SOFTWARE REFERENCE MANUAL

Digital Computer System

Model H8

HEATH COMPANY
BENTON HARBOR, MICHIGAN 49022

595-2048
Printed in the United States of America

Contents

NOTE: An individual Table of Contents is included at the beginning of each of the following sections.

Introduction ..... Page 0-3

Panel Monitor (PAM-8) ..... Page 1-2

Console Debugger (BUG-8) ..... Page 2-2

Heath Text Editor (TED-8) ..... Page 3-2

Heath Assembly Language (HASL-8) ..... Page 4-2

Benton Harbor BASIC ..... Page 5-2

SPECIAL DISCLAIMER

Heath cannot provide consultation on user-developed programs or modified versions of Heath Software products.

INTRODUCTION

TABLE OF CONTENTS

PANEL MONITOR (PAM-8) 0-7

CONSOLE DEBUGGER (BUG-8) 0-8

HEATH TEXT EDITOR (TED-8) 0-8

HEATH ASSEMBLY LANGUAGE (HASL-8) 0-10

BENTON HARBOR BASIC 0-11
TAPE FILES 0-12
- System Record Structure 0-13
- Label Record Format 0-14
- System Data Formats 0-14
- Reading the Displays 0-15

USING THE MAGNETIC TAPE SYSTEM 0-18
- Recorder Operating Hints 0-18

USING THE PAPER TAPE SYSTEM 0-18
- Paper Tape Operating Hints 0-19

PRODUCT INSTALLATION 0-19
- Creating a Configured Tape 0-19
- Loading From a Configured Tape 0-21
- Copying an Existing Memory Tape 0-22
- Installing a Pitch 0-22
- Using an ASR Console 0-23
- Using a 110 Baud Console Terminal 0-23
- Console Interface 0-23
- Reporting Software Problems 0-25

APPENDIX A (ASR Patches) 0-26

APPENDIX B (Console Driver Listing) 0-36

APPENDIX C (I/O and Memory Maps) 0-50

APPENDIX D (ASCII Characters) 0-52

APPENDIX E (Decimal to Octal Tables) 0-56

APPENDIX F (Memory Table, Offset Octal & Decimal Boundaries 0-58

INDEX 0-59

This Software Reference Manual includes all the information you will need to be thoroughly familiar with the software products supplied with your H8 Computer. These software products are: the front Panel Monitor, PAM-8; the Console Debugger routine, BUG-8; the Heath Test Editor, TED-8; the Heath Assembly Language, HASL-8; and Benton Harbor BASIC, Heath Company's version of Dartmouth BASIC. Extended Benton Harbor BASIC, which is available as an optional accessory and includes such additional features as string manipulation, is also included in this Manual.

This book is intended as a reference manual, and, as such, it is as complete as possible. Examples are included to help you understand exactly how the Heath software products carry out their instructions; but they are not designed to teach you programming. If you have never used a text editor and assembler, for example, we recommend that you obtain some instruction from other sources, such as the "Heathkit Continuing Education" courses, prior to reading this material. If you have used editors and assemblers, this Manual will tell you about the special features in the Heath Text Editor (TED-8) and the Heath Assembly Language (HASL-8).

This introduction describes each product briefly and covers those aspects of the packages that are common to all. A separate section then follows for each software product. Each section provides detailed reference information and is followed by one or more Appendices for that product. Be sure to read all of this introductory section so you have a good overview of all of the products.

Heath software products feature a high degree of commonality in many of the modules which make up the individual products. For example, all software products which use the console terminal employ a software module called the Console Terminal Driver. This common usage of the console terminal driver permits you to move easily from one software product to the other, as the operating features are similar. Likewise, all tape handling is carried out through a common tape handling package, and once these features are understood, they are applicable to all products.

Heath software is supplied in three forms: cassette magnetic tape, paper tape, and read/only memory (ROM). The Panel Monitor (PAM-8) is supplied in a ROM (programs supplied in ROM cannot be modified by the user). The Console Debugger (BUG-8), the Heath Text Editor (TED-8), the Heath Assembly Language (HASL-8), and BASIC are supplied with the H8 in cassette form. They are optionally available in paper tape form. The cassettes and the paper tapes are compatible with the required error checking and synchronizing characters used by the front panel monitor system.

A printed copy of the panel monitor source listing is provided to aid you in using PAM-8. The Console Driver Listing and the partial listing (including entry points) of the BASIC floating point package and other BASIC utility packages are also included. All other programs are supplied in binary object forms and listings are **not** available.

PANEL MONITOR (PAM-8)

The ROM Panel Monitor, which is permanently located in the lower 1024 bytes of memory, permits you to load, execute, and debug programs written in 8080 machine language. The Heath Panel Monitor also makes use of the first 64 locations of random access memory. The H8 front panel is used as an I/O device, and it is assigned port numbers 360 and 361. With the Heath Panel Monitor, you can:

1. Examine the contents of a memory location.
2. Change the contents of a memory location (enter a new program, for example, or modify an old program).
3. Examine the contents of any of the 8080 registers.
4. Change the contents of any of the 8080 registers.
5. Start or stop the execution of a user-written program.
6. Execute a user program, a single instruction at a time.
7. Dump a program onto either magnetic or paper tape, with error detection codes and synchronization data.
8. Load a program from paper or magnetic tape into the desired memory locations.
9. Breakpoint a user program.
10. Reinitialize to a power up status.

The Heath Panel Monitor also offers the following features:

1. The user may automatically increment or decrement memory addresses which are being examined or modified.
2. The user may automatically increment or decrement through the registers which are being examined or modified.
3. The user is provided with a visual indication of the current mode in which the panel monitor is operating.
4. The user is provided with audio feedback upon valid and invalid command and data entry.
5. The H8 front panel utilizes an octal display rather than the more difficult to read binary display.
6. The front panel key switches and display are available for your programs.
7. The front display is operated on a continuously updated basis and, therefore, is active even during the execution of a user program. This feature permits the user to monitor either registers or memory location while his program is operating.

PAM-8 provides the fundamental tape routines by which the user loads all other programs, including the Heath supplied software and user-written software into the computer.

CONSOLE DEBUGGER (BUG-8)

BUG-8 allows you to perform very sophisticated operations from a console terminal with a full active keyboard and display. BUG-8 resides in H8 memory, using approximately 3,000 bytes of storage. You can use BUG-8 to write, load, execute, and debug machine language programs in the H8 computer in octal, decimal, or ASCII format. This package also has many of the features included in PAM-8.

With the Heath Console Debugger, you can:

1. Examine the contents of memory locations.
2. Alter the contents of memory locations.
3. Examine the contents of the CPU registers.
4. Alter the contents of the CPU registers.
5. Start program execution.
6. Execute a program in a single step form.
7. Set break points with multiple hit capability.
8. Clear break points.
9. Load programs from magnetic tape or paper tape.
10. Dump programs onto magnetic tape or paper tape.

BUG-8 is an advanced monitor, permitting you to prepare extensive software in machine code format that can be readily debugged and then recorded on a mass storage unit for future use.

HEATH TEXT EDITOR (TED-8)

The Heath TED-8 Text Editor is a general purpose, line-oriented text editor that is used primiarly to prepare source code that can be assembled by the Heath Assembly Language (HASL-8). But while this is its primary purpose, it is also useful for such things as letter writing, preparation of club newspapers, and manuscript editing.

This software product requires an H8 system with 8192 bytes of memory, an ASCII keyboard for text entry, and an ASCII display for text display. If large files are to be used or files are to be saved, a separate input/output tape unit is recommended.

With the Heath Text Editor, you can:

1. Read text from a pre-existing text file.
2. Create text for a new file.
3. Output text to a named tape file.
4. Insert new text after a given line.
5. Search the text for a given character string.
6. Delete a given line or lines.
7. Print a particular line or lines.
8. Replace a given line.
9. Edit a given string; that is, replace a particular string with another string.

All the above functions are supported by a number of special features, some of which are only available on the Heath Text Editor. Some of these features are:

1. A wide scope of range expressions, including:

   A. First line.
   B. Last line.
   C. Single line.
   D. Line to line.

2. Count and string versions of range expressions, which permit you to edit lines, plus or minus a certain number of lines from a given line, or to edit all lines containing a certain string.
3. You have the option of selecting one of three optional modes. Optional mode A prints the line after operating on it, optional mode B prints the line before operating on it, and optional mode BA prints the line before and after operating on it.
4. The use of a Qualifier String (Qualifier Strings permit operating only on the lines containing designated strings).
5. Tab. This command lets you set tab stops for entering text. The editor is constructed so that tabs do not occupy extensive user storage.
6. A Use statement, which provides a line count and memory usage information.

7. File Labeling Procedures to create new file names in either the input or output mode.

Under the H8 text editor, source code is prepared for the Heath Assembly Language (HASL-8). Once the source code has been prepared, it is written to a cassette tape or paper tape output file. Once this has been done, the user proceeds to the assembler.

HEATH ASSEMBLY LANGUAGE (HASL-8)

Heath Assembly Language runs on a Heath H8 Computer using about 8192 bytes of memory. This program assembles source code and produces object code. HASL-8 utilizes all the standard 8080 mnemonics, extended mnemonics, and numerous psuedo instructions.

Some of the special features of HASL-8 are that it:

1. Recognizes five operators: plus, minus, *, /, unary−.
2. Recognizes four token operand expressions:

   A. Integers.
   B. Symbols.
   C. Character strings.
   D. The origin symbol.

HASL-8 is a two pass assembler. Before the user starts assembly, it asks if a binary output is to be generated. On the second pass, it produces the binary if directed to do so, as well as the appropriate listing. The binary object code may be placed on a specified output device or may be placed directly in memory.

HASL-8 features the same terminal controls as do other Heath programs, including a suspend output mode and a discard output mode.

BENTON HARBOR BASIC

BENTON HARBOR BASIC is a modified version of Dartmouth BASIC, an easy-to-learn-and-use conversational language.

The BENTON HARBOR BASIC system is interpretive. That is, it executes each statement as it comes to it. BENTON HARBOR BASIC utilizes an H8 computer with 8K of memory, and appropriate terminal and paper tape or magnetic tape handling capability.

Extended BENTON HARBOR BASIC requires 12K of memory and offers strings. Some of the features of BENTON HARBOR BASIC are:

1. Three different data types:

   A. Numeric data, which has over six digits of accuracy and lies in the range of 10^{-39} to 10^{+38}. Numeric data may be either fixed or floating point.
   B. Strings, which can be from 0 to 255 characters.
   C. Boolean values, which permit logical operations.

2. Multidimensioned variables.

3. BASIC supports fifteen operators, which are:

   A. −(unary) NOT
   B. ↑ Exponentiation
   C. * /
   D. + −
   E. <, <=, =, <>, >=, and >
   F. OR
   G. AND

4. Free Format Programs.
5. Multiple statements per line.
6. Enhanced expression and conditional statement facilities.

BASIC features both command and program modes, where statements may be executed immediately after the line is written or numbered lines may be used so the program will not be executed until a RUN statement is executed.

BASIC also features command completion. In the command mode, BASIC checks inputted characters and, as soon as there are sufficient characters to establish a unique command, the command is completed. This feature saves considerable typing time and reduces errors.

NOTE: In order to fully use the Heath Software package, you must not only review the special features of Heath programs, but you must also know how to use monitors, debuggers, Text Editors, Assemblers, and BASIC. Once you have learned to use such programs, this Software Reference Manual will be an invaluable quick reference on how to carry out specific functions within the H8 software packages.

TAPE FILES

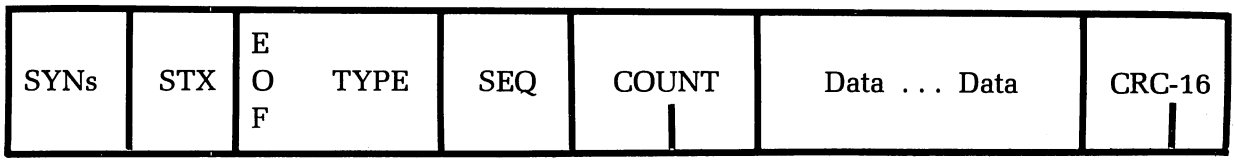
This section describes the tape format used in the Heath H8 Computer System. Tape formats are identical, regardless of the media used. The following terms are used to define the Heath H8 Tape format.

FILE — A logically complete set of data. For example, a memory dump causes the FILE to be written on the tape. Although several files may be written onto one tape, the files are each totally independant of any other information written on that tape. A file consists of one or more records.

RECORD — A record is a discrete block of data written to the tape transport. Each record must be read all at one time. It is not possible to read part of the record, pause, and then read the rest. Each record contains a CRC-16 Check. Each file has a first and last record. They may be the same record in a one-record file. The records in the file are numbered so a missing record can be detected.

System Record Structure

As discussed on Page 0-12, all H8 files consist of one or more records. All of the records have the same format.



SYN — From 20 to 40 ASCII Synchronizing Idle (026) characters.

STX — An ASCII STX character. This character, preceded by at least 10 SYN characters, indicates the start of a record. The SYN characters and the STX character are not included in the CRC. Note that a gap may be required between records to allow the tape transport to start and stop.

EOF — End of file. This flag is the high-order bit in the 'TYPE' byte. If set, it indicates that this is the last record in the file. The record is otherwise normal, and may contain data.

TYPE — This 7-bit field (the 8th bit is 'EOF') indicates the type of the record. All records in a file have the same type. The data field's format is type dependent. See below for a description of file types.

SEQ — This field is an 8-bit sequence counter, used to detect missing records. If a label record is present in the file, it is record #0. The first data record is record #1. If the file contains no label record, the first record is record #1. Note that the record following record #255 is record #0, but is not a label record.

COUNT — This two-byte field contains a count of the number of bytes in the Data field. The high-order byte of the count appears first. Note that the count may be zero, indicating that there is no data field.

DATA — This field contains the data. Its format is dependent upon the record type. Its length is set in 'count'.

CRC-16 This is a polynomial remainder check, computed byte-wise upon the entire record (starting with the EOF/TYPE byte) from (X + 1) * (X ↑5 + X + 1)
This checksum provides nearly flawless error detection.

ERROR	DETECTION RATE
Single bit error	100%
Double bit errors	100%
An odd number of bits in error	100%
An error burst <17 bits long	100%
An error burst >= 17 bits	99.997%

Label Record Format

Some file types require a label record to be present, and some require that no label record be present. A label record is detected by its record number of 0. Except for the contents of the data field, a label record has the same format as the other records in the file. The data field consists of a string of 7-bit ASCII characters which comprise the file's label. The 8th bit should be 0 for all characters.

System Data Formats

The following section describes the data formats associated with the various file types. There are currently three file types:

1. Memory Image.
2. BASIC programs.
3. Compressed Text.

MEMORY IMAGE (Type = 001)

The file type 'memory image' is used when you dump or load programs from H8 memory. This file type has no label record, the first record in the file is #1. The file may consist of one or more records. The format of the data field is:

ENTRY ADDR Program bytes

Where ENTRY = the program's entry point address, and ADDR = the address to start loading this group of program bytes. If there are multiple records in this data file, the 'entry' portion of each record should be identical.

NOTE: The COUNT field in the record header does not include the 4 bytes for ENTRY and ADDR. Thus, an empty record of this type has a zero COUNT field, but still contains the ENTRY and ADDR in the data field. Note that the high-order byte comes first for the ENTRY and ADDR fields.

BASIC PROGRAM (Type = 002)

This file type is used by BASIC when you load and dump programs. The file always has a label record (#0), and always has only one data record, #1. The data field contains the BASIC program in a special internal format. This file type can not be processed by the text editor.

COMPRESSED TEXT (Type = 003)

This file type is used by TED-8 and HASL-8 for source statements. It always has a label record (record #0) and has one or more data records. The data field in each record should not exceed 512 bytes. Lines should not be split between records. Each line is compressed according to the following format:

1. All characters are 7-bit ASCII, with the parity bit zero.
2. The carriage return and line-feed characters are not used. The end of line is indicated by a 000 byte.
3. Strings of spaces are represented by the value 200_8 + N, where 'N' is the number of spaces in the series. Thus, a single blank is encoded as 201_8; ten blanks are encoded as 212_8.
4. The maximum line length is 127 characters.

Reading the Displays

When the H8 computer is reading or writing data on a tape transport, the front panel displays are continually displaying data about the tape operation. Data about tape operation is displayed into two areas:

ADDRESS LEDs — Display the number of bytes left in the record when the transport is reading or writing data. The Address LEDs do not display any information during the inter-record gap. The address LEDs display the actual address being loaded when a memory image load or dump is executed. During a memory image operation, the Data LEDs display the data being entered into or read from memory.

DATA LEDs — Display the type of data and the record number. The right-hand-most data LED displays the type of data being read or written. This information is displayed as:

DISPLAY	DATA/TYPE
1.	Memory Image
2.	A BASIC program.
3.	Compressed text.

The two left hand LEDs display the octal record count within the particular file. As noted earlier, a file may contain one or more records. When 17_8 records are exceeded, the record count in the two left hand data LEDs starts over at 00. When the last record is read, the extreme left hand data LED displays a two or a three if the record count is between 10_8 and 17_8. Using this information, you can readily observe the type of data being handled by the H8. Figure 0-1 shows how these displays are used.

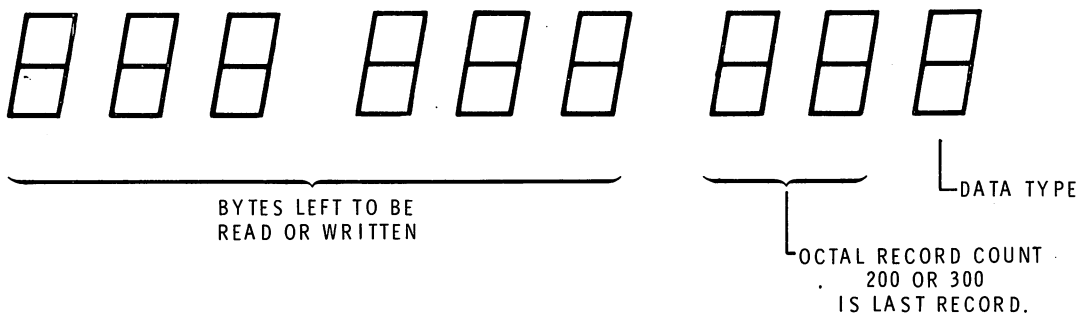
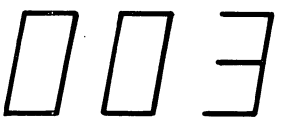


Figure 0-1

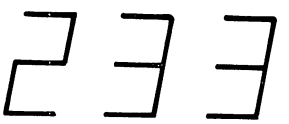
For example:

indicates the first and last data record of compressed text. NOTE: This could be the 1st or the 17_8th data record.

indicates first data record of a memory image file.



indicates a label record of a compressed text file. Note, the label record is record number 0.



indicates the third and last record of a compressed text file.

indicates the last record of a memory image file.

USING THE MAGNETIC TAPE SYSTEM

The Heath H8-5 serial card supports two different configurations of magnetic tape recorders. You may use a single tape recorder or a dual tape recorder. The most versatile system operation is obtained by using two tape recorders with independent control. However, you can achieve a perfectly workable and somewhat less expensive system with the single recorder. Dual recorders are preferable when you are assembling long programs, as it is necessary to read a few records of the source program and then assemble this source material, generating the appropriate binary output before reading additional source records. If you use a single recorder, you must change cassettes frequently.

Recorder Operating Hints

Observe the following guidelines carefully to get the maximum operating efficiency from your H8 Cassette Recorder system.

1. Use only Heath approved cassette tape recorders. Although there are a variety of very good tape recorders on the market, only those models tested and approved by the Heath Company will assure successful operation.

2. Use a high-quality cassette recording tape. Once again, the Heath approved tape is required to assure success.

3. Be sure the tape is off the leader before recording a file. Frequently cassettes have excessively long leaders. Therefore, the initial portion of a file placed on tape may lose the synchronizing characters, and be lost.

4. Keep the tape and the recorder clean. Dirty tapes tend to cause drop outs which cause tape errors.

5. Label your tapes. There is nothing more frustrating than a good program written onto a tape that is unlabeled and therefore not recoverable later.

USING THE PAPER TAPE SYSTEM

The H10 Paper Tape Reader/Punch is two independent devices. It is both a paper tape punch and a paper tape reader. When it is used with the H8-2 Parallel I/O Interface, operation of the H10 becomes identical to operation with dual cassette recorders.

Paper Tape Operating Hints

The following hints will help insure you of maximum effectiveness with your paper tape operating system.

1. Use oiled paper tape. Oiled paper tape helps keep the punch system operating smoothly and reduces punch wear.

2. Keep the punch well adjusted. After the first few hours of operation, carefully observe the punch mechanism to be sure it is still properly aligned.

3. Label all your tapes. Just like magnetic tapes, unlabeled paper tapes are virtually useless.

4. Keep the chad cleaned up. Excessive loose chad tends to cause reliability problems.

PRODUCT INSTALLATION

Use the following procedure as a guideline when you install your magnetic or paper tape operating system. Remember, a major part of your H8 System purchase is the software. The installation of the software should be treated with as much care and thought as the installation of the hardware products. Without the software, the hardware is of little or no value.

Creating a Configured Tape

The Heath H8 software is supplied on distribution tapes. These tapes enable you to configure the software for your own particular needs. Use the following procedure to create a configured tape from the software distribution tape.

A. Load tape in reader.
B. Ready tape transport.
C. Press LOAD on H8 front panel.
D. Wait for a single beep indicating successful load.
E. Press GO on H8 front panel.*

*NOTE: Any software patches received with the product should be entered before you execute their step. Once entered, they become a permanent part of the configured tape. (See Page 0-22.)

G. Configure the software product as desired, answering each of the following questions. Prompt each question by typing its first character on the console terminal keyboard. Simply type a return or do not prompt the question if you wish to leave them as distributed. The questions are:

AUTO NEWLINE (Y/N)?

A yes (Y) response to this question directs the product to generate a new line each time the print head (or cursor) moves out of the last column of the console terminal. This function is distributed preset to Y.

BKSP = 00008/

The backspace character is normally a control H (00008 decimal). When used with the video terminal or other backspacing devices, the control H generates a true backspace. The backspace character may be changed to other ASCII printing or nonprinting characters (See Appendix D) such as a backslash, if a non-backspacing terminal is used. This new character will be considered a true backspace by the software.

CONSOLE LENGTH = 00080/

The console length is initially set at 80 (decimal) characters, which is the normal width of a video terminal. This may be changed to other common values such as 132 characters for a wide printing terminal, or to 72 characters for a teleprinter.

NOTE: The maximum number of characters per line is a function of each software product. See the individual sections to determine the maximum permissible characters per line.

HIGH MEMORY = XXXXX/

When the software product is initially started, the limit of available high memory is determined. All products start at 040 100 (offset octal). If you wish not to use a certain portion of high memory, a new high memory limit (decimal count) should be typed in. If the upper memory limit is set too low, the new limit will be refused (the terminal bell will sound).

LOWER CASE (Y/N)?

A Yes (Y) response to LOWER CASE configures the software product to input lower case letters and output lower case letters. An N (no) in response to the question configures the product to work with upper case only terminals. This function is distributed preset to N.

PAD = 4/

The pad characters (nulls) are inserted following a carriage return. The pad characters are sent at this time to allow the print head time to return to the left-hand margin. For video terminals, and most teleprinters, the number of pad characters may be changed to zero. If you do not know how many pad characters are required for your terminal, initially try zero. If you appear to to be overtyping (or missing characters) at the beginning of lines, increase the pad count until the overtyping stops. You may enter up to a maximum of 9 pad characters.

RUB OUT = 00127/

The rub out character is set as 127 (decimal). If you desire to use a special rub out character, you may change it by entering in a new decimal number identifying a different ASCII character. See Appendix D.

SAVE?

A yes (Y) in response to this question directs the software product to generate a memory image of the configured product. This memory image of the configured product should be the tape you use regularly to load your program. This will avoid your having to configure the product on a regular basis. Before executing the save command, be sure the tape transport at the dump output is ready.

To use the product directly from the distribution tape, type the return key at any time rather than typing a key which prompts a question.

NOTE: It is very important that you immediately configure products as you will use them, and then place your original software distribution tape in an appropriate place for safe keeping. Use the above procedure any time you wish to configure the product.

Loading From a Configured Tape

Loading from a configured tape is a very simple procedure. It is the recommended way to normally load the software. The procedure is:

1. Load tape in the tape transport.
2. Ready the tape transport.
3. Press LOAD on the H8 front panel.
4. Wait for a single beep, indicating a successful load.
5. Press GO key on the H8 front panel.
6. The console terminal will respond with the product description and its prompt character. The product is now ready to use in a preconfigured form.

Copying an Existing Memory Tape

Use the following procedure to copy a memory image tape. Be sure to use this procedure. Memory image tapes should not be copied on an audio-to-audio basis. An audio-to-audio copy may not work. To copy a tape:

1. Load the source tape in the tape transport.
2. Ready the tape transport.
3. Press LOAD on the H8 front panel.
4. Wait for a single beep, indicating a successful load.
5. Load a blank tape into the dump tape transport.
6. Ready the dump tape transport.
7. Press DUMP on the H8 front panel.
8. Wait for a single beep, indicating a completed dump.
9. Repeat steps 7 and 8 to produce a second dump, creating a double copy.

The product is now copied and ready to be used. IMPORTANT: Make at least one double copy of the distribution tapes you received with the H8 as a protection against accidental tape damage. Once the tape is copied, if magnetic tape is being used, the read-only plugs should be knocked out of the back of the cassette.

Installing a Patch

To implement a patch supplied with the distribution software tapes or those in Appendix A, load the distribution tape following steps A - P in "Creating a Configured Tape" (Page 0-19). Alter the memory contents at the locations shown in the desired patch(s), inserting the new data given in the patch. Refer to Page 1-10 "Displaying and Altering Memory Locations" for the appropriate procedure to modify a memory location. For example, to use Option Patch #1 on BENTON HARBOR BASIC:

1. Load the distribution tape through step D.
2. Change the contents of location 041 010 to 316 and the contents of location 064 077 to 001. This completes step D.
3. Finish the configuration continuing with step E. BENTON HARBOR BASIC is now patched to 5.01.00.I and will supply two stop bits on each transmitted ASCII character.

Using an ASR Console

The following procedure allows you to use an ASR console as the main load/dump port, as well as the console terminal with an H8 system. An example of such an ASR (automatic send/receive) console would be the Teletype Corporation Model 33 Teleprinter. Perform the initial load by first setting the port interchange switch to the port interchange position on your H8-5 Serial I/O card. The tapes are then read in, in accordance with the procedure outlined under "Creating a Configured Tape." Once the tapes are read in, PAM-8 should be used to patch the software to the ASR console terminal configuration.

The appropriate patches for the ASR console terminal configurations of the software products are found in Appendix A of this section. Once the patches are accomplished, the GO key may be pressed and the normal configuration procedure takes place. Leave the Port Interchange switch in the Port Interchange position as long as the tape handler on the console terminal is used as the main load dump terminal.

Using a 110-Baud Console Terminal

If you use a 110-baud console terminal such as a teleprinter, one extra stop bit must be added to the ASCII characters sent by the H8. This change is made at configuration time and should be done any time you use a 110-baud console, regardless of whether or not you use the terminal as an ASR console (such as a teleprinter with a paper tape reader/punch) or simply as the console terminal. The patch is listed in Appendix A.

Console Interface

Appendix B contains a listing of the H8 Console Driver, a software module included in all H8 software packages which utilize a console terminal. A console driver is a general-purpose software package, providing you with such special characters as Control A, Control B, Control C, Control D, Control O, Control P, Control S, and Control Q.

The characters Control-A, -B, -C, and -D are available for use within the program. For example, most Heath programs use Control-C as a general purpose cancel. The other characters are permanently assigned in the console driver, and therefore in all Heath software products using the console driver. These characters are assigned as follows:

CONTROL-O

Control-O toggles the output discard flag. When the output discard flag is set, output to the console terminal is stopped, but program execution continues. Typing Control-O once sets the discard flag. Typing the Control-O again resets the output flag, permitting output to the console terminal to resume.

CONTROL-P

Control-P resets the output discard flag. Typing Control-P insures the output discard flag is not set. NOTE: Control-O toggles the flag, but Control-P only resets it.

CONTROL-S

Control-S sets the suspend output flag.

CONTROL-Q

Control-Q resets the suspend output flag.

The above control characters are not echoed to the console terminal when they are typed as is a normal character. NOTE: Many Heath Software products use Control-H, Control-I and Rubout. These characters are not used by the console driver but are passed directly through to the program for individual processing. They are also echoed to the console terminal.

CONSOLE DRIVER

The console driver also provides all capabilities for communicating with the console terminal and a tape transport at the load/ dump ports. If you, the user, develop any of your own software packages, we recommend that you incorporate this console driver rather than attempting to develop your own console driver.

The use of the control characters is explained on Page 3 of the console driver listing and in the individual software product reference sections.

The console driver also provides two front panel entry points. These are listed on Page 5 of the console driver listing. They are:

PROGRAM COUNTER*	ENTRY TYPE
040 100	Program Reset entry point. All text buffers, etc., are effectively cleared and the product is re-started. This is known as a "cold" or "hard" start.
040 103	Program restart entry point. Product is restarted with text buffers, etc., intact. This is known as a "warm" or "soft" start.

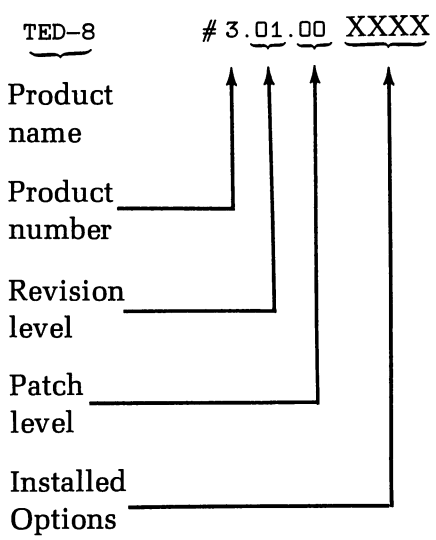
*NOTE: Set the value of the program counter using the front panel monitor. Then press GO.

Reporting Software Problems

Every effort has been made to keep the Heath H8 Software products free of defects. Should you suspect that a Heath H8 system software product may be defective, review the following procedure before contacting Heath Company.

1. Attempt to reload all software to be sure it has not been damaged.
2. Reconfigure the product from the distribution tape in an attempt to duplicate the problem.
3. If the problem persists, document the apparent product problem and the software version code. NOTE: It is extremely important that you know your exact software version code when you contact the Heath Company about your software product.

For example:



NOTE: Heath Company can not consult on user developed programs or modified versions of the Heath Software products.

APPENDIX A

This appendix is a listing of the patches required to use an ASR console terminal as both the load/dump port and as the console terminal (option patch #2). A patch to convert the output to two-stop bits is also included (option patch #1).

BUG-8
02.01.xx.

OPTION PATCH #2

ASR CONSOLE

USE: This patch is used for systems which have both their console terminal and load/dump device attached to the same port, such as a Teletype model 33 ASR. For those systems, these patches are required. Note that the port interchange switch must be set to interchange.

NOTES: These patches remove the VERIFY command from BUG-8. It will no longer be valid. Also, CTL-C is no longer effective during tape operations. The proper procedure for aborting a load or dump is to return to the front panel monitor (via the front panel keyboard), set PC to the warm start address: 040103, and press 'GO'.

```
040107  370
040112  370
040115  371
040120  371
040123  371
040126  371
041004  072
041014  072
044046  072
046067  315 036 041 257
046231  251 046
046252  036 041 303 360 052
051164  311
051171  057 053
053047  055 052 315 036 041 303 305 050 315 366 051 303 036 041
053321  003
056360  001
```

TED - 8
03.01.xx.

OPTION PATCH #2

ASR CONSOLE

USE: This patch is used for systems which have both their console terminal and load/dump device attached to the same port, such as a Teletype model 33 ASR. For those systems, these patches are required. Note that the port interchange switch must be set to interchange.

NOTES: These patches remove the VERIFY command from TED-8. It will no longer be valid. CNTRL-C is no longer effective during tape operations. The proper procedure for aborting a load or a dump is to return to PAM-8 (via RTM on the PAm-8 keyboard), set Pc to the warm start address: 040103, and press 'GO'.

```
040107  370
040112  370
040115  371
040120  371
040123  371
040126  371
041004  072
041014  072
041170  072
042036  072
046267  146 047
046274  324
046324  315 036 041
047146  356 200 310 303 324 046
047156  357 001 200 052 226 057 311
053064  315 156 047
055074  311
055101  336 056
056326  363 055 315 036 041 303 044 054 315 274 055 303 036 041
057143  003
064100  001
```

HASL - 8
04.01.xx.

OPTION PATCH #2

ASR CONSOLE

USE: This patch is used for systems which have both their console terminal and load/dump device attached to the same port, such as a Teletype model 33 ASR. For those systems, these patches are required. Note that the port interchange switch must be set to interchange.

NOTES: None.

```
040107  370
040112  370
040115  371
040120  371
040123  371
040126  371
041004  072
041014  072
057022  064 061
061057  104 060 303 036 041 315 015 060
061067  303 036 041
072316  001
```

BENTON HARBOR BASIC
05.01.xx.

OPTION PATCH #2

ASR CONSOLE

USE: This patch is used for systems which have both their console terminal and load/dump device attached to the same port, such as a Teletype model 33 ASR. For those systems, these patches are required. Note that the port interchange switch must be set to interchange.

NOTES: These patches remove the VERIFY command from B.H. BASIC. It will no longer be valid. Also, CTL-C is no longer effective during tape operations. The proper procedure for aborting a load or dump is to return to the front panel monitor (via the front panel keyboard), set Pc to the warm start address: 040103, and press 'GO'.

```
040107  370
040112  370
040115  371
040120  371
040123  371
040126  371
041004  072
041014  072
044356  303 252 046
046252  265 341 310 315 036 041 303 361
046262  044 302 160 067 341 303 340 070
057071  003
057171  036 041
067166  303 263 046
070034  311
070060  106 071
071076  017 003 315 036 041 303 020 067
071106  315 325 002 303 036 041
075202  001
```

EXTENDED BENTON HARBOR BASIC
10.01.xx.

OPTION PATCH #2

ASR CONSOLE

USE: This patch is used for systems which have both their console terminal and load/dump device attached to the same port, such as a Teletype model 33 ASR. For those systems, these patches are required. Note that the port interchange switch must be set to interchange.

NOTES: These patches remove the VERIFY command from BASIC. It will no longer be valid. Also, CTL-C is no longer effective during tape operations. The proper procedure for aborting a load or dump is to return to the front panel monitor (via the front panel keyboard), set Pc to the warm start address: 040103, and press 'GO'.

```
040107  370
040112  370
040115  371
040120  371
040123  371
040126  371
041004  072
041014  072
046307  303 316 050
050316  265 341 310 315 036 041 303 312
050326  046 302 151 100 341 303 011 100
064226  003
065110  036 041
100157  303 270 050
101116  311
101142  313 102
102303  024 102 315 036 041 303 011 100
102313  315 335 101 303 036 041
106173  001
```

BUG - 8
02.01.xx.

OPTION PATCH #1

2 STOP BITS

USE: This patch is inserted for systems which use a terminal device requiring 2 stop bits. This should not be used for devices which can run with only one stop bit.

NOTES: None.

041010 316
056357 001

TED - 8
03.01.xx.

OPTION PATCH #1

2 STOP BITS

USE: This patch is inserted for systems which use a terminal device requiring 2 stop bits. This should not be used for devices which can run with only one stop bit.

NOTES: None.

041010 316
064077 001

HASL - 8
04.01.xx.

OPTION PATCH #1

2 STOP BITS

USE: This patch is inserted for systems which use a terminal device requiring 2 stop bits. This should not be used for devices which can run with only one stop bit.

NOTES: None.

041010 316
072315 001

BENTON HARBOR BASIC
05.01.xx.

OPTION PATCH #1

2 STOP BITS

USE: This patch is inserted for systems which use a terminal device requiring 2 stop bits. This should not be used for devices which can run with only one stop bit.

NOTES: None.

041010 316
075201 001

EXTENDED BENTON HARBOR BASIC
10.01.xx.

OPTION PATCH #1

2 STOP BITS

USE: This patch is inserted for systems which use a terminal device requiring 2 stop bits. This should not be used for devices which can run with only one stop bit.

NOTES: None.

041010 316
106173 001

APPENDIX B

Console Driver Listing

```
                        2  ***     HEATH H8 SOFTWARE CONSOLE DRIVER.
                        3  *
                        4  *       JGL, 01/01/77, FOR *HEATH* COMPANY.
                        5  *
                        6  *       COPYRIGHT 1977 BY HEATH COMPANY,
                        7  *                         BENTON HARBOR, MI.
                        8
                        9
                       10  ***     THE FOLLOWING CONTAINS THE TEXT FOR THE HEATH H8
                       11  *       CONSOLE DRIVER. THESE EXACT ROUTINES ARE USED IN
                       12  *       ALL SOFTWARE PRODUCTS.
                       13  *
                       14  *       ALL PROGRAMS WISHING TO COMMUNICATE WITH THE CONSOLE
                       15  *       SHOULD USE THESE ROUTINES.
                       16
                       17
040.100                18          ORG     40100A          START OF USER RAM
                       19
                       20
                       21  **      ASSEMBLY CONSTANTS.
                       22
000.007                23  BELL    EQU     007Q            ASCII BELL
040.037                24  .UIVEC  EQU     040037A         H8MTR USER INTERRUPT VECTORS
111.111                25  START   EQU     111111A         DUMMY START LABEL
222.222                26  RESTART EQU     222222A         DUMMY RESTART LABEL
333.333                27  CONFIG  EQU     333333A         DUMMY CONFIGURE LABEL
                       28
040.100                29          XTEXT   U8251
```

```
                        32X **      8251 USART BIT DEFINITIONS.
                        33X *
                        34X
                        35X **      MODE INSTRUCTION CONTROL BITS.
                        36X
000.100                 37X UMI.1B  EQU     01000000B       1 STOP BIT
000.200                 38X UMI.HB  EQU     10000000B       1 1/2 STOP BITS
000.300                 39X UMI.2B  EQU     11000000B       2 STOP BITS
000.040                 40X UMI.PE  EQU     00100000B       EVEN PARITY
000.020                 41X UMI.PA  EQU     00010000B       USE PARITY
000.000                 42X UMI.L5  EQU     00000000B       5 BIT CHARACTERS
000.004                 43X UMI.L6  EQU     00000100B       6 BIT CHARACTERS
000.010                 44X UMI.L7  EQU     00001000B       7 BIT CHARACTERS
000.014                 45X UMI.L8  EQU     00001100B       8 BIT CHARACTERS
000.001                 46X UMI.1X  EQU     00000001B       CLOCK X 1
000.002                 47X UMI.16X EQU     00000010B       CLOCK X 16
000.003                 48X UMI.64X EQU     00000011B       CLOCK X 64
                        49X
                        50X **      COMMAND INSTRUCTION BITS.
                        51X
000.100                 52X UCI.IR  EQU     01000000B       INTERNAL RESET
000.040                 53X UCI.RO  EQU     00100000B       READER-ON CONTROL FLAG
000.020                 54X UCI.ER  EQU     00010000B       ERROR RESET
000.004                 55X UCI.RE  EQU     00000100B       RECEIVE ENABLE
000.002                 56X UCI.IE  EQU     00000010B       ENABLE INTERRUPTS FLAG
000.001                 57X UCI.TE  EQU     00000001B       TRANSMIT ENABLE
                        58X
                        59X **      STATUS READ COMMAND BITS.
                        60X
000.040                 61X USR.FE  EQU     00100000B       FRAMING ERROR
000.020                 62X USR.OE  EQU     00010000B       OVERRUN ERROR
000.010                 63X USR.PE  EQU     00001000B       PARITY ERROR
000.004                 64X USR.TXE EQU     00000100B       TRANSMITTER EMPTY
000.002                 65X USR.RXR EQU     00000010B       RECEIVER READY
000.001                 66X USR.TXR EQU     00000001B       TRANSMITTER READY
040.100                 67          XTEXT   CONSLX
```

```
                    70X ***     CONSL - SYSTEM CONSOLE AND I/O DRIVER.
                    71X *
                    72X *       CONSL IS A GENERAL PURPOSE CONSOLE DRIVER. IT IS A STANDARD
                    73X *       PRODUCT USED IN ALL HEATH H8 SOFTWARE (WHICH COMMUNICATES WITH
                    74X *       A CONSOLE DEVICE).
                    75X *
                    76X *       CONSL CONTAINS:
                    77X *
                    78X *       1) PORT ROUTINES. THESE ARE PLACED IN CONSOLE SO THEY HAVE THE SAME
                    79X *          LOCATION IN ALL PRODUCTS. PORT ADDRESSES MAY BE CHANGED BY PATCHING
                    80X *          THESE ROUTINES.
                    81X *
                    82X *       2) THE CONSOLE DRIVER PACKAGE. THIS PACKAGE CONSISTS OF THREE ROUTINES:
                    83X *             $RCHAR - READ A SINGLE CHARACTER
                    84X *             $WCHAR - WRITE A SINGLE CHARACTER
                    85X *             $PRSCL - PRESET CONSOLE AND TAPE UARTS.
                    86X *
                    87X *       THE CONSOLE PACKAGE PROVIDES SOPHISTICATED SUPPORT FOR
                    88X *       ITS CALLERS:
                    89X *                INTERRUPT PROCESSING FOR INPUT CHARACTERS
                    90X *                28 CHARACTER TYPE-AHEAD CAPABILITY
                    91X *                SPECIAL CONTROL CHARACTER PROCESSING.
                    92X *
                    93X *       NOTE THAT IF THE CONSOLE PACKAGE IS USED BY ANY RUNNING
                    94X *       ROUTINE, ALL ROUTINES RUNNING WITH IT MUST USE IT ALSO.
                    95X *       THIS IS BECAUSE *CONSL* WILL PROCESS INPUT CHARACTERS AT
                    96X *       INTERRUPT TIME, BEATING OUT ANY TASK-TIME ROUTINE WHICH
                    97X *       ATTEMPTS TO READ CHARACTERS.


                    99X **      SPECIAL CHARACTER PROCESSING.
                   100X *
                   101X *       *CONSL* SUPPORTS 8 SPECIAL CHARACTERS:
                   102X *
                   103X *       CTL-A          USER DEFINED CONTROL FLAGS. THESE CAN BE CHECKED
                   104X *       CTL-B           AT TASK TIME, OR THE USER PROGRAM CAN SET UP AN
                   105X *       CTL-C           INTERRUPT VECTOR WHICH IS ENTERED, AT INTERRUPT
                   106X *       CTL-D           TIME, WHEN ANY OF THESE CHARACTERS ARE ENTERED.
                   107X *
                   108X *       CTL-O          TOGGLE DISCARDING OUTPUT CHARACTERS
                   109X *       CTL-P          RESUME OUTPUTTING CHARACTERS
                   110X *       CTL-S          SUSPEND OUTPUT
                   111X *       CTL-Q          RESUME OUTPUT
```

```
                        113X **     CONSOLE DRIVER ASSEMBLY CONSTANTS.
                        114X *
                        115X
                        116X
                        117X **     I/O PORT ADDRESSES.
                        118X
000.372                 119X IP.CDP EQU   372Q          CONSOLE DATA IN PORT
000.372                 120X OP.CDP EQU   372Q          CONSOLE DATA OUT PORT
000.373                 121X IP.CIS EQU   373Q          CONSOLE INPUT STATUS IN PORT
000.373                 122X OP.CIS EQU   373Q          CONSOLE INPUT STATUS OUT PORT
000.373                 123X IP.COS EQU   373Q          CONSOLE OUTPUT STATUS IN PORT
000.373                 124X OP.COS EQU   373Q          CONSOLE OUTPUT STATUS OUT PORT
                        125X
000.370                 126X IP.TDP EQU   370Q          TAPE DATA IN PORT
000,370                 127X OP.TDP EQU   370Q          TAPE DATA OUT PORT
000.371                 128X IP.TSP EQU   371Q          TAPE STATUS IN PORT
000.371                 129X OP.TSP EQU   371Q          TAPE STATUS OUT PORT

                        131X **     $CSLCTL (CONSOLE CONTROL FLAG) BITS.
                        132X *
                        133X *      THESE BITS ARE SET IN $CSLCTL WHEN THE APPROPRIATE CONTROL
                        134X *      CHARACTERS ARE STRUCK.
                        135X *
                        136X *      THESE CAN BE EXAMINED AT TASK-TIME, OR WHEN THE USERS CONTROL
                        137X *      CHARACTER ROUTINE IS ENTERED (AT INTERRUPT TIME)
                        138X
                        139X
000.001                 140X CC.HLD EQU   01            SUSPEND OUTPUT
000.002                 141X CC.DMP EQU   02            DISCARD OUTPUT
000.010                 142X CC.CTLA EQU  010Q          CTL-A
000.020                 143X CC.CTLB EQU  020Q          CTL-B
000.040                 144X CC.CTLC EQU  040Q          CTL-C
000.100                 145X CC.CTLD EQU  100Q          CTL-D
```

```
                                147X **      PROGRAM ENTRY POINTS.
                                148X
040.100  303 333 333            149X ENTRY   JMP     CONFIG          PROGRAM RESET ENTRY POINT
040.103  303 222 222            150X         JMP     RESTART         PROGRAM RESTART ENTRY POINT


                                152X **      PORT ROUTINES.
                                153X *
                                154X *       ALL PROGRAMS MUST USE THESE ROUTINES, TO ALLOW PORT ADDRESS
                                155X *       CHANGEABILITY.
                                156X
                                157X
040.106  333 372                158X $CDIN   IN      IP.CDP          CONSOLE DATA IN
040.110  311                    159X $RET    RET
                                160X
040.111  323 372                161X $CDOUT  OUT     OP.CDP          CONSOLE DATA OUT
040.113  311                    162X         RET
                                163X
040.114  333 373                164X $CISI   IN      IP.CIS          CONSOLE INPUT STATUS IN
040.116  311                    165X         RET
                                166X
040.117  323 373                167X $CISO   OUT     OP.CIS          CONSOLE INPUT STATUS OUT
040.121  311                    168X         RET
                                169X
040.122  333 373                170X $COSI   IN      IP.COS          CONSOLE OUTPUT STATUS IN
040.124  311                    171X         RET
                                172X
040.125  323 373                173X $COSO   OUT     OP.COS          CONSOLE OUTPUT STATUS OUT
040.127  311                    174X         RET
                                175X
040.130  333 370                176X $TDIN   IN      IP.TDP          TAPE DATA IN
040.132  311                    177X         RET
                                178X
040.133  323 370                179X $TDOUT  OUT     OP.TDP          TAPE DATA OUT
040.135  311                    180X         RET
                                181X
040.136  333 371                182X $TSIN   IN      IP.TSP          TAPE STATUS IN
040.140  311                    183X         RET
                                184X
040.141  323 371                185X $TSOUT  OUT     OP.TSP          TAPE STATUS OUT
040.143  311                    186X         RET


                                188X **      REMOTE ENTRY POINTS FOR CONSOLE DRIVER.
                                189X *
                                190X
040.144  303 255 040            191X $RCHAR  JMP     $RCHAR.         READ ONE CHARACTER
                                192X
040.147  303 332 040            193X $WCHAR  JMP     $WCHAR.         WRITE ONE CHARACTER
                                194X
040.152  303 347 040            195X $PRSCL  JMP     $PRSCL.         PRESET CONSOLE UART
```

```
                                197X **     DATA AND BUFFERS.
                                198X *
                                199X
                                200X
040.155  000                   201X $INBUF  DB      0              INPUT BUFFER COUNT
040.156                        202X         DS      30             TYPE AHEAD BUFFER. FIRST BYTE = NEXT CHARACTER
                                203X
000.035                        204X $INBUFL EQU     *-$INBUF-2     MAX LENGTH OF BUFFER


                                206X **     CONTROL CHARACTER TABLE.
                                207X *
                                208X *      DB      CHAR,MASK,VALUE
                                209X *
                                210X *      IF CHAR, $CSLCTL = ($CSLCTL .AND. MASK) .OR. VALUE
                                211X
040,214  000 377 000            212X $CSIB   DB      00Q,377Q,000Q  CTL-@ (00 = NULL)
040.217  001 167 210            213X         DB      01Q,167Q,210Q  CTL-A
040.222  002 157 220            214X         DB      02Q,157Q,220Q  CTL-B
040.225  003 137 240            215X         DB      03Q,137Q,240Q  CTL-C
040.230  004 077 300            216X         DB      04Q,077Q,300Q  CTL-D
040.233  017 177 002            217X         DB      17Q,177Q,002Q  CTL-O
040.236  020 175 000            218X         DB      20Q,175Q,000Q  CTL-P
040.241  021 176 000            219X         DB      21Q,176Q,000Q  CTL-Q
040.244  023 176 001            220X         DB      23Q,176Q,001Q  CTL-S
040.247  177                    221X         DB      177Q           END OF TABLE


                                223X **     $CSIC - ADDRESS OF INTERRUPT TIME CONTROL CHARACTER PROCESSOR.
                                224X *
                                225X *      $CSIC CONTAINS THE ADDRESS OF THE ROUTINE CONSL ENTERS WHENEVER A
                                226X *      CONTROL CHARACTER (CTL-A THROUGH CTL-D) IS STRUCK. THE PROPER BITS
                                227X *      ARE SET/CLEARED IN $CSLCTL, AND THE ROUTINE IS ENTERED AT INTERRUPT
                                228X *      TIME. AFTER THE USER ROUTINE HAS COMPLETED PROCESSING, IT SHOULD CLEAR
                                229X *      THE BITS FROM $CSLCTL, AND RETURN TO THE SYMBOL '$RET'. IF INTERRUPT
                                230X *      PROCESSING IS NOT REQUIRED, LEAVE THE $RET ADDRESS IN $CSIC.
                                231X
040.250  110 040               232X $CSIC   DW      $RET           ADDRESS OF USER ROUTINE FOR CTL-A THROUGH CTL-D
                                233X
040.252  000                   234X $CSLCTL DB      0              CONSOLE CONTROL BYTE
040.253  000                   235X COLNO   DB      0              CONSOLE CURSOR POSITION
040.254  120                   236X $CSLLEN DB      80             CONSOLE WIDTH
                                237X
000.040                        238X         SET     $INBUF/1000A
000.000                        239X         ERRNZ   */1000A-.      ALL DATA MUST BE IN SAME PAGE
```

```
                                243X **      $RCHAR - READ SINGLE CHARACTER.
                                244X *
                                245X *       $RCHAR IS CALLED TO READ A CONSOLE CHARACTER.
                                246X *
                                247X *       ENTRY   NONE
                                248X *       EXIT    (A) = CHAR (PARITY BIT CLEARED)
                                249X *       USES    A,F
                                250X
                                251X
040.255  345                   252X $RCHAR: PUSH    H               SAVE (HL)
040.256  041 155 040           253X         LXI     H,$INBUF        (HL) = ADDRESS OF CHARACTER POINTER
040.261  176                   254X $RCHAR1 MOV     A,M             (A) = COUNTER
040.262  247                   255X         ANA     A
040.263  312 261 040           256X         JZ      $RCHAR1         WAIT FOR INTERRUPT TO READ CHARACTER
                                257X
040.266  363                   258X         DI                      INTERLOCK SEQUENCE
040.267  325                   259X         PUSH    D
040.270  065                   260X         DCR     M               DECREMENT COUNT
040.271  126                   261X         MOV     D,M             (D) = COUNT-1
040.272  043                   262X         INX     H
040.273  176                   263X         MOV     A,M             (A) = READ CHARACTER
040.274  376 173               264X         CPI     173Q            SEE IF LOWER CASE
040.276  322 310 040           265X         JNC     $RC1.5
040.301  376 141               266X         CPI     141Q
040.303  332 310 040           267X         JC      $RC1.5          IS NOT LOWER CASE
040.306  326 040               268X         SUI     40Q             MAKE UPPER
040.307                        269X $RCHARA EQU     *-1             ZEROED FOR LOWER CASE
040.310  365                   270X $RC1.5  PUSH    PSW             SAVE IT
                                271X
040.311  025                   272X $RCHAR2 DCR     D               MOVE OTHERS DOWN IN QUEUE
040.312  372 325 040           273X         JM      $RCHAR3         NO MORE
040.315  043                   274X         INX     H
040.316  176                   275X         MOV     A,M
040.317  053                   276X         DCX     H
040.320  167                   277X         MOV     M,A
040.321  043                   278X         INX     H
040.322  303 311 040           279X         JMP     $RCHAR2
                                280X
040.325  373                   281X $RCHAR3 EI                      RESTORE INTERRUPTS
040.326  361                   282X         POP     PSW
040.327  321                   283X         POP     D
040.330  341                   284X         POP     H
040.331  311                   285X         RET                     EXIT: (A) = CHAR
```

```
                                287X **      $WCHAR - WRITE SINGLE CHARACTER.
                                288X *
                                289X *       $WCHAR IS CALLED TO OUTPUT A SINGLE CHARACTER.
                                290X *
                                291X *       ENTRY   (A) = CHARACTER
                                292X *       EXIT    NONE
                                293X *       USES    NONE
                                294X
                                295X
040.332  365                   296X $WCHAR. PUSH    PSW             SAVE CHARACTER
040.333  072 252 040           297X $WCHAR1 LDA     $CSLCTL         CHECK CONTROL
000.000                        298X         ERRNZ   CC.HLD-1
040.336  037                 299X         RAR
040.337  332 333 040           300X         JC      $WCHAR1         AM TO WAIT
040.342  037                 301X         RAR
000.000                        302X         ERRNZ   CC.DMP-2
040.343  322 354 040           303X         JNC     $WCHAR2         AM TO PRINT
040.346  361                   304X         POP     PSW             CTL-O IN EFFECT, DISCARD CHAR
040.347  311                 305X         RET
                                306X
                                307X *       OVERFLOWN TYPE-AHEAD BUFFER, ECHO BELL.
                                308X
040.350  065                   309X $CSI2   DCR     M               REMOVE LAST CHARACTER INPUT TO MAKE ROOM
040.351  076 007               310X         MVI     A,BELL
040.353  365                 311X         PUSH    PSW             STORE CHARACTER
                                312X
                                313X *       TYPE CHARACTER.
                                314X
040.354  315 122 040           315X $WCHAR2 CALL    $COSI
000.000                        316X         ERRNZ   USR.TXR-1
040.357  037                 317X         RAR
040.360  322 354 040           318X         JNC     $WCHAR2         WAIT FOR ROOM
040.363  361                 319X         POP     PSW
040.364  303 111 040           320X         JMP     $CDOUT          OUTPUT AND EXIT


                                322X **      $PRSCL - PRESET CONSOLE DRIVERS.
                                323X *
                                324X *       $PRSCD IS CALLED TO PRESET TERMINAL OPERATIONS.
                                325X *       IT ASSUMES THAT THE SAME PORT IS BEING USED FOR CONSOLE INPUT AS FOR
                                326X *       CONSOLE OUTPUT, BECAUSE THE CONFIGURATION IS DONE THROUGH
                                327X *       THE INPUT PORT ROUTINES.
                                328X *
                                329X *       ENTRY   NONE
                                330X *       EXIT    NONE
                                331X *       USES    A,F
                                332X
                                333X
040.367  076 201               334X $PRSCL. MVI     A,201Q
040.371  315 117 040           335X         CALL    $CISO           GUARANTEE OUT OF MODE-SET
040.374  315 141 040           336X         CALL    $TSOUT
040.377  076 100               337X         MVI     A,UCI.IR
041.001  315 117 040           338X         CALL    $CISO
041.004  315 141 040           339X         CALL    $TSOUT          FORCE INTO MODE-SET
```

```
041.007  076 116        340X          MVI     A,UMI.1B+UMI.L8+UMI.16X
041.011  315 117 040    341X          CALL    $CISO
041.014  315 141 040    342X          CALL    $TSOUT
041.017  257            343X          XRA     A
041.020  062 155 040    344X          STA     $INBUF
041.023  076 303        345X          MVI     A,303Q          SET UP INTERRUPT VECTOR
041.025  062 045 040    346X          STA     .UIVEC+6
041.030  041 043 041    347X          LXI     H,$CSINT
041.033  042 046 040    348X          SHLD    .UIVEC+7
041.036  076 027        349X          MVI     A,UCI.ER+UCI.RE+UCI.IE+UCI.TE
041.040  303 117 040    350X          JMP     $CISO           ENABLE CONSOLE OUTPUT
```

```
                                354X **     $CSINT - CONSOLE INTERRUPT PROCESSOR.
                                355X *
                                356X *      CONSOLE READ INTERRUPTS ENTER HERE FROM PAM-8.
                                357X *
                                358X *      ENTRY   NONE
                                359X *      EXIT    'EI' AND 'RET'
                                360X *      USES    NONE
                                361X
                                362X
041.043  345                    363X $CSINT PUSH    H
041.044  365                    364X        PUSH    PSW             PRESERVE REGISTERS
041.045  315 114 040            365X        CALL    $CISI
041.050  346 002                366X        ANI     USR.RXR
041.052  304 061 041            367X        CNZ     $CSI1           IF HAVE DATA FROM INPUT
041.055  361                    368X $CSIX  POP     PSW
041.056  341                    369X        POP     H
041.057  373                    370X        EI
041.060  311                    371X        RET
                                372X
                                373X **     HAVE DATA INPUT INTERRUPT. SEE IF ROOM IN THE QUEUE.
                                374X
041.061  041 155 040            375X $CSI1  LXI     H,$INBUF
041.064  076 035                376X        MVI     A,$INBUFL
041.066  276                    377X        CMP     M
041.067  334 350 040            378X        CC      $CSI2           NO ROOM
                                379X
                                380X *      ADD CHARACTER TO QUEUE.
                                381X
041.072  064                    382X $CSI3  INR     M
041.073  176                    383X        MOV     A,M
041.074  205                    384X        ADD     L
041.075  157                    385X        MOV     L,A             (HL) = ADDRESS OF CHARACTER
041.076  315 106 040            386X        CALL    $CDIN           INPUT CHARACTER
041.101  346 177                387X        ANI     177Q            TRIM IT
041.103  167                    388X        MOV     M,A             STORE IT
                                389X
                                390X *      CHECK FOR SPECIAL CONTROL SEQUENCES.
                                391X
                                392X *      CTL-A TO CTL-D          SPECIAL USER INTERRUPT PROCESSING
                                393X *      CTLO                    SET DISCARD MODE
                                394X *      CTL-P                   CLEAR DISCARD MODE
                                395X *      CTL-Q                   CLEAR HOLD MODE
                                396X *      CTL-S                   SET HOLD MODE
                                397X
041.104  376 040                398X        CPI     ' '
041.106  320                    399X        RNC                     NOT CONTROL CHARACTER
                                400X
                                401X *      HAVE CONTROL CHARACTER.
                                402X
041.107  056 212                403X        MVI     L,#$CSIB-2
041.111  043                    404X $CSI5  INX     H
041.112  043                    405X        INX     H               POINT AT NEXT ELEMENT IN TABLE
041.113  276                    406X        CMP     M               COMPARE TO CHARACTER
041.114  330                    407X        RC                      IS NOT IN LIST
041.115  043                    408X        INX     H
041.116  302 111 041            409X        JNE     $CSI5           IS NOT THIS ONE
```

```
041.121  072 252 040   410X          LDA     $CSLCTL
041.124  246           411X          ANA     M               CLEAR $CSLCTL BITS
041.125  043           412X          INX     H
041.126  256           413X          XRA     M               SET $CTLCTL BITS
041.127  062 252 040   414X          STA     $CSLCTL
041.132  365           415X          PUSH    PSW             SAVE BITS
041.133  056 155       416X          MVI     L,#$INBUF
041.135  065           417X          DCR     M               DECREMENT CHARACTER FROM BUFFER
041.136  361           418X          POP     PSW
041.137  360           419X          RP                      RETURN IF NOT CTL-A THROUGH CTL-D
                       420X
                       421X *        HAVE SPECIAL CONTROL. CALL USER.
                       422X
041.140  052 250 040   423X          LHLD    $CSIC
041.143  351           424X          PCHL                    CALL USER ROUTINE


041.144                426           END
ASSEMBLY COMPLETE
  426 STATEMENTS
    0 ERRORS DETECTED
22612 BYTES FREE
```

$CDIN	040106	158L	386						
$CDOUT	040111	161L	320						
$CISI	040114	164L	365						
$CISO	040117	167L	335	338	341	350			
$COSI	040122	170L	315						
$COSO	040125	173L							
$CSI1	041061	367	375L						
$CSI2	040350	309L	378						
$CSI3	041072	382L							
$CSI5	041111	404L	409						
$CSIB	040214	212L	403						
$CSIC	040250	232L	423						
$CSINT	041043	347	363L						
$CSIX	041055	368L							
$CSLCTL	040252	234L	297	410	414				
$CSLLEN	040254	236L							
$INBUF	040155	201L	204	238	253	344	375	416	
$INBUFL	000035	204E	376						
$PRSCL	040152	195L							
$PRSCL.	040367	195	334L						
$RCI.S	040310	265	267	270L					
$RCHAR	040144	191L							
$RCHAR.	040255	191	252L						
$RCHAR1	040261	254L	256						
$RCHAR2	040311	272L	279						
$RCHAR3	040325	273	281L						
$RCHARA	040307	269E							
$RET	040110	159L	232						
$TDIN	040130	176L							
$TDOUT	040133	179L							
$TSIN	040136	182L							
$TSOUT	040141	185L	336	339	342				
$WCHAR	040147	193L							
$WCHAR.	040332	193	296L						
$WCHAR1	040333	297L	300						
$WCHAR2	040354	303	315L	318					
.	000040	238S	239						
.UIVEC	040037	24E	346	348					
BELL	000007	23E	310						
CC.CTLA	000010	142E							
CC.CTLB	000020	143E							
CC.CTLC	000040	144E							
CC.CTLD	000100	145E							
CC.DMP	000002	141E	302						
CC.HLD	000001	140E	298						
COLNO	040253	235L							
CONFIG	333333	27E	149						
ENTRY	040100	149L							
IP.CDP	000372	119E	158						
IP.CIS	000373	121E	164						
IP.COS	000373	123E	170						
IP.TDP	000370	126E	176						
IP.TSP	000371	128E	182						
OP.CDP	000372	120E	161						
OP.CIS	000373	122E	167						
OP.COS	000373	124E	173						
OP.TDP	000370	127E	179						
OP.TSP	000371	129E	185						

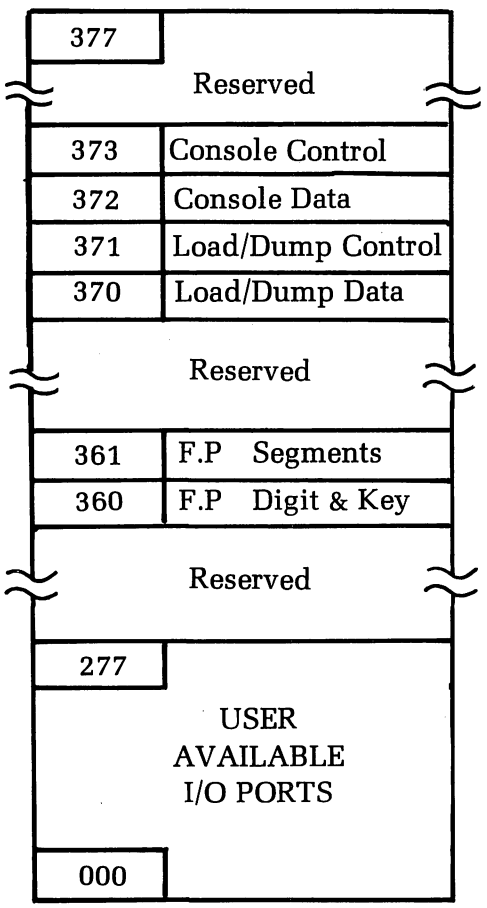
RESTART	222222	26E	150
START	111111	25E	
UCI.ER	000020	54E	349
UCI.IE	000002	56E	349
UCI.IR	000100	52E	337
UCI.RE	000004	55E	349
UCI.RO	000040	53E	
UCI.TE	000001	57E	349
UMI.16X	000002	47E	340
UMI.1B	000100	37E	340
UMI.1X	000001	46E	
UMI.2B	000300	39E	
UMI.64X	000003	48E	
UMI.HB	000200	38E	
UMI.L5	000000	42E	
UMI.L6	000004	43E	
UMI.L7	000010	44E	
UMI.L8	000014	45E	340
UMI.PA	000020	41E	
UMI.PE	000040	40E	
USR.FE	000040	61E	
USR.OE	000020	62E	
USR.PE	000010	63E	
USR.RXR	000002	65E	366
USR.TXE	000004	64E	
USR.TXR	000001	66E	316

31702 BYTES FREE

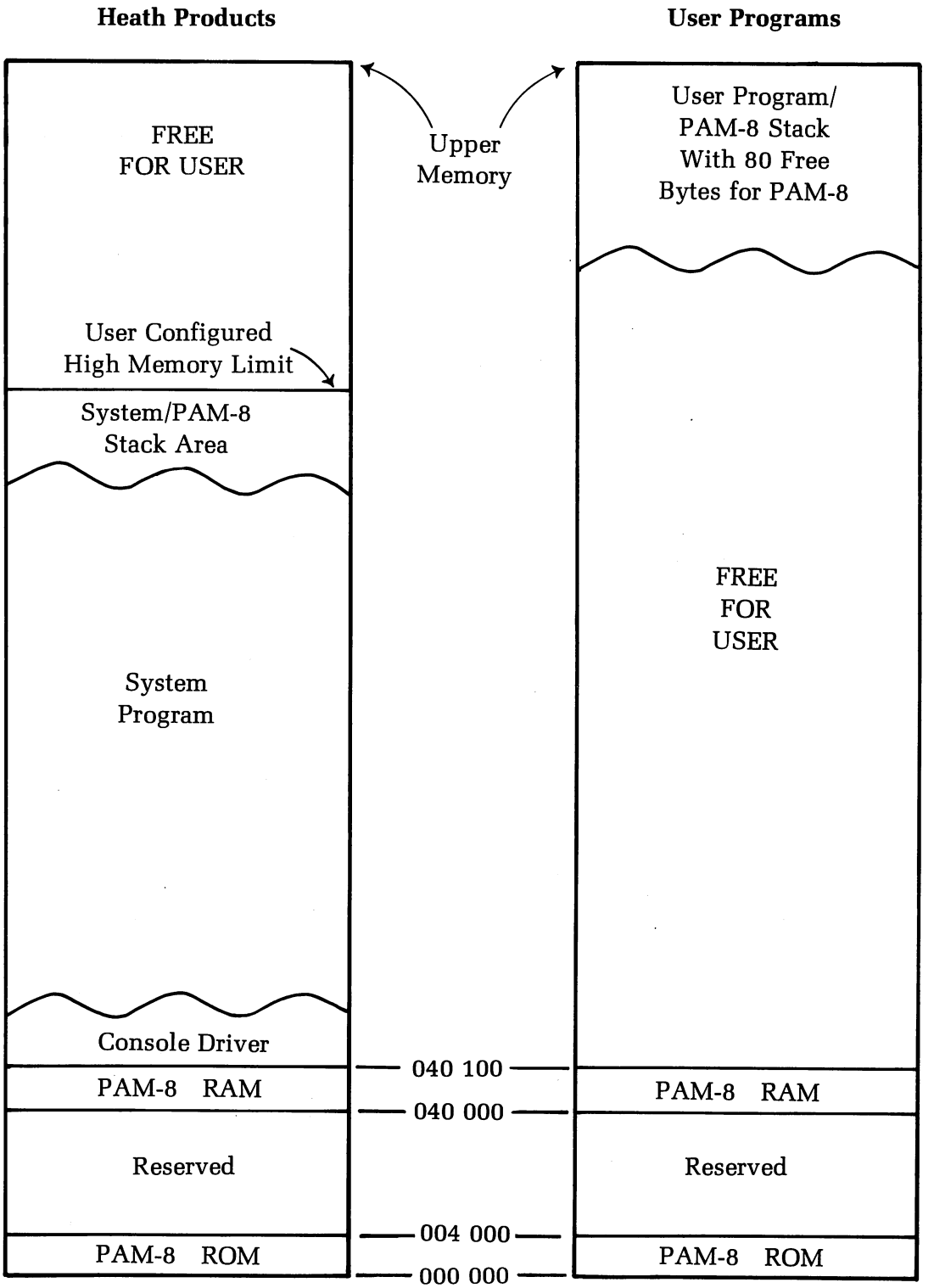
APPENDIX C

This appendix contains I/O and Memory Maps for the H8 software products.

I/O map



MEMORY MAP


Heath Products
User Programs
FREE
FOR USER
Upper
Memory
User Program/
PAM-8 Stack
With 80 Free
Bytes for PAM-8
User Configured
High Memory Limit
System/PAM-8
Stack Area
FREE
FOR
USER
System
Program
Console Driver
040 100
PAM-8 RAM
PAM-8 RAM
040 000
Reserved
Reserved
004 000
PAM-8 ROM
PAM-8 ROM
000 000

APPENDIX D

ASCII Characters

7-BIT OCTAL CODE	DECIMAL CODE	CHARACTER	DESCRIPTION
000	0	NUL	NULL, TAPE FEED, CONTROL SHIFT P.
001	1	SOH	START OF HEADING; ALSO SOM, START OF MESSAGE, CONTROL A,
002	2	STX	START OF TEXT; ALSO EOA, END OF ADDRESS, CONTROL B,
003	3	ETX	END OF TEXT: ALSO EOM, END OF MESSAGE CONTROL C,
004	4	EOT	END OF TRANSMISSION (END): CONTROL D,
005	5	ENQ	ENQUIRY; ALSO WRU, CONTROL E,
006	6	ACK	ACKNOWLEDGE. ALSO RU, CONTROL F.
007	7	BEL	RINGS THE BELL. CONTROL G.
010	8	BS	BACKSPACE: ALSO FEO, FORMAT EFFECTOR BACKSPACE SOME MACHINES, CONTROL H.
011	9	HT	HORIZONTAL TAB. CONTROL I
012	10	LF	LINE FEED (NEW LINE): ADVANCES PAPER TO NEXT LINE, DUPLICATED BY CONTROL J.
013	11	VT	VERTICAL TAB (VTAB). CONTROL K.
014	12	FF	FORM FEED TO TOP OF NEXT PAGE (PAGE). CONTROL L.
015	13	CR	CARRIAGE RETURN TO BEGINNING OF LINE. DUPLICATED BY CONTROL M.
016	14	SO	SHIFT OUT: CHANGES RIBBON COLOR TO RED. CONTROL N.
017	15	SI	SHIFT IN: CHANGES RIBBON COLOR TO BLACK. CONTROL O.
020	16	DLE	DATA LINK ESCAPE. CONTROL P (DCO).
021	17	DC1	DEVICE CONTROL 1, TURNS TRANSMITTER (READER) ON, CONTROL Q (XON).
022	18	DC2	DEVICE CONTROL 2, TURNS PUNCH OR AUXILIARY ON, CONTROL R (TAPE, AUX ON).
023	19	DC3	DEVICE CONTROL 3, TURNS TRANSMITTER (READER) OFF, CONTROL S (XOFF).
024	20	DC4	DEVICE CONTROL 4, TURNS PUNCH OR AUXILIARY OFF. CONTROL T (TAPE, AUX OFF).
025	21	NAK	NEGATIVE ACKNOWLEDGE: ALSO ERR. ERROR. CONTROL U.
026	22	SYN	SYNCHRONOUS IDLE (SYNC). CONTROL V.
027	34	ETB	END OF TRANSMISSION BLOCK: ALSO LEM. LOGICAL END OF MEDIUM. CONTROL W.

7-BIT OCTAL CODE	DECIMAL CODE	CHARACTER	DESCRIPTION
030	24	CAN	CANCEL (CANCL). CONTROL X.
031	25	EM	END OF MEDIUM. CONTROL Y.
032	26	SUB	SUBSTITUTE. CONTROL Z.
033	27	ESC	ESCAPE. PREFIX.
034	28	FS	FILE SEPARATOR. CONTROL SHIFT L.
035	29	GS	GROUP SEPARATOR. CONTROL SHIFT M.
036	30	RS	RECORD SEPARATOR. CONTROL SHIFT N.
037	31	US	UNIT SEPARATOR. CONTROL SHIFT O.
040	32	SP	SPACE.
041	33	!	
042	34	"	
043	35	#	
044	36	$	
045	37	%	
046	38	&	
047	39	'	ACUTE ACCENT OR APOSTROPHE.
050	40	(	
051	41	)	
052	42	*	
053	43	+	
054	44	,	
055	45	-	
056	46	.	
057	47	/	
060	48	0	
061	49	1	
062	50	2	
063	51	3	
064	52	4	
065	53	5	
066	54	6	
067	55	7	
070	56	8	
071	57	9	
072	58	:	
073	59	;	
074	60	<	
075	61	=	
076	62	>	
077	63	?	
100	64	@	
101	65	A	
102	66	B	
103	67	C	
104	68	D	
105	69	E	
106	70	F	
107	71	G	

7-BIT OCTAL CODE	DECIMAL CODE	CHARACTER	DESCRIPTION
110	72	H	
111	73	I	
112	74	J	
113	75	K	
114	76	L	
115	77	M	
116	78	N	
117	79	O	
120	80	P	
121	81	Q	
122	82	R	
123	83	S	
124	84	T	
125	85	U	
126	86	V	
127	87	W	
130	88	X	
131	89	Y	
132	90	Z	
133	91	[	SHIFT K
134	92	]	SHIFT L
135	93	↑	SHIFT M
136	94	←	SHIFT N
137	95		
140	96		ACCENT GRAVE.
141	97	a	
142	98	b	
143	99	c	
144	100	d	
145	101	e	
146	102	f	
147	103	g	
150	104	h	
151	105	i	
152	106	j	
153	107	k	
154	108	l	
155	109	m	
156	110	n	
157	111	o	
160	112	p	
161	113	q	
162	114	r	
163	115	s	
164	116	t	
165	117	u	
166	118	v	
167	119	w	

7-BIT OCTAL CODE	DECIMAL CODE	CHARACTER	DESCRIPTION
170	120	x	
171	121	y	
172	122	z	
173	123		
174	124		
175	125		THIS CODE GENERATED BY ALT MODE.
176	126		THIS CODE GENERATED BY ESC KEY (IF PRESENT)
177	127	DEL	DELETE, RUB OUT.

Appendix E

Decimal To Octal Tables

for 0 to 255_{10}

DECIMAL	OCTAL
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17
16	20
17	21
18	22
19	23
20	24
21	25
22	26
23	27
24	30
25	31
26	32
27	33
28	34
29	35
30	36
31	37
32	40
33	41
34	42
35	43
36	44

DECIMAL	OCTAL
37	45
38	46
39	47
40	50
41	51
42	52
43	53
44	54
45	55
46	56
47	57
48	60
49	61
50	62
51	63
52	64
53	65
54	66
55	67
56	70
57	71
58	72
59	73
60	74
61	75
62	76
63	77
64	100
65	101
66	102
67	103
68	104
69	105
70	106
71	107
72	110
73	111

DECIMAL	OCTAL
74	112
75	113
76	114
77	115
78	116
79	117
80	120
81	121
82	122
83	123
84	124
85	125
86	126
87	127
88	130
89	131
90	132
91	133
92	134
93	135
94	136
95	137
96	140
97	141
98	142
99	143
100	144
101	145
102	146
103	147
104	150
105	151
106	152
107	153
108	154
109	155
110	156

DECIMAL	OCTAL
111	157
112	160
113	161
114	162
115	163
116	164
117	165
118	166
119	167
120	170
121	171
122	172
123	173
124	174
125	175
126	176
127	177
128	200
129	201
130	202
131	203
132	204
133	205
134	206
135	207
136	210
137	211
138	212
139	213
140	214
141	215
142	216
143	217
144	220
145	221
146	222
147	223
148	224
149	225
150	226
151	227
152	230
153	231
154	232
155	233
156	234
157	235
158	236
159	237

DECIMAL	OCTAL
160	240
161	241
162	242
163	243
164	244
165	245
166	246
167	247
168	250
169	251
170	252
171	253
172	254
173	255
174	256
175	257
176	260
177	261
178	262
179	263
180	264
181	265
182	266
183	267
184	270
185	271
186	272
187	273
188	274
189	275
190	276
191	277
192	300
193	301
194	302
195	303
196	304
197	305
198	306
199	307
200	310
201	311
202	312
203	313
204	314
205	315
206	316
207	317
208	320

DECIMAL	OCTAL
209	321
210	322
211	323
212	324
213	325
214	326
215	327
216	330
217	331
218	332
219	333
220	334
221	335
222	336
223	337
224	340
225	341
226	342
227	343
228	344
229	345
230	346
231	347
232	350
233	351
234	352
235	353
236	354
237	355
238	356
239	357
240	360
241	361
242	362
243	363
244	364
245	365
246	366
247	367
248	370
249	371
250	372
251	373
252	374
253	375
254	376
255	377

APPENDIX F

Memory Table

Offset Octal and Decimal Boundaries

Hi Byte	Lo Byte	Decimal Boundary
A15.....A8	A7.....A0	
0 0 4	0 0 0	1024
0 2 0	0 0 0	4096
0 4 0	0 0 0	8192
0 6 0	0 0 0	12288
1 0 0	0 0 0	16384
1 2 0	0 0 0	20480
1 4 0	0 0 0	24576
1 6 0	0 0 0	28672
2 0 0	0 0 0	32768
2 2 0	0 0 0	36864
2 4 0	0 0 0	40960
2 6 0	0 0 0	45056
3 0 0	0 0 0	49152
3 2 0	0 0 0	53248
3 4 0	0 0 0	57344
3 6 0	0 0 0	61440
3 7 7	3 7 7	65535*

For example, if you have 12K bytes in an H8, the lower boundary is at 8192, or 040 000 offset octal. The upper boundary is at 8K + 12K = 20K (20480), or 120 000 Octal.

*NOTE: 65,535 is the last location in a memory addressed by 16 bits.

INDEX

ASR Console, 0-22

Basic Program Format, 0-15

Checksum, 0-12
Compressed, Text Format, 0-15
Configured Tape, 0-19
Console Debugger (BUG-8), 0-8
Console Driver, 0-23
Copying Tapes, 0-22
CRC-16, 0-12
Control Characters, 0-23

Data Formats, 0-14
Distribution Tapes, 0-19
Displays, 0-15

Entry Point, 0-23

File, 0-12
Front Panel Displays, 0-15
Front Panel Entry Points, 0-23

Heath Assembly Language (HASL-8), 0-10
Heath Text Editor (TED-8), 0-8

Installation, 0-19 ff.,

Label Record, 0-14
Loading, 0-21

Magnetic Tape, 0-18
Memory Image Format, 0-14

110-Baud Console, 0-22

Patch Installation 0-22
Pad Characters, 0-20
Panel Monitor (PAM-8), 0-7
Paper Tape, 0-18

ROM, 0-7
Record, 0-12
Record Structure, 0-13

Software Installation, 0-19 ff.,
Software Distribution Tape, 0-19
Software Problems, 0-24
Software Version Code, 0-24

Tape Files, 0-12
Tape Format, 0-12