MONITOR

MTR-89

595-2508

Operation Manual

ZENITH

data systems

ZENITH data systems

MONITOR

MTR-89

595-2508

ZENITH DATA SYSTEMS
SAINT JOSEPH, MICHIGAN 49085

Table of Contents

INTRODUCTION

This Manual describes the functions and operation of the Z89 Monitor Program, MTR-89, that is contained in a read-only memory (ROM) in your Z89. Some of the major features of MTR-89 are:

Memory contents display and alteration.
Program execution control.
Floppy diskette boot-strap routine.

In addition, MTR-89 can be instructed (by means of a flag byte maintained in read/write memory) 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 the information in the "Operations" and "Circuit Description" sections of your Z89 Operations Manual. In order to use all of the features of MTR-89, it is necessary to understand the Z80 operation codes and the circuit of your Z89. This section gives you details of the operation of MTR-89. The listing of MTR-89 is given in Appendix A.

Power Up and Reset

MTR-89 initializes the Z89 whenever you power-up or RESET. To power-up, use the switch on the back of the Z89. To RESET, simultaneously press the RESET key and the right-hand SHIFT key on the keyboard. MTR-89 sounds the electronic "bell" and resets to its normal state. During the initialization procedure, MTR-89 determines the high limit of continuous RAM in your Z89. Once this high limit has been determined, the Z80's stack pointer is set to this value. Then MTR-89 enters a loop waiting for you to enter a command.

Clock Interrupts

The Clock Interrupt is a crucial element in the operation of the Z89. It is a level one interrupt and is generated on the Z89 CPU board every 2 ms (millisecond). MTR-89 maintains "TICCNT" which counts up one every 2 ms. See the listing in Appendix A for the location of TICCNT.

Note that MTR-89 uses interrupts, so you should not disable interrupts for a long period of time. MTR-89 also requires a stack pointer at the top of memory with at least 80 bytes.

General Operations

When you RESET or power-up your Z89, MTR-89 responds by clearing the screen and displaying "H:". This tells you that it is ready to respond to your typed commands. When you type in something, MTR-89 will either accept it or give a beep, indicating an error.

If the letter you enter is the first letter of one of MTR-89's commands, it will display the remaining letters of the word. If the letter is not the start of a command, MTR-89 will sound the "bell" and ignore the letter.

The DELETE key will kill a partially entered line and cause MTR-89 to return to the "H:" prompt. You can use this to correct typing errors.

NOTE: In this manual, the symbol "△" means type a space and "Ⓒ" means type a RETURN.

The following is a list of the acceptable MTR-89 commands. You type the first letter of the command, and MTR-89 will supply the remainder of the word. You have to press the RETURN key before MTR-89 will respond.

TABLE OF MTR-89 COMMANDS

Substitute	— Display or alter memory.
Go	— Start a program
Program Counter	— Set an address in the PC
Boot	— Boot from a diskette

These commands are described in the remainder of this Manual.

DISPLAYING AND ALTERING MEMORY

One of the major features of MTR-89 is its ability to examine the contents of any Z89 memory location and to modify the contents of that location if it is in RAM.

The Substitute command is used to display memory locations. After a memory location has been displayed, its value can be changed before you proceed to something else. There is an example showing the Substitute procedure at the end of the description. You may jump ahead to it at any time.

To start the substitution process, first type "S". MTR-89 will respond by completing the word "Substitute". You should then enter the address of the memory location you want to inspect, followed by a RETURN. This address **must** be given in split-octal. Refer to Appendix B for the definitions of octal and split-octal.

MTR-89 will respond by re-displaying the address with leading zeros. Following the address, MTR-89 will display the contents of that memory location in octal.

Once the value of the memory location has been displayed, you may change it. To change it, simply type in the new value (in octal). The new value will be inserted after you complete the next step.

NOTE: MTR-89 will use the last three digits that you enter. That is, the entry "12345" will be entered as "345". You may use this to correct errors as entries are made.

After you have inspected or changed the value of a memory location, you have three options. First, you can cause MTR-89 to advance to the next memory location and display it by pressing the Space Bar. Second, you can cause MTR-89 to retrieve the previous memory location and display it by pressing the minus key, "−". Finally, you can cause MTR-89 to return to its initial "H:" by pressing the RETURN key.

The following example shows these features. To help you follow what you enter and what the computer responds, your entries and the computer's responses are shown on different lines. If a new line is really used, the new line will start at the left of the page. Otherwise, the output is shown just down a line.

EXAMPLE

```
  H:                                computer
     S                              you
      ubstitute                     computer
                  2146 (CR)         you
002146 041                          computer
           △                        you
002147 011                          computer
           △                        you
002150 040                          computer
           —                        you
002147 011                          computer
            (CR)                    you
  H:                                computer
     S                              you
      ubstitute                     computer
                40100 (CR)          you
040100 xxx                          computer
           123 △                    you
040101 xxx                          computer
           —                        you
040100 123                          computer
            (CR)                    you
  H:                                computer
```

PROGRAM EXECUTION CONTROL

MTR-89 allows you to start a program that you have loaded into memory. It also offers a form of breakpointing.

The standard way of starting a program is to use the Go command. After you type in "G", MTR-89 responds "o". You should then type in the address (in split octal) where you want execution of your program to start. For example, if you have loaded a program at 040100, you can start it with:

```
H:  Go 40100 CR
```

MTR-89 allows another method of starting programs. MTR-89 maintains in its working memory a value for the Program Counter. If you enter "G" and then a RETURN after MTR -89 prints "o", MTR-89 will use the value in the PC as the starting address of your program.

To set the value in the Program Counter, you use the "P" command. After you enter "P", MTR-89 will respond "rogram Counter" and you can then enter the value you want. For example:

```
H: Program Counter 40100CR
H: GoCR
```

Your program will now be started at 40100.

If you do not enter a value after "P", but simply press RETURN, then MTR-89 will display the current value of the PC on the next line. You can change the PC by typing in a new value or you can leave it un-altered by pressing RETURN. For example:

```
  H: Program CounterCR
277377 40100CR
```

(You type the second number.)

When you are debugging an assembly language program, you can use MTR-89 to set breakpoints at various places in the program. To set a breakpoint, use the Substitute command and put an HLT (166 octal) instruction where you want your program to stop.

When your program reaches the breakpoint HLT instruction, it will return to MTR-89, display an "H", and then advance to a new line and display "H:". You can now use any of the MTR-89 commands.

To continue your program, you will first have to restore the byte in the location where 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, you will have to decrease the PC value by one.

Do this by entering the "P" command and a RETURN. When the current value of the PC is shown, subtract one from it, and enter this value as the new value for the PC. Remember that you have to subtract in octal, so ten minus one is seven!

Alternatively, you can use the "Go" command to start the program from whatever address you want, including from the place where you put the HLT.

Note that if the program that you are debugging uses keyboard interrupts, MTR-89 and your program may "fight" for keyboard input! Your program will always see every character because it gets them by an interrupt. MTR-89 is continually testing if a character is available, and it will never see some of the characters that you enter.

ADVANCED CONTROL

One of the advanced features of MTR-89 is its provisions allowing sophisticated users to augment or replace MTR-89's functions. This is usually done in conjunction with assembly language programs, although it is sometimes possible to use these features in BASIC using the PEEK and POKE commands.

The following discussion refers to symbols and locations in MTR-89. In order to make the most of this information, you should refer to the listing of MTR-89 that is in Appendix A. Note that at the end of the listing the definitions of RAM locations from 40.000 to 40.077 and 41.120 to 41.125 are given. Following these is a symbol reference table that will help you find where symbols are used in the program.

The Tick Counter (TICCNT)

MTR-89 maintains in memory a 16-bit (2 byte) tick counter named TICCNT. This counter is incremented when the clock interrupts occur. As long as interrupts are enabled, this will occur every 2 ms. You may set TICCNT to any value and change it as often as you like. The low-order byte of TICCNT is in location 40.033 (8219 decimal) and the high-order byte is in 40.034.

Using Interrupts

All Z89 interrupts cause control to be transferred into the lowest 64 bytes of memory. Since MTR-89 occupies this area, it processes all interrupts first. Except for level zero interrupts (RESET function), you can supply a routine to process interrupts yourself.

Control is passed out of MTR-89 through the UIVECs (user interrupt vector) that are located at 40.037 and following. Each vector is three bytes long, and contains a JMP instruction to an interrupt processing routine. MTR-89 calls or jumps to the appropriate UIVEC, and control is passed to the processing routine. The exit from an interrupt processing routine should be the return instruction, RET.

I/O Interrupts

Interrupts numbered 3 through 7 are I/O interrupts of devices that you connect to your Z89. MTR-89 does not process these interrupts, but simply passes them on to a program in RAM by jumping to the appropriate UIVEC.

Zenith Data Systems software (except MTR-89) use interrupt 3 for input and output to and from the keyboard and screen. Additionally, interrupts 4 and 7 are reserved for certain applications. These programs set UIVEC themselves. If you want to use interrupts, your program has to place the appropriate jump in the appropriate UIVEC.

Clock Interrupts

The level one interrupt is generated by hardware in your Z89 every 2 ms. MTR-89 always processes these interrupts, but you can force it to pass control to your routine once it is done.

To do this, set the appropriate jump in the first UIVEC locations. Then set the UO.CLK bit (001) in .MFLAG (40.010). MTR-89 will then pass each clock interrupt to your routine when it finishes its own processing.

Single Instructions and Breakpoint Interrupts

Level two interrupts are generated by the single-instruction hardware contained in the Z89. When a single-instruction interrupt occurs, MTR-89 processes it, and jumps to the location specified by the second UIVEC.

If you have set up UIVEC for level two interrupts, you can use RST-2 as a breakpoint instruction. Control will be returned to the location specified by the second UIVEC.

FLOPPY BOOT

MTR-89 contains the code necessary to boot-up an operating system from a floppy disk. Two forms of "Boot" let you select the device (5-1/4" or Z47) and drive number (0-2 or 0-3). "Boot Primary" refers to the device that you will use most often. "Boot Secondary" provides you with a convenient way to boot from your alternate device, if you have one.

BOOT PRIMARY

The primary boot device is selected by switch SW501 sections 4, 1, and 0 on the CPU Logic Circuit Board. This switch is preset for 5-1/4" primary device. You may change the switch sections to select Z47 primary device.

`H: Boot` ⏎	Enter "B" and "RETURN"
`H: Bootd` ⏎	5-1/4" drive primary: Enter "B" and d(drive) = 0, 1, or 2 followed by "RETURN" OR Z47 primary: Enter "B" and d(drive) = 0, 1, 2, or 3 followed by "RETURN"

BOOT SECONDARY

`H:Boot SD` ↵	Enter "B", "S", and "RETURN"
`H:Boot SDd` ↵	5-1/4" secondary: Enter "B" and "S" and d(drive) = 0, 1, or 2 followed by "RETURN" OR Z47 secondary: Enter "B" and "S" and d(drive) = 0, 1, 2, or 3 followed by "RETURN"

Use the "DELETE" key to abort the boot command and return to the monitor.

ERRORS

The console will display a "?" if any of the following conditions occur:

1. The boot device does not respond within 15 seconds.
2. The "DELETE" key is pressed.
3. Switch SW501 section 2 is set to "0".
4. A disk error occurs.

SWITCH SW501

The sections of SW501 (on the Z89 CPU logic circuit board) have been redefined as follows:

SWITCH SECTION 7 6 5 4 3 2 1 0	DESCRIPTION
X X X X X X 0 0	Port 174/177 = 5-1/4" drive
X X X X X X X 0 1	Port 174/177 = Z47
X X X X 0 0 X X	Port 170/173 = unused
X X X X 0 1 X X	Port 170/173 = Z47
X X X 0 X X X X	Boot primary from port 174/177
X X X 1 X X X X	Boot primary from port 170/173
X X 0 X X X X X	Memory test
X X 1 X X X X X	Normal
X 0 X X X X X X	Baud = 9600
0 X X X X X X X	Normal

APPENDIX A

MTR-89 LISTING

This appendix contains a listing of MTR-89. It contains all the control for primitive keyboard input and screen output. MTR-89 needs RAM locations available in locations 40.000 to 40.077 and 41.120 to 41.125, and it also needs 80 bytes of stack area in high memory.

The first few pages of the listing show definitions that are used. The last portion of the listing contains references to the symbols that are used in MTR-89. Just before this cross reference listing is the definition of RAM locations in 40.000 through 40.077.

To allow compatibility with other hardware, the MTR-89 code is segmented throughout memory. The Memory Test entry point is 7.375 and the Floppy Speed Test (5-1/4" drive) entry point is 7.372.

```
                        ***   MTR89 - H89 MONITOR                    ISSUE 09.01.00
                        *
                        *     MTR89 IS A MODIFICATION OF MTR88 BY REX CHEN IN MAY, 1980.
                        *     MTR89 IS IDENTICAL TO THE MTR88 IN THAT ALL ENTRY POINTS TO
                        *     THE CURRENT ROUTINES REMAIN UNCHANGED AND ALL ROUTINES
                        *     REMAIN UNALTERED WITH THE FOLLOWING EXCEPTIONS:
                        *
                        *       (1). ALL CODE WHICH SUPPORTS THE CASSETTE IS REMOVED.
                        *            THIS INCLUDES THE LOAD ('L') AND DUMP ('D') COMMANDS
                        *            AS WELL AS ALL OF THE DEVICE DRIVERS.
                        *       (2). 'TYPE SPACES TO DETERMINE BAUD RATE' MESSAGE IS REMOVED.
                        *       (3). THE BOOTSTRAP FOR THE Z-47 IS INSTALLED.
                        *       (4). 15 SECONDS TIME OUT FOR Z-87, OR H-17 AND Z-47 IS INSERTED.
                        *       (5). <DELETE> KEY SERVES AS AN ABORT-BOOT KEY.
                        *       (6). ALLOWS BOOT FROM SELECT DEVICE AND UNIT.
                        *
                        *
                        *     MTR88 IS AN ADAPTATION OF PAM/8 ORIGINALLY WRITTEN FOR THE
                        *     HEATH H8 COMPUTER BY J. G. LETWIN IN 1976 AND MODIFIED BY
                        *     R. N. BORCHARDT IN 1979 FOR USE IN THE HEATH H88/H89
                        *     COMPUTERS.
                        *
                        *     MTR88 PROVIDES COMPATABILITY WITH PAM/8 SUCH THAT ALL ROUTINES
                        *     HAVE RETAINED PREVIOUSLY DESCRIBED ENTRY POINTS AND ENTRY AND
                        *     EXIT CONDITIONS.  ROUTINES WHICH ARE NOT APPLICABLE SUCH AS
                        *     THOSE PERTAINING TO THE FRONT PANEL DISPLAY HAVE BEEN DELETED.
                        *
                        *
                        *     COPYRIGHT  05/1976, WINTEK CORPORATION
                        *                         902 N. 9TH ST.
                        *                         LAFAYETTE, IND.
                        *
                        *     COPYRIGHT  01/1979, HEATH COMPANY
                        *                         BENTON HARBOR, MI.
                        *
                        *     COPYRIGHT  05/1980, ZENITH DATA SYSTEMS INC.
                        *                         ST. JOSEPH, MI.
```

```
000.001                 42  .RAM.   EQU     1
                        43
000.001                 44          IF      .RAM.
                        54          ENDIF



                        56  ***     MTR88 - H88/H89 MONITOR.
                        57  *
                        58  *       THIS PROGRAM RESIDES (IN ROM) IN THE LOW 2048 BYTES OF THE HEATH
                        59  *       H88/H89 COMPUTERS.


                        61  ***     INTERRUPTS.
                        62  *
                        63  *       MTR88 IS THE PRIMARY PROCESSOR FOR ALL INTERRUPTS.
                        64  *       THEY ARE PROCESSED AS FOLLOWS:
                        65  *
                        66  *       RST     USE
                        67  *
                        68  *       0       MASTER CLEAR. (NEVER USED FOR I/O OR RST)
                        69  *
                        70  *       1       CLOCK INTERRUPT. NORMALLY TAKEN BY MTR88,
                        71  *               SETTING BIT *UO.CLK* IN BYTE *.MFLAG* ALLOWS
                        72  *               USER PROCESSING (VIA A JUMP THROUGH *UIVEC*).
                        73  *               UPON ENTRY OF THE USER ROUTINE, THE STACK
                        74  *               CONTAINS:
                        75  *               (STACK+0)  = RETURN ADDRESS (TO MTR88)
                        76  *               (STACK+2)  = (STACKPTR+14)
                        77  *               (STACK+4)  = (AF)
                        78  *               (STACK+6)  = (BC)
                        79  *               (STACK+8)  = (DE)
                        80  *               (STACK+10) = (HL)
                        81  *               (STACK+12) = (PC)
                        82  *               THE USER'S ROUTINE SHOULD RETURN TO MTR88 VIA
                        83  *               A *RET* WITHOUT ENABLING INTERRUPTS.
                        84  *
                        85  *       2       SINGLE STEP INTERRUPTS RECEIVED WHEN IN
                        86  *               USER MODE CAUSES A JUMP THROUGH *UIVEC*+3.
                        87  *               STACK UPON USER ROUTINE ENTRY:
                        88  *               (STACK+0)  = (STACKPTR+12)
                        89  *               (STACK+2)  = (AF)
                        90  *               (STACK+4)  = (BC)
                        91  *               (STACK+6)  = (DE)
                        92  *               (STACK+8)  = (HL)
                        93  *               (STACK+10) = (PC)
                        94  *               THE USER'S ROUTINE SHOULD HANDLE IT'S OWN RETURN
                        95  *               FROM THE INTERRUPT.
                        96  *
                        97  *
                        98  *       THE FOLLOWING INTERRUPTS ARE VECTORED DIRECTLY THROUGH *UIVEC*.
                        99  *       THE USER ROUTINE MUST HAVE SETUP A JUMP IN *UIVEC* BEFORE ANY
                       100  *       OF THESE INTERRUPTS MAY OCCUR.
                       101  *
```

```
                    102  *       3       I/O 3, CAUSES A DIRECT JUMP THROUGH *UIVEC*+6
                    103  *
                    104  *       4       I/O 4, CAUSES A DIRECT JUMP THROUGH *UIVEC*+9
                    105  *
                    106  *       5       I/O 5, CAUSES A DIRECT JUMP THROUGH *UIVEC*+12
                    107  *
                    108  *       6       I/O 6, CAUSES A DIRECT JUMP THROUGH *UIVEC*+15
                    109  *
                    110  *       7       I/O 7, CAUSES A DIRECT JUMP THROUGH *UIVEC*+18

                    112  **      ASSEMBLY CONSTANTS

000,000             114          XTEXT   MTR88           DEFINE MTR88 OLD EQUATES
```

```
                        117X **     IO PORTS
                        118X
                        119X ***    ALL REFERENCES TO THE H8 FRONT PANEL PORTS ARE TRAPPED BY THE
                        120X *      Z80 NMI OF THE H88/H89.  OP.CTL WILL STILL PERFORM AS IN AN H8
                        121X *      IN RESPECT TO THE CLOCK AND SINGLE STEP CONTROL.  FOR MORE
                        122X *      INFORMATION SEE THE NMI ROUTINE.
                        123X
    000.360             124X IP.PAD EQU     360Q            PAD INPUT PORT
    000.360             125X OP.CTL EQU     360Q            CONTROL OUTPUT PORT
    000.360             126X OP.DIG EQU     360Q            DIGIT SELECT OUTPUT PORT
    000.361             127X OP.SEG EQU     361Q            SEGMENT SELECT OUTPUT PORT
                        128X
                        129X *      H88/H89 CONTROL PORT
    000.362             130X H88.CTL EQU    362Q            H88/H89 PORT FOR CLOCK AND SINGLE STEP
    000.002             131X H88B.CK EQU    00000010B       2MS CLOCK ENABLE/DISABLE
    000.001             132X H88B.SS EQU    00000001B       SINGLE STEP ENABLE/DISABLE
                        133X
    000.362             134X H88.SW EQU     362Q            8 POSITION DIP SWITCH
    000.200             135X H88S.AT EQU    10000000B       AUTO BOOT SWITCH
    000.100             136X H88S.BR EQU    01000000B       BAUD RATE SWITCH        **/RNC/**
    000.040             137X H88S.M EQU     00100000B       MEMORY TEST/NORMAL OPERATION SWITCH
    000.020             138X H88S.DV EQU    00010000B       = 0, BOOT FROM DEVICE AT 174-177Q
                        139X *                              = 1, BOOT FROM DEVICE AT 170-173Q
    000.014             140X H88S.0 EQU     00001100B       = 00, NO DEVICE INSTALLED AT 170-173Q
                        141X *                              = 01, DEVICE AT 170-173Q = Z47
    000.003             142X H88S.4 EQU     00000011B       = 00, DEVICE AT 174-177Q = H17
                        143X *                              = 01, DEVICE AT 174-177Q = Z47


                        145X **     CASSETTE PORTS
                        146X
    000.371             147X IP.TPC EQU     371Q            TAPE CONTROL IN
    000.371             148X OP.TPC EQU     371Q            TAPE CONTROL OUT
    000.370             149X IP.TPD EQU     370Q            TAPE DATA IN
    000.370             150X OP.TPD EQU     370Q            TAPE DATA OUT


                        152X **     ASCII CHARACTERS.
                        153X
    000.026             154X A.SYN  EQU     026Q            SYNC CHARACTER
    000.002             155X A.STX  EQU     002Q            STX CHARACTER
    000.007             156X A.BEL  EQU     007Q            BELL CHARACTER
    000.010             157X A.BKS  EQU     010Q            BACKSPACE CHARACTER
    000.012             158X A.LF   EQU     012Q            LINE FEED CHARACTER
    000.015             159X A.CR   EQU     015Q            CARRIAGE RETURN CHARACTER
    000.033             160X A.ESC  EQU     033Q            ESCAPE CHARACTER
    000.177             161X A.DEL  EQU     177Q            DELETE OR RUBOUT CHARACTER
```

```
                        163X **     FRONT PANEL HARDWARE CONTROL BITS.
                        164X
    000.020             165X CB.SSI EQU     00010000B       SINGLE STEP INTERRUPT
    000.040             166X CB.MTL EQU     00100000B       MONITOR LIGHT
    000.100             167X CB.CLI EQU     01000000B       CLOCK INTERRUPT ENABLE
    000.200             168X CB.SPK EQU     10000000B       SPEAKER ENABLE


                        170X **     DISPLAY MODE FLAGS (IN *DSPMOD*)
                        171X
    000.000             172X DM.MR  EQU     0               MEMORY READ
    000.001             173X DM.MW  EQU     1               MEMORY WRITE
    000.002             174X DM.RR  EQU     2               REGISTER READ
    000.003             175X DM.RW  EQU     3               REGISTER WRITE


                        177X **     MACHINE INSTRUCTIONS.
                        178X
    000.166             179X MI.HLT EQU     01110110B       HALT
    000.311             180X MI.RET EQU     11001001B       RETURN
    000.333             181X MI.IN  EQU     11011011B       INPUT
    000.323             182X MI.OUT EQU     11010011B       OUTPUT
    000.072             183X MI.LDA EQU     00111010B       LDA
    000.346             184X MI.ANI EQU     11100110B       ANI
    000.021             185X MI.LXID EQU    00010001B       LXI D
    000.303             186X MI.JMP EQU     11000011B       JMP
    000.335             187X MI.LDXA EQU    11011101B       LD IX, (BYTE A)
    000.041             188X MI.LDXB EQU    00100001B       LD IX, (BYTE B)
    000.375             189X MI.LDYA EQU    11111101B       LD IY, (BYTE A)
    000.041             190X MI.LDYB EQU    00100001B       LD IY, (BYTE B)
    000.010             191X MI.EXAF EQU    00001000B       EX AF,AF'
    000.335             192X MI.JIXA EQU    11011101B       JP (IX)  (BYTE A)
    000.351             193X MI.JIXB EQU    11101001B       JP (IX)  (BYTE B)
    000.375             194X MI.JIYA EQU    11111101B       JP (IY)  (BYTE A)
    000.351             195X MI.JIYB EQU    11101001B       JP (IY)  (BYTE B)


                        197X **     USER OPTION BITS.
                        198X *
                        199X *      THESE BITS ARE SET IN CELL .MFLAG.
                        200X
    000.200             201X UO.HLT EQU     10000000B       DISABLE HALT PROCESSING
    000.100             202X UO.NFR EQU     CB.CLI          NO REFRESH OF FRONT PANEL
    000.002             203X UO.DDU EQU     00000010B       DISABLE DISPLAY UPDATE
    000.001             204X UO.CLK EQU     00000001B       ALLOW PRIVATE INTERRUPT PROCESSING
    000.000             205         XTEXT   Z47DEF          DEFINE Z47 EQUATES
```

```
                        208X *
                        209X **     DISK INTERFACE CONSTANTS
                        210X *
    000.170             211X D.STA  EQU     170Q                INTERFACE STATUS PORT
    000.171             212X D.DAT  EQU     D.STA+1             INTERFACE DATA PORT
                        213X *
    000.001             214X S.ERR  EQU     00000001B           ERROR BIT
    000.040             215X S.DON  EQU     00100000B           DONE
    000.200             216X S.DTR  EQU     10000000B           DATA TRANSFER REQUEST
                        217X *
    000.002             218X W.RES  EQU     00000010B           RESET COMMAND


                        220X **     CONTROLLER STATUS REGISTER
                        221X *
    000.200             222X CS.UNR EQU     10000000B           UNIT NOT READY
    000.100             223X CS.WPD EQU     01000000B           WRITE PROTECTED DRIVE


                        225X **     AUXILLARY STATUS REGISTER
                        226X *
    000.100             227X AS.0DD EQU     01000000B           TRACK 0 DOUBLE DENSITY
    000.040             228X AS.1DD EQU     00100000B           TRACK 1 - 76 DOUBLE DENSITY
    000.020             229X AS.S1A EQU     00010000B           SIDE 1 AVAILABLE
    000.003             230X AS.SLW EQU     00000011B           SECTOR LENGTH MASK


                        232X **     DISK COMMANDS
                        233X *
    000.000             234X DC.BOOT EQU    0                   BOOT
    000.001             235X DC.RST EQU     1                   READ CONTROLLER STATUS
    000.002             236X DC.RAS EQU     2                   READ AUX. STATUS
    000.003             237X DC.LSC EQU     3                   LOAD SECTOR COUNT
    000.004             238X DC.RAD EQU     4                   READ ADDR. OF LAST SECTOR ACCESSED
    000.005             239X DC.REA EQU     5                   READ SECTORS
    000.006             240X DC.WRI EQU     6                   WRITE SECTORS
    000.007             241X DC.REAB EQU    7                   READ SECTORS BUFFERED
    000.010             242X DC.WRIB EQU    8                   WRITE SECTORS BUFFERED
    000.011             243X DC.WRD EQU     9                   WRITE SECTORS & DELETE
    000.012             244X DC.WRDB EQU    10                  WRITE SECTORS BUFFERED & DELETE
    000.013             245X DC.CPY EQU     11                  COPY
    000.014             246X DC.FRM0 EQU    12                  FORMAT IBM SD
    000.015             247X DC.FRM1 EQU    13                  FORMAT     SD
    000.016             248X DC.FRM2 EQU    14                  FORMAT IBM DD
    000.017             249X DC.FRM3 EQU    15                  FORMAT     DD
```

```
                        251X **      USEFUL FLAGS
                        252X *
    000.000             253X UNT.0   EQU     00000000B          UNIT 0
    000.040             254X UNT.1   EQU     00100000B          UNIT 1
    000.100             255X UNT.2   EQU     01000000B          UNIT 2
    000.140             256X UNT.3   EQU     01100000B          UNIT 3
                        257X *
    001.000             258X C.256   EQU     256                SECTOR SIZE = 256 BYTES
    000.200             259X C.128   EQU     128                SECTOR SIZE
    000.000             260          XTEXT   H17DEF        EQUATES FOR H17 BOOT ROM
```

```
                        263X **     H17 CONTROL INFORMATION.
                        264X
    000,177             265X DF.DC  EQU   07FH          DISK CONTROL PORT
                        266X
    000,001             267X DF.HD  EQU   00000001B     HOLE DETECT
    000.002             268X DF.T0  EQU   00000010B     TRACK 0 DETECT
    000,004             269X DF.WP  EQU   00000100B     WRITE PROTECT
    000.010             270X DF.SD  EQU   00001000B     SYNC DETECT
                        271X
    000.001             272X DF.WG  EQU   00000001B     WRITE GATE ENABLE
    000,002             273X DF.DS0 EQU   00000010B     DRIVE SELECT 0
    000.004             274X DF.DS1 EQU   00000100B     DRIVE SELECT 1
    000,010             275X DF.DS2 EQU   00001000B     DRIVE SELECT 2
    000.020             276X DF.MO  EQU   00010000B     MOTOR ON (BOTH DRIVES)
    000,040             277X DF.DI  EQU   00100000B     DIRECTION (0=OUT)
    000.100             278X DF.ST  EQU   01000000B     STEP COMMAND (ACTIVE HIGH)
    000,200             279X DF.WR  EQU   10000000B     WRITE ENABLE RAM
                        280X
                        281X
                        282X
                        283X **     DISK UART PORTS AND CONTROL FLAGS.
                        284X
    000,174             285X UF.DP  EQU   07CH          DATA PORT
    000.175             286X UF.FC  EQU   07DH          FILL CHARACTER
    000,175             287X UF.ST  EQU   07DH          STATUS FLAGS
    000.176             288X UF.SC  EQU   07EH          SYN CHARACTER (OUTPUT)
    000,176             289X UF.SR  EQU   07EH          SYNC RESET (INPUT)
                        290X
    000,001             291X UF.RDA EQU   00000001B     RECEIVE DATA AVAILABLE
    000.002             292X UF.ROR EQU   00000010B     RECEIVER OVERRUN
    000,004             293X UF.RPE EQU   00000100B     RECEIVER PARITY ERROR
    000,100             294X UF.FCT EQU   01000000B     FILL CHAR TRANSMITTED
    000,200             295X UF.TBM EQU   10000000B     TRANSMITTER BUFFER EMPTY
                        296X
                        297X
                        298X
                        299X **     CHARACTER DEFINITIONS.
                        300X
    000.375             301X C.PSYN EQU   0FDH          PREFIX SYNC CHARACTER
    000.000             302         XTEXT HOSEQU        HDOS EQUATES
```

```
                        305X **      HDOS SYSTEM EQUIVALENCES.
                        306X *
                        307X
    024.000             308X S.GRT   EQU    24000A          SYSTEM AREA FOR GRT0
    025.000             309X S.GRT1  EQU    25000A          SYSTEM AREA FOR GRT 1
    026.000             310X SECSCR  EQU    26000A          SYSTEM 512 BYTE SCRATCH AREA
    030.000             311X ROMBOOT EQU    30000A          ROM BOOT ENTRY
                        312X
    040.100             313X         ORG    40100A          FREE SPACE FROM PAM-8
                        314X
    040.100             315X         DS     8               JUMP TO SYSTEM EXIT
    040.110             316X D.CON   DS     16              DISK CONSTANTS
    040.130             317X SYDD    EQU    *               SYSTEM DISK ENTRY POINT
    040.130             318X D.VEC   DS     24*3            SYSTEM ROM ENTRY VECTORS
    040.240             319X D.RAM   DS     31              SYSTEM ROM WORK AREA
    040.277             320X S.VAL   DS     36              SYSTEM VALUES
    040.343             321X S.INT   DS     115             SYSTEM INTERNAL WORK AREAS
    041.126             322X         DS     16
    041.146             323X S.SOVR  DS     2               STACK OVERFLOW WARNING
    041.150             324X         DS     42200A-*        SYSTEM STACK
    001.032             325X STACKL  EQU    *-S.SOVR        STACK SIZE
                        326X
    042.200             327X STACK   EQU    *               LWA+1 SYSTEM STACK
    042.200             328X USERFWA EQU    *               USER FWA
    042.200             329          XTEXT  MISC            MISCELLANEOUS EQUATES FOR H17 BOOT ROM
```

```
                        332X ** MISCELLANEOUS EQUATES FROM H17 BOOT ROM.
                        333X *  REFER TO H17 BOOT ROM IF MORE INFROMATION DESIRED
                        334X
    036.235             335X WHD      EQU     36235A          WAIT FOR HOLE ROUTINE ENTRY POINT
    036.271             336X WNH      EQU     36271A          WAIT FOR NO HOLE ROUTINE ENTRY POINT
                        337X
    000.130             338X BOOTAL   EQU     130A            NUMBER OF RAM TO CLEAR
    037.132             339X BOOTA    EQU     37132A          RAM CLEAR START LOCATION
    030.252             340X $MOVE    EQU     30252A          MOVE DATA ROUTINE
    000.037             341X D.RAML   EQU     37Q
    031.212             342X $ZERO    EQU     31212A          ZERO RAM ROUTINE
    041.061             343X AIO.UNI  EQU     41061A          DISK UNIT NUMBER STORAGE
    040.037             344X .UIVEC   EQU     40037A          USER INTERRUPT VECTOR
    034.031             345X CLOCK17  EQU     34031A          Z17 TIMER INTERRUPT HANDLER LOCATION
    033.366             346X R.ABORT  EQU     33366A          RESET Z17 ROUTINE LOCATION
    034.077             347X R.READ   EQU     34077A          READ Z17 ROUTINE LOCATION
    040.206             348X D.SDP    EQU     40206A          SET DEVICE PARAMETER RAM LOCATION
    036.073             349X SDP3     EQU     36073A          SET DEVICE PARAMETER ENTRY
    034.027             350X EIXIT    EQU     34027A          EI/RET LOCATION
    000.012             351X ERPTCNT  EQU     12Q             ERROR COUNT
    040.264             352X D.DECNT  EQU     40264A
    042.200             353           XTEXT   U8251          DEFINE 8251 USART BITS
```

```
                        356X **     8251 USART BIT DEFINITIONS.
                        357X *
                        358X
                        359X **     PORT ADDRESSES
                        360X
000.000                 361X UDR    EQU     0           DATA REGISTER IS EVEN
000.001                 362X USR    EQU     1           STATUS REGISTER IS NEXT
                        363X
000.372                 364X SC.UART EQU    372Q        CONSOLE USART ADDRESS (IFF 8251)
                        365X
                        366X
                        367X **     MODE INSTRUCTION CONTROL BITS.
                        368X
000.100                 369X UMI.1B EQU     01000000B   1 STOP BIT
000.200                 370X UMI.HB EQU     10000000B   1 1/2 STOP BITS
000.300                 371X UMI.2B EQU     11000000B   2 STOP BITS
000.040                 372X UMI.PE EQU     00100000B   EVEN PARITY
000.020                 373X UMI.PA EQU     00010000B   USE PARITY
000.000                 374X UMI.L5 EQU     00000000B   5 BIT CHARACTERS
000.004                 375X UMI.L6 EQU     00000100B   6 BIT CHARACTERS
000.010                 376X UMI.L7 EQU     00001000B   7 BIT CHARACTERS
000.014                 377X UMI.L8 EQU     00001100B   8 BIT CHARACTERS
000.001                 378X UMI.1X EQU     00000001B   CLOCK X 1
000.002                 379X UMI.16X EQU    00000010B   CLOCK X 16
000.003                 380X UMI.64X EQU    00000011B   CLOCK X 64
                        381X
                        382X **     COMMAND INSTRUCTION BITS.
                        383X
000.100                 384X UCI.IR EQU     01000000B   INTERNAL RESET
000.040                 385X UCI.RO EQU     00100000B   READER-ON CONTROL FLAG
000.020                 386X UCI.ER EQU     00010000B   ERROR RESET
000.004                 387X UCI.RE EQU     00000100B   RECEIVE ENABLE
000.002                 388X UCI.IE EQU     00000010B   ENABLE INTERRUPTS FLAG
000.001                 389X UCI.TE EQU     00000001B   TRANSMIT ENABLE
                        390X
                        391X **     STATUS READ COMMAND BITS.
                        392X
000.040                 393X USR.FE EQU     00100000B   FRAMING ERROR
000.020                 394X USR.OE EQU     00010000B   OVERRUN ERROR
000.010                 395X USR.PE EQU     00001000B   PARITY ERROR
000.004                 396X USR.TXE EQU    00000100B   TRANSMITTER EMPTY
000.002                 397X USR.RXR EQU    00000010B   RECEIVER READY
000.001                 398X USR.TXR EQU    00000001B   TRANSMITTER READY
042.200                 399         XTEXT   U8250       DEFINE 8250 ACE BITS
```

```
                        402X **      8250 UART CONTROL AND BIT DEFINITIONS.
                        403X
000.350                 404X SC.ACE  EQU     350Q            SYSTEM CONSOLE PORT IF 8250 ACE
000.156                 405X AC.DLY  EQU     110             220 MIL. SEC. DELAY FOR 8250
                        406X
000.000                 407X UR.RBR  EQU     0               RECEIVER BUFFER REGISTER (READ ONLY)
                        408X
000.000                 409X UR.THR  EQU     0               TRANSMITTER HOLDING REGISTER (WRITE ONLY)
                        410X
000.000                 411X UR.DLL  EQU     0               DIVISOR LATCH (LEAST SIGNIFICANT)
                        412X
000.001                 413X UR.DLM  EQU     1               DIVISOR LATCH (MOST SIGNIFICANT)
                        414X
000.001                 415X UR.IER  EQU     1               INTERRUPT ENABLE REGISTER
000.001                 416X UC.EDA  EQU     00000001B        ENABLE RECEIVED DATA AVAILABLE INTERRUPT
000.002                 417X UC.TRE  EQU     00000010B        ENABLE TRANSMIT HOLD REGISTER EMPTY INTERRUPT
000.004                 418X UC.RSI  EQU     00000100B        ENABLE RECEIVE STATUS INTERRUPT
000.010                 419X UC.MSI  EQU     00001000B        ENABLE MODEM STATUS INTERRUPT
                        420X
000.002                 421X UR.IIR  EQU     2               INTERRUPT IDENTIFICATION REGISTER
000.001                 422X UC.IIP  EQU     00000001B        INVERTED INTERRUPT PENDING (0 MEANS PENDING)
000.006                 423X UC.IID  EQU     00000110B        INTERRUPT ID
                        424X
000.003                 425X UR.LCR  EQU     3               LINE CONTROL REGISTER
000.000                 426X UC.5BW  EQU     00000000B        5 BIT WORDS
000.001                 427X UC.6BW  EQU     00000001B        6 BIT WORDS
000.002                 428X UC.7BW  EQU     00000010B        7 BIT WORDS
000.003                 429X UC.8BW  EQU     00000011B        8 BIT WORDS
000.004                 430X UC.2SB  EQU     00000100B        TWO STOP BITS SELECTED
000.010                 431X UC.PEN  EQU     00001000B        PARITY COMPUTATION ENABLED
000.020                 432X UC.EPS  EQU     00010000B        EVEN PARITY SELECT
000.040                 433X UC.SKP  EQU     00100000B        STICK PARITY
000.100                 434X UC.SB   EQU     01000000B        SET BREAK
000.200                 435X UC.DLA  EQU     10000000B        DIVISOR LATCH ACCESS
                        436X
000.004                 437X UR.MCR  EQU     4               MODEM CONTROL REGISTER
000.001                 438X UC.DTR  EQU     00000001B        DATA TERMINAL READY
000.002                 439X UC.RTS  EQU     00000010B        REQUEST TO SEND
000.004                 440X UC.OU1  EQU     00000100B        OUT 1
000.010                 441X UC.OU2  EQU     00001000B        OUT 2
000.020                 442X UC.LOO  EQU     00010000B        LOOP
                        443X
000.005                 444X UR.LSR  EQU     5               LINE STATUS REGISTER
000.001                 445X UC.DR   EQU     00000001B        DATA READY
000.002                 446X UC.OR   EQU     00000010B        OVERRUN
000.004                 447X UC.PE   EQU     00000100B        PARITY ERROR
000.010                 448X UC.FE   EQU     00001000B        FRAMING ERROR
000.020                 449X UC.BI   EQU     00010000B        BREAK INTERRUPT
000.040                 450X UC.THE  EQU     00100000B        TRANSMITTER HOLDING REGISTER EMPTY
000.100                 451X UC.TSE  EQU     01000000B        TRANSMITTER SHIFT REGISTER EMPTY
                        452X
000.006                 453X UR.MSR  EQU     6               MODEM STATUS REGISTER
000.001                 454X UC.DCS  EQU     00000001B        DELTA CLEAR TO SEND
000.002                 455X UC.DDR  EQU     00000010B        DELTA DATA SET READY
000.004                 456X UC.TER  EQU     00000100B        TRAILING EDGE OF RING
000.010                 457X UC.DRL  EQU     00001000B        DELTA RECEIVE LINE SIGNAL DETECT
```

```
000.020                 458X UC.CTS   EQU     00010000B       CLEAR TO SEND
000.040                 459X UC.DSR   EQU     00100000B       DATA SET READY
000.100                 460X UC.RI    EQU     01000000B       RING INDICATOR
000.200                 461X UC.RLS   EQU     10000000B       RECEIVED LINE SIGNAL DETECT
```

```
                                464  ***     INTERRUPT VECTORS.
                                465  *
                                466

                                468  **      LEVEL 0 - RESET
                                469  *
                                470  *       THIS 'INTERRUPT' MAY NOT BE PROCESSED BY A USER PROGRAM.
                                471
    000.001                     472          IF      .RAM.
                                484          ELSE
    000.000                     485          ORG     00A
                                486          ENDIF
                                487
    000.000  303 000 004        488  INIT0   JMP     INIT0X          DO H88 EXTENSION OF INITIALIZATION
    000.003  041 012 040        489  INIT0.0 LXI     H,PRSRAM+PRSL-1 (HL) = RAM DESTINATION FOR CODE
    000.006  303 073 000        490          JMP     INIT            INITIALIZE
                                491
    000.001                     492          IF      .RAM.
                                493          ELSE
    377.073                     494          ERRPL   INIT-1000A      BYTE IN WORD 10A MUST BE 0
                                495          ENDIF
                                496

                                498  **      LEVEL 1 - CLOCK
                                499
    000.001                     500          IF      .RAM.
                                502          ELSE
    000.010                     503  INT1    EQU     10Q             INTERRUPT ENTRY POINT
                                504
    000.000                     505          ERRNZ   *-11Q           INT0 TAKES UP ONE BYTE
                                506          ENDIF
                                507
    000.011  315 132 000        508          CALL    SAVALL          SAVE USER REGISTERS
    000.014  026 000            509          MVI     D,0
    000.016  303 201 000        510          JMP     CLOCK           PROCESS CLOCK INTERRUPT
    000.001                     511          IF      .RAM.
                                512          ELSE
    377.201                     513          ERRPL   CLOCK-1000A     EXTRA BYTE MUST BE 0
                                514          ENDIF

                                516  **      LEVEL 2 - SINGLE STEP
                                517  *
                                518  *       IF THIS INTERRUPT IS RECEIVED WHEN NOT IN MONITOR MODE,
                                519  *       THEN IT IS ASSUMED TO BE GENERATED BY A USER PROGRAM
                                520  *       (SINGLE STEPPING OR BREAKPOINTING). IN SUCH CASE, THE
                                521  *       USER PROGRAM IS ENTERED THROUGH (UIVEC+3
                                522
    000.001                     523          IF      .RAM.
                                525          ELSE
    000.020                     526  INT2    EQU     20A             LEVEL 2 ENTRY
```

```
                                527
000.000                         528             ERRNZ   *-21A           INT1 TAKES EXTRA BYTE
                                529             ENDIF
                                530
000,021  315 132 000            531             CALL    SAVALL          SAVE REGISTERS
000.024  032                    532             LDAX    D               (A) = (CTLFLG)
040.011                         533 ,           SET     CTLFLG
000.025  303 244 001            534             JMP     STPRTN          STEP RETURN


                                536 ***     I/O INTERRUPT VECTORS.
                                537 *
                                538 *       INTERRUPTS 3 THROUGH 7 ARE AVAILABLE FOR GENERAL I/O USE.
                                539 *
                                540 *       THESE INTERRUPTS ARE NOT SUPPORTED BY MTR88, AND SHOULD
                                541 *       NEVER OCCUR UNLESS THE USER HAS SUPPLIED HANDLER ROUTINES
                                542 *       (THROUGH UIVEC)
                                543
000.001                         544             IF      .RAM.
                                545             ELSE
000.030                         546             ORG     30A
                                547             ENDIF
                                548
000,030  303 045 040            549 INT3        JMP     UIVEC+6         JUMP TO USER ROUTINE
                                550
000,033  064 064 064            551             DB      '44440'         HEATH PART NUMBER 444-40


                                553
000.001                         554             IF      .RAM.
                                555             ELSE
000.040                         556             ORG     40A
                                557             ENDIF
                                558
000,040  303 050 040            559 INT4        JMP     UIVEC+9         JUMP TO USER ROUTINE
                                560
000,043  044 122 116            561             DB      44Q,122Q,116Q,102Q,44Q  SUPPORT CODE


                                563
000.001                         564             IF      .RAM.
                                565             ELSE
000.050                         566             ORG     50A
                                567             ENDIF
                                568
000.050  303 053 040            569 INT5        JMP     UIVEC+12        JUMP TO USER ROUTINE
                                570
                                571
                                572 **      DLY - DELAY TIME INTERVAL.
                                573 *
                                574 *       ENTRY   (A) = MILLISECOND DELAY COUNT/2
                                575 *       EXIT    NONE
                                576 *       USES    A,F
```

```
                                577
000.001                         578             IF      .RAM.
                                579             ELSE
000.000                         580             ERRNZ   *-53A
                                581             ENDIF
                                582
000.053  365                    583  DLY        PUSH    PSW             SAVE COUNT
000.054  257                    584             XRA     A               DONT SOUND HORN
000.055  303 143 002            585             JMP     HRNO            PROCESS AS HORN


                                587
000.001                         588             IF      .RAM.
                                589             ELSE
000.060                         590             ORG     60A
                                591             ENDIF
                                592
000.060  303 056 040            593  INT6       JMP     UIVEC+15        JUMP TO USER ROUTINE
                                594
                                595
000.063  076 320                596  GO.        MVI     A,CB.SSI+CB.CLI+CB.SPK  OFF MONITOR MODE LIGHT
000.065  303 235 001            597             JMP     SST1            RETURN TO USER PROGRAM


                                599
000.001                         600             IF      .RAM.
                                601             ELSE
000.070                         602             ORG     70A
                                603             ENDIF
                                604
000.070  303 061 040            605  INT7       JMP     UIVEC+18        JUMP TO USER ROUTINE
```

```
                                608  **      INIT - INITIALIZE SYSTEM
                                609  *
                                610  *       INIT IS CALLED WHENEVER A HARDWARE MASTER-CLEAR IS INITIATED.
                                611  *
                                612  *       SETUP MTR88 CONTROL CELLS IN RAM,
                                613  *       DECODE HOW MUCH MEMORY EXISTS, SETUP STACKPOINTER, AND
                                614  *       ENTER THE MONITOR LOOP.
                                615  *
                                616  *       ENTRY   FROM MASTER CLEAR
                                617  *       EXIT    INTO MTR88 MAIN LOOP
                                618
    000.001                     619          IF      .RAM.
                                620          ELSE
    000.000                     621          ERRNZ   *-73Q
                                622          ENDIF
                                623
    000.073  032                624  INIT    LDAX    D               COPY *PRSROM* INTO RAM
    000.074  167                625          MOV     M,A             MOVE BYTE
    000.075  053                626          DCX     H               DECREMENT DESTINATION
    000.076  034                627          INR     E               INCREMENT SOURCE
    000.077  302 073 000        628          JNZ     INIT            IF NOT DONE
                                629
    004.000                     630  SINCR   EQU     4000A           SEARCH INCREMENT
                                631
    000.102  026 004            632          MVI     D,SINCR/256     (DE) = SEARCH INCREMENT
    000.104  041 000 034        633          LXI     H,START-SINCR   (HL) = FIRST RAM - SEARCH INCREMENT
                                634
                                635  *       DETERMINE MEMORY LIMIT.
                                636
    000.107  167                637  INIT1   MOV     M,A             RESTORE VALUE READ
    000.110  031                638          DAD     D               INCREMENT TRIAL ADDRESS
    000.111  176                639          MOV     A,M             (A) = CURRENT MEMORY VALUE
    000.112  065                640          DCR     M               TRY TO CHANGE IT
    000.113  276                641          CMP     M
    000.114  302 107 000        642          JNE     INIT1           IF MEMORY CHANGED
                                643
    000.117  053                644  INIT2   DCX     H
                                645
    000.001                     646          IF      .RAM.
                                649          ELSE
    000.120  371                650          SPHL                    SET STACKPOINTER = MEMORY LIMIT -1
                                651          ENDIF
                                652
    000.121  345                653          PUSH    H               SET *PC* VALUE ON STACK
    000.122  041 322 000        654          LXI     H,ERROR
    000.125  345                655          PUSH    H               SET 'RETURN ADDRESS'
                                656
                                657  *       CONFIGURE LOAD/DUMP UART
                                658
    000.126  076 116            659          MVI     A,UMI.1B+UMI.L8+UMI.16X
    000.130  323 371            660          OUT     OP.TPC          SET 8 BIT, NO PARITY, 1 STOP, X16
```

```
                                663  **      SAVALL - SAVE ALL REGISTERS ON STACK.
                                664  *
                                665  *       SAVALL IS CALLED WHEN AN INTERRUPT IS ACCEPTED, IN ORDER TO
                                666  *       SAVE THE CONTENTS OF THE REGISTERS ON THE STACK.
                                667  *
                                668  *       ENTRY   CALLED DIRECTLY FROM INTERRUPT ROUTINE.
                                669  *       EXIT    ALL REGISTERS PUSHED ON STACK,
                                670  *               IF NOT YET IN MONITOR MODE, REGPTR = ADDRESS OF REGISTERS
                                671  *               ON STACK.
                                672  *               (DE) = ADDRESS OF CTLFLG
                                673
    000.001                     674          IF      .RAM.
                                675          ELSE
    000.000                     676          ERRNZ   *-132A
                                677          ENDIF
                                678
    000.132  343                679  SAVALL  XTHL                    SET H,L ON STACK TOP
    000.133  325                680          PUSH    D
    000.134  305                681          PUSH    B
    000.135  365                682          PUSH    PSW
    000.136  353                683          XCHG                    (D,E) = RETURN ADDRESS
    000.137  041 012 000        684          LXI     H,10
    000.142  071                685          DAD     SP              (H,L) = ADDRESS OF USERS SP
                                686
                                687  **      REPLACE THESE INSTRUCTIONS WITH A JUMP AROUND THE NMI VECTOR JUMP
                                688  *
                                689  *       PUSH    H               SET ON STACK AS 'REGISTER'
                                690  *       PUSH    D               SET RETURN ADDRESS
                                691  *       LXI     D,CTLFLG
                                692  *       LDAX    D               (A) = CTLFLG
                                693
    000.143  303 105 004        694          JMP     SAVALLX         GO TO SAVALL EXTENSION
                                695
    000.001                     696          IF      .RAM.
                                697          ELSE
                                698  **      ENTRY POINT FOR THE Z80 NMI
                                699  *
                                700
                                701
    000.000                     702          ERRNZ   *-66H           Z80 NMI ADDRESS
                                703          ENDIF
                                704
    000.146  303 116 004        705  NMIENT  JMP     NMI
                                706
    000.001                     707          IF      .RAM.
                                708          ELSE
    000.000                     709          ERRNZ   SAVALLR-151A    DO NOT CHANGE ORGANIZATION
                                710          ENDIF
                                711
    000.151                     712  SAVALLR EQU     *               SAVALL EXTENSION RETURN ADDRESS
                                713
    000.151  057                714          CMA
    000.152  346 060            715          ANI     CB.MTL+CB.SSI   SAVE REGISTER ADDR IF USER OR SINGLE-STEP
    000.154  310                716          RZ                      RETURN IF WAS INTERRUPT OF MONITOR LOOP
    000.155  041 002 000        717          LXI     H,2
    000.160  071                718          DAD     SP              (H,L) = ADDRESS OF 'STACKPTR' ON STACK
```

```
000.161  042 035 040   719           SHLD    REGPTR
000.164  311           720           RET


                       722  **       CUI - CHECK FOR USER INTERRUPT PROCESSING.
                       723  *
                       724  *        CUI IS CALLED TO SEE IF THE USER HAS SPECIFIED PROCESSING
                       725  *        FOR THE CLOCK INTERRUPT.
                       726
000.001                727            IF      .RAM.
                       728            ELSE
000.000                729            ERRNZ   *-165A
                       730            ENDIF
                       731
040.010                732  .         SET     .MFLAG          REFERENCE TO MFLAG
000.165  012           733  CUI1      LDAX    B               (A) = .MFLAG
000.000                734            ERRNZ   UO.CLK-1        CODE ASSUMED = 01
000.166  017           735            RRC
000.167  334 037 040   736            CC      UIVEC           IF SPECIFIED, TRANSFER TO USER
                       737
                       738  *        RETURN TO PROGRAM FROM INTERRUPT.
                       739
000.001                740            IF      .RAM.
                       741            ELSE
000.000                742            ERRNZ   *-172A
                       743            ENDIF
                       744
000.172  361           745  INTXIT    POP     PSW             REMOVE FAKE 'STACK REGISTER'
000.173  361           746            POP     PSW
000.174  301           747            POP     B
000.175  321           748            POP     D
000.176  341           749            POP     H
000.177  373           750            EI
000.200  311           751            RET
```

```
                                754  ***     CLOCK - PROCESS CLOCK INTERRUPT
                                755  *
                                756  *       CLOCK IS ENTERED WHENEVER A MILLISECOND CLOCK INTERRUPT IS
                                757  *       PROCESSED.
                                758  *
                                759  *       TICCNT IS INCREMENTED EVERY INTERRUPT.
                                760
000.001                         761          IF      .RAM.
                                762          ELSE
000.000                         763          ERRNZ   *-201A
                                764          ENDIF
                                765
000.201  052 033 040            766  CLOCK   LHLD    TICCNT
000.204  043                    767          INX     H
000.205  042 033 040            768          SHLD    TICCNT          INCREMENT TICCOUNT
                                769
000.210  072 011 040            770          LDA     CTLFLG          CLEAR CLOCK INTERRUPT FLIP-FLOP
000.213  323 360                771          OUT     OP.CTL
                                772
                                773  *       EXIT CLOCK INTERRUPT.
                                774
000.215  001 011 040            775          LXI     B,CTLFLG
000.220  012                    776          LDAX    B               (A) = CTLFLG
000.221  346 040                777          ANI     CB.MTL
000.223  302 172 000            778          JNZ     INTXIT          IF IN MONITOR MODE
000.226  013                    779          DCX     B
000.000                         780          ERRNZ   CTLFLG-.MFLAG-1
000.227  012                    781          LDAX    B               (A) = .MFLAG
000.000                         782          ERRNZ   UO.HLT-200Q     ASSUME HIGH-ORDER
000.230  027                    783          RAL
000.231  332 270 000            784          JC      CLK4            SKIP IT
                                785
                                786  *       NOT IN MONITOR MODE, CHECK FOR HALT
                                787
000.234  076 012                788          MVI     A,10            (A) = INDEX OF *P* REG
000.236  315 052 003            789          CALL    LRA.            LOCATE REGISTER ADDRESS
000.241  136                    790          MOV     E,M
000.242  043                    791          INX     H
000.243  126                    792          MOV     D,M             (D,E) = PC CONTENTS
000.244  033                    793          DCX     D
000.245  032                    794          LDAX    D
000.246  376 166                795          CPI     MI.HLT          CHECK FOR HALT
000.250  302 165 000            796          JNZ     CUI1
000.253  076 007                797          MVI     A,A.BEL         DING BELL
000.255  315 302 003            798          CALL    WCC
000.260  076 110                799          MVI     A,'H'           'H' FOR HALT
000.262  315 302 003            800          CALL    WCC
000.265  303 322 000            801          JMP     ERROR
                                802
                                803  ***     JE      ERROR           IF HALT, BE IN MONITOR MODE
                                804
                                805  *       NONE OF THE ABOVE, SO ALLOW USER PROCESSING OF CLOCK INTERRUPT
                                806
000.270                         807  CLK4    EQU     *
000.270  303 165 000            808          JMP     CUI1            ALLOW USER PROCESSING OF CLOCK
```

```
                          811  **      THIS IS ONLY A PORTION OF THE DYNAMIC RAM TEST!!
                          812  *
                          813  *       WAIT BEFORE MAKING ANOTHER LOOP
                          814
 000.273  041 000 000     815  DYMEM6  LXI     H,0
 000.276  053             816  DYMEM7  DCX     H
 000.277  174             817          MOV     A,H
 000.300  265             818          ORA     L
 000.301  302 276 000     819          JNZ     DYMEM7          IF (B,C) NOT ZERO
                          820
 000.304  303 207 007     821          JMP     DYMEM4          TRY AGAIN BY INCREMENTING ONCE MORE
                          822
                          823  **      HAVE A FAILURE PRIOR TO REACHING END OF MEMORY!
                          824  *
 000.307  353             825  DYMEM9  XCHG
 000.310  041 047 001     826          LXI     H,MSG.ERR       DISPLAY ERROR MESSAGE
                          827
                          828  *       LD      IX,DY9.3        RETURN ADDRESS
 000.313  335 041         829          DB      MI.LDXA,MI.LDXB
 000.315  315 003         830          DW      DY9.3
                          831
 000.317  303 306 007     832          JMP     DYMSG
```

```
                                835  ***     ERROR - COMMAND ERROR.
                                836  *
                                837  *       ERROR IS CALLED AS A 'BAIL-OUT' ROUTINE.
                                838  *
                                839  *       IT RESETS THE OPERATIONAL MODE, AND RESTORES THE STACKPOINTER.
                                840  *
                                841  *       ENTRY   NONE
                                842  *       EXIT    TO MTR LOOP
                                843  *               CTLFLG SET
                                844  *               .MFLAG CLEARED
                                845  *       USES    ALL
                                846
    000.001                     847          IF      .RAM.
                                848          ELSE
    000.000                     849          ERRNZ   *-322A
                                850          ENDIF
                                851
    000.322                     852  ERROR   EQU     *
    000.322  041 010 040        853          LXI     H,.MFLAG
    000.325  176                854          MOV     A,M             (A) = .MFLAG
    000.326  346 275            855          ANI     377Q-UO.DDU-UO.NFR      RE-ENABLE DISPLAYS
    000.330  167                856          MOV     M,A             REPLACE
    000.331  043                857          INX     H
    000.332  066 360            858          MVI     M,CB.SSI+CB.MTL+CB.CLI+CB.SPK   RESTORE *CTLFLG*
    000.000                     859          ERRNZ   CTLFLG-.MFLAG-1
    000.334  373                860          EI
    000.335  052 035 040        861          LHLD    REGPTR
    000.340  371                862          SPHL                    RESTORE STACK POINTER TO EMPTY STATE
    000.341  315 136 002        863          CALL    ALARM           ALARM FOR 200 MS


                                865  **      MTR - MONITOR LOOP.
                                866  *
                                867
    000.001                     868          IF      .RAM.
                                869          ELSE
    000.000                     870          ERRNZ   *-344A
                                871          ENDIF
                                872
    000.344                     873  MTR     EQU     *
    000.344  373                874          EI
                                875
    000.345                     876  MTR1    EQU     *
    000.345  041 345 000        877          LXI     H,MTR1
    000.350  345                878          PUSH    H               SET 'MTR1' AS RETURN ADDRESS
    000.351  303 113 002        879          JMP     CKAUTO          CHECK AUTO BOOT, IF NOT CONTROL BACK TO NEXT
    000.354  315 100 006        880  MTR.15  CALL    TYPMSG          PRINT 'H:'
                                881
    000.357  315 262 003        882  MTR.2   CALL    RCC             READ A CONSOLE CHARACTER
    000.362  346 137            883          ANI     01011111B       MAKE SURE ITS UPPER CASE TO MATCH TABLE
    000.364  041 025 001        884          LXI     H,MTRA          LOOK UP CHARACTER IN *MTRA*
    000.367  006 004            885          MVI     B,MTRAL         (B) = LENGTH OF TABLE
    000.371  276                886  MTR.3   CMP     M               SEE IF CHARACTER FROM CONSOLE = TABLE ENTRY
    000.372  312 014 001        887          JZ      MTR.4           IF EQUAL
```

```
                                888
                                889             INX     H               POINT TO NEXT TABLE ENTRY
                                890             INX     H
                                891             INX     H
                                892             DCR     B               SEE IF PAST END OF TABLE
                                893             JNZ     MTR.3           IF NOT PAST
                                894
                                895             MVI     A,A.BEL         ELSE, DING ERROR
                                896             CALL    WCC
                                897             JMP     MTR.2           TRY AGAIN
                                898
                                899     MTR.4   CALL    WCC             WRITE CHARACTER BACK TO CONSOLE
                                900             INX     H               GET ROUTINE ADDRESS LSB
                                901             MOV     A,M
                                902             INX     H               GET MSB
                                903             MOV     H,M
                                904             MOV     L,A             (H,L) = ROUTINE ADDRESS
                                905             PCHL                    GO TO ROUTINE
                                906
                                907
                                908
                                909     MTRA    EQU     *               JUMP TABLE
                                910             SET     *#256           ALL ROUTINES MUST START IN THIS PAGE
                                911             DB      'G'             GO TO USER ROUTINE
                                912             DW      GO88
                                913
                                914             DB      'S'             SUBSTITUTE MEMORY MODE
                                915             DW      SUBM
                                916
                                917             DB      'P'             PROGRAM COUNTER ALTER MODE
                                918             DW      PCA
                                919
                                920             DB      'B'             BOOT H-17 OR Z-47 DRIVE
                                921             DW      BOOT
                                922
                                923     MTRAL   EQU     *-MTRA/3        NUMBER OF TABLE ENTRYS          /JWT 790507/
                                925     **      BSMSG   = BOOT SECONDARY DEVICE MESSAGE
                                926
                                927     BSMSG   DB      ' SD',0         'SECONDARY DEVICE'
                                929     ERRMSG  DB      '?',0           ERROR MESSAGE
```

```
001,047                 931             ORG     1047A
                        932  **         MSG.ERR - ERROR MESSAGE FOR RAM TEST
                        933  *
                        934  *          'ERROR @ '
                        935
001,047  015 012 012    936  MSG.ERR    DB      A.CR,A.LF,A.LF
001,052  105 122 122    937             DB      'ERROR @ '
001,062  000            938             DB      0
```

```
                                941  **      SAE - STORE ABUSS AND EXIT.
                                942  *
                                943  *       ENTRY   (HL) = ABUSS VALUE
                                944  *       EXIT    TO (RET)
                                945  *       USES    NONE
                                946
  000.001                       947           IF      ,RAM,
                                948           ELSE
  000.000                       949           ERRNZ   *-1063A
                                950           ENDIF
                                951
  001.063  042 024 040          952  SAE      SHLD    ABUSS
  001.066  311                  953           RET



                                955  **      PIN     -       PORT IN
                                956  *
                                957  *       PIN  INPUTS A BYTE FROM DISK
                                958  *
                                959  *       ENTRY:  NONE
                                960  *
                                961  *       EXIT:   (A) = INPUT BYTE FROM Z47
                                962  *
                                963  *       USE:    AF
                                964
  001.067                       965  PIN      EQU     *
  001.067  315 170 006          966           CALL    IN.                     GET STATUS
  001.072  346 200              967           ANI     S.DTR                   CHECK FOR DATA TERMINAL REQUEST
  001.074  050 371              968           JR      Z,PIN                   IF READY, WAIT
  001.076  315 156 006          969           CALL    IN1.                    INPUT A BYTE FROM PORT
  001.101  311                  970           RET
```

```
001.103                     973             ORG     1103A
                            974  **         PCA - PROGRAM COUNTER ALTER
                            975  *
                            976  *          PCA INPUTS AND/OR DISPLAYS THE CURRENT USER PROGRAM VALUE AND ALLOWS
                            977  *          A NEW VALUE TO BE ENTERED OR RETAINS THE CURRENT VALUE IF
                            978  *          A CR IS TYPED
                            979  *
                            980  *          ENTRY   NONE
                            981  *          EXIT    NONE
                            982  *          USES    A,D,E,H,L,F
                            983
                            984
001.103  041 214 006       985  PCA        LXI     H,MSG.PC        COMPLETE PC MESSAGE
001.106  315 100 006       986             CALL    TYPMSG
001.111  076 012           987             MVI     A,10            GET LOCATION OF USER PC
001.113  315 052 003       988             CALL    LRA.
001.116  136               989             MOV     E,M             (D,E) = USER PC VALUE
001.117  043               990             INX     H
001.120  126               991             MOV     D,M
001.121  353               992             XCHG                    (H,L) = USER PC VALUE
                            993
001.122  315 150 005       994             CALL    IROC            INPUT NEXT CHARACTER
001.125  332 137 001       995             JC      PCA1            IF FIRST CHARACTER WAS OCTAL, INPUT NEW PC
                            996
001.130  315 313 005       997             CALL    TOA             ELSE, OUTPUT CURRENT VALUE
001.133  315 150 005       998             CALL    IROC            SEE IF USER WANTS TO CHANGE IT NOW
001.136  320               999             RNC                     IF NO CHANGE, EXIT
                           1000
                           1001  *          ENTER NEW USER PC VALUE
                           1002
001.137  353               1003 PCA1       XCHG                    (H,L) = ADDRESS OF USER PC VALUE
001.140  026 015           1004            MVI     D,A.CR          END BYTE WITH A RETURN
001.142  315 062 003       1005            CALL    IOA             INPUT NEW ADDRESS
001.145  311               1006            RET                     EXIT


                           1008  **         GO88 - GO TO USER ROUTINE FROM H88 MONITOR
                           1009  *
                           1010  *          GO88 WAITS FOR A CARRIAGE RETURN OR A NEW ADDRESS TERMINATED WITH
                           1011  *          A CARRIAGE RETURN.  IF NO ADDRESS IS ENTERED, GO88 TRANSFERS
                           1012  *          CONTROL TO THE ADDRESS SPECIFIED BY THE USER PC VALUE
                           1013
                           1014
001.146  041 165 006       1015 GO88       LXI     H,MSG.GO        COMPLETE GO MESSAGE
001.151  315 100 006       1016            CALL    TYPMSG
001.154  315 150 005       1017            CALL    IROC            INPUT A RETURN OR AN OCTAL CHARACTER
001.157  322 177 001       1018            JNC     GO88.1          IF RETURN, GO TO CURRENT USER PC
                           1019
001.162  365               1020            PUSH    PSW             ELSE SAVE OCTAL CHARACTER AND FLAGS
001.163  076 012           1021            MVI     A,10            GET ADDRESS OF USER PC
001.165  315 052 003       1022            CALL    LRA.
001.170  043               1023            INX     H               POINT TO MSB
001.171  361               1024            POP     PSW             GET FIRST CHARACTER BACK
001.172  026 015           1025            MVI     D,A.CR          END ADDRESS WITH A RETURN
```

```
001.174  315 062 003  1026           CALL    IOA             INPUT NEW GO ADDRESS
001.177  315 302 003  1027  GO88.1   CALL    WCC             ECHO RETURN
001.202  076 012      1028           MVI     A,A.LF          LINE FEED
001.204  315 302 003  1029           CALL    WCC
001.207  303 222 001  1030           JMP     GO              EXECUTE USER ROUTINE



                      1032  **       AUTOBO  - AUTO BOOT
                      1033  *
                      1034  *        ENTRY:  NONE
                      1035  *
                      1036  *        EXIT:   (SEE 'DEVICE' ROUTINE)
                      1037  *
                      1038  *        USE:    ALL
                      1039
001.212  257          1040  AUTOBO   XRA     A               SET TO PRIMARY FLAG
001.213  315 301 002  1041           CALL    DEVICE          CHECK DEVICE INFROMATION
001.216  303 336 001  1042           JMP     BOOTO           GOTO BOOT IT



001.222                1044           ORG     1222A
                       1045  **       GO - RETURN TO USER MODE
                       1046  *
                       1047  *        ENTRY   NONE
                       1048
000.001                1049           IF      .RAM.
                       1050           ELSE
000.000                1051           ERRNZ   *-1222A
                       1052           ENDIF
                       1053
001.222  303 063 000  1054  GO       JMP     GO.



                       1056  **       SSTEP - SINGLE STEP INSTRUCTION.
                       1057  *
                       1058  *        ENTRY   NONE
                       1059
000.001                1060           IF      .RAM.
                       1061           ELSE
000.000                1062           ERRNZ   *-1225A
                       1063           ENDIF
                       1064
001.225                1065  SSTEP    EQU     *               SINGLE STEP
001.225  363           1066           DI                      DISABLE INTERRUPTS UNTIL THE RIGHT TIME
001.226  072 011 040  1067           LDA     CTLFLG
001.231  356 020      1068           XRI     CB.SSI          CLEAR SINGLE STEP INHIBIT
001.233  323 360      1069           OUT     OP.CTL          PRIME SINGLE STEP INTERRUPT
001.235  062 011 040  1070  SST1     STA     CTLFLG          SET NEW FLAG VALUES
001.240  341          1071           POP     H               CLEAN STACK
001.241  303 172 000  1072           JMP     INTXIT          RETURN TO USER ROUTINE FOR STEP
```

```
                                1074  **      STPRTN - SINGLE STEP RETURN
                                1075
    000.001                     1076          IF      .RAM.
                                1077          ELSE
    000.000                     1078          ERRNZ   *-1244A
                                1079          ENDIF
                                1080
    001.244                     1081  STPRTN  EQU     *
    001.244  366 020            1082          ORI     CB.SSI          DISABLE SINGLE STEP INTERRUPTION
    001.246  323 360            1083          OUT     OP.CTL          TURN OFF SINGLE STEP ENABLE
    040.011                     1084          SET     CTLFLG
    001.250  022                1085          STAX    D
    001.251  346 040            1086          ANI     CB.MTL          SEE IF IN MONITOR MODE
    001.253  302 344 000        1087          JNZ     MTR
    001.256  303 042 040        1088          JMP     UIVEC+3         TRANSFER TO USER'S ROUTINE
```

```
                        1091  **      NBOOT   -       NORMAL BOOT
                        1092  *
                        1093  *       NBOOT IS ENTERED WHEN USER TYPE 'BOOT' COMMAND FROM MONITOR,
                        1094  *       IT WILL ACCEPT THE BOOT DEVICE AS WELL AS THE UNIT NUMBER FROM
                        1095  *       CONSOLE AND GO TO THE BOOT CODE.
                        1096  *
                        1097  *       ENTRY:  NONE
                        1098  *
                        1099  *       EXIT:   (AIO.UNI) = UNIT NUMBER TO BOOT
                        1100  *               (PRIM)  = PORT ADDRESS OF THE BOOT DEVICE
                        1101  *               (TMFG)  = DEVICE TYPE, =1 IS Z47; =0 IS H17
                        1102  *
                        1103  *       USED:   ALL
                        1104
001.261  257            1105  NBOOT   XRA     A                       SET Z FLAG TO PRIMARY DEVICE
001.262  315 301 002    1106  NBOOT0  CALL    DEVICE                  READ SWITCH TO DETERMINE BOOT DEVICE
001.265  315 262 003    1107  START1  CALL    RCC                     INPUT FROM KB
001.270  376 015        1108          CPI     A,CR                    IF INPUT IS CR
001.272  050 042        1109          JR      Z,BOOT0                   THEN TAKE IT AS DRIVE 0
001.274  376 060        1110          CPI     '0'                     CHECK INPUT IS WITHIN DRIVE 0 - (B)
001.276  070 007        1111          JR      C,WRONG                   IF LESS THEN 0, WRONG INPUT
001.300  270            1112          CMP     B
001.301  070 036        1113          JR      C,BOOT5                   IF WITHIN THE RANGE, BOOT IT!
001.303  010            1114          DB      MI.EXAF                 SAVE INPUT, CHECK PRIM OR SEC?
001.304  050 010        1115          JR      Z,NB7                   IF PRIMARY, CHECK 'S'
001.306  010            1116          DB      MI.EXAF                 RESTORE (Z) FLAG
001.307                  1117  WRONG   EQU     *
001.307  076 007        1118          MVI     A,A.BEL                 NOT THE CASES, BEEP!
001.311  315 302 003    1119          CALL    WCC
001.314  030 347        1120          JR      START1                  AND TRY AGAIN
                        1121
001.316  010            1122  NB7     DB      MI.EXAF                 RESTORE INPUT & PRIM, SEC FLAG
001.317  346 137        1123          ANI     01011111B               MASK TO UPPER CASE LETTER
001.321  376 123        1124          CPI     'S'                     CHECK THE USER LIKE TO BOOT FROM
001.323  040 362        1125          JR      NZ,WRONG                BOOT SECONDARY DEVICE
                        1126
                        1127  *       USER WISHES TO BOOT FROM SECONDARY DEVICE
                        1128
001.325                 1129  BSEC    EQU     *
001.325  041 041 001    1130          LXI     H,BSMSG                 PRINT BOOT SECONDARY MESSAGE
001.330  315 100 006    1131          CALL    TYPMSG
001.333  074            1132          INR     A                       SET (Z)=0 FOR SECONDARY DEVICE
001.334  030 324        1133          JR      NBOOT0
                        1134
                        1135  *  SAVE THE AIO.UNI, CHECK IF THERE IS THE BOOT DEVICE AND GO!
                        1136
001.336  257            1137  BOOT0   XRA     A                       TAKE CR OR AUTO BOOT AS DRIVE 0
001.337  030 012        1138          JR      BOOT6
                        1139
001.341  315 302 003    1140  BOOT5   CALL    WCC                     PRINT UNIT NUMBER
001.344  326 060        1141          SUI     '0'                     MAKE IT BINARY
001.346  107            1142          MOV     B,A                     SAVE THE UNIT #
001.347  315 003 006    1143          CALL    WCR                     WAIT FOR A CR
001.352  170            1144          MOV     A,B                     GET UNIT NUMBER BACK
001.353  062 061 041    1145  BOOT6   STA     AIO.UNI                 STORE THE UNIT #
001.356  174            1146          MOV     A,H                     CHECK IF NO DEVICE AT ADDR. PORT
```

```
001.357  247          1147        ANA     A
001.360  312 171 002  1148        JZ      NODEV                NO DEVICE
001.363  351          1149        PCHL                         JMP TO THE EXECUTION ROUTINE
```

```
                        1152  **      Z47     -       BOOT FROM Z47 DISK DRIVE
                        1153  *
                        1154  *       Z47 WILL LOAD DATA FROM DISK TRACK 0 SECTOR 1 AND 2 TO
                        1155  *       USER FIRST AVALIABLE RAM LOCATION. IF THE BOOT IS SUCCED,
                        1156  *       CONTROL PASS TO THAT LOCATION.
                        1157  *
                        1158  *       ENTRY:  (AIO,UNI) = UNIT NUMBER TO BOOT
                        1159  *
                        1160  *       EXIT:   NONE
                        1161  *
                        1162  *       USE:    ALL
                        1163
001,364                 1164  Z47     EQU     *
                        1165  *       LD      (STK),SP                SAVE STACK POINTER FOR RE-BOOT
001,364 355 163         1166          DB      355Q,163Q
001.366 124 041         1167          DW      STK
                        1168
001.370                 1169  Z47A    EQU     *
001,370 373             1170          EI                              LET THE TIMER FLY
001.371 072 061 041     1171          LDA     AIO.UNI                 GET UNIT NUMBER
001,374 007             1172          RLC                             SET TO SIDE/UNIT/SECTOR FORMAT
001.375 007             1173          RLC
001,376 007             1174          RLC
001.377 007             1175          RLC
002,000 007             1176          RLC
002.001 074             1177          INR     A                       SET TO SECTOR 1
002,002 117             1178          MOV     C,A                     SAVE SIDE/UNIT/SECTOR (SIDE=0)
002.003 076 002         1179  RESET   MVI     A,W.RES                 RESET Z47
002,005 315 063 006     1180          CALL    OUT.
                        1181
                        1182  * DETERMINE THE DISK IS SINGLE OR DOUBLE DENSITY
                        1183
002,010 076 002         1184          MVI     A,DC,RAS                SEND READ AUX. STATUS COMMAND
002.012 315 027 006     1185          CALL    COM
002,015 171             1186          MOV     A,C                     GET SIDE/UNIT/SECTOR
002.016 315 023 006     1187          CALL    DAT                     SEND SECOND COMMAND BYTE
002,021 315 067 001     1188          CALL    PIN                     GET AUX. STATUS
002.024 346 100         1189          ANI     AS.ODD                  CHECK IT IS SINGLE OR DOUBLE DENSITY
002,026 007             1190          RLC
002.027 356 200         1191          XRI     10000000B               REVERSE THEN 7TH BIT, MAKE THE SECTOR
002,031 107             1192          MOV     B,A                     # TO 128 OR 256(B=0) BYTES
                        1193
                        1194  *       READ BOOT CODE FROM Z47
                        1195
002,032 041 200 042     1196          LXI     H,USERFWA               BOOT DESTINATION
002.035 305             1197          PUSH    B                       SAVE SECTOR SIZE & SIDE/UNIT/SECTOR
002,036 315 121 006     1198          CALL    RDBLCK                  READ A SECTOR FROM DISK
002.041 301             1199          POP     B                       GET SECTOR SIZE & SUS BACK
002,042 014             1200          INR     C                       SET TO NEXT SECTOR
002.043 315 121 006     1201          CALL    RDBLCK                  READ ANOTHER SECTOR
                        1202
                        1203  *       CHECK ANY ERROR DURING BOOT
                        1204
002.046 315 170 006     1205          CALL    IN.                     GET INTERFACE STATUS
002,051 346 001         1206          ANI     S.ERR                   IS THERE ANY ERROR WHEN BOOT
002.053 040 114         1207          JR      NZ,NODEV                      THEN ABORT
```

```
002.055  062 010 040  1208           STA     .MFLAG                  STOP TIMER
002.060  303 200 042  1209           JMP     USERFWA

                      1211  **       RETRY   -       RE-BOOT Z47
                      1212  *
                      1213  *       RETRY IS ENTERED WHEN 3.5 SECONDS TIME OUT & BOOT Z47
                      1214  *       STILL NOT SUCCEED. IT RESTORE STACK & JUMP TO BOOT Z47 ROUTINE
                      1215  *
                      1216  *       ENTRY:  NONE
                      1217  *
                      1218  *       EXIT:   (HL) = (SP)
                      1219  *
                      1220  *       USE:    HL, SP
                      1221
002.063  052 124 041  1222  RETRY    LHLD    STK                     GET OLD STACK ADDRESS
002.066  371         1223            SPHL                            SET TO STACK POINTER
002.067  030 277      1224           JR      Z47A                    RE-BOOT
```

```
                                1227  **      R.SDP   - SET DEVICE PARAMETER, ALLOW TO SET DRIVE 0, 1, AND 2,
                                1228  *       (MORE INFORMATION CAN BE FOUND IN H17 ROM CODE 36062A)
                                1229
     002.071                    1230  R.SDP   EQU     *
     002.071  076 012           1231          MVI     A,ERPTCNT
     002.073  062 264 040       1232          STA     D.OECNT                 SET MAX ERROR COUNT FOR OPERATION
     002.076  072 061 041       1233          LDA     AIO.UNI                 LOAD DRIVE NUMBER
     002.101  365               1234          PUSH    PSW                     SAVE IT
     002.102  376 002           1235          CPI     2                       IS IT DRIVE 2?
     002.104  070 002           1236          JR      C,R.SDP1                IF NOT JMP TO H17 ROM ROUTINE
     002.106  076 003           1237          MVI     A,3
     002.110  303 073 036       1238  R.SDP1  JMP     SDP3



                                1240  **      CKAUTO  -       CHECK IF IT IS AUTO BOOT
                                1241  *
                                1242  *       CKAUTO IS ENTERED FROM MONITOR LOOP, IT WILL CHECK IF AUTO BOOT
                                1243  *       CONDITION IS TRUE.IF NOT, BACK TO MONITOR LOOP
                                1244  *               IF AUTO BOOT, JUMP TO BOOT DEVICE ROUTINE
                                1245  *
                                1246  *       ENTRY:  NONE
                                1247  *
                                1248  *       EXIT:   NONE
                                1249  *
                                1250  *       USE:    ALL
                                1251
     002.113                    1252  CKAUTO  EQU     *
     002.113  333 362           1253          IN      H88.SW          GET SWITCH DATA
     002.115  346 200           1254          ANI     H88S.AT         CHECK AUTO BOOT SWITCH BIT SET
     002.117  050 007           1255          JR.     Z,CHAT2         NOT SET
     002.121  041 123 041       1256          LXI     H,AUTOB         SET AUTO BOOT FLAG ADDR,
     002.124  276               1257          CMP     M               CHECK AUTO BOOT BEFORE?
     002.125  302 243 004       1258          JNZ     ATB             YES, AUTO BOOT
     002.130  041 111 006       1259  CHAT2   LXI     H,MSG.PR        LOAD 'H:' ADDR.
     002.133  303 354 000       1260          JMP     MTR.15          BACK TO MONITOR LOOP
```

```
002.136                 1263            ORG     2136A
                        1264  **        HORN - MAKE NOISE.
                        1265  *
                        1266  *         ENTRY   (A) = (MILLISECOND COUNT)/2
                        1267  *         EXIT    NONE
                        1268  *         USES    A,F
                        1269
000.001                 1270            IF      .RAM.
                        1271            ELSE
000.000                 1272            ERRNZ   *-2136A
                        1273            ENDIF
                        1274
002.136                 1275  ALARM     EQU     *
002.136  030 026        1276            JR      ALARMB          BRANCH TO A JUMP TO NOISE TO DING BELL
                        1277
000.001                 1278            IF      .RAM.
                        1279            ELSE
000.000                 1280            ERRNZ   *-2140A
                        1281            ENDIF
                        1282
002.140  365            1283  HORN      PUSH    PSW
002.141  076 200        1284            MVI     A,CB.SPK        TURN ON SPEAKER
                        1285
002.143  343            1286  HRN0      XTHL                    SAVE (HL), (H)  = COUNT
002.144  325            1287            PUSH    D               SAVE (DE)
002.145  353            1288            XCHG                    (D) = LOOP COUNT
002.146  041 011 040     1289            LXI     H,CTLFLG
002.151  256            1290            XRA     M
002.152  136            1291            MOV     E,M             (E) = OLD CTLFLG VALUE
002.153  167            1292            MOV     M,A             TURN ON HORN
002.154  056 033        1293            MVI     L,#TICCNT
                        1294
002.156  172            1295            MOV     A,D             (A) = CYCLE COUNT
002.157  206            1296            ADD     M
002.160  276            1297  HRN2      CMP     M               WAIT REQUIRED TICCOUNTS
002.161  040 375        1298            JR      NZ,HRN2
                        1299
002.163  303 045 006    1300            JMP     HRNX            JUMP TO AN EXTENSION OF HORN SO ROOM
                        1301  *                                       CAN BE MADE FOR A JUMP TO NIOSE
                        1302
                        1303
002.166  303 053 006    1304  ALARMB    JMP     NOISE           SEND A BELL TO THE CONSOLE
```

```
                        1307  **      NODEV   - NO DEVICE AT THE UNIT USER INDICATE
                        1308  *
                        1309  *       NODEV IS ENTERED WHEN:  1. 15 SECONDS TIME OUT
                        1310  *                                OR 2. NO DEVICE IS INDICATED ON SWITCH
                        1311  *                                OR 3. USER HIT <DELETE> TO ABORT BOOT
                        1312  *                                OR 4. BOOT ERROR
                        1313  *       IT WILL EXIT TO 'ERROR' ROUTINE AND MONITOR LOOP
                        1314  *
                        1315  *       ENTRY:  NONE
                        1316  *
                        1317  *       EXIT:   (A) = 0
                        1318  *
                        1319  *       USE:    AF, HL
                        1320
002,171                 1321  NODEV   EQU     *
002,171  041 045 001    1322          LXI     H,ERRMSG        PRINT ERROR MESSAGE
002,174  315 100 006    1323          CALL    TYPMSG
002,177  062 010 040    1324          STA     .MFLAG          STOP TIMER
002,202  323 177        1325          OUT     DP.DC           OFF DISK
002,204  303 322 000    1326          JMP     ERROR           BACK TO MONITOR LOOP
```

```
                        1329  **      H17     -   BOOT FROM H17 DISK SYSTEM
                        1330  *                   (THIS IS THE MODIFICATION OF THE H17 BOOT ROUTINE,
                        1331  *                   MORE INFORMATION CAN BE FOUND ON H17 BOOT ROM 30000A)
                        1332  *
                        1333  *       ENTRY:  (AIO.UNI) = THE UNIT TO BOOT
                        1334  *
                        1335  *       EXIT:   NONE
                        1336  *
                        1337  *       USE:    ALL
                        1338
002.207                 1339  H17     EQU     *
002.207  001 130 000    1340          LXI     B,BOOTAL        SET THE COUNT TO MOVE IN CONSTANTS AND VECTORS
002.212  021 132 037    1341          LXI     D,BOOTA         SET THE SOURCE ADDRESS
002.215  041 110 040    1342          LXI     H,D.CON         SET THE DESTINATION ADDRESS
002.220  315 252 030    1343          CALL    *MOVE           MOVE IT
                        1344
                        1345  **      SET ADDRESS FOR 'SET DEVICE PARAMETER' ROUTINE
                        1346  *       TO HANDLE DISK DRIVE 0, 1, AND 2.
                        1347
002.223  041 071 002    1348          LXI     H,R.SDP         SET THIS ROM ROUTINE ADDRESS
002.226  042 206 040    1349          SHLD    D,SDP           SET INTO RAM JUMP VECTOR
002.231  373            1350          EI                      RESTORE INTERRUPT
                        1351
                        1352  *       WAIT TILL USER INSERT THE DISK AND CLOSE THE DOOR
                        1353  *       (TIMER INTERRUPT IS AFFECTED NOW)
                        1354
002.232  006 012        1355          MVI     B,10            LOOK FOR SOME HOLE AND NO HOLE
002.234  315 071 002    1356          CALL    R,SDP           SELECT UNIT & MOTOR ON
002.237  315 271 036    1357  H17A    CALL    WNH             WAIT FOR NO HOLE
002.242  315 235 036    1358          CALL    WHD             WAIT FOR HOLE
002.245  020 370        1359          DJNZ    H17A
                        1360
                        1361  *       READ BOOT CODE
                        1362
002.247  315 366 033    1363          CALL    R,ABORT         RESET DISK DRIVE
002.252  021 200 042    1364          LXI     D,USERFWA       SET THE LOAD LOCATION
002.255  001 000 011    1365          LXI     B,9*256         LOAD 9 SECTORS
002.260  041 000 000    1366          LXI     H,0             LOAD FROM TRACK 0 SECTOR 1
002.263  315 077 034    1367          CALL    R,READ          READ DISK BOOT CODE
002.266  070 301        1368          JR      C,NODEV           ERROR ON BOOT, BACK TO "H:"
                        1369
                        1370  **      SETUP CLOCK INTERRUPT  FOR H17 ONLY
                        1371
002.270  041 031 034    1372          LXI     H,CLOCK17       LOAD CLOCK ROUTINE ADDRESS
002.273  042 040 040    1373          SHLD    .UIVEC+1        SET IT INTO VECTOR LOCATION
002.276  303 200 042    1374          JMP     USERFWA         GOTO BOOT CODE
```

```
                        1377  ***     DEVICE  -       DETERMINE BOOT WHICH DEVICE AT WHICH PORT
                        1378  *
                        1379  *       ENTRY:  Z FLAG  ( Z=1 FOR PRIMARY, Z=0 FOR SECONDARY)
                        1380  *
                        1381  *       EXIT:   HL = DEVICE BOOT EXECUTION ADDRESS
                        1382  *                    IF H=0 THEN NO DEVICE THERE
                        1383  *                    (I.E. THE EXEC. ADDR. MUST RESIDENT > 1000A)
                        1384  *               REG B = PRIMARY MAXI. DRIVE NUMBER
                        1385  *                    IF Z47, #='4'; IF H17, #='3'
                        1386  *               (PRIM) = PRIMARY DEVICE PORT ADDRESS
                        1387  *                    IF Z47 THEN THE PORT IS EITHER 170Q OR 174Q
                        1388  *                    IF H17 THEN DON'T CARE (H17 BOOT ROM TAKE CARE IT)
                        1389  *               (TMFG) = 1 IF BOOT FROM Z47, = 0 IF FROM H17
                        1390  *
                        1391  *       USE:    ALL
                        1392
    002.301             1393  DEVICE  EQU     *
    002.301  010        1394          DB      MI.EXAF         SAVE Z FLAG
                        1395
                        1396  *       INITIAL VARIABLES
                        1397
    002.302  363        1398          DI                      NO INTERRUPT
    002.303  041 240 040 1399          LXI     H,D.RAM         CLEAR H17 WORK RAM AREA
    002.306  006 037    1400          MVI     B,D.RAML        LENGTH TO CLEAR
    002.310  315 212 031 1401          CALL    $ZERO
    002.313  323 177    1402          OUT     DP.DC           OFF DISK
    002.315  062 033 040 1403          STA     TICCNT          0 TIMER COUNTER
    002.320  062 122 041 1404          STA     MYCNT           0.5 SECOND TIMER = 0
                        1405
    002.323  074        1406          INR     A               (A)=1
    002.324  062 121 041 1407          STA     TMFG            SET TIMER TO Z47 FLAG
    000.000             1408          ERRNZ   UO.CLK-1        TIMER INTERRPT MUST = 1
    002.327  062 010 040 1409          STA     .MFLAG          ALLOW TIMER INTERRUPT
    002.332  041 037 040 1410          LXI     H,.UIVEC        SET ALL VECTOR TO EI/RET PROCESS
    002.335  066 303    1411  BOOT2   MVI     M,MI.JMP
    002.337  043        1412          INX     H
    002.340  066 027    1413          MVI     M,$EIXIT        STORE LS BYTE
    002.342  043        1414          INX     H
    002.343  066 034    1415          MVI     M,EIXIT/256     STORE MS BYTE
    002.345  043        1416          INX     H
    002.346  207        1417          ADD     A
    002.347  362 335 002 1418          JP      BOOT2
                        1419
    002.352  041 302 004 1420          LXI     H,TMOUT         SET TIMER INTERRUPT VECTOR
    002.355  042 040 040 1421          SHLD    .UIVEC+1
                        1422
    002.360  076 170    1423          MVI     A,D.STA         ASSUME ALL DEVICE ARE Z47 & BOOT AT 170Q
    002.362  062 120 041 1424          STA     PRIM            SINCE H17 BOOT ROM WILL TAKE CARE OF ITS MATTER
    002.365  041 364 001 1425          LXI     H,Z47           SET Z47 BOOT ADDR.
    002.370  006 064    1426          MVI     B,'4'           SET MAX. UNIT TO 4
                        1427
                        1428  *       DETERMINE BOOT DEVICE AND ITS INFORMATION
                        1429
    002.372  333 362    1430          IN      H88.SW          READ SWITCH DATA
    002.374  365        1431          PUSH    PSW             SAVE IN STACK
    002.375  346 020    1432          ANI     H88S.DV         CHECK PRIMARY DEVICE ADDRESS
```

```
003,377  010          1433           DB      MI,EXAF         SAVE Z FLAG & GET Z' FOR PRIM. SEC. FLAG
003,000  040 005      1434           JR      NZ,SECOND       IT SECONDARY
003,002  010          1435           DB      MI,EXAF
003,003  040 033      1436           JR      NZ,B170
003,005  030 003      1437           JR      B174
003,007  010          1438  SECOND   DB      MI,EXAF
003,010  050 026      1439           JR      Z,B170          BOOT PRIMARY AT 170Q
003,012  076 174      1440  B174     MVI     A,UP.DP         PRIMARY DEVICE IS AT 174Q
003,014  062 120 041  1441           STA     PRIM
003,017  361          1442           POP     PSW             GET SWITCH DATA BACK
003,020  346 003      1443           ANI     H88S,4          CHECK THIS IS Z47 OR H17
003,022  050 004      1444           JR      Z,BH17          IT H17
003,024  075          1445           DCR     A
003,025  310          1446  DEV2     RZ                      IT IS Z47
003,026  045          1447           DCR     H               NO DEVICE THERE, Z47 LOCATION MUST ON 1***A
003,027  311          1448           RET
000,000               1449           ERRNZ   Z47/256-1
                      1450
                      1451  *        PRIMARY DEVICE IS H17
                      1452
003,030  041 207 002  1453  BH17     LXI     H,H17           SET TO H17 EXECUTION LOACTION
003,033  005          1454           DCR     B               SET TO MAX 3 DRIVE
003,034  062 121 041  1455           STA     TMFG            SET TIMER INTERRUPT = 0 FOR H17
003,037  311          1456           RET
                      1457
                      1458  **       PRIMARY DEVICE IS AT PORT 170Q
                      1459
003,040               1460  B170     EQU     *
003,040  361          1461           POP     PSW             GET SWITCH DATA
003,041  346 014      1462           ANI     H88S.0          CHECK ANY DEVICE IN 170Q
003,043  376 004      1463           CPI     00000100B       CHECK IF IT IS Z47
003,045  030 356      1464           JR      DEV2
```

```
003.047                  1467          ORG     3047A
                         1468  **      LRA - LOCATE REGISTER ADDRESS.
                         1469  *
                         1470  *       ENTRY   NONE.
                         1471  *       EXIT    (A) = REGISTER INDEX
                         1472  *               (H,L) = STORAGE ADDRESS
                         1473  *               (D,E) = (0,A)
                         1474  *       USES    A,D,E,H,L,F
                         1475
                         1476
000.001                  1477          IF      .RAM.
                         1478          ELSE
000.000                  1479          ERRNZ   *-3047A
                         1480          ENDIF
                         1481
003.047  072 005 040    1482  LRA     LDA     REGI
003.052  137            1483  LRA.    MOV     E,A
003.053  026 000        1484          MVI     D,0
003.055  052 035 040    1485          LHLD    REGPTR
003.060  031            1486          DAD     D               (DE) = (REGPTR)+(REGI)
003.061  311            1487          RET


                         1489  **      IOA - INPUT OCTAL ADDRESS.
                         1490  *
                         1491  *       ENTRY   (H,L) = ADDRESS OF RECEPTION DOUBLE BYTE.
                         1492  *               (D) = TERMINATING CHARACTER
                         1493  *       EXIT    NONE
                         1494  *       USES    A,D,E,H,L,F
                         1495
                         1496
000.001                  1497          IF      .RAM.
                         1498          ELSE
000.000                  1499          ERRNZ   *-3062A
                         1500          ENDIF
                         1501
003.062  303 176 005    1502  IOA     JMP     IOA1
003.065  000            1503          NOP                     RETAIN H8 ORG


                         1505  **      IOB - INPUT OCTAL BYTE.
                         1506  *
                         1507  *       READ ONE OCTAL BYTE FROM THE KEYSET.
                         1508  *
                         1509  *       ENTRY   (H,L) = ADDRESS OF BYTE TO HOLD VALUE
                         1510  *               'C' SET IF FIRST DIGIT IN (A)
                         1511  *       EXIT    NONE
                         1512  *       USES    A,D,E,H,L,F
                         1513
                         1514
000.001                  1515          IF      .RAM.
                         1516          ELSE
```

```
000.000                 1517            ERRNZ   *-3066A
                        1518            ENDIF
                        1519
003.066  066 000        1520  IOB       MVI     M,0             ZERO OUT OLD VALUE
003.070  324 262 003    1521  IOB1      CNC     RCC             READ CONSOLE CHARACTER
                        1522
                        1523  *         SEE IF CHARACTER IS A VALID OCTAL VALUE
                        1524  *
003.073  376 060        1525            CPI     '0'             LESS THAN ZERO?
003.075  332 135 003    1526            JC      IOB2            IF (A) < 0, SEE IF A TERMINATING CHARACTER
003.100  376 070        1527            CPI     '8'             GREATER THAN 7?
003.102  322 070 003    1528            JNC     IOB1            IF TOO LARGE, TRY AGAIN
                        1529
                        1530  *         HAVE AN OCTAL DIGIT
                        1531  *
003.105  315 302 003    1532            CALL    WCC             ECHO CHARACTER
003.110  346 007        1533            ANI     00000111B       MASK FOR BINARY VALUE
003.112  137            1534            MOV     E,A             (E) = VALUE
003.113  176            1535            MOV     A,M             GET OLD VALUE
003.114  007            1536            RLC                     SHIFT 3
003.115  007            1537            RLC
003.116  007            1538            RLC
003.117  303 126 003    1539            JMP     IOB1.5          JUMP AROUND AN H88/H89 TO H8 FAKE ROUTINE
                        1540
                        1541  **        FAKE OUT ROUTINE FOR CALLERS OF *DOD* FROM THE H8 FRONT PANEL
                        1542
                        1543
000.001                 1544            IF      .RAM.
                        1545            ELSE
000.000                 1546            ERRNZ   *-3122A
                        1547            ENDIF
                        1548
003.122  043            1549  DOD       INX     H
003.123  043            1550            INX     H
003.124  043            1551            INX     H
003.125  311            1552            RET
                        1553
                        1554
                        1555  *         CONTINUE
                        1556
003.126  346 370        1557  IOB1.5    ANI     11111000B       TOSS OLD LSB DIGIT
003.130  263            1558            ORA     E               REPLACE WITH NEW VALUE
003.131  167            1559            MOV     M,A
003.132  303 070 003    1560            JMP     IOB1            INPUT ANOTHER CHARACTER
                        1561
                        1562  *         CHECK FOR A CARRIAGE RETURN TO TERMINATE BYTE
                        1563  *
003.135  376 015        1564  IOB2      CPI     A.CR            CARRIAGE RETURN?
003.137  310            1565            RZ                      RETURN IF CARRIAGE RETURN       /JWT 790507/
003.140  257            1566            XRA     A               CLEAR CARRY                     /JWT 790507/
003.141  030 325        1567            JR      IOB1            GET A NEW CHARACTER             /JWT 790507/
```

```
                                1570  **      DYASC - DYNAMIC RAM ASCII OUTPUT TO CONSOLE
                                1571  *
                                1572  *       ENTRY   (A) = CHARACTER TO OUTPUT
                                1573  *               (IY) = RETURN ADDRESS
                                1574  *       EXIT    TO (IY)
                                1575  *       USES    A,C,F
                                1576
    003.143                     1577  DYASC   EQU     *
                                1578  *       EX      AF,AF'          SAVE CHARACTER TO OUTPUT
    003.143  010               1579          DB      MI,EXAF
    003.144  333 355           1580  DYASC1  IN      SC.ACE+UR,LSR   READ LINE STATUS REGISTER
    003.146  346 040           1581          ANI     UC.THE
    003.150  312 144 003       1582          JZ      DYASC1          WAIT IF UART CAN'T HOLD ANOTHER CHARACTER
                                1583
                                1584  *       EX      AF,AF'          GET CHARACTER TO OUTPUT
    003.153  010               1585          DB      MI,EXAF
    003.154  323 350           1586          OUT     SC.ACE+UR,THR   OUTPUT TO UART
                                1587  *       JP      (IY)            RETURN TO CALLER
    003.156  375 351           1588          DB      MI,JIYA,MI,JIYB


                                1590  **      DYBYT - DYNAMIC RAM BYTE OUTPUT
                                1591  *
                                1592  *       ENTRY   (A) = BYTE TO OUTPUT AS OCTAL
                                1593  *               (IX) = RETURN ADDRESS
                                1594  *       EXIT    TO (IX)
                                1595  *       USES    A,C,IY,F
                                1596
    003.160  117               1597  DYBYT   MOV     C,A             SAVE CHARACTER
    003.161  346 300           1598          ANI     11000000B       OUTPUT FIRST CHARACTER OF OCTAL VALUE
    003.163  017               1599          RRC
    003.164  017               1600          RRC
    003.165  017               1601          RRC
    003.166  017               1602          RRC
    003.167  017               1603          RRC
    003.170  017               1604          RRC
    003.171  366 060           1605          ORI     00110000B       MAKE INTO ASCII
                                1606
                                1607  *       LD      IY,DYBYT.2
    003.173  375 041           1608          DB      MI,LDYA,MI,LDYB
    003.175  202 003           1609          DW      DYBYT.2
                                1610
    003.177  303 143 003       1611          JMP     DYASC
                                1612
    003.202  171               1613  DYBYT.2 MOV     A,C             OUTPUT SECOND CHARACTER
    003.203  346 070           1614          ANI     00111000B
    003.205  017               1615          RRC
    003.206  017               1616          RRC
    003.207  017               1617          RRC
    003.210  366 060           1618          ORI     00110000B       MAKE INTO ASCII
                                1619
                                1620  *       LD      IY,DYBYT.4      RETURN ADDRESS
    003.212  375 041           1621          DB      MI,LDYA,MI,LDYB
    003.214  221 003           1622          DW      DYBYT.4
                                1623
```

```
003,216  303 143 003  1624            JMP     DYASC
                      1625
003,221  171          1626  DYBYT,4   MOV     A,C             OUTPUT LAST CHARACTER
003,222  346 007      1627            ANI     00000111B
003,224  366 060      1628            ORI     00110000B       MAKE ASCII
                      1629
                      1630  *         LD      IY,DYBYT.6      RETURN ADDRESS
003,226  375 041      1631            DB      MI.LDYA,MI.LDYB
003,230  235 003      1632            DW      DYBYT.6
                      1633
003,232  303 143 003  1634            JMP     DYASC
                      1635
003,235               1636  DYBYT.6   EQU     *
                      1637  *         JP      (IX)            RETURN TO CALLER
003,235  335 351      1638            DB      MI,JIXA,MI,JIXB


                      1640  **        MSG.PAS - PASS MESSAGE FOR DYNAMIC RAM TEST
                      1641  *
                      1642
003,237  015 012      1643  MSG.PAS   DB      A.CR,A.LF
003,241  040 040 040  1644            DB      '     Pass =    '
003,257  000          1645            DB      0
```

```
003,260                 1648            ORG     3260A
                        1649    **      RCK - READ CONSOLE KEYPAD
                        1650    *
                        1651    *       RCK IS CALLED TO READ A KEYSTROKE FROM THE CONSOLE FRONT PANEL KEYPAD.
                        1652    *       SINCE THE H88/89 DOES NOT HAVE A FRONT PANEL, THIS ROUTINE IS PROVIDED
                        1653    *       ONLY TO MAINTAIN COMPATIBILITY WITH PAM-8.
                        1654    *       RCK WILL IMMEDIATELY RETURN WITH A VALUE OF 0 (ZERO) IN THE ACCUMULATOR.
                        1655    *
                        1656    *       ENTRY   NONE
                        1657    *       EXIT    (A) = 0
                        1658    *       USES    A,F
                        1659
                        1660    *       RCK MUST HAVE SAME ENTRY AS RCK IN PAM-8
000,000                 1661            ERRNZ   *-3260A
                        1662
003,260                 1663    RCK     EQU     *
                        1664
003,260  257           1665            XRA     A
003,261  311           1666            RET
                        1667
```

```
                        1671  **      RCC - READ CONSOLE CHARACTER.
                        1672  *
                        1673  *       RCC IS CALLED TO READ A KEYSTROKE FROM THE CONSOLE.
                        1674  *       IF A RUBOUT/DELETE IS RECEIVED, EXIT IS TO *ERROR*.
                        1675  *
                        1676  *       ENTRY   NONE
                        1677  *       EXIT    TO ERROR  - IF A DELETE OR RUBOUT IS ENCOUNTERED
                        1678  *               TO CALLER - WHEN A KEY IS HIT
                        1679  *               (A) =  ASCII KEY VALUE
                        1680  *       USES    A,F
                        1681
                        1682
                        1683
  003.262               1684  RCC     EQU     *
                        1685
  003.262  333 355      1686  RCC1    IN      SC.ACE+UR.LSR   INPUT ACE LINE STATUS REGISTER
  003.264  346 001      1687          ANI     UC.DR           SEE IF THERE IS A DATA READY
  003.266  050 372      1688          JR      Z,RCC1
                        1689
  003.270  333 350      1690  RCC2    IN      SC.ACE+UR.RBR   ELSE, INPUT CHARACTER
  003.272  346 177      1691          ANI     01111111B       TOSS ANY PARITY
  003.274  376 177      1692          CPI     A.DEL
  003.276  312 322 000  1693          JZ      ERROR           IF RUBOUT, EXIT TO ERROR
                        1694
  003.301  311          1695          RET                     ELSE, EXIT TO CALLER


                        1697  **      WCC - WRITE CONSOLE CHARACTER
                        1698  *
                        1699  *       WRITE A CHARACTER TO THE CONSOLE UART PORT
                        1700  *
                        1701  *       ENTRY   (A) = ASCII CHARACTER TO OUTPUT
                        1702  *       EXIT    NONE
                        1703  *       USES    NONE
                        1704
                        1705
  003.302  365          1706  WCC     PUSH    PSW             SAVE CHARACTER
  003.303  333 355      1707  WCC1    IN      SC.ACE+UR.LSR   INPUT ACE STATUS
  003.305  346 040      1708          ANI     UC.THE          SEE IF TRANSMITTER HOLDING REGISTER IS EMPTY
  003.307  050 372      1709          JR      Z,WCC1
                        1710
  003.311  361          1711          POP     PSW             GET CHARACTER
  003.312  323 350      1712          OUT     SC.ACE+UR.THR           OUTPUT TO CONSOLE
  003.314  311          1713          RET
```

```
                                1715  **      THE FOLLOWING IS ONLY A PORTION OF THE DYNAMIC RAM TEST!!!
                                1716  *
003.315  353                   1717  DY9.3   XCHG
003.316  174                   1718          MOV     A,H             OUTPUT MSB
                                1719
                                1720  *       LD      IX,DY9.4        RETURN ADDRESS
003.317  335 041               1721          DB      MI.LDXA,MI.LDXB
003.321  326 003               1722          DW      DY9.4
                                1723
003.323  303 160 003           1724          JMP     DYBYT
                                1725
003.326  175                   1726  DY9.4   MOV     A,L             OUTPUT LSB
                                1727
                                1728  *       LD      IX,DY9.5        RETURN ADDRESS
003.327  335 041               1729          DB      MI.LDXA,MI.LDXB
003.331  335 003               1730          DW      DY9.5
                                1731
003.333  030 223               1732          JR      DYBYT
                                1733
003.335  353                   1734  DY9.5   XCHG                    SAVE ERROR ADDRESS
003.336  041 362 007           1735          LXI     H,MSG.EQ        OUTPUT ' = '
                                1736
                                1737  *       LD      IX,DY9.8        RETURN ADDRESS
003.341  335 041               1738          DB      MI.LDXA,MI.LDXB
003.343  350 003               1739          DW      DY9.8
                                1740
003.345  303 306 007           1741          JMP     DYMSG           OUTPUT STRING
                                1742
003.350  032                   1743  DY9.8   LDAX    D               OUTPUT RAM CONTENTS
                                1744
                                1745  *       LD      IX,DYMEM10      RETURN ADDRESS
003.351  335 041               1746          DB      MI.LDXA,MI.LDXB
003.353  360 003               1747          DW      DYMEM10
                                1748
003.355  303 160 003           1749          JMP     DYBYT
                                1750
003.360  076 007               1751  DYMEM10 MVI     A,A.BEL         DING BELL
                                1752
                                1753  *       LD      IY,DY10.5       RETURN ADDRESS
003.362  375 041               1754          DB      MI.LDYA,MI.LDYB
003.364  265 007               1755          DW      DY10.5
                                1756
003.366  303 143 003           1757          JMP     DYASC
```

```
                        1760  **      IO ROUTINES TO BE COPIED INTO AND USED IN RAM.
                        1761  *
                        1762  *       MUST CONTINUE TO 3777A FOR PROPER COPY.
                        1763  *       THE TABLE MUST ALSO BE BACKWARDS TO THE FINAL RAM
                        1764
    000.001             1765          IF      .RAM.
                        1767          ELSE
    000.000             1768          ERRNZ   4000A-7-*
                        1769          ENDIF
                        1770
    003.371             1771  PRSROM  EQU     *
    003.371  001       1772          DB      1               REFIND
    003.372  000       1773          DB      0               CTLFLG
    003.373  000       1774          DB      0               .MFLAG
    003.374  000       1775          DB      0               DSPMOD
    003.375  000       1776          DB      0               DSPROT
    003.376  012       1777          DB      10              REGI
    003.377  311       1778          DB      MI.RET
                        1779
    000.001             1780          IF      .RAM.
                        1782          ELSE
    000.000             1783          ERRNZ *-4000A
                        1784          ENDIF
                        1785
```

```
                        1788  ***     INITOX          EXTENSION OF INITO TO SUPPORT H88
                        1789
    004.000  076 002    1790  INITOX  MVI     A,H88B.CK               ENABLE CLOCK
    004.002  323 362    1791          OUT     H88.CTL
                        1792
                        1793  *       SET UP ACE FOR CONSOLE COMMUNICATIONS
                        1794  *
    004.004  076 200    1795          MVI     A,UC.DLA        SET DIVISOR LATCH ACCESS BIT
    004.006  323 353    1796          OUT     SC.ACE+UR.LCR
    004.010  041 101 004 1797          LXI     H,BRTAB         (H,L) = BEGINNING OF BAUD RATE TABLE
    004.013  333 362    1798          IN      H88.SW          INPUT SWITCHES FOR DESIRED BAUD RATE
    004.015  346 100    1799          ANI     H88S.BR         MASK FOR BAUD RATE SWITCHES ONLY
    004.017  017        1800          RRC                     SHIFT FOR A *2 FOR TABLE
    004.020  017        1801          RRC
    004.021  017        1802          RRC
    004.022  017        1803          RRC
    004.023  017        1804          RRC
    004.024  205        1805          ADD     L               ADD DISPLACEMENT FROM BEGINNING OF TABLE
    004.025  157        1806          MOV     L,A
    004.026  176        1807          MOV     A,M             GET MSB OF DIVISOR
    004.027  323 351    1808          OUT     SC,ACE+UR.DLM
    004.031  043        1809          INX     H               GET LSB
    004.032  176        1810          MOV     A,M
    004.033  323 350    1811          OUT     SC.ACE+UR.DLL
    004.035  076 003    1812          MVI     A,UC.8BW        SET 8 BITS, 1 STOP BIT, NO PARITY
    004.037  323 353    1813          OUT     SC.ACE+UR.LCR
    004.041  076 000    1814          MVI     A,0             SET NO INTERRUPTS
    004.043  323 351    1815          OUT     SC.ACE+UR.IER
                        1816
                        1817  *       WAIT A WHILE TO ALLOW THE CONSOLE RESET TO FINISH SO IT CAN
                        1818  *       ACCEPT THE FIRST PROMPT
                        1819  *
    004.045  001 000 065 1820          LXI     B,65000A        APPROX. 100 MS
    004.050  015        1821  INITOX1 DCR     C
    004.051  040 375    1822          JR      NZ,INITOX1
                        1823
    004.053  020 373    1824          DJNZ    INITOX1
                        1825
                        1826  *       INPUT SWITCH TO SEE IF TO BEGIN OPERATION OR MEMORY TEST
                        1827  *
    004.055  333 362    1828          IN      H88.SW          GET SWITCHES
    004.057  346 040    1829          ANI     H88S.M          MASK FOR MEMORY TEST ONLY
    004.061  312 116 007 1830          JZ      DYMEM           IF TO PERFORM MEMORY TESTS
                        1831
                        1832  *       REPLACE WHAT WAS ORIGINALLY AT THE JUMP WHICH GOT US HERE
                        1833  *
    004.064  021 371 003 1834          LXI     D,PRSROM        (DE) = ROM COPY OF PRS CODE
    004.067  257        1835          XRA     A
    004.070  062 123 041 1836          STA     AUTOB           INITIAL AUTO BOOT FLAG
    004.073  062 066 040 1837          STA     DATA            INITIAL 362Q PORT DATA SAVE BYTE
    004.076  303 003 000 1838          JMP     INITO.0         RETURN TO ORIGINAL CODE
```

```
                        1840  **      BRTAB - BAUD RATE DIVISOR TABLE
                        1841  *
    004.101             1842  BRTAB   EQU     *
                        1843
    004.101 000 014     1844  BR96    DB      0,12            9600 BAUD
    004.103 000 006     1845  BR19.2  DB      0,6           19,200 BAUD
                        1846  *BR38.4 DB       0,3          38,400 BAUD
                        1847  *BR56.0 DB       0,2          56,000 BAUD
                        1848
    000.004             1849  .       SET     */256
    000.000             1850          ERRNZ   BRTAB/256-.   TABLE MUST BE IN ONE PAGE


                        1852  ***     SAVALLX - SAVALL EXTENSION TO MAKE ROOM FOR A JUMP TO THE NMI HANDLER
                        1853
    004.105             1854  SAVALLX EQU     *             REPLACE OLD CODE
    004.105 345         1855          PUSH    H             SET ON STACK AS 'REGISTER'
    004.106 325         1856          PUSH    D             SET RETURN ADDRESS
    004.107 021 011 040 1857          LXI     D,CTLFLG
    004.112 032         1858          LDAX    D
    004.113 303 151 000 1859          JMP     SAVALLR       RETURN TO OLD CODE
```

```
                        1862  ****    NMI - NON MASKABLE INTERRUPT
                        1863  *
                        1864  *       NMI IS USED AS THE TRAP FOR ALL ILLEGAL PORT REQUESTS
                        1865  *
                        1866  *       PORT ADDRESSES TRAPPED ARE:
                        1867  *
                        1868  *                       IN 360Q FRONT PANEL KEYBOARD INPUT
                        1869  *                      OUT 360Q FRONT PANEL CONTROL
                        1870  *                      OUT 361Q FRONT PANEL DISPLAY CONTROL
                        1871  *                   IN/OUT 372Q CONSOLE DATA FOR AN 8251A
                        1872  *                      OUT 373Q CONSOLE CONTROL FOR AN 8251A
                        1873  *
                        1874  *
                        1875  *               THESE PORT REQUESTS ARE RESPONDED TO AS FOLLOWS:
                        1876  *
                        1877  *                       IN 360Q RETURNS WITH (A) = 377Q TO SHOW THAT
                        1878  *                                NO FRONT PANEL SWITCHES ARE PRESSED
                        1879  *
                        1880  *                       OUT 360Q MOVES BIT 6 (CB.CLI) TO BIT 1, AND
                        1881  *                                BIT 4 (CB.SSI) INVERTED, TO BIT 0, AND
                        1882  *                                OUTPUTS THESE BITS TO PORT 362Q TO
                        1883  *                                CONTROL THE CLOCK AND SINGLE STEP INTERRUPTS
                        1884  *
                        1885  *                       OUTPUTS TO 361Q, 372Q, AND 373Q JUST RETURN
                        1886  *
                        1887  *                       INPUTS FROM 361Q, 372Q, AND 373Q RETURN WITH (A) = 0
                        1888  *                                TO INDICATE AN EMPTY BUSS
                        1889  *
                        1890  *
                        1891  *               ENTRY   NONE
                        1892  *
                        1893  *               EXIT    NONE
                        1894  *
                        1895  *               USES    (A) ONLY IF 'FAKING' AN INPUT
                        1896
                        1897
004.116  343           1898  NMI     XTHL                    GET RETURN ADDRESS FROM STACK
004.117  042 064 040    1899          SHLD    NMIRET          SAVE FOR LATER USE
004.122  343           1900          XTHL                    PUT RETURN ADDRESS BACK ON STACK
                        1901
004.123  345           1902          PUSH    H               SAVE REGISTERS
004.124  305           1903          PUSH    B
004.125  365           1904          PUSH    PSW
004.126  107           1905          MOV     B,A             SAVE (A) PRIOR TO I/O
004.127  052 064 040    1906          LHLD    NMIRET          GET RETURN ADDRESS
004.132  053           1907          DCX     H               BACK UP TO PORT # WHICH GOT US HERE
004.133  176           1908          MOV     A,M             GET PORT #
                        1909
004.134  376 360       1910          CPI     360Q            PORT 360?
004.136  050 033       1911          JR      Z,NMI1            IF PORT WAS 360Q
                        1912
                        1913  *       PORT REFERENCED WAS 361Q, 372Q, OR 373Q
                        1914  *
004.140  376 361       1915          CPI     361Q            MAKE SURE PORT IS LEGAL
004.142  050 010       1916          JR      Z,NMI0.5          IF LEGAL
                        1917
```

```
004.144  376 372      1918           CPI     372Q
004.146  050 004      1919           JR      Z,NMI0.5
                      1920
004.150  376 373      1921           CPI     373Q
004.152  040 062      1922           JR      NZ,NMI2.5       IF NONE OF THE ABOVE, EXIT
                      1923
004.154  053          1924  NMI0.5   DCX     H               POINT TO IN/OUT INSTRUCTION
004.155  176          1925           MOV     A,M             SEE IF INPUT OR OUTPUT
004.156  376 323      1926           CPI     MI.OUT
004.160  050 054      1927           JR      Z,NMI2.5          IF OUTPUT, JUST EXIT
                      1928
004.162  376 333      1929           CPI     MI.IN
004.164  040 050      1930           JR      NZ,NMI2.5         IF NOT INPUT EITHER, ILLEGAL SO EXIT
                      1931
004.166  361          1932           POP     PSW             RESTORE FLAGS
004.167  076 000      1933           MVI     A,0             ELSE, RETURN LIKE AN EMPTY BUSS
004.171  030 044      1934           JR      NMI3            EXIT
                      1935
004.173  053          1936  NMI1     DCX     H               POINT TO IN/OUT INSTRUCTION
004.174  176          1937           MOV     A,M             GET I/O INSTRUCTION
004.175  376 333      1938           CPI     MI.IN           INPUT?
004.177  040 005      1939           JR      NZ,NMI1.5         IF NOT 'IN'
                      1940
004.201  361          1941           POP     PSW             RESTORE FLAGS
004.202  076 377      1942           MVI     A,11111111B     SHOW 'NO KEYS PRESSED'
004.204  030 031      1943           JR      NMI3            EXIT
                      1944
004.206  376 323      1945  NMI1.5   CPI     MI.OUT          MAKE SURE INTRUCTION IS AN 'OUT'
004.210  040 024      1946           JR      NZ,NMI2.5         IF NOT
                      1947
004.212  170          1948  NMI2     MOV     A,B             GET OUTPUT DATA AGAIN
004.213  346 120      1949           ANI     CB.CLI+CB.SSI   MOVE CLOCK INFO TO BIT 1
004.215  017          1950           RRC
004.216  017          1951           RRC
004.217  017          1952           RRC
004.220  017          1953           RRC
004.221  017          1954           RRC
004.222  070 001      1955           JR      C,NMI2.2
004.224  074          1956           INR     A
004.225  041 066 040  1957  NMI2.2   LXI     H,DATA          OR WITH THE BYTE IN RAM
004.230  266          1958           ORA     M               BEFORE OUTPUT IT
004.231  323 362      1959           OUT     H88.CTL         SET IN HARDWARE
004.233  346 374      1960           ANI     11111100B
004.235  167          1961           MOV     M,A
                      1962
004.236  361          1963  NMI2.5   POP     PSW             RESTORE (A,F)
                      1964
004.237  301          1965  NMI3     POP     B
004.240  341          1966           POP     H
                      1967  *        RETN                    Z80 RETURN FROM NMI
004.241  355 105      1968           DB      355Q,105Q
```

```
                        1971  **      ATB     - AUTO BOOT ROUTINE CONTINUE
                        1972
004,243  167           1973  ATB     MOV     M,A             SET AUTO BOOT FLAG
004,244  076 012       1974          MVI     A,10            SET TO AUTO BOOT ROUTINE
004,246  315 052 003   1975          CALL    LRA.
004,251  021 212 001   1976          LXI     D,AUTOBO        SET AUTO BOOT ROUTINE
004,254  030 016       1977          JR      BOOTX


004,256                 1979          ORG     4256A
                        1980  **      BOOT H-17 OR Z47 ENTRY POINT FOR H88
                        1981  *
                        1982  *       ENTRY   NONE
                        1983  *
                        1984  *       EXIT    (DE) = NORMAL BOOT ROUTINE ADDRESS
                        1985  *
                        1986  *       USES    ALL
                        1987
004,256  041 234 006   1988  BOOT    LXI     H,MSG.BT        COMPLETE BOOT MESSAGE
004,261  315 100 006   1989          CALL    TYPMSG
004,264  076 012       1990          MVI     A,10
004,266  315 052 003   1991          CALL    LRA.            GET LOCATION OF USER PC
004,271  021 261 001   1992          LXI     D,NBOOT         SET ITS VALUE TO THE NORMAL BOOT ROUTINE
004,274  163           1993  BOOTX   MOV     M,E
004,275  043           1994          INX     H
004,276  162           1995          MOV     M,D
                        1996
004,277  303 063 000   1997          JMP     GO.             DO IT
```

```
                        2000  **      TMOUT   -  BOOT CODE TIME OUT ROUTINE
                        2001  *
                        2002  *       TMOUT IS ENTERED FROM TIMER INTERRUPT EVER 100 MS. AND IT WILL
                        2003  *       EXIT:   IF BOOT SUCCESS THEN TIMER OFF.
                        2004  *               IF 15 SECONDS TIME OUT AND BOOT IS NOT SUCCESS YES
                        2005  *                  THEN ABORT BOOT Z47 & TO MONITOR LOOP
                        2006  *               IF < 15S & 3.5S THEN RE-BOOT
                        2007  *
                        2008  *       ENTRY:  (TMFG)  = 1 IF THE TIME OUT IS FOR Z47
                        2009  *                       = 0 IF THE TIME OUT IS FOR H17
                        2010  *       EXIT:   NONE
                        2011  *
                        2012  *       USE:    ALL (WHEN RETURN, ALL REGISTERS ARE RESTORED)
                        2013
004.302                 2014  TMOUT   EQU     *
004.302  333 355        2015          IN      SC.ACE+UR.LSR           INPUT ACE LINE STATUS REGISTER
004.304  346 001        2016          ANI     UC.DR                   SEE IF THERE IS A DATA READY
004.306  050 011        2017          JR      Z,TMOUT4                CHECK IF IT IS <DELETE>
                        2018
004.310  333 350        2019          IN      SC.ACE+UR.RBR           INPUT DATA FROM KB
004.312  346 177        2020          ANI     01111111B               IS IT <DEL>?
004.314  376 177        2021          CPI     A.DEL
004.316  312 171 002    2022          JZ      NODEV                   IF IT, ABORT THE BOOT
                        2023  *                                       ELSE IGNORE THE INPUT
004.321  041 121 041    2024  TMOUT4  LXI     H,TMFG
004.324  176            2025          MOV     A,M
004.325  247            2026          ANA     A
004.326  010            2027          DB      MI.EXAF                 SAVE Z FLAG
004.327  072 033 040    2028          LDA     TICCNT                  GET TIC
004.332  247            2029          ANA     A                       SET ZERO FLAG
004.333  040 024        2030          JR      NZ,TMOUT2               NOT IN 0.5 SECOND
004.335  043            2031          INX     H                       SET TO MYCNT
000.000                 2032          ERRNZ   MYCNT-TMFG-1            MYCNT MUST FOLLOW TMFG
004.336  064            2033          INR     M                       INCREASE THE COUNT FOR 0.5 SECOND
004.337  176            2034          MOV     A,M
004.340  376 036        2035          CPI     30                      CHECK IF MORE THAN 15 SECONDS
004.342  322 171 002    2036          JNC     NODEV                   NO DEVICE ?
004.345  336 007        2037  TMOUT1  SBI     7                       IS IT 3.5 SECONDS?
004.347  070 010        2038          JR      C,TMOUT2                IF NOT, WAIT
004.351  040 372        2039          JR      NZ,TMOUT1               CHECK MORE
004.353  010            2040          DB      MI.EXAF
004.354  302 063 002    2041          JNZ     RETRY                   IF IT IS Z47, THEN RE-BOOT
004.357  030 002        2042          JR      TMOUT3                  IT IS H-17, CONTINUE IT CLOCK ROUTINE
004.361  010            2043  TMOUT2  DB      MI.EXAF                 CHECK IT IS Z47 OR H17
004.362  300            2044          RNZ                             Z47, THEN RETURN
004.363  303 031 034    2045  TMOUT3  JMP     CLOCK17                 CONTINUE H17 CLOCK ROUTINE
```

```
004.370                 2048            ORG     4370A
                        2049    **      SUBM - SUBSTITUTE MEMORY
                        2050    *
                        2051    *       SUBM INPUTS A MEMORY ADDRESS FROM THE CONSOLE AND THEN DISPLAYS
                        2052    *       THAT ADDRESS AND ITS CONTENTS.  IF A CARRIAGE RETURN IS THEN TYPED,
                        2053    *       CONTROL RETURNS TO THE MONITOR.  IF A SPACE IS TYPED, THE NEXT
                        2054    *       MEMORY LOCATION AND CONTENTS ARE DISPLAYED.  IF A MINUS SIGN IS
                        2055    *       TYPED, THE PREVIOUS MEMORY LOCATION AND CONTENTS ARE DISPLAYED.
                        2056    *       IF AN OCTAL CHARACTER IS TYPED, A BYTE IS ENTERED AN PLACED AT THE
                        2057    *       CURRENT MEMORY LOCATION.
                        2058    *
                        2059    *
                        2060    *       ENTRY   NONE
                        2061    *       EXIT    NONE
                        2062    *       USES    A,E,H,L,F
                        2063
                        2064
004.370 041 201 006     2065    SUBM    LXI     H,MSG.SUB       COMPLETE SUBSTITUTE MESSAGE
004.373 315 100 006     2066            CALL    TYPMSG
004.376 315 150 005     2067            CALL    IROC            INPUT FIRST CHARACTER
005.001 320             2068            RNC                     IF A RETURN, EXIT
                        2069
005.002 041 003 040     2070            LXI     H,IOWRK+1       ELSE, INPUT STARTING ADDRESS
005.005 026 015         2071            MVI     D,A.CR          ENDING WITH A RETURN
005.007 315 062 003     2072            CALL    IOA
005.012 353             2073            XCHG                    (H,L) = INPUT ADDRESS
                        2074
005.013 315 313 005     2075    SUBM1   CALL    TOA             TYPE CRLF, ADDRESS, AND A SPACE
005.016 176             2076            MOV     A,M             GET MEMORY CONTENTS FOR DISPLAY
005.017 315 343 005     2077            CALL    TOB
005.022 076 040         2078            MVI     A,' '           SPACE
005.024 315 302 003     2079            CALL    WCC
                        2080
005.027 315 301 005     2081    SUBM2   CALL    IOC             INPUT FIRST CHARACTER
005.032 322 075 005     2082            JNC     SUBM7           IF FIRST CHARACTER IS OCTAL
                        2083
005.035 376 040         2084            CPI     ' '             SPACE?
005.037 302 046 005     2085            JNZ     SUBM4           IF NOT A SPACE
                        2086
005.042 043             2087    SUBM3   INX     H               POINT TO NEXT ADDRESS
005.043 303 013 005     2088            JMP     SUBM1           DISPLAY NEXT
                        2089
005.046 376 055         2090    SUBM4   CPI     '-'             MINUS?
005.050 302 062 005     2091            JNZ     SUBM6           IF NOT
                        2092
005.053 315 302 003     2093    SUBM5   CALL    WCC             ECHO HYPHEN
005.056 053             2094            DCX     H               POINT TO PREVIOUS ADDRESS
005.057 303 013 005     2095            JMP     SUBM1           DISPLAY PREVIOUS
                        2096
005.062 376 015         2097    SUBM6   CPI     A.CR            RETURN?
005.064 310             2098            RZ                      IF RETURN, EXIT
                        2099
005.065 076 007         2100            MVI     A,A.BEL         ELSE, DING BELL
005.067 315 302 003     2101            CALL    WCC
005.072 303 027 005     2102            JMP     SUBM2           TRY AGAIN
                        2103
```

```
005.075 066 000      2104  SUBM7  MVI   M,0             ZERO BYTE TO BE BUILT
                     2105
005.077 315 302 003  2106  SUBM8  CALL  WCC             ECHO OCTAL CHARACTER
005.102 346 007      2107         ANI   00000111B       GET BINARY VALUE
005.104 137          2108         MOV   E,A             SAVE PARTIAL
005.105 176          2109         MOV   A,M             GET CURRENT
005.106 007          2110         RLC                   MAKE ROOM FOR NEW CHARACTER
005.107 007          2111         RLC
005.110 007          2112         RLC
005.111 346 370      2113         ANI   11111000B       TOSS PREVIOUS LSB
005.113 263          2114         ORA   E               ADD NEW
005.114 167          2115         MOV   M,A             SAVE NEW TOTAL
005.115 315 301 005  2116  SUBM9  CALL  IOC             INPUT NEXT CHARACTER
005.120 322 077 005  2117         JNC   SUBM8           IF OCTAL
                     2118
005.123 376 040      2119         CPI   ' '             SPACE?
005.125 312 042 005  2120         JZ    SUBM3           IF SPACE, DISPLAY NEXT BYTE
                     2121
005.130 376 055      2122         CPI   '-'             MINUS?
005.132 312 053 005  2123         JZ    SUBM5           IF MINUS, DISPLAY PREVIOUS
                     2124
005.135 376 015      2125         CPI   A.CR            RETURN?
005.137 310          2126         RZ                    IF RETURN, EXIT
                     2127
005.140 076 007      2128         MVI   A,A.BEL         ELSE, DING BELL
005.142 315 302 003  2129         CALL  WCC
005.145 303 115 005  2130         JMP   SUBM9           TRY AGAIN
```

```
                     2133  **     IROC - INPUT A RETURN OR AN OCTAL CHARACTER
                     2134  *
                     2135  *      IROC INPUTS A CHARACTER FROM THE CONSOLE AND WAITS UNTIL IT
                     2136  *      RECEIVES EITHER A VALID OCTAL CAHRACTER OR A CARRIAGE RETURN
                     2137  *
                     2138  *      ENTRY  NONE
                     2139  *      EXIT   (A) = INPUT CHARACTER
                     2140  *             'C' = SET IF CHARACTER IS OCTAL
                     2141  *      USES   A,F
                     2142
                     2143
005.150 315 262 003  2144  IROC   CALL  RCC             INPUT CHARACTER
005.153 376 015      2145         CPI   A.CR            RETURN?
005.155 310          2146         RZ                    IF A CR
                     2147
005.156 376 060      2148         CPI   '0'             < 0?
005.160 332 166 005  2149         JC    IROC1           IF < OCTAL
                     2150
005.163 376 070      2151         CPI   '8'             > 8?
005.165 330          2152         RC                    IF OCTAL
                     2153
005.166 076 007      2154  IROC1  MVI   A,A.BEL         ELSE, RING BELL
005.170 315 302 003  2155         CALL  WCC
005.173 303 150 005  2156         JMP   IROC            TRY AGAIN
```

```
                        2158  **      IOA1 - INPUT OCTAL ADDRESS
                        2159  *
                        2160  *       IOA1 IS A CONTINUATION OF *IOA* AND INPUTS A SPLIT OCTAL ADDRESS
                        2161  *       WITHOUT REQUIRING LEADING ZEROS
                        2162  *
                        2163  *       ENTRY   (H,L) = ADDRESS + 1 WHERE INPUT ADDRESS IS TO BE PLACED
                        2164  *               (A) = FIRST OCTAL CHARACTER IF 'C' IS SET
                        2165  *       EXIT    (D,E) = INPUT ADDRESS
                        2166  *               (A) = LAST INPUT CHARACTER
                        2167  *       USES    A,D,E,H,L,F
                        2168
                        2169
005,176  305           2170  IOA1    PUSH    B               SAVE (B,C)
005,177  102           2171          MOV     B,D             (B) = TERMINATION CHARACTER
005,200  345           2172          PUSH    H               SAVE ADDRESS WHERE INPUT IS TO BE PLACED
005,201  041 000 000    2173          LXI     H,0             SET NEW VALUE TO ZERO
005,204  324 262 003    2174  IOA2    CNC     RCC             IF CARRY SET, FIRST CHARACTER IS IN ACC
005,207  376 060        2175          CPI     '0'             MAKE SURE CHARACTER IS OCTAL
005,211  332 242 005    2176          JC      IOA3            IF < OCTAL
                        2177
005,214  376 070        2178          CPI     '8'
005,216  322 242 005    2179          JNC     IOA3            IF > OCTAL
                        2180
005,221  315 302 003    2181          CALL    WCC             ECHO OCTAL CHARACTER
005,224  346 007        2182          ANI     00000111B       GET BINARY VALUE
005,226  365           2183          PUSH    PSW             SAVE NEW CHARACTER VALUE
005,227  051           2184          DAD     H               SHIFT THREE TO MAKE ROOM FOR NEW CHARACTER
005,230  051           2185          DAD     H
005,231  051           2186          DAD     H
005,232  365           2187          PUSH    PSW             SAVE CARRY FROM DAD
005,233  321           2188          POP     D               SAVE FLAG RESULT IN E
005,234  361           2189          POP     PSW             RETURN NEW CHARACTER VALUE TO (A)
005,235  205           2190          ADD     L
005,236  157           2191          MOV     L,A
005,237  303 204 005    2192          JMP     IOA2            SEE IF MORE CHARACTERS
                        2193
005,242  270           2194  IOA3    CMP     B               TERMINATING CHARACTER?
005,243  312 260 005    2195          JZ      IOA4            IF EQUAL
                        2196
005,246  076 007        2197          MVI     A,A.BEL         ELSE, DING BELL
005,250  315 302 003    2198          CALL    WCC
005,253  067           2199          STC                     TRY AGAIN
005,254  077           2200          CMC
005,255  303 204 005    2201          JMP     IOA2
                        2202
                        2203  *       END OF INPUT, PUT VALUE IN MEMORY AND EXIT
                        2204
005,260  315 302 003    2205  IOA4    CALL    WCC             ECHO CHARACTER
005,263  127           2206          MOV     D,A             LAST CHARACTER TO D
005,264  325           2207          PUSH    D
005,265  361           2208          POP     PSW             (PSW) = RESULT OF DAD
005,266  174           2209          MOV     A,H             MAKE (H) INTO SPLIT OCTAL
005,267  037           2210          RAR
005,270  147           2211          MOV     H,A
005,271  172           2212          MOV     A,D             RESTORE LAST INPUT CHARACTER
005,272  353           2213          XCHG                    (D,E) = INPUT ADDRESS
```

```
005.273  341          2214          POP     H               (H,L) = LOCATION TO PLACE THIS ADDRESS
005.274  162          2215          MOV     M,D
005.275  053          2216          DCX     H
005.276  163          2217          MOV     M,E
005.277  301          2218          POP     B               RESTORE (B,C)
005.300  311          2219          RET


                      2221  **      IOC - INPUT OCTAL CHARACTER
                      2222  *
                      2223  *
                      2224  *       ENTRY   NONE
                      2225  *       EXIT    (A) = INPUT CHARACTER
                      2226  *               'C' = SET IF CHARACTER NOT OCTAL
                      2227  *       USES    A,F
                      2228
                      2229
005.301  315 262 003  2230  IOC     CALL    RCC             INPUT CHARACTER
005.304  376 060      2231          CPI     '0'
005.306  330          2232          RC                      IF CHARACTER < OCTAL
                      2233
005.307  376 070      2234          CPI     '8'             CHARACTER > OCTAL?
005.311  077          2235          CMC                     'C' IF GREATER THAN
005.312  311          2236          RET


                      2238  **      TOA - TYPE OCTAL ADDRESS
                      2239  *
                      2240  *       TOA OUTPUTS TO THE CONSOLE A CRLF, THE SPCECIFIED ADDRESS AND A SPACE
                      2241  *
                      2242  *       ENTRY   (H,L) = ADDRESS TO BE DISPLAYED
                      2243  *       EXIT    NONE
                      2244  *       USES    A,B,C,F
                      2245
                      2246
005.313  076 015      2247  TOA     MVI     A,A.CR          CRLF
005.315  315 302 003  2248          CALL    WCC
005.320  076 012      2249          MVI     A,A.LF
005.322  315 302 003  2250          CALL    WCC
                      2251
005.325  174          2252  TOA.    MOV     A,H             ADDRESS
005.326  315 343 005  2253          CALL    TOB
005.331  175          2254          MOV     A,L
005.332  315 343 005  2255          CALL    TOB
                      2256
005.335  076 040      2257          MVI     A,' '           SPACE
005.337  315 302 003  2258          CALL    WCC
005.342  311          2259          RET
```

```
                                2261  **      TOB - TYPE OCTAL BYTE
                                2262  *
                                2263  *       TOB OUTPUTS TO THE CONSOLE IN OCTAL, THE BYTE IN A
                                2264  *
                                2265  *       ENTRY   (A) = BYTE TO BE OUTPUT
                                2266  *       EXIT    NONE
                                2267  *       USES    A,F
                                2268
                                2269
005.343  305                   2270  TOB     PUSH    B
005.344  006 002               2271          MVI     B,2             NUMBER OF CHARACTERS - 1
005.346  117                   2272          MOV     C,A             SAVE ORIGINAL BYTE
005.347  267                   2273          ORA     A               ASSURE 'C' = ZERO
005.350  037                   2274          RAR
005.351  037                   2275          RAR                     SHIFT TOP BYTE TO LSB
005.352  037                   2276          RAR
005.353  037                   2277  TOB1    RAR                     SHIFT MIDDLE BYTE TO LSB
005.354  037                   2278          RAR
005.355  037                   2279          RAR
005.356  346 007               2280          ANI     00000111B       MASK FOR HALF ASCII
005.360  366 060               2281          ORI     00110000B       MAKE WHOLE ASCII
005.362  315 302 003           2282          CALL    WCC             OUTPUT TO CONSOLE
005.365  171                   2283          MOV     A,C             GET ORIGINAL BYTE
005.366  005                   2284          DCR     B
005.367  302 353 005           2285          JNZ     TOB1            IF SECOND BYTE STILL NEEDS TO BE OUTPUT
                                2286
005.372  346 007               2287          ANI     00000111B       ELSE, OUTPUT LAST CHARACTER
005.374  366 060               2288          ORI     00110000B
005.376  315 302 003           2289          CALL    WCC
006.001  301                   2290          POP     B
006.002  311                   2291          RET
```

```
                                2293  **      WCR - WAIT FOR A CARRIAGE RETURN
                                2294  *
                                2295  *       WCR INPUTS CHARACTERS FROM THE CONSOLE UNTIL A CARRIAGE RETURN
                                2296  *       IS RECEIVED AND THEN ECHOS A CRLF
                                2297  *
                                2298  *
                                2299  *       ENTRY   NONE
                                2300  *       EXIT    NONE
                                2301  *       USES    A,F
                                2302
                                2303
006.003  315 262 003           2304  WCR     CALL    RCC             INPUT CHARACTER
006.006  376 015               2305          CPI     A,CR
006.010  040 371               2306          JR      NZ,WCR            IF NOT A CR
                                2307
006.012  315 302 003           2308          CALL    WCC             ELSE, ECHO CR
006.015  076 012               2309          MVI     A,A.LF          LINE FEED
006.017  315 302 003           2310          CALL    WCC
006.022  311                   2311          RET
```

```
                        2313  ***   DAT     -       DATA BYTE OUTPUT TO Z-47
                        2314  *
                        2315  *     ENTRY:  (A) = BYTE TO OUTPUT
                        2316  *
                        2317  *     EXIT:   (A) = BYTE TO OUTPUT
                        2318  *             (D) = S.DTR
                        2319  *
                        2320  *     USE:    AF, D
                        2321
006.023                 2322  DAT     EQU     *
006.023  026 200        2323          MVI     D,S.DTR                 SET MATCH CONDITION TO DATA TRANSFER
006.025  030 002        2324          JR      COM1                    REQUEST BIT


                        2326  ***   COM     - OUTPUT COMMAND BYTE TO Z-47
                        2327  *
                        2328  *     ENTRY:  (A) = COMMAND BYTE
                        2329  *
                        2330  *     EXIT:   (A) = COMMAND BYTE
                        2331  *             (D) = S.DON
                        2332  *
                        2333  *     USE:    AF, D
                        2334
006.027                 2335  COM     EQU     *
006.027  026 040        2336          MVI     D,S.DON                 SET MATCH CONDITION TO DONE BIT
006.031  365            2337  COM1    PUSH    PSW
006.032  315 170 006    2338  WTDON1  CALL    IN.                     READ CONTROLLER STATUS REGISTER
006.035  242            2339          ANA     D                       GET MATCH BIT ONLY
006.036  050 372        2340          JR      Z,WTDON1                IF NO MATCH, WAIT
006.040  361            2341          POP     PSW
006.041  315 146 006    2342          CALL    OUT1.                   OUTPUT THE BYTE TO THE DATA PORT
006.044  311            2343          RET


006.045                 2345          ORG     6045A
                        2346  **    HRNX - HORN EXTENSION ROUTINE
                        2347  *
                        2348  *     THIS IS AN EXTENSION TO *HORN* TO MAKE ROOM FOR A JUMP
                        2349
006.045  056 011        2350  HRNX    MVI     L,#CTLFLG
006.047  163            2351          MOV     M,E                     TURN OFF HORN
006.050  321            2352          POP     D
006.051  341            2353          POP     H
006.052  311            2354          RET
```

```
                        2356  **      NOISE - DING BELL ON CONSOLE
                        2357  *
                        2358  *       THIS IS A MODIFICATION TO ALLOW THE H88/H89 TO USE THE CONSOLE BELL
                        2359
006.053 076 007         2360  NOISE   MVI     A,A,BEL
006.055 315 302 003     2361          CALL    WCC
006.060 303 140 002     2362          JMP     HORN            CONTINUE WITH NORMAL HORN DELAY



                        2364  **      OUT.    - OUTPUT BYTE TO Z-47
                        2365  *
                        2366  *       ENTRY:  (A) = OUTPUT BYTE
                        2367  *
                        2368  *       EXIT:   NONE
                        2369  *
                        2370  *       USE:    NONE
                        2371
006.063                 2372  OUT.    EQU     *
006.063 305             2373          PUSH    B
006.064 107             2374          MOV     B,A             SAVE THE OUTPUT DATA
006.065 072 120 041     2375          LDA     PRIM            GET PORT ADDRESS
006.070 117             2376  OUT.1   MOV     C,A             SET TO REG C
006.071 170             2377          MOV     A,B             GET OUTPUT BYTE DATA BACK
                        2378  *       OUT     (C),A           OUTPUT BYTE
006.072 355 171         2379          DB      355Q,171Q
006.074 301             2380          POP     B
006.075 311             2381          RET



006.100                 2383          ORG     6100A
                        2384  **      TYPMSG - TYPE MESSAGE TO CONSOLE
                        2385  *
                        2386  *       TYPMSG OUTPUTS AN ASCII MESSAGE FROM MEMORY TO THE CONSOLE
                        2387  *       UNTIL A NULL IS SENSED
                        2388  *
                        2389  *       ENTRY   (H,L) = ADDRESS OF MESSAGE
                        2390  *       EXIT    NONE
                        2391  *       USES    A,H,L,F
                        2392
                        2393
006.100 176             2394  TYPMSG  MOV     A,M             GET CHARACTER
006.101 267             2395          ORA     A               SEE IF A NULL
006.102 310             2396          RZ                      IF NULL, EXIT
                        2397
006.103 315 302 003     2398          CALL    WCC             ELSE OUTPUT CHARACTER TO CONSOLE
006.106 043             2399          INX     H               POINT TO NEXT CHARACTER
006.107 030 367         2400          JR      TYPMSG          OUTPUT IT
```

```
                          2402  **      MSG.PR - MESSAGE FOR MONITOR PROMPT
                          2403  *
                          2404  *       CRLF,' H: '
                          2405
                          2406
006.111  015 012 040     2407  MSG.PR  DB      A.CR,A.LF,'  H: ',0


                          2409  **      RDBLCK - INPUT A BLOCK FROM Z-47
                          2410  *
                          2411  *       RDBLCK  READS IN A BLOCK FROM THE DISK CONTROOLER
                          2412  *
                          2413  *       ENTRY:
                          2414  *               HL = LOAD ADDRESS
                          2415  *               B  = COUNT
                          2416  *               C  = SIDE/UNIT/SECTOR
                          2417  *
                          2418  *       EXIT:   NONE
                          2419  *
                          2420  *       USES:   ALL
                          2421
006.121                   2422  RDBLCK  EQU     *
006.121  076 007          2423  RD1     MVI     A,DC.REAR
006.123  315 027 006      2424          CALL    COM             SEND THE COMMAND
006.126  257              2425          XRA     A               FOR TRACK 0
006.127  315 023 006      2426          CALL    DAT             SEND IT TO DISK
006.132  171              2427          MOV     A,C             LOAD SIDE/UNIT/SECTOR
006.133  315 023 006      2428          CALL    DAT             SEND IT TO DISK
                          2429
006.136  315 067 001      2430  RD2     CALL    PIN             INPUT A BYTE FROM DISK
006.141  167              2431          MOV     M,A             STORE IN BUFFER
006.142  043              2432          INX     H               BUFFER TO NEXT ADDRESS
006.143  020 371          2433          DJNZ    RD2
006.145  311              2434          RET                     CONTINUE


                          2436  **      OUT1, - OUTPUT BYTE TO PORT (PRIM+1)
                          2437  *
                          2438  *       ENTRY:  (A) = OUTPUT PORT
                          2439  *
                          2440  *       EXIT:   NONE
                          2441  *
                          2442  *       USE:    NONE
                          2443
006.146                   2444  OUT1,   EQU     *
006.146  305              2445          PUSH    B
006.147  107              2446          MOV     B,A             SAVE THE OUTPUT DATA
006.150  072 120 041      2447          LDA     PRIM            GET PORT ADDRESS
006.153  074              2448          INR     A               SET TO (PRIM+1)
006.154  030 312          2449          JR      OUT.1           GO TO OUTPUT ROUTINE
```

```
                        2451  **      IN1.    - INPUT BYTE FROM (PRIM+1) PORT
                        2452  *
                        2453  *       ENTRY:  NONE
                        2454  *
                        2455  *       EXIT:   (A) = INPUT BYTE
                        2456  *
                        2457  *       USE:    A
                        2458
006.156                 2459  IN1.    EQU     *
006.156 305             2460          PUSH    B
006.157 072 120 041     2461          LDA     PRIM            GET PORT ADDRESS
006.162 074             2462          INR     A               SET TO (PRIM+1)
006.163 030 007         2463          JR      IN.1            GO TO INPUT ROUTINE


006.165                 2465          ORG     6165A
                        2466  **      MSG.GO - (G)O
                        2467  *
                        2468  *       'GO'
                        2469
006.165 157 040 000     2470  MSG.GO  DB      'o ',0


                        2472  **      IN.     - INPUT BYTE FROM PORT (PRIM)
                        2473  *
                        2474  *       ENTRY:  NONE
                        2475  *
                        2476  *       EXIT:   (A) = INPUT BYTE
                        2477  *
                        2478  *       USE:    A
                        2479
006.170                 2480  IN.     EQU     *
006.170 305             2481          PUSH    B
006.171 072 120 041     2482          LDA     PRIM            GET PORT ADDRESS
006.174 117             2483  IN.1    MOV     C,A             SET ADDR. TO REG C
                        2484  *       IN      A,(C)
006.175 355 170         2485          DB      355Q,170Q       INPUT BYTE
006.177 301             2486          POP     B
006.200 311             2487          RET


006.201                 2489          ORG     6201A
                        2490  **      MSG.SUB - (S)UBSTITUTE
                        2491  *
                        2492  *       'SUBSTITUTE'
                        2493
006.201 165 142 163     2494  MSG.SUB DB      'ubstitute ',0
```

```
                        2496  **      MSG.PC - (P)ROGRAM COUNTER
                        2497  *
                        2498  *       'PROGRAM COUNTER'
                        2499
 006.214 162 157 147    2500  MSG.PC  DB      'rogram Counter',0


                        2502  **      MSG.BT - (B)OOT
                        2503  *
                        2504  *       'BOOT'
                        2505
 006.234 157 157 164    2506  MSG.BT  DB      'oot',0
```

```
                        2509  ***   SPEED - ROTATIONAL SPEED TEST FOR 5.25 INCH DISK DRIVE
                        2510  *
                        2511  *     *SPEED* IS USED ONLY FOR GROSS ADJUSTMENT OF DRIVE ROTATIONAL
                        2512  *     SPEED IF THE FIRST READ/WRITE TEST OF THE UNIT FAILS DURING SET UP.
                        2513  *
                        2514  *              USE OF *SPEED* IS AS FOLLOWS:
                        2515  *
                        2516  *                      1.  ENTER 'GO AND THE ENTRY ADDRESS OF *SPEED*
                        2517  *                      2.  ADJUST DRIVE SPEED UNTIL DATA AT DISPLAYED
                        2518  *                          EQUALS 200
                        2519  *                              A.  IF SPEED < 200, TURN ADJUSTMENT CLOCKWISE
                        2520  *                              B.  IF SPEED > 200, TURN COUNTERCLOCKWISE
                        2521  *
                        2522  *     THE ABOVE TEST ADJUSTS SY0:.  TO ADJUST SY1:, USE HDOS


                        2524  **    LABLE EQUIVALENCES
                        2525  *
                        2526  *     I/O PORTS
   000,177              2527  OP.DC   EQU     177Q            DRIVE CONTROL OUTPUT PORT
   000,177              2528  IP.DS   EQU     177Q            DRIVE STATUS INPUT PORT

                        2530  *     MASKS
                        2531  *
   000.001              2532  DS.HOLE EQU     00000001B       DRIVE STATUS SECTOR/INDEX HOLE

                        2534  *     CONSTANTS
                        2535  *
   000.022              2536  ONDRO   EQU     022Q            TURN ON SY0:
```

```
006,240 041 371 006  2538  SPEED   LXI   H,MSG.SPD      OUTPUT SPEED MESSAGE
006,243 315 100 006  2539          CALL  TYPMSG
006,246 076 000      2540          MVI   A,0            SET FLAG AT IOWRK FOR 'WORKING' MESSAGE
006,250 062 002 040  2541          STA   IOWRK
006,253 076 022      2542          MVI   A,ONDRO        TURN ON DRIVE ZERO
006,255 323 177      2543          OUT   OP.DC
006,257 052 033 040  2544  SPEED1  LHLD  TICCNT         GET TICK COUNTER
006,262 174          2545          MOV   A,H            FORM TWO'S COMPLEMENT OF TICK COUNTER
006,263 057          2546          CMA
006,264 127          2547          MOV   D,A            (D,E) = NEGATIVE TICK COUNTER
006,265 175          2548          MOV   A,L
006,266 057          2549          CMA
006,267 074          2550          INR   A
006,270 137          2551          MOV   E,A
006,271 322 275 006  2552          JNC   SPEED2         IF NO CARRY FROM LSB
                     2553
006,274 024          2554          INR   D              ELSE, INCREMENT MSB
006,275 001 000 000  2555  SPEED2  LXI   B,0            ZERO REV COUNTERS
006,300 333 177      2556  SPEED3  IN    IP.DS          INPUT DISK STATUS
006,302 346 001      2557          ANI   DS.HOLE        MASK FOR SECTOR/INDEX PULSES
006,304 312 300 006  2558          JZ    SPEED3         IF NO HOLE PRESENT
                     2559
                     2560  *       HOLE PRESENT, WAIT FOR IT TO LEAVE
                     2561  *
006,307 333 177      2562  SPEED4  IN    IP.DS          GET DISK STATUS
006,311 346 001      2563          ANI   DS.HOLE        GET HOLE PULSES
006,313 302 307 006  2564          JNZ   SPEED4         WAIT UNTIL HOLE IS GONE AND WE HAVE MEDIA
                     2565
006,316 004          2566          INR   B              INCREMENT HOLE COUNTER
006,317 170          2567          MOV   A,B            TEST FOR FIVE REVOLUTIONS
006,320 376 070      2568          CPI   56
006,322 302 300 006  2569          JNZ   SPEED3         NOT FIVE, WAIT FOR MORE HOLES
                     2570
                     2571  *       HAVE FIVE REVS, DISPLAY DIFFERENCE OF TICK COUNTER AND EXPECTED TIME DIF
                     2572  *
006,325 052 033 040  2573          LHLD  TICCNT         GET CURRENT TICK VALUE
006,330 031          2574          DAD   D              SUBTRACT START VALUE
006,331 021 214 376  2575          LXI   D,377377A-500+1+200Q   SUBTRACT 500 FOR REVS, +200Q FOR OFFSET
006,334 031          2576          DAD   D              (H,L) = OFFSET RESULT
006,335 345          2577          PUSH  H              SAVE RESULT
006,336 041 062 007  2578          LXI   H,MSG.WRK      POINT TO 'WORKING' MESSAGE
006,341 072 002 040  2579          LDA   IOWRK          GET 'WORKING' FLAG
006,344 356 001      2580          XRI   1              INVERT LOWER BIT
006,346 062 002 040  2581          STA   IOWRK          SAVE NEW VALUE
006,351 302 357 006  2582          JNZ   SPEED5         IF TO DISPLAY 'WORKING'
                     2583
006,354 041 100 007  2584          LXI   H,MSG.HSS      POINT TO 'HOME', 'SPACES', AND SPEED MSG
006,357 315 100 006  2585  SPEED5  CALL  TYPMSG         OUTPUT MESSAGE
006,362 341          2586          POP   H              GET TEST RESULT
006,363 315 325 005  2587          CALL  TOA.           OUTPUT RESULT TO CONSOLE
006,366 303 257 006  2588          JMP   SPEED1         PERFORM ANOTHER SAMPLE
```

```
                                2590  **      MSG.SPD - SPEED TEST MESSAGE
                                2591  *
                                2592  *       .       Disk drive rotational speed test.
                                2593  *
                                2594  *
                                2595  *                       Drive speed = '
                                2596
006.371  033 105 012            2597  MSG.SPD DB      A.ESC,'E',A.LF
006.374  011 104 151            2598          DB      '       Disk drive rotational speed test.',A.CR,A.LF,A.LF
007.041  011 011 104            2599          DB      '               Drive speed = '
007.061  000                    2600          DB      0


                                2602  **      MSG.WRK - 'WORKING' MESSAGE FOR SPEED TEST
                                2603  *
                                2604  *       DISPLAYS 'WORKING' AT HOME POSITION AND RETURNS CURSOR TO SPEED =
                                2605
007.062  033 110                2606  MSG.WRK DB      A.ESC,'H'       CURSOR HOME
007.064  127 157 162            2607          DB      'Working'
007.073  033 131 043            2608          DB      A.ESC,'Y#>'     CURSOR ADDRESS OF SPEED = VALUE
007.077  000                    2609          DB      0               END MESSAGE


                                2611  **      MSG.HSS - BLANKS 'WORKING' MESSAGE
                                2612  *
                                2613
007.100  033 110                2614  MSG.HSS DB      A.ESC,'H'       CURSOR HOME
007.102  040 040 040            2615          DB      '   '           BLANKS
007.111  033 131 043            2616          DB      A.ESC,'Y#>'     CURSOR ADDRESS OF SPEED = VALUE
007.115  000                    2617          DB      0               END MESSAGE
```

```
                        2620  **      DYMEM - DYNAMIC MEMORY TEST
                        2621  *
                        2622  *       DYMEM TESTS THE DYNAMIC MEMORY IN THE H88/H89 BY PLACING
                        2623  *       A KNOWN PATTERN IN EACH DYNAMIC MEMORY CELL AND THEN
                        2624  *       PERFORMING A READ, INCREMENT, READ SEQUENCE WITH A DELAY
                        2625  *       BETWEEN EACH PASS OF THE TEST
                        2626  *
                        2627  *
                        2628  *       ENTRY   NONE
                        2629  *
                        2630  *       EXIT    ON RESET
                        2631  *
                        2632  *       USES    A,B,C,D,E,H,L,F,A',F',IX,IY
                        2633
                        2634
007.116  076 000        2635  DYMEM   MVI     A,0             MAKE SURE CLOCK AND SINGLE STEP ARE OFF
007.120  323 362        2636          OUT     H88.CTL
                        2637
                        2638  *       DETERMINE END OF MEMORY
                        2639  *
000.001                 2640          IF      .RAM.
                        2643          ELSE
007.122  041 000 040    2644  DYMEM1  LXI     H,START
                        2645          ENDIF
007.125  076 001        2646          MVI     A,1
007.127  066 000        2647  DYMEM2  MVI     M,0             SET RAM TO ZERO
007.131  064            2648          INR     M               SET MEMORY TO ONE
007.132  276            2649          CMP     M               SEE IF (A) = ((H,L))
007.133  040 003        2650          JR      NZ,DYMEM3       IF NOT EQUAL, THE END OF RAM HAS BEEN REACHED
                        2651
007.135  043            2652          INX     H               ELSE, POINT TO NEXT LOCATION IN RAM
007.136  030 367        2653          JR      DYMEM2
                        2654
                        2655
007.140  053            2656  DYMEM3  DCX     H               POINT TO LAST GOOD LOCATION
007.141  353            2657          XCHG                    PUT ENDING ADDRESS IN D,E
007.142  041 324 007    2658          LXI     H,MSG.RAM       OUTPUT ENDING ADDRESS
                        2659
                        2660  *       LD      IX,DY3.3        RETURN ADDRESS
007.145  335 041        2661          DB      MI.LDXA,MI.LDXB
007.147  153 007        2662          DW      DY3.3
                        2663
007.151  030 133        2664          JR      DYMSG
                        2665
007.153  172            2666  DY3.3   MOV     A,D             OUTPUT ADDRESS MSB
                        2667
                        2668  *       LD      IX,DY3.5        RETURN ADDRESS
007.154  335 041        2669          DB      MI.LDXA,MI.LDXB
007.156  163 007        2670          DW      DY3.5
                        2671
007.160  303 160 003    2672          JMP     DYBYT
                        2673
007.163  173            2674  DY3.5   MOV     A,E             LSB
                        2675
                        2676  *       LD      IX,DY3.7        RETURN ADDRESS
007.164  335 041        2677          DB      MI.LDXA,MI.LDXB
```

```
007.166  173 007        2678          DW      DY3.7
                        2679
007.170  303 160 003    2680          JMP     DYBYT
                        2681
007.173  023            2682  DY3.7   INX     D               (D,E) = LAST BYTE OF RAM + 1
                        2683
                        2684  *       TEST MEMORY
                        2685  *
007.174  006 001        2686          MVI     B,1             (B) = CONTENTS OF RAM AFTER SIZING
007.176  041 237 003    2687          LXI     H,MSG.PAS       OUTPUT PASS MESSAGE
                        2688
                        2689  *       LD      IX,DYMEM4       RETURN ADDRESS
007.201  335 041        2690          DB      MI.LDXA,MI.LDXB
007.203  207 007        2691          DW      DYMEM4
                        2692
007.205  030 077        2693          JR      DYMSG
                        2694
000.001                 2695          IF      .RAM.
                        2697          ELSE
007.207  041 000 040    2698  DYMEM4  LXI     H,START         POINT BACK TO BEGINNING OF RAM
                        2699          ENDIF
007.212  176            2700  DYMEM5  MOV     A,M             READ CURRENT CONTENTS
007.213  270            2701          CMP     B               SEE IF CORRECT CONTENTS STILL REMAIN
007.214  302 307 000    2702          JNZ     DYMEM9          FAILURE, SEE IF AT END OF RAM
                        2703
007.217  074            2704          INR     A
007.220  167            2705          MOV     M,A             INCREMENT RAM
007.221  276            2706          CMP     M               SEE IF WRITE WAS SUCCESSFUL
007.222  302 307 000    2707          JNZ     DYMEM9
                        2708
007.225  043            2709          INX     H
007.226  175            2710          MOV     A,L             GET LSB AND TEST FOR REACHING END OF RAM
007.227  273            2711          CMP     E
007.230  040 360        2712          JR      NZ,DYMEM5       IF LSB NOT EQUAL
                        2713
007.232  174            2714          MOV     A,H             CHECK LSB
007.233  272            2715          CMP     D
007.234  040 354        2716          JR      NZ,DYMEM5
                        2717
                        2718  *       HAVE REACHED END OF MEMORY!
                        2719  *       OUTPUT LAST VALUE TESTED
                        2720  *
007.236  046 003        2721          MVI     H,3             OUTPUT 3 BACKSPACES
007.240  076 010        2722          MVI     A,A.BKS
                        2723
007.242                 2724  DYME5.5 EQU     *
                        2725
                        2726  *       LD      IY,DY5.53       RETURN ADDRESS
007.242  375 041        2727          DB      MI.LDYA,MI.LDYB
007.244  251 007        2728          DW      DY5.53
                        2729
007.246  303 143 003    2730          JMP     DYASC
                        2731
007.251  045            2732  DY5.53  DCR     H
007.252  040 366        2733          JR      NZ,DYME5.5
                        2734
```

```
007.254  004              2735            INR     B               SHOW NEXT PASS VALUE
007.255  170              2736            MOV     A,B             VALUE TESTED
                          2737
                          2738  *         LD      IX,DYMEM6       RETURN ADDRESS
007.256  335 041          2739            DB      MI.LDXA,MI.LDXB
007.260  273 000          2740            DW      DYMEM6
                          2741
007.262  303 160 003      2742            JMP     DYBYT
                          2743
                          2744
                          2745  **        !!THE DYNAMIC RAM TEST CONTINUES ELSEWARE!!      **
                          2746  *         !!AND THEN RETURNS TO HERE!!!!!!!!!!!!!!!!!!      **
                          2747
                          2748
007.265  041 000 000      2749  DY10.5    LXI     H,0             DELAY AND DING BELL AGAIN
007.270  006 002          2750            MVI     B,2             2 LOOPS
007.272  045              2751  DYMEM11   DCR     H
007.273  040 375          2752            JR      NZ,DYMEM11
                          2753
007.275  055              2754            DCR     L
007.276  040 372          2755            JR      NZ,DYMEM11
                          2756
007.300  005              2757            DCR     B
007.301  040 367          2758            JR      NZ,DYMEM11
                          2759
007.303  303 360 003      2760            JMP     DYMEM10         AGAIN


                          2762  **        DYMSG - DYNAMIC RAM TEST MESSAGE OUTPUT ROUTINE
                          2763  *
                          2764  *         ENTRY   (H,L) = MESSAGE ADDRESS
                          2765  *                 (IX) = RETURN ADDRESS
                          2766  *
                          2767  *         EXIT    TO (IX)
                          2768  *
                          2769  *         USES    A,H,L,F,IY
                          2770
                          2771
007.306  176              2772  DYMSG     MOV     A,M             GET MESSAGE BYTE
                          2773
                          2774  *         LD      IY,DYMSG.5      RETURN ADDRESS
007.307  375 041          2775            DB      MI.LDYA,MI.LDYB
007.311  316 007          2776            DW      DYMSG.5
                          2777
007.313  303 143 003      2778            JMP     DYASC           OUTPUT ASCII
                          2779
007.316  267              2780  DYMSG.5   ORA     A               SEE IF NULL TO END STRING
007.317  043              2781            INX     H               POINT TO NEXT CHARACTER
007.320  040 364          2782            JR      NZ,DYMSG        IF NOT DONE YET
                          2783
                          2784  *         JP      (IX)            RETURN TO CALLER
007.322  335 351          2785            DB      MI.JIXA,MI.JIXB
```

```
                                2787  **      MSG.RAM - RAM TEST MESSAGE
                                2788  *
                                2789
    007.324  033 105            2790  MSG.RAM DB      A.ESC,'E'
    007.326  104 171 156        2791          DB      'Dynamic RAM test'
    007.346  015 012 012        2792          DB      A.CR,A.LF,A.LF
    007.351  011 040 114        2793          DB      '       LWA = '
    007.361  000                2794          DB      0


                                2796  **      MSG.EQ - EQUALS MESSAGE
                                2797  *
                                2798
    007.362  040 075 040        2799  MSG.EQ  DB      ' = '
    007.365  000                2800          DB      0
                                2801
    007.366  107 101 103        2802          DB      'GAC.'
```

```
                        2805  **      ENTRY POINT FOR FLOPPY DISK ROTATIONAL SPEED TEST
                        2806  *
    000.001             2807          IF      .RAM.
                        2808          ELSE
    000.000             2809          ERRNZ   10000A-6-*      MUST BE SIX BYTES BEFORE END
                        2810          ENDIF
                        2811
    007.372  303 240 006 2812  ESPEED  JMP     SPEED


                        2814  **      ENTRY POINT FOR DYNAMIC MEMORY TEST
                        2815  *
    000.001             2816          IF      .RAM.
                        2817          ELSE
    000.000             2818          ERRNZ   10000A-3-*      MUST BE THREE BYTES BEFORE END
                        2819          ENDIF
                        2820
    007.375  303 116 007 2821  EDYMEM  JMP     DYMEM
                        2822
    000.001             2823          IF      .RAM.
                        2824          ELSE
    000.000             2825          ERRNZ   *-10000A        MUST NOT EXCEED 2K BYTES
                        2826          ENDIF
                        2827
```

```
                    2830  **      THE FOLLOWING ARE CONTROL CELLS AND FLAGS USED BY THE KEYSET
                    2831  *       MONITOR.
                    2832
    040.000         2833          ORG     40000A          8192
    040.000         2834  START   DS      2               DUMP STARTING ADDRESS
    040.002         2835  IOWRK   DS      2               IN OR OUT INSTRUCTION
    040.004         2836  PRSRAM  EQU     *               FOLLOWING CELLS INITIALIZED FROM ROM
    040.004         2837          DS      1               RET
                    2838
    040.005         2839  REGI    DS      1               INDEX OF REGISTER UNDER DISPLAY
    040.006         2840  DSPROT  DS      1               PERIOD FLAG BYTE
    040.007         2841  DSPMOD  DS      1               DISPLAY MODE
                    2842
    040.010         2843  .MFLAG  DS      1               USER FLAG OPTIONS
                    2844  *                               SEE *UO.XXX* BITS DESCRIBED AT FRONT
                    2845
    040.011         2846  CTLFLG  DS      1               FRONT PANEL CONTROL BITS
    040.012         2847  REFIND  DS      1               REFRESH INDEX (0 TO 7)
    000.007         2848  PRSL    EQU     *-PRSRAM        END OF AREA INITIALIZED FROM ROM
                    2849
    040.013         2850  FPLEDS  EQU     *               FRONT PANEL LED PATTERNS
    040.013         2851  ALEDS   DS      1               ADDR 0
    040.014         2852          DS      1               ADDR 1
    040.015         2853          DS      1               ADDR 2
                    2854
    040.016         2855          DS      1               ADDR 3
    040.017         2856          DS      1               ADDR 4
    040.020         2857          DS      1               ADDR 5
                    2858
    040.021         2859  DLEDS   DS      1               DATA 0
    040.022         2860          DS      1               DATA 1
    040.023         2861          DS      1               DATA 2
                    2862
    040.024         2863  ABUSS   DS      2               ADDRESS BUSS
    040.026         2864  RCCA    DS      1               RCC SAVE AREA
    040.027         2865  CRCSUM  DS      2               CRC-16 CHECKSUM
    040.031         2866  TPERRX  DS      2               TAPE ERROR EXIT ADDRESS
    040.033         2867  TICCNT  DS      2               CLOCK TIC COUNTER
                    2868
    040.035         2869  REGPTR  DS      2               REGISETR CONTENTS POINTER
                    2870
    040.037         2871  UIVEC   DS      0               USER INTERRUPT VECTORS
    040.037         2872          DS      3               JUMP TO CLOCK PROCESSOR
    040.042         2873          DS      3               JUMP TO SINGLE STEP PROCESSOR
    040.045         2874          DS      3               JUMP TO I/O 3
    040.050         2875          DS      3               JUMP TO I/O 4
    040.053         2876          DS      3               JUMP TO I/O 5
    040.056         2877          DS      3               JUMP TO I/O 6
    040.061         2878          DS      3               JUMP TO I/O 7
                    2879
                    2880  **      H88/H89 RAM USAGE BEYOND THAT OF H8MTRF
                    2881  *
    040.064         2882  NMIRET  DS      2
                    2883  *
    041.120         2884          ORG     41120A
    041.120         2885  PRIM    DS      1               PRIMARY DEVICE ADDR. PORT
```

```
041,121  2886  TMFG   DS  1      TIMER INTERRUPT FLAG, =1 FOR Z47, =0 FOR H17
041,122  2887  MYCNT  DS  1      COUNTER FOR TIMER INTERRUPT
041,123  2888  AUTOR  DS  1      AUTO BOOT FLAG
041,124  2889  STK    DS  2      STACK POINTER FOR RE-BOOT
         2890
040,066  2891        ORG 40066A
040,066  2892  DATA  DS  1      OUTPUT 362Q DATA SAVE AREA
040,067  2893        END
ASSEMBLY COMPLETE
 2893 STATEMENTS
 0 ERRORS DETECTED
14204 BYTES FREE
```

```
MTR89 - H89 MONITOR  $09.01.00.                                    XREF V1.1
CROSS REFERENCE TABLE                                              PAGE  73

$MOVE   030252      340E    1343
$ZERO   031212      342E    1401
.       000004      533S     732S    910S   1084S   1849S   1850
.MFLAG  040010      732      780     853     859    1208    1324    1409    2843L
.RAM.   000001       42E      44     472     492     500     511     523     544     554     564     578     588
                     600      619     646     674     696     707     727     740     761     847     868     947    1049
                    1060     1076    1270    1278    1477    1497    1515    1544    1765    1780    2640    2695    2807
                    2816     2823
.UIVEC  040037      344E    1373    1410    1421
A.BEL   000007      156E     797     895    1118    1751    2100    2128    2154    2197    2360
A.BKS   000010      157E    2722
A.CR    000015      159E     936    1004    1025    1108    1564    1643    2071    2097    2125    2145    2247
                    2305    2407    2598    2792
A.DEL   000177      161E    1692    2021
A.ESC   000033      160E    2597    2606    2608    2614    2616    2790
A.LF    000012      158E     936     936    1028    1643    2249    2309    2407    2597    2598    2598    2792
                    2792
A.STX   000002      155E
A.SYN   000026      154E
ABUSS   040024      952    2863L
AC.DLY  000156      405E
AID.UNI 041061      343E    1145    1171    1233
ALARM   002136      863    1275E
ALARMB  002166     1276    1304L
ALEDS   040013     2851L
AS.ODD  000100      227E    1189
AS.IDD  000040      228E
AS.SIA  000020      229E
AS.SLW  000003      230E
ATB     004243     1258    1973L
AUTOB   041123     1256    1836    2888L
AUTOBO  001212     1040L   1976
B170    003040     1436    1439    1460E
B174    003012     1437    1440L
BH17    003030     1444    1453L
BOOT    004256      921    1988L
BOOTO   001336     1042    1109    1137L
BOOT2   002335     1411L   1418
BOOT5   001341     1113    1140L
BOOT6   001353     1138    1145L
BOOTA   037132      339E    1341
BOOTAL  000130      338E    1340
BOOTX   004274     1977    1993L
BR19.2  004103     1845L
BR96    004101     1844L
BRTAB   004101     1797    1842E   1850
BSEC    001325     1129E
BSMSG   001041      927L    1130
C.128   000200      259E
C.256   001000      258E
C.DSYN  000375      301E
CB.CLI  000100      167E     202     596     858    1949
CB.MTL  000040      166E     715     777     858    1086
CB.SPK  000200      168E     596     858    1284
CB.SSI  000020      165E     596     715     858    1068    1082    1949
CHAT2   002130     1255    1259L
CKAUTO  002113      879    1252E
CLK4    000270      784     807E
```

```
CLOCK    000201      510      513     766L
CLOCK17  034031      345E    1372    2045
COM      006027     1185     2335E   2424
COM1     006031     2324     2337L
CRCSUM   040027     2865L
CS.UNR   000200      222E
CS.WPD   000100      223E
CTLFLG   040011      533      770     775     780     859    1067    1070    1084    1289    1857    2350    2846L
CUI1     000165      733L     796     808
D.CON    040110      316L    1342
D.DAT    000171      212E
D.DECNT  040264      352E    1232
D.RAM    040240      319L    1399
D.RAML   000037      341E    1400
D.SDP    040206      348E    1349
D.STA    000170      211E     212    1423
D.VEC    040130      318L
DAT      006023     1187     2322E   2426    2428
DATA     040066     1837     1957    2892L
DC.BOOT  000000      234E
DC.CPY   000013      245E
DC.FRM0  000014      246E
DC.FRM1  000015      247E
DC.FRM2  000016      248E
DC.FRM3  000017      249E
DC.LSC   000003      237E
DC.RAD   000004      238E
DC.RAS   000002      236E    1184
DC.REA   000005      239E
DC.REAB  000007      241E    2423
DC.RST   000001      235E
DC.WRD   000011      243E
DC.WRDB  000012      244E
DC.WRI   000006      240E
DC.WRIB  000010      242E
DEV2     003025     1446L    1464
DEVICE   002301     1041     1106    1393E
DF.DI    000040      277E
DF.DS0   000002      273E
DF.DS1   000004      274E
DF.DS2   000010      275E
DF.HD    000001      267E
DF.MO    000020      276E
DF.SD    000010      270E
DF.ST    000100      278E
DF.TO    000002      268E
DF.WG    000001      272E
DF.WP    000004      269E
DF.WR    000200      279E
DLEDS    040021     2859L
DLY      000053      583L
DM.MR    000000      172E
DM.MW    000001      173E
DM.RR    000002      174E
DM.RW    000003      175E
DOD      003122     1549L
DP.DC    000177      265E    1325    1402
DS.HOLE  000001     2532E    2557    2563
```

DSPMOD	040007	2841L						
DSPROT	040006	2840L						
DY10.5	007265	1755	2749L					
DY3.3	007153	2662	2666L					
DY3.5	007163	2670	2674L					
DY3.7	007173	2678	2682L					
DY5.53	007251	2728	2732L					
DY9.3	003315	830	1717L					
DY9.4	003326	1722	1726L					
DY9.5	003335	1730	1734L					
DY9.8	003350	1739	1743L					
DYASC	003143	1577E	1611	1624	1634	1757	2730	2778
DYASC1	003144	1580L	1582					
DYBYT	003160	1597L	1724	1732	1749	2672	2680	2742
DYBYT.2	003202	1609	1613L					
DYBYT.4	003221	1622	1626L					
DYBYT.6	003235	1632	1636E					
DYME5.5	007242	2724E	2733					
DYMEM	007116	1830	2635L	2821				
DYMEM1	007122	2644L						
DYMEM10	003360	1747	1751L	2760				
DYMEM11	007272	2751L	2752	2755	2758			
DYMEM2	007127	2647L	2653					
DYMEM3	007140	2650	2656L					
DYMEM4	007207	821	2691	2698L				
DYMEM5	007212	2700L	2712	2716				
DYMEM6	000273	815L	2740					
DYMEM7	000276	816L	819					
DYMEM9	000307	825L	2702	2707				
DYMSG	007306	832	1741	2664	2693	2772L	2782	
DYMSG.5	007316	2776	2780L					
EDYMEM	007375	2821L						
EIXIT	034027	350E	1413	1415				
ERPTCNT	000012	351E	1231					
ERRMSG	001045	929L	1322					
ERROR	000322	654	801	852E	1326	1693		
ESPEED	007372	2812L						
FPLEDS	040013	2850E						
GO	001222	1030	1054L					
GO.	000063	596L	1054	1997				
GO88	001146	912	1015L					
GO88.1	001177	1018	1027L					
H17	002207	1339E	1453					
H17A	002237	1357L	1359					
H88.CTL	000362	130E	1791	1959	2636			
H88.SW	000362	134E	1253	1430	1798	1828		
H88B.CK	000002	131E	1790					
H88B.SS	000001	132E						
H88S.0	000014	140E	1462					
H88S.4	000003	142E	1443					
H88S.AT	000200	135E	1254					
H88S.BR	000100	136E	1799					
H88S.DV	000020	138E	1432					
H88S.M	000040	137E	1829					
HORN	002140	1283L	2362					
HRN0	002143	585	1286L					
HRN2	002160	1297L	1298					
HRNX	006045	1300	2350L					

```
IN.      006170    966    1205   2338   2480E
IN.1     006174   2463    2483L
IN1.     006156    969    2459E
INIT     000073    490     494    624L   628
INIT0    000000   488L
INIT0.0  000003   489L    1838
INITOX   004000    488    1790L
INITOX1  004050  1821L    1822   1824
INIT1    000107   637L     642
INIT2    000117   644L
INT1     000010   503E
INT2     000020   526E
INT3     000030   549L
INT4     000040   559L
INT5     000050   569L
INT6     000060   593L
INT7     000070   605L
INTXIT   000172   745L     778   1072
IOA      003062   1005    1026   1502L  2072
IOA1     005176   1502    2170L
IOA2     005204  2174L    2192   2201
IOA3     005242   2176    2179   2194L
IOA4     005260   2195    2205L
IOB      003066  1520L
IOB1     003070  1521L    1528   1560   1567
IOB1.5   003126   1539    1557L
IOB2     003135   1526    1564L
IOC      005301   2081    2116   2230L
IOWRK    040002   2070    2541   2579   2581   2835L
IP.DS    000177  2528E    2556   2562
IP.PAD   000360   124E
IP.TPC   000371   147E
IP.TPD   000370   149E
IROC     005150    994     998   1017   2067   2144L  2156
IROC1    005166   2149    2154L
LRA      003047  1482L
LRA.     003052    789     988   1022   1483L  1975   1991
MI.ANI   000346   184E
MI.EXAF  000010   191E    1114   1116   1122   1394   1433   1435   1438   1579   1585   2027   2040
                  2043
MI.HLT   000166   179E     795
MI.IN    000333   181E    1929   1938
MI.JIXA  000335   192E    1638   2785
MI.JIXB  000351   193E    1638   2785
MI.JIYA  000375   194E    1588
MI.JIYB  000351   195E    1588
MI.JMP   000303   186E    1411
MI.LDA   000072   183E
MI.LDXA  000335   187E     829   1721   1729   1738   1746   2661   2669   2677   2690   2739
MI.LDXB  000041   188E     829   1721   1729   1738   1746   2661   2669   2677   2690   2739
MI.LDYA  000375   189E    1608   1621   1631   1754   2727   2775
MI.LDYB  000041   190E    1608   1621   1631   1754   2727   2775
MI.LXID  000021   185E
MI.OUT   000323   182E    1926   1945
MI.RET   000311   180E    1778
MSG.BT   006234   1988    2506L
MSG.EQ   007362   1735    2799L
MSG.ERR  001047    826     936L
```

MTR89 - H89 MONITOR #09.01.00. XREF V1.1
CROSS REFERENCE TABLE PAGE 77

MSG.GO	006165	1015	2470L						
MSG.HSS	007100	2584	2614L						
MSG.PAS	003237	1643L	2687						
MSG.PC	006214	985	2500L						
MSG.PR	006111	1259	2407L						
MSG.RAM	007324	2658	2790L						
MSG.SPD	006371	2538	2597L						
MSG.SUB	006201	2065	2494L						
MSG.WRK	007062	2578	2606L						
MTR	000344	873E	1087						
MTR.15	000354	880L	1260						
MTR.2	000357	882L	897						
MTR.3	000371	886L	893						
MTR.4	001014	887	899L						
MTR1	000345	876E	877						
MTRA	001025	884	909E	923					
MTRAL	00000A	885	923E						
MYCNT	041122	1404	2032	2887L					
NB7	001316	1115	1122L						
NBOOT	001261	1105L	1992						
NBOOTO	001262	1106L	1133						
NMI	004116	705	1898L						
NMI0.5	004154	1916	1919	1924L					
NMI1	004173	1911	1936L						
NMI1.5	004206	1939	1945L						
NMI2	004212	1948L							
NMI2.2	004225	1955	1957L						
NMI2.5	004236	1922	1927	1930	1946	1963L			
NMI3	004237	1934	1943	1965L					
NMIENT	000146	705L							
NMIRET	040064	1899	1906	2882L					
NODEV	002171	1148	1207	1321E	1368	2022	2036		
NOISE	006053	1304	2360L						
ONDRO	000022	2536E	2542						
OP.CTL	000360	125E	771	1069	1083				
OP.DC	000177	2527E	2543						
OP.DIG	000360	126E							
OP.SEG	000361	127E							
OP.TPC	000371	148E	660						
OP.TPD	000370	150E							
OUT	006063	1180	2372E						
OUT.1	006070	2376L	2449						
OUT1	006146	2342	2444E						
PCA	001103	918	985L						
PCA1	001137	995	1003L						
PIN	001067	965E	968	1188	2430				
PRIM	041120	1424	1441	2375	2447	2461	2482	2885L	
PRSL	000007	489	2848E						
PRSRAM	040004	489	2836E	2848					
PRSROM	003371	1771E	1834						
R.ABORT	033366	346E	1363						
R.READ	034077	347E	1367						
R.SDP	002071	1230E	1348	1356					
R.SDP1	002110	1236	1238L						
RCC	003262	882	1107	1521	1684E	2144	2174	2230	2304
RCC1	003262	1686L	1688						
RCC2	003270	1690L							
RCCA	040026	2864L							

```
RCK      003260  1663E
RD1      006121  2423L
RD2      006136  2430L  2433
RDBLCK   006121  1198   1201  2422E
REFIND   040012  2847L
REGI     040005  1482   2839L
REGPTR   040035   719    861  1485  2869L
RESET    002003  1179L
RETRY    002063  1222L  2041
ROMBOOT  030000   311E
S.DON    000040   215E  2336
S.DTR    000200   216E   967  2323
S.ERR    000001   214E  1206
S.GRT    024000   308E
S.GRT1   025000   309E
S.INT    040343   321L
S.SOVR   041146   323L   325
S.VAL    040277   320L
SAE      001063   952L
SAVALL   000132   508    531   679L
SAVALLR  000151   709    712E  1859
SAVALLX  004105   694   1854E
SC.ACE   000350   404E  1580  1586  1686  1690  1707  1712  1796  1808  1811  1813  1815
                  2015  2019
SC.UART  000372   364E
SDF3     036073   349E  1238
SECOND   003007  1434   1438L
SECSCR   026000   310E
SINCR    004000   630E   632   633
SPEED    006240  2538L  2812
SPEED1   006257  2544L  2588
SPEED2   006275  2552   2555L
SPEED3   006300  2556L  2558  2569
SPEED4   006307  2562L  2564
SPEED5   006357  2582   2585L
SST1     001235   597   1070L
SSTEP    001225  1065E
STACK    042200   327E
STACKL   001032   325E
START    040000   633   2644  2698  2834L
START1   001265  1107L  1120
STK      041124  1167   1222  2889L
STPRTN   001244   534   1081E
SUBM     004370   915   2065L
SUBM1    005013  2075L  2088  2095
SUBM2    005027  2081L  2102
SUBM3    005042  2087L  2120
SUBM4    005046  2085   2090L
SUBM5    005053  2093L  2123
SUBM6    005062  2091   2097L
SUBM7    005075  2082   2104L
SUBM8    005077  2106L  2117
SUBM9    005115  2116L  2130
SYDD     040130   317E
TICCNT   040033   766    768  1293  1403  2028  2544  2573  2867L
TMFG     041121  1407   1455  2024  2032  2886L
TMOUT    004302  1420   2014E
TMOUT1   004345  2037L  2039
```

TMOUT2	004361	2030	2038	2043L									
TMOUT3	004363	2042	2045L										
TMOUT4	004321	2017	2024L										
TOA	005313	997	2075	2247L									
TOA.	005325	2252L	2587										
TOB	005343	2077	2253	2255	2270L								
TOB1	005353	2277L	2285										
TPERRX	040031	2866L											
TYPMSG	006100	880	986	1016	1131	1323	1989	2066	2394L	2400	2539	2585	
UC.2SB	000004	430E											
UC.5BW	000000	426E											
UC.6BW	000001	427E											
UC.7BW	000002	428E											
UC.8BW	000003	429E	1812										
UC.BI	000020	449E											
UC.CTS	000020	458E											
UC.DCS	000001	454E											
UC.DDR	000002	455E											
UC.DLA	000200	435E	1795										
UC.DR	000001	445E	1687	2016									
UC.DRL	000010	457E											
UC.DSR	000040	459E											
UC.DTR	000001	438E											
UC.EDA	000001	416E											
UC.EPS	000020	432E											
UC.FE	000010	448E											
UC.IID	000006	423E											
UC.IIP	000001	422E											
UC.LOO	000020	442E											
UC.MSI	000010	419E											
UC.OR	000002	446E											
UC.OU1	000004	440E											
UC.OU2	000010	441E											
UC.PE	000004	447E											
UC.PEN	000010	431E											
UC.RI	000100	460E											
UC.RLS	000200	461E											
UC.RSI	000004	418E											
UC.RTS	000002	439E											
UC.SB	000100	434E											
UC.SKP	000040	433E											
UC.TER	000004	456E											
UC.THE	000040	450E	1581	1708									
UC.TRE	000002	417E											
UC.TSE	000100	451E											
UCI.ER	000020	386E											
UCI.IE	000002	388E											
UCI.IR	000100	384E											
UCI.RE	000004	387E											
UCI.RO	000040	385E											
UCI.TE	000001	389E											
UDR	000000	361E											
UF.FCT	000100	294E											
UF.RDA	000001	291E											
UF.ROR	000002	292E											
UF.RPE	000004	293E											
UF.TBM	000200	295E											
UIVEC	040037	549	559	569	593	605	736	1088	2871L				

```
UMI.16X 000002  379E  659
UMI.1B  000100  369E  659
UMI.1X  000001  378E
UMI.2B  000300  371E
UMI.64X 000003  380E
UMI.HB  000200  370E
UMI.L5  000000  374E
UMI.L6  000004  375E
UMI.L7  000010  376E
UMI.L8  000014  377E  659
UMI.PA  000020  373E
UMI.PE  000040  372E
UNT.0   000000  253E
UNT.1   000040  254E
UNT.2   000100  255E
UNT.3   000140  256E
UO.CLK  000001  204E  734  1408
UO.DDU  000002  203E  855
UO.HLT  000200  201E  782
UO.NFR  000100  202E  855
UP.DP   000174  285E  1440
UP.FC   000175  286E
UP.SC   000176  288E
UP.SR   000176  289E
UP.ST   000175  287E
UR.DLL  000000  411E  1811
UR.DLM  000001  413E  1808
UR.IER  000001  415E  1815
UR.IIR  000002  421E
UR.LCR  000003  425E  1796  1813
UR.LSR  000005  444E  1580  1686  1707  2015
UR.MCR  000004  437E
UR.MSR  000006  453E
UR.RBR  000000  407E  1690  2019
UR.THR  000000  409E  1586  1712
USERFWA 042200  328E  1196  1209  1364  1374
USR     000001  362E
USR.FE  000040  393E
USR.OE  000020  394E
USR.PE  000010  395E
USR.RXR 000002  397E
USR.TXE 000004  396E
USR.TXR 000001  398E
W.RES   000002  218E  1179
WCC     003302  798   800   896   899   1027  1029  1119  1140  1532  1706L 2079  2093
                2101  2106  2129  2155  2181  2198  2205  2248  2250  2258  2282  2289  2308
                2310  2361  2398
WCC1    003303  1707L 1709
WCR     006003  1143  2304L 2306
WHD     036235  335E  1358
WNH     036271  336E  1357
WRONG   001307  1111  1117E 1125
WTDON1  006032  2338L 2340
Z47     001364  1164E 1425  1449
Z47A    001370  1169E 1224
```

22314 BYTES FREE

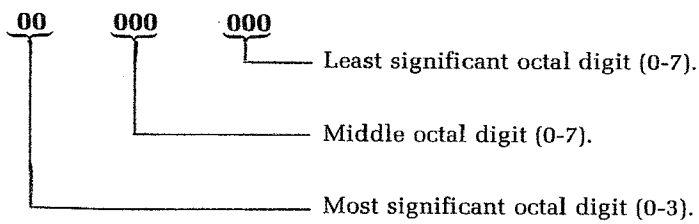
APPENDIX B

OCTAL DEFINITIONS

Binary numbers are converted to octal format for display. 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 byte is displayed as two-and-one-half octal digits. The octal numbers lie in the range of 000 to 377 for binary numbers in the range 00000000 to 11111111, as shown below.



NOTE: As 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 the user should inadvertently enter the octal digits 4 to 7 into the most significant digit, the most significant bit is lost. Losing this bit converts 4 through 7 into the digits 0 through 3 respectively.

Also note that 16-bit numbers, such as memory addresses and certain register contents, are displayed as two eight-bit numbers. Therefore, the representation of 16-bit numbers is made up of **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**, and is used consistantly for displays of 16-bit numbers.

Split-octal must 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 example shows true octal representation of a 16-bit binary number. True octal representation is never used in standard Zenith Data Systems software. Occassionally you will see split-octal numbers printed with a decimal point separating the upper and lower bytes. For example:

```
377.377
Hi Byte  Lo Byte
```

Note that 001.000 follows 000.377.