CP/M® VERSION 2.2.04
MANUAL

Volume II

HEATH | ZENITH data systems

First Printing
Printed in the
United States of America

595-2816

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.

HEATH COMPANY
BENTON HARBOR, MICHIGAN 49022

ZENITH DATA SYSTEMS
SAINT 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 wishes 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 SPECIFICATIONS

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 chooses 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/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 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 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.

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.

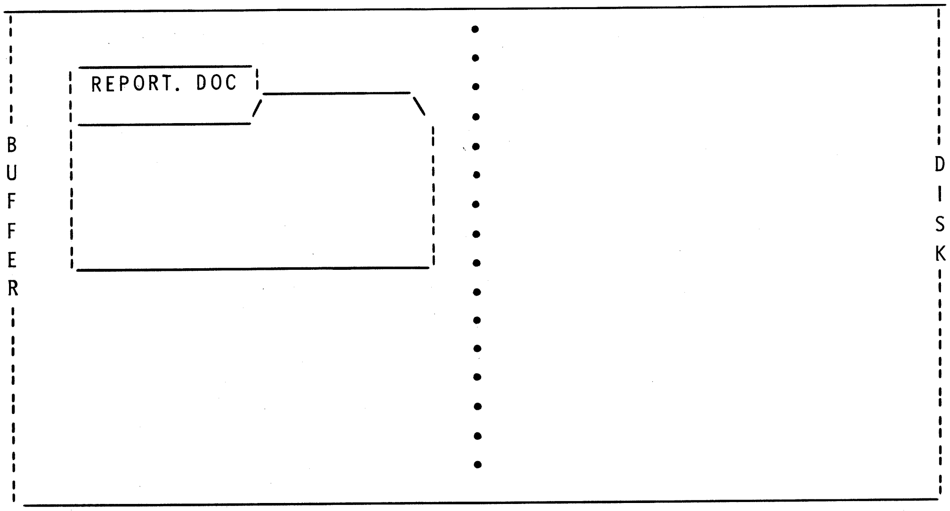

Figure 2-1

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.

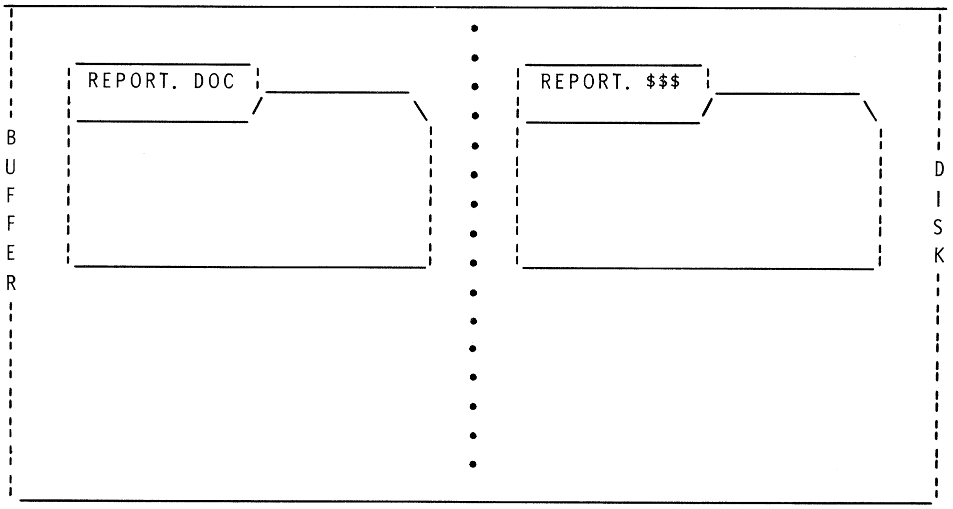

Figure 2-2

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.

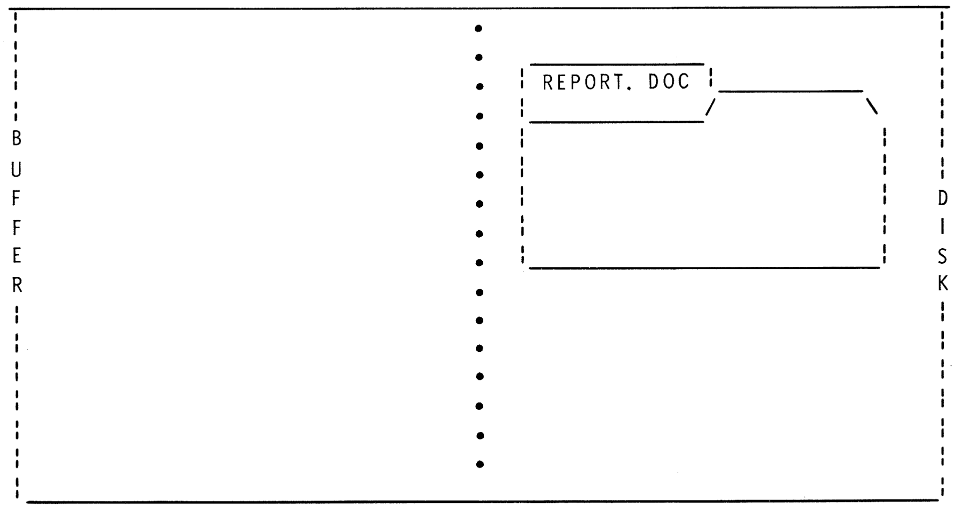

Figure 2-3

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.

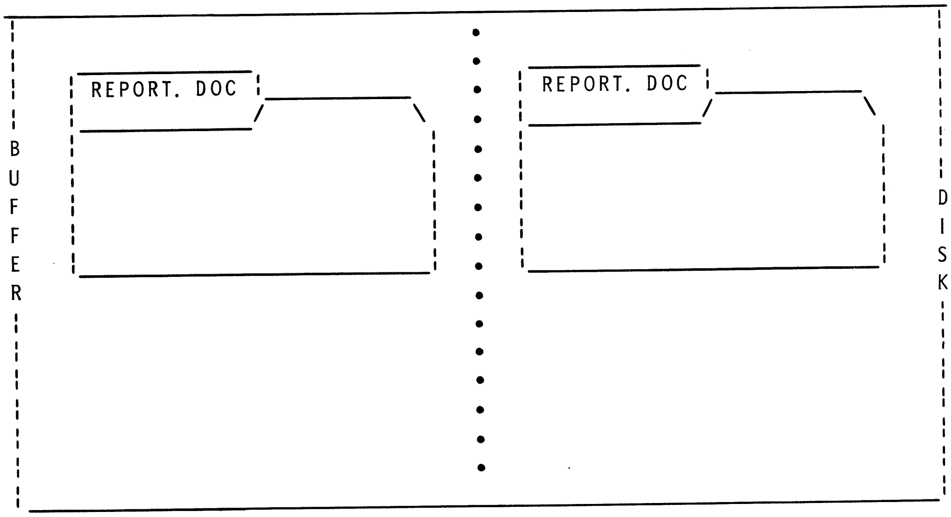

Figure 2-4

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.$$$".

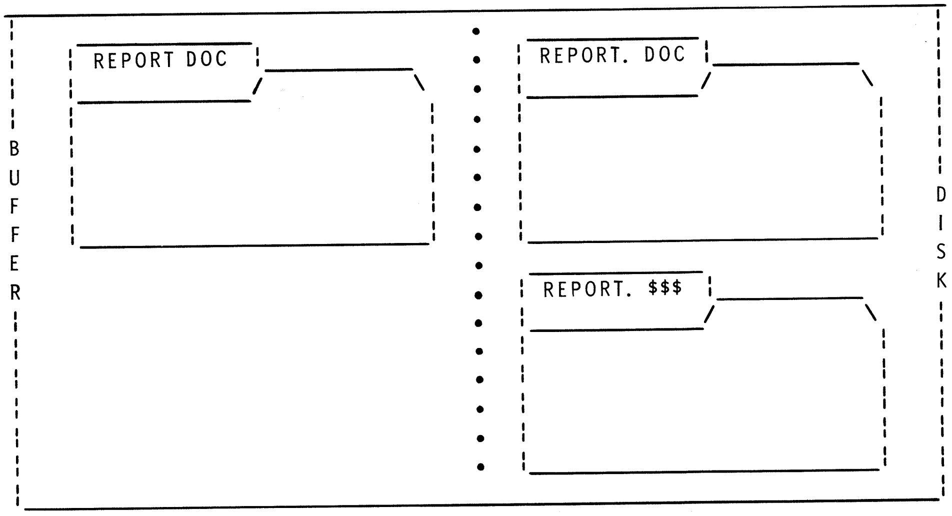

Figure 2-5

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.

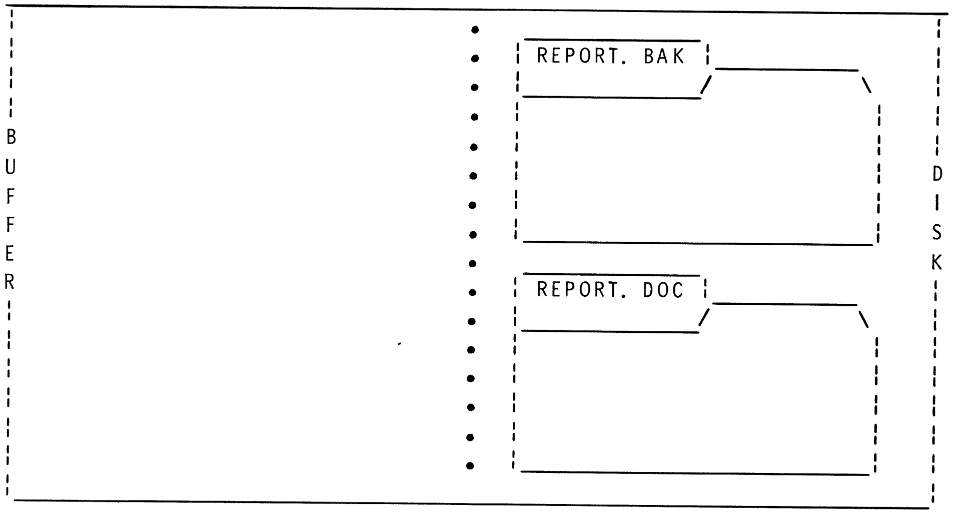

Figure 2-6

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.