MSX 2 BASIC INPUT/OUTPUT SYSTEM

Written October 4th, 1985 (ASCII)
Edited 31st january 1986 (MSX Europe)

1.0 FORMAT

Following notations are used.

Address	address in hexadecimal
Name	name of function *n
Function	function to be performed
Entry	Entry parameters
Returns	Returned parameters
Modifies	Registers to be modified
Notes	(optional)

*1 - No changes from MSX1.
*2 - Calls SUBROM if screen mode is 5, 6, 7, or 8.
*3 - Calls SUBROM.
*4 - Doesn't call SUBROM. But the routine is changed for screen 4~8.
*5 - This is a BASIC statement execution entry.

2.0 MAIN ROM

2.1 RST's

Following RST's (RST 0 thru RST 5) are reserved for BASIC interpreter, RST 6 for inter-slot calls, RST 7 for hardware interrupt.

2.1.1 CHKRAM -

Address:	0000H
Name:	CHKRAM *1
Function:	Checks RAM and sets slot for command area
Entry:	None
Returns:	None
Modifies:	All
Note:	When done, a jump to INIT must be made for further initialization

2.1.2 CGTABL -

Address:	0004H
Name:	CGTABL
Function:	Points to the character generator table
Note:	This 2 byte data holds the pointer to the character generator table

2.1.3 VDP I/O Port Address -

Data read port

Address: 0006H
Name: VDP.DR
Function: Points to the VDP data read port address
Note: This 1 byte data holds the VDP data read port address

Data write port

Address: 0007H
Name: VDP.DW
Function: Points to the VDP data write port address
Note: This 1 byte data holds the VDP data write port address

```
Status read port:   data read port address + 1
Command write port: data write port address + 1
Palette write port: data write port address + 2
Indirect access port: data write port address + 3
```

2.1.4 SYNCHR -

Address: 0008H
Name: SYNCHR *1
Function: Checks if the current character pointed by [HL] is the one we want. If not, generates 'Syntax error', otherwise falls into CHRGTR.
Entry: [HL], character to be checked must be placed at the next location to this RST.
Returns: [HL] points to next character, [A] has the character.
Carry flag set if number, Z flag set if end of statement.
Modifies: [AF], [HL]

2.1.5 RDSLT -

Address: 000CH
Name: RDSLT *1
Function: Selects the appropriate slot according to the value given through registers, and reads the contents of memory from the slot.

```
Entry:          [A]  - FxxxSSPP
                       |   ||||
                       |   ||++-- primary slot   (0-3)
                       |   ++---- secondary slot   (0-3)
                       +--------- 1 if secondary slot  specified
                [HL] - address of target memory
Returns:        [A]  - contents of memory
Modifies:       [AF], [BC], [DE]
```

Note: Interrupts are disabled automatically but never enabled by this routine.

2.1.6 CHRGTR -

Address: 0010H
Name: CHRGTR *1
Function: Gets next character (or token) from BASIC text.
Entry: [HL]
Returns: [HL] points to next character, [A] has the character. Carry flag set if number, Z flag set if end of statement encountered.
Modifies: [AF], [HL]

2.1.7 WRSLT -

Address: 0014H
Name: WRSLT *1
Function: Selects the appropriate slot according to the value given through registers, and writes to the memory.
Entry:

```
[A]  - FxxxSSPP  -
       |   ||||
       |   ||++-- primary slot   (0-3)
       |   ++---- secondary slot   (0-3)
       +--------- 1 if secondary slot specified
[HL] - address of target memory
[E]  - data to be written
```

Returns: None
Modifies: [AF], [BC], [D]
Note: Interrupts are disabled automatically but never enabled by this routine.

2.1.8 OUTDO -

Address: 0018H
Name: OUTDO *2
Function: Outputs to current device
Entry: [A], PTRFIL, PRTFLG
Returns: None
Modifies: None

2.1.9 CALSLT -

Address: 001CH
Name: CALSLT *1
Function: Performs inter-slot call to specified address.
Entry:

```
[IYH] - FxxxSSPP
        |   ||||
        |   ||++-- primary slot   (0-3)
        |   ++---- secondary slot   (0-3)
        +--------- 1 if secondary slot specified
[IX] - address to call
```

Returns: Who knows?
Modifies: Who knows?

Note: Interrupts are disabled automatically but never enabled by this routine. You can never pass arguments via alternate registers of Z80 or [IX], [IY].

2.1.10 DCOMPR -

Address: 0020H
Name: DCOMPR *1
Function: Compares [HL] with [DE]
Entry: [HL], [DE]
Returns: Flags
Modifies: [AF]

2.1.11 ENASLT -

Address: 0024H
Name: ENASLT *1
Function: Selects the appropriate slot according to the value given through registers, and permanently enables the slot.
Entry:

```
[A]  - FxxxSSPP
       |   ||||
       |   ||++-- primary slot   (0-3)
       |   ++---- secondary slot   (0-3)
       +--------- 1 if secondary slot specified
[HL] - address of target memory
```

Returns: None
Modifies: All
Note: Interrupts are disabled automatically but never enabled by this routine.

2.1.12 GETYPR -

Address: 0028H
Name: GETYPR *1
Function: Returns the type of FAC
Entry: FAC
Returns: Flags
Modifies: [AF]

Note: Interrupts are disabled automatically but never enabled by this routine. You can never pass arguments via alternate registers of Z80 or [IX], [IY].

2.1.10 DCOMPR -

Address: 0020H
Name: DCOMPR *1
Function: Compares [HL] with [DE]
Entry: [HL], [DE]
Returns: Flags
Modifies: [AF]

2.1.11 ENASLT -

Address: 0024H
Name: ENASLT *1
Function: Selects the appropriate slot according to the value given through registers, and permanently enables the slot.

```
        Entry:          [A]  - FxxxSSPP
                               |   ||||
                               |   ||++-- primary slot   (0-3)
                               |   ++---- secondary slot   (0-3)
                               +--------- 1 if secondary slot
specified
                        [HL] - address of target memory
```

Returns: None
Modifies: All
Note: Interrupts are disabled automatically but never enabled by this routine.

2.1.12 GETYPR -

Address: 0028H
Name: GETYPR *1
Function: Returns the type of FAC
Entry: FAC
Returns: Flags
Modifies: [AF]

2.1.13 BASIC ROM Version Number -

Address: 002BH,002CH
Name: None
Function: The format of ID byte is as follows

```
                  002BH: b7 b6 b5 b4 b3 b2 b1 b0
                         |  |  |  |  |  |  |  |
                         |  |  |  |  +--+--+--+-- kind of character
generator
```

```
                       |  |  |  |                0:Japanese
1:International
                       |  +--+--+--------------- format of date
                       |                           0:Y-M-D 1:M-D-Y 2:D-M-
Y
                       +------------------------ frequency of interrupt
                                                       1:50Hz  0:60Hz

                002CH: b7 b6 b5 b4 b3 b2 b1 b0
                       |  |  |  |  |  |  |  |
                       |  |  |  |  +--+--+--+--- kind of keyboard
                       |  |  |  |                  0:Japan
1:International
                       |  |  |  |                  2:French    3:UK
4:DIN
                       |  |  |  |
                       +--+--+--+--------------- version of BASIC
                                                       (print using
etc.)
```

2.1.14 Reserved Area -
Address: 002DH, 002EH and 002FH
Name: None
Function: Reserved area for future expansion
Note: This area is filled with zero

2.1.15 CALLF -
Address: 0030H
Name: CALLF *1
Function: Performs farcall (i.e., inter-slot call)
Entry: None
Returns: Who knows?
Modifies: ditto
Note: Calling sequence is as follows.

```
RST     6
DB      destination slot
DW      destination address
```

For precise description about parameters, see CALSLT.

2.1.16 KEYINT -
Address: 0038H
Name: KEYINT *1
Function: Performs hardware interrupt procedures necessary.
Entry: None
Returns: None
Modifies: None
Note: Calls the extended ROM for ROMA-KANA conversion.

2.1.16 KEYINT -

Address:	0038H
Name:	KEYINT *1
Function:	Performs hardware interrupt procedures necessary.
Entry:	None
Returns:	None
Modifies:	None
Note:	Calls the extended ROM for ROMA-KANA conversion.

2.2 I/O Initialization

2.2.1 INITIO -

Address:	003BH
Name:	INITIO *1
Function:	Performs device initialization
Entry:	None
Returns:	None
Modifies:	all

2.2.2 INIFNK -

Address:	003EH
Name:	INIFNK *1
Function:	Initializes function key string contents
Entry:	None
Returns:	None
Modifies:	All

2.3 Access VDP

2.3.1 DISSCR -

Address:	0041H
Name:	DISSCR *1
Function:	Disables screen display
Entry:	None
Returns:	None
Modifies:	[AF], [BC]

2.3.2 ENASCR -

Address:	0044H
Name:	ENASCR *1
Function:	Enables screen display
Entry:	None
Returns:	None
Modifies:	[AF], [BC]

2.3.3 WRTVDP -

Address:	0047H
Name:	WRTVDP *2
Function:	Writes to VDP register
Entry:	Register in [C], data in [B] The register number can be specified 0 through 23, 32 through 46
Returns:	None
Modifies:	[AF], [BC]
Note:	Calls the extended ROM if EV bit of register 0 is changed or register 8 through 46.

2.3.4 RDVRM -

Address:	004AH
Name:	RDVRM *1
Function:	Reads VRAM addressed by [HL] : Valid A13~A0 If you want to use full bits, call NRDVRM.
Entry:	Address in [HL]
Returns:	Data in [A]
Modifies:	[AF]

2.3.5 WRTVRM -

Address:	004DH
Name:	WRTVRM *1
Function:	Writes to VRAM addressed by [HL] : Valid A13~A0 If you want to use full bits, call NWRVRM.
Entry:	Address in [HL], data in [A]
Returns:	None
Modifies:	[AF]

2.3.6 SETRD -

Address:	0050H
Name:	SETRD *1
Function:	Sets up VDP for read
Entry:	[HL] = address : Valid A13~A0 If you want to use full bits, call NSETRD.
Returns:	None
Modifies:	[AF]

2.3.7 SETWRT -

Address:	0053H
Name:	SETWRT *1
Function:	Sets up VDP for write
Entry:	[HL] = address : Valid A13~A0 If you want to use full bits, call NSTWRT.
Returns:	None
Modifies:	[AF]

2.3.8 FILVRM -

Address:	0056H
Name:	FILVRM *4
Function:	Fills VRAM with specified data
Entry:	Address in [HL] : valid A13~A0 length in [BC], data in [A] If you want to use full bits, call BIGFIL.
Returns:	None
Modifies:	[AF], [BC]

2.3.9 LDIRMV -

Address:	0059H
Name:	LDIRMV *4
Function:	Moves block of memory from VRAM to memory
Entry:	Address of source in [HL] : all bits are valid destination in [DE], length in [BC].
Returns:	None
Modifies:	All

2.3.10 LDIRVM -

Address:	005CH
Name:	LDIRVM *4
Function:	Moves block of memory from memory to VRAM.
Entry:	Address of source in [HL] destination in [DE] : all bits are valid length in [BC].
Returns:	None
Modifies:	All

2.3.11 CHGMOD -

Address: 005FH
Name: CHGMOD *3
Function: Sets VDP mode according to SCRMOD
The palette is not initialized.
If you want to initialize palette, then call CHGMDP in the extended ROM.
Entry: screen mode in [A] (0~8).
Returns: None
Modifies: All

2.3.12 CHGCLR -

Address: 0062H
Name: CHGCLR *1
Function: Changes color of screen
Entry: Mode in [A]
Foreground color in FORCLR
Background color in BAKCLR
Border color in BDRCLR
Returns: None
Modifies: All

2.3.13 NMI -

Address: 0066H
Name: NMI *1
Function: Performs non-maskable interrupt procedures
Entry: None
Returns: None
Modifies: None

2.3.14 CLRSPR -

Address: 0069H
Name: CLRSPR *3
Function: Initializes all sprites
Patterns are set to nulls, sprite names are set to sprite plane number, sprite colors are set to foreground color, vertical positions are set to 209. If in the screen 4~8, then vertical positions are set to 217.
Entry: SCRMOD
Returns: None
Modifies: All

2.3.15 INITXT -

Address:	006CH
Name:	INITXT *3
Function:	Initializes screen for text mode (40*24), sets VDP. This routine doesn't initialize palette. If you want to initialize the palette, then call INIPLT in the extended ROM after calling this.
Entry:	TXTNAM, TXTCGP
Returns:	None
Modifies:	All

2.3.16 INIT32 -

Address:	006FH
Name:	INIT32 *3
Function:	Initializes screen for text mode (32*24), sets VDP. This routine doesn't initialize palette.
Entry:	T32NAM, T32CGP, T32COL, T32ATR, T32PAT
Returns:	None
Modifies:	All

2.3.17 INIGRP -

Address:	0072H
Name:	INIGRP *3
Function:	Initializes screen for hi-resolution mode, sets VDP. This routine doesn't initialize palette.
Entry:	GRPNAM, GRPCGP, GRPCOL, GRPATR, GRPPAT
Returns:	None
Modifies:	All

2.3.18 INIMLT -

Address:	0075H
Name:	INIMLT *3
Function:	Initializes screen for multicolor mode, sets VDP. This routine doesn't initialize palette.
Entry:	MLTNAM, MLTCGP, MLTCOL, MLTATR, MLTPAT
Returns:	None
Modifies:	All

2.3.19 SETTXT -

Address:	0078H
Name:	SETTXT *3
Function:	Sets VDP for text (40*24) mode
Entry:	TXTNAM, TXTCGP
Returns:	
Modifies:	

2.3.20 SETT32 -

Address:	007BH
Name:	SETT32 *3
Function:	Sets VDP for text (32*24) mode
Entry:	T32NAM, T32CGP, T32COL, T32ATR, T32PAT
Returns:	None
Modifies:	All

2.3.21 SETGRP -

Address:	007EH
Name:	SETGRP *3
Function:	Sets VDP for hi-resolution mode
Entry:	GRPNAM, GRPCGP, GRPCOL, GRPATR, GRPPAT
Returns:	None
Modifies:	All

2.3.22 SETMLT -

Address:	0081H
Name:	SETMLT *3
Function:	Sets VDP for multicolor mode
Entry:	MLTNAM, MLTCGP, MLTCOL, MLTATR, MLTPAT
Returns:	None
Modifies:	All

2.3.23 CALPAT -

Address:	0084H
Name:	CALPAT *1
Function:	Returns address of sprite pattern table
Entry:	Sprite ID in [A]
Returns:	Address in [HL]
Modifies:	[AF], [DE], [HL]

2.3.24 CALATR -

Address:	0087H
Name:	CALATR *1
Function:	Returns address of sprite attribute table.
Entry:	Sprite ID in [A]
Returns:	Address in [HL]
Modifies:	[AF], [DE], [HL]·

2.3.25 GSPSIZ -

Address:	008AH
Name:	GSPSIZ *1
Function:	Returns current sprite size
Entry:	None
Returns:	Sprite size (number of bytes) in [A]. Carry set if 16*16 sprite in use, reset otherwise.
Modifies:	[AF]

2.3.26 GRPPRT -

Address:	008DH
Name:	GRPPRT *2
Function:	Prints a character on graphic screen
Entry:	Code to output in [A] If screen 5~8: (LOGOPR):logical operation code
Returns:	None
Modifies:	None

2.4 Access PSG

2.4.1 GICINI -

Address: 0090H
Name: GICINI *1
Function: Initializes PSG, and static data for PLAY statement.
Entry: None
Returns: None
Modifies: All

2.4.2 WRTPSG -

Address: 0093H
Name: WRTPSG *1
Function: Writes data to PSG register
Entry: Register number in [A], data in [E]
Returns: None
Modifies: None

2.4.3 RDPSG -

Address: 0096H
Name: RDPSG *1
Function: Reads data from PSG register
Entry: Register number in [A]
Returns: Data in [A]
Modifies: None

2.4.4 STRTMS -

Address: 0099H
Name: STRTMS *1
Function: Checks and starts the background task for PLAY
Entry: None
Returns: None
Modifies: All

2.5 Access Console

2.5.1 CHSNS -

Address:	009CH
Name:	CHSNS *1
Function:	Checks the status of keyboard buffer.
Entry:	None
Returns:	Z flag reset if there's any characters in buffer
Modifies:	[AF]

2.5.2 CHGET -

Address:	009FH
Name:	CHGET *1
Function:	Waits until any characters are typed, and returns with the character code.
Entry:	None
Returns:	Character code in [A]
Modifies:	[AF]

2.5.3 CHPUT -

Address:	00A2H
Name:	CHPUT *1
Function:	Outputs a character to console.
Entry:	Character code to be output in [A]
Returns:	None
Modifies:	None

2.5.4 LPTOUT -

Address:	00A5H
Name:	LPTOUT *1
Function:	Outputs a character to LPT
Entry:	Character code to be output in [A]
Returns:	Carry flag set if aborted
Modifies:	[F]

2.5.5 LPTSTT -

Address:	00A8H
Name:	LPTSTT *1
Function:	Checks line printer status
Entry:	None
Returns:	255 in [A] and Z flag reset if printer ready, 0 and Z flag set if not.
Modifies:	[AF]

2.5.6 CNVCHR -

Address: 00ABH
Name: CNVCHR *1
Function: Checks graphic header byte and converts code
Entry: Character code in [A]
Returns: Carry flag reset - graphic header byte
Carry flag set, Z flag set - converted graphic code
Carry flag set, Z flag reset - non converted code
Modifies: [AF]

2.5.7 PINLIN -

Address: 00AEH
Name: PINLIN *1
Function: Accepts a line from console until a CR or STOP is typed, and stores the line in buffer
Entry: None
Returns: Address of buffer top-1 in [HL], carry flag set if STOP is typed.
Modifies: All

2.5.8 INLIN -

Address: 00B1H
Name: INLIN *1
Function: Same as PINLIN, except in this case AUTFLG is set.
Entry: None
Returns: Address of buffer top-1 in [HL], carry flag set if STOP is pressed.
Modifies: All

2.5.9 QINLIN -

Address: 00B4H
Name: QINLIN *1
Function: Outputs a '?' mark and a space then falls into INLIN.
Entry: None
Returns: Address of buffer top-1 in [HL], carry flag set if STOP is pressed.
Modifies: All

2.5.10 BREAKX -

Address: 00B7H
Name: BREAKX *1
Function: Checks the status of Control-STOP key
Entry: None
Returns: Carry flag set if being pressed
Modifies: [AF]
Note: This routine is used to check Control-STOP when interrupts are disabled.

2.5.11 ISCNTC -

Address:	00BAH
Name:	ISCNTC *1
Function:	Checks the status of SHIFT-STOP key. If the key is pressed, BASIC returns to command mode.
Entry:	None
Returns:	None
Modifies:	None

2.5.12 CKCNTC -

Address:	00BDH
Name:	CKCNTC *1
Function:	Checks the status of SHIFT-STOP key. If the key is pressed, BASIC returns to command mode.
Entry:	None
Returns:	None
Modifies:	None

2.5.13 BEEP -

Address:	00C0H
Name:	BEEP *3
Function:	Beeps buzzer
Entry:	None
Returns:	None
Modifies:	All

2.5.14 CLS -

Address:	00C3H
Name:	CLS *3
Function:	Clears screen
Entry:	Zero flag must be set.
Returns:	None
Modifies:	[AF], [BC], [DE]

2.5.15 POSIT -

Address:	00C6H
Name:	POSIT *1
Function:	Locates cursor at specified position.
Entry:	Column in [H], row in [L]
Returns:	None
Modifies:	[AF]

2.5.16 FNKSB -

Address:	00C9H
Name:	FNKSB *1
Function:	Checks if function key display is active. If so, displays it, otherwise erases it.
Entry:	FNKFLG
Returns:	None
Modifies:	All

2.5.17 ERAFNK -

Address:	00CCH
Name:	ERAFNK *1
Function:	Erases function key display
Entry:	None
Returns:	None
Modifies:	All

2.5.18 DSPFNK -

Address:	00CFH
Name:	DSPFNK *2
Function:	Displays function key display
Entry:	None
Returns:	None
Modifies:	All

2.5.19 TOTEXT -

Address:	00D2H
Name:	TOTEXT *1
Function:	Forces screen to text mode
Entry:	None
Returns:	None
Modifies:	All
Note:	TOTEXT is not changed. But this routine calls CHGMDP. So calls the extended ROM.

2.6 Access Game I/O

2.6.1 GTSTCK -

Address:	00D5H
Name:	GTSTCK *1
Function:	Returns the current status of joy-stick
Entry:	Joy-stick ID in [A]
Returns:	Direction in [A]
Modifies:	All

2.6.2 GTTRIG -

Address:	00D8H
Name:	GTTRIG *1
Function:	Returns the current status of trigger button
Entry:	Trigger button ID in [A]
Returns:	Returns 0 in [A] if not pressed, 255 otherwise.
Modifies:	[AF]

2.6.3 GTPAD -

Address:	00DBH
Name:	GTPAD *1
Function:	Checks current status of touch PAD
Entry:	ID in [A]
Returns:	Value in [A]
Modifies:	All

2.6.4 GTPDL -

Address:	00DEH
Name:	GTPDL *2
Function:	Returns the value of paddle
Entry:	Paddle ID in [A]
Returns:	Value in [A]
Modifies:	All

2.7 Access Cassette Tape

2.7.1 TAPION -

Address:	00E1H
Name:	TAPION *1
Function:	Turns motor on and reads header from tape
Entry:	None
Returns:	Carry flag set if aborted
Modifies:	All

2.7.2 TAPIN -

Address:	00E4H
Name:	TAPIN *1
Function:	Inputs from tape
Entry:	None
Returns:	Data in [A], carry flag set if aborted.
Modifies:	All

2.7.3 TAPIOF -

Address:	00E7H
Name:	TAPIOF *1
Function:	Stops reading from tape
Entry:	None
Returns:	None
Modifies:	None

2.7.4 TAPOON -

Address:	00EAH
Name:	TAPOON *1
Function:	Turns motor on and writes header block to cassette.
Entry:	[A] holds non-0 value if a long header desired, 0 if a short header desired.
Returns:	Carry flag set if aborted
Modifies:	All

2.7.5 TAPOUT -

Address:	00EDH
Name:	TAPOUT *1
Function:	Outputs to tape
Entry:	Data to be output in [A]
Returns:	Carry flag set if aborted
Modifies:	All

2.7.6 TAPOOF -

Address:	00F0H
Name:	TAPOOF *1
Function:	Stops writing to tape
Entry:	None
Returns:	Carry flag set if aborted
Modifies:	None

2.7.7 STMOTR -

Address:	00F3H
Name:	STMOTR *1
Function:	Sets cassette motor
Entry:	0 in [A] to stop, 1 to start, 255 to flip.
Returns:	None
Modifies:	[AF]

2.8 Handle Queue

2.8.1 LFTQ -

Address:	00F6H
Name:	LFTQ *1
Function:	Returns how many bytes are left in queue
Entry:	
Returns:	
Modifies:	

2.8.2 PUTQ -

Address:	00F9H
Name:	PUTQ *1
Function:	Puts a byte in queue
Entry:	
Returns:	
Modifies:	

2.9 Low Level Graphics

2.9.1 RIGHTC -

Address:	00FCH
Name:	RIGHTC *2
Function:	Moves one pixel right
Entry:	None
Returns:	None
Modifies:	[AF]
Note:	If the screen mode isn't hi-res, calls the extended ROM.

2.9.2 LEFTC -

Address:	00FFH
Name:	LEFTC *2
Function:	Moves one pixel left
Entry:	None
Returns:	None
Modifies:	[AF]
Note:	If the screen mode isn't hi-res, calls the extended ROM.

2.9.3 UPC -

Address:	0102H
Name:	UPC *2
Function:	Moves one pixel up
Entry:	None
Returns:	None
Modifies:	[AF]
Note:	If the screen mode isn't hi-res, then calls the extended ROM.

2.9.4 TUPC -

Address:	0105H
Name:	TUPC *2
Function:	Moves one pixel up
Entry:	None
Returns:	None
Modifies:	[AF]
Note:	If the screen mode isn't hi-res, calls the extended ROM.

2.9.5 DOWNC -

Address: 0108H
Name: DOWNC *2
Function: Moves one pixel down
Entry: None
Returns: None
Modifies: [AF]
Note: If the screen mode isn't hi-res, calls the extended ROM.

2.9.6 TDOWNC -

Address: 010BH
Name: TDOWNC *2
Function: Moves one pixel down
Entry: None
Returns: None
Modifies: [AF]
Note: If the screen mode isn't hi-res, calls the extended ROM.

2.9.7 SCALXY -

Address: 010EH
Name: SCALXY *2
Function: Scales X Y coordinates
Entry: Horizontal position is [BC], vertical position
is
[DE]
Returns: Clipped horizontal position is [BC],
Clipped vertical position is [DE]
Modifies: [AF]

2.9.8 MAPXYC -

Address: 0111H
Name: MAPXYC *2
Function: Maps coordinate to physical address
Entry: Horizontal position is [BC], vertical position is [DE]
Returns: In screen 2 to 4
  Physical address in [HL], Mask pattern in [A]
In screen 5 to 8
  horizontal position in [HL], vertical position in [A]
Modifies: [F]
Note: Calls the extended ROM when the screen is in multi color mode.

2.9.9 FETCHC -

Address: 0114H
Name: FETCHC *1
Function: Fetches current physical address and mask pattern.
Entry: None
Returns: Address in [HL], mask pattern in [A]
Modifies: [F]

2.9.10 STOREC -

Address: 0117H
Name: STOREC *1
Function: Stores physical address and mask pattern
Entry: Address in [HL], mask pattern in [A]
Returns: None
Modifies: None

2.9.11 SETATR -

Address: 011AH
Name: SETATR *4
Function: Sets attribute byte
Entry: Attribute code in [A]
Returns: Carry flag is set if illegal value
Modifies: [F]
Note: Works only in screen mode 0 to 4

2.9.12 READC -

Address: 011DH
Name: READC *2
Function: Reads attribute of current pixel
Entry: None
Returns: Attribute code in [A]
Modifies: [F]
Note: Calls the extended ROM when the screen is in multi color mode and bit map mode.

2.9.13 SETC -

Address: 0120H
Name: SETC *2
Function: Sets current pixel to specified attribute
Entry: None
Returns: None
Modifies: [AF]
Note: Calls the extended ROM when the screen is in multi color mode.

2.9.14 NSETCX -

Address:	0123H
Name:	NSETCX *1
Function:	Sets pixels horizontally
Entry:	Count in [HL]
Returns:	None
Modifies:	All

2.9.15 GTASPC -

Address:	0126H
Name:	GTASPC *1
Function:	Returns aspect ratio
Entry:	None
Returns:	[DE], [HL]
Modifies:	None

2.9.16 PNTINI -

Address:	0129H
Name:	PNTINI *1
Function:	Initializes for PAINT
Entry:	None
Returns:	None
Modifies:	[AF]

2.9.17 SCANR -

Address:	012CH
Name:	SCANR *2
Function:	Scans pixels to right
Entry:	Suspend flag in [B], border count in [DE]
Returns:	Border count in [DE], pixel changed flag in [C]
Modifies:	All
Note:	Calls the extended ROM when the screen is in multi color mode and bit map mode.

2.9.18 SCANL -

Address:	012FH
Name:	SCANL *2
Function:	Scans pixels to left
Entry:	Border count in [DE]
Returns:	Border count in [DE], pixel changed flag in [C]
Modifies:	All
Note:	Calls the extended ROM when the screen is in multi color mode and bit map mode.

2.10 Additional Entries

2.10.1 CHGCAP -

Address:	0132H
Name:	CHGCAP *1
Function:	Changes the status of CAP lamp
Entry:	0 in [A] to turn off the lamp, non 0 otherwise.
Returns:	None
Modifies:	[AF]

2.10.2 CHGSND -

Address:	0135H
Name:	CHGSND *1
Function:	Changes the status of 1 bit sound port.
Entry:	0 in [A] to turn off, non 0 otherwise.
Returns:	None
Modifies:	[AF]

2.10.3 RSLREG -

Address:	0138H
Name:	RSLREG *1
Function:	Reads what is currently output to primary slot register.
Entry:	None
Returns:	Result in [A]
Modifies:	None

2.10.4 WSLREG -

Address:	013BH
Name:	WSLREG *1
Function:	Writes to primary slot register.
Entry:	Value in [A]
Returns:	None
Modifies:	None

2.10.5 RDVDP -

Address:	013EH
Name:	RDVDP *1
Function:	Reads VDP's status register.
Entry:	None
Returns:	Data in [A]
Modifies:	None

2.10.6 SNSMAT -

Address: 0141H
Name: SNSMAT *1
Function: Returns the status of specified row of a keyboard matrix.
Entry: Row in [A]
Returns: Status in [A], corresponding bit is reset to 0 if being pressed.
Modifies: [AF], [C]

2.10.7 PHYDIO -

Address: 0144H
Name: PHYDIO *1
Function: Performs operation for mass storage devices (such as disks).
Entry: ???
Returns: ???
Modifies: ???
Note: In minimum configuration, only a hook is provided.

2.10.8 FORMAT -

Address: 0147H
Name: FORMAT *1
Function: Performs mass storage devices initialization.
Entry: ???
Returns: ???
Modifies: ???
Note: In minimum configuration, only a hook is provided.

2.10.9 ISFLIO -

Address: 014AH
Name: ISFLIO *1
Function: Checks if we're doing device I/O
Entry: None
Returns: Non zero if so, zero otherwise
Modifies: [AF]

2.10.10 OUTDLP -

Address: 014DH
Name: OUTDLP *1
Function: Outputs to LPT
Entry: Code in [A]
Returns: None
Modifies: [F]

Note:	This entry differs from LPTOUT in that: 1) TABs are expanded to spaces, 2) HIRAGANA and graphics symbols are converted when non-MSX printer is in use, 3) a jump to 'device I/O error' is made when aborted.

2.10.11 GETVCP -

Address:	0150H
Name:	GETVCP *1
Function:	Get pointer to music queue
Entry:	Channel number in [A]
Returns:	Pointer in [HL]
Modifies:	[AF]
Note:	Only used to play music as the background task.

2.10.12 GETVC2 -

Address:	0153H
Name:	GETVC2 *1
Function:	Get pointer to desired variable for voice VOICEN
Entry:	Desired displacement into voice buffer in [L]
Returns:	Pointer in [HL]
Modifies:	[AF]
Note:	Only used to play music as the background task.

2.10.13 KILBUF -

Address:	0156H
Name:	KILBUF *1
Function:	Clears keyboard buffer
Entry:	None
Returns:	None
Modifies:	[HL]

2.10.14 CALBAS -

Address:	0159H
Name:	CALBAS *1
Function:	Performs farcall (i.e., inter-slot call) into BASIC interpreter.
Entry:	Address in [IX]
Returns:	Who knows?
Modifies:	ditto

2.11 MSX2 Additional Entries

2.11.1 SUBROM -

Address:	015CH
Name:	SUBROM
Function:	Performs farcall (i.e., inter-slot call) into SUBROM.
Entry:	Address in [IX], saved [IX] on stack
Returns:	Who knows ?
Modifies:	alternative registers, [IY]

2.11.2 EXTROM -

Address:	015FH
Name:	EXTROM
Function:	Performs farcall (i.e., inter-slot call) into SUBROM.
Entry:	Address in [IX]
Returns:	Who knows?
Modifies:	alternative registers, [IY]

2.11.3 CHKSLZ -

Address:	0162H
Name:	CHKSLZ
Function:	does slot scan for SUBROM
Entry:	None
Returns:	None
Modifies:	All

2.11.4 CHKNEW -

Address:	0165H
Name:	CHKNEW
Function:	Check screen mode
Entry:	None
Return:	Carry flag is reset if screen 5 to 8
Modifies:	[AF]

2.11.5 EOL -

Address:	0168H
Name:	EOL
Function:	Erase to-end-of line
Entry:	Column number in [H], line number in [L] Cursor should remain unchanged
Returns:	None
Modifies:	All

2.11.6 BIGFIL -

Address: 016BH
Name: BIGFIL
Function: Same as FILVRM except for the following
FILVRM checks if the current screen mode is 0,1,2 or 3. If so, it behaves like the VDP has only 16K VRAM. This is to maintain compatibility with MSX1. BIGFIL, however, does not check the screen mode, and fills the VRAM just as specified by the parameters.
Entry: Same as FILVRM
Returns: Same as FILVRM
Modifies: Same as FILVRM

2.11.7 NSETRD -

Address: 016EH
Name: NSETRD
Function: Set-up VDP to read
Entry: Address in [HL]: Valid all bits
Returns: None
Modifies: [AF]

2.11.8 NSTWRT -

Address: 0171H
Name: NSTWRT
Function: Set-up VDP to write
Entry: Address in [HL]: Valid all bits
Returns: None
Modifies: [AF]

2.11.9 NRDVRM -

Address: 0174H
Name: NRDVRM
Function: Reads VRAM addressed by [HL]:Valid all bits
Entry: Address in [HL]
Returns: Data in [A]
Modifies: [F]

2.11.10 NWRVRM -

Address: 0177H
Name: NWRVRM
Function: Writes [A] to VRAM addressed by [HL]:Valid all bits
Entry: Address in [HL], data in [A]
Returns: None
Modifies: [AF]

3.0 EXTENDED ROM

How to call the extended ROM.

```
                .
        LD      IX,INIPLT
        CALL    EXTROM
                .               ;Returns here
        or
                .
INIPAL:
        PUSH    IX
        LD      IX,INIPLT
        JP      SUBROM          ;Returns caller of INIPAL
                .
        or
                .
        LD      IY,(EXBRSA-1)   ;get slot address of extended ROM
        LD      IX,INIPLT
        CALL    CALSLT
                .
```

3.1 Graphics Handler For BASIC

3.1.1 PAINT -

Address: 0069H
Name: PAINT *5
Function: Paints the graphic screen
Entry: [HL] has text pointer to BASIC token
Returns: [HL] has updated text pointer
Modifies: All
Note: for screen mode 5, 6, 7 or 8

3.1.2 PSET -

Address: 006DH
Name: PSET *5
Function: Sets the point
Entry: [HL] has text pointer to BASIC token
Returns: [HL] has updated text pointer
Modifies: All
Note: for screen mode 5, 6, 7 or 8

3.1.3 ATRSCN -

Address: 0071H
Name: ATRSCN *5
Function: Scans color attribute
Entry: [HL] has text pointer to BASIC token
Returns: [HL] has updated text pointer
Modifies: All

Note:	for screen mode 5, 6, 7 or 8

3.1.4 GLINE -

Address:	0075H
Name:	GLINE *5
Function:	Draws a line
Entry:	[HL] has text pointer to BASIC token
Returns:	[HL] has updated text pointer
Modifies:	All
Note:	for screen mode 5, 6, 7 or 8

3.1.5 DOBOXF -

Address:	0079H
Name:	DOBOXF *5
Function:	Draws a filled box
Entry:	[HL] has text pointer to BASIC token Start coordinate is ([BC],[DE]) End coordinate is (GXPOS, GYPOS)
Returns:	[HL] has updated text pointer
Modifies:	All
Note:	for screen mode 5, 6, 7 or 8

3.1.6 DOLINE -

Address:	007DH
Name:	DOLINE *5
Function:	Draws a line
Entry:	[HL] has text pointer to BASIC token Start coordinate is ([BC],[DE]) End coordinate is (GXPOS, GYPOS)
Returns:	[HL] has updated text pointer
Modifies:	All
Note:	for screen mode 5, 6, 7 or 8

3.1.7 BOXLIN -

Address:	0081H
Name:	BOXLIN *5
Function:	Draws a box
Entry:	[HL] has text pointer to BASIC token Start coordinate is ([BC],[DE]) End coordinate is (GXPOS, GYPOS)
Returns:	[HL] has updated text pointer
Modifies:	All
Note:	for screen mode 5, 6, 7 or 8

3.2 Low Level Graphics

3.2.1 DOGRPH -

Address:	0085H
Name:	DOGRPH
Function:	Draw a line
Entry:	Start coordinate in ([BC],[DE]) End coordinate in (GXPOS, GYPOS) Attribute in (ATRBYT) Logical operation code in (LOGOPR)
Returns:	None
Modifies:	[AF]
Note:	for screen mode 5, 6, 7 or 8

3.2.2 GRPPRT -

Address:	0089H
Name:	GRPPRT
Function:	Prints a character on graphic screen
Entry:	Code to output in [A] Attribute in (ATRBUT) Logical operation code is (LOGOPR)
Returns:	None
Modifies:	None
Note:	for screen mode 5, 6, 7 or 8

3.2.3 SCALXY -

Address:	008DH
Name:	SCALXY
Function:	Scales X Y coordinate
Entry:	Horizontal position is [BC], Vertical position is [DE]
Returns:	Clipped horizontal position is [BC], Clipped vertical position is [DE]
Modifies:	[AF]

3.2.4 MAPXYC -

Address:	0091H
Name:	MAPXYC
Function:	Maps coordinate to physical address
Entry:	The coordinate is ([BC], [DE])

Returns:	Screen mode 3
	  VRAM address is [HL] and (CLOC)
	  Bit mask is [A] and (CMASK)
	Screen mode 5, 6, 7 or 8
	  Horizontal position is [HL] and (CLOC)
	  Vertical position is [A] and (CMASK)

Modifies:	[F]
Note:	for screen mode 3, 5, 6, 7 or 8

3.2.5 READC -

Address:	0095H
Name:	READC
Function:	Reads attribute of current pixel
Entry:	Coordinate is (CLOC) and (CMASK)
Returns:	The attribute is [A]
Modifies:	[AF]
Note:	for screen mode 3, 5, 6, 7 or 8

3.2.6 SETATR -

Address:	0099H
Name:	SETATR
Function:	Sets attribute byte
Entry:	Attribute is [A]
Returns:	Carry flag is set if illegal attribute
Modifies:	[F]

3.2.7 SETC -

Address:	009DH
Name:	SETC
Function:	Sets current pixel to specified attribute
Entry:	Coordinate is (CLOC) and (CMASK) Attribute is (ATRBYT)
Returns:	None
Modifies:	[AF]
Note:	for screen mode 3, 5, 6, 7 or 8

3.2.8 TRIGHT -

Address:	00A1H
Name:	TRIGHT
Function:	Moves one pixel right
Entry:	Coordinate is (CLOC) and (CMASK)
Returns:	Updated coordinate is (CLOC) and (CMASK) Carry flag is set if the coordinate is on the edge of the screen
Modifies:	[AF]
Note:	for screen mode 3 only

3.2.9 RIGHTC -

Address:	00A5H
Name:	RIGHTC
Function:	Moves one pixel right
Entry:	Coordinate is (CLOC) and (CMASK)
Returns:	Updated coordinate is (CLOC) and (CMASK)
Modifies:	[AF]
Note:	for screen mode 3 only

3.2.10 TLEFTC -

Address:	00A9H
Name:	TLEFTC
Function:	Moves one pixel left
Entry:	Coordinate is (CLOC) and (CMASK)
Returns:	Updated coordinate is (CLOC) and (CMASK) Carry flag is set if the coordinate is on the edge of the screen
Modifies:	[AF]
Note:	for screen mode 3, 5, 6, 7 or 8

3.2.11 LEFTC -

Address:	00ADH
Name:	LEFTC
Function:	Moves one pixel left
Entry:	Coordinate is (CLOC) and (CMASK)
Returns:	Updated coordinate is (CLOC) and (CMASK)
Modifies:	[AF]
Note:	for screen mode 3 only

3.2.12 TDOWNC -

Address:	00B1H
Name:	TDOWNC
Function:	Moves one pixel down
Entry:	Coordinate is (CLOC) and (CMASK)
Returns:	Updated coordinate is (CLOC) and (CMASK) Carry flag is set if the coordinate is on the edge of the screen
Modifies:	[AF]
Note:	for screen mode 3, 5, 6, 7 or 8

3.2.13 DOWNC -

Address:	00B5H
Name:	DOWNC
Function:	Moves one pixel down
Entry:	Coordinate is (CLOC) and (CMASK)
Returns:	Updated coordinate is (CLOC) and (CMASK)
Modifies:	[AF]
Note:	for screen mode 3 only

3.2.14 TUOC -

Address:	00B9H
Name:	TUPC
Function:	Moves one pixel up
Entry:	Coordinate is (CLOC) and (CMASK)
Returns:	Updated coordinate is (CLOC) and (CMASK)

	Carry flag is set if the coordinate is on the edge of the screen
Modifies:	[AF]
Note:	for screen mode 3, 5, 6, 7 or 8

3.2.15 UPC -

Address:	00BDH
Name:	UPC
Function:	Moves one pixel up
Entry:	Coordinate is (CLOC) and (CMASK)
Returns:	Updated coordinate is (CLOC) and (CMASK)
Modifies:	[AF]
Note:	for screen mode 3 only

3.2.16 SCANR -

Address:	00C1H
Name:	SCANR
Function:	Scans pixels to right
Entry:	Suspend flag in [B], border count in [DE]
Returns:	Border count in [DE], pixel changed flag in [C]
Modifies:	All
Note:	for screen mode 3, 5, 6, 7 or 8

3.2.17 SCANL -

Address:	00C5H
Name:	SCANL
Function:	Scans pixels to left
Entry:	Border count in [DE]
Returns:	Border count in [DE], pixel changed flag in [C]
Modifies:	All
Note:	Works on screens 5~8 and multi color mode

3.2.18 NVBXLN -

Address:	00C9H
Name:	NVBXLN
Function:	Draws a box
Entry:	Start coordinate in ([BC],[DE]) End coordinate in ((GXPOS),(GYPOS)) Attribute code in (ATRBYT) Logical operation code in (LOGOPR)
Returns:	None
Modifies:	All
Note:	Works on screen 5, 6, 7 or 8

3.2.19 NVBXFL -

Address:	00CDH
Name:	NVBXFL
Function:	Draws a filled box
Entry:	Start coordinate in ([BC],[DE]) End coordinate in ((GXPOS),(GYPOS)) Attribute code in (ATRBYT) Logical operation code in (LOGOPR)
Returns:	None
Modifies:	All
Note:	Works on screen 5, 6, 7 or 8

3.3 Access VDP

3.3.1 CHGMOD -

Address:	00D1H
Name:	CHGMOD
Function:	Sets VDP mode according to SCRMOD
Entry:	Screen mode in [A] (0~8)
Returns:	None
Modifies:	All

3.3.2 INITXT -

Address:	00D5H
Name:	INITXT
Function:	Initializes screen for text mode (40*24), sets VDP.
Entry:	TXTNAM, TXTCGP
Returns:	None
Modifies:	All

3.3.3 INIT32 -

Address:	00D9H
Name:	INIT32
Function:	Initializes screen for text mode (32*24), sets VDP.
Entry:	T32NAM, T32CGP, T32COL, T32ATR, T32PAT
Returns:	None
Modifies:	All

3.3.4 INIGRP -

Address:	00DDH
Name:	INIGRP
Function:	Initializes screen for hi-resolution mode, sets VDP.
Entry:	GRPNAM, GRPCGP, GRPCOL, GRPATR, GRPPAT
Returns:	None
Modifies:	All

3.3.5 INIMLT -

Address:	00E1H
Name:	INIMLT
Function:	Initializes screen for multicolor mode, sets VDP.
Entry:	MLTNAM, MLTCGP, MLTCOL, MLTATR, MLTPAT
Returns:	None
Modifies:	All

3.3.6 SETTXT -

Address:	00E5H
Name:	SETTXT
Function:	Sets VDP for text (40*24) mode
Entry:	TXTNAM, TXTCGP
Returns:	None
Modifies:	All

3.3.7 SETT32 -

Address:	00E9H
Name:	SETT32
Function:	Sets VDP for text (32*24) mode
Entry:	T32NAM, T32CGP, T32COL, T32ATR, T32PAT
Returns:	None
Modifies:	All

3.3.8 SETGRP -

Address:	00EDH
Name:	SETGRP
Function:	Sets VDP for hi-resolution mode
Entry:	GRPNAM, GRPCGP, GRPCOL, GRPATR, GRPPAT
Returns:	None
Modifies:	All

3.3.9 SETMLT -

Address:	00F1H
Name:	SETMLT
Function:	Sets VDP for multicolor mode
Entry:	MLTNAM, MLTCGP, MLTCOL, MLTATR, MLTPAT
Returns:	None
Modifies:	All

3.3.10 CLRSPR -

Address:	00F5H
Name:	CLRSPR
Function:	Initializes all sprites Patterns are set to nulls, sprite names are set to sprite plane number, sprite colors are set to foreground color, vertical positions are set to 217.
Entry:	(SCRMOD)
Returns:	None
Modifies:	All

3.3.11 CALPAT -

Address: 00F9H
Name: CALPAT
Function: Returns address of sprite pattern table
Entry: Sprite ID in [A]
Returns: Address in [HL]
Modifies: [AF], [DE], [HL]
Note: This routine is equivalent to MSX1 BIOS.

3.3.12 CALATR -

Address: 00FDH
Name: CALATR
Function: Returns address of sprite attribute table.
Entry: Sprite ID in [A]
Returns: Address in [HL]
Modifies: [AF], [DE], [HL]
Note: This routine is equivalent to MSX1 BIOS.

3.3.13 GSPSIZ -

Address: 0101H
Name: GSPSIZ
Function: Returns current sprite size
Entry: None
Returns: Sprite size (number of bytes) in [A].
Carry set if 16*16 sprite in use, reset otherwise.
Modifies: [AF]
Note: This routine is equivalent to MSX1 BIOS.

3.3.14 GETPAT -

Address: 0105H
Name: GETPAT
Function: Returns a character pattern
Entry: ASCII character code in [A]
Returns: Character pattern in (PATWRK)
Modifies: All
Note: This routine is equivalent to MSX1 BIOS.

3.3.15 WRTVRM -

Address: 0109H
Name: WRTVRM
Function: Writes to VRAM addressed by [HL]
Entry: Address in [HL], data in [A]
Returns: None
Modifies: [AF]
Note: Supports 0~0FFFFH address.

3.3.16 RDVRM -

Address:	010DH
Name:	RDVRM
Function:	Reads VRAM addressed by [HL]
Entry:	Address in [HL]
Returns:	Data in [A]
Modifies:	[AF]
Note:	Supports 0~0FFFFH address.

3.3.17 CHGCLR -

Address:	0111H
Name:	CHGCLR
Function:	Changes color of screen
Entry:	Mode in [A] Foreground color in FORCLR Background color in BAKCLR Border color in BDRCLR
Returns:	None
Modifies:	All

3.3.18 CLS -

Address:	0115H
Name:	CLS
Function:	Clears screen
Entry:	None
Returns:	None
Modifies:	All

3.3.19 CLRTXT -

Address:	0119H
Name:	CLRTXT
Function:	Clears the text screen
Entry:	None
Returns:	None
Modifies:	All

3.3.20 DSPFNK -

Address:	011DH
Name:	DSPFNK
Function:	Displays function key display
Entry:	None
Returns:	None
Modifies:	All

3.3.21 DELLNO -

Address:	0121H
Name:	DELLNO
Function:	Deletes a line in text mode
Entry:	Line number in [L]
Returns:	None
Modifies:	All

3.3.22 INSLNO -

Address:	0125H
Name:	INSLNO
Function:	Inserts a line in text mode
Entry:	Line number in [L]
Returns:	None
Modifies:	All

3.3.23 PUTVRM -

Address:	0129H
Name:	PUTVRM
Function:	Put a character in text screen
Entry:	Column number in [H], line number in [L]
Returns:	None
Modifies:	[AF]

3.3.24 WRTVDP -

Address:	012DH
Name:	WRTVDP
Function:	Writes to VDP register
Entry:	Register in [C], data in [B]
Returns:	None
Modifies:	[AF], [BC]

3.3.25 VDPSTA -

Address:	0131H
Name:	VDPSTA
Function:	Read VDP status
Entry:	Status register in [A] (0~9)
Returns:	Data in [A]
Modifies:	[F]

3.4 Handle ROMA-KANA Conversion

3.4.1 KYKLOK -

Address:	0135H
Name:	KYKLOK
Function:	Handle kana key and lamp
Entry:	None
Returns:	None
Modifies:	[AF]

3.4.2 PUTCHR -

Address:	0139H
Name:	PUTCHR
Function:	Get a key code from keyboard, and convert it to kana-character, and put it into the buffer.
Entry:	Zero flag is set if no convert mode
Returns:	None
Modifies:	All

3.5 Access VDP

3.5.1 SETPAG -

Address:	013DH
Name:	SETPAG
Function:	Set VDP registers to page changes
Entry :	(ACPAGE),(DPPAGE)
Returns:	None
Modifies:	[AF]

3.6 Access Palette

VDP's palette has 3 colors (red, green, blue). Each color has 3 bits to display intensity of the color.
The current palette is saved in VRAM, because we can't read the palette from VDP.

3.6.1

Address:	0141H
Name:	INIPLT
Function:	Initialize palette and VRAM for palette saved area
Entry:	None
Returns:	None
Modifies:	[AF], [BC], [DE]

3.6.2 RSTPLT -

Address:	0145H
Name:	RSTPLT
Function:	Restore palette from VRAM
Entry:	None
Returns:	None
Modifies:	[AF], [BC], [DE]

3.6.3 GETPLT -

Address:	0149H
Name:	GETPLT
Function:	Get color codes from palette
Entry:	Palette in [A] (0~15)
Returns:	RED code in higher 4 bits of [B] BLUE code in lower 4 bits of [B] GREEN code in lower 4 bits of [C]
Modifies:	[AF], [DE]

3.6.4 SETPLT -

Address:	014DH
Name:	SETPLT
Function:	Set color codes to palette
Entry:	Palette in [D] (0~15) RED in higher 4 bits of [A] BLUE in lower 4 bits of [A] GREEN in lower 4 bits of [E]
Returns:	None
Modifies:	[AF]

3.7 BASIC Extended Statement

3.7.1 PUTSPR -

Address:	0151H
Name:	PUTSPR
Function:	Put sprites
Entry:	text pointer in [HL]
Returns:	updated text pointer in [HL]
Modifies:	All

3.7.2 COLOR -

Address:	0155H
Name:	COLOR
Function:	Change screen color, sprite color, palette
Entry:	text pointer in [HL]
Returns:	updated text pointer in [HL]
Modifies:	All

3.7.3 SCREEN -

Address:	0159H
Name:	SCREEN
Function:	Change screen mode
Entry:	text pointer in [HL]
Returns:	updated text pointer in [HL]
Modifies:	All

3.7.4 WIDTHS -

Address:	015DH
Name:	WIDTHS
Function:	Change text screen width
Entry:	text pointer in [HL]
Returns:	updated text pointer in [HL]
Modifies:	All

3.7.5 VDP -

Address:	0161H
Name:	VDP
Function:	Set VDP register
Entry:	text pointer in [HL]
Returns:	updated text pointer in [HL]
Modifies:	All

3.7.6 VDPF -

Address:	0165H
Name:	VDPF
Function:	Read current VDP register
Entry:	text pointer in [HL]
Returns:	updated text pointer in [HL]
Modifies:	All

3.7.7 BASE -

Address:	0169H
Name:	BASE
Function:	Set VDP base registers
Entry:	text pointer in [HL]
Returns:	updated text pointer in [HL]
Modifies:	All

3.7.8 BASEF -

Address:	016DH
Name:	BASEF
Function:	Read VDP base registers
Entry:	text pointer in [HL]
Returns:	updated text pointer in [HL]
Modifies:	All

3.7.9 VPOKE -

Address:	0171H
Name:	VPOKE
Function:	Write a byte to VRAM
Entry:	text pointer in [HL]
Returns:	updated text pointer in [HL]
Modifies:	All

3.7.10 VPEEK -

Address:	0175H
Name:	VPEEK
Function:	Read a byte from VRAM
Entry:	text pointer in [HL]
Returns:	updated text pointer in [HL]
Modifies:	All

3.7.11 SETS -

Address:	0179H
Name:	SETS
Function:	Sets beep sound, screen adjust, time and date
Entry:	text pointer in [HL]
Returns:	updated text pointer in [HL]
Modifies:	All

3.9 Restore Screen

3.9.1 SDFSCR -

Address:	0185H
Name:	SDFSCR
Function:	Restore screen related parameters from RAM on clock chip which is battery backed-up. Does not display function key if carry is reset on entry for call from DOS.
Entry:	Carry is reset on entry for call from DOS.
Returns:	None
Modifies:	All

3.9.2 SETSCR -

Address:	0189H
Name:	SETSCR
Function:	Restore screen and print opening message
Entry:	None
Returns:	None
Modifies:	All

3.10 VRAM Data Transfer Function

3.10.1 SCOPY -

Address:	018DH
Name:	SCOPY
Function:	Copies VRAM, array and disk file
Entry:	Text pointer in [HL]
Returns:	Updated text pointer in [HL]
Modifies:	All

3.10.2 BLTVV -

Address:	0191H
Name:	BLTVV
Function:	Copies VRAM to VRAM
Entry:	[HL] = 0F562H
Returns:	None
Modifies:	All

3.10.3 BLTVM -

Address:	0195H
Name:	BLTVM
Function:	Copies array to VRAM
Entry:	[HL] = 0F562H
Returns:	None
Modifies:	All

3.10.4 BLTMV -

Address:	0199H
Name:	BLTMV
Function:	Copies VRAM to array
Entry:	[HL] = 0F562H
Returns:	None
Modifies:	All

3.10.5 BLTVD -

Address:	019DH
Name:	BLTVD
Function:	Copies DISK file to VRAM
Entry:	[HL] = 0F562H
Returns:	None
Modifies:	All

3.10.6 BLTDV -

Address:	01A1H
Name:	BLTDV
Function:	Copies VRAM to DISK file
Entry:	[HL] = 0F562H
Returns:	None
Modifies:	All

3.10.7 BLTMD -

Address:	01A5H
Name:	BLTMD
Function:	Loads array data from DISK file
Entry:	[HL] = 0F562H
Returns:	None
Modifies:	All

3.10.8 BLTDM -

Address:	01A9H
Name:	BLTDM
Function:	Saves array data to DISK file
Entry:	[HL] = 0F562H
Returns:	None
Modifies:	All

3.11 Mouse And Track Ball

3.11.1 NEWPAD -

Address: 01ADH
Name: NEWPAD
Function: Read paddle, mouse and track ball
Entry: [A]
Mouse, Cat and Light pen support routines. Entered via GTPAD entry in the BIOS header (00DBH).

8	Sample light pen	(255 if valid)
9	Return X coordinate	
10	Return Y coordinate	
11	Return pen switch status	(255 if pressed)
12	Sample mouse/cat connected to port 1	(always 255)
13	Return X offset	
14	Return Y offset	
15	None	(always 0)
16	Sample mouse/cat connected to port 2	(always 255)
17	Return X offset	
18	Return Y offset	
19	None	(always 0)

Returns: Value in [A]
Modifies: All

3.12 Miscellaneous

3.12.1 GETPUT -

Address:	01B1H
Name:	GETPUT
Function:	GET TIME and GET DATE and PUT KANJI
Entry:	Text pointer in [HL]
Returns:	Updated text pointer in [HL]
Modifies:	All

3.12.2 CHGMDP -

Address:	01B5H
Name:	CHGMDP
Function:	Sets VDP mode according to SCRMOD The palette is initialized.
Entry:	screen mode in [A] (0~8).
Returns:	None
Modifies:	All

3.12.3 RESV1 -

Address:	01B9H
Name:	RESV1
Function:	Not used. Reserved.
Entry:	
Returns:	
Modifies:	

3.13 Kanji Print (KNJPRT)

Address:	01BDH
Name:	KNJPRT
Function:	Put a kanji character to graphic screen(5~8)
Entry:	[BC] = JIS kanji character code [A] = display mode (0 = full, 1 = even, 2 = odd)
Returns:	None
Modifies:	[AF]

3.14 Access Clock Chip

3.14.1 REDCLK -

Address:	01F5H
Name:	REDCLK
Function:	Read clock data
Entry :	[C]=Clock RAM address
bit -	7 6 5 4 3 2 1 0
[C] =	X X M1 M0 A3 A2 A1 A0
Returns:	[A]=Data read (valid lower 4bits)
Modifies:	[F]

3.14.2 WRTCLK -

Address:	01F9H
Name:	WRTCLK
Function:	Write clock data
Entry :	[C]=Clock RAM address [A]=Data to write
bit -	7 6 5 4 3 2 1 0
[C] =	X X M1 M0 A3 A2 A1 A0
Returns:	None
Modifies:	[F]