```
                                NAME    ('FPM80')

                                TITLE   'FPM/80 -- H8/DG-80 Front Panel Monitor'
                                SUBTTL  'Introduction.'

                                .XALL

                                .Z80

 0000                   FALSE   EQU     0                       ; False
 FFFF                   TRUE    EQU     NOT FALSE               ; True condition

                        ;       Conditional Assembly Parameters

                                .LIST

 FFFF                   ODSPLY  EQU     TRUE                    ; Default Display Mode is Octal
 0000                   RTMO    EQU     FALSE                   ; Return to monitor is "RTM-0"

 0001                   VER     EQU     1                       ; 1st try.
 0002                   SUBV    EQU     2                       ;

 0012                   BCDV    EQU     (VER*16)+SUBV           ; BCD Version #


                                IF1

                                IFT     ODSPLY
                                .PRINTX -  Default Display Mode is Octal  -
                                ELSE
                                .PRINTX -  Default Display Mode is Hexadecimal  -
                                ENDC

                                IFF     RTMO
                                .PRINTX -  RTM Keypad Entry is 'RTM'  -
                                ELSE
                                .PRINTX -  RTM Keypad Entry is 'RTM/0'  -
                                ENDC

                                ENDC

                                PAGE
 0000'

                                          .COMMENT *

                                           FPM/80 - Z80 Front Panel Monitor Program.

                                           Written by:     Bill Parrott & David Carroll

                                           For:            D-G Electronic Developments Company
                                                           Post Office Box 1124
                                                           1827 South Armstrong
                                                           Denison, Texas  75020

                                                           Copyright 1980, by D-G Electronic Developments

                                           Abstract:       This program resides in the low 1792 bytes of RAM
                                                           beginning at address 0 of the HEATH H8 computer.
                                                           Many routines used herein duplicate the functions
                                                           of the original PAM/8 monitor.  This version has
                                                           been written to maintain complete compatibility with
                                                           that monitor and applications which utilize its
                                                           various functions.  Because of the many capabilities
                                                           found in the Z-80 CPU which are not present in the
                                                           8080, the way many of the tasks are performed has
                                                           been revised to take advantage of the superior CPU.
                                                           In addition, many functions & routines have been
                                                           added to this montior to permit the user full Z-80
                                                           functionality from the front panel.

                                                           NOTE:  All of the major PAM/8 entry points have been
                                                           maintained to help ensure compatibility with existing
                                                           software including the HDOS & CP/M operating systems.

                                           Features:       Split Octal or Hexadecimal Display & Entry
                                                           Support for 4 MHz Operation
                                                           Support for Non-maskable interrupts
                                                           Support for Z80 Interrupt Mode 1
                                                           Display & Alter all primary & alternate CPU registers
                                                           Display & Alter index registers IX & IY
                                                           Display & Alter Interrupt Control Vector Register
                                                           Maintains all major PAM/8 entry points & functions
                                                           Maintains support for cassette tape
                                                           Supports hardware assisted 'Single Step' operation
                                                           User Real Time Clock in HH:MM:SS.mmm format.
                                                           Indexed display of memory through register and
                                                                   indirect through memory contents.


                                           Program ID:     Version 1.1
                                                           ROM Cyclic Redundancy Check = 10177

                                                   *

                                           SUBTTL  'Interrupt Processing.'
                                           PAGE
 0000'
                                        .COMMENT *

                                                Interrupts
                                                ----------

                                                FPM/80, like PAM/8, processes all interrupts received in
                                                Interrupt Mode 0 & Interrupt Mode 1.  In addition, the
                                                Z80 Non-maskable interrupt is processed by the monitor.

                                                They are processed as follows:

                                                Interrupt Mode 0

                                                RST     USE

                                                 0      Master Clear.  (Never used for I/O or Restart)

                                                 1      Clock Interrupt.  Normally processed by FPM/80,
                                                        the user may by setting a bit in .MFLAG process
                                                        the interrupt via a jump through the UIVEC table.
                                                        Upon entry of the users routine, the stack
                                                        contains:

                                                                (STACK+0)  = Return address to FPM/80
                                                                (STACK+2)  = (STACKPTR+18)
                                                                (STACK+4)  = (AF)
                                                                (STACK+6)  = (BC)
                                                                (STACK+8)  = (DE)
                                                                (STACK+10) = (HL)
                                                                (STACK+12) = (IY)
                                                                (STACK+14) = (IX)
                                                                (STACK+16) = (PC)

                                                        The user's routine should return to the monitor
                                                        after processing the interrupt via a 'RET',
                                                        without enabling interrupts.

                                                 2      Single Step.  Single step interrupts generated by
                                                        the front panel hardware are processed by FPM/80.
                                                        Any single step interrupt received when not in
                                                        monitor mode causes a jump through UIVEC+3.
                                                        Stack upon user routine entry contains:

                                                                (STACK+0)  = (STACKPTR+16)
                                                                (STACK+2)  = (AF)
                                                                (STACK+4)  = (BC)
                                                                (STACK+6)  = (DE)
                                                                (STACK+8)  = (HL)
                                                                (STACK+10) = (IY)
                                                                (STACK+12) = (IX)
                                                                (STACK+14) = (PC)

                                                        The user's routine should handle it's own return
                                                        from this interrupt.
```

```
                The following interrupts are vectored directly through UIVEC.
                The user must set up a jump entry in the proper location in
                UIVEC before any of these interrupts may occur.

                 3      I/O 3.  Causes a direct jump through UIVEC+6
                        (This vector is normally reserved for the console)

                 4      I/O 4.  Causes a direct jump through UIVEC+9

                 5      I/O 5.  Causes a direct jump through UIVEC+12

                 6      I/O 6.  Causes a direct jump through UIVEC+15

                 7      I/O 7.  Causes a conditional jump through UIVEC
                        depending on the current interrupt mode.  For
                        interrupt mode 0 (8080 mode), a jump is made
                        directly through UIVEC+18.  For interrupt mode
                        1, the user's routine is called through UIVEC+21.
                        Upon the user's return, the clock routine is called
                        and the interrupt is exited.

                Interrupt Mode 1

                All interrupts received in this mode are vectored via a
                CALL instruction through UIVEC+21.  Upon the user's return,
                the clock routine is called and the interrupt is exited.

                NOTE:

                The contents of UIVEC+21 are initialized to a RET
                instruction.  If the user sets IM1, he should first be
                sure to install the appropriate vector pointing to his
                routine.

                Non-maskable Interrupts

                All Non-maskable interrupts are passed directly through
                UIVEC+24.  Initially this vector is set to 'RETN' (Return
                from Non-maskable Interrupt).

                *

        SUBTTL  'Assembly Constants.'
        PAGE
```

```
 0000'

                                ;
                                ;       Assembly Constants
                                ;


                                ;       I/O Ports

 00F0                           IP.PAD  EQU     X'F0'                   ; Keypad input port
 00F0                           OP.CTL  EQU     X'F0'                   ; Front panel control output port
 00F0                           OP.DIG  EQU     X'F0'                   ; Digit select output port
 00F1                           OP.SEG  EQU     X'F1'                   ; Segment select output port

 00F9                           IP.TPC  EQU     X'F9'                   ; Tape control in
 00F9                           OP.TPC  EQU     X'F9'                   ; Tape control out
 00F8                           IP.TPD  EQU     X'F8'                   ; Tape data in
 00F8                           OP.TPD  EQU     X'F8'                   ; Tape data out

 003F                           OP.RAM  EQU     X'3F'                   ; DG-64D RAM/ROM Control Port



                                ;       ASCII Characters

 0016                           A.SYN   EQU     X'16'                   ; SYNC Character
 0002                           A.STX   EQU     X'02'                   ; STX Character



                                ;       Front panel hardware control bits

 0080                           CB.SPK  EQU     10000000B               ; Speaker Enable
 0040                           CB.CLI  EQU     01000000B               ; Clock Interrupt Enable
 0020                           CB.MTL  EQU     00100000B               ; Monitor Light
 0010                           CB.SSI  EQU     00010000B               ; Single Step Interrupt Enable



                                ;       Display Mode Flags  (In DSPMOD)

 0000                           DM.MR   EQU     0                       ; Memory Read
 0001                           DM.MW   EQU     1                       ; Memory Write (alter)
 0002                           DM.RR   EQU     2                       ; Register Read
 0003                           DM.RW   EQU     3                       ; Register Write (alter)



                                ;       Tape Equivalences

 0001                           RT.MI   EQU     1                       ; Record Type - Memory dump image
 0002                           RT.BP   EQU     2                       ; Record Type - BASIC program
 0003                           RT.CT   EQU     3                       ; Record Type - Compressed text

                                ;       Machine Instructions
 0076                        MI.HLT  EQU     X'76'           ; Halt
 00C9                        MI.RET  EQU     X'C9'           ; Return
 00DB                        MI.IN   EQU     X'DB'           ; Input
 00D3                        MI.OUT  EQU     X'D3'           ; Output
 003A                        MI.LDA  EQU     X'3A'           ; Load (A) Direct
 00E6                        MI.ANI  EQU     X'E6'           ; AND Immediate with (A)
 0011                        MI.LXID EQU     X'11'           ; Load Immediate Register (DE)
 003C                        MI.INCA EQU     X'3C'           ; Increment (A)

                             ;       Z80 Instructions

 ED46                        Z.IMO   EQU     X'ED46'         ; IMO (interrupt 0)
 45ED                        Z.RETN  EQU     X'45ED'         ; RETN (return from NMI) backwards



                             ;;;     User Option Bits.
                             ;
                             ;       These bits are set in .MFLAG

 0080                        UO.HLT  EQU     10000000B       ; Disable HLT processing by monitor
 0040                        UO.NFR  EQU     01000000B       ; No refresh of front panel
 0020                        UO.ALT  EQU     00100000B       ; Alternate registers are on stack
 0010                        UO.IM1  EQU     00010000B       ; Z80 Interrupt Mode 1 is set
 0008                        UO.HEX  EQU     00001000B       ; Current display is Hexadecimal
 0004                        UO.RCK  EQU     00000100B       ; Disable/enable real time clock
 0002                        UO.DDU  EQU     00000010B       ; Disable display update
 0001                        UO.CLK  EQU     00000001B       ; Allow user processing of clock

                                     .LIST

                                     SUBTTL  '8251 USART Bit Definitions.'
                                     PAGE
 0000'

                                        ;;;     8251 USART Bit Definitions
                                        ;
                                        ;       Mode Instruction Control Bits

 0040                                   UMI.1B  EQU     01000000B               ; 1 Stop bit
 0080                                   UMI.HB  EQU     10000000B               ; 1 1/2 Stop bits
 00C0                                   UMI.2B  EQU     11000000B               ; 2 Stop bits
 0020                                   UMI.PE  EQU     00100000B               ; Even parity
 0010                                   UMI.PA  EQU     00010000B               ; Use Parity
 0000                                   UMI.L5  EQU     00000000B               ; 5 Bit characters
 0004                                   UMI.L6  EQU     00000100B               ; 6 Bit characters
 0008                                   UMI.L7  EQU     00001000B               ; 7 Bit characters
 000C                                   UMI.L8  EQU     00001100B               ; 8 Bit characters
 0001                                   UMI.1X  EQU     00000001B               ; Clock X 1
 0002                                   UMI.16X EQU     00000010B               ; Clock X 16
 0003                                   UMI.64X EQU     00000011B               ; Clock X 64


                                        ;       Command Instruction Bits

 0040                                   UCI.IR  EQU     01000000B               ; Internal reset
 0020                                   UCI.RU  EQU     00100000B               ; Reader-on control flag
 0010                                   UCI.ER  EQU     00010000B               ; Error reset
 0004                                   UCI.RE  EQU     00000100B               ; Receiver enable
 0002                                   UCI.IE  EQU     00000010B               ; Enable interrupts flag
 0001                                   UCI.TE  EQU     00000001B               ; Transmitter enable


                                        ;       Status Read Commands

 0020                                   USR.FE  EQU     00100000B               ; Framing error
 0010                                   USR.OE  EQU     00010000B               ; Overrun error
 0008                                   USR.PE  EQU     00001000B               ; Parity error
 0004                                   USR.TE  EQU     00000100B               ; Transmitter empty
 0002                                   USR.RR  EQU     00000010B               ; Receiver ready
 0001                                   USR.TR  EQU     00000001B               ; Transmitter ready

                                                SUBTTL  'DG-64D Control Port Definitions'
                                                PAGE
```

```
0000'

                                     ;;;     Bit definitions for the DG-64D Bank Switching RAM Board
                                     ;

0080                                 ROMDIS  EQU     10000000B               ; ROM Disable

0001                                 BANK0   EQU     00000001B               ; Select Bank 0  (0 - 16K)
0002                                 BANK1   EQU     00000010B               ; Select Bank 1  (16 - 32K)
0004                                 BANK2   EQU     00000100B               ; Select Bank 2  (32 - 48K)
0008                                 BANK3   EQU     00001000B               ; Select Bank 3  (48 - 64K)

0000                                 BOARD0  EQU     00000000B               ; Select Board 0
0010                                 BOARD1  EQU     00010000B               ;    "      "  1
0020                                 BOARD2  EQU     00100000B               ;    "      "  2
0030                                 BOARD3  EQU     00110000B               ;    "      "  3
0040                                 BOARD4  EQU     01000000B               ;    "      "  4
0050                                 BOARD5  EQU     01010000B               ;    "      "  5
0060                                 BOARD6  EQU     01100000B               ;    "      "  6
0070                                 BOARD7  EQU     01110000B               ;    "      "  7

                                             NLIST
                                             .LIST

                                             SUBTTL  'Monitor System Initialization Program'
                                             PAGE
```

```
 026D'

                                        .PHASE  0                       ; Code is 100% relocatable!

                                       .COMMENT #       H8/DG-80 Z-80 System Initialization Program.

                                                        This code copies the panel monitor (FPM/80) & the
                                                        H17 disk code into low RAM.  If DG-64D memory
                                                        boards are in use, they should be addressed at
                                                        I/O port X'3F' as this routine will finish by
                                                        turning on all 64K on board #0 and disabling the
                                                        CPU ROM.  The system clock speed is also checked,
                                                        and if necessary, the proper disk timing constants
                                                        are altered for operation at 4 MHz.
                                                #

                                  ;
                                  ;     System Constant Definitions
                                  ;

 0900                             STACK   EQU     X'0900'               ; Our stack area
 1800                             H17ADR  EQU     X'1800'               ; Destination for H17 code
 0800                             H17LEN  EQU     X'0800'               ; Size of H17 code
 0000                             FPMADR  EQU     X'0000'               ; Destination for FPM/80 code
 0700                             FPMLEN  EQU     X'0700'               ; Size of FPM/80 code

                                        PAGE
 0000

                                ;;;     Main Code
                                ;

 0000                           SYSINIT:

 0000     F3                            DI                              ; Don't bother us, we're busy!

                                ;       Set RAM to known state on DG-64D

                                                                        ; Turn on 1st 48K on board #0
 0001     3E 07                         LD      A,BOARD0+BANK0+BANK1+BANK2
 0003     D3 3F                         OUT     (OP.RAM),A              ;

                                ;       Calculate our address so we can know where code source is.

 0005     31 0900                       LD      SP,STACK                ; Set up a stack
 0008     26 C9                         LD      H,MI.RET                ;       RET opcode
 000A     2E 3C                         LD      L,MI.INCA               ; INC A opcode
 000C     22 0008                       LD      (INT1),HL               ; Set it in at clock vector
 000F     CF                            RST     X'08'                   ; Cause an 'interrupt'

                                ;       Our address is now on the stack

 0010     2A 08FE                       LD      HL,(STACK-2)            ; Get the 'return' address
 0013     24                            INC     H                       ; FPM Monitor code is in next page
 0014     2E 00                         LD      L,0                     ;   on a 256 byte boundary.
 0016     E5                            PUSH    HL                      ; Save that address

 0017     11 0700                       LD      DE,FPMLEN               ; Figure out where the H17 code is
 001A     19                            ADD     HL,DE                   ;
 001B     11 1800                       LD      DE,H17ADR               ; Set destination
 001E     01 0800                       LD      BC,H17LEN               ;    and length.
 0021     ED B0                         LDIR                            ; Move it into place

                                        PAGE
 0023

                                ;
                                ;       Determine clock speed  (2 or 4 MHz)
                                ;

 0023    AF                             XOR     A                       ; Get a zero, clear 'Z'
 0024    FB                             EI                              ; Enable the interrupts

                                ;       At this point, we wait for a clock interrupt, at which time
                                ;       the (A) register will be incremented and the 'Z' flag will
                                ;       be cleared.

 0025    28 FE                  LOOP1:  JR      Z,LOOP1                 ; Just waiting around  (HO, HUM)

 0027    F3                             DI                              ;
 0028    3E F0                          LD      A,X'F0'                 ; Re-arm the clock
 002A    D3 F0                          OUT     (OP.CTL),A              ;

 002C    AF                             XOR     A                       ; Get a zero again & clear 'Z'
 002D    21 0000                        LD      HL,0                    ; Zero our counter
 0030    FB                             EI                              ; Enable interrupts

                                ;       This time, we will increment a counter until a clock interrupt
                                ;       occurs.  Based upon the contents of the counter after 2 ms, we
                                ;       can tell if we're running at 2 or 4 MHz.

 0031    23                     LOOP2:  INC     HL                      ; Bump counter
 0032    28 FD                          JR      Z,LOOP2                 ; No interrupt yet ...
 0034    F3                             DI                              ; Can't have any more interruptions.

 0035    25                             DEC     H                       ; Check counter
 0036    20 1A                          JR      NZ,.2MHZ                ; If (HL) = 001.224,  then 4MHz,
                                                                        ;  else, (HL) = 000.310 -> 2MHz.

 0038                           .4MHZ:                                  ; Patch H17 drivers for 4MHz operation
 0038    21 1C57                        LD      HL,X'1C57'              ;
 003B    34                             INC     (HL)                    ; READ2   Microsecond delay time
 003C    2E FE                          LD      L,X'FE'                 ;
 003E    34                             INC     (HL)                    ; WRITE2  Microsecond delay time
 003F    11 2028                        LD      DE,X'2028'              ;
 0042    21 1F5C                        LD      HL,X'1F5C'              ;
 0045    73                             LD      (HL),E                  ; D.WRITA   Guardband count for write
 0046    23                             INC     HL                      ;
 0047    23                             INC     HL                      ;
 0048    72                             LD      (HL),D                  ; D.WRITC   Two character delay before writing
 0049    16 B0                          LD      D,X'B0'                 ;
 004B    2E 65                          LD      L,X'65'                 ;
 004D    73                             LD      (HL),E                  ; D.WHDA    UDLY Count for hole debounce
 004E    23                             INC     HL                      ;
 004F    73                             LD      (HL),E                  ; D.WNHA    UDLY Count for hole debounce
 0050    23                             INC     HL                      ;
 0051    72                             LD      (HL),D                  ; D.WSCA    Loop count for 25 characters

                                        PAGE
 0052

 0052                              .2MHZ:
 0052     E1                               POP     HL                   ; Get the address of FPM
 0053     11 FFFB                          LD      DE,-5                ; Figure out where RAM fix-it-up is.
 0056     19                               ADD     HL,DE                ;
 0057     11 0800                          LD      DE,X'0800'           ; Area in low memory
 005A     01 0005                          LD      BC,5                 ; 5 bytes
 005D     ED B0                            LDIR                         ; Move it.

                                                                        ; (HL) := FPM source code address
 005F     11 0000                          LD      DE,FPMADR            ; Point to Destination
 0062     01 0700                          LD      BC,FPMLEN            ; Length
 0065     ED B0                            LDIR                         ; Move it down

                                   ;       End of system initialization

 0067     C3 0800                  EXIT:   JP      X'0800'              ; Go turn off ROM & enter FPM

                                           .LIST

                                   ;       This code is copied into low RAM above FPM/80 and then executed
                                   ;       to turn off the ROM & set the entire 64K address space on DG-64D
                                   ;       board # 0.

 000F                              ALL64K  EQU     BANK0+BANK1+BANK2+BANK3

 00FB     3E 8F                            LD      A,ROMDIS+ALL64K      ; Turn on all 64K & off ROM, board #0
 00FD     D3 3F                            OUT     (OP.RAM),A           ; Set on all 64K & off ROM
 00FF     C7                               RST     0                    ; Enter FPM

                                           TESTEQ  $,X'0100'            ; Monitor code must start here.

                                           .DEPHASE

                                           SUBTTL  'Hardware Interrupt Vectors.'
                                           PAGE
 036D'
                                                 .PHASE  0

                                         ;
                                         ;       Interrupt Vectors
                                         ;


                                         ;;;     Level 0 - Reset
                                         ;
                                         ;       This 'Interrupt' may not be processed by a user program.

 0000     11 2004                        INIT0:  LD      DE,PRSRAM               ; (DE) := RAM destination for code
 0003     21 061D                                LD      HL,PRSROM               ; (HL) := ROM copy of PRS code
 0006     18 33                                  JR      INIT                    ; Initialize


                                         ;       Level 1 - Clock

 0008     CD 005A                        INT1:   CALL    SAVALL                  ; Save user registers
 000B     16 00                                  LD      D,0                     ;
 000D     C3 0081                                JP      CLOCK                   ; Process clock interrupt


                                         ;;;     Level 2 - Single Step
                                         ;
                                         ;       If this interrupt is received when not in monitor mode,
                                         ;       then it is assumed to be generated by a user program
                                         ;       (single stepping ot breakpointing).  In such case, the
                                         ;       user program is entered through (UIVEC+3).

 0010     CD 005A                        INT2:   CALL    SAVALL                  ; Save registers
 0013     1A                                     LD      A,(DE)                  ; (A) := (CTLFLG)
 2009                                    .       DEFL    CTLFLG                  ;
 0014     C3 057A                                JP      STPRTN                  ; Step return

 0017     12                                     DEFB    BCDV                    ; Version number (BCD)

                                                 PAGE
 0018

                                 ;;;     I/O Interupt Vectors.
                                 ;
                                 ;       Interrupts 3 through 7 are available for general I/O use.
                                 ;
                                 ;       These interrupts are not supported by FPM/80, and should
                                 ;       never occur unless the user has supplied the necessary
                                 ;       handler routines (through UIVEC).

 0018    C3 2025                 INT3:   JP      UIVEC+6                 ; Jump to user routine

                                         .LIST

                                 ;       Interrupt 4

 0020    C3 2028                 INT4:   JP      UIVEC+9                 ; Jump to user routine

                                         .LIST

 0028    C3 202B                 INT5:   JP      UIVEC+12                ; Jump to user routine

                                 ;;;     DLY - Delay Time Interval
                                 ;
                                 ;       ENTRY:  (A) = Millisecond delay count / 2
                                 ;       EXIT:   None
                                 ;       USES:   A,F

                                         TESTEQ  DLY,X'002B'

 002B    F5                      DLY:    PUSH    AF                      ; Save count
 002C    AF                              XOR     A                       ; Don't sound horn
 002D    C3 0263                         JP      HRNO                    ;  but process as horn.

 0030    C3 202E                 INT6:   JP      UIVEC+15                ; Jump to user routine

 0033    3E D0                   GO.:    LD      A,CB.SSI+CB.CLI+CB.SPK  ; Off monitor mode light
 0035    C3 0587                         JP      SST1                    ; Return to user program

 0038    C3 060A                 INT7:   JP      IMOD1                   ; Go test interrupt mode

                                         SUBTTL  'Master Clear Processing.'
                                         PAGE
 003B

                                ;;;     INIT - Initialize System.
                                ;
                                ;       INIT is called whenever a hardware master clear is initiated.
                                ;
                                ;       Setup FPM/80 control cells in RAM.
                                ;       Decode how much memory exists, set up stack pointer, and
                                ;       enter the monitor loop.
                                ;
                                ;       ENTRY:  From master clear
                                ;       EXIT:   Into FPM/80 main loop

 003B    01 0007                INIT:   LD      BC,PRSL                 ; (BC) := Length of PRS code
 003E    ED B0                          LDIR                            ; Copy it into RAM

 0400                           SINCR   EQU     X'0400'                 ; Search increment for sizing memory

 0040    11 0400                        LD      DE,SINCR                ; (DE) := Search increment (1K)
 0043    21 2000                        LD      HL,START                ; (HL) := 1st RAM - Search increment

                                ;       Determine memory limit

 0046    77                     INIT1:  LD      (HL),A                  ; Restore value read
 0047    19                             ADD     HL,DE                   ; Increment trial address
 0048    38 05                          JR      C,INIT1A                ; If at end of 64K space.
 004A    7E                             LD      A,(HL)                  ; (A) := Current memory value
 004B    35                             DEC     (HL)                    ; Try to change it ...
 004C    BE                             CP      (HL)                    ; Did we do it?
 004D    20 F7                          JR      NZ,INIT1                ; Yes, go try for more

 004F    C3 05CC                INIT1A: JP      INIT2                   ; Finish up initialization


                                ;;;     IXIT1 - Conclusion of INTXIT
                                ;

 0052    FD E1                  IXIT1:  POP     IY                      ; Restore
 0054    DD E1                          POP     IX                      ;
 0056    FB                             EI                              ;
 0057    C9                             RET                             ; To user

                                        .LIST

                                        SUBTTL  'Interrupt Time Subroutines.'
                                        PAGE
 005A

                                 ;;;     SAVALL - Save all registers on stack
                                 ;
                                 ;       SAVALL is called when an interrupt is accepted, in order to
                                 ;       save the contents of the user's registers on the stack.
                                 ;
                                 ;       ENTRY:  Called directly from interrupt routine
                                 ;       EXIT:   All registers pushed onto stack.
                                 ;               If not yet in monitor mode, REGPTR = Address of registers
                                 ;               on stack.
                                 ;               (DE) = Address of CTLFLG.

                                         TESTEQ  SAVALL,X'005A'

 005A    DD E3                   SAVALL: EX      (SP),IX                 ; Set (IX) on top of stack
                                         IRP     ?REG,<IY,HL,DE,BC,AF>   ; Save registers on stack
                                         PUSH    ?REG                    ;; PUSH Register
                                         ENDM
 005C    FD E5           +               PUSH    IY
 005E    E5              +               PUSH    HL
 005F    D5              +               PUSH    DE
 0060    C5              +               PUSH    BC
 0061    F5              +               PUSH    AF
 0062    C3 05FF                         JP      SAVALX                  ; Do loop around NMI.

 0065    00                              DEFB    0                       ; Extra byte

                                 ;       NMI Entry point.

                                         TESTEQ  NMI,X'0066'

 0066    C3 2037                 NMI:    JP      UIVEC+24                ; Go to NMI vector in UIVEC

 0069    00                              DEFB    0                       ; Extra byte

                                 ;       Continue SAVALL routine

 006A    1A                      SAVALR: LD      A,(DE)                  ; Get CTLFLG
 006B    2F                              CPL                             ;
 006C    E6 30                           AND     CB.MTL+CB.SSI           ; Check for S. Step or Monitor mode
 006E    28 07                           JR      Z,SAV1                  ; It was, so exit.
 0070    21 0000                         LD      HL,0                    ; Set up REGPTR
 0073    39                              ADD     HL,SP                   ;
 0074    22 201D                         LD      (REGPTR),HL             ; Set it.
 0077    DD E9                   SAV1:   JP      (IX)                    ; Return

 0079    00                              DEFB    0                       ; Extra byte

                                         PAGE
 007A

                               ;      Return to program from interrupt

                                      TESTEQ  INTXIT,X'007A'

 007A                          INTXIT:                                ; Remove fake 'stack pointer' &
                                      IRP     ?REG,<AF,AF,BC,DE,HL>   ;   and user's registers.
                                      POP     ?REG                    ;; POP Register
                                      ENDM                            ;
 007A     F1              +           POP     AF
 007B     F1              +           POP     AF
 007C     C1              +           POP     BC
 007D     D1              +           POP     DE
 007E     E1              +           POP     HL
 007F     18 D1                       JR      IXIT1                   ; Go to rest of routine

                                      SUBTTL  'Process Clock Interrupts.'
                                      PAGE
 0081

                                    ;;;     CLOCK - Process clock interrupts.
                                    ;
                                    ;       CLOCK is entered whenever a 2 ms. clock interrupt is
                                    ;       processed.
                                    ;
                                    ;       TICCNT is incremented every interrupt.

                                            TESTEQ  CLOCK,X'0081'

 0081    2A 201B                    CLOCK:  LD      HL,(TICCNT)             ; Get TIC counter
 0084    23                                 INC     HL                      ; Increment it
 0085    22 201B                            LD      (TICCNT),HL             ; Now put it back
 0088    CD 05AC                            CALL    SIM                     ; Set interrupt mode

                                    ;;;     Refresh Front Panel.
                                    ;
                                    ;       This code displays the appropriate pattern on the
                                    ;       front panel LEDs.  The LEDs are painted in reverse order,
                                    ;       from right to left, one per interrupt.

 008B    7E                                 LD      A,(HL)                  ;
 008C    47                                 LD      B,A                     ; (B) := Current flag
 008D    E6 40                              AND     UO.NFR                  ; See if front panel refresh wanted
 008F    23                                 INC     HL                      ;
 0090    7E                                 LD      A,(HL)                  ; (A) := CTLFLG
 0091    4A                                 LD      C,D                     ; (C) = 0 in case no panel display
 0092    20 09                              JR      NZ,CLK3                 ; If not ...
 0094    23                                 INC     HL                      ; (HL) := (REFIND)
 0095    35                                 DEC     (HL)                    ; Decrement Digit Index
 0096    20 02                              JR      NZ,CLK2                 ; If not wrap-around
 0098    36 09                              LD      (HL),9                  ; Wrap digit display around
 009A    5E                         CLK2:   LD      E,(HL)                  ;
 009B    19                                 ADD     HL,DE                   ; (HL) := Address of pattern
 009C    4B                                 LD      C,E                     ;
 009D    B1                         CLK3:   OR      C                       ; (A) := Index + fixed bits
 009E    D3 F0                              OUT     (OP.DIG),A              ; Select digit
 00A0    7E                                 LD      A,(HL)                  ;
 00A1    D3 F1                              OUT     (OP.SEG),A              ; Select segment

                                    ;       See if time to decode display values.

 00A3    2E 1B                              LD      L,LOW(TICCNT)           ;
 00A5    7E                                 LD      A,(HL)                  ;
 00A6    E6 1F                              AND     X'1F'                   ; Every 32 interrupts
 00A8    CC 0370                            CALL    Z,UFD                   ; Update front panel displays

                                    ;       Exit from clock interrupt

 00AB    01 2009                            LD      BC,CTLFLG               ;
 00AE    0A                                 LD      A,(BC)                  ; (A) := CTLFLG
 00AF    E6 20                              AND     CB.MTL                  ; Monitor mode?
 00B1    20 C7                              JR      NZ,INTXIT               ;   Yep ...
 00B3    0B                                 DEC     BC                      ;
 00B4    0A                    LD      A,(BC)                  ; (A) := .MFLAG
 00B5    17                    RLA                             ;
 00B6    38 0E                 JR      C,CLK4                  ; Skip it.

                       ;       Not in monitor mode, check for a HALT.

 00B8    3E 0E                 LD      A,14                    ; (A) := Index of (PC) register
 00BA    CD 032A               CALL    LRA.                    ; Locate it on the stack
 00BD    5E                    LD      E,(HL)                  ;
 00BE    23                    INC     HL                      ;
 00BF    56                    LD      D,(HL)                  ; (DE) := User's (PC) contents
 00C0    1B                    DEC     DE                      ;
 00C1    1A                    LD      A,(DE)                  ; Get last instruction executed
 00C2    FE 76                 CP      MI.HLT                  ; Was it a HALT?
 00C4    28 0C                 JR      Z,ERROR                 ; Yes, go to monitor mode.

                       ;       Check for 'Return to Monitor' key entry.

 00C6    DB F0         CLK4:   IN      A,(IP.PAD)              ; Get keypad
                               IF      RTMO AND NOT BPROM AND NOT DCROM
                               CP      X'2E'                   ; See if '#' and '0' keys
                               ELSE
 00C8    FE 2F                 CP      X'2F'                   ; See if '#' key
                               ENDC
 00CA    C2 0572               JP      NZ,CUI1                 ; No, allow user processing of clock.
 00CD    18 03                 JR      ERROR                   ; Go to ERROR

                               .LIST

                               SUBTTL  'MTR - FPM/80 Main Executive Loop.'
                               PAGE
 00D2

                                        ;;;     ERROR - Command Error.
                                        ;
                                        ;       ERROR is called as a 'bail out' routine.  It resets the
                                        ;       operational mode to monitor mode & restores the stack pointer.
                                        ;
                                        ;       ENTRY:  None
                                        ;       EXIT:   To MTR loop.
                                        ;               CTLFLG set
                                        ;               .MFLAG cleared
                                        ;       USES:   All

                                                TESTEQ  ERROR,X'00D2'

 00D2     21 2008                       ERROR:  LD      HL,.MFLAG               ;
 00D5     7E                                    LD      A,(HL)                  ; (A) := .MFLAG
 00D6     E6 BD                                 AND     X'FF'-UO.DDU-UO.NFR     ; Re-enable displays
 00D8     77                                    LD      (HL),A                  ; Replace
 00D9     23                                    INC     HL                      ; Restore CTLFLG
 00DA     36 F0                                 LD      (HL),CB.SSI+CB.MTL+CB.CLI+CB.SPK
 00DC     FB                                    EI
 00DD     2A 201D                               LD      HL,(REGPTR)             ;
 00E0     F9                                    LD      SP,HL                   ; Restore (SP) to empty state
 00E1     CD 025E                               CALL    ALARM                   ; Sound the alarm!


                                        ;;;     MTR - Monitor loop.
                                        ;
                                        ;       This is the main executive loop for the front panel monitor.

 00E4     FB                            MTR:    EI                              ; On interrupts

 00E5     21 00E5                       MTR1:   LD      HL,MTR1                 ;
 00E8     E5                                    PUSH    HL                      ; Set 'MTR1' as return address
 00E9     01 2007                               LD      BC,DSPMOD               ; (BC) := DSPMOD
 00EC     0A                                    LD      A,(BC)                  ;
 00ED     E6 01                                 AND     1                       ; (A) = 1 if alter mode.
 00EF     2F                                    CPL                             ;
 00F0     32 2006                               LD      (DSPROT),A              ; Rotate LED periods if alter mode

                                        ;       Read keypad.

 00F3     CD 03B0                               CALL    RCK                     ; Read front panel keypad
 00F6     2A 2014                               LD      HL,(ABUSS)              ;
 00F9     FE 0A                                 CP      10                      ;
 00FB     CD 042E                               CALL    CKHEX                   ; Check for hexadecimal input
 00FE     30 06                                 JR      NC,MTR4                 ; If in 'always valid' group
 0100     5F                                    LD      E,A                     ; Save value
 2007                                   .       DEFL    DSPMOD                  ;
 0101     0A                                    LD      A,(BC)                  ; (A) := DSPMOD
 0102     0F                                    RRCA                            ;
 0103     38 24                                 JR      C,MTR5                  ; If in alter mode.
 0105     7B                                    LD      A,E                     ; (A) := code
```

```
                                ;       Have a command (not a value).

0106    D6 04                   MTR4:   SUB     4               ; (A) := Command
0108    38 C8                           JR      C,ERROR         ; Wrong!
010A    5F                              LD      E,A             ;
010B    E5                              PUSH    HL              ; Save ABUSS value
010C    21 011D                         LD      HL,MTRA         ; Point to table
010F    16 00                           LD      D,0             ;
0111    19                              ADD     HL,DE           ; (HL) := Address of table entry
0112    5E                              LD      E,(HL)          ; (E) := Table entry
0113    19                              ADD     HL,DE           ; (HL) := Address of processor
0114    E3                              EX      (SP),HL         ; Set address, (HL) := (ABUSS)
0115    11 2005                         LD      DE,REGI         ; (DE) := Adderss of register index
2007                            .       DEFL    DSPMOD          ;
0118    0A                              LD      A,(BC)          ; (A) := DSPMOD
0119    E6 02                           AND     2               ; Set 'Z' flag if displaying memory
011B    0A                              LD      A,(BC)          ; (A) := DSPMOD (again)
011C    C9                              RET                     ; Jump to processor

                                ;       Processor jump table.

011D    75                      MTRA:   DEFB    GO-$            ; 4 - Go
011E    5F                              DEFB    IN-$            ; 5 - Input byte
011F    61                              DEFB    OUT-$           ; 6 - Output byte
0120    75                              DEFB    SSTEP-$         ; 7 - Single step
0121    90                              DEFB    RMEM-$          ; 8 - Cassette load
0122    DA                              DEFB    WMEM-$          ; 9 - Cassette dump
0123    37                              DEFB    NEXT-$          ; A - Increment display
0124    43                              DEFB    LAST-$          ; B - Decrement display
0125    89                              DEFB    FUNCT-$         ; C - Function
0126    30                              DEFB    R$W-$           ; D - Toggle Display/Alter modes
0127    4C                              DEFB    MEMM-$          ; E - Memory mode
0128    1C                              DEFB    REGM-$          ; F - Register mode


                                ;;;     Process memory / register alterations.
                                ;
                                ;       This code is entered if in alter mode & a valid digit
                                ;       (HEX or OCTAL) was entered.

0129    0F                      MTR5:   RRCA                    ;
012A    7B                              LD      A,E             ; (A) := Value
012B    DA 0137                         JP      C,MTR6          ; Is register
012E    37                              SCF                     ; Indicate 1st digit is in (A)
012F    CD 0336                         CALL    INBYTE          ; Get octal byte
0132    23                              INC     HL              ; Display next location

                                        PAGE
```

```
 0133

                                  ;;;     SAE - Store ABUSS and exit.
                                  ;
                                  ;       ENTRY:  (HL) = ABUSS value
                                  ;       EXIT:   To (RET)
                                  ;       USES:   None

 0133     22 2014                 SAE:    LD      (ABUSS),HL              ;
 0136     C9                              RET                             ;


                                  ;       Alter register.

 0137     F5                      MTR6:   PUSH    AF                      ; Save code
 0138     CD 0327                         CALL    LRA                     ; Locate register on stack
 013B     A7                              AND     A                       ;
 013C     CA 0455                         JP      Z,BADALT                ; Not allowed to alter (SP)
 013F     23                              INC     HL                      ;
 0140     F1                              POP     AF                      ; Restore value & carry flag
 0141     C3 0332                         JP      INWORD                  ; Input octal address

                                          SUBTTL  'Monitor Task Subroutines.'
                                          PAGE
 0144

                                 ;;;     REGM - Enter register display mode.
                                 ;
                                 ;       ENTRY:  (A)  = (DSPMOD)
                                 ;               (BC) = DSPMOD

 0144    3E 02                   REGM:   LD      A,2                     ; Set display register mode
 2007                            .       DEFL    DSPMOD                  ;
 0146    02                              LD      (BC),A                  ;
 0147    0B                              DEC     BC                      ; (BC) := DSPROT
 0148    AF                              XOR     A                       ;
 0149    02                              LD      (BC),A                  ; Set all periods on.
 014A    CD 03B0                         CALL    RCK                     ; Read keypad
 014D    3D                              DEC     A                       ; Displace
 014E    FE 08                           CP      8                       ;
 0150    D2 00D2                         JP      NC,ERROR                ; Not 1-8
 0153    C3 059B                         JP      TSTREG                  ; Go test if (PC) or (IY)


                                 ;;;     R$W - Toggle display / alter modes.
                                 ;
                                 ;       ENTRY:  (A)  = (DSPMOD)
                                 ;               (BC) = DSPMOD

 2007                            .       DEFL    DSPMOD                  ;
 0156    EE 01                   R$W:    XOR     1                       ; Toggle bit 0
 0158    02                              LD      (BC),A                  ; Store it
 0159    C9                              RET                             ;


                                 ;;;     NEXT - Increment display element.
                                 ;
                                 ;       ENTRY:  (HL) = (ABUSS)
                                 ;               (DE) = REGI

 015A    23                      NEXT:   INC     HL                      ; Increment memory address
 015B    28 D6                           JR      Z,SAE                   ; If memory display, go store & exit.

                                 ;       In register mode.

 2005                            .       DEFL    REGI                    ;
 015D    1A                              LD      A,(DE)                  ; (A) := (REGI)
 015E    C6 02                           ADD     A,2                     ; Bump register index
 0160    12                              LD      (DE),A                  ; Set it.
 0161    FE 10                           CP      16                      ; Gone too far?
 0163    D8                              RET     C                       ; No, so exit.
 0164    AF                              XOR     A                       ; Yep, so wrap around to (SP)
 0165    12                              LD      (DE),A                  ; Set it right this time
 0166    C9                      ABORT:  RET                             ;
```

```
                                ;;;     LAST - Decrement display element.
                                ;
                                ;       ENTRY:  (HL) = (ABUSS)
                                ;               (DE) = REGI

 0167    2B                     LAST:   DEC     HL                      ; Decrement memory address
 0168    28 C9                          JR      Z,SAE                   ; If memory display, go store & exit.

                                ;       In register mode.

 2005                           .       DEFL    REGI                    ;
 016A    1A                             LD      A,(DE)                  ; (A) := (REGI)
 016B    D6 02                          SUB     2                       ; Previous register index
 016D    12                             LD      (DE),A                  ; Set it.
 016E    D0                             RET     NC                      ; If ok
 016F    3E 0E                          LD      A,14                    ; Gone too far, set for (PC).
 0171    12                             LD      (DE),A                  ; Set it.
 0172    C9                             RET                             ;


                                ;;;     MEMM - Enter memory display mode.
                                ;
                                ;       ENTRY:  (BC) = DSPMOD

 0173    AF                     MEMM:   XOR     A                       ; (A) := 0
 2007                           .       DEFL    DSPMOD                  ;
 0174    02                             LD      (BC),A                  ; Set display memory mode
 0175    0B                             DEC     BC                      ; (BC) := DSPROT
 0176    02                             LD      (BC),A                  ; Set all periods on
 0177    21 2015                        LD      HL,ABUSS+1              ; Point to address buss
 017A    C3 0332                        JP      INWORD                  ; Go set address to display

                                        PAGE
 017D

                                        ;;;     IN - Input data byte.
                                        ;
                                        ;;;     OUT - Output data byte.
                                        ;
                                        ;       ENTRY:  (HL) = (ABUSS)

 017D    0E DB                          IN:     LD      C,MI.IN                 ; (C) := 'IN' opcode
 017F    11                                     DEFB    MI.LXID                 ; Gobble up next instruction

 0180    0E D3                          OUT:    LD      C,MI.OUT                ; (C) := 'OUT' opcode
 0182    7C                                     LD      A,H                     ; (A) := Value
 0183    45                                     LD      B,L                     ; (B) := Port address
 0184    F3                                     DI                              ; No interrupts, please
 0185    ED 43 2002                             LD      (IOWRK),BC              ; Set up I/O instruction in RAM
 0189    CD 2002                                CALL    IOWRK                   ; Do I/O
 018C    FB                                     EI                              ; Ok to interrupt now ...
 018D    67                                     LD      H,A                     ; (H) := Value
 018E    18 A3                                  JR      SAE                     ; Go store & exit.

                                                .LIST

                                                SUBTTL  'GO & Single Step Functions'
                                                PAGE
 0192

                                  ;;;     GO - Return to user mode.
                                  ;
                                  ;       ENTRY:  None

 0192    C3 0033                  GO:     JP      GO.                     ; Routine is in 'waste space'


                                  ;;;     SSTEP - Single Step instruction.
                                  ;
                                  ;       ENTRY:  None

                                          TESTEQ  SSTEP,X'0195'

 0195    F3                       SSTEP:  DI                              ; Disable interrupts until right time
 0196    21 2009                          LD      HL,CTLFLG               ; Point to CTLFLG
 0199    CB A6                            RES     4,(HL)                  ; Reset Single Step Bit

                                  ;       Clear 'return' address, fake (SP), and all user registers

                                          IRP     ?REG,<AF,AF,AF,BC,DE,HL,IY,IX>
                                          POP     ?REG                    ;; POP Register
                                          ENDM
 019B    F1                +              POP     AF
 019C    F1                +              POP     AF
 019D    F1                +              POP     AF
 019E    C1                +              POP     BC
 019F    D1                +              POP     DE
 01A0    E1                +              POP     HL
 01A1    FD E1             +              POP     IY
 01A3    DD E1             +              POP     IX

 01A5    F5                               PUSH    AF                      ; Save PSW
 01A6    3A 2009                          LD      A,(CTLFLG)              ; Get CTLFLG
 01A9    D3 F0                            OUT     (OP.CTL),A              ; Arm Single Step interrupt
 01AB    F1                               POP     AF                      ; Restore User's PSW
 01AC    FB                               EI                              ; OK to interrupt now
 01AD    C9                               RET                             ; Go do user's instruction


                                  ;;;     FUNCT - Function Key Processor
                                  ;
                                  ;

 01AE    C3 0461                  FUNCT:  JP      FUNCT.                  ; Go to real routine

                                          SUBTTL  'Cassette Load Routines'
                                          PAGE
 01B1

                                ;;;     RMEM - Load memory from tape.
                                ;

                                        TESTEQ  RMEM,X'01B1'

 01B1     21 02A4               RMEM:   LD      HL,TPABT                ; Set up error exit address.
 01B4     22 2019                       LD      (TPERRX),HL             ;
                                ;       JP      LOAD                    ;


                                ;;;     LOAD - Load memory from tape.
                                ;
                                ;       Read the next record from cassette tape using the load
                                ;       address in the tape record.
                                ;
                                ;       ENTRY:  (HL) = Error exit address
                                ;       EXIT:   User (PC) on stack set to entry address
                                ;               To caller if OK
                                ;               To error exit if tape errors detected.

                                        TESTEQ  LOAD,X'01B7'

 FE00                           @@WORK  DEFL    (X'100'-RT.MI)*256-256  ; @@WORK := - Required type and #

 01B7     01 FE00               LOAD:   LD      BC,@@WORK               ; (BC) := @@WORK

 01BA     CD 02B5               LOA0:   CALL    SRS                     ; Scan for record start
 01BD     6F                            LD      L,A                     ; (HL) := Count
 01BE     EB                            EX      DE,HL                   ; (DE) := Count, (HL) := Type and #
 01BF     0D                            DEC     C                       ; (C) := - Next #
 01C0     09                            ADD     HL,BC                   ;
 01C1     7C                            LD      A,H                     ;
 01C2     C5                            PUSH    BC                      ; Save type and #
 01C3     F5                            PUSH    AF                      ; Save type code
 01C4     E6 7F                         AND     X'7F'                   ; Clear end flag bit
 01C6     B5                            OR      L                       ;
 01C7     3E 02                         LD      A,2                     ; Sequence error
 01C9     C2 0285                       JP      NZ,TPERR                ; If not right type or sequence
 01CC     CD 02D5                       CALL    RNP                     ; Read entry address
 01CF     44                            LD      B,H                     ;
 01D0     4F                            LD      C,A                     ; (BC) := Entry address
 01D1     3E 0E                         LD      A,14                    ;
 01D3     D5                            PUSH    DE                      ;
 01D4     CD 032A                       CALL    LRA.                    ; Locate register address
 01D7     D1                            POP     DE                      ;
 01D8     71                            LD      (HL),C                  ; Set entry address on stack
 01D9     23                            INC     HL                      ;
 01DA     70                            LD      (HL),B                  ;
 01DB     CD 02D5                       CALL    RNP                     ; Read load address
 01DE     6F                            LD      L,A                     ; (HL) := Address, (DE) := Count
 01DF     22 2000                       LD      (START),HL              ;

 01E2     CD 02D9               LOA1:   CALL    RNB                     ; Read a byte
 01E5     77                        LD      (HL),A              ; Store it
 01E6     22 2014                   LD      (ABUSS),HL          ; Set ABUSS for display
 01E9     23                        INC     HL                  ; Point to next loc'n
 01EA     1B                        DEC     DE                  ; Decrement count
 01EB     7A                        LD      A,D                 ; See if more to do
 01EC     B3                        OR      E                   ;
 01ED     C2 01E2                   JP      NZ,LOA1             ; Yes ...

 01F0     CD 027A                   CALL    CTC                 ; Check tape checksum

                            ;       Read next block

 01F3     F1                        POP     AF                  ; (A) := File type byte
 01F4     C1                        POP     BC                  ; (BC) := - Last type, last #
 01F5     07                        RLCA                        ;
 01F6     DA 025B                   JP      C,TFT               ; All done ... Go turn off tape.
 01F9     C3 01BA                   JP      LOA0                ; ... else, read another record.

                                    SUBTTL  'Cassette Dump Routines'
                                    PAGE
 01FC

                                    ;;;     WMEM - Dump memory to tape.
                                    ;

                                            TESTEQ  WMEM,X'01FC'

 01FC    21 02A4                    WMEM:   LD      HL,!PABT                ; Set up error exit
 01FF    22 2019                            LD      (TPERRX),HL             ;


                                    ;;;     DUMP - Dump memory to tape.
                                    ;
                                    ;       Dump specified memory range to cassette tape.
                                    ;
                                    ;       ENTRY:  (START) = Start address of dump
                                    ;               (ABUSS) = End address of dump
                                    ;               (PC)    = Entry address
                                    ;       EXIT:   To caller

                                            TESTEQ  DUMP,X'0202'

 0202    3E 01                      DUMP:   LD      A,UCI.TE                ;
 0204    D3 F9                              OUT     (OP.TPC),A              ; Set up tape control
 0206    3E 16                              LD      A,A.SYN                 ; (A) := SYNC character
 0208    26 20                              LD      H,32                    ; (H) := No. of SYNC characters
 020A    CD 0314                    WME1:   CALL    WNB                     ; Write SYNC
 020D    25                                 DEC     H                       ; One less to do ...
 020E    C2 020A                            JP      NZ,WME1                 ; If not done
 0211    3E 02                              LD      A,A.STX                 ; (A) := STX character
 0213    CD 0314                            CALL    WNB                     ; Write 'Start of Text'
 0216    6C                                 LD      L,H                     ; (HL) := 0
 0217    22 2017                            LD      (CRCSUM),HL             ; Clear CRC
 8101                               @@WORK  DEFL    (RT.MI+X'80')*256+1     ; 1st & last MI record
 021A    21 8101                            LD      HL,@@WORK               ; (HL) := @@WORK
 021D    CD 030F                            CALL    WNP                     ; Write header
 0220    2A 2000                            LD      HL,(START)              ;
 0223    EB                                 EX      DE,HL                   ; (DE) := Start address of dump
 0224    2A 2014                            LD      HL,(ABUSS)              ; (HL) := Ending address of dump
 0227    23                                 INC     HL                      ; Compute with stop + 1
 0228    7D                                 LD      A,L                     ; Compute (HL) := (HL) - (DE)
 0229    93                                 SUB     E                       ;
 022A    6F                                 LD      L,A                     ;
 022B    7C                                 LD      A,H                     ;
 022C    9A                                 SBC     A,D                     ;
 022D    67                                 LD      H,A                     ; (HL) := Count
 022E    CD 030F                            CALL    WNP                     ; Write count
 0231    E5                                 PUSH    HL                      ;
 0232    3E 0E                              LD      A,14                    ; (PC) index
 0234    D5                                 PUSH    DE                      ;
 0235    CD 032A                            CALL    LRA.                    ; Locate (PC) on stack
 0238    7E                                 LD      A,(HL)                  ;
 0239    23                                 INC     HL                      ;
 023A    66                                 LD      H,(HL)                  ;
 023B    6F                                 LD      L,A                     ; (HL) := Contents of (PC)
 023C    CD 030F                 CALL    WNP                     ; Write entry address
 023F    E1                      POP     HL                      ; (HL) := Address
 0240    D1                      POP     DE                      ; (DE) := Count
 0241    CD 030F                 CALL    WNP                     ;

 0244    7E              WME2:   LD      A,(HL)                  ; Get a byte
 0245    CD 0314                 CALL    WNB                     ; Write it
 0248    22 2014                 LD      (ABUSS),HL              ; Set ABUSS for display
 024B    23                      INC     HL                      ; Point to next byte
 024C    1B                      DEC     DE                      ; Decrement count
 024D    7A                      LD      A,D                     ;
 024E    B3                      OR      E                       ; Is (DE) = 0?
 024F    C2 0244                 JP      NZ,WME2                 ; No, do some more.

                         ;       Write CRC.

 0252    2A 2017                 LD      HL,(CRCSUM)             ; Get CRC
 0255    CD 030F                 CALL    WNP                     ; Write it.
 0258    CD 030F                 CALL    WNP                     ;
                         ;       JP      TFT


                         ;;;     TFT - Turn off tape.
                         ;
                         ;       Stop the cassette player/recorder.

                                 TESTEQ  TFT,X'025B'

 025B    AF              TFT:    XOR     A                       ; Get a 0
 025C    D3 F9                   OUT     (OP.TPC),A              ; Send it.

                                 SUBTTL  'Front Panel Horn Routine'
                                 PAGE
 025E

                                  ;;;     HORN - Sound the front panel horn
                                  ;
                                  ;       ENTRY:  (A) = ms/2
                                  ;       EXIT:   None
                                  ;       USES:   A,F

                                          TESTEQ  ALARM,X'025E'
                                          TESTEQ  HORN,X'0260'

 025E    3E 3C                    ALARM:  LD      A,120/2                 ; 120 ms beep

 0260    F5                       HORN:   PUSH    AF                      ; Save count
 0261    3E 80                            LD      A,CB.SPK                ; Turn on speaker

 0263    E3                       HRN0:   EX      (SP),HL                 ; Save (HL), (H) := Count
 0264    D5                               PUSH    DE                      ; Save DE
 0265    EB                               EX      DE,HL                   ; (D) := Loop count
 0266    21 2009                          LD      HL,CTLFLG               ; Point to CTLFLG
 0269    AE                               XOR     M                       ; Toggle horn bit
 026A    5E                               LD      E,(HL)                  ; (E) := Old CTLFLG value
 026B    77                               LD      (HL),A                  ; Set on horn in CTLFLG
 026C    2E 1B                            LD      L,LOW(TICCNT)           ;

 026E    7A                               LD      A,D                     ; (A) := Cycle count
 026F    86                               ADD     A,(HL)                  ; Value to wait for in TICCNT

 0270    BE                       HRN2:   CP      (HL)                    ; (A) = (TICCNT)?
 0271    C2 0270                          JP      NZ,HRN2                 ; No, wait ...

 0274    2E 09                            LD      L,LOW(CTLFLG)           ; (HL) := CTLFLG
 0276    73                               LD      (HL),E                  ; Restore old CTLFLG value
 0277    D1                               POP     DE                      ;
 0278    E1                               POP     HL                      ;
 0279    C9                               RET                             ;

                                          SUBTTL  'Casette Tape Processing Subroutines'
                                          PAGE
 027A

                                 ;;;     CTC - Verify checksum.
                                 ;
                                 ;       ENTRY:  Tape just before CRC.
                                 ;       EXIT:   To caller if Ok.
                                 ;               To TPERR if bad.
                                 ;       USES:   A,F,H,L

                                         TESTEQ  CTC,X'027A'

 027A     CD 02D5                CTC:    CALL    RNP                     ; Read next pair
 027D     2A 2017                        LD      HL,(CRCSUM)             ;
 0280     7C                             LD      A,H                     ;
 0281     B5                             OR      L                       ;
 0282     C8                             RET     Z                       ; Return if Ok.
 0283     3E 01                          LD      A,1                     ; Checksum error.
                                 ;       JP      TPERR                   ; (B) := Code


                                 ;;;     TPERR - Process tape error.
                                 ;
                                 ;       Display error number in low byte of (ABUSS)
                                 ;
                                 ;       If error number is even, don't allow '#'
                                 ;       If error number is odd, allow '#'
                                 ;
                                 ;       ENTRY:  (A) = Error number

                                         TESTEQ  TPERR,X'0285'

 0285     32 2014                TPERR:  LD      (ABUSS),A               ;
 0288     47                             LD      B,A                     ; (B) := Code
 0289     CD 025B                        CALL    TFT                     ; Stop tape.

                                 ;       Is '#', return (if parity error)

 028C     E6                             DB      MI.ANI                  ; Fall through with 'C' clear
 028D     78                     TER3:   LD      A,B                     ;

 028E     0F                             RRCA                            ;
 028F     D8                             RET     C                       ; Return if Ok.

                                 ;       Beep & flash error number.

 0290     DC 025E                TER1:   CALL    C,ALARM                 ; Alarm if proper time
 0293     CD 02AA                        CALL    TPXIT                   ; See if '*'
 0296     DB F0                          IN      A,(IP.PAD)              ;
 0298     FE 2F                          CP      X'2F'                   ; Check for '#'
 029A     CA 028D                        JP      Z,TER3                  ; It was.
 029D     3A 201C                        LD      A,(TICCNT+1)            ;
 02A0     1F                             RRA                             ; 'C' set if 1/2 second
 02A1     C3 0290                        JP      TER1                    ;
```

```
                                ;;;     TPABT - Abort tape load or dump.
                                ;
                                ;       Entered when loading or dumping, and the '*' key is pressed.

                                        TESTEQ  TPABT,X'02A4'

 02A4    AF                     TPABT:  XOR     A                       ; (A) := 0
 02A5    D3 F9                          OUT     (OP.TPC),A              ; Turn off the tape.
 02A7    C3 00D2                        JP      ERROR                   ; Sound the alarm & bail out.


                                ;;;     TPXIT - Check for user forced exit.
                                ;
                                ;       TPXIT checks for a '*' keypad entry.  If so, take
                                ;       the tape driver abnormal exit.
                                ;
                                ;       ENTRY:  None
                                ;       EXIT:   To (RET) if not '*'.
                                ;                (A) = Port status
                                ;               To (TPERRX) if '*' is depressed.
                                ;       USES:   A,F

                                        TESTEQ  TPXIT,X'02AA'

 02AA    DB F0                  TPXIT:  IN      A,(IP.PAD)              ; Read key pad
 02AC    FE 6F                          CP      X'6F'                   ; Is it '*'?
 02AE    DB F9                          IN      A,(IP.TPC)              ; Read tape status
 02B0    C0                             RET     NZ                      ; If not '*', return with status
 02B1    2A 2019                        LD      HL,(TPERRX)             ; (HL) := Error exit address
 02B4    E9                             JP      (HL)                    ; Go to (TPERRX)

                                        PAGE
```

```
  02B5

                                        ;;;     SRS - Scan record start.
                                        ;
                                        ;       SRS reads bytes from the tape until it recognizes the start
                                        ;       of a record.
                                        ;
                                        ;       This requires at least 10 SYNC characters & 1 STX character.
                                        ;
                                        ;       The CRC is then initialized.
                                        ;
                                        ;       ENTRY:  None
                                        ;       EXIT:   Tape positioned (and moving)
                                        ;               (CRCSUM) = 0
                                        ;               (DE) = Header bytes
                                        ;               (HA) = Record count
                                        ;       USES:   A,F,D,E,H,L

                                                TESTEQ  SRS,X'02B5'

  02B5                                  SRS:
  02B5      16 00                       SRS1:   LD      D,0                     ;
  02B7      62                                  LD      H,D                     ;
  02B8      6A                                  LD      L,D                     ; (HL) := 0

  02B9      CD 02D9                     SRS2:   CALL    RNB                     ; Read a byte
  02BC      14                                  INC     D                       ; Count 1 SYNC character
  02BD      FE 16                               CP      A,SYN                   ; Now see if it really was SYNC
  02BF      CA 02B9                             JP      Z,SRS2                  ; Yep, do it again

  02C2      FE 02                               CP      A,STX                   ; Was it a STX?
  02C4      C2 02B5                             JP      NZ,SRS1                 ; Nope, start over.

  02C7      3E 0A                               LD      A,10                    ;
  02C9      BA                                  CP      D                       ; Did we get at least 10 SYNCs?
  02CA      D2 02B5                             JP      NC,SRS1                 ; Nope ... start over.

  02CD      22 2017                             LD      (CRCSUM),HL             ; Clear CRC
  02D0      CD 02D5                             CALL    RNP                     ; Read leader
  02D3      54                                  LD      D,H                     ;
  02D4      5F                                  LD      E,A                     ; (DE) := Header
                                        ;       JP      RNP                     ; Read count

                                                PAGE
  02D5

                                ;;;     RNP - Read next pair.
                                ;
                                ;       RNP reads the next 2 bytes from the cassette tape.
                                ;
                                ;       ENTRY:  None
                                ;       EXIT:   (HA) = Byte pair
                                ;       USES:   A,F,H

                                        TESTEQ  RNP,X'02D5'

  02D5     CD 02D9              RNP:    CALL    RNB                     ; Read a byte
  02D8     67                           LD      H,A                     ; (H) := 1st byte
                                ;       JP      RNB                     ; Get 2nd byte


                                ;;;     RNB - Read next byte.
                                ;
                                ;       RNB reads a single byte from cassette tape.  The CRC
                                ;       is updated for the character.
                                ;
                                ;       ENTRY:  None
                                ;       EXIT:   (A) := Character
                                ;       USES:   A,F

                                        TESTEQ  RNB,X'02D9'

  02D9     3E 34                RNB:    LD      A,UCI.RO+UCI.ER+UCI.RE  ; Turn on reader for next byte
  02DB     D3 F9                        OUT     (OP.TPC),A              ;

  02DD     CD 02AA              RNB1:   CALL    TPXIT                   ; Check for '*' & read status
  02E0     E6 02                        AND     USR.RR                  ; Receiver ready?
  02E2     CA 02DD                      JP      Z,RNB1                  ; No ...
  02E5     DB F8                        IN      A,(IP.TPD)              ; Read a byte
                                ;       JP      CRC                     ; Do CRC

                                        PAGE
```

```
02E7

                            ;;;     CRC - Compute Cyclic Redundancy Check
                            ;
                            ;       CRC computes a CRC checksum from the polynomial:
                            ;
                            ;       (X + 1) * (X^15 + X + 1)
                            ;
                            ;       Since the checksum is a division remainder, a checksumed
                            ;       data sequence can be verified by running the data through
                            ;       CRC, and then running the previously obtained checksum
                            ;       through CRC.  The resultant checksum should be 0.
                            ;
                            ;       ENTRY:  (CRCSUM) = Current checksum
                            ;               (A)      = Byte
                            ;       EXIT:   (CRCSUM) Updated
                            ;               (A) Unchanged
                            ;       USES:   F

                                    TESTEQ  CRC,X'02E7'

02E7    C5                  CRC:    PUSH    BC                      ;
02E8    06 08                       LD      B,8                     ; (B) := Bit count
02EA    E5                          PUSH    HL                      ;
02EB    2A 2017                     LD      HL,(CRCSUM)             ; Get Current CRC

02EE    07                  CRC1:   RLCA                            ;
02EF    4F                          LD      C,A                     ; (C) = Bit
02F0    7D                          LD      A,L                     ;
02F1    87                          ADD     A,A                     ;
02F2    6F                          LD      L,A                     ;
02F3    7C                          LD      A,H                     ;
02F4    17                          RLA                             ;
02F5    67                          LD      H,A                     ;
02F6    17                          RLA                             ;
02F7    A9                          XOR     C                       ;
02F8    0F                          RRCA                            ;
02F9    D2 0304                     JP      NC,CRC2                 ; If not to XOR
02FC    7C                          LD      A,H                     ;
02FD    EE 80                       XOR     X'80'                   ;
02FF    67                          LD      H,A                     ;
0300    7D                          LD      A,L                     ;
0301    EE 05                       XOR     X'05'                   ;
0303    6F                          LD      L,A                     ;
0304    79                  CRC2:   LD      A,C                     ;
0305    05                          DEC     B                       ; Decrement bit count
0306    C2 02EE                     JP      NZ,CRC1                 ; Not done yet
0309    22 2017                     LD      (CRCSUM),HL             ; Set new CRC
030C    E1                          POP     HL                      ;
030D    C1                          POP     BC                      ;
030E    C9                          RET                             ;

                                    PAGE
```

```
 030F

                              ;;;      WNP - Write next pair.
                              ;
                              ;        WNP writes the next two bytes to cassette tape.
                              ;
                              ;        ENTRY:  (HL) = Bytes
                              ;        EXIT:   Bytes written
                              ;        USES:   A,F

                                       TESTEQ  WNP,X'030F'

 030F     7C                  WNP:     LD      A,H                  ; (A) := 1st byte
 0310     CD 0314                      CALL    WNB                  ; Write it.
 0313     7D                           LD      A,L                  ; (A) := 2nd byte
                              ;        JP      WNB                  ; Go write it


                              ;;;      WNB - Write next byte.
                              ;
                              ;        WNB writes the next byte to cassette tape.
                              ;
                              ;        ENTRY:  (A) = Byte
                              ;        EXIT:   None
                              ;        USES:   F

                                       TESTEQ  WNB,X'0314'

 0314     F5                  WNB:     PUSH    AF                   ; Save byte
 0315     CD 02AA             WNB1:    CALL    TPXIT                ; Check for '#' & get tape status
 0318     E6 01                        AND     USR.TR               ; Transmitter ready?
 031A     CA 0315                      JP      Z,WNB1               ; No, wait.
 031D     3E 11                        LD      A,UCI.ER+UCI.TE      ; Enable transmitter
 031F     D3 F9                        OUT     (OP.TPC),A           ;
 0321     F1                           POP     AF                   ; Get byte back
 0322     D3 F8                        OUT     (OP.TPD),A           ; Send it
 0324     C3 02E7                      JP      CRC                  ; Now go compute CRC & return.

                                       SUBTTL  'Miscellaneous Subroutines'
                                       PAGE
 0327

                                ;;;     LRA - Locate register address.
                                ;
                                ;       LRA locates a register on the monitor's stack.
                                ;
                                ;       ENTRY:  None
                                ;       EXIT:   (A) = Register index
                                ;               (HL) = Storage address
                                ;       USES:   A,F,D,E,H,L

 0327    3A 2005                LRA:    LD      A,(REGI)                ; Get register index
 032A    5F                     LRA.:   LD      E,A                     ;
 032B    16 00                          LD      D,0                     ; (DE) := Register index
 032D    2A 201D                        LD      HL,(REGPTR)             ; Get pointer to start of registers
 0330    19                             ADD     HL,DE                   ; (HL) := (REGPTR) + (REGI)
 0331    C9                             RET                             ;


                                ;;;     INWORD - Input 16 Bit Address from Keypad
                                ;
                                ;       INWORD reads a 16 bit address from the front panel keypad & stores
                                ;       it at ((HL)).
                                ;
                                ;       ENTRY:  (HL) := Address to store 16 bit value
                                ;       EXIT:   To (RET) if Ok
                                ;               To ERROR if bad digit entered.
                                ;       USES:   A,F,D,E,H,L

                                        TESTEQ  INWORD,X'0332'

 0332    CD 0336                INWORD: CALL    INBYTE                  ; Get 1st byte
 0335    2B                             DEC     HL                      ;


                                ;;;     INBYTE - Input 8 Bit Byte from Keypad
                                ;
                                ;       INBYTE reads an 8 bit value from the front panel keypad & stores
                                ;       it at ((HL)).
                                ;
                                ;       ENTRY:  (HL) = Address of byte to hold value.
                                ;       EXIT:   To (RET) if all Ok.
                                ;               To ERROR if error.
                                ;       USES:   A,F,D,E,H,L

                                        TESTEQ  INBYTE,X'0336'

 0336    EB                     INBYTE: EX      DE,HL                   ; Save (HL) in (DE)
 0337    21 2008                        LD      HL,.MFLAG               ; Point at .MFLAG
 033A    CB 5E                          BIT     3,(HL)                  ; Is hex mode set?
 033C    EB                             EX      DE,HL                   ; Put (HL) back
 033D    CA 043B                        JP      Z,IOB                   ; In octal mode, go to it.
```

```
                                ;       Input hexadecimal byte

 0340    06 02                  IHB:    LD      B,2                     ; 2 digits per hex byte
 0342    D4 03B0                IHB1:   CALL    NC,RCK                  ; Read a digit
 0345    A7                             AND     A                       ; Be sure 'C' is clear
 0346    ED 6F                          RLD                             ; Rotate digit into place
 0348    10 F8                          DJNZ    IHB1                    ; Decrement count & loop if not 0
 034A    CD 0459                        CALL    NOALTR                  ; Clear alter mode
 034D    3E 0F                          LD      A,30/2                  ; 30 ms
 034F    C3 0260                        JP      HORN                    ; *bip*


                                ;;;     DFD - Decode for front panel display
                                ;
                                ;       ENTRY:  (HL) = Address of LED refresh area
                                ;               (C)  = 'OR' pattern to force on bars or periods
                                ;               (A)  = Value
                                ;       EXIT:   (HL) = Next digit address
                                ;       USES:   A,B,C,H,L

                                        TESTED  DFD,X'0352'

 0352    E5                     DFD:    PUSH    HL                      ; Save (HL)
 0353    2E 08                          LD      L,LOW(.MFLAG)           ; (HL) := (.MFLAG)
 0355    CB 5E                          BIT     3,(HL)                  ; Check for HEX or S/OCTAL display
 0357    E1                             POP     HL                      ; Put (HL) back like we found it
                                        IF      ODSPLY
 0358    C2 040E                        JP      NZ,DHEX                 ; Go display HEX
                                        ELSE
                                        JP      Z,DHEX                  ;
                                        ENDC

 035B    D5                     DOCTAL: PUSH    DE                      ; Save (DE)
 035C    16 03                          LD      D,DODA/256              ; (D) := Page address of DODA
 035E    06 03                          LD      B,3                     ; (B) := Number of digits

 0360                           OCTAL1: REPT    3                       ; Shift it left 3 times
                                        RLA                             ;; Shift
                                        ENDM                            ;
 0360    17                 +           RLA                             ;; Shift
 0361    17                 +           RLA                             ;; Shift
 0362    17                 +           RLA                             ;; Shift
 0363    F5                             PUSH    AF                      ; Save value for next digit
 0364    E6 07                          AND     X'07'                   ; Remove unwanted bits
 0366    C6 EE                          ADD     A,LOW(DODA)             ; Index into DODA
 0368    5F                             LD      E,A                     ; (DE) := Address of pattern
 0369    1A                             LD      A,(DE)                  ; (A) := Pattern value
 036A    A9                             XOR     C                       ; Force pattern
 036B    E6 7F                          AND     X'7F'                   ;
 036D    A9                             XOR     C                       ;
 036E    18 6E                          JR      OCTAL2                  ; Go finish up.

                                        SUBTTL  'Update Front Panel Displays'
                                        PAGE
```

```
  0370

                                  ;;;     UFD - Update front panel display.
                                  ;
                                  ;       UFD is called by the clock interrupt processor when it is
                                  ;       time to update the display contents.  This is done every
                                  ;       32 interrupts, or about 32 times a second.
                                  ;
                                  ;       ENTRY:  (HL) = TICCNT
                                  ;       EXIT:   None
                                  ;       USES:   All

  0370    CB 48                   UFD:    BIT     1,B                     ; Test if display update wanted
  0372    C0                              RET     NZ                      ; If not ...
  0373    2E 06                           LD      L,LOW(DSPROT)           ; Rotate the little dots ...
  0375    CB 06                           RLC     (HL)                    ; Done ...
  0377    4E                              LD      C,(HL)                  ; Get the dots for update
  0378    23                              INC     HL                      ; Get to display mode
  0379    A7                              AND     A                       ; Make sure 'C' is clear
  037A    CB 4E                           BIT     1,(HL)                  ; See if in register mode
  037C    2A 2014                         LD      HL,(ABUSS)              ; If not set address mode
  037F    28 12                           JR      Z,UFD1                  ; If in memory mode don't set register
  0381    CD 0327                         CALL    LRA                     ; Get register address
  0384    E5                              PUSH    HL                      ; Save for later use
  0385    21 03FE                         LD      HL,DSPA                 ; Get register pattern from table by ...
  0388    19                              ADD     HL,DE                   ;   adding the register offset from LRA
  0389    7E                              LD      A,(HL)                  ;   and load into
  038A    23                              INC     HL                      ;   (H) and
  038B    66                              LD      H,(HL)                  ;   (L) for easier use
  038C    6F                              LD      L,A                     ;
  038D    E3                              EX      (SP),HL                 ; Set pattern on stack and get back
  038E    B4                              OR      H                       ;   register address with 'C' clear
  038F    7E                              LD      A,(HL)                  ; Load into (H,L) from memory
  0390    23                              INC     HL                      ;
  0391    66                              LD      H,(HL)                  ;
  0392    6F                              LD      L,A                     ;

                                          PAGE
```

```
 0393

 0393    F5                 UFD1:   PUSH    AF                      ; Save 'C' flag for mode
 0394    EB                         EX      DE,HL                   ; Swap to (DE) for display
 0395    21 200B                    LD      HL,ALEDS                ; Point to LED'S
 0398    7A                         LD      A,D                     ; Decode high byte
 0399    CD 0352                    CALL    DFD                     ;
 039C    7B                         LD      A,E                     ; Decode low byte
 039D    CD 0352                    CALL    DFD                     ;
 03A0    F1                         POP     AF                      ; Get back flags
 03A1    1A                         LD      A,(DE)                  ; If in memory mode get the data
 03A2    28 AE                      JR      Z,DFD                   ; Yup, in memory ... store and return
 03A4    CD 058E                    CALL    TALT                    ; Get prime marker for register sets
 03A7    E3                         EX      (SP),HL                 ; Get register pattern back
 03A8    22 2011                    LD      (DLEDS),HL              ; Store it
 03AB    E1                         POP     HL                      ; Get back LED address for prime mark
 03AC    23                         INC     HL                      ; Incrememnt past register pattern
 03AD    23                         INC     HL                      ;
 03AE    77                         LD      (HL),A                  ; Store it and ...

 03AF    C9                         RET                             ;   Return

                                    SUBTTL  'Read Console Keypad'
                                    PAGE
 03B0

                                    ;;;     RCK - Read front panel keypad.
                                    ;
                                    ;       RCK is called to read a keystroke from the front panel keypad.
                                    ;       RCK performs de-bouncing, and auto-repeat.  A *BIP* is sounded
                                    ;       when a value is accepted.
                                    ;
                                    ;       Keypad values:
                                    ;
                                    ;               1111 1110  -  0
                                    ;               1111 1100  -  1
                                    ;               1111 1010  -  2
                                    ;               1111 1000  -  3
                                    ;               1111 0110  -  4
                                    ;               1111 0100  -  5
                                    ;               1111 0010  -  6
                                    ;               1111 0000  -  7
                                    ;               1110 1111  -  8
                                    ;               1100 1111  -  9
                                    ;               1010 1111  -  A
                                    ;               1000 1111  -  B
                                    ;               0110 1111  -  C
                                    ;               0100 1111  -  D
                                    ;               0010 1111  -  E
                                    ;               0000 1111  -  F
                                    ;
                                    ;       ENTRY:  None
                                    ;       EXIT:   To caller when a key is hit.
                                    ;               (A) =    0 - '0'
                                    ;                        1 - '1'
                                    ;                        2 - '2'
                                    ;                        3 - '3'
                                    ;                        4 - '4'
                                    ;                        5 - '5'
                                    ;                        6 - '6'
                                    ;                        7 - '7'
                                    ;                        8 - '8'
                                    ;                        9 - '9'
                                    ;                       10 - 'A'
                                    ;                       11 - 'B'
                                    ;                       12 - 'C'
                                    ;                       13 - 'D'
                                    ;                       14 - 'E'
                                    ;                       15 - 'F'
                                    ;       USES:   A,F

                                            TESTEQ  RCK,X'03B0'

 03B0    E5                         RCK:    PUSH    HL                      ;
 03B1    C5                                 PUSH    BC                      ;
 03B2    0E 0A                              LD      C,200/20                ; Wait 200 ms
 03B4    21 2016                            LD      HL,RCKA                 ;

 03B7    DB F0                      RCK1:   IN      A,(IP.PAD)              ; Input pad value
```

```
  03B9    47                           LD      B,A               ; (B) := Value
  03BA    3E 0A                        LD      A,20/2            ; Wait 20 ms
  03BC    CD 002B                      CALL    DLY               ;
  03BF    78                           LD      A,B               ;
  03C0    BE                           CP      (HL)              ;
  03C1    20 03                        JR      NZ,RCK2           ; Have a change
  03C3    0D                           DEC     C                 ;
  03C4    20 F1                        JR      NZ,RCK1           ; Wait (C) cycles

                               ;       Have a key value.

  03C6    77                   RCK2:   LD      (HL),A            ; Update (RCKA)
  03C7    EE FE                        XOR     X'FE'             ;
  03C9    0F                           RRCA                      ;
  03CA    30 06                        JR      NC,RCK3           ; Hit '0' - '7'
                                       REPT    4                 ; Rotate right 4 times
                                       RRCA                      ;; Rotate
                                       ENDM                      ;
  03CC    0F                +          RRCA                      ;; Rotate
  03CD    0F                +          RRCA                      ;; Rotate
  03CE    0F                +          RRCA                      ;; Rotate
  03CF    0F                +          RRCA                      ;; Rotate
  03D0    30 E5                        JR      NC,RCK1           ; No hit at all ...
  03D2    47                   RCK3:   LD      B,A               ; (B) := Code
  03D3    3E 02                        LD      A,4/2             ; 4 ms
  03D5    CD 0260                      CALL    HORN              ; Make *BIP*
  03D8    78                           LD      A,B               ; Get value into (A)
  03D9    E6 0F                        AND     X'0F'             ; Make it 0 - 15
  03DB    C1                           POP     BC                ;
  03DC    E1                           POP     HL                ;
  03DD    C9                           RET                       ;

                                       SUBTTL  'Remainder of DOCTAL Routine'
                                       PAGE
```

```
 03DE

 03DE     77                  OCTAL2: LD      (HL),A              ; Set pattern into place
 03DF     23                          INC     HL                  ; Point to next pattern place
 03E0     CB 01                       RLC     C                   ; Rotate pattern
 03E2     F1                          POP     AF                  ; Get our value back
 03E3     05                          DEC     B                   ; One less digit to do.
 03E4     C2 0360                     JP      NZ,OCTAL1           ; If not 0 digits left
 03E7     D1                          POP     DE                  ; Restore (DE)
 03E8     C9                          RET                         ; All done.

                                      .LIST

                                      SUBTTL  'Front Panel Segment Patterns'
                                      PAGE
 03EE

                                  ;;;     Display segment coding:
                                  ;
                                  ;       Byte = 7654 3210
                                  ;
                                  ;                 ---1---
                                  ;                !       !
                                  ;                6       2
                                  ;                !       !
                                  ;                 ---0---
                                  ;                !       !
                                  ;                5       3
                                  ;                !       !
                                  ;                 ---4---  .7

                                  ;       Octal to 7 Segment Patterns

                                          TESTEQ  DODA,X'03EE'

 03EE     01                      DODA:   DEFB    00000001B            ; 0
 03EF     73                              DEFB    01110011B            ; 1
 03F0     48                              DEFB    01001000B            ; 2
 03F1     60                              DEFB    01100000B            ; 3
 03F2     32                              DEFB    00110010B            ; 4
 03F3     24                              DEFB    00100100B            ; 5
 03F4     04                              DEFB    00000100B            ; 6
 03F5     71                              DEFB    01110001B            ; 7
 03F6     00                              DEFB    00000000B            ; 8
 03F7     20                              DEFB    00100000B            ; 9
 03F8     10                              DEFB    00010000B            ; A
 03F9     06                              DEFB    00000110B            ; b
 03FA     0D                              DEFB    00001101B            ; C
 03FB     42                              DEFB    01000010B            ; d
 03FC     0C                              DEFB    00001100B            ; E
 03FD     1C                              DEFB    00011100B            ; F


                                  ;       Register to 7-Segment Patterns

 03FE     98A4                    DSPA:   DEFW    1001100010100100B    ; SP
 0400     9C90                            DEFW    1001110010010000B    ; AF
 0402     8D86                            DEFW    1000110110000110B    ; BC
 0404     8CC2                            DEFW    1000110011000010B    ; dE
 0406     8F92                            DEFW    1000111110010010B    ; HL
 0408     A2F3                            DEFW    1010001011110011B    ; IY
 040A     92F3                            DEFW    1001001011110011B    ; IX
 040C     CE98                            DEFW    1100111010011000B    ; Pc

                                          SUBTTL  'Hexadecimal Decode for Display'
                                          PAGE
 040E
                                     ;;;     DHEX - Called from DFD
                                     ;

 040E     D5                         DHEX:   PUSH    DE                      ; Save (DE)
 040F     16 03                              LD      D,DODA/256              ; (D) := Page address of DODA
 0411     06 02                              LD      B,2                     ; (B) := No. of digits
 0413     36 FF                              LD      (HL),X'FF'              ;   Blank out 1st digit of 3
 0415     23                                 INC     HL                      ;   Bump pointer, accordingly.

 0416                                DHEX1:  REPT    4                       ; Rotate left 4 times
                                             RLCA                            ;; Rotate
                                             ENDM                            ;
 0416     07                 +               RLCA                            ;; Rotate
 0417     07                 +               RLCA                            ;; Rotate
 0418     07                 +               RLCA                            ;; Rotate
 0419     07                 +               RLCA                            ;; Rotate
 041A     F5                                 PUSH    AF                      ; Save value on stack
 041B     E6 0F                              AND     X'0F'                   ; Remove high order NIBBLE
 041D     C6 EE                              ADD     A,LOW(DODA)             ; Index into DODA
 041F     5F                                 LD      E,A                     ; (DE) := Address of digit pattern
 0420     1A                                 LD      A,(DE)                  ; (A) := Pattern byte
 0421     A9                                 XOR     C                       ; Force pattern
 0422     E6 7F                              AND     X'7F'                   ;
 0424     A9                                 XOR     C                       ;
 0425     77                                 LD      (HL),A                  ; Set digit into memory
 0426     23                                 INC     HL                      ; Point to next digit place
 0427     CB 01                              RLC     C                       ; Rotate force pattern
 0429     F1                                 POP     AF                      ; Get the value back
 042A     10 EA                              DJNZ    DHEX1                   ; Decrement count & do more
 042C     D1                                 POP     DE                      ; Restore (DE)
 042D     C9                                 RET                             ; All done.

                                             SUBTTL  'Keypad Input Routines'
                                             PAGE
 042E

                                ;;;     CKHEX - Check for hexadecimal display mode.
                                ;
                                ;       CKHEX is called from the executive monitor loop.

 042E    F5                     CKHEX:  PUSH    AF                      ; Save value & status
 042F    3A 2008                        LD      A,(.MFLAG)              ; Get user options flag
 0432    E6 08                          AND     UO.HEX                  ; Hex mode set?
                                        IF      ODSPLY
 0434    28 03                          JR      Z,NHEX                  ; No, get out.
                                        ELSE
                                        JR      NZ,NHEX                 ;
                                        ENDC
 0436    F1                             POP     AF                      ; Get value back
 0437    37                             SCF                             ; Set 'C'
 0438    C9                             RET                             ; Done.

 0439    F1                     NHEX:   POP     AF                      ; Get value & status back
 043A    C9                             RET                             ; Return with PSW unaltered



                                ;;;     IOB - Input octal byte
                                ;

 043B    06 03                  IOB:    LD      B,3                     ; 3 digits per octal byte
 043D    D4 03B0                IOB1:   CALL    NC,RCK                  ; Read a digit

 0440    FE 08                          CP      8                       ; Was it a key from 0 - 7?
 0442    D2 00D2                        JP      NC,ERROR                ; No, refuse it!

 0445    5F                             LD      E,A                     ; Save value
 0446    7E                             LD      A,(HL)                  ;
 0447    07                             RLCA                            ; Shift over 3 bits
 0448    07                             RLCA                            ;
 0449    07                             RLCA                            ;
 044A    E6 F8                          AND     11111000B               ; Remove low digit of old value
 044C    B3                             OR      E                       ; 'OR' in this digit
 044D    77                             LD      (HL),A                  ; Replace value
 044E    10 ED                          DJNZ    IOB1                    ; Decrement count & repeat
 0450    3E 0F                          LD      A,30/2                  ; 30 ms beep
 0452    C3 0260                        JP      HORN

                                        PAGE
 0455

                                        ;;;     BADALT - Entered when trying to alter (SP)
                                        ;

 0455     21 00D2                       BADALT: LD      HL,ERROR                ; Set up 'return' address
 0458     E5                                    PUSH    HL                      ;
                                        ;       JP      NOALTR                  ; Clear alter mode & beep.


                                        ;;;     NOALTR - Clear alter mode
                                        ;

 0459     E5                            NOALTR: PUSH    HL                      ; Save (HL) on stack
 045A     21 2007                               LD      HL,DSPMOD               ; Mode is in DSPMOD
 045D     CB 86                                 RES     0,(HL)                  ; Clear alter bit
 045F     E1                                    POP     HL                      ; Restore (HL)
 0460     C9                                    RET                             ; Done (wasn't that easy?)

                                                SUBTTL  'Front Panel "FUNCTION" Processor'
                                                PAGE
 0461

                                  ;;;     FUNCT. - Function key processor
                                  ;
                                  ;       ENTRY:  NONE
                                  ;       EXIT:   If valid function,
                                  ;                To processor
                                  ;                (A) = Index
                                  ;               If error,
                                  ;                To ERROR.

 0461    3E 77                    FUNCT.: LD      A,01110111B             ; Rotate LEDS
 0463    32 2006                          LD      (DSPROT),A              ;
 0466    CD 03B0                          CALL    RCK                     ; Get function key
 0469    FE 04                            CP      NUMFUN                  ; Only valid functions allowed
 046B    D2 00D2                          JP      NC,ERROR                ; Wrong entry, so complain.
 046E    F5                               PUSH    AF                      ; Save index
 046F    3E FF                            LD      A,-1                    ; Stop rotating
 0471    32 2006                          LD      (DSPROT),A              ;
 0474    F1                               POP     AF                      ;
 0475    F5                               PUSH    AF                      ; Save index again
 0476    07                               RLCA                            ; (A) := index * 2
 0477    21 0485                          LD      HL,FNTBL                ; (HL) := Function table address
 047A    85                               ADD     A,L                     ; (HL) := (HL) + (0A)
 047B    6F                               LD      L,A                     ;
 047C    30 01                            JR      NC,FNC.                 ;
 047E    24                               INC     H                       ;
 047F    7E                       FNC.:   LD      A,(HL)                  ; Get low byte address of processor
 0480    23                               INC     HL                      ;
 0481    66                               LD      H,(HL)                  ; Get high byte
 0482    6F                               LD      L,A                     ; (HL) := Processor address
 0483    F1                               POP     AF                      ; (A) := index
 0484    E9                               JP      (HL)                    ; Go to processor through (HL)

 0485    04F5                     FNTBL:  DEFW    BOOTUP                  ; HDOS Bootstrap
 0487    04EC                             DEFW    DSP                     ; Hex / S. Octal Display
 0489    048D                             DEFW    XDISPL                  ; Indexed Display through Register
 048B    04AD                             DEFW    XCHGR                   ; Swap Primary & Alternate Registers

                                          NLIST
                                          .LIST

 0004                             NUMFUN  EQU     ($-FNTBL)/2             ; Number of defined functions

                                          SUBTTL  'Function Processing Routines'
                                          PAGE
 048D

                                   ;;;     Indexed Display of Memory through Register or Memory
                                   ;

 048D    3A 2007                   XDISPL: LD      A,(DSPMOD)              ; Get display mode byte
 0490    E6 02                             AND     DM.RR                   ; Register display?
 0492    28 0F                             JR      Z,XDMEM                 ; No, so index display through memory.

 0494    CD 0327                           CALL    LRA                     ; Get address of register on stack
 0497    5E                                LD      E,(HL)                  ; Get low half of address
 0498    23                                INC     HL                      ;
 0499    56                                LD      D,(HL)                  ; High half of address
 049A    ED 53 2014                        LD      (ABUSS),DE              ; ((ABUSS)) := Contents of register
 049E    AF                                XOR     A                       ; Get a zero in (A)
 049F    32 2007                           LD      (DSPMOD),A              ; Set memory display mode
 04A2    C9                                RET                             ; All done.

 04A3    2A 2014                   XDMEM:  LD      HL,(ABUSS)              ; Get memory address we're looking at
 04A6    CD 0596                           CALL    HLIHL                   ; Get value pointed to by (HL)
 04A9    22 2014                           LD      (ABUSS),HL              ; Set memory address
 04AC    C9                                RET                             ; Done ...


                                   ;;;     XCHGR - Swap alternate register set with
                                   ;               primary register set on stack.
                                   ;

 04AD    F3                        XCHGR:  DI                              ; No interrupts, please.
 04AE    2A 201D                           LD      HL,(REGPTR)             ; Point to registers on stack
 04B1    23                                INC     HL                      ; Move down to (AF)
 04B2    23                                INC     HL                      ;
 04B3    EB                                EX      DE,HL                   ; (DE) := Pointer to AF,BC,DE,HL
 04B4    21 FFF8                           LD      HL,-8                   ; Make room on bottom of stack
 04B7    39                                ADD     HL,SP                   ; (HL) := (SP) - 8
 04B8    F9                                LD      SP,HL                   ; Set new (SP)
 04B9    EB                                EX      DE,HL                   ; (DE) := To,  (HL) := From
 04BA    01 0008                           LD      BC,8                    ; (BC) := Length (4 registers * 2 bytes)
 04BD    ED B0                             LDIR                            ; Copy them
 04BF    F1                                POP     AF                      ; Get
 04C0    C1                                POP     BC                      ;   Registers
 04C1    D1                                POP     DE                      ;     From
 04C2    E1                                POP     HL                      ;       Stack
 04C3    CD 04DF                           CALL    XCHGR.                  ; Swap 'em
 04C6    E5                                PUSH    HL                      ; Put
 04C7    D5                                PUSH    DE                      ;   Alternate
 04C8    C5                                PUSH    BC                      ;     Registers
 04C9    F5                                PUSH    AF                      ;       On stack
 04CA    21 0008                           LD      HL,8                    ; Point to old (SP) loc'n
 04CD    39                                ADD     HL,SP                   ; (HL) := (SP) + 8
 04CE    F9                                LD      SP,HL                   ; Restore (SP) to original state
 04CF    2B                                DEC     HL                      ; Point to last of registers
 04D0    EB                                EX      DE,HL                   ; (DE) := Pointer to AF',BC',DE',HL'
 04D1    2A 201D                           LD      HL,(REGPTR)             ; Point to registers on stack
 04D4     01 0009                     LD      BC,9                 ; Move up to end of registers
 04D7     09                          ADD     HL,BC                ; (HL) := ((REGPTR)) + 9
 04D8     EB                          EX      DE,HL                ; (DE) := To, (HL) := From
 04D9     0E 08                       LD      C,8                  ; (BC) := Length of move (8 bytes)
 04DB     ED B8                       LDDR                         ; Set new registers onto stack
 04DD     FB                          EI                           ; Ok to interrupt now.
 04DE     C9                          RET                          ; We're done.


                              ;;;     XCHGR. - Actual Swap of Register Sets
                              ;
                              ;       EXIT:   (.MFLAG) Updated

 04DF     D9                  XCHGR.: EXX                          ; Swap BC,DE,HL
 04E0     08                          EX      AF,AF'               ; Swap PSW
 04E1     F5                          PUSH    AF                   ; Save PSW
 04E2     3A 2008                     LD      A,(.MFLAG)           ; (A) := .MFLAG
 04E5     EE 20                       XOR     UO.ALT               ; Toggle registers bit
 04E7     32 2008                     LD      (.MFLAG),A           ; Store new value
 04EA     F1                          POP     AF                   ; Restore AF
 04EB     C9                          RET                          ;


                              ;;;     DSP - Toggle Hex / Octal Display
                              ;

 04EC     3A 2008             DSP:    LD      A,(.MFLAG)           ; Get .MFLAG
 04EF     EE 08                       XOR     UO.HEX               ; Toggle bit
 04F1     32 2008                     LD      (.MFLAG),A           ; Replace with new values
 04F4     C9                          RET

                                      PAGE
 04F5

                                        ;;;     BOOTUP - Boot HDOS
                                        ;

 1F2D                                   ?BOOT   EQU     X'1F2D'                 ; Entry into boot code
 2280                                   ?STACK  EQU     X'2280'                 ; Stack address
 2048                                   ?D.CON  EQU     X'2048'                 ;
 1F5A                                   ?BOOTA  EQU     X'1F5A'                 ; Disk constants table
 0058                                   ?BOOTL  EQU     X'0058'                 ; Table length
 2048                                   ?D.CON  EQU     X'2048'                 ; Table storage area for HDOS
 20A0                                   ?D.RAM  EQU     X'20A0'                 ; System RAM work area
 001F                                   ?DRAML  EQU     X'001F'                 ; Work area length
 2131                                   ?IOUNI  EQU     X'2131'                 ; Disk unit number

 04F5    F3                             BOOTUP: DI                              ; No interrupts
 04F6    3E D0                                  LD      A,CB.SSI+CB.CLI+CB.SPK  ; Off monitor light
 04F8    32 2009                                LD      (CTLFLG),A              ;
 04FB    31 2280                                LD      SP,?STACK               ; Set new stack
 04FE    01 0058                                LD      BC,?BOOTL               ;
 0501    11 2048                                LD      DE,?D.CON               ;
 0504    21 1F5A                                LD      HL,?BOOTA               ;
 0507    ED B0                                  LDIR                            ; Copy Disk Constants

 0509    AF                                     XOR     A                       ; Get a zero in (A)

 050A    21 20A0                                LD      HL,?D.RAM               ; RAM work area
 050D    77                                     LD      (HL),A                  ;
 050E    11 20A1                                LD      DE,?D.RAM+1             ;
 0511    01 001E                                LD      BC,?DRAML-1             ;
 0514    ED B0                                  LDIR                            ; Zero it.

                                                NLIST
                                                .LIST

 0516    21 2008                                LD      HL,.MFLAG               ; Point to .MFLAG
 0519    CB C6                                  SET     0,(HL)                  ; Request clock interrupts
 051B    32 2131                                LD      (?IOUNI),A              ; Set proper boot device
 051E    D3 7F                                  OUT     (X'7F'),A               ; Reset Drives
 0520    3C                                     INC     A                       ; (A) := 1

 0521    C3 1F2D                                JP      ?BOOT                   ; Go to boot code.

                                                .LIST

                                                SUBTTL  'Real Time Clock Processor'
                                                PAGE
 0524

                                        ;;;     SYSCLK - Update System clock
                                        ;
                                        ;       This routine is called from SIM to keep up the system
                                        ;       clock. Exit will have (HL) := .MFLAG
                                        ;
                                        ;       The clock is stored at a location pointed to by CLKPTR,
                                        ;       in the following format:
                                        ;
                                        ;               (CLKPTR-4) - Hours   (0 - 23)
                                        ;               (CLKPTR-3) - Minutes (0 - 59)
                                        ;               (CLKPTR-2) - Seconds (0 - 59)
                                        ;               (CLKPTR-1) - Msec/2
                                        ;               (CLKPTR+1) - Clock timing constant
                                        ;                               This value contains the number of tics
                                        ;                               which will occur in 1 second.  It may
                                        ;                               be altered at the descretion of the
                                        ;                               user to compensate for disk I/O, etc.
                                        ;
                                        ;       If RAM is not present at the initial location set up by
                                        ;       this monitor (X'0804'),  the user may move it by merely
                                        ;       changing the address in CLKPTR to the desired location.
                                        ;
                                        ;       NOTE:  When the clock reaches midnight (24:00:00.000),
                                        ;               the date as maintained by HDOS is incremented.
                                        ;               No checks are made for the date before OR after
                                        ;               it is incremented, thus strange things may appear
                                        ;               at the end of a month.

 0524     E5                            SYSCLK: PUSH    HL                      ; Save .MFLAG
 0525     CD 052C                               CALL    SYSCK1                  ; Do the routine
 0528     E1                                    POP     HL                      ; Restore it all
 0529     16 00                                 LD      D,0                     ; Restore (D) to 0
 052B     C9                                    RET                             ; To caller

 052C     2A 203B                       SYSCK1: LD      HL,(CLKPTR)             ; Get clock pointer
 052F     2B                                    DEC     HL                      ; Point to MSEC
 0530     E5                                    PUSH    HL                      ; Hold that for a nano ...
 0531     CD 0596                               CALL    HLIHL                   ; Get MSEC value
 0534     23                                    INC     HL                      ; Bump it  (really 2 ms)
 0535     E3                                    EX      (SP),HL                 ; MSEC on stack, Addr. in (HL)
 0536     D1                                    POP     DE                      ; (DE) := MSEC
 0537     73                                    LD      (HL),E                  ; Set value into memory
 0538     23                                    INC     HL                      ;
 0539     72                                    LD      (HL),D                  ;
 053A     E5                                    PUSH    HL                      ; Save clock addr.
 053B     23                                    INC     HL                      ; Pointing to timing constant
 053C     CD 0596                               CALL    HLIHL                   ; Load it  (HL) := # tics / second
 053F     19                                    ADD     HL,DE                   ;
 0540     E1                                    POP     HL                      ; Get that address back
 0541     D0                                    RET     NC                      ; 'C' set if second is up.
 0542     AF                                    XOR     A                       ; Get us a zero
 0543     77                                    LD      (HL),A                  ; Zero the MSEC counter
 0544     2B                                    DEC     HL                      ;
```

```
  0545    77                         LD      (HL),A              ;
  0546    2B                         DEC     HL                  ; Now pointing at seconds
  0547    3E 3C                      LD      A,60                ; (A) := reference 60
  0549    01 0200                    LD      BC,X'200'           ; (B) := 2 , (C) := 0
  054C    34                 CLKTST: INC     (HL)                ; Bump the counter
  054D    BE                         CP      (HL)                ; See if new value
  054E    C0                         RET     NZ                  ; Nope ...
  054F    71                         LD      (HL),C              ; Zero the count
  0550    2B                         DEC     HL                  ; Drop to next value
  0551    10 F9                      DJNZ    CLKTST              ; Do the minutes
  0553    34                         INC     (HL)                ; Bump the hours now
  0554    3E 18                      LD      A,24                ; See if too high
  0556    BE                         CP      (HL)                ; Is it?
  0557    C0                         RET     NZ                  ; Nope ...
  0558    71                         LD      (HL),C              ; Zero it
  0559    21 20C8                    LD      HL,X'20C8'          ; HDOS's coded date
  055C    34                         INC     (HL)                ; Now its tomorrow ...
  055D    2E C0                      LD      L,X'C0'             ; Do the ASCII date
  055F    34                         INC     (HL)                ; Bump it up
  0560    7E                         LD      A,(HL)              ; Now get it
  0561    FE 3A                      CP      '9'+1               ; Is it too big?
  0563    D8                         RET     C                   ; Nope ...
  0564    36 30                      LD      (HL),'0'            ; Set it to 0
  0566    2B                         DEC     HL                  ; Point to tens
  0567    34                         INC     (HL)                ; Now its right ...
  0568    C9                         RET                         ; Done ...

  0569    E1                 CLKFIN: POP     HL                  ; Kill return address
  056A    01 2008                    LD      BC,.MFLAG           ; Set it up for CUI1
  056D    3A 2009                    LD      A,(CTLFLG)          ; Get hardware control flag
  0570    D3 F0                      OUT     (OP.CTL),A          ; Send it ...
                             ;       JP      CUI1                ; See if user wants clock.

                                     SUBTTL  'Miscellaneous Monitor Subroutines'
                                     PAGE
```

```
 0572

                                  ;;;     CUI1 - Check for user interrupt processing.
                                  ;
                                  ;       CUI1 is called to see if the user wants to process the
                                  ;       clock interrupt.  If so, his routine is called.

 2008                             .       DEFL    .MFLAG                  ; Make reference
 0572     0A                      CUI1:   LD      A,(BC)                  ; (A) := .MFLAG
 0573     0F                              RRCA                            ; Check bit 0 for user processing
 0574     DC 201F                         CALL    C,UIVEC                 ; If specified, call user's routine
 0577     C3 007A                         JP      INTXIT                  ; Return


                                  ;;;     STPRTN - Single Step Return
                                  ;
                                  ;       Return to here from single step interrupt.

 057A     F6 10                   STPRTN: OR      CB.SSI                  ; Set single step inhibit
 057C     D3 F0                           OUT     (OP.CTL),A              ; Disable the interrupt
 2009                             .       DEFL    CTLFLG                  ;
 057E     12                              LD      (DE),A                  ; Set the new flag values
 057F     E6 20                           AND     CB.MTL                  ; Are we in monitor mode?
 0581     C2 00E4                         JP      NZ,MTR                  ; Yep, get there.
 0584     C3 2022                         JP      UIVEC+3                 ; ... else go see user.


                                  ;;;     SSTI
                                  ;

 0587     32 2009                 SSTI:   LD      (CTLFLG),A              ; Set new flag values
 058A     E1                              POP     HL                      ;
 058B     C3 007A                         JP      INTXIT                  ;


                                  ;;;     TALT - Test for alternate registers.
                                  ;

 058E     3A 2008                 TALT:   LD      A,(.MFLAG)              ; Get .MFLAG
 0591     E6 20                           AND     UO.ALT                  ; Alternate registers?
 0593     07                              RLCA                            ; Rotate bit into place for display
 0594     2F                              CPL                             ; Make it 'on' and the rest 'off'
 0595     C9                              RET                             ;


                                  ;;;     HLIHL - Load (HL) indirect through (HL).
                                  ;

 0596     7E                      HLIHL:  LD      A,(HL)                  ; Get low byte
 0597     23                              INC     HL                      ;
 0598     66                              LD      H,(HL)                  ; Get high byte
```

```
0599    6F                      LD      L,A                     ; Set low byte in place.
059A    C9                      RET                             ;


                        ;;;     TSTREG - TSTREG tests for (PC) or (IY) display and
                        ;       returns the proper value.  This is done to keep
                        ;       the (PC) on key '6' for compatibility with PAM/8.

059B    FE 05           TSTREG: CP      5                       ; Was it (PC)?
059D    28 09                   JR      Z,PCREG                 ; Yep.
059F    FE 07                   CP      7                       ; How about (IY)?
05A1    20 02                   JR      NZ,TESTR1               ; Nope, go around.
05A3    3E 05                   LD      A,5                     ; Make it right for (IY)
05A5    07              TESTR1: RLCA                            ; (A) := (A) * 2
2005                    .       DEFL    REGI                    ;
05A6    12                      LD      (DE),A                  ; Set register index
05A7    C9                      RET                             ; Return
05A8    3E 07           PCREG:  LD      A,7                     ; Fix for (PC)
05AA    18 F9                   JR      TESTR1                  ; Go finish up ...


                        ;;;     SIM - Set Interrupt Mode.
                        ;
                        ;       SIM is called every clock interrupt to insure that the
                        ;       interrupt mode currently set agrees with that which is
                        ;       indicated in .MFLAG.

05AC    3A 203A         SIM:    LD      A,(IR)                  ; Get value for (I)
05AF    ED 47                   LD      I,A                     ; Set it.
05B1    21 2008                 LD      HL,.MFLAG               ; Get the flag
05B4    CB 56                   BIT     2,(HL)                  ; See if clock is wanted.
05B6    CC 0524                 CALL    Z,SYSCLK                ; Yes, do it.
05B9    CB 76                   BIT     6,(HL)                  ; See if no clock wanted.
05BB    20 AC                   JR      NZ,CLKFIN               ; Say all done.
05BD    7E                      LD      A,(HL)                  ;
05BE    E6 10                   AND     UO.IM1                  ; Check for interrupt mode 1 set
05C0    F6 46                   OR      LOW(Z.IMO)              ; Make either IMO or IM1 opcode.
05C2    47                      LD      B,A                     ; Set into (B)
05C3    0E ED                   LD      C,HIGH(Z.IMO)           ; Set high byte of instruction into (C)
05C5    ED 43 2002              LD      (IOWRK),BC              ; Put the whole thing into memory and
05C9    C3 2002                 JP      IOWRK                   ;   go execute it ...


                        ;;;     INIT2 - Initialize monitor stack, Cassette USART,
                        ;               NMI and IM1 vectors, & Real Time Clock.
                        ;

0804                    FPMCLK  EQU     X'0804'                 ; Address of default timer constant

05CC    2B              INIT2:  DEC     HL                      ; High Memory address
05CD    F9                      LD      SP,HL                   ; Set stack pointer
05CE    21 0804                 LD      HL,FPMCLK               ; Put the clock in 1st 5 bytes
05D1    22 203B                 LD      (CLKPTR),HL             ;   of available address space.
```

```
05D4    21 FE0C                 LD      HL,-500                 ; 500 tics in a second  (2ms / tic)
05D7    22 0605                 LD      (FPMCLK+1),HL           ; Set it for our clock
05DA    21 04F5                 LD      HL,[illegible]          ; Set initial (PC) address
05DD    E5                      PUSH    HL                      ;
05DE    21 [illegible]          LD      HL,ERROR                ; Set our 'return' address
05E1    E5                      PUSH    HL                      ;

05E2    3E 4E                   LD      A,UMI.1B+UMI.L8+UMI.16X ; Set 8 bits, no parity, 1 stop, 16X
05E4    D3 F9                   OUT     (UP.TPC),A              ; Done.

05E6    21 45ED                 LD      HL,Z.RETN               ; 'RETN' instruction.
05E9    22 2037                 LD      (UIVEC+24),HL           ; Set for NMI.
05EC    3E C9                   LD      A,MI.RET                ; 'RET' instruction
05EE    32 2034                 LD      (UIVEC+21),A            ; Set for IM1.

                        ;       Initialize the clock to a 00:00:00,000

05F1    0E 05                   LD      C,5                     ; Do 5 bytes
05F3    2A 203B                 LD      HL,(CLKPTR)             ; at clock pointer
05F6    54                      LD      D,H                     ;
05F7    5D                      LD      E,L                     ;
05F8    1B                      DEC     DE                      ; Gonna zero it backwards ...
05F9    70                      LD      (HL),B                  ; Start it off  ((B) has a zero)
05FA    ED B8                   LDDR                            ; Ripple it through
05FC    C3 005A                 JP      SAVALL                  ;



                        ;;;     SAVALX - Called to allow space for NMI entry.
                        ;

05FF    21 000E         SAVALX: LD      HL,14                   ;
0602    39                      ADD     HL,SP                   ; (HL) := Address of user's (SP)
0603    E5                      PUSH    HL                      ; Put it on stack as 'register'
0604    11 2009                 LD      DE,CTLFLG               ;
0607    C3 006A                 JP      SAVALR                  ; Now finish up.



                        ;;;     IMOD1 - Check for IM1 on a level 7 interrupt.
                        ;

060A    F5              IMOD1:  PUSH    AF                      ; Save (AF)
060B    3A 2008                 LD      A,(.MFLAG)              ; Get .MFLAG
060E    E6 10                   AND     UO.IM1                  ; See if IM1 is set
0610    20 04                   JR      NZ,IMOD1.               ; IM1, so do it.

0612    F1                      POP     AF                      ; Get PSW back
0613    C3 2031                 JP      UIVEC+18                ; Go to user's level 7 routine

0616    F1              IMOD1.: POP     AF                      ; Get PSW back
0617    CD 2034                 CALL    UIVEC+21                ; Call user's IM1 routine.
061A    F3                      DI                              ; No interrupts for the clock
061B    CF                      RST     08                      ; Now do it.
061C    C9                      RET                             ; All done.
```

```
                                        SUBTTL  'Constants'

                                ;;;     I/O routines to be copied into and used in RAM.
                                ;

 061D    C9                     PRSROM: DEFB    MI.RET                  ; 'RET' opcode
 061E    0E                             DEFB    14                      ; Register index
 061F    00                             DEFB    0                       ; DSPROT
 0620    02                             DEFB    2                       ; DSPMOD
 0621    00                             DEFB    0                       ; .MFLAG
 0622    00                             DEFB    0                       ; CTLFLG
 0623    01                             DEFB    1                       ; REFIND

                                        .LIST

                                        TESTEQ  $,X'700'

                                        .DEPHASE

 0A6D'                          ?END    EQU     $

                                        TESTEQ  (?END-?START),X'0800'   ; Must be EXACTLY 2K  (2048 bytes)

                                        SUBTTL  'RAM Cell Definitions'
                                        PAGE
 0A6D'

                                  ;;;     The following locations are used by the front panel monitor.
                                  ;

                                          .PHASE  X'2000'

 2000                             START:  DEFS    2                       ; Dump starting address
 2002                             IOWRK:  DEFS    2                       ; Input/Output work area
 2004                             PRSRAM  EQU     $                       ; The following cells initialized from ROM
 2004                                     DEFS    1                       ; (RET) opcode

 2005                             REGI:   DEFS    1                       ; Index of register under display
 2006                             DSPROT: DEFS    1                       ; Period flag byte
 2007                             DSPMOD: DEFS    1                       ; Display mode

 2008                             .MFLAG: DEFS    1                       ; User options flag
                                  ;                                         See UO.XXX bits described at front.

 2009                             CTLFLG: DEFS    1                       ; Front panel control bits.
 200A                             REFIND: DEFS    1                       ; Refresh index (0 to 7)
 0007                             PRSL    EQU     $-PRSRAM                ; End of ROM initialized area

 200B                             FPLEDS  EQU     $                       ; Front panel LED display area
 200B                             ALEDS:  DEFS    6                       ; Six LEDs for address
 2011                             DLEDS:  DEFS    3                       ; Three LEDs for data

 2014                             ABUSS:  DEFS    2                       ; Address buss (memory under display)
 2016                             RCKA:   DEFS    1                       ; RCK save area
 2017                             CRCSUM: DEFS    2                       ; CRC checksum
 2019                             TPERRX: DEFS    2                       ; Tape error exit address
 201B                             TICCNT: DEFS    2                       ; Clock 2 ms counter

 201D                             REGPTR: DEFS    2                       ; Register contents pointer

 201F                             UIVEC   EQU     $                       ; User interrupt vectors
 201F                                     DEFS    3                       ; Jump to clock processor
 2022                                     DEFS    3                       ; Jump to single step processor
 2025                                     DEFS    3                       ; Jump to I/O 3.  (HDOS console)
 2028                                     DEFS    3                       ; Jump to I/O 4.
 202B                                     DEFS    3                       ; Jump to I/O 5.
 202E                                     DEFS    3                       ; Jump to I/O 6.
 2031                                     DEFS    3                       ; Jump to I/O 7.  (HDOS SCALL routine)
 2034                                     DEFS    3                       ; Jump to Interrupt Mode 1 routine
 2037                                     DEFS    3                       ; Jump to NMI routine

 203A                             IR:     DEFS    1                       ; Storage for (I) register

 203B                             CLKPTR: DEFS    2                       ; Pointer to clock area

 203D                                     DEFS    3                       ; Unused bytes

                                          TESTEQ  $,X'2040'               ; Should be at end of scratch pad.

                                          .DEPHASE
 0AAD'   00                             DEFB    0
                                        END     CPYROM
```

```
Macros:
FILL      NLIST     SCALL     TESTEQ

Symbols:
$UDD      196F      .         2005      ..        000B      .2MHZ     0052
.4MHZ     0038      .CLRCO    0007      .CONSL    0006      .CTLC     0021
.EXIT     0000      .LOADO    0008      .MFLAG    2008      .PRINT    0003
.SCIN     0001      ?ABORT    0092'     ?AD1      0123'     ?AD2      0127'
?BOOT     1F2D      ?BOOTA    1F5A      ?BOOTL    0058      ?CR       0030'
?CR2      005A'     ?D.CON    2048      ?D.RAM    20A0      ?DRAML    001F
?END      0A6D'     ?EXIT     0087'     ?IOUNI    2131      ?ROMAD    8000
?STACK    2280      ?START    026D'     ?V.ERR    008C'     ?VLOOP    007A'
@@WORK    8101      A.STX     0002      A.SYN     0016      ABORT     0166
ABTMSG    0259'     ABUSS     2014      ALARM     025E      ALEDS     200B
ALL64K    000F      BADALT    0455      BANK0     0001      BANK1     0002
BANK2     0004      BANK3     0008      BCDV      0012      BDROM     0000
BOARD0    0000      BOARD1    0010      BOARD2    0020      BOARD3    0030
BOARD4    0040      BOARD5    0050      BOARD6    0060      BOARD7    0070
BOOTUP    04F5      BPROM     0000      CB.CLI    0040      CB.MTL    0020
CB.SPK    0080      CB.SSI    0010      CKHEX     042E      CLK2      009A
CLK3      009D      CLK4      00C6      CLKFIN    0569      CLKPTR    203B
CLKTST    054C      CLOCK     0081      CPYROM    0000I'    CRC       02E7
CRC1      02EE'     CRC2      0304      CRCSUM    2017      CTC       027A
CTLA      0001      CTLB      0002      CTLC      0003      CTLD      0004
CTLFLG    2009      CUI1      0572      DCROM     0000      DFD       0352
DHEX      040E      DHEX1     0416      DLEDS     2011      DLY       002B
DM.MR     0000      DM.MW     0001      DM.RR     0002      DM.RW     0003
DOCTAL    035B      DODA      03EE      DSP       04EC      DSPA      03FE
DSPMOD    2007      DSPROT    2006      DUMP      0202      ENL       008A
ERROR     00D2      ESC       001B      EXIT      0067      FALSE     0000
FNC.      047F      FNTBL     0485      FPLEDS    200B      FPMADR    0000
FPMCLK    0804      FPMLEN    0700      FUNCT     01AE      FUNCT.    0461
GO        0192      GO.       0033      H17ADR    1800      H17LEN    0800
HLIHL     0596      HORN      0260      HRN0      0263      HRN2      0270
IHB       0340      IHB1      0342      IMOD1     060A      IMOD1.    0616
IN        017D      INBYTE    0336      INIT      003B      INIT0     0000
INIT1     0046      INIT1A    004F      INIT2     05CC      INT1      0008
INT2      0010      INT3      0018      INT4      0020      INT5      0028
INT6      0030      INT7      0038      INTXIT    007A      INWORD    0332
IOB       043B      IOB1      043D      IOWRK     2002      IP.PAD    00F0
IP.TPC    00F9      IP.TPD    00F8      IR        203A      IXIT1     0052
LAST      0167      LOA0      01BA      LOA1      01E2      LOAD      01B7
LOOP1     0025      LOOP2     0031      LRA       0327      LRA.      032A
MASTER    0000      MEMM      0173      MI.ANI    00E6      MI.HLT    0076
MI.IN     00DB      MI.INC    003C      MI.LDA    003A      MI.LXI    0011
MI.OUT    00D3      MI.RET    00C9      MTR       00E4      MTR1      00E5
MTR4      0106      MTR5      0129      MTR6      0137      MTRA      011D
NEXT      015A      NHEX      0439      NL        000A      NMI       0066
NOALTR    0459      NUMFUN    0004      OCTAL1    0360      OCTAL2    03DE
ODSPLY    FFFF      OKMSG     01EA'     OP.CTL    00F0      OP.DIG    00F0
OP.RAM    003F      OP.SEG    00F1      OP.TPC    00F9      OP.TPD    00F8
OUT       0180      PCREG     05A8      PLWAIT    015A'     PRSL      0007
PRSRAM    2004      PRSROM    061D      R$W       0156      RCK       03B0
RCK1      03B7      RCK2      03C6      RCK3      03D2      RCKA      2016
REFIND    200A      REGI      2005      REGM      0144      REGPTR    201D
RMEM      01B1      RNB       02D9      RNB1      02DD      RNP       02D5
ROMDIS    0080      RT.BP     0002      RT.CT     0003      RT.MI     0001
RTM0      0000      SAE       0133      SAV1      0077      SAVALL    005A
SAVALR    006A      SAVALX    05FF      SIGNUN    0097'     SIM       05AC
```

```
SST1    0587    SSTEP   0195    STACK   0900    START   2000
STPRTN  057A    SUBV    0002    SYSCK1  052C    SYSCLK  0524
SYSINI  0000    TAB     0009    TALT    058E    TER1    0290
TER3    028D    TESIR1  05A5    TFT     025B    TICCNT  201B
TPABT   02A4    TPERR   0285    TPERRX  2019    TPXIT   02AA
TRUE    FFFF    TSTREG  059B    UCI.ER  0010    UCI.IE  0002
UCI.IR  0040    UCI.RE  0004    UCI.RO  0020    UCI.TE  0001
UFD     0370    UFD1    0393    UIVEC   201F    UMI.16  0002
UMI.1B  0040    UMI.1X  0001    UMI.2B  00C0    UMI.64  0003
UMI.HB  0080    UMI.L5  0000    UMI.L6  0004    UMI.L7  0008
UMI.L8  000C    UMI.PA  0010    UMI.PE  0020    UO.ALT  0020
UO.CLK  0001    UO.DDU  0002    UO.HEX  0008    UO.HLT  0080
UO.IM1  0010    UO.NFR  0040    UO.RCK  0004    USR.FE  0020
USR.OE  0010    USR.PE  0008    USR.RR  0002    USR.TE  0004
USR.TR  0001    V.ERR   022D'   VER     0001    VERIFY  0179'
VWAIT   01CE'   WME1    020A    WME2    0244    WMEM    01FC
WNB     0314    WNB1    0315    WNP     030F    XCHGR   04AD
XCHGR.  04DF    XDISPL  048D    XDMEM   04A3    Z.IMO   ED46
Z.RETN  45ED


No  Fatal error(s)
```