SOFTWARE REFERENCE MANUAL

H88/WH89 COMPUTER

MONITOR

MTR-88

Copyright © 1979
Heath Company
All Rights Reserved

HEATH COMPANY
BENTON HARBOR, MICHIGAN 49022

595-2349
Printed in the United States of America

2 Monitor

Table of Contents

INTRODUCTION

This manual describes the functions and operation of the H88 Monitor Program MTR-88 that is contained in a read-only memory (ROM) that is in your H88/H89. Some of the major features of MTR-88 are:

Memory contents display and alteration.
Program execution control.
Cassette load and dump routines.
Floppy diskette boot-strap routine.

In addition, MTR-88 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.

Since the H89 is an expansion of the H88, all the features of MTR-88 are available on an H89 also.

THEORY OF OPERATION

This section supplements the information in the "Operations" and "Circuit Description" sections of your H88 Operations Manual. In order to use all of the features of MTR-88, it is necessary to understand the Z80 operation codes and the circuit of your H88. This section gives you details of the operation of MTR-88. The listing of MTR-88 is given in Appendix A and a sophisticated example is given in Appendix B.

Power Up and Reset

MTR-88 initializes the H88 whenever you power-up or RESET. To power-up, use the switch on the back of the H88. To RESET, simultaneously press the RESET key and the right-hand SHIFT key on the keyboard. Both power-up and RESET cause a level zero interrupt (highest priority). MTR-88 sounds the audio alert and resets to its normal state. During the initialization procedure, MTR-88 determines the high limit of continuous RAM in your H88. Once this high limit has been determined, the Z80's stack pointer is set to this value. Then MTR-88 enters a loop waiting for you to enter a command.

Clock Interrupts

The Clock Interrupt is a crucial element in the operation of the H88. It is a level one interrupt and is generated on the H88 CPU board every 2 ms (millisecond). MTR-88 maintains "TICCNT" which counts up one every 2 ms. See the listing in Appendix A for the location of TICCNT.

Note that MTR-88 uses interrupts, so you should not disable interrupts for a long period of time. MTR-88 also requires a stack pointer at the top of memory with at least 80 bytes.

General Operations

When you RESET or power-up your H88 or H89, MTR-88 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-88 will either accept it or give a beep, indicating an error.

If the letter you enter is the first letter of one of MTR-88's commands, it will display the remaining letters of the word and start the appropriate program in MTR-88. If the letter is not the start of a command, MTR-88 will sound the horn and re-display the "H:".

The DELETE key will kill a partially entered line and cause MTR-88 to return to the "H:" prompt. You can use this to correct typing errors.

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

The following is a list of the acceptable MTR-88 commands. You type the first letter of the command, and MTR-88 will supply the remainder of the word. You have to press the (carriage) RETURN key before MTR-88 will respond.

TABLE OF MTR-88 COMMANDS

Boot	— Boot HDOS from a diskette
Dump	— Dump a program to cassette
Go	— Start a program
Load	— Load a program from cassette
Program Counter	— Set an address in the PC
Substitute	— Inspect or change memory

These commands are described more in the remainder of this manual.

DISPLAYING AND ALTERING MEMORY

One of the major features of MTR-88 is its ability to examine the contents of any H88 memory location and to modify the contents of that location if it is in RAM. This feature is described now.

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-88 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 C for the definitions of octal and split-octal.

MTR-88 will respond by re-displaying the address. Following the address, MTR-88 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-88 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-88 to advance to the next memory location and display it by pressing the Space Bar. Second, you can cause MTR-88 to retreat to the previous memory location and display it by pressing the minus key, "−". Finally, you can cause MTR-88 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
           Δ                          you
002147 011                           computer
           Δ                          you
002150 040                           computer
           -                          you
002147 011                           computer
           CR                         you
  H:                                  computer
     S                                you
      ubstitute                       computer
                 40100 CR             you
040100 xxx                           computer
           123 Δ                      you
040101 xxx                           computer
           -                          you
040100 123                           computer
           CR                         you
  H:                                  computer
```

PROGRAM EXECUTION CONTROL

MTR-88 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-88 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 040.100, you can start it with:

```
H:  Go 40100 CR
```

MTR-88 allows another method of starting programs. MTR-88 maintains in its working memory a value for the Program Counter. If you enter "G" and then a RETURN after MTR-88 prints "o", MTR-88 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-88 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 number after "P", but simply press RETURN, then MTR-88 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-88 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-88, display an "H", and then advance to a new line and display "H:". You can now inspect or change memory using the "Substitute" command.

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-88 and your program will "fight" for keyboard input! Your program will always see every character because it gets them by an interrupt. MTR-88 is continually testing if a character is available, and it will never see some of the characters that you enter. This can become very confusing, particularly in the debug program.

LOAD/DUMP ROUTINES

MTR-88 contains routines that let you load and dump memory contents from or to a cassette tape. These "boot strap" routines allow you to quickly and easily use your computer without entering a complex program by hand. These routines contain sophisticated error checking techniques that let you know if a problem has occurred.

Loading From Tape

To load from a cassette tape, ready the tape reader with the tape to be loaded. Then RESET MTR-88 and type "L". MTR-88 will respond "oad". When you enter a RETURN, MTR-88 will load the tape.

No change will be seen on the screen until MTR-88 finds the first file on the tape. The load routine places the entry point address into the H88's Program Counter and then continues loading. As data is loaded into memory, the address at which it is placed is shown on the screen. You can watch it change and see when the load is complete.

If the load is successful, MTR-88 will sound the alarm once to alert you. If a loading error occurs, MTR-88 will sound the alarm repeatedly. RESET the H88 and try loading the tape again.

You may RESET during loading, but the load will be invalid. To get a good load, you will have to start the procedure over.

Dumping a Tape

Before MTR-88 can dump a tape, it needs to have the first and last address of the section of memory that you want to have dumped. It also needs to have the starting address (which need not be the first address) so that when the program is loaded, the PC can be set, and "Go" can be used to start the program.

First, place the address of the program's starting address in the PC as described earlier. Later, when you load the tape, this value will be placed back in the PC so you can enter "Go" and start the program.

Next, give a "Dump" command by typing "D". MTR-88 will respond with "ump". You should then type in the first address you want saved on the tape, followed by the minus or dash character (-), and finally the last address you want saved. When you press the RETURN, MTR-88 will start recording the data on the tape.

The detail steps to DUMP to a tape are:

1. Ready a tape in the recorder and press the record button.
2. Enter the starting address of the program in the PC.
3. Enter the dump first and last addresses as:

   Dump 40100-43264

4. Press RETURN.

If you give a starting address, you must give an ending address separated by a dash. However, if you do not give a starting address, MTR-88 will use the starting and finishing address that were last used for a Load or a Dump. This is described next in Copying a Tape.

Copying a Tape

To copy a tape, simply load the tape as described in "Loading From Tape". Then ready the dump tape to receive the copy. Finally, type "D". When MTR-88 responds "ump", simply press RETURN. MTR-88 will remember the first, last, and starting addresses from the load.

You can modify the program before you dump it, but if the first or last addresses are different, your will have to enter them when you dump. The same is true for the PC.

Tape Errors

MTR-88 detects two types of tape errors: record errors and checksum errors. In either case, when an error is detected, the tape transport stops and an error number is printed on the screen. The error numbers are 001 for a checksum error and 002 for a record error. The alarm is repeatedly sounded when an error is detected. RESET the H88 to stop the alarm and return to MTR-88's command mode.

Record Errors

The following are typical causes of record errors.

- Attempting to load a file which is not a memory image. For example, loading an editor text file or a BASIC program file.
- Attempting to start a load in the middle of a file.
- A read error that causes a portion of the data to be lost, and records are not read in the proper sequence.

Checksum Errors

A checksum error occurs when the Cyclical Redundancy Check (CRC) checksum that follows a record does not match the CRC calculated by MTR-88. This error means that the record is either recorded incorrectly or the load was faulty. In either case, the load should be tried again. If repeated loads result in repeated failures, the tape is probably defective.

ADVANCED CONTROL

One of the advanced features of MTR-88 is its provisions allowing sophisticated users to augment or replace MTR-88'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 sample program in Appendix B shows how to use several of MTR-88's advanced features.

The following discussion refers to symbols and locations in MTR-88. In order to make the most of this information, you should refer to the listing of MTR-88 that is in Appendix A. Note that at the end of the listing the definitions of RAM locations from 40.000 to 40.077 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-88 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 H88 interrupts cause control to be transferred into the lowest 64 bytes of memory. Since MTR-88 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-88 through the UIVECs 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-88 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 H88. MTR-88 does not process these interrupts, but simply passes them on to a program in RAM by jumping to the appropriate UIVEC.

All Heath software (except MTR-88) uses interrupt 3 for input and output to and from the keyboard and screen. These programs set UIVEC themselves. If you want to use interrupts, your program has to place the appropriate jump in the appropriate UIVEC. See the sample program in Appendix B.

Clock Interrupts

The level one interrupt is generated by hardware in your H88 every 2 ms. MTR-88 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-88 will then pass each clock interrupt to your routine when it finishes its own processing. This is done in the example in Appendix B.

Single Instructions and Breakpoint Interrupts

Level two interrupts are generated by the single-instruction hardware contained in the H88. When a single-instruction interrupt occurs, MTR-88 processes it, and jumps to the location specified by the second UIVEC. This interrupt has no effect on MTR-88.

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. These features are used by the DBUG programs supplied by Heath.

FLOPPY BOOT

MTR-88 contains the code necessary to boot-up HDOS from a floppy disk. If you enter "B" after the "H:" prompt, then MTR-88 will respond "oot". When you then press RETURN, MTR-88 will jump to location 30.000 which is the entry point for the HDOS boot-up routine.

Unless you have the floppy disk controller board installed in your H88, there will be no ROM at 30.000, and the results of the "B" command are unpredictable. If you perform a "B" command, and do not have a floppy interface card, you should RESET your H88 to put it back in a known state.

APPENDIX A

MTR-88 LISTING

This appendix contains a listing of MTR-88. MTR-88 resides in the low 2K (2048) bytes of the H88 or H89 computer's memory. It contains all the control for primitive keyboard input and screen output as well as cassette tape load and dump facilities. MTR-88 needs RAM locations available in locations 40.000 through 40.077, 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-88. Just before this cross reference listing is the definition of RAM locations in 40.000 through 40.077.

Note that most of the PAM-8 entry points are preserved in MTR-88. (PAM-8 is the equivalent of MTR-88 on the H8 computer.) This was done to allow compatibility between H8 and H88 programs. Of course, H8 front panel routines will not operate, but they will return properly.

Because PAM-8 entry points have been preserved, the MTR-88 code has to jump around in a somewhat arbitrary manner. Also, the Memory Test and Floppy Disk Rotational Speed Test routines are scattered throughout memory. The listing of these two routines are not shown. The Memory Test entry point is 7.375 and the Floppy Speed Test entry point is 7.372.

```
                    4  ***     MTR88 - H88 MONITOR                    ISSUE 09.00.00
                    5  *
                    6  *       MTR88 IS AN ADAPTATION OF PAM/8 ORIGINALLY WRITTEN FOR THE
                    7  *       HEATH H8 COMPUTER BY J. G. LETWIN IN 1976 AND MODIFIED BY
                    8  *       R. N. BORCHARDT IN 1979 FOR USE IN THE HEATH H88/H89
                    9  *       COMPUTERS.
                   10  *
                   11  *       MTR88 PROVIDES COMPATABILITY WITH PAM/8 SUCH THAT ALL ROUTINES
                   12  *       HAVE RETAINED PREVIOUSLY DESCRIBED ENTRY POINTS AND ENTRY AND
                   13  *       EXIT CONDITIONS.  ROUTINES WHICH ARE NOT APPLICABLE SUCH AS
                   14  *       THOSE PERTAINING TO THE FRONT PANEL DISPLAY HAVE BEEN DELETED.
                   15  *
                   16  *
                   17  *       COPYRIGHT  05/1976, WINTEK CORPORATION,
                   18  *                           902 N. 9TH ST.
                   19  *                           LAFAYETTE, IND.
                   20  *
                   21  *       COPYRIGHT  01/1979, HEATH COMPANY
                   22  *                           BENTON HARBOR, MI.
                   23  *
                   24
                   25
 030.000           26  ROMID   EQU     30000A          HDOS BOOT ROM ADDRESS
                   27
                   28


                   30  ***     MTR88 - H88/H89 MONITOR.
                   31  *
                   32  *       THIS PROGRAM RESIDES (IN ROM) IN THE LOW 2048 BYTES OF THE HEATH
                   33  *       H88/H89 COMPUTERS.


                   35  ***     INTERRUPTS.
                   36  *
                   37  *       MTR88 IS THE PRIMARY PROCESSOR FOR ALL INTERRUPTS.
                   38  *       THEY ARE PROCESSED AS FOLLOWS:
                   39  *
                   40  *       RST     USE
                   41  *
                   42  *       0       MASTER CLEAR. (NEVER USED FOR I/O OR RST)
                   43  *
                   44  *       1       CLOCK INTERRUPT. NORMALLY TAKEN BY MTR88,
                   45  *               SETTING BIT *U0.CLK* IN BYTE *.MFLAG* ALLOWS
                   46  *               USER PROCESSING (VIA A JUMP THROUGH *UIVEC*).
                   47  *               UPON ENTRY OF THE USER ROUTINE, THE STACK
                   48  *               CONTAINS:
                   49  *               (STACK+0)  = RETURN ADDRESS (TO MTR88)
                   50  *               (STACK+2)  = (STACKPTR+14)
                   51  *               (STACK+4)  = (AF)
                   52  *               (STACK+6)  = (BC)
                   53  *               (STACK+8)  = (DE)
                   54  *               (STACK+10) = (HL)
```

```
                *               (STACK+12) = (PC)
                *               THE USER'S ROUTINE SHOULD RETURN TO MTR88 VIA
                *               A *RET* WITHOUT ENABLING INTERRUPTS.
                *
                *       2       SINGLE STEP INTERRUPTS RECEIVED WHEN IN
                *               USER MODE CAUSES A JUMP THROUGH *UIVEC*+3.
                *               STACK UPON USER ROUTINE ENTRY:
                *               (STACK+0)  = (STACKPTR+12)
                *               (STACK+2)  = (AF)
                *               (STACK+4)  = (BC)
                *               (STACK+6)  = (DE)
                *               (STACK+8)  = (HL)
                *               (STACK+10) = (PC)
                *               THE USER'S ROUTINE SHOULD HANDLE IT'S OWN RETURN
                *               FROM THE INTERRUPT.  THAT IS, *EI* FOLLOWED BY *RET*
                *
                *
                *       THE FOLLOWING INTERRUPTS ARE VECTORED DIRECTLY THROUGH *UIVEC*.
                *       THE USER ROUTINE MUST HAVE SETUP A JUMP IN *UIVEC* BEFORE ANY
                *       OF THESE INTERRUPTS MAY OCCUR.  RETURN IS VIA *EI* AND THEN *RET*
                *
                *       3       I/O 3. CAUSES A DIRECT JUMP THROUGH *UIVEC*+6
                *
                *       4       I/O 4. CAUSES A DIRECT JUMP THROUGH *UIVEC*+9
                *
                *       5       I/O 5. CAUSES A DIRECT JUMP THROUGH *UIVEC*+12
                *
                *       6       I/O 6. CAUSES A DIRECT JUMP THROUGH *UIVEC*+15
                *
                *       7       I/O 7. CAUSES A DIRECT JUMP THROUGH *UIVEC*+18
```

```
                        87  **      ASSEMBLY CONSTANTS

                        89  **      IO PORTS
                        90
                        91  ***     ALL REFERENCES TO THE H8 FRONT PANEL PORTS ARE TRAPPED BY THE
                        92  *       Z80 NMI OF THE H88/H89.  OP.CTL WILL STILL PERFORM AS IN AN H8
                        93  *       IN RESPECT TO THE CLOCK AND SINGLE STEP CONTROL.  FOR MORE
                        94  *       INFORMATION SEE THE NMI ROUTINE.
                        95
000.360                 96  IP.PAD  EQU     360Q            PAD INPUT PORT
000.360                 97  OP.CTL  EQU     360Q            CONTROL OUTPUT PORT
000.360                 98  OP.DIG  EQU     360Q            DIGIT SELECT OUTPUT PORT
000.361                 99  OP.SEG  EQU     361Q            SEGMENT SELECT OUTPUT PORT
                       100
                       101  *       H88/H89 CONTROL PORT
000.362                102  H88.CTL EQU     362Q            H88/H89 PORT FOR CLOCK AND SINGLE STEP
000.002                103  H88B.CK EQU     00000010B       2MS CLOCK ENABLE/DISABLE
000.001                104  H88B.SS EQU     00000001B       SINGLE STEP ENABLE/DISABLE
                       105
000.362                106  H88.SW  EQU     362Q            8 POSITION DIP SWITCH
000.300                107  H88S.BR EQU     11000000B       BAUD RATE SWITCHES
000.040                108  H88S.M  EQU     00100000B       MEMORY TEST/NORMAL OPERATION SWITCH


                       110  **      CASSETTE PORTS
                       111
000.371                112  IP.TPC  EQU     371Q            TAPE CONTROL IN
000.371                113  OP.TPC  EQU     371Q            TAPE CONTROL OUT
000.370                114  IP.TPD  EQU     370Q            TAPE DATA IN
000.370                115  OP.TPD  EQU     370Q            TAPE DATA OUT


                       117  **      ASCII CHARACTERS.
                       118
000.026                119  A.SYN   EQU     026Q            SYNC CHARACTER
000.002                120  A.STX   EQU     002Q            STX CHARACTER
000.007                121  A.BEL   EQU     007Q            BELL CHARACTER
000.010                122  A.BKS   EQU     010Q            BACKSPACE CHARACTER
000.012                123  A.LF    EQU     012Q            LINE FEED CHARACTER
000.015                124  A.CR    EQU     015Q            CARRIAGE RETURN CHARACTER
000.033                125  A.ESC   EQU     033Q            ESCAPE CHARACTER
000.177                126  A.DEL   EQU     177Q            DELETE OR RUBOUT CHARACTER
```

```
                        128  **      FRONT PANEL HARDWARE CONTROL BITS.
                        129
000.020                 130  CB.SSI  EQU     00010000B       SINGLE STEP INTERRUPT
000.040                 131  CB.MTL  EQU     00100000B       MONITOR LIGHT
000.100                 132  CB.CLI  EQU     01000000B       CLOCK INTERRUPT ENABLE
000.200                 133  CB.SPK  EQU     10000000B       SPEAKER ENABLE

                        135  **      DISPLAY MODE FLAGS (IN *DSPMOD*)
                        136
000.000                 137  DM.MR   EQU     0               MEMORY READ
000.001                 138  DM.MW   EQU     1               MEMORY WRITE
000.002                 139  DM.RR   EQU     2               REGISTER READ
000.003                 140  DM.RW   EQU     3               REGISTER WRITE
000.000                 141          XTEXT   TAPE            TAPE DEFINITIONS

                        163  **      MACHINE INSTRUCTIONS.
                        164
000.166                 165  MI.HLT  EQU     01110110B       HALT
000.311                 166  MI.RET  EQU     11001001B       RETURN
000.333                 167  MI.IN   EQU     11011011B       INPUT
000.323                 168  MI.OUT  EQU     11010011B       OUTPUT
000.072                 169  MI.LDA  EQU     00111010B       LDA
000.346                 170  MI.ANI  EQU     11100110B       ANI
000.021                 171  MI.LXID EQU     00010001B       LXI D
000.303                 172  MI.JMP  EQU     11000011B       JMP
000.335                 173  MI.LDXA EQU     11011101B       LD IX,  (BYTE A)
000.041                 174  MI.LDXB EQU     00100001B       LD IX,  (BYTE B)
000.375                 175  MI.LDYA EQU     11111101B       LD IY,  (BYTE A)
000.041                 176  MI.LDYB EQU     00100001B       LD IY,  (BYTE B)
000.010                 177  MI.EXAF EQU     00001000B       EX AF,AF'
000.335                 178  MI.JIXA EQU     11011101B       JP (IX)  (BYTE A)
000.351                 179  MI.JIXB EQU     11101001B       JP (IX)  (BYTE B)
000.375                 180  MI.JIYA EQU     11111101B       JP (IY)  (BYTE A)
000.351                 181  MI.JIYB EQU     11101001B       JP (IY)  (BYTE B)

                        183  **      USER OPTION BITS.
                        184  *
                        185  *       THESE BITS ARE SET IN CELL .MFLAG.
                        186
000.200                 187  UO.HLT  EQU     10000000B       DISABLE HALT PROCESSING
000.100                 188  UO.NFR  EQU     CB.CLI          NO REFRESH OF FRONT PANEL
000.002                 189  UO.DDU  EQU     00000010B       DISABLE DISPLAY UPDATE
000.001                 190  UO.CLK  EQU     00000001B       ALLOW PRIVATE INTERRUPT PROCESSING
```

```
000.000                192          XTEXT   U8251          DEFINE 8251 USART BITS

000.000                239          XTEXT   U8250          DEFINE 8250 ACE BITS
```

```
                                303  ***     INTERRUPT VECTORS.
                                304  *
                                305

                                307  **      LEVEL 0 - RESET
                                308  *
                                309  *       THIS 'INTERRUPT' MAY NOT BE PROCESSED BY A USER PROGRAM.
                                310
                                311
    000.000  303 000 004        312  INIT0   JMP     INITOX          DO H88 EXTENSION OF INITIALIZATION
    000.003  041 012 040        313  INIT0.0 LXI     H,PRSRAM+PRSL-1 (HL) = RAM DESTINATION FOR CODE
    000.006  303 073 000        314          JMP     INIT            INITIALIZE
                                315
    377.073                     316          ERRPL   INIT-1000A      BYTE IN WORD 10A MUST BE 0
                                317

                                319  **      LEVEL 1 - CLOCK
                                320
    000.010                     321  INT1    EQU     10Q             INTERRUPT ENTRY POINT
                                322
    000.000                     323          ERRNZ   *-11Q           INT0 TAKES UP ONE BYTE
                                324
    000.011  315 132 000        325          CALL    SAVALL          SAVE USER REGISTERS
    000.014  026 000            326          MVI     D,0
    000.016  303 201 000        327          JMP     CLOCK           PROCESS CLOCK INTERRUPT
    377.201                     328          ERRPL   CLOCK-1000A     EXTRA BYTE MUST BE 0

                                330  **      LEVEL 2 - SINGLE STEP
                                331  *
                                332  *       IF THIS INTERRUPT IS RECEIVED WHEN NOT IN MONITOR MODE,
                                333  *       THEN IT IS ASSUMED TO BE GENERATED BY A USER PROGRAM
                                334  *       (SINGLE STEPPING OR BREAKPOINTING). IN SUCH CASE, THE
                                335  *       USER PROGRAM IS ENTERED THROUGH (UIVEC+3
                                336
    000.020                     337  INT2    EQU     20A             LEVEL 2 ENTRY
                                338
    000.000                     339          ERRNZ   *-21A           INT1 TAKES EXTRA BYTE
                                340
    000.021  315 132 000        341          CALL    SAVALL          SAVE REGISTERS
    000.024  032                342          LDAX    D               (A) = (CTLFLG)
    040.011                     343          SET     CTLFLG
    000.025  303 244 001        344          JMP     STPRTN          STEP RETURN
```

```
                                346  ***     I/O INTERRUPT VECTORS.
                                347  *
                                348  *       INTERRUPTS 3 THROUGH 7 ARE AVAILABLE FOR GENERAL I/O USE.
                                349  *
                                350  *       THESE INTERRUPTS ARE NOT SUPPORTED BY MTR88, AND SHOULD
                                351  *       NEVER OCCUR UNLESS THE USER HAS SUPPLIED HANDLER ROUTINES
                                352  *       (THROUGH UIVEC)
                                353
000.030                         354          ORG     30A
                                355
000.030  303 045 040           356  INT3    JMP     UIVEC+6         JUMP TO USER ROUTINE
                                357
000.033  064 064 064           358          DB      '44440'         HEATH PART NUMBER 444-40


                                360
000.040                         361          ORG     40A
                                362
000.040  303 050 040           363  INT4    JMP     UIVEC+9         JUMP TO USER ROUTINE
                                364
000.043  044 122 116           365          DB      44Q,122Q,116Q,102Q,44Q  SUPPORT CODE


                                367
000.050                         368          ORG     50A
                                369
000.050  303 053 040           370  INT5    JMP     UIVEC+12        JUMP TO USER ROUTINE
                                371
                                372
                                373  **      DLY - DELAY TIME INTERVAL.
                                374  *
                                375  *       ENTRY   (A) = MILLISECOND DELAY COUNT/2
                                376  *       EXIT    NONE
                                377  *       USES    A,F
                                378
000.000                         379          ERRNZ   *-53A
                                380
000.053  365                   381  DLY     PUSH    PSW             SAVE COUNT
000.054  257                   382          XRA     A               DONT SOUND HORN
000.055  303 143 002           383          JMP     HRN0            PROCESS AS HORN


                                385
000.060                         386          ORG     60A
                                387
000.060  303 056 040           388  INT6    JMP     UIVEC+15        JUMP TO USER ROUTINE
                                389
                                390
000.063  076 320               391  GO.     MVI     A,CB.SSI+CB.CLI+CB.SPK  OFF MONITOR MODE LIGHT
000.065  303 235 001           392          JMP     SST1            RETURN TO USER PROGRAM
```

```
                                394
     000.070                    395          ORG     70A
                                396
     000.070  303 061 040       397  INT7    JMP     UIVEC+18        JUMP TO USER ROUTINE
```

```
                         400  **      INIT - INITIALIZE SYSTEM
                         401  *
                         402  *       INIT IS CALLED WHENEVER A HARDWARE MASTER-CLEAR IS INITIATED.
                         403  *
                         404  *       SETUP MTR88 CONTROL CELLS IN RAM.
                         405  *       DECODE HOW MUCH MEMORY EXISTS, SETUP STACKPOINTER, AND
                         406  *       ENTER THE MONITOR LOOP.
                         407  *
                         408  *       ENTRY   FROM MASTER CLEAR
                         409  *       EXIT    INTO MTR88 MAIN LOOP
                         410
  000.000                411           ERRNZ   *-73Q
                         412
  000.073  032           413  INIT     LDAX    D               COPY *PRSROM* INTO RAM
  000.074  167           414           MOV     M,A             MOVE BYTE
  000.075  053           415           DCX     H               DECREMENT DESTINATION
  000.076  034           416           INR     E               INCREMENT SOURCE
  000.077  302 073 000   417           JNZ     INIT            IF NOT DONE
                         418
  004.000                419  SINCR    EQU     4000A           SEARCH INCREMENT
                         420
  000.102  026 004       421           MVI     D,SINCR/256     (DE) = SEARCH INCREMENT
  000.104  041 000 034   422           LXI     H,START-SINCR   (HL) = FIRST RAM - SEARCH INCREMENT
                         423
                         424  *       DETERMINE MEMORY LIMIT.
                         425
  000.107  167           426  INIT1    MOV     M,A             RESTORE VALUE READ
  000.110  031           427           DAD     D               INCREMENT TRIAL ADDRESS
  000.111  176           428           MOV     A,M             (A) = CURRENT MEMORY VALUE
  000.112  065           429           DCR     M               TRY TO CHANGE IT
  000.113  276           430           CMP     M
  000.114  302 107 000   431           JNE     INIT1           IF MEMORY CHANGED
                         432
  000.117  053           433  INIT2    DCX     H
                         434
  000.120  371           435           SPHL                    SET STACKPOINTER = MEMORY LIMIT -1
                         436
  000.121  345           437           PUSH    H               SET *PC* VALUE ON STACK
  000.122  041 322 000   438           LXI     H,ERROR
  000.125  345           439           PUSH    H               SET 'RETURN ADDRESS'
                         440
                         441  *       CONFIGURE LOAD/DUMP UART
                         442
  000.126  076 116       443           MVI     A,UMI.1B+UMI.L8+UMI.16X
  000.130  323 371       444           OUT     OP.TPC          SET 8 BIT, NO PARITY, 1 STOP, X16
```

```
                                447  **      SAVALL - SAVE ALL REGISTERS ON STACK.
                                448  *
                                449  *       SAVALL IS CALLED WHEN AN INTERRUPT IS ACCEPTED, IN ORDER TO
                                450  *       SAVE THE CONTENTS OF THE REGISTERS ON THE STACK.
                                451  *
                                452  *       ENTRY   CALLED DIRECTLY FROM INTERRUPT ROUTINE.
                                453  *       EXIT    ALL REGISTERS PUSHED ON STACK,
                                454  *               IF NOT YET IN MONITOR MODE, REGPTR = ADDRESS OF REGISTERS
                                455  *               ON STACK.
                                456  *               (DE) = ADDRESS OF CTLFLG
                                457
    000.000                     458          ERRNZ   *-132A
                                459
    000.132  343                460  SAVALL  XTHL                    SET H,L ON STACK TOP
    000.133  325                461          PUSH    D
    000.134  305                462          PUSH    B
    000.135  365                463          PUSH    PSW
    000.136  353                464          XCHG                    (D,E) = RETURN ADDRESS
    000.137  041 012 000        465          LXI     H,10
    000.142  071                466          DAD     SP              (H,L) = ADDRESS OF USERS SP
                                467
                                468  **      REPLACE THESE INSTRUCTIONS WITH A JUMP AROUND THE NMI VECTOR JUMP
                                469  *
                                470  *       PUSH    H               SET ON STACK AS 'REGISTER'
                                471  *       PUSH    D               SET RETURN ADDRESS
                                472  *       LXI     D,CTLFLG
                                473  *       LDAX    D               (A) = CTLFLG
                                474
    000.143  303 105 004        475          JMP     SAVALLX         GO TO SAVALL EXTENSION
                                476
                                477  **      ENTRY POINT FOR THE Z80 NMI
                                478  *
                                479
                                480
    000.000                     481          ERRNZ   *-66H           Z80 NMI ADDRESS
                                482
    000.146  303 116 004        483  NMIENT  JMP     NMI
                                484
    000.000                     485          ERRNZ   SAVALLR-151A    DO NOT CHANGE ORGANIZATION
                                486
    000.151                     487  SAVALLR EQU     *               SAVALL EXTENSION RETURN ADDRESS
                                488
    000.151  057                489          CMA
    000.152  346 060            490          ANI     CB.MTL+CB.SSI   SAVE REGISTER ADDR IF USER OR SINGLE-STEP
    000.154  310                491          RZ                      RETURN IF WAS INTERRUPT OF MONITOR LOOP
    000.155  041 002 000        492          LXI     H,2
    000.160  071                493          DAD     SP              (H,L) = ADDRESS OF 'STACKPTR' ON STACK
    000.161  042 035 040        494          SHLD    REGPTR
    000.164  311                495          RET
```

```
                                497  **      CUI - CHECK FOR USER INTERRUPT PROCESSING.
                                498  *
                                499  *       CUI IS CALLED TO SEE IF THE USER HAS SPECIFIED PROCESSING
                                500  *       FOR THE CLOCK INTERRUPT.
                                501
     000.000                    502          ERRNZ   *-165A
                                503
     040.010                    504          SET     .MFLAG          REFERENCE TO MFLAG
     000.165  012               505  CUI1    LDAX    B               (A) = .MFLAG
     000.000                    506          ERRNZ   UO.CLK-1        CODE ASSUMED = 01
     000.166  017               507          RRC
     000.167  334 037 040       508          CC      UIVEC           IF SPECIFIED, TRANSFER TO USER
                                509
                                510  *       RETURN TO PROGRAM FROM INTERRUPT.
                                511
     000.000                    512          ERRNZ   *-172A
                                513
     000.172  361               514  INTXIT  POP     PSW             REMOVE FAKE 'STACK REGISTER'
     000.173  361               515          POP     PSW
     000.174  301               516          POP     B
     000.175  321               517          POP     D
     000.176  341               518          POP     H
     000.177  373               519          EI
     000.200  311               520          RET
```

```
                                523  ***     CLOCK - PROCESS CLOCK INTERRUPT
                                524  *
                                525  *       CLOCK IS ENTERED WHENEVER A 2-MILLISECOND CLOCK INTERRUPT IS
                                526  *       PROCESSED.
                                527  *
                                528  *       TICCNT IS INCREMENTED EVERY INTERRUPT.
                                529
    000.000                     530          ERRNZ   *-201A
                                531
    000.201  052 033 040        532  CLOCK   LHLD    TICCNT
    000.204  043                533          INX     H
    000.205  042 033 040        534          SHLD    TICCNT          INCREMENT TICCOUNT
                                535
    000.210  072 011 040        536          LDA     CTLFLG          CLEAR CLOCK INTERRUPT FLIP-FLOP
    000.213  323 360            537          OUT     OP.CTL
                                538
                                539  *       EXIT CLOCK INTERRUPT.
                                540
    000.215  001 011 040        541          LXI     B,CTLFLG
    000.220  012                542          LDAX    B               (A) = CTLFLG
    000.221  346 040            543          ANI     CB.MTL
    000.223  302 172 000        544          JNZ     INTXIT          IF IN MONITOR MODE
    000.226  013                545          DCX     B
    000.000                     546          ERRNZ   CTLFLG-.MFLAG-1
    000.227  012                547          LDAX    B               (A) = .MFLAG
    000.000                     548          ERRNZ   U0.HLT-200Q     ASSUME HIGH-ORDER
    000.230  027                549          RAL
    000.231  332 270 000        550          JC      CLK4            SKIP IT
                                551
                                552  *       NOT IN MONITOR MODE. CHECK FOR HALT
                                553
    000.234  076 012            554          MVI     A,10            (A) = INDEX OF *F* REG
    000.236  315 052 003        555          CALL    LRA.            LOCATE REGISTER ADDRESS
    000.241  136                556          MOV     E,M
    000.242  043                557          INX     H
    000.243  126                558          MOV     D,M             (D,E) = PC CONTENTS
    000.244  033                559          DCX     D
    000.245  032                560          LDAX    D
    000.246  376 166            561          CPI     MI.HLT          CHECK FOR HALT
    000.250  302 165 000        562          JNZ     CUI1
                                563
    000.253  076 007            564          MVI     A,A.BEL         DING BELL
    000.255  315 302 003        565          CALL    WCC
    000.260  076 110            566          MVI     A,'H'           'H' FOR HALT
    000.262  315 302 003        567          CALL    WCC
    000.265  303 322 000        568          JMP     ERROR
                                569
                                570  ***     JE      ERROR           IF HALT, BE IN MONITOR MODE
                                571
                                572  *       NONE OF THE ABOVE, SO ALLOW USER PROCESSING OF CLOCK INTERRUPT
                                573
    000.270                     574  CLK4    EQU     *
    000.270  303 165 000        575          JMP     CUI1            ALLOW USER PROCESSING OF CLOCK
                                600          LON     L
```

```
                                    604  ***     ERROR - COMMAND ERROR.
                                    605  *
                                    606  *       ERROR IS CALLED AS A 'BAIL-OUT' ROUTINE.
                                    607  *
                                    608  *       IT RESETS THE OPERATIONAL MODE, AND RESTORES THE STACKPOINTER.
                                    609  *
                                    610  *       ENTRY   NONE
                                    611  *       EXIT    TO MTR LOOP
                                    612  *               CTLFLG SET
                                    613  *               .MFLAG CLEARED
                                    614  *       USES    ALL
                                    615
    000.000                         616          ERRNZ   *-322A
                                    617
    000.322                         618  ERROR   EQU     *
    000.322  041 010 040            619          LXI     H,.MFLAG
    000.325  176                    620          MOV     A,M             (A) = .MFLAG
    000.326  346 275                621          ANI     377Q-UO.DDU-UO.NFR      RE-ENABLE DISPLAYS
    000.330  167                    622          MOV     M,A             REPLACE
    000.331  043                    623          INX     H
    000.332  066 360                624          MVI     M,CB.SSI+CB.MTL+CB.CLI+CB.SPK   RESTORE *CTLFLG*
    000.000                         625          ERRNZ   CTLFLG-.MFLAG-1
    000.334  373                    626          EI
    000.335  052 035 040            627          LHLD    REGPTR
    000.340  371                    628          SPHL                    RESTORE STACK POINTER TO EMPTY STATE
    000.341  315 136 002            629          CALL    ALARM           ALARM FOR 200 MS


                                    631  **      MTR - MONITOR LOOP.
                                    632  *
                                    633
    000.000                         634          ERRNZ   *-344A
                                    635
    000.344                         636  MTR     EQU     *
    000.344  373                    637          EI
                                    638
    000.345                         639  MTR1    EQU     *
    000.345  041 345 000            640          LXI     H,MTR1
    000.350  345                    641          PUSH    H               SET 'MTR1' AS RETURN ADDRESS
    000.351  041 112 006            642          LXI     H,MSG.PR        TYPE PROMPT MESSAGE
    000.354  315 100 006            643          CALL    TYPMSG
                                    644
    000.357  315 262 003            645  MTR.2   CALL    RCC             READ A CONSOLE CHARACTER
    000.362  346 137                646          ANI     01011111B       MAKE SURE ITS UPPER CASE TO MATCH TABLE
    000.364  041 025 001            647          LXI     H,MTRA          LOOK UP CHARACTER IN *MTRA*
    000.367  006 006                648          MVI     B,MTRAL         (B) = LENGTH OF TABLE
    000.371  276                    649  MTR.3   CMP     M               SEE IF CHARACTER FROM CONSOLE = TABLE ENTRY
    000.372  312 014 001            650          JZ      MTR.4           IF EQUAL
                                    651
    000.375  043                    652          INX     H               POINT TO NEXT TABLE ENTRY
    000.376  043                    653          INX     H
    000.377  043                    654          INX     H
    001.000  005                    655          DCR     B               SEE IF PAST END OF TABLE
    001.001  302 371 000            656          JNZ     MTR.3           IF NOT PAST
```

```
                          657
001.004  076 007         658           MVI     A,A.BEL         ELSE, DING ERROR
001.006  315 302 003     659           CALL    WCC
001.011  303 357 000     660           JMP     MTR.2           TRY AGAIN
                          661
001.014  315 302 003     662  MTR.4    CALL    WCC             WRITE CHARACTER BACK TO CONSOLE
001.017  043               663           INX     H               GET ROUTINE ADDRESS LSB
001.020  176               664           MOV     A,M
001.021  043               665           INX     H               GET MSB
001.022  146               666           MOV     H,M
001.023  157               667           MOV     L,A             (H,L) = ROUTINE ADDRESS
001.024  351               668           PCHL                    GO TO ROUTINE
                          669
                          670
                          671
001.025                   672  MTRA     EQU     *               JUMP TABLE
000.001                   673
001.025  107               674           DB      'G'             GO TO USER ROUTINE
001.026  146 001           675           DW      GO88
                          676
001.030  114               677           DB      'L'             CASSETTE LOAD
001.031  067 001           678           DW      SRMEM
                          679
001.033  104               680           DB      'D'             SET UP CASSETTE DUMP
001.034  313 004           681           DW      SWMEM
                          682
001.036  123               683           DB      'S'             SUBSTITUTE MEMORY MODE
001.037  370 004           684           DW      SUBM
                          685
001.041  120               686           DB      'P'             PROGRAM COUNTER ALTER MODE
001.042  103 001           687           DW      PCA
                          688
001.044  102               689           DB      'B'             BOOT HDOS
001.045  256 004           690           DW      BOOT
                          691
000.006                   692  MTRAL    EQU     *-MTRA/3        NUMBER OF TABLE ENTRYS          /JWT 790507/
                          703           LON     L


                          705  **       SAE - STORE ABUSS AND EXIT.
                          706  *
                          707  *        ENTRY   (HL) = ABUSS VALUE
                          708  *        EXIT    TO (RET)
                          709  *        USES    NONE
                          710
000.000                   711           ERRNZ   *-1063A
                          712
001.063  042 024 040     713  SAE      SHLD    ABUSS
001.066  311               714           RET
```

```
                                717  **      SRMEM - H88/H89 ENTRY POINT FOR A CASSETTE LOAD
                                718  *
                                719
001.067  041 170 006           720  SRMEM   LXI     H,MSG.LD        COMPLETE MESSAGE
001.072  315 100 006           721          CALL    TYPMSG
001.075  315 003 006           722          CALL    WCR             WAIT FOR A CARRIAGE RETURN
001.100  303 261 001           723          JMP     RMEM            LOAD TAPE


                                725  **      PCA - PROGRAM COUNTER ALTER
                                726  *
                                727  *       PCA INPUTS AND/OR DISPLAYS THE CURRENT USER PROGRAM VALUE AND ALLOWS
                                728  *       A NEW VALUE TO BE ENTERED OR RETAINS THE CURRENT VALUE IF
                                729  *       A CR IS TYPED
                                730  *
                                731  *       ENTRY   NONE
                                732  *       EXIT    NONE
                                733  *       USES    A,D,E,H,L,F
                                734
                                735
001.103  041 214 006           736  PCA     LXI     H,MSG.PC        COMPLETE PC MESSAGE
001.106  315 100 006           737          CALL    TYPMSG
001.111  076 012               738          MVI     A,10            GET LOCATION OF USER PC
001.113  315 052 003           739          CALL    LRA.
001.116  136                   740          MOV     E,M             (D,E) = USER PC VALUE
001.117  043                   741          INX     H
001.120  126                   742          MOV     D,M
001.121  353                   743          XCHG                    (H,L) = USER PC VALUE
                                744
001.122  315 150 005           745          CALL    IROC            INPUT NEXT CHARACTER
001.125  332 137 001           746          JC      PCA1            IF FIRST CHARACTER WAS OCTAL, INPUT NEW PC
                                747
001.130  315 313 005           748          CALL    TOA             ELSE, OUTPUT CURRENT VALUE
001.133  315 150 005           749          CALL    IROC            SEE IF USER WANTS TO CHANGE IT NOW
001.136  320                   750          RNC                     IF NO CHANGE, EXIT
                                751
                                752  *       ENTER NEW USER PC VALUE
                                753
001.137  353                   754  PCA1    XCHG                    (H,L) = ADDRESS OF USER PC VALUE
001.140  026 015               755          MVI     D,A.CR          END BYTE WITH A RETURN
001.142  315 062 003           756          CALL    IOA             INPUT NEW ADDRESS
001.145  311                   757          RET                     EXIT


                                759  **      GO88 - GO TO USER ROUTINE FROM H88 MONITOR
                                760  *
                                761  *       GO88 WAITS FOR A CARRIAGE RETURN OR A NEW ADDRESS TERMINATED WITH
                                762  *       A CARRIAGE RETURN.  IF NO ADDRESS IS ENTERED, GO88 TRANSFERS
                                763  *       CONTROL TO THE ADDRESS SPECIFIED BY THE USER PC VALUE
                                764
                                765
001.146  041 165 006           766  GO88    LXI     H,MSG.GO        COMPLETE GO MESSAGE
```

```
001.151  315 100 006   767          CALL    TYPMSG
001.154  315 150 005   768          CALL    IROC            INPUT A RETURN OR AN OCTAL CHARACTER
001.157  322 177 001   769          JNC     G088.1          IF RETURN, GO TO CURRENT USER PC
                       770
001.162  365          771          PUSH    PSW             ELSE SAVE OCTAL CHARACTER AND FLAGS
001.163  076 012      772          MVI     A,10            GET ADDRESS OF USER PC
001.165  315 052 003   773          CALL    LRA.
001.170  043          774          INX     H               POINT TO MSB
001.171  361          775          POP     PSW             GET FIRST CHARACTER BACK
001.172  026 015      776          MVI     D,A.CR          END ADDRESS WITH A RETURN
001.174  315 062 003   777          CALL    IOA             INPUT NEW GO ADDRESS
001.177  315 302 003   778  G088.1  CALL    WCC             ECHO RETURN
001.202  076 012      779          MVI     A,A.LF          LINE FEED
001.204  315 302 003   780          CALL    WCC
001.207  303 222 001   781          JMP     GO              EXECUTE USER ROUTINE


001.212  110 103 101   783          DB      110Q,103Q,101Q,046Q,103Q,101Q,107Q,056Q DESIGN CODE


                       785  **      GO - RETURN TO USER MODE
                       786  *
                       787  *       ENTRY   NONE
                       788
000.000                789          ERRNZ   *-1222A
                       790
001.222  303 063 000   791  GO      JMP     GO.             ROUTINE IS IN WASTE SPACE


                       793  **      SSTEP - SINGLE STEP INSTRUCTION.
                       794  *
                       795  *       ENTRY   NONE
                       796
000.000                797          ERRNZ   *-1225A
                       798
001.225                799  SSTEP   EQU     *               SINGLE STEP
001.225  363          800          DI                      DISABLE INTERRUPTS UNTIL THE RIGHT TIME
001.226  072 011 040   801          LDA     CTLFLG
001.231  356 020      802          XRI     CB.SSI          CLEAR SINGLE STEP INHIBIT
001.233  323 360      803          OUT     OP.CTL          PRIME SINGLE STEP INTERRUPT
001.235  062 011 040   804  SST1    STA     CTLFLG          SET NEW FLAG VALUES
001.240  341          805          POP     H               CLEAN STACK
001.241  303 172 000   806          JMP     INTXIT          RETURN TO USER ROUTINE FOR STEP
```

```
                                808  **      STPRTN - SINGLE STEP RETURN
                                809
     000.000                    810          ERRNZ   *-1244A
                                811
     001.244                    812  STPRTN  EQU     *
     001.244  366 020           813          ORI     CB.SSI          DISABLE SINGLE STEP INTERRUPTION
     001.246  323 360           814          OUT     OP.CTL          TURN OFF SINGLE STEP ENABLE
     040.011                    815          SET     CTLFLG
     001.250  022               816          STAX    D
     001.251  346 040           817          ANI     CB.MTL          SEE IF IN MONITOR MODE
     001.253  302 344 000       818          JNZ     MTR
     001.256  303 042 040       819          JMP     UIVEC+3         TRANSFER TO USER'S ROUTINE


                                821  **      RMEM - LOAD MEMORY FROM TAPE.
                                822  *
                                823
     000.000                    824          ERRNZ   *-1261A
                                825
     001.261  041 244 002       826  RMEM    LXI     H,TPABT
     001.264  042 031 040       827          SHLD    TPERRX          SETUP ERROR EXIT ADDRESS
                                828  *       JMP     LOAD


                                830  ***     LOAD - LOAD MEMORY FROM TAPE.
                                831  *
                                832  *       READ THE NEXT RECORD FROM THE CASSETTE TAPE.
                                833  *
                                834  *       USE THE LOAD ADDRESS IN THE TAPE RECORD.
                                835  *
                                836  *       ENTRY   (HL) = ERROR EXIT ADDRESS
                                837  *       EXIT    USER P-REG (IN STACK) SET TO ENTRY ADDRESS
                                838  *               TO CALLER IF ALL OK
                                839  *               TO ERROR EXIT IF TAPE ERRORS DETECTED.
                                840
                                841
     000.000                    842          ERRNZ   *-1267A
                                843
     001.267                    844  LOAD    EQU     *
     001.267  001 000 376       845          LXI     B,1000A-RT.MI*256-256   (BC) = - REQUIRED TYPE AND #
     001.272  315 265 002       846  LOA0    CALL    SRS             SCAN FOR RECORD START
     001.275  157               847          MOV     L,A             (HL) = COUNT
     001.276  353               848          XCHG                    (DE) = COUNT, (HL) = TYPE AND #
     001.277  015               849          DCR     C               (C) = - NEXT #
     001.300  011               850          DAD     B
     001.301  174               851          MOV     A,H
     001.302  305               852          PUSH    B               SAVE TYPE AND #
     001.303  365               853          PUSH    PSW             SAVE TYPE CODE
     001.304  346 177           854          ANI     177Q            CLEAR END FLAG BIT
     001.306  265               855          ORA     L
     001.307  076 002           856          MVI     A,2             SEQUENCE ERROR
     001.311  302 205 002       857          JNE     TPERR           IF NOT RIGHT TYPE OR SEQUENCE
```

```
001.314  315 325 002   858          CALL    RNF             READ ADDR
001.317  104           859          MOV     B,H
001.320  117           860          MOV     C,A             (BC) = P-REG ADDRESS
001.321  076 012       861          MVI     A,10
001.323  325           862          PUSH    D               SAVE (DE)
001.324  315 052 003   863          CALL    LRA.            LOCATE REG ADDRESS
001.327  321           864          POP     D               RESTORE (DE)
001.330  161           865          MOV     M,C             SET P-REG IN MEM
001.331  043           866          INX     H
001.332  160           867          MOV     M,B
001.333  315 325 002   868          CALL    RNF             READ ADDRESS
001.336  157           869          MOV     L,A             (HL) = ADDRESS, (DE) = COUNT
001.337  042 000 040   870          SHLD    START
                       871
001.342  315 331 002   872  LOA1    CALL    RNB             READ BYTE
001.345  167           873          MOV     M,A
                       874
                       875  *       SHOW H88 THAT SOMETHING IS HAPPENING
                       876
001.346  315 024 006   877          CALL    TPDSP           DISPLAY TO H88 USER THAT WE ARE LOADING
                       878
001.351  043           879          INX     H
001.352  033           880          DCX     D
001.353  172           881          MOV     A,D
001.354  263           882          ORA     E
001.355  302 342 001   883          JNZ     LOA1            IF MORE TO GO
                       884
001.360  315 172 002   885          CALL    CTC             CHECK  TAPE CHECKSUM
                       886
                       887  *       READ NEXT BLOCK
                       888
001.363  361           889          POP     PSW             (A) = FILE TYPE BYTE
001.364  301           890          POP     B               (BC) = -(LAST TYPE, LAST #)
001.365  007           891          RLC
001.366  332 133 002   892          JC      TFT             ALL DONE - TURN OFF TAPE
001.371  303 272 001   893          JMP     LOA0            READ ANOTHER RECORD
```

```
                                  896  ***  DUMP - DUMP MEMORY TO MAG TAPE.
                                  897  *
                                  898  *    DUMP SPECIFIED MEMORY RANGE TO MAG TAPE.
                                  899  *
                                  900  *    ENTRY   (START) = START ADDRESS
                                  901  *            (ABUSS) = END ADDRESS
                                  902  *            USER PC = ENTRY POINT ADDRESS
                                  903  *    EXIT    TO CALLER.
                                  904
000.000                           905          ERRNZ   *-1374A
                                  906
001.374                           907  WMEM    EQU     *
001.374  041 244 002              908          LXI     H,TPABT
001.377  042 031 040              909          SHLD    TPERRX          SETUP ERROR EXIT
                                  910
000.000                           911          ERRNZ   *-2002A
                                  912
002.002  076 001                  913  DUMP    MVI     A,UCI.TE
002.004  323 371                  914          OUT     OP.TPC          SETUP TAPE CONTROL
002.006  076 026                  915          MVI     A,A.SYN
002.010  046 040                  916          MVI     H,32            (H) = # OF SYNC CHARACTERS
002.012  315 024 003              917  WME1    CALL    WNB
002.015  045                      918          DCR     H
002.016  302 012 002              919          JNZ     WME1            WRITE SYN HEADER
002.021  076 002                  920          MVI     A,A.STX
002.023  315 024 003              921          CALL    WNB             WRITE STX
002.026  154                      922          MOV     L,H             (HL) = 00
002.027  042 027 040              923          SHLD    CRCSUM          CLEAR CRC 16
002.032  041 001 201              924          LXI     H,RT.MI+80H*256+1       FIRST AND LAST MI RECORD
002.035  315 017 003              925          CALL    WNF             WRITE HEADER
002.040  052 000 040              926          LHLD    START
002.043  353                      927          XCHG                    (D,E) = START ADDRESS
002.044  052 024 040              928          LHLD    ABUSS           (H,L) = STOP ADDR
002.047  043                      929          INX     H               COMPUTE WITH STOP+1
002.050  175                      930          MOV     A,L
002.051  223                      931          SUB     E
002.052  157                      932          MOV     L,A
002.053  174                      933          MOV     A,H
002.054  232                      934          SBB     D
002.055  147                      935          MOV     H,A             (HL) = COUNT
002.056  315 017 003              936          CALL    WNF             WRITE COUNT
002.061  345                      937          PUSH    H
002.062  076 012                  938          MVI     A,10
002.064  325                      939          PUSH    D               SAVE (DE)
002.065  315 052 003              940          CALL    LRA.            LOCATE P-REG ADDRESS
002.070  176                      941          MOV     A,M
002.071  043                      942          INX     H
002.072  146                      943          MOV     H,M
002.073  157                      944          MOV     L,A             (HL) = CONTENTS OF PC
002.074  315 017 003              945          CALL    WNF             WRITE HEADER
002.077  341                      946          POP     H               (HL) = ADDRESS
002.100  321                      947          POP     D               (DE) = COUNT
002.101  315 017 003              948          CALL    WNF
                                  949
002.104  176                      950  WME2    MOV     A,M
002.105  315 024 003              951          CALL    WNB             WRITE BYTE
```

```
                                952
                                953  *       SHOW H88 USER THAT DUMP IS DUMPING
                                954
     002.110  315 024 006      955          CALL    TPDSP           DISPLAY DUMP
                                956
     002.113  043              957          INX     H
     002.114  033              958          DCX     D
     002.115  172              959          MOV     A,D
     002.116  263              960          ORA     E
     002.117  302 104 002      961          JNZ     WME2            IF MORE TO GO
                                962
                                963  *       WRITE CHECKSUM
                                964
     002.122  052 027 040      965          LHLD    CRCSUM
     002.125  315 017 003      966          CALL    WNF             WRITE IT
     002.130  315 017 003      967          CALL    WNF             FLUSH CHECKSUM
                                968  *       JMP     TFT


                                970  **      TFT - TURN OFF TAPE.
                                971  *
                                972  *       STOP THE TAPE TRANSPORT.
                                973  *
                                974
     000.000                    975          ERRNZ   *-2133A
                                976
     002.133  257               977  TFT     XRA     A
     002.134  323 371           978          OUT     OP.TPC          TURN OFF TAPE


                                980  **      HORN - MAKE NOISE.
                                981  *
                                982  *       ENTRY   (A) = (MILLISECOND COUNT)/2
                                983  *       EXIT    NONE
                                984  *       USES    A,F
                                985
     000.000                    986          ERRNZ   *-2136A
                                987
     002.136                    988  ALARM   EQU     *
     002.136  030 027           989          JR      ALARMB          BRANCH TO A JUMP TO NOISE TO DING BELL
                                990
     000.000                    991          ERRNZ   *-2140A
                                992
     002.140  365               993  HORN    PUSH    PSW
     002.141  076 200           994          MVI     A,CB.SPK        TURN ON SPEAKER
                                995
     002.143  343              996  HRN0    XTHL                    SAVE (HL), (H) = COUNT
     002.144  325              997          PUSH    D               SAVE (DE)
     002.145  353              998          XCHG                    (D) = LOOP COUNT
     002.146  041 011 040      999          LXI     H,CTLFLG
     002.151  256             1000          XRA     M
     002.152  136             1001          MOV     E,M             (E) = OLD CTLFLG VALUE
```

```
                        1002            MOV     M;A             TURN ON HORN
002,153 167             1003            MVI     L,#TICCNT
002.154 056 033         1004
                        1005            MOV     A,D             (A) = CYCLE COUNT
002.156 172             1006            ADD     M
002.157 206             1007    HRN2    CMP     M               WAIT REQUIRED TICCOUNTS
002.160 276             1008            JNE     HRN2
002.161 302 160 002     1009
                        1010            JMP     HRNX            JUMP TO AN EXTENSION OF HORN SO ROOM
002.164 303 045 006     1011    *                                       CAN BE MADE FOR A JUMP TO NIOSE
                        1012
                        1013
                        1014    ALARMB  JMP     NOISE           SEND A BELL TO THE CONSOLE
002.167 303 053 006     1015
```

```
                                1020  **      CTC - VERIFY CHECKSUM.
                                1021  *
                                1022  *       ENTRY   TAPE JUST BEFORE CRC
                                1023  *       EXIT    TO CALLER IF OK
                                1024  *               TO *TPERR* IF BAD
                                1025  *       USES    A;F;H;L
                                1026
                                1027
     000.000                    1028            ERRNZ   *-2172A
                                1029
     002.172  315 325 002       1030  CTC       CALL    RNP             READ NEXT PAIR
     002.175  052 027 040       1031            LHLD    CRCSUM
     002.200  174               1032            MOV     A,H
     002.201  265               1033            ORA     L
     002.202  310               1034            RZ                      RETURN OF OK
     002.203  076 001           1035            MVI     A,1             CHECKSUM ERROR
                                1036  *         JMP     TPERR           (B) = CODE


                                1038  **      TPERR - PROCESS TAPE ERROR.
                                1039  *
                                1040  *       DISPLAY ERR NUMBER IN LOW BYTE OF ABUSS
                                1041  *
                                1042  *       IF ERROR NUMBER EVEN, DONT ALLOW #
                                1043  *       IF ERROR NUMBER ODD, ALLOW #
                                1044  *
                                1045  *       ENTRY   (B) = PATTERN
                                1046
     000.000                    1047            ERRNZ   *-2205A
                                1048
     002.205  107               1049  TPERR     MOV     B,A             (B) = CODE
     002.206  315 063 006       1050            CALL    TPERMSG         DISPLAY ERROR NUMBER ON CONSOLE
     002.211  315 133 002       1051            CALL    TFT             TURN OFF TAPE
                                1052
                                1053  *         IS #, RETURN (IF PARITY ERROR)
                                1054
     002.214  346               1055            DB      MI,ANI          FALL THROUGH WITH CARRY CLEAR
     002.215  170               1056  TER3      MOV     A,B
                                1057
     002.216  017               1058            RRC
     002.217  330               1059            RC                      RETURN IF OK
                                1060
                                1061  *         BEEP AND FLASH ERROR NUMBER
                                1062
     002.220  334 136 002       1063  TER1      CC      ALARM           ALARM IF PROPER TIME
     002.223  315 252 002       1064            CALL    TPXIT           SEE IF *
     002.226  333 360           1065            IN      IF;PAD
     002.230  376 057           1066            CPI     00101111B       CHECK FOR #
     002.232  312 215 002       1067            JE      TER3            IF #
     002.235  072 034 040       1068            LDA     TICCNT+1
     002.240  037               1069            RAR                     'C' SET IF 1/2 SECOND
     002.241  303 220 002       1070            JMP     TER1
```

```
                        1072  **      TPABT - ABORT TAPE LOAD OR DUMP.
                        1073  *
                        1074  *       ENTERED WHEN LOADING OR DUMPING, AND THE '*' KEY
                        1075  *       IS STRUCK.
                        1076
                        1077
    000.000             1078          ERRNZ   *-2244A
                        1079
    002.244  257        1080  TPABT   XRA     A
    002.245  323 371    1081          OUT     OP.TPC          OFF TAPE
    002.247  303 322 000 1082          JMP     ERROR


                        1084  **      TPXIT - CHECK FOR USER FORCED EXIT.
                        1085  *
                        1086  *       TPXIT CHECKS FOR AN '*' KEYPAD ENTRY. IF SO, TAKE
                        1087  *       THE TAPE DRIVER ABNORMAL EXIT.
                        1088  *
                        1089  *       ENTRY   NONE
                        1090  *       EXIT    TO *RET* IF NOT '*'
                        1091  *                 (A) = PORT STATUS
                        1092  *                 TO (TPERRX) IF '*' DOWN
                        1093  *       USES    A,F
                        1094
                        1095
    000.000             1096          ERRNZ   *-2252A
                        1097
    002.252  333 360    1098  TPXIT   IN      IP.PAD
    002.254  376 157    1099          CPI     01101111B       *
    002.256  333 371    1100          IN      IP.TPC          READ TAPE STATUS
    002.260  300        1101          RNE                     NOT '*', RETURN WITH STATUS
    002.261  052 031 040 1102          LHLD    TPERRX
                        1103
    000.000             1104          ERRNZ   *-2264A
                        1105
    002.264  351        1106          PCHL                    ENTER (TPERRX)


                        1108  **      SRS - SCAN RECORD START
                        1109  *
                        1110  *       SRS READS BYTES UNTIL IT RECOGNIZES THE START OF A RECORD.
                        1111  *
                        1112  *       THIS REQUIRES
                        1113  *       AT LEAST 10 SYNC CHARACTERS
                        1114  *       1 STX CHARACTER.
                        1115  *
                        1116  *       THE CRC-16 IS THEN INITIALIZED.
                        1117  *
                        1118  *       ENTRY   NONE
                        1119  *       EXIT    TAPE POSITIONED (AND MOVING), CRCSUM =0
                        1120  *                 (DE) = HEADER BYTES
                        1121  *                 (HA) = RECORD COUNT
```

```
                        1122  *       USES    A,F,D,E,H,L
                        1123
 000.000                1124           ERRNZ   *-2265A
                        1125
 002.265                1126  SRS      EQU     *
 002.265  026 000       1127  SRS1     MVI     D,0
 002.267  142           1128           MOV     H,D
 002.270  152           1129           MOV     L,D             (HL) = 0
 002.271  315 331 002   1130  SRS2     CALL    RNB             READ NEXT BYTE
 002.274  024           1131           INR     D
 002.275  376 026       1132           CPI     A.SYN
 002.277  312 271 002   1133           JE      SRS2            HAVE SYN
 002.302  376 002       1134           CPI     A.STX
 002.304  302 265 002   1135           JNE     SRS1            NOT STX - START OVER
                        1136
 002.307  076 012       1137           MVI     A,10
 002.311  272           1138           CMP     D               SEE IF ENOUGH SYN CHARACTERS
 002.312  322 265 002   1139           JNC     SRS1            NOT ENOUGH
 002.315  042 027 040   1140           SHLD    CRCSUM          CLEAR CRC-16
 002.320  315 325 002   1141           CALL    RNP             READ LEADER
 002.323  124           1142           MOV     D,H
 002.324  137           1143           MOV     E,A
                        1144  *        JMP     RNP             READ COUNT


                        1146  **       RNP - READ NEXT PAIR.
                        1147  *
                        1148  *        RNP READS THE NEXT TWO BYTES FROM THE INPUT DEVICE.
                        1149  *
                        1150  *        ENTRY   NONE
                        1151  *        EXIT    (H,A) = BYTE PAIR
                        1152  *        USES    A,F,H
                        1153
 000.000                1154           ERRNZ   *-2325A
                        1155
 002.325  315 331 002   1156  RNP      CALL    RNB             READ NEXT BYTE
 002.330  147           1157           MOV     H,A
                        1158  *        JMP     RNB             READ NEXT BYTE


                        1160  **       RNB - READ NEXT BYTE
                        1161  *
                        1162  *        RNB READS THE NEXT SINGLE BYTE FROM THE INPUT DEVICE.
                        1163  *        THE CHECKSUM IS TAKEN FOR THE CHARACTER.
                        1164  *
                        1165  *        ENTRY   NONE
                        1166  *        EXIT    (A) = CHARACTER
                        1167  *        USES    A,F
                        1168
 000.000                1169           ERRNZ   *-2331A
                        1170
 002.331  076 064       1171  RNB      MVI     A,UCI.RO+UCI.ER+UCI.RE  TURN ON READER FOR NEXT BYTE
```

```
002.333  323 371          1172           OUT     OP.TPC
002.335  315 252 002      1173   RNB1    CALL    TPXIT           CHECK FOR *, READ STATUS
002.340  346 002          1174           ANI     USR.RXR
002.342  312 335 002      1175           JZ      RNB1            IF NOT READY
002.345  333 370          1176           IN      IP.TPD          INPUT DATA
                          1177   *       JMP     CRC             CHECKSUM



                          1179   **      CRC - COMPUTE CRC-16
                          1180   *
                          1181   *       CRC COMPUTES A CRC-16 CHECKSUM FROM THE POLYNOMIAL
                          1182   *
                          1183   *       (X + 1) * (X^15 + X + 1)
                          1184   *
                          1185   *       SINCE THE CHECKSUM GENERATED IS A DIVISION REMAINDER,
                          1186   *       A CHECKSUMED DATA SEQUENCE CAN BE VERIFIED BY RUNNING
                          1187   *       THE DATA THROUGH CRC, AND THEN RUNNING THE PREVIOUSLY OBTAINED
                          1188   *       CHECKSUM THROUGH CRC. THE RESULTANT CHECKSUM SHOULD BE 0.
                          1189   *
                          1190   *       ENTRY   (CRCSUM) = CURRENT CHECKSUM
                          1191   *               (A) = BYTE
                          1192   *       EXIT    (CRCSUM) UPDATED
                          1193   *               (A) UNCHANGED.
                          1194   *       USES    F
                          1195
000.000                   1196           ERRNZ   *-2347A
                          1197
002.347  305              1198   CRC     PUSH    B               SAVE (BC)
002.350  006 010          1199           MVI     B,8             (B) = BIT COUNT
002.352  345              1200           PUSH    H
002.353  052 027 040      1201           LHLD    CRCSUM
002.356  007              1202   CRC1    RLC
002.357  117              1203           MOV     C,A             (C) = BIT
002.360  175              1204           MOV     A,L
002.361  207              1205           ADD     A
002.362  157              1206           MOV     L,A
002.363  174              1207           MOV     A,H
002.364  027              1208           RAL
002.365  147              1209           MOV     H,A
002.366  027              1210           RAL
002.367  251              1211           XRA     C
002.370  017              1212           RRC
002.371  322 004 003      1213           JNC     CRC2            IF NOT TO XOR
002.374  174              1214           MOV     A,H
002.375  356 200          1215           XRI     200Q
002.377  147              1216           MOV     H,A
003.000  175              1217           MOV     A,L
003.001  356 005          1218           XRI     5Q
003.003  157              1219           MOV     L,A
003.004  171              1220   CRC2    MOV     A,C
003.005  005              1221           DCR     B
003.006  302 356 002      1222           JNZ     CRC1            IF MORE TO GO
003.011  042 027 040      1223           SHLD    CRCSUM
003.014  341              1224           POP     H               RESTORE (HL)
```

```
003.015  301            1225            POP     B               RESTORE (BC)
003.016  311            1226            RET                     EXIT


                        1228  **        WNF - WRITE NEXT PAIR.
                        1229  *
                        1230  *         WNF WRITES THE NEXT TWO BYTES TO THE CASSETTE DRIVE.
                        1231  *
                        1232  *         ENTRY   (H,L) = BYTES
                        1233  *         EXIT    WRITTEN.
                        1234  *         USES    A,F
                        1235
000.000                 1236            ERRNZ   *-3017A
                        1237
003.017  174            1238  WNF       MOV     A,H
003.020  315 024 003    1239            CALL    WNB
003.023  175            1240            MOV     A,L
                        1241  *         JMP     WNB             WRITE NEXT BYTE


                        1243  **        WNB - WRITE BYTE
                        1244  *
                        1245  *         WNB WRITES THE NEXT BYTE TO THE CASSETTE TAPE.
                        1246  *
                        1247  *         ENTRY   (A) = BYTE
                        1248  *         EXIT    NONE.
                        1249  *         USES    F
                        1250
000.000                 1251            ERRNZ   *-3024A
                        1252
003.024  365            1253  WNB       PUSH    PSW
003.025  315 252 002    1254  WNB1      CALL    TPXIT           CHECK FOR *, READ STATUS
003.030  346 001        1255            ANI     USR.TXR
003.032  312 025 003    1256            JZ      WNB1            IF MORE TO GO
003.035  076 021        1257            MVI     A,UCI.ER+UCI.TE ENABLE TRANSMITTER
003.037  323 371        1258            OUT     OP.TPC          TURN ON TAPE
003.041  361            1259            POP     PSW
003.042  323 370        1260            OUT     OP.TPD          OUTPUT DATA
003.044  303 347 002    1261            JMP     CRC             COMPUTE CRC
```

```
                          1265  **      LRA - LOCATE REGISTER ADDRESS.
                          1266  *
                          1267  *       ENTRY   NONE.
                          1268  *       EXIT    (A) = REGISTER INDEX
                          1269  *               (H,L) = STORAGE ADDRESS
                          1270  *               (D,E) = (0,A)
                          1271  *       USES    A,D,E,H,L,F
                          1272
                          1273
     000.000              1274          ERRNZ   *-3047A
                          1275
     003.047  072 005 040 1276  LRA     LDA     REGI
     003.052  137         1277  LRA.    MOV     E,A
     003.053  026 000     1278          MVI     D,0
     003.055  052 035 040 1279          LHLD    REGPTR
     003.060  031         1280          DAD     D               (DE) = (REGPTR)+(REGI)
     003.061  311         1281          RET


                          1283  **      IOA - INPUT OCTAL ADDRESS.
                          1284  *
                          1285  *       ENTRY   (H,L) = ADDRESS OF RECEPTION DOUBLE BYTE.
                          1286  *               (D) = TERMINATING CHARACTER
                          1287  *       EXIT    NONE
                          1288  *       USES    A,D,E,H,L,F
                          1289
                          1290
     000.000              1291          ERRNZ   *-3062A
                          1292
     003.062  303 176 005 1293  IOA     JMP     IOA1
     003.065  000         1294          NOP                     RETAIN H8 ORG


                          1296  **      IOB - INPUT OCTAL BYTE.
                          1297  *
                          1298  *       READ ONE OCTAL BYTE FROM THE KEYSET.
                          1299  *
                          1300  *       ENTRY   (H,L) = ADDRESS OF BYTE TO HOLD VALUE
                          1301  *               'C' SET IF FIRST DIGIT IN (A)
                          1302  *       EXIT    NONE
                          1303  *       USES    A,D,E,H,L,F
                          1304
                          1305
     000.000              1306          ERRNZ   *-3066A
                          1307
     003.066  066 000     1308  IOB     MVI     M,0             ZERO OUT OLD VALUE
     003.070  324 262 003 1309  IOB1    CNC     RCC             READ CONSOLE CHARACTER
                          1310
                          1311  *       SEE IF CHARACTER IS A VALID OCTAL VALUE
                          1312  *
     003.073  376 060     1313          CPI     '0'             LESS THAN ZERO?
     003.075  332 135 003 1314          JC      IOB2            IF (A) < 0, SEE IF A TERMINATING CHARACTER
```

```
003.100  376 070      1315          CPI     '8'             GREATER THAN 7?
003.102  322 070 003  1316          JNC     IOB1            IF TOO LARGE, TRY AGAIN
                      1317
                      1318  *       HAVE AN OCTAL DIGIT
                      1319  *
003.105  315 302 003  1320          CALL    WCC             ECHO CHARACTER
003.110  346 007      1321          ANI     00000111B       MASK FOR BINARY VALUE
003.112  137          1322          MOV     E,A             (E) = VALUE
003.113  176          1323          MOV     A,M             GET OLD VALUE
003.114  007          1324          RLC                     SHIFT 3
003.115  007          1325          RLC
003.116  007          1326          RLC
003.117  303 126 003  1327          JMP     IOB1.5          JUMP AROUND AN H88/H89 TO H8 FAKE ROUTINE
                      1328
                      1329  **      FAKE OUT ROUTINE FOR CALLERS OF *DOD* FROM THE H8 FRONT PANEL
                      1330
                      1331
000.000               1332          ERRNZ   *-3122A
                      1333
003.122  043          1334  DOD     INX     H
003.123  043          1335          INX     H
003.124  043          1336          INX     H
003.125  311          1337          RET
                      1338
                      1339
                      1340  *       CONTINUE
                      1341
003.126  346 370      1342  IOB1.5  ANI     11111000B       TOSS OLD LSB DIGIT
003.130  263          1343          ORA     E               REPLACE WITH NEW VALUE
003.131  167          1344          MOV     M,A
003.132  303 070 003  1345          JMP     IOB1            INPUT ANOTHER CHARACTER
                      1346
                      1347  *       CHECK FOR A CARRIAGE RETURN TO TERMINATE BYTE
                      1348  *
003.135  376 015      1349  IOB2    CPI     A.CR            CARRIAGE RETURN?
003.137  310          1350          RZ                      RETURN IF CARRIAGE RETURN       /JWT 790507/
003.140  257          1351          XRA     A               CLEAR CARRY                     /JWT 790507/
003.141  030 325      1352          JR      IOB1            GET A NEW CHARACTER             /JWT 790507/
                      1353
                      1433          LON     L
```

```
                        1437  **      RCK - READ CONSOLE KEYPAD
                        1438  *
                        1439  *       RCK IS CALLED TO READ A KEYSTROKE FROM THE CONSOLE FRONT PANEL KEYPAD.
                        1440  *       SINCE THE H88/89 DOES NOT HAVE A FRONT PANEL, THIS ROUTINE IS PROVIDED
                        1441  *       ONLY TO MAINTAIN COMPATIBILITY WITH PAM-8.
                        1442  *       RCK WILL IMMEDIATELY RETURN WITH A VALUE OF 0 (ZERO) IN THE ACCUMULATOR.
                        1443  *
                        1444  *       ENTRY   NONE
                        1445  *       EXIT    (A) = 0
                        1446  *       USES    A,F
                        1447
                        1448  *       RCK MUST HAVE SAME ENTRY AS RCK IN PAM-8
   000.000              1449          ERRNZ   *-3260A
                        1450
   003.260              1451  RCK     EQU     *
                        1452
   003.260  257         1453          XRA     A
   003.261  311         1454          RET
                        1455
```

```
                        1459  **      RCC - READ CONSOLE CHARACTER.
                        1460  *
                        1461  *       RCC IS CALLED TO READ A KEYSTROKE FROM THE CONSOLE.
                        1462  *       IF A RUBOUT/DELETE IS RECEIVED, EXIT IS TO *ERROR*.
                        1463  *
                        1464  *       ENTRY   NONE
                        1465  *       EXIT    TO ERROR  - IF A DELETE OR RUBOUT IS ENCOUNTERED
                        1466  *               TO CALLER - WHEN A KEY IS HIT
                        1467  *                   (A) =  ASCII KEY VALUE
                        1468  *       USES    A;F
                        1469
                        1470
                        1471
 003.262                1472  RCC     EQU     *
                        1473
 003.262  333 355       1474  RCC1    IN      SC.ACE+UR.LSR   INPUT ACE LINE STATUS REGISTER
 003.264  346 001       1475          ANI     UC.DR           SEE IF THERE IS A DATA READY
 003.266  050 372       1476          JR      Z,RCC1
                        1477
 003.270  333 350       1478          IN      SC.ACE+UR.RBR   ELSE, INPUT CHARACTER
 003.272  346 177       1479          ANI     01111111B       TOSS ANY PARITY
 003.274  376 177       1480          CPI     A.DEL
 003.276  312 322 000   1481          JZ      ERROR           IF RUBOUT, EXIT TO ERROR
                        1482
 003.301  311           1483          RET                     ELSE, EXIT TO CALLER


                        1485  **      WCC - WRITE CONSOLE CHARACTER
                        1486  *
                        1487  *       WRITE A CHARACTER TO THE CONSOLE UART PORT
                        1488  *
                        1489  *       ENTRY   (A) = ASCII CHARACTER TO OUTPUT
                        1490  *       EXIT    NONE
                        1491  *       USES    NONE
                        1492
                        1493
 003.302  365           1494  WCC     PUSH    PSW             SAVE CHARACTER
 003.303  333 355       1495  WCC1    IN      SC.ACE+UR.LSR   INPUT ACE STATUS
 003.305  346 040       1496          ANI     UC.THE          SEE IF TRANSMITTER HOLDING REGISTER IS EMPTY
 003.307  050 372       1497          JR      Z,WCC1
                        1498
 003.311  361           1499          POP     PSW             GET CHARACTER
 003.312  323 350       1500          OUT     SC.ACE+UR.THR           OUTPUT TO CONSOLE
 003.314  311           1501          RET
                        1547          LON     L
```

```
                        1550  **      IO ROUTINES TO BE COPIED INTO AND USED IN RAM.
                        1551  *
                        1552  *       MUST CONTINUE TO 3777A FOR PROPER COPY.
                        1553  *       THE TABLE MUST ALSO BE BACKWARDS TO THE FINAL RAM
                        1554
    000.000             1555          ERRNZ   4000A-7-*
                        1556
    003.371             1557  PRSROM  EQU     *
    003.371  001       1558          DB      1               REFIND
    003.372  000       1559          DB      0               CTLFLG
    003.373  000       1560          DB      0               .MFLAG
    003.374  000       1561          DB      0               DSPMOD
    003.375  000       1562          DB      0               DSPROT
    003.376  012       1563          DB      10              REGI
    003.377  311       1564          DB      MI.RET
                        1565
    000.000             1566          ERRNZ *-4000A
                        1567
```

```
                                1570  ***     INITOX          EXTENSION OF INIT0 TO SUPPORT H88
                                1571
004.000  076 002               1572  INIT0X  MVI     A,00000010B
004.002  323 362               1573          OUT     H88.CTL
                                1574
                                1575  *       SET UP ACE FOR CONSOLE COMMUNICATIONS
                                1576  *
004.004  076 200               1577          MVI     A,UC.DLA        SET DIVISOR LATCH ACCESS BIT
004.006  323 353               1578          OUT     SC.ACE+UR.LCR
004.010  041 075 004           1579          LXI     H,BRTAB         (H,L) = BEGINNING OF BAUD RATE TABLE
004.013  333 362               1580          IN      H88.SW          INPUT SWITCHES FOR DESIRED BAUD RATE
004.015  346 300               1581          ANI     H88S.BR         MASK FOR BAUD RATE SWITCHES ONLY
004.017  017                   1582          RRC                     SHIFT FOR A *2 FOR TABLE
004.020  017                   1583          RRC
004.021  017                   1584          RRC
004.022  017                   1585          RRC
004.023  017                   1586          RRC
004.024  205                   1587          ADD     L               ADD DISPLACEMENT FROM BEGINNING OF TABLE
004.025  157                   1588          MOV     L,A
004.026  176                   1589          MOV     A,M             GET MSB OF DIVISOR
004.027  323 351               1590          OUT     SC.ACE+UR.DLM
004.031  043                   1591          INX     H               GET LSB
004.032  176                   1592          MOV     A,M
004.033  323 350               1593          OUT     SC.ACE+UR.DLL
004.035  076 003               1594          MVI     A,UC.8BW        SET 8 BITS, 1 STOP BIT, NO PARITY
004.037  323 353               1595          OUT     SC.ACE+UR.LCR
004.041  076 000               1596          MVI     A,0             SET NO INTERRUPTS
004.043  323 351               1597          OUT     SC.ACE+UR.IER
                                1598
                                1599  *       WAIT A WHILE TO ALLOW THE CONSOLE RESET TO FINISH SO IT CAN
                                1600  *       ACCEPT THE FIRST PROMPT
                                1601  *
004.045  001 000 065           1602          LXI     B,65000A        APPROX. 100 MS
004.050  015                   1603  INIT0X1 DCR     C
004.051  302 050 004           1604          JNZ     INIT0X1
                                1605
004.054  005                   1606          DCR     B
004.055  302 050 004           1607          JNZ     INIT0X1
                                1608
                                1609  *       INPUT SWITCH TO SEE IF TO BEGIN OPERATION OR MEMORY TEST
                                1610  *
004.060  333 362               1611          IN      H88.SW          GET SWITCHES
004.062  346 040               1612          ANI     H88S.M          MASK FOR MEMORY TEST ONLY
004.064  312 116 007           1613          JZ      DYMEM           IF TO PERFORM MEMORY TESTS
                                1614
                                1615  *       REPLACE WHAT WAS ORIGINALLY AT THE JUMP WHICH GOT US HERE
                                1616  *
004.067  021 371 003           1617          LXI     D,PRSROM        (DE) = ROM COPY OF PRS CODE
004.072  303 003 000           1618          JMP     INIT0.0         RETURN TO ORIGINAL CODE
```

```
                        1620  **      BRTAB - BAUD RATE DIVISOR TABLE
                        1621  *
    004.075             1622  BRTAB   EQU     *
                        1623
    004.075  000 014    1624  BR96    DB      0,12            9600 BAUD
    004.077  000 006    1625  BR19.2  DB      0,6           19,200 BAUD
    004.101  000 003    1626  BR38.4  DB      0,3           38,400 BAUD
    004.103  000 002    1627  BR56.0  DB      0,3           56,000 BAUD
                        1628
    000.004             1629          SET     *!256
    000.000             1630          ERRNZ   BRTAB!256-.   TABLE MUST BE IN ONE PAGE


                        1632  ***     SAVALLX - SAVALL EXTENSION TO MAKE ROOM FOR A JUMP TO THE NMI HANDLER
                        1633
    004.105             1634  SAVALLX EQU     *             REPLACE OLD CODE
    004.105  345        1635          PUSH    H             SET ON STACK AS 'REGISTER'
    004.106  325        1636          PUSH    D             SET RETURN ADDRESS
    004.107  021 011 040 1637          LXI     D,CTLFLG
    004.112  032        1638          LDAX    D
    004.113  303 151 000 1639          JMP     SAVALLR       RETURN TO OLD CODE
```

```
                        1642  ****    NMI - NON MASKABLE INTERRUPT
                        1643  *
                        1644  *       NMI IS USED AS THE TRAP FOR ALL ILLEGAL PORT REQUESTS
                        1645  *
                        1646  *       PORT ADDRESSES TRAPPED ARE:
                        1647  *
                        1648  *                       IN 360Q FRONT PANEL KEYBOARD INPUT
                        1649  *                      OUT 360Q FRONT PANEL CONTROL
                        1650  *                      OUT 361Q FRONT PANEL DISPLAY CONTROL
                        1651  *                   IN/OUT 372Q CONSOLE DATA FOR AN 8251A
                        1652  *                      OUT 373Q CONSOLE CONTROL FOR AN 8251A
                        1653  *
                        1654  *
                        1655  *              THESE PORT REQUESTS ARE RESPONDED TO AS FOLLOWS:
                        1656  *
                        1657  *                       IN 360Q RETURNS WITH (A) = 377Q TO SHOW THAT
                        1658  *                              NO FRONT PANEL SWITCHES ARE PRESSED
                        1659  *
                        1660  *                       OUT 360Q MOVES BIT 6 (CB.CLI) TO BIT 1, AND
                        1661  *                              BIT 4 (CB.SSI) INVERTED, TO BIT 0, AND
                        1662  *                              OUTPUTS THESE BITS TO PORT 362Q TO
                        1663  *                              CONTROL THE CLOCK AND SINGLE STEP INTERRUPTS
                        1664  *
                        1665  *                      OUTPUTS TO 361Q, 372Q, AND 373Q JUST RETURN
                        1666  *
                        1667  *                      INPUTS FROM 361Q, 372Q, AND 373Q RETURN WITH (A) = 0
                        1668  *                              TO INDICATE AN EMPTY BUSS
                        1669  *
                        1670  *
                        1671  *              ENTRY   NONE
                        1672  *
                        1673  *              EXIT    NONE
                        1674  *
                        1675  *              USES    (A) ONLY IF "FAKING" AN INPUT
                        1676
                        1677
004.116  343           1678  NMI     XTHL                    GET RETURN ADDRESS FROM STACK
004.117  042 064 040    1679          SHLD    NMIRET          SAVE FOR LATER USE
004.122  343           1680          XTHL                    PUT RETURN ADDRESS BACK ON STACK
                        1681
004.123  345           1682          PUSH    H               SAVE REGISTERS
004.124  305           1683          PUSH    B
004.125  365           1684          PUSH    PSW
004.126  107           1685          MOV     B,A             SAVE (A) PRIOR TO I/O
004.127  052 064 040    1686          LHLD    NMIRET          GET RETURN ADDRESS
004.132  053           1687          DCX     H               BACK UP TO PORT # WHICH GOT US HERE
004.133  176           1688          MOV     A,M             GET PORT #
                        1689
004.134  376 360       1690          CPI     360Q            PORT 360?
004.136  312 202 004    1691          JZ      NMI1            IF PORT WAS 360Q
                        1692
                        1693  *       PORT REFERENCED WAS 361Q, 372Q, OR 373Q
                        1694  *
004.141  376 361       1695          CPI     361Q            MAKE SURE PORT IS LEGAL
004.143  312 160 004    1696          JZ      NMI0.5          IF LEGAL
                        1697
```

```
004.146  376 372        1698            CPI     372Q
004.150  312 160 004    1699            JZ      NMI0.5
                        1700
004.153  376 373        1701            CPI     373Q
004.155  302 251 004    1702            JNZ     NMI2.5          IF NONE OF THE ABOVE, EXIT
                        1703
004.160  053            1704  NMI0.5    DCX     H               POINT TO IN/OUT INSTRUCTION
004.161  176            1705            MOV     A,M             SEE IF INPUT OR OUTPUT
004.162  376 323        1706            CPI     MI.OUT
004.164  312 251 004    1707            JZ      NMI2.5          IF OUTPUT, JUST EXIT
                        1708
004.167  376 333        1709            CPI     MI.IN
004.171  302 251 004    1710            JNZ     NMI2.5          IF NOT INPUT EITHER, ILLEGAL SO EXIT
                        1711
004.174  361            1712            POP     PSW             RESTORE FLAGS
004.175  076 000        1713            MVI     A,0             ELSE, RETURN LIKE AN EMPTY BUSS
004.177  303 252 004    1714            JMP     NMI3            EXIT
                        1715
004.202  053            1716  NMI1      DCX     H               POINT TO IN/OUT INSTRUCTION
004.203  176            1717            MOV     A,M             GET I/O INSTRUCTION
004.204  376 333        1718            CPI     MI.IN           INPUT?
004.206  302 217 004    1719            JNZ     NMI1.5          IF NOT "IN"
                        1720
004.211  361            1721            POP     PSW             RESTORE FLAGS
004.212  076 377        1722            MVI     A,11111111B     SHOW "NO KEYS PRESSED"
004.214  303 252 004    1723            JMP     NMI3            EXIT
                        1724
004.217  376 323        1725  NMI1.5    CPI     MI.OUT          MAKE SURE INTRUCTION IS AN "OUT"
004.221  302 251 004    1726            JNZ     NMI2.5          IF NOT
                        1727
004.224  170            1728  NMI2      MOV     A,B             GET OUTPUT DATA AGAIN
004.225  346 100        1729            ANI     CB.CLI          MOVE CLOCK INFO TO BIT 1
004.227  017            1730            RRC
004.230  017            1731            RRC
004.231  017            1732            RRC
004.232  017            1733            RRC
004.233  017            1734            RRC
004.234  117            1735            MOV     C,A             SAVE IN C
004.235  170            1736            MOV     A,B             GET OUTPUT DATA AGAIN
004.236  346 020        1737            ANI     CB.SST          GET SINGLE STEP INFO
004.240  017            1738            RRC                     MOVE TO BIT 0
004.241  017            1739            RRC
004.242  017            1740            RRC
004.243  017            1741            RRC
004.244  241            1742            ORA     C               ADD TO CLOCK DATA
004.245  356 001        1743            XRI     00000001B       INVERT THIS BIT PRIOR TO OUTPUT
004.247  323 362        1744            OUT     H88.CTL         SET IN HARDWARE
                        1745
004.251  361            1746  NMI2.5    POP     PSW             RESTORE (A,F)
                        1747
004.252  301            1748  NMI3      POP     B
004.253  341            1749            POP     H
                        1750  *         RETN                    Z80 RETURN FROM NMI
004.254  355 105        1751            DB      355Q,105Q
```

```
                                1754  **      BOOT HDOS ENTRY POINT FOR H88
                                1755  *
                                1756  *       ENTRY   NONE
                                1757  *       EXIT    TO HDOS BOOT ROM
                                1758  *       USES    ALL
                                1759
     004.256  041 234 006      1760  BOOT    LXI     H,MSG.BT        COMPLETE BOOT MESSAGE
     004.261  315 100 006      1761          CALL    TYPMSG
     004.264  315 003 006      1762          CALL    WCR             WAIT FOR A CARRIAGE RETURN
     004.267  076 012          1763          MVI     A,10
     004.271  315 052 003      1764          CALL    LRA.            GET LOCATION OF USER PC
     004.274  021 000 030      1765          LXI     D,ROMDD         SET ITS VALUE TO THE BOOT ROM
     004.277  163              1766          MOV     M,E
     004.300  043              1767          INX     H
     004.301  162              1768          MOV     M,D
                               1769
                               1770  *       TELL USER TO 'TYPE SPACES TO DETERMINE BAUD RATE'
                               1771  *
     004.302  041 122 006      1772          LXI     H,MSG.SP
     004.305  315 100 006      1773          CALL    TYPMSG
                               1774
     004.310  303 063 000      1775          JMP     GO.             DO IT


                               1777  **      SWMEM - SET UP FOR WMEM TO DUMP A CASSETTE FROM THE MONITOR LEVEL
                               1778  *
                               1779  *       SWMEM ALLOWS THE USER TO EITHER ENTER A NEW STARTING AND
                               1780  *       ENDING ADDRESS FOR THIS DUMP OR USE THE ADDRESS OF THE
                               1781  *       PREVIOUS LOAD OR DUMP.  THE PREVIOUS ADDRESSES ARE USED IF
                               1782  *       THE FIRST CHARACTER IS AN ASCII CARRIAGE RETURN.  IF THE
                               1783  *       FIRST CHARACTER IS AN OCTAL CHARACTER, BOTH BEGINNING AND
                               1784  *       ENDING ADDRESSES MUST BE ENTERED SEPARATED BY A DASH AND
                               1785  *       FOLLOWED BY A CARRIAGE RETURN
                               1786  *
                               1787  *
                               1788  *       ENTRY   USER PC VALUE ON STACK = PROGRAM START ADDRESS FOR THIS TAPE
                               1789  *       EXIT    TO WMEM
                               1790  *       USES    ALL
                               1791
                               1792
     004.313  041 174 006      1793  SWMEM   LXI     H,MSG.DMP       COMPLETE DUMP MESSAGE
     004.316  315 100 006      1794          CALL    TYPMSG
     004.321  315 150 005      1795          CALL    IROC            INPUT FIRST CHARACTER
     004.324  322 351 004      1796          JNC     SWMEM4          IF FIRST CHARACTER IS OCTAL
                               1797
     004.327  041 001 040      1798          LXI     H,START+1       ELSE, INPUT STARTING ADDRESS
     004.332  026 055          1799          MVI     D,'-'           FIRST BYTE MUST END WITH A DASH
     004.334  315 062 003      1800          CALL    IOA
     004.337  041 025 040      1801  SWMEM2  LXI     H,ABUSS+1       ENTER ENDING ADDRESS
     004.342  067              1802          STC                     SHOW NO CHARACTER IN (A)
     004.343  077              1803          CMC
     004.344  026 015          1804          MVI     D,A.CR          LAST CHARACTER MUST BE A RETURN
     004.346  315 062 003      1805          CALL    IOA
     004.351  076 012          1806  SWMEM4  MVI     A,10            GET USER PC VALUE FOR DISPLAY
```

```
004.353 315 052 003 1807          CALL    LRA.
004.356 136         1808          MOV     E,M
004.357 043         1809          INX     H
004.360 126         1810          MOV     D,M
004.361 353         1811          XCHG                    (H,L) = USER PC VALUE
004.362 315 313 005 1812          CALL    TOA             TYPE OCTAL ADDRESS
004.365 303 374 001 1813          JMP     WMEM            DO THE DUMP
                    1814
                    1816  **      SUBM - SUBSTITUTE MEMORY
                    1817  *
                    1818  *       SUBM INPUTS A MEMORY ADDRESS FROM THE CONSOLE AND THEN DISPLAYS
                    1819  *       THAT ADDRESS AND ITS CONTENTS.  IF A CARRIAGE RETURN IS THEN TYPED,
                    1820  *       CONTROL RETURNS TO THE MONITOR.  IF A SPACE IS TYPED, THE NEXT
                    1821  *       MEMORY LOCATION AND CONTENTS ARE DISPLAYED.  IF A MINUS SIGN IS
                    1822  *       TYPED, THE PREVIOUS MEMORY LOCATION AND CONTENTS ARE DISPLAYED.
                    1823  *       IF AN OCTAL CHARACTER IS TYPED, A BYTE IS ENTERED AN PLACED AT THE
                    1824  *       CURRENT MEMORY LOCATION.
                    1825  *
                    1826  *
                    1827  *       ENTRY   NONE
                    1828  *       EXIT    NONE
                    1829  *       USES    A,E,H,L,F
                    1830
                    1831
004.370 041 201 006 1832  SUBM    LXI     H,MSG.SUB       COMPLETE SUBSTITUTE MESSAGE
004.373 315 100 006 1833          CALL    TYPMSG
004.376 315 150 005 1834          CALL    IROC            INPUT FIRST CHARACTER
005.001 320         1835          RNC                     IF A RETURN, EXIT
                    1836
005.002 041 003 040 1837          LXI     H,IOWRK+1       ELSE, INPUT STARTING ADDRESS
005.005 026 015     1838          MVI     D,A.CR          ENDING WITH A RETURN
005.007 315 062 003 1839          CALL    IOA
005.012 353         1840          XCHG                    (H,L) = INPUT ADDRESS
                    1841
005.013 315 313 005 1842  SUBM1   CALL    TOA             TYPE CRLF, ADDRESS, AND A SPACE
005.016 176         1843          MOV     A,M             GET MEMORY CONTENTS FOR DISPLAY
005.017 315 343 005 1844          CALL    TOB
005.022 076 040     1845          MVI     A,' '           SPACE
005.024 315 302 003 1846          CALL    WCC
                    1847
005.027 315 301 005 1848  SUBM2   CALL    IOC             INPUT FIRST CHARACTER
005.032 322 075 005 1849          JNC     SUBM7           IF FIRST CHARACTER IS OCTAL
                    1850
005.035 376 040     1851          CPI     ' '             SPACE?
005.037 302 046 005 1852          JNZ     SUBM4           IF NOT A SPACE
                    1853
005.042 043         1854  SUBM3   INX     H               POINT TO NEXT ADDRESS
005.043 303 013 005 1855          JMP     SUBM1           DISPLAY NEXT
                    1856
005.046 376 055     1857  SUBM4   CPI     '-'             MINUS?
005.050 302 062 005 1858          JNZ     SUBM6           IF NOT
                    1859
```

```
005.053  315 302 003  1860  SUBM5   CALL    WCC             ECHO HYPHEN
005.056  053         1861          DCX     H               POINT TO PREVIOUS ADDRESS
005.057  303 013 005  1862          JMP     SUBM1           DISPLAY PREVIOUS
                      1863
005.062  376 015      1864  SUBM6   CPI     A.CR            RETURN?
005.064  310         1865          RZ                      IF RETURN, EXIT
                      1866
005.065  076 007      1867          MVI     A,A.BEL         ELSE, DING BELL
005.067  315 302 003  1868          CALL    WCC
005.072  303 027 005  1869          JMP     SUBM2           TRY AGAIN
                      1870
005.075  066 000      1871  SUBM7   MVI     M,0             ZERO BYTE TO BE BUILT
                      1872
005.077  315 302 003  1873  SUBM8   CALL    WCC             ECHO OCTAL CHARACTER
005.102  346 007      1874          ANI     00000111B       GET BINARY VALUE
005.104  137         1875          MOV     E,A             SAVE PARTIAL
005.105  176         1876          MOV     A,M             GET CURRENT
005.106  007         1877          RLC                     MAKE ROOM FOR NEW CHARACTER
005.107  007         1878          RLC
005.110  007         1879          RLC
005.111  346 370      1880          ANI     11111000B       TOSS PREVIOUS LSB
005.113  263         1881          ORA     E               ADD NEW
005.114  167         1882          MOV     M,A             SAVE NEW TOTAL
005.115  315 301 005  1883  SUBM9   CALL    IOC             INPUT NEXT CHARACTER
005.120  322 077 005  1884          JNC     SUBM8           IF OCTAL
                      1885
005.123  376 040      1886          CPI     ' '             SPACE?
005.125  312 042 005  1887          JZ      SUBM3           IF SPACE, DISPLAY NEXT BYTE
                      1888
005.130  376 055      1889          CPI     '-'             MINUS?
005.132  312 053 005  1890          JZ      SUBM5           IF MINUS, DISPLAY PREVIOUS
                      1891
005.135  376 015      1892          CPI     A.CR            RETURN?
005.137  310         1893          RZ                      IF RETURN, EXIT
                      1894
005.140  076 007      1895          MVI     A,A.BEL         ELSE, DING BELL
005.142  315 302 003  1896          CALL    WCC
005.145  303 115 005  1897          JMP     SUBM9           TRY AGAIN


                      1899  **      IROC - INPUT A RETURN OR AN OCTAL CHARACTER
                      1900  *
                      1901  *       IROC INPUTS A CHARACTER FROM THE CONSOLE AND WAITS UNTIL IT
                      1902  *       RECEIVES EITHER A VALID OCTAL CAHRACTER OR A CARRIAGE RETURN
                      1903  *
                      1904  *       ENTRY   NONE
                      1905  *       EXIT    (A) = INPUT CHARACTER
                      1906  *               'C' = SET IF CHARACTER IS OCTAL
                      1907  *       USES    A,F
                      1908
                      1909
005.150  315 262 003  1910  IROC    CALL    RCC             INPUT CHARACTER
005.153  376 015      1911          CPI     A.CR            RETURN?
005.155  310         1912          RZ                      IF A CR
```

```
                          1913
005,156  376 060          1914          CPI     '0'             < 0?
005.160  332 166 005      1915          JC      IROC1           IF < OCTAL
                          1916
005.163  376 070          1917          CPI     '8'             > 8?
005.165  330              1918          RC                      IF OCTAL
                          1919
005.166  076 007          1920  IROC1   MVI     A,A.BEL         ELSE, RING BELL
005.170  315 302 003      1921          CALL    WCC
005.173  303 150 005      1922          JMP     IROC            TRY AGAIN


                          1924  **      IOA1 - INPUT OCTAL ADDRESS
                          1925  *
                          1926  *       IOA1 IS A CONTINUATION OF *IOA* AND INPUTS A SPLIT OCTAL ADDRESS
                          1927  *       WITHOUT REQUIRING LEADING ZEROS
                          1928  *
                          1929  *       ENTRY   (H,L) = ADDRESS + 1 WHERE INPUT ADDRESS IS TO BE PLACED
                          1930  *               (A) = FIRST OCTAL CHARACTER IF 'C' IS SET
                          1931  *       EXIT    (D,E) = INPUT ADDRESS
                          1932  *               (A) = LAST INPUT CHARACTER
                          1933  *       USES    A,D,E,H,L,F
                          1934
                          1935
005.176  305              1936  IOA1    PUSH    B               SAVE (B,C)
005.177  102              1937          MOV     B,D             (B) = TERMINATION CHARACTER
005.200  345              1938          PUSH    H               SAVE ADDRESS WHERE INPUT IS TO BE PLACED
005.201  041 000 000      1939          LXI     H,0             SET NEW VALUE TO ZERO
005.204  324 262 003      1940  IOA2    CNC     RCC             IF CARRY SET, FIRST CHARACTER IS IN ACC
005.207  376 060          1941          CPI     '0'             MAKE SURE CHARACTER IS OCTAL
005.211  332 242 005      1942          JC      IOA3            IF < OCTAL
                          1943
005.214  376 070          1944          CPI     '8'
005.216  322 242 005      1945          JNC     IOA3            IF > OCTAL
                          1946
005.221  315 302 003      1947          CALL    WCC             ECHO OCTAL CHARACTER
005.224  346 007          1948          ANI     00000111B       GET BINARY VALUE
005.226  365              1949          PUSH    PSW             SAVE NEW CHARACTER VALUE
005.227  051              1950          DAD     H               SHIFT THREE TO MAKE ROOM FOR NEW CHARACTER
005.230  051              1951          DAD     H
005.231  051              1952          DAD     H
005.232  365              1953          PUSH    PSW             SAVE CARRY FROM DAD
005.233  321              1954          POP     D               SAVE FLAG RESULT IN E
005.234  361              1955          POP     PSW             RETURN NEW CHARACTER VALUE TO (A)
005.235  205              1956          ADD     L
005.236  157              1957          MOV     L,A
005.237  303 204 005      1958          JMP     IOA2            SEE IF MORE CHARACTERS
                          1959
005.242  270              1960  IOA3    CMP     B               TERMINATING CHARACTER?
005.243  312 260 005      1961          JZ      IOA4            IF EQUAL
                          1962
005.246  076 007          1963          MVI     A,A.BEL         ELSE, DING BELL
005.250  315 302 003      1964          CALL    WCC
005.253  067              1965          STC                     TRY AGAIN
```

```
005.254  077            1966           CMC
005.255  303 204 005     1967           JMP     IOA2
                         1968
                         1969  *        END OF INPUT, PUT VALUE IN MEMORY AND EXIT
                         1970
005.260  315 302 003     1971  IOA4     CALL    WCC             ECHO CHARACTER
005.263  127             1972           MOV     D,A             LAST CHARACTER TO D
005.264  325             1973           PUSH    D
005.265  361             1974           POP     PSW             (PSW) = RESULT OF DAD
005.266  174             1975           MOV     A,H             MAKE (H) INTO SPLIT OCTAL
005.267  037             1976           RAR
005.270  147             1977           MOV     H,A
005.271  172             1978           MOV     A,D             RESTORE LAST INPUT CHARACTER
005.272  353             1979           XCHG                    (D,E) = INPUT ADDRESS
005.273  341             1980           POP     H               (H,L) = LOCATION TO PLACE THIS ADDRESS
005.274  162             1981           MOV     M,D
005.275  053             1982           DCX     H
005.276  163             1983           MOV     M,E
005.277  301             1984           POP     B               RESTORE (B,C)
005.300  311             1985           RET
```

```
                         1987  **       IOC - INPUT OCTAL CHARACTER
                         1988  *
                         1989  *
                         1990  *        ENTRY   NONE
                         1991  *        EXIT    (A) = INPUT CHARACTER
                         1992  *                'C' = SET IF CHARACTER NOT OCTAL
                         1993  *        USES    A,F
                         1994
                         1995
005.301  315 262 003     1996  IOC      CALL    RCC             INPUT CHARACTER
005.304  376 060         1997           CPI     '0'
005.306  330             1998           RC                      IF CHARACTER < OCTAL
                         1999
005.307  376 070         2000           CPI     '8'             CHARACTER > OCTAL?
005.311  077             2001           CMC                     'C' IF GREATER THAN
005.312  311             2002           RET
```

```
                         2004  **       TOA - TYPE OCTAL ADDRESS
                         2005  *
                         2006  *        TOA OUTPUTS TO THE CONSOLE A CRLF, THE SPECIFIED ADDRESS AND A SPACE
                         2007  *
                         2008  *        ENTRY   (H,L) = ADDRESS TO BE DISPLAYED
                         2009  *        EXIT    NONE
                         2010  *        USES    A,B,C,F
                         2011
                         2012
005.313  076 015         2013  TOA      MVI     A,A.CR          CRLF
005.315  315 302 003     2014           CALL    WCC
005.320  076 012         2015           MVI     A,A.LF
```

```
005.322 315 302 003  2016          CALL    WCC
                     2017
005.325 174          2018  TOA.    MOV     A,H             ADDRESS
005.326 315 343 005  2019          CALL    TOB
005.331 175          2020          MOV     A,L
005.332 315 343 005  2021          CALL    TOB
                     2022
005.335 076 040      2023          MVI     A,' '           SPACE
005.337 315 302 003  2024          CALL    WCC
005.342 311          2025          RET


                     2027  **      TOB - TYPE OCTAL BYTE
                     2028  *
                     2029  *       TOB OUTPUTS TO THE CONSOLE IN OCTAL, THE BYTE IN A
                     2030  *
                     2031  *       ENTRY   (A) = BYTE TO BE OUTPUT
                     2032  *       EXIT    NONE
                     2033  *       USES    A,F
                     2034
                     2035
005.343 305          2036  TOB     PUSH    B
005.344 006 002      2037          MVI     B,2             NUMBER OF CHARACTERS - 1
005.346 117          2038          MOV     C,A             SAVE ORIGINAL BYTE
005.347 267          2039          ORA     A               ASSURE 'C' = ZERO
005.350 037          2040          RAR
005.351 037          2041          RAR                     SHIFT TOP BYTE TO LSB
005.352 037          2042          RAR
005.353 037          2043  TOB1    RAR                     SHIFT MIDDLE BYTE TO LSB
005.354 037          2044          RAR
005.355 037          2045          RAR
005.356 346 007      2046          ANI     00000111B       MASK FOR HALF ASCII
005.360 366 060      2047          ORI     00110000B       MAKE WHOLE ASCII
005.362 315 302 003  2048          CALL    WCC             OUTPUT TO CONSOLE
005.365 171          2049          MOV     A,C             GET ORIGINAL BYTE
005.366 005          2050          DCR     B
005.367 302 353 005  2051          JNZ     TOB1            IF SECOND BYTE STILL NEEDS TO BE OUTPUT
                     2052
005.372 346 007      2053          ANI     00000111B       ELSE, OUTPUT LAST CHARACTER
005.374 366 060      2054          ORI     00110000B
005.376 315 302 003  2055          CALL    WCC
006.001 301          2056          POP     B
006.002 311          2057          RET


                     2059  **      WCR - WAIT FOR A CARRIAGE RETURN
                     2060  *
                     2061  *       WCR INPUTS CHARACTERS FROM THE CONSOLE UNTIL A CARRIAGE RETURN
                     2062  *       IS RECEIVED AND THEN ECHOS A CRLF
                     2063  *
                     2064  *
                     2065  *       ENTRY   NONE
```

```
                          2066  *       EXIT    NONE
                          2067  *       USES    A;F
                          2068
                          2069
    006.003  315 262 003  2070  WCR     CALL    RCC             INPUT CHARACTER
    006.006  376 015      2071          CPI     A;CR
    006.010  302 003 006  2072          JNZ     WCR             IF NOT A CR
                          2073
    006.013  315 302 003  2074          CALL    WCC             ELSE, ECHO CR
    006.016  076 012      2075          MVI     A;A.LF          LINE FEED
    006.020  315 302 003  2076          CALL    WCC
    006.023  311          2077          RET


                          2079  **      TPDSP - TAPE DISPLAY
                          2080  *
                          2081  *       SHOW H88 USER THAT THERE IS SOME ACTIVITY DURING A LOAD OR A DUMP
                          2082
    006.024  042 024 040  2083  TPDSP   SHLD    ABUSS           UPDATE ABUSS
                          2084
    006.027  076 015      2085          MVI     A,A.CR          RETURN
    006.031  315 302 003  2086          CALL    WCC
                          2087
    006.034  174          2088          MOV     A,H             ADDRESS
    006.035  315 343 005  2089          CALL    TOB
    006.040  175          2090          MOV     A,L
    006.041  315 343 005  2091          CALL    TOB
                          2092
    006.044  311          2093          RET


                          2095  **      HRNX - HORN EXTENSION ROUTINE
                          2096  *
                          2097  *       THIS IS AN EXTENSION TO *HORN* TO MAKE ROOM FOR A JUMP
                          2098
    006.045  056 011      2099  HRNX    MVI     L;#CTLFLG
    006.047  163          2100          MOV     M,E             TURN OFF HORN
    006.050  321          2101          POP     D
    006.051  341          2102          POP     H
    006.052  311          2103          RET


                          2105  **      NOISE - DING BELL ON CONSOLE
                          2106  *
                          2107  *       THIS IS A MODIFICATION TO ALLOW THE H88/H89 TO USE THE CONSOLE BELL
                          2108
    006.053  076 007      2109  NOISE   MVI     A,A.BEL
    006.055  315 302 003  2110          CALL    WCC
    006.060  303 140 002  2111          JMP     HORN            CONTINUE WITH NORMAL HORN DELAY
```

```
                        2113  **      TPERMSG - TAPE ERROR MESSAGE
                        2114  *
                        2115  *       DISPLAY THE TAPE ERROR NUMBER ON THE CONSOLE
                        2116
006.063  062 024 040   2117  TPERMSG STA     ABUSS           SAVE ERROR NUMBER
006.066  076 040       2118          MVI     A,' '           OUTPUT A SPACE
006.070  315 302 003   2119          CALL    WCC
006.073  170           2120          MOV     A,B             OUTPUT NUMBER
006.074  315 343 005   2121          CALL    TOB
006.077  311           2122          RET


                        2124  **      TYPMSG - TYPE MESSAGE TO CONSOLE
                        2125  *
                        2126  *       TYPMSG OUTPUTS AN ASCII MESSAGE FROM MEMORY TO THE CONSOLE
                        2127  *       UNTIL A NULL IS SENSED
                        2128  *
                        2129  *       ENTRY   (H,L) = ADDRESS OF MESSAGE
                        2130  *       EXIT    NONE
                        2131  *       USES    A,H,L,F
                        2132
                        2133
006.100  176           2134  TYPMSG  MOV     A,M             GET CHARACTER
006.101  267           2135          ORA     A               SEE IF A NULL
006.102  310           2136          RZ                      IF NULL, EXIT
                        2137
006.103  315 302 003   2138          CALL    WCC             ELSE OUTPUT CHARACTER TO CONSOLE
006.106  043           2139          INX     H               POINT TO NEXT CHARACTER
006.107  303 100 006   2140          JMP     TYPMSG          OUTPUT IT


                        2142  **      MSG.PR - MESSAGE FOR MONITOR PROMPT
                        2143  *
                        2144  *       CRLF, '  H: '
                        2145
                        2146
006.112  015 012 040   2147  MSG.PR  DB      A.CR,A.LF,'  H: ',0


                        2149  **      MSG.SP - MESSAGE TO TELL USER TO TYPE SPACES
                        2150  *
                        2151  *       'Type spaces to determine baud rate'
                        2152
006.122  124 171 160   2153  MSG.SP  DB      'Type SPACES to determine baud rate',0
```

```
                         2155  **      MSG.GO - (G)O
                         2156  *
                         2157  *       'GO'
                         2158
    006.165  157 040 000 2159  MSG.GO  DB      'o ',0


                         2161  **      MSG.LD - (L)OAD
                         2162  *
                         2163  *       'LOAD'
                         2164
    006.170  157 141 144 2165  MSG.LD  DB      'oad',0


                         2167  **      MSG.DMP - (D)UMP
                         2168  *
                         2169  *       'DUMP'
                         2170
    006.174  165 155 160 2171  MSG.DMP DB      'ump ',0


                         2173  **      MSG.SUB - (S)UBSTITUTE
                         2174  *
                         2175  *       'SUBSTITUTE'
                         2176
    006.201  165 142 163 2177  MSG.SUB DB      'ubstitute ',0


                         2179  **      MSG.PC - (P)ROGRAM COUNTER
                         2180  *
                         2181  *       'PROGRAM COUNTER'
                         2182
    006.214  162 157 147 2183  MSG.PC  DB      'rogram Counter ',0


                         2185  **      MSG.BT - (B)OOT
                         2186  *
                         2187  *       'BOOT'
                         2188
    006.234  157 157 164 2189  MSG.BT  DB      'oot',0
                         2478          LON     L
```

```
                        2481  **      ENTRY POINT FOR FLOPPY DISK ROTATIONAL SPEED TEST
                        2482  *
000.000                 2483          ERRNZ   10000A-6-*      MUST BE SIX BYTES BEFORE END
                        2484
007.372 303 240 006     2485  ESPEED  JMP     SPEED


                        2487  **      ENTRY POINT FOR DYNAMIC MEMORY TEST
                        2488  *
000.000                 2489          ERRNZ   10000A-3-*      MUST BE THREE BYTES BEFORE END
                        2490
007.375 303 116 007     2491  EDYMEM  JMP     DYMEM
                        2492
000.000                 2493          ERRNZ   *-10000A        MUST NOT EXCEED 2K BYTES
                        2494
```

```
                        2497  **      THE FOLLOWING ARE CONTROL CELLS AND FLAGS USED BY THE KEYSET
                        2498  *       MONITOR.
                        2499
    040.000             2500          ORG     40000A          8192
    040.000             2501  START   DS      2               DUMP STARTING ADDRESS
    040.002             2502  IOWRK   DS      2               IN OR OUT INSTRUCTION
    040.004             2503  PRSRAM  EQU     *               FOLLOWING CELLS INITIALIZED FROM ROM
    040.004             2504          DS      1               RET
                        2505
    040.005             2506  REGI    DS      1               INDEX OF REGISTER UNDER DISPLAY
    040.006             2507  DSPROT  DS      1               PERIOD FLAG BYTE
    040.007             2508  DSPMOD  DS      1               DISPLAY MODE
                        2509
    040.010             2510  .MFLAG  DS      1               USER FLAG OPTIONS
                        2511  *                               SEE *UO.XXX* BITS DESCRIBED AT FRONT
                        2512
    040.011             2513  CTLFLG  DS      1               FRONT PANEL CONTROL BITS
    040.012             2514  REFIND  DS      1               REFRESH INDEX (0 TO 7)
    000.007             2515  PRSL    EQU     *-PRSRAM        END OF AREA INITIALIZED FROM ROM
                        2516
    040.013             2517  FPLEDS  EQU     *               FRONT PANEL LED PATTERNS
    040.013             2518  ALEDS   DS      1               ADDR 0
    040.014             2519          DS      1               ADDR 1
    040.015             2520          DS      1               ADDR 2
                        2521
    040.016             2522          DS      1               ADDR 3
    040.017             2523          DS      1               ADDR 4
    040.020             2524          DS      1               ADDR 5
                        2525
    040.021             2526  DLEDS   DS      1               DATA 0
    040.022             2527          DS      1               DATA 1
    040.023             2528          DS      1               DATA 2
                        2529
    040.024             2530  ABUSS   DS      2               ADDRESS BUSS
    040.026             2531  RCCA    DS      1               RCC SAVE AREA
    040.027             2532  CRCSUM  DS      2               CRC-16 CHECKSUM
    040.031             2533  TPERRX  DS      2               TAPE ERROR EXIT ADDRESS
    040.033             2534  TICCNT  DS      2               CLOCK TIC COUNTER
                        2535
    040.035             2536  REGPTR  DS      2               REGISETR CONTENTS POINTER
                        2537
    040.037             2538  UIVEC   DS      0               USER INTERRUPT VECTORS
    040.037             2539          DS      3               JUMP TO CLOCK PROCESSOR
    040.042             2540          DS      3               JUMP TO SINGLE STEP PROCESSOR
    040.045             2541          DS      3               JUMP TO I/O 3
    040.050             2542          DS      3               JUMP TO I/O 4
    040.053             2543          DS      3               JUMP TO I/O 5
    040.056             2544          DS      3               JUMP TO I/O 6
    040.061             2545          DS      3               JUMP TO I/O 7
                        2546
                        2547  **      H88/H89 RAM USAGE BEYOND THAT OF H8MTRF
                        2548  *
    040.064             2549  NMIRET  DS      2
    040.066             2550          END
 ASSEMBLY COMPLETE
   2550 STATEMENTS
      0 ERRORS DETECTED
 12984 BYTES FREE
```

MTR88 - H88 MONITOR #09.00.00. XREF V1.1
CROSS REFERENCE TABLE PAGE 47

```
         000004   343S   504S   673S   815S  1629S  1630
.MFLAG   040010   504    546    619    625   2510L
A.BEL    000007   121E   564    658   1540   1867   1895   1920   1963   2109
A.BKS    000010   122E  2397
A.CR     000015   124E   699    755    776   1349   1429   1804   1838   1864   1892   1911   2013
                 2071   2085   2147   2282   2467
A.DEL    000177   126E  1480
A.ESC    000033   125E  2281   2290   2292   2298   2300   2465
A.LF     000012   123E   699    699    779   1429   2015   2075   2147   2281   2282   2282   2467
                 2467
A.STX    000002   120E   920   1134
A.SYN    000026   119E   915   1132
ABUSS    040024   713    928   1801   2083   2117  2530L
AC.DLY   000110   244E
ALARM    002136   629    988E  1063
ALARMB   002167   989   1014L
ALEDS    040013  2518L
BLKSIZ   002000   154E
BOOT     004256   690   1760L
BR19.2   004077  1625L
BR38.4   004101  1626L
BR56.0   004103  1627L
BR96     004075  1624L
BRTAB    004075  1579   1622E  1630
CB.CLI   000100   132E   188    391    624   1729
CB.MIL   000040   131E   490    543    624    817
CB.SPK   000200   133E   391    624    994
CB.SSI   000020   130E   391    490    624    802    813   1737
CLK4     000270   550    574E
CLOCK    000201   327    328    532L
CRC      002347  1198L  1261
CRC1     002356  1202L  1222
CRC2     003004  1213   1220L
CRCSUM   040027   923    965   1031   1140   1201   1223  2532L
CTC      002172   885   1030L
CTLFLG   040011   343    536    541    546    625    801    804    815    999   1637   2099  2513L
CUI1     000165  505L    562    575
DLEDS    040021  2526L
DLY      000053   381L
DM.MR    000000   137E
DM.MW    000001   138E
DM.RR    000002   139E
DM.RW    000003   140E
DOD      003122  1334L
DS.HOLE  000001  2216E  2241   2247
DSPMOD   040007  2508L
DSPROT   040006  2507L
DUMP     002002   913L
DY10.5   007265  1544   2424L
DY3.3    007153  2341   2345L
DY3.5    007163  2349   2353L
DY3.7    007173  2357   2361L
DY5.53   007251  2403   2407L
DY9.3    003315   587   1506L
DY9.4    003326  1511   1515L
DY9.5    003335  1519   1523L
DY9.8    003350  1528   1532L
DYASC    003143  1363E  1397   1410   1420   1546   2405   2453
```

DYASC1	003144	1366L	1368					
DYBYT	003160	1383L	1513	1521	1538	2351	2359	2417
DYBYT.2	003202	1395	1399L					
DYBYT.4	003221	1408	1412L					
DYBYT.6	003235	1418	1422E					
DYME5.5	007242	2399E	2408					
DYMEM	007116	1613	2319L	2491				
DYMEM1	007122	2324L						
DYMEM10	003360	1536	1540L	2435				
DYMEM11	007272	2426L	2427	2430	2433			
DYMEM2	007127	2326L	2332					
DYMEM3	007140	2329	2335L					
DYMEM4	007207	588	2370	2374L				
DYMEM5	007212	2375L	2387	2391				
DYMEM6	000273	582L	2415					
DYMEM7	000276	583L	586					
DYMEM9	000307	592L	2377	2382				
DYMSG	007306	599	1530	2343	2372	2447L	2457	
DYMSG.5	007316	2451	2455L					
EDYMEM	007375	2491L						
ERROR	000322	438	568	618E	1082	1481		
ESPEED	007372	2485L						
FPLEDS	040013	2517E						
GO	001222	781	791L					
GO.	000063	391L	791	1775				
GO88	001146	675	766L					
GO88.1	001177	769	778L					
H88.CTL	000362	102E	1573	1744	2320			
H88.SW	000362	106E	1580	1611				
H88B.CK	000002	103E						
H88B.SS	000001	104E						
H88S.BR	000300	107E	1581					
H88S.M	000040	108E	1612					
HORN	002140	993L	2111					
HRN0	002143	383	996L					
HRN2	002160	1007L	1008					
HRNX	006045	1010	2099L					
INIT	000073	314	316	413L	417			
INIT0	000000	312L						
INIT0.0	000003	313L	1618					
INIT0X	004000	312	1572L					
INIT0X1	004050	1603L	1604	1607				
INIT1	000107	426L	431					
INIT2	000117	433L						
INT1	000010	321E						
INT2	000020	337E						
INT3	000030	356L						
INT4	000040	363L						
INT5	000050	370L						
INT6	000060	388L						
INT7	000070	397L						
INTXIT	000172	514L	544	806				
IOA	003062	756	777	1293L	1800	1805	1839	
IOA1	005176	1293	1936L					
IOA2	005204	1940L	1958	1967				
IOA3	005242	1942	1945	1960L				
IOA4	005260	1961	1971L					
IOB	003066	1308L						

```
IOB1      003070   1309L   1316    1345    1352
IOB1.5    003126   1327    1342L
IOB2      003135   1314    1349L
IOC       005301   1848    1883    1996L
IOWRK     040002   1837    2225    2263    2265    2502L
IP.DS     000177   2212E   2240    2246
IP.PAD    000360    96E    1065    1098
IP.TPC    000371   112E    1100
IP.TPD    000370   114E    1176
IROC      005150   745      749     768    1795    1834    1910L   1922
IROC1     005166   1915    1920L
LOA0      001272   846L     893
LOA1      001342   872L     883
LOAD      001267   844E
LRA       003047   1276L
LRA.      003052   555      739     773     863     940    1277L   1764    1807
MI.ANI    000346   170E    1055
MI.EXAF   000010   177E    1365    1371
MI.HLT    000166   165E     561
MI.IN     000333   167E    1709    1718
MI.JIXA   000335   178E    1424    2460
MI.JIXB   000351   179E    1424    2460
MI.JIYA   000375   180E    1374
MI.JIYB   000351   181E    1374
MI.JMP    000303   172E
MI.LDA    000072   169E
MI.LDXA   000335   173E     596    1510    1518    1527    1535    2340    2348    2356    2369    2414
MI.LDXB   000041   174E     596    1510    1518    1527    1535    2340    2348    2356    2369    2414
MI.LDYA   000375   175E    1394    1407    1417    1543    2402    2450
MI.LDYB   000041   176E    1394    1407    1417    1543    2402    2450
MI.LXID   000021   171E
MI.OUT    000323   168E    1706    1725
MI.RET    000311   166E    1564
MSG.BT    006234   1760    2189L
MSG.DMP   006174   1793    2171L
MSG.EQ    007362   1524    2474L
MSG.ERR   001047    593     699L
MSG.GO    006165    766    2159L
MSG.HSS   007100   2268    2298L
MSG.LD    006170    720    2165L
MSG.PAS   003237   1429L   2366
MSG.PC    006214    736    2183L
MSG.PR    006112    642    2147L
MSG.RAM   007324   2337    2465L
MSG.SP    006122   1772    2153L
MSG.SPD   006371   2222    2281L
MSG.SUB   006201   1832    2177L
MSG.WRK   007062   2262    2290L
MTR       000344    636E    818
MTR.2     000357    645L    660
MTR.3     000371    649L    656
MTR.4     001014    650     662L
MTR1      000345    639E    640
MTRA      001025    647     672E    692
MTRAL     000006    648     692E
NMI       004116    483    1678L
NMI0.5    004160   1696    1699    1704L
NMI1      004202   1691    1716L
```

NMI1.5	004217	1719	1725L									
NMI2	004224	1728L										
NMI2.5	004251	1702	1707	1710	1726	1746L						
NMI3	004252	1714	1723	1748L								
NMIENT	000146	483L										
NMIRET	040064	1679	1686	2549L								
NOISE	006053	1014	2109L									
ONDRO	000022	2220E	2226									
OP.CTL	000360	97E	537	803	814							
OP.DC	000177	2211E	2227									
OP.DIG	000360	98E										
OP.SEG	000361	99E										
OP.TPC	000371	113E	444	914	978	1081	1172	1258				
OP.TPD	000370	115E	1260									
PCA	001103	687	736L									
PCA1	001137	746	754L									
PRSL	000007	313	2515E									
PRSRAM	040004	313	2503E	2515								
PRSROM	003371	1557E	1617									
RCC	003262	645	1309	1472E	1910	1940	1996	2070				
RCC1	003262	1474L	1476									
RCCA	040026	2531L										
RCK	003260	1451E										
REFIND	040012	2514L										
REGI	040005	1276	2506L									
REGPTR	040035	494	627	1279	2536L							
RMEM	001261	723	826L									
RNB	002331	872	1130	1156	1171L							
RNB1	002335	1173L	1175									
RNP	002325	858	868	1030	1141	1156L						
ROMDD	030000	26E	1765									
RT.BD	000005	149E										
RT.BP	000002	146E										
RT.CT	000003	147E										
RT.MI	000001	145E	845	924								
RT.NB	000004	148E										
RT.PD	000006	150E										
SAE	001063	713L										
SAVALL	000132	325	341	460L								
SAVALLR	000151	485	487E	1639								
SAVALLX	004105	475	1634E									
SC.ACE	000350	243E	1366	1372	1474	1478	1495	1500	1578	1590	1593	1595	1597
SC.UART	000372	203E										
SINCR	004000	419E	421	422								
SPEED	006240	2222L	2485									
SPEED1	006257	2228L	2272									
SPEED2	006275	2236	2239L									
SPEED3	006300	2240L	2242	2253								
SPEED4	006307	2246L	2248									
SPEED5	006357	2266	2269L									
SRMEM	001067	678	720L									
SRS	002265	846	1126E									
SRS1	002265	1127L	1135	1139								
SRS2	002271	1130L	1133									
SST1	001235	392	804L									
SSTEP	001225	799E										
START	040000	422	870	926	1798	2324	2374	2501L				
STPRTN	001244	344	812E									

SUBM	004370	684	1832L										
SUBM1	005013	1842L	1855	1862									
SUBM2	005027	1848L	1869										
SUBM3	005042	1854L	1887										
SUBM4	005046	1852	1857L										
SUBM5	005053	1860L	1890										
SUBM6	005062	1858	1864L										
SUBM7	005075	1849	1871L										
SUBM8	005077	1873L	1884										
SUBM9	005115	1883L	1897										
SWMEM	004313	681	1793L										
SWMEM2	004337	1801L											
SWMEM4	004351	1796	1806L										
TD.IN	000370	158E											
TD.OUT	000370	159E											
TER1	002220	1063L	1070										
TER3	002215	1056L	1067										
TFT	002133	892	977L	1051									
TICCNT	040033	532	534	1003	1068	2228	2257	2534L					
TOA	005313	748	1812	1842	2013L								
TOA.	005325	2018L	2271										
TOB	005343	1844	2019	2021	2036L	2089	2091	2121					
TOB1	005353	2043L	2051										
TPABT	002244	826	908	1080L									
TPDSP	006024	877	955	2083L									
TPERMSG	006063	1050	2117L										
TPERR	002205	857	1049L										
TPERRX	040031	827	909	1102	2533L								
TPXIT	002252	1064	1098L	1173	1254								
TS.IN	000371	160E											
TS.OUT	000371	161E											
TYPMSG	006100	643	721	737	767	1761	1773	1794	1833	2134L	2140	2223	2269
UC.2SB	000004	269E											
UC.5BW	000000	265E											
UC.6BW	000001	266E											
UC.7BW	000002	267E											
UC.8BW	000003	268E	1594										
UC.BI	000020	288E											
UC.CTS	000020	297E											
UC.DCS	000001	293E											
UC.DDR	000002	294E											
UC.DLA	000200	274E	1577										
UC.DR	000001	284E	1475										
UC.DRL	000010	296E											
UC.DSR	000040	298E											
UC.DTR	000001	277E											
UC.EDA	000001	255E											
UC.EFS	000020	271E											
UC.FE	000010	287E											
UC.IID	000006	262E											
UC.IIP	000001	261E											
UC.LOO	000020	281E											
UC.MSI	000010	258E											
UC.OR	000002	285E											
UC.OU1	000004	279E											
UC.OU2	000010	280E											
UC.PE	000004	286E											
UC.PEN	000010	270E											

```
UC.RI    000100    299E
UC.RLS   000200    300E
UC.RSI   000004    257E
UC.RTS   000002    278E
UC.SB    000100    273E
UC.SKP   000040    272E
UC.TER   000004    295E
UC.THE   000040    289E   1367   1496
UC.TRE   000002    256E
UC.TSE   000100    290E
UCI.ER   000020    225E   1171   1257
UCI.IE   000002    227E
UCI.IR   000100    223E
UCI.RE   000004    226E   1171
UCI.RO   000040    224E   1171
UCI.TE   000001    228E    913   1257
UDR      000000    200E
UIVEC    040037    356     363    370    388    397    508    819   2538L
UMI.16X  000002    218E    443
UMI.1B   000100    208E    443
UMI.1X   000001    217E
UMI.2B   000300    210E
UMI.64X  000003    219E
UMI.HB   000200    209E
UMI.L5   000000    213E
UMI.L6   000004    214E
UMI.L7   000010    215E
UMI.L8   000014    216E    443
UMI.PA   000020    212E
UMI.PE   000040    211E
UO.CLK   000001    190E    506
UO.DDU   000002    189E    621
UO.HLT   000200    187E    548
UO.NFR   000100    188E    621
UR.DLL   000000    250E   1593
UR.DLM   000001    252E   1590
UR.IER   000001    254E   1597
UR.IIR   000002    260E
UR.LCR   000003    264E   1578   1595
UR.LSR   000005    283E   1366   1474   1495
UR.MCR   000004    276E
UR.MSR   000006    292E
UR.RBR   000000    246E   1478
UR.THR   000000    248E   1372   1500
USR      000001    201E
USR.FE   000040    232E
USR.OE   000020    233E
USR.PE   000010    234E
USR.RXR  000002    236E   1174
USR.TXE  000004    235E
USR.TXR  000001    237E   1255
WCC      003302    565     567    659    662    778    780   1320  1494L  1846  1860  1868  1873
                  1896    1921   1947   1964   1971   2014   2016   2024   2048  2055  2074  2076  2086
                  2110    2119   2138
WCC1     003303   1495L   1497
WCR      006003    722    1762  2070L  2072
WME1     002012    917L    919
WME2     002104    950L    961
```

```
MTR88 - H88 MONITOR  #09.00.00.                                  XREF V1.1
   CROSS REFERENCE TABLE                                         PAGE   53

   WMEM     001374       907E      1813
   WNB      003024        917       921      951     1239     1253L
   WNB1     003025      1254L      1256
   WNP      003017        925       936      945      948      966      967     1238L

24284 BYTES FREE
```

APPENDIX B

MTR-88 DEMO

The sample program that follows shows some of the advanced features that are available to you with MTR-88. The program is not designed to be efficient or particularly useful by itself. It uses the H88 clock, console terminal, and interrupt capability to create an accurate interval timer that will time up to 377(octal) seconds. When the interval ends, the H88 audio alarm is sounded.

Use the H88 keyboard and the "Substitute" command to enter the machine code and start the program. You will also use the keyboard to enter the octal time.

The demo uses the MTR-88 firmware (program in a ROM) for most of the working routines, and you should look up the details of these routines (in Appendix A). The listing of the demo was prepared using the text editor and assembler that are available for the H88. However, the program should be loaded by hand using the "Substitute" command.

THE SAMPLE PROGRAM

This program initially blanks the screen and then waits for you to enter an octal value. The MTR-88 routine WCC is used to send characters to the screen, and IOB is used to Input an Octal Byte.

The most subtle part of the program is the interrupt processing. First, a jump to the interrupt processor is planted in UIVEC to allow processing of the clock interrupts. Then .MFLAG is set so MTR-88 will pass interrupts to the program. Finally, interrupts are enabled.

The main part of the program is a "do-nothing" loop that waits for the time to count down to zero. When the time is exhausted, the program restores the original state of .MFLAG and stops.

The interrupt processor keeps its own local TICCNT and counts it down from 500. When this count reaches zero, one second has elapsed and the new reduced time is displayed on the screen using TOB (Type Octal Byte). The local TICCNT is reset to 500. When the time is exhausted, the main program stops clock processing, so the processor is not called again.

```
                                ***  ********************************************************
                                *
                                *       DEMO: MTR-88
                                *
                                *       SYSTEM DEFINITIONS
                                *
                002.136         ALARM   EQU     2136A   MAKE NOISE
                003.302         WCC     EQU     3302A   WRITE CHAR TO CONSOLE
                040.010         .MFLAG  EQU     40010A  USER FLAG OPTIONS
                040.037         UIVEC   EQU     40037A  USER INTERRUPT VECTOR
                000.001         UO.CLK  EQU     1A      ALLOW CLOCK INTERRUPT PROCESSING
                000.303         MI.JMP  EQU     303A    MACHINE INSTRUCTION JUMP
                003.066         IOB     EQU     3066A   INPUT OCTAL BYTE
                005.343         TOB     EQU     5343A   TYPE OCTAL BYTE

                000.033         ESC     EQU     33A
                000.015         CR      EQU     15A
                000.012         LF      EQU     12A

                040.100                 ORG     40100A

                                ***  ********************************************************
                                *
                                *       ERASE SCREEN
                                *
                040.100 076 033 MTR88   MVI     A,ESC           ESCAPE SEQUENCE TO
                040.102 315 301 003             CALL    WCC
                040.105 076 105                 MVI     A,'E'           ERASE SCREEN
                040.107 315 301 003             CALL    WCC

                                ***  ********************************************************
                                *
                                *       READ A OCTAL INTEGER FROM KEYBOARD
                                *         STORE THE NUMBER.
                                *
                040.112 041 234 040             LXI     H,NUMBER        SET ADDRESS OF NUMBER
                040.115 247                     ANA     A               CLEAR CARRY (SIDE EFFECT)
                040.116 315 066 003             CALL    IOB             INPUT OCTAL BYTE
                040.121 315 223 040             CALL    SETICK          SETUP TICK TO 500 FOR ONE SEC

                                ***  ********************************************************
                                *
                                *       INITIALIZE SERVICE INTERRUPT ROUTINE
                                *          LOAD THE USER INTERRUPT VECTOR (UIVEC) WITH A
                                *          JUMP INSTRUCTION AND THE ADDRESS OF THE SERVICE
                                *          ROUTINE.  ENABLE USER CLOCK INTERRUPT!
                                *
                040.124 076 303                 MVI     A,MI.JMP        SET-UP JUMP INSTRUCTION
                040.126 062 037 040             STA     UIVEC           STORE 'JMP' INSTRUCTION
                040.131 041 165 040             LXI     H,INTRP             USER INTERRUPT ADDRESS
                040.134 042 040 040             SHLD    UIVEC+1                 POSITIONED
                040.137 076 001                 MVI     A,UO.CLK
                040.141 062 010 040             STA     .MFLAG          ENABLE CLOCK INTERRUPT
```

```
                        ***  ************************************************************
                        *
                        *       WAIT FOR CLOCK TO REACH ZERO
                        *
040.144  072 234 040 LOOP    LDA     NUMBER          DO NOTHING LOOP.
040.147  376 000             CPI     0                  WAIT FOR END
040.151  302 144 040         JNZ     LOOP               OF COUNT DOWN.

                        ***  ************************************************************
                        *
                        *       RETURN TO NORMAL INTERRUPT STATUS & HALT
                        *          DISABLE INTERRUPT & TURN ON SPEAKER
                        *
040.154  076 000             MVI     A,0
040.156  062 010 040         STA     .MFLAG          DISABLE CLOCK INTERRUPT
040.161  315 136 002         CALL    ALARM
040.164  166                 HLT

                        ***  ************************************************************
                        *
                        *       INTERRUPT ROUTINE
                        *          CLOCK AND DISPLAY INTERRUPT
                        *
040.165  052 232 040 INTRP   LHLD    TICK            GET COUNT (BETWEEN 0 & 500)
040.170  053                 DCX     H               TICK=TICK-1
040.171  042 232 040         SHLD    TICK            STORE COUNT
040.174  175                 MOV     A,L             TEST FOR ZERO
040.175  264                 ORA     H                  COMPARE WITH 'H'
040.176  300                 RNE                        EXIT IF .NE. 0

                        ***  ************************************************************
                        *
                        *       UPDATE DISPLAY FOR 'NEW' NUMBER.
                        *
040.177  076 015             MVI     A,CR            DO CARRIAGE RETURN
040.201  315 302 003         CALL    WCC
040.204  076 012             MVI     A,LF               AND LINE FEED
040.206  315 302 003         CALL    WCC
040.211  072 234 040         LDA     NUMBER          GET NUMBER
040.214  075                 DCR     A               NUMBER=NUMBER-1
040.215  062 234 040         STA     NUMBER          SAVE NUMBER
040.220  315 343 005         CALL    TOB             TYPE OCTAL BYTE
040.223  041 364 001 SETICK  LXI     H,500           RESTORE COUNT
040.226  042 232 040         SHLD    TICK               WITH 500
040.231  311                 RET
                        *               STORAGE AREA & END ASSEMBLY
040.232                 TICK    DS      2
040.234                 NUMBER  DS      1
040.235  000                 END     MTR88
```

>

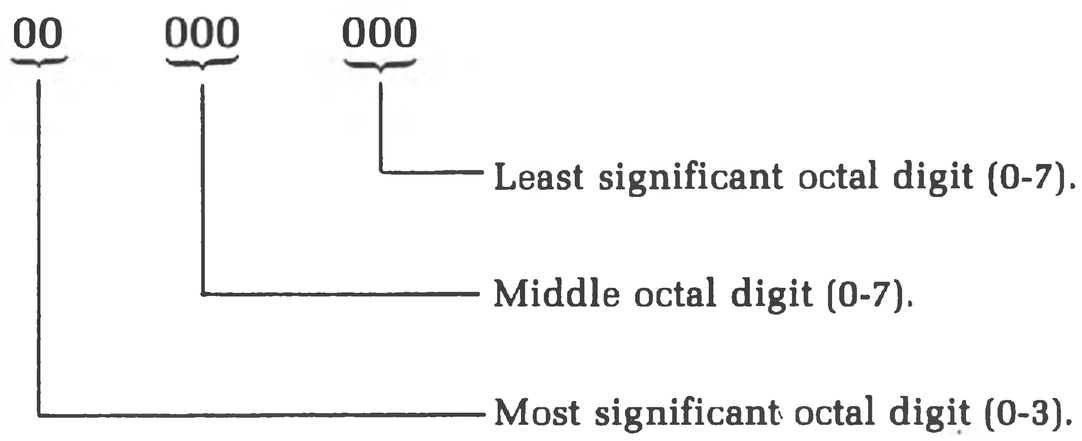
APPENDIX C

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

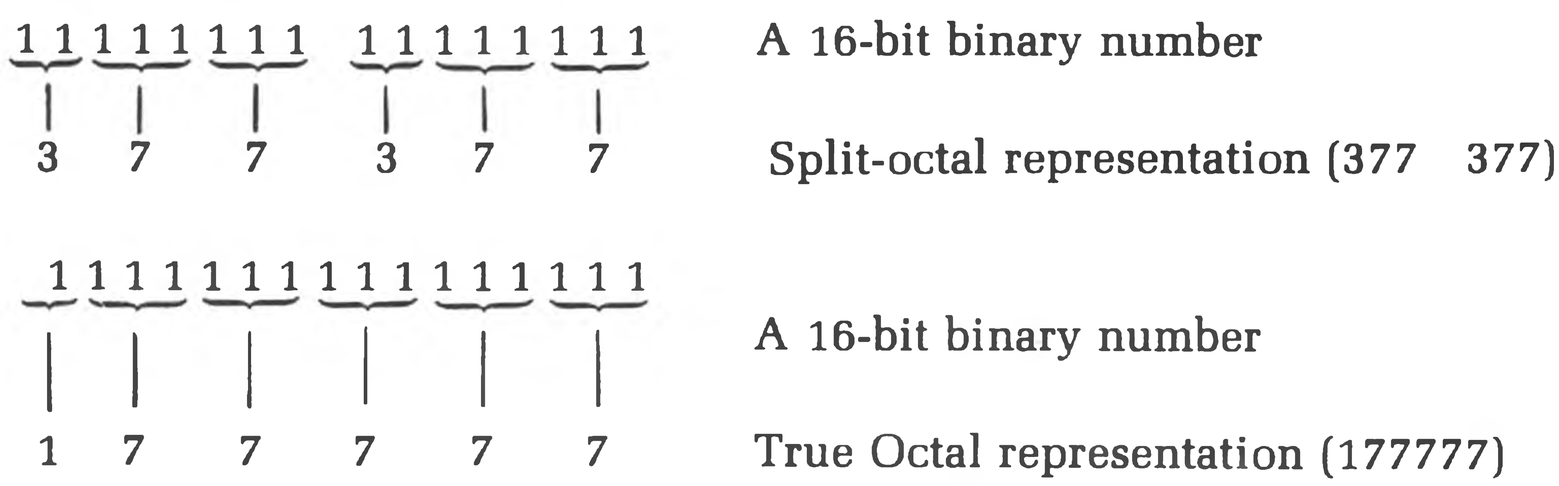
Each byte is displayed as two-and-one-half octal digits. The octal numbers lie in the range of 000 to 377 for binary numbers in the range 00000000 to 11111111, as shown below.



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 still 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:



The lower example shows true octal representation of a 16-bit binary number. True octal representation is never used in standard Heath software. Occasionally 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.