ZENITH | data systems

MONITOR
MTR-90

595-2696-02

ZENITH DATA SYSTEMS
SAINT JOSEPH, MICHIGAN 49085

TABLE OF CONTENTS

Introduction 3

Theory of Operation 4
Power Up and Reset 4
General Operations 4
Clock Interrupts 4

MTR-90 Commands 5
B(oot) 5
B(oot) S(D) 6
C(onvert) 7
G(o) 7
I(n) 7
O(ut) 8
P(rogram Counter) 8
R(adix) 9
S(ubstitute) 9
T(est Memory) 9
V(iew) 10

Program Execution Control 11

Appendix A
MTR-90 Listing 12

Appendix B
Octal Definitions 136

Appendix C
Hexadecimal Definitions 138

Appendix D
SW501 Switch Settings 140

Appendix E
CPU Jumpers 141

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
```

```
001.001  302 371 000  1214          JNZ     MTR.3           IF NOT PAST
                      1215
001.004  076 007      1216          MVI     A,A.BEL         ELSE, DING ERROR
001.006  315 302 003  1217          CALL    WCC
001.011  303 357 000  1218          JMP     MTR.2           TRY AGAIN
                      1219
001.014  315 302 003  1220  MTR.4   CALL    WCC             WRITE CHARACTER BACK TO CONSOLE
001.017  176          1221          MOV     A,M
001.020  043          1222          INX     H               GET MSB
001.021  146          1223          MOV     H,M
001.022  157          1224          MOV     L,A             (H,L) = ROUTINE ADDRESS
001.023  351          1225          PCHL                    GO TO ROUTINE


                      1227  **      GETBND1 - CONTINUATION OF GETBND
                      1228  *
                      1229
001.024  041 003 040  1230  GETBND1 LXI     H,IOWRK+1
001.027  026 015      1231          MVI     D,A.CR
001.031  315 023 015  1232          CALL    IOA
                      1233
001.034  052 002 040  1234          LHLD    IOWRK
001.037  115          1235          MOV     C,L
001.040  104          1236          MOV     B,H
001.041  321          1237          POP     D
001.042  341          1238          POP     H
001.043  311          1239          RET


                      1241  **      VIEW - VIEW MEMORY BLOCKS
                      1242  *
                      1243  *       VIEW START,STOP
                      1244
001.044  041 113 002  1245  VIEW    LXI     H,MSG.VEW
001.047  315 100 006  1246          CALL    TYPMSG
001.052  303 351 007  1247          JMP     VIEW3A                  GET START IN DE, STOP IN BC
                      1248
001.055  042 067 040  1249  VIEW1   SHLD    BLKICW                  SAVE START ADDRESS FOR ASCII STUFF
001.060  303 066 002  1250          JMP     VIEW2
```

```
                        1253  **      SAE - STORE ABUSS AND EXIT.
                        1254  *
                        1255  *       ENTRY   (HL) = ABUSS VALUE
                        1256  *       EXIT    TO (RET)
                        1257  *       USES    NONE
                        1258
000.000                 1259          ERRMI   1063A-*
001.063                 1260          ORG     1063A
                        1261
001.063  042 024 040    1262  SAE     SHLD    ABUSS
001.066  311            1263          RET


                        1265  **      PIN     -       PORT IN
                        1266  *
                        1267  *       PIN  INPUTS A BYTE FROM DISK
                        1268  *
                        1269  *       ENTRY:  NONE
                        1270  *
                        1271  *       EXIT:   (A) = INPUT BYTE FROM 247
                        1272  *
                        1273  *       USE:    AF
                        1274
001.067                 1275  PIN     EQU     *
001.067  315 170 006    1276          CALL    IN.             GET STATUS
001.072  346 240        1277          ANI     S.DTR+S.DON     CHECK FOR DATA TERMINAL REQUEST
001.074  050 371        1278          JR      Z,PIN           IF NOT READY, WAIT
001.076  067            1279          STC
001.077  360            1280          RP                      IF NO S.DTR, MUST BE S.DON
001.100  303 150 006    1281          JMP     IN1.            INPUT A BYTE FROM PORT
```

```
000.000                  1284          ERRMI   1103A-*
001.103                  1285          ORG     1103A
                         1286  **      PCA - PROGRAM COUNTER ALTER
                         1287  *
                         1288  *       PCA INPUTS AND/OR DISPLAYS THE CURRENT USER PROGRAM VALUE AND ALLOWS
                         1289  *       A NEW VALUE TO BE ENTERED OR RETAINS THE CURRENT VALUE IF
                         1290  *       A CR IS TYPED
                         1291  *
                         1292  *       ENTRY   NONE
                         1293  *       EXIT    NONE
                         1294  *       USES    A,D,E,H,L,F
                         1295
                         1296
001.103  041 214 006     1297  PCA     LXI     H,MSG.PC        COMPLETE PC MESSAGE
001.106  315 100 006     1298          CALL    TYPMSG
001.111  076 012         1299          MVI     A,10            GET LOCATION OF USER PC
001.113  315 052 003     1300          CALL    LRA.
001.116  136             1301          MOV     E,M             (D,E) = USER PC VALUE
001.117  043             1302          INX     H
001.120  126             1303          MOV     D,M
001.121  353             1304          XCHG                    (H,L) = USER PC VALUE
                         1305
001.122  315 012 015     1306          CALL    IROC            INPUT NEXT CHARACTER
001.125  332 137 001     1307          JC      PCA1            IF FIRST CHARACTER WAS OCTAL, INPUT NEW PC
                         1308
001.130  315 064 015     1309          CALL    TOA             ELSE, OUTPUT CURRENT VALUE
001.133  315 012 015     1310          CALL    IROC            SEE IF USER WANTS TO CHANGE IT NOW
001.136  320             1311          RNC                     IF NO CHANGE, EXIT
                         1312
                         1313  *       ENTER NEW USER PC VALUE
                         1314
001.137  353             1315  PCA1    XCHG                    (H,L) = ADDRESS OF USER PC VALUE
001.140  026 015         1316          MVI     D,A.CR          END BYTE WITH A RETURN
001.142  315 023 015     1317          CALL    IOA             INPUT NEW ADDRESS
001.145  311             1318          RET                     EXIT
```

```
                         1320  **      GO88 - GO TO USER ROUTINE FROM H88 MONITOR
                         1321  *
                         1322  *       GO88 WAITS FOR A CARRIAGE RETURN OR A NEW ADDRESS TERMINATED WITH
                         1323  *       A CARRIAGE RETURN.  IF NO ADDRESS IS ENTERED, GO88 TRANSFERS
                         1324  *       CONTROL TO THE ADDRESS SPECIFIED BY THE USER PC VALUE
                         1325
                         1326
001.146  041 165 006     1327  GO88    LXI     H,MSG.GO        COMPLETE GO MESSAGE
001.151  315 100 006     1328          CALL    TYPMSG
001.154  315 012 015     1329          CALL    IROC            INPUT A RETURN OR AN OCTAL CHARACTER
001.157  322 177 001     1330          JNC     GO88.1          IF RETURN, GO TO CURRENT USER PC
                         1331
001.162  365             1332          PUSH    PSW             ELSE SAVE OCTAL CHARACTER AND FLAGS
001.163  076 012         1333          MVI     A,10            GET ADDRESS OF USER PC
001.165  315 052 003     1334          CALL    LRA.
001.170  043             1335          INX     H               POINT TO MSB
001.171  361             1336          POP     PSW             GET FIRST CHARACTER BACK
```

```
001.172  026 015       1337           MVI     D,A.CR          END ADDRESS WITH A RETURN
001.174  315 023 015   1338           CALL    IOA             INPUT NEW GO ADDRESS
001.177  315 370 005   1339  G088.1   CALL    WCR.            ECHO RETURN
001.202  303 222 001   1340           JMP     GO              EXECUTE USER ROUTINE


                       1342  **       AUTOBO  - AUTO BOOT
                       1343  *
                       1344  *        ENTRY:  NONE
                       1345  *
                       1346  *        EXIT:   (SEE 'DEVICE' ROUTINE)
                       1347  *
                       1348  *        USE:    ALL
                       1349
001.205  257           1350  AUTOBO   XRA     A                       SET TO PRIMARY FLAG
001.206  315 273 002   1351           CALL    DEVICE                  CHECK DEVICE INFROMATION
001.211  303 342 001   1352           JMP     BOOTO                   GOTO BOOT IT


000.006                1354           ERRMI   1222A-*
001.222                1355           ORG     1222A
                       1356  **       GO - RETURN TO USER MODE
                       1357  *
                       1358  *        ENTRY   NONE
                       1359
000.000                1360           ERRNZ   *-1222A
                       1361
001.222  303 063 000   1362  GO       JMP     GO.


                       1364  **       SSTEP - SINGLE STEP INSTRUCTION.
                       1365  *
                       1366  *        ENTRY   NONE
                       1367
000.000                1368           ERRNZ   *-1225A
                       1369
001.225                1370  SSTEP    EQU     *               SINGLE STEP
001.225  363           1371           DI                      DISABLE INTERRUPTS UNTIL THE RIGHT TIME
001.226  072 011 040   1372           LDA     CTLFLG
001.231  356 020       1373           XRI     C8.SSI          CLEAR SINGLE STEP INHIBIT
001.233  323 360       1374           OUT     OP.CTL          PRIME SINGLE STEP INTERRUPT
001.235  062 011 040   1375  SST1     STA     CTLFLG          SET NEW FLAG VALUES
001.240  341           1376           POP     H               CLEAN STACK
001.241  303 172 000   1377           JMP     INTXIT          RETURN TO USER ROUTINE FOR STEP
```

```
                          1379  **      STPRTN - SINGLE STEP RETURN
                          1380
    000.000               1381          ERRNZ   *-1244A
                          1382
    001.244               1383  STPRTN  EQU     *
    001.244  366 020      1384          ORI     CB.SSI          DISABLE SINGLE STEP INTERRUPTION
    001.246  323 360      1385          OUT     OP.CTL          TURN OFF SINGLE STEP ENABLE
    040.011               1386  .       SET     CTLFLG
    001.250  022          1387          STAX    D
    001.251  346 040      1388          ANI     CB.MTL          SEE IF IN MONITOR MODE
    001.253  302 344 000  1389          JNZ     MTR
    001.256  303 042 040  1390          JMP     UIVEC+3         TRANSFER TO USER'S ROUTINE
```

```
                                1393  **      NBOOT   -       NORMAL BOOT
                                1394  *
                                1395  *       NBOOT IS ENTERED WHEN USER TYPE 'BOOT' COMMAND FROM MONITOR.
                                1396  *       IT WILL ACCEPT THE BOOT DEVICE AS WELL AS THE UNIT NUMBER FROM
                                1397  *       CONSOLE AND GO TO THE BOOT CODE.
                                1398  *
                                1399  *       ENTRY:  NONE
                                1400  *
                                1401  *       EXIT:   (AIO.UNI) = UNIT NUMBER TO BOOT
                                1402  *               (PRIM)  = PORT ADDRESS OF THE BOOT DEVICE
                                1403  *               (TMFG)  = DEVICE TYPE, =1 IS Z47; =0 IS H17
                                1404  *
                                1405  *       USED:   ALL
                                1406
001.261  257                   1407  NBOOT   XRA     A               SET Z FLAG TO PRIMARY DEVICE
001.262  315 273 002            1408  NBOOT0  CALL    DEVICE          READ SWITCH TO DETERMINE BOOT DEVICE
001.265  315 262 003            1409  START1  CALL    RCC             INPUT FROM KB
001.270  376 015                1410          CPI     A.CR            IF INPUT IS CR
001.272  050 043                1411          JR      Z,BOOT0.            THEN TAKE IT AS DRIVE 0
001.274  315 133 016            1412          CALL    BOOT7
001.277  070 007                1413          JR      C,WRONG
001.301  270                    1414          CMP     B
001.302  070 044                1415          JR      C,BOOT5           IF WITHIN THE RANGE, BOOT IT!
001.304  010                    1416          DB      MI.EXAF         SAVE INPUT, CHECK PRIM OR SEC?
001.305  050 010                1417          JR      Z,NB7           IF PRIMARY, CHECK 'S'
001.307  010                    1418          DB      MI.EXAF         RESTORE (Z) FLAG
001.310                         1419  WRONG   EQU     *
001.310  076 007                1420          MVI     A,A.BEL         NOT THE CASES, BEEP!
001.312  315 302 003            1421          CALL    WCC
001.315  030 346                1422          JR      START1          AND TRY AGAIN
                                1423
001.317  010                    1424  NB7     DB      MI.EXAF         RESTORE INPUT & PRIM, SEC FLAG
001.320  346 137                1425          ANI     01011111B       MASK TO UPPER CASE LETTER
001.322  376 123                1426          CPI     'S'             CHECK THE USER LIKE TO BOOT FROM
001.324  040 362                1427          JR      NZ,WRONG        BOOT SECONDARY DEVICE
                                1428
                                1429  *       USER WISHES TO BOOT FROM SECONDARY DEVICE
                                1430
001.326                         1431  BSEC    EQU     *
001.326  041 042 014            1432          LXI     H,BSMSG         PRINT BOOT SECONDARY MESSAGE
001.331  315 100 006            1433          CALL    TYPMSG
001.334  074                    1434          INR     A               SET (Z)=0 FOR SECONDARY DEVICE
001.335  030 323                1435          JR      NBOOT0
                                1436
                                1437  *  SAVE THE AIO.UNI, CHECK IF THERE IS THE BOOT DEVICE AND GO!
                                1438
001.337  315 370 005            1439  BOOT0.  CALL    WCR.            PRINT CR FOR GOOD LOOKS
                                1440
001.342  257                    1441  BOOT0   XRA     A               TAKE CR OR AUTO BOOT AS DRIVE 0
001.343  061 200 042            1442          LXI     SP,42200A       SET STACK FOR NO COMMAND LINE
001.346  030 010                1443          JR      BOOT6
                                1444
001.350  315 302 003            1445  BOOT5   CALL    WCC             PRINT UNIT NUMBER
001.353  326 060                1446          SUI     '0'             MAKE IT BINARY
001.355  303 263 013            1447          JMP     CCL             CHECK FOR COMMAND LINE
001.360  062 061 041            1448  BOOT6   STA     AIO.UNI         STORE THE UNIT #
```

```
001.363  174           1449      MOV     A,H     CHECK IF NO DEVICE AT ADDR. PORT
001.364  247           1450      ANA     A
001.365  373           1451      EI              INSURE INTERRUPTS READY
001.366  312 171 002   1452      JZ      NODEV   NO DEVICE
001.371  351           1453      PCHL            JMP TO THE EXECUTION ROUTINE
```

```
                        1456  **      Z47     -       BOOT FROM Z47 DISK DRIVE
                        1457  *
                        1458  *       Z47 WILL LOAD DATA FROM DISK TRACK 0 SECTOR 0 THRU 9 TO
                        1459  *       USER FIRST AVALIABLE RAM LOCATION. IF THE BOOT IS SUCCED,
                        1460  *       CONTROL PASS TO THAT LOCATION.
                        1461  *
                        1462  *       ENTRY:  (AIO.UNI) = UNIT NUMBER TO BOOT
                        1463  *
                        1464  *       EXIT:   NONE
                        1465  *
                        1466  *       USE:    ALL
                        1467
001.372                 1468  Z47     EQU     *
                        1469  *       LD      (STK),SP                SAVE STACK POINTER FOR RE-BOOT
001.372  355 163        1470          DB      355Q,163Q
001.374  124 041        1471          DW      STK
                        1472
001.376                 1473  Z47A    EQU     *
001.376  373            1474          EI                              LET THE TIMER FLY
001.377  072 061 041    1475          LDA     AIO.UNI                 GET UNIT NUMBER
002.002  007            1476          RLC                             SET TO SIDE/UNIT/SECTOR FORMAT
002.003  007            1477          RLC
002.004  007            1478          RLC
002.005  007            1479          RLC
002.006  007            1480          RLC
002.007  074            1481          INR     A                       SET TO SECTOR 1
002.010  117            1482          MOV     C,A                     SAVE SIDE/UNIT/SECTOR (SIDE=0)
                        1483
000.001                 1484  RESET   IF      .DEBUG
                        1485          XRA     A
                        1486          STA     DBFLG
                        1487          ENDIF
                        1488
002.011  076 002        1489          MVI     A,W.RES
002.013  315 000 010    1490          CALL    Z47X                    DO Z47 EXTENSION
                        1491
                        1492  *       READ BOOT CODE FROM Z47
                        1493
002.016  041 200 042    1494          LXI     H,USERFWA               BOOT DESTINATION
                        1495
000.001                 1496          IF      .DEBUG
                        1497          MVI     A,10
                        1498          STA     DBFLG                   MEMORY LOCATION FOR DEBUGGING
                        1499          ENDIF
                        1500
002.021  315 111 006    1501          CALL    RDBLCK                  READ A SECTOR FROM DISK
                        1502
000.001                 1503          IF      .DEBUG
                        1504          PUSH    PSW
                        1505          MVI     A,11
                        1506          STA     DBFLG
                        1507          POP     PSW
                        1508          ENDIF
                        1509
002.024  332 171 002    1510          JC      NODEV                   IF READ ERROR
                        1511
```

```
002.027  052 124 041  1512          LHLD    STK
002.032  371          1513          SPHL                    RESTORE STACK
                      1514
002.033  303 201 016  1515          JMP     EUC             SET CLOCK AND ENTER USER CODE



                      1517  **      RETRY   -       RE-BOOT Z47
                      1518  *
                      1519  *       RETRY IS ENTERED WHEN 3.5 SECONDS TIME OUT & BOOT Z47
                      1520  *       STILL NOT SUCCEED. IT RESTORE STACK & JUMP TO BOOT Z47 ROUTINE
                      1521  *
                      1522  *       ENTRY:  NONE
                      1523  *
                      1524  *       EXIT:   (HL) = (SP)
                      1525  *
                      1526  *       USE:    HL, SP
                      1527
002.036  052 124 041  1528  RETRY   LHLD    STK             GET OLD STACK ADDRESS
002.041  371          1529          SPHL                    SET TO STACK POINTER
002.042  030 332      1530          JR      Z47A            RE-BOOT
```

```
                         1533  **       R.SDP   - SET DEVICE PARAMETER, ALLOW TO SET DRIVE 0, 1, AND 2.
                         1534  *        (MORE INFORMATION CAN BE FOUND IN H17 ROM CODE 36062A)
                         1535
002.044                  1536  R.SDP    EQU     *
002.044  076 012         1537           MVI     A,ERPTCNT
002.046  062 264 040     1538           STA     D.OECNT         SET MAX ERROR COUNT FOR OPERATION
002.051  072 061 041     1539           LDA     AIO.UNI         LOAD DRIVE NUMBER
002.054  365             1540           PUSH    PSW             SAVE IT
002.055  376 002         1541           CPI     2               IS IT DRIVE 2?
002.057  070 002         1542           JR      C,R.SDP1        IF NOT JMP TO H17 ROM ROUTINE
002.061  076 003         1543           MVI     A,3
002.063  303 073 036     1544  R.SDP1   JMP     SDP3


                         1546  **       VIEW2 - CONTINUE *VIEW* COMMAND
                         1547  *
                         1548
002.066  176             1549  VIEW2    MOV     A,M             A = BYTE
002.067  315 077 015     1550           CALL    TOB
002.072  076 040         1551           MVI     A,' '           SPACE BETWEEN
002.074  315 302 003     1552           CALL    WCC
002.077  315 363 007     1553           CALL    VIEW4           CHECK FOR END
002.102  312 355 003     1554           JZ      VIEW9           IF ALL DONE
002.105  315 340 003     1555           CALL    VIEW3.          CHECK FOR END OF LINE
002.110  303 000 006     1556           JMP     VIEW3
                         1557
002.113  151 145 167     1558  MSG.VEW  DB      'iew ',0
```

```
000.016                 1561            ERRMI   2136A-*
002.136                 1562            ORG     2136A
                        1563    **      HORN - MAKE NOISE.
                        1564    *
                        1565    *       ENTRY   (A) = (MILLISECOND COUNT)/2
                        1566    *       EXIT    NONE
                        1567    *       USES    A,F
                        1568
000.000                 1569            ERRNZ   *-2136A
                        1570
002.136                 1571    ALARM   EQU     *
002.136  030 026        1572            JR      ALARMB          BRANCH TO A JUMP TO NOISE TO DING BELL
                        1573
000.000                 1574            ERRNZ   *-2140A
                        1575
002.140  365            1576    HORN    PUSH    PSW
002.141  076 200        1577            MVI     A,CB.SPK        TURN ON SPEAKER
                        1578
002.143  343            1579    HRN0    XTHL                    SAVE (HL), (H)  = COUNT
002.144  325            1580            PUSH    D               SAVE (DE)
002.145  353            1581            XCHG                    (D) = LOOP COUNT
002.146  041 011 040     1582            LXI     H,CTLFLG
002.151  256            1583            XRA     M
002.152  136            1584            MOV     E,M             (E) = OLD CTLFLG VALUE
002.153  167            1585            MOV     M,A             TURN ON HORN
002.154  056 033        1586            MVI     L,#TICCNT
                        1587
002.156  172            1588            MOV     A,D             (A) = CYCLE COUNT
002.157  206            1589            ADD     M
002.160  276            1590    HRN2    CMP     M               WAIT REQUIRED TICCOUNTS
002.161  040 375        1591            JR      NZ,HRN2
                        1592
002.163  303 045 006    1593            JMP     HRNX            JUMP TO AN EXTENSION OF HORN SO ROOM
                        1594    *                                       CAN BE MADE FOR A JUMP TO NIOSE
                        1595
                        1596
002.166  303 053 006    1597    ALARMB  JMP     NOISE           SEND A BELL TO THE CONSOLE
```

```
                                1600  **      NODEV   - NO DEVICE AT THE UNIT USER INDICATE
                                1601  *
                                1602  *       NODEV IS ENTERED WHEN:  1. 15 SECONDS TIME OUT
                                1603  *                               OR 2. NO DEVICE IS INDICATED ON SWITCH
                                1604  *                               OR 3. USER HIT <DELETE> TO ABORT BOOT
                                1605  *                               OR 4. BOOT ERROR
                                1606  *       IT WILL EXIT TO 'ERROR' ROUTINE AND MONITOR LOOP
                                1607  *
                                1608  *       ENTRY:  NONE
                                1609  *
                                1610  *       EXIT:   (A) = 0
                                1611  *
                                1612  *       USE:    AF, HL
                                1613
002.171                         1614  NODEV   EQU     *
002.171  041 046 014            1615          LXI     H,ERRMSG        PRINT ERROR MESSAGE
002.174  315 100 006            1616          CALL    TYPMSG
002.177  062 010 040            1617  NODEV1  STA     .MFLAG          STOP TIMER
002.202  323 177                1618          OUT     DP.DC           OFF DISK
002.204  303 322 000            1619          JMP     ERROR           BACK TO MONITOR LOOP
```

```
                                1622  **      H17     -   BOOT FROM H17 DISK SYSTEM
                                1623  *                   (THIS IS THE MODIFICATION OF THE H17 BOOT ROUTINE,
                                1624  *                    MORE INFORMATION CAN BE FOUND ON H17 BOOT ROM 30000A)
                                1625  *
                                1626  *       ENTRY:  (AIO.UNI) = THE UNIT TO BOOT
                                1627  *
                                1628  *       EXIT:   NONE
                                1629  *
                                1630  *       USE:    ALL
                                1631
002.207                         1632  H17     EQU     *
002.207  001 130 000           1633          LXI     B,BOOTAL        SET THE COUNT TO MOVE IN CONSTANTS AND VECTORS
002.212  021 132 037           1634          LXI     D,BOOTA         SET THE SOURCE ADDRESS
002.215  041 110 040           1635          LXI     H,D.CON         SET THE DESTINATION ADDRESS
002.220  315 252 030           1636          CALL    $MOVE           MOVE IT
                                1637
                                1638  **      SET ADDRESS FOR 'SET DEVICE PARAMETER' ROUTINE
                                1639  *       TO HANDLE DISK DRIVE 0, 1, AND 2.
                                1640
002.223  041 044 002           1641          LXI     H,R.SDP         SET THIS ROM ROUTINE ADDRESS
002.226  042 206 040           1642          SHLD    D.SDP           SET INTO RAM JUMP VECTOR
002.231  373                   1643          EI                      RESTORE INTERRUPT
                                1644
                                1645  *       WAIT TILL USER INSERT THE DISK AND CLOSE THE DOOR
                                1646  *       (TIMER INTERRUPT IS AFFECTED NOW)
                                1647
002.232  006 012               1648          MVI     B,10            LOOK FOR SOME HOLE AND NO HOLE
002.234  315 044 002           1649          CALL    R.SDP           SELECT UNIT & MOTOR ON
002.237  315 271 036           1650  H17A    CALL    WNH             WAIT FOR NO HOLE
002.242  315 235 036           1651          CALL    WHD             WAIT FOR HOLE
002.245  020 370               1652          DJNZ    H17A
                                1653
                                1654  *       READ BOOT CODE
                                1655
002.247  315 024 017           1656          CALL    H17X            H17 Extension Routine
002.252  021 200 042           1657          LXI     D,USERFWA       SET THE LOAD LOCATION
002.255  001 000 011           1658          LXI     B,9*256         LOAD 9 SECTORS
002.260  041 000 000           1659          LXI     H,0             LOAD FROM TRACK 0 SECTOR 1
002.263  315 077 034           1660          CALL    R.READ          READ DISK BOOT CODE
002.266  070 301               1661          JR      C,NODEV           ERROR ON BOOT, BACK TO "H:"
002.270  303 215 016           1662          JMP     EUC.            VECTORS ALREADY IN
```

```
                        1665  ***     DEVICE  -       DETERMINE BOOT WHICH DEVICE AT WHICH PORT
                        1666  *
                        1667  *       ENTRY:  Z FLAG  ( Z=1 FOR PRIMARY, Z=0 FOR SECONDARY)
                        1668  *
                        1669  *       EXIT:   HL = DEVICE BOOT EXECUTION ADDRESS
                        1670  *               REG B = PRIMARY MAXI. DRIVE NUMBER
                        1671  *                    IF Z47 ='4';H17 ='3';H37 ='4';H67 ='4'
                        1672  *               (PRIM) = PRIMARY DEVICE PORT ADDRESS
                        1673  *               (TMFG) = SET UP FROM TABLE
                        1674  *
                        1675  *       USE:    ALL
                        1676
002.273                 1677  DEVICE  EQU     *
002.273  010           1678          DB      MI.EXAF         SAVE Z FLAG
                        1679
                        1680  *       INITIAL VARIABLES
                        1681
002.274  363           1682          DI                      NO INTERRUPT
002.275  041 240 040   1683          LXI     H,D.RAM         CLEAR H17 WORK RAM AREA
002.300  006 037       1684          MVI     B,D.RAML        LENGTH TO CLEAR
002.302  315 212 031   1685          CALL    $ZERO
002.305  323 177       1686          OUT     DP.DC           OFF DISK
002.307  062 033 040   1687          STA     TICCNT          0 TIMER COUNTER
002.312  062 122 041   1688          STA     MYCNT           0.5 SECOND TIMER = 0
                        1689
002.315  074           1690          INR     A               (A)=1
000.000                 1691          ERRNZ   U0.CLK-1       TIMER INTERRPT MUST = 1
002.316  062 010 040   1692          STA     .MFLAG          ALLOW TIMER INTERRUPT
002.321  041 037 040   1693          LXI     H,UIVEC         SET ALL VECTOR TO EI/RET PROCESS
002.324  066 303       1694  BOOT2   MVI     M,MI.JMP
002.326  043           1695          INX     H
002.327  066 027       1696          MVI     M,#EIXIT        STORE LS BYTE
002.331  043           1697          INX     H
002.332  066 034       1698          MVI     M,EIXIT/256     STORE MS BYTE
002.334  043           1699          INX     H
002.335  207           1700          ADD     A
002.336  362 324 002   1701          JP      BOOT2
                        1702
002.341  041 304 004   1703          LXI     H,TMOUT         SET TIMER INTERRUPT VECTOR
002.344  042 040 040   1704          SHLD    UIVEC+1
                        1705
                        1706  *       DETERMINE BOOT DEVICE AND ITS INFORMATION
                        1707
002.347  333 362       1708          IN      H88.SW          READ SWITCH DATA
002.351  346 020       1709          ANI     H88S.DV         DETERMINE WHICH TABLE IS PRIMARY
002.353  050 012       1710          JR      Z,DEV174          IF PORT 174 IS PRIMARY
                        1711
                        1712  *       PRIMARY DEVICE IS AT 170q
                        1713
002.355  333 362       1714  DEV170  IN      H88.SW          GET DEVICE SWITCHES
                        1715
002.357  010           1716          DB      MI.EXAF         GET 'SD' FLAG
002.360  041 103 017   1717          LXI     H,BT174         ASSUME PORT 174
002.363  040 021       1718          JR      NZ,DEV2           IF WAS 174
                        1719
002.365  030 010       1720          JR      DEV1.           DO PORT 170 STUFF
```

```
                          1721
                          1722  *       DEVICE IS AT 174q
                          1723
002.367  333 362         1724  DEV174  IN      H88.SW          GET DEVICE DIPS
                          1725
002.371  010             1726          DB      MI.EXAF         SAVE DIPS, RESTORE 'SD' FLAG
002.372  041 103 017      1727          LXI     H,BT174         ASSUME PRIMARY
002.375  050 007         1728          JR      Z,DEV2
                          1729
002.377  010             1730  DEV1.   DB      MI.EXAF         GET SWITCHES BACK
003.000  017             1731          RRC
003.001  017             1732          RRC                     MOVE BITS DOWN
003.002  010             1733          DB      MI.EXAF         AND SAVE AGAIN
003.003  041 124 017      1734          LXI     H,BT170         WAS PORT 170
                          1735  *       JR      DEV2
000.000                   1736          ERRNZ   *-DEV2
                          1737
                          1738  *       HL = ADDRESS OF FWA OF PROPER TABLE
                          1739
003.006  176             1740  DEV2    MOV     A,M             FIRST BYTE IS PORT NUMBER
003.007  062 120 041      1741          STA     PRIM            (A) = DEVICE ADDRESS
                          1742
003.012  010             1743          DB      MI.EXAF         (A) = DEVICE SPECIFIC FLAG
003.013  346 003         1744          ANI     H88S.4          MASK OFF UNIT BITS
003.015  207             1745          ADD     A
003.016  207             1746          ADD     A               4 BYTE ENTRIES
003.017  043             1747          INX     H               HL = FWA OF TABLE ENTRIES
003.020  137             1748          MOV     E,A
003.021  026 000         1749          MVI     D,0             DE = OFFSET
003.023  031             1750          DAD     D               HL = ADDRES OF DEVICE ENTRY
                          1751
003.024  176             1752          MOV     A,M
003.025  062 121 041      1753          STA     TMFG            1ST ENTRY IS TIME-OUT FLAG
                          1754
003.030  043             1755          INX     H
003.031  106             1756          MOV     B,M             2ND ENTRY IS UNIT NUMBER
                          1757
003.032  043             1758          INX     H
003.033  136             1759          MOV     E,M
003.034  043             1760          INX     H               3RD ENTRY IS BOOT ROUTINE ADDRESS
003.035  126             1761          MOV     D,M
003.036  353             1762          XCHG                    MOVE IT INTO HL
003.037  311             1763          RET
```

```
000.007                    1766          ERRMI   3047A-*
003.047                    1767          ORG     3047A
                           1768  **      LRA - LOCATE REGISTER ADDRESS.
                           1769  *
                           1770  *       ENTRY   NONE.
                           1771  *       EXIT    (A) = REGISTER INDEX
                           1772  *               (H,L) = STORAGE ADDRESS
                           1773  *               (D,E) = (0,A)
                           1774  *       USES    A,D,E,H,L,F
                           1775
                           1776
000.000                    1777          ERRNZ   *-3047A
                           1778
003.047  072 005 040      1779  LRA     LDA     REGI
003.052  137              1780  LRA.    MOV     E,A
003.053  026 000          1781          MVI     D,0
003.055  052 035 040      1782          LHLD    REGPTR
003.060  031              1783          DAD     D               (DE) = (REGPTR)+(REGI)
003.061  311              1784          RET


                           1786  **      IOA - INPUT OCTAL ADDRESS.
                           1787  *
                           1788  *       ENTRY   (H,L) = ADDRESS OF RECEPTION DOUBLE BYTE.
                           1789  *               (D) = TERMINATING CHARACTER
                           1790  *       EXIT    NONE
                           1791  *       USES    A,D,E,H,L,F
                           1792
                           1793
000.000                    1794          ERRNZ   *-3062A
                           1795
003.062  303 166 005      1796  IOA0    JMP     IOA1
003.065  000              1797          NOP                     RETAIN H8 ORG


                           1799  **      IO8 - INPUT OCTAL BYTE.
                           1800  *
                           1801  *       READ ONE OCTAL BYTE FROM THE KEYSET.
                           1802  *
                           1803  *       ENTRY   (H,L) = ADDRESS OF BYTE TO HOLD VALUE
                           1804  *               'C' SET IF FIRST DIGIT IN (A)
                           1805  *       EXIT    NONE
                           1806  *       USES    A,D,E,H,L,F
                           1807
                           1808
000.000                    1809          ERRNZ   *-3066A
                           1810
003.066  066 000          1811  IO80    MVI     M,0             ZERO OUT OLD VALUE
003.070  324 262 003      1812  IO81    CNC     RCC             READ CONSOLE CHARACTER
                           1813
                           1814  *       SEE IF CHARACTER IS A VALID OCTAL VALUE
                           1815  *
```

```
003.073  376 060        1816          CPI     '0'             LESS THAN ZERO?
003.075  332 135 003    1817          JC      IOB2            IF (A) < 0, SEE IF A TERMINATING CHARACTER
003.100  376 070        1818          CPI     '8'             GREATER THAN 7?
003.102  322 070 003    1819          JNC     IOB1            IF TOO LARGE, TRY AGAIN
                        1820
                        1821  *       HAVE AN OCTAL DIGIT
                        1822  *
003.105  315 302 003    1823          CALL    HCC             ECHO CHARACTER
003.110  346 007        1824          ANI     00000111B       MASK FOR BINARY VALUE
003.112  137            1825          MOV     E,A             (E) = VALUE
003.113  176            1826          MOV     A,M             GET OLD VALUE
003.114  007            1827          RLC                     SHIFT 3
003.115  007            1828          RLC
003.116  007            1829          RLC
003.117  303 126 003    1830          JMP     IOB1.5          JUMP AROUND AN H88/H89 TO H8 FAKE ROUTINE
                        1831
                        1832  **      FAKE OUT ROUTINE FOR CALLERS OF *DOD* FROM THE H8 FRONT PANEL
                        1833
                        1834
000.000                 1835          ERRNZ   *-3122A
                        1836
003.122  043            1837  DOD     INX     H
003.123  043            1838          INX     H
003.124  043            1839          INX     H
003.125  311            1840          RET
                        1841
                        1842
                        1843  *       CONTINUE
                        1844
003.126  346 370        1845  IOB1.5  ANI     11111000B       TOSS OLD LSB DIGIT
003.130  263            1846          ORA     E               REPLACE WITH NEW VALUE
003.131  167            1847          MOV     M,A
003.132  303 070 003    1848          JMP     IOB1            INPUT ANOTHER CHARACTER
                        1849
                        1850  *       CHECK FOR A CARRIAGE RETURN TO TERMINATE BYTE
                        1851  *
003.135  376 015        1852  IOB2    CPI     A.CR            CARRIAGE RETURN?
003.137  310            1853          RZ                      RETURN IF CARRIAGE RETURN       /JWT 790507/
003.140  257            1854          XRA     A               CLEAR CARRY                     /JWT 790507/
003.141  030 325        1855          JR      IOB1            GET A NEW CHARACTER             /JWT 790507/
```

```
                        1858  **      DYASC - DYNAMIC RAM ASCII OUTPUT TO CONSOLE
                        1859  *
                        1860  *       ENTRY   (A) = CHARACTER TO OUTPUT
                        1861  *               (IY) = RETURN ADDRESS
                        1862  *       EXIT    TO (IY)
                        1863  *       USES    A,C,F
                        1864
003.143                 1865  DYASC   EQU     *
                        1866  *       EX      AF,AF'
003.143  010            1867          DB      MI.EXAF
003.144  333 355        1868  DYASC1  IN      SC.ACE+UR.LSR   TERMINAL READY?
003.146  346 040        1869          ANI     UC.THE
003.150  312 144 003    1870          JZ      DYASC1          NOT YET.
                        1871
                        1872  *       EX      AF,AF'
003.153  010            1873          DB      MI.EXAF
003.154  323 350        1874          OUT     SC.ACE+UR.THR
                        1875  *       JP      (IY)            RETURN TO CALLER
003.156  375 351        1876          DB      MI.JIYA,MI.JIYB


                        1878  **      DYBYT - DYNAMIC RAM BYTE OUTPUT
                        1879  *
                        1880  *       ENTRY   (A) = BYTE TO OUTPUT AS OCTAL
                        1881  *               (IX) = RETURN ADDRESS
                        1882  *       EXIT    TO (IX)
                        1883  *       USES    A,C,IY,F
                        1884
003.160  303 240 016    1885  DYBYT   JMP     DYBYTX
003.163  366 060        1886  DYBYT.1 ORI     '0'                     MAKE ASCII
                        1887
                        1888  *       LD      IY,DYBYT.2
003.165  375 041        1889          DB      MI.LDYA,MI.LDYB
003.167  174 003        1890          DW      DYBYT.2
                        1891
003.171  303 143 003    1892          JMP     DYASC
                        1893
003.174  171            1894  DYBYT.2 MOV     A,C
003.175  346 070        1895          ANI     00111000B
003.177  017            1896          RRC
003.200  017            1897          RRC
003.201  017            1898          RRC
003.202  366 060        1899          ORI     '0'
                        1900
                        1901  *       LD      IY,DYBYT.4
003.204  375 041        1902          DB      MI.LDYA,MI.LDYB
003.206  213 003        1903          DW      DYBYT.4
                        1904
003.210  303 143 003    1905          JMP     DYASC
                        1906
003.213  171            1907  DYBYT.4 MOV     A,C
003.214  346 007        1908          ANI     00000111B
003.216  366 060        1909          ORI     '0'
                        1910
                        1911  *       LD      IY,DYBYT.6
```

```
003.220  375 041        1912            DB      MI.LDYA,MI.LDYB
003.222  227 003        1913            DW      DYBYT.6
                        1914
003.224  303 143 003    1915            JMP     DYASC
                        1916
003.227                 1917    DYBYT.6 EQU     *
                        1918    *       JP      (IX)
003.227  335 351        1919            DB      MI.JIXA,MI.JIXB


                        1921    **      MSG.PAS - PASS MESSAGE FOR DYNAMIC RAM TEST
                        1922    *
                        1923
003.231  015 012 012    1924    MSG.PAS DB      A.CR,A.LF,A.LF
003.234  011 040 120    1925            DB      11Q,' Pass =',11Q,'    '
003.250  000           1926            DB      0
```

```
000.007                 1929            ERRMI   3260A-*
003.260                 1930            ORG     3260A
                        1931  **        RCK - READ CONSOLE KEYPAD
                        1932  *
                        1933  *         RCK IS CALLED TO READ A KEYSTROKE FROM THE CONSOLE FRONT PANEL KEYPAD.
                        1934  *         SINCE THE H88/89 DOES NOT HAVE A FRONT PANEL, THIS ROUTINE IS PROVIDED
                        1935  *         ONLY TO MAINTAIN COMPATIBILITY WITH PAM-8.
                        1936  *         RCK WILL IMMEDIATELY RETURN WITH A VALUE OF 0 (ZERO) IN THE ACCUMULATOR.
                        1937  *
                        1938  *         ENTRY   NONE
                        1939  *         EXIT    (A) = 0
                        1940  *         USES    A,F
                        1941  *
                        1942  *         RCK MUST HAVE SAME ENTRY AS RCK IN PAM-8
000.000                 1943            ERRNZ   *-3260A
                        1944
003.260                 1945  RCK       EQU     *
                        1946
003.260  257            1947            XRA     A
003.261  311            1948            RET
                        1949
```

```
                                1953  **      RCC - READ CONSOLE CHARACTER.
                                1954  *
                                1955  *       RCC IS CALLED TO READ A KEYSTROKE FROM THE CONSOLE.
                                1956  *       IF A RUBOUT/DELETE IS RECEIVED, EXIT IS TO *ERROR*.
                                1957  *
                                1958  *       ENTRY   NONE
                                1959  *       EXIT    TO ERROR  - IF A DELETE OR RUBOUT IS ENCOUNTERED
                                1960  *               TO CALLER - WHEN A KEY IS HIT
                                1961  *                           (A) = ASCII KEY VALUE
                                1962  *       USES    A,F
                                1963
                                1964
                                1965
003.262                         1966  RCC     EQU     *
                                1967
003.262  333 355                1968  RCC1    IN      SC.ACE+UR.LSR   INPUT ACE LINE STATUS REGISTER
003.264  346 001                1969          ANI     UC.DR           SEE IF THERE IS A DATA READY
003.266  050 372                1970          JR      Z,RCC1
                                1971
003.270  333 350                1972  RCC2    IN      SC.ACE+UR.RBR   ELSE, INPUT CHARACTER
003.272  346 177                1973          ANI     01111111B       TOSS ANY PARITY
003.274  376 177                1974          CPI     A.DEL
003.276  312 322 000            1975          JZ      ERROR           IF RUBOUT, EXIT TO ERROR
                                1976
003.301  311                   1977          RET                     ELSE, EXIT TO CALLER



                                1979  **      WCC - WRITE CONSOLE CHARACTER
                                1980  *
                                1981  *       WRITE A CHARACTER TO THE CONSOLE UART PORT
                                1982  *
                                1983  *       ENTRY   (A) = ASCII CHARACTER TO OUTPUT
                                1984  *       EXIT    NONE
                                1985  *       USES    NONE
                                1986
                                1987
003.302  365                   1988  WCC     PUSH    PSW             SAVE CHARACTER
003.303  333 355                1989  WCC1    IN      SC.ACE+UR.LSR   INPUT ACE STATUS
003.305  346 040                1990          ANI     UC.THE          SEE IF TRANSMITTER HOLDING REGISTER IS EMPTY
003.307  050 372                1991          JR      Z,WCC1
                                1992
003.311  361                   1993          POP     PSW             GET CHARACTER
003.312  323 350                1994          OUT     SC.ACE+UR.THR           OUTPUT TO CONSOLE
003.314  311                   1995          RET
```

```
                         1997  **       THE FOLLOWING IS ONLY A PORTION OF THE DYNAMIC RAM TEST!!!
                         1998  *
003.315  353            1999  DY9.5    XCHG                    SAVE ERROR ADDRESS
003.316  041 341 007     2000           LXI     H,MSG.EQ        OUTPUT " = "
                         2001
                         2002  *        LD      IX,DY9.8        RETURN ADDRESS
003.321  335 041         2003           DB      MI.LDXA,MI.LDX8
003.323  330 003         2004           DW      DY9.8
                         2005
003.325  303 265 007     2006           JMP     DYMSG           OUTPUT STRING
                         2007
003.330  032            2008  DY9.8    LDAX    D               OUTPUT RAM CONTENTS
                         2009
                         2010  *        LD      IX,DYMEM10      RETURN ADDRESS
003.331  335 041         2011           DB      MI.LDXA,MI.LDX8
003.333  252 013         2012           DW      DYMEM10
                         2013
003.335  303 160 003     2014           JMP     DYBYT


                         2016  **       VIEW3. - CONTINUATION OF *VIEW*
                         2017  *
                         2018  *        SEE IF END OF BYTES
                         2019  *
                         2020
003.340  043            2021  VIEW3.   INX     H               BUMP POINTER
003.341  315 112 015     2022           CALL    CHKRAD          GET RADIX
003.344  076 360         2023           MVI     A,11110000B     ASSUME HEX
003.346  040 002         2024           JR      NZ,VIEW3.A      IF WAS HEX
003.350  076 370         2025           MVI     A,11111000B
003.352  245            2026  VIEW3.A  ANA     L               (A) = MASKED ADDR LSB
003.353  275            2027           CMP     L               SAME?
003.354  311            2028           RET                     LET CALLER DECIDE


                         2030  **       VIEW9 - DO THE ASCII
                         2031  *
                         2032
003.355  052 067 040     2033  VIEW9    LHLD    BLKICH          RESTORE REGISTERS
003.360  303 171 010     2034           JMP     VIEW5
```

```
                        2037  **      IO ROUTINES TO BE COPIED INTO AND USED IN RAM.
                        2038  *
                        2039  *       MUST CONTINUE TO 3777A FOR PROPER COPY.
                        2040  *       THE TABLE MUST ALSO BE BACKWARDS TO THE FINAL RAM
                        2041
000.006                 2042          ERRMI   4000A-7-*
003.371                 2043          ORG     4000A-7
                        2044
003.371                 2045  PRSROM  EQU     *
003.371  001            2046          DB      1               REFIND
003.372  000            2047          DB      0               CTLFLG
003.373  000            2048          DB      0               .MFLAG
003.374  000            2049          DB      0               DSPMOD
003.375  000            2050          DB      0               DSPROT
003.376  012            2051          DB      10              REGI
003.377  311            2052          DB      MI.RET
                        2053
000.000                 2054          ERRNZ *-4000A
                        2055
```

```
                        2058  ***     INITOX          EXTENSION OF INITO TO SUPPORT H88
                        2059
004.000  076 002       2060  INITOX  MVI     A,H88B.CK               ENABLE CLOCK
004.002  323 362       2061          OUT     H88.CTL
                        2062
                        2063  *       SET UP ACE FOR CONSOLE COMMUNICATIONS
                        2064  *
004.004  076 200       2065          MVI     A,UC.DLA                SET DIVISOR LATCH ACCESS BIT
004.006  323 353       2066          OUT     SC.ACE+UR.LCR
004.010  041 101 004   2067          LXI     H,BRTAB                 (H,L) = BEGINNING OF BAUD RATE TABLE
004.013  333 362       2068          IN      H88.SW                  INPUT SWITCHES FOR DESIRED BAUD RATE
004.015  346 100       2069          ANI     H88S.BR                 MASK FOR BAUD RATE SWITCHES ONLY
004.017  017           2070          RRC                             SHIFT FOR A *2 FOR TABLE
004.020  017           2071          RRC
004.021  017           2072          RRC
004.022  017           2073          RRC
004.023  017           2074          RRC
004.024  205           2075          ADD     L                       ADD DISPLACEMENT FROM BEGINNING OF TABLE
004.025  157           2076          MOV     L,A
004.026  176           2077          MOV     A,M                     GET MSB OF DIVISOR
004.027  323 351       2078          OUT     SC.ACE+UR.DLM
004.031  043           2079          INX     H                       GET LSB
004.032  176           2080          MOV     A,M
004.033  323 350       2081          OUT     SC.ACE+UR.DLL
004.035  076 003       2082          MVI     A,UC.8BW                SET 8 BITS, 1 STOP BIT, NO PARITY
004.037  323 353       2083          OUT     SC.ACE+UR.LCR
004.041  076 000       2084          MVI     A,0                     SET NO INTERRUPTS
004.043  323 351       2085          OUT     SC.ACE+UR.IER
                        2086
                        2087  *       WAIT A WHILE TO ALLOW THE CONSOLE RESET TO FINISH SO IT CAN
                        2088  *       ACCEPT THE FIRST PROMPT
                        2089  *
004.045  303 113 016   2090          JMP     INTOXO                  DO OTHER STUFF FIRST
004.050  015           2091  INITOX1 DCR     C
004.051  040 375       2092          JR      NZ,INITOX1
                        2093
004.053  020 373       2094          DJNZ    INITOX1
                        2095
                        2096  *       INPUT SWITCH TO SEE IF TO BEGIN OPERATION OR MEMORY TEST
                        2097  *
004.055  333 362       2098          IN      H88.SW                  GET SWITCHES
004.057  346 040       2099          ANI     H88S.M                  MASK FOR MEMORY TEST ONLY
004.061  312 032 016   2100          JZ      MEMORY.                 IF TO PERFORM MEMORY TESTS
                        2101
                        2102  *       REPLACE WHAT WAS ORIGINALLY AT THE JUMP WHICH GOT US HERE
                        2103  *
004.064  021 371 003   2104          LXI     D,PRSROM                (DE) = ROM COPY OF PRS CODE
004.067  257           2105          XRA     A
004.070  062 123 041   2106          STA     AUTOB                   INITIAL AUTO BOOT FLAG
004.073  062 066 040   2107          STA     DATA                    INITIAL 362Q PORT DATA SAVE BYTE
004.076  303 003 000   2108          JMP     INITO.0                 RETURN TO ORIGINAL CODE
```

```
                        2110  **      BRTAB - BAUD RATE DIVISOR TABLE
                        2111  *
004.101                 2112  BRTAB   EQU     *
                        2113
004.101  000 014       2114  BR96    DB      0,12            9600 BAUD
004.103  000 006       2115  BR19.2  DB      0,6           19,200 BAUD
                        2116  *BR38.4 DB       0,3          38,400 BAUD
                        2117  *BR56.0 DB       0,2          56,000 BAUD
                        2118
000.004                 2119  .       SET     */256
000.000                 2120          ERRNZ   BRTAB/256-.   TABLE MUST BE IN ONE PAGE


                        2122  ***     SAVALLX - SAVALL EXTENSION TO MAKE ROOM FOR A JUMP TO THE NMI HANDLER
                        2123
004.105                 2124  SAVALLX EQU     *             REPLACE OLD CODE
004.105  345           2125          PUSH    H             SET ON STACK AS 'REGISTER'
004.106  325           2126          PUSH    D             SET RETURN ADDRESS
004.107  021 011 040   2127          LXI     D,CTLFLG
004.112  032           2128          LDAX    D
004.113  303 151 000   2129          JMP     SAVALLR       RETURN TO OLD CODE
```

```
                        2132  ****    NMI - NON MASKABLE INTERRUPT
                        2133  *
                        2134  *       NMI IS USED AS THE TRAP FOR ALL ILLEGAL PORT REQUESTS
                        2135  *
                        2136  *       PORT ADDRESSES TRAPPED ARE:
                        2137  *
                        2138  *                       IN 360Q FRONT PANEL KEYBOARD INPUT
                        2139  *                      OUT 360Q FRONT PANEL CONTROL
                        2140  *                      OUT 361Q FRONT PANEL DISPLAY CONTROL
                        2141  *                   IN/OUT 372Q CONSOLE DATA FOR AN 8251A
                        2142  *                      OUT 373Q CONSOLE CONTROL FOR AN 8251A
                        2143  *
                        2144  *
                        2145  *               THESE PORT REQUESTS ARE RESPONDED TO AS FOLLOWS:
                        2146  *
                        2147  *                       IN 360Q RETURNS WITH (A) = 377Q TO SHOW THAT
                        2148  *                            NO FRONT PANEL SWITCHES ARE PRESSED
                        2149  *
                        2150  *                       OUT 360Q MOVES BIT 6 (CB.CLI) TO BIT 1, AND
                        2151  *                            BIT 4 (CB.SSI) INVERTED, TO BIT 0, AND
                        2152  *                            OUTPUTS THESE BITS TO PORT 362Q TO
                        2153  *                            CONTROL THE CLOCK AND SINGLE STEP INTERRUPTS
                        2154  *
                        2155  *                       OUTPUTS TO 361Q, 372Q, AND 373Q JUST RETURN
                        2156  *
                        2157  *                       INPUTS FROM 361Q, 372Q, AND 373Q RETURN WITH (A) = 0
                        2158  *                            TO INDICATE AN EMPTY BUSS
                        2159  *
                        2160  *
                        2161  *               ENTRY   NONE
                        2162  *
                        2163  *               EXIT    NONE
                        2164  *
                        2165  *               USES    (A) ONLY IF "FAKING" AN INPUT
                        2166
                        2167
004.116 343             2168  NMI     XTHL                    GET RETURN ADDRESS FROM STACK
004.117 042 064 040     2169          SHLD    NMIRET          SAVE FOR LATER USE
004.122 343             2170          XTHL                    PUT RETURN ADDRESS BACK ON STACK
                        2171
004.123 345             2172          PUSH    H               SAVE REGISTERS
004.124 305             2173          PUSH    B
004.125 365             2174          PUSH    PSW
004.126 107             2175          MOV     B,A             SAVE (A) PRIOR TO I/O
004.127 052 064 040     2176          LHLD    NMIRET          GET RETURN ADDRESS
004.132 053             2177          DCX     H               BACK UP TO PORT # WHICH GOT US HERE
004.133 176             2178          MOV     A,M             GET PORT #
                        2179
004.134 376 360         2180          CPI     360Q            PORT 360?
004.136 050 033         2181          JR      Z,NMI1            IF PORT WAS 360Q
                        2182
                        2183  *       PORT REFERENCED WAS 361Q, 372Q, OR 373Q
                        2184  *
004.140 376 361         2185          CPI     361Q            MAKE SURE PORT IS LEGAL
004.142 050 010         2186          JR      Z,NMI0.5          IF LEGAL
                        2187
```

```
004.144  376 372        2188           CPI     372Q
004.146  050 004        2189           JR      Z,NMI0.5
                        2190
004.150  376 373        2191           CPI     373Q
004.152  040 062        2192           JR      NZ,NMI2.5          IF NONE OF THE ABOVE, EXIT
                        2193
004.154  053            2194  NMI0.5   DCX     H                  POINT TO IN/OUT INSTRUCTION
004.155  176            2195           MOV     A,M                SEE IF INPUT OR OUTPUT
004.156  376 323        2196           CPI     MI.OUT
004.160  050 054        2197           JR      Z,NMI2.5             IF OUTPUT, JUST EXIT
                        2198
004.162  376 333        2199           CPI     MI.IN
004.164  040 050        2200           JR      NZ,NMI2.5            IF NOT INPUT EITHER, ILLEGAL SO EXIT
                        2201
004.166  361            2202           POP     PSW                RESTORE FLAGS
004.167  076 000        2203           MVI     A,0                ELSE, RETURN LIKE AN EMPTY BUSS
004.171  030 044        2204           JR      NMI3               EXIT
                        2205
004.173  053            2206  NMI1     DCX     H                  POINT TO IN/OUT INSTRUCTION
004.174  176            2207           MOV     A,M                GET I/O INSTRUCTION
004.175  376 333        2208           CPI     MI.IN              INPUT?
004.177  040 005        2209           JR      NZ,NMI1.5            IF NOT "IN"
                        2210
004.201  361            2211           POP     PSW                RESTORE FLAGS
004.202  076 377        2212           MVI     A,11111111B        SHOW "NO KEYS PRESSED"
004.204  030 031        2213           JR      NMI3               EXIT
                        2214
004.206  376 323        2215  NMI1.5   CPI     MI.OUT             MAKE SURE INTRUCTION IS AN "OUT"
004.210  040 024        2216           JR      NZ,NMI2.5            IF NOT
                        2217
004.212  170            2218  NMI2     MOV     A,B                GET OUTPUT DATA AGAIN
004.213  346 120        2219           ANI     CB.CLI+CB.SSI      MOVE CLOCK INFO TO BIT 1
004.215  017            2220           RRC
004.216  017            2221           RRC
004.217  017            2222           RRC
004.220  017            2223           RRC
004.221  017            2224           RRC
004.222  070 001        2225           JR      C,NMI2.2
004.224  074            2226           INR     A
004.225  041 066 040    2227  NMI2.2   LXI     H,DATA             OR WITH THE BYTE IN RAM
004.230  266            2228           ORA     M                  BEFORE OUTPUT IT
004.231  323 362        2229           OUT     H88.CTL            SET IN HARDWARE
004.233  346 374        2230           ANI     11111100B
004.235  167            2231           MOV     M,A
                        2232
004.236  361            2233  NMI2.5   POP     PSW                RESTORE (A,F)
                        2234
004.237  301            2235  NMI3     POP     B
004.240  341            2236           POP     H
                        2237  *        RETN                       Z80 RETURN FROM NMI
004.241  355 105        2238           DB      355Q,105Q
```

```
                                2242  **      ATB     - AUTO BOOT ROUTINE CONTINUE
                                2243
004.243  167                    2244  ATB     MOV     M,A             SET AUTO BOOT FLAG
004.244  076 012                2245          MVI     A,10            SET TO AUTO BOOT ROUTINE
004.246  315 052 003            2246          CALL    LRA.
004.251  021 205 001            2247          LXI     D,AUTOBO        SET AUTO BOOT ROUTINE
004.254  030 017                2248          JR      BOOTX


000.000                         2250          ERRNI   4256A-*
004.256                         2251          ORG     4256A
                                2252  **      BOOT H-17 OR Z47 ENTRY POINT FOR H88
                                2253  *
                                2254  *       ENTRY   NONE
                                2255  *
                                2256  *       EXIT    (DE) = NORMAL BOOT ROUTINE ADDRESS
                                2257  *
                                2258  *       USES    ALL
                                2259
004.256  041 234 006            2260  BOOT    LXI     H,MSG.BT        COMPLETE BOOT MESSAGE
004.261  315 100 006            2261          CALL    TYPMSG
004.264  363                    2262          DI
004.265  076 012                2263          MVI     A,10
004.267  315 052 003            2264          CALL    LRA.            GET LOCATION OF USER PC
004.272  021 261 001            2265          LXI     D,NBOOT         SET ITS VALUE TO THE NORMAL BOOT ROUTINE
004.275  163                    2266  BOOTX   MOV     M,E
004.276  043                    2267          INX     H
004.277  162                    2268          MOV     M,D
004.300  373                    2269          EI
004.301  303 063 000            2270          JMP     GO.             DO IT
```

```
                                2273  **      TMOUT   -  BOOT CODE TIME OUT ROUTINE
                                2274  *
                                2275  *       TMOUT IS ENTERED FROM TIMER INTERRUPT EVER 100 MS. AND IT WILL
                                2276  *       EXIT:   IF BOOT SUCCESS THEN TIMER OFF.
                                2277  *               IF 15 SECONDS TIME OUT AND BOOT IS NOT SUCCESS YES
                                2278  *                  THEN ABORT BOOT Z47 & TO MONITOR LOOP
                                2279  *               IF < 15S & 3.5S THEN RE-BOOT
                                2280  *
                                2281  *       NOTE: Because the H37 and H67 run with interrupts disabled
                                2282  *             during portions of the code, they handle their own
                                2283  *             time outs.
                                2284  *
                                2285  *       ENTRY:  (TMFG)  = 1 IF THE TIME OUT IS FOR Z47
                                2286  *                       = 0 IF THE TIME OUT IS FOR H17
                                2287  *       EXIT:   NONE
                                2288  *
                                2289  *       USE:    ALL (WHEN RETURN, ALL REGISTERS ARE RESTORED)
                                2290
004.304                         2291  TMOUT   EQU     *
004.304  333 355                2292          IN      SC.ACE+UR.LSR           INPUT ACE LINE STATUS REGISTER
004.306  346 001                2293          ANI     UC.DR                   SEE IF THERE IS A DATA READY
004.310  050 011                2294          JR      Z,TMOUT4                CHECK IF IT IS <DELETE>
                                2295
004.312  333 350                2296          IN      SC.ACE+UR.RBR           INPUT DATA FROM KB
004.314  346 177                2297          ANI     01111111B               IS IT <DEL>?
004.316  376 177                2298          CPI     A.DEL
004.320  312 171 002            2299          JZ      NODEV                   IF IT, ABORT THE BOOT
                                2300  *                                       ELSE IGNORE THE INPUT
004.323  041 121 041            2301  TMOUT4  LXI     H,TMFG
004.326  176                    2302          MOV     A,M
004.327  247                    2303          ANA     A
004.330  010                    2304          DB      MI.EXAF                 SAVE Z FLAG
004.331  072 033 040            2305          LDA     TICCNT                  GET LOW ORDER COUNTER
004.334  247                    2306          ANA     A                       SET ZERO FLAG
004.335  040 024                2307          JR      NZ,TMOUT2               NOT IN 0.5 SECOND
004.337  043                    2308          INX     H                       SET TO MYCNT
000.000                         2309          ERRNZ   MYCNT-TMFG-1            MYCNT MUST FOLLOW TMFG
004.340  064                    2310          INR     M                       INCREASE THE COUNT FOR 0.5 SECOND
004.341  176                    2311          MOV     A,M
004.342  376 036                2312          CPI     30                      CHECK IF MORE THAN 15 SECONDS
004.344  322 171 002            2313          JNC     NODEV                   NO DEVICE ?
004.347  336 007                2314  TMOUT1  SBI     7                       IS IT 3.5 SECONDS?
004.351  070 010                2315          JR      C,TMOUT2                IF NOT, WAIT
004.353  040 372                2316          JR      NZ,TMOUT1               CHECK MORE
004.355  010                    2317          DB      MI.EXAF
004.356  302 036 002            2318          JNZ     RETRY                   IF IT IS Z47, THEN RE-BOOT
004.361  030 002                2319          JR      TMOUT3                  IT IS H-17, CONTINUE IT CLOCK ROUTINE
004.363  010                    2320  TMOUT2  DB      MI.EXAF                 CHECK IT IS Z47 OR H17
004.364  300                    2321          RNZ                             Z47, THEN RETURN
004.365  303 031 034            2322  TMOUT3  JMP     CLOCK17                 CONTINUE H17 CLOCK ROUTINE
```

```
000.000                  2325          ERRMI   4370A-*
004.370                  2326          ORG     4370A
                         2327  **      SUBM - SUBSTITUTE MEMORY
                         2328  *
                         2329  *       SUBM INPUTS A MEMORY ADDRESS FROM THE CONSOLE AND THEN DISPLAYS
                         2330  *       THAT ADDRESS AND ITS CONTENTS.  IF A CARRIAGE RETURN IS THEN TYPED,
                         2331  *       CONTROL RETURNS TO THE MONITOR.  IF A SPACE IS TYPED, THE NEXT
                         2332  *       MEMORY LOCATION AND CONTENTS ARE DISPLAYED.  IF A MINUS SIGN IS
                         2333  *       TYPED, THE PREVIOUS MEMORY LOCATION AND CONTENTS ARE DISPLAYED.
                         2334  *       IF AN OCTAL CHARACTER IS TYPED, A BYTE IS ENTERED AN PLACED AT THE
                         2335  *       CURRENT MEMORY LOCATION.
                         2336  *
                         2337  *
                         2338  *       ENTRY   NONE
                         2339  *       EXIT    NONE
                         2340  *       USES    A,E,H,L,F
                         2341
                         2342
004.370  041 201 006   2343  SUBM    LXI     H,MSG.SUB       COMPLETE SUBSTITUTE MESSAGE
004.373  315 100 006   2344          CALL    TYPMSG
004.376  315 012 015   2345          CALL    IKOC            INPUT FIRST CHARACTER
005.001  320           2346          RNC                     IF A RETURN, EXIT
                         2347
005.002  041 003 040   2348          LXI     H,IOWRK+1       ELSE, INPUT STARTING ADDRESS
005.005  026 015       2349          MVI     D,A.CR          ENDING WITH A RETURN
005.007  315 023 015   2350          CALL    IOA
005.012  353           2351          XCHG                    (H,L) = INPUT ADDRESS
                         2352
005.013  315 064 015   2353  SUBM1   CALL    TOA             TYPE CRLF, ADDRESS, AND A SPACE
005.016  176           2354          MOV     A,M             GET MEMORY CONTENTS FOR DISPLAY
005.017  315 077 015   2355          CALL    TOB
005.022  076 040       2356          MVI     A,' '           SPACE
005.024  315 302 003   2357          CALL    WCC
                         2358
005.027  315 051 015   2359  SUBM2   CALL    IOC             INPUT FIRST CHARACTER
005.032  322 075 005   2360          JNC     SUBM7           IF FIRST CHARACTER IS OCTAL
                         2361
005.035  376 040       2362          CPI     ' '             SPACE?
005.037  302 046 005   2363          JNZ     SUBM4           IF NOT A SPACE
                         2364
005.042  043           2365  SUBM3   INX     H               POINT TO NEXT ADDRESS
005.043  303 013 005   2366          JMP     SUBM1           DISPLAY NEXT
                         2367
005.046  376 055       2368  SUBM4   CPI     '-'             MINUS?
005.050  302 062 005   2369          JNZ     SUBM6           IF NOT
                         2370
005.053  315 302 003   2371  SUBM5   CALL    WCC             ECHO HYPHEN
005.056  053           2372          DCX     H               POINT TO PREVIOUS ADDRESS
005.057  303 013 005   2373          JMP     SUBM1           DISPLAY PREVIOUS
                         2374
005.062  376 015       2375  SUBM6   CPI     A.CR            RETURN?
005.064  310           2376          RZ                      IF RETURN, EXIT
                         2377
005.065  076 007       2378          MVI     A,A.BEL         ELSE, DING BELL
005.067  315 302 003   2379          CALL    WCC
005.072  303 027 005   2380          JMP     SUBM2           TRY AGAIN
```

```
                                2381
005.075  066 000               2382  SUBM7   MVI     M,0             ZERO BYTE TO BE BUILT
                                2383
005.077  315 302 003           2384  SUBM8   CALL    HCC             ECHO OCTAL CHARACTER
005.102  315 352 014           2385          CALL    SUBM10
005.105  315 051 015           2386  SUBM9   CALL    IOC             INPUT NEXT CHARACTER
005.110  322 077 005           2387          JNC     SUBM8           IF OCTAL
                                2388
005.113  376 040               2389          CPI     ' '             SPACE?
005.115  312 042 005           2390          JZ      SUBM3           IF SPACE, DISPLAY NEXT BYTE
                                2391
005.120  376 055               2392          CPI     '-'             MINUS?
005.122  312 053 005           2393          JZ      SUBM5           IF MINUS, DISPLAY PREVIOUS
                                2394
005.125  376 015               2395          CPI     A.CR            RETURN?
005.127  310                   2396          RZ                      IF RETURN, EXIT
                                2397
005.130  076 007               2398          MVI     A,A.BEL         ELSE, DING BELL
005.132  315 302 003           2399          CALL    HCC
005.135  303 105 005           2400          JMP     SUBM9           TRY AGAIN


                                2403  **      IROC - INPUT A RETURN OR AN OCTAL CHARACTER
                                2404  *
                                2405  *       IROC INPUTS A CHARACTER FROM THE CONSOLE AND WAITS UNTIL IT
                                2406  *       RECEIVES EITHER A VALID OCTAL CAHRACTER OR A CARRIAGE RETURN
                                2407  *
                                2408  *       ENTRY   NONE
                                2409  *       EXIT    (A) = INPUT CHARACTER
                                2410  *               'C' = SET IF CHARACTER IS OCTAL
                                2411  *       USES    A,F
                                2412
                                2413
005.140  315 262 003           2414  IROC0   CALL    RCC             INPUT CHARACTER
005.143  376 015               2415          CPI     A.CR            RETURN?
005.145  310                   2416          RZ                      IF A CR
                                2417
005.146  376 060               2418          CPI     '0'             < 0?
005.150  332 156 005           2419          JC      IROC1           IF < OCTAL
                                2420
005.153  376 070               2421          CPI     '8'             > 8?
005.155  330                   2422          RC                      IF OCTAL
                                2423
005.156  076 007               2424  IROC1   MVI     A,A.BEL         ELSE, RING BELL
005.160  315 302 003           2425          CALL    HCC
005.163  303 012 015           2426          JMP     IROC            TRY AGAIN
```

```
                          2428  **      IOA1 - INPUT OCTAL ADDRESS
                          2429  *
                          2430  *       IOA1 IS A CONTINUATION OF *IOA* AND INPUTS A SPLIT OCTAL ADDRESS
                          2431  *       WITHOUT REQUIRING LEADING ZEROS
                          2432  *
                          2433  *       ENTRY   (H,L) = ADDRESS + 1 WHERE INPUT ADDRESS IS TO BE PLACED
                          2434  *               (A) = FIRST OCTAL CHARACTER IF 'C' IS SET
                          2435  *       EXIT    (D,E) = INPUT ADDRESS
                          2436  *               (A) = LAST INPUT CHARACTER
                          2437  *       USES    A,D,E,H,L,F
                          2438
                          2439
005.166  305             2440  IOA1    PUSH    B               SAVE (B,C)
005.167  102             2441          MOV     B,D             (B) = TERMINATION CHARACTER
005.170  036 000          2442          MVI     E,0             CLEAR PSEUDO FLAGS
005.172  345             2443          PUSH    H               SAVE ADDRESS WHERE INPUT IS TO BE PLACED
005.173  041 000 000      2444          LXI     H,0             SET NEW VALUE TO ZERO
005.176  324 262 003      2445  IOA2    CNC     RCC             IF CARRY SET, FIRST CHARACTER IS IN ACC
005.201  315 271 005      2446          CALL    IOC.            CHECK VALIDITY
005.204  332 230 005      2447          JC      IOA3            IF < OCTAL
                          2448
005.207  315 302 003      2449          CALL    WCC             ECHO OCTAL CHARACTER
005.212  346 007          2450          ANI     00000111B       GET BINARY VALUE
005.214  365             2451          PUSH    PSW             SAVE NEW CHARACTER VALUE
005.215  051             2452          DAD     H               SHIFT THREE TO MAKE ROOM FOR NEW CHARACTER
005.216  051             2453          DAD     H
005.217  051             2454          DAD     H
005.220  365             2455          PUSH    PSW             SAVE CARRY FROM DAD
005.221  321             2456          POP     D               SAVE FLAG RESULT IN E
005.222  361             2457          POP     PSW             RETURN NEW CHARACTER VALUE TO (A)
005.223  205             2458          ADD     L
005.224  157             2459          MOV     L,A
005.225  303 176 005      2460          JMP     IOA2            SEE IF MORE CHARACTERS
                          2461
005.230  270             2462  IOA3    CMP     B               TERMINATING CHARACTER?
005.231  312 245 005      2463          JZ      IOA4            IF EQUAL
                          2464
                          2465
005.234  076 007          2466          MVI     A,A.BEL         ELSE, DING BELL
005.236  315 302 003      2467          CALL    WCC
005.241  247             2468          ANA     A               CLEAR CARRY
005.242  303 176 005      2469          JMP     IOA2
                          2470
                          2471  *       END OF INPUT, PUT VALUE IN MEMORY AND EXIT
                          2472
005.245  315 302 003      2473  IOA4    CALL    WCC             ECHO CHARACTER
005.250  127             2474          MOV     D,A             LAST CHARACTER TO D
005.251  325             2475          PUSH    D
005.252  361             2476          POP     PSW             (PSW) = RESULT OF DAD
005.253  174             2477          MOV     A,H             MAKE (H) INTO SPLIT OCTAL
005.254  037             2478          RAR
005.255  147             2479          MOV     H,A
005.256  172             2480          MOV     A,D             RESTORE LAST INPUT CHARACTER
005.257  353             2481          XCHG                    (D,E) = INPUT ADDRESS
005.260  341             2482          POP     H               (H,L) = LOCATION TO PLACE THIS ADDRESS
005.261  162             2483          MOV     M,D
```

```
005.262  053              2484          DCX   H
005.263  163              2485          MOV   M,E
005.264  301              2486          POP   B          RESTORE (B,C)
005.265  311              2487          RET

                          2489  **      IOC - INPUT OCTAL CHARACTER
                          2490  *
                          2491  *
                          2492  *       ENTRY NONE
                          2493  *       EXIT  (A) = INPUT CHARACTER
                          2494  *             'C' = SET IF CHARACTER NOT OCTAL
                          2495  *       USES  A,F
                          2496
                          2497
005.266  315 262 003      2498  IOCO    CALL  RCC        INPUT CHARACTER
005.271  376 060          2499  IOC.    CPI   '0'
005.273  330              2500          RC               IF CHARACTER < OCTAL
                          2501
005.274  376 070          2502          CPI   '8'        CHARACTER > OCTAL?
005.276  077              2503          CMC              'C' IF GREATER THAN
005.277  311              2504          RET

                          2506  **      TOA - TYPE OCTAL ADDRESS
                          2507  *
                          2508  *       TOA OUTPUTS TO THE CONSOLE A CRLF, THE SPCECIFIED ADDRESS AND A SPACE
                          2509  *
                          2510  *       ENTRY (H,L) = ADDRESS TO BE DISPLAYED
                          2511  *       EXIT  NONE
                          2512  *       USES  A,B,C,F
                          2513
                          2514
005.300  076 015          2515  TOAO    MVI   A,A.CR     CRLF
005.302  315 370 005      2516          CALL  HCR.
                          2517
005.305  174              2518  TOA.    MOV   A,H        ADDRESS
005.306  315 322 005      2519          CALL  TOBO
005.311  175              2520          MOV   A,L
005.312  315 322 005      2521          CALL  TOBO
                          2522
005.315  076 040          2523          MVI   A,' '      SPACE
005.317  303 302 003      2524          JMP   HCC
```

```
                                2526  **      TOB - TYPE OCTAL BYTE
                                2527  *
                                2528  *       TOB OUTPUTS TO THE CONSOLE IN OCTAL, THE BYTE IN A
                                2529  *
                                2530  *       ENTRY   (A) = BYTE TO BE OUTPUT
                                2531  *       EXIT    NONE
                                2532  *       USES    A,F
                                2533
                                2534
005.322  305                   2535  TOBO    PUSH    B
005.323  006 002               2536          MVI     B,2             NUMBER OF CHARACTERS - 1
005.325  117                   2537          MOV     C,A             SAVE ORIGINAL BYTE
005.326  247                   2538          ANA     A               CLEAR CARRY
005.327  037                   2539          RAR
005.330  037                   2540          RAR                     SHIFT TOP BYTE TO LSB
005.331  037                   2541          RAR
005.332  037                   2542  TOB1    RAR                     SHIFT MIDDLE BYTE TO LSB
005.333  037                   2543          RAR
005.334  037                   2544          RAR
005.335  346 007               2545          ANI     00000111B       MASK FOR HALF ASCII
005.337  366 060               2546          ORI     00110000B       MAKE WHOLE ASCII
005.341  315 302 003           2547          CALL    WCC             OUTPUT TO CONSOLE
005.344  171                   2548          MOV     A,C             GET ORIGINAL BYTE
005.345  005                   2549          DCR     B
005.346  302 332 005           2550          JNZ     TOB1            IF SECOND BYTE STILL NEEDS TO BE OUTPUT
                                2551
005.351  346 007               2552          ANI     00000111B       ELSE, OUTPUT LAST CHARACTER
005.353  366 060               2553          ORI     00110000B
005.355  301                   2554          POP     B
005.356  303 302 003           2555          JMP     WCC


                                2557  **      WCR - WAIT FOR A CARRIAGE RETURN
                                2558  *
                                2559  *       WCR INPUTS CHARACTERS FROM THE CONSOLE UNTIL A CARRIAGE RETURN
                                2560  *       IS RECEIVED AND THEN ECHOS A CRLF
                                2561  *
                                2562  *
                                2563  *       ENTRY   NONE
                                2564  *       EXIT    NONE
                                2565  *       USES    A,F
                                2566
                                2567
005.361  315 262 003           2568  WCR     CALL    RCC             INPUT CHARACTER
005.364  376 015               2569          CPI     A.CR
005.366  040 371               2570          JR      NZ,WCR              IF NOT A CR
                                2571
005.370  315 302 003           2572  WCR.    CALL    WCC             ELSE, ECHO CR
005.373  076 012               2573          MVI     A,A.LF          LINE FEED
005.375  303 302 003           2574          JMP     WCC
```

```
                        2576  **      VIEW3 - *VEIW* CONTINUATION
                        2577  *
                        2578
006.000  302 066 002    2579  VIEW3   JNZ     VIEW2           IF NOT END OF LINE
006.003  315 355 003    2580          CALL    VIEW9           END OF LINE, RESTORE ADDRESS
006.006  076 015        2581          MVI     A,A.CR
006.010  303 355 007    2582          JMP     VIEW3A.         DO ASCII STUFF
000.010                 2583          ERRMI   6023A-*
006.023                 2584          ORG     6023A



                        2586  ***     DAT     -       DATA BYTE OUTPUT TO Z-47
                        2587  *
                        2588  *       ENTRY:  (A) = BYTE TO OUTPUT
                        2589  *
                        2590  *       EXIT:   (A) = BYTE TO OUTPUT
                        2591  *               (D) = S.DTR
                        2592  *
                        2593  *       USE:    AF, D
                        2594
006.023                 2595  DAT     EQU     *
006.023  026 200        2596          MVI     D,S.DTR         SET MATCH CONDITION TO DATA TRANSFER
006.025  030 002        2597          JR      COM1            REQUEST BIT
000.000                 2598          ERRMI   6027A-*
006.027                 2599          ORG     6027A



                        2601  ***     COM     - OUTPUT COMMAND BYTE TO Z-47
                        2602  *
                        2603  *       ENTRY:  (A) = COMMAND BYTE
                        2604  *
                        2605  *       EXIT:   (A) = COMMAND BYTE
                        2606  *               (D) = S.DON
                        2607  *
                        2608  *       USE:    AF, D
                        2609
006.027                 2610  COM     EQU     *
006.027  026 040        2611          MVI     D,S.DON         SET MATCH CONDITION TO DONE BIT
006.031  365            2612  COM1    PUSH    PSW
006.032  315 170 006    2613  WTDON1  CALL    IN.             READ CONTROLLER STATUS REGISTER
006.035  242            2614          ANA     D               GET MATCH BIT ONLY
006.036  050 372        2615          JR      Z,WTDON1        IF NO MATCH, WAIT
006.040  361            2616          POP     PSW
006.041  303 156 010    2617          JMP     COM2            CONTINUE *COM* ROUTINE
```

```
000.001                 2619            ERRMI   6045A-*
006.045                 2620            ORG     6045A
                        2621    **      HRNX - HORN EXTENSION ROUTINE
                        2622    *
                        2623    *       THIS IS AN EXTENSION TO *HORN* TO MAKE ROOM FOR A JUMP
                        2624
006.045  056 011       2625    HRNX    MVI     L,#CTLFLG
006.047  163           2626            MOV     M,E             TURN OFF HORN
006.050  321           2627            POP     D
006.051  341           2628            POP     H
006.052  311           2629            RET


                        2631    **      NOISE - DING BELL ON CONSOLE
                        2632    *
                        2633    *       THIS IS A MODIFICATION TO ALLOW THE H88/H89 TO USE THE CONSOLE BELL
                        2634
006.053  076 007       2635    NOISE   MVI     A,A.BEL
006.055  315 302 003   2636            CALL    WCC
006.060  303 140 002   2637            JMP     HORN            CONTINUE WITH NORMAL HORN DELAY


                        2639    **      OUT.    - OUTPUT BYTE TO Z-47
                        2640    *
                        2641    *       ENTRY:  (A) = OUTPUT BYTE
                        2642    *
                        2643    *       EXIT:   NONE
                        2644    *
                        2645    *       USE:    NONE
                        2646
006.063                 2647    OUT.    EQU     *
006.063  305           2648            PUSH    B
006.064  107           2649            MOV     B,A             SAVE THE OUTPUT DATA
006.065  072 120 041   2650            LDA     PRIM            GET PORT ADDRESS
006.070  117           2651    OUT.1   MOV     C,A             SET TO REG C
006.071  170           2652            MOV     A,B             GET OUTPUT BYTE DATA BACK
                        2653    *       OUT     (C),A           OUTPUT BYTE
006.072  355 171       2654            DB      355Q,171Q
006.074  301           2655            POP     B
006.075  311           2656            RET


000.002                 2658            ERRMI   6100A-*
006.100                 2659            ORG     6100A
                        2660    **      TYPMSG - TYPE MESSAGE TO CONSOLE
                        2661    *
                        2662    *       TYPMSG OUTPUTS AN ASCII MESSAGE FROM MEMORY TO THE CONSOLE
                        2663    *       UNTIL A NULL IS SENSED
                        2664    *
                        2665    *       ENTRY   (H,L) = ADDRESS OF MESSAGE
```

```
                        2666  *       EXIT    NONE
                        2667  *       USES    A,H,L,F
                        2668
                        2669
006.100  176            2670  TYPMSG  MOV     A,M             GET CHARACTER
006.101  267            2671          ORA     A               SEE IF A NULL
006.102  310            2672          RZ                      IF NULL, EXIT
                        2673
006.103  315 302 003    2674          CALL    WCC             ELSE OUTPUT CHARACTER TO CONSOLE
006.106  043            2675          INX     H               POINT TO NEXT CHARACTER
006.107  030 367        2676          JR      TYPMSG          OUTPUT IT


                        2678  **      RDBLCK  - INPUT A BLOCK FROM Z-47
                        2679  *
                        2680  *       RDBLCK  READS IN A BLOCK FROM THE DISK CONTROOLER
                        2681  *
                        2682  *       ENTRY:  HL = LOAD ADDRESS
                        2683  *               C  = SIDE/UNIT/SECTOR
                        2684  *
                        2685  *       EXIT:   BLOCK IN READ IN MEMORY
                        2686  *
                        2687  *       USES:   ALL
                        2688  *
                        2689
006.111  076 007        2690  RDBLCK  MVI     A,DD.READ
006.113  315 027 006    2691          CALL    COM             SEND THE COMMAND
006.116  257            2692          XRA     A               FOR TRACK 0
006.117  315 023 006    2693          CALL    DAT             SEND IT TO DISK
006.122  171            2694          MOV     A,C             LOAD SIDE/UNIT/SECTOR
006.123  315 023 006    2695          CALL    DAT             SEND IT TO DISK
                        2696
006.126  315 067 001    2697  RD2     CALL    PIN             GET STATUS
006.131  332 104 010    2698          JC      WDN             'C' SET IF S.DON
                        2699
006.134  167            2700          MOV     M,A
006.135  043            2701          INX     H
006.136  030 366        2702          JR      RD2             CONTINUE TRANSFER


                        2704  **      OUT1.   -  OUTPUT BYTE TO PORT (PRIM+1)
                        2705  *
                        2706  *       ENTRY:  (A) = OUTPUT PORT
                        2707  *
                        2708  *       EXIT:   NONE
                        2709  *
                        2710  *       USE:    NONE
                        2711
006.140                 2712  OUT1.   EQU     *
006.140  305            2713          PUSH    B
006.141  107            2714          MOV     B,A             SAVE THE OUTPUT DATA
006.142  072 120 041    2715          LDA     PRIM            GET PORT ADDRESS
```

```
006.145  074              2716          INR     A                       SET TO (PRIM+1)
006.146  030 320          2717          JR      OUT.1                   GO TO OUTPUT ROUTINE


                          2719  **      IN1.    - INPUT BYTE FROM (PRIM+1) PORT
                          2720  *
                          2721  *       ENTRY:  NONE
                          2722  *
                          2723  *       EXIT:   (A) = INPUT BYTE
                          2724  *
                          2725  *       USE:    A
                          2726
006.150                   2727  IN1.    EQU     *
006.150  305              2728          PUSH    B
006.151  072 120 041      2729          LDA     PRIM                    GET PORT ADDRESS
006.154  074              2730          INR     A                       SET TO (PRIM+1)
006.155  247              2731          ANA     A
006.156  030 014          2732          JR      IN.1                    GO TO INPUT ROUTINE


000.005                   2734          ERRMI   6165A-*
006.165                   2735          ORG     6165A
                          2736  **      MSG.GO - (G)O
                          2737  *
                          2738  *       "GO"
                          2739
006.165  157 040 000      2740  MSG.GO  DB      'o ',0


                          2742  **      IN.     - INPUT BYTE FROM PORT (PRIM)
                          2743  *
                          2744  *       ENTRY:  NONE
                          2745  *
                          2746  *       EXIT:   (A) = INPUT BYTE
                          2747  *
                          2748  *       USE:    A
                          2749
006.170                   2750  IN.     EQU     *
006.170  305              2751          PUSH    B
006.171  072 120 041      2752          LDA     PRIM                    GET PORT ADDRESS
006.174  117              2753  IN.1    MOV     C,A                     SET ADDR. TO REG C
                          2754  *       IN      A,(C)
006.175  355 170          2755          DB      355Q,170Q               INPUT BYTE
006.177  301              2756          POP     B
006.200  311              2757          RET
```

```
000.000                  2759           ERRMI   6201A-*
006.201                  2760           ORG     6201A
                         2761  **       MSG.SUB - (S)UBSTITUTE
                         2762  *
                         2763  *        "SUBSTITUTE"
                         2764
006.201  165 142 163    2765  MSG.SUB  DB      'ubstitute ',0


                         2767  **       MSG.PC - (P)ROGRAM COUNTER
                         2768  *
                         2769  *        "PROGRAM COUNTER"
                         2770
006.214  162 157 147    2771  MSG.PC   DB      'rogram Counter ',0


                         2773  **       MSG.BT - (B)OOT
                         2774  *
                         2775  *        "BOOT"
                         2776
006.234  157 157 164    2777  MSG.BT   DB      'oot',0
```

```
                    2780  ***     SPEED - ROTATIONAL SPEED TEST FOR 5.25 INCH DISK DRIVE
                    2781  *
                    2782  *       *SPEED* IS USED ONLY FOR GROSS ADJUSTMENT OF DRIVE ROTATIONAL
                    2783  *       SPEED IF THE FIRST READ/WRITE TEST OF THE UNIT FAILS DURING SET UP.
                    2784  *
                    2785  *               USE OF *SPEED* IS AS FOLLOWS:
                    2786  *
                    2787  *                       1.  ENTER "GO AND THE ENTRY ADDRESS OF *SPEED*
                    2788  *                       2.  ADJUST DRIVE SPEED UNTIL DATA AT DISPLAYED
                    2789  *                           EQUALS 200
                    2790  *                              A.  IF SPEED < 200, TURN ADJUSTMENT CLOCKWISE
                    2791  *                              B.  IF SPEED > 200, TURN COUNTERCLOCKWISE
                    2792  *
                    2793  *       THE ABOVE TEST ADJUSTS SY0:.  TO ADJUST SY1:, USE HOOS


                    2795  **      LABLE EQUIVALENCES
                    2796  *
                    2797  *       I/O PORTS
000.177             2798  OP.DC   EQU     177Q            DRIVE CONTROL OUTPUT PORT
000.177             2799  IP.DS   EQU     177Q            DRIVE STATUS INPUT PORT


                    2801  *       MASKS
                    2802  *
000.001             2803  DS.HOLE EQU     00000001B       DRIVE STATUS SECTOR/INDEX HOLE


                    2805  *       CONSTANTS
                    2806  *
000.022             2807  ONDRO   EQU     022Q            TURN ON SY0:
```

```
006.240  041 371 006  2809  SPEED   LXI   H,MSG.SPD       OUTPUT SPEED MESSAGE
006.243  315 100 006  2810          CALL  TYPMSG
006.246  076 000      2811          MVI   A,0             SET FLAG AT IOWRK FOR "WORKING" MESSAGE
006.250  062 002 040  2812          STA   IOWRK
006.253  076 022      2813          MVI   A,ONDR0         TURN ON DRIVE ZERO
006.255  323 177      2814          OUT   OP.DC
006.257  052 033 040  2815  SPEED1  LHLD  TICCNT          GET TICK COUNTER
006.262  174          2816          MOV   A,H             FORM TWO'S COMPLEMENT OF TICK COUNTER
006.263  057          2817          CMA
006.264  127          2818          MOV   D,A             (D,E) = NEGATIVE TICK COUNTER
006.265  175          2819          MOV   A,L
006.266  057          2820          CMA
006.267  074          2821          INR   A
006.270  137          2822          MOV   E,A
006.271  322 275 006  2823          JNC   SPEED2          IF NO CARRY FROM LSB
                      2824
006.274  024          2825          INR   D               ELSE, INCREMENT MSB
006.275  001 000 000  2826  SPEED2  LXI   B,0             ZERO REV COUNTERS
006.300  333 177      2827  SPEED3  IN    IP.DS           INPUT DISK STATUS
006.302  346 001      2828          ANI   DS.HOLE         MASK FOR SECTOR/INDEX PULSES
006.304  312 300 006  2829          JZ    SPEED3          IF NO HOLE PRESENT
                      2830
                      2831  *       HOLE PRESENT, WAIT FOR IT TO LEAVE
                      2832  *
006.307  333 177      2833  SPEED4  IN    IP.DS           GET DISK STATUS
006.311  346 001      2834          ANI   DS.HOLE         GET HOLE PULSES
006.313  302 307 006  2835          JNZ   SPEED4          WAIT UNTIL HOLE IS GONE AND WE HAVE MEDIA
                      2836
006.316  004          2837          INR   B               INCREMENT HOLE COUNTER
006.317  170          2838          MOV   A,B             TEST FOR FIVE REVOLUTIONS
006.320  376 070      2839          CPI   56
006.322  302 300 006  2840          JNZ   SPEED3          NOT FIVE, WAIT FOR MORE HOLES
                      2841
                      2842  *       HAVE FIVE REVS, DISPLAY DIFFERENCE OF TICK COUNTER AND EXPECTED TIME DIF
                      2843  *
006.325  052 033 040  2844          LHLD  TICCNT          GET CURRENT TICK VALUE
006.330  031          2845          DAD   D               SUBTRACT START VALUE
006.331  021 214 376  2846          LXI   D,377377A-500+1+200Q   SUBTRACT 500 FOR REVS, +200Q FOR OFFSET
006.334  031          2847          DAD   D               (H,L) = OFFSET RESULT
006.335  345          2848          PUSH  H               SAVE RESULT
006.336  041 062 007  2849          LXI   H,MSG.WRK       POINT TO "WORKING" MESSAGE
006.341  072 002 040  2850          LDA   IOWRK           GET "WORKING" FLAG
006.344  356 001      2851          XRI   1               INVERT LOWER BIT
006.346  062 002 040  2852          STA   IOWRK           SAVE NEW VALUE
006.351  302 357 006  2853          JNZ   SPEED5          IF TO DISPLAY "WORKING"
                      2854
006.354  041 100 007  2855          LXI   H,MSG.HSS       POINT TO "HOME", "SPACES", AND SPEED MSG
006.357  315 100 006  2856  SPEED5  CALL  TYPMSG          OUTPUT MESSAGE
006.362  341          2857          POP   H               GET TEST RESULT
006.363  315 305 005  2858          CALL  TOA.            OUTPUT RESULT TO CONSOLE
006.366  303 257 006  2859          JMP   SPEED1          PERFORM ANOTHER SAMPLE
```

```
                                2861  **      MSG.SPD - SPEED TEST MESSAGE
                                2862  *
                                2863  *       "       Disk drive rotational speed test.
                                2864  *
                                2865  *
                                2866  *                       Drive speed = "
                                2867
006.371  033 105 012           2868  MSG.SPD DB      A.ESC,'E',A.LF
006.374  011 104 151           2869          DB      '       Disk drive rotational speed test.',A.CR,A.LF,A.LF
007.041  011 011 104           2870          DB      '               Drive speed = '
007.061  000                   2871          DB      0


                                2873  **      MSG.WRK - "WORKING" MESSAGE FOR SPEED TEST
                                2874  *
                                2875  *       DISPLAYS "WORKING" AT HOME POSITION AND RETURNS CURSOR TO SPEED =
                                2876
007.062  033 110               2877  MSG.WRK DB      A.ESC,'H'       CURSOR HOME
007.064  127 157 162           2878          DB      'Working'
007.073  033 131 043           2879          DB      A.ESC,'Y#>'     CURSOR ADDRESS OF SPEED = VALUE
007.077  000                   2880          DB      0               END MESSAGE


                                2882  **      MSG.HSS - BLANKS "WORKING" MESSAGE
                                2883  *
                                2884
007.100  033 110               2885  MSG.HSS DB      A.ESC,'H'       CURSOR HOME
007.102  040 040 040           2886          DB      '   '           BLANKS
007.111  033 131 043           2887          DB      A.ESC,'Y#>'     CURSOR ADDRESS OF SPEED = VALUE
007.115  000                   2888          DB      0               END MESSAGE
```

```
                        2891  **      DYMEM - DYNAMIC MEMORY TEST
                        2892  *
                        2893  *       DYMEM TESTS THE DYNAMIC MEMORY IN THE H88/H89 BY PLACING
                        2894  *       A KNOWN PATTERN IN EACH DYNAMIC MEMORY CELL AND THEN
                        2895  *       PERFORMING A READ, INCREMENT, READ SEQUENCE WITH A DELAY
                        2896  *       BETWEEN EACH PASS OF THE TEST
                        2897  *
                        2898  *
                        2899  *       ENTRY   NONE
                        2900  *
                        2901  *       EXIT    ON RESET
                        2902  *
                        2903  *       USES    A,B,C,D,E,H,L,F,A',F',IX,IY
                        2904
                        2905
007.116  076 000        2906  DYMEM   MVI     A,0             MAKE SURE CLOCK AND SINGLE STEP ARE OFF
007.120  323 362        2907          OUT     H88.CTL
                        2908
                        2909  *       DETERMINE END OF MEMORY
                        2910  *
                        2911
007.122  041 000 040    2912  DYMEM1  LXI     H,START
007.125  076 001        2913          MVI     A,1
007.127  066 000        2914  DYMEM2  MVI     M,0             SET RAM TO ZERO
007.131  064            2915          INR     M               SET MEMORY TO ONE
007.132  276            2916          CMP     M               SEE IF (A) = ((H,L))
007.133  040 003        2917          JR      NZ,DYMEM3       IF NOT EQUAL, THE END OF RAM HAS BEEN REACHED
                        2918
007.135  043            2919          INX     H               ELSE, POINT TO NEXT LOCATION IN RAM
007.136  030 367        2920          JR      DYMEM2
                        2921
                        2922
007.140  053            2923  DYMEM3  DCX     H               POINT TO LAST GOOD LOCATION
007.141  353            2924          XCHG                    PUT ENDING ADDRESS IN D,E
007.142  041 303 007    2925          LXI     H,MSG.RAM       OUTPUT ENDING ADDRESS
                        2926
                        2927  *       LD      IX,DY3.3        RETURN ADDRESS
007.145  335 041        2928          DB      MI.LDXA,MI.LDX8
007.147  201 011        2929          DW      DY3.3
                        2930
007.151  030 112        2931          JR      DYMSG
007.153  023            2932  DY3.7   INX     D               (D,E) = LAST BYTE OF RAM + 1
                        2933
                        2934  *       TEST MEMORY
                        2935  *
007.154  006 001        2936          MVI     B,1             (B) = CONTENTS OF RAM AFTER SIZING
007.156  041 231 003    2937          LXI     H,MSG.PAS       OUTPUT PASS MESSAGE
                        2938
                        2939  *       LD      IX,DYMEM4       RETURN ADDRESS
007.161  335 041        2940          DB      MI.LDXA,MI.LDX8
007.163  167 007        2941          DW      DYMEM4
                        2942
007.165  030 076        2943          JR      DYMSG
                        2944
007.167  041 000 040    2945  DYMEM4  LXI     H,START
007.172  176            2946  DYMEM5  MOV     A,M             READ CURRENT CONTENTS
```

```
007.173  270            2947          CMP   B              SEE IF CORRECT CONTENTS STILL REMAIN
007.174  302 307 000    2948          JNZ   DYMEM9         FAILURE, SEE IF AT END OF RAM
                        2949
007.177  074            2950          INR   A
007.200  167            2951          MOV   M,A            INCREMENT RAM
007.201  276            2952          CMP   M              SEE IF WRITE WAS SUCCESSFUL
007.202  302 307 000    2953          JNZ   DYMEM9
                        2954
007.205  043            2955          INX   H
007.206  175            2956          MOV   A,L            GET LSB AND TEST FOR REACHING END OF RAM
007.207  273            2957          CMP   E
007.210  040 360        2958          JR    NZ,DYMEM5      IF LSB NOT EQUAL
                        2959
007.212  174            2960          MOV   A,H            CHECK LSB
007.213  272            2961          CMP   D
007.214  040 354        2962          JR    NZ,DYMEM5
                        2963
                        2964  *       HAVE REACHED END OF MEMORY!
                        2965  *       OUTPUT LAST VALUE TESTED
                        2966  *
                        2967
007.216  303 336 016    2968          JMP   DYMM5          HOW MANY TO BACK SPACE?
                        2969
007.221                 2970  DYME5.5 EQU   *
                        2971
                        2972  *       LD    IY,DY5.53      RETURN ADDRESS
007.221  375 041        2973          DB    MI.LDYA,MI.LDYB
007.223  230 007        2974          DW    DY5.53
                        2975
007.225  303 143 003    2976          JMP   DYASC
                        2977
007.230  045            2978  DY5.53  DCR   H
007.231  040 366        2979          JR    NZ,DYME5.5
                        2980
007.233  004            2981          INR   B              SHOW NEXT PASS VALUE
007.234  170            2982          MOV   A,B            VALUE TESTED
                        2983
                        2984  *       LD    IX,DYMEM6      RETURN ADDRESS
007.235  335 041        2985          DB    MI.LDXA,MI.LDXB
007.237  273 000        2986          DW    DYMEM6
                        2987
007.241  303 160 003    2988          JMP   DYBYT
                        2989
                        2990
                        2991  **      !!THE DYNAMIC RAM TEST CONTINUES ELSEWARE!!      **
                        2992  *       !!AND THEN RETURNS TO HERE!!!!!!!!!!!!!!!!!      **
                        2993
                        2994
007.244  041 000 000    2995  DY10.5  LXI   H,0            DELAY AND DING BELL AGAIN
007.247  006 002        2996          MVI   B,2            2 LOOPS
007.251  045            2997  DYMEM11 DCR   H
007.252  040 375        2998          JR    NZ,DYMEM11
                        2999
007.254  055            3000          DCR   L
007.255  040 372        3001          JR    NZ,DYMEM11
                        3002
```

```
007.257  005              3003           DCR     B
007.260  040 367          3004           JR      NZ,DYMEM11
                           3005
007.262  303 252 013      3006           JMP     DYMEM10         AGAIN


                           3008   **      DYMSG - DYNAMIC RAM TEST MESSAGE OUTPUT ROUTINE
                           3009   *
                           3010   *       ENTRY   (H,L) = MESSAGE ADDRESS
                           3011   *               (IX) = RETURN ADDRESS
                           3012   *
                           3013   *       EXIT    TO (IX)
                           3014   *
                           3015   *       USES    A,H,L,F,IY
                           3016
                           3017
007.265  176              3018   DYMSG   MOV     A,M             GET MESSAGE BYTE
                           3019
                           3020   *       LD      IY,DYMSG.5      RETURN ADDRESS
007.266  375 041          3021           DB      MI.LDYA,MI.LDYB
007.270  275 007          3022           DW      DYMSG.5
                           3023
007.272  303 143 003      3024           JMP     DYASC           OUTPUT ASCII
                           3025
007.275  267              3026   DYMSG.5 ORA     A               SEE IF NULL TO END STRING
007.276  043              3027           INX     H               POINT TO NEXT CHARACTER
007.277  040 364          3028           JR      NZ,DYMSG        IF NOT DONE YET
                           3029
                           3030   *       JP      (IX)            RETURN TO CALLER
007.301  335 351          3031           DB      MI.JIXA,MI.JIXB


                           3033   **      MSG.RAM - RAM TEST MESSAGE
                           3034   *
                           3035
007.303  033 105          3036   MSG.RAM DB      A.ESC,'E'
007.305  104 171 156      3037           DB      'Dynamic RAM test'
007.325  015 012 012      3038           DB      A.CR,A.LF,A.LF
007.330  011 040 114      3039           DB      '       LWA = '
007.340  000              3040           DB      0


                           3042   **      MSG.EQ - EQUALS MESSAGE
                           3043   *
                           3044
007.341  040 075 040      3045   MSG.EQ  DB      ' = '
007.344  000              3046           DB      0
                           3047
007.345  107 101 103      3048           DB      'GAC.'
```

```
                        3050  **      VIEW3A - *VIEW* CONTINUED
                        3051  *
                        3052
007.351  315 032 011   3053  VIEW3A  CALL    VIEW8           GET BOUNDRIES
007.354  353           3054          XCHG
007.355  315 111 011   3055  VIEW3A. CALL    VIEW12          PRINT CRLF AND ADDRESS
007.360  303 055 001   3056          JMP     VIEW1           AND START NEXT LINE
                        3057
007.363  174           3058  VIEW4   MOV     A,H
007.364  270           3059          CMP     B               COMPARE BC AND DE
007.365  300           3060          RNZ
007.366  175           3061          MOV     A,L
007.367  271           3062          CMP     C
007.370  311           3063          RET
```

```
                        3066  **      ENTRY POINT FOR FLOPPY DISK ROTATIONAL SPEED TEST
                        3067  *
000.001                 3068          ERRMI   10000A-6-*       MUST BE SIX BYTES BEFORE END
007.372                 3069          ORG     10000A-6
                        3070
007.372 303 240 006     3071  ESPEED  JMP     SPEED



                        3073  **      ENTRY POINT FOR DYNAMIC MEMORY TEST
                        3074  *
000.000                 3075          ERRNZ   10000A-3-*       MUST BE THREE BYTES BEFORE END
                        3076
007.375 303 032 016     3077  EDYMEM  JMP     MEMORY.



                        3079  **      Z47X - EXTENSION TO Z47 ROUTINE
                        3080  *
                        3081
010.000 315 063 006     3082  Z47X    CALL    OUT.            SEND RESET COMMAND
                        3083
010.003 315 104 010     3084          CALL    WDN             WAIT FOR HIM TO WAKE UP
010.006 332 171 002     3085          JC      NODEV           ERROR WAITING FOR DONE
                        3086
000.001                 3087          IF      .DEBUG
                        3088          MVI     A,1             FLAG PAST THE RESET
                        3089          STA     DBFLG
                        3090          ENDIF
                        3091
010.011 315 142 010     3092  Z47X.   CALL    RRDY
010.014 332 171 002     3093          JC      NODEV
010.017 315 142 010     3094          CALL    RRDY
010.022 332 171 002     3095          JC      NODEV
                        3096
000.001                 3097          IF      .DEBUG
                        3098          MVI     A,2
                        3099          STA     DBFLG           FLAG THRU RRDY
                        3100          ENDIF
                        3101
010.025 072 061 041     3102          LDA     AIO.UNI         (A)=UNIT NUMBER
010.030 107             3103          MOV     B,A
010.031 257             3104          XRA     A
010.032 315 155 012     3105          CALL    BITS            SET UNIT BIT MASK
010.035 245             3106          ANA     L               CHECK AGAINST READY FLAGS
010.036 040 351         3107          JR      NZ,Z47X.
                        3108
000.001                 3109          IF      .DEBUG
                        3110          MVI     A,8
                        3111          STA     DBFLG           FLAG PAST READ
                        3112          ENDIF
                        3113
010.040 076 002         3114          MVI     A,DD.RAS
010.042 315 027 006     3115          CALL    COM             READ AUX STAT
```

```
010.045 171             3116            MOV     A,C
010.046 315 023 006     3117            CALL    DAT
010.051 315 067 001     3118            CALL    PIN
010.054 332 171 002     3119            JC      NODEV           PREMATURE DONE
                        3120
000.001                 3121            IF      .DEBUG
                        3122            MVI     A,9
                        3123            STA     DBFLG           FLAG PAST RAS
                        3124            ENDIF
                        3125
                        3126    *       SET TRANSFER COUNT TO 9 SECTORS
                        3127
010.057 076 003         3128            MVI     A,DD.LSC
010.061 315 027 006     3129            CALL    COM             SEND 'LOAD COUNT'
                        3130
010.064 257             3131            XRA     A
010.065 315 023 006     3132            CALL    DAT             SEND HIGH ORDER BYTE
                        3133
010.070 076 012         3134            MVI     A,10
010.072 315 023 006     3135            CALL    DAT             SEND LOW ORDER BYTE
                        3136
010.075 315 104 010     3137            CALL    WDN             WAIT FOR DONE, THEN EXIT
010.100 332 171 002     3138            JC      NODEV
                        3139
010.103 311             3140            RET



                        3142    **      WDN     - WAIT FOR DONE
                        3143    *
                        3144    *       WDN waits for the done bit to be set.
                        3145    *
                        3146    *       time-out is in effect at this point
                        3147    *
                        3148    *       ENTRY:  NONE
                        3149    *
                        3150    *       EXIT:   PSW     'C' SET IF ERROR
                        3151    *                       'C' CLEAR IF DONE
                        3152    *
                        3153    *       USES:   PSW
                        3154    *
                        3155
010.104 363             3156    WDN     DI
010.105 305             3157            PUSH    B               SAVE BC
010.106 001 000 175     3158            LXI     B,WDNA
                        3159
010.111 013             3160    WDN1    DCX     B
010.112 170             3161            MOV     A,B
010.113 261             3162            ORA     C               IF TIMED-OUT
010.114 067             3163            STC
010.115 050 020         3164            JR      Z,WDN2
                        3165
010.117 315 170 006     3166            CALL    IN.
010.122 346 040         3167            ANI     S.DON
010.124 050 363         3168            JR      Z,WDN1          IF NOT DONE YET
```

```
                         3169
010.126  315 170 006    3170          CALL    IN.             S.ERR VALID ONLY IF S.DON SET
010.131  346 001        3171          ANI     S.ERR
010.133  067            3172          STC
010.134  040 001        3173          JR      NZ,MDN2         IF ERROR BIT SET
                         3174
010.136  247            3175          ANA     A               CLEAR CARRY
                         3176
010.137  301            3177  MDN2    POP     B
010.140  373            3178          EI
010.141  311            3179          RET                     ALL OK.
                         3180
175.000                  3181  MDNA    EQU     32000           TIME OUT COUNTER


                         3183  **      RRDY    - CHECK DEVICE READY
                         3184  *
                         3185  *       RRDY RETURNS THE DEVICE READY BITS IN
                         3186  *       THE L REGISTER. BITS 'ON' INDICATE
                         3187  *       UNIT NOT READY.
                         3188  *
                         3189
010.142  076 020        3190  RRDY    MVI     A,DD.RRDY
010.144  315 027 006    3191          CALL    COM
                         3192
010.147  315 067 001    3193          CALL    PIN
                         3194
000.001                  3195          IF      .DEBUG
                         3196          MVI     A,3             FLAG PAST PIN
                         3197          STA     DBFLG
                         3198          ENDIF
                         3199
010.152  157            3200          MOV     L,A
010.153  303 104 010    3201          JMP     MDN


                         3203  **      COM2 - *COM* ROUTINE CONTINUATION
                         3204  *
                         3205  *       OUTPUT COMMAND TO 47 AND THEN DELAY
                         3206  *
                         3207
010.156  315 140 006    3208  COM2    CALL    OUT1.           SEND COMMAND BYTE
010.161  076 040        3209          MVI     A,40Q
010.163  247            3210          ANA     A               CLEARY 'Z'
                         3211
010.164  075            3212  COM3    DCR     A
010.165  302 164 010    3213          JNZ     COM3            SHORT DELAY
                         3214
010.170  311            3215          RET
```

```
                                3217  **      VIEW5 - *VIEW* CONTINUED
                                3218  *
                                3219  *       VIEW5 DOES THE ASCII PORTION OF THE *VIEW* ROUTINE
                                3220  *
                                3221
010.171 315 316 010             3222  VIEW5   CALL    PCFA            PUSITION CURSOR FOR ASCII
010.174 176                     3223  VIEW5A  MOV     A,M             GET A BYTE
010.175 247                     3224          ANA     A               CHECK PARITY
010.176 372 241 010             3225          JM      VIEW7
010.201 376 177                 3226          CPI     177Q
010.203 312 212 010             3227          JZ      VIEW5.          IF DELETE
010.206 376 040                 3228          CPI     ' '             PRINTABLE?
010.210 060 012                 3229          JR      NC,VIEW6        YES
010.212 345                     3230  VIEW5.  PUSH    H
010.213 041 024 011             3231          LXI     H,VEW.NPC       NON-PRINTABLE CHARACTER
010.216 315 100 006             3232          CALL    TYPMSG
010.221 341                     3233          POP     H
010.222 030 003                 3234          JR      VIEW6.
010.224 315 302 003             3235  VIEW6   CALL    WCC             PRINT IT
010.227 315 363 007             3236  VIEW6.  CALL    VIEW4
010.232 310                     3237          RZ                      IF LAST BYTE DONE
010.233 315 340 003             3238          CALL    VIEW3.          CHECK FOR END
010.236 040 334                 3239          JR      NZ,VIEW5A       NO, DO MORE
010.240 311                     3240          RET
                                3241
010.241 346 177                 3242  VIEW7   ANI     177Q            STRIP PARITY
010.243 365                     3243          PUSH    PSW             SAVE IT
010.244 076 033                 3244          MVI     A,33Q
010.246 315 302 003             3245          CALL    WCC
010.251 076 160                 3246          MVI     A,'p'
010.253 315 302 003             3247          CALL    WCC             GO TO REVERSE VIDEO
010.256 361                     3248          POP     PSW
010.257 376 177                 3249          CPI     177Q
010.261 312 270 010             3250          JZ      VIEW7A
010.264 376 040                 3251          CPI     ' '
010.266 060 012                 3252          JR      NC,VIEW7.
010.270 345                     3253  VIEW7A  PUSH    H
010.271 041 024 011             3254          LXI     H,VEW.NPC
010.274 315 100 006             3255          CALL    TYPMSG
010.277 341                     3256          POP     H
010.300 030 003                 3257          JR      VIEW7..
010.302 315 302 003             3258  VIEW7.  CALL    WCC             PRINT IT
010.305 076 033                 3259  VIEW7.. MVI     A,33Q
010.307 315 302 003             3260          CALL    WCC
010.312 076 161                 3261          MVI     A,'q'           EXIT REVERSE VIDEO
010.314 030 306                 3262          JR      VIEW6           AND FINISH UP
                                3263
010.316 345                     3264  PCFA    PUSH    H
010.317 315 112 015             3265          CALL    CHKRAD
010.322 041 017 011             3266          LXI     H,PCF.MO        ASSUME OCTAL
010.325 312 333 010             3267          JZ      PCFAA
010.330 041 012 011             3268          LXI     H,PCF.MH        WAS HEX
010.333 315 100 006             3269  PCFAA   CALL    TYPMSG
010.336 341                     3270          POP     H
010.337 076 001                 3271          MVI     A,1             Skip 1 space per letter
                                3272
```

```
010.341  345            3273  PCFA.   PUSH   H
010.342  305            3274          PUSH   B
010.343  107            3275          MOV    B,A              B=SKIP COUNT
010.344  315 112 015     3276          CALL   CHKRAD
010.347  312 357 010     3277          JZ     PCFA1
                         3278
010.352  076 360         3279          MVI    A,11110000B      MASK FOR HEX
010.354  303 361 010     3280          JMP    PCFA2
                         3281
010.357  076 370         3282  PCFA1   MVI    A,11111000B      MASK FOR OCTAL
                         3283
010.361  245            3284  PCFA2   ANA    L                MASK LOW ORDER, RESULT IN A
                         3285
010.362  275            3286  PCFA3   CMP    L
010.363  312 007 011     3287          JZ     PCFA4            IF A=L, DONE
010.366  055            3288          DCR    L
010.367  365            3289          PUSH   PSW
010.370  305            3290          PUSH   B
010.371  076 040         3291  PCFA3.  MVI    A,' '
010.373  315 302 003     3292          CALL   WCC
010.376  005            3293          DCR    B
010.377  302 371 010     3294          JNZ    PCFA3.
                         3295
011.002  301            3296          POP    B
011.003  361            3297          POP    PSW
011.004  303 362 010     3298          JMP    PCFA3            PRINT (B) SPACES AND CHECK AGAIN
                         3299
011.007  301            3300  PCFA4   POP    B
011.010  341            3301          POP    H
011.011  311            3302          RET
                         3303
011.012  033 131 001     3304  PCF.MH  DB     33Q,'Y',1,54+31,0     Hex Version
011.017  033 131 001     3305  PCF.MO  DB     33Q,'Y',1,40+31,0     Octal Version
                         3306
011.024  033 106 136     3307  VEW.NPC DB     33Q,'F^',33Q,'G',0    ESC,GRAPHICS,1,ESC,NO-GRAPHICS



                         3309  **      VIEW8 - GET BOUNDRIES
                         3310  *
                         3311  *       VIEW8 GETS THE BOUNDRIES OF THE *VIEW* COMMAND
                         3312  *
                         3313
011.032  315 012 015     3314  VIEW8   CALL   IROC             GET CHARACTER OR RETURN
011.035  040 012         3315          JR     NZ,VIEW8A
                         3316
011.037  052 072 040     3317          LHLD   VEWHLD           GET LAST ON
011.042  043            3318          INX    H                START AT NEXT ONE
011.043  353            3319          XCHG
011.044  001 000 000     3320          LXI    B,0              SET LENGTH TO 0
011.047  030 003         3321          JR     VIEW8B
                         3322
011.051  315 073 016     3323  VIEW8A  CALL   GET8ND.          'C' IS SET, FIRST CHARACTER IN A
                         3324
011.054  151            3325  VIEW8B  MOV    L,C
```

```
011.055 140             3326            MOV     H,B
011.056 042 072 040     3327            SHLD    VEWHLD          SAVE LAST
                        3328
011.061 170             3329            MOV     A,B
011.062 261             3330            ORA     C               LAST = 0
011.063 300             3331            RNZ                     NO, OK
011.064 041 177 000     3332            LXI     H,200Q-1        ADD 177Q TO VALUE
011.067 031             3333            DAD     D               HL = DE + 177Q
                        3334
011.070 315 112 015     3335            CALL    CHKRAD
011.073 050 006         3336            JR      Z,VIEW8.
                        3337
011.075 325             3338            PUSH    D
011.076 021 200 000     3339            LXI     D,200Q
011.101 031             3340            DAD     D               ADD IN ANOTHER
011.102 321             3341            POP     D
                        3342
011.103 042 072 040     3343    VIEW8.  SHLD    VEWHLD          UPDATE END ADDRESS
011.106 115             3344            MOV     C,L
011.107 104             3345            MOV     B,H
011.110 311             3346            RET


                        3348    **      VIEW12 - Print address and position cursor
                        3349    *
                        3350
011.111                 3351    VIEW12  EQU     *
011.111 315 064 015     3352            CALL    TOA
011.114 315 112 015     3353            CALL    CHKRAD
011.117 312 127 011     3354            JZ      VIEW3B
011.122 076 003         3355            MVI     A,3                     NUMBER OF ASCII FOR BYTES
011.124 303 131 011     3356            JMP     VIEW3C
011.127 076 004         3357    VIEW3B  MVI     A,4
011.131 315 341 010     3358    VIEW3C  CALL    PCFA.                   SKIP TO START ON SCREEN
011.134 311             3359            RET


                        3361    **      CKAUTO - CHECK FOR AUTO BOOT
                        3362    *
                        3363    *       CKAUTO IS ENTERED DURING THE MONITOR LOOP TO CHECK
                        3364    *       IF THE AUTO BOOT SWITCH IS SET.
                        3365    *
                        3366    *
                        3367    *       THIS ROUTINE WAS MOVED FROM UP FRONT TO MAKE ROOM
                        3368    *
                        3369
011.135 333 362         3370    CKAUTO  IN      H88.SW
011.137 346 200         3371            ANI     H88S.AT         CHECK SWITCH
011.141 050 007         3372            JR      Z,CHAT2         NOT AUTO BOOT
011.143 041 123 041     3373            LXI     H,AUTO8
011.146 276             3374            CMP     M               HAVE WE BEEN HERE BEFORE?
011.147 302 243 004     3375            JNZ     AT8             NO, DO AUTO BOOT
```

```
                          3376
011.152  041 062 014     3377  CHAT2   LXI     H,MSG.PR
011.155  303 354 000     3378          JMP     MTR.15          RETURN TO MOITOR LOOP



                          3380  **      DYMEM EXTENSION
                          3381  *
011.160  353              3382  DY9.3   XCHG
011.161  174              3383          MOV     A,H
                          3384  *       LD      IX,DY9.4
011.162  335 041          3385          DB      MI.LDXA,MI.LDXB
011.164  171 011          3386          DW      DY9.4
                          3387
011.166  303 160 003      3388          JMP     DY8YT
                          3389
011.171  175              3390  DY9.4   MOV     A,L
                          3391
                          3392  *       LD      IX,DY9.5
011.172  335 041          3393          DB      MI.LDXA,MI.LDXB
011.174  315 003          3394          DW      DY9.5
                          3395
011.176  303 160 003      3396          JMP     DY8YT
                          3397
                          3398  *       ANOTHER EXTENSION!
                          3399
011.201  172              3400  DY3.3   MOV     A,D
                          3401  *       LD      IX,DY3.5
011.202  335 041          3402          DB      MI.LDXA,MI.LDXB
011.204  211 011          3403          DW      DY3.5
                          3404
011.206  303 160 003      3405          JMP     DY8YT
                          3406
011.211  173              3407  DY3.5   MOV     A,E
                          3408  *       LD      IX,DY3.7
011.212  335 041          3409          DB      MI.LDXA,MI.LDXB
011.214  153 007          3410          DW      DY3.7
011.216  303 160 003      3411          JMP     DY8YT



                          3413  **      H37 - ENTRY POINT TO BOOT FROM H37
                          3414  *
                          3415
011.221  257              3416  H37     XRA     A
011.222  323 171          3417          OUT     DK.INT          SET FLIP LATCH
                          3418
011.224  076 320          3419          MVI     A,FDC.FI
011.226  323 172          3420          OUT     FD.CMD          SET NOT BUSY
                          3421
011.230  076 001          3422          MVI     A,1
011.232  315 053 000      3423          CALL    DLY             DLY 2 MILLISECONDS
                          3424
011.235  333 172          3425          IN      FD.STAT         CLEAR INTERRUPTS
```

```
                                3426
011.237  041 145 012   3427            LXI     H,MYINT
011.242  042 051 040   3428            SHLD    UIVEC+9+1       SET INTERRUPT RUTINE
011.245  076 303       3429            MVI     A,MI.JMP
011.247  062 050 040   3430            STA     UIVEC+9
                                3431
011.252  072 061 041   3432            LDA     AIO.UNI
011.255  306 004       3433            ADI     4
011.257  107           3434            MOV     B,A
011.260  257           3435            XRA     A
011.261  315 155 012   3436            CALL    BITS            GET DEVICE CODE
                                3437
011.264  366 015       3438            ORI     CON.MO+CON.EI+CON.MFM
011.266  323 170       3439            OUT     DK.CON
011.270  107           3440            MOV     B,A
011.271  305           3441            PUSH    B
                                3442
011.272  373           3443            EI                      insure interrupts on
                                3444
011.273  076 226       3445            MVI     A,150           300MS ON DELAY
011.275  315 053 000   3446            CALL    DLY
                                3447
011.300  041 335 011   3448            LXI     H,H371
011.303  042 067 040   3449            SHLD    BLKICW          SET RETURN ADDRESS
011.306  076 003       3450            MVI     A,FDC.RST+FDF.S30
011.310  323 172       3451            OUT     FD.CMD
                                3452
011.312  001 377 377   3453            LXI     B,-1            ABOUT 5 SECONDS
011.315  026 004       3454            MVI     D,4               DOUBLED.
011.317  013           3455  H37.      DCX     B
011.320  170           3456            MOV     A,B
011.321  261           3457            ORA     C
011.322  040 373       3458            JR      NZ,H37.         IF BC>0
                                3459
011.324  025           3460            DCR     D
011.325  040 370       3461            JR      NZ,H37.         IF D>0
                                3462
011.327  076 320       3463            MVI     A,FDC.FI
011.331  323 172       3464            OUT     FD.CMD
                                3465
011.333  030 132       3466            JR      H373            TIMED OUT
                                3467
011.335  041 356 011   3468  H371      LXI     H,H371B
011.340  042 067 040   3469            SHLD    BLKICW          LOOP RETURN ADDRESS
011.343  076 012       3470            MVI     A,10            NUMBER OF TRACKS TO STEP
011.345  323 173       3471            OUT     FD.DAT          SET TRACK NUMBER TO 10
011.347  076 023       3472            MVI     A,FDC.SEK+FDF.S30
011.351  323 172       3473            OUT     FD.CMD
011.353  303 353 011   3474            JMP     *               Wait for interrupt
                                3475
                                3476  *         Return here after doing seek
                                3477
011.356  041 373 011   3478  H371B     LXI     H,H371C
011.361  042 067 040   3479            SHLD    BLKICW
011.364  076 003       3480            MVI     A,FDC.RST+FDF.S30
011.366  323 172       3481            OUT     FD.CMD
```

```
011.370  303 370 011  3482          JMP     *
                      3483
                      3484  *       Here after final RESTORE
                      3485
011.373  346 004      3486  H371C   ANI     FDS.TK0         Be sure track zero switch on
011.375  050 070      3487          JR      Z,H373          If not there
                      3488
                      3489  *       Over track zero, Wait for head to settle
                      3490
011.377  001 200 014  3491          LXI     B,3200          40 mS DELAY
012.002  013          3492  H371.   DCX     B
012.003  170          3493          MOV     A,B
012.004  261          3494          ORA     C
012.005  040 373      3495          JR      NZ,H371.            ALLOW HEAD SETTLE TIME
                      3496
012.007  301          3497          POP     B
012.010  170          3498          MOV     A,B             (A) = Device Control Bits
012.011  366 002      3499          ORI     CON.DRQ         Turn on DRQ interrupt
012.013  107          3500          MOV     B,A
012.014  305          3501          PUSH    B               save device control bits
012.015  323 170      3502          OUT     DK.CON          READY FOR TRANSFERS
                      3503
012.017  315 075 012  3504          CALL    READT           Read a track
012.022  301          3505          POP     B
012.023  365          3506          PUSH    PSW             SAVE RETURN STATUS
012.024  170          3507          MOV     A,B
012.025  346 373      3508          ANI     377Q-CON.MFM    OFF DBL DENSITY
012.027  107          3509          MOV     B,A
012.030  361          3510          POP     PSW
012.031  040 012      3511          JR      NZ,H372             IF READ FAILURE
                      3512
012.033  041 200 335  3513          LXI     H,-USERFWA
012.036  031          3514          DAD     D                   HL = Bytes Read
012.037  174          3515          MOV     A,H
012.040  376 011      3516          CPI     2048+256/256        See if 2.25K
012.042  322 201 016  3517          JNC     EUC             If got it all
                      3518
012.045  170          3519  H372    MOV     A,B
012.046  323 170      3520          OUT     DK.CON
                      3521
012.050  315 075 012  3522          CALL    READT           TRY SINGLE DENSITY
012.053  040 012      3523          JR      NZ,H373             IF FAILURE
                      3524
012.055  041 200 335  3525          LXI     H,-USERFWA
012.060  031          3526          DAD     D                   HL = Bytes Read
012.061  174          3527          MOV     A,H
012.062  376 011      3528          CPI     2048+256/256        See if 2.25K
012.064  322 201 016  3529          JNC     EUC                 More than 2.25K read, is ok
                      3530
012.067  257          3531  H373    XRA     A
012.070  323 170      3532          OUT     DK.CON          TURN OFF DEVICE
012.072  303 171 002  3533          JMP     NODEV
                      3534
012.075  076 001      3535  READT   MVI     A,CON.ST
012.077  323 171      3536          OUT     DK.INT
012.101  323 172      3537          OUT     FD.SEC
```

```
000.000                  3538          ERRNZ   CON.ST-1
012.103  257            3539          XRA     A
012.104  323 171        3540          OUT     DK.INT
000.000                  3541          ERRNZ   CON.CD
                         3542
012.106  041 134 012     3543          LXI     H,READT2
012.111  042 067 040     3544          SHLD    BLKICW          SET RETURN ADDRESS
012.114  041 126 012     3545          LXI     H,READT1
012.117  021 200 042     3546          LXI     D,USERFWA
012.122  076 234         3547          MVI     A,FDC.RDS+FDF.DLF+FDF.MRF+FDF.SLF
012.124  323 172         3548          OUT     FD.CMD
                         3549
012.126  166            3550  READT1  HLT
012.127  333 173        3551          IN      FD.DAT          *TIME DEPENDENT*
012.131  022            3552          STAX    D
012.132  023            3553          INX     D
012.133  351            3554          PCHL
                         3555
012.134  365            3556  READT2  PUSH    PSW
012.135  076 010        3557          MVI     A,CON.MO
012.137  323 170        3558          OUT     DK.CON
012.141  361            3559          POP     PSW
012.142  346 254        3560          ANI     FDS.NRD+FDS.LOT+FDS.CRC+FDS.RTE
012.144  311            3561          RET
                         3562
                         3563  ***     MYINT - H37 Interrupt Routine
                         3564  *
                         3565  *       This routine is entered when a level 4 interrupt
                         3566  *       is recieved from the H37 Hardware.
                         3567  *
                         3568  *       Control is passed to the address in BLKICW
                         3569  *
                         3570  *       ENTRY:  NONE    (From disk routine via level 4 interrupt)
                         3571  *
                         3572  *       EXIT:   PSW     =       Status byte from controller
                         3573  *               HL      =       Return address to routine
                         3574  *
                         3575
012.145  333 172        3576  MYINT   IN      FD.STAT
012.147  341            3577          POP     H
012.150  052 067 040     3578          LHLD    BLKICW
012.153  373            3579          EI
012.154  351            3580          PCHL
012.155                  3581          XTEXT   BITS


                         3583X **      BITS    -  BIT SET
                         3584X *
                         3585X *       BITS SETS THE SPECIFIED BIT IN THE ACCUMULATOR.
                         3586X *
                         3587X *       ENTRY:  A       = ORIGINAL  A
                         3588X *               B       = NUMBER OF BIT TO SET ( 7=HIGH,...,0=LOW )
                         3589X *
                         3590X *       EXIT:   A       = ORIGINAL  A  WITH  BIT(B) SET
```

```
                        3591X *
                        3592X *         USES:   PSW
                        3593X *
                        3594X
012.155  305            3595X BITS      PUSH    B
                        3596X
012.156  365            3597X           PUSH    PSW
012.157  076 200        3598X           MVI     A,10000000B
012.161  004            3599X           INR     B
012.162  007            3600X BITS1     RLC
012.163  005            3601X           DCR     B
012.164  302 162 012    3602X           JNZ     BITS1
                        3603X
012.167  117            3604X           MOV     C,A
012.170  361            3605X           POP     PSW
012.171  261            3606X           ORA     C
                        3607X
012.172  301            3608X           POP     BC
012.173  311            3609X           RET
                        3610  **        H67 - BOOT H67
                        3611  *
                        3612  *         The section of this code most likely to 'HANG' because
                        3613  *         of no controller is timed using the BC register pair
                        3614  *         for approximately 3 seconds.
                        3615  *
                        3616
012.174  076 020        3617  H67       MVI     A,BC.RST
012.176  315 140 006    3618            CALL    OUT1.           RESET THE CONTROLLER
                        3619
012.201  076 004        3620            MVI     A,4
012.203  315 053 000    3621            CALL    DLY
                        3622
012.206  041 062 041    3623            LXI     H,AIO.DIR       SCRATCH AREA FOR CDB
012.211  066 000        3624            MVI     M,D.TDR         TEST FOR READY
012.213  016 005        3625            MVI     C,5
012.215  043            3626  H671      INX     H
012.216  066 000        3627            MVI     M,0             FILL CDB WITH 0
012.220  015            3628            DCR     C
012.221  040 372        3629            JR      NZ,H671
                        3630
012.223  315 365 012    3631            CALL    H67UNI          GET UNIT NUMBER
012.226  062 063 041    3632            STA     AIO.DIR+1       SET THE LUN
                        3633
012.231  315 376 012    3634  H671.     CALL    GETCON          CHECK READY
012.234  060 012        3635            JR      NC,H672         IF DRIVE IS READY
                        3636
012.236  312 171 002    3637            JZ      NODEV           IF WAS TIME-OUT PROBLEM
                        3638
012.241  076 377        3639            MVI     A,377Q
012.243  315 053 000    3640            CALL    DLY             WAIT ABOUT 1/2 SECOND
012.246  030 361        3641            JR      H671.
                        3642
012.250  041 062 041    3643  H672      LXI     H,AIO.DIR
012.253  066 001        3644            MVI     M,D.REC         RECAL THE DRIVE
                        3645
012.255  315 376 012    3646            CALL    GETCON          DO THE RECAL
```

```
012.260  332 171 002  3647           JC      NODEV           ERROR IN RECAL
                      3648
                      3649  *        Now cause the drive to step out 10 tracks
                      3650
012.263  072 061 041  3651           LDA     AIO.UNI         Only for the hard disk
012.266  247         3652           ANA     A
012.267  302 330 012  3653           JNZ     H673            If unit = 1, is 8" floppy
                      3654
012.272  041 062 041  3655           LXI     H,AIO.DIR
012.275  066 013      3656           MVI     M,D.SEK
012.277  043         3657           INX     H               HL over logical address 0
012.300  043         3658           INX     H
012.301  066 007      3659           MVI     M,7             Seek block (7*256)
                      3660
012.303  315 376 012  3661           CALL    GETCON          Do the seek
012.306  332 171 002  3662           JC      NODEV           If error doing Seek
                      3663
012.311  041 062 041  3664           LXI     H,AIO.DIR
012.314  066 001      3665           MVI     M,D.REC
012.316  043         3666           INX     H
012.317  043         3667           INX     H
012.320  066 000      3668           MVI     M,0             Do another Recal
012.322  315 376 012  3669           CALL    GETCON
012.325  332 171 002  3670           JC      NODEV
                      3671
012.330  041 062 041  3672  H673     LXI     H,AIO.DIR       SET UP READ COMMAND
012.333  066 010      3673           MVI     M,D.REA
012.335  043         3674           INX     H               HL = LUN
012.336  043         3675           INX     H
012.337  043         3676           INX     H
012.340  043         3677           INX     H
012.341  066 012      3678           MVI     M,10            SET 10 SECTOR READ
012.343  043         3679           INX     H
012.344  066 200      3680           MVI     M,080H          CONTROL BYTE
                      3681
012.346  315 365 012  3682           CALL    H67UNI
012.351  062 063 041  3683           STA     AIO.DIR+1       SET LUN TO READ
                      3684
012.354  315 376 012  3685           CALL    GETCON
012.357  332 171 002  3686           JC      NODEV           IF READ ERROR
012.362  303 201 016  3687           JMP     EUC             ENTER USER CODE
                      3688
012.365  072 061 041  3689  H67UNI   LDA     AIO.UNI         (A)=UNIT NUMBER
012.370  017         3690           RRC
012.371  017         3691           RRC
012.372  017         3692           RRC                     MOVE IT INTO PLACE
012.373  346 140      3693           ANI     ST.LUN
012.375  311         3694           RET
                      3695
012.376  363         3696  GETCON   DI                      GET CONTROLLER ATTENTION
                      3697
012.377  001 377 377  3698           LXI     B,65535         ABOUT 5 SECONDS FOR RESPONSE
013.002  026 002      3699           MVI     D,2             3 BYTE COUNTER (D,B,C)
                      3700
013.004  315 150 006  3701  GTCON    CALL    IN1.            GET BUSS STATUS
013.007  346 010      3702           ANI     BS.BSY
```

```
013.011  050 012        3703           JR      Z,GTCON1          WAIT FOR BUSY TO LEAVE
                        3704
013.013  013            3705           DCX     B                 COUNT DOWN
013.014  170            3706           MOV     A,B
013.015  261            3707           ORA     C
013.016  040 364        3708           JR      NZ,GTCON          NO TIMEOUT YET
                        3709
013.020  025            3710           DCR     D
013.021  040 361        3711           JR      NZ,GTCON          DEC 3RD BYTE
                        3712
013.023  067            3713           STC                       INDICATE ERROR
013.024  311            3714           RET
                        3715
013.025  076 100        3716  GTCON1   MVI     A,BC.SEL
013.027  315 140 006    3717           CALL    OUT1.              OUTPUT TO (PRIM)
                        3718
013.032  315 150 006    3719  CBUSY    CALL    IN1.
013.035  346 010        3720           ANI     BS.BSY
013.037  040 007        3721           JR      NZ,CBUSY1            WAIT FOR CONTROLLER
                        3722
013.041  013            3723           DCX     B                 CONTINUE COUNTING
013.042  170            3724           MOV     A,B
013.043  261            3725           ORA     C
013.044  040 364        3726           JR      NZ,CBUSY
013.046  067            3727           STC
013.047  311            3728           RET                       TIMED OUT
                        3729
013.050  076 002        3730  CBUSY1   MVI     A,BC.EDT
013.052  315 140 006    3731           CALL    OUT1.
                        3732  *
                        3733  *        HAVE CONTROLLER, SEND HIM COMMAND
                        3734  *
013.055  041 062 041    3735  OUTCOM   LXI     H,AIO.DIR          FWA OF COMMAND BUFFER
                        3736
013.060  315 150 006    3737  COMREQ   CALL    IN1.
013.063  117            3738           MOV     C,A
013.064  247            3739           ANA     A
013.065  362 060 013    3740           JP      COMREQ
000.000                 3741           ERRNZ   200Q-BS.REQ         WAIT FOR REQUEST BIT
                        3742
013.070  346 020        3743           ANI     BS.COM
013.072  050 077        3744           JR      Z,TFDATA             COMMAND DONE, SEND DATA
                        3745
013.074  171            3746           MOV     A,C               (A)=BUSS STATUS BYTE
013.075  346 100        3747           ANI     BS.DTD            CHECK DIRECTION
013.077  050 007        3748           JR      Z,GETST             IF DONE, GET STATUS
                        3749
013.101  176            3750           MOV     A,M               (A)=COMMAND BYTE
013.102  315 063 006    3751           CALL    OUT.              SEND OUT DATA
013.105  043            3752           INX     H                 BUMP POINTER
013.106  030 350        3753           JR      COMREQ            CONTINUE SENDIG BYTES
                        3754  *
                        3755  *        GET STATUS - COMPLETION BYTE SHOULD BE ZEROS AND
                        3756  *        STATUS BYTE SHOULD BE ZERO IN 2 LS BITS
                        3757  *
013.110  315 150 006    3758  GETST    CALL    IN1.
```

```
013.113  346 320        3759           ANI   BS.REQ+BS.DTD+BS.COM
013.115  376 220        3760           CPI   BS.REQ+BS.COM
013.117  040 367        3761           JR    NZ,GETST                  WAIT FOR CONTROLLER
                        3762
013.121  315 170 006    3763           CALL  IN.                                        /2.1b/
013.124  117            3764           MOV   C,A                       (C)=STATUS BYTE
013.125  062 070 041    3765           STA   AIO.DIR+6
                        3766
013.130  315 150 006    3767  GETCPT   CALL  IN1.
013.133  107            3768           MOV   B,A
013.134  062 071 041    3769           STA   AIO.DIR+7
                        3770
013.137  346 340        3771           ANI   BS.REQ+BS.DTD+BS.MTY+BS.COM
013.141  376 240        3772           CPI   BS.REQ+BS.MTY+BS.COM
013.143  040 363        3773           JR    NZ,GETCPT                 WAIT FOR MESSAGE
013.145  062 072 041    3774           STA   AIO.DIR+8                 SAVE BYTES FOR DEBUG
                        3775
013.150  373            3776           EI
                        3777
013.151  315 170 006    3778           CALL  IN.                       (A)=COMPLETION BYTE
013.154  267            3779           ORA   A                         CHECK COMPLETION
013.155  067            3780           STC
013.156  300            3781           RNZ                             SHOULD BE ZERO
                        3782
013.157  171            3783           MOV   A,C
013.160  346 003        3784           ANI   00000011B                 CHECK FOR ERRORS
013.162  067            3785           STC
013.163  300            3786           RNZ                             IF A BIT IS SET
                        3787
013.164  170            3788           MOV   A,B
013.165  346 002        3789           ANI   000000108
013.167  067            3790           STC
013.170  300            3791           RNZ                             IF INTERFACE ERROR
                        3792
013.171  257            3793           XRA   A                         CLEAR CARRY
013.172  311            3794           RET
                        3795
013.173  041 200 042    3796  TFDATA   LXI   H,USERFWA                 HL = LOAD ADDRESS
                        3797
013.176  315 150 006    3798  TFREQ    CALL  IN1.
013.201  117            3799           MOV   C,A
013.202  346 200        3800           ANI   BS.REQ
013.204  050 370        3801           JR    Z,TFREQ                   WAIT FOR REQUEST
                        3802
013.206  171            3803           MOV   A,C
013.207  346 020        3804           ANI   BS.COM
013.211  040 275        3805           JR    NZ,GETST                  IF DONE, CHECK STATUS
                        3806
013.213  315 170 006    3807           CALL  IN.                       GET DATA BYTE
013.216  167            3808           MOV   M,A
013.217  043            3809           INX   H
013.220  030 354        3810           JR    TFREQ                     CONTINUE UNTIL DONE
```

```
                        3812  **      FEDEV - FUTURE EXPANSION DEVICE
                        3813  *
                        3814  *       CURRENTLY, FEDEV JUST PRINTS "UNKNOWN DEVICE"
                        3815  *
                        3816
013.222  041 233 013    3817  FEDEV   LXI     H,MSG.FE
013.225  315 100 006    3818          CALL    TYPMSG
013.230  303 177 002    3819          JMP     NODEV1          ENTER COMMON RECOVERY CODE
                        3820
013.233  077 125 156    3821  MSG.FE  DB      '?Unkown Device',0


                        3823  **      DYMEM10 - DYNAMIC RAM TEST CONTINUED
                        3824  *
                        3825
013.252  076 007        3826  DYMEM10 MVI     A,A.BEL
013.254  375 041        3827          DB      MI.LDYA,MI.LDYB
013.256  244 007        3828          DW      DY10.5
                        3829
013.260  303 143 003    3830          JMP     DYASC


                        3832  **      CCL - CHECK COMMAND LINE
                        3833  *
                        3834  *       CCL CHECKS TO SEE IF THE USER WISHES TO PASS A COMMAND
                        3835  *       TO THE BOOT ROUTINE. IF THE USER SIMPLY TYPES A CARRIAGE
                        3836  *       RETURN, THEN NO COMMAND LINE IS PRESENT AND (SP) = 42.200
                        3837  *       OTHERWISE THE COMMAND LINE IS PUSHED ONTO THE STACK ALA HDOS
                        3838  *       AND THE BOOT ROUTINES CAN DO WITH AS THEY SEE FIT.
                        3839  *
                        3840  *       ENTRY:  NONE
                        3841  *
                        3842  *       EXIT:   (SP)    =       42.200
                        3843  *                               NO COMMAND LINE
                        3844  *               (SP)    <>      42.200
                        3845  *                               COMMAND ON STACK TERMINATED WITH 000Q
                        3846  *
                        3847  *       USES:   SP
                        3848  *
                        3849
013.263  062 000 040    3850  CCL     STA     START           SAVE UNIT NUMBER
013.266  042 002 040    3851          SHLD    IOWRK           SAVE DEVICE ADDRESS
013.271  061 200 042    3852          LXI     SP,42200A       SET STACK
                        3853
013.274  041 062 041    3854          LXI     H,AIO.DIR
013.277  016 035        3855          MVI     C,PRIM-AIO.DIR-1        (C) = MAXIMUM ALLOWABLE LENGTH
                        3856
                        3857  *       GET 1ST CHARACTER
                        3858
013.301  315 262 003    3859  CCL1    CALL    RCC             READ KEYBOARD
013.304  376 015        3860          CPI     A.CR            IS HE DONE?
013.306  050 017        3861          JR      Z,CCL3
```

```
013.310  376 072        3862          CPI     ':'             COMMAND LINE FOLLOWS
013.312  050 027        3863          JR      Z,CCL4
013.314  376 040        3864          CPI     ' '             ALLOW A SPACE
013.316  050 002        3865          JR      Z,CCL2
013.320  076 007        3866          MVI     A,A.BEL
013.322  315 302 003    3867  CCL2    CALL    WCC             ECHO CHARACTER
013.325  030 352        3868          JR      CCL1
                        3869
                        3870  *       JUST A CARRIAGE RETURN, NO COMMAND
                        3871
013.327  315 370 005    3872  CCL3    CALL    WCR.            ECHO CRLF
013.332  052 002 040    3873  CCL3.   LHLD    IOWRK
013.335  072 000 040    3874          LDA     START           RESTORE REGISTERS
013.340  303 360 001    3875          JMP     BOOT6           RETURN TO CALLER
                        3876
                        3877  *       HAD ':', COMMAND LINE FOLLOWS
                        3878
013.343  315 302 003    3879  CCL4    CALL    WCC             ECHO THE CHARACTER
013.346  315 262 003    3880  CCL5    CALL    RCC             GET NEXT
013.351  376 015        3881          CPI     A.CR
013.353  050 013        3882          JR      Z,CCL6            IF END OF LINE
013.355  167            3883          MOV     M,A             SAVE CHARACTER
013.356  043            3884          INX     H
013.357  015            3885          DCR     C
013.360  040 361        3886          JR      NZ,CCL4           IF NOT TOO MANY
013.362  014            3887          INR     C               RESET COUNTER
013.363  053            3888          DCX     H               IGNORE IT
013.364  076 007        3889          MVI     A,A.BEL         BEEP
013.366  030 353        3890          JR      CCL4
                        3891
                        3892  *       END OF COMMAND LINE
                        3893
013.370  315 370 005    3894  CCL6    CALL    WCR.
013.373  066 000        3895          MVI     M,0             NUL TERMINATER
013.375  353            3896          XCHG                    (DE)=LWA OF COMMAND
013.376  041 000 000    3897          LXI     H,0
014.001  071            3898          DAD     SP              HL = STACK
014.002  053            3899          DCX     H               HL = LWA OF COMMAND LINE (NULL BYTE)
                        3900
                        3901  *       MOVE COMMAND INTO STACK AREA
                        3902
014.003  363            3903          DI                      NO CLOCK INTERRUPTS
                        3904
014.004  032            3905  CCL7    LDAX    D               DE = COMMAND BYTE
014.005  167            3906          MOV     M,A             MOVE IT IN
014.006  053            3907          DCX     H
014.007  033            3908          DCX     D               BUMP POINTERS
014.010  042 067 040    3909          SHLD    BLKICH          SAVE FOR A SECOND
014.013  041 061 041    3910          LXI     H,AIO.DIR-1     AM I DONE?
014.016  174            3911          MOV     A,H
014.017  272            3912          CMP     D
014.020  040 004        3913          JR      NZ,CCL8           NO
014.022  175            3914          MOV     A,L
014.023  273            3915          CMP     E
014.024  050 005        3916          JR      Z,CCL9            YES, FINISH UP
                        3917
```

```
014.026  052 067 040  3918  CCL8    LHLD    BLKICW
014.031  030 351      3919          JR      CCL7
                      3920
                      3921  *       FINISHED WITH COMMAND LINE, (BLKICW)=FWA-1
                      3922
014.033  052 067 040  3923  CCL9    LHLD    BLKICW
014.036  043          3924          INX     H
014.037  371          3925          SPHL
014.040  030 270      3926          JR      CCL3.           AND GO BACK


                      3928  **      BSMSG - BOOT SECONDARY MESSAGE
                      3929  *
                      3930
014.042  040 123 104  3931  BSMSG   DB      ' SD',0


                      3933  **      ERRMSG - GENERAL ERROR MESSAGE
                      3934
014.046  077 102 157  3935  ERRMSG  DB      '?Boot Error',0


                      3937  **      MSG.PR - Prompt Message
                      3938  *
014.062  015 012 040  3939  MSG.PR  DB      A.CR,A.LF,'  H: ',0
```

```
                                3942  **      RADIX - ASSIGN DEFAULT RADIX
                                3943  *
                                3944  *       RADIX SETS THE SYSTEM RADIX TO OCTAL OR HEX
                                3945  *
                                3946
014.072  041 203 014           3947  RADIX   LXI     H,MSG.RAD
014.075  315 100 006           3948          CALL    TYPMSG          COMPLETE NAME
                                3949
014.100  315 262 003           3950  RADIX1  CALL    RCC             READ CHARACTER
014.103  315 223 015           3951          CALL    MCU             MAP TO UPPER
014.106  376 117               3952          CPI     'O'
014.110  050 017               3953          JR      Z,RADIX2
014.112  376 110               3954          CPI     'H'
014.114  050 026               3955          JR      Z,RADIX3
014.116  376 015               3956          CPI     A.CR
014.120  050 036               3957          JR      Z,RADIX4
014.122  076 007               3958          MVI     A,A.BEL
014.124  315 302 003           3959          CALL    WCC
014.127  030 347               3960          JR      RADIX1
                                3961
                                3962  *       SET OCTAL RADIX
                                3963
014.131  041 211 014           3964  RADIX2  LXI     H,RAD.OCT
014.134  315 100 006           3965          CALL    TYPMSG
014.137  257                   3966          XRA     A
014.140  062 071 040           3967          STA     RADFLG          SET FLAG
014.143  311                   3968          RET
                                3969
                                3970  *       SET     HEX RADIX
                                3971
014.144  041 217 014           3972  RADIX3  LXI     H,RAD.HEX
014.147  315 100 006           3973          CALL    TYPMSG
014.152  076 001               3974          MVI     A,1
014.154  062 071 040           3975          STA     RADFLG
014.157  311                   3976          RET
                                3977
                                3978  *       SHOW CURRENT SETTING
                                3979
014.160  041 211 014           3980  RADIX4  LXI     H,RAD.OCT
014.163  315 112 015           3981          CALL    CHKRAD          ASSUME OCTAL
014.166  050 003               3982          JR      Z,RADIX5          WAS OCTAL
014.170  041 217 014           3983          LXI     H,RAD.HEX
                                3984
014.173  076 015               3985  RADIX5  MVI     A,A.CR
014.175  315 370 005           3986          CALL    WCR.            PRINT CRLF
014.200  303 100 006           3987          JMP     TYPMSG          TYPE NAME
                                3988
                                3989  *       MESSAGES
                                3990
014.203  141 144 151           3991  MSG.RAD DB      'adix ',0
014.211  117 143 164           3992  RAD.OCT DB      'Octal',0
014.217  110 145 170           3993  RAD.HEX DB      'Hexadecimal',0
```

```
                        3996  **      INPUT - PORT INPUT
                        3997  *
                        3998  *       INPUT INPUTS THE VALUE FROM THE SPECIFIED
                        3999  *       PORT NUMBER. THIS VALUE IS THEN PRINTED
                        4000  *
                        4001
014.233 041 327 014     4002  INPUT   LXI     H,MSG.INP
014.236 315 100 006     4003          CALL    TYPMSG          FINISH COMMAND
                        4004
                        4005  *       GET DESIRED PORT NUMBER
                        4006
014.241 041 120 041     4007          LXI     H,PRIM
014.244 247             4008          ANA     A               CLEAR CARRY
014.245 315 036 015     4009          CALL    IOB             GET PORT
                        4010
                        4011  *       READ DATA FROM THAT PORT
                        4012
014.250 315 170 006     4013          CALL    IN.             GET DATA AT (PRIM)
                        4014
                        4015  *       NOW PRINT RESULT
                        4016
014.253 365             4017          PUSH    PSW
014.254 076 015         4018          MVI     A,A.CR
014.256 315 370 005     4019          CALL    WCR.            PRINT CRLF
014.261 361             4020          POP     PSW
014.262 303 077 015     4021          JMP     TOB             TYPE THE BYTE


                        4023  **      OUTPUT - PORT OUTPUT
                        4024  *
                        4025  *       OUTPUT SENDS DATA OUT THE DESIRED PORT
                        4026  *       IN KEEPING WITH THE TAPE LOAD/DUMP ROUTINES, THE
                        4027  *       PORT NUMBER IS SPECIFIED FIRST, FOLLOEND BY A HYPHEN
                        4028  *       AND FOLLWED BY DATA:
                        4029  *
                        4030  *       OUTPUT AAA,DDD<CR>
                        4031  *
                        4032
014.265 041 332 014     4033  OUTPUT  LXI     H,MSG.OUT
014.270 315 100 006     4034          CALL    TYPMSG
                        4035
014.273 041 003 040     4036          LXI     H,IOWRK+1       STORE INFO IN IOWRK
014.276 026 054         4037          MVI     D,'-'           TERMINATE PORT BY HYPHEN
014.300 247             4038          ANA     A               CLEAR CARRY
                        4039
014.301 315 023 015     4040  3100    CALL    IOA             INPUT ADDRESS
                        4041
014.304 072 002 040     4042          LDA     IOWRK           (A)=PORT NUMBER
014.307 062 120 041     4043          STA     PRIM            SAVE IT
                        4044
014.312 247             4045          ANA     A
014.313 041 002 040     4046          LXI     H,IOWRK         GET DATA
014.316 315 036 015     4047          CALL    IOB             GET BYTE AND <CR>
                        4048
```

```
014.321  072 002 040  4049          LDA     IOWRK           GET DATA IN (A)
                      4050
014.324  303 063 006  4051          JMP     OUT.            OUT (PRIM) WITH (A)



                      4053  **      MSG.XXX - INPUT/OUTPUT MESSAGES
                      4054  *
                      4055
014.327  156 040 000  4056  MSG.INP DB      'n ',0
014.332  165 164 040  4057  MSG.OUT DB      'ut ',0
014.336  015 012 012  4058  MSG.ERR DB      A.CR,A.LF,A.LF,'Error a ',0



                      4060  **      SUBM10 - SUBSTITUE PREFIX
                      4061  *
                      4062
014.352  315 112 015  4063  SUBM10  CALL    CHKRAD
014.355  040 014      4064          JR      NZ,SUBM11
014.357  346 007      4065          ANI     00000111B       GET BINARY VALUE
014.361  137          4066          MOV     E,A             SAVE PARTIAL
014.362  176          4067          MOV     A,M             GET CURRENT
014.363  007          4068          RLC                     MAKE ROOM FOR NEW CHARACTER
014.364  007          4069          RLC
014.365  007          4070          RLC
014.366  346 370      4071          ANI     11111000B       TOSS PREVIOUS LSB
014.370  263          4072  SUBM10. ORA     E               ADD NEW
014.371  167          4073          MOV     M,A             SAVE NEW TOTAL
014.372  311          4074          RET
014.373  315 234 015  4075  SUBM11  CALL    CHC             CONVERT IT TO HEX
014.376  346 017      4076          ANI     00001111B
015.000  137          4077          MOV     E,A
015.001  176          4078          MOV     A,M
015.002  007          4079          RLC
015.003  007          4080          RLC
015.004  007          4081          RLC
015.005  007          4082          RLC
015.006  346 360      4083          ANI     11110000B
015.010  030 356      4084          JR      SUBM10.
```

```
                          4087  **      PREFIXES
                          4088  *
                          4089  *       THESE ROUTINES ARE PREFIXES TO THE IOA, IOB
                          4090  *       TOA, AND TOB ROUTINES. THESE PREFIXES DETERMINE
                          4091  *       THE PROPER BASE TO USE, AND TRANSFER CONTROL
                          4092  *       TO THE NEEDED ROUTINES
                          4093  *
                          4094
015.012  315 112 015      4095  IROC    CALL    CHKRAD
015.015  312 140 005      4096          JZ      IROCO
015.020  303 326 015      4097          JMP     IROCH
                          4098
015.023  365              4099  IOA     PUSH    PSW
015.024  315 112 015      4100          CALL    CHKRAD          CHECK RADIX
015.027  302 244 015      4101          JNZ     IHA
015.032  361              4102          POP     PSW             SAVE CARRY FLAG
015.033  303 062 003      4103          JMP     IOAO
                          4104
015.036  365              4105  IOB     PUSH    PSW
015.037  315 112 015      4106          CALL    CHKRAD
015.042  302 123 015      4107          JNZ     IHB
015.045  361              4108          POP     PSW
015.046  303 066 003      4109          JMP     IOBO
                          4110
015.051  365              4111  IOC     PUSH    PSW
015.052  315 112 015      4112          CALL    CHKRAD
015.055  302 214 015      4113          JNZ     IHC
015.060  361              4114          POP     PSW
015.061  303 266 005      4115          JMP     IOCO
                          4116
015.064  365              4117  TOA     PUSH    PSW
015.065  315 112 015      4118          CALL    CHKRAD
015.070  302 001 016      4119          JNZ     THA
015.073  361              4120          POP     PSW
015.074  303 300 005      4121          JMP     TOAO
                          4122
015.077  365              4123  TOB     PUSH    PSW
015.100  315 112 015      4124          CALL    CHKRAD
015.103  302 350 015      4125          JNZ     THB
015.106  361              4126          POP     PSW
015.107  303 322 005      4127          JMP     TOBO
                          4128
                          4129  *       CHECK CURRENT RADIX
                          4130
015.112  305              4131  CHKRAD  PUSH    B
015.113  107              4132          MOV     B,A
015.114  072 071 040      4133          LDA     RADFLG
015.117  247              4134          ANA     A
015.120  170              4135          MOV     A,B
015.121  301              4136          POP     B
015.122  311              4137          RET
```

```
                        4139  **      HEX ROUTINES
                        4140  *
                        4141  *       THESE ROUTINES ARE THE HEX EQUIVELANT OF THE
                        4142  *       OCTAL ROUTINES PREFIXED ABOVE
                        4143  *
                        4144  *       NOTE: THESE ROUTINES ARE ENTERED WITH PSW ON THE STACK
                        4145  *
                        4146
                        4147
015.123  066 000        4148  IHB     MVI     M,0             CLEAR RESULT
015.125  361            4149          POP     PSW
015.126  324 262 003    4150  IHB1    CNC     RCC
                        4151
                        4152  *       CHECK FOR VALIDITY
                        4153
015.131  315 173 015    4154          CALL    CCH             CHECK CHARACTER FOR VALID HEX
015.134  060 013        4155          JR      NC,IHB2
015.136  376 015        4156          CPI     A.CR            RETURN?
015.140  310            4157          RZ                      YES, DONE
015.141  247            4158          ANA     A               INSURE CARRY OFF
015.142  076 007        4159          MVI     A,A.BEL
015.144  315 302 003    4160          CALL    WCC
015.147  030 355        4161          JR      IHB1
                        4162
                        4163  *       HAVE A VALID HEX CHARACTER
                        4164
015.151  315 302 003    4165  IHB2    CALL    WCC
015.154  315 234 015    4166          CALL    CHC             CONVERT HEX CHARACTER
015.157  137            4167          MOV     E,A
015.160  176            4168          MOV     A,M             GET VALUE SO FAR
015.161  007            4169          RLC
015.162  007            4170          RLC                     MOVE UP NIBBLE
015.163  007            4171          RLC
015.164  007            4172          RLC
015.165  346 360        4173          ANI     11110000B       THROW AWAY LAST
015.167  263            4174          ORA     E
015.170  167            4175          MOV     M,A             SET NEW NIBBLE
015.171  030 333        4176          JR      IHB1
                        4177  **      CHECK FOR VALID HEX CHARACTER
                        4178  *
                        4179  *       CCH CHECKS (A) FOR HEX VALIDITY
                        4180  *       'C' IS SET IF INVALID
                        4181  *
                        4182
015.173  315 223 015    4183  CCH     CALL    MCU             MAP TO UPPER
015.176  376 060        4184          CPI     '0'
015.200  330            4185          RC                      IF LESS THAN ZERO
015.201  376 072        4186          CPI     '9'+1
015.203  077            4187          CMC
015.204  320            4188          RNC                     BETWEEN 0 AND 9
015.205  376 101        4189          CPI     'A'
015.207  330            4190          RC                      LOWER CASE IS NOT VALID
015.210  376 107        4191          CPI     'F'+1
015.212  077            4192          CMC
015.213  311            4193          RET
```

```
                        4195  **      IHC - INPUT HEX CHARACTER
                        4196  *
                        4197
015.214  361           4198  IHC     POP     PSW
015.215  315 262 003    4199          CALL    RCC             GET CHARACTER
015.220  303 173 015    4200          JMP     CCH             CHECK FOR VALID HEX


                        4202  **      MCU - MAP CASE TO UPPER
                        4203  *
                        4204
015.223  376 141        4205  MCU     CPI     'a'
015.225  330            4206          RC                      LESS THAN 'A'
015.226  376 173        4207          CPI     'z'+1
015.230  320            4208          RNC
015.231  346 137        4209          ANI     01011111B
015.233  311            4210          RET
```

```
                        4213  **      CONVERT HEX TO BINARY
                        4214  *
                        4215  *       CHC CONVERTS THE ASCII CHARACTER IN (A) INTO
                        4216  *       IT'S 4BIT HEX EQUIVELANT
                        4217  *
                        4218
015.234  326 060       4219  CHC     SUI     '0'
015.236  376 012       4220          CPI     9+1
015.240  330           4221          RC                      IF DONE
015.241  326 007       4222          SUI     7
015.243  311           4223          RET                     CONVERT A - F
                        4224
                        4225  *       INPUT HEX ADDRESS
                        4226
015.244  361           4227  IHA     POP     PSW
015.245  305           4228  IHA.    PUSH    B
015.246  102           4229          MOV     B,D
015.247  345           4230          PUSH    H               B = DELIMITER
015.250  041 000 000    4231          LXI     H,0
015.253  324 262 003    4232  IHA1    CNC     RCC
015.256  315 173 015    4233          CALL    CCH             CHECK FOR HEX
015.261  070 016        4234          JR      C,IHA3            IF NOT, CHECK DELIMITER
015.263  315 302 003    4235          CALL    WCC
015.266  315 234 015    4236          CALL    CHC             CONVERT IT
015.271  051           4237          DAD     H
015.272  051           4238          DAD     H
015.273  051           4239          DAD     H
015.274  051           4240          DAD     H               HL = HL * 16
015.275  205           4241          ADD     L
015.276  157           4242          MOV     L,A             MOVE IN NEW NIBBLE
015.277  030 352       4243          JR      IHA1
                        4244
015.301  270           4245  IHA3    CMP     B
015.302  050 010       4246          JR      Z,IHA4            IF VALID DELIMITER
015.304  076 007       4247          MVI     A,A.BEL
015.306  315 302 003    4248          CALL    WCC
015.311  247           4249          ANA     A
015.312  030 337       4250          JR      IHA1
                        4251
                        4252  *       END OF INPUT
                        4253
015.314  315 302 003    4254  IHA4    CALL    WCC             PRINT DELIMITER
015.317  353           4255          XCHG
015.320  341           4256          POP     H
015.321  162           4257          MOV     M,D
015.322  053           4258          DCX     H
015.323  163           4259          MOV     M,E
015.324  301           4260          POP     B
015.325  311           4261          RET
                        4262
                        4263  *       IROC REPLACEMENT
                        4264
015.326  315 262 003    4265  IROCH   CALL    RCC
015.331  376 015       4266          CPI     A.CR
015.333  310           4267          RZ                      IF CARRIAGE RETURN
                        4268
```

```
015.334  315 173 015  4269          CALL    CCH
015.337  077          4270          CMC
015.340  330          4271          RC                      IF VALID
                       4272
015.341  076 007      4273          MVI     A,A.BEL
015.343  315 302 003  4274          CALL    WCC
015.346  030 356      4275          JR      IROCH
                       4276
                       4277  *       TYPE BYTE REPLACEMENT
                       4278
015.350  361          4279  THB     POP     PSW
015.351  365          4280  THB1    PUSH    PSW
015.352  346 360      4281          ANI     11110000B
015.354  017          4282          RRC
015.355  017          4283          RRC
015.356  017          4284          RRC
015.357  017          4285          RRC                     DO HIGH NIBBLE FIRST
015.360  315 366 015  4286          CALL    THB2
015.363  361          4287          POP     PSW
015.364  346 017      4288          ANI     00001111B
                       4289
                       4290  *       THB1 - TYPE NIBBLE
                       4291
015.366  306 060      4292  THB2    ADI     '0'
015.370  376 072      4293          CPI     '9'+1
015.372  070 002      4294          JR      C,THB3
015.374  306 007      4295          ADI     7
015.376  303 302 003  4296  THB3    JMP     WCC
                       4297  *       THA - TYPE HEX ADDRESS
                       4298
016.001  361          4299  THA     POP     PSW
016.002  076 015      4300          MVI     A,A.CR
016.004  315 370 005  4301          CALL    WCR.
                       4302
016.007  174          4303  THA1    MOV     A,H
016.010  315 351 015  4304          CALL    THB1
016.013  175          4305          MOV     A,L
016.014  315 351 015  4306          CALL    THB1
016.017  076 040      4307          MVI     A,' '
016.021  303 302 003  4308          JMP     WCC


                       4310  **      MEMORY - MEMORY DIAGNOSTIC
                       4311  *
                       4312  *       MEMORY IS THE PREFACE TO THE MEMORY
                       4313  *       DIAGNOSTIC UTILTIY
                       4314  *
016.024  041 043 016  4315  MEMORY  LXI     H,MSG.MEM
016.027  315 100 006  4316          CALL    TYPMSG
016.032  072 071 040  4317  MEMORY. LDA     RADFLG
                       4318  *       EXX
016.035  331          4319          DB      MI.EXX
016.036  157          4320          MOV     L,A
                       4321  *       EXX
```

```
016.037  331           4322           DB      MI.EXX
016.040  303 116 007   4323           JMP     DYMEM
                       4324
016.043  145 163 164   4325  MSG.MEM  DB      'est Memory',A.CR,A.LF,0


                       4327  **       GETBND - GET BOUNDRIES
                       4328  *
                       4329  *        GETBND GETS THREE ADDRESS BOUNDRIES, RETURNING
                       4330  *        THE FIRST IN HL, THE SECOND IN DE AND THE THIRD
                       4331  *        IN BC.
                       4332  *
                       4333
016.060  041 003 040   4334  GETBND   LXI     H,IOWRK+1
016.063  026 054       4335           MVI     D,',
016.065  315 023 015   4336           CALL    IOA             GET FIRST
                       4337
016.070  052 002 040   4338           LHLD    IOWRK
016.073  345           4339  GETBND.  PUSH    H               ENTRY POINT FOR DE,BC ONLY
                       4340
016.074  041 003 040   4341           LXI     H,IOWRK+1
016.077  026 054       4342           MVI     D,',
016.101  315 023 015   4343           CALL    IOA
                       4344
016.104  052 002 040   4345           LHLD    IOWRK           SAVE SECOND
016.107  345           4346           PUSH    H
016.110  303 024 001   4347           JMP     GETBND1         CONTINUE ELSEWHERE


                       4349  **       INTOX0 - EXTENSION TO INTOX
                       4350  *
                       4351  *        INTOX0 CLEANS UP SOME OF THE RAM CELLS
                       4352  *
                       4353
016.113  257           4354  INTOX0   XRA     A
016.114  062 071 040   4355           STA     RADFLG
016.117  041 377 377   4356           LXI     H,-1
016.122  042 072 040   4357           SHLD    VEWHLD
016.125  001 200 076   4358           LXI     B,16000
016.130  303 050 004   4359           JMP     INITOX1


                       4361  **       BOOT7 - EXTENSION TO BOOT ROUTINE
                       4362  *
                       4363  *        THIS ROUTINE HANDLES BOOTING FROM DEVICE
                       4364  *        ZERO WITH COMMAND LINES
                       4365  *
                       4366
016.133  376 040       4367  BOOT7    CPI     ' '
016.135  050 007       4368           JR      Z,BOOT71        TYPED SPACE, MUST WANT COMMAND LINE
```

```
016.137  376 072        4369          CPI     ':'
016.141  050 012        4370          JR      Z,BOOT72        TYPE :, HERE COMES COMMAND
016.143  376 060        4371          CPI     '0'
016.145  311            4372          RET                     OTHERWISE, MAYBE UNIT NUMBER
016.146  315 302 003    4373  BOOT71  CALL    MCC
016.151  257            4374          XRA     A               ENTER CCL AS UNIT 0
016.152  303 263 013    4375          JMP     CCL
                        4376
                        4377  *       HE ALREADY STARTED THE COMMAND LINE, LETS CATCH UP!
                        4378
016.155  257            4379  BOOT72  XRA     A
016.156  062 000 040    4380          STA     START           SAVE UNIT NUMBER
016.161  042 002 040    4381          SHLD    IOWRK           SAVE DEVICE ADDRESS
016.164  061 200 042    4382          LXI     SP,42200A       SET UP STACK
016.167  041 062 041    4383          LXI     H,AIO.DIR
016.172  016 035        4384          MVI     C,PRIM-AIO.DIR-1
016.174  076 072        4385          MVI     A,':'           ECHO THE COLON
016.176  303 343 013    4386          JMP     CCL4            CONTINUE FROM HERE


                        4388  **      EUC - ENTER USER CODE
                        4389  *
                        4390  *       EUC ENTERS THE USER BOOT CODE, AFTER RE-VECTORING
                        4391  *       THE CLOCK INTERRUPT REQUEST VECTORS
                        4392  *
                        4393  *       THE H17 RAM CONSTANTS ETC. ARE ALSO MOVED IN
                        4394  *
                        4395
016.201  001 130 000    4396  EUC     LXI     B,BOOTAL                SET THE COUNT TO MOVE IN CONSTANTS AND VECTORS
016.204  021 132 037    4397          LXI     D,BOOTA                 SET THE SOURCE ADDRESS
016.207  041 110 040    4398          LXI     H,D.CON                 SET THE DESTINATION ADDRESS
016.212  315 252 030    4399          CALL    $MOVE                   MOVE IT
                        4400
                        4401  *       ENTRY POINT FROM H17 (CONSTANTS ALREADY MOVED IN)
                        4402
016.215  363            4403  EUC.    DI                              STOP CLOCK
016.216  041 031 034    4404          LXI     H,CLOCK17               LOAD CLOCK ROUTINE ADDRESS
016.221  042 040 040    4405          SHLD    UIVEC+1                 SET IT INTO VECTOR LOCATION
016.224  373            4406          EI
                        4407
                        4408  *       Zero out H67 operating system info
                        4409
000.000                 4410          ERRNZ   S.OSZ-S.OSI-2           MUST BE CONTIGUOUS BYTES
016.225  041 126 041    4411          LXI     H,S.OSI
016.230  006 005        4412          MVI     B,1+1+3
016.232  315 212 031    4413          CALL    $ZERO                   Zero area
                        4414
016.235  303 200 042    4415          JMP     USERFWA                 GOTO BOOT CODE
```

```
                        4417  **      DYBYTX
                        4418  *
                        4419  *       DYBYTX DETERMINES WHETHER TO OUPUT THE BYTE
                        4420  *       IN HEX OR OCTAL. IF IN OCTAL WE MUST REPLACE
                        4421  *       THE CODE WE PATCHED TO GET US HERE.
                        4422  *
                        4423  *       ENTRY:  (A) = BYTE OF TO OUTPUT
                        4424  *
                        4425
016.240  117          4426  DYBYTX  MOV     C,A             SAVE BYTE
                        4427  *       EXX
016.241  331          4428          DB      MI.EXX
016.242  175          4429          MOV     A,L             GET RADIX FLAG
                        4430  *       EXX
016.243  331          4431          DB      MI.EXX
016.244  247          4432          ANA     A               'Z' SET IF OCTAL
016.245  171          4433          MOV     A,C             RESTORE A
016.246  040 012       4434          JR      NZ,DYBYTH       IF IN HEX
                        4435
016.250  117          4436  DYBYTO  MOV     C,A
016.251  346 300       4437          ANI     11000000B
016.253  007          4438          RLC
016.254  007          4439          RLC
016.255  346 003       4440          ANI     000000011B
016.257  303 163 003   4441          JMP     DYBYT.1         FINISH UP OLD OCTAL ROUTINE
                        4442
                        4443  *       IS HEX
                        4444
016.262  117          4445  DYBYTH  MOV     C,A
016.263  346 360       4446          ANI     11110000B
016.265  017          4447          RRC
016.266  017          4448          RRC
016.267  017          4449          RRC
016.270  017          4450          RRC             MOVE DOWN HIGH HALF
016.271  346 017       4451          ANI     000011111B
016.273  306 060       4452          ADI     '0'
016.275  376 072       4453          CPI     '9'+1
016.277  070 002       4454          JR      C,DYBYTH1
016.301  306 007       4455          ADI     7
016.303  375 041       4456  DYBYTH1 DB      MI.LDYA,MI.LDYB
016.305  312 016       4457          DW      DYBYTH2         SET RETURN ADDRESS
016.307  303 143 003   4458          JMP     DYASC
                        4459
016.312  171          4460  DYBYTH2 MOV     A,C
016.313  346 017       4461          ANI     000011111B
016.315  306 060       4462          ADI     '0'
016.317  376 072       4463          CPI     '9'+1
016.321  070 002       4464          JR      C,DYBYTH3
016.323  306 007       4465          ADI     7
016.325  375 041       4466  DYBYTH3 DB      MI.LDYA,MI.LDYB
016.327  334 016       4467          DW      DYBYTH4
016.331  303 143 003   4468          JMP     DYASC
                        4469
                        4470  *       JP      (IX)
                        4471
016.334  335 351       4472  DYBYTH4 DB      MI.JIXA,MI.JIXB
```

```
                                4474  **      DYMM5
                                4475  *
                                4476  *       DETERMINE NUMBER OF BACKSPACES
                                4477  *       TO TYPE FOR EACH CHARACTER OUTPUT
                                4478  *
                                4479
016.336  331                    4480  DYMM5   DB      MI.EXX
016.337  175                    4481          MOV     A,L
016.340  331                    4482          DB      MI.EXX
016.341  247                    4483          ANA     A
016.342  076 010                4484          MVI     A,A.BKS
016.344  046 003                4485          MVI     H,3
016.346  312 221 007            4486          JZ      DYME5.5
016.351  045                    4487          DCR     H
016.352  303 221 007            4488          JMP     DYME5.5


                                4490  **      CONVERT - BASE CONVERSION
                                4491  *
                                4492  *       CONVERT CONVERTS THE INPUT IN THE OPPOSITE
                                4493  *       RADIX AND CHANGES IT TO THE CURRENT RADIX
                                4494  *
                                4495
016.355  041 014 017            4496  CONVERT LXI     H,MSG.CON
016.360  315 100 006            4497          CALL    TYPMSG
016.363  041 003 040            4498          LXI     H,IOWRK+1
016.366  026 015                4499          MVI     D,A.CR
016.370  315 112 015            4500          CALL    CHKRAD
016.373  050 005                4501          JR      Z,CONV.O                IF OCTAL
                                4502
016.375  315 062 003            4503  CONV.H  CALL    IOAO
017.000  030 003                4504          JR      CONV.E
                                4505
017.002  315 245 015            4506  CONV.O  CALL    IHA.
                                4507
017.005  052 002 040            4508  CONV.E  LHLD    IOWRK
017.010  353                    4509          XCHG
017.011  303 064 015            4510          JMP     TOA
                                4511
017.014  157 156 166            4512  MSG.CON DB      'onvert ',0
```

```
                          4515  ***     H17X - H17 Extension routine
                          4516  *
                          4517  *       H17x is the extension to the H17 Abort command
                          4518  *
                          4519
017.024  315 366 033      4520  H17X    CALL    R.ABORT
                          4521
                          4522  *       Step the head out 10 tracks
                          4523
017.027  315 044 002      4524          CALL    R.SDP           Set up device
017.032  076 012          4525          MVI     A,10
017.034  062 240 040      4526          STA     D.TT            Set target track to 10
017.037  315 166 040      4527          CALL    D.SDT           Seek Desired track
                          4528
017.042  303 366 033      4529          JMP     R.ABORT         Abort and return
```

```
                        4532  **      MTRA - COMMAND DESCRIPTOR TABLE
                        4533  *
                        4534  *       THIS TABLE CONTAINS THE SINGLE LETTER COMMANDS
                        4535  *       UNDERSTOOD BY MTR90. THE ENTRIES IN THIS TABLE
                        4536  *       CONSIST OF A SINGLE LETTER (THE COMMAND KEY) FOLLOWED
                        4537  *       BY A WORD ADDRESS OF THE ROUTINE.
                        4538  *
                        4539  *       NOTE: THIS TABLE WAS MOVED FROM UP FRONT BECAUSE
                        4540  *             OF SIZE CONSIDERATIONS
                        4541  *
                        4542
017.045                 4543  MTRA    EQU     *
017.045  107            4544          DB      'G'             *GO*
017.046  146 001        4545          DW      GO88
                        4546
017.050  123            4547          DB      'S'             *SUBSTITUTE*
017.051  370 004        4548          DW      SUBM
                        4549
017.053  120            4550          DB      'P'             *PROGRAM COUNTER*
017.054  103 001        4551          DW      PCA
                        4552
017.056  102            4553          DB      'B'             *BOOT*
017.057  256 004        4554          DW      BOOT
                        4555
017.061  111            4556          DB      'I'             *INPUT*
017.062  233 014        4557          DW      INPUT
                        4558
017.064  117            4559          DB      'O'             *OUTPUT*
017.065  265 014        4560          DW      OUTPUT
                        4561
017.067  122            4562          DB      'R'             *RADIX*
017.070  072 014        4563          DW      RADIX
                        4564
017.072  124            4565          DB      'T'
017.073  024 016        4566          DW      MEMORY          *TEST RAM*
                        4567
017.075  126            4568          DB      'V'             *VIEW*
017.076  044 001        4569          DW      VIEW
                        4570
017.100  103            4571          DB      'C'
017.101  355 016        4572          DW      CONVERT
000.012                 4573  MTRAL   EQU     *-MTRA/3        NUMBER OF ENTRIES       /JWT  790507/


                        4575  **      BT170, BT174 - BOOT TABLES
                        4576  *
                        4577  *       THESE TABLES DEFINE DEVICE DEPENDENT INFORMATION USED
                        4578  *       TO DETERMINE WHICH DEVICE IS TO BE BOOTED FROM
                        4579  *
                        4580  *       THE ORGANIZATION OF THE TWO TABLES IS IDENTICAL:
                        4581  *
                        4582  *       BYTE 1          -       PORT NUMBER OF THESE DEVICES
                        4583  *
                        4584  *       BYTE 2          -       DEVICE 0 TMFG
```

```
                        4585  *       BYTE 3          -               MAX UNITS
                        4586  *       BYTE 4,5        -               BOOT CODE ADDRESS
                        4587  *
                        4588  *       BYTE 6          -       DEVICE 1 TMFG
                        4589  *       BYTE 7          -               MAX UNITS
                        4590  *       BYTE 8,9        -               BOOT CODE ADDRESS
                        4591  *
                        4592  *       etc., etc., etc. THRU DEVICE 3
                        4593  *
                        4594  *       NO END-OF-TABLE CHECK IS MADE, THEREFORE, 4 ENTRIES
                        4595  *       MUST EXIST PER TABLE
                        4596  *
                        4597
017.103  174           4598  BT174   DB      174Q            PORT ADDRESS
017.104                 4599  BT174E  EQU     *
                        4600
017.104  000           4601  BTH174  DB      0               TMFG = 0
017.105  063           4602          DB      '3'             MAX UNIT = 3
017.106  207 002       4603          DW      H17             BOOT ADDRESS
                        4604
017.110  001           4605  BTH474  DB      1
017.111  064           4606          DB      '4'
017.112  372 001       4607          DW      Z47
                        4608
017.114  000           4609  BTH674  DB      0
017.115  064           4610          DB      '4'
017.116  174 012       4611          DW      H67
                        4612
017.120  000           4613  BTHFE4  DB      0
017.121  061           4614          DB      '1'
017.122  222 013       4615          DW      FEDEV
                        4616
000.004                 4617  BT174L  EQU     *-BT174E/4      INSURE ALL ENTRIES FILLED
000.000                 4618          ERRNZ   BT174L-4
                        4619
017.124  170           4620  BT170   DB      170Q            PORT ADDRESS
017.125                 4621  BT170E  EQU     *
                        4622
017.125  000           4623  BTH370  DB      0
017.126  064           4624          DB      '4'
017.127  221 011       4625          DW      H37
                        4626
017.131  001           4627  BTH470  DB      1
017.132  064           4628          DB      '4'
017.133  372 001       4629          DW      Z47
                        4630
017.135  000           4631  BTH670  DB      0
017.136  064           4632          DB      '4'
017.137  174 012       4633          DW      H67
                        4634
017.141  000           4635  BTHFE0  DB      0
017.142  061           4636          DB      '1'
017.143  222 013       4637          DW      FEDEV
                        4638
000.004                 4639  BT170L  EQU     *-BT170E/4
000.000                 4640          ERRNZ   BT170L-4
```

```
000.233                 4642            ERRMI   20000A-*        MUST NOT EXCEED 4K BYTES
```

```
                    4645  **      THE FOLLOWING ARE CONTROL CELLS AND FLAGS USED BY THE KEYSET
                    4646  *       MONITOR.
                    4647
040.000             4648          ORG     40000A          8192
040.000             4649  START   DS      2               DUMP STARTING ADDRESS
040.002             4650  IOWRK   DS      2               IN OR OUT INSTRUCTION
040.004             4651  PRSRAM  EQU     *               FOLLOWING CELLS INITIALIZED FROM ROM
040.004             4652          DS      1               RET
                    4653
040.005             4654  REGI    DS      1               INDEX OF REGISTER UNDER DISPLAY
040.006             4655  DSPROT  DS      1               PERIOD FLAG BYTE
040.007             4656  DSPMOD  DS      1               DISPLAY MODE
                    4657
040.010             4658  .MFLAG  DS      1               USER FLAG OPTIONS
                    4659  *                               SEE *U0.XXX* BITS DESCRIBED AT FRONT
                    4660
040.011             4661  CTLFLG  DS      1               FRONT PANEL CONTROL BITS
040.012             4662  REFIND  DS      1               REFRESH INDEX (0 TO 7)
000.007             4663  PRSL    EQU     *-PRSRAM        END OF AREA INITIALIZED FROM ROM
                    4664
040.013             4665  FPLEDS  EQU     *               FRONT PANEL LED PATTERNS
040.013             4666  ALEDS   DS      1               ADDR 0
040.014             4667          DS      1               ADDR 1
040.015             4668          DS      1               ADDR 2
                    4669
040.016             4670          DS      1               ADDR 3
040.017             4671          DS      1               ADDR 4
040.020             4672          DS      1               ADDR 5
                    4673
040.021             4674  DLEDS   DS      1               DATA 0
040.022             4675          DS      1               DATA 1
040.023             4676          DS      1               DATA 2
                    4677
040.024             4678  ABUSS   DS      2               ADDRESS BUSS
040.026             4679  RCCA    DS      1               RCC SAVE AREA
040.027             4680  CRCSUM  DS      2               CRC-16 CHECKSUM
040.031             4681  TPERRX  DS      2               TAPE ERROR EXIT ADDRESS
040.033             4682  TICCNT  DS      2               CLOCK TIC COUNTER
                    4683
040.035             4684  REGPTR  DS      2               REGISETR CONTENTS POINTER
                    4685
040.037             4686  UIVEC   DS      0               USER INTERRUPT VECTORS
040.037             4687          DS      3               JUMP TO CLOCK PROCESSOR
040.042             4688          DS      3               JUMP TO SINGLE STEP PROCESSOR
040.045             4689          DS      3               JUMP TO I/O 3
040.050             4690          DS      3               JUMP TO I/O 4
040.053             4691          DS      3               JUMP TO I/O 5
040.056             4692          DS      3               JUMP TO I/O 6
040.061             4693          DS      3               JUMP TO I/O 7
                    4694
                    4695  **      H88/H89 RAM USAGE BEYOND THAT OF H8MTRF
                    4696  *
040.064             4697  NMIRET  DS      2
040.066             4698  DATA    DS      1               OUTPUT TO 3620 DATA SAVE
040.067             4699  BLKICW  DS      2               H37 INTERRUPT RETURN ADDRESS
040.071             4700  RADFLG  DS      1               RADIX FLAG
```

```
    040.072             4701  VEHHLD  DS      2
    000.001             4702          IF      .DEBUG
                        4703  DBFLG   DS      1           FOR DEBUG RESULTS
                        4704          ENDIF
    040.074             4705  MEML    EQU     *
    001.024             4706          ERRMI   41120A-MEML
    041.120             4707          ORG     41120A
    041.120             4708  PRIM    DS      1           PRIMARY DEVICE ADDR. PORT
    041.121             4709  TMFG    DS      1           TIMER INTERRUPT FLAG, =1 FOR Z47, =0 FOR H17
    041.122             4710  MYCNT   DS      1           COUNTER FOR TIMER INTERRUPT
    041.123             4711  AUTOB   DS      1           AUTO BOOT FLAG
    041.124             4712  STK     DS      2           STACK POINTER FOR RE-BOOT
                        4713
    041.126             4714          END
Assembly complete
  4714 statements
     0 errors detected
22422 bytes free
```

$MOVE	030252	745E	1636	4399										
$ZERO	031212	747E	1685	4413										
.	000004	910S	1065S	1386S	2119S	2120								
.DEBUG	000001	1E	1484	1496	1503	3087	3097	3109	3121	3195	4702			
.MFLAG	040010	1065	1107	1176	1182	1617	1692	4658L						
.UIVEC	040037	748E												
3100	014301	4040L												
A.BEL	000007	155E	1124	1216	1420	2378	2398	2424	2466	2635	3826	3866	3889	
		3958	4159	4247	4273									
A.BKS	000010	156E	4484											
A.CR	000015	158E	1231	1316	1337	1410	1852	1924	2349	2375	2395	2415	2515	
		2569	2581	2869	3038	3860	3881	3939	3956	3985	4018	4058	4156	4266
		4300	4325	4499										
A.DEL	000177	160E	1974	2298										
A.ESC	000033	159E	2868	2877	2879	2885	2887	3036						
A.LF	000012	157E	1924	1924	2573	2868	2869	2869	3038	3038	3939	4058	4058	
		4325												
A.STX	000002	154E												
A.SYN	000026	153E												
ABUSS	040024	1262	4678L											
AC.DLY	000156	810E												
AIO.CGN	041047	710L												
AIO.CHA	041116	725L												
AIO.CNT	041111	721L												
AIO.CSI	041050	711L												
AIO.DDA	041041	706E												
AIO.DES	041055	715L												
AIO.DEV	041057	716L												
AIO.DIR	041062	719L	3623	3632	3643	3655	3664	3672	3683	3735	3765	3769	3774	
		3854	3855	3910	4383	4384								
AIO.DTA	041053	714L												
AIO.EOF	041113	723L												
AIO.EOM	041112	722L												
AIO.FLG	041043	707L												
AIO.GRT	041044	708L												
AIO.LGN	041051	712L												
AIO.LSI	041052	713L												
AIO.SPG	041046	709L												
AIO.TFP	041114	724L												
AIO.UNI	041061	717L	1448	1475	1539	3102	3432	3651	3689					
AIO.VEC	041040	705L												
ALARM	002136	1186	1571E											
ALARMB	002166	1572	1597L											
ALEDS	040013	4666L												
AS.0DD	000100	369E												
AS.1DD	000040	370E												
AS.S1A	000020	371E												
AS.SLM	000003	372E												
ATB	004243	2244L	3375											
AUTO8	041123	2106	3373	4711L										
AUTO80	001205	1350L	2247											
BASE	000170	459E												
BC.EDT	000002	470E	3730											
BC.IE	000040	468E												
BC.RST	000020	469E	3617											
BC.SEL	000100	467E	3716											
BITS	012155	3105	3436	3595L										

BITS1	012162	3600L	3602									
BLKICH	040067	1249	2033	3449	3469	3479	3544	3578	3909	3918	3923	4699L
BOOT	004256	2260L	4554									
BOOT.P	000001	684E										
BOOT.SY	000002	685E										
BOOT0	001342	1352	1441L									
BOOT0.	001337	1411	1439L									
BOOT2	002324	1694L	1701									
BOOT5	001350	1415	1445L									
BOOT6	001360	1443	1448L	3875								
BOOT7	016133	1412	4367L									
BOOT71	016146	4368	4373L									
BOOT72	016155	4370	4379L									
BOOTA	037132	744E	1634	4397								
BOOTAL	000130	743E	1633	4396								
BOOTX	004275	2248	2266L									
BR19.2	004103	2115L										
BR96	004101	2114L										
BRTAB	004101	2067	2112E	2120								
BS.BSY	000010	482E	3702	3720								
BS.COM	000020	481E	3743	3759	3760	3771	3772	3804				
BS.DAT	000000	480E										
BS.DTD	000100	475E	3747	3759	3771							
BS.HID	000001	485E										
BS.IN	000000	476E										
BS.INT	000004	483E										
BS.LMB	000040	478E										
BS.MTY	000020	479E	3771	3772								
BS.OUT	000100	477E										
BS.PE	000002	484E										
BS.REQ	000200	474E	3741	3759	3760	3771	3772	3800				
BSEC	001326	1431E										
BSMSG	014042	1432	3931L									
BT170	017124	1734	4620L									
BT170E	017125	4621E	4639									
BT170L	000004	4639E	4640									
BT174	017103	1717	1727	4598L								
BT174E	017104	4599E	4617									
BT174L	000004	4617E	4618									
BTH174	017104	4601L										
BTH370	017125	4623L										
BTH470	017131	4627L										
BTH474	017110	4605L										
BTH670	017135	4631L										
BTH674	017114	4609L										
BTHFE0	017141	4635L										
BTHFE4	017120	4613L										
C.DSYN	000375	244E										
CB.CLI	000100	166E	201	958	1181	2219						
CB.MTL	000040	165E	1051	1104	1181	1388						
CB.SPK	000200	167E	958	1181	1577							
CB.SSI	000020	164E	958	1051	1181	1373	1384	2219				
CBUSY	013032	3719L	3726									
CBUSY1	013050	3721	3730L									
CCH	015173	4154	4183L	4200	4233	4269						
CCL	013263	1447	3850L	4375								
CCL1	013301	3859L	3868									

CCL2	013322	3865	3867L										
CCL3	013327	3861	3872L										
CCL3.	013332	3873L	3926										
CCL4	013343	3863	3879L	3886	3890	4386							
CCL5	013346	3880L											
CCL6	013370	3882	3894L										
CCL7	014004	3905L	3919										
CCL8	014026	3913	3918L										
CCL9	014033	3916	3923L										
CDB.H84	000001	627E											
CDB.H85	000000	626E											
CHAT2	011152	3372	3377L										
CHC	015234	4075	4166	4219L	4236								
CHKRAD	015112	2022	3265	3276	3335	3353	3981	4063	4095	4100	4106	4112	4118
		4124	4131L	4500									
CKAUTO	011135	1199	3370L										
CLASS0	000000	499E	509	510	511	512	513	514	515	516	517	518	519
		520											
CLASS1	000040	500E	524										
CLASS6	000300	501E	528										
CLASSM	000340	497E											
CLK4	000270	1111	1134E										
CLOCK	000201	894	895	1093L									
CLOCK17	034031	749E	2322	4404									
COM	006027	2610E	2691	3115	3129	3191							
COM1	006031	2597	2612L										
COM2	010156	2617	3208L										
COM3	010164	3212L	3213										
COMREQ	013060	3737L	3740	3753									
CON.CD	000000	324E	3541										
CON.DRQ	000002	313E	3499										
CON.DS0	000020	316E											
CON.DS1	000040	317E											
CON.DS2	000100	318E											
CON.DS3	000200	319E											
CON.EI	000001	312E	3438										
CON.MFM	000004	314E	3438	3508									
CON.MO	000010	315E	3438	3557									
CON.ST	000001	325E	3535	3538									
CONV.E	017005	4504	4508L										
CONV.H	016375	4503L											
CONV.O	017002	4501	4506L										
CONVERT	016355	4496L	4572										
CRCSUM	040027	4680L											
CTLFLG	040011	910	1097	1102	1107	1182	1372	1375	1386	1582	2127	2625	4661L
CUI1	000165	1066L	1123	1135									
D.CON	040110	574L	1635	4398									
D.CPB	000040	524E											
D.CTF	000005	514E											
D.DAT	000171	340E											
D.FBS	000007	516E											
D.FFD	000300	528E											
D.FOR	000004	513E											
D.FT	000006	515E											
D.OECNT	040264	758E	1538										
D.RAM	040240	577L	1683										
D.RAML	000037	746E	1684										

D.REA	000010	517E	3673				
D.REC	000001	510E	3644	3665			
D.RSE	000003	512E					
D.RSY	000002	511E					
D.SDP	040206	752E	1642				
D.SDT	040166	753E	4527				
D.SEK	000013	520E	3656				
D.STA	000170	339E	340				
D.TDR	000000	509E	3624				
D.TT	040240	754E	4526				
D.VEC	040130	576L					
D.WPS	000011	518E					
D.WRI	000012	519E					
DAT	006023	2595E	2693	2695	3117	3132	3135
DATA	040066	2107	2227	4698L			
DD.BOOT	000000	378L					
DD.CPY	000013	389L					
DD.OS	000202	413L					
DD.FRM0	000014	390L					
DD.FRM1	000015	391L					
DD.FRM2	000016	392L					
DD.FRM3	000017	393L					
DD.LSC	000003	381L	3128				
DD.RAD	000004	382L					
DD.RAS	000002	380L	3114				
DD.RDBL	000205	416L					
DD.REA	000005	383L					
DD.REAB	000007	385L	2690				
DD.ROL	000203	414L					
DD.RRDY	000020	394L	3190				
DD.RST	000001	379L					
DD.SDC	000200	411L					
DD.SPF0	000020	400L					
DD.SPF1	000021	401L					
DD.SPF2	000022	402L					
DD.SPF3	000023	403L					
DD.SPF4	000024	404L					
DD.SPF5	000025	405L					
DD.ST	000201	412L					
DD.WDLB	000210	419L					
DD.WRD	000011	387L					
DD.WRDB	000012	388L					
DD.WRI	000006	384L					
DD.WRIB	000010	386L					
DD.WTBL	000206	417L					
DD.WTDL	000207	418L					
DD.WTL	000204	415L					
DEV1.	002377	1720	1730L				
DEV170	002355	1714L					
DEV174	002367	1710	1724L				
DEV2	003006	1718	1728	1736	1740L		
DEVICE	002273	1351	1408	1677E			
DF.CLR	000376	595E					
DF.DI	000040	220E					
DF.DS0	000002	216E					
DF.DS1	000004	217E					
DF.DS2	000010	218E					

DF.EMP	000377	594E								
DF.HD	000001	210E								
DF.MD	000020	219E								
DF.SD	000010	213E								
DF.ST	000100	221E								
DF.TO	000002	211E								
DF.WG	000001	215E								
DF.WP	000004	212E								
DF.WR	000200	222E								
DIR.ALD	000025	610L								
DIR.CLU	000015	603L								
DIR.CRD	000023	609L								
DIR.EXT	000010	598L								
DIR.FGN	000020	606L								
DIR.FLG	000016	604L								
DIR.LGN	000021	607L								
DIR.LSI	000022	608L								
DIR.NAM	000000	597L								
DIR.PRO	000013	599L								
DIR.VER	000014	600L								
DIRELEN	000027	612E	719							
DIRIDL	000015	601E								
DK.CON	000170	255E	3439	3502	3520	3532	3558			
DK.INT	000171	256E	3417	3536	3540					
DK.PORT	000170	248E	250	251	252	253	254	255	256	
DLEDS	040021	4674L								
DLY	000053	948L	3423	3446	3621	3640				
DM.MR	000000	171E								
DM.MW	000001	172E								
DM.RR	000002	173E								
DM.RW	000003	174E								
DOD	003122	1837L								
DP.DC	000177	208E	1618	1686						
DS.HOLE	000001	2803E	2828	2834						
DSPMOD	040007	4656L								
DSPROT	040006	4655L								
DY10.5	007244	2995L	3828							
DY3.3	011201	2929	3400L							
DY3.5	011211	3403	3407L							
DY3.7	007153	2932L	3410							
DY5.53	007230	2974	2978L							
DY9.3	011160	1157	3382L							
DY9.4	011171	3386	3390L							
DY9.5	003315	1999L	3394							
DY9.8	003330	2004	2008L							
DYASC	003143	1865E	1892	1905	1915	2976	3024	3830	4458	4468
DYASC1	003144	1868L	1870							
DYBYT	003160	1885L	2014	2988	3388	3396	3405	3411		
DYBYT.1	003163	1886L	4441							
DYBYT.2	003174	1890	1894L							
DYBYT.4	003213	1903	1907L							
DYBYT.6	003227	1913	1917E							
DYBYTH	016262	4434	4445L							
DYBYTH1	016303	4454	4456L							
DYBYTH2	016312	4457	4460L							
DYBYTH3	016325	4464	4466L							
DYBYTH4	016334	4467	4472L							

DYBYTO	016250	4436L						
DYBYTX	016240	1885	4426L					
DYME5.5	007221	2970E	2979	4486	4488			
DYMEM	007116	2906L	4323					
DYMEM1	007122	2912L						
DYMEM10	013252	2012	3006	3826L				
DYMEM11	007251	2997L	2998	3001	3004			
DYMEM2	007127	2914L	2920					
DYMEM3	007140	2917	2923L					
DYMEM4	007167	1148	2941	2945L				
DYMEM5	007172	2946L	2958	2962				
DYMEM6	000273	1142L	2986					
DYMEM7	000276	1143L	1146					
DYMEM9	000307	1152L	2948	2953				
DYMM5	016336	2968	4480L					
DYMSG	007265	1158	2006	2931	2943	3018L	3028	
DYMSG.5	007275	3022	3026L					
EDYMEM	007375	3077L						
EIXIT	034027	756E	1696	1698				
ERPTCNT	000012	757E	1537					
ERRMSG	014046	1615	3935L					
ERROR	000322	1005	1128	1175E	1619	1975		
ESPEED	007372	3071L						
EUC	016201	1515	3517	3529	3687	4396L		
EUC.	016215	1662	4403L					
FD.CMD	000172	251E	3420	3451	3464	3473	3481	3548
FD.DAT	000173	254E	3471	3551				
FD.SEC	000172	253E	3537					
FD.STAT	000172	250E	3425	3576				
FD.TRK	000173	252E						
FDC.FI	000320	273E	3419	3463				
FDC.RDA	000300	269E						
FDC.RDS	000200	266E	3547					
FDC.RDT	000340	270E						
FDC.RST	000000	260E	3450	3480				
FDC.SEK	000020	261E	3472					
FDC.STI	000100	263E						
FDC.STO	000140	264E						
FDC.STP	000040	262E						
FDC.WTS	000240	267E						
FDC.WTT	000360	271E						
FDF.DDM	000001	292E						
FDF.DLF	000004	290E	3547					
FDF.HLB	000010	278E						
FDF.MRF	000020	288E	3547					
FDF.S12	000001	282E						
FDF.S20	000002	283E						
FDF.S30	000003	284E	3450	3472	3480			
FDF.S6	000000	281E						
FDF.SLF	000010	289E	3547					
FDF.SS1	000002	291E						
FDF.UTR	000020	277E						
FDF.VRF	000004	279E						
FDS.BSY	000001	308E						
FDS.CRC	000010	303E	3560					
FDS.DRQ	000002	307E						
FDS.HLD	000040	298E						

FDS.IND	000002	306E						
FDS.LDT	000004	305E	3560					
FDS.NRD	000200	296E	3560					
FDS.RNF	000020	302E						
FDS.RTE	000040	299E	3560					
FDS.SEK	000020	301E						
FDS.TKO	000004	304E	3486					
FDS.WPV	000100	297E						
FDS.WTF	000040	300E						
FEDEV	013222	3817L	4615	4637				
FPLEDS	040013	4665E						
GETBND	016060	4334L						
GETBND.	016073	3323	4339L					
GETBND1	001024	1230L	4347					
GETCON	012376	3634	3646	3661	3669	3685	3696L	
GETCPT	013130	3767L	3773					
GETST	013110	3748	3758L	3761	3805			
GO	001222	1340	1362L					
GO.	000063	958L	1362	2270				
GO88	001146	1327L	4545					
GO88.1	001177	1330	1339L					
GTCON	013004	3701L	3708	3711				
GTCON1	013025	3703	3716L					
H17	002207	1632E	4603					
H17A	002237	1650L	1652					
H17X	017024	1656	4520L					
H37	011221	3416L	4625					
H37.	011317	3455L	3458	3461				
H371	011335	3448	3468L					
H371.	012002	3492L	3495					
H371B	011356	3468	3478L					
H371C	011373	3478	3486L					
H372	012045	3511	3519L					
H373	012067	3466	3487	3523	3531L			
H67	012174	3617L	4611	4633				
H671	012215	3626L	3629					
H671.	012231	3634L	3641					
H672	012250	3635	3643L					
H673	012330	3653	3672L					
H67UNI	012365	3631	3682	3689L				
H88.CTL	000362	129E	2061	2229	2907			
H88.SW	000362	133E	1708	1714	1724	2068	2098	3370
H88B.CK	000002	130E	2060					
H88B.SS	000001	131E						
H88S.0	000014	139E						
H88S.4	000003	141E	1744					
H88S.AT	000200	134E	3371					
H88S.BR	000100	135E	2069					
H88S.DV	000020	137E	1709					
H88S.M	000040	136E	2099					
HORN	002140	1576L	2637					
HRN0	002143	950	1579L					
HRN2	002160	1590L	1591					
HRNX	006045	1593	2625L					
IHA	015244	4101	4227L					
IHA.	015245	4228L	4506					
IHA1	015253	4232L	4243	4250				

```
IHA3    015301  4234    4245L
IHA4    015314  4246    4254L
IHB     015123  4107    4148L
IHB1    015126  4150L   4161    4176
IHB2    015151  4155    4165L
IHC     015214  4113    4198L
IN.     006170  1276    2613    2750E   3166    3170    3763    3778    3807    4013
IN.1    006174  2732    2753L
IN1.    006150  1281    2727E   3701    3719    3737    3758    3767    3798
INIT    000073   881     883     980L    984
INIT0   000000   879L
INIT0.0 000003   880L    2108
INITOX  004000   879     2060L
INITOX1 004050  2091L    2092    2094    4359
INIT1   000107   993L    998
INIT2   000117  1000L
INPUT   014233  4002L    4557
INTOX0  016113  2090    4354L
INT1    000010   888E
INT2    000020   904E
INT3    000030   923L
INT4    000040   930L
INT5    000050   937L
INT6    000060   955L
INT7    000070   964L
INTXIT  000172  1075L    1105    1377
IOA     015023  1232    1317    1338    2350    4040    4099L   4336    4343
IOA1    005166  1796    2440L
IOA2    005176  2445L   2460    2469
IOA3    005230  2447    2462L
IOA4    005245  2463    2473L
IOA0    003062  1796L   4103    4503
IOB     015036  4009    4047    4105L
IOB1    003070  1812L   1819    1848    1855
IOB1.5  003126  1830    1845L
IOB2    003135  1817    1852L
IOB0    003066  1811L   4109
IOC     015051  2359    2386    4111L
IOC.    005271  2446    2499L
IOCO    005266  2498L   4115
IOWRK   040002  1230    1234    2348    2812    2850    2852    3851    3873    4036    4042    4046    4049
                4334    4338    4341    4345    4381    4498    4508    4650L
IP.DS   000177  2799E   2827    2833
IP.PAD  000360   123E
IP.TPC  000371   146E
IP.TPD  000370   148E
IROC    015012  1306    1310    1329    2345    2426    3314    4095L
IROC1   005156  2419    2424L
IROCH   015326  4097    4265L   4275
IROCO   005140  2414L   4096
LRA     003047  1779L
LRA.    003052  1116    1300    1334    1780L   2246    2264
LSA.2   000037   505E
LUNM    000140   504E
M.INI   242355   333E
M.OUTI  243355   334E
MCU     015223  3951    4183    4205L
```

MEML	040074	4705E	4706										
MEMORY	016024	4315L	4566										
MEMORY.	016032	2100	3077	4317L									
MI.ANI	000346	183E											
MI.EXAF	000010	190E	1416	1418	1424	1678	1716	1726	1730	1733	1743	1867	1873
		2304	2317	2320									
MI.EXX	000331	111E	4319	4322	4428	4431	4480	4482					
MI.HLT	000166	178E	1122										
MI.IN	000333	180E	2199	2208									
MI.JIXA	000335	191E	1919	3031	4472								
MI.JIXB	000351	192E	1919	3031	4472								
MI.JIYA	000375	193E	1876										
MI.JIYB	000351	194E	1876										
MI.JMP	000303	185E	1694	3429									
MI.LDA	000072	182E											
MI.LDXA	000335	186E	1156	2003	2011	2928	2940	2985	3385	3393	3402	3409	
MI.LDXB	000041	187E	1156	2003	2011	2928	2940	2985	3385	3393	3402	3409	
MI.LDYA	000375	188E	1889	1902	1912	2973	3021	3827	4456	4466			
MI.LDYB	000041	189E	1889	1902	1912	2973	3021	3827	4456	4466			
MI.LXID	000021	184E											
MI.OUT	000323	181E	2196	2215									
MI.RET	000311	179E	2052										
MSG.BT	006234	2260	2777L										
MSG.CON	017014	4496	4512L										
MSG.EQ	007341	2000	3045L										
MSG.ERR	014336	1153	4058L										
MSG.FE	013233	3817	3821L										
MSG.GO	006165	1327	2740L										
MSG.HSS	007100	2855	2885L										
MSG.INP	014327	4002	4056L										
MSG.MEM	016043	4315	4325L										
MSG.OUT	014332	4033	4057L										
MSG.PAS	003231	1924L	2937										
MSG.PC	006214	1297	2771L										
MSG.PR	014062	3377	3939L										
MSG.RAD	014203	3947	3991L										
MSG.RAM	007303	2925	3036L										
MSG.SPD	006371	2809	2868L										
MSG.SUB	006201	2343	2765L										
MSG.VEH	002113	1245	1558L										
MSG.WRK	007062	2849	2877L										
MTR	000344	1193E	1389										
MTR.15	000354	1200L	3378										
MTR.2	000357	1202L	1218										
MTR.3	000371	1207L	1214										
MTR.4	001014	1209	1220L										
MTR1	000345	1196E	1197										
MTRA	017045	1204	4543E	4573									
MTRAL	000012	1205	4573E										
MYCNT	041122	1688	2309	4710L									
MYINT	012145	3427	3576L										
NB7	001317	1417	1424L										
NBOOT	001261	1407L	2265										
NBOOTO	001262	1408L	1435										
NMI	004116	1044	2168L										
NMI0.5	004154	2186	2189	2194L									
NMI1	004173	2181	2206L										

NMI1.5	004206	2209	2215L										
NMI2	004212	2218L											
NMI2.2	004225	2225	2227L										
NMI2.5	004236	2192	2197	2200	2216	2233L							
NMI3	004237	2204	2213	2235L									
NMIENT	000146	1044L											
NMIRET	040064	2169	2176	4697L									
NODEV	002171	1452	1510	1614E	1661	2299	2313	3085	3093	3095	3119	3138	3533
		3637	3647	3662	3670	3686							
NODEV1	002177	1617L	3819										
NOISE	006053	1597	2635L										
ONDRO	000022	2807E	2813										
OP.CTL	000360	124E	1098	1374	1385								
OP.DC	000177	2798E	2814										
OP.DIG	000360	125E											
OP.SEG	000361	126E											
OP.TPC	000371	147E											
OP.TPD	000370	149E											
OPCOOM	000037	503E											
OUT.	006063	2647E	3082	3751	4051								
OUT.1	006070	2651L	2717										
OUT1.	006140	2712E	3208	3618	3717	3731							
OUTCOM	013055	3735L											
OUTPUT	014265	4033L	4560										
OVL.IN	000001	651E											
OVL.NUM	000014	653E											
OVL.RES	000002	652E											
OVL.UCS	000200	654E											
PCA	001103	1297L	4551										
PCA1	001137	1307	1315L										
PCF.MH	011012	3268	3304L										
PCF.MO	011017	3266	3305L										
PCFA	010316	3222	3264L										
PCFA.	010341	3273L	3358										
PCFA1	010357	3277	3282L										
PCFA2	010361	3280	3284L										
PCFA3	010362	3286L	3298										
PCFA3.	010371	3291L	3294										
PCFA4	011007	3287	3300L										
PCFAA	010333	3267	3269L										
PIN	001067	1275E	1278	2697	3118	3193							
PRIM	041120	1741	2650	2715	2729	2752	3855	4007	4043	4384	4708L		
PRSL	000007	880	4663E										
PRSRAM	040004	880	4651E	4663									
PRSROM	003371	2045E	2104										
R.ABORT	033366	750E	4520	4529									
R.READ	034077	751E	1660										
R.SOP	002044	1536E	1641	1649	4524								
R.SOP1	002063	1542	1544L										
RAD.HEX	014217	3972	3983	3993L									
RAD.OCT	014211	3964	3980	3992L									
RADFLG	040071	3967	3975	4133	4317	4355	4700L						
RADIX	014072	3947L	4563										
RADIX1	014100	3950L	3960										
RADIX2	014131	3953	3964L										
RADIX3	014144	3955	3972L										
RADIX4	014160	3957	3980L										

RADIX5	014173	3982	3985L										
RCC	003262	1202	1409	1812	1966E	2414	2445	2498	2568	3859	3880	3950	4150
		4199	4232	4265									
RCC1	003262	1968L	1970										
RCC2	003270	1972L											
RCCA	040026	4679L											
RCK	003260	1945E											
RD2	006126	2697L	2702										
RDBLCK	006111	1501	2690L										
READT	012075	3504	3522	3535L									
READT1	012126	3545	3550L										
READT2	012134	3543	3556L										
REFIND	040012	4662L											
REGI	040005	1779	4654L										
REGPTR	040035	1055	1184	1782	4684L								
RESET	002011	1484L											
RETRY	002036	1528L	2318										
RI.BST	000001	463E											
RI.CON	000001	462E											
RI.DAT	000000	461E											
ROMBOOT	030000	569E											
RRDY	010142	3092	3094	3190L									
S.BAUD	040344	628L											
S.BDA	041120	727L											
S.BOOTF	041034	683L											
S.CACC	041006	667L											
S.CDB	040343	625L											
S.CFWA	040352	635L											
S.CODE	041007	668L											
S.DCS	041033	681L											
S.DDDTA	040366	646L											
S.DDGRP	040364	643L											
S.DDLDA	040360	641L											
S.DDLEN	040362	642L											
S.DDOPC	040370	647L											
S.DFWA	040354	636L											
S.DIREA	041016	675L											
S.DLINK	040346	633L											
S.DON	000040	343E	1277	2611	3167								
S.DTR	000200	345E	1277	2596									
S.ERR	000001	342E	3171										
S.FASER	041013	674L											
S.FCI	041021	676L											
S.GRT0	024000	565E											
S.GRT1	025000	566E											
S.GRT2	026000	567E											
S.GUP	041027	678L											
S.IEN	000100	344E											
S.INT	040343	579L	621										
S.JUMPS	041010	672L											
S.MOUNT	041032	680L											
S.OFWA	040350	634L											
S.OSI	041126	731L	4410	4411									
S.OSN	041004	663L											
S.OSO	041127	732L											
S.OSZ	041130	733L	4410										
S.OVLE	041000	660L											

S.OVLFL	040371	656L											
S.OVLS	040376	659L											
S.OVSTK	041035	689L											
S.RFHA	040356	637L											
S.SCI	041024	677L											
S.SCR	041121	728L											
S.SDD	041010	673L											
S.SOVR	041146	581L	583										
S.SSN	041002	662L											
S.SW0	000002	347E											
S.SW1	000004	348E											
S.SW2	000010	349E											
S.SW3	000020	350E											
S.UCSF	040372	657L											
S.UCSL	040374	658L											
S.VAL	040277	578L											
SAE	001063	1262L											
SAVALL	000132	892	908	1022L									
SAVALLR	000151	1046	1048E	2129									
SAVALLX	004105	1037	2124E										
SB.BTO	000001	364E											
SB.CRC	000010	361E											
SB.DLD	000040	359E											
SB.ILC	000002	363E											
SB.LTO	000004	362E											
SB.NRF	000020	360E											
SB.UNR	000200	357E											
SB.WPD	000100	358E											
SC.ACE	000350	809E	1868	1874	1968	1972	1989	1994	2066	2078	2081	2083	2085
		2292	2296										
SC.UART	000372	770E											
SDP3	036073	755E	1544										
SEC.M	000037	440E											
SID.0	000000	433E											
SID.1	000200	434E											
SID.M	000200	436E											
SINCR	004000	986E	988	989									
SPEED	006240	2809L	3071										
SPEED1	006257	2815L	2859										
SPEED2	006275	2823	2826L										
SPEED3	006300	2827L	2829	2840									
SPEED4	006307	2833L	2835										
SPEED5	006357	2853	2856L										
SSIZ.M	004000	444E											
SST1	001235	959	1375L										
SSTEP	001225	1370E											
ST.ERR	000002	491E											
ST.LUN	000140	489E	3693										
ST.PER	000001	492E											
ST.SPR	000034	490E											
STACK	042200	585E											
STACKL	001032	583E											
START	040000	989	2912	2945	3850	3874	4380	4649L					
START1	001265	1409L	1422										
STK	041124	1471	1512	1528	4712L								
STPRTN	001244	911	1383E										
SUBM	004370	2343L	4548										

SUBM1	005013	2353L	2366	2373					
SUBM10	014352	2385	4063L						
SUBM10.	014370	4072L	4084						
SUBM11	014373	4064	4075L						
SUBM2	005027	2359L	2380						
SUBM3	005042	2365L	2390						
SUBM4	005046	2363	2368L						
SUBM5	005053	2371L	2393						
SUBM6	005062	2369	2375L						
SUBM7	005075	2360	2382L						
SUBM8	005077	2384L	2387						
SUBM9	005105	2386L	2400						
SYDD	040130	575E							
TO.DNR	000004	536E							
TO.DNS	000005	537E							
TO.MDS	000007	539E							
TO.NIS	000001	533E							
TO.NSC	000002	534E							
TO.NST	000000	532E							
TO.NTO	000006	538E							
TO.WFT	000003	535E							
T1.BBF	000011	552E							
T1.CDE	000010	551E							
T1.DMNF	000003	546E							
T1.DTE	000006	549E							
T1.FE	000012	553E							
T1.ID	000000	543E							
T1.IDNF	000002	545E							
T1.RNF	000004	547E							
T1.SKE	000005	548E							
T1.UDE	000001	544E							
T1.WP	000007	550E							
T2.IDA	000001	558E							
T2.IFN	000002	559E							
T2.ILC	000000	557E							
TFDATA	013173	3744	3796L						
TFREQ	013176	3798L	3801	3810					
THA	016001	4119	4299L						
THA1	016007	4303L							
THB	015350	4125	4279L						
THB1	015351	4280L	4304	4306					
THB2	015366	4286	4292L						
THB3	015376	4294	4296L						
TICCNT	040033	1093	1095	1586	1687	2305	2815	2844	4682L
TMFG	041121	1753	2301	2309	4709L				
TMOUT	004304	1703	2291E						
TMOUT1	004347	2314L	2316						
TMOUT2	004363	2307	2315	2320L					
TMOUT3	004365	2319	2322L						
TMOUT4	004323	2294	2301L						
TOA	015064	1309	2353	3352	4117L	4510			
TOA.	005305	2518L	2858						
TOAD	005300	2515L	4121						
TOB	015077	1550	2355	4021	4123L				
TOB1	005332	2542L	2550						
TOBO	005322	2519	2521	2535L	4127				
TPERRX	040031	4681L							

TYPMSG	006100	1200	1246	1298	1328	1433	1616	2261	2344	2670L	2676	2810	2856
		3232	3255	3269	3818	3948	3965	3973	3987	4003	4034	4316	4497
UC.2SB	000004	835E											
UC.5BW	000000	831E											
UC.6BW	000001	832E											
UC.7BW	000002	833E											
UC.8BW	000003	834E	2082										
UC.BI	000020	854E											
UC.CTS	000020	863E											
UC.DCS	000001	859E											
UC.DDR	000002	860E											
UC.DLA	000200	840E	2065										
UC.DR	000001	850E	1969	2293									
UC.DRL	000010	862E											
UC.DSR	000040	864E											
UC.DTR	000001	843E											
UC.EDA	000001	821E											
UC.EPS	000020	837E											
UC.FE	000010	853E											
UC.IID	000006	828E											
UC.IIP	000001	827E											
UC.LOO	000020	847E											
UC.MSI	000010	824E											
UC.OR	000002	851E											
UC.OU1	000004	845E											
UC.OU2	000010	846E											
UC.PE	000004	852E											
UC.PEN	000010	836E											
UC.RI	000100	865E											
UC.RLS	000200	866E											
UC.RSI	000004	823E											
UC.RTS	000002	844E											
UC.SB	000100	839E											
UC.SKP	000040	838E											
UC.TER	000004	861E											
UC.THE	000040	855E	1869	1990									
UC.TRE	000002	822E											
UC.TSE	000100	856E											
UCI.ER	000020	792E											
UCI.IE	000002	794E											
UCI.IR	000100	790E											
UCI.RE	000004	793E											
UCI.RO	000040	791E											
UCI.TE	000001	795E											
UDR	000000	767E											
UF.FCT	000100	237E											
UF.RDA	000001	234E											
UF.ROR	000002	235E											
UF.RPE	000004	236E											
UF.TBM	000200	238E											
UIVEC	040037	923	930	937	955	964	1069	1390	1693	1704	3428	3430	4405
		4686L											
UMI.16X	000002	785E											
UMI.1B	000100	775E											
UMI.1X	000001	784E											
UMI.2B	000300	777E											
UMI.64X	000003	786E											

UMI.HB	000200	776E						
UMI.L5	000000	780E						
UMI.L6	000004	781E						
UMI.L7	000010	782E						
UMI.L8	000014	783E						
UMI.PA	000020	779E						
UMI.PE	000040	778E						
UNT.0	000000	424E						
UNT.1	000040	425E						
UNT.2	000100	426E						
UNT.3	000140	427E						
UNT.M	000140	429E						
UO.CLK	000001	203E	1067	1691				
UO.DDU	000002	202E	1178					
UO.HLT	000200	200E	1109					
UO.NFR	000100	201E	1178					
UP.DP	000174	228E						
UP.FC	000175	229E						
UP.SC	000176	231E						
UP.SR	000176	232E						
UP.ST	000175	230E						
UR.DLL	000000	816E	2081					
UR.DLM	000001	818E	2078					
UR.IER	000001	820E	2085					
UR.IIR	000002	826E						
UR.LCR	000003	830E	2066	2083				
UR.LSR	000005	849E	1868	1968	1989	2292		
UR.MCR	000004	842E						
UR.MSR	000006	858E						
UR.RBR	000000	812E	1972	2296				
UR.THR	000000	814E	1874	1994				
USERFWA	042200	586E	1494	1657	3513	3525	3546	3796 4415
USR	000001	768E						
USR.FE	000040	799E						
USR.OE	000020	800E						
USR.PE	000010	801E						
USR.RXR	000002	803E						
USR.TXE	000004	802E						
USR.TXR	000001	804E						
VEW.NPC	011024	3231	3254	3307L				
VEWHLD	040072	3317	3327	3343	4357	4701L		
VIEW	001044	1245L	4569					
VIEW1	001055	1249L	3056					
VIEW12	011111	3055	3351E					
VIEW2	002066	1250	1549L	2579				
VIEW3	006000	1556	2579L					
VIEW3.	003340	1555	2021L	3238				
VIEW3.A	003352	2024	2026L					
VIEW3A	007351	1247	3053L					
VIEW3A.	007355	2582	3055L					
VIEW3B	011127	3354	3357L					
VIEW3C	011131	3356	3358L					
VIEW4	007363	1553	3058L	3236				
VIEW5	010171	2034	3222L					
VIEW5.	010212	3227	3230L					
VIEW5A	010174	3223L	3239					
VIEW6	010224	3229	3235L	3262				

VIEW6.	010227	3234	3236L											
VIEW7	010241	3225	3242L											
VIEW7.	010302	3252	3258L											
VIEW7..	010305	3257	3259L											
VIEW7A	010270	3250	3253L											
VIEW8	011032	3053	3314L											
VIEW8.	011103	3336	3343L											
VIEW8A	011051	3315	3323L											
VIEW8B	011054	3321	3325L											
VIEW9	003355	1554	2033L	2580										
W.RES	000002	352E	1489											
WCC	003302	1125	1127	1217	1220	1421	1445	1552	1823	1988L	2357	2371	2379	
		2384	2399	2425	2449	2467	2473	2524	2547	2555	2572	2574	2636	2674
		3235	3245	3247	3258	3260	3292	3867	3879	3959	4160	4165	4235	4248
		4254	4274	4296	4308	4373								
WCC1	003303	1989L	1991											
WCR	005361	2568L	2570											
WCR.	005370	1339	1439	2516	2572L	3872	3894	3986	4019	4301				
WDN	010104	2698	3084	3137	3156L	3201								
WDN1	010111	3160L	3168											
WDN2	010137	3164	3173	3177L										
WDNA	175000	3158	3181E											
WHD	036235	740E	1651											
WNH	036271	741E	1650											
WRONG	001310	1413	1419E	1427										
WTDON1	006032	2613L	2615											
Z47	001372	1468E	4607	4629										
Z47A	001376	1473E	1530											
Z47X	010000	1490	3082L											
Z47X.	010011	3092L	3107											

21374 bytes free

Appendix B

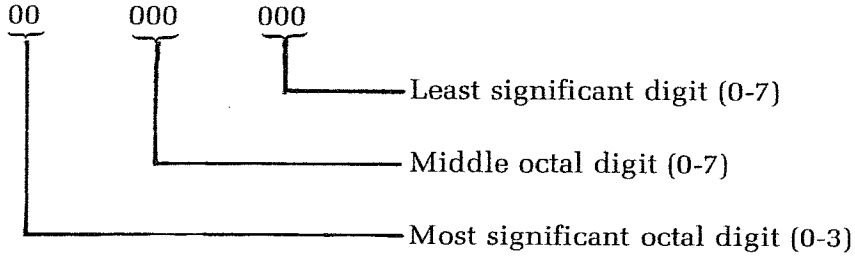
Octal Definitions

Binary numbers are converted to either hexadecimal or octal numbers for display, depending on the current radix setting. This section describes binary to octal conversion. For a description of binary to hexadecimal and octal to hexadecimal conversion, see Appendix C.

The following table shows binary to octal conversion.

BINARY NUMBER	OCTAL DIGIT
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Each 8-bit byte is displayed as three octal digits as shown below. The octal numbers lie in the range 000 to 377 for binary numbers in the range 00000000 to 11111111.



Since there are only eight bits in a byte, the most significant octal digit only represents two bits and is therefore displayed as 0 to 3. If you inadvertantly enter the octal digits 4 through 7 as the most significant digit, the most significant two bits will be interpreted in modulus 4. That is to say, entering the octal digits 4 through 7 will cause MTR-90 to interpret the digits as 0 through 3, respectively.

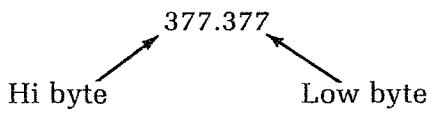
Also note that 16-bit numbers, such as memory addresses and certain register contents, are made up of two eight-bit binary numbers. The two groups of eight-bit numbers can be represented by two groups of three octal numbers in the range of 000 to 377. This representation of 16-bit binary numbers is known as offset octal or **split octal**. Where the current radix setting is octal, split octal is used consistently for displays of 16-bit numbers.

Split octal should not be confused with octal. For example:

```
11  111 111   11  111 111     A 16-bit binary number
 |   |   |    |   |   |
 3   7   7    3   7   7       Split octal representation (377 377)

 1  111 111  111 111 111      A 16-bit binary number
 |   |   |    |   |   |
 1   7   7    7   7   7       True octal representation (177777)
```

The lower representation shows true octal representation of a 16-bit binary number. True octal representation is rarely used in standard Zenith Data Systems software. Occasionally, you will see split octal numbers printed with a decimal point separating the upper and lower bytes. For example:



Note that 001.000 follows 000.377.

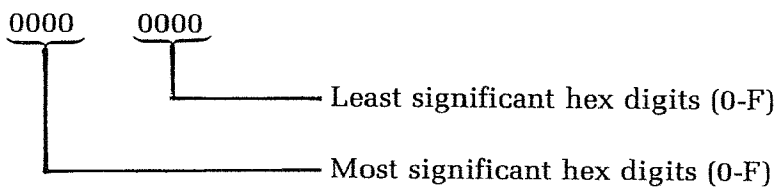
Appendix C

Hexadecimal Definitions

If the radix setting is hexadecimal, all display addresses are given in hexadecimal. The following table shows binary to hexadecimal conversion.

BINARY NUMBER	HEX DIGIT
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

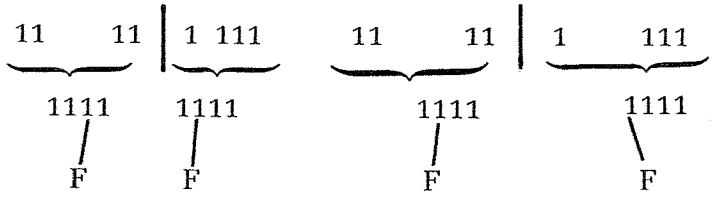
Each byte is displayed as two hexadecimal digits as shown below. Hexadecimal numbers lie in the range 00 to FF for binary numbers 00000000 to 11111111.



Converting Split Octal to Hexadecimal

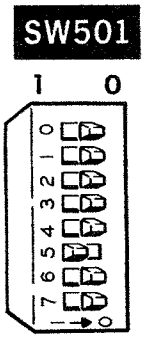
To convert a split octal number to hexadecimal, first convert the split octal number to binary. In split octal representation, each of the two bytes of a sixteen-bit number are converted independently to octal. Thus, the most significant split octal digit only represents two bits. The following illustrates the conversion to binary of the split octal value 377.377:

Having converted the split octal number to binary, regroup the bits into nybbles, then convert to hexadecimal. For example:



Appendix D

SW501 Switch Settings



DIP Switch SW501, located on the lower right of the Z-89/90 CPU logic circuit board, incorporates eight individual bits (0-7) which may be set to either one or zero. These switch settings perform the following functions:

SWITCH SECTION 7 6 5 4 3 2 1 0	DESCRIPTION
	Selects the device located at port 07CH (174Q). The settings of these two sections are:
X X X X X X 0 0 -	Hard-sectored 5.25-inch disk
X X X X X X 0 1 -	H/Z-47
X X X X X X 1 0 -	H/Z-67 hard disk/floppy
X X X X X X 1 1 -	No device
	Selects the device located at port 078H (170Q). The settings of these two sections are:
X X X X 0 0 X X -	Soft-sectored 5.25-inch disk
X X X X 0 1 X X -	H/Z-47
X X X X 1 0 X X -	H/Z-67 Winchester disk-floppy
X X X X 1 1 X X -	No device
	Determines whether the primary boot device is at port 07CH (174Q) or at 078H (170Q). The port not configured as primary becomes the secondary device.
X X X 0 X X X X -	Primary boot from device at 07CH (174Q)
X X X 1 X X X X -	Primary boot from device at 078H (170Q)
	Disables/enables memory diagnostic on power up.
X X 0 X X X X X -	Initiate memory test on power up
X X 1 X X X X X -	Disable memory test on power up
	Sets console (H/Z-19) baud rate.
X 0 X X X X X X -	Sets console baud rate at 9600
X 1 X X X X X X -	Sets console baud rate at 19,200 (not currently supported)
	Selects type of boot process
0 X X X X X X X -	Normal boot (normal)
1 X X X X X X X -	Auto boot on power up or reset (not recommended)

Appendix E

CPU Jumpers

MEMORY JUMPERS

The memory decode ROM is located at U517. An older ROM (#442-42) allowed for a maximum or 48K of memory. A newer ROM allows for up to 64K of memory. Memory jumpers should be set as follows:

Older CPU boards have four jumpers.

When you are using the old ROM (#444-42):

	JJ501	JJ502	JJ503	JJ504
16K	0	0	0	0 (or B)
32K	1	0	0	0 (or B)
64K	0	1	0	0 (or B)

When you are using the new ROM (#444-66):

16K	0	0	**	0 (or B)
32K	1	0	**	0 (or B)
48K	0	1	**	0 (or B)
64K*	1	1	**	0 (or B)

* Requires WH-88-16 Accessory Board

** A jumper is required between the center pin of JJ503 and pin 17 of P509 — or P4 of WH-88-16 (which connects to pin 17 of P509). This jumper was permanently installed on some boards.

Newer CPU boards have three jumpers.

They also have the newer decoder ROM, and the jumper wire is part of the circuit foils. These boards should not be used with the old ROM (#444-42). The jumpers are set as follows:

	JJ501	JJ502	JJ503
16K	0	0	0 (or B)
32K	1	0	0 (or B)
48K	0	1	0 (or B)
64K	1	1	0 (or B)

SECONDARY ADDRESS DECODER — U516

MTR-90 requires the use of IC #444-83 at location U516 on the left-hand side of the CPU circuit board. Four nearby jumpers (JJ505, JJ506, JJ507, and JJ508 on older units; or JJ504, JJ505, JJ506, and JJ507 on newer units) must be set as follows:

Older Units	JJ505	JJ506	JJ507	JJ508
Newer Units	JJ504	JJ505	JJ506	JJ507
	1	*	1	1 (or B)

* MTR-90 requires a jumper (#134-1159) from the center pin of JJ506 (or JJ505) to pin 14 of P508.