CP/M® VERSION 2.2.04 MANUAL

Volume II

HEATH | ZENITH data systems

Printed in the
United States of America

595-2816-01

NOTICE

This software *is licensed (not sold). It is licensed to sublicensees, including end-users, without either express or implied warranties of any kind on an "as is" basis.*

The owner and distributors make no express or implied warranties to sublicensees, including end-users, with regard to this software, including merchantability, fitness for any purpose or non-infringement of patents, copyrights or other proprietary rights of others. Neither of them shall have any liability or responsibility to sublicensees, including end-users, for damages of any kind, including special, indirect or consequential damages, arising out of or resulting from any program, services or materials made available hereunder or the use or modification thereof.

This publication *could contain technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of this publication.*

Technical consultation *is available for any problems you encounter in verifying the proper operation of this product. Sorry, but we are not able to evaluate or assist in the debugging of any programs you may develop. For technical assistance, write:*

Zenith Data Systems Corporation
Software Consultation
Hilltop Road
St. Joseph, Michigan 49085

or call:

(616) 982-3884 Application Software/SoftStuff Products
(616) 982-3860 Operating Systems/Languages/Utilities

Consultation is available from 8:00 AM to 7:30 PM (Eastern Time Zone) on regular business days.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in paragraph (b) (3) (B) of the Rights in Technical Data and Computer Software clause in DAR 7-104.9(a). Contractor/Manufacturer is Zenith Data Systems Corporation of Hilltop Road, St. Joseph, Michigan 49085.

Trademarks and Copyrights

CP/M is a registered trademark of Digital Research Corporation
Digital Research is a trademark of Digital Research Corporation

HEATH COMPANY
BENTON HARBOR, MICHIGAN 49022

ZENITH DATA SYSTEMS CORPORATION
ST. JOSEPH, MICHIGAN 49085

The CP/M Reference Guide

This Reference Guide contains detailed descriptions of each of the commands accompanying Heath/Zenith CP/M release 2.2.04. The guide deals with all of the resident commands and transient commands, presenting each command in alphabetical order. The text explains the following aspects of commands (where applicable):

- Function
- Invocation
- Options or Parameters
- Common applications
- Error message explanations

The text on each command is divided into numbered sections and subsections concerning specific aspects of the command. Examples are provided where appropriate.

The Reference Guide features several examples of user interaction with the microcomputer. In these examples, displays presented by the microcomputer console will be represented by the following typestyle:

```
    THIS TYPESTYLE represents console displays

                    (0123456789#$*?: =.A>)
```

User input will be represented by boldface type, as shown:

BOLDFACE TYPE represents the things the user types

(0123456789#$*?: = [.])

Text that instructs you to press a key that is labelled with more than one character (such as the RETURN key of the CTRL key) will specify the label of this key in dark (boldface), slanted (italicized), capitalized characters, as shown:

Press ***RETURN*** Hold down ***CTRL***

An example of user interaction with the terminal might appear as follows:

A>**STAT {drive}:{file name}** ***RETURN***

Where "A>" is a console display (issued by CP/M);

where **STAT** is a command entry typed through the console by the user;

where {**drive**} or any other description enclosed in curved braces is an entry that the user should supply;

where {**file name**} is the primary name of a file within the specified drive; and

where ***RETURN*** is a key label of a key pressed by the user.

The "where" statements that immediately follow a sample screen display or sample entry describe the important components of the display or entry.

These examples appear throughout the Reference Guide to demonstrate how you should communicate with the microcomputer.

ASM

The Utility that Creates an Intel Hexadecimal File and a Printout File from an Assembly Language Program File

> This text on the ASM utility assumes that you are familiar with assembly language programming. If you are not familiar with assembly language programming, we recommend that you read the Heathkit Continuing Education Course in Assembly Language Programming before using the ASM utility. Information on your computer's architecture, opcodes, and other processor features is available in your computer manual.

The ASM utility reads assembly language source files (files with the "ASM" extension) from a disk and translates them into two output files (1):

- An output file with the "HEX" extension, containing 8080 machine code in Intel hexadecimal form (2). This file can be loaded into the computer (using the LOAD utility) and executed under the CP/M Operating System.

- An output file with the "PRN" extension, containing ASCII characters. This file can be printed out, displayed on a console screen, or not produced (1.1). It includes the same mnemonics as the "ASM" file (3, 4, and 5), plus the hexadecimal address of each source line and signal characters for errors that occurred during the assembly (6.2).

1. ASM INVOCATION

The ASM utility is invoked in response to the system prompt by entering a command in the following form:

A>**ASM {file name}.{asm}{hex}{prn}** ***RETURN***

Where {**file name**} is the primary name of an assembly language source file whose extension is assumed to be "ASM". This file will be translated into the two output files by ASM;

where {**asm**} represents a one-letter parameter entered to specify the drive in which the assembly language source file is located;

where {**hex**} represents a one-letter parameter entered to specify the drive to which the "HEX" output file will be written; and

where {**prn**} represents a one-letter parameter entered to specify the drive to which the "PRN" output file will be written.

CP/M assumes that the source file has the extension "ASM". Therefore, the "ASM" extension does not have to be entered in the invocation command.

1.1 Assembler Parameters

Three single-letter parameters can be entered immediately after the period (.) in the invocation command line, as explained below:

A>**ASM PROGRAM.ABC** ***RETURN***

The first parameter letter specifies the drive which contains the "ASM" source file which is to be assembled.

The second parameter letter specifies the drive that will receive the "HEX" file produced by the assembly. If a **Z** is entered in the place of a drive letter, the assembler will not produce a "HEX" file.

The third parameter letter specifies the drive that will receive the "PRN" file produced by the assembly. If a **Z** is entered in place of a drive letter, the assembler will not produce a "PRN" file. If an **X** is entered in place of a drive letter, the listing will be displayed at the console instead of being written to a disk as a "PRN" file.

If you omit all of the parameter letters (as in the command **ASM PROGRAM** ***RETURN***), then ASM will try to read the source file from the default drive and to write both the HEX file and the PRN file to the default drive.

If you omit the the last two parameter letters (as in the command **ASM PROGRAM.A** ***RETURN***), then ASM will try to read the source file from the specified drive, to write the HEX file to the default drive, and to write the PRN file to the default drive.

If you omit only the final parameter letter (as in the command **ASM PROGRAM.AB** ***RETURN***), then ASM will try to read the source file from the specified drive, to write the HEX file to the specified drive, and to write the PRN file to the default drive.

If you leave a space for the first parameter letter and then specify a letter for any remaining parameters, (as in the command **ASM PROGRAM. BC** ***RETURN*** or **ASM PROGRAM. C** ***RETURN***), then ASM will ignore the parameters that follow the space. Thus ASM will try to read the source file from the default drive and to write both the HEX file and the PRN file to the default drive.

1.2 Example ASM Command Lines

The following examples and explanations illustrate the different forms of the ASM command line and the results they produce.

```
A>ASM PROGRAM   RETURN
```

Source file PROGRAM.ASM is read from the default drive and assembled. Hex file PROGRAM.HEX and print file PROGRAM.PRN are written to the default drive.

```
A>ASM PROGRAM.ABB   RETURN
```

Source file PROGRAM.ASM is read from drive A and assembled. Output files PROGRAM.HEX and PROGRAM.PRN are both written to drive B.

```
A>ASM PROGRAM.BAX   RETURN
```

Source file PROGRAM.ASM is read from drive B and assembled. Hex file PROGRAM.HEX is written to disk A, and print file PROGRAM.PRN is displayed on the console.

```
A>ASM PROGRAM.BZZ   RETURN
```

Source file PROGRAM.ASM is read from drive B and assembled. Neither output file is created. Such an entry is used to check for source file errors.

2. ASM EXECUTION

When invoked, ASM identifies itself with the following display:

```
CP/M ASSEMBLER - VER 2.0
```

Where "2.0" indicates the version number of the ASM utility.

If you specified (in a parameter) that the PRN file be displayed rather than stored, then this display will now appear on the console. This display can be suspended by entering a ***CTRL-S***, and resumed by entering another ***CTRL-S*** or any other character. The display can only be aborted by resetting the computer, which would abort the entire assembly operation as well. The HEX file is not assembled until the PRN file has been fully displayed.

After an assembly operation is completed, ASM displays a message in the form of the following example:

```
0FD4
02AH USE FACTOR
END OF ASSEMBLY

A>
```

where "0FD4" (or any number appearing in this position) is the next free address (in hexadecimal) after the end of the HEX file produced by the assembly; and

where "02AH" (or any number appearing in this position) represents the percentage of free system memory used for ASM's symbol table. This percentage is expressed in hexadecimal values, with 000H being 0 percent and 0FFFH being 100 percent.

3. FORM FOR SOURCE FILE STATEMENTS

Assembly language source files must be composed of program statements in the following form:

line# **label** **operator** **operand** **;comment**

Where the **line#** field is an optional integer value representing the source program line number. These numbers are only for the convenience of the user and are ignored by the assembler when present;

where the **label** field is optional, except when required by particular statement types. It consists of a maximum of 16 alphanumeric characters (letters and numbers). The first character must be a letter. Labels can be freely used by the programmer to identify elements such as program steps.

A single label should not contain any spaces, because ASM interprets a space as the end of the label field. However, the dollar sign ($) can be used to improve the readability of labels that contain more than one word. ASM will ignore the dollar sign, and consider the other characters in the label as if they were in a continuous string. The label "DATES$OF$BIRTH" is somewhat easier to read than the label "DATESOFBIRTH". ASM will treat the two labels as being identical.

Labels may also be followed by a colon (:) to maintain compatibility with some other assemblers. The colon will not become part of the label and will be ignored by ASM.

where **operator** is either a pseudo operation, or an assembler directive or an 8080 Machine Opcode (see your microcomputer manual);

where the **operand**, in general, contains an expression formed from constants and labels together with arithmetic and logical operations on these elements; and

where a **comment** consists of any characters following a semicolon (;) until an end-of-line is encountered. An asterisk (*) used as the first character on a line will also indicate a comment. All comments are read and listed in the print file but otherwise ignored by ASM.

Any or all of these fields may be present in a statement. Each assembly language statement is terminated by a carriage return and a line feed (both of which are inserted when the RETURN key is pressed with many text editors), or with the exclamation point character (which is recognized by the ASM utility as an "end-of-line" character). If the exclamation point (!) is used to signify the end of an assembly language statement, other statements can be entered on the same physical line.

4. FORMING THE OPERAND

In order to completely describe the operation codes and pseudo operations, it is first necessary to present the form of the operand since it is used in nearly all statements.

Expressions consist of simple operands (labels, constants, and reserved words) combined in properly formed subexpressions by arithmetic and logical operators. Expression computation is carried out by the assembler as the assembly proceeds. Each expression must produce a 16-bit value during the assembly.

The number of significant digits in the result must not exceed the intended use. If an expression is to be used in a byte move-immediate instruction, then the most significant 8 bits of the expression must be zero. The restrictions on the expression significance are given with the individual instructions.

4.1 Labels

A label is an identifier which occurs on a particular statement. Generally, the label is given a value determined by the type of statement which it precedes.

If the label occurs on a statement which generates machine code or reserves memory space (such as a MOV instruction, or a DS pseudo operation), the label is given the value of the program address that it labels. If the label precedes an EQU or SET instruction, the label is given the value resulting from evaluating the operand. Except for the SET statement, an identifier can only label one statement.

When a label appears in the operand, its value is substituted by the assembler. This value can then be combined with other operands and operators to form an operand for a particular instruction.

4.2 Numeric Constants

A numeric constant is a 16-bit value in one of several bases. The base, or radix, of the constant is denoted by a trailing radix indicator. The radix indicators are:

B	Binary constant (base 2)
O	Octal constant (base 8)
Q	Octal constant (base 8)
D	Decimal constant (base 10)
H	Hexadecimal constant (base 16)

"Q" is an alternate radix indicator for octal numbers since the letter O is easily confused with the digit zero (0).

A numeric constant which does not terminate with a radix indicator is assumed to be a decimal constant.

A constant is composed of a sequence of digits, sometimes followed by a radix indicator. The digits are in the appropriate range for the radix. Binary constants must be composed of 0 and 1 digits, octal constants can contain digits in the range 0–7, while decimal constants contain decimal digits 0–9. Hexadecimal constants contain decimal digits and hexadecimal digits A (10D), B (11D), C (12D), D (13D), E(14D), and F (15D). The leading digit of a hexadecimal constant must be a decimal digit in order to avoid confusing a hexadecimal constant must be a decimal digit in order to avoid confusing a hexadecimal constant with an identifier. If a number begins with a hexadecimal digit (A–F), precede this digit with a zero.

A constant composed in this manner must correspond to a binary number that can be contained within a 16-bit counter, otherwise it is truncated on the right by the assembler. As with a label, dollar signs ($) can be imbedded in a constant to improve readability. Finally, the radix indicator is translated to upper case if a lower case letter is encountered. The following are all valid instances of numeric constants.

1234	1234D	1100B	1111$0000$1111$0000B
1234H	0FFEH	33770	33$77$22Q
3377o	0fe3h	1234d	0ffffh

4.3 Reserved Words

There are several reserved character sequences with predefined meanings in the operand field of a statement. Names of 8080 registers are given below. When ASM encounters one of these register names, the numeric value shown in the following table is produced:

REGISTER NAME	REGISTER VALUE
A	7
B	0
C	1
D	2
E	3
H	4
L	5
M	6
SP	6
PSW	6

Lower case names have the same value as their upper case equivalents.

When the symbol "$" occurs in the operand field (not imbedded within identifiers and numeric constants) its value becomes the address of the next instruction to generate, not including the instruction contained within the current logical line.

4.4 String Constants

String constants, sequences of ASCII characters, are represented by enclosing the characters within apostrophe symbols ('). All strings must be fully contained within the current physical line (thus allowing "!" symbols within strings) and may not exceed 64 characters in length. The apostrophe character can be included within a string by representing it as a double apostrophe (press the apostrophe key two consecutive times), which ASM interprets as a single apostrophe.

In most cases, the string length is restricted to either one or two characters (the DB pseudo operation is an exception). If the string consists of one character, it becomes an 8-bit value. If the string consists of two characters, it becomes a 16-bit value. Two character strings become a 16-bit constant, with the second character as the low order byte, and the first character as the high order byte.

The value of a character is its corresponding ASCII code. There is no case translation within strings. Therefore, both upper and lower case characters can be represented. Only ASCII characters that print are allowed within strings. Valid strings are:

’A’ ‘AB’ ‘ab’ ’c’
’She said “Hello” to me.’
’I said “Hello” to her.’

4.5 Arithmetic and Logical Operators

The operands described previously can be combined in normal algebraic notation using any combination of properly formed operands, operators, and expressions enclosed in parentheses. The operators recognized in the operand field are shown by the following list:

a + b	Unsigned arithmetic sum of a and b
a – b	Unsigned arithmetic difference between a and b
+ b	Unary plus (produces b)
– b)	Unary minus (identical to 0 – b)
a * b	Unsigned magnitude multiplication of a and b
a / b	Unsigned magnitude division of a by b
a MOD b	Remainder after a / b NOT b Logical inverse of b (0’s become 1’s, 1’s become 0’s), where b is considered a 16-bit value
a AND b	Bit-by-bit logical and of a and b
a OR b	Bit-by-bit logical or of a and b
a XOR b	Bit-by-bit logical exclusive or of a and b
a SHL b	The value which results from shifting a to the left by an amount b, with zero fill
a SHR b	The value which results from shifting a to the right by an amount b, with zero fill

In each case, a and b represent simple operands (labels, numeric constants, reserved words, and one or two character strings), or subexpressions fully enclosed in parentheses, such as:

```
10+20     10h+37Q     L1/3     (L2+4)     SHR 3
('a' and 5fh)  +  '0'  ('B' +B)  OR  (PSW+M)
(1+(2+c))  shr  (A-(B+1))
```

All computations are performed at assembly time as 16-bit unsigned operations. Thus, −1 is computed as 0-1 which results in the value 0ffffh (i.e., all 1's). The resulting expression must fit the operation code in which it is used. If, the expression is used in an ADI (add immediate) instruction, then the high order eight bits of the expression must be zero. As a result, the operation "ADI −1" produces an error message (−1 becomes 0ffffh, which cannot be represented as an 8-bit value), while "ADI (−1) AND 0FFH" is accepted by the assembler since the "AND" operation zeros the high order bits of the expression.

4.6 Precedence of Operators

ASM assumes that operators have a relative precedence of application which allows the programmer to write expressions without nested levels of parenthesis. The resulting expression has assumed parentheses which are defined by the relative precedence.

The order of application of operators in expressions not enclosed in parentheses is listed below. Operators listed first have highest precedence (they are applied first in an expression), while operators listed last have lowest precedence. Operators listed on the same line have equal precedence, and are applied from left to right as they are encountered in an expression such as the following:

```
* / MOD SHL SHR
      − +
      NOT
      AND
     OR XOR
```

In the following examples, the expressions shown to the left are interpreted by ASM as the expressions enclosed in parentheses shown to the right:

APPEARANCE IN PROGRAM	AS INTERPRETED BY ASM UTILITY
a * b + c	(a * b) + c
a + b * c	a + (b * c)
a MOD b * c SHL d	((a MOD b) * c) SHL d
a OR b AND NOT c + d SHL e	a OR (b AND (NOT(c + (d SHL e))))

Balanced subexpressions enclosed in parentheses can always be used to override the assumed parentheses. The last expression in the preceding example could be rewritten to force application of operators in a different order, as shown:

(a OR b) AND (NOT c) + d SHL e

This expression is interpreted by ASM as the following expression with assumed parentheses:

(a OR b) AND ((NOT c) + (d SHL e))

An expression not enclosed in parentheses is only correct if the expression resulting from inserting the assumed parentheses is correctly formed.

5. ASSEMBLER DIRECTIVES

Assembler directives are used to set labels to specific values during the assembly, perform conditional assembly, define storage areas, and specify starting addresses in the program. Each assembler directive is denoted by a "pseudo operation" that appears in the operation field of the line. These pseudo operations are acceptable to ASM:

Pseudo Operation	Function	Text Subsection
ORG	Set the program or data origin	(5.1)
END	End program, optional start address	(5.2)
EQU	Numeric "equate"	(5.3)
SET	Numeric "set"	(5.4)
IF	Begin conditional assembly	(5.5)
ENDIF	End of conditional assembly	(5.6)
DB	Define data bytes	(5.7)
DW	Define data words	(5.8)
DS	Define data storage area	(5.9)

5.1 The ORG Directive

The ORG Directive takes the following form:

Label ORG expression

Where "label" is an optional program label; and

where "expression" is a 16-bit expression, consisting of operands which are defined previous to the ORG statement.

The assembler begins machine code generation at the location specified in the expression. There can be any number of ORG statements within a particular program. There are no checks to ensure that the programmer is not defining overlapping memory areas. Most programs written for the CP/M operating system begin with an ORG statement of:

ORG 100H

Machine code generation begins at the base of the CP/M transient program area (hexadecimal address 100H). If a label is specified in the ORG statement, then the label is given the value of the expression. (This label can then be used in the operand field of other statements to represent this expression.)

5.2 The END Directive

The END Directive is optional in an assembly language program. If it is present, it should be the last statement. All subsequent statements will be ignored by ASM.

The two forms of the END directive are:

label END

Label END expression

Where the "label" field is optional; and

where the "expression" field is the program starting address.

If the first form is used, the assembly process stops, and the default starting address of the program is taken as 0000.

If the second form is used, the expression in the statement becomes the program starting address. (This starting address is included in the last record of the Intel formatted machine code "HEX" file that results from the assembly.)

Most CP/M assembly language programs should end with the statement:

END 100H

resulting in the default starting address of 100H (which is the beginning of the Transient Program Area).

NOTE: When an assembled .HEX file is loaded using the CP/M LOAD utility (included with your CP/M distribution software), you do not need to use the expression "100H" with the END directive. LOAD produces .COM files that automatically execute from address 100H regardless of the expression used in an END directive that might exist at the end of a program.

5.3 The EQU Directive

The EQU Directive is used to set up synonyms for particular numeric values. It takes the following form:

label EQU expression

Where the "label" field must be present, and must not label any other statement; and

where the "expression" field is assigned to an identifier given in the label field. The identifier is usually a name which describes the value in a more human-oriented manner. This name is used throughout the program to add parameters to certain functions.

Suppose, for example, that data received from a Teletype appears at a particular input port, and data is sent to the Teletype through the next output port in sequence. The following series of equate statements could be used to define these ports for a particular hardware environment:

```
TTYBASE  EQU  10H         ;BASE PORT NUMBER FOR TTY
TTYIN    EQU  TTYBASE     ;TTY DATA IN
TTYOUT   EQU  TTYBASE+1   ;TTY DATA OUT
```

At a later point in the program, the statements which access the Teletype could appear as shown:

```
IN       TTYIN      ;READ TTY DATA TO REG-A
...
OUT      TTYOUT     ;WRITE DATA TO TTY FROM REG-A
```

This directive makes the program more readable than if the absolute I/O ports had been used.

If the hardware environment is redefined to start the Teletype communications ports at 7FH instead of 10H, the first statement need only be changed to:

```
TTYBASE  EQU  7FH   ;BASE PORT NUMBER FOR TTY
```

Then the program can be reassembled without changing any other statements.

5.4 The SET Directive

The SET Directive takes the following form:

label SET expression

Where the "label" field must be present, and may appear in other SET statements throughout the program; and

where the "expression" field is evaluated by ASM, and becomes the current value associated with the label.

The EQU Directive defines a label with a single value while the SET Directive defines a value which is valid from the current SET statement to the point where the label appears in the next SET statement. The use of the SET is similar to the EQU statement, but SET is most often used in controlling conditional assembly.

5.5 The IF and 5.6 ENDIF Directives

The IF and ENDIF Directives define a range of assembly language statements which are included or excluded during the assembly process. The form is:

```
IF expression
statement#1
statement#2
...
statement#n
ENDIF
```

When ASM encounters the IF statement, it evaluates the expression following the IF. (All operands in the expression must be defined ahead of the IF statement.) If the expression evaluates to a value where the low order bit (bit 0) is one, then statement #1 through statement #n are assembled. If the expression evaluates to a value where the low order bit (bit 0) of the expression is zero, then the statements are listed but not assembled.

Conditional assembly is often used to write a single "generic" program that includes a number of possible run-time environments, with only a few specific portions of the program selected for any particular assembly. The following program segments, might be part of a program that communicates with either a Teletype console or a CRT console by selecting a particular value for TTY before the assembly begins:

```
TRUE      EQU 0FFFFH        ;DEFINE VALUE OF TRUE
FALSE     EQU NOT TRUE      ;DEFINE VALUE OF FALSE
;
TTY       EQU TRUE          ;TRUE IF TTY, FALSE IF CRT
;
TTYBASE   EQU 10H           ;BASE OF TTY I/O PORTS
CRTBASE   EQU 20H           ;BASE OF CRT I/O PORTS
          IF TTY            ;ASSEMBLE RELATIVE TO TTYBASE
CONTIN    EQU TTYBASE       ;CONSOLE INPUT
CONOUT    EQU TTYBASE+1     ;CONSOLE OUTPUT
          ENDIF
;
          IF NOT TTY        ;ASSEMBLE RELATIVE TO CRTBASE
CONTIN    EQU CRTBASE       ;CONSOLE INPUT
CONOUT    EQU CRTBASE+1      ;CONSOLE OUTPUT
          ENDIF
          ...
          IN CONTIN         ;READ CONSOLE DATA
          ...
          OUT CONOUT        ;WRITE CONSOLE DATA
```

In this case, the program would assemble for an environment where a Teletype is connected, based at port 10H. The statement defining TTY could be changed to:

```
    TTY       EQU FALSE
```

and, in this case, the program would assemble for a CRT based at port 20H.

5.7 The DB Directive

The DB Directive allows the programmer to define initial storage areas in single-byte format. The statement appears in the form:

```
label DB e#1,e#2, ...,e#n
```

Where "e#1" through "e#n" are either:

- Expressions which evaluate to 8-bit values (the high order eight bits must be zero), or
- ASCII strings of fewer than 65 characters.

There is no practical restriction on the number of expressions that can be included on a single source line. The expressions are evaluated and placed sequentially into the machine code file following the last program address generated by ASM. String characters are similarly placed into memory, starting with the first character and ending with the last character.

Strings containing more than two characters cannot be used as operands in more complicated expressions. They must stand alone between the commas. ASCII characters are always placed in memory with the parity bit reset (to 0). There is no translation from lower to upper case with strings. The optional label can be used to reference the data area throughout the remainder of the program. Examples of valid DB statements are:

```
data:     DB 0,1,2,3,4,5
          DB data and 0ffh,5,377Q,1 + 2 + 3 + 4
signon:   DB 'please type your name',cr,lf,0
          DB 'AB' SHR 8, 'C', 'DE' and 7FH
```

5.8 The DW Directive

The DW Directive lets you define initial storage areas in double-byte words. The statement appears in the form:

```
label    DW    e#1,    e#2,    ...,    e#n
```

Where "e#1" through "e#n" are expressions which ASM evaluates in 16-bit results.

ASCII strings are limited to one or two characters. Data storage is consistent with the 8080 processor. The least significant byte of the expression is stored first in memory, followed by the most significant byte. The following are examples of DW usage:

```
doub:    DW 0ffefh,doub+4,signon-$,255+255
         DW 'a', 5, 'ab', 'CD', 6shl 8 or 11b
```

5.9 The DS Directive

The DS Directive is used to reserve an area of uninitialized memory. This directive takes the following form:

```
label    DS   expression
```

Where the label is optional.

The assembler begins subsequent code generation after the area reserved by the DS. The DS statement given previously has exactly the same effect as the following statement:

```
label:   EQU  $              ;LABEL VALUE IS CURRENT CODE LOCATION
         ORG  $+expression   ;MOVE PAST RESERVED AREA
```

6. ASM ERROR MESSAGES

The ASM utility has two kinds of error messages. One signals problems that occur during ASM file manipulation (6.1), and another signals errors in the assembly language source file (6.2).

6.1 File Manipulation Error Messages

The following error messages are displayed when ASM encounters difficulty in manipulating the source and/or output files involved in operation of the ASM utility:

```
NO SOURCE FILE PRESENT
```

EXPLANATION: The file specified in the ASM command does not exist on the accessed disk.

```
NO DIRECTORY SPACE
```

EXPLANATION: The disk directory is full. Unnecessary files should be erased, and the operation attempted again.

```
SOURCE FILE NAME ERROR
```

EXPLANATION: Improperly formed ASM file name (such as it is specified with "?" fields).

```
SOURCE FILE READ ERROR
```

EXPLANATION: The source File cannot be read properly by the assembler. The TYPE command should be used to determine the location of the unreadable code.

```
OUTPUT FILE WRITE ERROR
```

EXPLANATION: Output files cannot be written properly. The most likely cause is a full disk. Unnecessary files should be erased, and the operation attempted again.

```
CANNOT CLOSE FILES
```

EXPLANATION: Output files cannot be closed. Disk should be checked for write protection.

```
SYMBOL TABLE OVERFLOW
```

EXPLANATION: The symbol table has exceeded memory capacity.

6.2 Assembly Program Error Messages

When errors occur within the assembly language program, they are listed as single character flags in the left-most position of the source listing. The line in error is also echoed in a terminal display so that the source listing need not be examined to determine if errors are present. The error codes are:

D Data error: Element in data statement cannot be placed in the specified data area.

E Expression error: Expression is badly formed and cannot be computed at assembly time.

L Label error: Label cannot appear in this context (may be duplicated label).

N Not implemented: Features which will appear in future ASM versions (e.g., macros) are recognized, but flagged in this version.

O Overflow: Expression is too complicated (i.e., too many pending operators) to compute, simplify it.

P Phase error: Label does not have the same value on two subsequent passes through the program.

R Register error: The value specified as a register is not compatible with the operation code.

S Syntax error: The program included a character string that does not comprise a valid label, operator, or operand.

U Undefined symbol: The program made a reference to a label or variable that had not been defined earlier in the program.

V Value error: Operand encountered in expression is improperly formed.

ASSIGN

The Utility that Assigns Winchester Disk Partitions to Drive Names

(Used with H/Z-67 Drive Model)

The ASSIGN utility enables you to access software and data from Winchester Disk partitions by assigning these partitions to drive names (4), or by changing assignments (5). ASSIGN can also identify all of the partitions on a Winchester Disk and identify which of these are currently accessible through a drive name.

1 WINCHESTER DISK PARTITIONS

Partitions are independent logical devices that are established on a Winchester Disk using the PART program. (For a detailed explanation of the PART program, refer to the documentation accompanying the H/Z-67 disk drive hardware.)

The Winchester Disk can be divided into as many as 63 partitions to accommodate separate groups of data files and operating systems. Two of these partitions can be made accessible at one time by assigning them to a drive name with the ASSIGN utility.

A Winchester Disk partition can be identified by the following:

- a partition name of 16 or fewer characters;
- a semicolon (;); and
- an occurrence number ranging from 0 to 62.

The semicolon and occurrence number are optional. When not used, the occurrence number zero is assumed.

Partitions can bear the same partition name as long as they have different occurrence numbers.

2 INQUIRY OF AVAILABLE PARTITIONS

To find out the names of **all** of the partitions on the Winchester Disk, enter the following command:

A>**ASSIGN ?** ***RETURN***

ASSIGN will respond with a display in the following form:

```
PARTITION NAME        OCCUR       SIZE

--------------        -----      ------
CPM                    ;  0       3200k
CPM                    ;  1       2860k
CPM                    ;  2       2020k
HDOS                   ;  0       1600k
```

Where the letters listed beneath "PARTITION NAME" are the names that were designated for each partition when the PART utility was run;

where the numbers listed beneath "OCCUR" are the occurrence numbers that help you to distinguish between different partitions that have the same partition name (such as "CPM;0" and "CPM;1); and

where the numbers listed beneath "SIZE" show the total file capacity (in kilobytes) of the partition named to the left. If a listed partition already contains files, some or all of this space is already occupied. (To determine the amount of **empty** space on a partition, assign the partition to a drive and then run the STAT utility.

NOTE: The partitions listed in the example shown above may or may not be assigned.

3 INQUIRY OF CURRENT ASSIGNMENTS

To find out the names of the partitions on the Winchester Disk that are **currently** accessible through a drive name, enter the following command:

```
A>ASSIGN RETURN
```

The ASSIGN utility will display equations to indicate which drive names can be used to access which partition names, as in the following example:

```
A:  =  CPM; 1
B:  =  HDOS
```

If you boot up with a Winchester Disk partition, that partition is automatically assigned to drive A. If you wish to access an additional partition, you can assign the additional partition to drive B using the ASSIGN utility.

If you boot up with the floppy disk drive in the H/Z-67, that floppy disk drive is automatically drive A. You can then assign partitions to drives B and C using the ASSIGN utility.

4 ASSIGNING PARTITIONS TO DRIVE NAMES

To assign a partition to a drive name, enter a command in the following form:

```
A>ASSIGN {drive}:={partition name};{occurrence number} RETURN
```

Where {**drive**} is the letter for the drive;

where {**partition name**} is a name of 16 or fewer characters (characters that are valid in CP/M file names can be used); and

where {**occurrence number**} is a one-digit or two-digit number from 0 to 62, preceded by a semicolon. If this number is omitted, 0 will be assumed.

If you booted up with the floppy disk drive slot of the Z67, drive A would be assigned that floppy drive, but drive B and drive C would initially be unassigned and thus inaccessible.

Assuming that your partitions were named as those in Section 2, you could assign drive B by entering the command:

A>**ASSIGN B: = CPM;1** ***RETURN***

and assign drive C by entering the command:

A>**ASSIGN C: = HDOS** ***RETURN***

NOTE: The occurrence number of the "HDOS" partition can be omitted from this command because the occurrence number of this partition is 0 and ASSIGN assumes an occurrence number of 0 with any partition for which you do not specify the occurrence number.

5 CHANGING AN ASSIGNMENT

If a drive has already been assigned a particular partition and you wish to assign a different partition to this drive, enter a command in the following form:

A>**ASSIGN {drive}: = {new partition}** ***RETURN***

Where {**drive**} is the letter for the drive; and

where {**new partition**} is the partition name and occurrence number of a partition that was not assigned to any drive at the time of this command.

NOTE: You cannot assign a partition to a drive if that partition is assigned to a different drive at the time of the command.

You can remove an assignment from a drive, by entering a command in the following form:

A>**ASSIGN {drive}: =** ***RETURN***

Where {**drive**} is the letter of a drive that is currently assigned a partition.

NOTE: While this command will unassign any partition that was assigned to the specified drive, it will also cause the "BAD PARTITION NAME" message to be displayed.

6 ASSIGN RESTRICTIONS

A maximum of two Winchester Disk partitions can be assigned to drive names at one time, regardless of how many partitions have been established on the disk.

If you boot up to one of the Winchester Disk partitions, that partition will be drive A. The 8-inch drive will be drive C, which leaves drive name B open for another Winchester Disk partition. In this case, it is not advisable for you to later assign a different partition to drive A. An error might occur if you reassign drive A and then perform a warm boot or exit from an application program, because CP/M will try to reload part of the operating system from the partition that has been newly assigned to drive A. This newly assigned partition might have a different kind of CP/M system or no CP/M system at all.

If you boot up to the 8-inch drive of the Z67, that drive will begin the session as drive A, and two Winchester Disk partitions can be assigned B and C.

The drive name B can never be assigned to the 8-inch drive, and the drive names A and C can never be assigned to two Winchester Disk partitions at the same time.

When you boot up to the Z67 disk drive unit (to the floppy disk or to a Winchester Disk partition) then any other drives in the hardware environment are automatically assigned the drive names D, E, and F.

7 ASSIGN ERROR MESSAGES

```
BAD PARTITION NAME
```

EXPLANATION: You tried to assign a valid drive name to an invalid partition name, or user made syntax error during command entry. You must perform cold boot, and re-enter any assignments made.

```
BAD DRIVE NAME
```

EXPLANATION: You tried to assign partitions to both drive names A and C, or to assign drive name B to the H/Z-67 floppy disk. Assignment commands must be re-entered.

```
PARTITION ALREADY IN USE
```

EXPLANATION: You tried to assign a new drive name to a partition name that already has an assigned drive name. The assignment of the partition's currently assigned partition must be removed before a new drive name can be assigned to the partition.

```
BAD OCCURRENCE NUMBER
```

EXPLANATION: You tried to assign a partition name with an invalid occurrence number to a valid drive name. This assignment should be attempted again with an occurrence number that identifies an existing partition.

```
BIOS DOESN'T SUPPORT PARTITIONING
```

EXPLANATION: The operating system used for bootstrap will not accommodate Winchester Disk partitions. This system can be modified to accommodate partitions by performing the MAKEBIOS, MOVCPM67, SYSGEN activities in sequence.

```
INCORRECT VERSION OF BIOS
```

EXPLANATION: The operating system in use is the wrong version of CP/M. CP/M Version 2.2.04 must be used to perform bootstrap.

```
SECTOR SIZE OF DISK IS ZERO -- FATAL ERROR
```

EXPLANATION: The superblock may have been damaged. (The superblock is a unit of software recorded on the first sectors of your Winchester disk. This software makes it possible for you to access your partitions.) Repeat the ASSIGN command. If the command fails repeatedly, then copy all accessible data and software from your CP/M partitions, using BRS, PIP, and/or MOVCPM and SYSGEN. Then use the PREP and PART utilities that were supplied with your H/Z-67 hardware.

```
DISK READ ERROR
```

EXPLANATION: Part of the reserved software recorded on the first sectors of your Winchester disk may have been damaged. Repeat the ASSIGN command. If the command fails repeatedly, then copy all accessible data and software from your CP/M partitions, using BRS, PIP, and/or MOVCPM and SYSGEN. Then use the PREP and PART utilities that were supplied with your H/Z-67 hardware.

BRS

The Utility that Facilitates Backup of Winchester Disk Files to Floppy Disks

(Used with the H/Z-67 drive model)

The BRS (Backup and Restore System) utility enables you to create backup copies of all files on a Winchester Disk, and to put those files onto floppy disks (4). It also enables you to restore the backup copies to the Winchester Disk (5). In addition, BRS will compare floppy disk files with Winchester Disk files to verify the accuracy of the copies (6), and display a directory list of files that have been backed up (7).

1. BEGINNING BRS USE

Before using BRS, you must be certain that the Winchester Disk Partition to be used has a currently assigned drive name. Drive names can be assigned to partitions by using the ASSIGN utility.

To begin using BRS, you invoke the utility, enter the date, specify a drive name for a Winchester Disk partition, and specify a drive name for a floppy disk.

1.1 BRS Invocation

The Backup and Restore Utility is stored on the distribution disk under the file name BRS.COM. When the system prompt appears, it can be invoked by the entry of the primary file name, as shown:

```
A>BRS RETURN
```

1.2 Initial BRS Display

This entry will cause BRS to produce a display in the following form:

```
    H/Z67 Winchester Disk Backup/Restore System  (BRS)
--------------------------------------------------------------

 Release Ver. 1.2              Copyright (C)   March 1982
                 Heath/Zenith Data Systems


 Enter Today's Date. (MM/DD/YY)    ===>
```

1.3 Entering Today's Date

The initial BRS display prompts you to enter the date on which the utility is being operated.

```
Enter Today's Date.  (MM/DD/YY)    ===>
```

DATE ENTRY FORM

This date must be entered in a specific form:

- Numerals must be entered to represent the month, day, and year—in that order.
- Month, day, and year must be separated by the slash symbol.
- One or two digits may be entered for the month and day, while two digits must be entered for the year.
- The value entered for month can be no greater than 12.
- The value entered for day can be no greater than 31.
- Zeros can be entered for any component of the date

The following example shows the display that could result from entering a valid date response:

```
Enter Today's Date.  (MM/DD/YY)    ===>7/4/76  RETURN
```

DATE ENTRY FORM

This date must be entered in a specific form:

- Numerals must be entered to represent the month, day, and year—in that order.
- Month, day, and year must be separated by the slash symbol.
- One or two digits may be entered for the month and day, while two digits must be entered for the year.
- The value entered for month can be no greater than 12.
- The value entered for day can be no greater than 31.
- Zeros can be entered for any component of the date

The following example shows the display that could result from entering a valid date response:

```
Enter Today's Date.  (MM/DD/YY)  ===>7/4/76  RETURN
```

DATE ENTRY ERRORS

When you enter a date with an improper quantity or type of characters, the utility displays the message:

```
Syntax Error
```

and reprompts you for a valid entry. If you enter a date with month or day values exceeding the limits for these components, the utility displays the message:

```
Invalid Date Entered:
```

and reprompts you for a valid entry.

If you enter a date in the proper form, but wish to change it, then you may select a "DATE Change" operation from an upcoming menu.

1.4 Disk Drive Specification

After the date prompt has been answered, BRS will prompt for drive device specifications.

WINCHESTER DISK SPECIFICATION

The first drive prompt will ask for a drive name that has been assigned to a Winchester Disk partition, as shown:

```
Enter Winchester Disk Drive Spec. (A:-P:) ===>
```

Enter the drive name that has been assigned the Winchester Disk partition involved in the BRS operation. Then enter a carriage return. (No drive name colon is necessary.)

FLOPPY DISK SPECIFICATION

After you enter the drive name of the Winchester Disk partition, BRS will prompt for the drive name of the floppy disk drive being used for backup copies, as shown:

```
Enter Floppy Disk Drive Spec.   (A:-P:) ===>
```

Enter the name of the drive containing the floppy disk involved in the BRS operation. Then enter a carriage return. (No drive name colon is necessary.)

DRIVE SPECIFICATION ERRORS

If you enter an invalid drive name (only drive names A through F are valid), the utility displays the error message:

```
Invalid Drive Specification
```

The utility then reprompts for a valid drive name.

If you enter a drive name from A to F which does not exist in your hardware environment, or which has not been assigned to a Winchester Disk partition since the last cold boot, then BRS will display the message:

```
Drive not available in Current Configuration
```

The utility then reprompts for a valid drive name. You can use the ASSIGN utility to assign a drive name to a Winchester Disk partition, and then reinvoke BRS. When a valid, assigned drive name is entered for the Floppy Disk Drive, BRS will display the Master Menu (See “2 BRS Master Menu Transactions”).

2 BRS MASTER MENU TRANSACTIONS

The BRS Master Menu gives you the choice of performing any of six operations coordinated by the BRS utility. The most important of these operations are "BACKUP Creation" and "RESTORE Files". These and the other menu operations ("DATE Change", "COMPARE Files", "LIST Directory", and "EXIT to CP/M") are explained in upcoming text.

2.1 The Master Menu

After you invoke the BRS utility and enters the date and drive specifications, BRS will display the following menu and prompt:

```
                 BACKUP AND RESTORE MASTER MENU
          ------------------------------------------

                   D   DATE Change
                   B   BACKUP Creation
                   R   RESTORE Files
                   C   COMPARE Files
                   L   LIST Directory
                   E   EXIT to CP/M

   OPTIONS: Valid Only on BACKUP(B) and RESTORE (R) Operations

         V = Verify Files After Writing. (B/R)
         W = Warning Messages Disabled. (B/R)
         A = Allow R/O Files For Restore. (R)
         F = File Deletion After Saving. (B)
         P = Prompt all actions. (B/R)

   Select Operation (X /Y/Y/Y)      ====>
```

You can select a BRS Operation and, in some cases, one or more Options from the Master Menu. The selection is entered in response to the prompt at the bottom of the display.

OPERATION AND OPTION ENTRY METHOD

An Operation can be selected at the

```
Select Operation (X /Y/Y/Y)       ====>
```

prompt by entering the Operation letter (D, B, R, C, L, or E). BRS immediately spells out the rest of the title of the Operation.

If you are performing the BACKUP Operation (B) or the RESTORE Operation (R), then Option letters (V, W, A, F, or P) can be entered to govern the characteristics of the operation. Each Option should be preceded by a slash mark ("/"). Enter Option letters immediately after the Operation letter is entered. However, on the screen display, it will appear that the Option follows the word "Backup" or "Restore" in the Operation title, because BRS finished spelling the word automatically.

EXAMPLE MASTER MENU SELECTION

Your entry of the "B" menu Operation will cause BRS to immediately spell out the command by adding the characters "ackup". The resulting screen display would look like this:

```
Select Operation (X /Y/Y/Y)       ====> Backup
```

To further define this Backup Operation, you can enter slash marks and Option letters, such as "V" and "P", at the right end of the word "Backup". The entry is terminated by a carriage return. The resulting display would look like this:

```
Select Operation (X /Y/Y/Y)       ====> Backup/V/P
```

After such an entry, BRS will begin the Operation that backs up files from a Winchester Disk partition to a floppy disk. After copying the files to the floppy disk, the program will Verify (/V) the backup files to ensure that they are identical to the files on the Winchester Disk partition. In addition, BRS will Prompt (/P) you before it performs any step in the Backup Operation. (All Operations and Options are explained in detail in upcoming text sections.)

2.2 Master Menu Operations

The paragraphs on this page summarize the properties of each BRS Master Menu Operation.

Date Change

This Operation enables you to correct the date entered when the BRS program was invoked.

Backup

This Operation sends copies of a selected group of files from a Winchester Disk partition to one or more floppy disks. It can be used with Options "V", "W", "F", and "P".

Restore

This Operation sends copies of a selected group of files from one or more floppy disks to a Winchester Disk partition. It can be used with Options "V", "W", "A", and "P".

Compare Files

This Operation checks file backups on a floppy disk against file originals on a Winchester Disk to detect any differences between them.

List Directory

This Operation displays a directory of your backup files, the date they were created, the number of volumes within the disk set, the volume number of the disk containing the file, and the user number of the file.

Exit to CP/M

This Operation ends all other BRS Operations and returns control to the CP/M Operating System, which displays the CP/M system prompt.

2.3 Master Menu Options

If you select Master Menu Operation "B" or "R", then you can also enter options to tell BRS how to perform the BACKUP or RESTORE Operation. The letters that stand for Options should be entered with a slash mark ("/") after the letter standing for the selected Operation. (BRS will instantly display the entire word or title for the Operation after you enter the first letter.) Options **cannot** be entered with the "D", "C", "L", or "E" Operations.

For example, you could respond to the "Select Operation" prompt at the bottom of the menu with the following Operation and Option entry:

```
Select Operation (X /Y/Y/Y)        ====> Backup/V/P
```

The following descriptions explain how each Master Menu Option works:

/V **The Verify Option:** This Option checks each byte of the destination file against each byte of the source file to see that both files are identical. If the two files are not identical, BRS displays an error message.

NOTE: If a discrepancy between the data of two files does occur, you must decide which file is the accurate one, delete the inaccurate file, and make another attempt at creating an identical file. If this option is performed during all Backup and Restore operations, you may assume that the oldest file of the two being compared is the correct file because it has been verified previously. Hence the file that is the most recent copy would be the one that is in error. That file should be deleted, and a new copy made.

The "/V" Option can be used with both the BACKUP and RESTORE Operations.

/W **The Warning Message Disable Option:** Selection of this Option suppresses all warning messages that may occur during an Operation. These warning messages are of two types, and occur when you attempt to:

- write to a disk that is not blank
- delete files from the Winchester Disk

When the "/W" Option is specified, the program will perform the requested command without warning you beforehand. The "/W" Option can be used with both the BACKUP and RESTORE Operations.

/A **The Allow R/O File Restoration Option:** Selection of this Option allows you to restore files that have been given R/O (Read/Only) status. It allows the utility to delete R/O files if they exist on the Winchester Disk. If the "/A" option is not selected, and an attempt is made to restore the R/O file, then BRS will respond:

```
Found User# 00 BRS.COM File is R/O, Not Being Restored
         Enter Carriage Return When Ready
```

This Option can be entered only with the RESTORE Operation.

/F **The File Deletion Option:** Selection of this Option causes the program to delete the files from the source disk after the files have been copied to the destination disk that is being used for backup. The "/F" Option is only available when the BACKUP Operation is selected.

/P **The Prompt Option:** If the "/P" Option is selected, you must respond to a prompt for each action before BRS performs that action. When the "/P" Option is selected during the RESTORE Operation, for example, BRS would ask you if the existing files on the Winchester disk may be deleted so that they can be replaced with restored backup files. Only those files on the Winchester Disk that have names that are identical to those being restored can be deleted. The "/P" Option can be selected with both the BACKUP and RESTORE Operations.

3 THE DATE CHANGE OPERATION

The "DATE Change" Menu Operation enables you to change the date that was entered during the initial date prompt, in case that date was incorrect. The date helps BRS to distinguish between different disks that may contain files with identical file names.

3.1 Selecting the DATE Change Operation

The "DATE change" operation is invoked by entering the letter **D** and a carriage return in response to the Master Menu prompt. (BRS will instantly complete the command with the characters "ate Change".)

This entry will cause BRS to display the date you entered in response to the initial BRS display, and a prompt requesting the corrected date. If you initially entered the date "02/30/84", and now wants to correct it, the BRS utility would display the following message and prompt:

```
Current Date is 02/30/84

Enter Change else hit RETURN ====>
```

3.2 DATE Change Execution

To change the date entered initially, you should enter a new date in the appropriate form, and then a carriage return. The same form for date entry is used here as with the original date entry (see 1.3 "Entering Today's Date"), and the same error messages will result if an invalid date entry is made.

The date listed as "Current Date" will be retained if the you answer the prompt with a carriage return alone.

After you enter the carriage return, the Master Menu reappears.

4 THE BACKUP CREATION OPERATION

The BACKUP Operation transfers files from Winchester disk partitions to floppy disks. During this Operation, you must specify which files on the partition should be left alone ("rejected" by the Operation) and which files should be copied ("selected" by the Operation).

To make file specification more convenient, you can enclose the names of several files in a batch file. When the name of a batch file is specified, the names of all of the enclosed files will be specified automatically.

4.1 Selecting the BACKUP Creation Operation

To begin the BACKUP operation, enter the letter **B** at the Master Menu selection prompt. After the "B" you may enter four of the five options listed at the bottom of the Master Menu (**V**, **W**, **F**, and **P**) with a slash mark immediately before each one. The entry should end with a carriage return.

In the following example, the "**B**" is the command for the BACKUP Operation, and the "**/V/F/P**" are the three options selected as an example:

```
Select Operation (X /Y/Y/Y)        ====> Backup/V/F/P
```

BRS adds part of the command by instantly displaying "ackup" after the "B" that you entered.

4.2 Rejection File Specification

After you choose the BACKUP operation, the following prompt is displayed:

```
Enter Rejection File Spec(s).  ====>
```

Any file specified at this prompt will be executed from the upcoming BACKUP operation.

ENTERING EXPLICIT FILE NAMES

The Rejection File prompt will reappear each time you responds to it with a file name and carriage return, as shown:

```
Enter Rejection File Spec(s).  ====> TESTFILE.DOC   RETURN

Enter Rejection File Spec(s).  ====> RETURN
```

The file named "TESTFILE.DOC" will be excluded from the upcoming BACKUP operation (rejected).

ENTERING "WILD CARD" FILE NAMES

You can also answer the prompt with a "wild card" file name (a file name with "*" or "?" characters in it), as shown:

```
Enter Rejection File Spec(s).  ====> *.ASM   RETURN
```

When this entry is made, BRS automatically displays the explicit file names that are implied by the "wild card" file name. These explicit file names are displayed at the end of rejection prompts beneath the prompt you just answered, so it appears as if you entered them. For example, the explicit file names implied by specifying "*.ASM" might display as shown:

```
Enter Rejection File Spec(s).  ====> *.ASM   RETURN

Enter Rejection File Spec(s).  ====> DUMP.ASM   RETURN

Enter Rejection File Spec(s).  ====> BIOS.ASM   RETURN

Enter Rejection File Spec(s).  ====> PROGRAM9.ASM   RETURN

Enter Rejection File Spec(s).  ====>  RETURN
```

ENTERING BATCH FILE NAMES

Another way you can reject more than one file at once is to specify a batch file name. A batch file contains the names of several other files that will be specified automatically. (See 4.4 Creating Batch Files.) A batch file name must be preceded by a semicolon (;) when specified at the rejection prompt, as shown:

```
    Enter Rejection File Spec(s).  ====> ;BATCHFIL.DOC  RETURN
```

The files within batch file "BATCHFIL.DOC" will be excluded from the BACKUP operation (rejected).

When this type of entry is made, BRS automatically displays the names of the files that are stored in the specified batch file. These file names are displayed at the end of rejection prompts beneath the prompt you just answered, so it will appear as if you had entered them. For example, BRS might display the names of files contained in "BATCHFIL.DOC" as shown:

```
Enter Rejection File Spec(s).  ====> PIP.COM   RETURN

Enter Rejection File Spec(s).  ====> STAT.COM   RETURN

Enter Rejection File Spec(s).  ====> MBASIC.COM   RETURN

Enter Rejection File Spec(s).  ====> PROG2A.BAS   RETURN

Enter Rejection File Spec(s).  ====>   RETURN
```

REJECTING FILES WITH THE MASTER MENU PROMPT OPTION

If the /P ("Prompt all actions") Option was selected with the BACKUP operation, BRS will prompt you to confirm the rejection of each specified file. For the specification "TESTFILE.DOC", one of these prompts might be displayed as follows:

```
    Found User# 00 TESTFILE.DOC Reject ? (Y/N) <Y> ====>
```

To confirm the rejection, enter the letter **Y** or a carriage return. To nullify the rejection, enter the letter **N**.

INVALID FILE SPECIFICATION

If you respond to the prompt by entering the name of a file that does not exist on the specified drive device, BRS will display the message:

```
No Files Found For Entered Spec.
```

and redisplay the prompt.

ENDING REJECTION SPECIFICATIONS

You can stop the specification of rejection files by entering a carriage return alone in response to the prompt:

```
Enter Rejection File Spec(s)?  ====> RETURN
```

This response will produce the menu for File Selection Specifications and the "Selection File Spec(s). (X /Y/Y/Y) = = = =>" prompt.

4.3 Selection File Specification

After you have specified a desired number of rejection files, the following display and prompt will appear:

```
          Options for File Selection Spec(s).
 ------------------------------------------------
             A  - All File Types (Default)
             N  - Nonsystem
             S  - System
             U  - Current User #  (Default)
             U* - All Users
             Un - One Specific User #

Enter Selection File Spec(s). (X /Y/Y/Y) ====>
```

Any file specified at this prompt will be included in the upcoming BACKUP Operation (unless it was first specified at a rejection prompt).

ENTERING EXPLICIT FILE NAMES

The Selection File prompt will reappear each time you respond to it with a file name and carriage return, as shown:

```
Enter Selection File Spec(s). (X /Y/Y/Y)  ====> TESTFILE.DOC  RETURN

Enter Selection File Spec(s). (X /Y/Y/Y)  ====> RETURN
```

The file "TESTFILE.DOC" will be included in the upcoming BACKUP operation (selected).

ENTERING "WILD CARD" FILE NAMES

You can also answer the prompt with a "wild card" file name (a file name with "*" or "?" characters in it), as shown:

```
Enter Selection File Spec(s). (X /Y/Y/Y)  ====> *.ASM RETURN
```

When this entry is made, BRS automatically displays the explicit file names that are implied by the "wild card" file name. These explicit file names are displayed at the end of the selection prompts beneath the prompt you just answered, so it appears as if you entered them. For example, the explicit file names implied by specifying "*.ASM" might display as shown:

```
Enter Selection File Spec(s). (X /Y/Y/Y) ====> *.ASM  RETURN

Enter Selection File Spec(s). (X /Y/Y/Y)  ====> DUMP.ASM  RETURN

Enter Selection File Spec(s). (X /Y/Y/Y)  ====> BIOS.ASM  RETURN

Enter Selection File Spec(s). (X /Y/Y/Y)  ====> PROGRAM9.ASM  RETURN

Enter Selection File Spec(s). (X /Y/Y/Y)  ====> RETURN
```

ENTERING BATCH FILE NAMES

You can select more than one file at once by specifying a batch file name. A batch file contains the names of several other files that will be specified automatically. (See 4.4 Creating Batch Files.) A batch file name must be preceded by a semicolon (;) when specified at the selection prompt, as shown:

```
    Enter Selection File Spec(s). (X /Y/Y/Y)  ====>;BATCHFIL.DOC  RETURN
```

The files within batch file "BATCHFIL.DOC" will be included in the BACKUP operation (selected).

When this entry is made, BRS automatically displays the names of the files that are stored in the specified batch file. These file names are displayed at the end of selection prompts beneath the prompt you just answered, so it appears as if you entered them. For example, BRS might display the names of files contained in "BATCHFIL.DOC" as shown:

```
Enter Selection File Spec(s). (X /Y/Y/Y)  ====> PIP.COM  RETURN

Enter Selection File Spec(s). (X /Y/Y/Y) ====> STAT.COM  RETURN

Enter Selection File Spec(s). (X /Y/Y/Y) ====> MBASIC.COM  RETURN

Enter Selection File Spec(s). (X /Y/Y/Y) ====> PROG2A.BAS  RETURN

Enter Selection File Spec(s). (X /Y/Y/Y) ====>  RETURN
```

SELECTING FILES WITH THE MASTER MENU PROMPT OPTION

If the /P ("Prompt all actions") Option was selected with the B ("BACKUP") operation, BRS will prompt you to confirm the selection of each specified file. For the specification "TESTFILE.DOC", one of these prompts might be displayed as follows:

```
    Found User# 00 TESTFILE.DOC Select ? (Y/N) <Y> ====>
```

To confirm the selection, enter the letter **Y** or a carriage return. To nullify the selection, enter the letter **N**. After one of these responses is entered, BRS displays a similar prompt for the next explicit file name implied by the specified "wild card" name, and another, and another, until you have confirmed or nullified selections for all of the implicated files.

INVALID FILE SPECIFICATION

If you respond to the prompt by entering the name of a file that does not exist on the specified drive device, BRS will display the message:

```
No Files Found For Entered Spec.
```

and redisplay the prompt. This message will also appear if you specify a file (explicit, "wild card", or batch) for both file rejection and file selection. A rejected file cannot be selected during the same run of the BACKUP operation.

```
Ending Selection Specifications
```

ENDING SELECTION SPECIFICATIONS

You can end the specification of selection files by entering a carriage return alone in response to the prompt (as shown),

```
Enter Selection File Spec(s).  ====>
```

This response will lead to execution of the BACKUP operation, which begins with erasure and preparation of the floppy disk (See "4.5 BACKUP Creation Execution").

SELECTION FILE OPTIONS

You can also limit the types of files that may be affected during the BACKUP operation by specifying Selection File Options. (These options are not to be confused with the Main Menu Options for Operation Selection, although they are entered in a similar fashion.)

These Options are entered after the name of any file specified for selection. Each option letter must be preceded by a slash mark. If no Options are entered, BRS assumes default values. (See "Selection File Option Defaults".) A description of each of these Options follows.

/A **All System and Non-System files Option.** It defines the Backup operation so that both System and/or Non-System files will be affected. This is one of the default Options.

/N **Non-System files Option.** It defines the Backup Operation so that only Non-System files will be affected.

/S **System files Option.** It defines the Backup Operation so that only System files will be affected.

/U **Current User Option.** It defines the Backup Operation so that only those files accessible through the currently logged user area will be affected. This is one of the default Options.

/U* **All User files Option.** It defines the Backup Operation so that the files to be affected can be accessed through any user area on the partition.

/Un **User Number "n" files Option**, where "n" is a number from zero (0) to fifteen (15), inclusive. It defines the Backup Operation so that only the files of user number "n" will be affected.

NOTE: User areas are accessed by implementing the USER resident command. Files can be transferred between user areas by implementing the PIP utility with the [G] parameter. USER and PIP can only be invoked in response to the CP/M system prompt.

SELECTION FILE OPTION DEFAULTS

Two of the Options are applied to Selection File Specifications by default if you do not specify any Options. These two default Options are "A—All File Types" and "U—Current User #".

These Options automatically affect the characteristics of any Selection File Specification for which other Options are not specified. Therefore, a Selection File Specification entered with no Options will select files of all types from the currently logged user area number. (Unless one has been designated since the last cold boot, the currently-logged user area number is zero.)

EXAMPLE ENTRIES WITH SELECTION FILE OPTIONS

Selection File Specifications can be entered with Options as shown:

```
Enter Selection File Spec(s) ====> TESTFILE.COM/N/U* RETURN
```

In the preceding example, **TESTFILE.COM** is the filename, **/N** limits the selection at this prompt to Non-System files only, and **U*** allows the selection to be made from any user area.

Options are also used in this example:

```
Enter Selection File Spec(s) ====> *.*/S/U14   RETURN
```

In this example, ***.*** is the file name, **/S** limits file selection at this prompt to system files only, and **/U14** further limits file selection at this prompt to files accessible from user area 14.

SELECTION FILE OPTION ERRORS

The following two error messages occur when Selection File Options are entered improperly:

```
Invalid Option Format or Selection
```

and

```
Invalid User Number or Format
```

If these messages do occur, the "Enter Selection File Spec(s)" prompt will appear again and await an appropriate entry.

4.4 Creating Batch Files

You can enter several file names in response to a file rejection or specification prompt by storing these file names in a batch file.

A batch file is a text file composed entirely of file names and Options.

The name of the batch file itself is entered at file rejection or specification prompt to trigger the entry of all of the file names within it. When entered, the name of the batch file itself must be preceded by a semicolon (;), as in the following example:

`Enter Rejection File Spec(s).  ====>` **;BATCHFIL.DOC** ***RETURN***

The batch file can be composed using the ED utility or any other text editor or word processor. It can contain any file name that conforms to CP/M file naming conventions.

The file names within the batch file should be entered on separate, successive lines. Each line must be terminated with a carriage return.

CONTENTS OF EXAMPLE BATCH FILE

The example batch file BATCHFIL.DOC, for instance, contains the following file names:

```
ACCOUNTS1.DAT
ACCOUNTS2.DAT
FINANCE.COB
STATS.FOR
```

When you specify the batch file BATCHFIL.DOC at a rejection or selection prompt, these files will be automatically specified for rejection or selection.

FILE SELECTION OPTIONS WITHIN A BATCH FILE

If you enter a batch file name at a selection file specification prompt, the file names within the batch file can have Selection File Options. The following example shows the contents of a batch file that contains options with some of its files:

```
BIOS.SYS/S
PAYROLL5.DAT/U14
MEMO????.DOC/U*
*.*/N/U
```

In addition, you can enter a batch file name with options when responding to a selection file prompt. The options will apply to all of the files within the batch file.

4.5 BACKUP Execution

After you have specified the names of the files that should be backed up, the "Enter Selection File Spec(s). (X /Y/Y/Y)= = = = >" prompt should be answered with a carriage return alone. This entry tells BRS to begin execution.

BACKUP EXECUTION MESSAGE

BRS will display the following message when it is ready to execute the BACKUP Operation:

```
              Load Drive B: with a Blank Floppy Disk

      *** WARNING Data on This Disk Will be DESTROYED !! ***

                      Press RETURN to begin
```

(This warning message will not appear if you entered the "/W" option at the Master Menu.)

PREPARING BACKUP FLOPPY DISKS FOR DATA STORAGE

You should insert a floppy disk into the floppy disk drive in the H/Z-67 drive unit. This disk need not be blank, but any data on it must be expendable, because the BACKUP Operation begins by preparing the disk surface for the storage of **new** data.

After inserting a disk into the floppy disk drive, enter a carriage return. The following prompt will be displayed:

```
Which density? (S=single, D=double) ====>
```

Enter **S** or **D** depending upon the amount of data desired on the Backup Disk. Then BRS will prompt:

```
Press RETURN to begin.
```

The entry of a carriage return will begin execution of the BACKUP operation.

If one of the floppy disks used for backup cannot be formatted, then BRS will display:

```
Unable to format this disk. Place a different disk in the drive.
```

Make sure the disk is properly inserted and write enabled. Then, if it still cannot be formatted, replace it with a different disk for this Operation.

ACTUAL BACKUP EXECUTION

When actual BACKUP execution begins, BRS will display the following message:

```
********  BACKUP OPERATION IN PROGRESS  ********
```

If BACKUP execution stops because all of the selected files have been backed up to the floppy disk, then the operation is finished and the Master Menu will be displayed again.

If BACKUP execution stops because the floppy disk has been backed up to capacity while more selected files remain on the Winchester Disk, then more floppy disk space is needed. Refer to the next text section for guidance.

BACKING UP TO SEVERAL FLOPPY DISKS

If the files selected for the BACKUP Operation will not all fit on a single floppy disk, then BRS will interrupt BACKUP execution with the following message:

```
                   Current Floppy Disk is Full

             Load Drive X: with a Blank Floppy Disk

    *** WARNING  Data on This Disk Will be DESTROYED !! ***

                Enter Carriage Return When Ready
```

Where “Drive X: ” is the drive already specified as the Floppy Disk Drive for the operation.

After inserting a new disk and entering a carriage return, the density prompts will appear again, as shown:

```
    Which density? (S=single, D=double) ====>
```

Enter **S** or **D** to indicate the density desired on the backup floppy disk. Then the following prompt will appear as shown:

```
    Press RETURN to begin.
```

Enter a carriage return to begin the formatting and copying of data to this floppy disk.

BACKUP DISK SET DIRECTORY

When Winchester Disk files are backed up to more than one floppy disk, then the floppy disks used are referred to as the "Backup Disk Set". BRS automatically gives each disk in this set a volume number, starting with the first disk inserted (Volume 01).

Volume 01 of the Backup Disk Set is also referred to as the "Directory Disk".

To ensure that the volumes in a Backup Disk Set are kept in order, BRS records a directory on the Volume 01 disk. This directory is stored in a file named "MASTBRS.DIR". BRS prompts you to reinsert the Volume 01 disk so that the directory can be recorded:

```
Load Drive X: With Volume 01 of Backup Disk Set

        Enter Carriage Return When Ready
```

Where "Drive X: " is the drive already specified as the Floppy Disk Drive for the Operation.

WRONG DISK ERROR MESSAGE

If you are prompted to load "Volume 01" and insert a disk other than Volume 01 (the Directory Disk), the following error message will appear:

```
Wrong Disk Inserted in Drive

 Enter (CR) for Master Menu
```

Respond to this message by entering a carriage return. The Master Menu will appear.

5 THE RESTORE FILES OPERATION

The RESTORE operation transfers backup files from floppy disks to a Winchester disk partition. During execution of this Operation, you must specify which files on the floppy disk should be left alone ("rejected" by the Operation), and which files should be copied ("selected" by the Operation).

RESTORE is similar to BACKUP in the form of its messages and prompts. The biggest difference is the appearance of two extra screen displays after you select RESTORE from the Master Menu.

5.1 Selecting the RESTORE Files Operation

To begin the RESTORE Operation, enter the letter **R** at the Master Menu selection prompt. After entering the "R", you may enter four of the five options listed at the bottom of the Master Menu (**V**, **W**, **A**, and **P**)

In the following example, the "**R**" is the command for the RESTORE Operation, and the "**/V/A/P**" are the three options selected:

```
    Select Operation (X /Y/Y/Y) ====>Restore /V/A/P
```

BRS adds part of the command by instantly displaying "estore" after the "R" that you entered.

5.2 The Disk Set Directory

The files transferred during a single BACKUP Operation can reside on several floppy disks. These disks are collectively called a Disk Set. BRS maintains a Disk Set Directory that helps it to restore files to the Winchester disk in the same order that they existed before they were backed up.

The Disk Set Directory is stored on the first disk that you inserted during the BACKUP Operation, which becomes known as the Directory Disk. This directory assigns a "Volume" number to each disk in the set, and "Volume 01" is always assigned to the Directory Disk.

When you want to RESTORE files from a floppy Disk Set, BRS must access the directory from the Directory Disk (Volume 01 of the Disk Set) to determine where the files are.

INSERTING THE DIRECTORY DISK

To remind you to insert the Directory Disk, BRS displays the following prompt:

```
Load Drive    X: With Volume 01   of Backup Disk Set.
           Enter Carriage Return When Ready.
```

Where "Drive X:" refers to the drive (A through P) that you assigned as the Floppy Disk Drive before the Master Menu appeared.

Respond to this prompt by inserting the Directory Disk (Volume 01) into the specified drive and entering a carriage return.

WRONG DISK ERROR MESSAGE

If you are prompted to load "Volume 01" and insert a disk other than Volume 01 (the Directory Disk), the following error message will appear:

```
Wrong Disk Inserted in Drive
 Enter (CR) for Master Menu
```

Respond to this message by entering a carriage return. The Master Menu will appear.

DIRECTORY VERIFICATION

When a Directory Disk (Volume 01) is inserted, BRS examines it and displays some of the characteristics of the Backup Disk Set in the following form:

```
Backup Disk Set:  Created mm/dd/yy
                  Start vol. 01
                  Last  vol. nnn

Is This The Correct Backup Set?  (Y/N) <Y> ====>
```

Where "mm/dd/yy" is the date you performed the BACKUP operation that created the file copies in this Disk Set; and

where "nnn" is the volume number of the last disk in the set.

If the “Created” date and the “Last vol.” number shown in this display do not match those of the Disk Set that you want to restore, the prompt at the bottom of the display should be answered with an “N”. BRS will redisplay the “Load Drive X:” prompt to allow you to insert a different Directory Disk.

If the date and volume numbers match, enter **Y** or a carriage return, and read about “Rejection File Specification”.

5.3 Rejection File Specification

After you verify that the disk in the floppy disk drive is Volume 01 of the “Correct Backup Set”, the following prompt is displayed:

```
    Enter Rejection File Spec(s).  ====>
```

Any file specified at this prompt will be excluded from the ensuing RESTORE operation.

USER RESPONSE

This prompt will reappear each time you respond to it with a file name and carriage return, as shown:

```
    Enter Rejection File Spec(s).  ====>TESTFILE.DOC   RETURN

    Enter Rejection File Spec(s).  ====> RETURN
```

or a batch file name,as shown:

```
    Enter Rejection File Spec(s).  ====>;BATCHFIL.DOC   RETURN

    Enter Rejection File Spec(s).  ====> RETURN
```

The file named “TESTFILE.DOC” will **not** be backed up. The files within batch file “BATCHFIL.DOC” will not be backed up either. The specification of a batch file must be preceded by a semicolon. (See 5.5 “Creating Batch Files”.)

INVALID FILE SPECIFICATION

If you respond to the prompt by entering the name of a file that does not exist on the specified drive device, BRS will display the message:

```
    No Files Found For Entered Spec.
```

and redisplay the prompt.

Each file name entry must be followed by a carriage return. Ambiguous file names can be entered (using wild card symbols "*" and "?"). But no drive names can be specified at this prompt.

ENDING REJECTION SPECIFICATIONS

You can end the specification of rejection files by entering a carriage return alone in response to the prompt:

```
    Enter Rejection File Spec(s).   ====>
```

This response will produce the "Selection File Specification" display and prompt.

5.4 Selection File Specification

After you have specified the desired number of rejection files, the following display and prompt will appear:

```
          Options for File Selection Spec(s).
    --------------------------------------------------
             A  - All File Types (Default)
             N  - Nonsystem
             S  - System
             U  - Current User =   (Default)
             U* - All Users
             Un - One Specific User =

    Enter Selection File Spec(s).  (X /Y  (X /Y/Y/Y)   ====>
```

Any file specified at this prompt will be included in the ensuing RESTORE Operation.

USER ENTRY

This prompt will reappear each time you respond to it with a file name and carriage return, as shown:

```
Enter Selection File Spec(s).  (X /Y/Y/Y) ====> TESTFILE.DOC  RETURN

Enter Selection File Spec(s).  (X /Y/Y/Y) ====> RETURN
```

or a batch file name, as shown:

```
Enter Selection File Spec(s).  (X /Y/Y/Y) ====> ;BATCHFIL.DOC  RETURN

Enter Selection File Spec(s).  (X /Y/Y/Y) ====> RETURN
```

The file named "TESTFILE.DOC" will be restored. The files within batch file "BATCHFIL.DOC" will also be restored. The specification of a batch file must be preceded by a semicolon. (See 5.5 "Creating Batch Files".)

Each file name entry must be followed by a carriage return. Ambiguous file names can be entered (using wild card symbols "*" and "?"). No drive names can be specified at this prompt.

SELECTION FILE OPTIONS

You can also limit the types of files that may be affected during the RESTORE Operation by specifying Selection File Options with file names. (These Options are not to be confused with the Main Menu Options for Operation Selection, although they are entered in a similar fashion.)

These Options are entered after the name of any file specified for selection. Each Option letter must be preceded by a slash mark. If no Options are entered, BRS assumes default values. (See "Selection File Option Defaults".) A description of each of these Options follows.

/A **All System and Non-System files Option.** It defines the Backup Operation so that System and/or Non-System files are affected. This is a default Option.

/N **Non-System files Option.** It defines the Backup Operation so that only Non-System files may be affected.

/S **System files Option.** It defines the Backup Operation so that only System files may be affected.

/U **Current User Option.** It defines the Backup Operation so that only those files accessible through the currently logged User area may be affected. This is a default Option.

/U* **All User files Option.** It defines the Backup Operation so that the files to be affected can be accessed through any User area.

/Un **User Number "n" files Option**, where "n" is an integer from zero (0) to fifteen (15), inclusive. It defines the Backup Operation so that only the files of User number "n" are affected.

You can respond to a single Selection File prompt with both a file name and Options, as shown in the following example:

```
Selection File Spec(s) ==>TESTFILE.COM/N/U* RETURN
```

In this example, **TESTFILE.COM** is the filename, **/N** is a limit for only Non-System files, and **U*** allows all user files to be affected.

FILE SELECTION OPTION DEFAULTS

The default values for the Backup and Restore Options are marked on the menu: "A—All File Types (Default)", and "U—Current User # (Default)". If a file name is entered in response to a Selection File prompt **without** File Selection Options, then the RESTORE operation will affect files of all types that belong to the current user.

INVALID FILE SPECIFICATION

If you respond to the prompt by entering the name of a file that does not exist on the specified drive device, BRS will display the message:

```
No Files Found for Entered Spec.
```

and redisplay the prompt. File Selection Specifications often produce these errors as well:

```
Invalid Option Format or Selection
```

and

```
Invalid User Number or Format
```

OVERRULED FILE SELECTION SPECIFICATION

If you enter the same file name in response to both the Rejection File prompt and the Selection File prompt, the file's "rejection" will prevail. The twice specified file will not be affected by the Operation.

ENDING FILE SELECTION SPECIFICATIONS

You may end the specification of selection files by entering a carriage return alone in response to the prompt (as shown),

```
Enter Selection File Spec(s)?  ====>
```

The immediate entry of just a carriage return at this prompt will cause the RESTORE operation to be executed. After execution, BRS will redisplay the Master Menu and prompt for another Operation.

5.5 Creating Batch Files

You may enter several file names in response to a single specification prompt by storing these file names in a batch file. A batch file is a text file composed entirely of file names.

The name of the batch file itself can then be entered at the prompt to trigger the entry of all of the file names within the batch file. The batch file will be read by BRS, and each file named within it will be excluded from (if entered at a rejection prompt) or included in (if entered at a selection prompt) the RESTORE Operation. When entered, the name of the batch file must be preceded by a semicolon (;), as in the following example:

```
Enter Rejection File Spec(s).  ====>;BATCHFIL.DOC   RETURN
```

The batch file can be composed using the ED utility or another text editor or word processor. It can be given any file name that fits the CP/M file naming conventions.

The file names within the batch file should be entered on separate, successive lines. Each line must be terminated with a carriage return. The example batch file BATCHFIL.DOC, for instance, contains the following file names:

```
ABSTRACT.REC
TAXES1.REC
TAXES2.REC
TAXES3.REC
```

When "BATCHFIL.DOC" is specified, BRS automatically displays the files it contains, so it appears as if you entered them at separate prompts. Hence, specifying "BATCHFIL.DOC" would produce the following display:

```
Enter Selection File Spec(s).  ====>;BATCHFIL.DOC   RETURN

Enter Selection File Spec(s).  ====>ABSTRACT.REC   RETURN

Enter Selection File Spec(s).  ====>TAXES1.REC   RETURN

Enter Selection File Spec(s).  ====>TAXES2.REC   RETURN

Enter Selection File Spec(s).  ====>TAXES3.REC   RETURN

Enter Selection File Spec(s).  ====>  RETURN
```

Batch file names can be specified at either rejection or selection prompts. Additionally, files within a batch file can have selection Options when their batch file is specified at a selection prompt.

5.6 RESTORE Execution

When the actual RESTORE execution begins, BRS will display the following message:

```
    ********  RESTORE OPERATION IN PROGRESS  ********
```

When a Disk Set includes several floppy disks, BRS must transfer the selected files from each of these disks one at a time onto the Winchester disk. This disk-by-disk transfer process requires an interruption in the execution of the RESTORE Operation as BRS prompts you to insert the next disk (Disk Set Volume) that contains a specified file. Whenever the directory indicates to BRS that a file is on a disk (Disk Set Volume) that is not in the specified drive, BRS displays a prompt in the following form:

```
    Load Drive    X: with Volume   nn of Backup Disk Set.
             Enter Carriage Return When Ready.
```

Where "x: " is the drive specified in response to one of the initial BRS prompts; and

where "nn" is the number of the the Disk Set Volume that BRS is now ready to restore to the Winchester disk partition.

You should insert the disk with a volume number corresponding to the one in the prompt, and then a carriage return. When BRS finishes restoring all of the selected files that it can find on the inserted disk, it will display the message:

```
    Load Drive    X: with Volume   nn of Backup Disk Set.
             Enter Carriage Return When Ready.
```

Where "x: " is the drive specified in response to one of the initial BRS prompts; and

where "nn" is the number of the Disk Set Volume that BRS is now ready to restore. This number increases in increments of one each time the prompt is displayed during a RESTORE Operation. Hence you should insert a disk with a higher volume number each time.

The preceding display reappears until you insert the last disk (volume) in the Disk Set. After BRS restores the selected files from the final Disk Set Volume, the RESTORE Operation is terminated and the Master Menu reappears.

WRONG DISK ERROR

If you are prompted to load "Volume 01" and insert a disk other than Volume 01 (the Directory Disk), the following error message will appear:

```
Wrong Disk Inserted in Drive

 Enter (CR) for Master Menu
```

Respond to this message by entering a carriage return. The Master Menu will appear.

If you are prompted to load a disk volume other than Volume 01 and insert the wrong disk volume, the following error message will appear:

```
             Wrong Disk Inserted in Drive

Load Drive     X: with Volume    nn of Backup Disk Set
           Enter Carriage Return When Ready
```

Where "x:" is the drive specified in response to one of the initial BRS prompts; and

where "nn" is the number of the Disk Set Volume that BRS is now ready to restore.

A disk will be considered "Wrong" by BRS is it does not belong to the same Disk Set as the Directory Disk, or if it belongs to the disk set but has a different volume number than the one that is requested in the prompt.

DELETING OLD WINCHESTER DISK FILES

When files are restored to the Winchester Disk from Backup Floppy Disks, the old original file (from which the Backup was copied) must be erased to allow space for the restored copy from the Backup.

If you also responded to the Master Menu prompt by specifying the "/P" (Prompt) Option, then BRS will ask whether or not you want to delete a given file before the program actually deletes that file. The following message would appear:

```
Found User# 00 TEST.ASM     Select ? (Y/N) <Y>
```

Where TEST.ASM is the name of the file that BRS had located from its list of files to be deleted. The question is to be answered with either a "Y" or an "N". The default of "Y" (shown as <Y> on the monitor) is assumed if you enter only a carriage return. Entering "N" will cause BRS to skip over that particular file and not delete it.

If you did **not** respond to the Master Menu prompt by specifying the "/W" (Warning messages suppressed) Option, then the warning messages will be active. Under these conditions, you will be prompted to verify that the files are to be deleted with the following display:

```
*****Warning! Files on Winchester Disk are Going to be Deleted*****
                Enter (CR) to Cancel Deletion Process
```

To stop the deletion, you must enter a carriage return. To begin the deletion, you can enter any other keyboard character.

However, if the "/W" Option was entered, BRS will not offer you the alternative of canceling the deletion process.

6 COMPARE FILES

When large amounts of data are transferred between disks, it is extremely important to be able to determine whether or not the copied data (on the Backup Floppy Disk) is exactly the same as the original data (on the Winchester Disk).

The COMPARE Files Operation compares files on a Backup Disk against the corresponding files on the Winchester Disk to make sure that the data within the two files is exactly the same.

6.1 Selecting the COMPARE Files Operation

To select the COMPARE Files Operation, enter the letter C and a carriage return at the Master Menu Selection prompt. (No Master Menu Options can be selected with this operation.)

```
Select Operation (X /Y/Y/Y)  ====> C
```

BRS will complete the Operation title by displaying the characters "ompare Files" after the "C".

When this Operation is selected, BRS displays the following prompt:

```
Load Drive    X: with Volume   01 of Backup Disk Set.
              Press RETURN when ready.
```

Where "Drive X:" refers to the drive specified before the Master Menu appeared.

After you insert the first disk of a Backup Disk Set and enter a carriage return, BRS will check to see if the inserted disk contains a Backup Disk Set Directory (stored in a file named MASTERBRS.DIR).

If BRS finds a directory on the disk, it will display the following information from the directory:

```
Backup Disk Set:  Created mm/dd/yy
                  Start vol. 01
                  Last  vol. nnn

Is This The Correct Backup Set? (Y/N)<Y> ====>
```

Where "mm/dd/yy" is the date of the BACKUP Operation that created the file copies in this Backup Disk Set; and

where "nnn" is the number of disks in the Backup Disk Set, and also the volume number of the last disk in the set.

If the creation date and volume number are the same as those of the Backup Disk Set being compared, then enter the letter Y or a carriage return. BRS will then prompt for file names. If the date and volume number are not those of the Disk Set you wish to compare, then N should be entered.

6.2 Entering File Names

ENTERING WINCHESTER DISK FILE NAME

After you have confirmed that the Volume 01 disk from the correct Backup Disk Set was inserted, BRS will then prompt:

```
               Compare Function
---------------------------------------------

  Enter Winchester Disk Filename ====>
```

At this prompt, you respond by entering the name of the Winchester Disk file to be compared and a carriage return. After you enter a valid file name, BRS prompts for a floppy disk file name.

INVALID WINCHESTER DISK FILE NAME ENTERED

If you specified the name of a file that is not on the currently logged Winchester Disk partition, the following error message is displayed:

```
Source File Not Found
```

ENTERING FLOPPY DISK FILE NAMES

After you have entered a valid Winchester Disk file name, BRS will display the following prompt:

```
Enter Floppy Disk Filename ====>
```

At this prompt, enter the name of the Floppy Disk file to be compared with its corresponding Winchester Disk file. Then end this entry with a carriage return.

INVALID FLOPPY DISK FILE NAME ENTERED

If you specified the name of a file that is not in the Backup Disk Set, the following error message is displayed:

```
Destination File Not Found
```

FILE NAME SPECIFICATION ERROR MESSAGE

If you specify a drive name (such as "A:") at either of the file name prompts, the following error message is displayed:

```
Drive Specifications NOT allowed.
```

6.3 COMPARE Execution

When BRS finds the specified file name in the directory of the Volume 01 Disk, it will prompt you to insert the Disk Set Volume that contains the file. The following prompt will appear:

```
Load Drive X: with Volume nn of Disk Set.
Enter RETURN when ready.
```

Where "X: " is the floppy disk drive in the H/Z-67 drive unit; and

where "nn" is the number of the volume that contains the file that you specified for the COMPARE operation.

The COMPARE operation compares each byte in both files. If the data in each file is identical, the following prompt is redisplayed:

```
              Compare Function
---------------------------------------------

Enter Winchester Disk Filename ====>
```

You can now COMPARE another file.

6.4 Compare Error

If the files do not compare exactly, BRS will notify you with a display in the following form: .

```
Data Miscompared at Record rrrr         Reading file COMPFILE.NAM

              Enter Carriage Return When Ready
```

Where "rrrr" is the location of the discrepancy between the data in the two files, expressed in a hexadecimal number; and

where "COMPFILE.NAM" represents the name of the Winchester Disk file that does not match the Floppy Disk file.

BRS will redisplay the following prompt after you enter a carriage return:

```
              Compare Function
---------------------------------------------

Enter Winchester Disk Filename ====>
```

6.5 Remedy for Dissimilar Files

If discrepancies are found between files, you should first be sure that they were supposed to have been identical in the first place. If the files were supposed to have been identical, you should determine which of the two files contains the error, and delete that file. Then you should perform the BACKUP or RESTORE operation again. You should also run the COMPARE Operation again to test the newly made file against his source file to see that they are indeed identical.

6.6 Ending the COMPARE Operation

You can end the COMPARE Operation by entering a carriage return alone at the "Enter Winchester Disk Filename" prompt or at the "Enter Floppy Disk Filename" prompt. BRS will then redisplay the Master Menu.

7 THE LIST DIRECTORY OPERATION

The LIST operation enables you to obtain the following information about a Backup Disk Set:

- creation date of set;
- number of volumes in set;
- drive name assigned to Winchester Disk partition;
- file names within set;
- user area numbers of files within set; and
- volume numbers of disk(s) on which files reside.

7.1 Selecting the LIST Operation

To select the LIST Operation, enter the letter **L** and a carriage return at the Master Menu selection prompt. (No Master Menu Options can be entered with the LIST Operation.)

```
Select Operation (X /Y/Y/Y) ====> List Directory
```

BRS will complete the operation title by displaying the characters "ist Directory" after the "L" you enter.

When this Operation is selected, BRS displays the following prompt:

```
Load Drive   X: with Volume   01 of Backup Disk Set.
              Enter Carriage Return When Ready.
```

Where "Drive X:" refers to the drive specified before the Master Menu appeared.

After you insert a disk and enters a carriage return, BRS will check to see if the inserted disk is Volume I and contains a Backup Disk Set Directory (stored in a file named MASTERBRS.DIR).

If BRS finds a directory on the disk, it will display the following information from the directory:

```
Backup Disk Set:  Created mm/dd/yy
                   Start vol. 01
                   Last vol. nnn

Is This The Correct Backup Set? (Y/N)<Y> ====>
```

Where "mm/dd/yy" is the date of the BACKUP Operation that created the file copies in this Backup Disk Set; and

where "nnn" is the number of disks in the Backup Disk Set, and also the volume number of the last disk in the set.

7.2 LIST Directory Display

After you enter the letter **Y** or a carriage return at the "Correct Backup Set?" prompt, BRS displays a "Listing of Backup Directory". This display appears in the form of the following example:

```
Listing of Backup Directory
Created: 02/31/82
Highest Volume Written: 003

   FILENAME      USER   VOLS        FILENAME      USER   VOLS
   --------      ----   ----        --------      ----   ----
C:TEST1    COM    00    01-01    C:TEST     REL    03    01-01
C:TEST     ASM    03    01-01    C:TEST     PRN    15    01-02
C:TEST     COM    07    01-01    C:FILE1    ASM    15    01-02
C:FILE2    ASM    15    02-02    C:FILE3    ASM    09    02-03
C:FILE4    ASM    15    03-03
```

If you have a terminal with a video screen, the entire display might not fit on the screen at once. In such a case, the directory listing will scroll up the screen. You can suspend the scroll by entering a ***CTRL-S***, and resume the scroll by entering another ***CTRL-S***

8. THE EXIT TO CP/M OPERATION

To select the EXIT Operation, you should enter the letter **E** and a carriage return at the Master Menu Selection prompt. (No Master Menu Options can be entered with the EXIT Operation.)

```
Select Operation (X /Y/Y/Y) ====> Exit to CP/M
```

BRS will complete the operation title by displaying the characters "xit to CP/M" after the "E".

The BRS utility will end and the CP/M operating system will return. CP/M will display the system prompt as shown:

```
A>
```

NOTE: If you performed bootstrap with the floppy disk drive of the H/Z-67 drive model prior to invoking BRS, then CP/M will display the following message when you exits from BRS:

```
Place a Bootable Disk in Drive A:
Enter Carriage Return When Ready
```

You must perform the activities described in the prompt to produce the "A>" system prompt.

9 BRS ERROR MESSAGES

```
Invalid Date Entered
```

EXPLANATION: You entered a date specifying a month value greater than 12, or a day value greater than 31. "Enter Today's Date" prompt will reappear to accept a valid entry.

```
Source File Not Found
```

EXPLANATION: You entered the name of a file that does not exist on an assigned Winchester Disk partition. Enter the name of an existing file or assign a drive name to the partition that contains the desired file (see ASSIGN).

```
Destination File Not Found
```

EXPLANATION: You entered the name of a file that does not exist on the floppy disk in the H/Z-67 drive slot. Enter the name of a file on this floppy disk.

```
Invalid file format or Contains Wildcards
```

EXPLANATION: You entered a file name that did not conform to CP/M file naming conventions, or a file name that contained wildcard characters (* or ?). Enter a valid file name without wildcard characters.

```
Invalid drive specification
```

EXPLANATION: You entered a drive name that is not possible with the CP/M Operating System. Enter a drive name within the range "A:" through "P:".

```
Drive Specifications NOT allowed
```

EXPLANATION: You answered a prompt by specifying both a drive name and a file name. You must specify the desired drive names when beginning the BRS session, before selecting an Operation from the Master Menu.

```
No Files Found For Entered Spec
```

EXPLANATION: You entered a file name that does not exist on the disk. Enter the name of an existing file.

```
Invalid User Number or Format
```

EXPLANATION: You entered a Selection File Specification with an invalid user number, or in an improper form. Enter the specification with a user number in the range 0-15.

```
Rejection Table is Full       Hit (CR) to Continue
```

EXPLANATION: You tried to enter more than 40 Rejection File Specifications. Enter a carriage return; then BRS will begin prompting for Selection File Specifications.

```
Directory is Full             Must Enter RETURN!!
```

EXPLANATION: User attempted to perform a Backup or Restore Operation involving more than 2000 file names. User must enter a carriage return to access the Master Menu, and should perform a number of separate Backup or Restore Operations involving no more than 2000 files each.

```
  Current Floppy Disk is Full
Load Drive X: with a Blank Floppy Disk
```

EXPLANATION: BRS is executing a Backup Operation, and has filled a floppy disk. User should remove full disk, insert a blank floppy disk, and enter a carriage return to continue the Operation.

```
Submit File Not Found
```

EXPLANATION: User entered the name of a batch file that does not exist on the disk or partition involved in the Operation. User should enter the name of an existing batch file.

```
Drive not Available in Current Configuration
```

EXPLANATION: User specified a partition that has not yet been assigned, or a floppy disk that is not within the H/Z-67 drive model. User should specify an assigned partition or the floppy disk drive in the H/Z-67 drive model.

```
Data Miscompared at Record nnnn     Reading File

               ****  JOB ABORTED  ****

             Fatal Errors Have Occurred !!

              Enter (CR) for Master Menu
```

EXPLANATION: Data within two compared files is not identical, and one of the discrepancies between the files occurs at the record indicated by "nnnn".

```
   ****  JOB ABORTED  ****

Fatal Errors Have Occurred !!

 Enter (CR) for Master Menu
```

EXPLANATION: BRS failed in an attempt to execute the selected Operation. Execution of the selected Operation ends. User should enter a carriage return to access the Master Menu, and repeat the Operation. If this error message occurs repeatedly, you should make a new copy of the BRS.COM file from the Distribution Disk.

```
Read Error Condition Code  nn     Reading File

                    ****  JOB ABORTED  ****

                 Fatal Errors Have Occurred !!

                  Enter (CR) for Master Menu
```

EXPLANATION: BRS failed in attempt to read data from a file. Execution of the selected Operation ends. User should enter a carriage return to access the Master Menu, and repeat the Operation. If this error message occurs repeatedly, you should make a new copy of the BRS.COM file from the Distribution Disk.

```
Write Error Occurred, Condition Code nn  Writing File
                    ****  JOB ABORTED  ****

                 Fatal Errors Have Occurred !!

                  Enter (CR) for Master Menu
```

EXPLANATION: BRS failed in an attempt to write data to a Winchester Disk partition. Execution of the selected Operation ends. User should enter a carriage return to access the Master Menu, and repeat the Operation. If this error message occurs repeatedly, you should make a new copy of the BRS.COM file from the Distribution Disk.

```
Winchester Disk is Full. Allow 32k to Run Program
                    ****  JOB ABORTED  ****

                 Fatal Errors Have Occurred !!

                  Enter (CR) for Master Menu
```

EXPLANATION: During the Backup Operation, BRS makes a Master Directory of file names on the partition that is being backed up. If the Backup Operation involves a large number of file names, then the partition might not have space for the directory. Hence you must create more empty space on the partition. If this error occurs, you should enter a carriage return to access the Master Menu, exit from BRS to the CP/M Operating System, and erase (see ERA) enough data from the partition to create at least 32 kilobytes of unused storage space (see STAT). Then you can invoke BRS again and repeat the Backup Operation. During this Operation, you should enter 2000 or fewer Selection File Specifications.

```
Wrong Disk Inserted in Drive

 Enter (CR) for Master Menu
```

EXPLANATION: BRS prompted you to insert Volume 01 (the Directory Disk), and the disk you inserted either was not a Volume 01 disk or was the Volume 01 disk of the wrong Backup Disk Set. Enter a carriage return at this prompt and the Master Menu will appear. At the Master Menu, you can select a BRS operation again.

```
                 Wrong Disk Inserted in Drive

LoadDrive     X: with Volume    nn of Backup Disk Set
              Enter Carriage Return When Ready
```

EXPLANATION: BRS prompted you to insert a specific volume (other than Volume 01) from your Backup Disk Set, and the disk you inserted either was not the specified volume or was the specified volume for a different Backup Disk Set. Insert the nnth volume of the appropriate Backup Disk Set and enter a carriage return.

BSYSGEN

The Utility that Copies the Operating System Between Disks

The BSYSGEN utility is used to transfer either part or all of the CP/M operating system to a disk, depending on the circumstances. Unlike the SYSGEN utility the BSYSGEN utility can **not** be used to copy the system kernel directly from memory to a disk after running a MOVCPMxx utility, although it can copy a file that was recorded by the SAVE command after execution of MOVCPM.

NOTE: Heath/Zenith releases of the CP/M Operating System consist of a system kernel and the file BIOS.SYS. To make a disk bootable, you must put the system kernel on the disk's system tracks and the file BIOS.SYS on the disk's file area. BSYSGEN will copy the system kernel under all circumstances, and the file BIOS.SYS under some circumstances.

BSYSGEN can be used by two methods: the BSYSGEN Prompt Method or the System Prompt Method.

1 BSYSGEN PROMPT METHOD

Under the BSYSGEN Prompt Method, you first load the BSYSGEN utility into computer memory, and then respond to BSYSGEN prompts that define the operation.

1.1 BSYSGEN Invocation

To begin under this method, type the following command at the system prompt:

A>**B SYSGEN** ***RETURN***

The following display will appear:

```
BSYSGEN VER 2.0.04
SOURCE DRIVE NAME :
```

1.2 Specifying the Source

At the "SOURCE DRIVE NAME :" prompt, you can specify the drive containing the disk from which the system will be copied. Enter the letter that stands for that drive.

The following example shows how you would answer this prompt if the source of the system was to be the disk in drive A:

```
SOURCE DRIVE NAME (OR RETURN TO SKIP): A
```

NOTE: BSYSGEN can only copy the system between disks of the same type. Therefore, you can **not** enter a carriage return at this BSYSGEN prompt to copy a system that has been moved into computer memory by a MOVCPMxx utility. (If you do wish to copy the system from memory after a MOVCPMxx activity, use the SYSGEN utility.)

BSYSGEN will now prompt you to confirm your selection of the source drive, with a prompt in the following form:

```
SOURCE ON A, THEN TYPE RETURN:
```

You can confirm your specification of the source drive name by entering a carriage return at this prompt. You can also abort the BSYSGEN operation and return control to the operating system by holding down the ***CTRL*** key and pressing the ***C*** key.

If you confirm the "SOURCE ON" prompt with a carriage return, BSYSGEN will then display the message:

```
FUNCTION COMPLETE
COPY BIOS.SYS (Y/N):
```

1.3 Copying BIOS.SYS with BSYSGEN

To instruct BSYSGEN to copy the file BIOS.SYS from the source disk to the destination disk, press **Y** at the "COPY BIOS.SYS (Y/N):" prompt. If you do **not** wish to copy BIOS.SYS, press **N**.

NOTE: If you decline to copy the BIOS.SYS file using the BSYSGEN utility, you can copy it to a disk using the PIP utility or record a new one on a disk using the MAKEBIOS utility.

If you pressed Y to copy BIOS.SYS, BSYSGEN will display the message "FUNCTION COMPLETE", and then prompt for destination. If you pressed N to forgo the copying of BIOS.SYS, BSYSGEN will immediately prompt for destination.

1.4 Specifying the Destination

A few seconds after you have made an entry at the "COPY BIOS.SYS" prompt, BSYSGEN will prompt for destination as shown:

```
DESTINATION DRIVE NAME (OR RETURN TO REBOOT):
```

The first time this prompt appears, you should type the drive letter for the disk that you wish to receive the system. For instance, type **B**. BSYSGEN would then display a prompt in the following form:

```
DESTINATION ON B, THEN TYPE RETURN
```

Confirm your destination specification by entering a carriage return at such a prompt.

BSYSGEN will again display the "DESTINATION DRIVE NAME (OR RETURN TO REBOOT):" prompt. This time, you can specify a different drive name, insert a new disk into the former destination drive and specify this drive again as the destination, or enter a carriage return to cause a warm boot. (A warm boot will exit you from the BSYSGEN utility to the CP/M Operating System. Then a system prompt will be displayed.)

2 SYSTEM PROMPT METHOD

The System Prompt Method enables you to enter all of the specifications necessary for a BSYSGEN operation in a single command line entered at the CP/M system prompt.

2.1 Command Line Entry

System Prompt Method BSYSGEN commands are entered in the following form:

A>**BSYSGEN** {destination}={**source**}{**[option,option]**} ***RETURN***

Where **BSYSGEN** is the command line function, stored in the file BSYSGEN.COM on the logged disk;

where {**destination**} is the name of the drive that contains the disk receiving the copy of the system;

where {**source**} can be either a drive name, a file name, or both; and

where {**[option,option]**} represents letters enclosed in square brackets [] and separated by a comma , to specify how the BSYSGEN operation should be conducted.

NOTE: In a command line equation, the data source is always on the right and the data destination is always on the left.

2.2 System Sources

The system source can be one of the following four types:

- Drive Name, including a letter for a drive within your hardware environment and a colon, as with **A:**, **B:**, **C:**, **D:**, **E:**, **or F:**;
- File Name, which specifies a file that was created and stored by consecutive MOVCPMxx and SAVE commands, as with **CPM32.SYS**, **CPM48.SYS**, or **CPM64.SYS;** or
- Drive Name and File Name, where the file desired for the system kernel source resides on a disk in a non-default drive and the drive name must specify that drive, as with **B:CPM48.SYS** or **C:CPM64.SYS**.

2.3 BSYSGEN Options and Defaults

BSYSGEN command lines entered by the System Prompt Method can include the following options (enclosed in square brackets []):

B BIOS.SYS file will be copied with the system kernel, from the specified source to the specified destination. If a file named BIOS.SYS already exists on the destination disk, it will be overwritten.

N No prompt will be displayed during this operation.

When you enter a BSYSGEN command line with source and destination specifications, and neglect to specify options, BSYSGEN will perform the operation according to these default criteria:

- BIOS.SYS file will **not** be copied, regardless of the source of the rest of the system (unless option B is entered when source is not memory).
- Prompt **will** be displayed to confirm which drive will receive the copy of the system (unless option N is entered). The BSYSGEN prompt appears as shown:

```
BSYSGEN VER 2.0.04

DESTINATION ON B, THEN TYPE RETURN
```

2.4 System Prompt Method Examples

A>**BSYSGEN B:=A:** ***RETURN***

BSYSGEN will copy the system kernel from the disk in drive A to the disk in drive B. The BIOS.SYS file from A will **not** be copied and a prompt **will** appear before the copying, by default.

A>**BSYSGEN B:=D:[N]** ***RETURN***

BSYSGEN will copy the system kernel from the disk in drive D to the system tracks of the disk in drive B. The BIOS.SYS file will **not** be copied, by default. A prompt will **not** appear before the copying, as specified by the N option.

```
A>D:BSYSGEN B:=C:CPM64.SYS[B,N]   RETURN
```

The BSYSGEN utility, in this case, is stored on the disk in non-default drive D. It will copy the system kernel from the file named "CPM96.SYS" (recorded onto the disk in drive C by the SAVE command after creation in memory by the MOVCPMxx command), and put it on the system tracks of the disk in drive B. It will also copy the BIOS.SYS file from drive C to drive B, and display no prompts during the operation, as specified by options.

3 BSYSGEN ERROR MESSAGES

```
INVALID DRIVE NAME
```

EXPLANATION: You must specify drive names using the names of drives that exist in the hardware environment, and are recognized by the operating system that was loaded at bootstrap.

```
NO SOURCE FILE ON DISK
```

EXPLANATION: The drive specified as "SOURCE DRIVE" did not contain the file BIOS.SYS. You should use a different disk in the source drive, or rename a BIOS file that has been given a different name to "BIOS.SYS".

```
SOURCE FILE INCOMPLETE
```

EXPLANATION: BSYSGEN failed in an attempt to copy the file BIOS.SYS from the disk in the source drive. This file might have been damaged by disk media flaws or partially overwritten. Reset, perform bootstrap, and re-enter the BSYSGEN command using a different disk in the source drive.

```
WRITE ERROR DURING BIOS.SYS
```

EXPLANATION: Try BSYSGEN again with a destination disk that is write enabled, formatted, and has at least 6 kilobytes of free space.

```
ERROR READING BIOS.SYS
```

EXPLANATION: BSYSGEN failed in an attempt to copy the file BIOS.SYS from the disk in the source drive. This file might have been damaged by disk media flaws or partially overwritten. Reset, perform bootstrap, and re-enter the BSYSGEN command using a different disk in the source drive or using a different disk to perform bootstrap.

```
PERMANENT ERROR, TYPE RETURN TO IGNORE
```

EXPLANATION: The system kernel or BIOS.SYS file are either incompatible with the destination disk type or otherwise flawed. Reset, perform bootstrap, and re-enter the BSYSGEN command using a different disk in the source drive or using a different disk to perform bootstrap. Under some circumstances, you must use a MOVCPM utility before BSYSGEN.

```
UNABLE TO SELECT DRIVE
```

EXPLANATION: Specify the name of a drive that can be accessed by BSYSGEN. Such a drive must be a valid drive that is recognized by the operating system.

```
COMMAND SYNTAX ERROR
```

EXPLANATION: System Prompt Method command line was entered without following the entry form explained in "2.1 Command Line Entry". Enter command again after reviewing this entry form.

```
ILLEGAL OPTION
```

EXPLANATION: System Prompt Method command line was entered with an option other than a B or an N. Re-enter command with either, none, or all of the BSYSGEN options B and N. Enclose the option(s) in square brackets and separate them with a comma if both are used.

CONFIGUR

The Utility that Customizes the CP/M Operating System for Several Characteristics of the Hardware Environment

The CONFIGUR utility helps you to change the CP/M Operating System so that it will control specific hardware devices.

When invoked (Section 1), CONFIGUR offers you two methods for changing the system: the Standard System Method (Section 2) or the Menu Selection Method (Section 3). The former method automatically changes the operating system to accommodate only certain devices. The latter method enables you to type selections at menus to change the system to accommodate a wide variety of hardware characteristics and/or your preferences (Sections 4 through 7).

CONFIGUR is usually run during the first session of CP/M use in a particular hardware environment. But it should also be run whenever a component of the hardware environment is added or changed.

NOTE: CONFIGUR always makes some changes to the image of the CP/M Operating System that was loaded into computer memory during bootstrap, but these changes are only recorded on the disk if the disk is not write protected. If the disk is write protected, then the changes will only remain in effect until the computer is reset.

Changing only the system image in memory and not the system on the disk is an effective way to test the changes that CONFIGUR makes to the system before recording these changes on a disk.

1 CONFIGUR INVOCATION

CONFIGUR is invoked by responding to the system prompt with a command line in the following form:

```
A>CONFIGUR   RETURN
```

NOTE: Heath/Zenith CP/M Distribution Disks are preset at the factory so that the CONFIGUR utility will be invoked automatically when you perform bootstrap with the write-protected distribution disks. This automatic invocation feature can be changed (Section 7).

When invoked, CONFIGUR identifies itself with a display in the following form:

```
Heath/Zenith Configuration Program
Version 2.2.04
Serial Number: sss-ssss

This program configures the CP/M operating
system to a particular hardware environment.

Please wait during hardware verification...
```

At this point, the CONFIGUR utility will probe parts of your hardware environment. After this probe, CONFIGUR displays some of the characteristics of some of your hardware devices, for example:

```
H/Z89 with 48k of random access memory (RAM)
03 H/Z17 minifloppy drive(s)
CRT baud rate is 9600
00 additional serial ports found
```

This display differs depending upon the assortment of devices in your hardware environment. However CONFIGUR always notes the type of microcomputer, the terminal baud rate, and at least one type of disk drive.

If the disk with which you performed bootstrap is write protected, then CONFIGUR cannot make any changes to the operating system recorded on this disk, and displays the following message:

```
Drive A disk is write-protected.
Modifications will NOT be made to this disk for this CONFIGUR run.
```

The bottom line of this display is a prompt that appears as follows:

```
Standard system (Y or N)? <Y>:
```

This prompt presents you with the choice of changing the system through the Standard System Method (Section 2) or through the Menu Selection Method (Section 3).

2 STANDARD SYSTEM METHOD

CONFIGUR can automatically apply limited changes to the operating system to allow the system to control your terminal, at least one type of disk drive, and a particular type of printer.

These standard system changes will be made if you enter the letter **Y** or a carriage return at the standard system prompt, as shown:

```
Standard system (Y or N)? <Y>: Y
```

The automatic standard system changes adjust the operating system so that it recognizes the hardware characteristics CONFIGUR found in its limited probe of the hardware environment.

In addition, the standard system changes adjust the operating system to recognize the following other criteria:

- A CRT: device set at a baud rate of 9600, will transfer data through port 350Q, with no nulls after carriage returns;
- A TTY: device set at a baud rate of 300, will transfer data through port 320Q, with no nulls after carriage returns, and without translating all letters to uppercase;
- A modem set at a baud rate of 300, will transfer data through port 330Q, with no nulls after carriage returns, and without translating all letters to uppercase;
- A printer set at a baud rate of 4800, will transfer data through port 340Q, with no nulls after carriage returns, and without printing all letters in uppercase;
- Use of the DELETE key will not cause characters to be echoed (repeated) on the console display;
- Serial Printer ready signal polarity will be set "LOW";
- Serial Printer ready signal will be carried by "RTS (Pin 4)";
- Parallel Printer ready signal polarity will be set "HIGH";
- A Z89-11 interface card's LPT device will facilitate "PARALLEL" data transmission;
- 5.25-inch disks will write 48 tracks per inch (tpi);
- 5.25-inch disk step rates are set to 30ms;
- Detailed disk error messages will not appear;
- The logical/physical device pairings (IOBYTE) are set as CON: = CRT:, RDR: = UR1:, PUN: = UP1:, and LST: = LPT:; and
- No program will be automatically invoked after the next warm or cold boot.

After you type **Y** at the "Standard system" prompt, CP/M immediately displays the system prompt.

3 MENU SELECTION METHOD

CONFIGUR also enables you to change the operating system through the Menu Selection Method, by typing selections that are listed on a series of menus.

Actually, CONFIGUR has already made a few automatic changes to reflect the hardware characteristics it found in its limited probe of your environment. But most users will benefit from making some additional changes through CONFIGUR menus. To begin making these changes, you should refuse the changes offered at the "Standard system" prompt by typing the letter **N**, as shown:

```
Standard system (Y or N)? <Y>: N
```

When an **N** is typed at this prompt, CONFIGUR will display the main menu, as shown:

```
CP/M Configuration

A    Set Terminal and Printer Characteristics
B    Set Disk Parameters
C    Change the Default I/O Configuration
D    Automatic Program Control

X    Configure, making changes to memory only
Y    Configure, making changes to both memory and disk
Z    Quit, making no changes
```

From CONFIGUR's main menu (shown above) you can access a submenu by typing one of the letters **A, B, C,** or **D**; or return to the system prompt by typing **X, Y,** or **Z**. Each submenu enables you to select changes or return to the main menu.

NOTE: If the disk you booted up with is write protected, the "Y" selection will not appear and you will not be able to make "changes to both memory and disk".

3.1 CONFIGUR Submenu Options

The following key describes the different uses for each of CONFIGUR's four submenus.

Submenu A Option:

- To change the system baud rate and/or port address setting for a video terminal, printing terminal, printer, input modem or output modem;
- to change the treatment of lowercase letters sent to a video terminal, printing terminal, or printer;
- to change the number of nulls sent after each carriage return to a video terminal, printing terminal, or printer;
- to change the way a terminal treats characters that are deleted with the "DELETE" key;
- to change the voltage level (polarity) and/or the connection pin used for the "printer ready signal" between the terminal and either a serial printer or a parallel printer;
- to determine whether a Z89-11 interface card will use the LPT physical device for parallel or serial data transmission; and/or
- to return to the main menu while saving or ignoring the changes selected at this submenu.

Submenu B Option:

- To change the step rate of 5.25-inch disk drives;
- to change the number of tracks per inch (tpi) used to store data on 5.25-inch soft-sectored disk drives;
- to turn on/off the display of detailed disk error messages; and/or
- to return to the main menu while saving or ignoring the changes selected at this submenu.

Submenu C Option:

- To change the match-up of logical input/output devices with physical input/output devices; and/or
- to return to the main menu while saving or ignoring the changes selected at this submenu.

Submenu D Option:

- To turn on/off the automatic invocation of a command;
- to change the text of the command that should be invoked after every cold boot and/or after every warm boot; and/or
- to return to the main menu while saving or ignoring the changes selected at this submenu.

3.2 CONFIGUR Exit Options

By typing one of the **X, Y,** or **Z** options at the bottom of the main menu, you can exit from CONFIGUR to the CP/M Operating System. The following key describes the other functions of each exit option.

Exit Option X: You can apply the selected changes **only to the system in computer memory** by typing the letter **X** at the main menu.

Exit Option Y: You can apply the selected changes **both to the system in memory and the system on the disk** by typing **Y** at the main menu. However, this option will not be displayed if the disk used to perform bootstrap was write protected when CONFIGUR was invoked. Changes entered through CONFIGUR cannot be recorded on a write-protected disk.

Exit Option Z: You can **nullify all changes** selected at all of the submenus by typing **Z.**

When either X, Y, or Z is used to exit from CONFIGUR, the CP/M Operating System will display the system prompt beneath the main menu.

NOTE: The CONFIGUR utility can detect many invalid entries. If you make an invalid entry, the terminal might emit audible beeps or refuse to display a selection. In such a case, you should re-enter the selection.

4 SUBMENU A: TERMINAL/PRINTER CHARACTERISTICS

Typing **A** at the main menu selection prompt will cause CONFIGUR to display the following submenu:

```
A    CRT:  Baud rate: 9600 Port: 0E8H = 350Q
B    TTY:  Baud rate:  300 Port: 0D0H = 320Q
C    LST:  Baud rate: 4800 Port: 0E0H = 340Q
D    UR1:  Baud rate:  300 Port: 0D8H = 330Q
E    UP1:  Baud rate:  300 Port: 0D8H = 330Q

F    Force output to upper case on CRT: TRUE
G    Force output to upper case on TTY: FALSE
H    Force output to upper case on LST: FALSE
I    Nulls outputted after CR on CRT: 0
J    Nulls outputted after CR on TTY: 0
K    Nulls outputted after CR on LST: 0
L    Echo on DELETE:  TRUE

M    Serial Printer Ready Signal Polarity <HIGH,LOW>: LOW
N    Serial Printer Ready Signal <DTR (Pin20),RTS (Pin4)>: RTS
O    Parallel Printer Ready Signal Polarity <HIGH,LOW>: HIGH
P    Z89-11 LPT Selection <PARALLEL,SERIAL>: PARALLEL

Y    Finished, make changes and return to main menu
Z    Quit, make no changes and return to main menu

Selection:
```

(This sample display may differ from the displays of some users.)

The selections in submenu A enable you to do the following:

- Change the system baud rate and/or port address setting for a video terminal, printing terminal, printer, input modem or output modem;
- change the treatment of lowercase letters sent to a video terminal, printing terminal, or printer;
- change the number of nulls sent after each carriage return to a video terminal, printing terminal, or printer;
- change the way a terminal treats characters deleted with the "DELETE" key;
- change the properties of the printer ready signal; and
- change the voltage levels (polarity) and/or the connection pins used for the "printer ready signals";
- select whether a Z89-11 interface card will use the LPT physical device for parallel or serial data transmission; and/or
- return to the main menu while saving or ignoring the changes selected at this submenu.

This text guides you through these operating system changes.

4.1 Change Baud Rate and/or Port Address

(Selections A, B, C, D, and E)

This section explains how to make baud rate and port address changes to physical devices. Users unfamiliar with these terms might benefit from reading the following explanations.

PHYSICAL DEVICE NAMES

The four-character names in selections A through E (such as CRT: and TTY:) are physical device names. Physical Device Names are abbreviations by which you will identify a particular kind of hardware device, as shown:

"CRT:" refers to video terminals (Cathode Ray Tube)
"TTY:" refers to printing terminals (TeleTYpe)
"LST:" refers to line printers (LiSTing device)*
"UR1:" refers to input modems (User-defined Reader 1)
"UP1:" refers to output modems (User-defined Punch 1)

BAUD RATE

Baud rate is the speed with which a physical device transmits or receives data. Since the CP/M operating system is responsible for controlling the passage of data between your devices, it must know how fast these devices are able to pass data.

The baud rate for a particular physical device is displayed on the line with that device's physical name. (Valid baud rates are: 38400, 19200, 9600, 4800, 2400, 1800, 1200, 600, 300, 150, 134, 110, and 75.)

PORT ADDRESSES

The CP/M operating system sends and receives data through electrical connections known as ports. A port address enables CP/M to locate a specific port through which it can access a device.

In the submenu A selections, port addresses are expressed in values from two different numbering systems: hexadecimal and octal. When the port address value is entered, it is in the form of either a hexadecimal number or an octal number. Some valid port values are: 320Q = 0D0H, 330Q = 0D8H, 340Q = 0E0H, and 350Q = 0E8H.

* Actually, "LST:" is not a physical device name, but a logical device name. This distinction will be explained in Section 6. During submenu A operations, LST: should be treated as though it were a physical device name.

HOW TO CHANGE BAUD RATES AND PORT ADDRESSES

To change a baud rate and/or port address, you should first determine what physical device names apply to the machines within the hardware environment (e.g. terminal, printer, modem). Considering each machine, one at a time, you should type the letter (A, B, C, D, or E) that corresponds to that machine.

```
A    CRT:  Baud rate: 9600 Port: 0E8H = 350Q
B    TTY:  Baud rate:  300 Port: 0D0H = 320Q
C    LST:  Baud rate: 4800 Port: 0E0H = 340Q
D    UR1:  Baud rate:  300 Port: 0D8H = 330Q
E    UP1:  Baud rate:  300 Port: 0D8H = 330Q
```

This portion of submenu. A (as it appears on your terminal) display shows the baud rate and terminal values currently used by the system. If a particular device does not currently work as it should, you should change it's baud rate to match the rate listed in Table 2-1. If a machine being used is not listed in Table 2-1, you should refer to that machine's manual.

If you have a Heath/Zenith microcomputer, then it should not be necessary to change the port address values. If you have a microcomputer other than one furnished by Heath/Zenith, then you should refer to the manual of the microcomputer used to determine the proper port addresses for each device.

You can start a change by typing one of the selection letters (**A**, **B**, **C**, or **D**). CONFIGUR will first display the corresponding physical device name and prompt you to enter the baud rate, as shown in the following example:

```
LST Baud rate:
```

The baud rate numbers must be entered carefully, because CONFIGUR anticipates certain baud rate values, and usually only needs to know the first digit or two before it assumes and displays the entire number.

Immediately after CONFIGUR accepts the baud rate value, it displays the word "Port:" and waits for you to enter the port address value, as shown:

```
    LST Baud rate: 9600  Port:
```

This value must be entered in hexadecimal or octal values, but the "H" or "Q" radix symbol that usually accompanies values in these number systems is not necessary. If you do not wish to change the port address, a carriage return can be entered at this prompt.

When an entry is made at the "Port:" prompt, CONFIGUR will redisplay submenu A with any new values you entered.

NOTE: CONFIGUR changes only the operating system's **expectations** of a particular baud rate for a particular device. CONFIGUR does **not** change the baud rate of the device itself. The baud rate of a peripheral device can sometimes be changed according to instructions in the device's hardware manual.

If you wish to change the baud rate of the CRT: or TTY: (when one of these devices is used as the terminal), change the CRT: or TTY: value in submenu A first. Then, after exiting from CONFIGUR, manually change the baud rate of the terminal to match the selection made through CONFIGUR. (If the baud rate had been manually changed on the terminal first, you will be unable to use the terminal to communicate with CP/M or CONFIGUR.)

This table illustrates the selection letter, baud rate, and port address you should enter for a particular hardware item. Some baud rates are subject to change. A device's hardware manual should be consulted to verify baud rates.

SELECTION LETTER	DEVICE NAME	DEVICE DESCRIPTION	INTERFACE	HEATH/ZENITH CATALOG NAME	BAUD RATE	PORT ADDRESS
A	CRT:	video terminal	Z-89-3	H/Z-19	9600	0E8H = 350Q
			Z-89-3	H/Z-89	9600	0E8H = 350Q
			Z-89-3	H/Z-90	9600	0E8H = 350Q
			Z-89-11	H/Z-19	9600	0E8H = 350Q
			Z-89-11	H/Z-89	9600	0E8H = 350Q
			Z-89-11	H/Z-90	9600	0E8H = 350Q
B	TTY:	printing terminal	Z89-3	LA-34 DECwriter (WH-34)	300	0D0H = 320Q
			Z89-3	LA-36 DECwriter (WH-36)	300	0D0H = 320Q
			Z89-3	Diablo KSR 1640	1200	0D0H = 320Q
			Z89-11	LA-34 DECwriter (WH-34)	300	0D8H = 330Q
			Z89-11	LA-36 DECwriter (WH-36)	300	0D8H = 330Q
			Z89-11	Diablo KSR 1640	1200	0D8H = 330Q
C	LST:	line printer	Z89-3	Heath H-14	4800	0E0H = 340Q
			Z89-3	Texas Instruments TI-810 (WH-24)	4800	0E0H = 340Q
			Z89-3	Heath/Zenith H/Z-25	4800	0E0H = 340Q
			Z89-3	LA-36 DECwriter (WH-36)	300	0E0H = 340Q
			Z89-3	Diablo 1640 (WH-44)	1200	0E0H = 340Q
			Z89-3	Diablo KSR 1640 (WH-44K)	1200	0E0H = 340Q
			Z89-3	Diablo 630 (WH-54)	1200	0E0H = 340Q
			Z89-3	Diablo 1610/1620/1650	1200	0E0H = 340Q
			Z89-3	Epson MX-80 (serial)	4800	0E0H = 340Q
			Z89-11	Heath H-14	4800	0E0H = 340Q
			Z89-11	Texas Instruments TI-810 (WH-24)	4800	0E0H = 340Q
			Z89-11	Heath/Zenith H/Z-25	4800	0E0H = 340Q
			Z89-11	LA-36 DECwriter (WH-36)	300	0E0H = 340Q
			Z89-11	Diablo 1640 (WH-44)	1200	0E0H = 340Q
			Z89-11	Diablo KSR 1640 (WH-44K)	1200	0E0H = 340Q
			Z89-11	Diablo 630 (WH-54)	1200	0E0H = 340Q
			Z89-11	Diablo 1610/1620/1650	1200	0E0H = 340Q
			Z89-11	Epson MX-80 (serial)	4800	0E0H = 340Q
			Z89-11	Epson MX-80 (parallel)	none	not applicable
D	UR1:	input modem		Novation CAT Acoustic (WH-13)	300	0D8H = 330Q
				Lexicon LEX-11 (WH-23)	300	0D8H = 330Q
and	and	and		UDS Direct Connect (WH-23)	300	0D8H = 330Q
E	UP1:	output modem		Hayes SMARTMODEM (WH-43)	300	0D8H = 330Q

Table 2-1
Submenu A Baud/Port Values

4.2 Change Treatment of Lowercase Letters
(Selections F, G, and H)

This portion of submenu A involves three devices. These selections enable you to change the operating system so that it translates lowercase letters into uppercase letters before they are sent to an output device such as a video terminal (CRT:), printing terminal (TTY:), or printer (LST:).

```
     F     Force output to upper case on CRT: FALSE
     G     Force output to upper case on TTY: FALSE
     H     Force output to upper case on LST: FALSE
```

These three selections offer you only two alternatives: "TRUE" and "FALSE". Therefore, you merely have to type one of the letters (**F**, **G**, or **H**) to cause the opposite value to be displayed for that device.

When one of the displayed lines for a device ends with the word "TRUE", then CP/M will automatically translate all lowercase letters sent to that device into upper case (if you choose to save these changes to the system).

This type of letter translation is necessary for devices that can produce only uppercase letters (such as the H9 video terminal). You should consult a device's hardware manual to determine whether it can produce lowercase letters.

4.3 Change Number of Nulls Sent After Carriage Returns

(Selections I, J, and K)

This portion of submenu A involves three devices. These selections enable you to change the operating system so that it sends a specified number of null characters to a particular device after each carriage return that is sent to that device.

```
I    Nulls outputted after CR on CRT: 0
J    Nulls outputted after CR on TTY: 0
K    Nulls outputted after CR on LST: 0
```

You can start a change to the number of nulls by typing one of the selection letters (**I**, **J**, or **K**). CONFIGUR will display a prompt for the selected device, as shown:

```
Nulls after CR on LST :
```

You should then answer this prompt with a number from zero to seven, depending on the number of nulls the device requires. CONFIGUR will immediately redisplay submenu A, showing any changes you made to the null settings.

Video terminals, printing terminals, and printers sold by Heath/Zenith require zero nulls after each carriage return. Users with devices not obtained through Heath/Zenith should consult the devices' hardware manuals to determine the number of nulls the devices requires.

4.4 Change Terminal Treatment of Deleted Characters

(Selection L)

The "L" selection from submenu A involves the video terminal (CRT:) or printing terminal (TTY:) device. These selections enable you to change the operating system so that it causes the terminal to redisplay characters deleted with the "DELETE" key ("DLT" key on some terminals). The display will be activated if the "L" line ends with the word "TRUE".

```
L    Echo on DELETE:  TRUE
```

This selection offers you only two alternatives values: "TRUE" and "FALSE". Therefore, you merely have to type the letter **L** to cause a different alternative to be applied to the operating system (if you choose to save these changes). When you type **L**, submenu A is redisplayed with the new value.

The "Echo on DELETE" feature is often desired by users of printing terminals, but seldom desired by users of video terminals that can back space.

NOTE: This feature is automatically put into effect (becomes "TRUE") whenever the MOVCPMxx and SYSGEN utilities are run, and it is disabled (becomes "FALSE") when **any** changes made by the CONFIGUR utility are saved to memory or to disk.

4.5 Change Properties of Printer Ready Signal *(Selections M, N, O, and P)*

This portion of submenu A enables you to change the operating system's expectations of the Printer Ready Signal and the ports used to communicate with a printer through the Z89-11 interface card.

```
M    Serial Printer Ready Signal Polarity <HIGH,LOW>: LOW
N    Serial Printer Ready Signal <DTR (Pin 20),RTS (Pin 4)>: RTS
O    Parallel Printer Ready Signal Polarity <HIGH,LOW>: HIGH
P    Z89-11 LPT Selection <PARALLEL,SERIAL>: PARALLEL
```

These selections involve the electrical signal that a printer sends, through a cable, to the computer when it is ready to print more data. This signal is called the "printer ready signal".

The printer ready signal is sent at a certain "polarity" (level of voltage). It can be sent at a "HIGH" voltage level or a "LOW" voltage level, depending on the kind of printer being used.

By typing **M**, the user of a **serial** printer can change the operating system's expectations of a particular voltage level from a serial printer.

Most serial printers available through Heath/Zenith send this signal at the "LOW" voltage level. However, to determine appropriate voltage level setting for a particular printer, you should refer to the printer's manual.

The serial printer ready signal travels through a special cable identified by the model number "RS-232C". This cable contains a bundle of wires, each connecting a pair of metal "pins". (One member of this pin pair plugs into the computer, and the other member plugs into the printer.) Because each pin carries a specific kind of signal between the printer and the computer, each pin is numbered. And the pin that helps transmit the printer ready signal is either pin number 20 or pin number 4, depending on the kind of printer being used.

By typing **N**, the user of a serial printer can change the operating system's expectations of which numbered pin will transmit the printer ready signal from a serial printer. The system will expect this signal to come in either the Data Terminal Ready fashion (DTR through Pin 20), or in the Ready To Send fashion (RTS through Pin 4) To determine which pin in the RS-232C cable a particular serial printer uses for the printer ready signal, refer to the serial printer's manual.

By typing **O**, the users of a parallel printer can change the operating system's expectations of a particular voltage level from a parallel printer.

Most parallel printers available through Heath/Zenith send this signal at the "HIGH" voltage level. However, to determine appropriate voltage level setting for a particular printer, refer to the printer's manual.

The Z89-11 interface card (available with this CP/M version) is capable of data transmission to and from either serial or parallel printers through the LPT physical device. However, the type of printer used must be specified through the P selection.

By typing **P**, you can change the specification of which kind of printer (serial or parallel) will be used.

When you type the letter for any of these four selections, the opposite value goes into effect (if you save this change). In addition, CONFIGUR redisplays submenu A with the newly selected values in place.

4.6 Return to Main Menu
(Selections Y and Z)

These submenu A selections enable you to exit from the submenu to the main menu.

```
Y    Finished, make changes and return to main menu
Z    Quit, make no changes and return to main menu
```

By typing **Y**, you also instruct CONFIGUR to preserve the selections entered at submenu A (although none of the changes from any submenu can be applied to the operating system until you exit from the CONFIGUR utility entirely).

By typing **Z**, you instruct CONFIGUR to abandon any changes entered at submenu A.

After either of these entries, CONFIGUR will redisplay the main menu, as shown:

```
CP/M Configuration

A    Set Terminal and Printer Characteristics
B    Set Disk Parameters
C    Change the Default I/O Configuration
D    Automatic Program Control

X    Configure, making changes to memory only
Y    Configure, making changes to both memory and disk
Z    Quit, making no changes
```

You can now access a submenu or exit from CONFIGUR entirely.

NOTE: If the disk being configured is write protected, the Y selection from the main menu will not be available, and changes cannot be recorded on disk.

5 SUBMENU B: SET DISK PARAMETERS

Typing **B** at the main menu selection prompt will cause CONFIGUR to display the following submenu:

```
A     5.25 Inch Soft Sectored Unit 0  Step Rate: 30ms  Track Density: 48tpi
B     5.25 Inch Soft Sectored Unit 1  Step Rate: 30ms  Track Density: 48tpi
C     5.25 Inch Soft Sectored Unit 2  Step Rate: 30ms  Track Density: 48tpi

D     5.25 Inch Hard Sectored Unit 0 **undefined**
E     5.25 Inch Hard Sectored Unit 1 **undefined**
F     5.25 Inch Hard Sectored Unit 2 **undefined**
G     Detailed Disk Error Messages:  FALSE

Y     Finished, make changes and return to main menu
Z     Quit, make no changes and return to main menu
```

(This display might differ slightly for some users.)

The selections in submenu B enable you to do the following:

- Change the step rate of 5.25-inch disk drives;
- change the number of tracks per inch (TPI) for 5.25-inch soft-sectored disk drives;
- turn on/off the display of detailed disk error messages; and/or
- return to the main menu while saving or ignoring the changes selected at this submenu.

5.1 Change 5.25-inch Disk Step Rate

This text explains how you can change the step rate and/or track density for 5.25-inch disk drives. Users unfamiliar with disk drive technology might benefit from the brief explanations provided here.

DISK DRIVE UNIT

A disk drive unit is a single physical disk drive, or one of the drive slots visible to you. Each of the 5.25-inch disk drive slots (within each disk drive group) are given distinct unit numbers from zero through two.

CONFIGUR automatically tries to find out how many disk drive units are connected to the computer when it probes the hardware environment. If CONFIGUR finds that an environment contains a particular unit, it will reflect these findings by displaying a step rate (ms) value next to that unit in submenu B. If CONFIGUR does not find a particular unit, then CONFIGUR will display the word "**undefined**" next to the corresponding unit in submenu B.

NOTE: If you have both hard-sectored and soft-sectored 5.25-inch drives, and have not yet run the MAKEBIOS program for the operating system in use, then this operating system only acknowledges one type of 5.25-inch drive in the hardware environment. Therefore, CONFIGUR could only find one type of drive during its hardware probe, and submenu B will show all three units for one type of drive as "**undefined* *".

STEP RATE

Step rate is the rate of speed at which the read/write head moves from one track of a disk to another while reading or writing data. Step rates are measured in thousandths of a second, or milliseconds (ms).

Step rates can range from 4 to 36 milliseconds, where 4ms is faster than 36ms. Only even numbered step rates can be entered. A faster step rate can hasten the transfer of data to and from the disk, but it might also cause the read/write head to move across the disk too fast to transfer data reliably.

With CP/M Version 2.2.04, Heath/Zenith recommends that the step rate be left at the preset step rate of 30ms for most work with 5.25-inch hard-sectored disks (which run under the H17 controller card).

Heath/Zenith recommends that 5.25-inch soft-sectored drives (which run under the H37 controller card) be set to operate at a step rate of 30ms when the disks used are 48 tpi, and 6ms when the disks used are 96 tpi. ("tpi" is an acronym meaning "tracks per inch".) Users who change the operating system to run disks at other step rate should first read the text on Step Rate Error Checking.

The step rate for 8-inch disks cannot be changed through CONFIGUR.

STEP RATE ERROR CHECKING

To check the current disk error rate, use the DDT utility (Dynamic Debugging Tool) and examine the value of the SECNT label in the BIOS that is active in memory. You can determine the address at which this value exists, by assembling BIOS.ASM, producing a BIOS.PRN file during the assembly, and analyzing the BIOS.PRN file for the address of the SECNT label you wish to check. The SECNT value shows the number of soft (recoverable) errors that have occurred in read/write operations with the disk. This value needs to be maintained at a relatively low level. If the value becomes too large, the step rate should be increased so that the read/write head is allowed additional time to step from track to track while reading or writing.

TRACK DENSITY

There are two types of 5.25-inch soft-sectored disk that can be used in an H/Z-37 drive unit: a "48 tpi disk" and a "96 tpi disk". The distinction between the two lies in their track density, or the number of tracks per inch (tpi) that cover the surface of the disk.

The 48 tpi disk has a total of 40 tracks, while the 96 tpi disk has a total of 80 tracks. Each H/Z-37 drive unit must be set to accommodate the type of disk used in it. This setting will determine the manner in which the FORMAT utility prepares disks in that drive, and thus the manner in which data is stored on these disks.

HOW TO CHANGE STEP RATES AND/OR TRACK DENSITIES

To change the step rate for a 5.25-inch disk, you must first type the selection letter (**A, B, C, D, E,** or **F**) for a disk drive unit that currently has a step rate number listed next to it.

```
A    5.25 Inch Soft Sectored Unit 0  Step Rate: 30ms  Track Density: 48tpi
B    5.25 Inch Soft Sectored Unit 1  Step Rate:  30ms Track Density: 48tpi
C    5.25 Inch Soft Sectored Unit 2  Step Rate:  30ms Track Density: 48tpi

D    5.25 Inch Hard Sectored Unit 0 **undefined**
E    5.25 Inch Hard Sectored Unit 1 **undefined**
F    5.25 Inch Hard Sectored Unit 2 **undefined**
```

NOTE: The selection letters in submenu B do not necessarily correspond to the drive name letters of the drive units.

When you type a selection letter, CONFIGUR will display a prompt in the following form:

```
Soft Sector Unit 0  Step Rate ?
```

You can now enter a number for the new step rate of the selected drive unit, and a carriage return. If you selected a hard-sectored drive unit, then any even number from 4 through 36 can be entered (at your own risk). If you selected a soft-sectored drive unit, then only the numbers 30, 20, 12, or 6 can be entered.

After you enter a step rate number and carriage return, CONFIGUR displays a prompt for a track density change, as shown:

```
Soft Sector Unit 0 Track Density ?
```

At this prompt, you should enter either **48** or **96** (depending on the desired density for the specified unit) and a carriage return.

After the carriage return, CONFIGUR will redisplay submenu B with the newly entered step rate and/or track density.

NOTE: Heath/Zenith recommends that a step rate of 30 be applied to drives controlling 48 tpi disks, and that a step rate of 6 be applied to drives controlling 96 tpi disks.

5.2 Turn On/Off Detailed Disk Error Messages

Most users will encounter error messages at some time during use of the CP/M Operating System. However these error messages are usually brief and general in their description of the problem that occurred. By causing submenu B selection G to indicate "TRUE", you can change the operating system so that it displays a code with each operating system error message. This code includes a two-digit hexadecimal number that can be translated into a more detailed explanation of the problem.

```
G    Detailed Disk Error Messages: FALSE
```

This selection offers you only two alternative values: "TRUE" and "FALSE". Therefore, you merely have to type the letter **G** to cause the opposite alternative to be applied to the operating system (if you choose to save these changes). When you type **G**, submenu B is redisplayed with the new value.

More information about detailed error messages is available in Appendix A.

5.3 Return to Main Menu
(Selections Y and Z)

These submenu B selections enable you to exit from the submenu to the main menu.

```
Y    Finished, make changes and return to main menu
Z    Quit, make no changes and return to main menu
```

By typing **Y**, you also instruct CONFIGUR to preserve the selections entered at submenu B (although none of the changes from any submenu can be applied to the operating system until you exit from the CONFIGUR utility entirely).

By typing **Z**, you also instruct CONFIGUR to abandon any changes entered at submenu B.

After either of these entries, CONFIGUR will redisplay the main menu, as shown:

```
CP/M Configuration

A    Set Terminal and Printer Characteristics
B    Set Disk Parameters
C    Change the Default I/OConfiguration
D    Automatic Program Control

X    Configure, making changes to memory only
Y    Configure, making changes to both memory and disk
Z    Quit, making no changes
```

You can now access a submenu or exit from CONFIGUR entirely.

NOTE: If the disk being configured is write protected, the Y selection from the main menu will not be available, and it will not be possible to record changes from this CONFIGUR session on the disk.

6 SUBMENU C: CHANGE THE DEFAULT I/O CONFIGURATION

Typing **C** at the main menu selection prompt will cause CONFIGUR to display the following submenu:

```
A    CON:  = CRT: Available TTY: CRT: BAT: UC1:
B    RDR:  = UR1: Available TTY: PTR: UR1: UR2:
C    PUN:  = UP1: Available TTY: PTP: UP1: UP2:
D    LST:  = LPT: Available TTY: CRT: LPT: UL1:

Y    Finished, make changes and return to main menu
Z    Quit, make no changes and return to main menu

Selection:
```

(This sample display might differ from the user's display.)

The menu shows how the CP/M Operating System currently matches logical device names with physical device names. Immediately to the right of the current devices are lists of other available devices.

The selections in submenu C enable you to do the following:

- Change the selection of logical input/output devices with physical input/output devices; and/or
- Return to the main menu while saving or ignoring the changes selected at this submenu.

6.1 Change the Logical/Physical Device Selection

(Selections A, B, C, and D)

This portion of submenu C enables you to match logical devices with the appropriate physical devices. These selections are essential to inform the operating system of how the different components of the hardware environment are connected.

A logical device name is the most general name used for a category of hardware items. Therefore, each logical device name can apply to a wide range of hardware items. However, CP/M uses only four logical device names. And all input/output machines within the hardware environment must be matched up with these four names.

You match a specific hardware item to a logical device name indirectly, by using a physical device name as an intermediate name. A physical device name is more specific than a logical device name.

Table 2-2 will help you to determine which physical device names can be matched with which logical device names.

NOTE: The suggestions in this table may become invalid if you change device port addresses.

This table suggests which physical devices can be matched up with each logical device, based on the kind of hardware product you have.

SUBMENU C SELECTION LETTER	LOGICAL DEVICE NAME	PHYSICAL DEVICE NAME	DESCRIPTION AND/OR CATALOG NAME OF RECOMMANDED INPUT/OUTPUT MACHINE
A	CON:	TTY:	Any non-handshaking RS-232 ASCII terminal at port 0D0H
		CRT:	Any non-handshaking RS-232 video terminal at port 0E8H
		BAT:	A pseudo batch device using RDR: for input and LST: for output
		UC1:	Any handshaking RS-232 terminal with ETX/ACK protocol, eg. Diablo KSR 1640 printing terminal
B	RDR:	TTY:	Any non-handshaking RS-232 ASCII terminal at port 0D0H
		PTR:	Null source, not implemented, returns an "end-of-file" character when accessed
		UR1:	Any non-handshaking serial device at port 0D8H (eg. a modem)
		UR2:	System terminal
C	PUN:	TTY:	Any non-handshaking RS-232 ASCII terminal at port 0D0H
		PTP:	Null sink, not implemented
		UP1:	Any non-handshaking serial device at port 0D8H (eg. a modem)
		UP2:	System terminal
D	LST:	TTY:	Any non-handshaking RS-232 ASCII terminal at port 0D0H (eg. WH-34)
		CRT:	Any non-handshaking RS-232 video terminal at port 0E8H
		LPT:	Serial printer with hardware handshaking (at port 0E0H) or parallel printer (at port 0D0H)
		UL1:	Any handshaking RS-232 printer with ETX/ACK protocol (eg. a Diablo printer)

Table 2-2

You can start a device assignment change by typing any of the selection letters (**A**, **B**, **C**, or **D**). CONFIGUR will display the logical device name and the " = " sign for the selected letter, in the following form:

```
LST: =
```

At a prompt such as this, enter the first three letters of the new physical device assignment. (No colon or carriage return is necessary.) CONFIGUR will redisplay submenu C with the new assignments.

NOTE: The logical/physical device assignments displayed on CONFIGUR's submenu C can be temporarily changed outside of CONFIGUR, by using the STAT utility, as explained in the text on STAT.

6.2 Return to Main Menu
(Selections Y and Z)

These submenu C selections enable you to exit from the submenu to the main menu.

```
Y    Finished, make changes and return to main menu
Z    Quit, make no changes and return to main menu
```

By typing **Y**, you also instructs CONFIGUR to preserve the selections entered at submenu C (although none of the changes from any submenu can be applied to the operating system until you exit from the CONFIGUR utility entirely).

By typing **Z**, you instruct CONFIGUR to abandon any changes entered at submenu C.

After either of these entries, CONFIGUR will redisplay the main menu, as shown:

```
CP/M Configuration

A    Set Terminal and Printer Characteristics
B    Set Disk Parameters
C    Change the Default I/O Configuration
D    Automatic Program Control

X    Configure, making changes to memory only
Y    Configure, making changes to both memory and disk
Z    Quit, making no changes
```

You can now access a submenu or exit from CONFIGUR entirely.

NOTE: If the disk being configured is write protected, the Y selection from the main menu will not be available, and changes cannot be recorded on disk.

7 SUBMENU D: AUTOMATIC PROGRAM CONTROL

Typing **D** at the main menu selection prompt will cause CONFIGUR to display the following submenu:

```
A    Run automatic command line on Cold Boot: TRUE
B    Run automatic command line on Warm Boot: FALSE
C    Automatic command line: CONFIGUR
Y    Finished, make changes and return to main menu.
Z    Quit, make no changes and return to main menu.

     Selection:
```

(This sample might differ slightly from the user's display.)

The selections in submenu D enable you to do the following:

- Turn on/off the automatic invocation of a command;
- change the command that should be automatically invoked after every cold boot and/or warm boot; and/or
- return to the main menu while saving or ignoring the changes selected at this submenu.

7.1 Turn On/Off Automatic Command Invocation *(Selections A and B)*

This portion of submenu D enable you to change the operating system so that it starts running a command automatically, after a cold boot or warm boot.

```
A    Run automatic command line on Cold Boot: TRUE
B    Run automatic command line on Warm Boot: FALSE
```

These selections each offer you only two alternative values: "TRUE" and "FALSE". Therefore, you merely have to type the letter **A** or **B** to cause the opposite value to be immediately displayed.

7.2 Change Automatic Command Line *(Selection C)*

This submenu D selection enables you to change (or add) the command that can be automatically invoked upon a cold boot or warm boot.

```
    C    Automatic command line: CONFIGUR
```

To start this change, you must first type **C**. CONFIGUR will display the following prompt:

```
    Automatic command line:
```

You can now enter any valid CP/M command line that ends with a carriage return. After you enter the carriage return, CONFIGUR will redisplay submenu D with the new automatic command line.

Any valid resident command, transient command (utility), or application program is acceptable in the automatic command line. However, any file that the command line refers to must reside on the disk specified in the command line. For example, if the line reads:

```
    C    Automatic Command Line: B:SC
```

Then the file "SC.COM" must reside on the disk in drive "B:" for the command to work. If any data referenced in the command line cannot be found in the specified drives, then command execution will be aborted and an error message will be displayed.

NOTE: We recommend that you do not cause a command to be run automatically on **both** a cold boot and a warm boot if this command performs a warm boot as it finishes execution. Doing so would cause the command to execute on cold boot, then execute again whenever you tried to exit to the operating system. Thus no activities outside of this command would be possible when you booted up with this disk.

7.3 Return to Main Menu
(Selections Y and Z)

These submenu D selections enable you to exit from the submenu to the main menu.

```
Y    Finished, make changes and return to main menu
Z    Quit, make no changes and return to main menu
```

By typing **Y**, you also instruct CONFIGUR to preserve the selections entered at submenu D (although none of the changes from any submenu can be applied to the operating system until you exit from the CONFIGUR utility entirely).

By typing **Z**, you also instruct CONFIGUR to abandon any changes entered at submenu D.

After either of these entries, CONFIGUR will redisplay the main menu, as shown:

```
CP/M Configuration

A    Set Terminal and Printer Characteristics
B    Set Disk Parameters
C    Change the Default I/O Configuration
D    Automatic Program Control

X    Configure, making changes to memory only
Y    Configure, making changes to both memory and disk
Z    Quit, making no changes
```

You can now access a submenu or exit from CONFIGUR entirely.

NOTE: If the disk being configured is write protected, the Y selection from the main menu will not be available, and changes cannot be recorded on disk.

8 CONFIGUR ERROR MESSAGES

```
    Unable to open BIOS.SYS
```

EXPLANATION: The CONFIGUR utility is trying to access the file BIOS.SYS (since this file is part of the operating system) but the file is not present. You must have a BIOS.SYS file (by that name) on the disk being used for bootstrap. This file must be for the appropriate version of CP/M, and it must have the ability to control the disk devices being used.

```
    INCONSISTENT Version number. . .can NOT configur!!
```

EXPLANATION: You must implement CONFIGUR program with same version number as the CP/M Operating System being used.

```
    INCONSISTENT version number or CONFLICTING program in memory. . .
    can NOT configure!!
```

EXPLANATION: You must implement CONFIGUR program with same version number as the CP/M Operating System being used.

```
Drive A disk is write protected.
Modifications will NOT be made to the disk for this CONFIGUR run.
```

EXPLANATION: To make CONFIGUR changes go into effect, the disk used to perform bootstrap should be write enabled, by removing the adhesive tab from a 5.25-inch disk or by adhering the adhesive tab to an 8-inch disk.

DDT

The Dynamic Debugging Tool

This text assumes that you are familiar with assembly language programming, the hexadecimal number system, and 8080 CPU registers. Caution is advised when using DDT.

The Dynamic Debugging Tool (DDT) utility enables you to debug machine language programs (files with the extension "HEX" or "COM"). DDT loads a program into the Transient Program Area of the computer's memory (1). During a debugging session, the utility exposes and manipulates the hexadecimal, assembly language, and ASCII forms of the loaded program when you implement the special DDT commands (3). You can copy the results of the debugging session to a disk file by exiting from DDT and implementing the SAVE resident command (2).

1 DDT INVOCATION

The DDT utility can be invoked using two methods.

1.1 System Prompt Invocation Method

You respond to the system prompt with a command line in the form:

A>**DDT** {**file name**} ***RETURN***

Where {**file name**} is the complete name of a program file residing on the default disk with a "HEX" or "COM" extension. This file is automatically loaded into the computer's memory beginning at address 100H (the Transient Program Area start.)

DDT will identify itself with the message:

```
DDT VERS n.n
NEXT PC
aaaa pppp
-
```

Where "n.n" is the utility's version number;

where "aaaa" is the next available memory address after the program is loaded (or the address after the last address occupied by the loaded program);

where "pppp" is the current position of the program counter within the Transient Program Area (this value is 0100 when the program is first loaded into memory); and

where "-" (the hyphen character) is the DDT prompt, at which you can enter special DDT commands.

Because DDT loads programs into the Transient Program Area beginning at address 0100, you can approximate a program's size by subtracting 0100 from the "aaaa" value in the display.

1.2 DDT Prompt Invocation Method

You respond to the system prompt with the following command:

```
A>DDT   RETURN
```

The DDT utility will identify itself with the following display:

```
DDT VERS n.n
-
```

where "n.n" is the version number; and

where "-" (the hyphen character) is the DDT prompt, at which you can enter special DDT commands.

To load a file from the default disk into memory under this invocation method, you must now make the following entries, in order: the letter **I**, (no space), the complete file name of the program to be loaded, a carriage return, the letter **R**, and another carriage return.

The DDT utility will **I**nsert the name of the program into memory, **R**ead the named program into memory, and display values for both the "NEXT" available address and the "PC" program counter. For example, if you desire to debug the program file named "PROGRAM.HEX", the console display of these transactions might appear as follows:

```
A>DDT   RETURN
DDT VERS 2.2
-IPROGRAM.HEX   RETURN
-R RETURN
NEXT PC
3E80 0100
-
```

Where "3E80" is the next available memory address after the end of the program (or the address after the last address occupied by the loaded program);

where "0100" is the current value of the program counter (or the first memory address occupied by the program); and

where "-" (the hyphen character) is the DDT prompt, at which you can enter special DDT commands.

LOADING A PROGRAM FROM A NON-DEFAULT DRIVE

Using the second DDT invocation method, you can access a file from a non-default disk by doing the following, in order:

- Enter the letter **I**, the complete file name of the program to be loaded, and a carriage return.
- Enter the character string **S5C** and a carriage return. This entry will produce a six-character display.
- Enter the two-digit drive number for the drive containing the program that is to be debugged, and a carriage return. (See the table below to find the drive number that corresponds to the appropriate drive letter. This table will not be displayed on the console.)

Drive Number	Drive Letter
00	DEFAULT
01	A
02	B
03	C
04	D
05	E
06	F

NOTE: "DEFAULT" is the drive logged before DDT invocation.

The drive number you entered will be displayed on the right of the six-digit display, and a second six-digit display will appear beneath.

- Enter a **.** (period) and a carriage return. This entry will be displayed on the right of the second six-digit display.

- Enter the letter **R** and a carriage return. DDT will read the program file into memory, and display values for both the "NEXT" available address and the "PC" program counter.

For instance, to access the file named "PROGRAM.HEX" from the disk within non-default drive B (02), you must interact with DDT as shown in the following display:

```
A>DDT   RETURN
DDT VERS 2.2
-IPROGRAM.HEX   RETURN
-S5C   RETURN
005C 00  02   RETURN
005D 57  .   RETURN
-R   RETURN
NEXT  PC
3880 0100
```

2 SAVING A DEBUGGED PROGRAM

You can preserve the results of a debugging session by copying a block of data from the Transient Program Area (TPA) to a disk file. The SAVE resident command will assist in this task. SAVE copies a user-specified number of pages from the TPA to a user-specified file name on a disk.

To SAVE a program that has been loaded into memory and debugged by DDT, first exit from the DDT utility to the operating system. If the exit is performed properly, the image of the program in the TPA will remain undisturbed as SAVE copies from it to a disk file.

You can exit from the DDT utility by either of the following two methods:

- **Performing a Warm Boot** — Any time the DDT hyphen (–) prompt is displayed, enter a ***CTRL-C*** (by pressing the **C** key while holding down the ***CTRL*** key). CP/M will display the system prompt.
- **Jumping to the Operating System's Execution Address** — Any time the DDT hyphen (–) prompt is displayed, you can trigger execution of the operating system, by entering the DDT command **G0** (where "0" is zero). This command sends the program counter to computer memory 0000H (zero), where the operating system resides. CP/M will display the system prompt.

You must proceed immediately to entry of the SAVE command. The SAVE command is entered in the form:

A>**SAVE {pages} {file name}** ***RETURN***

Where {**pages**} are 256-byte units of data that are expressed in decimal (not hexadecimal) numerals, and

where {**file name**} specifies the file name under which you wishes to store the results of the debugging session.

You can SAVE the entire program by determining the (decimal) number of pages between the "pppp" value and the "aaaa" value displayed when DDT initially loads the program into the TPA, as shown:

```
NEXT  PC
aaaa pppp
```

Unless the program has been moved from the beginning of the TPA, the two left-hand digits in the "aaaa" value will be the hexadecimal number of pages the program occupies, as long as the two right-hand digits are not zeros. If the two right-hand digits are zeros, one should be subtracted from this total to determine the number of hexadecimal pages. In either case, the hexadecimal number of pages should be converted to a decimal number suitable for the SAVE command.

3 DDT COMMANDS

The DDT utility has its own assortment of commands. They are entered in response to the DDT hyphen (-) prompt. DDT command lines begin with a single command letter. These command letters are defined in the following list:

A	Assemble: Assembly language mnemonics are inserted
D	Display: Display memory contents in hexadecimal an ASCII form
F	Fill: Fill a block of memory with a specified data constant
G	Go: Go to specified address to run a program
H	Hex: Hexadecimal computation of sums and differences
I	Insert: Insert file name into file control block
L	List: List assembly language mnemonics of a program
M	Move:Move a data block
R	Read: Read a file into memory from disk
S	Substitute: Substitute hexadecimal values
T	Trace: Trace program execution
U	Untrace: Untrace program execution
X	eXamine: Examine or change registers or flags

Most of these command letters can be followed by parameters such as hexadecimal values or a file name. All DDT command lines must end with a carriage return.

When hexadecimal parameters are used, these values consist of one to four digits. (Longer numbers are automatically truncated on the right.) One, two, or three such values can be entered in some command lines. Values are separated by commas or single blank spaces.

Only one DDT command can be entered in response to a single DDT hyphen (-) prompt. Each DDT command line, however, can be composed using most of the same line editing keys and techniques as are used for commands entered at the CP/M system prompt.

No DDT command line can exceed 32 characters in length. If a thirty-third character is entered, it is interpreted as a carriage return and execution begins based on the first 32 characters in the command line.

Many DDT commands operate under a "CPU state" which corresponds to the program being tested. The CPU state holds the program's registers. Initially, all registers and flags contain zeroes—except for the program counter (P) and the stack pointer (S), which default to the value 100H.

The program counter is a CPU register that is used as a movable reference point for DDT commands. It keeps track of the last hexadecimal address that was displayed and/or altered by a DDT command. The address immediately after this last address will be the starting address for the next DDT command you enter (unless the next command specifies a different starting address).

3.1 A Assembly Language Mnemonics are Inserted

The A (assembly) command enables you to insert assembly language instructions into the program being tested. The command is entered in the form:

As

Where **s** is the memory address at which you desire to start inserting assembly language instruction statements.

DDT responds to such an entry by echoing the value ("s") entered. You can then enter an assembly language statement to the right of the echoed "s" value. The statement must end with a carriage return.

DDT will display the next available memory address after the new statement is appended to the program. You can enter another statement to the right of the displayed address, or enter a carriage return alone to end A command operations and retrieve the DDT prompt.

For example, if you want to insert a "MOVE IMMEDIATE to register C" statement into a program at memory address 0104H, the following entry should be made in response to the DDT prompt:

A104 ***RETURN***

DDT will echo the address with the display:

```
0104
```

To the right of the address display, you can insert the statement:

MVI C,{data} ***RETURN***

Where {**data**} is the data to be moved into register C.

DDT will then display the value for the next available memory location. Since the "MVI" statement takes up two locations, DDT displays:

```
0106
```

You can enter another statement at location 0106H or end the operations of the A command by entering a carriage return alone.

NOTE: When the A command inserts a statement at a particular memory address, the statement(s) that formerly occupied that part of memory will be overwritten, and therefore destroyed. If you insert an statement that does not occupy the same number of locations as the statement(s) being replaced, the meaning of subsequent statements might be changed. You should use the L command immediately after finishing A command operations, to verify that the desired results were achieved during use of the A command.

The following example demonstrates this problem. You want to replace a "jump" instruction (JMP) with a "return from subroutine" statement (RET). The JMP statement occupies three locations, and the RET instruction will occupy one. Inserting the one-byte RET into the first location of the three-byte JMP will leave the last two-thirds of the JMP statement in the program. This partial statement could cause problems when the program is run.

3.2 D Display Memory Contents in Hexadecimal and ASCII Form

The D (display) command allows you to view the contents of memory in both hexadecimal and ASCII formats. The display appears in the following form:

```
aaaa bb bb bb bb bb bb bb bb bb bb bb bb bb bb bb bb cccccccccccccccc
```

Where "aaaa" is the address of the first memory location displayed in this line;

where "bb" represents the hexadecimal contents of a memory location; and

where "cccccccccccccccc" represents the ASCII translation of the contents of each memory location.

If the contents of a memory location cannot be displayed as an ASCII character, a period (.) will be displayed instead.

The display address acts as a pointer in memory which is initially set to 100H. As each memory location is displayed, the pointer is incremented by one so at the the end of a D command, the pointer is positioned ready to display the next 256 memory locations.

The four forms of the command are:

D Display memory from the current display address until one page of data (256 locations) have been shown. DDT will display 16 lines, each representing 16 bytes of data.

Ds Change the starting display address to "s", then display memory beginning with address "s" and continuing until one page of data (256 locations) is displayed.

Ds f Change the starting display address to "s", then display memory beginning with address "s" and continuing until memory address "f" is reached.

D,f Display memory from the current display address until address "f" is reached.

Displays triggered by any of these commands can be suspended if a ***CTRL-S*** is entered during the display. The display will resume if anything other than CTRL-C is entered.

A display will be aborted if any character other than CTRL-S is entered. However, it is recommended that the display be intentionally aborted by pressing the DELETE key, because other characters will appear at the next DDT hyphen (-) prompt if they are used to abort the display.

For example if the file SYSGEN.COM had been loaded into memory by DDT and the user wanted to see a hexadecimal and ASCII display of it, then **D** should be entered at the DDT hyphen prompt. This entry would cause a display something like the following:

```
0100 C3 79 02 43 4F 50 59 52 49 47 48 54 20 28 43 29 .y.COPYRIGHT (C)
0110 20 31 39 37 38 2C 20 44 49 47 49 54 41 4C 20 52 1978, DIGITAL R
0120 45 53 45 41 52 43 48 20 6F 26 00 29 29 29 29 29 ESEARCH o&.)))))
0130 29 29 C9 0E 01 CD 05 00 FE 61 D8 FE 7B D0 E6 5F )).......a..{..
0140 C9 5F 0E 02 CD 05 00 C9 3E 0D CD 41 01 3E 0A CD . ......>..A.>..
0150 41 01 C9 E5 CD 48 01 E1 7E B7 C8 E5 CD 41 01 E1 A....H..~....A..
0160 23 C3 58 01 D5 4F 2A 01 00 11 18 00 19 D1 E9 2A #.X..O*........*
0170 01 00 11 1B 00 19 E9 2A 01 00 11 1E 00 19 E9 2A .......*.......*
0180 01 00 11 21 00 19 E9 2A 01 00 11 24 00 19 E9 2A ...!...*...$...*
0190 01 00 11 27 00 19 E5 21 14 07 3A 12 07 BE F5 35 ...'...!..:....5
01A0 CA AE 01 F1 0E 00 C2 B7 01 0E 02 C3 B7 01 F1 3A ...............:
01B0 12 07 32 14 07 0E 01 E1 E9 0E 14 C3 05 00 0E 0F ..2.............
01C0 C3 05 00 21 00 09 22 08 07 3E FF 32 05 07 21 05 ...!.."..>.2..!.
01D0 07 34 7E B7 CA EB 01 3A 0F 07 32 0C 07 3A 13 07 .4~....:..2..:..
01E0 32 12 07 E5 2A 10 07 22 0D 07 E1 3A 12 07 32 14 2...*.."...:..2.
01F0 07 3A 0B 07 BE CA 78 02 4E CD 6F 01 3E FF 32 06 .:....x.N.o.>.2.
```

3.3 F Fill a Block of Memory with a Specified Data Constant

The F (fill) command allows you to fill a block of memory with a specific constant. The form of the command is:

Fs f c

Where **s** is the address at which the filling should begin;

where **f** is the address at which the filling should end; and

where **c** is the data constant that should occupy each memory address in between.

Any data that resided between addresses "s" and "f" prior to the entry of the command will be overwritten by the constant, and therefore destroyed.

Only hexadecimal values should be entered in such a command, and value "f" must be greater than value "s". If "s" is greater than "f", the operation will not be executed and the DDT prompt will reappear.

For example, the following command:

-F9200 B400 E5 ***RETURN***

would fill every memory location from address 9200H through address B400H with the hexadecimal value "E5".

3.4 G Go to Specified Address to Run a Program

The G (Go to) command enables you to begin execution of the program from any address, and to specify one or two execution breakpoints if desired. (A breakpoint is the address of an instruction which, when reached, will stop the execution of the program and redisplay the DDT prompt.) Execution begins with the instruction at the memory address immediately following the one specified in the command. The instruction at the specified address is not executed.

If no breakpoint is entered, the only other way in which the control of the program may be returned to you is if an "RST 7" instruction is encountered within the program. This instruction will immediately stop program execution and redisplay the DDT prompt to allow further DDT commands from you.

The G command can be entered in the following forms:

G Begins execution of the program at the current value of the program counter, with no breakpoints set. The program will run to completion.

Gs Sets the program counter to address "s" and begins execution of the program from that address, with no breakpoints set. The program will run to completion.

G,b Begins execution of the program at the current value of the program counter and continues until the instruction at address "b" (the breakpoint) is reached. Then program execution stops.

Gs b Sets the program counter to "s" and begins execution of the program at address "s". When the instruction at address "b" (the breakpoint) is reached, program execution stops.

G b c Begins execution of the program at the current value of the program counter and continues until either address "b" or address "c" is reached. When either of these breakpoint addresses is reached, program execution stops.

Gs b c Sets the program counter to address "s" and begins execution of the program at this address. When either address "b" or address "c" is reached, program execution stops.

At a breakpoint, program execution stops and DDT displays:

```
*bbbb

-
```

Where "bbbb" is the address at which program execution stopped; and

where "-" is the DDT prompt

For example, you could "go to" the very beginning of computer memory (address 0000H) and trigger execution of the program that is situated there. This program is, of course, the CP/M operating system. Its execution can be triggered by entry of the following command:

```
-G0 RETURN
```

The operating system would respond by displaying the system prompt, as shown:

```
A>
```

This command has the same effect as a warm boot.

3.5 H Hexadecimal Computation of Sums and Differences

The H (hexadecimal value) command simultaneously adds and subtracts two hexadecimal values. This command is entered in the following form:

```
Ha,b
```

Where **a** is a hexadecimal value; and

where **b** is another hexadecimal value.

The resulting display appears in the form:

```
ssss dddd
```

Where "ssss" represents the sum of two values; and

where "dddd" represents the difference between the two values.

This command is helpful in determining addresses to which programs will be relocated with DDT command "M".

For example, if you have a program that begins at address 0311H, and wishes to move this program 0126H bytes higher in memory, then the H command could be used to calculate the new starting address, as shown:

```
-H311,126   RETURN
0437 01EB
```

0437H would be the new starting address for the program.

However, if you enter an "a" value that is smaller than the "b" value, the sum ("ssss") will be the same, but the difference ("dddd") will be equal to 10000H minus the amount that "b" is greater than "a".
Such a case is demonstrated by the following entry:

-H1,2 ***RETURN***

which will produce the following displayed solution:

```
0003 FFFF
```

3.6 I Insert File Name Into File Control Block

The I (input) command allows you to insert a file name into the area of memory that is used to store the names of files to be read from the disk. This area of memory begins at address 5CH. This is one of the memory areas from which the Console Command Processor (a functioning part of the CP/M operating system) distributes control to utilities and resident commands. This DDT command is entered in one of the following forms:

-I{primary file name} ***RETURN***

-I{primary file name}.{extension} ***RETURN***

If the second form of the command is used and the {extension} entered is either "HEX" or "COM", then subsequent R commands can be used to read the pure binary or hexadecimal machine code.

The I command will not read the file from the disk and store it into memory. It will only insert the filename into the File Control Block portion of the Console Command Processor, so that a subsequent R command can read the named file into memory.

3.7 L List Assembly Language Mnemonics of a Program

The L (list) DDT command enables you to disassemble the instructions wthin a span of memory, and to display the assembly language mnemonics of the disassembled code on the console. The command can be entered in any of the following three forms:

L Lists 12 lines of disassembled machine code, beginning at the current list address. The list address acts as a pointer in memory which is initially set to 100H. As each memory location is disassembled, the pointer is incremented by one.

Ls Changes the list address to "s", and then lists 12 lines of disassembled machine code beginning at address "s".

Ls f Lists disassembled code from starting address "s" to the final address "f".

The list appears in the form:

```
aaaa mmmm oooo
```

Where "aaaa" is the address of the instruction,

where "mmmm" is the mnemonic of the operator, and

where "oooo" is the operand.

Listings triggered by an L command can be suspended if a ***CTRL-S*** is entered during the listing. The listing will resume if anything other than CTRL-C is entered.

A listing will be aborted if any character other than CTRL-S is entered during the listing. If the listing is to be aborted intentionally, we recommend that the DELETE key be pressed, because other characters might appear at the next DDT hyphen (-) prompt if they are used to abort the listing.

If an invalid mnemonic is encountered in a statement of a disassembled program, question marks ("??") will be used to represent the invalid mnemonic in the listing.

The disassembled mnemonics from address 0919 through address 091D of the program in memory can be listed with the entry of the following command:

-L0919,091D ***RETURN***

Such a listing might appears as follows:

```
0919 OUT  F2
091A INX  H
091B MOV  A,M
091D ORA  A
```

3.8 M Move a Data Block

The M (move) command allows you to move a block of data from one area of memory to another. This command is entered in the form:

Ms f d

Where **s** is the starting address;

where **f** is the final address of the block of data to be moved; and

where **d** is the starting point of the memory area to which the data is moved.

The data is moved to the area of memory beginning at the address "d". An example of the command follows:

M100 200 1000 ***RETURN***

This command would take the contents of the block of memory starting at address 0100H and running through address 0200H, and move these contents to the area of memory beginning with the address 1000H.

3.9 R Read a File Into Memory From Disk

The R (read) command is used after the I command to read COM and HEX files from the diskette into the transient program area in preparation for a debugging operation. The R command requires a previous I command, specifying the name of the HEX or COM file to be read. The command can be entered in either of the following two forms:

R Reads the file whose name is in the file control block at address 5CH from the disk and places it in the Transient Program Area. (The file name was placed in this location with the I command.)

Rb Reads the file whose name is in the file control block at address 5CH from the disk and places it in the Transient Program Area with the addition of a bias factor, "b", which is a hexadecimal number added to each program instruction address or data address as it is read. This factor allows you to locate the program at any location in memory. When the bias factor is omitted, then b = 0000 is assumed.

The read operation must not place the file in the first page of memory (0-0FFH) because this would write over the system parameters which are stored in this area. If the file specified in the preceding I command is a HEX file, the load address is derived from each individual HEX record. If the file to be loaded is a COM file, a load address of 100H is assumed. Any number of R commands may be issued following an I command to reread the program under test.

The R command reads the desired file from the default drive. If the desired file resides on a non-default drive, the command **S5C** should be entered, and the value for the non-default drive should be substituted according to the following table:

Drive Number	Drive Letter
00	DEFAULT
01	A
02	B
03	C
04	D
05	E
06	F

This substitution should be performed between the I command and the R command.

When the R command loads a named file into the Transient Program Area, a message in the following form is displayed:

```
NEXT PC
nnnn pppp
```

Where "nnnn" is the the address immediately following the loaded program; and

where "pppp" is the current value of the program counter (100H for COM files, or it is taken from the last record if a HEX file is specified.)

The next address "nnnn" can be used to determine the size of the file which was loaded. If the beginning address is 100H, then subtracting 100H from "nnnn" will display the size of the program in bytes. The size derived in this manner is in hexadecimal units, and may have to be converted to decimal units before it is used.

3.10 S Substitute Hexadecimal Values

The S (substitute) command enables you to examine—and optionally alter—the contents of specified memory locations. This command is entered in the form:

Sb

Where **b** is the hexadecimal address of the first memory location to examine.

DDT responds with a display of addresses and bytes in the form:

```
aaaa cc
```

Where "aaaa" is the hexadecimal address, and

where "cc" is the hexadecimal contents of the memory location.

You may substitute a new value for "cc" by entering the new value (in one or two hexadecimal digits) and a carriage return when DDT displays "aaaa" and "cc". Your entry will appear on the right side of this display, and it will replace "cc" in the memory image of the program.

The next address "aaaa" and its contents "cc" are then displayed, inviting you to substitute a new value for this "cc". When you are finished altering address contents in this sequence of addresses, a period (.) and a carriage return should be entered, rather than a new value. Your alterations will be retained in memory, and the DDT hyphen (-) prompt will reappear.

If you wish to skip an address without changing it, then a carriage return (without a period) should be entered in response to one of the "aaaa" "cc" displays. The DDT hyphen (-) prompt will reappear.

For example, if you enter the following command:

```
-S100  RETURN
```

then memory addresses and their contents will be displayed on the screen, starting with address 0100H, as shown. your substituted values for the address contents are in boldface print on the right side of the following example display:

```
0100 C3  3C  RETURN
0101 C0 C  RETURN
0102 01 10  RETURN
0103 20   RETURN
0104 43   RETURN
0105 4F F4  RETURN
0106 50 .  RETURN
-
```

3.11 T Trace program execution

The T (trace) command allows to trace the execution of one to 65,535 (0FFFFH) program steps. During the trace, the contents of all registers and the status of all flags within the central processing unit (CPU) are displayed. This command can be entered in either of the following forms:

T Displays the contents of the CPU registers and the status of the flags; then executes one program instruction. The DDT hyphen prompt (-) reappears.

Tn Displays the contents of the CPU registers and the status of the flags; then executes "n" program instructions and stops. The DDT hyphen prompt (-) reappears.

Displays caused by the T command take the following form:

```
CfXfMfEfIf A=bb B=dddd D=dddd H= dddd S=dddd P=dddd inst *hhhh
```

Where "f" is a 0 or 1 flag value;

where "bb" is a byte value;

where "dddd" is a double byte quantity corresponding to a register pair;

where the "inst" field contains the disassembled instruction which occurs at the location addressed by the program counter; and

where "hhhh" is the next address available for execution.

The display address (used in the D command) is set to the value of the H and L registers. The list address (used in the L command) is set to the value of "hhhh" so it will be ready to list the next program steps to be executed if desired. Since the state of the flags and registers of the CPU displayed by the T command occur before each instruction is executed, it may be helpful to use an X command to view the state of the CPU after the trace command.

The second form of the T command will trace the execution for "n" steps ("n" is a hexadecimal value) before a program breakpoint occurs. A breakpoint can be forced during long trace displays by using the DELETE key. The state of the CPU is displayed before each program step is executed in the trace mode.

If the program being tested must access the disk or input/output (I/O) devices through the CP/M system, the program tracing is discontinued at the interface to CP/M, and resumes after returning from CP/M to the program being tested. Thus, CP/M functions which access I/O devices, such as the disk drive, operate at the proper speed (real time), thereby avoiding I/O timing problems. Programs running in the trace mode execute aproximately 500 times slower than real time because DDT gets control after each user instruction is executed. In programs which use interrupt instructions, the interrupts are always enabled during the trace mode.

3.12 U Untrace Program Execution

The U (untrace) command allows you to trace the execution of one to 65,535 (0FFFFH) program steps. During the untrace, the contents of all registers and the status of all flags within the central processing unit (CPU) are displayed. Intermediate program steps are not displayed. This command can be entered in either of the following forms:

U Displays the contents of the CPU registers and the status of the flags. Then executes one program instruction. The DDT hyphen prompt (-) reappears.

Un Displays the contents of the CPU registers and the status of the flags. Then executes "n" program instructions, and stops. The DDT hyphen prompt (-) reappears.

Displays caused by the T command take the following form:

```
CfXfMfEfIf A=bb B=dddd D=dddd H=dddd S=dddd P=dddd inst *hhhh
```

Where "f" is a 0 or 1 flag value;

where "bb" is a byte value;

where "dddd" is a double byte quantity corresponding to a register pair;

where the "inst" field contains the disassembled instruction which occurs at the location addressed by the program counter; and

where "hhhh" is the next address available for execution.

The display address (used in the D command)1 is set to the value of the H and L registers. The list address (used in the L command) is set to the value of "hhhh" so it will be ready to list the next program steps to be executed if desired. Since the state of the flags and registers of the CPU displayed by the U command occur before each instruction is executed, it may be helpful to use an X command to view the state of the CPU after the trace command.

The second form of the U command will untrace the execution for "n" steps ("n" is a hexadecimal value) before a program breakpoint occurs. A breakpoint can be forced during long untrace displays by using the DELETE key. The state of the CPU is displayed before each program step is executed in the untrace mode.

If the program being tested must access the disk or input/output (I/O) devices through the CP/M system, the program tracing is discontinued at the interface to CP/M, and resumes after returning from CP/M to the program being tested. Thus, CP/M functions which access I/O devices operate at the proper speed (real time), thereby avoiding I/O timing problems. Programs running in the untrace mode execute aproximately 500 times slower than real time because DDT gets control after each user instruction is executed. In programs which use interrupt instructions, the interrupts are always enabled during the untrace mode.

3.13 X eXamine or Change Registers or Flags

The X (examine) command enables you to display and alter the state of the registers and flags of the CPU at any time during the debugging process. This command can be entered in either of the following forms:

X

Xr

Where **r** is one of the 8080 CPU registers in the following table:

8080 CPU REGISTER SYMBOL	REGISTER NAME	RANGE OF REGISTER CONTENTS
C	Carry Flag	(0/1)
Z	Zero Flag	(0/1)
M	Minus Flag	(0/1)
E	Even Parity Flag	(0/1)
I	Inter-Digit Carry	(0/1)
A	Accumulator	(0-FF)
B	BC Register Pair	(0-FFFF)
D	DE Register Pair	(0-FFFF)
H	HL Register Pair	(0-FFFF)
S	Stack Pointer	(0-FFFF)
P	Program Counter	(0-FFFF)

The first form of the command displays the state of the CPU in the following form:

```
CfXfMfEfIf A=bb B=dddd D=dddd H=dddd S=dddd P=dddd inst
```

Where "f" is a 0 or 1 flag value;

where "bb" is a byte value;

where "dddd" is a double byte quantity corresponding to a register pair; and

where the "inst" field contains the disassembled instruction which occurs at the location addressed by the program counter.

The second form of the command displays the flag or register value of the specified register, and allows alteration of the hexadecimal value within this flag or register. You can substitute a new value for the value held in the register. You makes this substitution by entering the new value and a carriage return to the right of the existing value.

This example demonstrates how the value in a register can be altered.

```
-XS
S=00FE EF00   RETURN
-
```

When you substitute a value into a flag or register, ending the substitution with a carriage return, the DDT hyphen prompt (-) reappears. If you wish to make no changes to the values, then a carriage return alone should be entered.

If a value in the proper range is typed, then the flag or register value is altered. Note that BC, DE, and HL are displayed as register pairs. Thus you enter the entire register pair when B, C, or the BC pair is altered.

4 DDT Error Signals

DDT does not display entire messages when you make an erroneous entry. However, DDT will display a question mark (?) if your entry does not conform to valid entry syntax restrictions.

In addition, DDT will display question marks (??) in place of invalid mnemonics that it encounters when dealing with the assembly language form of a program.

DIR

The Resident Command that Displays Disk File Directories

The DIR resident command is issued to determine the presence of (1) all of the files on a disk, (2) a specified file, or (3) a group of specified files. After command entry, DIR displays file names to the console (4). Some file names cannot be accessed by DIR (5).

1 DIRECTORY OF All FILES ON A DISK

DIR can be used to determine the names of all files on a disk by answering the system prompt with the following entry:

A>**DIR** ***RETURN***

If you desire a DIRectory of the files on a disk that does not reside in the default drive (drive B for instance) the DIR command should be entered with a drive specification, as shown:

A>**DIR B:** ***RETURN***

NOTE: Because DIR is a resident command, it is automatically loaded into the computer with the rest of the operating system. Therefore, it is never necessary (or valid) to specify a drive at the **beginning** of a DIR command line. For example, the command A>**B:DIR** ***RETURN*** is invalid.

2 DIRECTORY OF A SPECIFIED FILE

To find out whether one particular file resides on a disk, the complete name of that file is entered one space after the resident command specification "DIR". For example, the entry of the following command line will check the disk in default drive A for the file named "THISFILE.DOC":

```
A>DIR THISFILE.DOC   RETURN
```

The presence of a specific file on a disk in a non-default drive can be determined by entering the appropriate drive name and a colon immediately before the name of the specified file.

3 DIRECTORY OF A GROUP OF FILES

To inquire about several files that belong to a group with similar names, the you can enter an ambiguous file name (a file name with wildcard characters "*" and "?"). For instance, to check the default drive disk for all of the files with the extension "BAK", you would enter the following command line:

```
A>DIR *.BAK   RETURN
```

The command line:

```
A>DIR PROGRAM?.HEX   RETURN
```

will check the disk for files such as "PROGRAM1.HEX", "PROGRAM2.HEX", and "PROGRAM3.HEX". In addition, the command:

```
A>DIR S*.COM   RETURN
```

will check the disk for files such as "SC.COM", "STAT.COM", and "SYSGEN.COM".

4 CONSOLE DISPLAY OF FILE DIRECTORY

The entry of a DIR command line will produce a console display showing up to four file names in a horizontal line, with the name of the logged drive preceding each line. The following example shows a DIRectory display of all of the files on the disk in drive A:

```
A: MOVCPM5  COM : LIST     COM : PIP      COM : SUBMIT   COM
A: STAT     COM : XSUB     COM : ED       COM : ASM      COM
A: DDT      COM : LOAD     COM : CONFIGUR COM : SYSGEN   COM
A: DUMP     COM : DUMP     ASM : DUP      COM : FORMAT   COM
A: CPM48    COM : BATCH    SUB : DUMP     PRN : DUMP     HEX
```

If a specified file or group of files is not found on the disk being investigated, the console will display the message:

```
NO FILE
```

This message will also appear if a DIR command line references files on an empty disk.

5 FILES NOT ACCESSIBLE BY DIR

DIR commands will not produce a display indicating the presence of files that maintain the "SYS" status. (See STAT 2.3.)

Files assigned to an unlogged user area are also inaccessible to a DIR command unless a USER command is issued before the DIR command. (See USER.)

DUMP

The Utility that Displays a File in Hexadecimal Form

The DUMP utility is invoked with the name of a file (1) to display the hexadecimal contents of each address in file (2). The file contents appear in lines containing 16 bytes of data each. To the left of each line is the address of the first byte in each line. This display will continue to the end of the program, unless suspended or aborted (3).

1 DUMP INVOCATION

The hexadecimal contents of a file can be displayed on the console by responding to the system prompt with the command:

A>**DUMP** {**file name**} ***RETURN***

Where **file name** is the complete file name of the disk file that you wish to examine in a hexadecimal display.

2 EXAMPLE DUMP DISPLAY

The command **DUMP THISFILE.HEX** ***RETURN*** might produce this display:

```
0000 C3 C0 01 20 43 4F 50 59 52 49 47 48 54 20 28 43
0010 29 20 31 39 37 39 2C 20 44 49 47 49 54 41 4C 20
0020 52 45 53 45 41 52 43 48 20 44 49 53 4B 20 4F 52
0030 20 44 49 52 45 43 54 4F 52 59 20 46 55 4C 4C 24
0040 46 49 4C 45 20 45 58 49 53 54 53 2C 20 45 52 41
0050 53 45 20 49 54 24 4E 45 57 20 46 49 4C 45 24 2A
0060 2A 20 46 49 4C 45 20 49 53 20 52 45 41 44 2F 4F
etc. . .
```

3 SUSPENDING LONG SCROLLING DISPLAYS

The displays produced by the DUMP utility often scroll by quickly on the console. However, they can be halted temporarily by entering a ***CTRL-S*** at any time during the scroll. The display can then be resumed by entering any keyboard character except CTRL-C (which executes a warm boot and aborts the program). A run of the DUMP utility can be aborted altogether by entering any keyboard character (other than CTRL-S) during the display.

4 DUMP ERROR MESSAGE

```
NO INPUT FILE PRESENT ON DISK
```

EXPLANATION: The file specified to be dumped does not exist on the specified disk. Command should be re-entered specifying the proper file name or drive name, or inserting the proper disk.

DUP

The Utility for Copying and/or Verifying Entire Disks

The DUP utility can be used to duplicate **all** of the data from one disk to another disk. It can also compare the two disks to verify whether the data recorded on one disk correspond exactly to the data recorded on another disk. If desired, DUP will even perform both operations consecutively, to ensure accurate duplication of a disk.

NOTE: Both of the disks involved in a DUP operation must be prepared in the exact same fashion by the FORMAT utility. Thus the density and number of sides used for data storage on each disk must be identical. Furthermore, you can not duplicate disks that were initialized with a different operating system, such as HDOS or MS™-DOS.

You can use the DUP utility through either of two methods: the DUP Prompt Method or the System Prompt Method.

1 DUP PROMPT METHOD

With this DUP method, you invoke the DUP utility from a disk by entering a command at the system prompt. Then you answer a series of DUP prompts to define the duplication operation.

™MS is a trademark of Microsoft Corporation.

1.1 DUP Invocation

Invoke DUP by typing a command at the system prompt in the following form:

A>**DUP** ***RETURN***

The DUP utility will present the following display when invoked:

```
Disk Utility Program
Version 2.04

Do you want to:

    A  copy and verify
    B  copy only
    C  verify only

    Z  exit to operating system

Selection:
```

This display includes a menu listing the four operations DUP offers. You can begin execution of an operation by typing the letter listed to the left of that operation. Each operation is explained in the following sections.

1.2 A copy and verify

The "copy and verify" operation makes an exact duplicate of a disk, and then verifies that the operation was performed flawlessly, by comparing the two disks.

To begin this operation, type **A** at the "Selection:" prompt. (No carriage return is necessary.)

DUP will ask, in a series of consecutive prompts, for the letter of the drive that contains the disk to be copied **from** ("Source unit:") and for the letter of the drive that contains the disk to be copied **to** ("Destination unit:"). Enter the appropriate drive letter for each prompt. (No carriage return is necessary.)

When you have specified both a source and destination, DUP will instruct you to put the appropriate disks in the specified drives (even if you have already done so). Then the screen display will look something like the following:

```
Source unit:C
Destination unit:D
Put source disk in drive C.
Put destination disk in drive D.

Press RETURN to begin:
```

The last three lines in this display give you a final opportunity to make certain that you have specified the proper drives and inserted the proper disks. (It is important to make certain of these factors, lest you accidentally duplicate the contents of a blank disk over to a disk that has useful data.)

When both disks are positioned in the proper drives, enter a carriage return to start the copying process. The lights on the specified drives will glow alternately to signify DUP activity. (The duration of the copy and verify operation varies depending on the density and number of sides used on the disks.) When finished copying, DUP will display the message:

```
Copy finished.
```

The verification process will begin automatically. DUP will compare the source and destination disks to verify the accuracy of the copy. Then DUP will display the message:

```
Verification finished.
```

and redisplay the DUP selection menu.

1.3 B copy only

The "copy only" operation makes an exact duplicate of a disk.

To begin this operation, type **B** at the "Selection:" prompt. (No carriage return is necessary.)

DUP will ask, in a series of consecutive prompts, for the letter of the drive that contains the disk to be copied **from** ("Source unit:") and for the letter of the drive that contains the disk to be copied **to** ("Destination unit:"). The screen display for this operation may appear as:

```
Source unit:C
Destination unit:D

Put source disk in drive C.
Put destination disk in disk D.

Press RETURN to begin:
```

The last three lines in this display give you a final opportunity to make certain that you have specified the proper drives and inserted the proper disks. (It is important to make certain of these factors, lest you accidentally duplicate the contents of a blank disk over to a disk that has useful data.)

When both disks are positioned in the proper drives, enter a carriage return to start the copying process. The lights on the specified drives will glow alternately to signify DUP activity. (The duration of the copy operation varies depending on the density and number of sides used on the disks.) When finished, DUP will display:

```
Copy finished.
```

Then DUP will redisplay the selection menu.

1.4 C verify only

The "verify only" operation helps you determine whether two disks are identical in media and data contents.

To begin this operation, type **C** at the "Selection" prompt. (No carriage return is necessary.)

DUP will ask, in two consecutive prompts, for the letter of the drives that contain the disks to be compared. (These prompts will ask you to specify "Source unit:" and "Destination unit:", although "source" and "destination" are not pertinent to this operation.)

Answer each prompt with the name of one of the drives containing a disk to be compared. The screen display for this operation might appear as follows:

```
Source unit:C
Destination unit:D
Put source disk in drive C.
Put destination disk in drive D.

Press RETURN to begin:
```

Enter a carriage return to start the verification process. The lights on the specified drives will glow alternately to signify DUP activity. (The duration of the verification operation varies depending on the density and number of sides used by the disks.)

If DUP finishes comparing the two disks and finds that they are absolutely identical, DUP will display the message:

```
Verification finished
```

Then DUP will redisplay the selection menu.

If the two disks do not correspond exactly, in data content or data position, then DUP displays the following message:

```
Verification error
```

Then DUP will redisplay the selection menu.

NOTE: Conceivably, two disks could contain the exact same data, but in different positions on the disk surface. DUP verification would regard such disks as **different**.

1.5 Z exit to operating system

When the DUP selection menu is displayed, you can end the program and return to the operating system by typing the Z alternative at the "Selection:" prompt. Such an entry will produce the prompt:

```
Place a bootable disk in drive A and type any character:
```

At this prompt, you should insert the disk you used to perform bootstrap into drive A (if you have removed it since bootstrap) and enter a carriage return.

1.6 Invalid Entries During a DUP Prompt Operation

If you answer a DUP prompt with an invalid character, either the prompt or the menu will be repeated in most cases. However, if you specify a drive that does not exist in the hardware environment (when responding to the "Source unit:" prompt or the "Destination unit:" prompt) then the terminal may "hang up", freezing the keyboard. In such a case, you must reset the computer and perform bootstrap to proceed with any CP/M operation.

2 SYSTEM PROMPT METHOD

The System Prompt Method enables you to include all of the specifications necessary for the DUP operation in a single command line entered at the CP/M system prompt.

2.1 Command Line Entry

System Prompt Method DUP commands are entered in the following form:

```
A>DUP {destination}:={source}:{[option]}    RETURN
```

Where **DUP** is the command line function;

where {**destination**} is the letter of the drive containing the blank disk that you wish to receive the copied data;

where {**source**} is the letter of the drive containing the data disk that you wish to duplicate; and

where {**[option]**} represents letters enclosed in square brackets [] and separated by a comma , to specify how the DUP operation should be conducted. One or two of the letters **C**, **V**, or **N** can be used, although none of these options are mandatory.

NOTE: In a command line equation, the data source is always on the right and the data destination is always on the left.

2.2 DUP Options

Through the System Prompt Method, you can enter the following options to perform a DUP operation as indicated. These option letters should always be enclosed in square brackets. If two option letters are used, they should be separated by a comma.

C Copy only: DUP will copy all of the data from the source disk to the destination disk, without comparing them to verify the accuracy of the copy.

V Verify only: DUP will compare the source and destination disks to verify that they contain the exact same data in the exact same locations on the disk surface. When this option is specified, either disk can be the "source" or "destination".

N No inquiry prompt: DUP will perform the operation you have specified without displaying a prompt to confirm whether the disks are in the appropriate drives.

Without the C or V Options Copy and verify: DUP will copy all of the data from the source disk to the destination disk, and then automatically compare the two disks to verify that the copying was accurate. This operation will be performed by default if you enter a DUP command with source drive and destination drive specifications but without specifying a C or V option.

2.3 DUP Defaults

If you enter a System Prompt Method DUP command line (specifying destination and source drives) and abstain from specifying certain options, the following default conditions will be in effect during the DUP operation:

- Copy and Verify operation will be performed. Hence, the entire contents of the source disk will be copied to the destination disk and then the two disks will be compared to verify whether they are exact duplicates. Occurs if neither the C nor the V options are specified.

- Prompts are displayed to encourage you to insert the source and destination disks in the specified drives, and then to trigger the start of the DUP operation with a carriage return. These prompts appear as shown:

```
Disk Utility Program
Version 2.04


Put source disk in drive A.
Put destination disk in drive B.

Press RETURN to begin:
```

Prompts in this form will appear if the N option is not specified.

2.4 System Prompt Examples

A>**DUP B:=A:** ***RETURN***

DUP will prompt you to insert the proper disks into drives B and A, then copy all of the data from drive A to drive B, and then verify whether the copy was performed accurately.

A>**B:DUP C:=D:[C,N]** ***RETURN***

The DUP utility, in this case, is stored on the disk in non-default drive B. It will copy all of the data from the disk in drive D to the disk in drive C. As specified in the options, DUP will **not** prompt you to insert the disks in the appropriate drive and DUP will **not** verify the accuracy of the copy.

A>**DUP D:=E:[V,C]** ***RETURN***

If your command line options are contradictory, (both C and V use in same line) DUP will acknowledge only the last one. Hence, in this case, DUP will prompt you to insert the proper disks into drives D and E, then **copy** all of the data from drive E to drive D, DUP will **not** verify whether the copy was performed accurately.

A>**dup c:=e:[v,n]** ***RETURN***

DUP will verify that the data stored on the disks in drives C and E is exactly the same and arranged in the same locations on the surfaces of these disks. As specified in the options, DUP will **not** copy any data and it will **not** prompt you to insert your disks. Notice that the letters in a DUP command line do not have to be upper case (capitalized).

NOTE: Conceivably, two disks could contain the exact same data, but in different positions on the disk surface. A DUP verification operation would regard such disks as **different**, and display a "Verification Error" message.

3 DUP ERROR MESSAGES

```
Media incompatible on diskettes.
```

EXPLANATION: Disks used for duplication must be identical in size, density, number of sides, and tracks per inch.

```
Drives incompatible for copy operation.
```

EXPLANATION: DUP operations can only be performed between two drives that write the same type of disk media. If you have Z-37 drives, make certain that the drives you have selected as your "Source unit" and "Destination unit" have been set to write data at the same step rate and number of tracks per inch (tpi). Specify drives that write to identical disk types for your "Source unit" and "Destination unit".

```
Drive not available in current configuration.
```

EXPLANATION: Drives specified as source or destination must be drives that are connected in the hardware environment, turned on and recognized by the operating system. Specify such drives.

```
Hard read error on source disk. Copy/Verify aborted.
```

EXPLANATION: DUP failed in an attempt to read data from a source disk. Try the operation again. If DUP failures persist, use the PIP utility to copy files from the source disk, and the SYSGEN utility to copy the operating system from the source disk.

```
Hard read error on destination disk. Copy/Verify aborted.
```

EXPLANATION: DUP failed in an attempt to write data to the destination disk. Use the FORMAT utility to prepare the destination disk before using DUP. If DUP failures persist, use a different disk as the data destination. Make certain that the destination disk media is the same as the source disk media.

```
Verification error.
```

EXPLANATION: DUP's comparison of two disks found them to be different. Determine which disk contains the desired data, call this disk the "source disk." Then use the FORMAT utility to erase and prepare the inferior disk, and perform DUP's "copy and verify" operation. If the second verification of these two disks produces this error, use a new disk of the same media type for the destination and repeat the "copy and verify" operation.

```
Source and Destination cannot be the same drive.
```

EXPLANATION: Different drive units must be specified as source and destination unit.

```
Unable to copy to this disk. It is write protected.
```

EXPLANATION: Disk should be write enabled by removing the adhesive tab from the write-enable notch in the disk jacket.

```
Command line syntax error
```

EXPLANATION: System Prompt Method command line was entered incorrectly. Re-enter using the entry form explained in "2.1 Command Line Entry."

```
Unknown command line option
```

EXPLANATION: System Prompt Method command line was entered with invalid characters used as options. Re-enter using only the options listed in "2.2 DUP Options."

```
Invalid Source Unit
```

EXPLANATION: The entry made at the "Source unit:" prompt did not correspond to a valid disk drive within the hardware environment. The DUP operation must be reselected, and a valid drive name letter entered.

```
Invalid Destination Unit
```

EXPLANATION: The entry made at the "Destination unit:" prompt did not correspond to a valid disk drive within the hardware environment. The DUP operation must be reselected, and a valid drive name letter entered.

ED

The Line Editing Utility that Creates and Edits Text Files

The ED utility enables you to compose, alter, and manipulate files containing ASCII characters. The files composed by ED are often referenced or manipulated by other system utilities and commands.

When invoking ED, you either create a new file or summon an old file into the computer (1). ED works on files using an area of computer memory known as the memory buffer (2). Files being edited can be moved, altered, or displayed when you enter various ED commands (3).

1 ED INVOCATION

The ED utility is invoked by the entry of a command line in the following form:

A>**ED {file name}** *RETURN*

Where {**file name**} is the complete name of a file that you wish to compose or edit. you must specify the name of the text file here.

If the file resides in a non-default drive, this drive should be specified immediately before the file name in the command line. If the file does not yet exist at all, ED will create it on the disk in the default drive, or on a disk in a specified drive.

The following entry, for example, would cause ED to open the file named "THISFILE.TXT", which resides on the disk in non-default drive B:

```
A>ED B:THISFILE.TXT   RETURN
```

When ED "opens" a file, it checks the disk directory for the name of the file. Then ED reserves the computer's Transient Program Area (TPA) as a memory buffer to be used for text editing and file manipulation.

If the name of the file is not in the disk directory, ED creates the empty file and displays both the "NEW FILE" message and the ": *" prompt. Then you can begin inserting text into the empty file by using the "I" command at the prompt.

If the name of the file does exist in the disk directory, then ED locates it, and displays the ": *" prompt on the console. You can then move the existing file into the memory buffer by using the "A" command at the prompt.

2 ED STRUCTURE AND FEATURES

Text sections 2.1, 2.2, and 2.3 explain the structure and features of the ED utility. It is recommended that you understand these concepts before trying to implement ED commands.

2.1 Text Files in the CP/M Environment

To be properly read from, written to, and transferred, CP/M text files must be composed entirely of American Standard Code for Information Interchange (ASCII) characters. Text files must end with the entry of the CTRL-Z "end-of-file" character. Text can be moved to and from the disk in units of "lines". A line is defined as a string of ASCII characters that ends with the carriage return and line feed characters. (The "carriage return and line feed" combination can be entered by pressing the RETURN key.)

2.2 The Memory Buffer

The memory buffer is an area in the computer's memory that the ED utility uses as a "scratch pad" on which to compose and edit text before it is transferred to a disk for storage.

You can send a specified number of lines of text to the buffer from the disk or send text characters into the buffer by entering them directly through the keyboard. The memory buffer in some computers can hold about 36,000 text characters at one time. When full, you can purge the buffer of text by writing a specified number of text lines from the full buffer to the disk.

2.3 The Character Pointer

Text in the buffer is usually arranged on numbered lines. To help you to access locations within the text, the buffer contains an invisible character pointer, which can be moved to specific locations within the text by various user commands.

The character pointer resides before the first character in the text, after the last character, or between any two text characters. When it is moved to a position in the text, a specified number of characters or lines can be inserted or deleted starting at the pointer's current position. Text is inserted through the use of special commands, sometimes followed by direct keyboard typing.

The character pointer is positioned by user commands that move it up or down to different lines, left or right along a line, or to the top or bottom of the file. The character pointer can also search through the text to locate a user-specified text string.

All ED commands are executed starting at the current position of the character pointer.

3 ED COMMANDS

Commands used within the CP/M EDitor are entered in response to the ":*" prompt, or to a prompt in the form "n:*" where "n" is the number of the text line upon which the character pointer resides. All ED commands (except those which end an ED session) can be entered in a series on the same command line. The last command entered on any line must be followed by a carriage return to initiate command processing.

The ED utility uses four commands (A, W, X, and R) to transfer text lines between the disk and the memory buffer (3.1).

When a file is in the memory buffer, four commands (B, C, L, and n) can be used to move the character pointer to a specified line or character (3.2).

Once the character pointer is in position, four commands (I, D, and K) are used to insert text into or delete text from the buffer (3.3).

Whenever text is within the memory buffer, two commands (T and P) can be used to display it to the screen. Three other commands (Z, V, and U) are used to alter characteristics of screen display (3.4).

When a file is in the memory buffer, four commands (F, S, N, and J) can be used to move the character pointer to an occurrence of a specified text string (3.5).

To remove an entire file from the memory buffer, four commands (E, Q, H, and O) can be used to send it from the buffer to the disk, or to dispose of it in some manner. These commands must be entered alone and followed immediately by a carriage return (3.6).

One command (M) is used to trigger multiple execution of other ED commands (3.7).

3.1 Moving Text to or from Memory Buffer

nA **Append lines from disk to buffer**
This command will copy "n" lines of text from the disk file specified in the invocation command to the memory buffer, where the text image can be edited.

The nA command must be implemented when you wish to edit text from an existing file. This command is entered in response to the ":*" prompt, and causes the "1:*" prompt to be displayed. The counterpart of the nA command, the nW command, transfers edited text lines back to the disk.

If you do not specify the number (n) of lines to be appended, ED will append one line from the disk. If the "#" character is entered in place of "n" (#A), then all of the text lines in the disk file will be copied to the memory buffer.

nW **Write lines from buffer to disk**
This command transfers "n" lines of edited text from the memory buffer to the disk. Text that is written to the disk in this manner will no longer exist in the memory buffer.

The W command is implemented when the memory buffer becomes full during text editing. It is entered in response to the ":*" or "n:*" prompt, and causes the ":*" prompt to be displayed.

The W command always starts at the top of the buffer, transferring the first line in the buffer through the "nth" line in the buffer. The buffer text line that occurs after the "nth" line then becomes the first line, moving up to the top of the buffer.

If you do not specify the number of lines to be transferred to the disk, one line will be transferred. If the "#" character is entered in place of "n" (#W), then all of the text lines in the memory buffer will be transferred to the disk, and the buffer will be empty.

As the edited text is written back to the disk, a few lines at a time, it accumulates in a temporary file that has a "$$$" extension.

nX **eXtradite text block from buffer to temporary library file**

This command transfers a block of text from the memory buffer to a temporary disk file, so that it can be transferred back to the buffer (by the R command) at a desired location.

The block of text begins with the line containing the character pointer, and ends "n" lines later. The block is stored on the disk in a temporary file which is automatically named $$$.LIB (the standard name for a temporary library file). After this file is created on the disk, you should implement other ED commands to move the character pointer to the text location at which the temporary file should be inserted.

When the character pointer is at the desired location, the R command is used to transfer the $$$.LIB file back to the buffer at the desired location. The letter "R" and a carriage return should be entered to insert the text at the current location. The same text can then be inserted at another text location by moving the character pointer and, again, entering "R" and a carriage return.

Before a different block of text is transferred to the $$$.LIB file, the "0X" (a zero followed by an X) command should be entered to clear the old text from the file.

If no number is specified in the space preceding the "X", then one line of text (starting from the current character pointer location) is transferred to the temporary file. If the "#" symbol is specified, then all of the text lines within the buffer which follow the character pointer are transferred.

Rf **Read library file f from disk to buffer**

This command copies the text from a disk library file to the memory buffer, inserting this text at the current location of the character pointer.

The file being read into the buffer should usually be specified in place of the "f" in the command "Rf". However, these files are always assumed to have the "LIB" extension. Therefore, you need only enter the primary file name in such a command. For example, the text from library file "ROUTINEX.LIB" could be read into the buffer with the entry of the following command:

RROUTINEX

If you wish to read a temporary library file into the buffer text, then no part of the file name need be entered with the R command. The standard temporary library file name "$$$.LIB" will be assumed.

3.2 Positioning the Character Pointer

+/− **B Beginning/Bottom of text character pointer movement**
This command will move the character pointer to the beginning of the first line of the text in the buffer (if entered in the form **B**), or to the end of the last line in the buffer (if entered in the form −**B**).

+/−**nL** **Line down/up character pointer movement**
This command will move the character pointer from its current line within the memory buffer text to another line a specified number (**n**) of lines away.

When the command is entered in the form **nL**, the character pointer will move ahead (down) the specified number of lines to the beginning of a text line.

When the command is entered in the form −**nL**, the character pointer will move backward (up) the specified number of lines to the beginning of a text line.

If the number of lines to move is not specified when the **L** command is entered, the pointer will move one line ahead (down).

In order to move the pointer to the beginning of the line upon which it currently resides, enter the "0L" command (with a zero preceding the "L").

+/−n **Advance to a line and display it**

This command moves the character pointer a specified number of lines (n) and displays the text of the line on which it lands. This command produces the same results as the simultaneous entry of both the "L" and "T" commands. If no number is entered before the carriage return, ED assumes the number one.

+/−nC **Character pointer movement to right/left**

This command moves the character pointer a specified number (n) of character spaces, usually toward the right or left edge of a text line. (When a command-driven character pointer reaches the edge of a text line, the carriage return and line feed characters will cause it to change its direction momentarily.)

When entered with a plus sign ("+"), this command will move the pointer the specified number of spaces to the right, and/or down to successive lines. When entered with a minus sign ("−"), this command will move the pointer the specified number of spaces to the left, and/or up to preceding lines. If you wish to move the character pointer past the edge of a text line using the "C" command, the number specified in the command will have to include two character spaces to get past the carriage return and line feed characters at the end of the line.

3.3 Inserting or Deleting Text

I **Insert characters from keyboard to buffer**

The "I" command enables you to insert characters directly into the text at the current position of the character pointer. This command is entered in response to the ": *" or "n: *" prompt.

If you enter a carriage return immediately after the "I" command, then the "n:" prompt appears on the next line and text characters can be inserted on successive lines until you enter a CTRL-Z "end-of-file" character. The entry of a CTRL-Z causes the ": *" prompt to be displayed. After such an insertion operation, the character pointer will be positioned at the end of the last inserted text line.

If text is inserted on the same line as the "I" command, then the insertion operation will end when the next carriage return is entered, and a "*" command prompt will appear at the left edge of the screen. After such an insertion operation, the character pointer will be positioned at the beginning of the line following the line of inserted text.

If upper and lower case insertion text is desired, enter the "I" command with a lower case "i". Entering the command with an upper case "I" will automatically translate all inserted text to upper case.

+/−nD Delete characters from buffer text

This command will delete a specified number (**n**) of characters from memory buffer text, starting at the location of the character pointer.

Deletions will take effect to the right of the pointer if the specified number of deletions is preceded by a " − " sign. If the specified number of deletions is preceded by a " + " sign or by nothing, then characters to the right of the pointer will be deleted.

If no number of characters is specified for the deletion operation, then one character will be deleted. If "#" is specified as the number of characters to be deleted, then all text characters before or after the pointer (depending on the sign preceding the number) will be deleted.

The carriage return and line feed characters at the end of each text line are counted as two separate characters, even though they are produced by pressing only the RETURN key.

+/−nK Kill lines from buffer text

This command will delete a specified number (**n**) of lines from the memory buffer text, starting at the position of the character pointer.

If a " + " (plus sign) precedes the number of lines to be killed, then that number of lines occurring **after** the character pointer will be killed. If a " − " (minus sign) precedes the number, then that number of lines occurring **before** the pointer will be killed.

If the character pointer is positioned in the middle of a line during a "K" command, the portion of the text line to the left or right of the pointer will be deleted as if it were one entire line.

If no number (**n**) is specified for the deletion operation, then **one** line will be deleted. If "#" is specified as the number of lines to be deleted, then **all** text lines before or after the pointer (depending on the sign preceding the number) will be deleted.

3.4 Displaying Text to Console

+/−nT **Type text lines on console**

This command will cause a console display of a specified number (n) of text lines, starting at the position of the character pointer.

If the character pointer is in the middle of a line, the portion of the line between the pointer and the end of the line will be counted as an entire line. If such a command begins with a minus sign ("-"), then the specified number of lines before the line containing the character pointer and the line containing the pointer are displayed. If the pointer is positioned in the middle of a line and a zero is specified in the command, then only the portion of the line from its beginning to the pointer will be displayed.

If no number (**n**) of lines is specified, then one line will be displayed. If the "#" symbol is specified, then all of the lines in one direction will be displayed.

You can interrupt a console display which scrolls too quickly by entering a ***CTRL-S*** character. The scroll will resume when another CTRL-S is entered. You can abort a long scrolling screen display by entering any other keyboard character while the display is in progress.

The "T" command will not effect the position of the character pointer. Therefore, at the end of a "T" operation, the pointer marks the position at which the operation began. This position will be indicated by the number in the "n:*" prompt.

The "T" command can be entered in response to a "n:*" or ":*" prompt.

nP **Page display on console**

This command causes text to be displayed on the video screen in units of one page (23 lines), and deposits the character pointer at the end of the display.

If a display of the first page (23 lines) beginning at the character pointer is desired, then a zero should precede the **P** in such a command. Hence, a **0P** command has the same effect as a **23T** command.

If the number one or no number is specified before the "P", then one page (23 lines) of text, starting 23 lines past the character pointer, will be displayed. The command **2P** will cause the display of two pages starting 23 lines past the pointer. The command **3P** will cause the display of three pages starting 23 lines past the character pointer, and so on. Hence, a **2P** command has the same effect as a **23L46T-23L**.

nZ **Zone interruption of text display scroll**

This command can be entered into a command line in front of the "T" or "P" display commands to interrupt a long console display scroll at time-regulated intervals so you can view the text one zone at a time.

When "T" and/or "P" commands are entered in a series within the same command line, "Z" commands can be placed in between to interrupt their execution for time periods determined by the number preceding the "Z".

The number (n) preceding the **Z** in the command stands for the number of half seconds that the display scroll will be interrupted. Hence, if a command line contains a **10Z** between two display commands, the scrolling caused by the display commands will be interrupted for five seconds.

+/−V **aVert or replace line numbers in console displays**
If you prefer not to use the line numbers, the **−V** command will eliminate them from the console display.

The command **V** will restore line numbers to the console display.

A special form of this command, in which a zero precedes the V (**0V**), will produce a display showing how many locations remain unused in the memory buffer (r), and the total number of buffer locations that are accessible through the ED utility (t). The display appears in the form: "r/t".

+/−U **Uppercase/lowercase text translation**
If you would like all characters entered into text through the ED utility to be put into uppercase, the "U" command can be entered.

The "−U" command can be entered to allow the inserted text to be displayed in both lower and uppercase.

3.5 Searching for Text Strings

nFt **Find text string t within buffer text**
This command is used to locate the specified number of occurrences (n) of a particular string of characters (t) within the text.

The string of characters (**t**) being sought is specified immediately after the **F** in the command,and ended with the entry of a ***CTRL-Z*** character and a carriage return.

If you do not specify the number of occurrences (n) of the string to be found within the text, then only the first occurrence will be found.

If you desire to locate a string of characters (t) that contains the carriage return and line feed characters, these two characters can be specified in the command line with the ***CTRL-L*** character.

The string specified in the command (t) must match the actual text in spelling, spacing, capitalization, etc.

The specification of a string of characters that is not found in the text will produce the error message:

```
BREAK "#" AT
```

sending the character pointer back to its position before the search operation failed.

nSdt **Search and replace text string**

This command performs the operations of the "F", "D", and "I" commands simultaneously, by finding a specified string (d) within the buffer text, deleting it, and inserting a second specified string (t) at the same location.

The number (n) in the command represents the number of text string substitutions desired by you throughout the text. The omission of this number will cause a substitution to be made only at the first occurrence of the sought-after string (d). The entry of the "#" symbol in place of this number will cause the substitution to be made at every occurrence of the sought-after string throughout the text.

The string of characters to be found and deleted in the buffer text (d) is specified immediately after the "S" in the command, and ended by a CTRL-Z character. Immediately after this CTRL-Z, the text string to be inserted (t) is entered and ended by a second CTRL-Z and a carriage return.

The specification of a string of characters that is not found in the text will produce the error message:

```
BREAK "#" AT
```

sending the character pointer back to its position before the failed search and replace operation.

nNt **fiNd text string on disk**

This command performs the same search operation as the "F" command except that it can search an entire file for a text string (t).

If the specified string (t) is not found in the memory buffer, then this command will automatically write the contents of the buffer to the disk (into a temporary file, as the "W" command does) and append an image of another portion of the disk file's text to the buffer (as the "A" command does) until the entire file has been searched for the string the specified number of times (n).

nJftd **Juxtaposition substitution and deletion**

This command finds a first string (f), inserts a second string (t) after the first, and then deletes all of the text between the end of the inserted string (t) and the beginning of the third string (d).

The "J" in the command is immediately followed by the first text string (f), a CTRL-Z, the text string to be inserted (t), a CTRL-Z, the third text string (d), a CTRL-Z, and a carriage return.

The third string (d) serves as a restraining boundary for the text deletion.

This multi-faceted operation is performed a specified number of times (n), or once, if no number is specified. If the "#" is specified, the operation will be performed for all occurrences of the first text string (f).

If the third command line string cannot be found in the buffer text, then no text is deleted.

3.6 Closing a Text File

E **End session while buffer text becomes permanent disk file**
All text in the memory buffer is copied to the disk, where it is combined with any text that has accumulated in a temporary file, and assigned the original file name.

At the same time, the version of the file that was copied to perform the edit is assigned the extension ".BAK" in place of its original extension.

The operating system then regains control and displays the system prompt.

Q **Quit session by deleting edited copy of file**
All text in the memory buffer and/or any temporary file created during the session is deleted, and any existing versions of the file on the disk maintain the status and names they held prior to the editing session.

If the file being edited existed before the session, then the original version remains intact, as if the editing session never took place.

If the file in the buffer is an original composition (was created as a "NEW FILE" during this editing session) then all copies or versions of this file will be destroyed.

Since the accidental use of this command could delete important text composed or edited during the session, its entry will produce the "Q- (Y/N)?" confirmation message. The **Y** character must be entered before the deletion will be executed. If the **N** is entered, the current editing session will continue.

H **Halt session temporarily to save alterations**
All alterations made to text (or any text composed) will be saved under the active file name, and the editing session will continue with an image of the currently edited file automatically appended to the memory buffer. This command has the same effect as entering a combination of both the "E" and "A" commands.

O **Omit recent alterations and restart edit session**

Any text in the memory buffer or in a temporary disk file is deleted, and the editing session continues, using the same text with which it began.

In effect, this command nullifies any text alterations or composition performed in the most recent ED session and starts the session over, as if the "Q" and "A" commands had been entered consecutively.

3.7 Causing Multiple Command Execution

nM **Multiple command execution**

This command allows you to execute one or more commands a specified number of times (n) without additional command entries. Commands are entered on the same line, following the **nM** command in a string terminated by either a carriage return or a CTRL-Z.

All commands following the **nM** will be executed the number of times specified at the beginning of the entire command line (n), or until an error condition is encountered. If no number (n) is specified, then the operations invoked by the command line will be implemented from the position of the character pointer through the end of the text, or until an error condition is encountered.

This command is commonly used with the search and replace command ("nS"), to facilitate text string substitution throughout a large text area. When such a search reaches the end of the text in the memory buffer, an error condition to indicate that the substitution can no longer be executed.

Multiple commands are executed from the position of the character pointer toward the end of the text. Hence, the pointer should be positioned at the beginning of the buffer text if multiple commands are to be executed throughout the text.

4 THE FILE EDITING CYCLE

The following sequence of diagrams shows the file named "REPORT.DOC" as it undergoes ED's file editing cycle. The left side of the diagrams represents the memory buffer, and the right side of the diagrams represents the logged disk.

In Figure 2-1, you have opened the file "REPORT.DOC" by entering **ED REPORT.DOC** at the "A>" system prompt. You then enter ED's I command, and begin typing text into the memory buffer file. No text has yet been recorded on the disk.

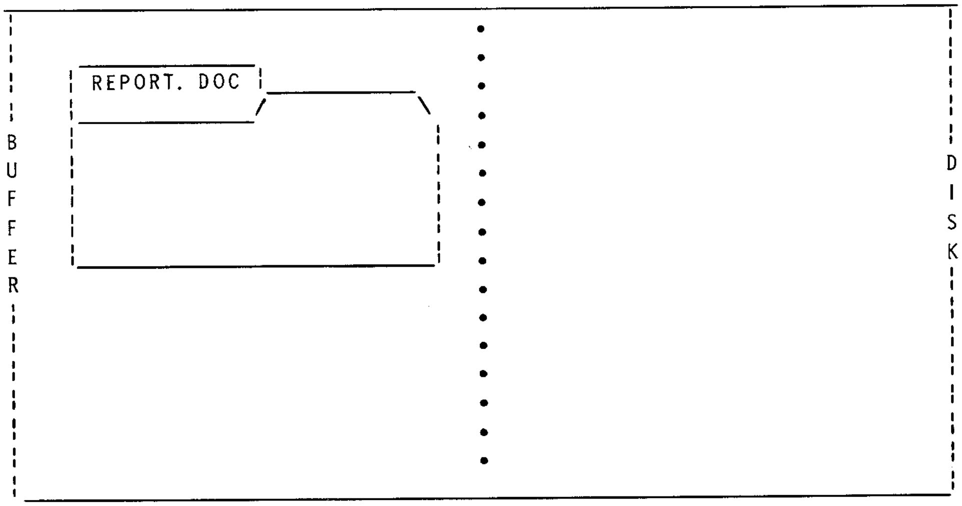


Figure 2-1

In Figure 2-2, you have inserted text into the file, and want to save some number of text lines. First you end the insert by entering a CTRL-Z. Then to save the text, you enter ED's **W** command to send a specified number of text lines from the buffer to a temporary file on the disk. ED gives this temporary file the name "REPORT.$$$". The buffer file remains.

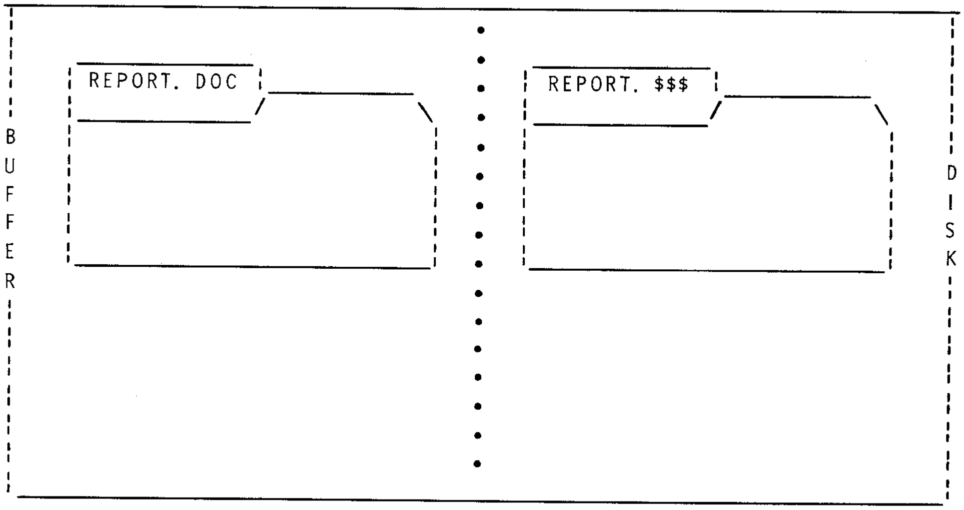


Figure 2-2

In Figure 2-3 you end the editing session and save all of the text composed for "REPORT.DOC" by entering ED's **E** command. This saved text is recorded on the disk under the file name "REPORT.DOC". Both the buffer file "REPORT.DOC" and the temporary disk file "REPORT.$$$" are erased.

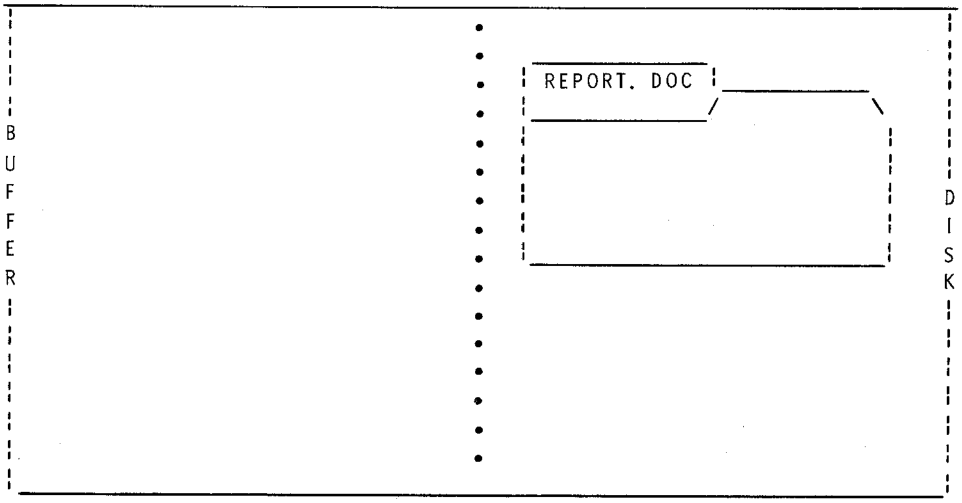


Figure 2-3

In Figure 2-4, you are reopening the file "REPORT.DOC" to edit it. First you enter **ED REPORT.DOC** to invoke ED. Then you must enter ED's **A** command to bring a copy of a specified number of text lines from the disk into the buffer. The disk copy of "REPORT.DOC" remains on the disk.

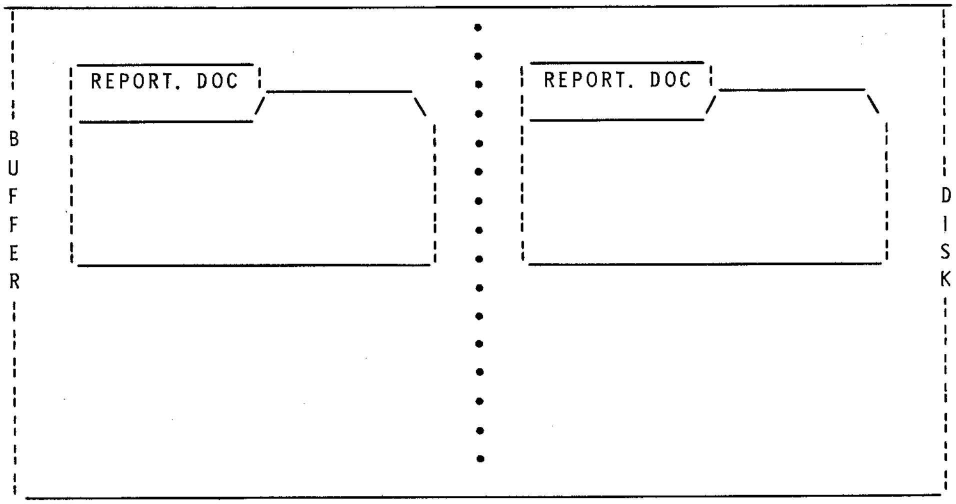


Figure 2-4

You changed and/or added to the text lines that were brought into the buffer. To save these changes and/or additions, you again enter a CTRL-Z and ED's **W** command. Figure 2-5 reflects the state of the files after these activities. The original disk copy of "REPORT.DOC" remains on the disk, a copy of some of the files text remains in the buffer under the name "REPORT.DOC", and some number of edited text lines from the file have been sent to the temporary disk file named "REPORT.$$$".

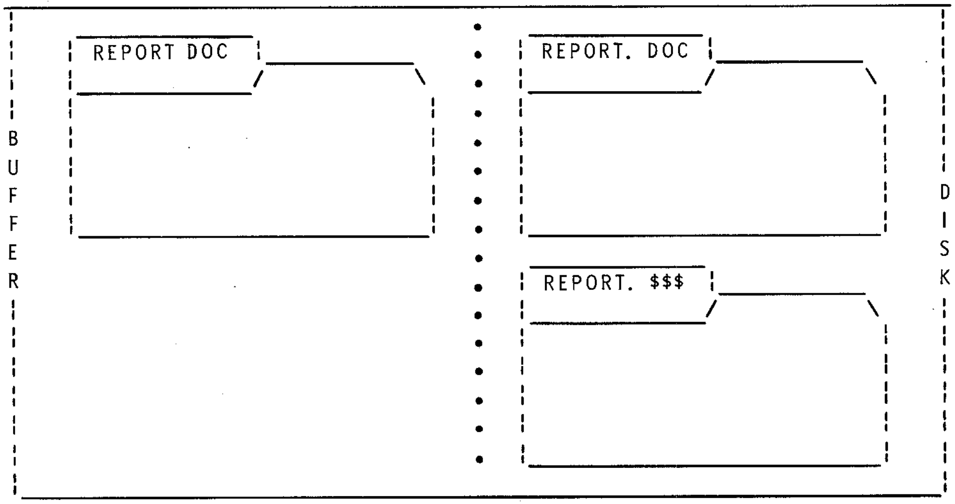


Figure 2-5

In Figure 2-6, you have closed the file editing session and saved the change and/or additions made to the text, by entering ED's **E** command. ED reacts to this entry by combining all text from buffer file "REPORT.DOC", all text from temporary disk file "REPORT.$$$", and any unchanged text from the original disk file "REPORT.DOC" into a new disk file named "REPORT.DOC". Then ED renames the original "REPORT.DOC" disk file to "REPORT.BAK". The buffer file and the temporary disk file disappear.

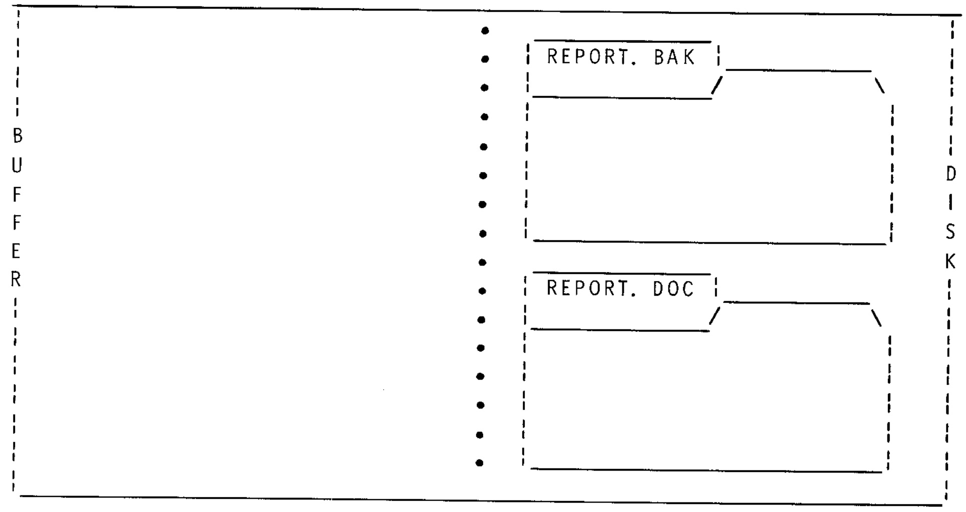


Figure 2-6

Thus a file with a file name extension assigned by you is usually the latest version of the file, whereas a file with the extension "BAK" is usually an old copy of the file.

If you re-edit the disk file "REPORT.DOC" and save the newly-edited version, the current disk file "REPORT.DOC" will become the new disk file "REPORT.BAK", as the memory file "REPORT.DOC" and the disk file "REPORT.$$$" combine to become the new disk file "REPORT.DOC". The old disk file "REPORT.BAK" will be automatically deleted after the third ED session with the file.

NOTE: These diagrams only illustrate a few of the basic options for file manipulation. The text on ED commands explains several other options.

5 ED ERROR MESSAGES

```
BREAK"?" AT x
     : *
```

EXPLANATION: ("x" is an invalid character that you entered.) you entered an ED command under inappropriate circumstances, at the wrong kind of prompt, or with improper syntax. Command should be re-entered.

```
DISK OR DIRECTORY FULL
```

EXPLANATION: You either entered **ED** without a file name argument at the CP/M system prompt, or composed more text than the disk could hold. For the former, you should invoke the ED utility by typing a command line with a file name argument. For the latter, you cannot save the overflow of text. When composing future documents, you should use ED's "W" command or CP/M's STAT utility more often.

```
** FILE IS READ/ONLY **
```

EXPLANATION: You tried to save newly-edited text to a file that cannot be written to because it has read/only status. You should abandon the text in the buffer with ED's "Q" exit command, and then use CP/M's STAT command to assign read/write status to the file before the next edit.

```
"SYSTEM" FILE NOT ACCESSIBLE
```

EXPLANATION: You tried to edit a file that had been given the "system" status by the STAT command. This status hides a file from commands such as ED and DIR. You should assign the "directory" status to the file, using the STAT command, before trying to edit the file again.

```
NO MEMORY
```

EXPLANATION: You have filled memory buffer, and should use the "W" command to send some of the buffer text to the disk.

ERA

The Resident Command for Erasing Files

The ERA resident command frees space once occupied by disk files, thereby permitting the storage of new files. ERA will erase a single file (1) or a group of files (2). ERA also displays a message when it cannot erase a file (3). Caution should be exercised when using ERA because erased files cannot be recovered under most circumstances.

1 ERASING A DISK FILE

To erase a file from a disk, you should enter a command line in the following form:

A>**ERA {filename.ext}** ***RETURN***

Where {**filename.ext**} is the complete name of the file you want to delete.

Files that do not reside on a disk in the default drive can only be erased when their file name is preceded in the command line by the appropriate drive specification.

2 ERASING GROUPS OF FILES

A group of files with similar names can be deleted by a single ERA command line when ambiguous file names (names with the "*" or "?" characters) are used, as in the following example:

```
A>ERA B:PROGRAM?.PRN RETURN
```

which would delete files with names such as PROGRAM1.PRN, PROGRAM2.PRN, PROGRAM7.PRN, and PROGRAM/.PRN from the disk in drive B.

The following command example shows how to delete even more files at once:

```
A>ERA B:*.* RETURN
```

This entry would erase every file from a disk. Because of the destructive potential of this form of the command, erase will ask you for confirmation with this message:

```
ALL FILES (Y/N)?
```

The command will not be executed until you confirms it by pressing the "**Y**" key. If the "**N**" key is pressed, the system prompt will be displayed.

3 CONSOLE RESPONSE TO ERA COMMANDS

When ERA finds the specified file and erases it, the system prompt returns.

If the specified file does not reside on a disk in the specified drive, then the console will display the following message:

```
NO FILE
```

If the file you desire to erase is write protected by the "R/O" (read only) status, or if you switched disks between drives without performing a warm boot, a message like the following will be displayed:

```
Bdos Err On x: File R/O
```

Where "x" is the letter of the drive containing the write-protected file.

If the disk containing the file to be erased is mechanically write protected (with adhesive tabs for 5.25-inch, without adhesive tabs for 8-inch, or with disk drive cabinet switches for 8-inch and Winchester Disks), then a message in the following form will be displayed:

```
Bdos Err On x: Bad Sector
```

Where "x" is the drive from which you tried to erase a file.

If you specified a drive in an ERA command, and that drive contains no disk, then a message in the following form will be displayed:

```
Bdos Err On x: Select
```

Where "x" is the drive from which you tried to erase a file.

FORMAT

The Utility that Prepares the Disk Surface

FORMAT prepares a floppy disk or a Winchester Disk partition for the storage of data by establishing storage areas on the disk surface. At the same time, FORMAT erases any data that remains on the disk from prior use, and sometimes inspects the recording surface for imperfections that could impair data storage or transmission. FORMAT also enables you to determine how much data you will be able to store on the disk.

CAUTION: Because FORMAT erases all existing data on a disk, make certain that you only format blank disks or disks containing expendable data. You can use the DIR (refer to the text titled "DIR") or STAT (refer to the text titled "STAT") commands to check a disk for valuable data files before formatting it. Notice, however, that the DIR and STAT commands cannot always find all of the data on a disk.

You can use the FORMAT utility through either of two methods: the FORMAT Prompt Method or the System Prompt Method.

1 FORMAT PROMPT METHOD

With this FORMAT method, you load the FORMAT utility into memory by entering a command at the system prompt. Then you answer a series of FORMAT prompts to define the formatting operation.

1.1 FORMAT Invocation

Answer the system prompt with a command in the following form:

```
A>FORMAT RETURN
```

When invoked through the FORMAT prompt method, FORMAT first identifies itself with name, version number, and a caution about its capabilities. Then it asks you if you wish to continue the operation, as shown:

```
Format Version 2.04
This program is used to initialize a disk.
All information currently on the disk will be destroyed.
Is that what you want? (y/n):
```

If you wish to prepare a disk, enter a Y at this prompt and continue reading.

If you do **not** wish to prepare a disk, enter any character other than a Y. Control will return to the operating system, which will display a system prompt.

1.2 Specifying The Disk to be Formatted

After you have confirmed your intention to format a disk, FORMAT asks:

```
Which drive do you wish to use for this operation?:
```

Answer this prompt by entering the letter of the drive containing the disk you wish to prepare.

1.3 Disk Format Specification

Heath/Zenith disk drive models accommodate four different kinds of disk:

- 5.25-inch hard-sectored floppy disks;
- 5.25-inch soft-sectored floppy disks;
- 8-inch floppy disks; and
- Winchester Disk

Each of these disks types are formatted differently. Once you have specified your drive, the messages and prompts that FORMAT displays depend on the type of the disk you are formatting.

FORMATTING A 5.25-INCH HARD-SECTORED FLOPPY DISK

After you have selected a drive for this FORMAT operation, FORMAT prompts you as follows:

```
Put the disk you wish to be formatted in drive x.
Press RETURN to begin, anything else to abort.
```

Entering a carriage return at this prompt will begin the actual formatting operation, while entering any other keyboard character will cause all FORMAT activities to end as CP/M displays the system prompt.

When the operation is finished, FORMAT will display:

```
Do you have any other disks to be formatted?  (y/n):
```

After formatting, this disk has a file capacity of 90 kilobytes.

FORMATTING A 5.25-INCH SOFT-SECTORED FLOPPY DISK

After you select a drive for this FORMAT operation, FORMAT displays the prompt:

```
Which density? (S=single, D=double):
```

The "density" of a disk refers to the concentration of data on its surface. You can decide whether data will be stored on the disk at "single" density or "double" density by entering "S" or "D" at this prompt.

If you specify "D" for "double" density, data will be recorded on the disk at approximately twice the concentration as it would at "single" density. Higher density levels might also decrease data access reliability.

After the density is selected, FORMAT will display the prompt:

```
Number of Sides? (1=single, 2=double):
```

Since 5.25-inch soft-sectored disks have two usable sides, you have the option of selecting how many of these sides will be prepared to store data.

After you enter a valid response to each of these prompts, FORMAT will display either of the following two messages:

```
48 TPI drive -- 40 tracks will be formatted

Put the disk you wish to be formatted in drive x.
Press RETURN to begin, anything else to abort.
```

or

```
96 TPI drive -- 80 tracks will be formatted

Put the disk you wish to be formatted in drive x.
Press RETURN to begin, anything else to abort.
```

Entering a carriage return at this prompt will begin the actual formatting operation, while entering any other keyboard character will cause all FORMAT activities to end as CP/M displays the system prompt.

When the operation is finished, FORMAT will display the message:

```
Do you have any more disks to format? (y/n):
```

NOTE: Soft-sectored 5.25-inch disks supported by the Heath/Zenith CP/M are available in two varieties: those with 48 tracks per inch (48 tpi) and those with 96 tracks per inch (96 tpi). Each variety has different storage capacities, depending on the density and number of sides you select when formatting.

The following table shows the file capacities of 5.25-inch, soft-sectored, 48 tpi disks:

	Single-sided	Double-sided
Single density	90 kilobytes	190 kilobytes
Double density	148 kilobytes	308 kilobytes

The following table shows the file capacities of 5.25-inch, soft-sectored, 96 tpi disks:

	Single-sided	Double-sided
Single density	190 kilobytes	388 kilobytes
Double density	308 kilobytes	624 kilobytes

NOTE: The CP/M Operating System is set to accommodate either 48 tpi disks or 96 tpi disks in a Z-37 drive model. This setting can be adjusted through the CONFIGUR utility. (See "Submenu B: Set Disk Parameters" in the text on CONFIGUR.)

FORMATTING AN 8-INCH DISK

After you select the drive to be used to FORMAT this disk, the FORMAT utility will display one of the following prompts:

```
Which density? (S=single, D=double, E=extended double):
```

or

```
Which density? (S=single, D=double):
```

The "density" of a disk refers to the concentration of data on its surface. You can decide whether data will be stored on the disk at "single" density or "double" density or "extended double" density by entering "S" or "D" or "E" at this prompt. If you specify "D" for "double" density, data will be recorded on the disk at about twice the concentration as it would at "single" density. Extended double density is the highest level of data concentration possible on that disk. Higher density levels might also decrease data access reliability.

After you answer the density prompt, FORMAT displays:

```
Put the disk you wish to be formatted in drive x.
Press RETURN to begin, anything else to abort.
```

Entering a carriage return here will begin the formatting operation, while entering any other keyboard character will cause all FORMAT activities to end as CP/M displays the system prompt.

When the operation is finished, FORMAT will display the message:

```
Do you have any more disks to format? (y/n):
```

Heath/Zenith CP/M supports two varieties of 8-inch disk: single-sided and double-sided. Each variety has different storage capacities, depending on the density and number of sides you select during formatting.

The following table shows the file capacities of single-sided 8-inch disks:

Single density	Double density	Extended double density
241 kilobytes	482 kilobytes	596 kilobytes

The following table shows the file capacities of double-sided 8-inch disks:

Single density	Double density	Extended double density
490 kilobytes	980 kilobytes	1208 kilobytes

NOTE: The 8-inch floppy disk drive slot of the H/Z-67 drive model cannot be used to FORMAT a disk to extended double density. Furthermore, an 8-inch disk that is formatted to extended double density cannot be used in the H/Z-67 model.

FORMATTING A WINCHESTER DISK PARTITION

After you specify the drive name assigned to the partition, FORMAT displays the prompt:

```
Press RETURN to begin, anything else to abort.
```

Entering a carriage return here will begin the formatting operation, while entering any other keyboard character will cause all FORMAT activities to end as CP/M displays the system prompt.

When the formatting operation begins, the following message is displayed:

```
Formatting partition
```

When the operation is finished, FORMAT will display the message:

```
Do you have any more disks to format? (y/n):
```

1.4 Ending a FORMAT Operation

When FORMAT finishes preparing the surface of a disk or partition, it will display the following prompt:

```
Do you have more disks to format? (y/n):
```

The specified disk or partition is now ready for data storage.

If you wish to FORMAT another disk or partition immediately, press **Y** at this prompt.

If you have no other disks to FORMAT, press **N** at this prompt and FORMAT will display the following message:

```
Place a bootable disk in drive x and press any character:
```

Where "x" is the letter for the default drive.

If the disk that was used to perform bootstrap has been removed from drive A since bootstrap, then it should be inserted back into drive A now. The entry of any keyboard character will transfer control back to the operating system, which will display a system prompt.

2 SYSTEM PROMPT METHOD

The System Prompt Method enables you to include all of the specifications necessary for the FORMAT operation in a single command line entered at the CP/M system prompt.

2.1 Command Line Entry

FORMAT commands that are entered by the System Prompt Method should be entered in the following form:

A>**FORMAT {drive}:{[option,option]}** ***RETURN***

Where **FORMAT** is the command line function

where {**drive**} is the letter of the drive that contains the disk you wish to format (this letter must represent a valid drive in your hardware environment, such as **A, B, C, D, E,** or **F**); and

where {**[option,option]**} represents letters and/or numbers enclosed in square brackets [] and separated by commas , to specify how the formatting operation should be conducted.

2.2 FORMAT Options

FORMAT command lines entered by the System Prompt Method can include the following options:

SD Disk formatted to Single Density.

DD Disk formatted to Double Density.

1S One Side of disk formatted.

2S Two Sides of disk formatted.

F Fast formatting, because the routine test of disk surface media is not performed.

N No prompt displayed between FORMAT command entry and FORMAT execution.

T1 Type of disk to be formatted must be a 5.25-inch hard-sectored disk.

T3 Type of disk to be formatted must be a 5.25-inch soft-sectored disk.

T4 Type of disk to be formatted must be an 8-inch disk, within a Z47 or H47 drive model.

TH6 Type of "disk" to be formatted must be a partition on the Winchester Disk within a Z67 drive model. This partition must have been assigned a drive name by the ASSIGN utility.

TF6 Type of disk to be formatted must be an 8-inch disk, within the floppy disk slot of a Z67 drive model.

NOTE: It is not necessary to enter an option for disk type (T1, T3, T4, TH6, TF6) in a System Prompt Method FORMAT command, because FORMAT automatically checks the type of the disk within the specified drive.

However, disk type should be specified when you desire to format a disk **only if** it is of a particular type. If a disk type **is** specified as an option, and the drive specified for this operation contains a disk of a different type, then the following error message is displayed:

```
DISK IS NOT OF TYPE SPECIFIED
```

2.3 FORMAT Defaults

When you enter a FORMAT command line with a drive specification, and neglect to specify some or all of the possible options, FORMAT will prepare the disk according to the following default criteria:

- Disk formatted to Single Density (as if you specified the SD option);
- One side of disk formatted (as if you specified the 1S option);
- Disk surface will be tested for data retention (as if you did **not** specify the F option); and
- Prompts will be displayed between FORMAT command entry and FORMAT execution (as if you did **not** specify the N option)

Unless the N option is entered in the command line, the following prompt will be displayed after you enter the FORMAT command:

```
Format Version 2.04
This program is used to initialize a disk.
All information currently on the disk will be destroyed.
Is that what you want? (y/n):
```

To confirm your intention to run a FORMAT operation, enter a **Y** at this prompt. FORMAT will display a prompt in the form of the example below.

To abort the FORMAT utility at this point, enter any other keyboard character. The operating system will take control and display a system prompt.

```
Put the disk you wish to be formatted in drive x.
Press RETURN to begin, anything else to abort.
```

To begin execution of the FORMAT operation, enter a carriage return at this prompt.

To abort the FORMAT utility, enter any other keyboard character. The operating system will take control and display a system prompt.

2.4 System Prompt Method Examples

A>**FORMAT B:** ***RETURN***

FORMAT will prepare the surface of the disk in drive B. By default, this disk will be formatted to single density on only one side. Also by default, FORMAT will display prompts before formatting, and test the disk surface while formatting.

A>**FORMAT B:[DD,2S,T4]** ***RETURN***

FORMAT will prepare the surface of the disk in drive B to double density and on both sides, as specified by options. FORMAT will display prompts before formatting and test the disk surface while formatting, by default. However, if drive B is not a Z47 or H47 drive containing an 8-inch disk, an error message will be displayed and no formatting operation will occur.

A>**FORMAT B:[SD,1S,DD,2S]** ***RETURN***

If your command line contains contradictory options, FORMAT will acknowledge the last one. Hence, in this case, FORMAT will prepare the surface of the disk in drive B to double density, as specified by the last density option (DD), and on both sides, as specified by the last side quantity option (2S). FORMAT will display prompts before formatting and test disk surface while formatting, by default.

A>**C:format B:[Dd,f,N]** ***RETURN***

The FORMAT utility, in this case, is stored on the disk in non-default drive C. It will prepare the surface of the disk in drive B to double density, as specified by the "Dd" option. But since no option is specified for the number of sides, only one side will be formatted (by default). The "f" option specifies that this formatting operation will be performed without a disk media test. The "N" option specifies that FORMAT will not prompt you to confirm your intentions before the formatting operation begins.

NOTE: FORMAT command lines can be edited (by pressing the ***DELETE*** key) or erased entirely (by holding down the ***CTRL*** key and pressing the ***X*** key).

3 FORMAT ERROR MESSAGES

```
Drive not available in current configuration.
```

EXPLANATION: If you entered a drive name that does not exist in the hardware environment, then a different drive name should be entered. If you entered a drive name for which your copy of the CP/M Operating System has not yet been customized, then MAKEBIOS or CONFIGUR must be run before FORMAT will accept the drive name entry. If you are using a Z67 drive model and entered the drive name for a partition that has not yet been established by the ASSIGN utility, then ASSIGN must be run before FORMAT.

```
Unable to format this disk. It is write protected.
Do you have any more disks to format? (y/n):
```

EXPLANATION: If the disk is a 5.25-inch disk you should remove the adhesive tab from the notch on the disk cover. If the disk is an 8-inch disk the you should cover the notch on the disk cover using the tabs provided, and push down the write protect switch on the front of the H/Z47 or H/Z67 cabinet.

```
Unable to format this disk. Place a different disk in the drive
and press any key to begin...
```

EXPLANATION: The disk to be formatted is damaged or improperly inserted in the drive. You should try the operation again and replace the disk if the message appears again.

```
Media error
```

EXPLANATION: The disk to be formatted is damaged or improperly inserted in the drive. You should try the operation again and replace the disk if the message appears again.

```
Wrong type of media, or media inserted improperly,
or media damaged.
```

EXPLANATION: You may have tried to FORMAT a hard-sectored disk in a soft-sectored drive, or vice versa. You should check that the proper type of disk is being used. If the proper disk type is being used, then the disk is probably damaged and a different disk should be used for the operation.

```
ILLEGAL FORMAT OPTION
```

EXPLANATION: System Prompt Method command line was entered with undefined characters in place of options.

```
ILLEGAL COMMAND SYNTAX
```

EXPLANATION: System Prompt Method command line was entered with undefined characters in place of options.

```
DISK IS NOT OF TYPE SPECIFIED
```

EXPLANATION: A System Prompt Method command line specified a drive that contained a disk which did not match the specified disk type.

```
OPTION NOT AVAILABLE
```

EXPLANATION: A System Prompt Method command line included option characters which were not possible under the circumstances. Re-enter command with pertinent options.

```
Disk is not partitioned. The complete disk surface will be formatted.
Is that what you want? (y/n):
```

EXPLANATION: The Winchester Disk has not yet been partitioned. Therefore, any FORMAT operation will erase and prepare the surface of the entire Winchester Disk. To continue FORMAT operation, press Y. If you wish to partition the Winchester Disk before formatting, run the PART utility as instructed in the PART manual.

```
PARTITION IS SMALLER THAN MINIMUM ALLOWABLE SIZE
```

EXPLANATION: Winchester Disk partitions must contain a minimum number of sectors before they can be prepared by FORMAT. Either ASSIGN a partition containing this minimum quantity of sectors, or use the PART utility to repartition the Winchester Disk with few enough partitions so that at least one is of minimum size. Refer to the Z67 manual and the PART manual to determine the minimum allowable size for a partition.

```
PARTITION IS LARGER THAN CP/M MAXIMUM SIZE -- ONLY 8 MEGABYTES USABLE
```

EXPLANATION: Winchester Disk partitions should contain no more than eight megabytes of storage capacity if used with the CP/M Operating System. If a partition contains more, it cannot be formatted. Either ASSIGN a partition of suitable size for formatting purposes, or repartition the Winchester Disk with the PART utility and form a partition within the eight megabyte limit. Then format the properly sized partition.

LIST

The Utility that Prints File Contents on Paper

The LIST utility enables you to obtain paper copies of files by entering a command (1) for one or more files (2) to be printed. Special printout characteristics can be set when you enters the command with LIST parameters (3). You can stop a LIST printout in progress (4). LIST is used to print out only certain types of files (5).

1 METHODS OF ENTERING LIST COMMANDS

Two different methods can be used to enter LIST commands: the LIST Prompt Entry Method and the System Prompt Entry Method.

1.1 LIST Prompt Entry Method

With the LIST Prompt Entry Method, you enter LIST in response to the system prompt. This entry is sufficient to invoke LIST, which displays its own prompt—the asterisk (*). You can now enter the argument portion of the command line in response to the "*" prompt supplied by LIST. LIST prompt entries are made in the following form:

A>**LIST** ***RETURN***

{argument}** ***RETURN

Where {**argument**} is the name of the file(s) to be listed.

After the LIST operation is finished, LIST again displays the "*" prompt. You can enter another argument or return to the operating system by entering a carriage return.

1.2 System Prompt Entry Method

To invoke a similar operation without splitting up the command line, you must include the argument in the response to the system prompt, as in the following example:

A>**LIST {argument}** ***RETURN***

Where {**argument**} is the name of the file(s) to be listed.

After LIST finishes the latter printing operation, it automatically returns you to the operating system, and the system prompt is displayed.

EXAMPLE OF LIST USAGE

The following command, entered using the LIST Prompt Method:

```
A>LIST RETURN

*REPORT.DOC RETURN
```

and this command, entered using the System Prompt Method:

```
A>LIST REPORT.DOC RETURN
```

will produce the same results. Both commands will produce a paper copy of the file named "REPORT.DOC".

2 PRINTING CONTENTS OF MORE THAN ONE FILE

Any number of files can be specified in a single LIST command. To initiate a listing of several files, the names of the files are entered in the argument separated by single spaces.

If the files to be listed reside on different disks, their names should be preceded by a drive specification (drive letter and colon).

Both of the following examples demonstrate how to LIST the contents of the specified files:

```
A>LIST RETURN

*PRINTOUT.DOC B:PROGRAM.PRN B:REPORT.DOC C:SYSTEMX.PRN RETURN
```

or

```
A>LIST PRINTOUT.DOC B:PROGRAM.PRN B:REPORT.DOC C:SYSTEMX.PRN RETURN
```

3 LIST PARAMETERS

You can specify parameters in a LIST argument to alter the characteristics of a standard printout. These parameters allow you to select printout characteristics such as the date, the number of copies desired, the width of tabs, etc. If no parameters are specified, LIST will print a document with default value characteristics.

The parameters used to specify printout characteristics are as follows:

PARAMETER NAME	KEYBOARD ENTRY	DESCRIPTION OF PRINTOUT CHARACTERISTIC	DEFAULT VALUE
Date	**[D xxx...x]**	First 10 characters of specified date printed on upper left corner of each document page.	none
Heading	**[H xxx...x]**	First 60 characters of specified heading printed on top line of each document page.	file name
No Heading	**[N]**	No heading or date printed on any document page.	file name
Lines per Page	**[L nn]**	Each document page has the specified number of lines. Specification of zero/page lines causes continuous printing.	60 lines per page
Tab Stop Width	**[T nn]**	Each tab stop within text is expanded or contracted to specified number of spaces.	8 spaces

PARAMETER NAME	KEYBOARD ENTRY	DESCRIPTION OF PRINTOUT CHARACTERISTIC	DEFAULT VALUE
Page Number	**[P nn]**	Page numbering sequence begins with specified number on first file page.	page 1
Upper Case	**[U]**	All letters in document printed in upper case.	upper and lower case
Copies Desired	**[C nn]**	Specified number of copies are printed.	1 copy
Erase	**[E]**	File is erased from disk after LIST operation is completed.	file retained on disk

USE OF LIST COMMAND LINE PARAMETERS

Parameters are entered, enclosed in square brackets, after the last file name in the LIST command. If more than one parameter is entered, each should be enclosed in a set of brackets.

A LIST command entered with parameters using the LIST Prompt Method might appear as follows:

A>**LIST** ***RETURN***
PRINTOUT.DOC PROGRAM.PRN [H Today's Work] [D 31-Feb-81] [P 09]** ***RETURN

This command will cause the contents of the files PRINTOUT.DOC and PROGRAM.PRN to be printed, with the following heading across the top of the first page of each file:

```
Today's Work                                          31-Feb-81  Page  9
```

The parameters will take effect on all of the files specified in that command line.

When the command is entered using the LIST Prompt Entry Method, any parameters entered (except for the starting Page Number parameter and Erase parameter) will remain in effect with any files specified at successive asterisk (*) prompts, until new values are entered for the parameters, or until control is returned to the operating system.

When a LIST command line is entered using the System Prompt Method, letters in the heading, date, and page number line might be automatically translated into upper case. The following System Prompt command demonstrates this character translation:

A>**LIST PRINTOUT.DOC PROGRAM.PRN [H Today's Work] [D 31-Feb-81] [P 09]** ***RETURN***

This command will cause the contents of the files PRINTOUT.DOC and PROGRAM.PRN to be printed with the following heading across the top of the first page of each file:

```
TODAY'S WORK                                               31-FEB-81  PAGE  9
```

4 ABORTING A LIST OPERATION

After a LIST command has been entered, the printout can be aborted by pressing any keyboard character. When you abort a command that was entered by the LIST Prompt Method, the next printout might also be automatically aborted soon after it begins. Therefore, it is advisable to exit from, and reinvoke, the LIST utility after aborting a printout.

5 FILES THAT SHOULD BE LISTED

Only files containing ASCII characters should be listed. ASCII character files include files composed using a text editor (such as ED), or file with the "PRN" extension created by an assembler (such as ASM).

Files such as those bearing the extension "COM" or "HEX" will produce meaningless printouts if LISTed because they are not composed of ASCII characters.

Files composed in a word processor will LIST, but they might possess features (such as bold face characters or page breaks) that LIST will not print in the same way as the word processor's printout command.

6 LIST ERROR MESSAGES

```
File not found
```

EXPLANATION: A file specified in the LIST command argument does not exist on the disk in the logged drive. If the file does exist on a disk, that disk should be inserted in a drive and the appropriate drive should be specified before the file name in the argument.

```
Syntax error in command line
```

EXPLANATION: The LIST command was improperly entered. Often occurs if a space is not entered between file names, or when an invalid parameter is specified.

LOAD

The Utility that Loads a "HEX" File for Execution

The LOAD utility puts an assembled hexadecimal file in the Transient Program Area (1), and translates the file into a "COM" file, which is executable under the CP/M operating system (2). Only certain types of files can be loaded (3).

1 LOAD INVOCATION

To LOAD a hexadecimal file into the Transient Program Area (TPA), you should respond to the system prompt with an entry in the form:

A>**LOAD {hex file}** ***RETURN***

Where {**hex file**} is the primary file name of an Intel hexadecimal file. The file extension is omitted from this entry because LOAD always assumed the extension to be "HEX".

This command translates the hex file into machine code that can be executed under CP/M, gives it a "COM" extension, and writes it back to the disk.

2 EXECUTING A LOADED PROGRAM

You can start execution of the program by entering the primary name of the file (and a carriage return) in response to the system prompt. Thus the file is executed as if it were a regular CP/M utility.

It is only necessary to LOAD a hex file once. The "COM" version of the file will be stored on the disk from which the "HEX" version came. Thereafter, CP/M treats it like another "COM" file.

The operation can take place on a non-default drive if the file name is prefixed by a drive name. Thus the entry of the following command:

```
A>LOAD B:BETA RETURN
```

brings the LOAD program into the TPA from the default disk and then operates on the file "BETA.HEX", which resides in drive B. The file named "BETA.COM" is written to the disk in drive B. (The file "BETA.HEX" will also remain on the disk.) Now you can execute BETA.COM be responding to the system prompt as shown:

```
A>BETA RETURN
```

Thus, you can "invent" new CP/M commands by using LOAD to translate a "HEX" file into a CP/M-executable "COM" file.

3 LOAD REQUIREMENTS FOR HEX FILES

To be loaded, a file must contain valid Intel hexadecimal format records. The ASM utility can be used to produce such a file from a file with the "ASM" extension. The hex file must begin at address 100H, which is the memory location at which the Transient Program Area (TPA) begins. In addition, the addresses in the hex records must be in ascending order. Gaps in memory regions are filled with zeroes by the LOAD command, as the hex records are read. Thus, LOAD must be used only for creating CP/M command files, which operate in the TPA. Program files that occupy memory locations other than the beginning of the TPA (address 100H) can be loaded using the DDT utility. After a program is loaded, LOAD displays a message in the following form:

```
FIRST ADDRESS nnnn
LAST ADDRESS nnnn
BYTES READ nnnn
RECORDS WRITTEN nn
```

4 LOAD ERROR MESSAGES

```
INVALID HEX DIGIT
```

EXPLANATION: The hexadecimal file you tried to LOAD contains upper bits set in ASCII words.

```
ERROR ADDRESS nnnn
```

EXPLANATION: An error occurred at address "nnnn".

```
CHECK SUM ERROR
```

EXPLANATION: The hexadecimal file did not produce the correct check sum during LOAD execution.

```
ERROR: CANNOT OPEN SOURCE, LOAD ADDRESS nnnn
```

EXPLANATION: The hexadecimal source file specified in the LOAD command line is not present on the specified disk.

```
NO MORE DIRECTORY SPACE
```

EXPLANATION: The directory on the disk is full.

```
CANNOT CLOSE FILES
```

EXPLANATION: The loaded file is not present when LOAD tries to close the "COM" file.

```
INVERTED LOAD ADDRESS, LOAD ADDRESS nnnn
```

EXPLANATION: The loaded hexadecimal file did not start at beginning of the TPA (0100H) as it should have.

MAKEBIOS

The Utility that Customizes the Operating System for Disk Drives

The MAKEBIOS utility helps you to create the part of CP/M known as the Basic Input/Output System (BIOS). The kind of BIOS that MAKEBIOS creates is selected by you to accommodate any combination of disk drives offered by Heath/Zenith. You can implement this utility by running MAKEBIOS (1) and selecting a drive combination (2). MAKEBIOS execution is automatic (3).

NOTE: The BIOS is a component of the CP/M Operating System that makes the system work on Heath/Zenith hardware. The file "BIOS.SYS" (which is included with your CP/M software) contains the BIOS.

1 RUNNING MAKEBIOS

After making a backup copy of the CP/M Distribution Disk, you can run MAKEBIOS by entering a command line in the form:

A>**SUBMIT MAKEBIOS {destination drive}:{file name} {source drive}** ***RETURN***

Where {**destination drive**} is the name of the drive/partition device to which you wish to send the customized BIOS;

where {**file name**} is the optional name of the customized BIOS. If no name is entered, the file containing the tailored BIOS will be called "BIOS.SYS". Any file on the destination disk previously named "BIOS.SYS" will be overwritten.;

where {**source drive**} is the optional name of the drive/partition upon which the following files reside: MAKEBIOS.COM, MAKEBIOS.SUB, PREL.COM, BIOS.ASM; and

where the following files must reside on the default drive: ASM.COM and SUBMIT.COM.

An example of such a command line would be as follows:

A>**SUBMIT B:MAKEBIOS C: B:** ***RETURN***

NOTE: Neither the default disk nor the disk containing the MAKEBIOS files can be write protected during a MAKEBIOS run.

2 SELECTING THE APPROPRIATE DRIVE COMBINATION

The entry of the preceding example command line will first produce the display of a menu listing assortments of Heath/Zenith drive controllers:

```
A>MAKEBIOS A:1  C:

BIOS SELECTION MENU

A -- H17 ONLY
B -- H37 ONLY
C -- H47 ONLY
D -- H67 ONLY
E -- H17 AND H37
F -- H17 AND H47
G -- H17 AND H67
H -- H37 AND H47
I -- H37 AND H67
J -- H47 AND H67

ENTER SELECTION:
```

You should enter the selection letter corresponding to the disk drive combination being used.

NOTE: The above selections refer to disk drive controllers. Most disk drive models bear the same number as that of the controller that controls them, except that the "H-17" controller can control the H-77 drive unit, the H/Z-87 drive unit, or the H/Z-17 drive within the H/Z-89 microcomputer model.

3 MAKEBIOS EXECUTION

From this point on, MAKEBIOS execution is automatic. No further user participation is required.

After you enter a letter at the "ENTER SELECTION:" prompt, a series of command lines and execution messages will appear on the screen. The following example shows the form of this display. (Numeric values will vary.)

```
A>ASM BIOS.AAZ
CP/M ASSEMBLER - VER 2.0
1537
02DH USE FACTOR
END OF ASSEMBLY

A>REN A:BIOS.HX0=BIOS.HEX
A>MAKEBIOS A:2

A>ASM BIOS.AAZ
CP/M ASSEMBLER - VER 2.0
1637
02DH USE FACTOR
END OF ASSEMBLY

A>REN A:BIOS.HX1=BIOS.HEX
A>PREL A:BIOS C:BIOS

A>REN C:BIOS.SYS=BIOS.HX1
A>MAKEBIOS C:3

MAKEBIOS FUNCTION COMPLETE
```

When the display is complete, the destination drive will contain a BIOS suitable for your disk drive combination. This BIOS will bear the file name "BIOS.SYS", or whatever name you might have specified.

This new file will now help control input/output operations for your hardware. It may also be transferred to other disks or partitions, using the PIP utility with the "[R]" parameter.

4 MAKEBIOS FUNCTION

The CP/M Operating System requires a Basic Input/Output System (BIOS) to control input/output operations for any peripheral devices that are connected to the Heath/Zenith computer. Among these devices may be the disk drives, a keyboard, a video monitor and a printer.

Your CP/M Distribution Disk (or disk set) comes with a BIOS that will adequately control one of your disk drive types. However, Heath/Zenith offers four different kinds of disk drive, and each hardware environment can include any combination of one or two different drive types. Because a BIOS capable of controlling all combinations would take up too much memory space, the MAKEBIOS files are provided to enable to create a BIOS that will control your own disk drive combination.

Therefore, users with two different kinds of disk drive should run MAKEBIOS.

5 A PERSPECTIVE ON MAKEBIOS INVOCATION

You do not invoke the MAKEBIOS.COM utility directly. You enter a SUBMIT command, which triggers the automatic entry of several command lines. The command lines that invoke MAKEBIOS.COM and other CP/M commands are stored in the file named "MAKEBIOS.SUB".

The utilities and resident commands specified within the MAKEBIOS.SUB file automatically modify, assemble, and relocate the source file BIOS.ASM. After these activities are performed, the assembled BIOS file is written to the specified file name (or default name BIOS.SYS) on the specified drive.

6 FURTHER BIOS MODIFICATIONS

You can perform further modifications to the BIOS before running MAKEBIOS by carefully altering the source file BIOS.ASM, which is included with your CP/M distribution media.

However, BIOS.ASM should not be altered until after the the distribution disk has been copied, and the MAKEBIOS.SUB file has been SUBMITted.

As noted within the BIOS.ASM program, no alterations should be made to the first five lines of the program under any circumstances. The first five lines of the program affect subsequent locations in the program, and user alterations to them could easily render the entire program useless.

7 MAKEBIOS ERROR MESSAGE

```
ERROR IN EXECUTION OF MAKEBIOS
```

EXPLANATION: Check the parameters in the SUBMIT MAKEBIOS command line, check the arrangement of the necessary disk files, and run MAKEBIOS again.

MOVCPMxx (MOVCPM17, MOVCPM37, MOVCPM47, and MOVCPM67)

The Utilities that Customize a CP/M Operating System Kernel to Fit Memory Size and Disk Type

After selecting the proper MOVCPM utility to use (1), you can implement it to adjust the system kernel so that it has the proper memory limit and will work on disks of a particular type (3). This utility should be followed immediately with another utility or command (5).

1 SELECTING THE PROPER MOVCPM UTILITY

MOVCPM17 is run when you wish to move a system kernel to a 5.25-inch, hard-sectored disk.

MOVCPM37 is run when you wish to move a system kernel to a 5.25-inch, soft-sectored disk.

MOVCPM47 is run when you wish to move a system kernel to an 8-inch disk that will be used to perform bootstrap in an H/Z47 drive model.

MOVCPM67 is run when the you wish to move a system kernel to a Winchester Disk partition, or to an 8-inch disk that will be used to perform bootstrap in an H/Z67 drive model.

In this text section, MOVCPMxx will be used to indicate any of these four utilities designed for use with CP/M Version 2.2.04., where "xx" is the model number of the drive unit receiving the system kernel.

2 FUNCTION OF MOVCPMxx

MOVCPMxx loads the kernel of a CP/M Operating System (the part exclusive of the file BIOS.SYS) into a special location in computer memory. At this location, it adjusts the system kernel to either a specified memory size or the total available memory size of the computer. MOVCPMxx must also observe and measure the BIOS.SYS file that will eventually be used with the system kernel, in order to allow sufficient space for the BIOS.SYS file. In addition, MOVCPMxx must make adjustments for the type of disk to which the system kernel will eventually be transferred. MOVCPMxx, however, will rely upon the SYSGEN utility to copy the system kernel that MOVCPMxx loaded into memory.

3 MOVCPMxx COMMAND LINE ENTRY

The MOVCPMxx command line is entered in the following form, with one mandatory specification, two optional specifications and a space separating each, as shown:

MOVCPMxx nn z ***RETURN***

Where **xx** is the number that identifies a particular MOVCPM utility being used. It also stands for the type of disk drive that will be the destination of the system kernel (17, 37, 47, or 67). One of these numbers is a mandatory component of any MOVCPM command;

where the **nn** variable represents the memory size the transferred system kernel will occupy, in multiples of 1024 bytes (kilobytes). This is an optional value. If the "*" character or no value is entered, the system kernel will be set to occupy the entire memory capacity of the computer being used, by default. (H/Z89 computers have RAM capacities of 32K, 48K, or 64K.) This value can be less than or equal to the actual memory capacity of the computer. It should not be greater than the memory capacity of the computer.

where the **z** variable represents the location of the BIOS that is to be matched up with the system kernel being moved. You should enter the name of the drive that contains the appropriate BIOS. If the you has renamed the BIOS to be used (by using the MAKEBIOS or REN commands), then this variable must include the name of the drive containing this BIOS and the name of the file containing the BIOS. This variable is optional. If omitted, the MOVCPM utility will assume that the created system kernel will be matched with a copy of the BIOS that was loaded into memory during bootstrap.

The "*" character must be entered when you specify no value for the memory ("nn") variable, and **do** specify a value for the BIOS source ("z"). In this sort of command line, the "*" character acts as a space filler so that the computer does not interpret the source value entered as a memory value because it was entered in the memory value space.

During execution, the MOVCPMxx utility will respond with a message in the following form:

```
MOVCPMxx VERSION 2.v

CONSTRUCTING nnk CP/M vers 2.v
READY FOR "SYSGEN" OR
"SAVE 38 CPMnn.COM"
```

4 MOVCPMxx EXAMPLES

The following command lines and accompanying explanations are specific examples:

A>**MOVCPM67 64** ***RETURN***

The system kernel created by this command will be bootable from an H/Z67 drive. It will operate with 64K of RAM. The kernel will be adjusted using the BIOS in computer memory for reference.

A>**MOVCPM37* C:** ***RETURN***

The system kernel created by this command will be bootable from an H/Z37 drive. It will probe computer memory and operate at computer's memory capacity. This kernel will be adjusted using the BIOS.SYS file stored in drive C: for reference.

A>**MOVCPM1732 B:MYBIOS.SYS** ***RETURN***

This command will adjust a system kernel that will eventually help to make a 5.25-inch hard-sectored disk bootable. It will operate with 32K of RAM. This kernel will be adjusted by referring to the BIOS file "MYBIOS.SYS", which resides in drive B:.

5 AFTER RUNNING MOVCPMxx...

MOVCPMxx should be immediately followed by the SYSGEN utility, which will transfer the adjusted CP/M system kernel from the TPA to the system tracks of a specified blank disk; or by the SAVE utility, which will transfer the adjusted CP/M system kernel from the TPA to a disk file.

If you perform any other activity immediately after running MOVCPMxx, the work of MOVCPMxx will probably be destroyed.

NOTE: The BSYSGEN utility, although similar to the SYSGEN utility, cannot be used to transfer a system kernel moved by a MOVCPMxx command.

6 MOVCPMxx ERROR MESSAGES

```
INVALID MEMORY SIZE
```

EXPLANATION: Valid memory sizes are between 32K and 64K.

```
SYNCHRONIZATION ERROR
```

EXPLANATION: The serial number of the MOVCPMxx utility used must match that of the operating system used.

```
READ ERROR
```

EXPLANATION: MOVCPMxx cannot read data from a file you specified because the file is flawed.

```
NO FILE
```

EXPLANATION: MOVCPMxx cannot read data from a file you specified because it cannot find the file on the specified drive.

```
NO SPACE
```

EXPLANATION: The BIOS file you specified will not fit in the Transient Program Area.

```
BAD LOAD
```

EXPLANATION: A file specified by you did not load properly. You should try the MOVCPMxx command again or specify a different file.

```
CAN'T OPEN BIOS.SYS
```

EXPLANATION: MOVCPMxx is unable to use the specified BIOS because it is not stored in a file named "BIOS.SYS", or because it is a defective file.

```
FATAL ERR F25: NOT ENUF MEMORY
```

EXPLANATION: MOVCPMxx execution was aborted; you should perform bootstrap and try again.

```
File not found.
```

EXPLANATION: MOVCPMxx could not find the file you specified.

PIP

The Utility that Copies Data between Files, Disks, and/or Devices

PIP stands for Peripheral Interchange Program, the CP/M utility that can be invoked (1, 2, 3) to copy (4) and link (5) files or parts of files (6), and to direct intput and output between logical devices (8). PIP also allows files to be transferred between different disks when using a one-drive hardware environment (9). PIP operations can be regulated by parameters (11).

1 METHODS OF ENTERING PIP COMMANDS

Two different methods can be deployed to enter PIP commands: the PIP prompt method and the system prompt method.

1.1 PIP Prompt Method

With the PIP prompt mode, you enter "PIP" in response to the system prompt. This entry is sufficient to invoke PIP, which displays its own prompt—the asterisk (*). You can now enter the argument half of the command line in response to the "*" prompt supplied by PIP, as shown in the following example:

```
A>PIP RETURN
*{argument} RETURN
*
```

After this PIP operation is finished, PIP again displays the "*" prompt. In reply, you can enter another argument to command PIP to perform another operation, or enter a carriage return only to return to the operating system and obtain a system prompt.

1.2 System Prompt Method

To induce a similar operation without splitting up the command line, the system prompt mode of command entry can be used. With this mode, you must include the argument in the response to the system prompt, as shown in the following example:

```
A>PIP {argument} RETURN
A>
```

After PIP finishes this operation, it automatically returns you to the operating system, and the system prompt is displayed.

All characters specified in such a command line will be automatically translated into UPPER CASE.

NOTE: PIP sends data to a destination from a source, but it does not remove the data from the source. It merely copies an image of the data, and then sends it. Therefore, any data that is copied will remain intact at the source after the PIP operation.

2 THE ARGUMENT IN A PIP COMMAND LINE

PIP commands require an argument regardless of which invocation method is used. PIP arguments must specify a "data source" (which can be either a file or a logical input device), and a "data destination" (which can be either a file, a disk, or a logical output device). The "data destination" is entered first and followed by an "=" sign, which is followed by the "data source", as shown in these examples:

```
A>PIP RETURN
*{data destination}={data source} RETURN
```

or

```
A>PIP {data destination}={data source} RETURN
```

In all cases, copied data is transferred in a right-to-left direction, with respect to the command line components. Hence, the file, disk, or logical output device to receive the data is always specified on the left hand side of the "=" sign; and the file or logical input device to submit the data is always specified on the right hand side of the "=" sign.

If entered in the system prompt mode, characteristics entered in a PIP command line will be in upper case in any future displays. Characters entered in a PIP prompt mode command line will reappear in the case in which they were entered.

3 CHARACTERISTICS OF DATA DESTINATION AND SOURCE SPECIFICATIONS

A data destination can be a file, a disk, or a logical device. A data source can be a file or a logical device. Each must be specified in the command line as shown below:

DATA SPECIFICATION CHARACTERISTICS	EXAMPLES

Data can be transferred to a disk by specifying the letter of the drive in which the disk resides, and a colon.	B:=SENDOVER.DOC
	B:=CON:

Files on non-default disks are identified by specifying the drive in which they reside immediately before the file name.	B:NEW.DOC=C:SEND.DOC B:CREATED.DOC=CON: CON:=B:SENDOVER.DOC

Logical devices are specified by entering the three-letter code for that particular device and a colon.	CON:=LST:
	LST:=SENDOVER.DOC
	CREATED.DOC=CON:

4 FILE COPYING EXAMPLES

The following argument will cause PIP to transfer a copy of the file SENDOVER.DOC from drive B to the same disk, and give it the file name CREATED.DOC. In effect, this operation creates a file backup with a different name on the same disk.

B:CREATED.DOC = B:SENDOVER.DOC *RETURN*

The following argument will cause PIP to transfer a copy of the file SENDOVER.DOC from drive C to the disk in drive B, and give it the file name CREATED.DOC.

B:CREATED.DOC = C:SENDOVER.DOC *RETURN*

The following argument will cause PIP to transfer a copy of the file SENDOVER.DOC from the currently logged drive to the disk in drive B, and give it the same file name.

B: = SENDOVER.DOC *RETURN*

The following argument will cause PIP to transfer a copy of the file SENDOVER.DOC from drive C to the disk in drive B, and give it the same file name.

B: = C:SENDOVER.DOC *RETURN*

Each of the following arguments will cause PIP to transfer a copy of the file SENDOVER.DOC from the disk in drive B to the disk in the currently logged drive, and give it the file name SENDOVER.DOC. It should be noted that the data source in this argument is **not** the disk in drive B, but the file SENDOVER.DOC which resides on the disk in drive B. Because the source file is not specified in the typical data source location, PIP assumes that the source file is the same as the destination file, which is specified in its typical location.

SENDOVER.DOC = B: *RETURN*

or

SENDOVER.DOC = B:SENDOVER.DOC *RETURN*

5 DATA SOURCE CONCATENATION

While only one data destination can be specified in a single PIP command line, several data sources can be specified. Thus you can merge data from several locations to one location.

When more than one data source is specified, each data source specification is separated by a comma.

The length of a PIP command line cannot exceed 255 characters. (If the user tries to enter a 256th character into the command line, PIP will begin execution based on the first 255 characters entered.)

PIP will concatenate data sources in the order in which their specifications are entered in the command line.

The following PIP argument will cause PIP to get a copy of the files THISFILE.DOC and THATFILE.DOC from the disk in drive C and the file YOURFILE.DOC from the disk in the default drive. PIP will then combine all three files into one file (in the order they are specified), place this file on the disk in drive B, and assign the name COMBINED.DOC to the transferred combination of the three source files.

```
*B:COMBINED.DOC=C:THISFILE.DOC,C:THATFILE.DOC,YOURFILE.DOC RETURN
```

Both files and devices can be specified as data sources in the same command line.

NOTE: If you wish to concatenate object files with a PIP operation, you must specify the O parameter after the file name of each object file that you wish to concatenate.

6 COPYING A BLOCK OF DATA

Blocks of files, as well as whole files, can be copied to another file. To do this, you enter a PIP command that specifies the source file name and the beginning and end of the block being transferred. Such an entry requires the use of PIP parameters as in the following example:

BLOCKFIL.TXT = WHOLEFIL.TXT[Sbeginning ^ ZQend ^ Z]** ***RETURN

In this example, "BLOCKFIL.TXT" is the name of the destination file to receive a block of text from the source file named "WHOLEFIL.TXT". The block is specified by entering a bracket, the parameter "S", a unique string of text from the "beginning" of the block, a CTRL-Z character, the parameter "Q", a unique string of text from the "end" of the block, a CTRL-Z character, and another bracket.

The following example demonstrates how the user can concatenate blocks from several files, by using the "S" and "Q" parameters to specify the beginning and end of each block:

USBLOCK.TXT = GETADD.TXT[SFour ^ ZQearth ^ Z],CONSTIT.TXT[SWhen ^ ZQAmerica ^ Z]** ***RETURN

With the entry of this command line, PIP would combine the specified block from the file "GETADD.TXT" with the specified block from "CONSTIT.TXT". With this data, PIP would create the file "USBLOCK.TXT".

NOTE: The strings following the S and Q parameters will be translated to UPPERCASE if you are using PIP by the System Prompt Method. They will **not** be translated to upper case if you are using the Utility Prompt Method.

7 COPYING DATA TO AN INTEL HEXADECIMAL FILE

PIP performs a special function if the data destination is a file with the "HEX" extension (an Intel hex formatted machine code file) and the data source (for one of several data sources) is an input/output device, such as a paper tape reader.

In such a case, PIP checks to ensure that the source is a properly formed Intel hexadecimal file, with legal hexadecimal values and checksum records. If an invalid input record is found, PIP reports an error message at the console and waits for corrective action.

It is usually sufficient to open the reader and rerun a section of the tape (by pulling the tape back about 20 inches). When the tape is ready for the second reading, you should enter a carriage return at the console, and PIP will attempt another read. If the tape position cannot be properly read, then you should simply continue the read (by entering a carriage return after the error message), and enter the record manually, using the ED utility after the disk file is constructed from the other data source components.

For convenience, PIP allows a CTRL-Z ("end-of-file" character) to be entered from the console if the source file is a RDR: device. In such a case, PIP reads the specified device and monitors the keyboard. If a CTRL-Z is entered at the keyboard, then the read operation is terminated normally.

For instance, the following PIP argument could be entered to create a "HEX" file:

CREATED.HEX = CON:,B:SENDOVER.HEX,PTR:** ***RETURN

The preceding argument will cause PIP to create the data destination file CREATED.HEX by reading the data from source device CON: (as you enter hexadecimal values at the keyboard and eventually enter a CTRL-Z "end-of-file" character), the data from source file SENDOVER.HEX (which includes a CTRL-Z), and finally the data from source device PTR: (until a CTRL-Z is read from the paper tape).

NOTE: Hexadecimal data can be checked for valid format by specifying the "H" or "I" format in the PIP argument.

8 COPYING DATA TO OR FROM LOGICAL DEVICES

The following examples, and their ensuing explanations, demonstrate PIP commands that transfer data to and/or from computer-recognized logical devices:

LST:=SENDOVER.PRN** ***RETURN

The preceding argument will cause PIP to copy the source file SENDOVER.PRN to the LST: device.

CON:=B:THISFILE.ASM,C:THATFILE.ASM,YOURFILE.ASM** ***RETURN

The preceding argument will cause PIP to concatenate three ASM source files from disk B, disk C, and the logged disk, respectively; and to copy the combination to the CON: destination device.

PUN:=NUL:,SENDOVER.ASMmr,EOF:,NUL:** ***RETURN

The preceding argument will cause PIP to send 40 nulls from the NUL: source device to the PUN: destination device, and then to copy the SENDOVER.ASM source file to the PUN:, then to send a CTRL-Z "end-of-file" character from the EOF: source device to the PUN:, and finally to send 40 more nulls from the NUL: source device to the PUN: data destination device.

8.1 Input/Output Devices Accessible through PIP

The PIP utility enables you to transfer data directly to or from the logical and physical devices being used. PIP supports data transfer with respect to the devices indicated in the following table:

LOGICAL DEVICE NAME	PHYSICAL DEVICE NAME	DEVICE DEVICE DESCRIPTION AND/ OR CATALOG NAME OF RECOMMENDED INPUT/OUTPUT MACHINE
CON:	TTY:	Any non-handshaking RS-232 ASCII terminal at port 0D0H
	CRT:	Any non-handshaking RS-232 video terminal at port 0E8H
	BAT:	A batch pseudo device using RDR: for input and LST: for output
	UC1:	Any handshaking RS-232 terminal with ETX/ACK protocol, eg. Diablo KSR 1640 printing terminal
RDR:	TTY:	Any non-handshaking RS-232 ASCII terminal at port 0D0H
	PTR:	Null source, not implemented, returns an "end-of-file" character when accessed
	UR1:	An input modem at port 0D8H
	UR2:	System terminal

PUN:	TTY:	Any non-handshaking RS-232 ASCII terminal at port 0D0H
	PTP:	Null sink, not implemented
	UP1:	An output modem at port 0D8H
	UP2:	System terminal
LST:	TTY:	Any non-handshaking RS-232 ASCII terminal at port 0D0H (eg. WH-34)
	CRT:	Any non-handshaking RS-232 video terminal at port 0E8H
	LPT:	line printer (eg. H-14, WH-14, WH-24, H/Z25, WH-36, Epson MX80)
	UL1:	Any Diablo printer

Table 2-3:
Logical/Physical Devices

You must be certain that the destination device that is specified is capable of receiving data, and that the specified source device is capable of sending data.

ADDITIONAL LOGICAL DEVICES USED WITH PIP

All of the devices in the preceding table can also be referenced using the STAT utility, which assigns physical devices to logical devices on a temporary basis.

The PIP utility can also gain access to five additional devices, which are defined below:

NUL: Sends 40 "nulls" (ASCII zeroes) to the device (This device is usually accessed at the beginning and/or end of the output when a paper tape punch device is used.)

EOF: Sends an "end-of-file" character (ASCII CTRL-Z) to the destination device. (Such a transfer is performed automatically after PIP operation involving a file composed with ASCII characters.)

INP: Special input source which can be "patched" into the PIP program itself. Through this source, PIP accepts data input character by character using a system call to memory location 103H. The data returns from location 109H. The parity bit must be preset at zero.

OUT: Special output destination which can be "patched" into the PIP program. PIP transmits data from register C to this destination using a system call to memory location 106H. It should be noted that locations 109H through 1FFH of the PIP memory image are not used, and can be replaced by special purpose drivers using DDT.

PRN: This device is accessed for the same purposes as the LST: device. In its operation, however, tabs are expanded at every eighth character position, lines are numbered, and page breaks occur every 60 lines. Output directed to the LST: device will be treated identically if the [T8NP] or [T8NP60] parameter is entered with the LST: argument.

8.2 Suspending PIP Operations

When sending material with PIP to the CON: logical device, the copy operation can be suspended by entering the ***CTRL-S*** character at the keyboard, and resumed by entering any character other than CTRL-C. The operation can be aborted by entering any keyboard character other than CTRL-S while data is being transferred. PIP will respond to such an entry with the message:

```
ABORTED
```

9 USING PIP WITH A ONE-DRIVE HARDWARE ENVIRONMENT

When performing a PIP operation in a one-drive hardware environment, the "PIP" command should be entered alone at the system prompt (using invocation method 1.1) to produce the "*" "*" prompt.

When using the PIP utility to copy a file to a file on another disk, the disks involved must be inserted in the drive alternately to allow PIP to read from one file and then write to another. The command line argument should be entered as follows:

B:{destination file name} = A:{source file name}** ***RETURN

Where "**B:**" specifies logical drive device B;

where "**A:**" specifies logical drive device A; and

where both A and B are referred to as physical drive device 0 (the only physical drive device in the hardware environment).

This entry causes PIP to display the message:

```
Put disk B in drive A: and press RETURN
```

The disk desired to receive the file copy should be placed in the drive, and a carriage return entered. PIP will then display the following message:

```
Put disk A in drive A: and press RETURN
```

The disk containing the data source file should be placed in the drive, and a carriage return entered. You should repeat these two steps alternately, as the PIP prompts indicate, until the entire source file has been copied to the destination file. This process will vary in length, depending on the size of the file being copied. You should be careful to keep track of which disk is A and which is B.

10 PIP's METHOD OF OPERATION

File names and device names can both be used in a single data source argument, with PIP reading each name entered (starting with the name entered on the left-hand side of the source argument) until it reaches "end-of-file". "End-of-file" is indicated by a CTRL-Z character in ASCII files, and by the actual end of the file in non-ASCII files. Each of the data source names entered is read and then concatenated to the preceding name (to its left) until the last name has been read.

The data destination device or file receives a copy of the data from the source files and/or devices. When data from an ASCII file is written, the CTRL-Z "end-of-file" character is appended to the result.

As the PIP operation begins, a temporary file is established in the directory of the destination disk. This file has a name that consists of the primary name specified in the data destination, and a "$$$" extension. This temporary file is not changed to the actual file (with the extension specified in the command line argument) until successful completion of the PIP operation.

Files with a "COM" extension are always assumed to be non-ASCII files.

11 PIP PARAMETERS

The PIP utility performs enhanced data transfer operations when command line parameters are used. Most PIP parameters should be entered after the data source and enclosed between a single set of square brackets. A carriage return must follow the closing bracket. Most parameters are entered in the following form:

{data destination} = {data source}{[parameters]}** ***RETURN

Where {**data destination**} is a specification of the file, disk, or device to receive the transferred data;

where {**data source**} is one or more specifications of the file(s) or device(s) from which the transferred data will be copied; and

where {**[parameters]**} is the specification of PIP parameters used to regulate the transfer operation.

Several parameters can be enclosed between brackets in a single PIP command line.

NOTE: If you wish to concatenate object files with a PIP operation, you must specify the O parameter after **each** file name that is specified as part of the data source.

The parameters used to regulate transfer operations are as follows:

B Block mode transfer: PIP transfers data from a source device to a buffer (data reservoir in computer memory), which retains bits of a continuous flow of data before writing it on a disk. The buffer is emptied to a disk when the CTRL-S character is transmitted to the buffer from the source device. PIP then resumes transmitting the continuous flow of data from the source device. If the buffer overflows with data, PIP will display an error message.

Dn Delete characters: Causes PIP to transfer all of the characters within the source file or source device—except for the characters to the right of the nth column, which are truncated so that the file can be sent to a narrow printer or console.

E Echo transfers: Causes display indicating what units of data are being transferred, as they are being transferred.

F Filter Form Feeds From File: Removes form feeds that are embedded in a file. The “P” parameter can be entered in the same command line to insert new form feeds.

Gn Get file from user area “n”: Enables user to access files through a different user area number “n” (in the range 0-15). Can only be implemented if the file “PIP.COM” is in the current user number.

H Hex data transfer: Transferred data is checked for proper Intel hexadecimal file format, and unnecessary characters between hex records are removed during the transfer operation. If errors occur, PIP will display prompts for corrective action.

I Ignore records: Transferred Intel hex files will not include “:00” records. The I parameter automatically sets the H parameter.

L Lower case letters: Translates all copied alphabetic characters to lower case.

N Number lines: Applies line numbers to the beginning of each transferred line of data. The numbers begin at one and increase sequentially by ones. The numbers are separated from data lines by a colon. If “N2” is specified, then each number is preceded by a zero and followed by a tab space. This tab space is expanded if the “T” parameter is also set.

O

Object file transfer: The normal CP/M end of file will be ignored in any data transferred with this parameter. This parameter is useful in transferring files with non-ASCII characters.

NOTE: If you wish to concatenate object files with a PIP operation, you must specify the O parameter after the file name of each object file that you wish to concatenate. Such an operation can be caused by a command line in the following form:

```
A>PIP B:ALL.OBJ=FILE0.OBJ[O],FILE1.OBJ[O],FILE2.OBJ[O] RETURN
```

Pn

Page ejects: Transferred data is interrupted at the beginning of the file and every “n” lines thereafter. If “n” is omitted from the parameter, the value “1” is assumed. If the value “1” is entered or assumed with the parameter, page ejects will occur every 60 lines. If the “F” parameter is entered in the same command line, then form feed suppression takes place before the new page ejects are inserted.

Qstring ^ Z

Quit copying block: The text “string” between the “Q” and the “ ^ Z” in this parameter marks the end of a block of text to be transferred. (Block begins with text “string” in the “Sstring ^ Z” parameter.

R

Read system files: Enables transferral of files with the “SYS” status

Sstring ^ Z

Start copying block: The text “string” between the “S” and the “ ^ Z” in this parameter marks the start of a block of text to be transferred. (Block ends with text “string” in the “Qstring ^ Z” parameter.)

NOTE: the strings in the S and Q parameters are translated to upper case by the Console Command Processor if PIP is invoked using the system prompt method (entire command on one line):

```
A>PIP B:THIS.DOC = C:THAT.DOC[STHE BEGINNING ^ ZQTHE END. ^ Z]
```

Therefore, the file's actual text must be in upper case to match the string. However, if PIP is invoked using the PIP prompt method (command line argument entered in response to PIP "*" prompt):

```
A>PIP
*B:THIS.DOC = C:THAT.DOC[SThe beginning ^ ZQthe end. ^ Z]
```

then the string text will not be automatically translated into upper case.

Tn Tab expansion: Expands each occurrence of tab character to every "n"th column in transferred text.

U Upper case: Translates lower case alphabetic characters to upper case during text transferral.

V Verify operation: Compares copy of transferred data with original to ensure exact correspondence. Might increase command execution time by as much as two times. This parameter only works if the data destination is a disk file.

W Write over files: Writes newly transferred data file over old data file, even if old file has been given R/O (read only) status. PIP will not prompt you before deleting the old file.

Z Zero parity bit: Sets parity bit on input to 0 for each ASCII character in transferred data.

12 PIP ERROR MESSAGES

```
DISK READ ERROR
```

EXPLANATION: Source of data is flawed. You should repeat PIP command or replace data source.

```
DISK WRITE ERROR
```

EXPLANATION: Destination disk did not have enough space for the copied data. You should clear out some space on the destination and repeat PIP command.

```
VERIFY ERROR
```

EXPLANATION: Destination disk has flawed media. You should replace destination disk and repeat PIP command.

```
ABORTED
```

EXPLANATION: A keyboard entry during a PIP data transfer caused this message and an end to PIP execution.

```
BAD PARAMETER
```

EXPLANATION: You entered an invalid parameter or an invalid bracket in a PIP command line.

```
INVALID USER NUMBER
```

EXPLANATION: You specified a user number out of the 0-15 range.

```
INVALID DIGIT
```

EXPLANATION: You entered an invalid digit for a PIP parameter, and should re-enter the entire command line.

```
CHECKSUM ERROR
```

EXPLANATION: PIP verification found a discrepancy between a data source and data destination file. The PIP operation should be repeated.

```
INVALID FORMAT
```

EXPLANATION: You entered PIP command line in improper form, and should repeat entry of command.

```
NO DIRECTORY SPACE
```

EXPLANATION: The directory of the destination disk is full.

```
NO FILE: =x:filename.ext
```

EXPLANATION: PIP could find no source file by the name "filename.ext" on the disk in drive "x".

```
START NOT FOUND
```

EXPLANATION: PIP could not find the beginning of a string that was specified with the "S" and "Q" parameters. You should re-enter command specifying a string beginning that exists in the data source.

```
QUIT NOT FOUND
```

EXPLANATION: PIP could not find the end of a string that was specified with the "S" and "Q" parameters. You should re-enter command specifying a string end that exists in the data source.

```
DESTINATION IS R/O, DELETE (Y/N)?
```

EXPLANATION: This message prompts you that the destination already has a file by that name. Use can have the old file destroyed by answering Y to this prompt.

```
NOT FOUND
```

EXPLANATION: You should re-enter the PIP command with the name of an accessible data source.

```
REQUIRES CP/M 2.0.0 OR NEWER FOR OPERATION
```

EXPLANATION: PIP must be used with CP/M version 2.2.02 or 2.2.03 or 2.2.04.

```
UNRECOGNIZED DESTINATION
CANNOT WRITE
```

EXPLANATION: You should re-enter a PIP command with a valid data destination.

```
INVALID PIP FORMAT
CANNOT READ
```

EXPLANATION: You made syntax error in command line, and PIP could not write to the data destination.

```
INVALID SEPARATORS
```

EXPLANATION: You should re-enter PIP command with the proper square brackets around a parameter.

PREL

The Utility that Creates a Relocatable File from Two Hexadecimal Output Files

The PREL utility is invoked (1) to take the hex output files of two assemblies and generates a relocatable file from them (2). The first of the assemblies should have as an initial origin of 000H and the second assembly should be one page higher, or 0100H. The hex output of the first assembly should be given a file name extension of "HX0", and the hex output of the second should have "HX1" for its extension. PREL can be effectively implemented as one of the batched commands in a SUB file (3).

1 PREL INVOCATION

A PREL command line is entered in the following form:

A>**PREL {hex file name} {relocatable file name}** ***RETURN***

Where {**hex file name**} is the primary name of the two assembled hex files (These files are given the extensions HX0 and HX1); and

where {**relocatable file name**} is the primary name of the output page relocatable file. (The output file will have the extension PRE.)

For example, the command line:

A>**PREL MYPROG TESTBIOS** ***RETURN***

would create the file TESTBIOS.PRE from the two hex files MYPROG.HX0 and MYPROG.HX1.

2 PREL FUNCTION

PREL will compare the two hex files and perform two separate functions:

- PREL converts the hex file to a binary file (similar to the function of the LOAD utility.)

- PREL appends a relocation table to the end of the file which includes a bit for each byte in the module. If a particular bit has a value of one, then the associated byte must be offset when the program is loaded into memory. An error message will result if the hex values for a given byte differ by more than 0 or 1. An error message will also result if the addresses in the hex files do not increase sequentially.

3 PREL OPERATION

All addresses that need to be relocated should be expressed relative to the base of the module, and not defined absolutely. A convenient way to do this is to omit "ORG" statements from the program and to use the SUBMIT utility to enter batched commands from a SUB file. The commands in the SUB file should perform the required assemblies and the conversion to a relocatable format. An example of this type of batched command might be:

```
PIP TEMP0.ASM=ORG0.ASM,$1.ASM
ASM TEMP0.AAZ
REN $1.HX0=TEMP0.HEX
ERA TEMP0.ASM
PIP TEMP1.ASM=ORG1.ASM,$1.ASM
ASM TEMP1.AAZ
ERA TEMP1.ASM
REN $1.HX1=TEMP1.HEX
PREL $1 $1
ERA $1.HX0
ERA $1.HX1
```

where the files ORG0.ASM and ORG1.ASM contain nothing but "org 0000H" and "org 0100H" statements respectively. This command script could be invoked with the entry:

A>**SUBMIT MAKEPRE MYPROG** ***RETURN***

Where the command sequence is contained in a file called MAKEPRE.SUB, the original assembly file is called MYPROG.ASM and the final result is a page relocatable file called MYPROG.PRE.

4 PREL ERROR MESSAGES

```
Phase error
```

EXPLANATION: Hexadecimal files do not contain correct starting addresses.

```
Cannot open input file
```

EXPLANATION: The file with the extension "HX0" or "HX1" was not found.

```
Cannot create output file
```

EXPLANATION: The directory on the destination disk is full.

```
Hex files not monotonic
```

EXPLANATION: The hexadecimal addresses are not increasing sequentially.

```
Write error
```

EXPLANATION: The destination disk is full.

REN

The Resident Command that Renames Files

The REN command allows you to assign a new name to any disk file (1). If it cannot rename a file, it also displays the reason (2).

1 RENAMING A FILE

To rename a file, you should respond to the system prompt with a command line in the following form:

```
A>REN {new name}={old name} RETURN
```

Where {**new name**} is the name you wishes to assign to the file; and

where {**old name**} is the name of the file before it is renamed.

For instance, the name of the file "BIOS.SYS" can be assigned to a file currently named "1747.SYS" by entering the following command:

```
A>REN BIOS.SYS=1747.SYS RETURN
```

If file receiving the new name does not reside on a disk in the default drive, then the REN command line must include a drive specification which precedes the new file name, as shown:

```
A>REN C:BIOS.SYS=1747.SYS RETURN
```

2 CONSOLE RESPONSE TO RENAMING COMMANDS

If a file is successfully renamed, the system prompt will appear.

```
A>
```

If the file being renamed does not exist on the default drive and no drive is specified (or if an incorrect drive is specified), then REN cannot find the file you want to rename and will display the error message:

```
NO FILE
```

If you try to assign a file a name that is actually the current name of an existing file, REN will not perform the change and displays the error message:

```
FILE EXISTS
```

If the file being renamed has Read/Only access status (as described in the text on the STAT utility), REN does not rename the file. However CP/M will display an error message in the following form:

```
Bdos Err on x: File R/O
```

SAVE

The Resident Command that Copies Data from Memory to a Disk File

The SAVE command is used to copy the hexadecimal contents of a span of memory locations to a disk file (1). The amount of data that SAVE will copy is measured in pages (2). The data that SAVE copies from the computer's memory space can come from a variety of sources (3).

1 SAVE COMMAND LINE

To SAVE a program onto a disk and create a file for it, you should respond to the system prompt with a command in the form:

A>**SAVE {pages} {file name}** ***RETURN***

Where {**pages**} is the decimal number of pages of memory contents that SAVE should copy to the disk, and

where {**file name**} is the name of the file created to store the data.

NOTE: A page of memory in CP/M is 256 (decimal) bytes, or a span of 0100H (hexadecimal) memory locations.

For example, to copy 3 pages (the memory contents between address 100H and address 3FFH) of data from main memory to the disk in non-default drive B and give this block of data the file name "PROGRAM.ASM", you would enter the following command line:

A>**SAVE 3 B:PROGRAM.ASM** ***RETURN***

2 SAVE DESCRIPTION

Data in the computer's Transient Program Area (TPA) memory space can be blocked off by specifying the first and last memory locations that it occupies. Programs loaded into the TPA always begin at memory location 0100H (hexadecimal). This is also the location from which SAVE starts copying blocks of data. Therefore, if you specified one page (100H bytes) in the SAVE command line, SAVE will start copying data from memory location 100H and continue through memory location 1FFH.

Only whole pages can be transferred; and SAVE recognizes only decimal integers for pages.

3 COMMON APPLICATIONS FOR SAVE

SAVE will copy the hexadecimal values for any data that resides in memory locations 100H through 32768H (in a 32K computer). This area of memory is known as the Transient Program Area, and it is used to accommodate data manipulated by utilities and application programs. Data often remains in these locations after being manipulated by CP/M utilities.

One useful application is to SAVE a program that has been placed in memory by the DDT utility.

Another common application is to SAVE a copy of the operating system onto a file. The CP/M operating system always occupies the area of memory between locations 0H and 100H. However, if this operating system is loaded into the Transient Program Area by a MOVCPM utility, it can be saved onto a file. To do this, you must enter a command in the following form:

```
A>SAVE 38 CPMkk.COM    RETURN
```

Where "**kk**" is the number of kilobytes of memory to which the system has been adjusted by the MOVCPM utility.

NOTE: To produce the system prompt (A>) and allow entry of the SAVE command, you must exit from the utility with a warm boot (performed by entering ***CTRL-C***).

4 COMPUTING PAGES TO BE SAVED

To SAVE a program from memory, you must determine how many pages this block occupies. This requires a conversion of the program's hexadecimal size to decimal pages.

The hexadecimal size of a program is displayed when the program is loaded into the TPA under DDT. This display indicates that the program begins at address 0100H and ends at an address identified as "NEXT". You should take the two left-hand digits from the "NEXT" value, and convert them into a decimal number. This decimal number is the number of pages to be saved.

For example, if you are debugging a program that occupies memory locations 0100H through 28FFH, the results of this debugging activity can be recorded by taking the hexadecimal digits "28", converting them into the decimal value "40", and entering the following command:

```
A>SAVE 40 PROGRAMX.HEX    RETURN
```

If the disk to which you try to SAVE a data block does not have enough space to record a file of the specified size, then the following error message will be displayed:

```
NO SPACE
```

SETLP

The Utility that Temporarily Sets Printer Characteristics

The SETLP utility enables you to change the operating system's baud rate setting for a serial printer and to determine whether the LPT device will be used to run a serial printer or a parallel printer. The SETLP utility can be run with a single command line.

Baud rate is the speed with which data is transferred between the computer and a serial printer. Baud rate is approximately equal to the number of bits transferred per second. A baud rate setting is not necessary when you are using a parallel printer.

1 REASONS TO USE SETLP

The system's baud rate setting for the LST: device (usually used for a printer) can be changed by running the CONFIGUR utility. However, SETLP was designed as a convenience so that you could change the baud rate setting without encountering all of the menus and options of the more extensive CONFIGUR program. SETLP can temporarily alter the baud rate values kept in memory, but it will not affect the baud rate values stored on the disk by CONFIGUR.

Hence a baud rate set by SETLP will remain in effect only until the computer is reset or until another SETLP command is executed to change the baud rate.

SETLP is useful if you have more that one kind of printer and use each of these printers alternately. However, if the printers used are linked to the input/output ports by different physical device names, then CONFIGUR or STAT must be used to change the operating system for these physical device names.

NOTE: SETLP does not change the baud rate of the printer itself. A printer's baud rate is usually changed by manual printer adjustments that are explained in the manual(s) supplied with your printer(s).

SETLP can also be used to quickly set the system to run either a serial printer or a parallel printer through the LPT: device.

2 SETLP COMMAND METHODS

The SETLP utility is invoked and executed by entering a command line in the H-89-3 Command Method, the H-89-11/LPT: Command Method, or the H-89-11/UL1: Command Method.

2.1 H-89-3 Command Form

This SETLP method is helpful to users with an H-89-3 interface card:

```
A>SETLP {baud} RETURN
```

Where **SETLP** is the command line function; and

where {**baud**} is the number for the baud rate to which the user wishes to temporarily adjust the operating system. The baud rate must be specified one space to the right of the SETLP function. No further command line argument is necessary.

The printer for which you are setting this baud rate must use the physical device assigned to logical device LST:. This match is made through the CONFIGUR utility or the STAT utility.

For example, the command

```
A>SETLP 1200 RETURN
```

could be entered to temporarily change the system baud rate setting to 1200.

2.2 H-89-11/UL1: Command Method

This SETLP method is helpful to users with H-89-11 interface cards and printers that are accessed through the UL1: physical device. (Diablo daisy wheel printers are accessed through this device.) It enables these users to temporarily change the baud rate that the operating system uses to send data to these printers.

A>**SETLP** {**baud**} ***RETURN***

Where **SETLP** is the command line function; and

where {**baud**} is the number for the baud rate to which the user wishes to temporarily adjust the operating system. The UL1: physical device must currently be matched with the LST: logical device. (The CONFIGUR or STAT utilities can help you to match these devices.)

For example, the command

A>**SETLP 300** ***RETURN***

could be entered to temporarily change the baud rate to 1200 of a serial printer accessed through the UL1: device.

NOTE: Using the H-89-11/UL1: Command Method, the baud rate for the UL1: physical device can be changed without affecting the baud rate for the LPT: physical device.

2.3 H-89-11/LPT: Command Method

This SETLP method is helpful to users with an H-89-11 interface card and a printer that is accessed through the LPT: physical device. (Most dot-matrix printers are accessed through this device.) It enables these users to temporarily change the system's baud rate setting for serial printers, and/or to quickly set the system to run either a serial printer or a parallel printer through the LPT: device.

A>**SETLP {parameter} {baud}** ***RETURN***

Where **SETLP** is the command line function;

where {**parameter**} is the letter that stands for a type of printer. The letter **S** is used for serial printers, and the letter **P** is used for parallel printers; and

where {**baud**} is a decimal number for the baud rate to which the user wishes to temporarily adjust the operating system. A baud rate can be specified for serial printers, but not parallel printers. The LPT: physical device must currently be matched with the LST: logical device. (The CONFIGUR or STAT utilities can help you to match these devices.)

For example, the command

A>**SETLP S 4800** ***RETURN***

could be entered to temporarily change the system baud rate to 4800, for a serial printer accessed through the LPT: device.

You can also temporarily change the system baud rate to 4800, for a serial printer accessed through the LPT: device. To make this change, enter the following command:

A>**SETLP 4800** ***RETURN***

The result of entering the preceding two example commands is identical—even though the S parameter is not used in the second command—because the system assumes that you wish to set up for a serial printer whenever you specify a baud rate in a SETLP command.

The following example command

A>**SETLP P** ***RETURN***

could be entered to temporarily change the system so that it can access a parallel printer through the LPT: device.

NOTE: The "Default I/O Configuration" menu (submenu C) in the CONFIGUR utility and the STAT utility both show whether the LST: logical device (for printers) has been matched with the LPT: or UL1: or an other physical device.

3 POSSIBLE BAUD RATES

SETLP can be used to adjust the operating system for any of the following baud rates:

75
110
134
150
300
600
1200
2400
4800
9600
19200
38400

Make certain that any baud rate entered in a SETLP command line matches the baud rate that has been set on the serial printer being used. Instructions for setting (or resetting) the baud rate of a printer can be found in your printer's manual.

4 SETLP ERROR MESSAGES

```
ERROR – Invalid argument
```

EXPLANATION: The command entered did not contain syntax for the parameter and/or baud rate; or SETLP was used with the wrong interface card. Enter a command that uses the syntax described in "2 SETLP Command Methods", and sets up the system for hardware that is connected in your hardware environment.

```
ERROR – Improper configuration
```

EXPLANATION: You tried to run SETLP with the wrong version of CP/M. Use a CP/M Operating System and a SETLP utility that has been copied from the same set of CP/M distribution media. Then enter the command again.

STAT

The Utility that Reports Disk Statistics and Assigns Types of Status

The STAT utility has two functions: to provide statistical information about disk space or file size (1), and to allow you to change the assignment of various types of status (2).

1 REPORTING STATISTICS CONCERNING DISKS AND FILES

1.1 Available Disk Space

The amount of unoccupied space on the disk in the default drive that is available for use can be determined by entering the command "STAT" and a carriage return at the default drive prompt, as in the following example:

```
A>STAT RETURN
```

If drive A is the only drive that has been logged since the system was booted, then STAT will respond with a display like the following:

```
A: R/W, Space: 3k
```

Where "A: " indicates that the disk in drive A was investigated by STAT;

where "R/W," stands for Read/Write, meaning that data and files can be written to or read from the disk. If STAT encounters a disk that has been write protected, it will report that the disk is "R/O" or read/only. Data and files cannot be recorded to or erased from a read/only disk; and

where "3k" indicates the amount of space that is unoccupied on the disk, and thus available to record files. Memory space is expressed in units of 1024 bytes, or kilobytes (k). One byte holds one character. The disk that was investigated for the preceding example has enough remaining memory space to store 3072 characters.

If you have logged other drives since the system was last booted, the preceding response to the basic "STAT" command line will also produce a listing for the disks in each of the drives that have been logged, as shown:

```
A: R/W, Space: 3k
B: R/W, Space: 18k
C: R/W, Space: 65k
```

Even if a non-default drive has not yet been logged, you can still ascertain the space remaining on the disk within it by specifying the drive in the STAT command line:

```
A>STAT B: RETURN
```

The entry of this command line might produce a response in the following form:

```
Bytes Remaining ON B: 24k
```

1.2 Disk Space Consumed by a File

To find out how much space one particular file occupies, requires specification of the file name and the drive in which the file resides. An appropriate entry for such a command would be:

```
A>STAT C:PRINTOUT.DOC RETURN
```

Such an entry might produce a display in the following form:

```
Recs  Bytes  Ext Acc
 140    18k    2 R/W C:PRINTOUT.DOC
Bytes Remaining ON C:  72k
```

Where "Recs" stands for the number of records within the file. The file PRINTOUT.DOC contains 140 data records. A record is the amount of data it takes to fill up 128 bytes (one-eighth of a kilobyte) of memory space. Thus, one record consists of 128 data characters. However, STAT cannot report fractions of records; and if a file contains anywhere from 1 to 128 characters, STAT will still report that the file contains one record;

where "Bytes" stands for the amount of memory space the file occupies, expressed in kilobytes (1024 bytes). In this display, STAT reports that the file PRINTOUT.DOC occupies 18 kilobytes of memory space. One kilobyte of space will hold 8 records. However, STAT cannot report fractions of kilobytes. If a file contains anywhere from 1 to 8 records, STAT will still report that it occupies one kilobyte of space;

where "Ext" stands for the number of extents that the file occupies. In this display, STAT reports that the file PRINTOUT.DOC occupies two extents. One extent will process 16 kilobytes of data. However, STAT cannot report fractions of extents. If a file occupies anywhere from 0 to 16 kilobytes, STAT will still report that it requires one extent to be processed; and

where "Acc" indicates the file's access status. The "R/W" access status (read/write) enables you to run commands to freely erase data from the file, or add data to the file. The "R/O" access status imposes limitations on the commands that you can run to change the data in a file. The file PRINTOUT.DOC has R/W status.

1.3 Disk Space Consumed by Groups of Files

To obtain the status of all of the files on a disk without entering a command line specifying each file on the disk, the drive letter and the ambiguous filename "*. *" can be entered, as in the following command line:

```
A>STAT B:*. * RETURN
```

Such an entry might produce a display like this:

```
  Recs  Bytes  Ext Acc
   175    22k    2 R/W    B:DATE0927.DOC
   160    20K    2 R/W    B:DATE1003.DOC
    15     2k    1 R/W    B:REPORT1.TXT
     9     1K    1 R/W    B:REPORT2.FIL
   119    15k    1 R/O    B:THATFILE.COM
   201    25k    2 R/O    B:THISFILE.COM
Bytes Remaining On B:  26k
```

(Such an operation will only reveal the files within the currently logged user area.) STAT can also convey the status of groups of files on a disk when ambiguous file names are entered with the asterisk and/or question mark representing only part of the file names, as in the following sample command lines:

```
A>STAT B:*.COM RETURN

A>STAT B:DATE????.DOC RETURN

A>STAT B:REPORT?.* RETURN
```

1.4 Measuring the Size of a File

In addition to ascertaining the actual amount of disk space occupied by a file, STAT can also report on the outermost boundaries of the disk space occupied by a file, or the difference between the first and last sectors occupied by the file.

In the case of sequential files, the size of the file is the same as the number of records because records are recorded on adjacent sectors. The records of Random files, however, are often scattered over a wider disk area than they actually occupy. To determine the size of a file, the normal command line for status of a particular file is entered, followed by a space and the characters "$" and "S", as shown:

A>**STAT C:PRINTOUT.DOC $S** ***RETURN***

The preceding entry might produce the response:

```
   Size  Recs  Bytes  Ext Acc
    140   140    18k    1 R/W C:PRINTOUT.DOC
 Bytes Remaining On C:  72k
```

NOTE: If disks are switched from one drive to another between STAT runs, the individual file statistics will be accurate, but the "Bytes Remaining on X: nnk" message will display a kilobyte count that reflects the remaining disk space on the disk that was first logged within that particular drive since bootup. Therefore, to ensure a correct listing of remaining disk capacity, you should perform a warm boot (enter ***CTRL-C***) if disks are switched between STAT runs.

2 ASSIGNING STATUS

The STAT utility also performs functions which allow disk protection, manipulation of file indicators, disk parameter statistics, user number display, and input/output device assignment. A list of these special STAT functions can be invoked by entering the following command:

A>**STAT VAL:** ***RETURN***

In response, STAT will display the following list:

```
Temp R/O Disks: d: =R/O                                (2.1)
Set Indicator:  d:filename.typ $R/O $R/W $SYS $DIR  (2.2, 2.3)
Disk Status   : DSK: d:DSK:                            (2.4)
User Status   : USR:                                   (2.5)
Iobyte Assign:                                         (2.6)
CON:  = TTY: CRT: BAT: UC1:
RDR:  = TTY: PTR: UR1: UR2:
PUN:  = TTY: PTP: UP1: UP2:
LST:  = TTY: CRT: LPT: UL1:
```

2.1 Temporarily Changing a Disk's Access Mode

The category title "Acc" in many STAT displays stands for the access mode of a file. A file's access mode could be either R/W or R/O. The R/W (Read/Write) mode allows you to write data to the file or erase data from it. The R/O (Read/Only) mode prevents insertion or deletion of data within the file.

To temporarily change the access mode of all files on a disk from R/W to R/O, requires the entry of "STAT", a space, the name of the drive containing the disk to be protected, a colon, an "=" sign, the new access mode "R/O", and a carriage return. The following example illustrates such a command line:

A>**STAT B: = R/O** ***RETURN***

After the entry of this command line, no data or software on disk B can be erased or overwritten. The protection furnished by the R/O access mode will remain in effect until a warm or cold boot is performed.

If you protect files in this manner, invoke the ED utility, and try to write to or delete from a file on the disk, the following error message is displayed:

```
Bdos Err On B: R/O
```

CP/M will not allow alteration to file contents, and will perform a warm boot with the next key pressed after this message is displayed. After this warm boot is performed, the disk's files are reset to the R/W access mode.

2.2 Changing a File's Access Mode

The R/W (Read/Write) mode allows you to write data to the file or erase data from it. The R/O (Read/Only) mode prevents insertion or deletion of file data, and it can be applied to files by the set indicator commands "$R/O" and "$R/W".

Individual files, or groups of them, can be given the R/O status with the entry of "STAT", a space, the drive specification, the file name, a space, the characters "$R/O", and a carriage return. For instance, the file named "ARCHIVAL.DOC" on the disk in drive B can be protected from alteration by the following entry:

```
A>STAT B:ARCHIVAL.DOC $R/O RETURN
```

To give R/O status to a group of files, enter the "*" or "?" wildcard characters in place of the variable part of the names of the files in the group. After this command has been given, the following message is displayed:

```
ARCHIVAL.DOC set to R/O
```

Attempts to write to or erase from this file using any text editor will produce the message:

```
Bdos Err On B: File R/O
```

You must enter a keyboard character to warm boot the system. The R/O attribute is recorded in the directory and will not change if the system is rebooted. The file can be made Read/Write again by entering a similar command with the characters "$R/W" in place of "$R/O", as in the following example:

```
A>STAT B:ARCHIVAL.DOC $R/W RETURN
```

2.3 Changing a File's System Status

Individual files, or groups of them, can be given the system status, which prevents their mention in directory listings, prevents access by the editor, and prevents copying by PIP (except with the "[R]" parameter). This status is applied to a file when the entry of "STAT", the drive specification, and the file name is followed by the characters "$SYS". For instance, the file "SYSTEM.COM" on drive B becomes ineligible for directory mention, editing, and copying (except with [R]) when the following command is entered:

```
A>STAT B:HIDEFILE.COM $SYS RETURN
```

After this command has been given, the message:

```
HIDEFILE.COM set to SYS
```

is displayed.

To illustrate how this status protects a file, efforts to edit the file using the ED utility will produce the message:

```
"SYSTEM" FILE NOT ACCESSIBLE
```

In addition, a check of this file using a different form of the STAT command will also produce noteworthy results, as shown. If you now enter the command:

```
A>STAT B:HIDEFILE.COM RETURN
```

the following display will appear:

```
 Recs  Bytes  Ext  Acc
   38     5k    1  R/W B: (HIDEFILE.COM)
Bytes Remaining On B:  19k
```

When a file bears the SYS status, it will have parenthesis around its listing in a STAT file report.

The system attribute can be removed from this file by the following entry:

A>**STAT B:HIDEFILE.COM $DIR** ***RETURN***

The **$DIR** argument in a STAT command line lifts all of the restrictions imposed by the $SYS argument in a previous STAT command.

2.4 Listing Disk Characteristics

Characteristics of all disks being used can be obtained by entering "STAT", a drive specification, "DSK:", and a carriage return, as shown:

A>**STAT B:DSK:** ***RETURN***

STAT will list a display similar to the following:

```
  B: Drive Characteristics
736: 128 Byte Record Capacity
 92: Kilobyte Drive Capacity
 64: 32 Byte Directory Entries
 64: Checked Directory Entries
128: Records/ Extent
  8: Records/ Block
 20: Sectors/ Track
  3: Reserved Tracks
```

Where the letter "B" to the left of "Drive Characteristics" indicates the selected drive, which contains the disk being analyzed (a 5.25-inch hard-sectored disk in the example above);

where the second and third lines indicate the disk's total "Record Capacity" and "Kilobyte Drive Capacity", respectively (these totals are sometimes dependent upon the selections made during the FORMAT utility);

where the directory can hold a maximum of "64" entries;

where the number of "Checked Directory Entries" is usually identical to the directory size for floppy disks, since this mechanism is used to detect changed disk media during CP/M operation without an intervening warm boot;

where the number of "Records/ Extent" indicates the addressing capacity of each directory entry;

where the number of "Records/ Block" indicates the basic size of a block. By knowing that a record contains 128 bytes, you can calculate the number of bytes in a block;

where the number of "Sectors/ Track" can help you to determine the number of sectors on the disk if you multiply this number by the number of tracks on the disk; and

where the "Reserved Tracks" are reserved for the operating system and the disk file directory.

Any drive from A through P can be specified in this command, as long as it is a valid drive in the hardware environment and operating system being used. If no drive is specified in the command line, then disk characteristics will be displayed for the disk in each drive that has been logged since the last cold boot.

2.5 Examining User Areas

User access to certain files can be controlled by assigning user areas to the files. (See text on USER resident command.) The following form of the STAT utility will display a list of the numbered user areas which have files on the currently logged disk.

A>**STAT USR:** ***RETURN***

This entry might produce the response:

```
Active User : 0
Active Files: 0 1 3
```

The first line of the display lists the currently logged user area, or "Active User", which was set at zero either by the last USER command, or by a cold boot.

The second line lists the other user area numbers that have been established for files on the disk. This list is displayed as "Active Files", and indicates that additional files can be accessed by logging user area one or three.

To determine the "Active User" and "Active Files" on a non-default disk, a similar STAT command can be entered with a drive specification at the end of the command line, as in the following example:

A>**STAT USR:C:** ***RETURN***

You can examine the directories of the other user areas by first issuing the USER resident command with the appropriate area number. The DIR (directory) resident command will then help to determine what files are in a particular user area.

2.6 Temporarily Matching Logical and Physical Devices

The STAT utility also enables you to monitor and temporarily control the assignment of physical input/output devices to the appropriate logical device names. In general, there are four logical peripheral devices which, at any given time, are each assigned to one of several physical devices. The four logical devices, and the physical devices assigned to each of them, are shown in the display invoked by the **STAT VAL:** command, and in the following table:

LOGICAL DEVICE NAME	PHYSICAL DEVICE NAME	DESCRIPTION AND/OR CATALOG NAME OF RECOMMENDED NAME NAME INPUT/OUTPUT MACHINE
CON:	TTY:	Any non-handshaking RS-232 ASCII terminal at port 0D0H
	CRT:	Any non-handshaking RS-232 video terminal at port 0E8H
	BAT:	A batch pseudo device using RDR: for input and LST: for output
	UC1:	Any handshaking RS-232 terminal with ETX/ACK protocol, eg. Diablo KSR 1640 printing terminal
RDR:	TTY:	Any non-handshaking RS-232 ASCII terminal at port 0D0H
	PTR:	Null source, not implemented, returns an "end-of-file" character when accessed
	UR1:	An input modem at port 0D8H
	UR2:	System terminal
PUN:	TTY:	Any non-handshaking RS-232 ASCII terminal at port 0D0H
	PTP:	Null sink, not implemented
	UP1:	An output modem at port 0D8H
	UP2:	System terminal
LST:	TTY:	Any non-handshaking RS-232 ASCII terminal at port 0D0H (eg. WH-34)
	CRT:	Any non-handshaking RS-232 video terminal at port 0E8H
	LPT:	line printer (eg. H-14, WH-14, WH-24, H/Z25, WH-36, Epson MX80)
	UL1:	Any Diablo printer

Table 2-4
Possible Device Assignments

EXAMINATION OF CURRENT DEVICE ASSIGNMENTS

The current assignments of the physical devices to logical devices are shown in the display invoked by the following command entry:

```
A>STAT DEV: RETURN
```

This command produces a display similar in form to the following:

```
CON: is CRT:
RDR: is UR1:
PUN: is UP1:
LST: is UL1:
```

CHANGING DEVICE ASSIGNMENTS

The current logical to physical device assignment can be changed by entering a STAT command in the following form:

```
A>STAT {logical device}:={physical device}: RETURN
```

Where both the "**logical device**" and the "**physical device**" are specified by their four-character names.

For example, the following entry will change the physical device assigned to the console logical device (CON:) to a teletype printer terminal (TTY:):

```
A>STAT CON:=TTY: RETURN
```

Several device assignments can be affected in one command by entering equations in series, separated by commas, as shown:

```
A>STAT RDR:=PTR:,PUN:=PTP:,LST:=LPT: RETURN
```

The physical device names may or may not actually correspond to the devices which the names imply. For instance, the PTR: device could be implemented as a cassette/write operation if you wish. The exact correspondence and driving subroutine is defined in the BIOS portion of CP/M.

Physical to logical device pairings assigned by the STAT utility will survive a warm boot, but are replaced by default assignments when a cold boot is performed. Default assignments, that will survive a cold boot, may be changed through the CONFIGUR utility.

3 STAT ERROR MESSAGES

```
Bad Delimiter
```

EXPLANATION: You entered a command line with improper syntax while assigning some type of status (2). Command should be re-entered in the proper form.

```
Invalid Assignment
```

EXPLANATION: You tried to make a device assignment with a device name that is not listed in Table 2-4. You should re-enter assignment using valid device names.

```
Invalid File Indicator
```

EXPLANATION: You tried to set a file for a particular status using an indicator or syntax that is not acceptable. You should re-enter command with a proper indicator.

```
File Not Found
```

EXPLANATION: STAT found none of the specified files on the logged disk. You should re-enter command specifying different file names, or perform a warm boot and insert a disk that has the specified file(s).

```
Wrong CP/M Version (Requires n.n)
```

EXPLANATION: The STAT.COM file in use must be used with the operating system version corresponding to the number ("n.n") in the error message.

```
Invalid Disk Assignment
```

You entered a disk assignment command that included a status other than "R/O" or "R/W". Re-enter the command with correct syntax.

```
** Aborted **
```

You tried to enter a command while a STAT status command was still executing. When you assign status (such as R/O, R/W, SYS, or DIR) to a file, you must wait for the status command to execute completely and for the system prompt to appear before you type the next command.

SUBMIT

The Utility that Triggers Automatic Execution of CP/M Commands

The SUBMIT utility enables you to initiate automatic execution of several sequential commands (3) by issuing only one SUBMIT command (2). The SUBMIT command accesses a file containing a precomposed sequence of commands (1). You can also vary the execution of this command sequence with each invocation by substituting command line parameters such as file names, drive names, and device names for variables inserted into the command file.

1 SUB FILES

A SUB file is a file with the "SUB" extension, which is comprised of a sequence of command lines composed using a text editor (eg ED). The SUB file is the source of the command sequence referenced by the SUBMIT command. The command lines within a SUB file can contain prototype parameters to represent command line fields. These prototype parameters appear in the SUB file in the form:

```
    $n
```

Where each numeral ("n") preceded by a dollar sign ("$") takes the place of a file name, drive name, or device name which you will substitute into the SUB file with an actual parameter entered in the SUBMIT command line.

The following example shows the text of the SUB file with the name "MULTIJOB.SUB", which contains two prototype parameters:

```
ASM $1
DIR $1
ERA *.BAK
PIP $2:=$1.PRN
ERA $1.PRN
```

2 SUBMIT COMMAND LINES

The SUBMIT utility is invoked and operated by user entry of a command line in the form of the following example:

SUBMIT MULTIJOB PROGRAMX LST *RETURN*

Where **MULTIJOB** is the name of the SUB file filled with batch commands; and

where **PROGRAMX** and **LST** are parameters which will be substituted into one or more locations in the batch file.

An actual SUBMIT parameter is the explicit specification of a command, file, drive, or device which you want to substitute for a prototype parameter within the SUB file. The number of parameters specified in a SUBMIT command line must correspond to the number of different prototype parameters that appear within the SUB file in the form "$n".

3 SUBMIT EXECUTION

After the entry of a SUBMIT command line, SUBMIT will create a file named $$$.SUB, and send it to the Console Command Processor (CCP) portion of the operating system. The CCP recognizes the commands in the SUB file and executes them in sequence. The $$$.SUB file contains the actual parameters expressed in the SUBMIT command substituted for the corresponding prototype parameters which are imbedded in the original SUB file. If created in response to the preceding example command line, the $$$.SUB file would contain the following commands:

```
ASM PROGRAMX
DIR PROGRAMX.*
ERA *.BAK
PIP LST: = PROGRAMX.PRN
ERA PROGRAMX.PRN
```

4 SUBMITTING A SUB FILE FROM A DRIVE OTHER THAN A

The $$$.SUB file must exist on a disk in drive A for the commands to be executed. However, the SUB file used by SUBMIT to create the $$$.SUB file can reside on a disk in any existing drive; and the $$$.SUB file can be created on a disk in any existing drive.

If $$$.SUB is created on a disk in a drive other than drive A, you must adhere to one of the following sets of stipulations to initiate command execution:

- The drive location of the disk containing the SUB file must be specified in the SUBMIT command line. $$$.SUB is created on a bootable disk, the disk is placed in drive A, and the system must be rebooted with drive A as the default drive.

- The drive location of the disk containing the SUB file must be specified in the SUBMIT command line. The $$$.SUB file can be created on any disk and then copied to a disk in drive A using the PIP utility. The disk in drive A must be write enabled and have sufficient space to accommodate the $$$.SUB file.

5 ABORTING SUBMIT COMMAND EXECUTION

Execution of submitted commands can be aborted at any time by entering a ***DELETE, RETURN,*** or ***CTRL-C*** when the command is read and echoed. In this case, the $$$.SUB file is removed, and the subsequent commands come from the console. Command processing is also aborted if the CCP detects an error in any of the commands. Programs which execute under CP/M can abort processing of command files when error conditions occur by simply erasing any existing $$$.SUB file.

6 UNCONVENTIONAL TEXT IN A SUB FILE

You can insert comments into a SUB file so that they will be displayed with the other messages produced by the processing of SUB file commands. To ensure that the CCP ignores the comments and does not interpret them as commands, they must be entered on a line which begins with a semicolon (;).

To introduce literal dollar signs into a SUB file, you may enter "$$", which translates to a single "$" within the SUB file. Furthermore, a caret symbol (^) can precede the alphabetic character X, which produces a single CTRL-X character within the file.

7 CHAINED BATCH COMMANDS

The last command in a SUB file can be a SUBMIT command, which could either initiate execution of the commands within another SUB file, or repeat execution of the commands within the same file. You can include different parameters to be substituted for SUB file prototypes to cause varied repetition of the same set of commands.

8 SUBMIT ERROR MESSAGES

```
Error on Line nnn     No 'SUB' File Present
```

EXPLANATION: SUBMIT command can only be issued if a file with the "SUB" extension exists on a logged disk, and if that file is specified in the SUBMIT command line. You should re-enter the command with a different argument, or log the disk with the appropriate SUB file.

```
Disk Write Error
```

EXPLANATION: SUBMIT found insufficient disk space while trying to create $$$.SUB file on the disk. You should clear off some space on the disk or use another disk.

```
Parameter Error
```

EXPLANATION: You entered an invalid parameter in the SUB file. You should edit SUB file, with only decimal integers following the "$" sign for a parameter.

```
Directory Full
```

EXPLANATION: SUBMIT tried to create SUB file on a disk that didn't have enough directory space to accommodate the SUB file name. You should clear off some space on the disk or use another disk.

```
Cannot Close, Read/Only?
```

EXPLANATION: SUBMIT is trying to write data to a disk (as one of the activities prescribed in a SUB file command line), but the file being written to has R/O (read/only) status. You must use the STAT utility to change the file's status to "R/W" (read/write); or change the SUB file command line that is trying to write to the file; or enter different SUBMIT parameters.

SYSGEN

The Utility that Puts the Operating System on a Disk

The SYSGEN utility is used to transfer the operating system to a disk. Under some circumstances, SYSGEN does this task by itself. Sometimes SYSGEN needs the help of other utilities before an entire, usable system can be put on the disk.

1 SYSGEN INVOCATION

No matter which SYSGEN method is to be performed, the SYSGEN utility is invoked by entering the following command at the system prompt:

A>**SYSGEN *RETURN***

A display in the following form will appear:

```
SYSGEN VER 2.0.04

SOURCE DRIVE NAME (OR RETURN TO SKIP):
```

The next entry you makes depends on the method used.

2 SYSGEN METHODS

You must consider the situation before running SYSGEN, so that the appropriate SYSGEN method is used.

An operating system must be compatible with the disk it is copied to, and Heath/Zenith supports four types of disk: (5.25-inch hard-sectored, 5.25-inch soft-sectored, 8-inch, and any disk used in the H/Z67 Winchester Disk drive model). Furthermore, Heath/Zenith computers can accommodate systems of different memory capacities. The MOVCPM utilities are used to customize the system for both disk type and memory capacity.

- If a MOVCPM utility was not necessary to customize the operating system for disk type or memory capacity, then the "Disk to Disk System Copying" method (2.1) should be used. (The BSYSGEN utility could also be used in this case.)
- If a MOVCPM utility is used to customize an operating system before transfer, then this system will reside in the computer when SYSGEN is invoked, and the "Computer to Disk System Copying" method (2.2) should be used.

NOTE: When the CP/M Operating System is copied to a disk, it is moved in two parts: the system kernel and the BIOS.SYS file. The "Computer to Disk System Copying" method does not copy the file BIOS.SYS. If this method is used, then the PIP or MAKEBIOS utility must also be used to copy the BIOS.SYS file.

2.1 Disk to Disk System Copying

If the operating system is being copied between two disks of the same type, you can copy both the system kernel and the BIOS.SYS file to the destination disk using this SYSGEN method. (This method is used when a MOVCPM activity does **not** precede the copy activity.)

Under these circumstances, you should answer the "SOURCE DRIVE NAME" prompt by typing the letter for a drive name, as shown:

```
SOURCE DRIVE NAME (OR RETURN TO SKIP): x
```

Where **x** is the letter of the source drive.

SYSGEN will respond with a prompt in the following form:

```
SOURCE ON x, THEN TYPE RETURN
```

You should answer this prompt with a carriage return. SYSGEN will read the system kernel from the source disk, and signal that it has done so with the following message:

```
FUNCTION COMPLETE
```

Then SYSGEN will offer the option of copying BIOS.SYS from the source disk with the prompt:

```
COPY BIOS.SYS (Y/N):
```

If the you wish to copy the file BIOS.SYS from the source disk, then **Y** should be pressed, and SYSGEN will again display the message:

```
FUNCTION COMPLETE
```

If you wish to copy a BIOS.SYS file that is not stored on the source disk, (or a file by another name that contains a BIOS) then **N** should be pressed.

After either entry, SYSGEN will prompt for the drive that contains the destination disk. You should answer the prompt as shown:

```
DESTINATION DRIVE NAME (OR RETURN TO REBOOT): y
```

Where **y** is the letter of a valid, working drive.

Then SYSGEN will display the prompt:

```
DESTINATION ON y, THEN TYPE RETURN
```

You should enter a carriage return at this prompt, to confirm the destination drive choice. SYSGEN will put the system kernel, and in some cases the BIOS.SYS file, onto the destination disk. (The disk in drive “y”).

Then SYSGEN will again display the prompt:

```
DESTINATION DRIVE NAME (OR RETURN TO REBOOT):
```

If you wish to copy the same system components to a different destination disk, then the letter of the drive containing this disk should be entered.

If you do not wish to SYSGEN other disks, then a carriage return should be pressed. The SYSGEN activity will end, and CP/M will display the system prompt.

2.2 Computer to Disk System Copying

If you have just run a MOVCPM utility to customize the operating system for disk type or memory capacity, then this system still resides in the memory of the computer. You can copy this system from the computer to the disk by using this method.

When SYSGEN prompts for "SOURCE DRIVE NAME", you must enter a carriage return—**not** a drive name.

```
SOURCE DRIVE NAME (OR RETURN TO SKIP) RETURN
```

SYSGEN will now prompt you for the drive that contains the destination disk. You should answer the prompt as shown:

```
DESTINATION DRIVE NAME (OR RETURN TO REBOOT) y
```

Where **y** is the letter of the drive containing the disk that is to receive the system kernel. (The BIOS.SYS file can not be copied through this SYSGEN method.)

SYSGEN will request confirmation with the prompt:

```
DESTINATION ON y, THEN TYPE RETURN
```

You should enter a carriage return at this prompt. SYSGEN will put the system kernel onto the destination disk (the disk in drive y).

Then SYSGEN will again display the prompt:

```
DESTINATION DRIVE NAME (OR RETURN TO REBOOT):
```

If you wish to put the system kernel on another disk, then type the letter of the drive containing the other disk.

If you do not wish to put the system kernel on another disk, and if you have not specified drive A at a previous "DESTINATION DRIVE" prompt, then enter a carriage return at this "DESTINATION DRIVE" prompt. The SYSGEN activity will end and CP/M will display the system prompt.

Under this method, a carriage return should **not** be entered at this prompt.

If you do not wish to SYSGEN other disks, and you did specify drive A at a previous "DESTINATION DRIVE" prompt, then you must reset the computer at this "DESTINATION DRIVE" prompt (rather than entering a carriage return). Resetting the computer at this prompt is necessary because entry of a carriage return at this prompt causes a warm boot, which would cause the new system kernel (recently recorded on the disk in drive A) to be loaded into computer memory. It is undesirable to load any part of this new system kernel into memory on a warm boot because it might have just been changed by a MOVCPM utility. Therefore, it might be of a different size than the system kernel that was loaded into memory at bootup.

When you use SYSGEN by the "Computer to Disk" method, SYSGEN will not copy the BIOS.SYS file. Therefore, in order to make the destination disk bootable, you must copy the BIOS.SYS file to it using the PIP utility or the SUBMIT and MAKEBIOS utilities.

The PIP utility can be used to transfer the BIOS.SYS file, as shown:

A>**PIP y:=x:BIOS.SYS[RV]** ***RETURN***

Where **y** is the destination disk receiving BIOS.SYS;

where **x** is the source disk from which BIOS.SYS is copied; and

where **[RV]** are PIP parameters used to help you copy a file that has system status, and to help you verify the accuracy of the copy operation.

The MAKEBIOS utility can be used to transfer the BIOS.SYS file, as shown:

A>**SUBMIT MAKEBIOS y x** ***RETURN***

Where **y** is the destination disk receiving BIOS.SYS; and

where **x** is the disk containing the files BIOS.ASM, MAKEBIOS.COM, MAKEBIOS.SUB, and PREL.COM.

NOTE: The MAKEBIOS utility customizes the system before transferring it to the destination disk. See the text on MAKEBIOS for details on its use.

3 SYSGEN ERROR MESSAGES

```
INVALID DRIVE NAME
```

EXPLANATION: You must specify a drive letter corresponding to the names of drives that exist in the hardware environment, and are recognized by the operating system that was loaded at bootstrap.

```
NO SOURCE FILE ON DISK
```

EXPLANATION: The drive specified as "SOURCE DRIVE" did not contain the file BIOS.SYS. You should use a different disk in the source drive, or rename a BIOS file that has been given a different name to "BIOS.SYS".

```
SOURCE FILE INCOMPLETE
```

EXPLANATION: SYSGEN failed in an attempt to copy the file BIOS.SYS from the disk in the source drive. This file might have been damaged by disk media flaws or partially overwritten. You should reset, perform bootstrap, and re-enter the SYSGEN command using a different disk in the source drive.

```
WRITE ERROR DURING BIOS.SYS
```

EXPLANATION: You should try SYSGEN again with a destination disk that is write enabled, formatted, and has at least 6 kilobytes of free space.

```
ERROR READING BIOS.SYS
```

EXPLANATION: SYSGEN failed in an attempt to copy the file BIOS.SYS from the disk in the source drive. This file might have been damaged by disk media flaws or partially overwritten. You should reset, perform bootstrap, and re-enter the SYSGEN command using a different disk in the source drive or using a different disk to perform bootstrap.

```
PERMANENT ERROR, TYPE RETURN TO IGNORE
```

EXPLANATION: The system kernel or BIOS.SYS file are either incompatible with the destination disk type or otherwise flawed. You should reset, perform bootstrap, and re-enter the SYSGEN command using a different disk in the source drive or using a different disk to perform bootstrap. Under some circumstances, you must use a MOVCPM utility before SYSGEN.

TYPE

The Resident Command that Displays File Contents on the Console

The TYPE command displays to the console the contents of specified files (1). Fast scrolling displays can be controlled to make them readable (2). TYPE is best used on certain kinds of files (3).

1 DISPLAYING A FILE

To produce a console display of the contents of a disk file, you should respond to the system prompt with a command line in the following form:

```
A>TYPE {file name} RETURN
```

Where {**file name**} is the complete, explicit name of a file.

If you must examine a file that resides on disk in a non-default drive, this drive must be specified immediately before the file name specification. For example, if the file "LOGDRIVE.TXT" resides on a disk in drive B, its contents will be displayed on the screen when the following command is entered:

```
A>TYPE B:LOGDRIVE.TXT RETURN
```

2 SCREEN DISPLAY CHARACTERISTICS

The display produced by TYPE will scroll by on the console until the entire contents of the file have been displayed. You can terminate the display during its execution and return to the operating system by pressing any character other than CTRL-S. The display scroll can be halted temporarily by entering ***CTRL-S*** character, and restarted by pressing CTRL-S (or any other character) again.

Some of the special features (boldface type, underlining, etc.) inserted into text files by some word processing systems might not be indicated in screen displays produced by the TYPE command.

3 PRACTICAL USAGE OF TYPE

The most desirable usage of the TYPE command is to display files composed with ASCII characters (printable letters and numbers). If the TYPE command is issued for a file with a ".COM" extension (or for any file stored on the disk in binary form) the resulting screen display will contain a meaningless series of characters.

To obtain a paper printout of the file, enter a ***CTRL-P*** before entering the TYPE command line. Printouts will not exhibit special features (boldface type, underlining, etc.) that are inserted into some text files by word processing programs.

USER

The Resident Command that Controls User Access to File Areas

The USER command enables a user to access only files in a specified area of the disk directory (1). You can log into USER areas by invoking a single USER command line (2). You must implement a sequence of commands to put files into a non-zero USER area (3).

1 USER INVOCATION

File directories supported by the CP/M system are divided into 16 user areas. When a user is logged into a particular USER area, then only the files within that area are accessible (except by implementing the PIP utility with the "G" parameter).

The USER command is useful when several users have files stored on the same disk. File directory (DIR) and status checks (STAT) invoked by a particular user will list only the files in the user's specified area.

Whenever a cold boot is performed, you are automatically logged in to USER area number 0. When in this area, only files in USER area 0 of the directory are accessible.

2 INVOKING USER

To make a user area other than number 0 accessible, you should respond to the system prompt by entering a command line in the form:

A>**USER** {**area number**} ***RETURN***

Where{**area number**} indicates the number, from 0 to 15, of an area that contains files that you wish to access.

The USER command can be issued whenever the system prompt appears. The currently-logged USER area will be in effect for all of the disks in the hardware environment, until a different USER area is logged or a cold boot is performed.

3 PULLING FILES INTO USER AREAS

Files can be pulled into a USER area that contains the file PIP.COM, but to put the file PIP.COM into that USER area in the first place requires a special process.

To put the file PIP.COM into a USER area (other than area zero), you should follow the following steps, in sequence:

3.1 Insert disks that have the PIP.COM utility and at least 8 kilobytes of space available.

3.2 Log USER area zero (or whatever USER area contains PIP.COM) by entering the command **USER 0 *RETURN*.**

3.3 Invoke the PIP utility by entering the command **PIP *RETURN*.** (Specify the name of the drive that contains PIP.COM if it is not the default drive.)

3.4 Press ***RETURN*** at the asterisk (*) prompt.

3.5 Log the desired USER area by entering the command **USER n *RETURN*,** where **n** is the number (0-15) of the USER area you wish to establish.

3.6 Save the portion of the computer memory that now contains the file PIP.COM by entering the command **SAVE 29 PIP.COM *RETURN*.** (Specify the name of the drive on which you wish to establish the USER area if it is not the default drive.)

These six steps will place the file "PIP.COM" into the desired USER area (n). Once PIP.COM resides within a USER area, it can help you to pull other files into that USER area. The following example display demonstrates how this procedure might appear on the console:

```
(3.2)     A>USER 0 RETURN
(3.3)     A>PIP RETURN
(3.4)     A>* RETURN
(3.5)     A>USER 5 RETURN
(3.6)     A>SAVE 29 PIP.COM RETURN
```

If you have followed the preceding example procedure, file copies could now be pulled into USER area 5 by the entry of commands in the form:

```
A>PIP B:={drive}:{file}[G{source area}] RETURN
```

Where {**drive**} is the name of the drive from which file copies are being pulled;

where {**file**} is the name of the file being pulled into the new USER area; and

where{**source area**} is the number of the USER area from which files are being pulled.

XSUB

The Utility that Batches Commands Within Utility Programs for Automatic Processing

The XSUB utility extends the power of the SUBMIT utility to enable you to invoke utility programs, and then execute commands within these programs, while making only one entry at the terminal.

1 XSUB OPERATION

The XSUB command line is entered as the first line in a SUB file. A SUB file is a text file composed of command lines that are executed in sequence when a SUBMIT command line is entered at the system prompt.

When the XSUB command is executed, it moves beneath the component of the operating system known as the Console Command Processor (CCP). All command lines after the XSUB command line are processed by XSUB, so that programs which usually prompt you to make entries on the keyboard will accept entries directly from the SUB file.

The XSUB program remains in memory after execution, and displays the message "(xsub active)" each time a warm boot is performed during the execution of the $$$.SUB file. Thus the initial XSUB command in a SUB file can remain in effect throughout the execution of every command in the SUB file.

XSUB will display the message "XSUB already present" as you enter the command **XSUB** at the system prompt (through the console). However, this remnant of the XSUB program will not go into effect when subsequent SUB files are submitted. Therefore, you must make "XSUB" the first line of any SUB file in which XSUB execution is desired.

NOTE: XSUB will not support the submission of input for programs which, when run, prompt for single character input. Hence, the only programs that can effectively be used in a SUB file under XSUB are programs that accept input lines ended with a carriage return, such as DDT, PIP, LIST, and ED.

2 XSUB EXAMPLE

The file "MEGAJOB.SUB" contains the following command lines:

```
XSUB
DDT
I$1.HEX
R
G0
SAVE 4 $2.COM
```

Processing of this SUB file could be initiated by entering the following command line:

```
A>SUBMIT MEGAJOB PRGRMY PRGRMZ RETURN
```

The preceding entry creates a $$$.SUB file in which the primary file name "PRGRMY" is substituted for the prototype name "$1" of the SUB file, and "PRGRMZ" is substituted for the "$2" prototype parameter.

As commands are read from the $$$.SUB file, the XSUB utility enters computer memory. Then DDT enters and executes the DDT commands "IPRGRMY.HEX", "R", and "G0", which are also included in the $$$.SUB file. The DDT command "G0" has the same effect as a warm boot, which enables the operating system to process the final command in the $$$.SUB file, "SAVE 4 PRGRMZ.COM".

Appendix A:

Operating System Error Messages

This appendix explains error messages produced by the CP/M Operating System.

CONVENTIONAL ERROR MESSAGES

```
BAD LOAD
```

Causes: You tried to run 3) a program that requires 1) more memory than CP/M has available or 2) more memory than your computer has available.

Remedies: 1) Use a MOVCPM utility and the SYSGEN utility in sequence to expand CP/M to the memory limit of your computer. 2) Add more memory to your computer. 3) Make your program smaller.

```
Bdos Err On x: Bad Sector
```

Causes: (Where "x:" is the name of a drive which CP/M tried and failed to read data from, or write data to, a disk.) 1) Disk used for a data transfer operation is damaged or extremely worn. 2) Disk drive controller is malfunctioning. 3) Disk being accessed was formatted by a disk drive controller that is not IBM compatible. 4) Disk being accessed has not been formatted at all. 5) Drive being accessed contains no disk. 6) Accessed disk is write protected.

Remedies: Enter a ***CTRL-C*** to abort the activity and perform a warm boot, or enter a carriage return to skip over the location of the error condition, or press **R** to retry the activity at the location of the error condition. If the activity is impossible at this location, then 1) inspect or replace disk. 2) inspect or replace controller card. 3) & 4) FORMAT disks in your own hardware environment. 5) Access a drive with a disk. 6) Write enable disk and perform warm boot before trying to write data to disk.

```
Bdos Err On x: Select
```

Causes: (Where "x:" is the name of a drive which you referred to in a command, but which CP/M cannot access or does not recognize.) 1) You entered a drive name that does not correspond to a drive in your hardware. 2) You tried to access a Winchester partition that was not first assigned a drive name.

Remedies: In all cases, pressing any keyboard character after such an error message will cause a warm boot. 1) Enter drive names that correspond to drives in your hardware. 2) Use ASSIGN utility on a partition before trying to access the partition.

```
Bdos Err On x: R/O
```

Causes: (Where "x:" is the name of a drive at which CP/M refuses to read or write data. 1) You tried to write to a file that had been given "Read/Only" status by the STAT utility. 2) You tried to write to a disk that had been given temporary "Read/Only" status by the STAT utility. 3) You tried to switch disks in a drive and then write to the new disk without performing a warm boot.

Remedies: In all cases, pressing any keyboard character after such an error message will cause a warm boot. 1) Change file status to "Read/Write" using the STAT utility. 2) Perform a warm boot in any fashion (such as the ***CTRL-C*** entry) before trying to write data to the disk. 3) Perform a warm boot (with a ***CTRL-C***) after switching disks, but before accessing a switched disk.

```
NO FILE
```

Causes: 1) You invoked DIR Resident Command for a directory of disk files, and the file(s) specified not on the disk in the logged user area. 2) You invoked TYPE Resident Command to display contents of a file that was not on the disk in the logged user area. 3) You invoked ERA Resident Command to erase a file that was not on the disk in the logged user area. 4) You invoked REN Resident Command to try to rename a file that was not on the disk in the logged user area.

Remedies: 1) Unless such a message gives you adequate information, log to a different user area with the USER Resident Command before trying DIR command. 2), 3), & 4) Check disk directory using DIR Resident Command or STAT utility to obtain correct file name.

```
FILE EXISTS
```

Cause: You tried to rename a file (with REN Resident Command) using a name that another file on the same disk already has.

Remedy: Give a different name to the file you are renaming, or move the file that already has the name to a different disk or user area.

```
NO SPACE
```

Cause: You invoked SAVE Resident Command to store memory contents on a disk that didn't have enough room for the data from memory.

Remedy: Re-enter SAVE command specifying the name of a drive containing a disk that has sufficient room for the file, or specifying fewer pages of data.

```
SYNCHRONIZATION ERROR
```

Causes: You tried to invoke a copy of the SYSGEN utility that did not bear the same serial number identification as the Operating System.

Remedies: Only use the MOVCPM or SYSGEN utilities when the utility and the system (or backup copies of both) originally came from the same Distribution Disk.

```
CAN'T OPEN BIOS.SYS
```

Cause: You tried to perform bootstrap with a disk that does not contain an acceptable BIOS.SYS file.

Remedy: You should copy a properly customized BIOS.SYS file to the disk, or rename an already resident BIOS file to the name "BIOS.SYS".

```
ERROR DURING WARM BOOT - PRESS ANY KEY
```

Cause: You tried to perform a warm boot or exit from a utility after changing the position of the disk that originally resided in drive A.

Remedy: You should replace the disk that was originally in drive A: or the current default drive when bootstrap was performed. Then you must perform a warm boot by entering a ***CTRL-C***.

DETAILED DISK ERROR MESSAGES

Detailed disk error messages are additional messages that are displayed with conventional system error messages to provide more specific information about the cause of the error condition. These detailed messages will occur if you turn them "on" through the "Set Disk Parameters" submenu of the CONFIGUR utility.

If the operating system has difficulty in reading from or writing to a disk, it tries the operation again and again, up to ten times. If the disk still can't be read or written at this location, then the system displays a "Bdos Err" error message. (When a 5.25-inch disk drive has such difficulty, it records a "soft error" for each of the ten attempts, incrementing the SECNTnn variable each time.)

Detailed Disk Error Messages are displayed in the following form:

```
Hdd READ ERROR nn

Bdos Err On x: Bad Sector
```

Where "dd" is the drive type that obtained the error;

where "READ" or "WRITE" is the activity that the operating system was trying to perform;

where "nn" is the detailed error code; and

where "x" is the drive name of the drive at which the error was obtained.

Most of the specific detailed error codes are listed in the hardware manuals for the H/Z-37, H/Z-47, and H/Z-67 drive models. This text contains explanations of the detailed codes that occur with the H/Z-17 drive model, and some of those that occur with the H/Z-67 drive model.

H/Z-67 DETAILED ERROR MESSAGES (PARTIAL LISTING)

80 Hardware interface error

81 Hardware interface error

82 Hardware interface error

83 Attempt made to exceed partition boundaries

H/Z-17 DETAILED ERROR MESSAGES

01 Bad Track Error: Probably a problem in the floppy disk mechanism, often caused by setting the disk step rate too fast during a run of CONFIGUR (submenu B).

02 Head Sync Error: An error has occurred in the format of the floppy disk.

04 Header CRC Error: An error in reading the sector identification header, generally caused by defective media.

08 Data CRC Error: An error in reading the data on the sector, can be caused by power failure, power surge, defective media, or defective hardware.

10 Record Not Found: The specified track and sector cannot be located, due to poor alignment, flawed disk media, incorrect disk rotation speed, or the specification of a nonexistent track or sector.

20 Missing Data Sync: An error has occurred in the format of the floppy disk.

40 Write Protect Error: An attempt was made to write to a mechanically write-protected disk.

80 Unit Not Ready: The drive unit was not ready to read or write data. This can be caused by not having a disk in the drive or by leaving the drive open.

Appendix B:

Recommended Hardware for Heath/Zenith CP/M

WITH H/Z-89 COMPUTER

The minimum supportable system that will run under this implementation of CP/M is:

- An H/Z-89 computer system with at least 32K of random access read/write memory, with the Z89-7 or H88-7 ROM modification kit installed (already done on Z89-FA and later models), and with the MTR90 ROM. (If only H/Z-17 and H/Z-47 drives are used, MTR89 ROM will work.)

- One connected, functional disk drive (the H/Z-89 comes with a single drive with a H/Z-17 controller).

This implementation of CP/M for the H/Z-89 will support the following pieces of hardware:

THE COMPUTER:

- An H/Z-89 computer system with 32K or more (64K max) of random access read/write memory, with the Z-89-7 or H-88-7 ROM modification kit installed (already done on Z89-FA and later models), and with the MTR90 ROM. (If only H/Z-17 and H/Z-47 drives are used, MTR89 ROM will work.)

DISK DRIVES:

- From one to three hard-sectored 5.25-inch floppy disk drives (H/Z-17 controller).
- From one to three soft-sectored 5.25-inch floppy disk drives (Z-37 controller).
- Two 8-inch floppy disk drives (H/Z-47 controller).
- One Winchester Disk System—Winchester Disk and 8-inch Floppy Disk (Z-67 controller).

With a maximum of six drives connected and functional. (Note that only two types of drives may be connected at any given time.)

PRINTERS:

- A diablo 630, 1610, 1620, 1640, or 1650 (compatible) printer set to 1200 baud, and an RS-232C cable connected to the socket labeled DCE 340-347. The UL1: logical device driver should be connected to this port. It uses an ETX/ACK handshake protocol to control the printer output.
- An H14, WH14, WH24 or H/Z-25 dot-matrix printer, utilizing the RTS modem control for handshaking, connected to the socket labeled DCE 340-347.

If your printer requires a handshaking convention that is not discussed in the text on the CONFIGUR utility, it will be your responsibility to modify the BIOS physical device driver. The Heath/Zenith Software Group can not assume the responsibility for any such modifications.

- A WH36, or any printer that does not require "busy" handshaking, may be connected to the TTY: logical device, again it is plugged into the socket labeled DCE 340-347. In the Configuration program, specify the port number for the TTY: as being 340Q and move the LST: device to 320Q. (H/Z-89 computers are not currently supplied with port 320Q populated, it is made up of the empty set of sockets on the serial interface card.)
- A WH13 modem connected to the plug labeled DTE 330-337.

WITH H8 COMPUTER

The minimum supportable system that will run under this implementation of CP/M is:

- An H8 computer system with at least 32K of random access read/write memory and with the HA-8-8 Extended Configuration kit installed.

- A terminal connected to the computer with a serial interface at port 350Q (0E8H) on an H8-4 card or port 372Q (0FAH) on an H8-5 card.

- One connected, functional floppy disk drive.

This implementation of CP/M for the H8 will support the following pieces of hardware:

- An H8 computer system with 32K or more (64K max) of random access read/write memory and with the HA-8-8 Extended Configuration kit installed.

- A terminal connected to the computer with a serial interface at port 350Q (0E8H) on an H8-4 card or port 372Q (0FAH) on an H8-5 card.

- From one to three hard-sectored 5.25-inch floppy disk drives (H17 controller).

- Two 8-inch floppy disk drives (H/Z-47 controller)

With a maximum of six drives connected and functional. (Note that only two types of drives may be connected at any given time.)

PRINTERS:

- A Diablo 630, 1610, 1620, 1640, or 1650 (compatible) printer set to 1200 baud, and the RS-232 cable connected to port 340-347 on an H8-4 card. The UL1: logical device driver should be connected to this port. It uses ETX/ACK handshake protocol to control printer output.

- An H14, WH14, WH24 or H/Z-25 dot-matrix printer, utilizing the RTS modem control for handshaking, connected to port 340-347 on an H8-4 card.

If your printer requires a handshaking convention that is not discussed in the text on the CONFIGUR utility (page 2-70), it will be your responsibility to modify the BIOS physical device driver. The Heath/Zenith Software Group can not assume the responsibility for any such modifications.

- A WH36, or any printer that does not require "busy" handshaking, can be connected to the TTY: logical device, again it is plugged into port 340-347 on an 8-4 card. In the Configuration program, specify the number for the TTY: as being 340Q and move the LST: to 320Q.

- A WH13 modem connected to port 330-337 on an H8-4 card.

Appendix C:

Bootstrap

The "Startup Procedures" in "Volume I: CP/M Introductory Guide" provide detailed instructions on performing bootstrap. However, some microcomputers allow you to vary startup procedures by entering alternative bootstrap commands, or by causing the microcomputer to perform bootstrap automatically. This text explains the startup variations that are possible with some Heath/Zenith microcomputers.

ALTERNATIVE BOOTSTRAP COMMANDS

This text section shows you how to use different bootstrap commands to boot up from different drives. But it is first necessary that you understand some of the technical terms used to define the relationship between the microcomputer and disk drives. This text will introduce some new terms used to refer to disk drives. But these terms will apply to your drives only until you've completed bootstrap. After a successful bootstrap, the drive containing your bootable disk automatically becomes drive "A:", and the other disks will assume drive names such as "B:", "C:", "D", "E", or "F".

Your microcomputer contains circuit boards that control your disk drives. Each of these circuit boards, called "controller cards", can control two or three disk drive slots of the same type. For instance, the H/Z-47 controller card controls two 8-inch drives in the H/Z-47 drive model.

All of the drive slots controlled by a single controller card are referred to collectively as one "device". If you have only one kind of disk drive, then your microcomputer contains only one controller card, and the drives controlled by this card are collectively called the "primary device". If you have two kinds of disk drive, then your microcomputer contains two controller cards, and the drives controlled by each card are called either the "primary device" or the "secondary device".

COMMANDS FOR THE H/Z-88, H/Z-89, AND H/Z-90

With an H/Z-88, H/Z-89, or H/Z-90 microcomputer you can boot up with a disk in any of your drive slots because this microcomputer refers to each drive slot by a unique and constant name.

Since each device consists of two or three drives, the individual drive slots within each device are given "unit" numbers 0, 1, or 2. Thus each physical drive slot is identified by a device type ("primary" or "secondary") and a unit number ("0", "1", or "2").

Table C-1 shows all of the bootstrap commands possible within your hardware environment, and the drive that will be accessed by each. (Drives are identified in this table by their "device" and "unit" designations.)

BOOTSTRAP COMMAND ENTRY	VIDEO CONSOLE DISPLAY	DEVICE TYPE ACCESSED	UNIT NUMBER ACCESSED
B (or **B0**)	H:Boot (or H:Boot0)	primary	0
B1	H:Boot1	primary	1
B2	H:Boot2	primary	2
BS (or **BS0**)	H:Boot SD (or H:Boot SD0)	secondary	0
BS1	H:Boot SD1	secondary	1
BS2	H:Boot SD2	secondary	2

Table C-1:
Alternative Bootstrap Commands for the H/Z-88, H/Z-89, and H/Z-90

To perform bootstrap with a bootable disk in a particular drive slot, you type the boostrap command entry and a carriage return. Then the microcomputer will access the drive slot identified by the bootstrap command to copy the CP/M Operating System from the bootable disk.

NOTE: The actual drive slot that is accessed by each of these bootstrap commands depends upon the assortment of disk drives you have and the settings of the switches inside your drives and microcomputer.

BOOTSTRAP COMMAND ENTRY	H8 LED DISPLAY	DEVICE TYPE ACCESSED	DRIVE SLOT ACCESSED
1	Pri H17	primary	slot in H/Z-89; or left-hand slot of H/Z-87, H-77, or H-17-3
	Pri H47	primary	left-hand slot of H/Z-47 (8-inch drives)
2	Sec H17	secondary	slot in H/Z-89; or left-hand slot of H/Z-87, H-77, or H-17-3
	Sec H47	secondary	left-hand slot of H/Z-47 (8-inch drives)

Table C-2
Alternative Bootstrap Commands for H-8

ENABLING THE AUTOMATIC BOOT

FOR THE H/Z-89

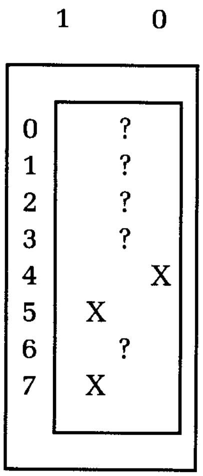
The H/Z-89 has the capability to automatically boot from the primary device. The feature makes it possible to simply turn on (or reset) the machine and have the CP/M system automatically come up.

CAUTION: Never have the disks inserted with the drive doors closed when the power is turned on or off. Current surges that occur when power is initially turned on (and when a unit is turned off) could magnetically damage the disk as well as any data that is stored there.

To enable the auto boot, you will have to make a minor change to the hardware. First, be certain that the computer and all of its peripheral devices are turned off and unplugged from the wall outlets. Carefully release the top of the computer cabinet by inserting a small screwdriver blade into the slot in each side of the unit. Slide the latch plates toward either the front or the back of the screen and carefully tilt the cabinet shell back.

The auto boot feature is enabled by changing one DIP switch setting on the CPU board. The DIP switch is light blue, and is located in the lower right quadrant of the CPU board. (The switch SW-501, is immediately to the right of the resistors RP508 and RP509.) Change only the position of the lowest switch, which is labeled "7", by sliding the small white switch to the left. You can do this by using the tip of a pen or a small screwdriver. Refer to Figure C-1 for a graphic representation of the proper position of the "7" switch. (Some of the other number switches are marked with a question mark because their positions are different depending on the user's combination of hardware.)

After setting the "7" switch, close the cover of the machine, by reversing the steps you used to open the cover. Plug the units in and turn on the power. Place a bootable disk into the drive, close the door, and the system goes directly into CP/M without the user performing the reset and bootstrap steps.



	1	0	
0		?	Device Type (port 174Q)
1		?	Device Type (port 174Q)
2		?	Device Type (port 174Q)
3		?	Device Type (port 170Q)
4		X	1 = Secondary (174Q) 0 = Primary(170Q)
5	X		1 = No Memory Test 0 = Memory Test
6		?	Baud Rate
7	X		1 = Auto Boot 0 = Not Auto Boot

Figure C-1:
The SW-501 Switch,
Set for Automatic Bootstrap
To the Primary Device

FOR THE H8

The Extended Configuration Option board (HA8-8) in the H8 has a single set of eight switches labeled 0-7. To enable the automatic boot function, change the position of switch 7.

Precautionary Notes:

Remember that if your hardware configuration includes a Z87, make sure that it is always powered up whenever you use the computer whether it is in immediate use or not.

Disks may get damaged if the disks are placed in the drives with the doors closed prior to powering up or down. Before using your system either wait for CP/M to display the system prompt before you insert your disks, or leave the drive doors open until power is applied. Conversely, prior to turning the power off, either remove all disks from your drives or open all the drive doors.

Appendix D

Basic Input/Output System (BIOS) Organization

The Heath/Zenith 2.2.04 release of CP/M includes several features to enhance the flexibility of the original CP/M system. As new hardware devices are introduced, they will be supported under this operating system. As a result, the BIOS (Basic Input Output System) occupies more than the amount of space provided for that purpose by Digital Research.

Therefore, Heath/Zenith has designed a program to fit where the BIOS is normally recorded on the system tracks. This program is capable of using the CP/M file system to open a disk file and read in the complete BIOS. It is referred to as the BIOS Loader, and contains just enough intelligence to read from the disk and do some simple terminal I/O in case of an error.

When the system is cold booted, the monitor ROM reads the first sectors of the disk in the selected drive into memory and then control is transferred to that program. The cold start loader contained in the first half of those sectors reads the contents of the first tracks into memory at the appropriate locations for the configured memory size. When that is complete, control is passed to the cold boot entry point of the newly loaded system.

The following BIOS features are explained in this appendix:

- The Heath/Zenith BIOS Loader (BLDR)
- Basic Addresses within the Heath/Zenith BIOS
- BIOS Options
- Algorithm for Disk Deblocking
- Disk Formats
- Page Zero and Clock Organization

THE HEATH/ZENITH BIOS LOADER (BLDR)

With the Heath/Zenith CP/M, instead of being the standard BIOS that is opened, it is the BIOS loader (BLDR) that gets opened, which in turn opens a file called BIOS.SYS and reads it into the appropriate locations in memory.

By using the BLDR concept, the Heath/Zenith implementation overcomes the size limitation imposed by the Digital Research design and at the same time simplifies the otherwise error-prone process of integrating one's own modified BIOS into the system. The penalties paid are slight—the disk from which you cold boot must contain the BIOS.SYS file, and it takes somewhat longer to cold boot because the second file must be opened, read, and relocated in memory.

Note that the BIOS.SYS file is only referenced during the cold boot process, so it only needs to be on the disk used to cold boot your system. The warm boot function of CP/M does not reread the BIOS, and therefore this file does not necessarily have to appear on disks that are used during a warm boot.

The BIOS.SYS file contains the BIOS in a relocatable file format that is compatible with the page relocatable format used in Digital Research's MP/M operating system. This relocatable format is reasonably compact, and storing the BIOS in relocatable form provides the advantage of not requiring relocation of the BIOS during MOVCPMnn.

BASIC ADDRESSES IN THE HEATH/ZENITH BIOS

The Heath/Zenith BIOS implementation is structured so that system utilities can readily access its device tables. This is primarily a requirement of CONFIGUR, which alters some of the values which are used at run-time. Many of these values would be assembly-time constants in the normal implementation of CP/M.

A user program can find the BIOS by using the address at location 0001H as a pointer (the operand of the jmp wboot instruction). That value will be 3 greater than the address of the start of the BIOS.

The first part of the BIOS is the standard Digital Research entry point "jump" table. This table includes jump instructions to each of the 17 entry points into the BIOS.

The "listst" entry point for checking the output status of the list device is implemented in this release. This entry point is used by some background printing spoolers.

Immediately following the standard entries is a one byte BIOS version number (in binary). It is important that programs which alter values in the BIOS in a direct way know exactly which version they are dealing with. For example, CONFIGUR must patch tables at the beginning of the BIOS and therefore needs to be reasonably certain that those tables are where it expects to find them.

Upon cold boot, CP/M will display a sign-on message like the following:

```
nnK HEATH/ZENITH CP/M 2.2.04 09/15/82
FOR Hdd DISKS WITH OPTION(S) cccc
```

Where "nnK" is the amount of Random Access Memory space that the currently loaded CP/M Operating System will occupy; and where "2.2.04" is the version number.

The next byte is the mask used for the serial printer ready status check.

The next byte is used as the first mode byte for the BIOS. Six of the bits are assigned; the remainder are reserved. Bit 7 causes the CCP to attempt to execute the automatic command line on the default disk every time the disk is cold booted. Bit 6 performs the same function for each warm boot. Bit 3 contains the serial printer ready flag, 0 = low and 1 = high. Bit 2 supports the partitioning of the Z-67 Winchester Disk. Bit l enables the printing of disk error messages by the BIOS so the user will receive extended error-status messages. Bit 0 is set when the BIOS is configured for use with a H8-5 serial card for,the console port.

The next byte is used as the second mode byte for the BIOS. Bit 2 contains the Parallel Printer Ready flag, 0 = low and 1 = high. Bit 1 indicates which printer to use as the LPT: device if a Z89-11 interface card is installed, 0 = serial and 1 = parallel. Bit 0 is set when at cold boot it is determined that a Z89-11 interface card is installed.

The next several bytes are used as character I/O device control structures. These specify the port number, baud rate, number of nulls required after a carriage return, and upper case mapping flag for each of the serial character I/O devices. The data is arranged as port number, followed by the baud rate divisor word.

The high order nibble of the baud rate divisor forces all output to upper case if the MSB is 1, and the least significant 3 bits of that nibble are the number of nulls sent after a carriage return to that device. There is one of these structures for each device.

Then two words (four bytes) are used as a soft error counter for the 5.25-inch disk drives. These words are incremented after each soft (recoverable) error. By using DDT to examine this value you can get an indication of the performance of your disk system.

The next byte is set for the maximum number of disks that this BIOS may be configured for. This allows programs to determine the length of the disk tables and to access only those tables that really exist.

This is followed by the disk parameter base and the disk parameter entries for each of the disk drives. In addition to the 16 bytes per disk assigned by Digital Research, the Heath/Zenith implementation has added 8 bytes to each entry.

The first of these 8 is a multi-purpose byte that is used to identify the drive type and its port address. The three most significant bits indicate the drive type. The second byte is the select code necessary for this drive. The next byte is the number of 128 byte records per physical sector. It is followed by a byte indicating the number of 128 byte records per allocation unit. The next four bytes are used for all drive controllers except the H/Z-47.

For the 5.25-inch disk drives the first of these indicates the current track position of the head for this drive. The next byte controls the step rate. When set, the most significant bit of that byte indicates that the drive will be up to speed in 250 mS instead of the normal 1000 mS. The next byte contains flags. The last byte is reserved for use during the physical to logical drive mapping process.

For the Z-67, these bytes are used a bit differently. The first word (two bytes) is the logical sector number of track 0 in a partition. The last word is the logical sector number used in upper bound limit checks.

In addition to these tables, it is necessary for some system utilities to have access to some of the values used for disk control. These values have been located in BIOS and in page 0 of memory. These values control the motor and select timeouts as well as the head settle and write-gate step delays.

At 000DH the current value of the H/Z-89 control latch at port 362Q is stored. This port controls the presence of RAM at zero and enables/resets the clock. In the H8, the clock enable and RAM at zero are split between two different ports.

The H/Z-89 contains special hardware that traps input and output instructions to some of the H8 control ports and generates a nonmaskable interrupt (NMI). Since the Z80 NMI vector falls within the RAM allocated for the default file control block (FCB), it is not possible to guard it with a return instruction or to install a jump to a user routine. Therefore, care must be taken that no inputs or outputs to the H8 control ports (360Q, 361Q, 372Q, 373Q) are attempted in the H/Z-89. Location 000EH is used for the value required to reset the clock on the H8, or 0 if on an H/Z-89. Location 000FH is reserved for the contents of the H17/H77/Z87 control port. The 16 bytes located at 0040-004FH are reserved.

An effort will be made to keep these values in their same relative positions. However, it is possible that between this release and future releases of CP/M from Heath/Zenith there will be some redefinition of the BIOS structure. There is a diminishing probability of change with each release thereafter. Software that accesses these tables should be written using an easily redefinable template to simplify adaptation to such changes.

BIOS OPTIONS

When CP/M is cold booted, it displays a message in the following form:

```
nnK HEATH/ZENITH CP/M 2.2.04 09/15/82
FOR {type} DISKS WITH OPTION(S) {code}
```

The "code" will show the options that are currently selected active within the BIOS.

The options are as follows:

P	The Z-67 is Partitioned
P2	BIOS supports two Z-67 Partitions
I	CRT Input is Interrupt Driven
B	CRT "BREAK" key's Interrupt is available (causes an immediate warm boot)
E3	BIOS supports Extended Double Density on Z-37 Disk Drive
E4	BIOS supports Extended Double Density on H/Z-47 Disk Drive
T	Time of Day Clock
E	Event Counter

ALGORITHM FOR DISK DEBLOCKING

The Heath/Zenith BIOS uses a deblocking algorithm which transparently converts the sectors stored on the 5.25-inch, the 8-inch disks, and on the Winchester Disk to the 128 byte logical records used by CP/M. The BIOS uses information provided by the BDOS to minimize the number of physical read and write operations.

When accessing the disk, the deblocker keeps track of the logical records that are currently buffered within the BIOS. If the BIOS is processing a read call, it is possible for it to retrieve the information from its buffers without any disk activity if the proper sector is already in memory as a by-product of a previous operation.

The buffering that occurs during the write operation means that records can be buffered in memory until the next disk operation. The buffer is only written to the disk if the calling program accesses a logical record that is not contained in the current physical sector, the buffer overflows, or the write is flagged as a "directory access."

One important ramification of this "write-behind" is that programs that do direct BIOS calls to perform track/sector disk I/O should make certain that the write buffer is emptied before the disk is removed or changed, or a cold or warm boot is performed. This can be done by specifying that the last write operation should be performed immediately without write-behind as the BDOS does when it accesses the disk directory.

DISK FORMATS

5.25-INCH HARD-SECTOR DISK FORMAT

The logical records are numbered from 1 to 20 on each track, and are stored in physical sectors 0 thru 9. The tracks of the disk are numbered 0 to 39. The CP/M system reserves the first three tracks for the Cold Start Loader, the CCP, the BDOS, and the BIOS Loader. The CP/M directory begins on track 3, and the usable space begins immediately following it.

8-INCH AND 5.25-INCH SOFT-SECTOR DISK FORMATS

The Heath/Zenith implementation of the BIOS transparently supports a mixture of single, double, and extended double density disks. In addition, each of the densities can be recorded on either single or double-sided media. The first time a disk is "logged on" after a disk reset function (such as a warm boot), the density and whether the disk is double or single sided is checked and used to dynamically modify the disk parameter tables for the particular drive. The most important ramification of this is that disks of different densities or number of recordable sides should be logged on after being interchanged.

The Heath/Zenith configuration offers the user three densities available on the 8-inch disk. Single density is comprised of 26 sectors per track with 128 bytes per sector. Double density contains 26 sectors per track with 256 bytes per sector. Extended density uses 8 sectors per track with 1024 bytes per sector. Deblocking is used on double and extended densities to make each track appear to have 52 and 64 sectors respectively.

The 5.25-inch soft-sectored disks have two possible densities. Single density offers 10 sectors per track with 256 bytes per sector. Double density has 16 sectors of 256 bytes each per track.

PAGE ZERO AND CLOCK ORGANIZATION

```
0000H               jmp    BIOS+3          ;wboot entry use value
                                           ; at 0001H as a pointer
                                           ; to the BIOS to access
                                           ; jump vectors and tables
0003H               db     IOBYTE          ;current IOBYTE
0004H               db     DEFAULT$DISK    ;current default disk
0005H               jmp    BDOS+6          ;BDOS entry point use value
                                           ; at 0006H to find the top
                                           ; of user memory (subtract
                                           ; CCP size if it is to be
                                           ; preserved)
0008H               jmp    clock           ;clock interrupt service
                                           ; vector
                    dw     TICCNT          ;1 work counter incremented
                                           ; every 2 mS interrupts are
                                           ; enabled and clock is on
000DH               db     H88CTL          ;current contents of H88/Z89
                                           ; control latch (362Q)
000EH               db     H8CTL           ;current contents of H8
                                           ; control latch (360Q), or
                                           ; zero if on H/Z-89
000FH               db     DEVCTL          ;current contents of H17/H77
                                           ; Z87 control latch (177Q)
0018H               jmp    CRTINT          ;CRT input interrupt service
                                           ; vector
0020H               jmp    H37INT          ;H37 interrupt service vector
0023H   dlymo37     db     0               ;H37 motor turn-off delay
                                           ; counter
0024H   dlyh37      db     0               ;H37 deselect delay counter
0025H   h37ctl      db     0               ;H37 control register image
0026H   h37iret     dw     0               ;where to go after H37
                                           ; interrupt
                                           ;
;using the value at location 0009H as a pointer to the clock routine,
; the disk (H17/H77/Z87) timers can be accessed, they occupy the three
;bytes immediately preceding the clock interrupt
;
        dlymo:      db     0               ;motor timer decremented
                                           ; every 512 mS until 0
        dlyh:       db     0               ;head load timer decremented
                                           ; every 512 mS until 0
        dlyw:       db     0               ;head settle timer
                                           ; decremented every 4 mS
                                           ; until 0
        clock:      ...                    ;clock interrupt service
                    ...                    ; routine, located in BIOS
                    ei
                    ret
```

Index

TECHNICAL INFORMATION EXPLANATION

The technical information portion of your CP/M documentation package is intended for users who wish to modify the CP/M Operating System or examine the internal components of the system.

Knowledge of this information is not necessary for most CP/M users who only wish to use CP/M resident commands and to run utilities and application programs with the CP/M system.

This technical information consists of the following documentation items bound within three booklets:

- CP/M 2 System Interface: Chapter 5
- CP/M 2 Alteration: Chapter 6
- Appendix A: The MDS Basic I/O System (BIOS)
- Appendix B: A Skeletal CBIOS
- Appendix C: A Skeletal GETSYS/PUTSYS Program
- Appendix D: The MDS-800 Cold Start Loader for CP/M 2
- Appendix E: A Skeletal Cold Start Loader
- Appendix F: CP/M Disk Definition Library
- Appendix G: Blocking and Deblocking Algorithms.

This technical information was written by Digital Research Corporation, the original producers of CP/M version 2. The source listings in the appendices are for illustration purposes only and do not correspond to the modules actually supplied with your system.

The CP/M software products you have purchased are modifications of CP/M version 2. In these modifications, the Basic Input/Output System (BIOS) has been customized for your hardware by Heath and Zenith Data Systems.