ZENITH | data systems

MONITOR
MTR-90

595-2696-02

ZENITH DATA SYSTEMS
SAINT JOSEPH, MICHIGAN 49085

TABLE OF CONTENTS

INTRODUCTION

This Manual describes the functions and operation of the Z-89/90 Monitor Program, MTR-90, which is contained in a read-only memory (ROM) on the Z-89/90 CPU board. Some of the major features of MTR-90 include:

Disk system bootstrap routines.

Memory contents display and alteration.

Program execution control.

Variable radix settings for display addresses and conversion.

Input/output to specified ports.

Memory diagnostic routine.

In addition, by means of a flag byte maintained in read/write memory, MTR-90 can be instructed to bypass some or all of its normal functions. In this manner, a sophisticated user can augment or replace these functions.

THEORY OF OPERATION

This section supplements information in the "Operations" and "Circuit Description" sections of your Z-89/Z-90 Operations Manual. In order to use all of the features of MTR-90, it is necessary to understand the 8080 and Z-80 opcodes and the circuitry of your Z-89/90. This section details the operation of MTR-90. For a listing of the MTR-90 program, see Appendix A.

Power Up and Reset

MTR-90 initializes the Z-89/90 whenever you apply power to or reset the computer. To power up, use the switch on the right rear of the Z-89/90. To reset, simultaneously press the RESET key and the right-hand shift key on the keyboard. When reset, MTR-90 sounds the electronic "bell" and displays the "H:" prompt on the terminal screen.

During the initialization procedure, MTR-90 determines the high limit of continuous RAM. Once MTR-90 has established this high memory limit, the Z-80 stack pointer is set to the value of the upper memory limit. Then MTR-90 enters a loop and awaits a command.

General Operations

When you power up or reset your Z-89/90, MTR-90 responds by clearing the screen and displaying "H:". This "H:" prompt informs you that MTR-90 is ready to respond to commands. When you enter a character, MTR-90 will either accept it, completing a command word, or beep, which signifies an invalid command word or an inability to boot.

The DELETE key kills a partially entered line and causes MTR-90 to return the "H:" prompt. This is useful for correcting typing errors.

Clock Interrupts

The clock interrupt is a crucial element in the operation of the Z-89/90. It is a level one interrupt, and is generated on the Z-89/90 CPU board every two milliseconds. MTR-90 maintains a tick counter called "TICCNT", which counts one tick every 2 milliseconds. Refer to the listing in Appendix A for the location of TICCNT.

Note that MTR-90 uses interrupts, so you should not disable interrupts using the DI instruction for other than very short periods of time. MTR-90 also requires a stack pointer at the top of memory with at least 80 bytes of stack area.

NOTE: In this manual, the symbol △ means press the space bar and ⓒⓡ means press the RETURN key.

MTR-90 COMMANDS

The following section summarizes valid commands to MTR-90. Each command is listed in alphabetical order along with a brief explanation and examples. You need only enter the first letter of these commands — MTR-90 will respond with what is enclosed in parentheses. In most cases, you will need to press RETURN before MTR-90 will respond. Where a command requires numeric input, we have used the hexadecimal, octal, and split octal number bases.

B(oot)

Typing B(oot) and pressing RETURN initiates boot from drive 0 of the disk drives which have been configured for primary boot using SW501 switch 4 (see Appendix D). This command may optionally be followed by a unit number which specifies a drive other than drive zero. The unit number may be optionally followed by a command string which begins with a colon. The command string is currently used only by those Heath/Zenith operating systems which support the H/Z-67 Winchester disk subsystem. For more detailed information about how MTR-90 accomplishes bootstrap, see Appendix A.

If the boot fails, the computer will display a question mark, beep, and display the H: again. The possible causes for a boot failure include:

1. The boot device is not activated within 15 seconds.
2. The DELETE key is pressed during boot.
3. Switch SW501 is not set properly.
4. A disk error occurs.

The DELETE Key cancels the B(oot) command and repeats the H: prompt, unless boot has already begun, in which case the system displays the message "?Boot Error".

EXAMPLE 1: Boot from unit zero of the primary boot drives.

```
H: B(oot) ↵
```

EXAMPLE 2: Boot from unit 2 of the primary boot drives.

```
H: B(oot)2 ↵
```

EXAMPLE 3: Boot from primary boot Z-67 unit 2, passing the command line "HDOS;1" to the secondary Z-67 boot routine.

```
H: B(oot)2:HDOS;1 ↵
```

B(oot) S(D)

The B(oot) S(D) command initiates boot from unit zero of the drives which have been configured using SW501 switch 4 as secondary boot drives (see Appendix D). This command may optionally be followed by a unit number which specifies a drive other than drive zero. The unit number may be optionally followed by a command string which begins with a colon. The command string is currently used only by those Heath/Zenith operating systems which support the H/Z-67 Winchester disk subsystem.

If the boot fails, the computer will display a question mark, beep, and display the H: again. The possible causes for a boot failure are:

1. The boot device is not activated within 15 seconds.
2. The DELETE key is pressed during boot.
3. Switch SW501 is not set properly.
4. A disk error occurs.

The DELETE key cancels the B(oot) S(D) command and repeats the H: prompt, unless boot has already begun, in which case the system prints the message "?Boot Error".

EXAMPLE 1: Boot from secondary boot drives, unit zero.

```
H: B(oot) S(D) ↵
```

EXAMPLE 2: Boot from unit 2 of the secondary boot drives.

```
H: B(oot) S(D)2 ↵
```

EXAMPLE 3: Boot from secondary boot Z-67 unit 2, passing the command line "HDOS;1" to the Z-67 boot routine.

```
H: B(oot) S(D)2:HDOS;1 CR
```

C(onvert)

The C(onvert) command converts a sixteen-bit number specified in the opposite radix to the current radix. To set the current radix, see the **R(adix)** command on Page 9.

EXAMPLE: Convert FFFF hex to split octal, where octal is the current radix.

```
   H: (C(onvert)FFFF  CR
377377
   H:
```

G(o)

The G(o) command initiates a user program, beginning at the address specified in the current radix as an argument to the G(o) command. If no argument is supplied with the G(o) command, then execution begins at the address contained in the program counter.

EXAMPLE: Go to address 40200 octal.

```
H: G(o)40200 CR
```

I(n)

I(n) inputs a number from the port specified as an argument to the I(n) command. The port number must be specified in the current radix.

EXAMPLE: Input data from port 177 octal, where octal is the current radix.

```
   H: I(n)177  CR
370
   H:
```

O(ut)

The O(ut) command outputs the specified data to the specified port. The first number is the port, and the second the data. Both values should be expressed in the current radix, and should be separated by a comma.

EXAMPLE: Send FF out port A7, where hex is the current radix.

```
H: O(ut)A7,FF  CR
```

P(rogram Counter)

The P(rogram Counter) command sets the current address in the program counter. This command is used to specify the object of the G(o) command. The address specified should be expressed in the current radix.

Simply typing P and RETURN causes the system to display the current contents of the program counter and then to await a new value. Typing P followed by a value sets the PC to that value. Typing P and RETURN, then pressing RETURN again without entering a value terminates the command and does not alter the PC.

EXAMPLE 1: Set the program counter to 100 hex, where the current radix is hexadecimal.

```
H: P(rogram Counter)  100  CR
H:
```

EXAMPLE 2: Display the contents of the program counter without altering its contents.

```
   H: P(rogram Counter)  CR
FFFF  CR
   H:
```

EXAMPLE 3: Set the program counter to 40100 octal after examining the current value, where the current radix is octal.

```
   H: P(rogram Counter)  CR
377377 40100
   H:
```

Note that the operator entered the 40100 in this example.

R(adix)

The R(adix) command sets the current working radix for all other commands.

Valid arguments to radix are O(ctal) and H(exadecimal). The default current radix on power up is octal. Typing R and RETURN with no argument displays the current radix.

EXAMPLE: Set the current radix to hexadecimal and then check it.

```
   H: R(adix) H(exadecimal)
   H: R(adix)    CR
Hexadecimal
   H:
```

S(ubstitute)

The substitute command can be used to examine or alter the contents of a memory location. The argument to S(ubstitute) is the first address to be examined (and optionally changed). When the starting address has been entered and terminated by pressing RETURN, the system displays address/value pairs. To replace the old value with a new one, type a new value, then a space. To proceed to the next memory location, type a space without entering anything else. To examine a previously displayed memory location, type a hyphen. To terminate, press RETURN.

EXAMPLE: Modify address 40100 octal, where octal is the current radix, then check the memory location.

```
   H: S(ubstitute) 40100   CR
40100 000  377 △  [operator types 377 and a space]
40101 000  -      [operator types a hyphen]
40100 377  CR     [operator presses RETURN]
   H:
```

T(est Memory)

The T(est Memory) command initiates the RAM memory test. The test references memory locations in the current radix. Error messages report the addresses of any bad memory locations.

V(iew)

The V(iew) command displays the contents of blocks of memory on the screen in the current radix and in ASCII. Non-printable characters appear as a graphics dot. Characters with the high order (parity) bit set appear in reverse video. The display begins at the first address specified, and continues through the second address. Starting and ending addresses should be separated with a comma.

If no starting or ending address is given, or if an address of zero is specified as the starting or ending address, the display begins at zero. V(iew) displays 128 bytes of data in octal if the current radix is octal, or 256 bytes in hexadecimal if the current radix is hexadecimal. Subsequent V(iew) commands which do not supply an argument display the next 128 or 256 bytes, depending on the setting of the current radix.

EXAMPLE 1: View the contents of memory locations 2280 through 2300 hex, where hexadecimal is the current radix.

```
          H: V(iew)2280,2300  ⏎
2280   20 21 32 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F  !"#$%&'()*+,-./
2290   20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F  !"#$%&'()*+,-./
          H:
```

EXAMPLE 2: View the contents of memory locations 0000 through 0128, where the current radix is octal. Then proceed to examine the next 128 bytes.

```
     H: V(iew)  ⏎
000000  303 000 004 041 012 040 303 073   C 0 0 ! 0   C ;
000010  000 315 132 000 026 000 303 201   0 M Z 0 0 0 C 0
000020  000 315 132 000 032 303 244 001   0 M Z 0 0 C $ 0
   .
   .
   .
000170  037 040 361 361 301 321 341 373   0 q q A Q a {
     H: V(iew)  ⏎
000200  311 052 033 040 043 042 033 040   I * 0   # " 0
etc.
```

EXAMPLE 3: View memory locations beginning with 2280 hex and continuing for 256 bytes, where hexadecimal is the current radix.

```
   H: V(iew) 2280,  ⏎
2280  4A 4A 4A 4A ... 4A  JJJJJJJJJJJJJJJJ
          .
          .
          .
2370  4A 4A 4A 4A ... 4A  JJJJJJJJJJJJJJJJ
```

PROGRAM EXECUTION CONTROL

When debugging an assembly language program, you can use MTR-90 commands to set breakpoints at, and continue execution from, various points in the program. Debugging can take place at any location above the lower 4K of memory. Be careful not to attempt to debug a program in the lower 4K of memory, as this area is occupied by MTR-90.

To set a breakpoint, use the S(ubstitute) command and put an HLT (hexadecimal 76, octal 166) instruction wherever you want the program to stop.

When your program reaches the breakpoint HLT instruction, it will return control to MTR-90 which will display an "H", then advance to a new line and display "H:". You can then use any MTR-90 command.

To continue your program, first restore the byte in the location into which you placed the breakpoint HLT. Since the computer had to execute the HLT instruction, the PC will point one beyond where you placed the HLT. To continue, decrement the PC value by one. Do this by entering the P(rogram Counter) command and pressing RETURN. When MTR-90 has displayed the current value of the PC, subtract one from that value, then enter the result into the PC.

You can alternatively use the G(o) command to start the program from whatever address you prefer, including from the location where you put the HLT.

Note that if the program which you are debugging uses keyboard interrupts, your program may contend with MTR-90 for console input. Your program should see every character input because the program receives the input via interrupts. But if the MTR-90 checks the keyboard for input after your program, the MTR-90 will not receive the input and no characters will be displayed on the screen. In other words, the fact that your keyboard input does not appear on the screen during program debugging using breakpoints does not mean that your program is at fault.

Appendix A

MTR-90 Listing

This appendix contains a listing of MTR-90. This program contains control routines for primitive keyboard input and screen output. MTR-90 needs available RAM in locations 2000H (040 000 octal) to 203FH (40 077 octal) and from 2150H (41 120 octal) to 2155 (41 125 octal). MTR-90 also needs 80 bytes of stack area in high memory.

```
000.001                 1  .DEBUG  EQU     1                       ASSEMBLE FOR DEBUG
                        2
                        6  ***     MTR90 - H/Z-89 MONITOR                  ISSUE 09.02.01
                        7  *
                        8  *       MTR89 IS A MODIFICATION OF MTR88 BY REX CHEN IN MAY, 1980.
                        9  *       MTR89 IS IDENTICAL TO THE MTR88 IN THAT ALL ENTRY POINTS TO
                       10  *       THE CURRENT ROUTINES REMAIN UNCHANGED AND ALL ROUTINES
                       11  *       REMAIN UNALTERED WITH THE FOLLOWING EXECPTIONS:
                       12  *
                       13  *         (1). "TYPE SPACES TO DETERMINE BAUD RATE" MESSAGE IS REMOVED.
                       14  *         (2). THE BOOTSTRAP FOR THE Z-47 IS INSTALLED.
                       15  *         (3). 15 SECONDS TIME OUT FOR Z-87, OR H-17 AND Z-47 IS INSERTED.
                       16  *         (4). <DELETE> KEY SERVES AS AN ABORT-BOOT KEY.
                       17  *         (5). ALLOWS BOOT FROM SELECT DEVICE AND UNIT.
                       18  *
                       19  *       MTR90 IS A MODIFICATION OF MTR89 TO ALLOW BOOTING FROM
                       20  *       THE H67, H37, AND 1 FUTURE DEVICE. ALSO THE H47 CODE WAS
                       21  *       CHANGED, AND HEXIDECIMAL ROUTINES WERE ADDED.
                       22  *       SEVERAL NEW "CONVENIENCE" COMMANDS WERE ADDED, THANKS TO
                       23  *       THE ADDITION OF THE EXTRA 2K SPACE.
                       24  *
                       25  *       MTR90-1 Employs a software fix for a hardware deficiency in disk
                       26  *       drives. It seems that a disk drive head may go into the negative
                       27  *       track area (-1, -2, ...) and not know it, so all disk drivers
                       28  *       have been modified to step the head in and then issuing a second
                       29  *       restore command. This can be taken care of in the hardware, but
                       30  *       people are opposed to adjusting hardware properly.
                       31  *
                       32  *       MTR88 IS AN ADAPTATION OF PAM/8 ORIGINALLY WRITTEN FOR THE
                       33  *       HEATH H8 COMPUTER BY J. G. LETWIN IN 1976 AND MODIFIED BY
                       34  *       R. N. BORCHARDT IN 1979 FOR USE IN THE HEATH H88/H89
                       35  *       COMPUTERS.
                       36  *
                       37  *       MTR88 PROVIDES COMPATABILITY WITH PAM/8 SUCH THAT ALL ROUTINES
                       38  *       HAVE RETAINED PREVIOUSLY DESCRIBED ENTRY POINTS AND ENTRY AND
                       39  *       EXIT CONDITIONS.  ROUTINES WHICH ARE NOT APPLICABLE SUCH AS
                       40  *       THOSE PERTAINING TO THE FRONT PANEL DISPLAY HAVE BEEN DELETED.
                       41  *
                       42  *       COPYRIGHT  05/1976, WINTEK CORPORATION
                       43  *                           902 N. 9TH ST.
                       44  *                           LAFAYETTE, IND.
                       45  *
                       46  *       COPYRIGHT  01/1979, HEATH COMPANY
                       47  *                           BENTON HARBOR, MI.
                       48  *
                       49  *       COPYRIGHT  05/1980, ZENITH DATA SYSTEMS INC.
                       50  *                           ST. JOSEPH, MI.
```

```
                ***     MTR88 - H88/H89 MONITOR.
                *
                *       THIS PROGRAM RESIDES (IN ROM) IN THE LOW 2048 BYTES OF THE HEATH
                *       H88/H89 COMPUTERS.


                ***     INTERRUPTS.
                *
                *       MTR88 IS THE PRIMARY PROCESSOR FOR ALL INTERRUPTS.
                *       THEY ARE PROCESSED AS FOLLOWS:
                *
                *       RST     USE
                *
                *       0       MASTER CLEAR. (NEVER USED FOR I/O OR RST)
                *
                *       1       CLOCK INTERRUPT. NORMALLY TAKEN BY MTR88,
                *               SETTING BIT *UO.CLK* IN BYTE *.MFLAG* ALLOWS
                *               USER PROCESSING (VIA A JUMP THROUGH *UIVEC*).
                *               UPON ENTRY OF THE USER ROUTINE, THE STACK
                *               CONTAINS:
                *               (STACK+0)  = RETURN ADDRESS (TO MTR88)
                *               (STACK+2)  = (STACKPTR+14)
                *               (STACK+4)  = (AF)
                *               (STACK+6)  = (BC)
                *               (STACK+8)  = (DE)
                *               (STACK+10) = (HL)
                *               (STACK+12) = (PC)
                *               THE USER'S ROUTINE SHOULD RETURN TO MTR88 VIA
                *               A *RET* WITHOUT ENABLING INTERRUPTS.
                *
                *       2       SINGLE STEP INTERRUPTS RECEIVED WHEN IN
                *               USER MODE CAUSES A JUMP THROUGH *UIVEC*+3.
                *               STACK UPON USER ROUTINE ENTRY:
                *               (STACK+0)  = (STACKPTR+12)
                *               (STACK+2)  = (AF)
                *               (STACK+4)  = (BC)
                *               (STACK+6)  = (DE)
                *               (STACK+8)  = (HL)
                *               (STACK+10) = (PC)
                *               THE USER'S ROUTINE SHOULD HANDLE IT'S OWN RETURN
                *               FROM THE INTERRUPT.
                *
                *
                *       THE FOLLOWING INTERRUPTS ARE VECTORED DIRECTLY THROUGH *UIVEC*.
                *       THE USER ROUTINE MUST HAVE SETUP A JUMP IN *UIVEC* BEFORE ANY
                *       OF THESE INTERRUPTS MAY OCCUR.
                *
                *       3       I/O 3. CAUSES A DIRECT JUMP THROUGH *UIVEC*+6
                *
                *       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
                *
```

```
                        107  *       7       I/O 7. CAUSES A DIRECT JUMP THROUGH *UIVEC*+18

                        109  **      ASSEMBLY CONSTANTS

    000.331             111  MI.EXX  EQU     331Q            Z80 EXX INSTRUCTION
                        112
    000.000             113          XTEXT   MTR88           DEFINE MTR88 OLD EQUATES
```

```
                    116X **     IO PORTS
                    117X
                    118X ***    ALL REFERENCES TO THE H8 FRONT PANEL PORTS ARE TRAPPED BY THE
                    119X *      Z80 NMI OF THE H88/H89.  OP.CTL WILL STILL PERFORM AS IN AN H8
                    120X *      IN RESPECT TO THE CLOCK AND SINGLE STEP CONTROL.  FOR MOTE
                    121X *      INFORMATION SEE THE NMI ROUTINE.
                    122X
000.360             123X IP.PAD EQU     360Q            PAD INPUT PORT
000.360             124X OP.CTL EQU     360Q            CONTROL OUTPUT PORT
000.360             125X OP.DIG EQU     360Q            DIGIT SELECT OUTPUT PORT
000.361             126X OP.SEG EQU     361Q            SEGMENT SELECT OUTPUT PORT
                    127X
                    128X *      H88/H89 CONTROL PORT
000.362             129X H88.CTL EQU    362Q            H88/H89 PORT FOR THE CLOCK AND SINGLE STEP
000.002             130X H88B.CK EQU    00000010B       2MS CLOCK ENABLE/DISABLE
000.001             131X H88B.SS EQU    00000001B       SINGLE STEP ENABLE/DISABLE
                    132X
000.362             133X H88.SW EQU     362Q            8 POSITION DIP SWITCH
000.200             134X H88S.AT EQU    10000000B       AUTO BOOT SWITCH
000.100             135X H88S.BR EQU    01000000B       BAUD RATE SWITCH
000.040             136X H88S.M EQU     00100000B       MEMORY TEST/NORMAL OPERATION SWITCH
000.020             137X H88S.DV EQU    00010000B       =0, BOOT FROM DEVICE AT 174-177Q
                    138X *                              =1, BOOT FROM DEVICE AT 170-173Q
000.014             139X H88S.0 EQU     00001100B       DEVICE AT 170-173Q: 0 = Z37, 1 = Z47
                    140X *                              2 = Z67, 3 = UNKNOWN
000.003             141X H88S.4 EQU     00000011B       DEVICE AT 174-177Q: 0 = H17, 1 = Z47
                    142X *                              2 = Z67, 3 = UNKNOWN


                    144X **     CASSETTE PORTS
                    145X
000.371             146X IP.TPC EQU     371Q            TAPE CONTROL IN
000.371             147X OP.TPC EQU     371Q            TAPE CONTROL OUT
000.370             148X IP.TPD EQU     370Q            TAPE DATA IN
000.370             149X OP.TPD EQU     370Q            TAPE DATA OUT


                    151X **     ASCII CHARACTERS
                    152X
000.026             153X A.SYN  EQU     026Q            SYNC CHARACTER
000.002             154X A.STX  EQU     002Q            STX CHARACTER
000.007             155X A.BEL  EQU     007Q            BELL CHARACTER
000.010             156X A.BKS  EQU     010Q            BACKSPACE CHARACTER
000.012             157X A.LF   EQU     012Q            LINE FEED CHARACTER
000.015             158X A.CR   EQU     015Q            CARRIAGE RETURN CHARACTER
000.033             159X A.ESC  EQU     033Q            ESCAPE CHARACTER
000.177             160X A.DEL  EQU     177Q            DELETE OR RUBOUT CHARACTER
```

```
                    162X **     FRONT PANEL HARDWARE CONTROL BITS
                    163X
000.020             164X CB.SSI EQU     00010000B       SINGLE STEP INTERRUPT
000.040             165X CB.MTL EQU     00100000B       MONITOR LIGHT
000.100             166X CB.CLI EQU     01000000B       CLOCK INTERRUPT ENABLE
000.200             167X CB.SPK EQU     10000000B       SPEAKER ENABLE


                    169X **     DISPLAY MODE FLAGS (IN *DSPMOD*)
                    170X
000.000             171X DM.MR  EQU     0               MEMORY READ
000.001             172X DM.MW  EQU     1               MEMORY WRITE
000.002             173X DM.RR  EQU     2               REGISTER READ
000.003             174X DM.RW  EQU     3               REGISTER WRITE


                    176X **     MACHINE INSTRUCTIONS
                    177X
000.166             178X MI.HLT EQU     01110110B       HALT
000.311             179X MI.RET EQU     11001001B       RETURN
000.333             180X MI.IN  EQU     11011011B       INPUT
000.323             181X MI.OUT EQU     11010011B       OUTPUT
000.072             182X MI.LDA EQU     00111010B       LDA
000.346             183X MI.ANI EQU     11100110B       ANI
000.021             184X MI.LXID EQU    00010001B       LXI D
000.303             185X MI.JMP EQU     11000011B       JMP
000.335             186X MI.LDXA EQU    11011101B       LD IX, (BYTE A)
000.041             187X MI.LDXB EQU    00100001B       LD IX, (BYTE B)
000.375             188X MI.LDYA EQU    11111101B       LD IY, (BYTE A)
000.041             189X MI.LDYB EQU    00100001B       LD IY, (BYTE B)
000.010             190X MI.EXAF EQU    00001000B       EX AF,AF'
000.335             191X MI.JIXA EQU    11011101B       JP (IX)  (BYTE A)
000.351             192X MI.JIXB EQU    11101001B       JP (IX)  (BYTE B)
000.375             193X MI.JIYA EQU    11111101B       JP (IY)  (BYTE A)
000.351             194X MI.JIYB EQU    11101001B       JP (IY)  (BYTE B)


                    196X **     USER OPTION BITS.
                    197X *
                    198X *      THESE BITS ARE SET IN CELL .MFLAG.
                    199X
000.200             200X UO.HLT EQU     10000000B       DISABLE HALT PROCESSING
000.100             201X UO.NFR EQU     CB.CLI          NO REFRESH FRONT PANEL
000.002             202X UO.DDU EQU     00000010B       DISABLE DISPLAY UPDATE
000.001             203X UO.CLK EQU     00000001B       ALLOW PRIVATE INTERRUPT PROCESSING
000.000             204         XTEXT   H17DEF          EQUATES FOR H17 BOOT ROM
```

```
                    206X **       H17 CONTROL INFORMATION.
                    207X
000.177             208X DP.DC    EQU     07FH            DISK CONTROL PORT
                    209X
000.001             210X DF.HD    EQU     00000001B       HOLE DETECT
000.002             211X DF.T0    EQU     00000010B       TRACK 0 DETECT
000.004             212X DF.WP    EQU     00000100B       WRITE PROTECT
000.010             213X DF.SD    EQU     00001000B       SYNC DETECT
                    214X
000.001             215X DF.WG    EQU     00000001B       WRITE GATE ENABLE
000.002             216X DF.DS0   EQU     00000010B       DRIVE SELECT 0
000.004             217X DF.DS1   EQU     00000100B       DRIVE SELECT 1
000.010             218X DF.DS2   EQU     00001000B       DRIVE SELECT 2
000.020             219X DF.MO    EQU     00010000B       MOTOR ON (BOTH DRIVES)
000.040             220X DF.DI    EQU     00100000B       DIRECTION (0=OUT)
000.100             221X DF.ST    EQU     01000000B       STEP COMMAND (ACTIVE HIGH)
000.200             222X DF.WR    EQU     10000000B       WRITE ENABLE RAM
                    223X
                    224X
                    225X
                    226X **       DISK UART PORTS AND CONTROL FLAGS.
                    227X
000.174             228X UP.DP    EQU     07CH            DATA PORT
000.175             229X UP.FC    EQU     07DH            FILL CHARACTER
000.175             230X UP.ST    EQU     07DH            STATUS FLAGS
000.176             231X UP.SC    EQU     07EH            SYN CHARACTER (OUTPUT)
000.176             232X UP.SR    EQU     07EH            SYNC RESET (INPUT)
                    233X
000.001             234X UF.RDA   EQU     00000001B       RECEIVE DATA AVAILABLE
000.002             235X UF.ROR   EQU     00000010B       RECEIVER OVERRUN
000.004             236X UF.RPE   EQU     00000100B       RECEIVER PARITY ERROR
000.100             237X UF.FCT   EQU     01000000B       FILL CHAR TRANSMITTED
000.200             238X UF.TBM   EQU     10000000B       TRANSMITTER BUFFER EMPTY
                    239X
                    240X
                    241X
                    242X **       CHARACTER DEFINITIONS.
                    243X
000.375             244X C.DSYN   EQU     0FDH            PREFIX SYNC CHARACTER
000.000             245           XTEXT   H37DEF          DEFINE H37 PARAMETERS
                    246X **       H37DEF - H37 DISK CONTROLLER DEFINITIONS
                    247X
000.170             248X DK.PORT  EQU     170Q    BASE UART PORT
                    249X
000.172             250X FD.STAT  EQU     DK.PORT+2       STATUS PORT
000.172             251X FD.CMD   EQU     DK.PORT+2       COMMAND PORT
000.173             252X FD.TRK   EQU     DK.PORT+3       TRACK REGISTER
000.172             253X FD.SEC   EQU     DK.PORT+2       SECTOR REGISTER
000.173             254X FD.DAT   EQU     DK.PORT+3       DATA PORT
000.170             255X DK.CON   EQU     DK.PORT         CONTROL PORT
000.171             256X DK.INT   EQU     DK.PORT+1       INTERFACE CONTROL
                    257X
                    258X **       COMMANDS SENT TO FD.CMD
                    259X
000.000             260X FDC.RST  EQU     00000000B       RESTORE
000.020             261X FDC.SEK  EQU     00010000B       SEEK TRACK IN FD.TRK
```

```
000.040          262X FDC.STP EQU     001000008      STEP IN SAME DIR AS LAST
000.100          263X FDC.STI EQU     010000008      STEP IN
000.140          264X FDC.STO EQU     011000008      STEP OUT
                 265X
000.200          266X FDC.RDS EQU     100000008      READ SECTOR
000.240          267X FDC.WTS EQU     101000008      WRITE SECTOR
                 268X
000.300          269X FDC.RDA EQU     110000008      READ ADDRESS
000.340          270X FDC.RDT EQU     111000008      READ TRACK
000.360          271X FDC.WTT EQU     111100008      WRITE TRACK
                 272X
000.320          273X FDC.FI  EQU     110100008      FORCE INTERRUPT
                 274X
                 275X **      OPTIONS FOR FDC.RST THRU FDC.STO
                 276X
000.020          277X FDF.UTR EQU     000100008      UPDATE TRACK REGISTER
000.010          278X FDF.HLB EQU     000010008      LOAD HEAD AT BEGINING
000.004          279X FDF.VRF EQU     000001008      VERIFY DESTINATION
                 280X
000.000          281X FDF.S6  EQU     000000008      STEP  6 MS
000.001          282X FDF.S12 EQU     000000018      STEP 12 MS
000.002          283X FDF.S20 EQU     000000108      STEP 20 MS
000.003          284X FDF.S30 EQU     000000118      STEP 30 MS
                 285X
                 286X **      OPTIONS FOR FDC.RDS THRU FDC.WTT
                 287X
000.020          288X FDF.MRF EQU     000100008      MULTY RECORD FLAG
000.010          289X FDF.SLF EQU     000010008      SECTOR LENGTH SHIFT RIGHT
000.004          290X FDF.DLF EQU     000001008      15 (30) MS DELAY
000.002          291X FDF.SS1 EQU     000000108      SELECT SIDE 1
000.001          292X FDF.DDM EQU     000000018      DELETED DATA MARK
                 293X
                 294X **      STATUS BIT DEFINITIONS
                 295X
000.200          296X FDS.NRD EQU     100000008      NOT READY
000.100          297X FDS.WPV EQU     010000008      WRITE PROTECT
000.040          298X FDS.HLD EQU     001000008      HEAD IS LOADED
000.040          299X FDS.RTE EQU     001000008      RECORD TYPE
000.040          300X FDS.WTF EQU     001000008      WRITE FAULT
000.020          301X FDS.SEK EQU     000100008      SEEK ERROR
000.020          302X FDS.RNF EQU     000100008      RECORD NOT FOUND
000.010          303X FDS.CRC EQU     000010008      CRC ERROR
000.004          304X FDS.TK0 EQU     000001008      OVER TRACK ZERO
000.004          305X FDS.LDT EQU     000001008      LOST DATA
000.002          306X FDS.IND EQU     000000108      INDEXZ PULSE
000.002          307X FDS.DRQ EQU     000000108      DATA REQUEST
000.001          308X FDS.BSY EQU     000000018      BUSY
                 309X
                 310X *       BITS SET IN DK.CON
                 311X
000.001          312X CON.EI  EQU     000000018              ENABLE INT-REQ
000.002          313X CON.DRQ EQU     000000108              ENABLE DRQ INTERRUPT
000.004          314X CON.MFM EQU     000001008              SET MFM RECORDING
000.010          315X CON.MO  EQU     000010008              ALL MOTORS ON
000.020          316X CON.DS0 EQU     000100008              DRIVE 0
000.040          317X CON.DS1 EQU     001000008              DRIVE 1
```

```
000.100  318X CON.DS2 EQU   010000008          DRIVE 2
000.200  319X CON.DS3 EQU   100000008          DRIVE 3
         320X
         321X
         322X *       Bits set to select alternate registers
         323X
000.000  324X CON.CD  EQU   000000008          SELECT COMMAND/DATA
000.001  325X CON.ST  EQU   000000018          SELECT SECTOR/TRACK
000.000  326         XTEXT  Z47DEF       DEFINE Z47 EQUATES


         328X **      H47DEF -      H47 Constant Definitions
         329X *


         331X *       Z80 Instructions
         332X
242.355  333X M.INI   EQU   10100010B*256+111011018    INI     Instruction
243.355  334X M.OUTI  EQU   10100011B*256+111011018    OUTI    Instructions


         336X **      DISK INTERFACE CONSTANTS
         337X *
         338X
000.170  339X D.STA   EQU   170Q                 INTERFACE STATUS PORT
000.171  340X D.DAT   EQU   D.STA+1              INTERFACE DATA PORT
         341X
000.001  342X S.ERR   EQU   000000018          ERROR BIT
000.040  343X S.DON   EQU   001000008          DONE
000.100  344X S.IEN   EQU   010000008          INTERRUPT ENABLE
000.200  345X S.DTR   EQU   100000008          DATA TRANSFER REQUEST
         346X
000.002  347X S.SW0   EQU   000000108          DIP SWITCH 0
000.004  348X S.SW1   EQU   000001008          DIP SWITCH 1
000.010  349X S.SW2   EQU   000010008          DIP SWITCH 2
000.020  350X S.SW3   EQU   000100008          DIP SWITCH 3
         351X
000.002  352X W.RES   EQU   000000108          RESET COMMAND


         354X **      STATUS BYTE FLAGS
         355X *
         356X
000.200  357X SB.UNR  EQU   100000008          UNIT NOT READY
000.100  358X SB.WPD  EQU   010000008          WRITE PROTECTED DRIVE
000.040  359X SB.DLD  EQU   001000008          DELETED DATA
000.020  360X SB.NRF  EQU   000100008          NO RECORD FOUND
000.010  361X SB.CRC  EQU   000010008          CRC ERROR
```

```
000.004                 362X SB.LTD  EQU     00000100B       LATE DATA
000.002                 363X SB.ILC  EQU     00000010B       ILLEGAL COMMAND
000.001                 364X SB.BTO  EQU     00000001B       BAD TRACK OVERFLOW


                        366X **      AUXILLARY STATUS BYTE FLAGS
                        367X *
                        368X
000.100                 369X AS.0DD  EQU     01000000B       TRACK 0 DOUBLE DENSITY
000.040                 370X AS.1DD  EQU     00100000B       TRACK 1-76 DOUBLE DENSITY
000.020                 371X AS.S1A  EQU     00010000B       SIDE 1 AVAILABLE
000.003                 372X AS.SLM  EQU     00000011B       SECTOR LENGTH MASK


                        374X **      DISK COMMANDS
                        375X *
                        376X
000.000                 377X         ORG     0
000.000                 378X DD.BOOT DS      1               BOOT
000.001                 379X DD.RST  DS      1               READ CONTROLLER STATUS
000.002                 380X DD.RAS  DS      1               READ AUX. STATUS
000.003                 381X DD.LSC  DS      1               LOAD SECTOR COUNT
000.004                 382X DD.RAD  DS      1               READ ADDR. OF LAST SECTOR ACCESSED
000.005                 383X DD.REA  DS      1               READ SECTORS
000.006                 384X DD.WRI  DS      1               WRITE SECTORS
000.007                 385X DD.REAB DS      1               READ SECTORS BUFFERED
000.010                 386X DD.WRIB DS      1               WRITE SECTORS BUFFERED
000.011                 387X DD.WRD  DS      1               WRITE SECTORS & DELETE
000.012                 388X DD.WRDB DS      1               WRITE SECTORS BUFFERED & DELETE
000.013                 389X DD.CPY  DS      1               COPY
000.014                 390X DD.FRM0 DS      1               FORMAT IBM SD
000.015                 391X DD.FRM1 DS      1               FORMAT     SD
000.016                 392X DD.FRM2 DS      1               FORMAT IBM DD
000.017                 393X DD.FRM3 DS      1               FORMAT     DD
000.020                 394X DD.RRDY DS      1               READ READY


                        396X **      Special De-Bug functions
                        397X *
                        398X
000.020                 399X         ORG     010H
000.020                 400X DD.SPF0 DS      1               SPECIAL FUNCTION 0
000.021                 401X DD.SPF1 DS      1               SPECIAL FUNCTION 1
000.022                 402X DD.SPF2 DS      1               SPECIAL FUNCTION 2
000.023                 403X DD.SPF3 DS      1               SPECIAL FUNCTION 3
000.024                 404X DD.SPF4 DS      1               SPECIAL FUNCTION 4
000.025                 405X DD.SPF5 DS      1               SPECIAL FUNCTION 5
```

```
                    407X **       Special Heath Functions
                    408X *
                    409X
000.200             410X          ORG     080H
000.200             411X DD.SDC   DS      1       SET DRIVE CHARACTERISTIC
000.201             412X DD.ST    DS      1       SEEK TO TRACK
000.202             413X DD.DS    DS      1       DISK STATUS
000.203             414X DD.RDL   DS      1       READ LOGICAL
000.204             415X DD.WTL   DS      1       WRITE LOGICAL
000.205             416X DD.RDBL  DS      1       READ BUFFERED LOGICAL
000.206             417X DD.WTBL  DS      1       WRITE BUFFERED LOGICAL
000.207             418X DD.WTDL  DS      1       WRITE DELETED DATA LOGICAL
000.210             419X DD.WDLB  DS      1       WRITE BUFFERED DELETED DATA LOGICAL


                    421X **       USEFUL FLAGS
                    422X *
                    423X
000.000             424X UNT.0    EQU     00000000B       UNIT 0
000.040             425X UNT.1    EQU     00100000B       UNIT 1
000.100             426X UNT.2    EQU     01000000B       UNIT 2
000.140             427X UNT.3    EQU     01100000B       UNIT 3
                    428X
000.140             429X UNT.M    EQU     01100000B       Unit mask
                    430X
                    431X
                    432X
000.000             433X SID.0    EQU     00000000B       Side: 0
000.200             434X SID.1    EQU     10000000B       Side: 1
                    435X
000.200             436X SID.M    EQU     10000000B       Side Mask
                    437X
                    438X
                    439X
000.037             440X SEC.M    EQU     00011111B       Track Mask
                    441X
                    442X
                    443X
004.000             444X SSIZ.M   EQU     1024            Maximum Sector Size
                    445X
                    446X
                    447X *C.256   EQU     256             SECTOR SIZE = 256 BYTES
                    448X *C.128   EQU     128             SECTOR SIZE
                    449X *C.26    EQU     26
000.211             450           XTEXT   H67DEF          H67 DEFINITIONS
```

```
                        453X **      H67 Disk Controller Definitions
                        454X *

                        456X **      Register addresses
                        457X *
                        458X
000.170                 459X BASE    EQU     170Q        Controller base address
                        460X
000.000                 461X RI.DAT  EQU     0           Data In/Out (Read/Write)
000.001                 462X RI.CON  EQU     1           Control (Write Only)
000.001                 463X RI.BST  EQU     1           Bus Status (Read Only)

                        465X *       Control Register Definition
                        466X
000.100                 467X BC.SEL  EQU     010000000B  Select and data bit 0
000.040                 468X BC.IE   EQU     001000000B  Interrupt Enable
000.020                 469X BC.RST  EQU     000100000B  Reset
000.002                 470X BC.EDT  EQU     000000010B  Enable Data

                        472X *       Bus Status Register Definition
                        473X
000.200                 474X BS.REQ  EQU     100000000B  Data Transfer Request
000.100                 475X BS.DTD  EQU     010000000B  Data Transfer Direction
000.000                 476X BS.IN   EQU     000000000B   Data to Host
000.100                 477X BS.OUT  EQU     010000000B   Data to Controller
000.040                 478X BS.LMB  EQU     001000000B  Last byte in data/command string
000.020                 479X BS.MTY  EQU     000100000B  Message type
000.000                 480X BS.DAT  EQU     000000000B   Data
000.020                 481X BS.COM  EQU     000100000B   Command
000.010                 482X BS.BSY  EQU     000010000B  Busy
000.004                 483X BS.INT  EQU     000001000B  Interrupt Pending
000.002                 484X BS.PE   EQU     000000100B  Parity Error
000.001                 485X BS.HID  EQU     000000001B  Hardware Identification

                        487X *       Status Byte Definitions
                        488X
000.140                 489X ST.LUN  EQU     011000000B  Logical Unit
000.034                 490X ST.SPR  EQU     000111000B  Spare
000.002                 491X ST.ERR  EQU     000000010B  Error
000.001                 492X ST.PER  EQU     000000001B  Parity Error
```

```
                        494X **       Commands
                        495X *
                        496X
000.340                 497X CLASSM   EQU     11100000B       Class Mask
                        498X
000.000                 499X CLASS0   EQU     00000000B       Class 0
000.040                 500X CLASS1   EQU     00100000B       Class 1
000.300                 501X CLASS6   EQU     11000000B       Class 6
                        502X
000.037                 503X OPCODM   EQU     00011111B       Op-code Mask
000.140                 504X LUNM     EQU     01100000B       Logical Unit Mask
000.037                 505X LSA.2    EQU     00011111B       Logical Sector Address (2)


                        507X *        Class 0 Commands
                        508X
000.000                 509X D.TDR    EQU     CLASS0+0        Test drive ready
000.001                 510X D.REC    EQU     CLASS0+1        Recalibrate drive
000.002                 511X D.RSY    EQU     CLASS0+2        Request Syndrome
000.003                 512X D.RSE    EQU     CLASS0+3        Request Sense
000.004                 513X D.FOR    EQU     CLASS0+4        Format Drive
000.005                 514X D.CTF    EQU     CLASS0+5        Check track format
000.006                 515X D.FT     EQU     CLASS0+6        Format Track
000.007                 516X D.FBS    EQU     CLASS0+7        Format bad sector
000.010                 517X D.REA    EQU     CLASS0+8        Read
000.011                 518X D.WPS    EQU     CLASS0+9        Write protect the sector
000.012                 519X D.WRI    EQU     CLASS0+10       Write
000.013                 520X D.SEK    EQU     CLASS0+11       Seek


                        522X *        Class 1 Commands
                        523X
000.040                 524X D.CPB    EQU     CLASS1+0        Copy block


                        526X *        Class 6 Commands
                        527X
000.300                 528X D.FFD    EQU     CLASS6+0        Format floppy disk
```

```
                    530X *      Type 0 error codes (Drive error Codes)
                    531X
000.000             532X T0.NST EQU     0           No status
000.001             533X T0.NIS EQU     1           No Index signal
000.002             534X T0.NSC EQU     2           No seek complete
000.003             535X T0.WFT EQU     3           Write fault
000.004             536X T0.DNR EQU     4           Drive not ready
000.005             537X T0.DNS EQU     5           Drive not selected
000.006             538X T0.NT0 EQU     6           No track zero
000.007             539X T0.MDS EQU     7           Multi-drive selected


                    541X *      Type 1 error codes (data error codes)
                    542X
000.000             543X T1.ID  EQU     0           ID Read Error
000.001             544X T1.UDE EQU     1           Uncorrectable data error
000.002             545X T1.IDNF EQU    2           ID Address Mark not found
000.003             546X T1.DMNF EQU    3           Data Address Mark Not Found
000.004             547X T1.RNF EQU     4           Record Not Found
000.005             548X T1.SKE EQU     5           Seek Error
000.006             549X T1.DTE EQU     6           DMA Time-out Error  (not used)
000.007             550X T1.WP  EQU     7           Write protected
000.010             551X T1.CDE EQU     8           Correctable Data field Error
000.011             552X T1.BBF EQU     9           Bad Block Found
000.012             553X T1.FE  EQU     10          Format Error


                    555X *      Type 2 Error Codes (Command error codes)
                    556X
000.000             557X T2.ILC EQU     0           Illegal Command
000.001             558X T2.IDA EQU     1           Illegal Disk Address
000.002             559X T2.IFN EQU     2           Illegal Function
000.211             560         XTEXT   HDOSEQU     HDOS EQUATES


                    562X **     HDOS SYSTEM EQUIVALENCES.
                    563X *
                    564X
024.000             565X S.GRT0 EQU     24000A      SYSTEM AREA FOR  GRT0
025.000             566X S.GRT1 EQU     25000A      SYSTEM AREA FOR  GRT1
026.000             567X S.GRT2 EQU     26000A      SYSTEM AREA FOR  GRT2
                    568X
030.000             569X ROMBOOT EQU    30000A      ROM BOOT ENTRY
                    570X
040.100             571X        ORG     40100A      FREE SPACE FROM PAM-8
                    572X
040.100             573X        DS      8           JUMP TO SYSTEM EXIT
040.110             574X D.CON  DS      16          DISK CONSTANTS
040.130             575X SYDD   EQU     *           SYSTEM DISK ENTRY POINT
040.130             576X D.VEC  DS      24*3        SYSTEM ROM ENTRY VECTORS
```

```
040.240          577X D.RAM   DS      31              SYSTEM ROM WORK AREA
040.277          578X S.VAL   DS      36              SYSTEM VALUES
040.343          579X S.INT   DS      115             SYSTEM INTERNAL WORK AREAS
041.126          580X         DS      16
041.146          581X S.SOVR  DS      2               STACK OVERFLOW WARNING
041.150          582X         DS      42200A-*        SYSTEM STACK
001.032          583X STACKL  EQU     *-S.SOVR        STACK SIZE
                 584X
042.200          585X STACK   EQU     *               LWA+1 SYSTEM STACK
042.200          586X USERFWA EQU     *               USER FWA
042.200          587          XTEXT   DIRDEF


                 589X **      DIRECTORY ENTRY FORMAT.
                 590X
000.000          591X         ORG     0
                 592X
                 593X
000.377          594X DF.EMP  EQU     377Q            FLAGS ENTRY EMPTY
000.376          595X DF.CLR  EQU     376Q            FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
                 596X
000.000          597X DIR.NAM DS      8               NAME
000.010          598X DIR.EXT DS      3               EXTENSION
000.013          599X DIR.PRO DS      1               PROJECT
000.014          600X DIR.VER DS      1               VERSION
000.015          601X DIRIDL  EQU     *               FILE IDENTIFICATION LENGTH
                 602X
000.015          603X DIR.CLU DS      1               CLUSTER FACTOR
000.016          604X DIR.FLG DS      1               FLAGS
000.017          605X         DS      1               RESERVED
000.020          606X DIR.FGN DS      1               FIRST GROUP NUMBER
000.021          607X DIR.LGN DS      1               LAST GROUP NUMBER
000.022          608X DIR.LSI DS      1               LAST SECTOR INDEX (IN LAST GROUP)
000.023          609X DIR.CRD DS      2               CREATION DATE
000.025          610X DIR.ALD DS      2               LAST ALTERATION DATE
                 611X
000.027          612X DIRELEN EQU     *               DIRECTORY ENTRY LENGTH
000.027          613          XTEXT   ESINT


                 615X **      S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
                 616X *
                 617X *       THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
                 618X *       MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
                 619X
                 620X
040.343          621X         ORG     S.INT
                 622X
                 623X **      CONSOLE STATUS FLAGS
                 624X
040.343          625X S.CDB   DS      1               CONSOLE DESCRIPTOR BYTE
000.000          626X CDB.H85 EQU     000000008
000.001          627X CDB.H84 EQU     000000018       =0 IF H8-5, =1 IF H8-4
```

```
040.344  628X S.BAUD  DS      2          [0-14]  H8-4 BAUD RATE, =0 IF H8-5
         629X *                           [15]    =1 IF 2 STOP BITS
         630X
         631X **       TABLE ADDRESS WORDS
         632X
040.346  633X S.DLINK DS      2          ADDRESS OF DATA IN HDOS CODE
040.350  634X S.OFWA  DS      2          FWA  OVERLAY  TABLE
040.352  635X S.CFWA  DS      2          FWA  CHANNEL  TABLE
040.354  636X S.DFWA  DS      2          FWA  DEVICE   TABLE
040.356  637X S.RFWA  DS      2          FWA  RESIDENT HDOS CODE
         638X
         639X **       DEVICE DRIVER DELAYED LOAD FLAGS
         640X
040.360  641X S.DDLDA DS      2          DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
040.362  642X S.DDLEN DS      2          CODE LENGTH IN BYTES
040.364  643X S.DDGRP DS      1          GROUP NUMBER FOR DRIVER
040.365  644X         DS      1          HOLD PLACE
         645X *S.DDSEC DS      2          SECTOR NUMBER FOR DRIVER ( * OBSOLETE ! * )
040.366  646X S.DDDTA DS      2          DEVICE'S ADDRESS IN DEVLST +DEV.RES
040.370  647X S.DDOPC DS      1          OPEN OPCODE PENDING
         648X
         649X **       OVERLAY MANAGEMENT FLAGS
         650X
000.001  651X OVL.IN  EQU     000000001B IN MEMORY
000.002  652X OVL.RES EQU     000000010B PERMANENTLY RESIDENT
000.014  653X OVL.NUM EQU     000011000B OVERLAY NUMBER MASK
000.200  654X OVL.UCS EQU     100000000B USER CODE SWAPPED FOR OVERLAY
         655X
040.371  656X S.OVLFL DS      1          OVERLAY FLAG
040.372  657X S.UCSF  DS      2          FWA SWAPPED USER CODE
040.374  658X S.UCSL  DS      2          LENGTH SWAPPED USER CODE
040.376  659X S.OVLS  DS      2          SIZE OF OVERLAY CODE
041.000  660X S.OVLE  DS      2          ENTRY POINT OF OVERLAY CODE
         661X
041.002  662X S.SSN   DS      2          SWAP AREA SECTOR NUMBER
041.004  663X S.OSN   DS      2          OVERLAY SECTOR NUMBER
         664X
         665X *       SYSCALL PROCESSING WORK AREAS
         666X
041.006  667X S.CACC  DS      1          (ACC) UPON SYSCALL
041.007  668X S.CODE  DS      1          SYSCALL INDEX IN PROGRESS
         669X
         670X *       JUMPS TO ROUTINES IN RESIDENT HDOS CODE
         671X
041.010  672X S.JUMPS DS      0          START OF JUMP VECTORS
041.010  673X S.SDD   DS      3          JUMP TO STAND-IN DEVICE DRIVER
041.013  674X S.FASER DS      3          JUMP TO FATSERR (FATAL SYSTEM ERROR)
041.016  675X S.DIREA DS      3          JUMP TO DIREAD (DISK FILE READ)
041.021  676X S.FCI   DS      3          JUMP TO FCI (FETCH CHANNEL INFO)
041.024  677X S.SCI   DS      3          JUMP TO SCI (STORE CHANNEL INFO)
041.027  678X S.GUP   DS      3          JUMP TO GUP (GET UNIT POINTER)
         679X
041.032  680X S.MOUNT DS      1          <>0 IF THE SYSTEM DISK IS MOUNTED
041.033  681X S.DCS   DS      1          DEFAULT CLUSTER SIZE-1
         682X
041.034  683X S.BOOTF DS      1          BOOT FLAGS
```

```
000.001                 684X BOOT.P  EQU     000000018       EXECUTE PROLOGUE UPON BOOTUP
000.002                 685X BOOT.SY EQU     000000108       SY: Device Driver loaded                /2.1b/
                        686X
                        687X *       STACK VALUE SAVED FOR OVERLAY SYSCALLS
                        688X
041.035                 689X S.OVSTK DS      2               VALUE OF SP UPON SYSCALLS USING OVERLAY
                        690X
041.037                 691X         DS      1               RESERVED


                        693X **      ACTIVE I/O AREA.
                        694X *
                        695X *       THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION
                        696X *       CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM
                        697X *       THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.
                        698X *
                        699X *       NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY
                        700X *       FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE
                        701X *       8080 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY
                        702X *       COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND
                        703X *       BACKDATED AFTER PROCESSING.
                        704X
041.040                 705X AIO.VEC DS      3               JUMP INSTRUCTION
041.041                 706X AIO.DDA EQU     *-2             DEVICE DRIVER ADDRESS
041.043                 707X AIO.FLG DS      1               FLAG BYTE
041.044                 708X AIO.GRT DS      2               ADDRESS OF GROUP RESERV TABLE
041.046                 709X AIO.SPG DS      1               SECTORS PER GROUP
041.047                 710X AIO.CGN DS      1               CURRENT GROUP NUMBER
041.050                 711X AIO.CSI DS      1               CURRENT SECTOR INDEX
041.051                 712X AIO.LGN DS      1               LAST GROUP NUMBER
041.052                 713X AIO.LSI DS      1               LAST SECTOR INDEX
041.053                 714X AIO.DTA DS      2               DEVICE TABLE ADDRESS
041.055                 715X AIO.DES DS      2               DIRECTORY SECTOR
041.057                 716X AIO.DEV DS      2               DEVICE CODE
041.061                 717X AIO.UNI DS      1               UNIT NUMBER (0-9)
                        718X
041.062                 719X AIO.DIR DS      DIRELEN         DIRECTORY ENTRY
                        720X
041.111                 721X AIO.CNT DS      1               SECTOR COUNT
041.112                 722X AIO.EOM DS      1               END OF MEDIA FLAG
041.113                 723X AIO.EOF DS      1               END OF FILE FLAG
041.114                 724X AIO.TFP DS      2               TEMP FILE POINTERS
041.116                 725X AIO.CHA DS      2               ADDRESS OF CHANNEL BLOCK (IOC.DDA)


041.120                 727X S.BDA   DS      1               Boot Device address (Setup by ROM) /80.09.gc/
041.121                 728X S.SCR   DS      2               SYSTEM SCRATCH AREA ADDRESS
041.123                 729X         DS      3
000.000                 730X         ERRNZ   *-41126A
041.126                 731X S.OSI   DS      1               Operating System ID                     /2.1b/
041.127                 732X S.OSO   DS      1               Operating System Occurance              /2.1b/
041.130                 733X S.OSZ   DS      3               Operating System Sector Zero            /2.1b/
```

```
041.133                 734             XTEXT   MISC            MISCELLANEOUS EQUATES FOR H17 BOOT ROM


                        737X **         MISCELLANEOUS EQUATES FROM H17 BOOT ROM.
                        738X *          REFER TO H17 BOOT ROM IF MORE INFORMATION DESIRED
                        739X
036.235                 740X WHD        EQU     36235A          WAIT FOR HOLE ROUTINE ENTRY POINT
036.271                 741X WNH        EQU     36271A          WAIT FOR NO HOLE ROUTINE ENTRY POINT
                        742X
000.130                 743X BOOTAL     EQU     130A            NUMBER OF RAM TO CLEAR
037.132                 744X BOOTA      EQU     37132A          RAM CLEAR START LOCATION
030.252                 745X $MOVE      EQU     30252A          MOVE DATA ROUTINE
000.037                 746X D.RAML     EQU     37Q
031.212                 747X $ZERO      EQU     31212A          ZERO RAM ROUTINE
040.037                 748X .UIVEC     EQU     40037A          USER INTERRUPT VECTOR
034.031                 749X CLOCK17    EQU     34031A          Z17 TIMER INTERRUPT HANDLER LOCATION
033.366                 750X R.ABORT    EQU     33366A          RESET Z17 ROUTINE LOCATION
034.077                 751X R.READ     EQU     34077A          READ Z17 ROUTINE LOCATION
040.206                 752X D.SDP      EQU     40206A          SET DEVICE PARAMETER RAM LOCATION
040.166                 753X D.SDT      EQU     40166A          SEEK DESIRED TRACK
040.240                 754X D.TT       EQU     40240A          TARGET TRACK BYTE
036.073                 755X SDP3       EQU     36073A          SET DEVICE PARAMETER ENTRY
034.027                 756X EIXIT      EQU     34027A          EI/RET LOCATION
000.012                 757X ERPTCNT    EQU     12Q             ERROR COUNT
040.264                 758X D.OECNT    EQU     40264A
041.133                 759             XTEXT   U8251           DEFINE 8251 USART BITS
```

```
                        762X **         8251 USART BIT DEFINITIONS.
                        763X *
                        764X
                        765X **         PORT ADDRESSES
                        766X
000.000                 767X UDR        EQU     0               DATA REGISTER IS EVEN
000.001                 768X USR        EQU     1               STATUS REGISTER IS NEXT
                        769X
000.372                 770X SC.UART    EQU     372Q            CONSOLE USART ADDRESS (IFF 8251)
                        771X
                        772X
                        773X **         MODE INSTRUCTION CONTROL BITS.
                        774X
000.100                 775X UMI.1B     EQU     01000000B       1 STOP BIT
000.200                 776X UMI.HB     EQU     10000000B       1 1/2 STOP BITS
000.300                 777X UMI.2B     EQU     11000000B       2 STOP BITS
000.040                 778X UMI.PE     EQU     00100000B       EVEN PARITY
000.020                 779X UMI.PA     EQU     00010000B       USE PARITY
000.000                 780X UMI.L5     EQU     00000000B       5 BIT CHARACTERS
000.004                 781X UMI.L6     EQU     00000100B       6 BIT CHARACTERS
000.010                 782X UMI.L7     EQU     00001000B       7 BIT CHARACTERS
000.014                 783X UMI.L8     EQU     00001100B       8 BIT CHARACTERS
000.001                 784X UMI.1X     EQU     00000001B       CLOCK X 1
000.002                 785X UMI.16X    EQU     00000010B       CLOCK X 16
000.003                 786X UMI.64X    EQU     00000011B       CLOCK X 64
                        787X
                        788X **         COMMAND INSTRUCTION BITS.
                        789X
000.100                 790X UCI.IR     EQU     01000000B       INTERNAL RESET
000.040                 791X UCI.RO     EQU     00100000B       READER-ON CONTROL FLAG
000.020                 792X UCI.ER     EQU     00010000B       ERROR RESET
000.004                 793X UCI.RE     EQU     00001000B       RECEIVE ENABLE
000.002                 794X UCI.IE     EQU     00000010B       ENABLE INTERRUPTS FLAG
000.001                 795X UCI.TE     EQU     00000001B       TRANSMIT ENABLE
                        796X
                        797X **         STATUS READ COMMAND BITS.
                        798X
000.040                 799X USR.FE     EQU     00100000B       FRAMING ERROR
000.020                 800X USR.OE     EQU     00010000B       OVERRUN ERROR
000.010                 801X USR.PE     EQU     00001000B       PARITY ERROR
000.004                 802X USR.TXE    EQU     00000100B       TRANSMITTER EMPTY
000.002                 803X USR.RXR    EQU     00000010B       RECEIVER READY
000.001                 804X USR.TXR    EQU     00000001B       TRANSMITTER READY
041.133                 805             XTEXT   U8250           DEFINE 8250 ACE BITS


                        807X **         8250 UART CONTROL AND BIT DEFINITIONS.
                        808X
000.350                 809X SC.ACE     EQU     350Q            SYSTEM CONSOLE PORT IF 8250 ACE
000.156                 810X AC.DLY     EQU     110             220 MIL. SEC. DELAY FOR 8250
                        811X
000.000                 812X UR.RBR     EQU     0               RECEIVER BUFFER REGISTER (READ ONLY)
                        813X
000.000                 814X UR.THR     EQU     0               TRANSMITTER HOLDING REGISTER (WRITE ONLY)
```

```
                    815X
000.000             816X UR.DLL  EQU  0          DIVISOR LATCH (LEAST SIGNIFICANT)
                    817X
000.001             818X UR.DLM  EQU  1          DIVISOR LATCH (MOST SIGNIFICANT)
                    819X
000.001             820X UR.IER  EQU  1          INTERRUPT ENABLE REGISTER
000.001             821X UC.EDA  EQU  00000001B   ENABLE RECEIVED DATA AVAILABLE INTERRUPT
000.002             822X UC.TRE  EQU  00000010B   ENABLE TRANSMIT HOLD REGISTER EMPTY INTERRUPT
000.004             823X UC.RSI  EQU  00000100B   ENABLE RECEIVE STATUS INTERRUPT
000.010             824X UC.MSI  EQU  00001000B   ENABLE MODEM STATUS INTERRUPT
                    825X
000.002             826X UR.IIR  EQU  2          INTERRUPT IDENTIFICATION REGISTER
000.001             827X UC.IIP  EQU  00000001B   INVERTED INTERRUPT PENDING (0 MEANS PENDING)
000.006             828X UC.IID  EQU  00000110B   INTERRUPT ID
                    829X
000.003             830X UR.LCR  EQU  3          LINE CONTROL REGISTER
000.000             831X UC.5BW  EQU  00000000B   5 BIT WORDS
000.001             832X UC.6BW  EQU  00000001B   6 BIT WORDS
000.002             833X UC.7BW  EQU  00000010B   7 BIT WORDS
000.003             834X UC.8BW  EQU  00000011B   8 BIT WORDS
000.004             835X UC.2SB  EQU  00000100B   TWO STOP BITS SELECTED
000.010             836X UC.PEN  EQU  00001000B   PARITY COMPUTATION ENABLED
000.020             837X UC.EPS  EQU  00010000B   EVEN PARITY SELECT
000.040             838X UC.SKP  EQU  00100000B   STICK PARITY
000.100             839X UC.SB   EQU  01000000B   SET BREAK
000.200             840X UC.DLA  EQU  10000000B   DIVISOR LATCH ACCESS
                    841X
000.004             842X UR.MCR  EQU  4          MODEM CONTROL REGISTER
000.001             843X UC.DTR  EQU  00000001B   DATA TERMINAL READY
000.002             844X UC.RTS  EQU  00000010B   REQUEST TO SEND
000.004             845X UC.OU1  EQU  00000100B   OUT 1
000.010             846X UC.OU2  EQU  00001000B   OUT 2
000.020             847X UC.LOO  EQU  00010000B   LOOP
                    848X
000.005             849X UR.LSR  EQU  5          LINE STATUS REGISTER
000.001             850X UC.DR   EQU  00000001B   DATA READY
000.002             851X UC.OR   EQU  00000010B   OVERRUN
000.004             852X UC.PE   EQU  00000100B   PARITY ERROR
000.010             853X UC.FE   EQU  00001000B   FRAMING ERROR
000.020             854X UC.BI   EQU  00010000B   BREAK INTERRUPT
000.040             855X UC.THE  EQU  00100000B   TRANSMITTER HOLDING REGISTER EMPTY
000.100             856X UC.TSE  EQU  01000000B   TRANSMITTER SHIFT REGISTER EMPTY
                    857X
000.006             858X UR.MSR  EQU  6          MODEM STATUS REGISTER
000.001             859X UC.DCS  EQU  00000001B   DELTA CLEAR TO SEND
000.002             860X UC.DDR  EQU  00000010B   DELTA DATA SET READY
000.004             861X UC.TER  EQU  00000100B   TRAILING EDGE OF RING
000.010             862X UC.DRL  EQU  00001000B   DELTA RECEIVE LINE SIGNAL DETECT
000.020             863X UC.CTS  EQU  00010000B   CLEAR TO SEND
000.040             864X UC.DSR  EQU  00100000B   DATA SET READY
000.100             865X UC.RI   EQU  01000000B   RING INDICATOR
000.200             866X UC.RLS  EQU  10000000B   RECEIVED LINE SIGNAL DETECT
```

```
                                869  ***     INTERRUPT VECTORS.
                                870  *
                                871

                                873  **      LEVEL 0 - RESET
                                874  *
                                875  *       THIS 'INTERRUPT' MAY NOT BE PROCESSED BY A USER PROGRAM.
                                876
000.000                         877          ORG     00A
                                878
000.000  303 000 004           879  INIT0   JMP     INITOX          DO H88 EXTENSION OF INITIALIZATION
000.003  041 012 040           880  INIT0.0 LXI     H,PRSRAM+PRSL-1 (HL) = RAM DESTINATION FOR CODE
000.006  303 073 000           881          JMP     INIT            INITIALIZE
                                882
377.073                         883          ERRPL   INIT-1000A     BYTE IN WORD 10A MUST BE 0
                                884

                                886  **      LEVEL 1 - CLOCK
                                887
000.010                         888  INT1    EQU     10Q             INTERRUPT ENTRY POINT
                                889
000.000                         890          ERRNZ   *-11Q          INT0 TAKES UP ONE BYTE
                                891
000.011  315 132 000           892          CALL    SAVALL          SAVE USER REGISTERS
000.014  026 000               893          MVI     D,0
000.016  303 201 000           894          JMP     CLOCK           PROCESS CLOCK INTERRUPT
377.201                         895          ERRPL   CLOCK-1000A     EXTRA BYTE MUST BE 0

                                897  **      LEVEL 2 - SINGLE STEP
                                898  *
                                899  *       IF THIS INTERRUPT IS RECEIVED WHEN NOT IN MONITOR MODE,
                                900  *       THEN IT IS ASSUMED TO BE GENERATED BY A USER PROGRAM
                                901  *       (SINGLE STEPPING OR BREAKPOINTING). IN SUCH CASE, THE
                                902  *       USER PROGRAM IS ENTERED THROUGH (UIVEC+3
                                903
000.020                         904  INT2    EQU     20A             LEVEL 2 ENTRY
                                905
000.000                         906          ERRNZ   *-21A          INT1 TAKES EXTRA BYTE
                                907
000.021  315 132 000           908          CALL    SAVALL          SAVE REGISTERS
000.024  032                   909          LDAX    D               (A) = (CTLFLG)
040.011                         910  .       SET     CTLFLG
000.025  303 244 001           911          JMP     STPRTN          STEP RETURN
```

```
                          913  ***     I/O INTERRUPT VECTORS.
                          914  *
                          915  *       INTERRUPTS 3 THROUGH 7 ARE AVAILABLE FOR GENERAL I/O USE.
                          916  *
                          917  *       THESE INTERRUPTS ARE NOT SUPPORTED BY MTR88, AND SHOULD
                          918  *       NEVER OCCUR UNLESS THE USER HAS SUPPLIED HANDLER ROUTINES
                          919  *       (THROUGH UIVEC)
                          920
000.030                   921          ORG     30A
                          922
000.030  303 045 040     923  INT3    JMP     UIVEC+6         JUMP TO USER ROUTINE
                          924
000.033  102 061 064     925          DB      102Q,61Q,64Q,62Q,102Q   PART NUMBER 444-142


                          927
000.040                   928          ORG     40A
                          929
000.040  303 050 040     930  INT4    JMP     UIVEC+9         JUMP TO USER ROUTINE
                          931
000.043  044 122 116     932          DB      44Q,122Q,116Q,102Q,44Q  SUPPORT CODE


                          934
000.050                   935          ORG     50A
                          936
000.050  303 053 040     937  INT5    JMP     UIVEC+12        JUMP TO USER ROUTINE
                          938
                          939
                          940  **      DLY - DELAY TIME INTERVAL.
                          941  *
                          942  *       ENTRY   (A) = MILLISECOND DELAY COUNT/2
                          943  *       EXIT    NONE
                          944  *       USES    A,F
                          945
000.000                   946          ERRNZ   *-53A
                          947
000.053  365             948  DLY     PUSH    PSW             SAVE COUNT
000.054  257             949          XRA     A               DONT SOUND HORN
000.055  303 143 002     950          JMP     HRN0            PROCESS AS HORN


                          952
000.060                   953          ORG     60A
                          954
000.060  303 056 040     955  INT6    JMP     UIVEC+15        JUMP TO USER ROUTINE
                          956
                          957
000.063  076 320         958  GO.     MVI     A,C8.SSI+C8.CLI+C8.SPK  OFF MONITOR MODE LIGHT
000.065  303 235 001     959          JMP     SST1            RETURN TO USER PROGRAM
```

```
                                961
000.070                         962             ORG     70A
                                963
000.070  303 061 040            964     INT7    JMP     UIVEC+18        JUMP TO USER ROUTINE
```

```
                          967  **      INIT - INITIALIZE SYSTEM
                          968  *
                          969  *       INIT IS CALLED WHENEVER A HARDWARE MASTER-CLEAR IS INITIATED.
                          970  *
                          971  *       SETUP MTR88 CONTROL CELLS IN RAM.
                          972  *       DECODE HOW MUCH MEMORY EXISTS, SETUP STACKPOINTER, AND
                          973  *       ENTER THE MONITOR LOOP.
                          974  *
                          975  *       ENTRY   FROM MASTER CLEAR
                          976  *       EXIT    INTO MTR88 MAIN LOOP
                          977
 000.000                  978          ERRNZ   *-730
                          979
 000.073  032             980  INIT    LDAX    D               COPY *PRSROM* INTO RAM
 000.074  167             981          MOV     M,A             MOVE BYTE
 000.075  053             982          DCX     H               DECREMENT DESTINATION
 000.076  034             983          INR     E               INCREMENT SOURCE
 000.077  302 073 000      984          JNZ     INIT            IF NOT DONE
                          985
 004.000                  986  SINCR   EQU     4000A           SEARCH INCREMENT
                          987
 000.102  026 004         988          MVI     D,SINCR/256     (DE) = SEARCH INCREMENT
 000.104  041 000 034     989          LXI     H,START-SINCR   (HL) = FIRST RAM - SEARCH INCREMENT
                          990
                          991  *       DETERMINE MEMORY LIMIT.
                          992
 000.107  167             993  INIT1   MOV     M,A             RESTORE VALUE READ
 000.110  031             994          DAD     D               INCREMENT TRIAL ADDRESS
 000.111  176             995          MOV     A,M             (A) = CURRENT MEMORY VALUE
 000.112  065             996          DCR     M               TRY TO CHANGE IT
 000.113  276             997          CMP     M
 000.114  302 107 000      998          JNE     INIT1           IF MEMORY CHANGED
                          999
 000.117  053            1000  INIT2   DCX     H
                         1001
 000.120  371            1002          SPHL                    SET STACKPOINTER = MEMORY LIMIT -1
                         1003
 000.121  345            1004          PUSH    H               SET *PC* VALUE ON STACK
 000.122  041 322 000     1005          LXI     H,ERROR
 000.125  345            1006          PUSH    H               SET 'RETURN ADDRESS'
```

```
                        1009  **      SAVALL - SAVE ALL REGISTERS ON STACK.
                        1010  *
                        1011  *       SAVALL IS CALLED WHEN AN INTERRUPT IS ACCEPTED, IN ORDER TO
                        1012  *       SAVE THE CONTENTS OF THE REGISTERS ON THE STACK.
                        1013  *
                        1014  *       ENTRY   CALLED DIRECTLY FROM INTERRUPT ROUTINE.
                        1015  *       EXIT    ALL REGISTERS PUSHED ON STACK,
                        1016  *               IF NOT YET IN MONITOR MODE, REGPTR = ADDRESS OF REGISTERS
                        1017  *               ON STACK.
                        1018  *               (DE) = ADDRESS OF CTLFLG
                        1019
000.004                 1020          ERRMI   132A-*
000.132                 1021          ORG     132A
000.132  343            1022  SAVALL  XTHL                    SET H,L ON STACK TOP
000.133  325            1023          PUSH    D
000.134  305            1024          PUSH    B
000.135  365            1025          PUSH    PSW
000.136  353            1026          XCHG                    (D,E) = RETURN ADDRESS
000.137  041 012 000     1027          LXI     H,10
000.142  071            1028          DAD     SP              (H,L) = ADDRESS OF USERS SP
                        1029
                        1030  **      REPLACE THESE INSTRUCTIONS WITH A JUMP AROUND THE NMI VECTOR JUMP
                        1031  *
                        1032  *       PUSH    H               SET ON STACK AS 'REGISTER'
                        1033  *       PUSH    D               SET RETURN ADDRESS
                        1034  *       LXI     D,CTLFLG
                        1035  *       LDAX    D               (A) = CTLFLG
                        1036
000.143  303 105 004     1037          JMP     SAVALLX         GO TO SAVALL EXTENSION
                        1038
                        1039  **      ENTRY POINT FOR THE Z80 NMI
                        1040  *
                        1041
000.000                 1042          ERRNZ   *-66H           Z80 NMI ADDRESS
                        1043
000.146  303 116 004     1044  NMIENT  JMP     NMI
                        1045
000.000                 1046          ERRNZ   SAVALLR-151A    DO NOT CHANGE ORGANIZATION
                        1047
000.151                 1048  SAVALLR EQU     *               SAVALL EXTENSION RETURN ADDRESS
                        1049
000.151  057            1050          CMA
000.152  346 060        1051          ANI     CB.MTL+CB.SSI   SAVE REGISTER ADDR IF USER OR SINGLE-STEP
000.154  310            1052          RZ                      RETURN IF WAS INTERRUPT OF MONITOR LOOP
000.155  041 002 000     1053          LXI     H,2
000.160  071            1054          DAD     SP              (H,L) = ADDRESS OF 'STACKPTR' ON STACK
000.161  042 035 040     1055          SHLD    REGPTR
000.164  311            1056          RET
```

```
                        1058  **      CUI - CHECK FOR USER INTERRUPT PROCESSING.
                        1059  *
                        1060  *       CUI IS CALLED TO SEE IF THE USER HAS SPECIFIED PROCESSING
                        1061  *       FOR THE CLOCK INTERRUPT.
                        1062
000.000                 1063          ERRNZ   *-165A
                        1064
040.010                 1065  .       SET     .MFLAG          REFERENCE TO MFLAG
000.165  012            1066  CUI1    LDAX    B               (A) = .MFLAG
000.000                 1067          ERRNZ   UO.CLK-1        CODE ASSUMED = 01
000.166  017            1068          RRC
000.167  334 037 040    1069          CC      UIVEC           IF SPECIFIED, TRANSFER TO USER
                        1070
                        1071  *       RETURN TO PROGRAM FROM INTERRUPT.
                        1072
000.000                 1073          ERRNZ   *-172A
                        1074
000.172  361            1075  INTXIT  POP     PSW             REMOVE FAKE 'STACK REGISTER'
000.173  361            1076          POP     PSW
000.174  301            1077          POP     B
000.175  321            1078          POP     D
000.176  341            1079          POP     H
000.177  373            1080          EI
000.200  311            1081          RET
```

```
                                1084  ***     CLOCK - PROCESS CLOCK INTERRUPT
                                1085  *
                                1086  *       CLOCK IS ENTERED WHENEVER A MILLISECOND CLOCK INTERRUPT IS
                                1087  *       PROCESSED.
                                1088  *
                                1089  *       TICCNT IS INCREMENTED EVERY INTERRUPT.
                                1090
000.000                         1091          ERRNZ   *-201A
                                1092
000.201  052 033 040            1093  CLOCK   LHLD    TICCNT
000.204  043                    1094          INX     H
000.205  042 033 040            1095          SHLD    TICCNT          INCREMENT TICCOUNT
                                1096
000.210  072 011 040            1097          LDA     CTLFLG          CLEAR CLOCK INTERRUPT FLIP-FLOP
000.213  323 360                1098          OUT     OP.CTL
                                1099
                                1100  *       EXIT CLOCK INTERRUPT.
                                1101
000.215  001 011 040            1102          LXI     B,CTLFLG
000.220  012                    1103          LDAX    B               (A) = CTLFLG
000.221  346 040                1104          ANI     CB.MTL
000.223  302 172 000            1105          JNZ     INTXIT          IF IN MONITOR MODE
000.226  013                    1106          DCX     B
000.000                         1107          ERRNZ   CTLFLG-.MFLAG-1
000.227  012                    1108          LDAX    B               (A) = .MFLAG
000.000                         1109          ERRNZ   U0.HLT-200Q     ASSUME HIGH-ORDER
000.230  027                    1110          RAL
000.231  332 270 000            1111          JC      CLK4            SKIP IT
                                1112
                                1113  *       NOT IN MONITOR MODE. CHECK FOR HALT
                                1114
000.234  076 012                1115          MVI     A,10            (A) = INDEX OF *P* REG
000.236  315 052 003            1116          CALL    LRA.            LOCATE REGISTER ADDRESS
000.241  136                    1117          MOV     E,M             *
000.242  043                    1118          INX     H
000.243  126                    1119          MOV     D,M             (D,E) = PC CONTENTS
000.244  033                    1120          DCX     D
000.245  032                    1121          LDAX    D
000.246  376 166                1122          CPI     MI.HLT          CHECK FOR HALT
000.250  302 165 000            1123          JNZ     CUI1
000.253  076 007                1124          MVI     A,A.BEL         DING BELL
000.255  315 302 003            1125          CALL    HCC
000.260  076 110                1126          MVI     A,'H'           "H" FOR HALT
000.262  315 302 003            1127          CALL    HCC
000.265  303 322 000            1128          JMP     ERROR
                                1129
                                1130  ***     JE      ERROR           IF HALT, BE IN MONITOR MODE
                                1131
                                1132  *       NONE OF THE ABOVE, SO ALLOW USER PROCESSING OF CLOCK INTERRUPT
                                1133
000.270                         1134  CLK4    EQU     *
000.270  303 165 000            1135          JMP     CUI1            ALLOW USER PROCESSING OF CLOCK
```

```
                        1138  **      THIS IS ONLY A PORTION OF THE DYNAMIC RAM TEST!!
                        1139  *
                        1140  *       WAIT BEFORE MAKING ANOTHER LOOP
                        1141
000.273  041 000 000   1142  DYMEM6  LXI     H,0
000.276  053           1143  DYMEM7  DCX     H
000.277  174           1144          MOV     A,H
000.300  265           1145          ORA     L
000.301  302 276 000   1146          JNZ     DYMEM7          IF (B,C) NOT ZERO
                        1147
000.304  303 167 007   1148          JMP     DYMEM4          TRY AGAIN BY INCREMENTING ONCE MORE
                        1149
                        1150  **      HAVE A FAILURE PRIOR TO REACHING END OF MEMORY!
                        1151  *
000.307  353           1152  DYMEM9  XCHG
000.310  041 336 014   1153          LXI     H,MSG.ERR       DISPLAY ERROR MESSAGE
                        1154
                        1155  *       LD      IX,DY9.3        RETURN ADDRESS
000.313  335 041        1156          DB      MI.LDXA,MI.LDXB
000.315  160 011        1157          DW      DY9.3
000.317  303 265 007   1158          JMP     DYMSG
```

```
                                1161  ***     ERROR - COMMAND ERROR.
                                1162  *
                                1163  *       ERROR IS CALLED AS A 'BAIL-OUT' ROUTINE.
                                1164  *
                                1165  *       IT RESETS THE OPERATIONAL MODE, AND RESTORES THE STACKPOINTER.
                                1166  *
                                1167  *       ENTRY   NONE
                                1168  *       EXIT    TO MTR LOOP
                                1169  *               CTLFLG SET
                                1170  *               .MFLAG CLEARED
                                1171  *       USES    ALL
                                1172
000.000                         1173            ERRNZ   *-322A
                                1174
000.322                         1175  ERROR     EQU     *
000.322  041 010 040            1176            LXI     H,.MFLAG
000.325  176                    1177            MOV     A,M             (A) = .MFLAG
000.326  346 275                1178            ANI     377Q-UO.DDU-UO.NFR      RE-ENABLE DISPLAYS
000.330  167                    1179            MOV     M,A             REPLACE
000.331  043                    1180            INX     H
000.332  066 360                1181            MVI     M,CB.SSI+CB.MTL+CB.CLI+CB.SPK   RESTORE *CTLFLG*
000.000                         1182            ERRNZ   CTLFLG-.MFLAG-1
000.334  373                    1183            EI
000.335  052 035 040            1184            LHLD    REGPTR
000.340  371                    1185            SPHL                    RESTORE STACK POINTER TO EMPTY STATE
000.341  315 136 002            1186            CALL    ALARM           ALARM FOR 200 MS


                                1188  **      MTR - MONITOR LOOP.
                                1189  *
                                1190
000.000                         1191            ERRNZ   *-344A
                                1192
000.344                         1193  MTR       EQU     *
000.344  373                    1194            EI
                                1195
000.345                         1196  MTR1      EQU     *
000.345  041 345 000            1197            LXI     H,MTR1
000.350  345                    1198            PUSH    H               SET 'MTR1' AS RETURN ADDRESS
000.351  303 135 011            1199            JMP     CKAUTO          CHECK AUTO BOOT, IF NOT CONTROL BACK TO NEXT
000.354  315 100 006            1200  MTR.15    CALL    TYPMSG          PRINT 'H:'
                                1201
000.357  315 262 003            1202  MTR.2     CALL    RCC             READ A CONSOLE CHARACTER
000.362  346 137                1203            ANI     01011111B       MAKE SURE ITS UPPER CASE TO MATCH TABLE
000.364  041 045 017            1204            LXI     H,MTRA          LOOK UP CHARACTER IN *MTRA*
000.367  006 012                1205            MVI     B,MTRAL         (B) = LENGTH OF TABLE
                                1206
000.371  276                    1207  MTR.3     CMP     M               SEE IF CHARACTER FROM CONSOLE = TABLE ENTRY
000.372  043                    1208            INX     H               HL = ADDRESS
000.373  312 014 001            1209            JZ      MTR.4           IF EQUAL
                                1210
000.376  043                    1211            INX     H
000.377  043                    1212            INX     H
001.000  005                    1213            DCR     B               SEE IF PAST END OF TABLE
```