BH,'L05'   ;L BLUE DISP SPEC (ESC "L0n")
DB      'ABC '      ;DISPLAY DATA
DB      0B1H,0B2H,0B3H,020H
DB      0A4H,0C1H,0A4H,0C2H,0A4H,0C3H,0
```

Result of execution: ABC 123 ｱｲｳ

* BIOS memory is rewritten as follows.

A) When using DDT

1 DDT

2 -SFE50

3 FE50 00 FF

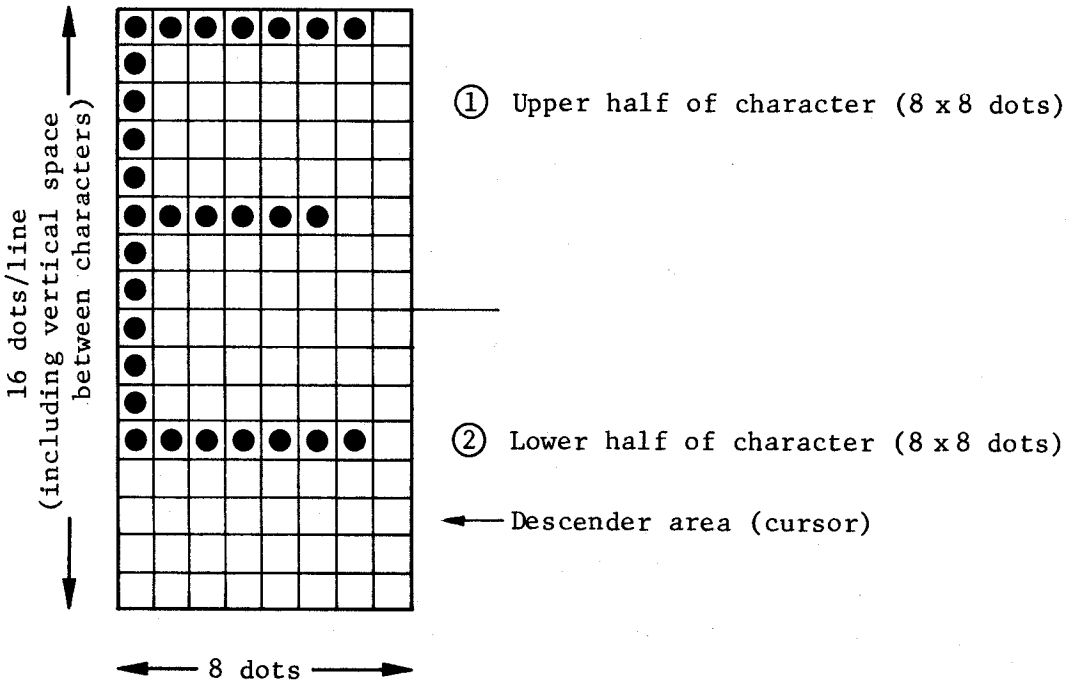
Changes the mode.

Indicates current contents of 0FE50H.

```
4  FE51 00 .
5  - C              (BREAK or CTRL-C; ends DDT)
```

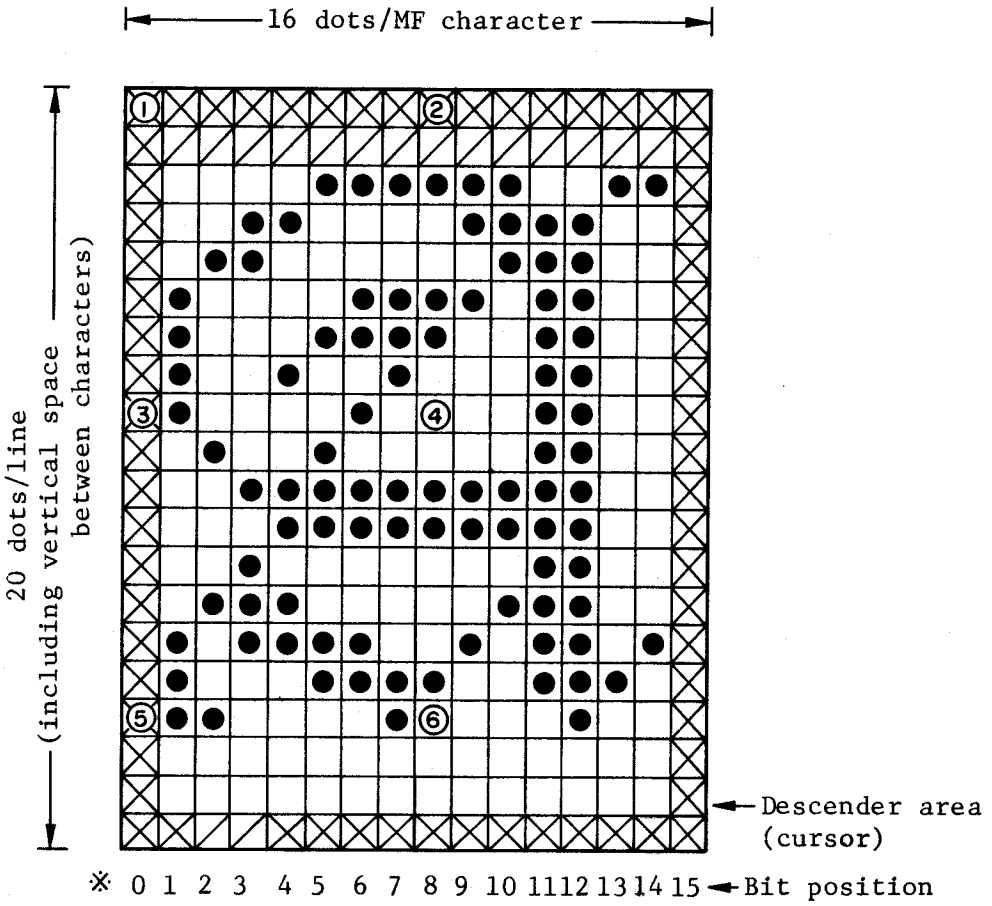
* Display of individual characters is as follows for a green monitor in the non-MFBASIC MF mode or the MFBASIC mode, or for a color monitor.

A) Color monitor, ASCII character in the Non-MFBASIC normal mode



Since an 8 x 16 dot pattern cannot be displayed in one operation, 8 x 8 dot sections 1 and 2 are displayed in sequence by two operations.

B) Display in the Non-MFBASIC MF mode (color or green monitors)



Here, a total of six operations are required to display the four 8 x 8 dot areas and the two 8 x 4 dot areas.

Dots marked with an "X" in the figure above are used for extending character display outward according to the settings of four extension bits (one for each of the four sides of the character) which are provided for each character in the character generator. When the extension bit for a given side of the character is 1, the "X" dots on the corresponding side of the character are set (turned on) if adjacent inner dots are set; if adjacent inner dots are reset (turned off), corresponding "X" dots are not set. When the extension bit is 0, the "X" dots on the corresponding side are reset whenever the character is displayed.

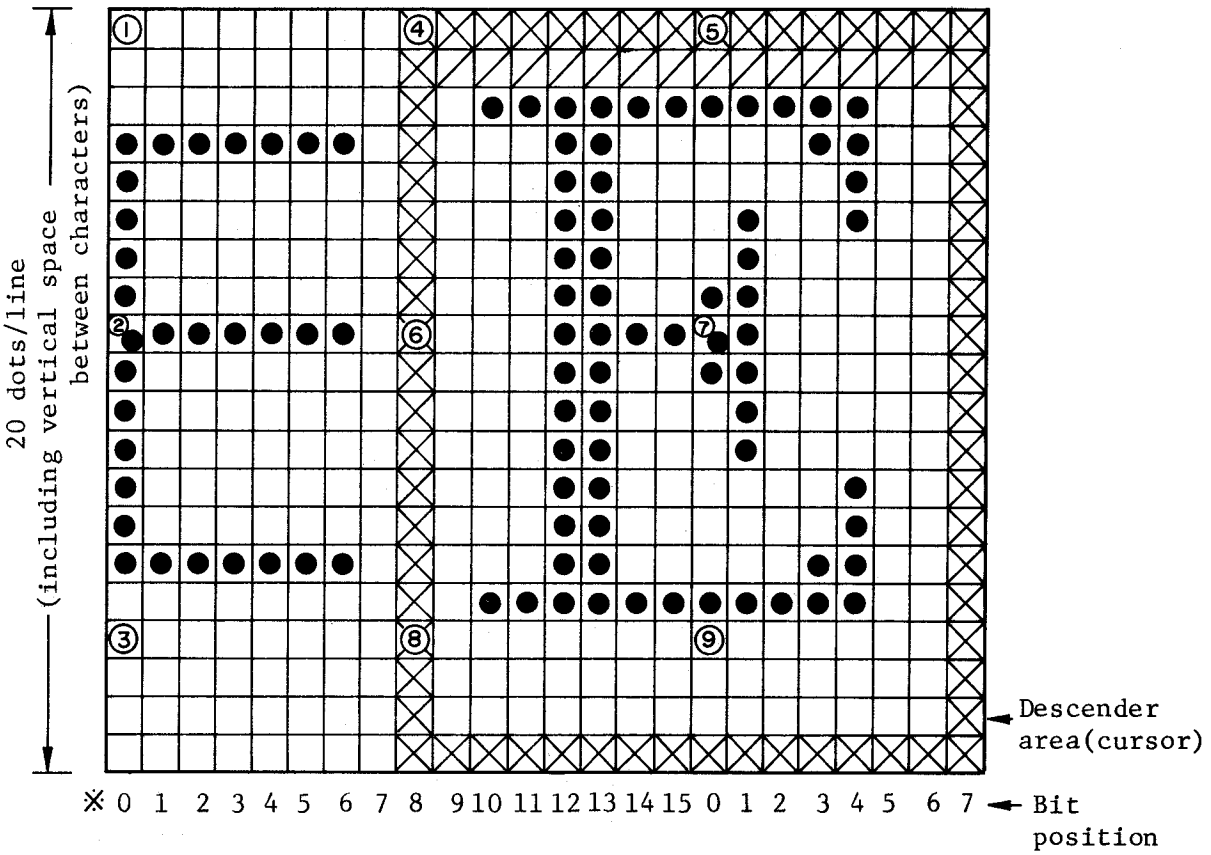
Dots marked with a diagonal act in the same manner as "X" dots for 2-byte characters with codes from 0D0A0H to 0D5BFH.

* VRAM addressing for bit specification

All bits from 0 to 15 are addressed under the same address (the address is incremented by one for each 2 bytes).

C) Display in the MFBASIC mode (with color/green monitor)

(C-1) Display in the WIDTH 80 mode



With 1-byte characters, 3 bits are added to the top of the pattern described in A) and 1 space bit is added to the bottom, then areas 1, 2, and 3 are displayed in succession. With 2-byte characters, display is the same as described in B).

(C-2) Display in the WIDTH 40 mode

In this mode, 1-byte characters are displayed as described in B). Two-byte characters are doubled in size in the horizontal direction and displayed by performing the procedure described in B) two times.

* When 2-byte (MultiFonts) characters are displayed in the Non-MFBASIC MF mode or MFBASIC mode, characters displayed at the right side of the screen may be split into two parts (with the right-hand part displayed at the beginning of the following line). This occurs when the right half of a 2-byte character extends beyond the right edge of the screen.

  BIOS does not check for this situation; therefore care must be taken when displaying 2-byte characters starting at an even-numbered column. (Structure programs so that 2-byte characters are not displayed in the 80th column.)

* VRAM display addresses

VRAM display addresses and display modes vary according to the type of monitor used as indicated below.

a) Display modes

System display mode	Non-MFBASIC normal mode	Non-MFBASIC MF mode/ MFBASIC mode
Green monitor	Character/graphic mode (only 1-byte characters can be displayed)	Character/graphic mode (only graphics can be displayed)
Color monitor	Graphic mode	Graphic mode

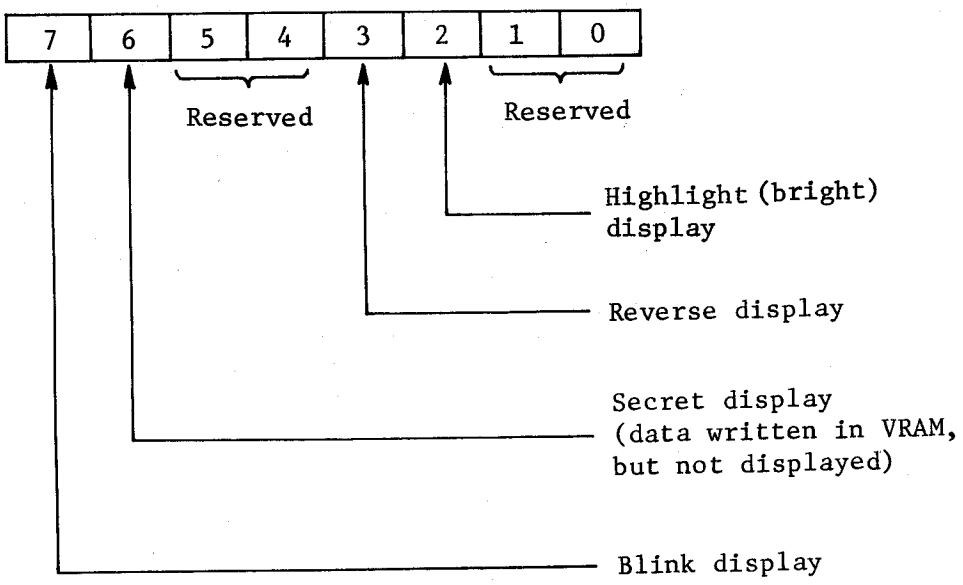
b) VRAM display addresses

(i) With a green monitor in the non-MFBASIC mode (the character/graphic mode with character display only), display addresses are from 0 to 16383 (0H to 03FFFH). The address is incremented as each character is displayed. After characters have been displayed to address 16383 (03FFFH), address 0 is written even if the program specifies address 16384 (04000H).

Each address indicates a 16-bit VRAM location. Therefore, the write data for each character consists of the 8-bit character code (ASCII code) and its attribute character.

The attribute character is composed of 8 bits which can be specified as follows.

Attribute character



Attribute characters are written into memory following the character code. Since the attribute is not changed until it is rewritten, subsequent characters can also be displayed just by writing the character code. It is also possible to set the attribute by console escape sequence.

After characters have been displayed on the entire screen of a green monitor (80 columns x 25 lines), the next display address becomes 2000 (07D0H).

ii) Display addresses are incremented in 16-bit units for a green monitor in the non-MFBASIC MF mode or MFBASIC mode, or a color monitor in any mode.

Therefore, 640/16=40 addresses are used to display the top line (the 640 dots at the top of the screen); i.e., the top line display addresses range from 0 to 39 (027H). When the address of character section 1 on page is 41, that of section 2 is 40+40X8=360 (0168H). The address of section 4 is the same as that of section 1, but with bit position 8 specified. (For section 1, the bit position is 0.)

Therefore, VRAM addresses range from 0 to 16383 (03FFFH), but the number of data bytes is 32768.

* Differentiation between green and color monitors

The type of monitor used is determined by reading I/O port 02CH; a green monitor is connected if the value read is 0FEH, and a color monitor is connected if it is 0FDH.

*Color specification for a color monitor (VRAM selection)

The RGB setting for a color monitor is made by writing the following data to I/O port 02DH.

Blue (B) 02DH<--01B (01H)

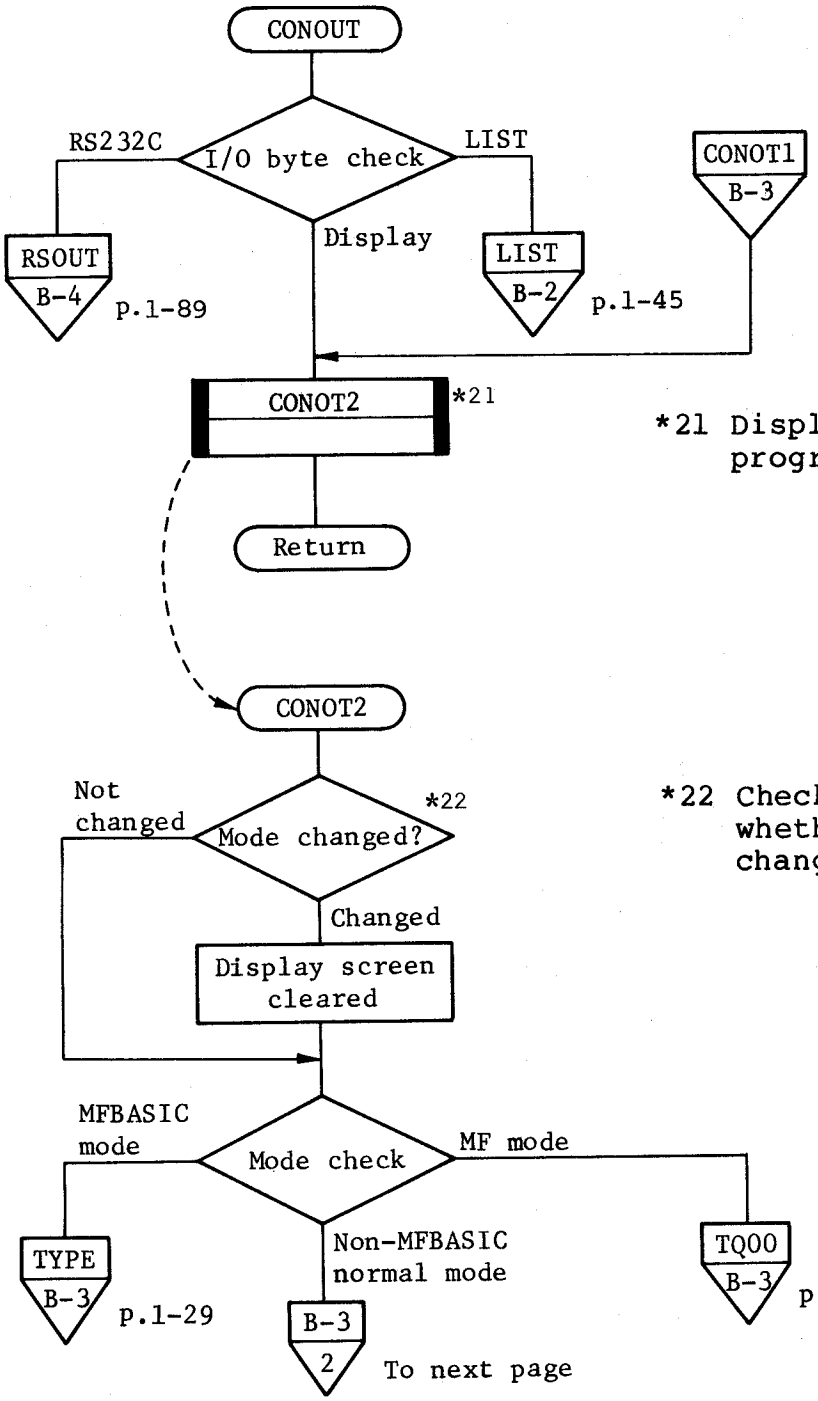
Green (G) 02DH<--010B (02H)

Red (R) 02DH<--0100B (04H)

Warning: Only one bit can be set to "1" at a time. Setting two or more bits to "1" may damage the hardware.

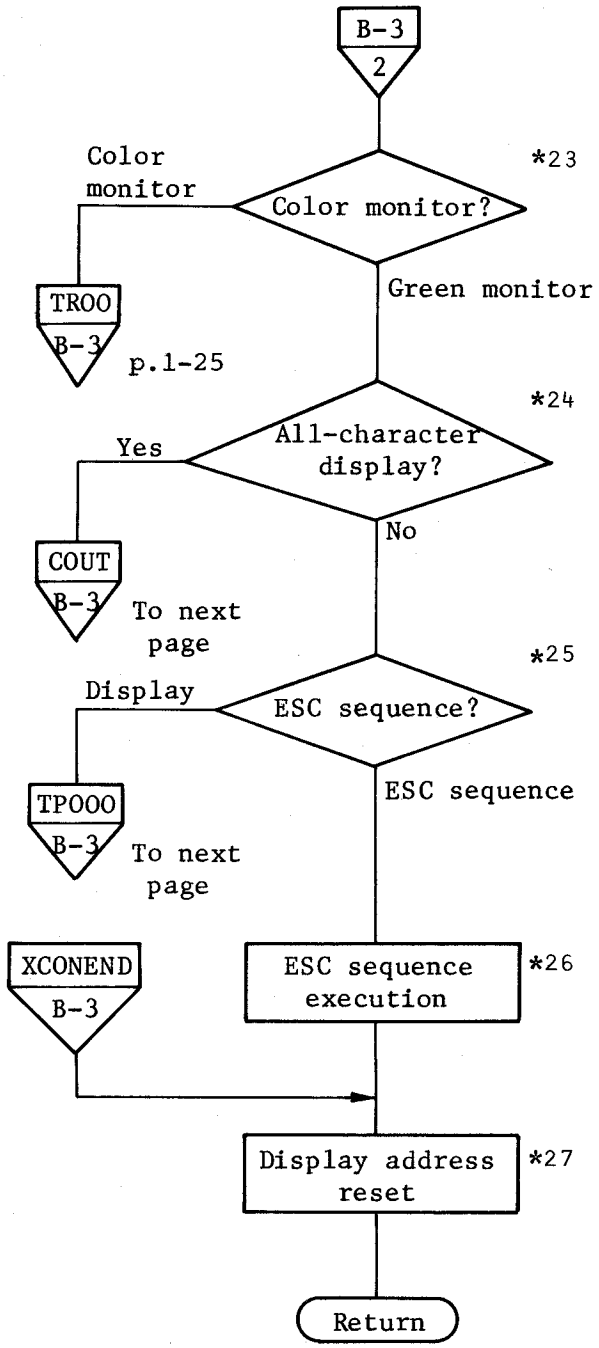
A total of seven colors can be displayed by displaying the three basic color in combinations; e.g., data (for blue, green, and red) must be written three times in order to display white.

3.3 General flowchart



*21 Display output control program

*22 Check made to determine whether the mode has been changed.



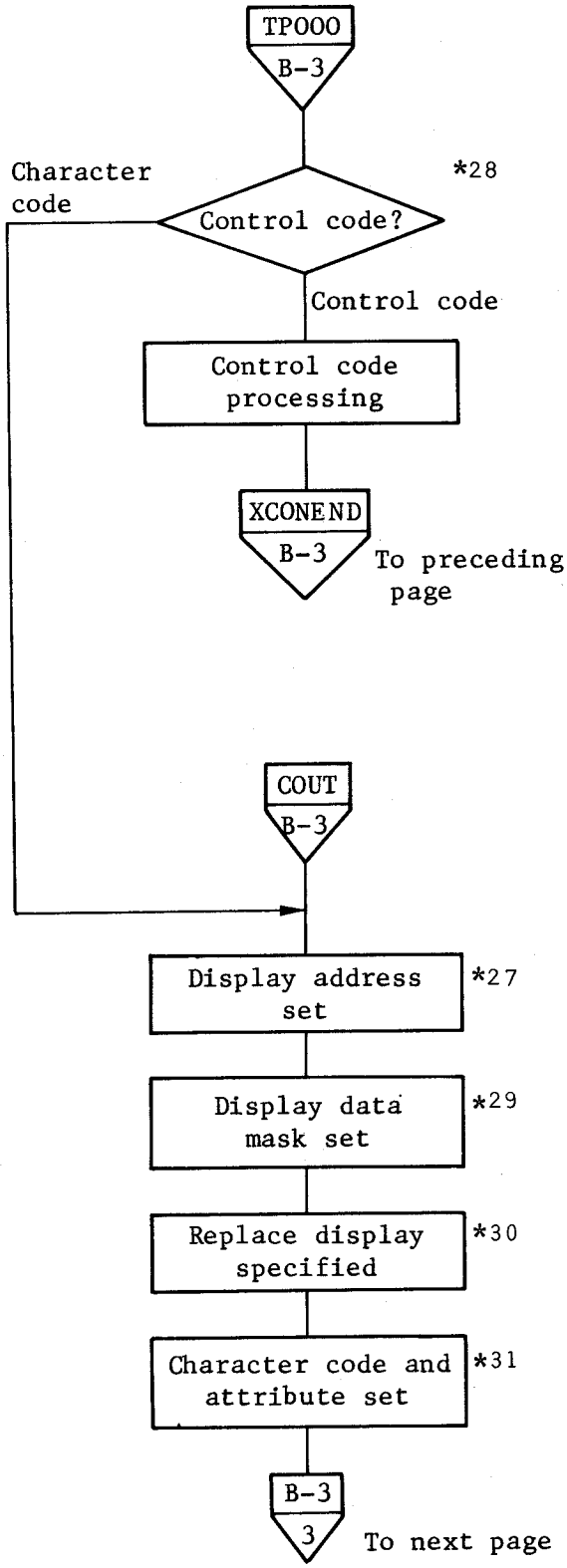
*23 Display I/O port checked.

*24 Check made as to whether the ESC "%" mode has been specified.

*25 Check made as to whether the preceding data is 01BH (the ESC code).

*26 ESC sequences executed and ESC flag cleared.

*27 Subroutine CSRW called to send the CSRW command to the μPD7220.

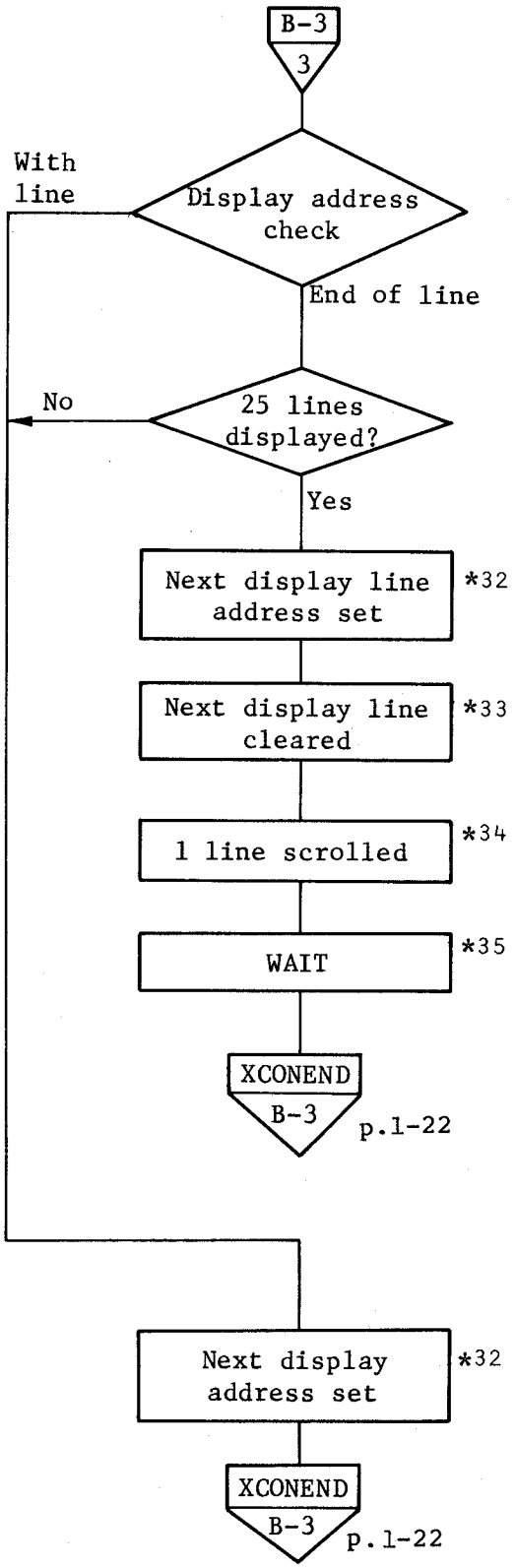


*28 See the character set in the Non-MFBASIC normal mode for the control codes (EL, BEL, BS, TAB, LF, HOME, CLS, ES, ESC, and the cursor control codes).

*29 Subroutine MASKFF called to send the MASK command to the μPD7220.

*30 Subroutine CRTWTE called to send a WRITE command to the μPD7220 and specify the replace mode.

*31 Character code and attribute code parameters for the WRITE command to the μPD7220. Character is displayed when these parameters are output.

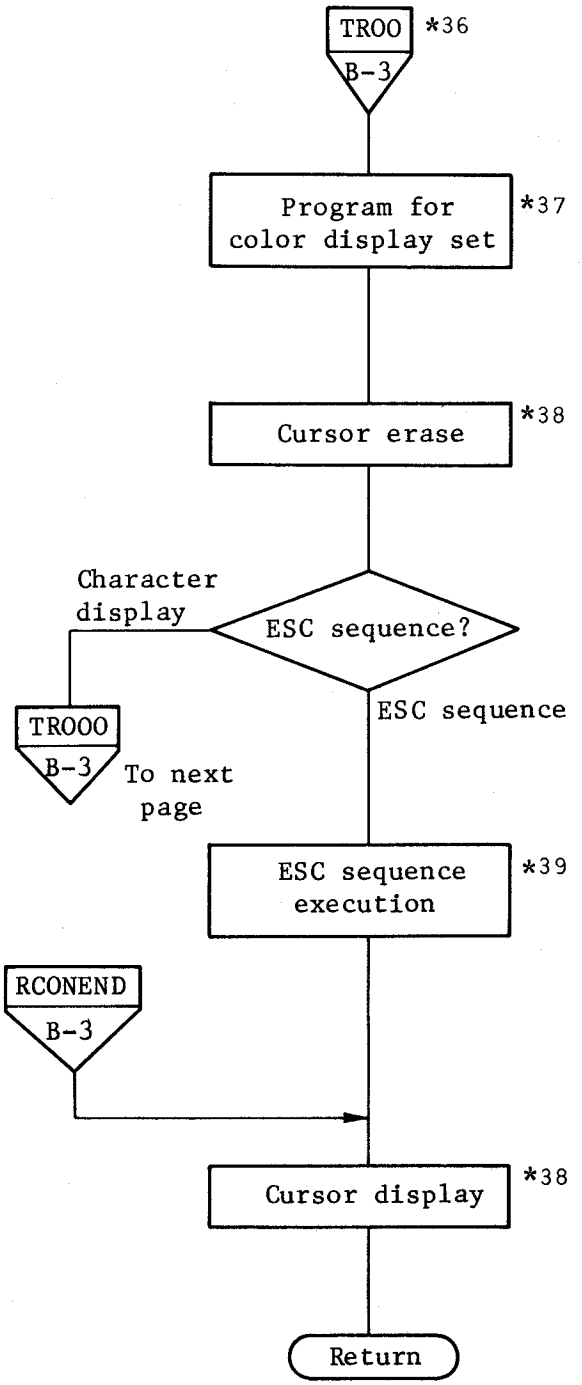


*32 Subroutine XCALCUSR called.

*33 Subroutine CLRSCRN called to send the VECTN, MASK, and WRITE commands to the μPD7220.

*34 Subroutine SCRLCHR called to send the SCROLL command to the μPD7220.

*35 Wait for scrolling.

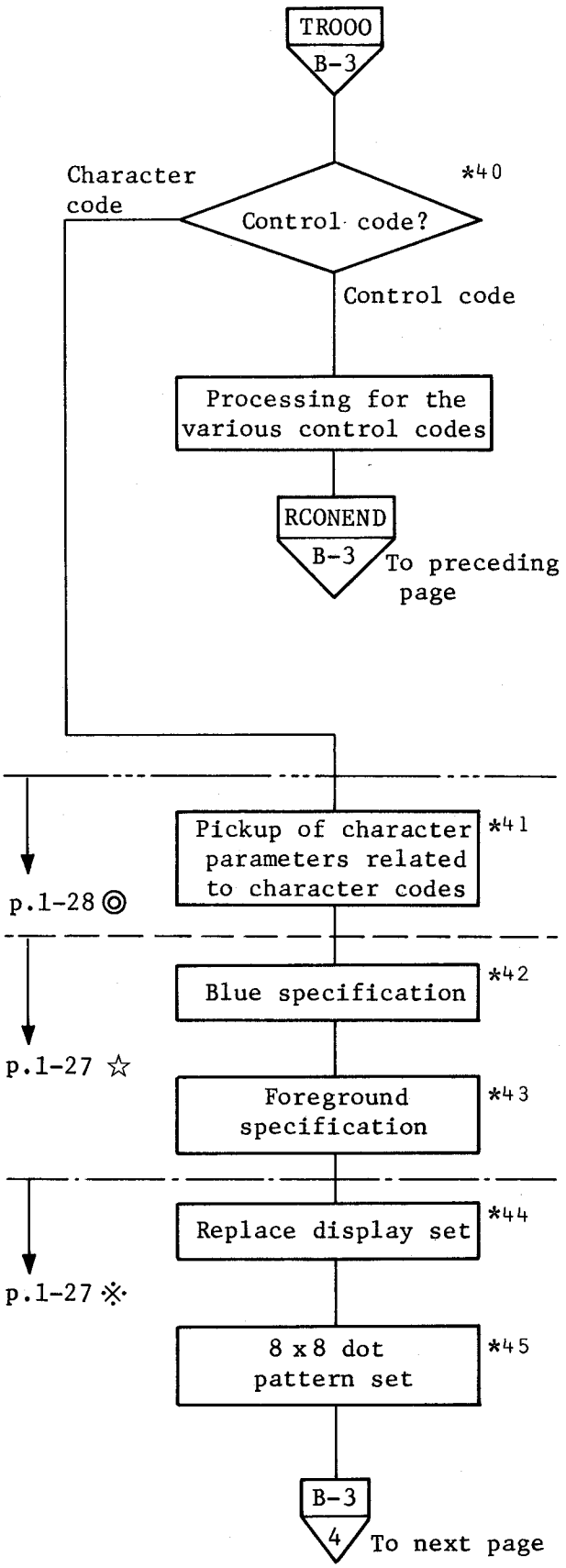


*36 Non-MFBASIC normal mode character display for color monitor.

*37 Subroutine COLCHK called to make settings required for color display. This is necessary because the parameters sent to the μPD7220 are different for the green and color monitors.

*38 Subroutine DYCUSR called to send the MASK, WRITE (in the complement mode), TEXTW, DSRW, VECTW, and TEXTE commands to the μPD7220. In the cursor display mode, the cursor display is turned off if on, and vice versa.

*39 ESC sequences executed and the ESC flag cleared.



*40 See the character set in the Non-MFBASIC normal mode for the control codes (EL, BEL, BS, TAB, LF, HOME, CLS, ESC, and the cursor control codes).

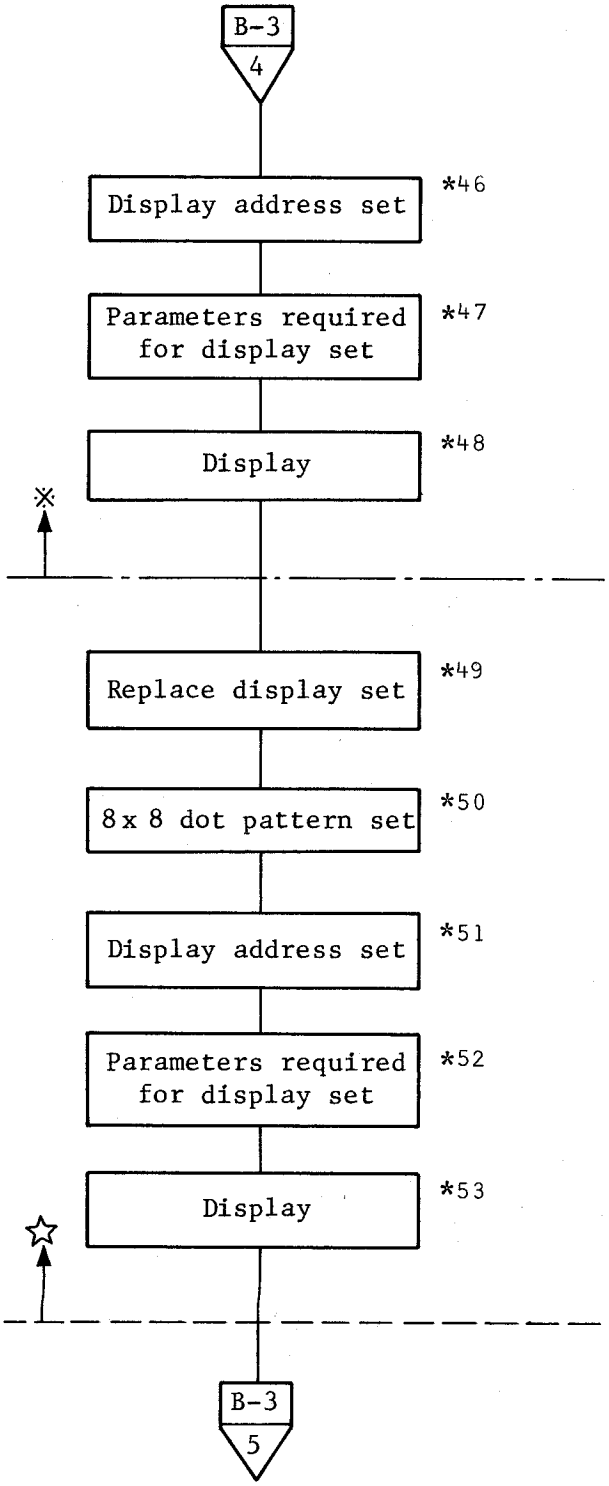
*41 The character set for the color monitor is stored in the form of bit patterns; related data is stored in the system bank area starting at address 09900H.

*42 Subroutine BLUE called to set blue VRAM.

*43 Foreground display specification.

*44 Subroutine CRTWTE called to send the WRITE command to the μPD7220 and specify the replace mode.

*45 Subroutine TEXTW1 called to send the TEXTW command to the μPD7220 for display of the upper half of a character in the Non-MFBASIC normal mode (section 1 of the character pattern on page 1-16).



*46 Subroutine CSRW called to send the CSRW command to the μPD7220.

*47 Subroutine VECTW1 called to send the VECTW command to the μPD7220 and set the 8x8 dot pattern.

*48 Subroutine TEXTE called to send the TEXTE command to the μPD7220.

†) In the Non-MFBASIC mode, upper half of character is displayed by steps *44 to *48. See page 1-16.

*49 Subroutine CRTWTE called to send the WRITE command to the μPD7220.

*50 Subroutine TEXTW1 called to send the TEXTW command to the μPD7220.

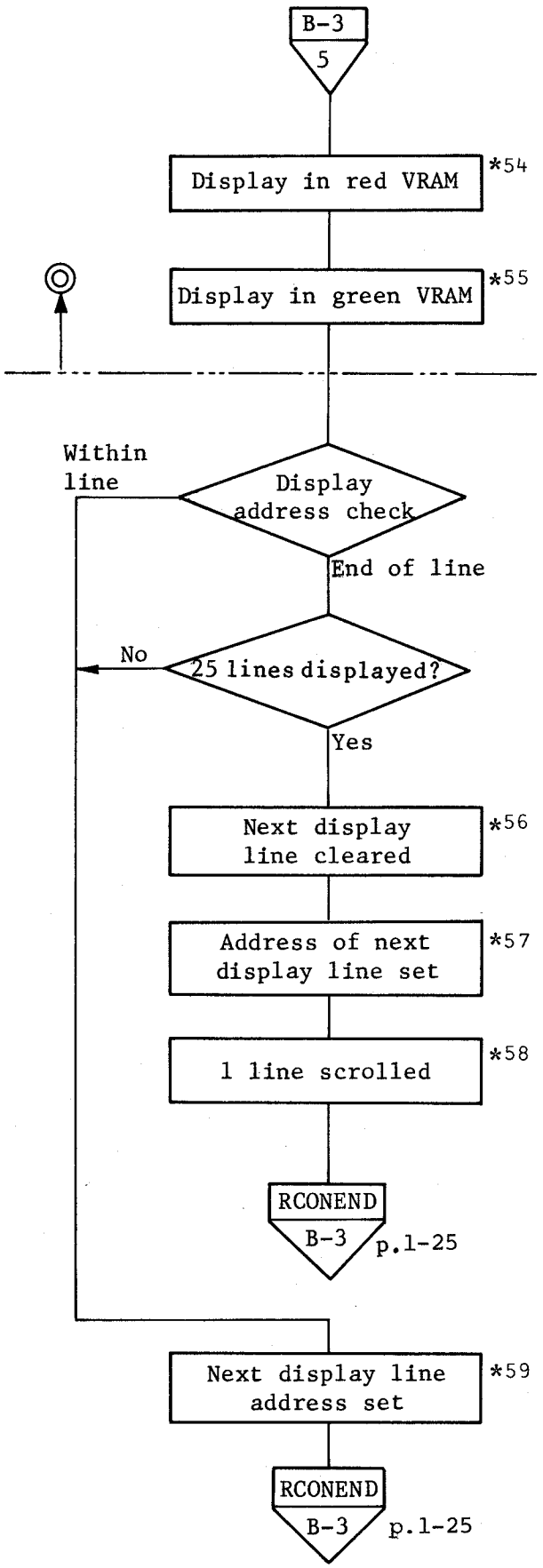
*51 Subroutine CSRW called to send the CSRW command to the μPD7220.

*52 Subroutine VECTW1 called to send the VECTW command to the μPD7220 and set the 8x8 dot pattern.

*53 Subroutine TEXTE called to send the TEXTE command to the μPD7220.

†) In the Non-MFBASIC normal mode, lower half of character is displayed by steps *49 to *53. See page 1-16.

†) In the Non-MFBASIC normal mode, display of one character in a specific color (blue in this case) is accomplished by steps *42 to *53.



*54 Red VRAM specified and steps *43 to *53 performed.

*55 Green VRAM set and steps *43 to *53 performed.

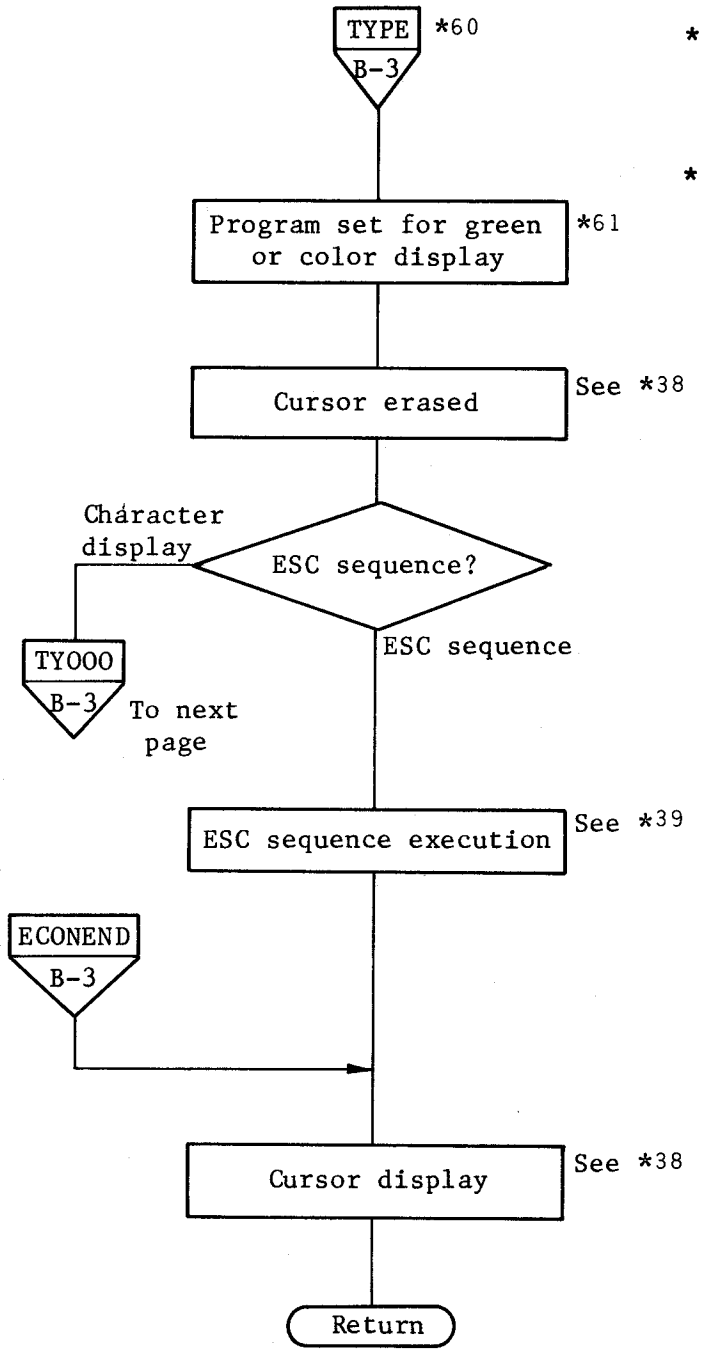
◎) One character displayed in white by steps *41 to *55.

*56 Subroutine CLRLINE called for scrolling to clear the next line for display. At this time, 40*16*25 is added to the lowest display address (the top left corner of the screen) in VRAM.

*57 Subroutine RCALCUSR called.

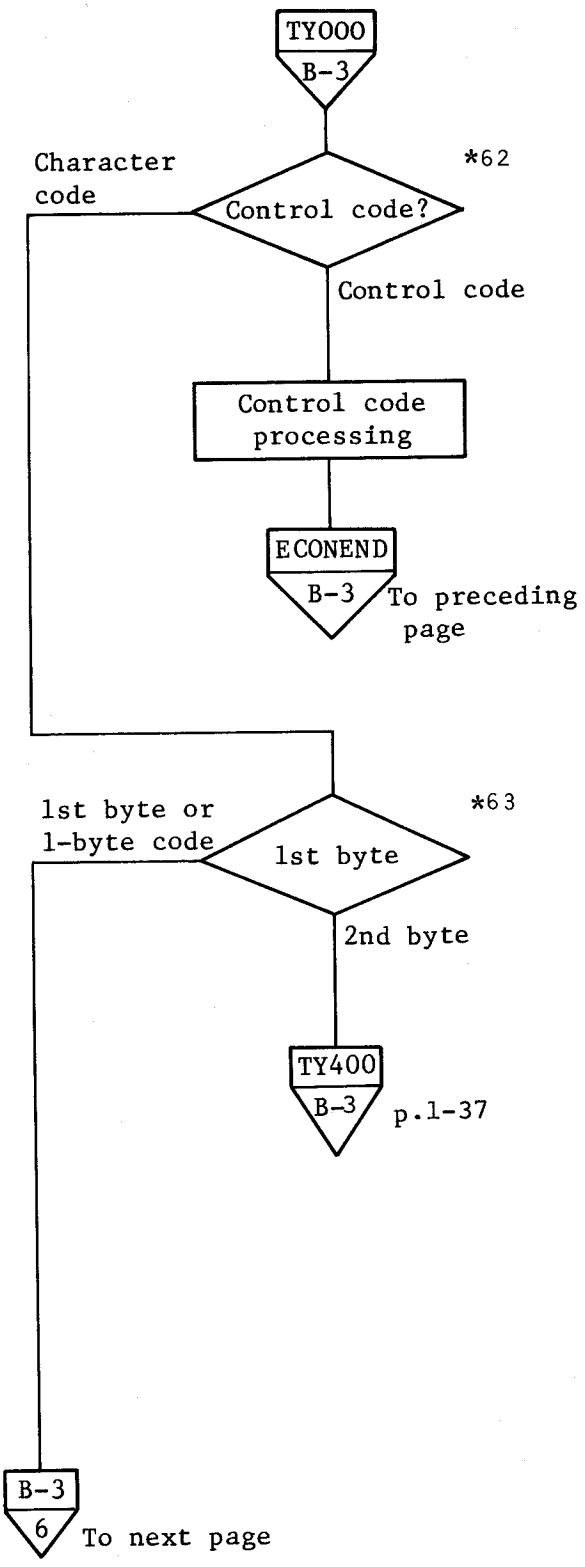
*58 Subroutine SCRLGRP called to send the SCROLL command to the μPD7220.

*59 Subroutine RCALCUSR called.



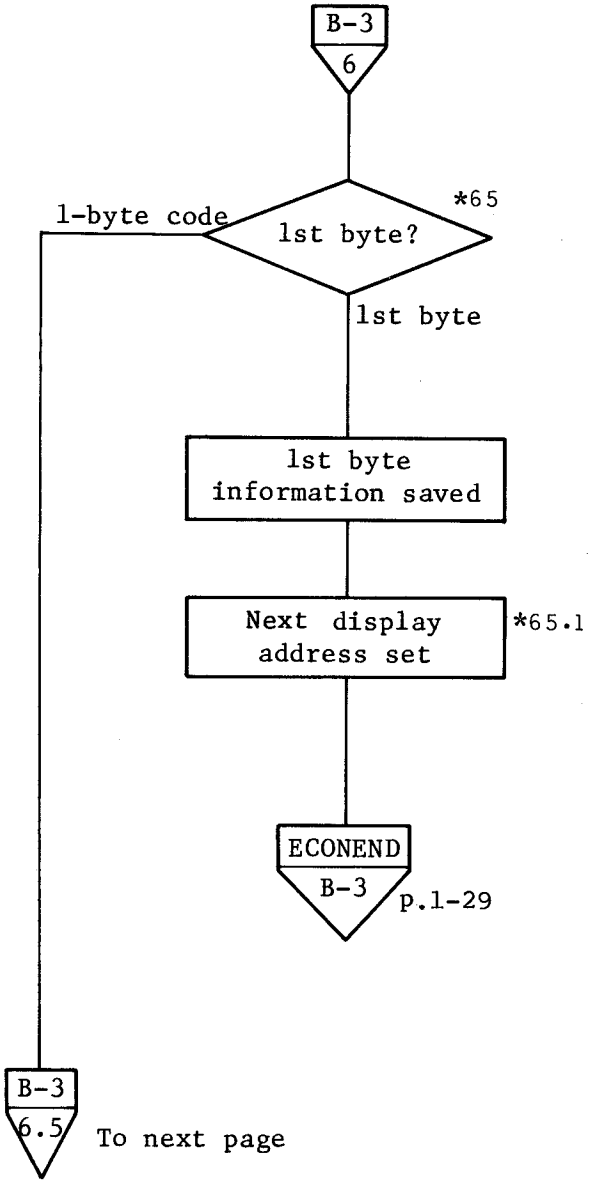
*60 MFBASIC mode character output routines for green and color monitors.

*61 Subroutine COLCHK called to check the type of monitor connected (green or color) and to set the program for output in the respective mode.



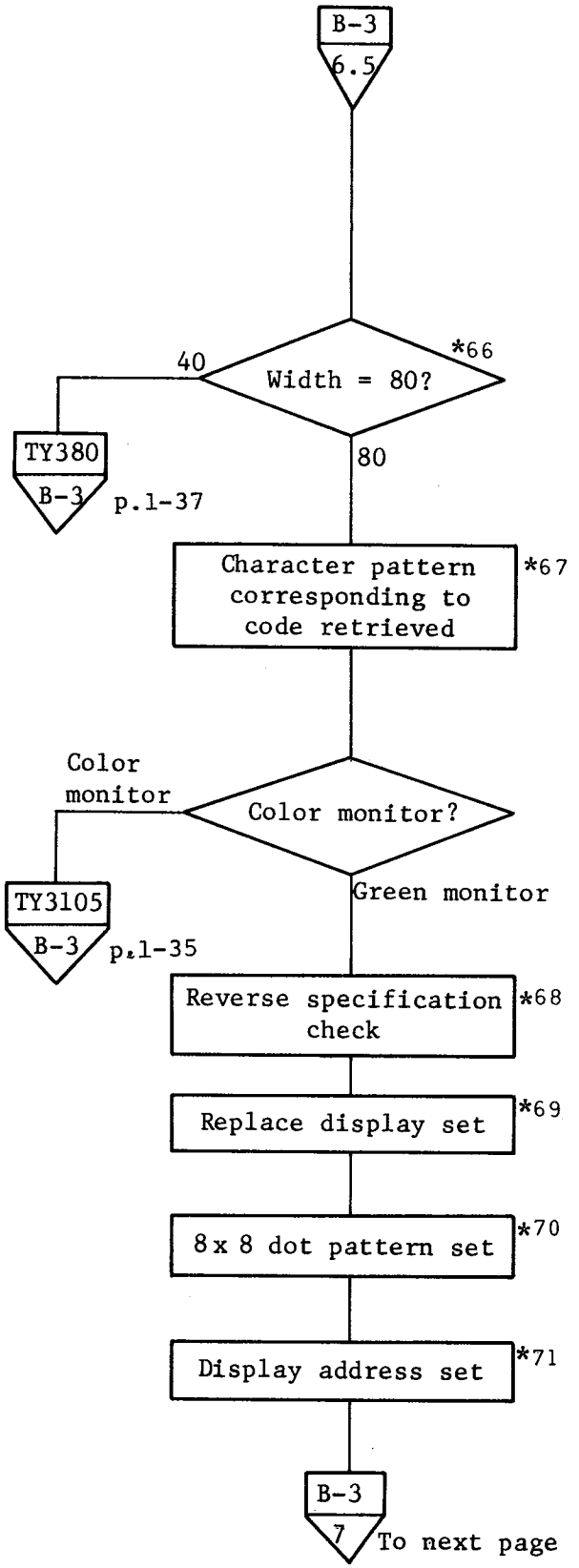
*62 See the character set in the Non-MFBASIC normal mode for the control codes (EL, BEL, BS, TAB, LF, HOME, CLS, ES, ESC, and the cursor control codes).

*63 Checks whether this is the 1st byte of a 2-byte code or the 2nd byte. 1-byte codes are handled in the same manner as the 1st byte of 2-byte codes.



*65 Checks whether this is a 1-byte code or the 1st byte of a 2-byte code.

*65.1 Subroutine ADVANCE called.



*66 Is this the WIDTH 80 display mode?

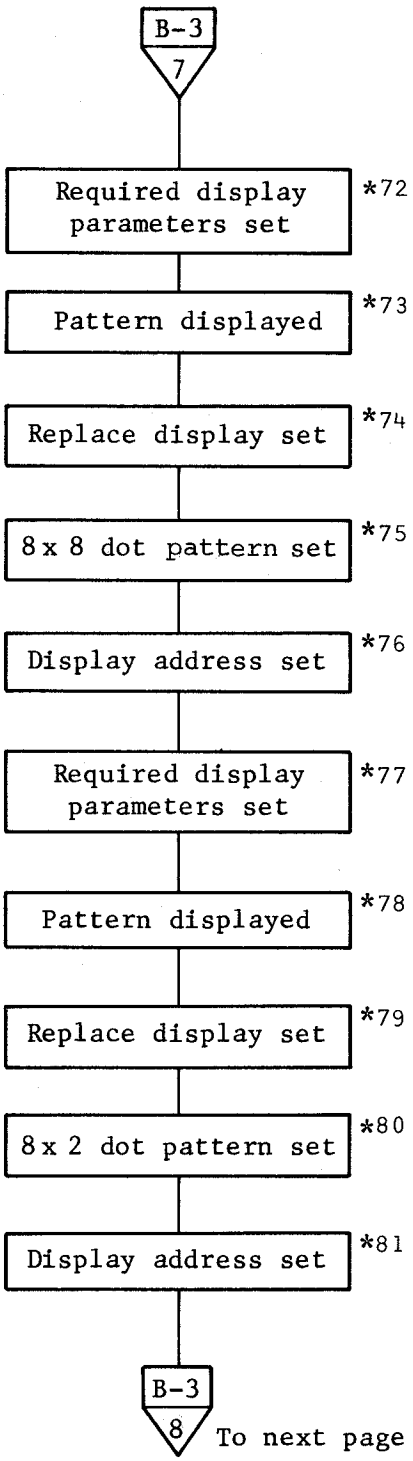
*67 In the MFBASIC mode, bit patterns are used for the character set with both the green and color monitors. Relevant data is contained in the system bank area starting at address 09900H.

*68 Reverse display set if so specified.

*69 Subroutine CRTWTE called to send a WRITE command to the μPD7220, thus setting the replace mode.

*70 Subroutine TEXTW1 called to send the TEXTW command to the μPD7220, thus setting the upper 8x8-dot half of the character in the MFBASIC mode (section 1 on page 1-18.

*71 Subroutine CSRW called to send the CSRW command to the μPD7220.



*72 Subroutine VECTW1 called to send the VECTW command to the μPD7220.

*73 Subroutine TEXTE called to send the TEXTE command to the μPD7220.

†) The upper 8x8 dot section of characters (section 1 on page 1-18) is displayed in the MFBASIC mode by steps *68 to *73.

*74 Same as step *69.

*75 Same as step *70; displays the center 8x8 dot section of characters (section 2 on page 1-18) in the MFBASIC mode.

*76 Same as step *71.

*77 Same as step *72.

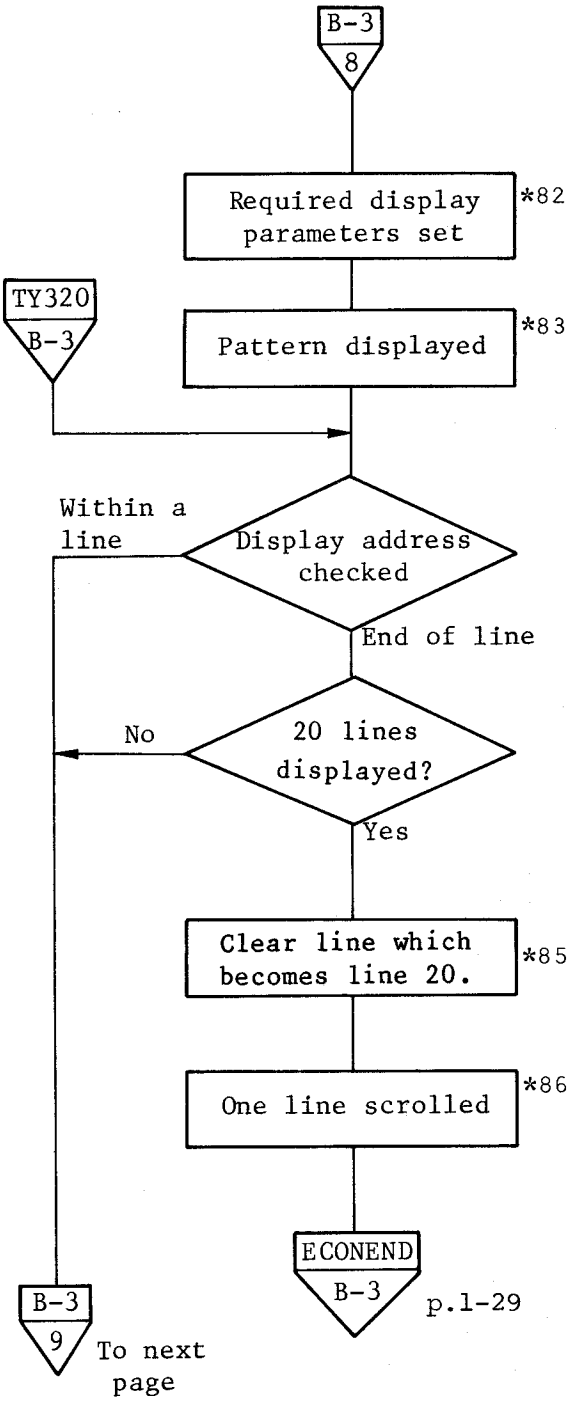
*78 Same as step *73.

†) Steps *74 to *78 display the 8x8 dot pattern corresponding to section 2 on page 1-18.

*79 Same as step *69.

*80 Subroutine TEXTW2 called.

*81 Same as step *71.



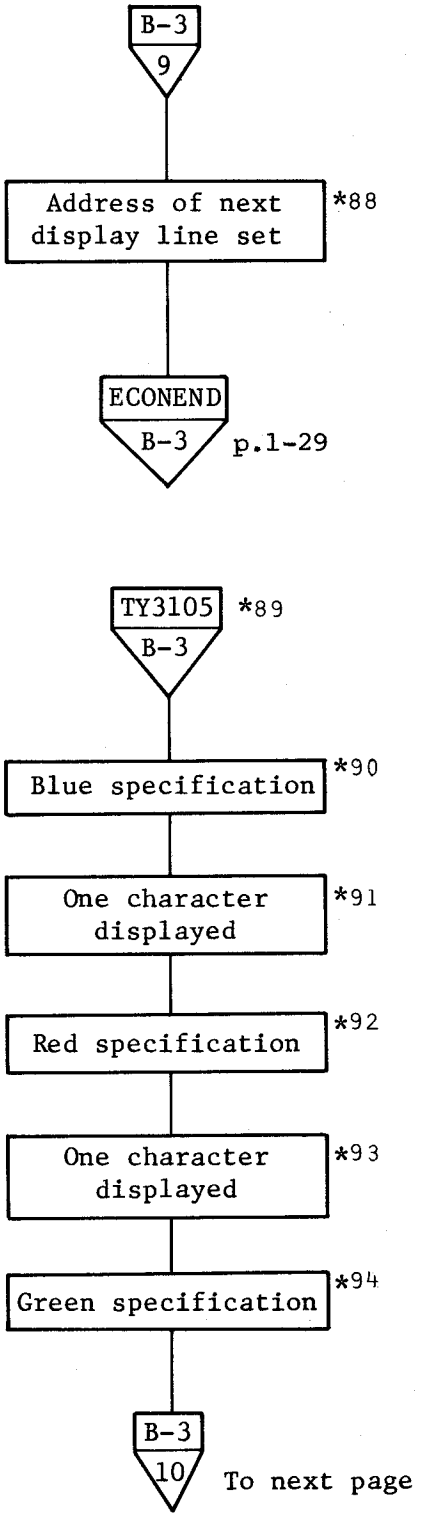
*82 Subroutine VECTW2 called to send VECTW command to μPD7220.

*83 Same as step *71.

†) One alphanumeric character (sections 1, 2, and 3 on page 1-18) displayed in the MFBASIC mode by steps *68 to *83.

*85 Subroutine CLRLINE clears line which becomes line 20 after scrolling.

*86 Subroutines ECALCUSR and SCRLGRP called.



*88 Subroutine ECALCUSR called.

*89 For display on color monitor.

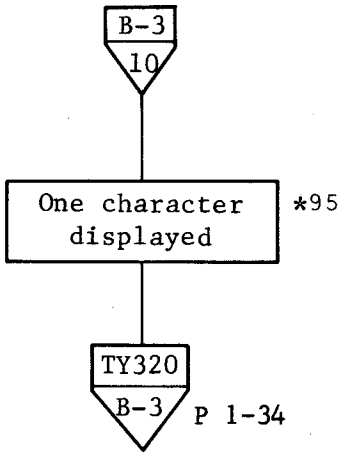
*90 Subroutine BLUE called.

*91 One character written into blue VRAM by steps *68 to *83.

*92 Subroutine RED called.

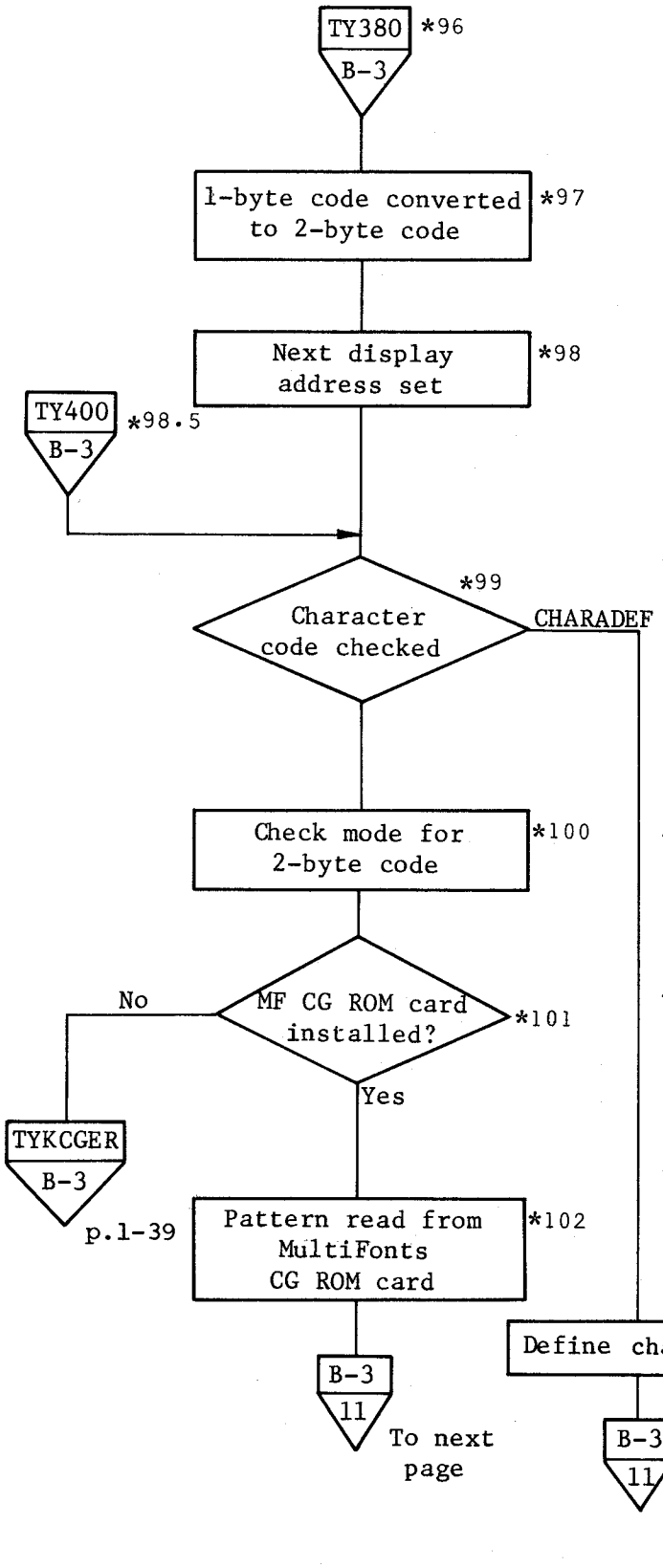
*93 One character written into red VRAM by steps *68 to *83.

*94 Subroutine GREEN called.



*95 One character written into green VRAM by steps *68 to *83.

* One alphanumeric character is displayed on a color monitor in the MFBASIC mode by steps *90 to *95.



*96 Displays a character in the MFBASIC WIDTH 40 mode.

*97 Converts an alphanumeric 1-byte character code into a 2-byte code. The conversion table is located in the system bank area starting at address 09800H.

*98 Subroutine ADVANCE called.

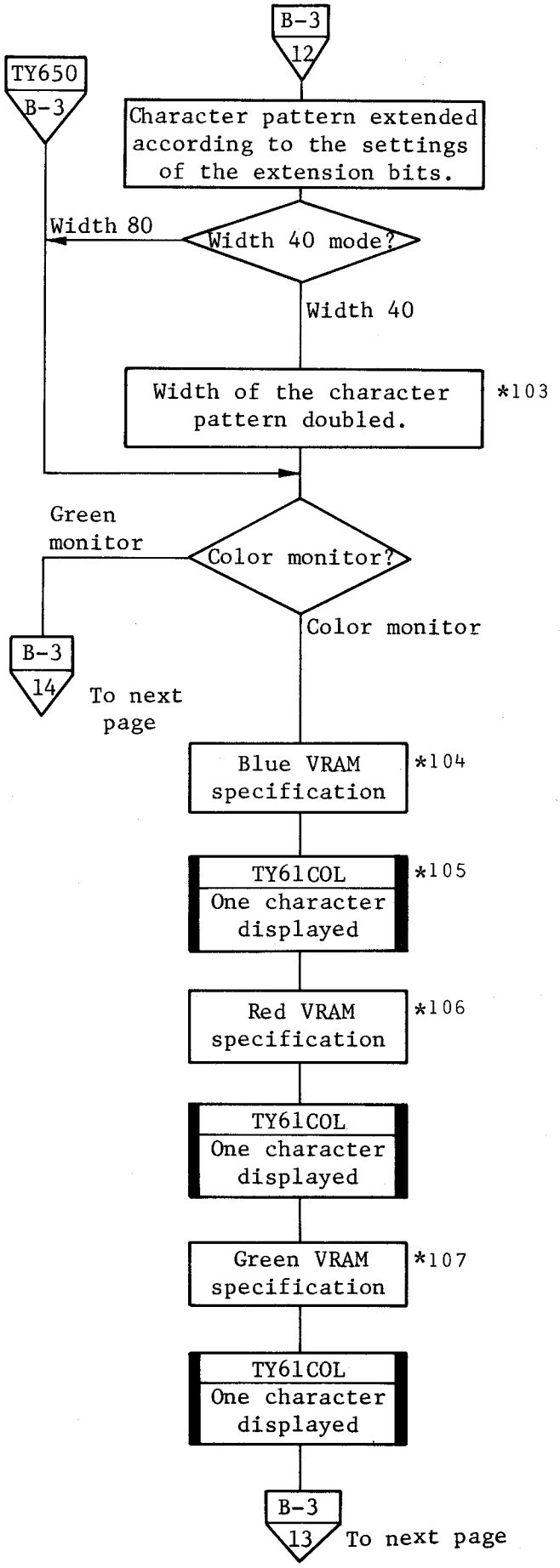
*98.5 From TY400 on, applies to both the MFBASIC WIDTH 80 and WIDTH 40 modes.

*99 Checks whether the code is that of a user defined character.

*100 Subroutine CONVCD called.

*101 Checks whether the MultiFonts CG ROM card is installed.

*102 Subroutines KCGOT and XCGIN called to read the 20x16 dot pattern from the MultiFonts CG ROM card.



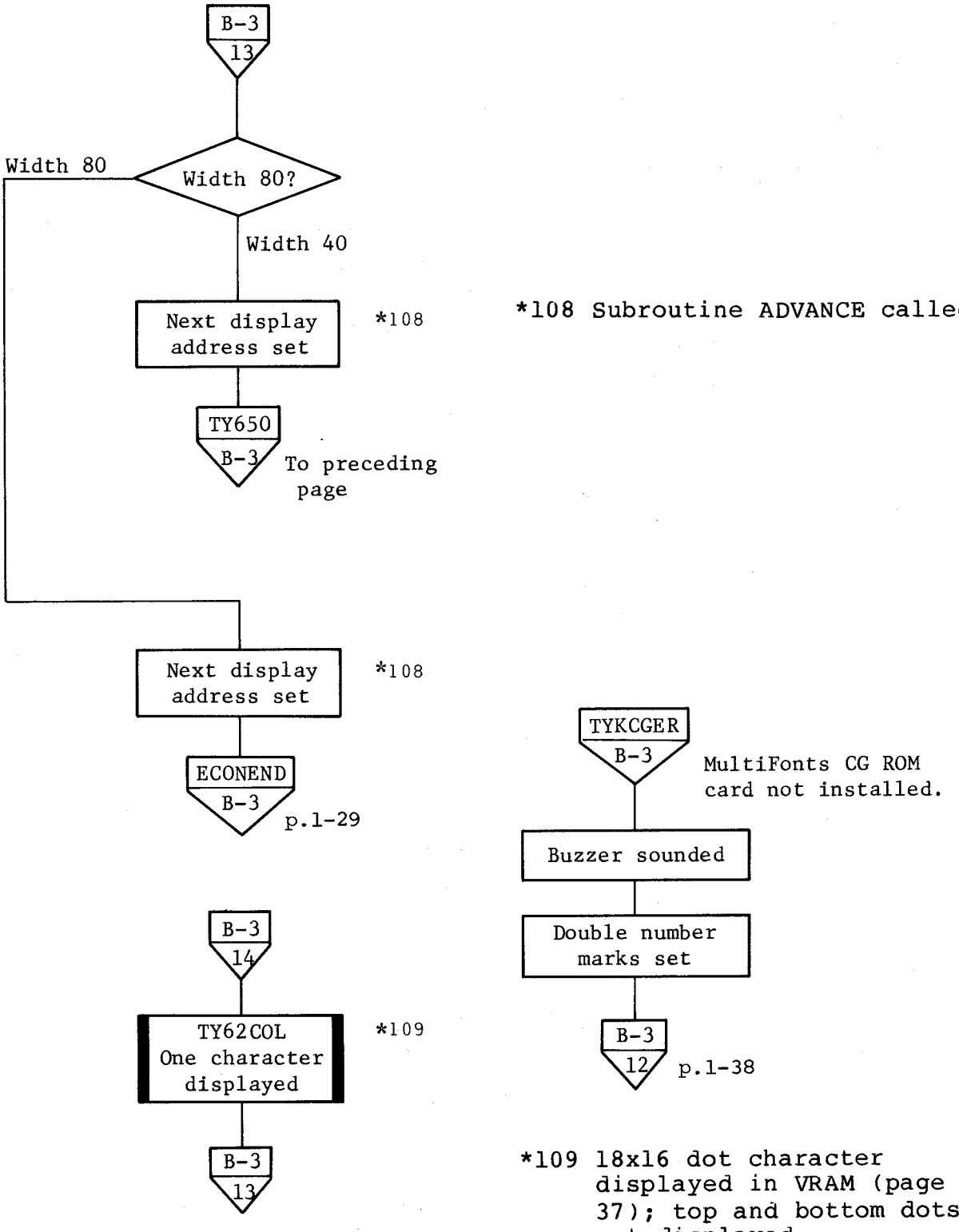
*103 Doubles the width of the character pattern from 20x16 dots to 20x32 dots.

*104 Subroutine BLUE called.

*105 One 18x16 dot character displayed in blue VRAM (top and bottom dots not displayed).

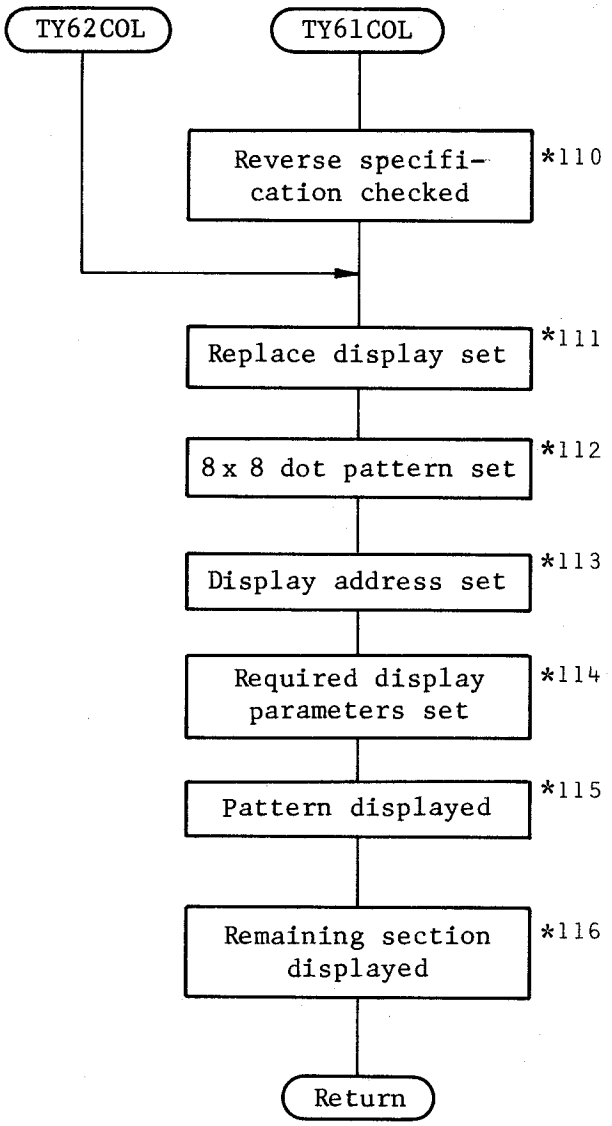
*106 Subroutine RED called.

*107 Subroutine GREEN called.



*108 Subroutine ADVANCE called.

*109 18x16 dot character displayed in VRAM (page 37); top and bottom dots not displayed.



*110 Reverse display specified if applicable.

*111 Subroutine CRTWTE called to send WRITE command to the μPD7220.

*112 Subroutine TEXTW1 called to send TEXTW command to the μPD7220.

*113 Subroutine CSRW called to send CSRW command to the μPD7220.

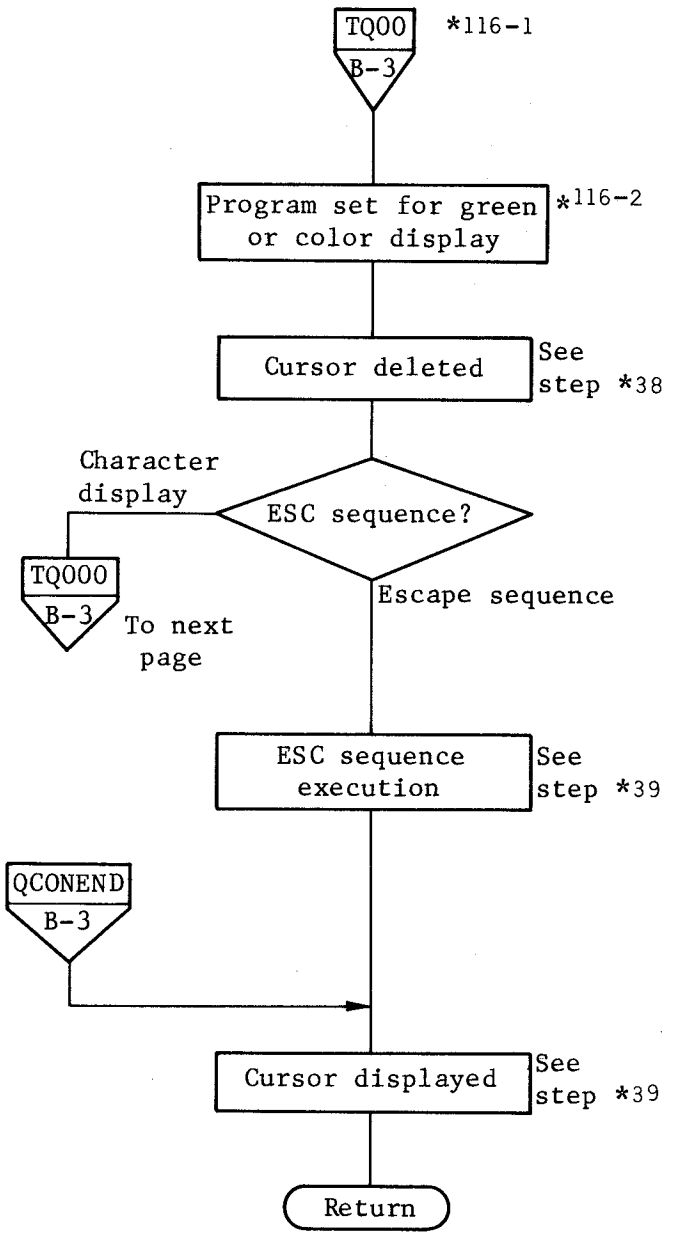
*114 Subroutine VECTW1 called to send VECTW command to the μPD7220.

*115 Subroutine TEXTE called to send TEXTE command to the μPD7220.

†) Upper 8x8 dot section (section 4 on page 1-17) of the multiple font character displayed by steps *111 to *115.

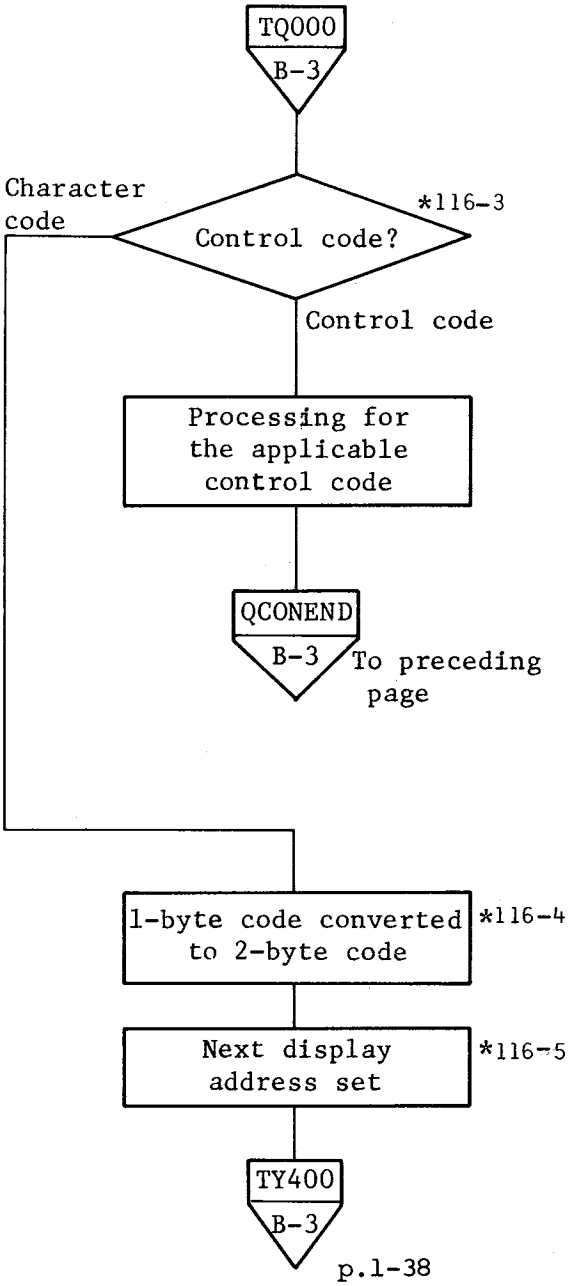
*116 Steps *111 to *115 repeated three times to display sections 5 to 7 of the character (see page 1-17.) At *112, the two 8x2 dot sections (sections 8 and 9) are displayed by calling subroutine VECTW2 twice.

†) One 2-byte character displayed by steps *110 to *116.



*116-1 MF mode output routine called.

*116-2 Subroutine COLCHK called to determine whether a green or color monitor is being used and set the program as applicable.



*116-3 See the character set in the Non-MFBASIC normal mode for the control codes (EL, BEL, BS, TAB, LF, HOME, CLS, ES, and ESC).

*116-4 1-byte (ASCII) code converted to MF character code (2 bytes).

*116-5 Subroutine ADVANCE called.

§4 LIST (List Output - Address: 0F60FH)

4.1 General

This routine outputs the character code set in register C to the device currently assigned to "LST:". All registers are changed by execution of this routine.

The device assigned to "LST:" is determined by the setting of the I/O byte. LIST devices include the printer, console, and RS-232C interfaces; the device assigned upon completion of a cold start is the printer.

Output to the printer is processed for the printer selected with the CONFIG transient command. Normal results cannot be expected when the printer actually connected differs from that specified.

If the BREAK key is pressed during output to the printer, output is aborted.

Notes concerning output to the printer

1. Printer status check procedures

Call LISTST to check the printer ready status, out-of-paper condition, and other printer errors. Results of this routine are set in registers A and B.

A register: 0 - Not ready
0FFH - Ready

B register: As follows, regardless of the A register value

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
SELECT OUT	POWER	$\overline{\text{READY}}$	OUT-OF-PAPER	$\overline{\text{ERROR}}$	0	0	0

When printer error processing is to be performed by application program, all that is necessary is to call the LISTST routine before calling the LIST routine.

2. Printer problems during printout

Ordinarily, BIOS operation is as follows when printer problems occur during printout.

A. When the printer is offline, the buzzer sounds after a certain time has passed; the buzzer can be stopped by pressing any key, then BIOS waits until the printer goes back online. Press the BREAK key if printout is to be abandoned.

B. When a printer error occurs, the buzzer sounds immediately; the buzzer can be stopped by pressing any key, then BIOS waits until the error condition has been cleared. If printout is to be abandoned, press the BREAK key.

To avoid situations such as the above, store 0FFH in BASIC (address 0FE7dH). If an error occurs, the printer goes offline, or the BREAK key is pressed, the printer status is stored in LISTERR (0FE9DH) and a return is made. (The buzzer does not sound.)

	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
LISTERR (0FE9DH)	SELECT OUT	POWER	$\overline{\text{READY}}$	OUT-OF-PAPER	$\overline{\text{ERROR}}$	0	0	1

4.2 Call procedures

Entry parameters: Register C=Output data

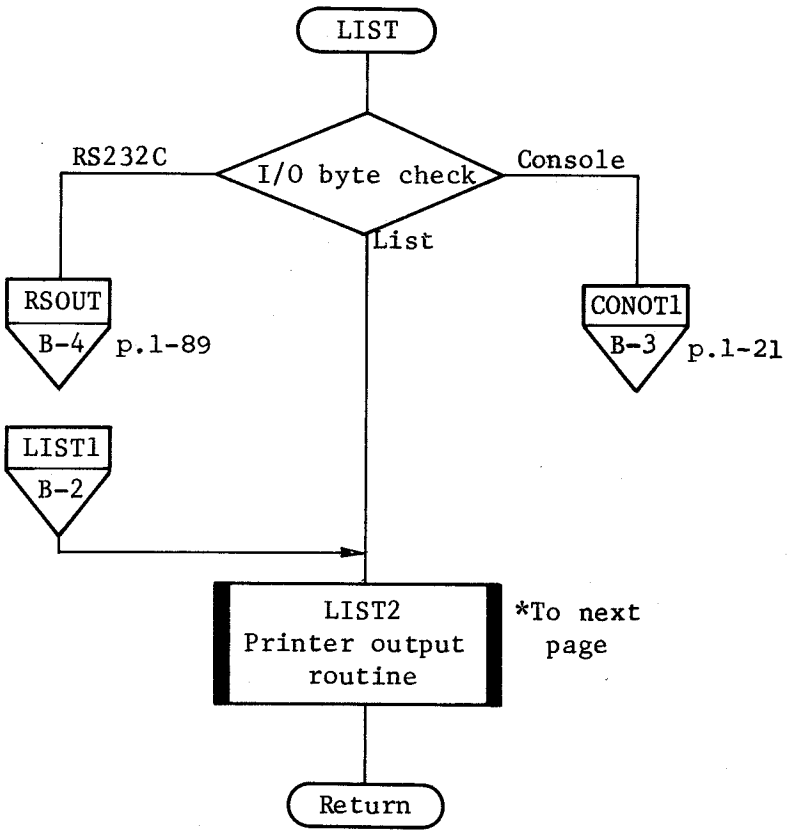
Example:

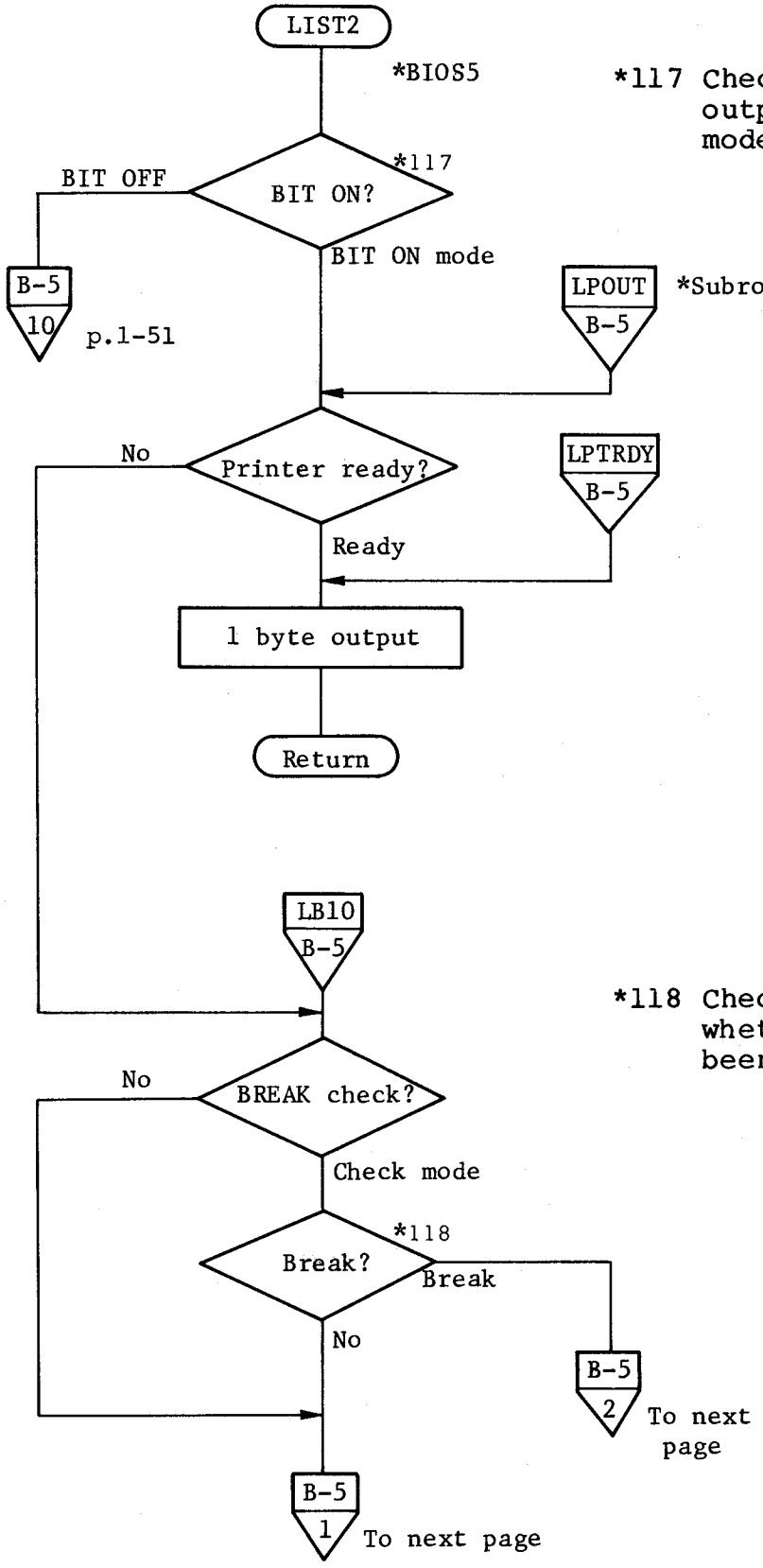
```
LIST      EQU     0F60FH
          .
          .
          LD      HL,OUTDATA1
OUTLOOP:  LD      A,(HL)
          CP      0
          OR      Z,OUTEND
          LD      C,A
          PUSH    HL
          CALL    LIST
          POP     HL
          INC     HL
          JR      OUTLOOP
;
OUTEND:   .
          .
          .
OUTDATA:  DB      'ABC '031H,032H,033H,020H
          DB      0A4H,0C1H,0A4H,0C2H,0A4H,0C3H,0
```

Result of execution: ABC 123 ｱｲｳ

Return information: None

4.3 General flowchart





*117 Check made as to whether output is in the BIT ON mode.

*118 Check made to determine whether the BREAK key has been pressed.