MSX® BIOS

Edited: January 1985
by Steven M. Ting

Graphic design: Mervin Fong.

The information in this document is subject to change without notice. ASCII Corp. makes no warranty with regard to the manual, including but not limited to, implied warranties of merchantability and fitness for a particular purpose. The parties above assume no responsibility for any errors which may appear in this document.

This document is not intended as "Consumer goods" under applicable federal or state law(s).

Printed in United States

TABLE OF CONTENTS

```
                                .list
                        ;
                        ;
                        ;       (C) Copyright by ASCII Corp., 1983
                        ;       Proprietary information.  All rights reserved.
                        ;
                        ;       File:   BIOHDR.MAC
                        ;       USE:    Restart calls and ROM entries table
                        ;       Written by Jey Suzuki, Rick Yamashita
                        ;               ASCII Corporation, Japan
                        ;
                        ;       Edit:   January, 1985
                        ;       Reason: Zilog Z80 Mnemonic version and cleanup
                        ;       Edited by:      Steven M. Ting
                        ;
                        ;
                        ; Labels referenced in this listing, are the absolute locations
                        ; within the MSX ROM. However, "ONLY" this BIOS entry point table,
                        ; and RAM variables are guaranteed to be permanent.
                        ;
                        ; All other locations in the ROM, will be changed without notice.
                        ;
                        SUBTTL -BIOS header- BIOS calls (Basic Interpreter, Slot I/O)
```

```

                        ;
                        ;  The following RST's (RST 0 thru RST 5) are reserved for BASIC
                        ;  interpreter, RST 6 for inter-slot calls, and RST 7 for
                        ;  hardware interrupt
                        ;
0000    F3              BEGIN:  DI                      ;Fail safe
0001    C3 02D7                 JP      CHKRAM          ;Finds all connected RAM
                                                        ;and cartridges
                        ;
                        ;
                        ;  ** Special information for the VDP. **
                        ;  Any program that accesses the VDP hardware directly
                        ;  should read the I/O port address found here, to be certain
                        ;  the software is compatible with future versions of the VDP.
                        ;
0004    1BBF                    DW      CGTABL          ;Address of character generator table,
                                                        ;to allow use of other character ROM.
                                                        ;
0006    98                      DB      98H             ;Current port address for VDP Data read
0007    98                      DB      98H             ;  "    "     "    "   "    "  write
                        ;
0008    C3 2683                 JP      SYNCHR          ;Check byte following the RST 8, see
                                                        ;if equal to the byte pointed by HL
000B    00                      DB      0
000C    C3 01B6                 JP      RDSLT           ;Read a byte from another slot
000F    00                      DB      0
0010    C3 2686                 JP      CHRGTR          ;Fetch next char from BASIC text
0013    00                      DB      0
0014    C3 01D1                 JP      WRSLT           ;Write a byte to another slot
0017    00                      DB      0
```

```
0018    C3 1B45             JP      OUTDO           ;Output a char to the Console or printer
001B    00                  DB      0
001C    C3 0217             JP      CALSLT          ;Perform Inter-slot call
001F    00                  DB      0
0020    C3 146A             JP      DCOMPR          ;Compares [HL] to [DE]
0023    00                  DB      0
0024    C3 025E             JP      ENASLT          ;Permanently enables a slot
0027    00                  DB      0
0028    C3 2689             JP      GETYPR          ;Returns the [FAC] type
002B    00                  DB      0               ;ID Byte (1)
                                                    ;Format:
                                                    ;  B7 B6 B5 B4 B3 B2 B1 B0
                                                    ;   +  +  +  +  +  +  +  +
                                                    ;   +  +  +  +  - - - - - - Type of character
                                                    ;   +  +  +  +              generator.
                                                    ;   +  +  +  +              0:Japanese
                                                    ;   +  +  +  +              1:International
                                                    ;   +  +  +  +              2:Korea
                                                    ;   +  +  - - - - - - - - - Date format
                                                    ;   +  +                    0: Y-M-D  1: M-D-Y
                                                    ;   +  +                    2: D-M-Y
                                                    ;   - - - - - - - - - - - - Interrupt frequency
                                                    ;                           0: 60 Hz  1: 50 Hz
002C    00                  DB      0               ;ID Byte (2)
                                                    ;Format:
                                                    ;  B7 B6 B5 B4 B3 B2 B1 B0
                                                    ;   +  +  +  +  +  +  +  +
                                                    ;   +  +  +  +  - - - - - - Type of Keyboard
                                                    ;   +  +  +  +              0:Japanese 2:French
                                                    ;   +  +  +  +              1:Int      3:UK
                                                    ;   +  +  +  +              4:DIN
```

```
                                                    ;   - - - - - - - - - - - - Version of BASIC
                                                    ;                           0: Japanese
                                                    ;                           1: International
002D    00 00 00            DB      0,0,0
0030    C3 0205             JP      CALLF           ;Performs Far-call (i.e., Inter-slot)
0033    00 00 00 00         DB      0,0,0,0,0
0037    00
                            ;
                    ;
                    ;  Following are used for I/O initialization
                    ;
0038    C3 0C3C             JP      KEYINT          ;Handlers for hardware interrupt
003B    C3 049D             JP      INITIO          ;Do device initialization
003E    C3 139D             JP      INIFNK          ;Reset all function key's text
                            ;
                    SUBTTL -BIOS header- BIOS calls (Video display processor)
```

```
 102
 103                               ;
 104                               ;  The following entry points provides control of the
 105                               ;  VDP's registers, screen mode settings, and memory block
 106                               ;  move between DRAM and VRAM.
 107                               ;
 108    0041     C3 0577                  JP      DISSCR          ;Disables screen display
 109    0044     C3 0570                  JP      ENASCR          ;Enables screen display
 110    0047     C3 057F                  JP      WRTVDP          ;Write a byte to any VDP register
 111    004A     C3 07D7                  JP      RDVRM           ;Read VRAM addressed using [HL]
 112    004D     C3 07CD                  JP      WRTVRM          ;Write VRAM addressed using [HL]
 113    0050     C3 07EC                  JP      SETRD           ;Sets up VDP for read
 114    0053     C3 07DF                  JP      SETWRT          ;Sets up VDP for write
 115    0056     C3 0815                  JP      FILVRM          ;Fills VRAM with specified data
 116    0059     C3 070F                  JP      LDIRMV          ;Moves block of data from VRAM to memory
 117    005C     C3 0744                  JP      LDIRVM          ;  "     "   "   "    "   memory to VRAM
 118    005F     C3 084F                  JP      CHGMOD          ;Change screen mode of VDP to [SCRMOD]
 119    0062     C3 07F7                  JP      CHGCLR          ;change Foreground, background,
 120                                                              ;border, color
 121    0065     00                       DB      0
 122                                      ;
 123                               ;
 124    0066     C3 1398                  JP      NMI             ;Handler for non-maskable interrupt
 125                               ;
 126    0069     C3 06A8                  JP      CLRSPR          ;Init sprite data
 127    006C     C3 050E                  JP      INITXT          ;Init VDP for 40 X 24 text mode (SCREEN 0)
 128    006F     C3 0538                  JP      INIT32          ; "    "   "  32 X 24 text mode (SCREEN 1)
 129    0072     C3 05D2                  JP      INIGRP          ; "    "   "  High resolution mode (SCREEN 2)
 130    0075     C3 061F                  JP      INIMLT          ; "    "   "  Multi color mode (SCREEN 3)
 131    0078     C3 0594                  JP      SETTXT          ;Sets VDP to display 40 X 24 text mode
 132    007B     C3 05B4                  JP      SETT32          ; "    "  "     "    32 X 24 text mode
```

```
 133    007E     C3 0602                  JP      SETGRP          ; "    "  "     "    High-res mode
 134    0081     C3 0659                  JP      SETMLT          ; "    "  "     "    Multi color mode
 135    0084     C3 06E4                  JP      CALPAT          ;Get address of sprite pattern table
 136    0087     C3 06F9                  JP      CALATR          ; "     "     "    "   attribute table
 137    008A     C3 0704                  JP      GSPSIZ          ;Returns current sprite size
 138    008D     C3 1510                  JP      GRPPRT          ;Print a character on the graphic screen
 139                                      ;
 140                               SUBTTL -BIOS header- BIOS calls (Programmable Sound Generator control)
```

```
 141
 142                               ;
 143                               ;  Following entry points are used for PSG initialization,
 144                               ;  read and write PSG registers, and PLAY statement execution.
 145                               ;
 146    0090     C3 04BD                  JP      GICINI          ;Init PSG, and static data for PLAY
 147    0093     C3 1102                  JP      WRTPSG          ;Write data to PSG
 148    0096     C3 110E                  JP      RDPSG           ;Read data from PSG
 149    0099     C3 11C4                  JP      STRTMS          ;Checks and start background task for PLAY
 150                                      ;
 151                               SUBTTL -BIOS header- BIOS calls (Keyboard, CRT, and Printer)
```

```

                          ;
                          ;  General INPUT and PRINT utilities.
                          ;
009C    C3 0D6A                JP      CHSNS          ;Checks status of keyboard status
009F    C3 10CB                JP      CHGET          ;Return char typed, with wait
00A2    C3 08BC                JP      CHPUT          ;Output character to console
00A5    C3 085D                JP      LPTOUT         ;  "         "      to printer, if possible
00A8    C3 0884                JP      LPTSTT         ;Checks status of line printer
00AB    C3 089D                JP      CNVCHR         ;Checks for graphic header byte
                                                      ;and convert code
00AE    C3 23BF                JP      PINLIN         ;Read line from keyboard to buffer
00B1    C3 23D5                JP      INLIN          ;Same as above, except in case of
                                                      ;AUTFLG is set
00B4    C3 23CC                JP      QINLIN         ;Print a "?", then jump to INLIN
00B7    C3 046F                JP      BREAKX         ;[Control-STOP] pressed??
00BA    C3 03FB                JP      ISCNTC         ;[Shift-STOP] pressed??
00BD    C3 10F9                JP      CKCNTC         ;Same as ISCNTC, but used by BASIC
00C0    C3 1113                JP      BEEP           ;Buzz
00C3    C3 0848                JP      CLS            ;Clear screen
00C6    C3 088E                JP      POSIT          ;Place cursor at Column [H], Row [L]
00C9    C3 0B26                JP      FNKSB          ;Display Function key, if neccessary
00CC    C3 0B15                JP      ERAFNK         ;Stop displaying the Function keys
00CF    C3 0B2B                JP      DSPFNK         ;Enable Function key display
00D2    C3 083B                JP      TOTEXT         ;Force screen to text mode
                               ;
                          SUBTTL -BIOS header- BIOS calls (Game and Cassette I/O, Queue handler)
```

```

                          ;
                          ;  Following are used to read the value from Joysticks,
                          ;  Graphic pad (tablet), and Paddles.
                          ;
00D5    C3 11EE                JP      GTSTCK         ;Return status of joystick
00D8    C3 1253                JP      GTTRIG         ;Read joystick trigger button
00DB    C3 12AC                JP      GTPAD          ;Returns status of graphic pad
00DE    C3 1273                JP      GTPDL          ;Read paddle
                               ;
                          ;
                          ;  Following are used to access the cassette tape,
                          ;  data read/write, and motor on/off
                          ;
00E1    C3 1A63                JP      TAPION         ;Turn on motor and read tape header
00E4    C3 1ABC                JP      TAPIN          ;Read tape data
00E7    C3 19E9                JP      TAPIOF         ;Stops reading from tape
00EA    C3 19F1                JP      TAPOON         ;Turn on motor and write tape header
00ED    C3 1A19                JP      TAPOUT         ;Write data to tape
00F0    C3 19DD                JP      TAPOFF         ;Stops writing to tape
00F3    C3 1384                JP      STMOTR         ;Start, stop cassette motor, or
                                                      ;flip motor(on to off, off to on)
                               ;
                          ;
                          ;  BASIC queues
                          ;
00F6    C3 14EB                JP      LFTQ           ;Bytes left in queue
00F9    C3 1492                JP      PUTQ           ;Send a byte to queue
                               ;
                          SUBTTL -BIOS header- BIOS calls (Generalized graphics)

                          ;
                          ;  For BASIC interpreter's GENGRP and ADVGRP modules use
 00FC    C3 16C5                JP      RIGHTC          ;Moves one pixel right
 00FF    C3 16EE                JP      LEFTC           ;  "    "    "   left
 0102    C3 175D                JP      UPC             ;  "    "    "   up
 0105    C3 173C                JP      TUPC            ;  "    "    "   "
 0108    C3 172A                JP      DOWNC           ;  "    "    "   down
 010B    C3 170A                JP      TDOWNC          ;  "    "    "   "
 010E    C3 1599                JP      SCALXY          ;Scales X Y cordinates
 0111    C3 15DF                JP      MAPXYC          ;Maps cordinates to physical address
 0114    C3 1639                JP      FETCHC          ;Get current physical address and
                                                        ;mask pattern
 0117    C3 1640                JP      STOREC          ;Put current physical address and
                                                        ;mask pattern
 011A    C3 1676                JP      SETATR          ;Sets the color attribute byte
 011D    C3 1647                JP      READC           ;Reads attribute of current pixel
 0120    C3 167E                JP      SETC            ;Sets current pixel to specified attribute
 0123    C3 1809                JP      NSETCX          ;Sets pixel horizontally
 0126    C3 18C7                JP      GTASPC          ;Returns aspect ratio
 0129    C3 18CF                JP      PNTINI          ;Do paint initialization
 012C    C3 18E4                JP      SCANR           ;Scan pixels to the right
 012F    C3 197A                JP      SCANL           ; "    "    "    "  left
                                ;
                          SUBTTL -BIOS header- BIOS calls (Misc. Entries)
```

```
( MSX ROM BASIC BIOS )  Macro-80      3.44    01-Jan-85       PAGE    8                11
-BIOS header- BIOS calls (Misc. Entries)

                          ;
                          ;
 0132    C3 0F3D                JP      CHGCAP          ;Turn [CAPSLOCK] light, on/off
 0135    C3 0F7A                JP      CHGSND          ;Change status of 1 bit sound port
 0138    C3 144C                JP      RSLREG          ;Return output of primary slot register
 013B    C3 144F                JP      WSLREG          ;Write to primary slot register
 013E    C3 1449                JP      RDVDP           ;Read VDP status register
 0141    C3 1452                JP      SNSMAT          ;Read a specified row in the
                                                        ;keyboard matrix
 0144    C3 148A                JP      PHYDIO          ;Performs operation for mass storage
                                                        ;devices (such as disks)
 0147    C3 148E                JP      FORMAT          ;Initialize mass storage device
 014A    C3 145F                JP      ISFLIO          ;Are we doing device I/O
 014D    C3 1B63                JP      OUTDLP          ;Output to line printer
 0150    C3 1470                JP      GETVCP          ;Used by Music background tasking
 0153    C3 1474                JP      GETVC2          ; "   "    "      "          "
 0156    C3 0468                JP      KILBUF          ;Clear the keyboard buffer
 0159    C3 01FF                JP      CALBAS          ;Performs far-call into BASIC
 015C                           DS      005AH           ;RESERVED FOR EXPANSION
                          ;
                          SUBTTL - SLOT -  Slot handler stuff
```

```
( MSX ROM BASIC BIOS )  Macro-80      3.44    01-Jan-85       PAGE    9                12
- SLOT -  Slot handler stuff

 00A8                     PPI.AR  EQU     0A8h    ;A8H    read from PPI Port A
 00A8                     PPI.AW  EQU     0A8h    ;A8H    Write to PPI Port A
                          ;
                          ; Every cartridge located at 0000-3FFFH must contain codes in
                          ; this module which are entered via following addresses.
                          ;
                          ;    000CH RDSLT
                          ;    0014H WRSLT
                          ;    001CH CALSLT
                          ;    0024H ENASLT
                          ;
                          ;
                          ; ----------------------------- RDSLT ------------------------------
                          ;
                          ; Selects the appropriate slot according to the value given
                          ; through registers, and read the content of memory from the
                          ; slot.
                          ;
                          ; Input parameters:
                          ; A  - FxxxSSPP
                          ;      |   ||||
                          ;      |   ||++-- primary slot # (0-3)
                          ;      |   ++---- secondary slot # (0-3)
                          ;      +--------- 1 if secondary slot # specified
                          ;
                          ;               HL - address of target memory
                          ; Returned value
                          ;               A  - content of memory
                          ;
                          ;  Note: Interrupts are disabled automatically but never enabled
```

```
                                 ;        by this routine.
                                 ;
 01B6                            RDSLT:
 01B6      CD 027E                        CALL     SELPRM          ;Calculate bit pattern and mask code
 01B9      FA 01C6                        JP       M,RDESLT        ;Expanded slot specified
 01BC      DB A8                          IN       A,(PPI.AR)
 01BE      57                             LD       D,A             ;Save current setting
 01BF      A1                             AND      C               ;Cancel current setting for target address
 01C0      B0                             OR       B               ;Add new setting
 01C1      CD F380                        CALL     RAMLOW          ;Call read primitive routine (in system area)
 01C4      7B                             LD       A,E             ;Return value via [Acc]
 01C5      C9                             RET
 01C6                            RDESLT:
 01C6      E5                             PUSH     HL              ;Save target address
 01C7      CD 02A3                        CALL     SELEXP          ;Select secondary slot
 01CA      E3                             EX       (SP),HL         ;Restore target address and save [HL]
 01CB      C5                             PUSH     BC
 01CC      CD 01B6                        CALL     RDSLT
 01CF      18 1B                          JR       WRESED          ;Restore old slot select register
                                 SUBTTL   -SLOT-   Slot handler (Write slot)
```

```
( MSX ROM BASIC BIOS )  Macro-80        3.44    01-Jan-85        PAGE    10                                14
-SLOT-  Slot handler (Write slot)


                                 ;
                                 ; ------------------------------ WRSLT ------------------------------
                                 ;
                                 ; Selects the appropriate slot according to the value given
                                 ; through registers, and write to the memory in the specified
                                 ; slot.
                                 ;
                                 ; Input parameters:
                                 ; A  - FxxxSSPP
                                 ;      |   ||||
                                 ;      |   ||++-- primary slot # (0-3)
                                 ;      |   ++---- secondary slot # (0-3)
                                 ;      +--------- 1 if secondary slot # specified
                                 ;
                                 ;           HL - address of target memory
                                 ;
                                 ;           E  - value to be written
                                 ;
                                 ;  Note: Interrupts are disabled automatically but never enabled
                                 ;        by this routine.
                                 ;
 01D1                            WRSLT:
 01D1      D5                             PUSH     DE              ;Save data to be written
 01D2      CD 027E                        CALL     SELPRM          ;Calculate bit pattern and mask code
 01D5      FA 01E1                        JP       M,WRESLT        ;Expanded slot specified
 01D8      D1                             POP      DE              ;Restore data to be written
 01D9      DB A8                          IN       A,(PPI.AR)
 01DB      57                             LD       D,A             ;Save current setting
 01DC      A1                             AND      C               ;Cancel current setting for target address
 01DD      B0                             OR       B               ;Add new setting
```

```
( MSX ROM BASIC BIOS )  Macro-80        3.44    01-Jan-85        PAGE    10-1                               15
--SLOT-  Slot handler (Write slot)

 01DE      C3 F385                        JP       WRPRIM          ;Call write primitive routine (in system area)
 01E1                            WRESLT:
 01E1      E3                             EX       (SP),HL         ;Save target address, get data to be written
 01E2      E5                             PUSH     HL              ;Save data to be written
 01E3      CD 02A3                        CALL     SELEXP          ;Select secondary slot
 01E6      D1                             POP      DE              ;Restore data to be written
 01E7      E3                             EX       (SP),HL         ;Restore target address and save [HL]
 01E8      C5                             PUSH     BC
 01E9      CD 01D1                        CALL     WRSLT
 01EC                            WRESED:
 01EC      C1                             POP      BC
 01ED      E3                             EX       (SP),HL         ;Save target address and get old [HL]
 01EE      F5                             PUSH     AF              ;Save value returned by RDSLT
 01EF      78                             LD       A,B             ;Get current setting
 01F0      E6 3F                          AND      00111111B       ;Cancel current setting for 0C000H..0FFFFH
 01F2      B1                             OR       C
 01F3      D3 A8                          OUT      (PPI.AW),A      ;Enable 0C000H..0FFFFH of target bank
 01F5      7D                             LD       A,L             ;Restore old setting of slot register
 01F6      32 FFFF                        LD       (0FFFFH),A
 01F9      78                             LD       A,B             ;Finally restore old primary slot register
 01FA      D3 A8                          OUT      (PPI.AW),A
 01FC      F1                             POP      AF              ;Restore value returned by RDSLT
 01FD      E1                             POP      HL              ;Restore target address
 01FE      C9                             RET
```

```

01FF                  CALBAS:
01FF     FD 2A FCC0           LD      IY,(EXPTBL-1)
0203     18 12                JR      CALSLT
0205                  CALLF:
0205     E3                   EX      (SP),HL         ;Get return address, save [HL]
0206     F5                   PUSH    AF              ;Save working registers
0207     D5                   PUSH    DE
0208     7E                   LD      A,(HL)          ;Get destination slot
0209     F5                   PUSH    AF
020A     FD E1                POP     IY              ;Move it to IYH
020C     23                   INC     HL
020D     5E                   LD      E,(HL)          ;Get destination address
020E     23                   INC     HL
020F     56                   LD      D,(HL)
0210     23                   INC     HL              ;Prepare true return address
0211     D5                   PUSH    DE
0212     DD E1                POP     IX              ;Move it to IX
0214     D1                   POP     DE              ;Restore working registers
0215     F1                   POP     AF
0216     E3                   EX      (SP),HL         ;Resture [HL], save true return address
                      SUBTTL -SLOT-
```

```

                      ;
                      ; ------------------------------ CALSLT ------------------------------
                      ;
                      ; Performs inter-slot call to specified address.
                      ;
                      ; Input parameters:
                      ; IY  - FxxxSSPP
                      ;       |   ||||
                      ;       |   ||++-- primary slot # (0-3)
                      ;       |   ++---- secondary slot # (0-3)
                      ;       +--------- 1 if secondary slot # specified
                      ;
                      ; IX  - address to call
                      ;
                      ;  Note: Interrupts are disabled automatically but never enabled
                      ;        by this routine.
                      ;        You can never pass arguments via alternate registers
                      ;        of Z80.
                      ;
0217                  CALSLT:
0217     D9                   EXX                     ;Save environments
0218     08                   EX      AF,AF'
0219     FD E5                PUSH    IY
021B     F1                   POP     AF              ;Get target slot information
021C     DD E5                PUSH    IX
021E     E1                   POP     HL              ;Get target address
021F     CD 027E              CALL    SELPRM
0222     FA 022E              JP      M,CALESL        ;Call expanded slot
0225     DB A8                IN      A,(PPI.AR)
0227     F5                   PUSH    AF              ;Save current value of primary slot register
```

```
0228    A1                      AND     C               ;Cancel current setting for target address
0229    B0                      OR      B               ;Add new setting
022A    D9                      EXX                     ;Restore environments except PSW
022B    C3 F38C                 JP      CLPRIM          ;Jump to primitive routine (in system area)
022E                    CALESL:
022E    CD 02A3                 CALL    SELEXP          ;Select secondary slot register
0231    F5                      PUSH    AF              ;Move primary slot # in [IYH]
0232    FD E1                   POP     IY
0234    E5                      PUSH    HL              ;Save [B,C,L]  which contain information
0235    C5                      PUSH    BC              ;for restoring slot environments
0236    4F                      LD      C,A             ;Move primary slot # to [BC]
0237    06 00                   LD      B,0
0239    7D                      LD      A,L             ;Re-calculate what is currently output
023A    A4                      AND     H               ;to expansion slot register
023B    B2                      OR      D
023C    21 FCC5                 LD      HL,SLTTBL       ;Calculate address into SLTTBL
023F    09                      ADD     HL,BC
0240    77                      LD      (HL),A          ;Set current value output to expansion
                                                        ;slot register
0241    E5                      PUSH    HL              ;Remember this address
0242    08                      EX      AF,AF'          ;Restore possible arguments passed
0243    D9                      EXX                     ;via registers
0244    CD 0217                 CALL    CALSLT          ;Call by primary slot #
0247    D9                      EXX                     :Save possible values returned via
0248    08                      EX      AF,AF'          ;registers
0249    E1                      POP     HL              ;Restore address into SLTTBL
024A    C1                      POP     BC              ;Restore information about old slots
024B    D1                      POP     DE
024C    78                      LD      A,B             ;Get current setting
024D    E6 3F                   AND     00111111B       ;Cancel current setting for 0C000H..0FFFFH
024F    B1                      OR      C
```

```
0250    F3                      DI
0251    D3 A8                   OUT     (PPI.AW),A      ;Enable 0C000H..0FFFFH of target bank
0253    7B                      LD      A,E             ;Restore old setting of slot register
0254    32 FFFF                 LD      (0FFFFH),A
0257    78                      LD      A,B             ;Finally restore old primary slot register
0258    D3 A8                   OUT     (PPI.AW),A
025A    73                      LD      (HL),E          ;And change SLTTBL also
025B    08                      EX      AF,AF'          ;Restore possible returned values
025C    D9                      EXX
025D    C9                      RET
```

```

                        ;
                        ; ------------------------------ ENASLT ------------------------------
                        ;
                        ; Selects the appropriate slot according to the value given
                        ; through registers, and permanently enables the slot.
                        ;
                        ; Input parameters:
                        ;
                        ; A  - FxxxSSPP
                        ;       |   ||||
                        ;       |   ||++-- primary slot # (0-3)
                        ;       |   ++---- secondary slot # (0-3)
                        ;       +--------- 1 if secondary slot # specified
                        ;
                        ; HL - address of target memory
                        ;
                        ;  Note: Interrupts are disabled automatically but never enabled
                        ;        by this routine.
                        ;
025E                    ENASLT:
025E    CD 027E                 CALL    SELPRM          ;Calculate bit pattern and mask code
0261    FA 026B                 JP      M,ENESLT        ;Expanded slot specified
0264    DB A8                   IN      A,(PPI.AR)
0266    A1                      AND     C               ;Cancel current setting for target address
0267    B0                      OR      B               ;Add new setting
0268    D3 A8                   OUT     (PPI.AW),A
026A    C9                      RET
026B                    ENESLT:
026B    E5                      PUSH    HL              ;Save target address
026C    CD 02A3                 CALL    SELEXP          ;Select secondary slot
```

```
026F    4F                LD      C,A             ;Move primary slot # to [BC]
0270    06 00             LD      B,0
0272    7D                LD      A,L             ;Re-calculate what is  currently  output
0273    A4                AND     H               ;to expansion slot register
0274    B2                OR      D
0275    21 FCC5           LD      HL,SLTTBL       ;Calculate address into SLTTBL
0278    09                ADD     HL,BC
0279    77                LD      (HL),A          ;Set current  value  output to expansion
                                                  ;slot register
027A    E1                POP     HL              ;Restore target address
027B    79                LD      A,C             ;Restore primary slot # to [Acc]
027C    18 E0             JR      ENASLT          ;Enable by primary slot register
```

```

027E              SELPRM:
027E    F3                DI
027F    F5                PUSH    AF              ;Save slot address
0280    7C                LD      A,H             ;Extract upper 2 bits
0281    07                RLCA
0282    07                RLCA
0283    E6 03             AND     00000011B
0285    5F                LD      E,A
0286    3E C0             LD      A,0C0H          ;Format mask pat. correspond to address
0288              SLPRM1:
0288    07                RLCA
0289    07                RLCA
028A    1D                DEC     E
028B    F2 0288           JP      P,SLPRM1
028E    5F                LD      E,A             ;Save mask pattern
                                                  ;       00000011        0000-3FFF
                                                  ;       00001100        4000-7FFF
                                                  ;       00110000        8000-BFFF
                                                  ;       11000000        C000-FFFF
028F    2F                CPL
0290    4F                LD      C,A             ;Save mask pattern
                                                  ;       11111100        0000-3FFF
                                                  ;       11110011        4000-7FFF
                                                  ;       11001111        8000-BFFF
                                                  ;       00111111        C000-FFFF
0291    F1                POP     AF              ;Restore slot address
0292    F5                PUSH    AF
0293    E6 03             AND     00000011B       ;Extract primary slot #
0295    3C                INC     A
0296    47                LD      B,A
```

```
0297    3E AB             LD      A,10101011B     ;Convert slot # to proper bit pattern
0299              SLPRM2:
0299    C6 55             ADD     A,01010101B
029B    10 FC             DJNZ    SLPRM2
029D    57                LD      D,A             ;Save bit pattern for primary slot #
                                                  ;       00000000        slot #0
                                                  ;       01010101        slot #1
                                                  ;       10101010        slot #2
                                                  ;       11111111        slot #3
029E    A3                AND     E               ;Extract significant bits
029F    47                LD      B,A             ;Set it to [B]
02A0    F1                POP     AF              ;Expanded slot specified?
02A1    A7                AND     A               ;Set sign flag if so
02A2    C9                RET
02A3              SELEXP:
02A3    F5                PUSH    AF              ;Save target slot
02A4    7A                LD      A,D             ;Get bit pattern for primary slot
02A5    E6 C0             AND     11000000B       ;Extract slot # for 0C000H..0FFFFH
02A7    4F                LD      C,A             ;Save it
02A8    F1                POP     AF              ;Restore target slot
02A9    F5                PUSH    AF              ;Save target slot
02AA    57                LD      D,A             ;Load [D] with specified slot address
02AB    DB A8             IN      A,(PPI.AR)
02AD    47                LD      B,A             ;Save current setting
02AE    E6 3F             AND     00111111B       ;Cancel current setting for 0C000H..0FFFFH
02B0    B1                OR      C
02B1    D3 A8             OUT     (PPI.AW),A      ;Enable 0C000H..0FFFFH or target bank
02B3    7A                LD      A,D             ;Load slot information
02B4    0F                RRCA
02B5    0F                RRCA
02B6    E6 03             AND     00000011B       ;Extract secondary slot #
```

```
  561     02B8     57                    LD      D,A
  562     02B9     3E AB                 LD      A,10101011B      ;Convert secondary   slot  #  to  proper
  563     02BB                   SLEXP1:
  564     02BB     C6 55                 ADD     A,01010101B      ;bit pattern
  565     02BD     15                    DEC     D
  566     02BE     F2 02BB               JP      P,SLEXP1         ;        00000000        slot #0
  567                                                           ;        01010101        slot #1
  568                                                           ;        10101010        slot #2
  569                                                           ;        11111111        slot #3
  570     02C1     A3                    AND     E                ;Make bit pattern to be added
  571     02C2     57                    LD      D,A              ;Save this
  572     02C3     7B                    LD      A,E              ;Make bit pattern to strip off old value
  573     02C4     2F                    CPL
  574     02C5     67                    LD      H,A              ;Save this
  575     02C6     3A FFFF               LD      A,(0FFFFH)       ;Read expanded slot register
  576     02C9     2F                    CPL
  577     02CA     6F                    LD      L,A              ;Save current setting
  578     02CB     A4                    AND     H                ;Strip off old bits
  579     02CC     B2                    OR      D                ;And set new bits
  580     02CD     32 FFFF               LD      (0FFFFH),A       ;Set secondary slot register
  581     02D0     78                    LD      A,B
  582     02D1     D3 A8                 OUT     (PPI.AW),A       ;Restore original primary port
  583     02D3     F1                    POP     AF               ;Restore target slot
  584     02D4     E6 03                 AND     00000011B        ;Fake read from primary slot
  585     02D6     C9                    RET
  586                            SUBTTL  - MSXIO -  I/O Module
( MSX ROM BASIC BIOS )  Macro-80        3.44    01-Jan-85       PAGE    15                                25
- MSXIO -  I/O Module

  587
  588                            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  589                            ;                            ;
  590                            ;       Port definition      ;
  591                            ;                            ;
  592                            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  593                            ;
  594                            ;           VDP address definition
  595                            ;
  596     0098                   VDP.DRW EQU     10011000B        ;98H    Read/write data VDP
  597     0099                   VDP.CW  EQU     10011001B        ;99H    write command to VDP
  598     0099                   VDP.SR  EQU     10011001B        ;99H    read status from VDP
  599                            ;
  600     0007                   V.COLR  EQU     7                ;In text mode, foreground and background color
  601                                                           ;Otherwise background color
  602                            ;
  603                            ;           PSG address definition
  604                            ;
  605     00A0                   PSG.LW  EQU     10100000B        ;A0H    latch address for PSG
  606     00A1                   PSG.DW  EQU     10100001B        ;A1H    write data to PSG
  607     00A2                   PSG.DR  EQU     10100010B        ;A2H    read data from PSG
  608                            ;
  609     000E                   PSG.PA  EQU     14                ;Port A of PSG
  610     000F                   PSG.PB  EQU     15                ;Port B of PSG
  611                            ;
  612                            ;           PPI address definition
  613                            ;
  614     00A8                   PPI.AR  EQU     10101000B        ;A8H    read from PPI Port A
  615     00A9                   PPI.BR  EQU     10101001B        ;A9H    read from PPI Port B
  616     00AA                   PPI.CR  EQU     10101010B        ;AAH    read from PPI Port C
  617     00A8                   PPI.AW  EQU     10101000B        ;A8H    Write to PPI Port A
```

```
00AA                 PPI.CW  EQU     10101010B      ;AAH    write to PPI Port C
00AB                 PPI.CM  EQU     10101011B      ;ABH    write to PPI command register
                     ;
                     ;          Printer port definition
                     ;
0091                 LPT.DW  EQU     10010001B      ;Data port
0090                 LPT.SB  EQU     10010000B      ;Strobe output
0090                 LPT.ST  EQU     10010000B      ;Printer status
                     ;
                     ;          Text mode (40*24)                       SCREEN 0
                     ;
                     ;               TXTNAM,TXTCGP
                     ;
                     ;          Text mode (graphics 1)                  SCREEN 1
                     ;
                     ;               T32NAM,T32COL,T32CGP,T32ATR,T32PAT
                     ;
                     ;          Hires mode                              SCREEN 2
                     ;
                     ;               GRPNAM,GRPCOL,GRPCGP,GRPATR,GRPPAT
                     ;
                     ;          Multi-color mode                        SCREEN 3
                     ;
                     ;               MLTNAM,MLTCGP,MLTATR,MLTPAT
                     ;
                     ;          Screen size
                     ;
                     ;               LINLEN,CRTCNT,LINL32,LINL40
                     ;
                     ;          External constants
                     ;
```

```
                     ;               CGTABL         Character generator table
                     ;
                     ;          External variables
                     ;
                     ;               FORCLR         Foreground color
                     ;               BAKCLR         Background color
                     ;               BDRCLR         Border color for PAINT
                     ;               SCRMOD         Current screen mode
                                                    ;       0 - 40*24 text
                                                    ;       1 - 32*24 text
                                                    ;       2 - hiresolution graphics
                                                    ;       3 - Multicolor graphics
                     ;               OLDSCR
                     ;               NAMBAS         Base of current name table
                     ;               CGPBAS         Base of current cgen table
                     ;               PATBAS         Base of current sprite pattern table
                     ;               ATRBAS         Base of current sprite attribute table
                     ;               JIFFY          Jiffy count
                     ;               CLIKSW         Click switch
                     ;               CLIKFL         Click flag to suppress multiple key clicks
                     ;               RG0SAV         VDP register #0 save area
                     ;               RG1SAV         VDP register #1 save area
                     ;               STATFL         VDP status register
                     ;               PATWRK         Work area for pattern converter
                     ;
                     ;          External routines
                     ;
                     ;               GETQ
                     ;               PUTQ
                     ;               INITQ
                     SUBTTL - MSXIO -  Find available RAM
```

```

02D7                    CHKRAM:
                        ;
                        ; ------------------------------ CHKRAM ------------------------------
                        ;
                        ; Look into every slot from 0FFFFH to C000H, and set system work
                        ; area. Note that we cannot use RAM as work area nor perform
                        ; subroutine calls 'cause we do not yet know where the available
                        ; RAM exits. Everything has to be done inside ROM and CPU's
                        ; register until the RAM is found.
                        ;
02D7    3E 82                   LD      A,82H           ;Port A - output (mode 0)
02D9    D3 AB                   OUT     (PPI.CM),A      ;Port B - input (mode 0)
02DB    AF                      XOR     A               ;Port C - output (mode 0)
02DC    D3 A8                   OUT     (PPI.AW),A      ;Select slot 0 for all addresses
02DE    3E 50                   LD      A,'P'           ;Disable all cassette related outputs
02E0    D3 AA                   OUT     (PPI.CW),A      ;Motor off
                        ;
                        ; Start searching
                        ;
                        ; Register usage:
                        ; B - non 0 if we're now checking secondary slot
                        ; SPH - slot # of the biggest RAM block
                        ; SPL - secondary slot # of the biggest RAM block (if any)
                        ; DE - lowest address of the biggest RAM block ever found
                        ; C - 'slot-expanded' flag
                        ;
                        ; 0000xxxx
                        ;     ||||
                        ;     |||+- slot #3 expanded
                        ;     ||+-- slot #2 expanded
```

```
                        ;     |+--- slot #1 expanded
                        ;     +---- slot #0 expanded
                        ;
02E2    11 FFFF                 LD      DE,0FFFFH       ;Initialize lowest address ever found
02E5    AF                      XOR     A               ;Start from slot #0
02E6    4F                      LD      C,A             ;Clear bit pattern
02E7                    CKRM05:
02E7    D3 A8                   OUT     (PPI.AW),A      ;Select the slot
02E9    CB 21                   SLA     C               ;Shift bit pattern
02EB    06 00                   LD      B,0             ;Assume this slot is not expanded
02ED    21 FFFF                 LD      HL,0FFFFH       ;Read from possible expansion slot register
02F0    36 F0                   LD      (HL),0F0H       ;Write  a binary 11110000
02F2    7E                      LD      A,(HL)
02F3    D6 0F                   SUB     0FH             ;Read back as 00001111?
02F5    20 0B                   JR      NZ,CKRM15       ;Nop, this is not an expanded slot
02F7    77                      LD      (HL),A          ;Write 00000000
02F8    7E                      LD      A,(HL)
02F9    3C                      INC     A               ;Read back as 11111111?
02FA    20 06                   JR      NZ,CKRM15       ;Nop, not expanded slot
02FC    04                      INC     B               ;We're checking expanded slot
02FD    CB C1                   SET     0,C             ;Say this slot is expanded
02FF                    CKRM10:
                        ;
                        ;Start from expansion slot #0
                        ;
02FF    32 FFFF                 LD      (0FFFFH),A      ;Select the expanded slot
0302                    CKRM15:
0302    21 BF00                 LD      HL,0BF00H       ;Start checking from 0BF00H to 8000H
0305                    CKRM20:
0305    7E                      LD      A,(HL)
0306    2F                      CPL
```

```
0307    77                      LD      (HL),A
0308    BE                      CP      (HL)
0309    2F                      CPL
030A    77                      LD      (HL),A
030B    20 07                   JR      NZ,CKRM25       ;RAM not equipped in this page
030D    2C                      INC     L               ;Make sure it's not a coincidence
030E    20 F5                   JR      NZ,CKRM20       ;Check more
0310    25                      DEC     H
0311    FA 0305                 JP      M,CKRM20        ;Check next page
0314                    CKRM25:
0314    2E 00                   LD      L,0
0316    24                      INC     H
0317    7D                      LD      A,L             ;Below the one ever found
0318    93                      SUB     E
0319    7C                      LD      A,H
031A    9A                      SBC     A,D
031B    30 0A                   JR      NC,CKRM30       ;No
031D    EB                      EX      DE,HL           ;Register this address as the lowest
031E    3A FFFF                 LD      A,(0FFFFH)      ;Set possible secondary slot #
0321    2F                      CPL
0322    6F                      LD      L,A
0323    DB A8                   IN      A,(PPI.AR)      ;Set primary slot #
0325    67                      LD      H,A
0326    F9                      LD      SP,HL           ;Register these slot #'s
0327                    CKRM30:
0327    78                      LD      A,B
0328    A7                      AND     A               ;Are we checking secondary slot
0329    28 0A                   JR      Z,CKRM35        ;No
032B    3A FFFF                 LD      A,(0FFFFH)
032E    2F                      CPL
032F    C6 10                   ADD     A,10H           ;Prepare to select next secondary slot
```

```
0331    FE 40                   CP      01000000B
0333    38 CA                   JR      C,CKRM10        ;Continue if more secondary slots remain
0335                    CKRM35:
0335    DB A8                   IN      A,(PPI.AR)
0337    C6 50                   ADD     A,01010000B     ;Prepare to select next slot
0339    30 AC                   JR      NC,CKRM05       ;Continue if more primary slots remain
```

```

                        ;
                        ; Check is done, select the biggest one
                        ;
033B    21 0000                 LD      HL,0
033E    39                      ADD     HL,SP
033F    7C                      LD      A,H
0340    D3 A8                   OUT     (PPI.AW),A      ;Set primary slot register
0342    7D                      LD      A,L
0343    32 FFFF                 LD      (0FFFFH),A      ;Set possible secondary slot register
                        ;
                        ; Next, check 0C000H..0FFFFH
                        ;
0346    79                      LD      A,C
0347    07                      RLCA
0348    07                      RLCA
0349    07                      RLCA
034A    07                      RLCA
034B    4F                      LD      C,A
034C    11 FFFF                 LD      DE,0FFFFH       ;Initialize lowest address ever found
034F    DB A8                   IN      A,(PPI.AR)      ;Start from slot #0
0351    E6 3F                   AND     00111111B
0353                    CKRM50:
0353    D3 A8                   OUT     (PPI.AW),A      ;Select the slot
0355    06 00                   LD      B,0             ;Assume this slot is not expanded
0357    CB 01                   RLC     C               ;Shift bit pattern
0359    30 0A                   JR      NC,CKRM60       ;This slot is not expanded
035B    04                      INC     B               ;We're checking expanded slot
035C    3A FFFF                 LD      A,(0FFFFH)
035F    2F                      CPL
0360    E6 3F                   AND     00111111B
```

```
0362                  CKRM55:
0362    32 FFFF                LD      (0FFFFH),A      ;Select the expanded slot
0365                  CKRM60:
0365    21 FE00                LD      HL,0FE00H       ;Start checking from 0FE00H to 0C000H
0368                  CKRM65:
0368    7E                     LD      A,(HL)
0369    2F                     CPL
036A    77                     LD      (HL),A
036B    BE                     CP      (HL)
036C    2F                     CPL
036D    77                     LD      (HL),A
036E    20 09                  JR      NZ,CKRM70       ;RAM not equipped in this page
0370    2C                     INC     L               ;Make sure it's not a coincidence
0371    20 F5                  JR      NZ,CKRM65       ;Check more
0373    25                     DEC     H
0374    7C                     LD      A,H
0375    FE C0                  CP      0C0H
0377    30 EF                  JR      NC,CKRM65       ;Check next page
0379                  CKRM70:
0379    2E 00                  LD      L,0
037B    24                     INC     H
037C    7D                     LD      A,L             ;Below the one ever found
037D    93                     SUB     E
037E    7C                     LD      A,H
037F    9A                     SBC     A,D
0380    30 0A                  JR      NC,CKRM75       ;No
0382    EB                     EX      DE,HL           ;Register this address as the lowest
0383    3A FFFF                LD      A,(0FFFFH)      ;Set possible secondary slot #
0386    2F                     CPL
0387    6F                     LD      L,A
0388    DB A8                  IN      A,(PPI.AR)      ;Set primary slot #
```

```
( MSX ROM BASIC BIOS )  Macro-80        3.44    01-Jan-85       PAGE    17-2        34
- MSXIO -  Find available RAM

038A    67                     LD      H,A
038B    F9                     LD      SP,HL           ;Register these slot #'s
038C                  CKRM75:
038C    78                     LD      A,B
038D    A7                     AND     A               ;Are we checking secondary slot
038E    28 08                  JR      Z,CKRM80        ;No
0390    3A FFFF                LD      A,(0FFFFH)
0393    2F                     CPL
0394    C6 40                  ADD     A,01000000B     ;Prepare to select next secondary slot
0396    30 CA                  JR      NC,CKRM55       ;Continue if more secondary slots remain
0398                  CKRM80:
0398    DB A8                  IN      A,(PPI.AR)
039A    C6 40                  ADD     A,01000000B     ;Prepare to select next slot
039C    30 B5                  JR      NC,CKRM50       ;Continue if more primary slots remain
                      SUBTTL - MSXIO -  Slot attribute setup
```

```
( MSX ROM BASIC BIOS )  Macro-80        3.44    01-Jan-85       PAGE    18          35
- MSXIO -  Slot attribute setup


                      ;
                      ; Check is done, select the biggest one
                      ;
039E    21 0000                LD      HL,0
03A1    39                     ADD     HL,SP
03A2    7C                     LD      A,H
03A3    D3 A8                  OUT     (PPI.AW),A      ;Set primary slot register
03A5    7D                     LD      A,L
03A6    32 FFFF                LD      (0FFFFH),A      ;Set possible secondary slot register
03A9    79                     LD      A,C             ;Set 'slot expanded' flag
                      ;
                      ; Clear work area with zero
                      ;
03AA    01 0C49                LD      BC,0C49H        ;length of work area
03AD    11 F381                LD      DE,RAMLOW+1
03B0    21 F380                LD      HL,RAMLOW       ;beginning of work
03B3    36 00                  LD      (HL),0          ;init first byte
03B5    ED B0                  LDIR                    ;transfer it to rest of area
                      ;
                      ; Set EXPTBL
                      ;
03B7    4F                     LD      C,A             ;Get 'slot-expanded' flag
03B8    06 04                  LD      B,4             ;Loop 4 times
03BA    21 FCC4                LD      HL,EXPTBL+3
03BD                  SSLTLP:
03BD    CB 19                  RR      C               ;Set carry if LSB is set
03BF    9F                     SBC     A,A             ;[Acc]=255 if expanded, 0 if not expanded
03C0    E6 80                  AND     80H             ;Affects only MSB
03C2    77                     LD      (HL),A          ;Set table for each slot
03C3    2B                     DEC     HL
```

```
03C4    10 F7                   DJNZ    SSLTLP
                        ;
                        ; Set SLTTBL
                        ;
03C6    DB A8                   IN      A,(PPI.AR)      ;Remember primary slot register's content
03C8    4F                      LD      C,A
03C9    AF                      XOR     A               ;Read from slot #0
03CA    D3 A8                   OUT     (PPI.AW),A
03CC    3A FFFF                 LD      A,(0FFFFH)
03CF    2F                      CPL
03D0    6F                      LD      L,A
03D1    3E 40                   LD      A,01000000B     ;Read from slot #1
03D3    D3 A8                   OUT     (PPI.AW),A
03D5    3A FFFF                 LD      A,(0FFFFH)
03D8    2F                      CPL
03D9    67                      LD      H,A
03DA    3E 80                   LD      A,80H           ;Read from slot #2
03DC    D3 A8                   OUT     (PPI.AW),A
03DE    3A FFFF                 LD      A,(0FFFFH)
03E1    2F                      CPL
03E2    5F                      LD      E,A
03E3    3E C0                   LD      A,0C0H          ;Read from slot #3
03E5    D3 A8                   OUT     (PPI.AW),A
03E7    3A FFFF                 LD      A,(0FFFFH)
03EA    2F                      CPL
03EB    57                      LD      D,A
03EC    79                      LD      A,C             ;Restore primary slot register
03ED    D3 A8                   OUT     (PPI.AW),A
03EF    22 FCC5                 LD      (SLTTBL),HL     ;Set SLTTBL
03F2    EB                      EX      DE,HL
03F3    22 FCC7                 LD      (SLTTBL+2),HL
```

```
03F6    ED 56                   IM      1               ;IM 1
03F8    C3 2680                 JP      INIT
                        SUBTTL - MSXIO -  Control-[C] processing
```

```

03FB                    ISCNTC:
03FB    3A FBB1                 LD      A,(BASROM)      ;Is BASIC text in ROM
03FE    A7                      AND     A
03FF    C0                      RET     NZ              ;Yes
0400    E5                      PUSH    HL
0401    21 FC9B                 LD      HL,INTFLG       ;Seen any interesting key
0404    F3                      DI
0405    7E                      LD      A,(HL)
0406    36 00                   LD      (HL),0
0408    E1                      POP     HL
0409    FB                      EI
040A    A7                      AND     A
040B    C8                      RET     Z               ;No
040C    FE 03                   CP      3               ;Is it ctrl-stop?
040E    28 1C                   JR      Z,EXCABO        ;Yes, execution aborted
                        ;
                        ; Pause until next STOP is pressed
                        ;
0410    E5                      PUSH    HL              ;STOP pressed (pause)
0411    D5                      PUSH    DE
0412    C5                      PUSH    BC
0413    CD 09DA                 CALL    CKDPC0          ;Display cursor if disabled
0416    21 FC9B                 LD      HL,INTFLG       ;Wait for next interesting key
0419                    WATINT:
0419    F3                      DI
041A    7E                      LD      A,(HL)
041B    36 00                   LD      (HL),0
041D    FB                      EI                      ;Wait for character if SELECT pressed
041E    A7                      AND     A               ;Seen?
041F    28 F8                   JR      Z,WATINT        ;Not yet
```