PANEL MONITOR

XCON-8

HEATH COMPANY
BENTON HARBOR, MICHIGAN 49022

595-2546

Printed in the United States of America

TABLE OF CONTENTS

INTRODUCTION

This Manual describes the functions and operations of the Heath H8 Panel Monitor Program, XCON-8, which resides permanently in a ROM on the H8 CPU board. XCON-8 provides a sophisticated front panel display and keyboard emulation as well as handling master clear and interrupt operations. Some of the major features of XCON-8 are:

- Memory contents display and alteration.
- Register contents display and alteration.
- Program execution control (both breakpoint and single instruction operation).
- Self-contained bootstraps for program loading and dumping.
- Port input and output routines.

In addition to the above features, XCON-8 can be instructed (by means of a flag byte contained in the H8 RAM) to bypass some or all of its normal functions so the sophisticated user can augment or totally replace them.

Communication with the Panel Monitor is accomplished through three devices: the keypad, the 7-segment displays, and the audio alert. The user enters commands and values through the 16-key keypad, and XCON-8 responds visually through the front panel displays. In addition to the front panel displays, XCON-8 provides the keypad entry and function feedback to the built-in speaker. Appropriate signals (short, medium, and long beeps) indicate that commands and data are accepted or rejected.

THEORY OF OPERATION

This section will supplement the information contained in the "Operation" and "Circuit Description" sections of your H8 Operation Manual. In order to fully understand how XCON-8 operates, you must be familiar with the H8 front panel and CPU. A thorough knowledge of the 8080 instruction set and its architecture is also essential.

Power Up and Master Clear

XCON-8 initializes the H8 whenever you power-up or master clear (RST). You initiate the power-up operation by turning on the rear panel Power switch. You can master clear by simultaneously depressing both the lower right-hand (RSTØ) and lower left-hand (Ø) keys of the H8 front panel keypad. Both power-up and RST cause a level zero (highest priority) interrupt and result in a long beep from the audio alert.

During initialization, XCON-8 enters a routine which determines the high limit of continuous RAM. Once the high limit of available RAM is determined, the H8 stack pointer (SP) is set to this value. XCON-8 then determines if the RAM starts at Ø, and copies itself from ROM into that RAM space. Control is passed to the front panel command loop. Using this feature, you can immediately determine the total amount of continuous memory above 8K by displaying the stack pointer value.

Clock Interrupts

The Clock Interrupt is a crucial element in the operation of the H8 front panel system. This level one interrupt is generated by the front panel hardware every 2,000 μS. XCON-8 uses this interrupt to check for some keyboard commands, to check for user program breakpoints, and to refresh the front panel displays.

XCON-8 performs these functions using a series of subroutines which are executed as necessary when indicated by the interrupts. For this reason, all user programs must maintain a valid stack (at high memory) containing at least 80 free bytes at all times. If this stack space is not available and XCON-8 is running (it can be disabled; see the Advanced Control Section), unpredictable software damage can occur in your program. In the same manner, if your program should execute a DI (Disable Interrupt) instruction, no front panel services including the RTM (Return To Monitor) function are available until an EI (Enable Interrupt) instruction is executed or until a master clear (RST/Ø) is performed.

XCON-8 Modes /Using RST and RTM

XCON-8 is always in either the monitor mode or the user mode. In the monitor mode no user program is executing, XCON-8 loops reading the keypad and refreshing the displays. All commands entered via the keypad are valid; however, the RTM command is meaningless.

When your program is being executed, XCON-8 is in the user mode and the MON LED on the front panel is extinguished. Only two keyboard commands are valid in this mode: RST (master clear) and RTM (Return To Monitor). NOTE: Both of these commands are dual key commands. No single key command is recognized, so a user program may have free use of the entire keypad.

You can return XCON-8 to the monitor mode by using the RTM command (simultaneously press the ø and the # keys). This command stops program execution at the end of the current instruction, stores the current value of each register, and returns XCON-8 to the monitor mode. You can then continue your program by pressing the GO key. The RST command (simultaneously press the ø and the / keys) performs the master clear operation described earlier and does not save any register values.

Normally, when a user program is running, XCON-8 is also running. Thus, if XCON-8 is displaying the contents of the HL register pair and the user program is started, it continues to display the contents of this register pair as the program is run. If the user program changes the contents of the HL pair, the change is immediately reflected in the front panel displays. In a similar manner, if a memory location is displayed when a user program is started, it is displayed during the time the user program is run. If the user program changes the contents of the display memory location, the front panel display changes.

Since XCON-8 does not recognize keypad commands in the user mode, the RTM command must be used before the memory location or register being displayed is changed to a new location or a different register. Once you select the new location or different register, you can resume program execution by pressing GO.

NOTE: XCON-8 requires about 10% of the H8 CPU's resources to process the display interrupts. Programs which are compute-bound may be slowed down by simultaneous operation of XCON-8. In this situation, you may wish to turn off the clock interrupts to improve execution time. See "Using Interrupts" on Page 1-24.

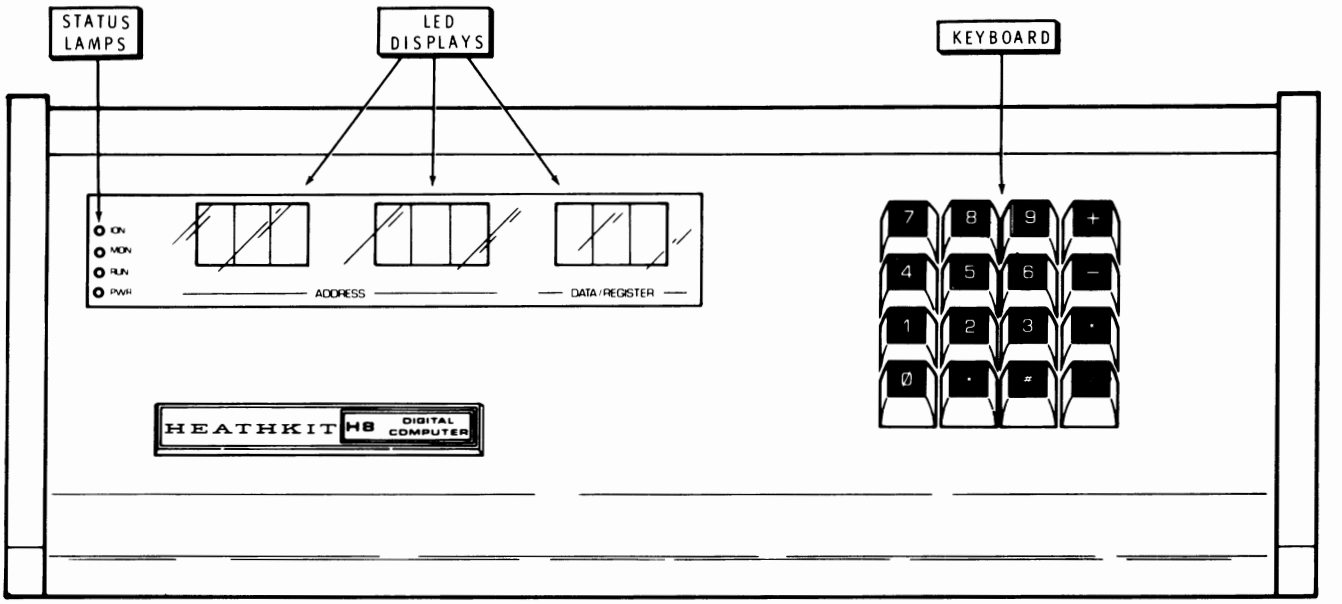


Figure 1-1

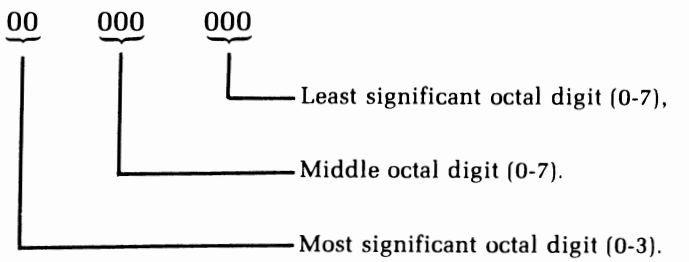
H8 Displays

You must understand the H8 front panel presentation in order to use XCON-8. The display is made up of 9 digits, in three groups of three digits each. See Figure 1-1. Each group of three digits displays one byte (eight bits) of information. This information may be the contents of a designated register or memory location, or it may be the address of a memory location itself. The register names are also displayed.

All binary numbers are converted to octal format for display on the H8 front panel. The following table shows binary to octal conversion.

BINARY NUMBER	OCTAL NUMBER
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 H8 front panel representation of the number 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**, and is used consistantly throughout all H8 displays of 16-bit numbers. Offset octal must not be confused with octal. For example:

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	A 16-bit binary number
3 7 7 3 7 7	Offset octal representation (377 377)
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	A 16-bit binary number
1 7 7 7 7 7	True Octal representation (177777)

The lower example shows true octal representation of a 16-bit binary number. This is **not** used by the H8 front panel displays or any H8 software. Occasionally you will see offset octal numbers printed with a decimal point separating the upper and lower bytes. For example:

377.377

Hi Byte Lo Byte

H8 Keypad

The H8 Keypad consists of 16 keys, as shown in Figure 1-1 on Page 1-7. When the keypad is operating under the control of XCON-8, it exhibits a number of unique properties.

- Each keystroke is verified by a short beep from the audio alert.
- Octal digits are entered using the keys 0 through 7.
- Holding a key down continuously repeats the key's function.
- The + key increments memory port or register locations.
- The − key decrements memory port or register locations.
- The * key cancels previous keypad entries.
- The ALTER key causes XCON-8 to enter the alter mode.
- The MEM key causes XCON-8 to enter the display memory mode.
- The REG key causes XCON-8 to enter the register mode.

Many of the keys on the keypad have multiple functions, depending on the XCON-8 mode being used. In the register mode, for example, the numeric keys (1-6) call the register indicated in the upper left-hand corner of the key. When the XCON-8 is in neither the register nor the memory mode, the keys perform the functions indicated in the lower right-hand corner of the key.

The # and / keys have additional special functions, as indicated earlier. When the / key is pressed simultaneously with the Ø key, the RST (master clear) sequence is initiated. When the # sign key is pressed simultaneously with the Ø key, the RTM (Return To Monitor) function is initiated, the user program is stopped, and XCON-8 regains control.

Each key is covered in greater detail as the various function are discussed.

DISPLAYING AND ALTERING MEMORY LOCATIONS

One of the major features of XCON-8 is its ability to examine the contents of any H8 memory location and to modify the contents of that memory location if it is RAM.

When the H8 is first powered up, XCON-8 is in the display memory mode. This mode is indicated by all digits displaying octal numbers and no decimal points being on.

Specifying a Memory Address

If you wish to display or alter the contents of a memory location, you must first place XCON-8 in the memory address mode and then enter the desired memory address. Place XCON-8 in the memory address mode (if not already there) by pressing the MEM (Memory) key. Specify the address to be displayed or altered by entering the 6-digit address (offset octal).

When you press the MEM key, all the decimal points will light. This indicates that the address may now be entered. Once the full 6-digit address is entered, the decimal points turn off, indicating that address entry is completed. After all 6 digits are entered, the address is displayed in the left-most six displays, and the contents of the addressed memory location are displayed in the right-hand 3 digits.

NOTE: As you press each key, including the MEM key, a short beep indicates successful entry. As each group of three octal digits is successfully entered, a medium beep is sounded. The sequence by which you specify a memory address is shown in Figure 1-2.

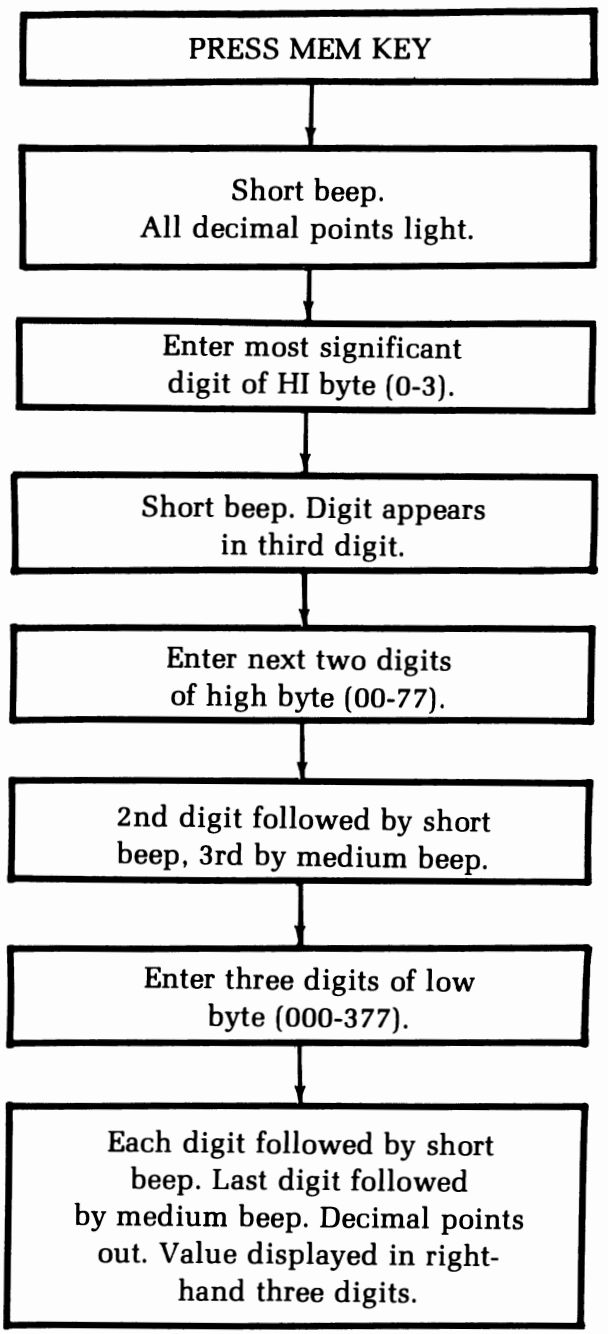


Figure 1-2
Entering a memory address through XCON-8.

NOTE: If you press a non-octal digit key as one of the six address digits, an error is flagged (a long beep). Once this error is flagged, the XCON-8 considers the address complete and extinguishes the decimal points. The entire sequence must be repeated.

Altering a Memory Location

Before you can alter a memory location, you must first display the contents of the memory location by specifying the memory address as described in the preceding paragraphs. After you specify the memory address, press the ALTER key. This will cause XCON-8 to enter the memory alter mode.

When XCON-8 enters the memory alter mode, a single decimal point rotates from right to left through all 9 digits. You can now alter the contents of the displayed location by entering the new octal value (three digits on the keypad). When the three digits have been entered, acoustical verification (a short beep) is given **and the memory address is incremented.** You can then alter this new location by entering three more digits or pressing one of the following keys, causing the monitor to perform the indicated function:

KEY	FUNCTION
+	Increment the address.
−	Decrement the address.
MEM	Specify a new memory address (leave the memory alter mode).
REG	Specify a register for display (leave the memory alter mode).
ALTER	Exit from the alter mode (into the display mode).

NOTE: XCON-8 automatically increments the memory address as each entry (3 octal digits) is complete. Therefore, you may load a program in sequential locations very rapidly. Each location is modified by simply entering the three octal digits.

The following example reviews each step as the H8 is turned on; the memory address mode is entered; and the location 040 123 is addressed, altered to 345, checked, and closed.

DISPLAY			COMMENTS
X X X	X X X	X X X	Random memory display at power up (X=random number.)
X.X.X.	X.X.X.	X.X.X.	MEM key pressed. (In memory address mode, a short beep.)
X.X.0.	X.X.X.	X.X.X.	0 key pressed. (Short beep.)
X.0.4.	X.X.X.	X.X.X.	4 key pressed. (Short beep.)
0.4.0.	X.X.X.	X.X.X.	0 key pressed. (Medium beep.) Contents of location 040 XXX displayed.)
0.4.0.	X.X.1.	X.X.X.	1 key pressed. (Short beep. Contents of 040 XX1 displayed.)
0.4.0.	X.1.2.	X.X.X.	2 key pressed. (Short beep. Contents of 040 X12 displayed.)
0 4 0	1 2 3	X X X	3 key pressed. (Medium beep. Contents of desired location 040 123 displayed, decimal points out.)
0.4.0	1.2.3	X.X.X	ALTER key pressed. (Short beep. Decimal points **rotate**.)
0.4.0.	1.2.3.	X.X.3.	3 key pressed. (Short beep. Decimal points **rotate**.)
0.4.0.	1.2.3.	X.3.4.	4 key pressed. (Short beep. Decimal points **rotate**.)
0.4.0.	1.2.4.	X.X.X.	5 key pressed. (Medium beep. Address increments one location. Decimal points **rotate**.)
0.4.0	1.2.3	3.4.5	−key pressed. (Short beep. Address decrements one location. Decimal points **rotate**.)
0 4 0	1 2 3	3 4 5	ALTER key pressed. (Short beep. Decimal points go out.)

Stepping Through Memory

When XCON-8 is either in the display memory or alter memory modes, the + and − keys increment and decrement the memory address. Each time you press the key, XCON-8 increments (or decrements) the memory address one location. If you hold the key down, the auto-repeat function of XCON-8 causes the memory address to increment or decrement repeatedly (approximately one location every second).

DISPLAYING AND ALTERING REGISTERS

XCON-8 can display and alter the contents of the 8080 CPU registers, just as it displays and alters the contents of H8 memory locations. Although the process is quite similar, a few special features should be noted.

Specifying a Register for Display

Press the REG key to specify that a register is to be displayed. After you press the REG key, press a second key (SP through PC, see the Table below) to specify the desired register or register pair.

When the REG key is pressed, six decimal points light, indicating that you must now select a register. NOTE: Simply pressing the REG key causes a register name to appear in the right-hand digits. However, you must select a register using the Register Select key before a register is definitely selected and its true contents are displayed. Once a register is selected, the decimal points are extinguished.

The contents of the selected register pair are displayed in the six left-most displays. The register name (or names) are displayed in the two right-most digits of the right-hand three displays. The registers are selected and displayed in accordance with the following table:

KEY	LEFT 3 DIGITS	MIDDLE 3 DIGITS	RIGHT PAIR	COMMENTS
SP (1)	000 to 377	000 to 377	SP	Stack pointer
AF (2)	000 to 377	000 to 377	AF	AF Register pair
BC (3)	000 to 377	000 to 377	bC	BC Register pair
DE (4)	000 to 377	000 to 377	dE	DE Register pair
HL (5)	000 to 377	000 to 377	HL	HL Register pair
PC (6)	000 to 377	000 to 377	PC	Program counter

NOTE: The contents of any single eight-bit register may lie in the range of 000 to 377 octal. The stack pointer (SP) and the program counter (PC) are 16-bit registers and are displayed as two sets of three octal numbers. Each 3-digit grouping corresponds to one byte (8 bit number). When a register pair is displayed, the left three digits correspond to the left register and the middle three digits correspond to the right register. For example:

256 312 AF

Register A contains 256 and register F contains 312.

Altering the Contents of a Selected Register

To alter the contents of a register (or register pair), you must first specify it as described in the preceding paragraphs. After you select the register or register pair, press the ALTER key. This will cause the six left-hand decimal points to rotate right to left, indicating that you may enter 6 digits to alter the contents of the indicated register or register pair.

Alternately, you may press one of the following command keys.

KEY	FUNCTION
+	Changes the register pair being displayed.
–	Changes the register pair being displayed.
MEM	Specify a new memory address (leave the alter register mode).
REG	Specify a new register for display (leave the alter register mode).
ALTER	Exit the register alter mode.

NOTE: Stack pointer register (SP) is not a direct display of the real stack pointer register, but simply a copy of the real stack pointer register and is used for display purposes only. The stack pointer cannot be altered from the front panel. To alter the stack pointer register, an SPHL (SPHL = 371) instruction must be written into memory. The desired new stack pointer value is then placed in the HL register pair. XCON-8 single instruction mode is used to execute the SPHL swap instructions, loading the stack pointer with the contents loaded in the HL register pair.

Stepping Through the Registers

Use + and – keys to change the register pair being displayed. For example, if the DE register pair is being displayed, pressing the + key causes the next sequential register pair to be displayed (the HL pair). In the same manner, pressing the – key causes the register to decrement to the preceding pair. For example, if the DE pair is being displayed, pressing the – key displays the BC register pair. NOTE: Holding down either the + key or the – key causes the display to continuously increment or decrement through all the six registers/register pairs.

PROGRAM EXECUTION CONTROL

XCON-8 supports three basic program execution control facilities:

- Beginning or starting execution.
- Breakpointing.
- Single instruction.

Each of these execution controls permits the programmer to execute the desired portions of a program and examine its effects. He may execute the entire program, or a small group of instructions, or a single program instruction.

Initiating Program Execution

To begin the execution of a program residing in H8 memory, place the address of the first instruction to be executed in the PC (program counter). Use the methods described in "Displaying and Altering Registers" (Page 1-14). Once the address of this first instruction is placed in the program counter, press the GO key and program execution will begin. NOTE: Unless the program disables the front panel, the display continues to be actively updated, although the front panel commands are no longer active (except for RST and RTM). If the program counter is displayed when you press the GO key, XCON-8 continuously monitors the program counter.

Breakpointing

Breakpointing permits the programmer to execute small portions of a program and then return to XCON-8. Breakpointing is especially useful when a program is being "debugged." Small portions of the program may be executed and their results observed. If there is an error, it may be corrected before an entire program is involved.

When the H8 executes a program and encounters a halt instruction, it re-enters XCON-8 and sounds the alarm. All of the registers are preserved and the program counter points to the address **following** the address of the halt instruction. Thus, you can breakpoint a program from the front panel by inserting halt instructions (HLT = 166) at the desired points throughout the program. When a particular

section of the program is tested and the breakpoint feature is no longer required, you can change the halt to a "no operation" (NOP = 000). Once the halts are changed to NOPs, execution of the NOP simply passes control to the next successive instruction. Program execution for breakpointing uses the GO key as previously described.

NOTE: If you temporarily replace an existing instruction with a halt, you must restore the instruction before resuming program execution. The contents of the program counter point to the address **following** the halt. Therefore, if the instruction which replaced the halt is to be executed, when the program continues, the contents of the program counter must be decremented one location before execution is resumed.

Single Instruction Operation

Any user program may be operated in the single instruction mode. This procedure is identical to the GO command, except that the SI key is pressed rather than the GO key. When the SI key is pressed, a single **instruction** (not a single machine cycle) is executed and then control is returned to XCON-8. Single instruction operation is available for careful inspection of program results and for executing special programs, such as swapping the HL register pair with the stack pointer as discussed in "Altering the Contents of a Selected Register" (Page 1-15).

Interrupting a Program During Execution

You can interrupt a running program (with all registers preserved at the point of interruption) by pressing RTM & Ø. You can then examine and/or alter the contents of various memory locations and all the registers as required. Resume execution of the program at the next sequential instruction by simply pressing the GO key. NOTE: Although all registers and memory locations are preserved when RTM & Ø are pressed, it is very difficult to stop a program at an exact location. Therefore, use the breakpoint feature if you want to stop the program at an exact location.

LOAD/DUMP ROUTINES

XCON-8 contains a routine that lets you load and dump memory contents from or to a tape. This feature is especially important, as most computers require one of two successive "boot strap" routines to be hand-loaded before a desired program can be loaded into the main memory. All these "boot strap" routines are contained within the XCON-8 ROM, and use sophisticated error checking techniques. Thus, a program can be loaded or dumped by simply pressing a single key.

Loading From Tape

To load from a tape, ready the reader device with the tape to be loaded prior to executing the load command. Place XCON-8 in the display memory mode and press the LOAD key. Once the LOAD key is pressed, XCON-8 starts the tape transport and scans the tape for the first file record.

No change will be seen on the front panel displays until XCON-8 finds the first file. When the first file record is located, XCON-8 checks it to see if it is the first (or only) record in a sequence, and the record is a memory dump record. If it is not a memory dump record, a number two error is flagged (see "Tape Errors" on Page 1-20).

Once a correct record is found, loading proceeds. The loading procedure places the entry point address of the program being loaded in the H8 program counter. The H8 memory is then loaded. The displays continuously show the address being loaded and the data being loaded at these addresses. When the load is complete, XCON-8 sounds a long beep and displays the final memory address. If the load is faulty, a number one error is displayed and the audio alarm continuously beeps. (See "Tape Errors," Page 1-20.)

NOTE: You may abort a partial load by using the CANCEL key. Naturally, the load image resulting from this action is incorrect, and should not be executed.

Dumping to Tape

Before dumping a memory image onto tape, the following three dump parameters are required:

- The entry point address (the program starting address).
- The dump starting address.
- The dump ending address.

Set the desired entry point address by placing this value in the program counter (PC). This value will be placed in the program counter whenever you load the program so execution will begin at this address when you press the GO key.

Place the dump starting address into the first two H8 RAM cells. These are: 040 000 (offset octal) and 040 001 (offset octal). NOTE: The low order byte of the address should be placed into location 040 000 and the high order byte of the starting address should be placed into location 040 001.

Enter the dump ending address as a memory address using the # (MEM) key. Then ready the tape transport and press the DUMP key. As the tape dump takes place, the number of bytes left to be dumped and the contents of the memory location being dumped are displayed on the front panel. You can abort a dump by using the CANCEL key. If the CANCEL key is used, an incomplete dump image is left on the tape. This cannot be loaded at a future date. NOTE: A successful load automatically sets up the following three dump parameters:

A. The program starting locations are stored in locations 040 000 and 040 001.
B. The program ending location is displayed.
C. The program counter contains the program entry point.

Figure 1-3A shows the steps of a typical dump sequence and Figure 1-3B shows the steps of a typical load sequence.

1. Set PC to 040 100; (040 100 = entry address).
2. Set 040 000 to 100 (100 = low byte of dump start).
3. Set 040 001 to 040 (040 = high byte of dump start).
4. Enter memory address 052 340 (052 340 = end address of dump).
5. Be sure tape is ready.
6. Press DUMP.

Figure 1-3A
The H8 memory image dump.

1. Be sure tape is ready.
2. Press LOAD.

Figure 1-3B
The H8 memory image load.

Copying a Tape

The beginning and final address of the load image are placed at the appropriate points. Thus, to copy a tape, simply load the tape as described in "Loading From Tape" (Page 1-18). Then ready the dump tape drive and press the DUMP key. A dump then takes place, including entry point, initial address, and final address.

In a similar manner, to load, alter, and then dump, enter only the ending address. The other parameters are unchanged from the load if locations 040 000, 040 001 or the program counter have not been modified during the altering procedure.

Tape Errors

XCON-8 detects two types of tape errors: record errors and checksum errors. In either case, when an error is detected, the tape transport is halted. The error number is then displayed in the center three digits (001 for a checksum error, 002 for a record error) and the alarm is repeatedly sounded. To halt the alarm and return to the command mode, press the CANCEL key.

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 load image. Therefore missing the initialization information at the start of the file.
- A tape error which causes a portion of the load image to be missed so the next record read is not in the proper sequence.

CHECKSUM ERRORS

A checksum error is flagged when the CRC (Cyclical Redundancy Check) checksum following a record does not match the CRC calculated by PAM-8. This error means that the record is either incorrectly recorded or the load is faulty. In either case, the load should be attempted again. If successive loads result in repeated failures, the original tape must be suspected as faulty.

I/O FACILITIES

XCON-8 supports two commands that allow you to perform input and output functions on H8 I/O ports. These front panel instructions permit simple manipulation of the H8 I/O ports without your having to write extensive routines to perform these functions.

Inputting From a Port

To input from a port, press the # key. Then enter three zero digits and the three-digit address (octal) of the desired port. NOTE: The front panel should now display 000 AAA, where AAA is the port address and 000 is meaningless. Press the IN key to read the port, the value is displayed in the three left-most digits of the front panel display.

Outputting to a Port

To output to a specified port, press the # key. Then enter the value to be supplied to the port in the three left-most displays. The port address is entered into the middle three displays. The display is of the form VVV AAA, where V stands for value, and A for address. Pressing the OUT key causes the value to be outputted to the indicated port.

Addressing Port Pairs

Frequently, ports are assigned in pairs, where one of the two port addresses is the control and status register and the other port is the data port. Address port pairs by using the + and − key to change ports. Once the initial port has been defined, the + key increments the port address to a new higher numbered port, and the − key is used to decrement to a lower numbered port.

ADVANCED CONTROL

One of the advanced features of XCON-8 is its provisions allowing sophisticated users to augment or replace XCON-8's functions. Augmenting or replacing XCON-8 functions is usually done in conjunction with assembly language programs. Sometimes it is possible to implement these features by using the POKE and PEEK commands in BASIC.

16-Bit Tick Counter (TICCNT)

XCON-8 maintains a 16-bit (2 byte) tick counter known as TICCNT. The value of this counter is incremented each time a clock interrupt is processed. As an interrupt occurs once every 2 mS, the counter is incremented once every 2 mS. As long as clock interrupts are not disabled, this value can be used by any program to compute elapsed time. The tick counter may be set to any desired value, but it should not be frequently reset, as this interferes with the front panel refresh cycle. The contents of the tick counter are contained in memory locations 040 033 (the least significant byte) and 040 034 (the most significant byte).

Using the Keypad

When your program is running, XCON-8 does not recognize any single key command. Thus, all single key patterns are available for your program. To read keypad patterns, you can use one of two routines. First, you may take an input from port IP. PAD; or second, your program may use XCON-8 RCK (read Console Keypad) routine. The input port IP. PAD is permanently assigned to port location 360. Inputting a binary number from this port detects which of the 16 keys are depressed.

The RCK routine provides keypad decoding, keypad debounce routines, auto-repeat routines, and acoustical feedback.

NOTE: If you use two key combinations, each key must reside in a separate bank. The first bank includes keys 0-7 and the second bank includes keys 8-#. RCK cannot decode two key combinations.

Display Usage

When a user program is running, XCON-8 normally displays the contents of the selected register or memory location. However, you may disable this process and display any arbitrary segment pattern, or completely disable the display to provide greater computational through-put. The display usage is primarily controlled by setting various bits in the .MFLAG memory cell. This memory cell is found at location 040 010.

MANUAL UPDATING

By setting the UO.DDU bit in the .MFLAG memory location, you can instruct XCON-8 to continue refreshing the front panel displays and to disable updating. When this is done, XCON-8 continues to refresh the LED's from a 9-byte block of RAM cells found at locations 040 013 thorugh 040 023. When the UO.DDU bit is set in .MFLAG, the contents of these bytes are not altered in any manner by XCON-8.

You can use this technique to display numbers, letters, or arbitrary bar patterns on the front panel displays. For instance, your program may alter the display by inserting any value into FPLEDS. The front panel LED segments will display a decimal integer if you use the octal to 7-segment pattern (DODA) display.

MANUAL DISPLAY REFRESHING

By setting the UO.NFR (User Option.No Front Panel Refresh) bit in the .MFLAG memory cell, you can instruct XCON-8 to stop refreshing the front panel displays. Setting the UO.NFR bit does not disable the clock interrupts; therefore, the tick counter (TICCNT) is still incremented. But XCON-8 does not refresh the displays from the information contained in the FPLEDS bytes.

NOTE: If you desire, you may write a program to refresh the front panel LED displays. Usually this is done using the clock interrupts. If you undertake an independent front panel refresh program, take extreme care to avoid burning the displays due to excessive refreshing. **The total power dissipated in the LEDs is determined by the refresh cycle, and too frequent refreshing will result in excessive display heating.**

Using Interrupts

All H8 interrupts cause control to be transferred into the low 64 bytes of memory. XCON-8 occupies this memory space so all interrupts are first processed by XCON-8. Except for level zero interrupts, which are used as master clears, you can supply an interrupt processing routine for each of the seven additional interrupts. The following sections explain the use of each of these interrupts.

I/O INTERRUPTS

Interrupts numbered 3 through 7 are I/O interrupts. XCON-8 does not process these interrupts in any way. When a level 3 through level 7 interrupt is received, XCON-8 immediately transfers to the user interrupt vectors contained in memory locations 040 037 through 040 064. Each location must contain a jump instruction pointing to the appropriate program location which processes these interrupts.

NOTE: If any of these interrupts occur, you must supply a processing routine for them. This routine must be complete including both entry and exit processing. When you use H8 interrupts, you must use only the available vector which is 6 to insure compatibility with future H8 products. You may also use 2 if you will not be using BUG-8.

CLOCK INTERRUPTS

The level one interrupts are generated by the front panel hardware every 2 mS. XCON-8 normally processes these interrupts. However, by setting a processing vector in UIVEC and setting the UO.INT bit in the .MFLAG cell, XCON-8 enters the users routine each time a clock interrupt is generated.

SINGLE INSTRUCTION AND BREAKPOINT INTERRUPTS

Level two interrupts are generated by the single instruction hardware contained on the CPU card. When a single instruction is requested, the result of the interrupt is processed by XCON-8. If the single instruction interrupt was generated by XCON-8 in response to a Monitor Mode Single Instruction register condition, XCON-8 processes it. Otherwise, XCON-8 jumps to the user level two interrupt vector (UIVEC). Since the level two interrupt does not affect XCON-8, a level two restart instruction can be used as a breakpoint instruction by the user programs.

FLOPPY BOOT

XCON-8 contains the code necessary to boot-up an operating system from a floppy disk. Two forms of "Boot" let you select the device (H17 or H47) and drive number (0-2 or 0-3). "Boot Primary" refers to the device that you will use most often. "Boot Secondary" provides you with a convenient way to boot from your alternate device, if you have one.

BOOT PRIMARY

The primary boot device is selected by switch SW1 sections 4, 1, and 0 on the extended configuration board. This switch is preset for H17 primary device. You may change the switch sections to select H47 primary device.

DISPLAY	ACTION	COMMENTS
	Press "1"	Boot H17 primary
	or	
	Press "1"	Boot H47 primary

BOOT SECONDARY

	Press "2"	Boot H17 secondary
	or	
	Press "2"	Boot H47 secondary

You may use the "CANCEL" key to abort the boot command and return to the monitor.

AUTO BOOT

If Switch SW1 section 7 is set to 1, the floppy disk will boot from the primary device automatically at power-up and master clear.

NOTE: We do not recommend auto-booting with a diskette in the drive and the door closed at power-up. Damage could occur to the diskette if you attempt to do so. Rather, power-up the H8 and H17 (H47), insert the diskette, and close the door within 15 seconds. Rebooting with Auto-Boot is the prime reason for its implementation. Software may accomplish this by executing an RST Ø.

BOOT FROM DRIVE OTHER THAN DRIVE Ø

Primary and secondary Boot are both designed to access drive Ø on either the H17 or the H47. However, if you have not selected Auto Boot, you may boot from H47 drive 1 or 2 or H17 drive 1, 2, or 3 by following this procedure:

1. Use XCON-8 "Altering the Contents of a Selected Register" procedures to set register A to the drive number that you want to boot from.

2. If you are booting from a primary alternate drive, simply press "GO."

3. If you are booting from a secondary alternate drive, set register PC to 007 367 (the secondary drive address) and press "GO."

NOTE: Register PC is already set for the primary drive address (007 364) at power-up and master clear.

ERRORS

The front panel will display 
if any of the following conditions occur:

1. The boot device does not respond within 15 seconds.

2. Switch SW1 is set to an undefined setting.

3. A disk error occurs.

NOTE: The "boot Err" message will only remain on the display a few seconds. XCON-8 will then return to the panel monitor mode.

SWITCH SW1

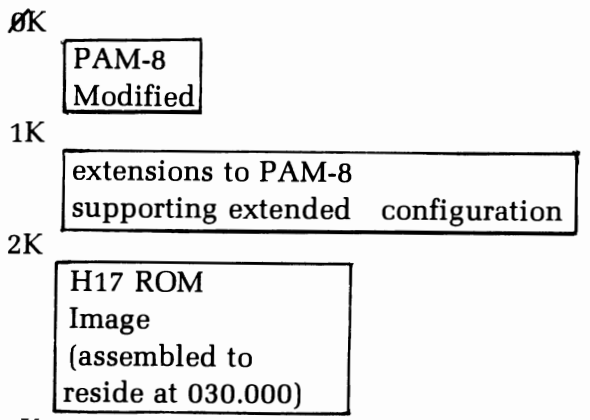
The sections of SW1 (on the HA8-8 Extended Configuration Board) have been defined as follows:

SWITCH SECTION 7 6 5 4 3 2 1 0	DESCRIPTION
X X X X X X 0 0	Port 174/177 = H17
X X X X X X 0 1	Port 174/177 = H47
X X X X 0 0 X X	Port 170/173 = unused
X X X X 0 1 X X	Port 170/173 = H47
X X X 0 X X X X	Boot primary from port 174/177
X X X 1 X X X X	Boot primary from port 170/173
0 X X X X X X X	Normal
1 X X X X X X X	Auto-Boot

Note that switches 5 and 6 are reserved.

MEMORY MAP

The lower 4K of memory is used as follows:



```
                4  ***     PAM/8 - H8 FRONT PANEL MONITOR.
                5  *
                6  *       J. G. LETWIN, 05/01/76.
                7  *
                8  *       FOR *WINTEK* INC.
                9  *
               10  *       COPYRIGHT  05/1976, WINTEK CORPORATION,
               11  *                          902 N. 9TH ST.
               12  *                          LAFAYETTE, IND.
               13  *       Modified:
               14  *              25 Aug 79       PAM8GO          JMTittsler
               15  *                              added single button boot from H17
               16  *              31 Dec 79       PAM8AT          JMTittsler
               17  *                              added automatic power on boot from H17
               18  *              04 Mar 80       PAM8GO          JMTittsler
               19  *                              changed default display to PC
               20  *              20 Jun 80       RAM8GO          JMTittsler
               21  *                              added RAM at zero capability
               22  *              07 Aug 80       Ram8Go          G. Chandler     /Ram8Go 2/
               23  *                              Issue:  01.02.00
               24  *                              H17 Boot
               25  *                              H47 Boot
               26  *                              Auto-Boot
               27  *                              Primary/Secondary Support
               28  *                              Modified RAM at Zero (No double move)
               29  *                              Remove 030.000 default PC to avoid confusion
               30  *


               32  ***     PAM/8 - H8 FRONT PANEL MONITOR.
               33  *
               34  *       THIS PROGRAM RESIDES (IN ROM) IN THE LOW 1024 BYTES OF THE HEATH
               35  *       H8 COMPUTER. IT ACTUALLY CONSISTS OF TWO VIRTUALLY INDEPENDANT
               36  *       ROUTINES: A TASK-TIME PROGRAM WHICH PROVIDES SOPHISTICATED
               37  *       FRONT PANEL MONITOR SERVICE, AND AN INTERRUPT-TIME PROGRAM WHICH
               38  *       PROVIDES BOTH A REAL-TIME CLOCK AND EMULATES AN EFFECTIVE
               39  *       HARDWARE FRONT PANEL.


               41  ***     INTERRUPTS.
               42  *
               43  *       PAM/8 IS THE PRIMARY PROCESSOR FOR ALL INTERRUPTS.
               44  *       THEY ARE PROCESSED AS FOLLOWS:
               45  *
               46  *       RST     USE
               47  *
               48  *       0       MASTER CLEAR. (NEVER USED FOR I/O OR RST)
               49  *
               50  *       1       CLOCK INTERRUPT. NORMALLY TAKEN BY PAM/8,
               51  *               SETTING BIT *UO.CLK* IN BYTE *.MFLAG* ALLOWS
               52  *               USER PROCESSING (VIA A JUMP THROUGH *UIVEC*).
               53  *               UPON ENTRY OF THE USER ROUTINE, THE STACK
               54  *               CONTAINS:
```

```
                *               (STACK+0)  = RETURN ADDRESS (TO PAM/8)
                *               (STACK+2)  = (STACKPTR+14)
                *               (STACK+4)  = (AF)
                *               (STACK+6)  = (BC)
                *               (STACK+8)  = (DE)
                *               (STACK+10) = (HL)
                *               (STACK+12) = (PC)
                *               THE USER'S ROUTINE SHOULD RETURN TO PAM/8 VIA
                *               A *RET* WITHOUT ENABLING INTERRUPTS.
                *
                *       2       SINGLE STEP. SINGLE STEP INTERRUPTS GENERATED
                *               BY PAM/8 ARE PROCESSED BY PAM/8.
                *               ANY SINGLE STEP INTERRUPT RECEIVED WHEN IN
                *               USER MODE CAUSES A JUMP THROUGH *UIVEC*+3.
                *               STACK UPON USER ROUTINE ENTRY:
                *               (STACK+0)  = (STACKPTR+12)
                *               (STACK+2)  = (AF)
                *               (STACK+4)  = (BC)
                *               (STACK+6)  = (DE)
                *               (STACK+8)  = (HL)
                *               (STACK+10) = (PC)
                *               THE USER'S ROUTINE SHOULD HANDLE IT'S OWN RETURN
                *               FROM THE INTERRUPT.
                *
                *
                *       THE FOLLOWING INTERRUPTS ARE VECTORED DIRECTLY THROUGH *UIVEC*.
                *       THE USER ROUTINE MUST HAVE SETUP A JUMP IN *UIVEC* BEFORE ANY
                *       OF THESE INTERRUPTS MAY OCCUR.
                *
                *       3       I/O 3. CAUSES A DIRECT JUMP THROUGH *UIVEC*+6
                *
                *       4       I/O 4. CAUSES A DIRECT JUMP THROUGH *UIVEC*+9
                *
                *       5       I/O 5. CAUSES A DIRECT JUMP THROUGH *UIVEC*+12
                *
                *       6       I/O 6. CAUSES A DIRECT JUMP THROUGH *UIVEC*+15
                *
                *       7       I/O 7. CAUSES A DIRECT JUMP THROUGH *UIVEC*+18
```

```
                         95  **      ASSEMBLY CONSTANTS

                         97  **      IO PORTS
                         98
000.360                  99  IP.PAD  EQU     360Q            PAD INPUT PORT
000.360                 100  OP.CTL  EQU     360Q            CONTROL OUTPUT PORT
000.360                 101  OP.DIG  EQU     360Q            DIGIT SELECT OUTPUT PORT
000.361                 102  OP.SEG  EQU     361Q            SEGMENT SELECT OUTPUT PORT
000.371                 103  IP.TPC  EQU     371Q            TAPE CONTROL IN
000.371                 104  OP.TPC  EQU     371Q            TAPE CONTROL OUT
000.370                 105  IP.TPD  EQU     370Q            TAPE DATA IN
000.370                 106  OP.TPD  EQU     370Q            TAPE DATA OUT
000.362                 107  IP.CON  EQU     362Q            Configure Port          /Ram8Go 2/
000.362                 108  OP.CTL2 EQU     362Q            Secondary Control Port  /Ram8Go 2/

                        110  **      ASCII CHARACTERS.
                        111
000.026                 112  A.SYN   EQU     026Q            SYNC CHARACTER
000.002                 113  A.STX   EQU     002Q            STX CHARACTER

                        115  **      FRONT PANEL HARDWARE CONTROL BITS.
                        116
000.020                 117  CB.SSI  EQU     000100008       SINGLE STEP INTERRUPT
000.040                 118  CB.MTL  EQU     001000008       MONITOR LIGHT
000.100                 119  CB.CLI  EQU     010000008       CLOCK INTERRUPT ENABLE
000.200                 120  CB.SPK  EQU     100000008       SPEAKER ENABLE

                        122  **      Secondary Control Bytes                         /Ram8Go 2/
                        123
000.001                 124  CB2.SSI EQU     00000001B       Single-Step Enable
000.002                 125  CB2.CLI EQU     00000010B       Clock Interrupt Enable
000.040                 126  CB2.ORG EQU     001000008       ORG-0 Enable
000.100                 127  CB2.SID EQU     010000008       Side-1 Select

                        129  **      DISPLAY MODE FLAGS (IN *DSPMOD*)
                        130
000.000                 131  DM.MR   EQU     0               MEMORY READ
000.001                 132  DM.MW   EQU     1               MEMORY WRITE
000.002                 133  DM.RR   EQU     2               REGISTER READ
000.003                 134  DM.RW   EQU     3               REGISTER WRITE
```

```
                        136  **      Configuration Flags                                    /Ram8Go 2/
                        137
    000.003             138  CN.174M EQU     000000118       Port 174Q Device-Type Mask
    000.014             139  CN.170M EQU     000011008       Port 170Q Device-Type Mask
    000.020             140  CN.PRI  EQU     000100008       Primary/Secondary:  1 => Primary == 170Q
    000.040             141  CN.MEM  EQU     001000008       Memory Test/Normal
    000.100             142  CN.BAU  EQU     010000008       Baud Rate:  0 => 9600; 1 => 19200
    000.200             143  CN.ABO  EQU     100000008       Auto-Boot:  1 => Auto-Boot
                        144
    000.000             145  CND.H17 EQU     008             H-17 Disk               Valid only in CN.174M
    000.000             146  CND.NDI EQU     008             No Disk Installed       Valid only in CN.170M
    000.001             147  CND.H47 EQU     018             H-47


                        149  **      Boot Constants (H17 Rom Dependant)                     /Ram8Go 2/
                        150
    041.061             151  AIO.UNI EQU     41061A          Boot Device Unit Number
    037.132             152  BOOTA   EQU     37132A          Disk Constants ROM Source
    000.130             153  BOOTAL  EQU       130Q          Disk Constants Length
    000.012             154  ERPTCNT EQU        10           Soft Error Retry Count
    036.073             155  R.SDP.  EQU     36073A          Common ROM Code
    034.031             156  ROMCLK  EQU     34031A          H17 Clock Vector


                        158  **      Segment Definitions                                    /Ram8Go 2/
                        159
    000.001             160  S0      EQU     000000018
    000.002             161  S1      EQU     000000108
    000.004             162  S2      EQU     000001008
    000.010             163  S3      EQU     000010008
    000.020             164  S4      EQU     000100008
    000.040             165  S5      EQU     001000008
    000.100             166  S6      EQU     010000008
    000.200             167  S7      EQU     100000008


                        169  **      Key Definitions                                        /Ram8Go 2/
                        170
    000.257             171  K.PLUS  EQU     101011118               +
    000.217             172  K.MINU  EQU     100011118               -
    000.157             173  K.STAR  EQU     011011118               *
    000.117             174  K.DIVD  EQU     010011118               /
    000.057             175  K.NUMB  EQU     001011118               #
    000.017             176  K.DOT   EQU     000011118               .
    000.000             177          XTEXT   TAPE            TAPE DEFINITIONS
```

```
                        179X **     TAPE EQUIVALENCES.
                        180X
000.001                 181X RT.MI  EQU   1           RECORD TYPE - MEMORY DUMP IMAGE
000.002                 182X RT.BP  EQU   2           RECORD TYPE - BASIC PROGRAM
000.003                 183X RT.CT  EQU   3           RECORD TYPE - COMPRESSED TEXT
000.004                 184X RT.NB  EQU   4           RECORD TYPE - NEW BASIC PROG.
000.005                 185X RT.BD  EQU   5           RECORD TYPE - BASIC DATA
000.006                 186X RT.PD  EQU   6           RECORD TYPE - BASIC PROG. AND DATA
                        187X
                        188X **     BLOCK SIZE FOR INTER-PRODUCT COMMUNICATION.
                        189X
002.000                 190X BLKSIZ EQU   512
                        191X
                        192X **     IO PORT VALUES.
                        193X
000.370                 194X TD.IN  EQU   370Q          TAPE DATA IN
000.370                 195X TD.OUT EQU   370Q          TAPE DATA OUT
000.371                 196X TS.IN  EQU   371Q          TAPE STATUS IN
000.371                 197X TS.OUT EQU   371Q          TAPE STATUS OUT


                        199  **     MACHINE INSTRUCTIONS.
                        200
000.166                 201  MI.HLT EQU   01110110B     HALT
000.311                 202  MI.RET EQU   11001001B     RETURN
000.333                 203  MI.IN  EQU   11011011B     INPUT
000.303                 204  MI.JMP EQU   11000011B     Jump                    /Ram8Go 2/
000.323                 205  MI.OUT EQU   11010011B     OUTPUT
000.072                 206  MI.LDA EQU   00111010B     LDA
000.346                 207  MI.ANI EQU   11100110B     ANI
000.021                 208  MI.LXID EQU  00010001B     LXI D


                        210  **     USER OPTION BITS.
                        211  *
                        212  *      THESE BITS ARE SET IN CELL .MFLAG.
                        213
000.200                 214  UO.HLT EQU   10000000B     DISABLE HALT PROCESSING
000.100                 215  UO.NFR EQU   CB.CLI        NO REFRESH OF FRONT PANEL
000.002                 216  UO.DDU EQU   00000010B     DISABLE DISPLAY UPDATE
000.001                 217  UO.CLK EQU   00000001B     ALLOW PRIVATE INTERRUPT PROCESSING


000.000                 219          XTEXT  H8SEQU
```

```
                        221X **      HDOS SYSTEM EQUIVALENCES.
                        222X *
                        223X
    024.000             224X S.GRT0  EQU     24000A          SYSTEM AREA FOR  GRT0
    025.000             225X S.GRT1  EQU     25000A          SYSTEM AREA FOR  GRT1
    026.000             226X S.GRT2  EQU     26000A          SYSTEM AREA FOR  GRT2
                        227X
    030.000             228X ROMBOOT EQU     30000A          ROM BOOT ENTRY
                        229X
    040.100             230X         ORG     40100A          FREE SPACE FROM PAM-8
                        231X
    040.100             232X         DS      8               JUMP TO SYSTEM EXIT
    040.110             233X D.CON   DS      16              DISK CONSTANTS
    040.130             234X SYDD    EQU     *               SYSTEM DISK ENTRY POINT
    040.130             235X D.VEC   DS      24*3            SYSTEM ROM ENTRY VECTORS
    040.240             236X D.RAM   DS      31              SYSTEM ROM WORK AREA
    040.277             237X S.VAL   DS      36              SYSTEM VALUES
    040.343             238X S.INT   DS      115             SYSTEM INTERNAL WORK AREAS
    041.126             239X         DS      16
    041.146             240X S.SOVR  DS      2               STACK OVERFLOW WARNING
    041.150             241X         DS      42200A-*        SYSTEM STACK
    001.032             242X STACKL  EQU     *-S.SOVR        STACK SIZE
                        243X
    042.200             244X STACK   EQU     *               LWA+1 SYSTEM STACK
    042.200             245X USERFWA EQU     *               USER FWA
    042.200             246          XTEXT   EDRAM


                        248X **      EDRAM - DISK RAM WORKAREA DEFINITION.
                        249X *
                        250X *       ZEROED UPON BOOTING UP.
                        251X *
                        252X *       HOSEQU MUST BE CHANGED WHEN THIS DECK IS CHANGED.
                        253X
                        254X
    040.240             255X         ORG     D.RAM
                        256X
    040.240             257X D.TT    DS      1               TARGET TRACK (CURRENT OPERATION)
    040.241             258X D.TS    DS      1               TARGET SECTOR (CURRENT OPERATION)
                        259X
    040.242             260X D.DVCTL DS      1               DEVICE CONTROL BYTE
                        261X
    040.243             262X D.DLYMO DS      1               MOTOR ON DELAY COUNT
    040.244             263X D.DLYHS DS      1               HEAD SETTLE DELAY COUNTER
                        264X
    040.245             265X D.TRKPT DS      2               ADDRESS IN D.DRVTB FOR TRACK NUMBER
    040.247             266X D.VOLPT DS      2               ADDRESS IN D.DRVTB FOR VOLUME NUMBER
                        267X
    040.251             268X D.DRVTB DS      2*4             TRACK NUMBER AND VOLUME NUMBER FOR 4 DRIVES
                        269X
    040.261             270X D.HECNT DS      1               HARD ERROR COUNT
    040.262             271X D.SECNT DS      2               SOFT ERROR COUNT
    040.264             272X D.OECNT DS      1               OPERATION ERROR COUNT
                        273X
```

```
                    274X *      GLOBAL DISK ERROR COUNTERS
                    275X
040.265             276X D.ERR  DS    0          BEGINNING OF ERROR BLOCK
040.265             277X D.E.MDS DS   1          MISSING DATA SYNC
040.266             278X D.E.MSY DS   1          MISSING HEADER SYNC
040.267             279X D.E.CHK DS   1          DATA CHECKSUM
040.270             280X D.E.HCK DS   1          HEADER CHECKSUM
040.271             281X D.E.VOL DS   1          WRONG VOLUME NUMBER
040.272             282X D.E.TRK DS   1          BAD TRACK SEEK
040.273             283X D.ERRL DS    0          LIMIT OF ERROR COUNTERS
                    284X
                    285X *      I/O OPERATION COUNTS
                    286X
040.273             287X D.OPR  DS    2
040.275             288X D.OPW  DS    2
                    289X
000.037             290X D.RAML EQU   *-D.RAM
040.277             291         XTEXT EDVEC


                    293X **     JMP VECTORS FOR ROM CODE
                    294X *
                    295X *      SEE DISK ROM FOR ADDRESSES
                    296X *
                    297X *      HOSEQU MUST BE ALTERED WHEN THIS TABLE IS ALTERED.
                    298X
040.130             299X        ORG   D.VEC
                    300X
040.130             301X D.SYDD DS    3          JMP  R.SYDD (MUST BE FIRST)
040.133             302X D.MOUNT DS   3          JMP  R.MOUNT
040.136             303X D.XOK  DS    3          JMP  R.XOK
040.141             304X D.ABORT DS   3          JMP  R.ABORT
040.144             305X D.XIT  DS    3          JMP  R.XIT
040.147             306X D.READ DS    3          JMP  R.READ
040.152             307X D.READR DS   3          JMP  R.READR
040.155             308X D.WRITE DS   3          JMP  R.WRITE
040.160             309X D.CDE  DS    3          JMP  R.CDE
040.163             310X D.DTS  DS    3          JMP  R.DTS
040.166             311X D.SDT  DS    3          JMP  R.SDT
040.171             312X D.MAI  DS    3          JMP  R.MAI
040.174             313X D.MAO  DS    3          JMP  R.MAO
040.177             314X D.LPS  DS    3          JMP  R.LPS
040.202             315X D.RDB  DS    3          JMP  R.RDB
040.205             316X D.SDP  DS    3          JMP  R.SDP
040.210             317X D.STS  DS    3          JMP  R.STS
040.213             318X D.STZ  DS    3          JMP  R.STZ
040.216             319X D.UDLY DS    3          JMP  R.UDLY
040.221             320X D.WSC  DS    3          JMP  R.WSC
040.224             321X D.WSP  DS    3          JMP  R.WSP
040.227             322X D.WNB  DS    3          JMP  R.WNB
040.232             323X D.ERRT DS    3          JMP  R.ERRT
040.235             324X D.DLY  DS    3          JMP  R.DLY
040.240             325         XTEXT H17DEF
```

```
                        327X **     H17 CONTROL INFORMATION.
                        328X
      000.177           329X DP.DC  EQU   07FH          DISK CONTROL PORT
                        330X
      000.001           331X DF.HD  EQU   000000001B    HOLE DETECT
      000.002           332X DF.TO  EQU   000000010B    TRACK 0 DETECT
      000.004           333X DF.WP  EQU   000000100B    WRITE PROTECT
      000.010           334X DF.SD  EQU   000001000B    SYNC DETECT
                        335X
      000.001           336X DF.WG  EQU   000000001B    WRITE GATE ENABLE
      000.002           337X DF.DS0 EQU   000000010B    DRIVE SELECT 0
      000.004           338X DF.DS1 EQU   000000100B    DRIVE SELECT 1
      000.010           339X DF.DS2 EQU   000001000B    DRIVE SELECT 2
      000.020           340X DF.MO  EQU   000100000B    MOTOR ON (BOTH DRIVES)
      000.040           341X DF.DI  EQU   001000000B    DIRECTION (0=OUT)
      000.100           342X DF.ST  EQU   010000000B    STEP COMMAND (ACTIVE HIGH)
      000.200           343X DF.WR  EQU   100000000B    WRITE ENABLE RAM
                        344X
                        345X
                        346X
                        347X **     DISK UART PORTS AND CONTROL FLAGS.
                        348X
      000.174           349X UP.DP  EQU   07CH          DATA PORT
      000.175           350X UP.FC  EQU   07DH          FILL CHARACTER
      000.175           351X UP.ST  EQU   07DH          STATUS FLAGS
      000.176           352X UP.SC  EQU   07EH          SYN CHARACTER (OUTPUT)
      000.176           353X UP.SR  EQU   07EH          SYNC RESET (INPUT)
                        354X
      000.001           355X UF.RDA EQU   000000001B    RECEIVE DATA AVAILABLE
      000.002           356X UF.ROR EQU   000000010B    RECEIVER OVERRUN
      000.004           357X UF.RPE EQU   000000100B    RECEIVER PARITY ERROR
      000.100           358X UF.FCT EQU   010000000B    FILL CHAR TRANSMITTED
      000.200           359X UF.TBM EQU   100000000B    TRANSMITTER BUFFER EMPTY
                        360X
                        361X
                        362X
                        363X **     CHARACTER DEFINITIONS.
                        364X
      000.375           365X C.DSYN EQU   0FDH          PREFIX SYNC CHARACTER
      040.240           366         XTEXT H47DEF


                        368X **     H47DEF  -  H47 Constant Definitions
                        369X *
```

```
                        371X *      Z-80 INSTRUCTIONS
                        372X
    242.355             373X M.INI  EQU     10100010B*256+11101101B     INI  INSTRUCTION
    243.355             374X M.OUTI EQU     10100011B*256+11101101B     OUTI INSTRUCTION


                        376X **     DISK INTERFACE CONSTANTS
                        377X *
                        378X
    000.000             379X D.STAI EQU     0                           INTERFACE STATUS PORT Index
    000.001             380X D.DATI EQU     D.STAI+1                    DATA PORT Index
                        381X
    000.001             382X S.ERR  EQU     00000001B                   ERROR BIT
    000.040             383X S.DON  EQU     00100000B                   DONE
    000.100             384X S.IEN  EQU     01000000B                   INTERRUPT ENABLE
    000.200             385X S.DTR  EQU     10000000B                   DATA TERMINAL REQUEST
                        386X
    000.002             387X S.SW0  EQU     00000010B                   DIP SWITCH:  0
    000.004             388X S.SW1  EQU     00000100B                   DIP SWITCH:  1
    000.010             389X S.SW2  EQU     00001000B                   DIP SWITCH:  2
    000.020             390X S.SW3  EQU     00010000B                   DIP SWITCH:  3
                        391X
    000.002             392X W.RES  EQU     00000010B                   RESET COMMAND


                        394X **     STATUS BYTE FLAGS
                        395X *
                        396X
    000.200             397X SB.UNR EQU     10000000B                   UNIT NOT READY
    000.100             398X SB.WPD EQU     01000000B                   WRITE PROTECTED DRIVE
    000.040             399X SB.DLD EQU     00100000B                   DELETED DATA
    000.020             400X SB.NRF EQU     00010000B                   NO RECORD FOUND
    000.010             401X SB.CRC EQU     00001000B                   CRC ERROR
    000.004             402X SB.LTD EQU     00000100B                   LATE DATA
    000.002             403X SB.ILC EQU     00000010B                   ILLEGAL COMMAND
    000.001             404X SB.BTO EQU     00000001B                   BAD TRACK OVERFLOW


                        406X **     AUXILLARY STATUS BYTE FLAGS
                        407X *
                        408X
    000.100             409X AS.ODD EQU     01000000B                   TRACK 0 DOUBLE DENSITY
    000.040             410X AS.1DD EQU     00100000B                   TRACK 1-76 DOUBLE DENSITY
    000.020             411X AS.S1A EQU     00010000B                   SIDE 1 AVAILABLE
    000.003             412X AS.SLM EQU     00000011B                   SECTOR LENGTH MASK
```

```
                    414X **     DISK COMMANDS
                    415X *
                    416X
000.000             417X        ORG     0
000.000             418X DD.BOOT DS     1           BOOT
000.001             419X DD.RST  DS     1           READ STATUS
000.002             420X DD.RAS  DS     1           READ AUX. STATUS
000.003             421X DD.LSC  DS     1           LOAD SECTOR COUNT
000.004             422X DD.RAD  DS     1           READ ADDRESS OF LAST SECTOR ACCESSED
000.005             423X DD.REA  DS     1           READ SECTORS
000.006             424X DD.WRI  DS     1           WRITE SECTORS
000.007             425X DD.REAB DS     1           READ SECTORS BUFFERED
000.010             426X DD.WRIB DS     1           WRITE SECTORS BUFFERED
000.011             427X DD.WRD  DS     1           DD.WRI  + DELETED
000.012             428X DD.WRBD DS     1           DD.WRIB + DELETED
000.013             429X DD.CPY  DS     1           COPY
000.014             430X DD.FRM0 DS     1           FORMAT IBM SD
000.015             431X DD.FRM1 DS     1           FORMAT     SD
000.016             432X DD.FRM2 DS     1           FORMAT IBM DD
000.017             433X DD.FRM3 DS     1           FORMAT     DD
000.020             434X DD.RRDY DS     1           Read Ready (conflict with DD.SPF0)


                    436X **     Special De-Bug Functions
                    437X *
                    438X
000.020             439X        ORG     010H
000.020             440X DD.SPF0 DS     1           SPECIAL FUNCTION  0
000.021             441X DD.SPF1 DS     1           SPECIAL FUNCTION  1
000.022             442X DD.SPF2 DS     1           SPECIAL FUNCTION  2
000.023             443X DD.SPF3 DS     1           SPECIAL FUNCTION  3
000.024             444X DD.SPF4 DS     1           SPECIAL FUNCTION  4
000.025             445X DD.SPF5 DS     1           SPECIAL FUNCTION  5


                    447X **     Special Heath Functions
                    448X *
                    449X
000.200             450X        ORG     080H
000.200             451X DD.SDC  DS     1           SET DRIVE CHARACTERISTICS
000.201             452X DD.ST   DS     1           SEEK TO TRACK
000.202             453X DD.DS   DS     1           DISK STATUS
000.203             454X DD.RDL  DS     1           READ LOGICAL
000.204             455X DD.WTL  DS     1           WRITE LOGICAL
000.205             456X DD.RDBL DS     1           READ BUFFERED LOGICAL
000.206             457X DD.WTBL DS     1           WRITE BUFFERED LOGICAL
000.207             458X DD.WTDL DS     1           WRITE DELETED DATA LOGICAL
000.210             459X DD.WDLB DS     1           WRITE BUFFERED DELETED DATA LOGICAL
```

```
                    461X **      Useful Flags
                    462X *
                    463X
000.000             464X UNT.0   EQU     000000008           Unit: 0
000.040             465X UNT.1   EQU     001000008           Unit: 1
000.100             466X UNT.2   EQU     010000008           Unit: 2
000.140             467X UNT.3   EQU     011000008           Unit: 3
                    468X
000.140             469X UNT.M   EQU     UNT.0!UNT.1!UNT.2!UNT.3 Unit Mask
                    470X
                    471X
                    472X
000.000             473X SID.0   EQU     000000008           Side: 0
000.200             474X SID.1   EQU     100000008           Side: 1
                    475X
000.200             476X SID.M   EQU     SID.0!SID.1         Side Mask
                    477X
                    478X
                    479X
000.037             480X SEC.M   EQU     000111118           Track Mask
                    481X
                    482X
                    483X
004.000             484X SSIZ.M  EQU     1024                Maximum Sector Size
                    485X
                    486X
                    487X *C.128  EQU     128
                    488X *C.256  EQU     256
                    489X *C.26   EQU     26
000.211             490          XTEXT   U8251       DEFINE 8251 USART BITS
```

```
                        493X **      8251 USART BIT DEFINITIONS.
                        494X *
                        495X
                        496X **      PORT ADDRESSES
                        497X
000.000                 498X UDR     EQU     0              DATA REGISTER IS EVEN
000.001                 499X USR     EQU     1              STATUS REGISTER IS NEXT
                        500X
000.372                 501X SC.UART EQU     372Q           CONSOLE USART ADDRESS (IFF 8251)
                        502X
                        503X
                        504X **      MODE INSTRUCTION CONTROL BITS.
                        505X
000.100                 506X UMI.1B  EQU     010000000B     1 STOP BIT
000.200                 507X UMI.HB  EQU     100000000B     1 1/2 STOP BITS
000.300                 508X UMI.2B  EQU     110000000B     2 STOP BITS
000.040                 509X UMI.PE  EQU     001000000B     EVEN PARITY
000.020                 510X UMI.PA  EQU     000100000B     USE PARITY
000.000                 511X UMI.L5  EQU     000000000B     5 BIT CHARACTERS
000.004                 512X UMI.L6  EQU     000001000B     6 BIT CHARACTERS
000.010                 513X UMI.L7  EQU     000010000B     7 BIT CHARACTERS
000.014                 514X UMI.L8  EQU     000011000B     8 BIT CHARACTERS
000.001                 515X UMI.1X  EQU     000000001B     CLOCK X 1
000.002                 516X UMI.16X EQU     000000010B     CLOCK X 16
000.003                 517X UMI.64X EQU     000000011B     CLOCK X 64
                        518X
                        519X **      COMMAND INSTRUCTION BITS.
                        520X
000.100                 521X UCI.IR  EQU     010000000B     INTERNAL RESET
000.040                 522X UCI.RO  EQU     001000000B     READER-ON CONTROL FLAG
000.020                 523X UCI.ER  EQU     000100000B     ERROR RESET
000.004                 524X UCI.RE  EQU     000001000B     RECEIVE ENABLE
000.002                 525X UCI.IE  EQU     000000100B     ENABLE INTERRUPTS FLAG
000.001                 526X UCI.TE  EQU     000000001B     TRANSMIT ENABLE
                        527X
                        528X **      STATUS READ COMMAND BITS.
                        529X
000.040                 530X USR.FE  EQU     001000000B     FRAMING ERROR
000.020                 531X USR.OE  EQU     000100000B     OVERRUN ERROR
000.010                 532X USR.PE  EQU     000010000B     PARITY ERROR
000.004                 533X USR.TXE EQU     000001000B     TRANSMITTER EMPTY
000.002                 534X USR.RXR EQU     000000100B     RECEIVER READY
000.001                 535X USR.TXR EQU     000000001B     TRANSMITTER READY
```

```
                        538  ***     INTERRUPT VECTORS.
                        539  *
                        540


                        542  **      LEVEL 0 - RESET
                        543  *
                        544  *       THIS 'INTERRUPT' MAY NOT BE PROCESSED BY A USER PROGRAM.
                        545
000.000                 546          ORG     00A
                        547
000.000                 548  RAM8GO  EQU     *               RAM8GO Address                  /Ram8Go 2/
                        549
                        550  *
000.000  021 000 000   551  INIT0   LXI     D,RAM8GO        ROM COPY OF RAM                 /Ram8Go 2/
000.003  303 016 004   552          JMP     XINIT           DO EXTENDED INITIALIZATION (ROM)/RAM8GO JUN80/
000.006  303 073 000   553          JMP     INIT            INITIALIZE
377.073                 554          ERRPL   INIT-1000A      BYTE IN WORD 10A MUST BE 0


                        556  **      LEVEL 1 - CLOCK
                        557
000.010                 558  INT1    EQU     10Q             INTERRUPT ENTRY POINT
                        559
000.000                 560          ERRNZ   *-11Q           INT0 TAKES UP ONE BYTE
000.011  315 132 000   561          CALL    SAVALL          SAVE USER REGISTERS
000.014  026 000       562          MVI     D,0
000.016  303 201 000   563          JMP     CLOCK           PROCESS CLOCK INTERRUPT
377.201                 564          ERRPL   CLOCK-1000A     EXTRA BYTE MUST BE 0


                        566  **      LEVEL 2 - SINGLE STEP
                        567  *
                        568  *       IF THIS INTERRUPT IS RECEIVED WHEN NOT IN MONITOR MODE,
                        569  *       THEN IT IS ASSUMED TO BE GENERATED BY A USER PROGRAM
                        570  *       (SINGLE STEPPING OR BREAKPOINTING). IN SUCH CASE, THE
                        571  *       USER PROGRAM IS ENTERED THROUGH (UIVEC+3
                        572
000.020                 573  INT2    EQU     20A             LEVEL 2 ENTRY
                        574
000.000                 575          ERRNZ   *-21A           INT1 TAKES EXTRA BYTE
000.021  315 132 000   576          CALL    SAVALL          SAVE REGISTERS
000.024  032           577          LDAX    D               (A) = (CTLFLG)
040.011                 578  .       SET     CTLFLG
000.025  303 244 001   579          JMP     STPRTN          STEP RETURN
```

```
                        581  ***     I/O INTERRUPT VECTORS.
                        582  *
                        583  *       INTERRUPTS 3 THROUGH 7 ARE AVAILABLE FOR GENERAL I/O USE.
                        584  *
                        585  *       THESE INTERRUPTS ARE NOT SUPPORTED BY PAM/8, AND SHOULD
                        586  *       NEVER OCCUR UNLESS THE USER HAS SUPPLIED HANDLER ROUTINES
                        587  *       (THROUGH UIVEC)
                        588
000.030                 589          ORG     30A
000.030  303 045 040   590  INT3    JMP     UIVEC+6         JUMP TO USER ROUTINE
                        591
000.033  064 064 064   592          DB      '44470'         Heath Part Number               /Ram8Go 2/


000.040                 594          ORG     40A
000.040  303 050 040   595  INT4    JMP     UIVEC+9         JUMP TO USER ROUTINE
                        596
000.043  100 112 107   597          DB      100Q,112Q,107Q,114Q,100Q        Support Code    /Ram8Go 2/


000.050                 599          ORG     50A
000.050  303 053 040   600  INT5    JMP     UIVEC+12        JUMP TO USER ROUTINE
                        601
                        602
                        603  **      DLY - DELAY TIME INTERVAL.
                        604  *
                        605  *       ENTRY   (A) = MILLISECOND DELAY COUNT/2
                        606  *       EXIT    NONE
                        607  *       USES    A,F
                        608
000.053  365           609  DLY     PUSH    PSW             SAVE COUNT
000.054  257           610          XRA     A               DONT SOUND HORN
000.055  303 143 002   611          JMP     HRNO            PROCESS AS HORN


000.060                 613          ORG     60A
000.060  303 056 040   614  INT6    JMP     UIVEC+15        JUMP TO USER ROUTINE
                        615
                        616
000.063  076 320       617  GO.     MVI     A,CB.SSI+CB.CLI+CB.SPK  OFF MONITOR MODE LIGHT
000.065  303 235 001   618          JMP     SST1            RETURN TO USER PROGRAM


000.070                 620          ORG     70A
000.070  303 061 040   621  INT7    JMP     UIVEC+18        JUMP TO USER ROUTINE
```

```
                        624  **      INIT - INITIALIZE SYSTEM
                        625  *
                        626  *       INIT IS CALLED WHENEVER A HARDWARE MASTER-CLEAR IS INITIATED.
                        627  *
                        628  *       SETUP PAM/8 CONTROL CELLS IN RAM.
                        629  *       DECODE HOW MUCH MEMORY EXISTS, SETUP STACKPOINTER, AND
                        630  *       ENTER THE MONITOR LOOP.
                        631  *
                        632  *       ENTRY   FROM MASTER CLEAR
                        633  *       EXIT    INTO PAM/8 MAIN LOOP
                        634
                        635
000.073  032            636  INIT    LDAX    D               COPY *PRSROM* INTO RAM
000.074  167            637          MOV     M,A             MOVE BYTE
000.075  053            638          DCX     H               DECREMENT DESTINATION
000.076  034            639          INR     E               INCREMENT SOURCE
000.077  302 073 000     640          JNZ     INIT            IF NOT DONE
                        641
004.000                 642  SINCR   EQU     4000A           SEARCH INCREMENT
                        643
000.102  026 004        644          MVI     D,SINCR/256     (DE) = SEARCH INCREMENT
000.104  041 000 040    645          LXI     H,START         (HL) = FIRST RAM                /RAM8GO JUN80/
                        646
                        647  *       DETERMINE MEMORY LIMIT.
                        648
000.107  303 000 004    649  INIT1   JMP     XINIT1          JUMP TO FREE SPACE              /RAM8GO JUN80/
                        650
000.112  110 105 101    651          DB      'HEATH'                                         /RAM8GO JUN80/
000.000                 652          ERRNZ   *-000117A                                       /RAM8GO JUN80/
                        653
                        654  *       RETURN TO INLINE CODE WITH HL SET TO FIRST NON-EXISTANT LOCATION
                        655
000.117  053            656  INIT2   DCX     H
000.120  371            657          SPHL                    SET STACKPOINTER = MEMORY LIMIT -1
000.121  041 364 007    658          LXI     H,DEFPC                                         /Ram8Go 2/
000.124  345            659          PUSH    H               Set *PC* value on stack         /Ram8Go 2/
000.125  315 254 007    660          CALL    PATCH1          Tape UART/Auto-Boot             /Ram8Go 2/
000.130  345            661          PUSH    H               Set Return Address              /Ram8Go 2/
000.131  257            662          XRA     A               Leave addresses the same & A=0  /Ram8Go 2/
```

```
                                665  **      SAVALL - SAVE ALL REGISTERS ON STACK.
                                666  *
                                667  *       SAVALL IS CALLED WHEN AN INTERRUPT IS ACCEPTED, IN ORDER TO
                                668  *       SAVE THE CONTENTS OF THE REGISTERS ON THE STACK.
                                669  *
                                670  *       ENTRY   CALLED DIRECTLY FROM INTERRUPT ROUTINE.
                                671  *       EXIT    ALL REGISTERS PUSHED ON STACK,
                                672  *               IF NOT YET IN MONITOR MODE, REGPTR = ADDRESS OF REGISTERS
                                673  *               ON STACK.
                                674  *               (DE) = ADDRESS OF CTLFLG
                                675
                                676
000.132  343                    677  SAVALL  XTHL                    SET H,L ON STACK TOP
000.133  325                    678          PUSH    D
000.134  305                    679          PUSH    B
000.135  365                    680          PUSH    PSW
000.136  353                    681          XCHG                    (D,E) = RETURN ADDRESS
000.137  041 012 000            682          LXI     H,10
000.142  071                    683          DAD     SP              (H,L) = ADDRESS OF USERS SP
000.143  345                    684          PUSH    H               SET ON STACK AS 'REGISTER'
000.144  325                    685          PUSH    D               SET RETURN ADDRESS
000.145  021 011 040            686          LXI     D,CTLFLG
000.150  032                    687          LDAX    D               (A) = CTLFLG
000.151  057                    688          CMA
000.152  346 060                689          ANI     CB.MTL+CB.SSI   SAVE REGISTER ADDR IF USER OR SINGLE-STEP
000.154  310                    690          RZ                      RETURN IF WAS INTERRUPT OF MONITOR LOOP
000.155  041 002 000            691          LXI     H,2
000.160  071                    692          DAD     SP              (H,L) = ADDRESS OF 'STACKPTR' ON STACK
000.161  042 035 040            693          SHLD    REGPTR
000.164  311                    694          RET


                                696  **      CUI - CHECK FOR USER INTERRUPT PROCESSING.
                                697  *
                                698  *       CUI IS CALLED TO SEE IF THE USER HAS SPECIFIED PROCESSING
                                699  *       FOR THE CLOCK INTERRUPT.
                                700
                                701
040.010                         702  .       SET     .MFLAG          REFERENCE TO MFLAG
000.165  012                    703  CUI1    LDAX    B               (A) = .MFLAG
000.000                         704          ERRNZ   UD.CLK-1        CODE ASSUMED = 01
000.166  017                    705          RRC
000.167  334 037 040            706          CC      UIVEC           IF SPECIFIED, TRANSFER TO USER
                                707
                                708  *       RETURN TO PROGRAM FROM INTERRUPT.
                                709
000.172  361                    710  INTXIT  POP     PSW             REMOVE FAKE 'STACK REGISTER'
000.173  361                    711          POP     PSW
000.174  301                    712          POP     B
000.175  321                    713          POP     D
000.176  341                    714          POP     H
000.177  373                    715          EI
000.200  311                    716          RET
```

```
                          719  ***     CLOCK - PROCESS CLOCK INTERRUPT
                          720  *
                          721  *       CLOCK IS ENTERED WHENEVER A MILLISECOND CLOCK INTERRUPT IS
                          722  *       PROCESSED.
                          723  *
                          724  *       TICCNT IS INCREMENTED EVERY INTERRUPT.
                          725
                          726
000.201  052 033 040      727  CLOCK   LHLD    TICCNT
000.204  043              728          INX     H
000.205  042 033 040      729          SHLD    TICCNT          INCREMENT TICCOUNT
                          730
                          731  **      REFRESH FRONT PANEL.
                          732  *
                          733  *       THIS CODE DISPLAYS THE APPROPRIATE PATTERN ON THE
                          734  *       FRONT PANEL LEDS. THE LEDS ARE PAINTED IN REVERSE ORDER,
                          735  *       ONE PER INTERRUPT. FIRST, NUMBER 9 IS LIT, THEN NUMBER 8,
                          736  *       ETC.
                          737
                          738
000.210  041 010 040      739          LXI     H,.MFLAG
000.213  176              740          MOV     A,M
000.214  107              741          MOV     B,A             (B) = CURRENT FLAG
000.215  346 100          742          ANI     UO.NFR          SEE IF FRONT PANEL REFRESH WANTED
000.217  043              743          INX     H
000.000                   744          ERRNZ   CTLFLG-.MFLAG-1
000.220  176              745          MOV     A,M             (A) = CTLFLG
000.221  112              746          MOV     C,D             (C) = 0 IN CASE NO PANEL DISPLAY
000.222  302 237 000      747          JNZ     CLK3            IF NOT
000.225  043              748          INX     H               (H,L) = (REFIND)
000.000                   749          ERRNZ   REFIND-CTLFLG-1
000.226  065              750          DCR     M               DECREMENT DIGIT INDEX
000.227  302 234 000      751          JNZ     CLK2            IF NOT WRAP-AROUND
000.232  066 011          752          MVI     M,9             WRAP DISPLAY AROUND
000.234  136              753  CLK2    MOV     E,M
000.235  031              754          DAD     D               (H,L) = ADDRESS OF PATTERN
000.236  113              755          MOV     C,E
000.237                   756  CLK3    EQU     *               (A) = CTLNLG
000.237  261              757          ORA     C               (A) = INDEX + FIXED BITS
000.240  323 360          758          OUT     OP.DIG          SELECT DIGIT
000.242  176              759          MOV     A,M
000.243  323 361          760          OUT     OP.SEG          SELECT SEGMENT
                          761
                          762  *       SEE IF TIME TO DECODE DISPLAY VALUES.
                          763
000.245  056 033          764          MVI     L,#TICCNT
000.247  176              765          MOV     A,M
000.250  346 037          766          ANI     37Q             EVERY 32 INTERRUPTS
000.252  314 161 003      767          CZ      UFD             UPDATE FRONT PANEL DISPLAYS
                          768
                          769  *       EXIT CLOCK INTERRUPT.
                          770
000.255  001 011 040      771          LXI     B,CTLFLG
000.260  012              772          LDAX    B               (A) = CTLFLG
000.261  346 040          773          ANI     CB.MTL
000.263  302 172 000      774          JNZ     INTXIT          IF IN MONITOR MODE
```

```
000.266  013            775          DCX     B
000.000                  776          ERRNZ   CTLFLG-.MFLAG-1
000.267  012            777          LDAX    B               (A) = .MFLAG
000.000                  778          ERRNZ   UD.HLT-200Q     ASSUME HIGH-ORDER
000.270  027            779          RAL
000.271  332 313 000    780          JC      CLK4            SKIP IT
                         781
                         782  *       NOT IN MONITOR MODE. CHECK FOR HALT
                         783
000.274  076 012        784          MVI     A,10            (A) = INDEX OF *P* REG
000.276  315 052 003    785          CALL    LRA.            LOCATE REGISTER ADDRESS
000.301  136            786          MOV     E,M
000.302  043            787          INX     H
000.303  126            788          MOV     D,M             (D,E) = PC CONTENTS
000.304  033            789          DCX     D
000.305  032            790          LDAX    D
000.306  376 166        791          CPI     MI.HLT          CHECK FOR HALT
000.310  312 322 000    792          JE      ERROR           IF HALT, BE IN MONITOR MODE
                         793
                         794  *       CHECK FOR 'RETURN TO MONITOR' KEY ENTRY.
                         795
000.313                  796  CLK4    EQU     *
000.313  333 360        797          IN      IP.PAD
000.315  376 056        798          CPI     56Q             SEE IF '0' AND '#'
000.317  302 165 000    799          JNE     CUI1            IF NOT, ALLOW USER PROCESSING OF CLOCK
```

```
                                803  ***     ERROR - COMMAND ERROR.
                                804  *
                                805  *       ERROR IS CALLED AS A 'BAIL-OUT' ROUTINE.
                                806  *
                                807  *       IT RESETS THE OPERATIONAL MODE, AND RESTORES THE STACKPOINTER.
                                808  *
                                809  *       ENTRY   NONE
                                810  *       EXIT    TO MTR LOOP
                                811  *               CTLFLG SET
                                812  *               .MFLAG CLEARED
                                813  *       USES    ALL
                                814
                                815
000.322                         816  ERROR   EQU     *
000.322  041 010 040            817          LXI     H,.MFLAG
000.325  176                    818          MOV     A,M             (A) = .MFLAG
000.326  346 275                819          ANI     377Q-U0.DDU-U0.NFR      RE-ENABLE DISPLAYS
000.330  167                    820          MOV     M,A             REPLACE
000.331  043                    821          INX     H
000.332  066 360                822          MVI     M,C8.SSI+C8.MTL+C8.CLI+C8.SPK   RESTORE *CTLFLG*
000.000                         823          ERRNZ   CTLFLG-.MFLAG-1
000.334  373                    824          EI
000.335  052 035 040            825          LHLD    REGPTR
000.340  371                    826          SPHL                    RESTORE STACK POINTER TO EMPTY STATE
000.341  315 136 002            827          CALL    ALARM           ALARM FOR 200 MS


                                829  **      MTR - MONITOR LOOP.
                                830  *
                                831  *       THIS IS THE MAIN EXECUTIVE LOOP FOR THE FRONT PANEL EMULATOR.
                                832
                                833
000.344                         834  MTR     EQU     *
000.344  373                    835          EI
                                836
000.345  041 345 000            837  MTR1    LXI     H,MTR1
000.350  345                    838          PUSH    H               SET 'MTR1' AS RETURN ADDRESS
000.351  001 007 040            839          LXI     B,DSPMOD                (BC) = #DSPMOD
000.354  012                    840          LDAX    B
000.355  346 001                841          ANI     1               (A) = 1 IF ALTER
000.357  057                    842          CMA
000.360  062 006 040            843          STA     DSPROT          SET FLAG BIT IF ALTER
                                844
                                845  *       READ KEY
                                846
000.363  315 260 003            847          CALL    RCK             READ CONSOLE KEYSET
000.366  052 024 040            848          LHLD    ABUSS
000.371  376 012                849          CPI     10
000.373  322 005 001            850          JNC     MTR4            IF IN 'ALWAYS VALID' GROUP
000.376  137                    851          MOV     E,A             SAVE VALUE
040.007                         852  .       SET     DSPMOD
000.377  012                    853          LDAX    B               (A) = DSPMOD
001.000  017                    854          RRC
001.001  332 051 001            855          JC      MTR5            IF IN ALTER MODE
```

```
001.004  173            856          MOV     A,E             (A) = CODE
                        857
                        858  *       HAVE A COMMAND (NOT A VALUE)
                        859
001.005  326 004        860  MTR4    SUI     4               (A) = COMMAND
001.007  332 160 004    861          JC      EXTCMD          Extended Commands          /Ram8Go 2/
001.012  137            862          MOV     E,A
001.013  345            863          PUSH    H               SAVE ABUSS VALUE
001.014  041 035 001    864          LXI     H,MTRA
001.017  026 000        865          MVI     D,0
001.021  031            866          DAD     D               (H,L) = ADDRESS OF TABLE ENTRY
001.022  136            867          MOV     E,M
001.023  031            868          DAD     D               (H,L) = ADDRESS OF PROCESSOR
001.024  343            869          XTHL                    SET ADDRESS, (H,L) = (ABUSS)
001.025  021 005 040    870          LXI     D,REGI          (D,E) = ADDRESS OF REG INDEX
040.007                 871  .       SET     DSPMOD
001.030  012            872          LDAX    B               (A) = DSPMOD
001.031  346 002        873          ANI     2               SET 'Z' IF MEMORY
001.033  012            874          LDAX    B               (A) = DSPMOD
001.034  311            875          RET                     JUMP TO PROCESSOR
                        876
                        877
001.035                 878  MTRA    EQU     *               JUMP TABLE
001.035  165            879          DB      GO-*            4 - GO
001.036  141            880          DB      IN-*            5 - INPUT
001.037  143            881          DB      OUT-*           6 - OUTPUT
001.040  165            882          DB      SSTEP-*         7 - SINGLE STEP
001.041  220            883          DB      RMEM-*          8 - CASSETTE LOAD
001.042  332            884          DB      WMEM-*          9 - CASSETTE DUMP
001.043  067            885          DB      NEXT-*          + - NEXT
001.044  104            886          DB      LAST-*          - - LAST
001.045  102            887          DB      ABORT-*         * - ABORT
001.046  060            888          DB      R$W-*           / - DISPLAY/ALTER
001.047  116            889          DB      MEMM-*          # - MEMORY MODE
001.050  034            890          DB      REGM-*          . - REGISTER MODE


                        892  **      PROCESS MEMORY/REGISTER ALTERATIONS.
                        893  *
                        894  *       THIS CODE IS ENTERED IF
                        895  *
                        896  *       1) AM IN ALTER MODE, AND
                        897  *       2) A KEY FROM 0-7 WAS ENTERED.
                        898
001.051  017            899  MTR5    RRC
001.052  173            900          MOV     A,E             (A) = VALUE
001.053  332 072 001    901          JC      MTR6            IS REGISTER
001.056  067            902          STC                     INDICATE 1ST DIGIT IS IN (A)
001.057  315 066 003    903          CALL    IO8             INPUT OCTAL BYTE
001.062  043            904          INX     H               DISPLAY NEXT LOCATION
```

```
                          906  **      SAE - STORE ABUSS AND EXIT.
                          907  *
                          908  *       ENTRY   (HL) = ABUSS VALUE
                          909  *       EXIT    TO (RET)
                          910  *       USES    NONE
                          911
001.063  042 024 040      912  SAE     SHLD    ABUSS
001.066  311              913          RET
                          914
000.000                   915          ERRNZ   *-1067A
001.067  303 066 007      916  H89PIN  JMP     PIN             H89 Compatible PIN routine      /Ram8Go 2/
                          917
001.072  365              918  MTR6    PUSH    PSW             SAVE CODE
001.073  315 047 003      919          CALL    LRA             LOCATE REGISTER ADDRESS
001.076  247              920          ANA     A
001.077  303 274 007      921          JMP     PATCH2                                  /Ram8Go 2/
001.102                   922          DS      2               Reserve Space           /Ram8Go 2/
000.000                   923          ERRNZ   *-1104A         Insure good routine addresses   /Ram8Go 2/
```

```
                          927  **      REGM - ENTER REGISTER DISPLAY MODE.
                          928  *
                          929  *       ENTRY   (A) = DSPMOD
                          930
001.104  076 002          931  REGM    MVI     A,2             SET DISPLAY REGISTER MODE
040.007                   932  .       SET     DSPMOD
001.106  002              933          STAX    B               SET DISPLAY REGISTER MODE
000.000                   934          ERRNZ   DSPMOD-DSPROT-1
001.107  013              935          DCX     B               (BC) = #DSPROT
001.110  257              936          XRA     A
001.111  002              937          STAX    B               SET ALL PERIODS ON
001.112  315 260 003      938          CALL    RCK             READ KEY ENTRY
001.115  075              939          DCR     A               DISPLACE
001.116  376 006          940          CPI     6
001.120  322 322 000      941          JNC     ERROR           NOT 1-6
001.123  007              942          RLC
001.124  022              943          STAX    D               SET NEW REG IND
040.005                   944  .       SET     REGI
001.125  311              945          RET


                          947  **      R$W - TOGGLE DISPLAY/ALTER MODE.
                          948  *
                          949  *       ENTRY   (A) = DSPMOD
                          950  *               (BC) = ADDRESS OF DSPMOD
                          951
040.007                   952  .       SET     DSPMOD
001.126  356 001          953  R$W     XRI     1
001.130  002              954          STAX    B
001.131  311              955          RET


                          957  **      NEXT - INCREMENT DISPLAY ELEMENT.
                          958  *
                          959  *       ENTRY   (HL) = (ABUSS)
                          960  *               (DE) = ADDRESS OF REGIND
                          961
001.132  043              962  NEXT    INX     H
001.133  312 063 001      963          JZ      SAE             IF MEMORY, STORE ABUSS AND EXIT
                          964
                          965  *       IS REGISTER MODE.
                          966
040.005                   967  .       SET     REGI
001.136  032              968          LDAX    D               (A) = REGI
001.137  306 002          969          ADI     2               INCREMENT REG INNEX
001.141  022              970          STAX    D               WRAP TO *SP*
001.142  376 014          971          CPI     12
001.144  330              972          RC                      IF NOT TOO LARGE, EXIT
001.145  257              973          XRA     A               OVERFLOW
001.146  022              974          STAX    D
001.147  311              975  ABORT   RET
```

```
                                977  **      LAST - DECREMENT DISPLAY ELEMENT.
                                978  *
                                979  *       ENTRY   (HL) = (ABUSS)
                                980  *               (DE) = ADDRESS OF REGIND
                                981
                                982
001.150  053                    983  LAST    DCX     H
001.151  312 063 001            984          JZ      SAE             IF MEMORY, STORE AND EXIT
                                985
                                986  *       IS REGISTER MODE.
                                987
040.005                         988  .       SET     REGI
001.154  032                    989  LST2    LDAX    D               (A) = REGI
001.155  326 002                990          SUI     2
001.157  022                    991          STAX    D
001.160  320                    992          RNC                     IF OK
001.161  076 012                993          MVI     A,10            UNDERFLOW TO *PC*
001.163  022                    994          STAX    D
001.164  311                    995          RET
                                996


                                998  **      MEMM - ENTER DISPLAY MEMORY MODE.
                                999  *
                               1000  *       ENTRY   (BC) = ADDRESS OF DSPMOD
                               1001
001.165  257                   1002  MEMM    XRA     A               (A) = 0
040.007                        1003  .       SET     DSPMOD
001.166  002                   1004          STAX    B               SET DISPLAY MEMORY MODE
000.000                        1005          ERRNZ   DSPMOD-DSPROT-1
001.167  013                   1006          DCX     B               (BC) = #DSPROT
001.170  002                   1007          STAX    B               SET ALL PERIODS ON
001.171  041 025 040           1008          LXI     H,ABUSS+1
001.174  303 062 003           1009          JMP     IOA             INPUT OCTAL ADDRESS


                               1011  **      IN - INPUT DATA BYTE.
                               1012  *
                               1013
                               1014  **      OUT - OUTPUT DATA BYTE.
                               1015  *
                               1016  *       ENTRY   (HL) = (ABUSS)
                               1017
001.177  006 333               1018  IN      MVI     B,MI.IN
001.201  021                   1019          DB      MI.LXID         SKIP NEXT INSTRUCTION
001.202  006 323               1020  OUT     MVI     B,MI.OUT
001.204  174                   1021          MOV     A,H             (A) = VALUE
001.205  145                   1022          MOV     H,L             (H) = PORT
001.206  150                   1023          MOV     L,B             (L) = IN/OUT INSTRUCTION
001.207  042 002 040           1024          SHLD    IOWRK
001.212  315 002 040           1025          CALL    IOWRK           PERFORM IO
001.215  154                   1026          MOV     L,H             (L) = PORT
001.216  147                   1027          MOV     H,A             (H) = VALUE
```

```
001.217  303 063 001  1028          JMP     SAE             STORE ABUSS AND EXIT
```

```
                        1033  **      GO - RETURN TO USER MODE
                        1034  *
                        1035  *       ENTRY   NONE
                        1036
001.222  303 063 000   1037  GO      JMP     GO.             ROUTINE IS IN WASTE SPACE


                        1039  **      SSTEP - SINGLE STEP INSTRUCTION.
                        1040  *
                        1041  *       ENTRY   NONE
                        1042
001.225                 1043  SSTEP   EQU     *               SINGLE STEP
001.225  363            1044          DI                      DISABLE INTERRUPTS UNTIL THE RIGHT TIME
001.226  072 011 040    1045          LDA     CTLFLG
001.231  356 020        1046          XRI     CB.SSI          CLEAR SINGLE STEP INHIBIT
001.233  323 360        1047          OUT     OP.CTL          PRIME SINGLE STEP INTERRUPT
001.235  062 011 040    1048  SST1    STA     CTLFLG          SET NEW FLAG VALUES
001.240  341            1049          POP     H               CLEAN STACK
001.241  303 172 000    1050          JMP     INTXIT          RETURN TO USER ROUTINE FOR STEP


                        1052  **      STPRTN - SINGLE STEP RETURN
                        1053
001.244                 1054  STPRTN  EQU     *
001.244  366 020        1055          ORI     CB.SSI          DISABLE SINGLE STEP INTERRUPTION
001.246  323 360        1056          OUT     OP.CTL          TURN OFF SINGLE STEP ENABLE
040.011                 1057  .       SET     CTLFLG
001.250  022            1058          STAX    D
001.251  346 040        1059          ANI     CB.MTL          SEE IF IN MONITOR MODE
001.253  302 344 000    1060          JNZ     MTR
001.256  303 042 040    1061          JMP     UIVEC+3         TRANSFER TO USER'S ROUTINE


                        1063  **      RMEM - LOAD MEMORY FROM TAPE.
                        1064  *
                        1065
001.261  041 244 002    1066  RMEM    LXI     H,TPABT
001.264  042 031 040    1067          SHLD    TPERRX          SETUP ERROR EXIT ADDRESS
                        1068  *       JMP     LOAD
```

```
                        1070  ***     LOAD - LOAD MEMORY FROM TAPE.
                        1071  *
                        1072  *       READ THE NEXT RECORD FROM THE CASSETTE TAPE.
                        1073  *
                        1074  *       USE THE LOAD ADDRESS IN THE TAPE RECORD.
                        1075  *
                        1076  *       ENTRY   (HL) = ERROR EXIT ADDRESS
                        1077  *       EXIT    USER P-REG (IN STACK) SET TO ENTRY ADDRESS
                        1078  *               TO CALLER IF ALL OK
                        1079  *               TO ERROR EXIT IF TAPE ERRORS DETECTED.
                        1080
                        1081
001.267                 1082  LOAD    EQU     *
001.267  001 000 376    1083          LXI     B,1000A-RT.MI*256-256   (BC) = - REQUIRED TYPE AND #
001.272  315 265 002    1084  LOA0    CALL    SRS             SCAN FOR RECORD START
001.275  157            1085          MOV     L,A             (HL) = COUNT
001.276  353            1086          XCHG                    (DE) = COUNT, (HL) = TYPE AND #
001.277  015            1087          DCR     C               (C) = - NEXT #
001.300  011            1088          DAD     B
001.301  174            1089          MOV     A,H
001.302  305            1090          PUSH    B               SAVE TYPE AND #
001.303  365            1091          PUSH    PSW             SAVE TYPE CODE
001.304  346 177        1092          ANI     177Q            CLEAR END FLAG BIT
001.306  265            1093          ORA     L
001.307  076 002        1094          MVI     A,2             SEQUENCE ERROR
001.311  302 205 002    1095          JNE     TPERR           IF NOT RIGHT TYPE OR SEQUENCE
001.314  315 325 002    1096          CALL    RNP             READ ADDR
001.317  104            1097          MOV     B,H
001.320  117            1098          MOV     C,A             (BC) = P-REG ADDRESS
001.321  076 012        1099          MVI     A,10
001.323  325            1100          PUSH    D               SAVE (DE)
001.324  315 052 003    1101          CALL    LRA.            LOCATE REG ADDRESS
001.327  321            1102          POP     D               RESTORE (DE)
001.330  161            1103          MOV     M,C             SET P-REG IN MEM
001.331  043            1104          INX     H
001.332  160            1105          MOV     M,B
001.333  315 325 002    1106          CALL    RNP             READ ADDRESS
001.336  157            1107          MOV     L,A             (HL) = ADDRESS, (DE) = COUNT
001.337  042 000 040    1108          SHLD    START
                        1109
001.342  315 331 002    1110  LOA1    CALL    RNB             READ BYTE
001.345  167            1111          MOV     M,A
001.346  042 024 040    1112          SHLD    ABUSS           SET ABUSS FOR DISPLAY
001.351  043            1113          INX     H
001.352  033            1114          DCX     D
001.353  172            1115          MOV     A,D
001.354  263            1116          ORA     E
001.355  302 342 001    1117          JNZ     LOA1            IF MORE TO GO
                        1118
001.360  315 172 002    1119          CALL    CTC             CHECK  TAPE CHECKSUM
                        1120
                        1121  *       READ NEXT BLOCK
                        1122
001.363  361            1123          POP     PSW             (A) = FILE TYPE BYTE
001.364  301            1124          POP     B               (BC) = -(LAST TYPE, LAST #)
001.365  007            1125          RLC
```

```
001.366  332 133 002  1126          JC      TFT          ALL DONE - TURN OFF TAPE
001.371  303 272 001  1127          JMP     LOAD         READ ANOTHER RECORD
```

```
                          1130  ***     DUMP - DUMP MEMORY TO MAG TAPE.
                          1131  *
                          1132  *       DUMP SPECIFIED MEMORY RANGE TO MAG TAPE.
                          1133  *
                          1134  *       ENTRY   (START) = START ADDRESS
                          1135  *               (ABUSS) = END ADDRESS
                          1136  *               USER PC = ENTRY POINT ADDRESS
                          1137  *       EXIT    TO CALLER.
                          1138
                          1139
     001.374              1140  WMEM    EQU     *
     001.374  041 244 002 1141          LXI     H,TPABT
     001.377  042 031 040 1142          SHLD    TPERRX          SETUP ERROR EXIT
                          1143
     002.002  076 001     1144  DUMP    MVI     A,UCI.TE
     002.004  323 371     1145          OUT     OP.TPC          SETUP TAPE CONTROL
     002.006  076 026     1146          MVI     A,A.SYN
     002.010  046 040     1147          MVI     H,32            (H) = # OF SYNC CHARACTERS
     002.012  315 024 003 1148  WME1    CALL    WNB
     002.015  045         1149          DCR     H
     002.016  302 012 002 1150          JNZ     WME1            WRITE SYN HEADER
     002.021  076 002     1151          MVI     A,A.STX
     002.023  315 024 003 1152          CALL    WNB             WRITE STX
     002.026  154         1153          MOV     L,H             (HL) = 00
     002.027  042 027 040 1154          SHLD    CRCSUM          CLEAR CRC 16
     002.032  041 001 201 1155          LXI     H,RT.MI+80H*256+1       FIRST AND LAST MI RECORD
     002.035  315 017 003 1156          CALL    WNP             WRITE HEADER
     002.040  052 000 040 1157          LHLD    START
     002.043  353         1158          XCHG                    (D,E) = START ADDRESS
     002.044  052 024 040 1159          LHLD    ABUSS           (H,L) = STOP ADDR
     002.047  043         1160          INX     H               COMPUTE WITH STOP+1
     002.050  175         1161          MOV     A,L
     002.051  223         1162          SUB     E
     002.052  157         1163          MOV     L,A
     002.053  174         1164          MOV     A,H
     002.054  232         1165          SBB     D
     002.055  147         1166          MOV     H,A             (HL) = COUNT
     002.056  315 017 003 1167          CALL    WNP             WRITE COUNT
     002.061  345         1168          PUSH    H
     002.062  076 012     1169          MVI     A,10
     002.064  325         1170          PUSH    D               SAVE (DE)
     002.065  315 052 003 1171          CALL    LRA.            LOCATE P-REG ADDRESS
     002.070  176         1172          MOV     A,M
     002.071  043         1173          INX     H
     002.072  146         1174          MOV     H,M
     002.073  157         1175          MOV     L,A             (HL) = CONTENTS OF PC
     002.074  315 017 003 1176          CALL    WNP             WRITE HEADER
     002.077  341         1177          POP     H               (HL) = ADDRESS
     002.100  321         1178          POP     D               (DE) = COUNT
     002.101  315 017 003 1179          CALL    WNP
                          1180
     002.104  176         1181  WME2    MOV     A,M
     002.105  315 024 003 1182          CALL    WNB             WRITE BYTE
     002.110  042 024 040 1183          SHLD    ABUSS           SET ADDRESS FOR DISPLAY
     002.113  043         1184          INX     H
     002.114  033         1185          DCX     D
```

```
002.115  172           1186           MOV     A,D
002.116  263           1187           ORA     E
002.117  302 104 002   1188           JNZ     WME2            IF MORE TO GO
                       1189
                       1190   *       WRITE CHECKSUM
                       1191
002.122  052 027 040   1192           LHLD    CRCSUM
002.125  315 017 003   1193           CALL    WNP             WRITE IT
002.130  315 017 003   1194           CALL    WNP             FLUSH CHECKSUM
                       1195   *       JMP     TFT


                       1197   **      TFT - TURN OFF TAPE.
                       1198   *
                       1199   *       STOP THE TAPE TRANSPORT.
                       1200   *
                       1201
002.133  257           1202   TFT     XRA     A
002.134  323 371       1203           OUT     OP.TPC          TURN OFF TAPE


                       1205   **      HORN - MAKE NOISE.
                       1206   *
                       1207   *       ENTRY   (A) = (MILLISECOND COUNT)/2
                       1208   *       EXIT    NONE
                       1209   *       USES    A,F
                       1210
                       1211
002.136  076 144       1212   ALARM   MVI     A,200/2         200 MS BEEP
002.140  365           1213   HORN    PUSH    PSW
002.141  076 200       1214           MVI     A,CB.SPK        TURN ON SPEAKER
                       1215
002.143  343           1216   HRNO    XTHL                    SAVE (HL), (H)  = COUNT
002.144  325           1217           PUSH    D               SAVE (DE)
002.145  353           1218           XCHG                    (D) = LOOP COUNT
002.146  041 011 040   1219           LXI     H,CTLFLG
002.151  256           1220           XRA     M
002.152  136           1221           MOV     E,M             (E) = OLD CTLFLG VALUE
002.153  167           1222           MOV     M,A             TURN ON HORN
002.154  056 033       1223           MVI     L,#TICCNT
                       1224
002.156  172           1225           MOV     A,D             (A) = CYCLE COUNT
002.157  206           1226           ADD     M
002.160  276           1227   HRN2    CMP     M               WAIT REQUIRED TICCOUNTS
002.161  302 160 002   1228           JNE     HRN2
002.164  056 011       1229           MVI     L,#CTLFLG
002.166  163           1230           MOV     M,E             TURN HORN OFF
002.167  321           1231           POP     D
002.170  341           1232           POP     H
002.171  311           1233           RET
```

```
                                1238  **      CTC - VERIFY CHECKSUM.
                                1239  *
                                1240  *       ENTRY   TAPE JUST BEFORE CRC
                                1241  *       EXIT    TO CALLER IF OK
                                1242  *               TO *TPERR* IF BAD
                                1243  *       USES    A,F,H,L
                                1244
                                1245
      002.172  315 325 002      1246  CTC     CALL    RNP             READ NEXT PAIR
      002.175  052 027 040      1247          LHLD    CRCSUM
      002.200  174              1248          MOV     A,H
      002.201  265              1249          ORA     L
      002.202  310              1250          RZ                      RETURN OF OK
      002.203  076 001          1251          MVI     A,1             CHECKSUM ERROR
                                1252  *       JMP     TPERR           (B) = CODE


                                1254  **      TPERR - PROCESS TAPE ERROR.
                                1255  *
                                1256  *       DISPLAY ERR NUMBER IN LOW BYTE OF ABUSS
                                1257  *
                                1258  *       IF ERROR NUMBER EVEN, DONT ALLOW #
                                1259  *       IF ERROR NUMBER ODD, ALLOW #
                                1260  *
                                1261  *       ENTRY   (B) = PATTERN
                                1262
                                1263
      002.205  062 024 040      1264  TPERR   STA     ABUSS
      002.210  107              1265          MOV     B,A             (B) = CODE
      002.211  315 133 002      1266          CALL    TFT             TURN OFF TAPE
                                1267
                                1268  *       IS #, RETURN (IF PARITY ERROR)
                                1269
      002.214  346              1270          DB      MI.ANI          FALL THROUGH WITH CARRY CLEAR
      002.215  170              1271  TER3    MOV     A,B
                                1272
      002.216  017              1273          RRC
      002.217  330              1274          RC                      RETURN IF OK
                                1275
                                1276  *       BEEP AND FLASH ERROR NUMBER
                                1277
      002.220  334 136 002      1278  TER1    CC      ALARM           ALARM IF PROPER TIME
      002.223  315 252 002      1279          CALL    TPXIT           SEE IF *
      002.226  333 360          1280          IN      IP.PAD
      002.230  376 057          1281          CPI     00101111B       CHECK FOR #
      002.232  312 215 002      1282          JE      TER3            IF #
      002.235  072 034 040      1283          LDA     TICCNT+1
      002.240  037              1284          RAR                     'C' SET IF 1/2 SECOND
      002.241  303 220 002      1285          JMP     TER1
```

```
                        1287  **      TPABT - ABORT TAPE LOAD OR DUMP.
                        1288  *
                        1289  *       ENTERED WHEN LOADING OR DUMPING, AND THE '*' KEY
                        1290  *       IS STRUCK.
                        1291
                        1292
002.244  257           1293  TPABT   XRA     A
002.245  323 371       1294          OUT     OP.TPC          OFF TAPE
002.247  303 322 000   1295          JMP     ERROR


                        1297  **      TPXIT - CHECK FOR USER FORCED EXIT.
                        1298  *
                        1299  *       TPXIT CHECKS FOR AN '*' KEYPAD ENTRY. IF SO, TAKE
                        1300  *       THE TAPE DRIVER ABNORMAL EXIT.
                        1301  *
                        1302  *       ENTRY   NONE
                        1303  *       EXIT    TO *RET* IF NOT '*'
                        1304  *                (A) = PORT STATUS
                        1305  *               TO (TPERRX) IF '*' DOWN
                        1306  *       USES    A,F
                        1307
                        1308
002.252  333 360       1309  TPXIT   IN      IP.PAD
002.254  376 157       1310          CPI     01101111B       *
002.256  333 371       1311          IN      IP.TPC          READ TAPE STATUS
002.260  300           1312          RNE                     NOT '*', RETURN WITH STATUS
002.261  052 031 040   1313          LHLD    TPERRX
002.264  351           1314          PCHL                    ENTER (TPERRX)


                        1316  **      SRS - SCAN RECORD START
                        1317  *
                        1318  *       SRS READS BYTES UNTIL IT RECOGNIZES THE START OF A RECORD.
                        1319  *
                        1320  *       THIS REQUIRES
                        1321  *       AT LEAST 10 SYNC CHARACTERS
                        1322  *       1 STX CHARACTER.
                        1323  *
                        1324  *       THE CRC-16 IS THEN INITIALIZED.
                        1325  *
                        1326  *       ENTRY   NONE
                        1327  *       EXIT    TAPE POSITIONED (AND MOVING), CRCSUM =0
                        1328  *                (DE) = HEADER BYTES
                        1329  *                (HA) = RECORD COUNT
                        1330  *       USES    A,F,D,E,H,L
                        1331
                        1332
002.265                 1333  SRS     EQU     *
002.265  026 000       1334  SRS1    MVI     D,0
002.267  142           1335          MOV     H,D
002.270  152           1336          MOV     L,D             (HL) = 0
```

```
002.271  315 331 002  1337  SRS2   CALL   RNB            READ NEXT BYTE
002.274  024          1338         INR    D
002.275  376 026      1339         CPI    A.SYN
002.277  312 271 002  1340         JE     SRS2           HAVE SYN
002.302  376 002      1341         CPI    A.STX
002.304  302 265 002  1342         JNE    SRS1           NOT STX - START OVER
                      1343
002.307  076 012      1344         MVI    A,10
002.311  272          1345         CMP    D              SEE IF ENOUGH SYN CHARACTERS
002.312  322 265 002  1346         JNC    SRS1           NOT ENOUGH
002.315  042 027 040  1347         SHLD   CRCSUM         CLEAR CRC-16
002.320  315 325 002  1348         CALL   RNP            READ LEADER
002.323  124          1349         MOV    D,H
002.324  137          1350         MOV    E,A
                      1351  *      JMP    RNP            READ COUNT


                      1353  **     RNP - READ NEXT PAIR.
                      1354  *
                      1355  *      RNP READS THE NEXT TWO BYTES FROM THE INPUT DEVICE.
                      1356  *
                      1357  *      ENTRY  NONE
                      1358  *      EXIT   (H,A) = BYTE PAIR
                      1359  *      USES   A,F,H
                      1360
                      1361
002.325  315 331 002  1362  RNP    CALL   RNB            READ NEXT BYTE
002.330  147          1363         MOV    H,A
                      1364  *      JMP    RNB            READ NEXT BYTE


                      1366  **     RNB - READ NEXT BYTE
                      1367  *
                      1368  *      RNB READS THE NEXT SINGLE BYTE FROM THE INPUT DEVICE.
                      1369  *      THE CHECKSUM IS TAKEN FOR THE CHARACTER.
                      1370  *
                      1371  *      ENTRY  NONE
                      1372  *      EXIT   (A) = CHARACTER
                      1373  *      USES   A,F
                      1374
                      1375
002.331  076 064      1376  RNB    MVI    A,UCI.RO+UCI.ER+UCI.RE  TURN ON READER FOR NEXT BYTE
002.333  323 371      1377         OUT    OP.TPC
002.335  315 252 002  1378  RNB1   CALL   TPXIT          CHECK FOR *, READ STATUS
002.340  346 002      1379         ANI    USR.RXR
002.342  312 335 002  1380         JZ     RNB1           IF NOT READY
002.345  333 370      1381         IN     IP.TPD         INPUT DATA
                      1382  *      JMP    CRC            CHECKSUM
```

```
                        1384  **      CRC - COMPUTE CRC-16
                        1385  *
                        1386  *       CRC COMPUTES A CRC-16 CHECKSUM FROM THE POLYNOMIAL
                        1387  *
                        1388  *       (X + 1) * (X^15 + X + 1)
                        1389  *
                        1390  *       SINCE THE CHECKSUM GENERATED IS A DIVISION REMAINDER,
                        1391  *       A CHECKSUMED DATA SEQUENCE CAN BE VERIFIED BY RUNNING
                        1392  *       THE DATA THROUGH CRC, AND THEN RUNNING THE PREVIOUSLY OBTAINED
                        1393  *       CHECKSUM THROUGH CRC. THE RESULTANT CHECKSUM SHOULD BE 0.
                        1394  *
                        1395  *       ENTRY   (CRCSUM) = CURRENT CHECKSUM
                        1396  *               (A) = BYTE
                        1397  *       EXIT    (CRCSUM) UPDATED
                        1398  *               (A) UNCHANGED.
                        1399  *       USES    F
                        1400
                        1401
002.347  305           1402  CRC     PUSH    B               SAVE (BC)
002.350  006 010       1403          MVI     B,8             (B) = BIT COUNT
002.352  345           1404          PUSH    H
002.353  052 027 040   1405          LHLD    CRCSUM
002.356  007           1406  CRC1    RLC
002.357  117           1407          MOV     C,A             (C) = BIT
002.360  175           1408          MOV     A,L
002.361  207           1409          ADD     A
002.362  157           1410          MOV     L,A
002.363  174           1411          MOV     A,H
002.364  027           1412          RAL
002.365  147           1413          MOV     H,A
002.366  027           1414          RAL
002.367  251           1415          XRA     C
002.370  017           1416          RRC
002.371  322 004 003   1417          JNC     CRC2            IF NOT TO XOR
002.374  174           1418          MOV     A,H
002.375  356 200       1419          XRI     200Q
002.377  147           1420          MOV     H,A
003.000  175           1421          MOV     A,L
003.001  356 005       1422          XRI     5Q
003.003  157           1423          MOV     L,A
003.004  171           1424  CRC2    MOV     A,C
003.005  005           1425          DCR     B
003.006  302 356 002   1426          JNZ     CRC1            IF MORE TO GO
003.011  042 027 040   1427          SHLD    CRCSUM
003.014  341           1428          POP     H               RESTORE (HL)
003.015  301           1429          POP     B               RESTORE (BC)
003.016  311           1430          RET                     EXIT
```

```
                        1432  **      WNP - WRITE NEXT PAIR.
                        1433  *
                        1434  *       WPT WRITES THE NEXT TWO BYTES TO THE CASSETTE DRIVE.
                        1435  *
                        1436  *       ENTRY   (H,L) = BYTES
                        1437  *       EXIT    WRITTEN.
                        1438  *       USES    A,F
                        1439
                        1440
003.017  174            1441  WNP     MOV     A,H
003.020  315 024 003    1442          CALL    WNB
003.023  175            1443          MOV     A,L
                        1444  *       JMP     WNB             WRITE NEXT BYTE


                        1446  **      WNB - WRITE BYTE
                        1447  *
                        1448  *       WNB WRITES THE NEXT BYTE TO THE CASSETTE TAPE.
                        1449  *
                        1450  *       ENTRY   (A) = BYTE
                        1451  *       EXIT    NONE.
                        1452  *       USES    F
                        1453
                        1454
003.024  365            1455  WNB     PUSH    PSW
003.025  315 252 002    1456  WNB1    CALL    TPXIT           CHECK FOR *, READ STATUS
003.030  346 001        1457          ANI     USR.TXR
003.032  312 025 003    1458          JZ      WNB1            IF MORE TO GO
003.035  076 021        1459          MVI     A,UCI.ER+UCI.TE ENABLE TRANSMITTER
003.037  323 371        1460          OUT     OP.TPC          TURN ON TAPE
003.041  361            1461          POP     PSW
003.042  323 370        1462          OUT     OP.TPD          OUTPUT DATA
003.044  303 347 002    1463          JMP     CRC             COMPUTE CRC
```

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


                                1484  **      IOA - INPUT OCTAL ADDRESS.
                                1485  *
                                1486  *       ENTRY   (H,L) = ADDRESS OF RECEPTION DOUBLE BYTE.
                                1487  *       EXIT    TO *RET* IF ERROR.
                                1488  *               TO *RET*+1 IF OK, VALUE IN MEMORY.
                                1489  *       USES    A,D,E,H,L,F
                                1490
                                1491
003.062  315 066 003            1492  IOA     CALL    IO8             INPUT BYTE
003.065  053                    1493          DCX     H


                                1495  **      IO8 - INPUT OCTAL BYTE.
                                1496  *
                                1497  *       READ ONE OCTAL BYTE FROM THE KEYSET.
                                1498  *
                                1499  *       ENTRY   (H,L) = ADDRESS OF BYTE TO HOLD VALUE
                                1500  *               'C' SET IF FIRST DIGIT IN (A)
                                1501  *       EXIT    TO *RET* IF ALL OK
                                1502  *               TO *ERROR* IF ERROR
                                1503  *       USES    A,D,E,H,L,F
                                1504
                                1505
                                1506
003.066  026 003                1507  IO8     MVI     D,3             (D) = DIGIT COUNT
003.070  324 260 003            1508  IO81    CNC     RCK             READ CONSOLE KEYSET
                                1509
003.073  376 010                1510          CPI     8
003.075  322 322 000            1511          JNC     ERROR           IF ILLEGAL DIGIT
                                1512
003.100  137                    1513          MOV     E,A             (E) = VALUE
003.101  176                    1514          MOV     A,M
003.102  007                    1515          RLC                     SHIFT 3
003.103  007                    1516          RLC
```

```
003.104  007           1517           RLC
003.105  346 370       1518           ANI     370Q
003.107  263           1519           ORA     E
003.110  167           1520           MOV     M,A             REPLACE
003.111  025           1521           DCR     D
003.112  302 070 003   1522           JNZ     IOB1            IF NOT DONE
003.115  076 017       1523           MVI     A,30/2          BEEP FOR 30 MS
003.117  303 140 002   1524           JMP     HORN


                       1526   **      DOD - DECODE FOR OCTAL DISPLAY.
                       1527   *
                       1528   *       ENTRY   (H,L) = ADDRESS OF LED REFRESH AREA
                       1529   *               (B) = *OR* PATTERN TO FORCE ON BARS OR PERIODS
                       1530   *               (A) = OCTAL VALUE
                       1531   *       EXIT    (H,L) = NEX DIGIT ADDRESS
                       1532   *       USES    A,B,C,D,H,L
                       1533
                       1534
003.122  325           1535   DOD     PUSH    D
003.123  026 003       1536           MVI     D,DODA/256
003.125  016 003       1537           MVI     C,3
003.127  027           1538   DOD1    RAL                     LEFT 3 PLACES
003.130  027           1539           RAL
003.131  027           1540           RAL
003.132  365           1541           PUSH    PSW             SAVE FOR NEXT DIGIT
003.133  346 007       1542           ANI     7
003.135  306 356       1543           ADI     #DODA
003.137  137           1544           MOV     E,A             (D) = INDEX
003.140  032           1545           LDAX    D               (A) = PATTERN
003.141  250           1546           XRA     B
003.142  346 177       1547           ANI     177Q
003.144  250           1548           XRA     B
003.145  167           1549           MOV     M,A             SET IN MEMORY
003.146  043           1550           INX     H
003.147  170           1551           MOV     A,B
003.150  007           1552           RLC
003.151  107           1553           MOV     B,A
003.152  361           1554           POP     PSW             (A) = VALUE
003.153  015           1555           DCR     C
003.154  302 127 003   1556           JNZ     DOD1            IF MORE TO GO
003.157  321           1557           POP     D
003.160  311           1558           RET                     RETURN
```

```
                                1561  **      UFD - UPDATE FRONT PANEL DISPLAYS.
                                1562  *
                                1563  *
                                1564  *       UFD IS CALLED BY THE CLOCK INTERRUPT PROCESSOR WHEN IT IS
                                1565  *       TIME TO UPDATE THE DISPLAY CONTENTS. CURRENTLY, THIS IS DONE
                                1566  *       EVERY 32 INTERRUPTS, OR ABOUT 32 TIMES A SECOND.
                                1567  *
                                1568  *       ENTRY   (H,L) = ADDRESS OF REFCNT
                                1569  *       EXIT    NONE
                                1570  *       USES    ALL
                                1571
                                1572
      003.161                   1573  UFD     EQU     *
      003.161  076 002          1574          MVI     A,UO.DDU
      003.163  240              1575          ANA     B
      003.164  300              1576          RNZ                     IF NOT TO HANDLE UPDATE
                                1577
      003.165  056 006          1578          MVI     L,#DSPROT
      003.167  176              1579          MOV     A,M
      003.170  007              1580          RLC
      003.171  167              1581          MOV     M,A             ROTATE PATTERN
      003.172  107              1582          MOV     B,A
      003.173  043              1583          INX     H
      000.000                   1584          ERRNZ   DSPMOD-DSPROT-1
      003.174  176              1585          MOV     A,M             (A) = DSPMOD
      003.175  346 002          1586          ANI     2
      003.177  052 024 040      1587          LHLD    ABUSS
      003.202  312 227 003      1588          JZ      UFD1            IF MEMORY
                                1589
                                1590  *       AM DISPLAYING REGISTERS.
                                1591
      003.205  315 047 003      1592          CALL    LRA             LOCATE REGISTER ADDRESS
      003.210  345              1593          PUSH    H
      003.211  041 342 003      1594          LXI     H,DSPA
      003.214  031              1595          DAD     D               (H,L) = ADDRESS OF REG NAME PATTERNS
      003.215  176              1596          MOV     A,M
      003.216  043              1597          INX     H
      003.217  146              1598          MOV     H,M
      003.220  157              1599          MOV     L,A             (H,L) = REG NAME PATTERN
      003.221  343              1600          XTHL
      003.222  264              1601          ORA     H               CLEAR 'Z'
      003.223  176              1602          MOV     A,M
      003.224  043              1603          INX     H
      003.225  146              1604          MOV     H,M
      003.226  157              1605          MOV     L,A             (HL) = ADDRESS OF REGISTER PAIR CONTENTS
                                1606
                                1607  *       SETUP DISPLAY
                                1608
      003.227  365              1609  UFD1    PUSH    PSW
      003.230  353              1610          XCHG
      003.231  041 013 040      1611          LXI     H,ALEDS
      003.234  172              1612          MOV     A,D
      003.235  315 122 003      1613          CALL    DOD             FORMAT ABANK HIGH HALF
      003.240  173              1614          MOV     A,E
      003.241  315 122 003      1615          CALL    DOD             FORMAT ABANK LOW HALF
      003.244  361              1616          POP     PSW
```

```
003.245  032          1617          LDAX    D
003.246  312 122 003  1618          JZ      DDD             IF MEMORY, DECODE BYTE VALUE
                      1619
                      1620  *       IS REGISTER. SET REGISTER NAME.
                      1621
003.251  066 377      1622          MVI     M,3770          CLEAR DIGIT
003.253  341          1623          POP     H
003.254  042 022 040  1624          SHLD    DLEDS+1
003.257  311          1625          RET
```

```
                        1629  **      RCK - READ CONSOLE KEYSET.
                        1630  *
                        1631  *       RCK IS CALLED TO READ A KEYSTROKE FROM THE CONSOLE KEYSET.
                        1632  *       WHENEVER A KEY IS ACCEPTED.
                        1633  *       RCK PERFORMS DEBOUNCING, AND AUTO-REPEAT. A *BIP* IS SOUNDED
                        1634  *       WHEN A VALUE IS ACCEPTED.
                        1635  *
                        1636  *       KEY PAD VALUES:
                        1637  *
                        1638  *       1111 1110  -  0
                        1639  *       1111 1100  -  1
                        1640  *       1111 1010  -  2
                        1641  *       1111 1000  -  3
                        1642  *       1111 0110  -  4
                        1643  *       1111 0100  -  5
                        1644  *       1111 0010  -  6
                        1645  *       1111 0000  -  7
                        1646  *       1110 1111  -  8
                        1647  *       1100 1111  -  9
                        1648  *       1010 1111  -  +
                        1649  *       1000 1111  -  -
                        1650  *       0110 1111  -  *
                        1651  *       0100 1111  -  /
                        1652  *       0010 1111  -  #
                        1653  *       0000 1111  -  .
                        1654  *
                        1655  *
                        1656  *       ENTRY   NONE
                        1657  *       EXIT    TO CALLER WHEN A KEY IS HIT
                        1658  *               (A) =  0 - '0'
                        1659  *                      1 - '1'
                        1660  *                      2 - '2'
                        1661  *                      3 - '3'
                        1662  *                      4 - '4'
                        1663  *                      5 - '5'
                        1664  *                      6 - '6'
                        1665  *                      7 - '7'
                        1666  *                      8 - '8'
                        1667  *                      9 - '9'
                        1668  *                     10 - '+'
                        1669  *                     11 - '-'
                        1670  *                     12 - '*'
                        1671  *                     13 - '/'
                        1672  *                     14 - '#'
                        1673  *                     15 - '.'
                        1674  *       USES    A,F
                        1675
                        1676
003.260                 1677  RCK     EQU     *
003.260  345            1678          PUSH    H
003.261  305            1679          PUSH    B
003.262  016 024        1680          MVI     C,400/20        WAIT 400 MS
003.264  041 026 040    1681          LXI     H,RCKA
                        1682
003.267  333 360        1683  RCK1    IN      IP.PAD          INPUT PAD VALUE
003.271  107            1684          MOV     B,A             (B) = VALUE
```

```
003.272  076 012       1685           MVI     A,20/2
003.274  315 053 000   1686           CALL    DLY             WAIT 20 MS
003.277  170          1687           MOV     A,B
003.300  276          1688           CMP     M
003.301  302 310 003  1689           JNE     RCK2            HAVE A CHANGE
003.304  015          1690           DCR     C  .
003.305  302 267 003  1691           JNZ     RCK1            WAIT N CYCLES
                       1692
                       1693  *        HAVE KEY VALUE
                       1694
003.310  167          1695  RCK2     MOV     M,A             UPDATE RCKA
003.311  356 376       1696           XRI     376Q            INVERT ALL BUT GROUP 0 FLAG
003.313  017          1697           RRC
003.314  322 326 003  1698           JNC     RCK3            HIT BANK 0
003.317  017          1699           RRC
003.320  017          1700           RRC
003.321  017          1701           RRC
003.322  017          1702           RRC
003.323  322 267 003  1703           JNC     RCK1            NO HIT AT ALL
003.326  107          1704  RCK3     MOV     B,A             (B) = CODE
003.327  076 002       1705           MVI     A,4/2
003.331  315 140 002  1706           CALL    HORN            MAKE BIP
003.334  170          1707           MOV     A,B
003.335  346 017       1708           ANI     17Q
003.337  301          1709           POP     B
003.340  341          1710           POP     H
003.341  311          1711           RET                     RETURN
```

```
                        1715  **      DISPLAY SEGMENT CODING:
                        1716  *
                        1717  *       BYTE = 76 543 210
                        1718  *
                        1719  *        1
                        1720  *       6 2
                        1721  *        0
                        1722  *       5 3
                        1723  *        4
                        1724  *           7


                        1728  **      REGISTER INDEX TO 7-SEGMENT PATTERN
                        1729
     003.342            1730  DSPA    DS      0
     003.342  244 230   1731          DW      1001100010100100B    SP
     003.344  220 234   1732          DW      1001110010010000B    AF
     003.346  206 215   1733          DW      1000110110000110B    BC
     003.350  302 214   1734          DW      1000110011000010B    DE
     003.352  222 217   1735          DW      1000111110010010B    HL
     003.354  230 316   1736          DW      1100111010011000B    PC


                        1738  **      OCTAL TO 7-SETMENT PATTERN
                        1739
     003.356            1740  DODA    DS      0
     003.356  001       1741          DB      00000001B       0
     003.357  163       1742          DB      01110011B       1
     003.360  110       1743          DB      01001000B       2
     003.361  140       1744          DB      01100000B       3
     003.362  062       1745          DB      00110010B       4
     003.363  044       1746          DB      00100100B       5
     003.364  004       1747          DB      00000100B       6
     003.365  161       1748          DB      01110001B       7
     003.366  000       1749          DB      00000000B       8
     003.367  040       1750          DB      00100000B       9


                        1752  **      IO ROUTINES TO BE COPIED INTO AND USED IN RAM.
                        1753  *
                        1754  *       MUST CONTINUE TO 3777A FOR PROPER COPY.
                        1755  *       THE TABLE MUST ALSO BE BACKWARDS TO THE FINAL RAM
                        1756
     003.371            1757          ORG     4000A-7
                        1758
     003.371            1759  PRSROM  EQU     *
     003.371  001       1760          DB      1               REFIND
     003.372  000       1761          DB      0               CTLFLG
     003.373  000       1762          DB      0               .MFLAG
```

```
003.374  002          1763          DB      DM.RR           DSPMOD  DISPLAY REGISTER        /PAM8GO 04MAR80/
003.375  000          1764          DB      0               DSPROT
003.376  000          1765          DB      0               REGI    Show  *SP*              /Ram8Go 2/
003.377  311          1766          DB      MI.RET
                      1767
000.000               1768          ERRNZ *-4000A
```

```
                          1772  **      XINIT1 - SIZE MEMORY                                    /RAM8GO JUN80/
                          1773  *
                          1774  *       XINIT1 IS JUMPED TO DURING PAM8'S MEMORY SIZING
                          1775  *       THIS ROUTINE DIFFERS FROM THE STANDARD PAM8 FUNCTION
                          1776  *       IN THAT IT IS NON-DESTRUCTIVE TO WHAT MAY BE RAM BELOW
                          1777  *       0400000A, AND IT WILL NOT WRAP-AROUND IN A 64K RAM SYSTEM
                          1778  *
                          1779  *       ENTRY   JUMPED TO FROM OLD INIT1
                          1780  *               (DE) = SEARCH INCREMENT
                          1781  *               (HL) = FIRST RAM SEARCH LOCATION
                          1782  *       EXIT    (HL) = FIRST LOCATION WHERE NO RAM FOUND
                          1783  *                      (OR ZERO IF RAM THROUGH 64K)
                          1784  *               (E)  = 0 AS REQUIRED
                          1785
004.000  176              1786  XINIT1  MOV     A,M             GET THE VALUE OF THE CURRENT TRIAL ADDRESS
004.001  065              1787          DCR     M               ATTEMPT TO CHANGE IT
004.002  276              1788          CMP     M               COMPARE IT TO ITS OLD VALUE
004.003  167              1789          MOV     M,A             RESTORE OLD VALUE
004.004  312 117 000      1790          JZ      INIT2           THERE WAS NO CHANGE => NO RAM
004.007  031              1791          DAD     D               INCREMENT THE SEARCH ADDRESS
004.010  322 000 004      1792          JNC     XINIT1          HAS NOT WRAPPED AROUND
004.013  303 117 000      1793          JMP     INIT2           BACK INTO INLINE CODE
```

```
                                1797  **      XINIT - EXTENDED INITIALIZATION                          /Ram8Go 2/
                                1798  *
                                1799  *       DECIDE IF THERE IS RAM AT ZERO,
                                1800  *         IF THERE IS, THEN COPY RAM FRONT PANEL AND H17 ROM TO
                                1801  *               APPROPRIATE POSITIONS
                                1802  *       JUMP BACK TO INLINE INIT
                                1803  *
                                1804  *       Modified to only do one move directly to RAM.            /Ram8Go 2/
                                1805  *
                                1806  *       ENTRY   (DE) = RAM8GO                                    /Ram8Go 2/
                                1807  *
                                1808  *       EXIT    (DE) = PRSROM
                                1809  *               (HL) = PRSRAM+PRSL-1
                                1810  *               RAM AT ZERO SET UP IF PRESENT
                                1811  *
                                1812
      030.000                   1813  H17ROM  EQU     030000A         H17 Rom Address
      010.000                   1814  H17ROML EQU     2*1024          Length of H17 ROM
                                1815
      004.016  257              1816  XINIT   XRA     A
      004.017  062 066 040      1817          STA     CTLFLG2         Initialize the flag
                                1818
                                1819  *       Copy check routine to RAM
                                1820
      004.022  016 012          1821          MVI     C,XINAL
      004.024  021 146 004      1822          LXI     D,XINA
      004.027  041 004 040      1823          LXI     H,XINB
      004.032  032              1824  XIN1    LDAX    D
      004.033  167              1825          MOV     M,A
      004.034  023              1826          INX     D
      004.035  043              1827          INX     H
      004.036  015              1828          DCR     C
      004.037  302 032 004      1829          JNZ     XIN1
                                1830
                                1831  *       Check for RAM at Zero
                                1832
      004.042  041 000 000      1833          LXI     H,RAM8GO
      004.045  072 066 040      1834          LDA     CTLFLG2
      004.050  107              1835          MOV     B,A             Save original in B
      004.051  366 362          1836          ORI     OP.CTL2         Turn on RAM at Zero
      004.053  021 061 004      1837          LXI     D,XIN2          DE = return address
      004.056  303 004 040      1838          JMP     XIN8
      004.061  312 135 004      1839  XIN2    JZ      XIN5            No change with decrement
                                1840
                                1841  *       Copy ROM to RAM
                                1842
      004.064  001 000 010      1843          LXI     B,RAM8GOL       Length to Copy
      004.067  021 000 000      1844          LXI     D,RAM8GO
      004.072  032              1845  XIN3    LDAX    D               Move RAM8GO into place
      004.073  022              1846          STAX    D
      004.074  023              1847          INX     D
      004.075  013              1848          DCX     B
      004.076  170              1849          MOV     A,B
      004.077  261              1850          ORA     C
      004.100  302 072 004      1851          JNZ     XIN3            Not all moved yet
                                1852
```

```
                          1853  *       Copy H17 ROM to its final resting place (RIP)
                          1854
004.103  001 000 010     1855          LXI     B,H17ROML
004.106  041 000 030     1856          LXI     H,H17ROM
004.111  032             1857  XIN4    LDAX    D               Move H17 ROM into place
004.112  167             1858          MOV     M,A
004.113  023             1859          INX     D
004.114  043             1860          INX     H
004.115  013             1861          DCX     B
004.116  170             1862          MOV     A,B
004.117  261             1863          ORA     C
004.120  302 111 004     1864          JNZ     XIN4            Not all moved yet
                          1865
004.123  072 066 040     1866          LDA     CTLFLG2
004.126  366 040         1867          ORI     C82.ORG
004.130  062 066 040     1868          STA     CTLFLG2
004.133  323 362         1869          OUT     OP.CTL2         Turn on Ram at 0
                          1870
004.135  021 371 003     1871  XIN5    LXI     D,PRSROM        RESTORE NORMAL VALUES
004.140  041 012 040     1872          LXI     H,PRSRAM+PRSL-1
                          1873
004.143  303 073 000     1874          JMP     INIT            RETURN TO INLINE CODE
                          1875
004.146  323 362         1876  XINA    OUT     OP.CTL2         Select RAM
004.150  176             1877          MOV     A,M
004.151  065             1878          DCR     M
004.152  276             1879          CMP     M               check for a change
004.153  170             1880          MOV     A,B
004.154  323 362         1881          OUT     OP.CTL2         Select ROM
004.156  353             1882          XCHG
004.157  351             1883          PCHL
000.012                  1884  XINAL   EQU     *-XINA
```

```
                        1887  ***     Extended Command Table                          /Ram8Go 2/
                        1888  *
                        1889
004.160  306 004       1890  EXTCMD  ADI     4
004.162  207           1891          ADD     A               A = 2 * A
004.163  021 177 004    1892          LXI     D,EXTCMDA
004.166  157           1893          MOV     L,A
004.167  046 000       1894          MVI     H,0
004.171  031           1895          DAD     D               *(HL) = Processor Address
004.172  176           1896          MOV     A,M
004.173  043           1897          INX     H
004.174  146           1898          MOV     H,M
004.175  157           1899          MOV     L,A             HL = Processor Address
004.176  351           1900          PCHL                    Enter Processor
                        1901
004.177  322 000       1902  EXTCMDA DW      ERROR           '0'    Illegal
004.201  253 004       1903          DW      PRIBOO          '1' => Primary Boot
004.203  236 004       1904          DW      SECBOO          '2' => Secondary Boot
004.205  322 000       1905          DW      ERROR           '3'    Illegal


                        1907  **      AUTOB   - Auto Boot                             /Ram8Go 2/
                        1908  *
                        1909  *       AUTOB performs an auto boot of the primary device.
                        1910  *
                        1911  *       ENTRY:  NONE
                        1912  *
                        1913  *       EXIT:   To PRIBOO
                        1914  *
                        1915  *       USES:   ALL
                        1916  *
                        1917
004.207  041 010 040    1918  AUTOB   LXI     H,.MFLAG
004.212  176           1919          MOV     A,M
004.213  346 275       1920          ANI     377Q-U0.DDU-U0.NFR
004.215  167           1921          MOV     M,A             Enable Display Update, and Re-Fresh
004.216  043           1922          INX     H
000.000                1923          ERRNZ   CTLFLG-.MFLAG-1
004.217  066 360       1924          MVI     M,CB.SSI+C8.MTL+C8.CLI+CB.SPK
004.221  076 377       1925          MVI     A,-1
004.223  062 006 040    1926          STA     OSPROT          All Periods OFF
004.226  373           1927          EI
004.227  052 035 040    1928          LHLD    REGPTR
004.232  371           1929          SPHL
004.233  303 253 004    1930          JMP     PRIBOO          Boot PRImary device
```

```
                                1933  ***     PRIBOO  - Primary Boot                                   /Ram8Go 2/
                                1934  *
                                1935  *       PRIBOO is called to boot from the primary boot device
                                1936  *       as defined in the configuration port IP.CON.  The
                                1937  *       alternate entry, SECBOO, is called to boot from the
                                1938  *       secondary boot device.  If the CN.PRI switch is one,
                                1939  *       then address 170 is the primary device, otherwise,
                                1940  *       address 174 is the boot device.  From there, the
                                1941  *       configuration switch further determines device type
                                1942  *       with the appropriate masks.
                                1943  *
                                1944
004.236  257                   1945  SECBOO  XRA     A
004.237  062 061 041           1946  SECBOO. STA     AIO.UNI                 Zero Boot Unit
004.242  001 140 005           1947          LXI     B,MSGSEC
004.245  333 362               1948          IN      IP.CON
004.247  057                   1949          CMA                             Invert Primary Flag
004.250  303 264 004           1950          JMP     B001                    Boot Secondary Device
                                1951
004.253  257                   1952  PRIBOO  XRA     A
004.254  062 061 041           1953  PRIBOO. STA     AIO.UNI
004.257  001 135 005           1954          LXI     B,MSGPRI
004.262  333 362               1955          IN      IP.CON
                                1956
004.264  061 200 042           1957  B001    LXI     SP,STACK        Initialize the stack-pointer
004.267  346 020               1958          ANI     CN.PRI
004.271  333 362               1959          IN      IP.CON
004.273  365                   1960          PUSH    PSW
004.274  312 317 004           1961          JZ      B002                    174 is the device to boot
                                1962
                                1963  *       Boot Device is 170
                                1964
004.277  076 170               1965          MVI     A,170Q
004.301  062 120 041           1966          STA     BDA                     Save Boot Device Address
004.304  361                   1967          POP     PSW
004.305  346 014               1968          ANI     CN.170M                 Mask Device Type
004.307  312 143 005           1969          JZ      BERR                    No Device Installed at 170
000.000                        1970          ERRNZ   CND.NDI                 No-Device-Installed Flag
004.312  017                   1971          RRC
004.313  017                   1972          RRC                     Device Type converted to index
004.314  303 327 004           1973          JMP     B003
                                1974
                                1975  *       Boot Device is 174
                                1976
004.317  076 174               1977  B002    MVI     A,174Q
004.321  062 120 041           1978          STA     BDA             Save Boot Device Address
004.324  361                   1979          POP     PSW
004.325  346 003               1980          ANI     CN.174M         Mask out device type
                                1981
                                1982  *       Initialize Vectors and Display
                                1983
004.327  365                   1984  B003    PUSH    PSW             Save Device Index
004.330  305                   1985          PUSH    B
                                1986
004.331  076 320               1987          MVI     A,CB.SSI+CB.CLI+CB.SPK
004.333  062 011 040           1988          STA     CTLFLG          Turn off monitor mode
```

```
                                1989
004.336  076 007               1990            MVI     A,7
004.340  041 037 040           1991            LXI     H,UIVEC
004.343  066 303               1992  8004      MVI     M,MI.JMP        Stuff Interrupt Vectors
004.345  043                   1993            INX     H
004.346  066 002               1994            MVI     M,#EIRET
004.350  043                   1995            INX     H
004.351  066 007               1996            MVI     M,EIRET/256
004.353  043                   1997            INX     H
004.354  075                   1998            DCR     A
004.355  302 343 004           1999            JNZ     8004
                                2000
004.360  257                   2001            XRA     A
004.361  062 122 041           2002            STA     TIMEOUT         Zero Time-Out Counter
004.364  041 031 034           2003            LXI     H,ROMCLK        H17 Rom Clock Routine
004.367  042 124 041           2004            SHLD    USRCLK
004.372  041 221 005           2005            LXI     H,CLKINT
004.375  042 040 040           2006            SHLD    UIVEC+1         Initialize Clock Interrupt Vector
                                2007
005.000  001 132 037           2008            LXI     B,BOOTA
005.003  021 110 040           2009            LXI     D,D.CON
005.006  315 151 007           2010            CALL    SMOV            Initialize Rom/Ram Vectors
005.011  130                   2011            DB      BOOTAL
                                2012
005.012  041 240 040           2013            LXI     H,D.RAM
005.015  006 037               2014            MVI     B,D.RAML
005.017  315 323 007           2015            CALL    $ZERO           Zero Memory
                                2016
005.022  072 010 040           2017            LDA     .MFLAG
005.025  366 003               2018            ORI     UO.CLK+UO.DDU   Enable Clock Int. turn off Display Update
005.027  062 010 040           2019            STA     .MFLAG
                                2020
005.032  301                   2021            POP     B
005.033  021 013 040           2022            LXI     D,FPLEDS
005.036  315 151 007           2023            CALL    SMOV
005.041  003                   2024            DB      MSGLEN
005.042  056 006               2025            MVI     L,9-MSGLEN
005.044  076 377               2026            MVI     A,-1
005.046  022                   2027  8005      STAX    D               Blank some
005.047  023                   2028            INX     D
005.050  055                   2029            DCR     L
005.051  302 046 005           2030            JNZ     8005
                                2031
005.054  361                   2032            POP     PSW
005.055  021 101 005           2033            LXI     D,8006
005.060  325                   2034            PUSH    D               Force Boot to return to 8006
                                2035
005.061  062 121 041           2036            STA     BDF             Save Boot Device Flag
005.064  207                   2037            ADD     A               A = 2 * A
005.065  157                   2038            MOV     L,A
005.066  046 000               2039            MVI     H,0
005.070  021 125 005           2040            LXI     D,BOOA
005.073  031                   2041            DAD     D
005.074  176                   2042            MOV     A,M
005.075  043                   2043            INX     H
005.076  146                   2044            MOV     H,M
```

```
005.077  157           2045          MOV     L,A             HL = Device Processor Address
005.100  351           2046          PCHL
                       2047
005.101  072 010 040   2048  8006    LDA     .MFLAG
005.104  346 375       2049          ANI     377Q-UO.DDU     Turn on Display Update
005.106  062 010 040   2050          STA     .MFLAG          Restore original front panel mode
005.111  052 124 041   2051          LHLD    USRCLK
005.114  042 040 040   2052          SHLD    UIVEC+1         Clear Time-Out Vector to just user vector
005.117  332 322 000   2053          JC      ERROR           Boot Routines return here
                       2054
005.122  303 200 042   2055          JMP     USERFWA



                       2057  **      Device Processors
                       2058  *
                       2059
005.125                2060  800A    EQU     *
                       2061
000.000                2062          ERRNZ   *-800A/2-CNO.H17
005.125  032 006       2063          DW      8H17
                       2064
000.000                2065          ERRNZ   *-800A/2-CNO.H47
005.127  152 006       2066          DW      8H47
                       2067
005.131  143 005       2068          DW      BERR            Illegal Device
005.133  143 005       2069          DW      BERR
                       2070
                       2071
005.135  230 336 337   2072  MSGPRI  DB      10011000B,11011110B,11011111B           'Pri'
000.003                2073  MSGLEN  EQU     *-MSGPRI
                       2074
005.140  244 214 215   2075  MSGSEC  DB      10100100B,10001100B,10001101B           'SEC'
000.000                2076          ERRNZ   *-MSGSEC-MSGLEN
```

```
                        2080  **      BERR    - Boot Error                                    /Ram8Go 2/
                        2081  *
                        2082  *       BERR handles boot errors.
                        2083  *
                        2084
005.143  072 010 040   2085  BERR    LDA     .MFLAG
005.146  366 002       2086          ORI     UO.DDU          Disable Display Update
005.150  062 010 040   2087          STA     .MFLAG
                        2088
005.153  001 210 005   2089          LXI     B,BERRA
005.156  021 013 040   2090          LXI     D,FPLEDS
005.161  315 151 007   2091          CALL    SMOV            Set up Error Message in LED's
005.164  011           2092          DB      BERRAL
                        2093
005.165  001 000 000   2094          LXI     B,BERR8         BC = Time-Out Count
005.170  013           2095  BERR1   DCX     B
005.171  170           2096          MOV     A,B
005.172  261           2097          ORA     C
005.173  312 322 000   2098          JZ      ERROR           Done displaying message
                        2099
005.176  333 360       2100          IN      IP.PAD
005.200  376 157       2101          CPI     K.STAR          *
005.202  312 322 000   2102          JZ      ERROR           Cancel was hit
005.205  303 170 005   2103          JMP     BERR1
                        2104
005.210  206           2105  BERRA   DB      377Q-S0-S3-S4-S5-S6             b
005.211  306           2106          DB      377Q-S0-S3-S4-S5                o
005.212  306           2107          DB      377Q-S0-S3-S4-S5                o
005.213  362           2108          DB      377Q-S0-S2-S3                   t
005.214  377           2109          DB      377Q
005.215  377           2110          DB      377Q
005.216  214           2111          DB      377Q-S0-S1-S4-S5-S6             E
005.217  336           2112          DB      377Q-S0-S5                      r
005.220  336           2113          DB      377Q-S0-S5                      r
000.011               2114  BERRAL  EQU     *-BERRA
                        2115
000.000               2116  BERR8   EQU     0               Time-Out Count  (Full Wrap)
```

```
                        2119  ***     CLKINT  - Clock Interrupt Processor                    /Ram8Go 2/
                        2120  *
                        2121  *       CLKINT processes the clock interrupts by:
                        2122  *
                        2123  *               Checking for abort
                        2124  *               Checking for Time-Out
                        2125  *               Passing the Interrupts on to the user
                        2126  *
                        2127  *       This clock routine is only to be used at boot
                        2128  *       time.
                        2129  *
                        2130
     005.221  365      2131  CLKINT  PUSH    PSW
     005.222  333 360   2132          IN      IP.PAD
     005.224  376 157   2133          CPI     K.STAR
     005.226  312 322 000 2134        JZ      ERROR           Cancel is down, so abort the BOOT
                        2135
     005.231  072 033 040 2136        LDA     TICCNT
     005.234  247       2137          ANA     A
     005.235  302 331 005 2138        JNZ     CKI3            Not time to increment internal timer
                        2139
     005.240  072 122 041 2140        LDA     TIMEOUT
     005.243  074       2141          INR     A
     005.244  062 122 041 2142        STA     TIMEOUT
     005.247  376 036   2143          CPI     30
     005.251  332 331 005 2144        JC      CKI3            Not the end yet
                        2145
                        2146  *       Time-Out Error
                        2147
     005.254  072 121 041 2148        LDA     BOF             A = Boot device flag
     005.257  376 000   2149          CPI     CND.H17
     005.261  302 305 005 2150        JNZ     CKI1            Not an H17
                        2151
                        2152  *       Abort H17
                        2153
     005.264  257       2154          XRA     A
     005.265  062 243 040 2155        STA     D.DLYMO
     005.270  072 242 040 2156        LDA     D.DVCTL         A = Device Control
     005.273  346 200   2157          ANI     DF.WR           Remove all but Ram/Write
     005.275  062 242 040 2158        STA     D.DVCTL
     005.300  323 177   2159          OUT     DP.DC           Turn of Motor
     005.302  303 317 005 2160        JMP     CKI2
                        2161
                        2162  *       Abort H47
                        2163
     005.305  376 001   2164  CKI1    CPI     CND.H47
     005.307  302 317 005 2165        JNZ     CKI2            Should never happen
     005.312  315 033 007 2166        CALL    OBD.            Reset H47
     005.315  002 000   2167          DB      N.RES,D.STAI
     000.000           2168          ERRNZ   CKI2-*
                        2169
                        2170  *       Restore User Clock Vector
                        2171
     005.317  052 124 041 2172  CKI2    LHLD    USRCLK          Restore User Clock
     005.322  042 040 040 2173          SHLD    UIVEC+1
     005.325  373       2174          EI
```

```
005.326  303 143 005  2175          JMP     BERR            Boot Time-Out Error
                      2176
005.331  361          2177  CKI3    POP     PSW
005.332  345          2178          PUSH    H
005.333  052 124 041  2179          LHLD    USRCLK
005.336  343          2180          XTHL
005.337  311          2181          RET                     Enter User's Clock Routine
```

```
                          2184  ***     H89 COM/DAT                                    /Ram8Go 2/
                          2185  *
                          2186  *       H89COM and H89DAT are provided as common entry points
                          2187  *       between MTR-89 and RAM8GO.
                          2188  *
                          2189
     006.023              2190  .       SET     6023A
     000.063              2191          ERRMI   .-*
     005.340              2192          DS      .-*
     006.023  303 361 006 2193  H89DAT  JMP     DAT.
                          2194
     006.027              2195  .       SET     6027A
     000.001              2196          ERRMI   .-*
     006.026              2197          DS      .-*
     006.027  303 334 006 2198  H89COM  JMP     COM.
```

```
                        2201  **      8H17  - Boot H17                                        /Ram8Go 2/
                        2202  *
                        2203  *       8H17 boots the specified unit of an H-17 disk.  The
                        2204  *       unit to boot is specified in AIO.UNI.
                        2205  *
                        2206  *       ENTRY:  AIO.UNI = Unit of H-17 to boot.
                        2207  *               all H-17 Ram Vectors Initialized
                        2208  *
                        2209
    006.032  001 146 006  2210  8H17    LXI     B,8H17A
    006.035  021 020 040  2211          LXI     D,FPLEDS+3+2
    006.040  315 151 007  2212          CALL    SMOV            Move in H17 Message
    006.043  004          2213          DB      8H17AL
                        2214
    006.044  257          2215          XRA     A
    006.045  323 177      2216          OUT     DP.DC           Turn off disk
                        2217
    006.047  041 111 007  2218          LXI     H,R.SDP
    006.052  042 206 040  2219          SHLD    D.SDP+1         Re-Vector set device parameter for SY2:
                        2220
    006.055  036 012      2221  8H170   MVI     E,10            Watch for 10 holes
    006.057  315 126 006  2222  8H171   CALL    8H174           Watch inter-hole gap
    006.062  346 001      2223          ANI     DF.HD
    006.064  312 057 006  2224          JZ      8H171
    006.067  315 126 006  2225  8H172   CALL    8H174           Watch entire hole
    006.072  346 001      2226          ANI     DF.HD
    006.074  302 067 006  2227          JNZ     8H172
    006.077  035          2228          DCR     E
    006.100  302 057 006  2229          JNZ     8H171           Find another hole
                        2230
    006.103  315 141 040  2231  8H173   CALL    D.ABORT
    006.106  021 200 042  2232          LXI     D,USERFWA
    006.111  001 000 011  2233          LXI     B,9*256
    006.114  041 000 000  2234          LXI     H,0
    006.117  315 147 040  2235          CALL    D.READ
    006.122  332 055 006  2236          JC      8H170           Error Reading Sectors, keep trying
                        2237
    006.125  311          2238          RET
                        2239
    006.126  072 061 041  2240  8H174   LDA     AIO.UNI
    006.131  107          2241          MOV     B,A
    006.132  004          2242          INR     B
    006.133  257          2243          XRA     A
    006.134  315 304 007  2244          CALL    BITS            Select device
    000.000                2245          ERRNZ   DF.DSO-2
    000.000                2246          ERRNZ   DF.DS1-4
    000.000                2247          ERRNZ   DF.DS2-8
    006.137  366 020      2248          ORI     DF.MO           Turn on motor
    006.141  323 177      2249          OUT     DP.DC           Turn on Motor and drive select
    006.143  333 177      2250          IN      DP.DC           Look at the drive status
    006.145  311          2251          RET
                        2252
    006.146  222          2253  8H17A   DB      377Q-S0-S2-S3-S5-S6     'H'
    006.147  377          2254          DB      377Q
    006.150  363          2255          DB      377Q-S2-S3              '1'
    006.151  361          2256          DB      377Q-S1-S2-S3           '7'
```

```
000.004              2257  BH17AL  EQU     *-BH17A
```

```
                          2260  **      BH47    - Boot H47                                      /Ram8Go 2/
                          2261  *
                          2262  *       BH47 boots the specified unit of an H-47 disk.  The
                          2263  *       unit to boot is specified in AIO.UNI.
                          2264  *
                          2265  *       ENTRY:  AIO.UNI = Unit of H-47 to boot.
                          2266  *               all H-17 Ram Vectors initialized
                          2267  *
                          2268
006.152  001 324 006     2269  BH47    LXI     B,BH47A
006.155  021 020 040     2270          LXI     D,FPLEDS+3+2
006.160  315 151 007     2271          CALL    SMOV            Move in H-47 Message
006.163  004             2272          DB      BH47AL
                          2273
006.164  315 200 006     2274  BH471   CALL    BH472
006.167  320             2275          RNC                     NO errors at boot
                          2276
006.170  076 372         2277          MVI     A,500/2         Wait 1/2 Second
006.172  315 053 000     2278          CALL    DLY
006.175  303 164 006     2279          JMP     BH471           Errors, so try again


                          2281  **      BH472
                          2282  *
                          2283
                          2284  *       Wait for Done
                          2285
006.200  315 033 007     2286  BH472   CALL    OBD.
006.203  002 000         2287          DB      W.RES,D.STAI
006.205  315 167 007     2288          CALL    WDN
006.210  330             2289          RC                      Try again
                          2290
                          2291  *       Wait for Device Ready
                          2292
006.211  315 134 007     2293  BH473   CALL    RRDY
006.214  330             2294          RC
006.215  315 134 007     2295          CALL    RRDY            L = Ready Bits
006.220  330             2296          RC
                          2297
006.221  072 061 041     2298          LDA     AIO.UNI
006.224  107             2299          MOV     B,A
006.225  257             2300          XRA     A
006.226  315 304 007     2301          CALL    BITS
006.231  245             2302          ANA     L
006.232  302 211 006     2303          JNZ     BH473           Specified Unit is not ready
                          2304
                          2305  *       Boot the Device
                          2306
006.235  315 330 006     2307          CALL    COM             Output Load Sector Count Command
006.240  003             2308          DB      DD.LSC
006.241  330             2309          RC
006.242  315 355 006     2310          CALL    DAT             Output data
006.245  000             2311          DB      0
006.246  330             2312          RC
```

```
006.247  315 355 006  2313          CALL    DAT             Transfer count of 2
006.252  002          2314          DB      2
006.253  330          2315          RC
006.254  315 167 007  2316          CALL    WDN
006.257  330          2317          RC                      Try again
                      2318
006.260  315 330 006  2319          CALL    COM             Output Read Command
006.263  007          2320          DB      DD.READ
006.264  330          2321          RC
006.265  315 355 006  2322          CALL    DAT             Track Number
006.270  000          2323          DB      0
006.271  330          2324          RC
006.272  072 061 041  2325          LDA     AIO.UNI
006.275  017          2326          RRC
006.276  017          2327          RRC
006.277  017          2328          RRC
000.000               2329          ERRNZ   UNT.M-01100000B
006.300  366 001       2330          ORI     1               Start at sector 1
000.000               2331          ERRNZ   SEC.M-00011111B
006.302  315 361 006  2332          CALL    DAT.
006.305  330          2333          RC
                      2334
006.306  021 200 042  2335          LXI     D,USERFWA
006.311  315 066 007  2336  8H474   CALL    PIN
006.314  332 167 007  2337          JC      WDN             Pre-Mature DONE means end, error set if S.ERR
006.317  022          2338          STAX    D
006.320  023          2339          INX     D
006.321  303 311 006  2340          JMP     8H474           Get another byte
                      2341
006.324  222          2342  8H47A   DB      377Q-S0-S2-S3-S5-S6     'H'
006.325  377          2343          DB      377Q
006.326  262          2344          DB      377Q-S0-S2-S3-S6        '4'
006.327  361          2345          DB      377Q-S2-S3-S1           '7'
000.004               2346  8H47AL  EQU     $-8H47A
```

```
                        2350  **      COM     - Command                                       /Ram8Go 2/
                        2351  *
                        2352  *       COM output a command byte.
                        2353  *
                        2354  *       ENTRY:  *(RET+1)= Command byte
                        2355  *
                        2356  *       EXIT:   PSW     = 'C' Set   if    ERROR
                        2357  *                         'C' Clear if NO error
                        2358  *
                        2359  *       USES:   PSW,BC
                        2360  *
                        2361
006.330  343           2362  COM     XTHL
006.331  176           2363          MOV     A,M             A = command byte
006.332  043           2364          INX     H
006.333  343           2365          XTHL                    Restore return address
                        2366
006.334  365           2367  COM.    PUSH    PSW
006.335  315 167 007   2368          CALL    WDN
006.340  332 352 006   2369          JC      COM1            Error
006.343  361           2370          POP     PSW
006.344  315 037 007   2371          CALL    OBD             Output to data port
006.347  001           2372          DB      D.DATI
006.350  247           2373          ANA     A
006.351  311           2374          RET
                        2375
006.352  063           2376  COM1    INX     SP              Ignore saved PSW
006.353  063           2377          INX     SP
006.354  311           2378          RET


                        2380  **      DAT     - Data                                          /Ram8Go 2/
                        2381  *
                        2382  *       DAT outputs data to the boot H47 with a DTR handshake.
                        2383  *
                        2384  *       ENTRY:  *(RET+1)= data to output
                        2385  *
                        2386  *       EXIT:   To RET+1
                        2387  *
                        2388  *       USES:   PSW,BC
                        2389  *
                        2390
006.355  343           2391  DAT     XTHL
006.356  176           2392          MOV     A,M             A  = Data
006.357  043           2393          INX     H
006.360  343           2394          XTHL
                        2395
006.361  365           2396  DAT.    PUSH    PSW
006.362  315 222 007   2397          CALL    WTR
006.365  332 377 006   2398          JC      DAT1            Error
006.370  361           2399          POP     PSW
006.371  315 037 007   2400          CALL    OBD             Output to Data Port
006.374  001           2401          DB      D.DATI
006.375  247           2402          ANA     A
```

```
006.376  311          2403          RET
                      2404
006.377  063          2405  DAT1    INX     SP              Discard Saved Data
007.000  063          2406          INX     SP
007.001  311          2407          RET


                      2409  **      EIRET   - EI RETurn                                  /Ram8Go 2/
                      2410  *
                      2411  *       EIRET is a simple routine which Re-Enables Interrupts,
                      2412  *       and executes a RETurn instruction
                      2413  *
                      2414
007.002  373          2415  EIRET   EI
007.003  311          2416          RET


                      2418  **      IBD     - Input from Boot Device                     /Ram8Go 2/
                      2419  *
                      2420  *       IBD inputs data from the Boot Port as saved at boot time.
                      2421  *
                      2422  *       ENTRY:  BDA     = Boot Device Address
                      2423  *               *(RET+1)= Port Index
                      2424  *
                      2425  *       EXIT:   A       = Data input from port
                      2426  *               IOWRK destroyed
                      2427  *
                      2428  *       USES:   PSW
                      2429  *
                      2430
007.004  343          2431  IBD     XTHL
007.005  365          2432          PUSH    PSW
007.006  325          2433          PUSH    D
007.007  126          2434          MOV     D,M             D  = Port Index
007.010  043          2435          INX     H
007.011  072 120 041  2436          LDA     BDA
007.014  202          2437          ADD     D
007.015  353          2438          XCHG
007.016  147          2439          MOV     H,A             H  = Actual Output Address
007.017  056 333      2440          MVI     L,MI.IN
007.021  042 002 040  2441          SHLD    IOWRK           Stuff Instruction and Port
007.024  353          2442          XCHG
007.025  321          2443          POP     D
007.026  361          2444          POP     PSW
007.027  343          2445          XTHL
007.030  303 002 040  2446          JMP     IOWRK           Do the actual input
```

```
                        2448  **      OBD     - Output to Boot Device                 /Ram8Go 2/
                        2449  *
                        2450  *       OBD outputs the data to the indexed Boot Device.
                        2451  *
                        2452  *       ENTRY:  BDA     = Boot Device Address
                        2453  *               *(RET+1)= Port Address Index
                        2454  *
                        2455  *       EXIT:   IOWRK destroyed
                        2456  *               Data output to the port
                        2457  *
                        2458  *       USES:   PSW,IOWRK
                        2459  *
                        2460
007.033  343           2461  OBD.    XTHL
007.034  176           2462          MOV     A,M             A  = data byte to output
007.035  043           2463          INX     H
007.036  343           2464          XTHL
                        2465
007.037  343           2466  OBD     XTHL
007.040  365           2467          PUSH    PSW
007.041  325           2468          PUSH    D
007.042  126           2469          MOV     D,M             D  = Port Index
007.043  043           2470          INX     H
007.044  072 120 041   2471          LDA     BDA             A  = Boot Device Address
007.047  202           2472          ADD     D
007.050  353           2473          XCHG
007.051  147           2474          MOV     H,A             H  = Actual Device address
007.052  056 323       2475          MVI     L,MI.OUT
007.054  042 002 040   2476          SHLD    IOWRK           Stuff Instruction and address
007.057  353           2477          XCHG
007.060  321           2478          POP     D
007.061  361           2479          POP     PSW
007.062  343           2480          XTHL
007.063  303 002 040   2481          JMP     IOWRK           Do the actual I/O


                        2483  **      PIN     - Port In                               /Ram8Go 2/
                        2484  *
                        2485  *       PIN inputs a byte of data from the H-47 with
                        2486  *       a data-transfer-ready handshake.
                        2487  *
                        2488  *       ENTRY:  NONE
                        2489  *
                        2490  *       EXIT:   PSW     = 'C' Set   if    ERROR
                        2491  *                         'C' Clear if NO error
                        2492  *                          A  = data
                        2493  *
                        2494  *       USES:   PSW
                        2495  *
                        2496
007.066  315 004 007   2497  PIN     CALL    IBD
007.071  000           2498          DB      D.STAI
007.072  346 240       2499          ANI     S.DTR+S.DDN
007.074  312 066 007   2500          JZ      PIN             Not done, and not ready to transfer
```

```
                        2501
007.077  346 040       2502          ANI     S.DON
007.101  067           2503          STC
007.102  300           2504          RNZ                     Error because done before DTR
                        2505
007.103  315 004 007   2506          CALL    IBD
007.106  001           2507          DB      D.DATI
007.107  247           2508          ANA     A
007.110  311           2509          RET


                        2511  **      R.SDP   - Set-Up Device Parameters                    /Ram8Go 2/
                        2512  *
                        2513  *       R.SDP sets up arguments forthe specific unit.
                        2514  *
                        2515  *               D.DVCTL = Motor ON
                        2516  *               D.TRKPT = Address of device track number
                        2517  *
                        2518  *       Modified to access drive 3, or sy2:.
                        2519  *
                        2520  *       ENTRY:  AIO.UNI = Unit Number
                        2521  *
                        2522  *       EXIT:   HL      = D.TRKPT
                        2523  *
                        2524  *       USES:   PSW,HL
                        2525  *
                        2526
007.111  076 012       2527  R.SDP   MVI     A,ERPTCNT
007.113  062 264 040   2528          STA     D.OECNT         Set the max error count for the operation
007.116  072 061 041   2529          LDA     AIO.UNI
007.121  365           2530          PUSH    PSW
007.122  376 002       2531          CPI     1+1
                        2532
007.124  332 073 036   2533          JC      R.SDP.          Unit 0 or 1
000.000                 2534          ERRNZ   DF.DS0-2
000.000                 2535          ERRNZ   DF.DS1-4
                        2536
007.127  076 003       2537          MVI     A,3             Unit 2
000.000                 2538          ERRNZ   DF.DS2-8
007.131  303 073 036   2539          JMP     R.SDP.


                        2541  **      RRDY    - Read Ready                                 /Ram8Go 2/
                        2542  *
                        2543  *       RRDY checks to see if the drive specified in
                        2544  *       AIO.UNI is ready.
                        2545  *
                        2546  *       ENTRY:  AIO.UNI = unit number
                        2547  *
                        2548  *       EXIT:   L       = Ready Bits
                        2549  *
                        2550  *       USES:   PSW,L
```

```
                        2551  *
                        2552
007.134  315 330 006   2553  RRDY    CALL    COM
007.137  020           2554          DB      DD.RRDY
007.140  315 066 007   2555          CALL    PIN
007.143  330           2556          RC                      Bad Error(Pre-mature DONE)
007.144  157           2557          MOV     L,A             L  = Ready Bits
007.145  315 167 007   2558          CALL    WDN
007.150  311           2559          RET                     Unit return ERROR


                        2561  **      SMOV    - Short Move                                    /Ram8Go 2/
                        2562  *
                        2563  *       SMOV performs a short (<256) byte move)
                        2564  *
                        2565  *       ENTRY:  BC      = source
                        2566  *               DE      = destination
                        2567  *               *RET    = byte count
                        2568  *
                        2569  *       EXIT:   RET+1
                        2570  *
                        2571  *       USES:   PSW,BC,DE,L
                        2572  *
                        2573
007.151  343           2574  SMOV    XTHL
007.152  176           2575          MOV     A,M
007.153  043           2576          INX     H
007.154  343           2577          XTHL
007.155  157           2578          MOV     L,A             L  = Byte Count
                        2579
007.156  012           2580  SMOV.   LDAX    B
007.157  022           2581          STAX    D
007.160  003           2582          INX     B
007.161  023           2583          INX     D
007.162  055           2584          DCR     L
007.163  302 156 007   2585          JNZ     SMOV.           Move more bytes
                        2586
007.166  311           2587          RET


                        2589  **      WDN     - Wait for Done                                 /Ram8Go 2/
                        2590  *
                        2591  *       WDN waits for the done bit to be set.  A time-out
                        2592  *       is kept track of in order that the command may be
                        2593  *       re-tried.
                        2594  *
                        2595  *       ENTRY:  NONE
                        2596  *
                        2597  *       EXIT:   PSW     = 'C' set   if there is an error or time-out
                        2598  *                         'C' clear if no error
                        2599  *
                        2600  *       USES:   PSW,BC
```

```
                        2601  *
                        2602
007.167  001 000 175   2603  WDN     LXI     B,WDNA
                        2604
007.172  013           2605  WDN1    DCX     B
007.173  170           2606          MOV     A,B
007.174  261           2607          ORA     C
007.175  067           2608          STC
007.176  310           2609          RZ                      Time-Out
                        2610
007.177  315 004 007   2611          CALL    IBD
007.202  000           2612          DB      D.STAI
007.203  346 040       2613          ANI     S.DON
007.205  312 172 007   2614          JZ      WDN1            Wait longer
                        2615
007.210  315 004 007   2616          CALL    IBD             Error is only valid after DONE is set
007.213  000           2617          DB      D.STAI
007.214  346 001       2618          ANI     S.ERR
007.216  067           2619          STC
007.217  300           2620          RNZ                     Error from H47
                        2621
007.220  247           2622          ANA     A               Clear Error flag
007.221  311           2623          RET
                        2624
175.000                 2625  WDNA    EQU     32000           Time-Out Counter
```

```
                        2627  **      WTR     - Wait for Transfer Request                     /Ram8Go 2/
                        2628  *
                        2629  *       WTR waits for a transfer request.  It checks for DONE
                        2630  *       first, and if it is found, flags an error.  The code
                        2631  *       will also time-out waiting for *S.DTR*.
                        2632  *
                        2633  *       ENTRY:  NONE
                        2634  *
                        2635  *       EXIT:   PSW     = 'C' set   if    ERROR
                        2636  *                         'C' clear if NO error
                        2637  *
                        2638  *       USES:   PSW,BC
                        2639  *
                        2640
007.222  001 000 175   2641  WTR     LXI     B,WTRA          BC = time-out count
                        2642
007.225  315 004 007   2643  WTR1    CALL    IBD             Check for DONE
007.230  000           2644          DB      D.STAI
007.231  346 040       2645          ANI     S.DON
007.233  067           2646          STC
007.234  300           2647          RNZ                     DONE is set, must be problems
                        2648
007.235  013           2649          DCX     B
007.236  170           2650          MOV     A,B
007.237  261           2651          ORA     C
007.240  067           2652          STC
007.241  310           2653          RZ                      Time-out ERROR
```

```
                                2654
      007.242  315 004 007      2655            CALL    IBD
      007.245  000              2656            DB      D.STAI
      007.246  346 200          2657            ANI     S.DTR
      007.250  312 225 007      2658            JZ      WTR1            No DTR yet
                                2659
      007.253  311              2660            RET
                                2661
      175.000                   2662  WTRA      EQU     32000           Time-Out count
```

```
                        2666  **      PATCH1                                          /Ram8Go 2/
                        2667  *
                        2668  *       PATCH1 replaces code which initially only initialized
                        2669  *       the Tape UART with code which also checks for Auto-Boot.
                        2670  *       Since the return address is already on the stack, if
                        2671  *       Auto-Boot is set, INIT exits to AUTOB instead of ERROR.
                        2672  *
                        2673  *
                        2674  *       ENTRY:  NONE
                        2675  *
                        2676  *       EXIT:   HL = INIT exit address
                        2677  *               Tape UART Initialized
                        2678  *
                        2679  *       USES:   PSW,BC
                        2680  *
                        2681
007.254                 2682  PATCH1  EQU     *
                        2683
                        2684  *       Initialize LOAD/DUMP Uart
                        2685
007.254  076 116       2686          MVI     A,UMI.1B+UMI.L8+UMI.16X
007.256  323 371       2687          OUT     OP.TPC          SET 8 BIT, NO PARITY, 1 STOP, X16
                        2688
                        2689  *       Check for Auto-Boot
                        2690
007.260  041 322 000   2691          LXI     H,ERROR
007.263  333 362       2692          IN      OP.CTL2
007.265  346 200       2693          ANI     CN.ABO
007.267  310           2694          RZ                      No Auto-Boot
                        2695
007.270  041 207 004   2696          LXI     H,AUTOB
007.273  311           2697          RET
```

```
                        2699  **      PATCH2                                          /Ram8Go 2/
                        2700  *
                        2701  *       PATCH2  moves the MTR6 code out of its original place
                        2702  *       in the ROM to permit providing for H89/H8 common PIN
                        2703  *       routine.
                        2704  *
                        2705
007.274  312 322 000   2706  PATCH2  JZ      ERROR           NOT ALLOWED TO ALTER STACKPOINTER
007.277  043           2707          INX     H
007.300  361           2708          POP     PSW             RESTORE VALUE AND CARRY FLAG
007.301  303 062 003   2709          JMP     IOA             INPUT OCTAL ADDRESS
```

```
007.304                2712          XTEXT   BITS                                    /Ram8Go 2/


                       2714X **      BITS    -  BIT SET
                       2715X *
                       2716X *       BITS SETS THE SPECIFIED BIT IN THE ACCUMULATOR.
                       2717X *
                       2718X *       ENTRY:  A       = ORIGINAL  A
                       2719X *               B       = NUMBER OF BIT TO SET ( 7=HIGH,...,0=LOW )
                       2720X *
                       2721X *       EXIT:   A       = ORIGINAL  A  WITH  BIT(B) SET
                       2722X *
                       2723X *       USES:   PSW
                       2724X *
                       2725X
007.304  305           2726X BITS    PUSH    B
                       2727X
007.305  365           2728X         PUSH    PSW
007.306  076 200       2729X         MVI     A,10000000B
007.310  004           2730X         INR     B
007.311  007           2731X BITS1   RLC
007.312  005           2732X         DCR     B
007.313  302 311 007   2733X         JNZ     BITS1
                       2734X
007.316  117           2735X         MOV     C,A
007.317  361           2736X         POP     PSW
007.320  261           2737X         ORA     C
                       2738X
007.321  301           2739X         POP     BC
007.322  311           2740X         RET
007.323                2741          XTEXT   ZERO                                    /Ram8Go 2/


                       2743X **      $ZERO - ZERO MEMORY
                       2744X *
                       2745X *       $ZERO ZEROS A BLOCK OF MEMORY.
                       2746X *
                       2747X *       ENTRY   (HL) = ADDRESS
                       2748X *               (B) = COUNT
                       2749X *       EXIT    (A) = 0
                       2750X *       USES    A,B,F,H,L
                       2751X
                       2752X
007.323  257           2753X $ZERO   XRA     A
007.324  167           2754X ZR01    MOV     M,A
007.325  043           2755X         INX     H
007.326  005           2756X         DCR     B
007.327  302 324 007   2757X         JNZ     ZR01            IF MORE
007.332  311           2758X         RET
```

```
                        2761  ***     Secondary Entry Vectors                          /Ram8Go 2/
                        2762  *
                        2763
007.364                 2764  .       SET     2*1024-12       Start the vectors at the end
000.031                 2765          ERRMI   .-*
007.364                 2766          ORG     .
                        2767
007.364  303 254 004   2768  DEFPC   JMP     PRI800.         Enter Primary Boot
                        2769
007.367  303 237 004   2770          JMP     SEC800.         Enter Secondary Boot
                        2771
007.372  303 372 007   2772          JMP     *               H89 Compatibility
                        2773
007.375  303 375 007   2774          JMP     *               H89 Compatibility
                        2775
010.000                 2776  RAM8GOL EQU     *-RAM8GO        Length of RAM8GO
                        2777
000.000                 2778          ERRNZ   2*1024-*        Copy of H17 ROM starts here
```

```
                    2781
                    2782  **      THE FOLLOWING ARE CONTROL CELLS AND FLAGS USED BY THE KEYSET
                    2783  *       MONITOR.
                    2784
040.000             2785          ORG     40000A          8192
040.000             2786  START   DS      2               DUMP STARTING ADDRESS
040.002             2787  IOWRK   DS      2               IN OR OUT INSTRUCTION
040.004             2788  XINB    EQU     *               Transient Routine Area          /Ram8Go 2/
040.004             2789  PRSRAM  EQU     *               FOLLOWING CELLS INITIALIZED FROM ROM
040.004             2790          DS      1               RET
                    2791
040.005             2792  REGI    DS      1               INDEX OF REGISTER UNDER DISPLAY
040.006             2793  DSPROT  DS      1               PERIOD FLAG BYTE
040.007             2794  DSPMOD  DS      1               DISPLAY MODE
                    2795
040.010             2796  .MFLAG  DS      1               USER FLAG OPTIONS
                    2797  *                               SEE *U0.XXX* BITS DESCRIBED AT FRONT
                    2798
040.011             2799  CTLFLG  DS      1               FRONT PANEL CONTROL BITS
040.012             2800  REFIND  DS      1               REFRESH INDEX (0 TO 7)
000.007             2801  PRSL    EQU     *-PRSRAM        END OF AREA INITIALIZED FROM ROM
                    2802
040.013             2803  FPLEDS  EQU     *               FRONT PANEL LED PATTERNS
040.013             2804  ALEDS   DS      1               ADDR 0
040.014             2805          DS      1               ADDR 1
040.015             2806          DS      1               ADDR 2
                    2807
040.016             2808          DS      1               ADDR 3
040.017             2809          DS      1               ADDR 4
040.020             2810          DS      1               ADDR 5
                    2811
040.021             2812  DLEDS   DS      1               DATA 0
040.022             2813          DS      1               DATA 1
040.023             2814          DS      1               DATA 2
                    2815
040.024             2816  ABUSS   DS      2               ADDRESS BUSS
040.026             2817  RCKA    DS      1               RCK SAVE AREA
040.027             2818  CRCSUM  DS      2               CRC-16 CHECKSUM
040.031             2819  TPERRX  DS      2               TAPE ERROR EXIT ADDRESS
040.033             2820  TICCNT  DS      2               CLOCK TIC COUNTER
                    2821
040.035             2822  REGPTR  DS      2               REGISETR CONTENTS POINTER
                    2823
040.037             2824  UIVEC   DS      0               USER INTERRUPT VECTORS
040.037             2825          DS      3               JUMP TO CLOCK PROCESSOR
040.042             2826          DS      3               JUMP TO SINGLE STEP PROCESSOR
040.045             2827          DS      3               JUMP TO I/O 3
040.050             2828          DS      3               JUMP TO I/O 4
040.053             2829          DS      3               JUMP TO I/O 5
040.056             2830          DS      3               JUMP TO I/O 6
040.061             2831          DS      3               JUMP TO I/O 7
                    2832
040.064             2833  NMIRET  DS      2               Used by H-88/H-89               /Ram8Go 2/
040.066             2834  CTLFLG2 DS      1               Control byte for OPZ.CTL        /Ram8Go 2/
                    2835
                    2836
```

```
    041.120             2837          ORG     41120A
                        2838
    041.120             2839  BDA     DS      1       Boot Device Address          /Ram8Go 2/
    041.121             2840  BDF     DS      1       Boot Device Flag             /Ram8Go 2/
    041.122             2841  TIMEOUT DS      1       Counter for Time-Out         /Ram8Go 2/
    041.123             2842          DS      1                                    /Ram8Go 2/
    041.124             2843  USRCLK  DS      2       Secondary User Clock for Boot /Ram8Go 2/
                        2844
    041.126             2845          END
Assembly complete
 2845 statements
    0 errors detected
26126 bytes free
```

```
$ZERO   007323  2015   2753L
.       007364   578S   702S   852S   871S   932S   944S   952S   967S   988S  1003S  1057S  2190S
                2191   2192  2195S  2196   2197  2764S  2765   2766
.MFLAG  040010   702    739    744    776    817    823   1918   1923   2017   2019   2048   2050
                2085   2087  2796L
A.STX   000002   113E   1151   1341
A.SYN   000026   112E   1146   1339
ABORT   001147   887    975L
ABUSS   040024   848    912    1008   1112   1159   1183   1264   1587   2816L
AIO.UNI 041061   151E   1946   1953   2240   2298   2325   2529
ALARM   002136   827    1212L  1278
ALEDS   040013  1611   2804L
AS.ODD  000100   409E
AS.IDD  000040   410E
AS.SIA  000020   411E
AS.SLM  000003   412E
AUTOB   004207  1918L   2696
BDA     041120  1966   1978   2436   2471   2839L
BDF     041121  2036   2148   2840L
BERR    005143  1969   2068   2069   2085L  2175
BERR1   005170  2095L  2103
BERRA   005210  2089   2105L  2114
BERRAL  000011  2092   2114E
BERRB   000000  2094   2116E
BH17    006032  2063   2210L
BH170   006055  2221L  2236
BH171   006057  2222L  2224   2229
BH172   006067  2225L  2227
BH173   006103  2231L
BH174   006126  2222   2225   2240L
BH17A   006146  2210   2253L  2257
BH17AL  000004  2213   2257E
BH47    006152  2066   2269L
BH471   006164  2274L  2279
BH472   006200  2274   2286L
BH473   006211  2293L  2303
BH474   006311  2336L  2340
BH47A   006324  2269   2342L  2346
BH47AL  000004  2272   2346E
BITS    007304  2244   2301   2726L
BITS1   007311  2731L  2733
BLKSIZ  002000   190E
BOO1    004264  1950   1957L
BOO2    004317  1961   1977L
BOO3    004327  1973   1984L
BOO4    004343  1992L  1999
BOO5    005046  2027L  2030
BOO6    005101  2033   2048L
BOOA    005125  2040   2060E  2062   2065
BOOTA   037132   152E   2008
BOOTAL  000130   153E   2011
C.DSYN  000375   365E
CB.CLI  000100   119E    215    617    822   1924   1987
CB.MTL  000040   118E    689    773    822   1059   1924
CB.SPK  000200   120E    617    822   1214   1924   1987
CB.SSI  000020   117E    617    689    822   1046   1055  1924   1987
CB2.CLI 000002   125E
```

```
C82.ORG 000040    126E    1867
C82.SID 000100    127E
C82.SSI 000001    124E
CKI1    005305   2150    2164L
CKI2    005317   2160    2165   2168   2172L
CKI3    005331   2138    2144   2177L
CLK2    000234    751     753L
CLK3    000237    747     756E
CLK4    000313    780     796E
CLKINT  005221   2005    2131L
CLOCK   000201    563     564    727L
CN.170M 000014   139E    1968
CN.174M 000003   138E    1980
CN.ABD  000200   143E    2693
CN.BAU  000100   142E
CN.MEM  000040   141E
CN.PRI  000020   140E    1958
CND.H17 000000   145E    2062   2149
CND.H47 000001   147E    2065   2164
CND.NDI 000000   146E    1970
COM     006330   2307    2319   2362L  2553
COM.    006334   2198    2367L
COM1    006352   2369    2376L
CRC     002347   1402L   1463
CRC1    002356   1406L   1426
CRC2    003004   1417    1424L
CRCSUM  040027   1154    1192   1247   1347   1405   1427   2818L
CTC     002172   1119    1246L
CTLFLG  040011    578     686    744    749    771    776    823   1045   1048   1057   1219   1229
                 1923    1988   2799L
CTLFLG2 040066   1817    1834   1866   1868   2834L
CUI1    000165    703L    799
D.ABORT 040141    304L   2231
D.CDE   040160    309L
D.CON   040110    233L   2009
D.DATI  000001    380E   2372   2401   2507
D.DLY   040235    324L
D.DLYMS 040244    263L
D.DLYMO 040243    262L   2155
D.DRVTB 040251    268L
D.DTS   040163    310L
D.DVCTL 040242    260L   2156   2158
D.E.CHK 040267    279L
D.E.HCK 040270    280L
D.E.HSY 040266    278L
D.E.MDS 040265    277L
D.E.TRK 040272    282L
D.E.VOL 040271    281L
D.ERR   040265    276L
D.ERRL  040273    283L
D.ERRT  040232    323L
D.HECNT 040261    270L
D.LPS   040177    314L
D.MAI   040171    312L
D.MAO   040174    313L
D.MOUNT 040133    302L
D.OECNT 040264    272L   2528
```

```
D.OPR    040273   287L
D.OPW    040275   288L
D.RAM    040240   236L   255   290   2013
D.RAML   000037   290E   2014
D.RDB    040202   315L
D.READ   040147   306L   2235
D.READR  040152   307L
D.SDP    040205   316L   2219
D.SDT    040166   311L
D.SECNT  040262   271L
D.STAI   000000   379E   380   2167  2287  2498  2612  2617  2644  2656
D.STS    040210   317L
D.STZ    040213   318L
D.SYDD   040130   301L
D.TRKPT  040245   265L
D.TS     040241   258L
D.TT     040240   257L
D.UDLY   040216   319L
D.VEC    040130   235L   299
D.VOLPT  040247   266L
D.WNB    040227   322L
D.WRITE  040155   308L
D.WSC    040221   320L
D.WSP    040224   321L
D.XIT    040144   305L
D.XOK    040136   303L
DAT      006355   2310   2313  2322  2391L
DAT.     006361   2193   2332  2396L
DAT1     006377   2398   2405L
DD.BOOT  000000   418L
DD.CPY   000013   429L
DD.DS    000202   453L
DD.FRM0  000014   430L
DD.FRM1  000015   431L
DD.FRM2  000016   432L
DD.FRM3  000017   433L
DD.LSC   000003   421L   2308
DD.RAD   000004   422L
DD.RAS   000002   420L
DD.RDBL  000205   456L
DD.RDL   000203   454L
DD.REA   000005   423L
DD.REAB  000007   425L   2320
DD.RRDY  000020   434L   2554
DD.RST   000001   419L
DD.SOC   000200   451L
DD.SPF0  000020   440L
DD.SPF1  000021   441L
DD.SPF2  000022   442L
DD.SPF3  000023   443L
DD.SPF4  000024   444L
DD.SPF5  000025   445L
DD.ST    000201   452L
DD.WDLB  000210   459L
DD.WRBD  000012   428L
DD.WRD   000011   427L
DD.WRI   000006   424L
```

```
DD.WRIB  000010   426L
DD.WTBL  000206   457L
DD.WTDL  000207   458L
DD.WTL   000204   455L
DEFPC    007364   658    2768L
DF.DI    000040   341E
DF.DSO   000002   337E   2245   2534
DF.DS1   000004   338E   2246   2535
DF.DS2   000010   339E   2247   2538
DF.HD    000001   331E   2223   2226
DF.MO    000020   340E   2248
DF.SD    000010   334E
DF.ST    000100   342E
DF.TO    000002   332E
DF.WG    000001   336E
DF.WP    000004   333E
DF.WR    000200   343E   2157
DLEDS    040021  1624    2812L
DLY      000053   609L   1686   2278
DM.MR    000000   131E
DM.MW    000001   132E
DM.RR    000002   133E   1763
DM.RW    000003   134E
DOD      003122  1535L   1613   1615   1618
DOD1     003127  1538L   1556
DODA     003356  1536    1543   1740L
DP.DC    000177   329E   2159   2216   2249   2250
DSPA     003342  1594    1730L
DSPMOD   040007   839     852    871    932    934    952   1003   1005   1584   2794L
DSPROT   040006   843     934   1005   1578   1584   1926   2793L
DUMP     002002  1144L
EIRET    007002  1994    1996   2415L
ERPTCNT  000012   154E   2527
ERROR    000322   792     816E   941   1295   1511   1902   1905   2053   2098   2102   2134   2691
                 2706
EXTCMD   004160   861    1890L
EXTCMDA  004177  1892    1902L
FPLEDS   040013  2022    2090   2211   2270   2803E
GO       001222   879    1037L
GO.      000063   617L   1037
H17ROM   030000  1813E   1856
H17ROML  010000  1814E   1855
H89COM   006027  2198L
H89DAT   006023  2193L
H89PIN   001067   916L
HORN     002140  1213L   1524   1706
HRN0     002143   611    1216L
HRN2     002160  1227L   1228
IBD      007004  2431L   2497   2506   2611   2616   2643   2655
IN       001177   880    1018L
INIT     000073   553     554    636L   640   1874
INIT0    000000   551L
INIT1    000107   649L
INIT2    000117   656L   1790   1793
INT1     000010   558E
INT2     000020   573E
INT3     000030   590L
```

INT4	000040	595L						
INT5	000050	600L						
INT6	000060	614L						
INT7	000070	621L						
INTXIT	000172	710L	774	1050				
IOA	003062	1009	1492L	2709				
IOB	003066	903	1492	1507L				
IOB1	003070	1508L	1522					
IOWRK	040002	1024	1025	2441	2446	2476	2481	2787L
IP.CON	000362	107E	1948	1955	1959			
IP.PAD	000360	99E	797	1280	1309	1683	2100	2132
IP.TPC	000371	103E	1311					
IP.TPD	000370	105E	1381					
K.DIVD	000117	174E						
K.DOT	000017	176E						
K.MINU	000217	172E						
K.NUMB	000057	175E						
K.PLUS	000257	171E						
K.STAR	000157	173E	2101	2133				
LAST	001150	886	983L					
LOA0	001272	1084L	1127					
LOA1	001342	1110L	1117					
LOAD	001267	1082E						
LRA	003047	919	1477L	1592				
LRA.	003052	785	1101	1171	1478L			
LST2	001154	989L						
M.INI	242355	373E						
M.OUTI	243355	374E						
MEMM	001165	889	1002L					
MI.ANI	000346	207E	1270					
MI.HLT	000166	201E	791					
MI.IN	000333	203E	1018	2440				
MI.JMP	000303	204E	1992					
MI.LDA	000072	206E						
MI.LXID	000021	208E	1019					
MI.OUT	000323	205E	1020	2475				
MI.RET	000311	202E	1766					
MSGLEN	000003	2024	2025	2073E	2076			
MSGPRI	005135	1954	2072L	2073				
MSGSEC	005140	1947	2075L	2076				
MTR	000344	834E	1060					
MTR1	000345	837	837L					
MTR4	001005	850	860L					
MTR5	001051	855	899L					
MTR6	001072	901	918L					
MTRA	001035	864	878E					
NEXT	001132	885	962L					
NMIRET	040064	2833L						
OBD	007037	2371	2400	2466L				
OBD.	007033	2166	2286	2461L				
OP.CTL	000360	100E	1047	1056				
OP.CTL2	000362	108E	1836	1869	1876	1881	2692	
OP.DIG	000360	101E	758					
OP.SEG	000361	102E	760					
OP.TPC	000371	104E	1145	1203	1294	1377	1460	2687
OP.TPD	000370	106E	1462					
OUT	001202	881	1020L					

PATCH1	007254	660	2682E									
PATCH2	007274	921	2706L									
PIN	007066	916	2336	2497L	2500	2555						
PRI800	004253	1903	1930	1952L								
PRI800.	004254	1953L	2768									
PRSL	000007	1872	2801E									
PRSRAM	040004	1872	2789E	2801								
PRSROM	003371	1759E	1871									
R$W	001126	888	953L									
R.SDP	007111	2218	2527L									
R.SDP.	036073	155E	2533	2539								
RAM8GO	000000	548E	551	1833	1844	2776						
RAM8GOL	010000	1843	2776E									
RCK	003260	847	938	1508	1677E							
RCK1	003267	1683L	1691	1703								
RCK2	003310	1689	1695L									
RCK3	003326	1698	1704L									
RCKA	040026	1681	2817L									
REFIND	040012	749	2800L									
REGI	040005	870	944	967	988	1477	2792L					
REGM	001104	890	931L									
REGPTR	040035	693	825	1480	1928	2822L						
RMEM	001261	883	1066L									
RNB	002331	1110	1337	1362	1376L							
RNB1	002335	1378L	1380									
RNP	002325	1096	1106	1246	1348	1362L						
ROMBOOT	030000	228E										
ROMCLK	034031	156E	2003									
RRDY	007134	2293	2295	2553L								
RT.BD	000005	185E										
RT.BP	000002	182E										
RT.CT	000003	183E										
RT.MI	000001	181E	1083	1155								
RT.NB	000004	184E										
RT.PD	000006	186E										
S.DON	000040	383E	2499	2502	2613	2645						
S.DTR	000200	385E	2499	2657								
S.ERR	000001	382E	2618									
S.GRT0	024000	224E										
S.GRT1	025000	225E										
S.GRT2	026000	226E										
S.IEN	000100	384E										
S.INT	040343	238L										
S.SOVR	041146	240L	242									
S.SW0	000002	387E										
S.SW1	000004	388E										
S.SW2	000010	389E										
S.SW3	000020	390E										
S.VAL	040277	237L										
S0	000001	160E	2105	2106	2107	2108	2111	2112	2113	2253	2342	2344
S1	000002	161E	2111	2256	2345							
S2	000004	162E	2108	2253	2255	2256	2342	2344	2345			
S3	000010	163E	2105	2106	2107	2108	2253	2255	2256	2342	2344	2345
S4	000020	164E	2105	2106	2107	2111						
S5	000040	165E	2105	2106	2107	2111	2112	2113	2253	2342		
S6	000100	166E	2105	2111	2253	2342	2344					
S7	000200	167E										

SAE	001063	912L	963	984	1028			
SAVALL	000132	561	576	677L				
SB.BTO	000001	404E						
SB.CRC	000010	401E						
SB.DLD	000040	399E						
SB.ILC	000002	403E						
SB.LTD	000004	402E						
SB.NRF	000020	400E						
SB.UNR	000200	397E						
SB.WPD	000100	398E						
SC.UART	000372	501E						
SEC.M	000037	480E	2331					
SEC800	004236	1904	1945L					
SEC800.	004237	1946L	2770					
SID.0	000000	473E	476					
SID.1	000200	474E	476					
SID.M	000200	476E						
SINCR	004000	642E	644					
SMOV	007151	2010	2023	2091	2212	2271	2574L	
SMOV.	007156	2580L	2585					
SRS	002265	1084	1333E					
SRS1	002265	1334L	1342	1346				
SRS2	002271	1337L	1340					
SSIZ.M	004000	484E						
SST1	001235	618	1048L					
SSTEP	001225	882	1043E					
STACK	042200	244E	1957					
STACKL	001032	242E						
START	040000	645	1108	1157	2786L			
STPRTN	001244	579	1054E					
SYDD	040130	234E						
TD.IN	000370	194E						
TD.OUT	000370	195E						
TER1	002220	1278L	1285					
TER3	002215	1271L	1282					
TFT	002133	1126	1202L	1266				
TICCNT	040033	727	729	764	1223	1283	2136	2820L
TIMEOUT	041122	2002	2140	2142	2841L			
TPABT	002244	1066	1141	1293L				
TPERR	002205	1095	1264L					
TPERRX	040031	1067	1142	1313	2819L			
TPXIT	002252	1279	1309L	1378	1456			
TS.IN	000371	196E						
TS.OUT	000371	197E						
UCI.ER	000020	523E	1376	1459				
UCI.IE	000002	525E						
UCI.IR	000100	521E						
UCI.RE	000004	524E	1376					
UCI.RO	000040	522E	1376					
UCI.TE	000001	526E	1144	1459				
UDR	000000	498E						
UF.FCT	000100	358E						
UF.RDA	000001	355E						
UF.ROR	000002	356E						
UF.RPE	000004	357E						
UF.TBM	000200	359E						
UFD	003161	767	1573E					

```
UFD1    003227   1588    1609L
UIVEC   040037    590     595   600   614   621   706  1061  1991  2006  2052  2173  2824L
UMI.16X 000002    516E   2686
UMI.1B  000100    506E   2686
UMI.1X  000001    515E
UMI.2B  000300    508E
UMI.64X 000003    517E
UMI.HB  000200    507E
UMI.L5  000000    511E
UMI.L6  000004    512E
UMI.L7  000010    513E
UMI.L8  000014    514E   2686
UMI.PA  000020    510E
UMI.PE  000040    509E
UNT.0   000000    464E    469
UNT.1   000040    465E    469
UNT.2   000100    466E    469
UNT.3   000140    467E    469
UNT.M   000140    469E   2329
UO.CLK  000001    217E    704  2018
UO.DDU  000002    216E    819  1574  1920  2018  2049  2086
UO.HLT  000200    214E    778
UO.NFR  000100    215E    742   819  1920
UP.DP   000174    349E
UP.FC   000175    350E
UP.SC   000176    352E
UP.SR   000176    353E
UP.ST   000175    351E
USERFWA 042200    245E   2055  2232  2335
USR     000001    499E
USR.FE  000040    530E
USR.OE  000020    531E
USR.PE  000010    532E
USR.RXR 000002    534E   1379
USR.TXE 000004    533E
USR.TXR 000001    535E   1457
USRCLK  041124   2004   2051  2172  2179  2843L
W.RES   000002     392E  2167  2287
WDN     007167   2288   2316  2337  2368  2558  2603L
WDN1    007172   2605L  2614
WDNA    175000   2603   2625E
WME1    002012   1148L  1150
WME2    002104   1181L  1188
WMEM    001374    884   1140E
WNB     003024   1148   1152  1182  1442  1455L
WNB1    003025   1456L  1458
WNP     003017   1156   1167  1176  1179  1193  1194  1441L
WTR     007222   2397   2641L
WTR1    007225   2643L  2658
WTRA    175000   2641   2662E
XIN1    004032   1824L  1829
XIN2    004061   1837   1839L
XIN3    004072   1845L  1851
XIN4    004111   1857L  1864
XIN5    004135   1839   1871L
XINA    004146   1822   1876L  1884
XINAL   000012   1821   1884E
```

XINB	040004	1823	1838	2788E
XINIT	004016	552	1816L	
XINITI	004000	649	1786L	1797
ZRO1	007324	2754L	2757	

30454 bytes free

INDEX