

# HEWLETT-PACKARD



**HP-UX Technical BASIC  
Reference Manual  
Volume 2**

---

**HP-UX Technical BASIC  
Reference Manual  
Volume 2**

---



HEWLETT  
PACKARD

**Edition 1 October 1984**



---

## Notice

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

© Copyright 1984, Hewlett-Packard Company.

This document contains proprietary information, which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of Hewlett-Packard Company. The information contained in this document is subject to change without notice.

**Restricted Rights Legend.** Use, duplication, or disclosure by the Government is subject to restrictions as set forth in paragraph (b)(3)(B) of the Rights in Technical Data and Software clause in DAR 7-104.9(a).

© Copyright 1979, 1980, 1983, The Regents of the University of California.

This software and documentation is based in part on the Fourth Berkeley Software Distribution under license from The Regents of the University of California.

© Copyright 1980, 1984, AT&T Technologies. All Rights Reserved.

**Portable Computer Division**  
1000 N.E. Circle Blvd.  
Corvallis, OR 97330, U.S.A.

---

**Printing History**    Edition 1    October 1984    Volume 2    Mfg. No. 82860-90003

## Volume 1

# Contents

---

## Chapter 1

### Introduction

- 1-1** How to Use This Manual
- 1-2** Using the Keyword Dictionary
- 1-2** Legal Usage Table
- 1-2** The Syntax Diagram
- 1-3** Table of Parameters
- 1-3** Spaces
- 1-4** Line Length
- 1-5** Variables
- 1-6** Line Numbers and Line Labels
- 1-6** Comments
- 1-6** Multi-statement Lines
- 1-7** Hardware Dependencies
- 1-7** BASIC Files
- 1-7** File Structure
- 1-10** File Types
- 1-10** File Security
- 1-10** The Configuration File
- 1-11** The BASIC Metacharacter
- 1-12** BASIC Function Keys

---

## Chapter 2

### Keyword Dictionary

## Volume 2

---

### Chapter 3

#### Glossary

---

### Chapter 4

#### Reference Tables

|             |  |
|-------------|--|
| <b>4-1</b>  | Math Hierarchy                                 |
| <b>4-1</b>  | String Hierarchy                               |
| <b>4-2</b>  | US ASCII ASCII Character Set                   |
| <b>4-5</b>  | Roman Extension Character Set                  |
| <b>4-8</b>  | Reset Conditions                               |
| <b>4-10</b> | Boundaries and Scaling                         |
| <b>4-11</b> | Reflecting Plots With LIMIT, LOCATE, and SCALE |
| <b>4-11</b> | Pen Up/Dpwn Status                             |
| <b>4-12</b> | Pen Control With PLOT, IPLOT, and RPLOT        |
| <b>4-12</b> | Graphics Display Pen Numbers                   |
| <b>4-12</b> | Pen Numbers With GCLEAR                        |
| <b>4-13</b> | Branch Precedence Table                        |
| <b>4-14</b> | HP-IB Single Line Messages                     |
| <b>4-15</b> | HP-IB Multiple-Line Messages                   |

---

### Chapter 5

#### I/O Registers

|             |                      |
|-------------|----------------------|
| <b>5-1</b>  | I/O Buffer Registers |
| <b>5-2</b>  | HP-IB Interface      |
| <b>5-6</b>  | Serial Interface     |
| <b>5-12</b> | BCD Interface        |
| <b>5-17</b> | GPIO Interface       |
| <b>5-21</b> | HP-IL Interface      |

---

### Chapter 6

#### Errors

---

## Chapter 7

### Keyword Summary

|             |   |
|-------------|---|
| <b>7-1</b>  | General Math Functions and Operators      |
| <b>7-3</b>  | Binary Functions                          |
| <b>7-4</b>  | String Operations                         |
| <b>7-5</b>  | Clock and Time Functions                  |
| <b>7-5</b>  | Program Entry and Editing                 |
| <b>7-6</b>  | Debugging                                 |
| <b>7-6</b>  | Variable Allocation                       |
| <b>7-7</b>  | Display Control                           |
| <b>7-8</b>  | Program Control                           |
| <b>7-10</b> | Subprogram Control                        |
| <b>7-10</b> | Mass Storage                              |
| <b>7-11</b> | Graphics Boundaries, Scaling, and Control |
| <b>7-12</b> | Graphics Plotting                         |
| <b>7-13</b> | Graphics Labeling                         |
| <b>7-14</b> | Event-Initiated Branching                 |
| <b>7-15</b> | Input/Output                              |
| <b>7-16</b> | Numeric Array Functions                   |
| <b>7-17</b> | Numeric Array Operations                  |

# 3

## Glossary

---

### A

**alpha display:** The portion of the display that receives alphanumeric information, including errors, warnings, characters entered from the keyboard, and output from statements such as CAT, DIRECTORY, XREF, SCAN, DISP and LIST.

**angle mode:** The current units used for interpreting angles—degrees, radians, or grads. The angle mode is changed by executing DEG, RAD, or GRAD.

**argument:** The parameter of a function.

**arithmetic operators:** +, -, \*, /, ^, \ (or DIV), MOD.

**array:** A numeric or string variable that has been defined explicitly or implicitly to have one or two dimensions. An array is explicitly dimensioned when memory is reserved for it by a DIM, REAL, SHORT, INTEGER, or COM statement. Each item in an array is called an *element*. A numeric array can be dimensioned so that all its elements have REAL, SHORT, or INTEGER precision. A string array is dimensioned such that all its elements have the same maximum string length.

When an array is dimensioned, the number of elements in the array is defined by specifying the array *upper bound*. The *lower bound* is defined by the program option base (default=0). An array is implicitly dimensioned when a program references an array element before the array has been explicitly dimensioned. Implicitly dimensioned arrays have an upper bound equal to 10.

**assignment statement:** A statement in which a value is assigned to a variable.



---

**B**

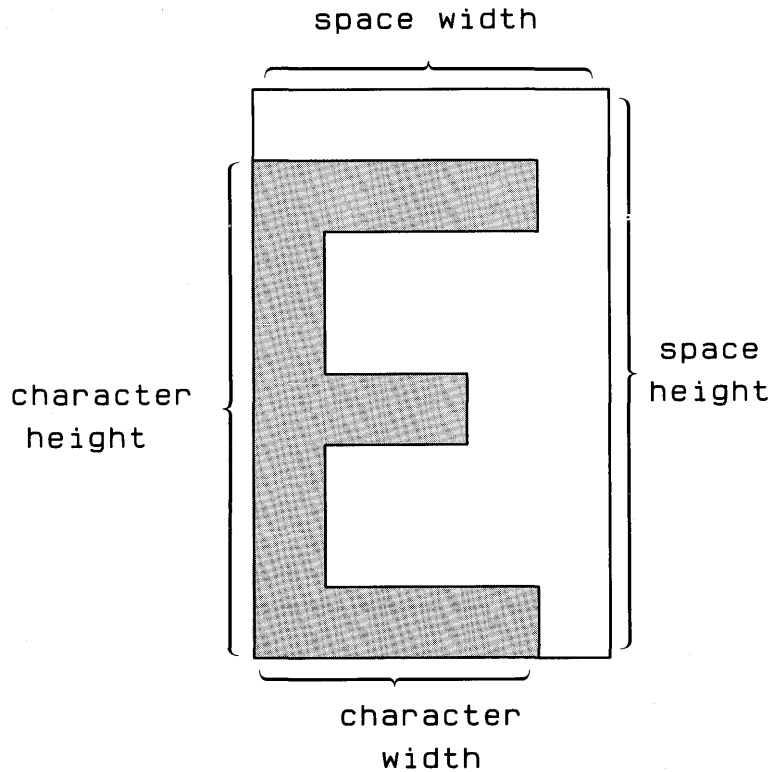
**bit:** A single digit in base 2 that must be 1 or 0.

**byte:** A group of eight bits. A byte evaluates to a decimal number in the range 0 through 255.

---

**C**

**character space:** The area occupied by both a character and the space surrounding the character that separates it horizontally and vertically from other characters.



**clip:** Establishing plotting boundaries. Plotting boundaries restrict the plotting area to a portion of the total area available (the physical limits or graphics limits specified by `LIMIT`). Restrictions imposed by the plotting boundaries do not affect the logical pen position. Pen position is set by the current scale, and may lie outside the plotting boundaries. However, no lines, axes, or grids are drawn outside the plotting boundaries.

**concatenate:** To join together two or more strings.

**conditional branching:** A form of branching that occurs only when a specified condition or set of conditions is met.

**control characters:** Characters interpreted by devices as instructions.

**current working directory:** The current working directory is accessed by mass storage statements when a file name is used alone, rather than as part of part of an HP-UX path name. The current working directory can be changed by executing the `MASS STORAGE IS` statement.

---

## D

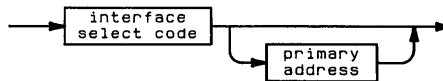
**data pointer:** An internal mechanism used to indicate the next `DATA` statement item to be read.

**default:** The action taken or value used unless otherwise specified. The system "wakes up" using certain default values. In addition, many BASIC statements have optional parameters which use default values when no parameter is specified.

**default graphics conditions:** See `graphics default conditions`.

**device name:**\* The name assigned to each disc drive unit by the operating system at power on, used as part of an HP-UX path name. The device names are listed in the /dev directory.

**device selector:** A numeric expression used to designate the device or interface acting as source or destination of data in an input/output operation. The device selector consists of a valid interface select code or a valid combination of interface select code and primary address.



The interface select code is a one- or two-digit integer in the range 0 through 10. Select codes 0 through 2 are reserved for internal devices. The primary address is a two-digit number in the range 00 through 31.

---

## E

**end-of-line (EOL) sequence:** The sequence of characters sent to an output device at the end of a PRINT, DISP, LABEL, or OUTPUT list. The default EOL is carriage return/line feed.

**escape sequence:** A sequence of characters beginning with the escape (ESC) character, CHR\$(27).

\* Machine-dependent parameter. Device names may be used by single-user systems with removable file structures.

---

## F

**file name:** A sequence of 1 to 14 characters used to identify a particular file. Any keyboard character can be used except slash and leading colon. The file name is entered into the directory in which the file is located.

**file pointer:** A mechanism used to indicate where in an open data file the next item of data will be read or printed.

**file selector:** An integer in the range 11 through 20, used to assign an I/O path to a file.

**flag:** A bit that can be individually set (1), cleared (0), and read. Individual flags are set using the `SFLAG` statement, and cleared using the `CFLAG` statement. The `SFLAG` statement can also be used to set and/or clear up to the entire 64 flags at a time.

**flat file structure:** A file structure in which each disc has only one directory—the top-level directory.

**free field format:** An output format used with simple (without `USING`) `DISP`, `PRINT`, and `OUTPUT` statements in which items are output left-justified in a field of 11, 21, or 32 columns. Free field format is specified by separating items by commas.

**function:** A procedural call that returns a value. The call can be to a user-defined function or to a function provided by BASIC. The value returned by the function replaces the function call as the expression containing the function is evaluated.

---

## G

**global declarations:** Declarations and system status parameters that are unaffected by switching between the main BASIC program and subprograms. All non-local declarations are global. (See **local declarations**.)

**graphics area:** The area enclosed by the graphics limits. No plotting or labeling can occur outside this area.

**graphics default conditions:** The graphics default conditions are activated at power-on, at reset, and whenever a PLOTTER IS or LIMIT statement is executed. The default conditions are:

- Plotting boundaries (set by CLIP and LOCATE) are set to the graphics limits.
- The plotting area is scaled in graphics units (GU's), the default scale.
- The computer is set to user units mode with user units (UU's) equal to graphics units (GU's).
- Pen color is set to pen 1.
- Lines are drawn using line type 1.
- Labels are drawn using the default character size.
- Labels are positioned as LORG 1.
- Labeling direction is left-to-right (LDIR 0).
- The logical pen moves to the origin (lower left corner).

**graphics display:** The portion of display memory used as destination for graphics output.

**graphics limits:** The area of a plotting device beyond which no plotting or labeling can occur. Graphics limits can be set manually and read by the PLOTTER IS statement, or they can be set by LIMIT.

**graphics units scale:**  $\frac{1}{100}$  of the shortest axis on the plotting device. Graphic units scaling is active at reset and whenever PLOTTER IS, LIMIT, or SETGU is executed.

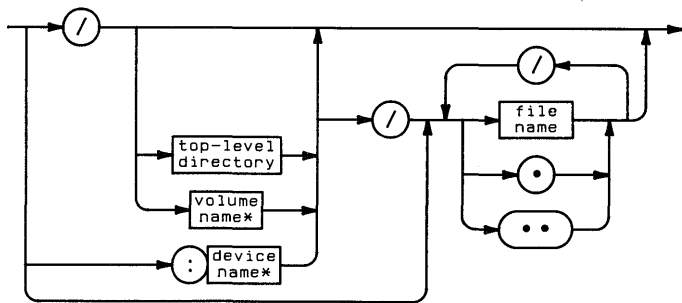
---

## H

**hard clip limits:** The physical limits of a plotting device.

**hierarchy:** The precedence system that determines the order in which operations are performed in numeric and string expressions. Operations with the highest precedence are performed first. Multiple operations with the same precedence are performed from left to right. Refer to **Reference Tables** for tables of the math and string hierarchies.

**HP-UX path name:** The complete name of a file. The pathname starts at the root directory (absolute path name) or at the current working directory (relative path name) and traces the path leading to the file through the file hierarchy.



\* Implemented for certain single-user systems with removable file systems.

---

**interface select code:** An integer in the range 0 through 10, used to provide an I/O path to an interface or internal device. Numbers 0 through 2 are reserved for internal devices.

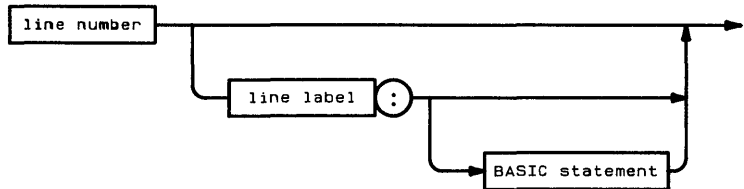
**interrupt:** An interruption to normal program execution caused by a particular event. Event-initiated interrupts include ON KEY#, ON KYBD, ON ERROR, ON INTR, ON TIMER, and ON TIMEOUT branching.

---

**Julian Day number:** An astronomical convention representing the number of days since January 1, 4713 B.C.

## L

**line label:** A character string up to 31 characters long used to identify a program line. The label can contain letters, numbers, and the underscore character; the first character must be a letter. A colon separates the line label from the BASIC statement it identifies.



**literal:** A string constant containing characters entered from the keyboard, including the metacharacter, ~.

**local declarations**—declarations and system status parameters that are in affect only within the main program or subprogram in which they are declared. The local declarations are:

```
OFF EOT      ↔  ON EOT
OFF ERROR   ↔  ON ERROR
OFF INTR    ↔  ON INTR
OFF KEY#     ↔  ON KEY#
OFF KEYBD   ↔  ON KEYBD
OFF TIMEOUT ↔  ON TIMEOUT
OFF TIMER    ↔  ON TIMER#
TRACE  TRACE VAR  TRACE ALL
```

**logical expression:** A numeric expression that evaluates to 1 (true) or 0 (false). Logical expressions may contain relational (= < > <= >= #) and logical (AND, OR, NOT, EXOR) operators.

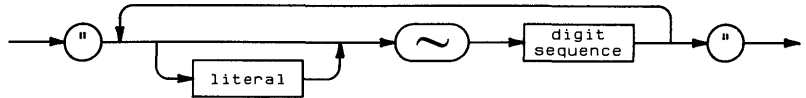
**logical pen:** The position of the plotting pen as specified in a plotting statement. The logical pen position is different from the actual pen position when a plotting statement specifies coordinates outside the plotting boundaries or graphics limits.

---

# M

**matrix:** A two-dimensional array.

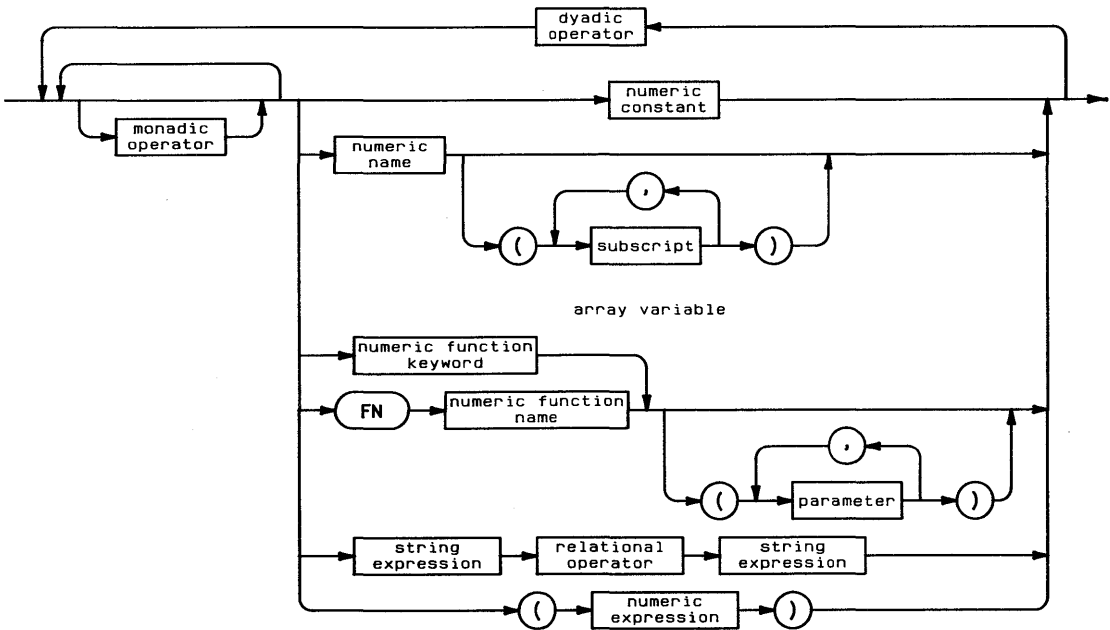
**metacharacter:** A character (~) used within a literal to indicate that the next character or group of characters has special significance.



---

# N

**numeric expression:** An expression that evaluates to a numeric result.





| Item                     | Description   |
|--------------------------|---|
| monadic operator         | An operator that performs its operation on the expression immediately to its right: +, -, NOT.  |
| dyadic operator          | An operator that performs its operation on the two expressions it is between: ^, *, /, \, MOD, DIV, +, -, =, <, >, #, <=, >=, AND, OR, EXOR.  |
| numeric constant         | A numeric quantity whose value is expressed using numerals and optional decimal point and exponent.   |
| numeric name             | The name of a numeric variable.   |
| subscript                | A numeric expression used to reference an element of an array.  |
| numeric function keyword | A BASIC keyword that invokes a function, returning a numeric value.   |
| numeric function name    | The name of a user-defined function that returns a numeric value.   |
| parameter                | A numeric or string expression that is passed to a function.  |
| relational operator      | An operator which returns a 1 (true) or 0 (false) based on the results of a relational test of the operands it separates: =, <, >, #, <=, >=. |

---

**O** **option base:** The explicit or implied lower bound of all arrays in a program. The default option base is 0.

---

**P** **path name:** See **HP-UX path name**.

**plotting area:** The area, designated by CLIP or LOCATE, in which lines and axes may be drawn.

**plotting boundaries:** Boundaries of the plotting area. Labels may be placed outside the plotting boundaries; however, they must be within the graphics limits. The plotting area specified by `CLIP` and `LOCATE` can be entirely within the graphics limits, or it can extend outside the graphics limits or physical limits of the device. However, no plotting or labeling is permitted outside the graphics limits. Plotting boundaries are in effect when the computer is in users units (UU's) mode. The plotting boundaries are set equal to the graphics limits when the computer is set to graphics units (GU's) mode.

**prerun error:** An error occurring in the context of a program, such as referencing a non-existent line, duplicate user-defined functions, and illegal array dimensions.

**primary address:** A number used in conjunction with the interface select code to define the device selector. On certain interfaces, the primary address describes the location of a device on an interface. On other interfaces, the primary address describes certain characteristics of the device. The primary address must be a two-digit number in the range 00 through 31.

**print-all mode:** An output mode, enabled by executing the `PRINT ALL` statement, in which all displayed alphanumeric output is also sent to the `PRINTER IS` printer. Print-all mode is canceled by executing `NORMAL`.

---

## R

**relational expression:** An expression consisting of two numeric expressions or two string expressions separated by a relational operator. A relational expression evaluates to true (1) or false (0).

**relational operator:** `=`, `>`, `<`, `<>`, `#`, `<=`, or `>=`

---

## S

**simple variable:** A variable in which one value can be stored; a non-array variable.

**standard number format:** The format used to output numbers when no other format is specified. Numbers are output as follows:

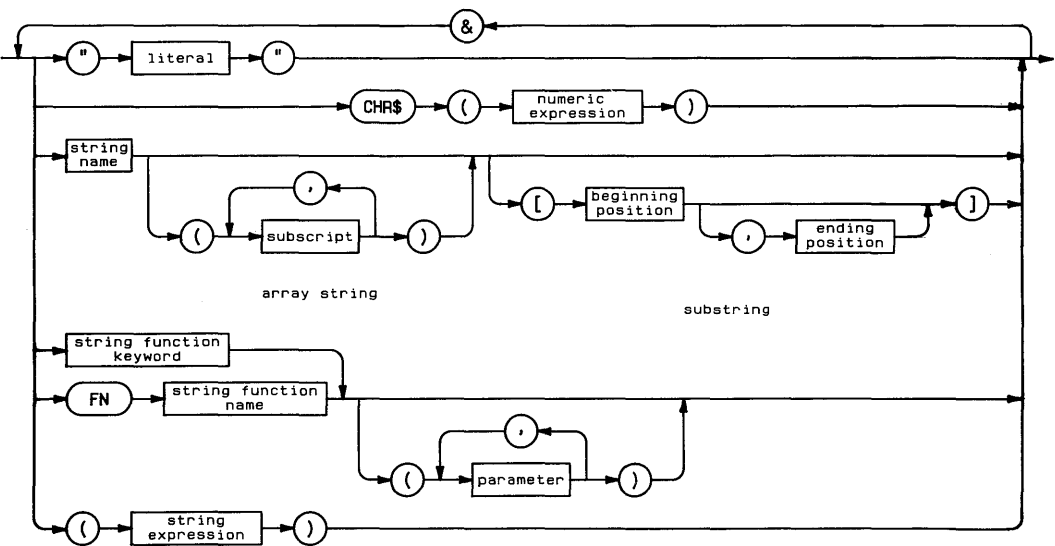
- All significant digits of a number are output.
- Excess zeros to the right of the decimal point are suppressed.
- Leading zeros to the left of the decimal point are truncated.
- Numbers whose absolute values are greater than or equal to 1 are output with no exponents if they can be represented precisely in the number of digit places available.\*
- Numbers between  $-1$  and  $1$  are output showing all significant digits and no exponent if they can be represented precisely in the number of decimal places available.\*
- All other numbers are expressed in scientific notation with a mantissa<sup>†</sup> in the range 1 through 10, followed by E, a minus sign if necessary, and the numeric value of the exponent.

\* Machine-dependent value.

† The number of decimal places in the mantissa is machine-dependent.

**string constant:** A data type that may contain literals and concatenated CHR# functions. The first character in the string is in position 1. The length of the string is the current number of characters in the string, excluding the metacharacter (~), and cannot exceed the dimensioned length. If a string is not explicitly dimensioned, it is implicitly dimensioned to 18 characters. When a string is empty, it is called a null string and has a length of zero. A null string can be represented as an empty literal (for example, A\$="") or as a substring in which the ending position is one less than the beginning position (for example, A\$[4,3]).

**string expression:** An expression that evaluates to a string result.



| Item                    | Description   |
|-------------------------|---|
| literal                 | A string constant composed of any character generated from the keyboard.            |
| numeric expression      | —   |
| string name             | The name of a string variable.  |
| subscript               | A numeric expression used to specify an element of an array.                        |
| beginning position      | A numeric expression specifying the position of the first character in a substring. |
| ending position         | A numeric expression specifying the position of the last character in a substring.  |
| string function keyword | A BASIC keyword that invokes a function returning a string value.                   |
| string function name    | The name of a user-defined function that returns a string value.                    |
| parameter               | A numeric or string expression that is passed to a function.                        |

**subprogram:** A program segment that can be detached from the main program and stored in its own subprogram file. When a subprogram is called by a program or other subprogram, the called subprogram is loaded, if necessary, into computer memory at the end of the calling (sub)program and automatically run. Calling a subprogram has no effect on BASIC and binary programs currently in memory.

All subprograms must begin with a `SUB` statement and end with a `SUBEND` or `SUBEXIT` statement. Line numbers and line labels within the subprogram are independent of the main program or other subprograms. For example, both the main program and subprogram can have the same line numbers.

A subprogram is invoked by execution of a `CALL` statement. The `CALL` statement includes an optional list of parameters passed to the subprogram by value or address.

**subscript:** A number that specifies the row or column location of an element of an array.

**substring:** A contiguous series of characters that comprises all or part of a string. If no ending position is specified, the substring includes all characters from the specified beginning position to the end of the string.

**syntax error:** An error returned when attempting to enter an improperly constructed statement or command.

---

## T

**top-level directory:** The highest-level directory on a disc. In the system hierarchy of files, the top-level directory of each mounted disc is located directly beneath the operating system root directory. In single-user systems with removable file systems, the volume name of the disc becomes the top-level directory when the disc is mounted. In a "flat" file structure, the top-level directory is the only directory file on the disc.

**trigonometric mode:** The current units for interpreting angles—degrees, radians, or grads. The trigonometric mode is changed by executing `DEG`, `RAD`, or `GRAD`.

---

## V

**variable name:** A name of a numeric or string variable. All string variable names must end with the character `$`. Names can be up to 32 characters long, and can be any sequence of letters, numbers and the underscore character, except that the first character must be a letter.

**vector:** A one-dimensional array.

**volume name:** A name used by single-user systems with removable file systems to identify a particular disc. A volume name is assigned to the disc when it is formatted. When the disc is mounted by the operating system, the volume name is entered into the root directory as the name of the top-level directory on the disc.



# 4

## Reference Tables

### Math Hierarchy

| Precedence             | Operator  |
|------------------------|---|
| Highest<br>↓<br>Lowest | Parentheses; may be used to force any order of operations<br>Functions; user-defined and BASIC<br>Exponentiation: ^<br>Monadic operators: +, -, NOT<br>Multiplication and division: *, /, MOD, DIV or \<br>Addition and subtraction: +, -<br>Relational operators: =, <, >, <=, >=, # or <><br>AND<br>OR EXOR |

### String Hierarchy

| Precedence             | Operator  |
|------------------------|---|
| Highest<br>↓<br>Lowest | Parentheses<br>Functions (user-defined and BASIC), substring operations<br>Concatenation: & |



## US ASCII Character Set

| ASCII Char. | Equivalent Forms |          |     |     | HP-IB |
|-------------|------------------|----------|-----|-----|-------|
|             | Dec              | Binary   | Oct | Hex |       |
| NUL         | 0                | 00000000 | 000 | 00  |       |
| SOH         | 1                | 00000001 | 001 | 01  | GTL   |
| STX         | 2                | 00000010 | 002 | 02  |       |
| ETX         | 3                | 00000011 | 003 | 03  |       |
| EOT         | 4                | 00000100 | 004 | 04  | SDC   |
| ENQ         | 5                | 00000101 | 005 | 05  |       |
| ACK         | 6                | 00000110 | 006 | 06  |       |
| BEL         | 7                | 00000111 | 007 | 07  |       |
| BS          | 8                | 00001000 | 010 | 08  | GET   |
| HT          | 9                | 00001001 | 011 | 09  | TCT   |
| LF          | 10               | 00001010 | 012 | 0A  |       |
| VT          | 11               | 00001011 | 013 | 0B  |       |
| FF          | 12               | 00001100 | 014 | 0C  |       |
| CR          | 13               | 00001101 | 015 | 0D  |       |
| SO          | 14               | 00001110 | 016 | 0E  |       |
| SI          | 15               | 00001111 | 017 | 0F  |       |
| DLE         | 16               | 00010000 | 020 | 10  |       |
| DC1         | 17               | 00010001 | 021 | 11  | LLO   |
| DC2         | 18               | 00010010 | 022 | 12  |       |
| DC3         | 19               | 00010011 | 023 | 13  |       |
| DC4         | 20               | 00010100 | 024 | 14  | DCL   |
| NAK         | 21               | 00010101 | 025 | 15  |       |

| ASCII Char. | Equivalent Forms |          |     |     | HP-IB |
|-------------|------------------|----------|-----|-----|-------|
|             | Dec              | Binary   | Oct | Hex |       |
| SYNC        | 22               | 00010110 | 026 | 16  |       |
| ETB         | 23               | 00010111 | 027 | 17  |       |
| CAN         | 24               | 00011000 | 030 | 18  | SPE   |
| EM          | 25               | 00011001 | 031 | 19  | SPD   |
| SUB         | 26               | 00011010 | 032 | 1A  |       |
| ESC         | 27               | 00011011 | 033 | 1B  |       |
| FS          | 28               | 00011100 | 034 | 1C  |       |
| GS          | 29               | 00011101 | 035 | 1D  |       |
| RS          | 30               | 00011110 | 036 | 1E  |       |
| US          | 31               | 00011111 | 037 | 1F  |       |
| space       | 32               | 00100000 | 040 | 20  | LA0   |
| !           | 33               | 00100001 | 041 | 21  | LA1   |
| "           | 34               | 00100010 | 042 | 22  | LA2   |
| #           | 35               | 00100011 | 043 | 23  | LA3   |
| \$          | 36               | 00100100 | 044 | 24  | LA4   |
| %           | 37               | 00100101 | 045 | 25  | LA5   |
| &           | 38               | 00100110 | 046 | 26  | LA6   |
| '           | 39               | 00100111 | 047 | 27  | LA7   |
| (           | 40               | 00101000 | 050 | 28  | LA8   |
| )           | 41               | 00101001 | 051 | 29  | LA9   |
| *           | 42               | 00101010 | 052 | 2A  | LA10  |
| +           | 43               | 00101011 | 053 | 2B  | LA11  |

## US ASCII Character Set (continued)

| ASCII Char. | Equivalent Forms |          |     |     | HP-IB |
|-------------|------------------|----------|-----|-----|-------|
|             | Dec              | Binary   | Oct | Hex |       |
| ,           | 44               | 00101100 | 054 | 2C  | LA12  |
| -           | 45               | 00101101 | 055 | 2D  | LA13  |
| .           | 46               | 00101110 | 056 | 2E  | LA14  |
| /           | 47               | 00101111 | 057 | 2F  | LA15  |
| 0           | 48               | 00110000 | 060 | 30  | LA16  |
| 1           | 49               | 00110001 | 061 | 31  | LA17  |
| 2           | 50               | 00110010 | 062 | 32  | LA18  |
| 3           | 51               | 00110011 | 063 | 33  | LA19  |
| 4           | 52               | 00110100 | 064 | 34  | LA20  |
| 5           | 53               | 00110101 | 065 | 35  | LA21  |
| 6           | 54               | 00110110 | 066 | 36  | LA22  |
| 7           | 55               | 00110111 | 067 | 37  | LA23  |
| 8           | 56               | 00111000 | 070 | 38  | LA24  |
| 9           | 57               | 00111001 | 071 | 39  | LA25  |
| :           | 58               | 00111010 | 072 | 3A  | LA26  |
| ;           | 59               | 00111011 | 073 | 3B  | LA27  |
| <           | 60               | 00111100 | 074 | 3C  | LA28  |
| =           | 61               | 00111101 | 075 | 3D  | LA29  |
| >           | 62               | 00111110 | 076 | 3E  | LA30  |
| ?           | 63               | 00111111 | 077 | 3F  | UNL   |
| @           | 64               | 01000000 | 100 | 40  | TA0   |

| ASCII Char. | Equivalent Forms |          |     |     | HP-IB |
|-------------|------------------|----------|-----|-----|-------|
|             | Dec              | Binary   | Oct | Hex |       |
| A           | 65               | 01000001 | 101 | 41  | TA1   |
| B           | 66               | 01000010 | 102 | 42  | TA2   |
| C           | 67               | 01000011 | 103 | 43  | TA3   |
| D           | 68               | 01000100 | 104 | 44  | TA4   |
| E           | 69               | 01000101 | 105 | 45  | TA5   |
| F           | 70               | 01000110 | 106 | 46  | TA6   |
| G           | 71               | 01000111 | 107 | 47  | TA7   |
| H           | 72               | 01001000 | 108 | 48  | TA8   |
| I           | 73               | 01001001 | 109 | 49  | TA9   |
| J           | 74               | 01001010 | 110 | 50  | TA10  |
| K           | 75               | 01001011 | 111 | 51  | TA11  |
| L           | 76               | 01001100 | 114 | 4C  | TA12  |
| M           | 77               | 01001101 | 115 | 4D  | TA13  |
| N           | 78               | 01001110 | 116 | 4E  | TA14  |
| O           | 79               | 01001111 | 117 | 4F  | TA15  |
| P           | 80               | 01010000 | 120 | 50  | TA16  |
| Q           | 81               | 01010001 | 121 | 51  | TA17  |
| R           | 82               | 01010010 | 122 | 52  | TA18  |
| S           | 83               | 01010011 | 123 | 53  | TA19  |
| T           | 84               | 01010100 | 124 | 54  | TA20  |
| U           | 85               | 01010100 | 125 | 55  | TA21  |

## US ASCII Character Set (continued)

| ASCII Char. | Equivalent Forms |          |     |     | HP-IB |
|-------------|------------------|----------|-----|-----|-------|
|             | Dec              | Binary   | Oct | Hex |       |
| V           | 86               | 01010110 | 126 | 56  | TA22  |
| W           | 87               | 01010111 | 127 | 57  | TA23  |
| X           | 88               | 01011000 | 130 | 58  | TA24  |
| Y           | 89               | 01011001 | 131 | 59  | TA25  |
| Z           | 90               | 01011010 | 132 | 5A  | TA26  |
| [           | 91               | 01011011 | 133 | 5B  | TA27  |
| \           | 92               | 01011100 | 134 | 5C  | TA28  |
| ]           | 93               | 01011101 | 135 | 5D  | TA29  |
| ^           | 94               | 01011110 | 136 | 5E  | TA30  |
| _           | 95               | 01011111 | 137 | 5F  | UNT   |
| '           | 96               | 01100000 | 140 | 60  | SC0   |
| a           | 97               | 01100001 | 141 | 61  | SC1   |
| b           | 98               | 01100010 | 142 | 62  | SC2   |
| c           | 99               | 01100011 | 143 | 63  | SC3   |
| d           | 100              | 01100100 | 144 | 64  | SC4   |
| e           | 101              | 01100101 | 145 | 65  | SC5   |
| f           | 102              | 01100110 | 146 | 66  | SC6   |
| g           | 103              | 01100111 | 147 | 67  | SC7   |
| h           | 104              | 01101000 | 150 | 68  | SC8   |
| i           | 105              | 01101001 | 151 | 69  | SC9   |
| j           | 106              | 01101010 | 152 | 6A  | SC10  |

| ASCII Char. | Equivalent Forms |          |     |     | HP-IB |
|-------------|------------------|----------|-----|-----|-------|
|             | Dec              | Binary   | Oct | Hex |       |
| k           | 107              | 01101011 | 153 | 6B  | SC11  |
| l           | 108              | 01101100 | 154 | 6C  | SC12  |
| m           | 109              | 01101101 | 155 | 6D  | SC13  |
| n           | 110              | 01101110 | 156 | 6E  | SC14  |
| o           | 111              | 01101111 | 157 | 6F  | SC15  |
| p           | 112              | 01110000 | 160 | 70  | SC16  |
| q           | 113              | 01110001 | 161 | 71  | SC17  |
| r           | 114              | 01110010 | 162 | 72  | SC18  |
| s           | 115              | 01110011 | 163 | 73  | SC19  |
| t           | 116              | 01110100 | 164 | 74  | SC20  |
| u           | 117              | 01110101 | 165 | 75  | SC21  |
| v           | 118              | 01110110 | 166 | 76  | SC22  |
| w           | 119              | 01110111 | 167 | 77  | SC23  |
| x           | 120              | 01111000 | 170 | 78  | SC24  |
| y           | 121              | 01111001 | 171 | 79  | SC25  |
| z           | 122              | 01111010 | 172 | 7A  | SC26  |
| {           | 123              | 01111011 | 173 | 7B  | SC27  |
|             | 124              | 01111100 | 174 | 7C  | SC28  |
| }           | 125              | 01111101 | 175 | 7D  | SC29  |
| ~           | 126              | 01111110 | 176 | 7E  | SC30  |
| DEL         | 127              | 01111111 | 177 | 7F  | SC31  |

## Roman Extension Character Set

| ASCII Char | Equivalent Forms |          | ASCII Char | Equivalent Forms |          |
|------------|------------------|----------|------------|------------------|----------|
|            | Dec              | Binary   |            | Dec              | Binary   |
|            | 128              | 10000000 |            | 150              | 10010110 |
|            | 129              | 10000001 |            | 151              | 10010111 |
|            | 130              | 10000010 |            | 152              | 10110000 |
|            | 131              | 10000011 |            | 153              | 10011001 |
|            | 132              | 10000100 |            | 154              | 10011010 |
|            | 133              | 10000101 |            | 155              | 10011011 |
|            | 134              | 10000110 |            | 156              | 10011100 |
|            | 135              | 10000111 |            | 157              | 10011101 |
|            | 136              | 10001000 |            | 158              | 10011110 |
|            | 137              | 10001001 |            | 159              | 10011111 |
|            | 138              | 10001010 | space      | 160              | 10100000 |
|            | 139              | 10001011 | À          | 161              | 10100001 |
|            | 140              | 10001100 | Á          | 162              | 10100010 |
|            | 141              | 10001101 | Ê          | 163              | 10100011 |
|            | 142              | 10001110 | Ë          | 164              | 10100100 |
|            | 143              | 10001111 | Ë          | 165              | 10100101 |
|            | 144              | 10010000 | Ï          | 166              | 10100110 |
|            | 145              | 10010001 | Ï          | 167              | 10100111 |
|            | 146              | 10010010 | Ï          | 168              | 10101000 |
|            | 147              | 10010011 | Ï          | 169              | 10101001 |
|            | 148              | 10010100 | Ï          | 170              | 10101010 |
|            | 149              | 10010101 | Ï          | 171              | 10101011 |

## Roman Extension Character Set (continued)

| ASCII Char | Equivalent Forms |          | ASCII Char | Equivalent Forms |          |
|------------|------------------|----------|------------|------------------|----------|
|            | Dec              | Binary   |            | Dec              | Binary   |
| ˘          | 172              | 10101100 | ë          | 193              | 11000001 |
| ò          | 173              | 10101101 | ô          | 194              | 11000010 |
| ó          | 174              | 10101110 | û          | 195              | 11000011 |
| £          | 175              | 10101111 | ä          | 196              | 11000100 |
| -          | 176              | 10110000 | é          | 197              | 11000101 |
|            | 177              | 10110001 | ó          | 198              | 11000110 |
|            | 178              | 10110010 | ü          | 199              | 11000111 |
| ª          | 179              | 10110011 | à          | 200              | 11001000 |
| ç          | 180              | 10110100 | ê          | 201              | 11001001 |
| ç          | 181              | 10110101 | ò          | 202              | 11001010 |
| ñ          | 182              | 10110110 | û          | 203              | 11001011 |
| ñ          | 183              | 10110111 | ä          | 204              | 11001100 |
| !          | 184              | 10111000 | ë          | 205              | 11001101 |
| ó          | 185              | 10111001 | ö          | 206              | 11001110 |
| ü          | 186              | 10111010 | ü          | 207              | 11001111 |
| £          | 187              | 10111011 | ñ          | 208              | 11010000 |
| ¥          | 188              | 10111100 | î          | 209              | 11010001 |
| §          | 189              | 10111101 | ø          | 210              | 11010010 |
| f          | 190              | 10111110 | Æ          | 211              | 11010011 |
| ¢          | 191              | 10111111 | à          | 212              | 11010100 |
| â          | 192              | 11000000 | í          | 213              | 11010101 |

## Roman Extension Character Set (continued)

| ASCII Char | Equivalent Forms |          |
|------------|------------------|----------|
|            | Dec              | Binary   |
| ø          | 214              | 11010110 |
| æ          | 215              | 11010111 |
| å          | 216              | 11011000 |
| ı          | 217              | 11011001 |
| ö          | 218              | 11011010 |
| ü          | 219              | 11011011 |
| é          | 220              | 11011100 |
| ı          | 221              | 11011101 |
| þ          | 222              | 11011110 |
| ó          | 223              | 11011111 |
| á          | 224              | 11100000 |
| â          | 225              | 11100001 |
| ä          | 226              | 11100010 |
| ð          | 227              | 11100011 |
| đ          | 228              | 11100100 |
| ı          | 229              | 11100101 |
| ı̇         | 230              | 11100110 |
| ó          | 231              | 11100111 |
| ò          | 232              | 11101000 |
| õ          | 233              | 11101001 |
| ö          | 234              | 11101010 |

| ASCII Char | Equivalent Forms |          |
|------------|------------------|----------|
|            | Dec              | Binary   |
| š          | 235              | 11101011 |
| ṧ         | 236              | 11101100 |
| ú          | 237              | 11101101 |
| ý          | 238              | 11101110 |
| ğ          | 239              | 11101111 |
| ƒ          | 240              | 11110000 |
| ƒ̇         | 241              | 11110001 |
|            | 242              | 11110010 |
|            | 243              | 11110011 |
|            | 244              | 11110100 |
|            | 245              | 11110101 |
| -          | 246              | 11110110 |
| +          | 247              | 11110111 |
| ×          | 248              | 11111000 |
| ÷          | 249              | 11111001 |
| ◊          | 250              | 11111010 |
| ◊̇         | 251              | 11111011 |
| ■          | 252              | 11111100 |
| ◊          | 253              | 11111101 |
| ±          | 254              | 11111110 |
|            | 255              | 11111111 |

## Reset Conditions

| Condition   | Power-on           | Reset   | Scratch | Run | Chain | Init | Call | Subend | Load |
|---|--------------------|---------|---------|-----|-------|------|------|--------|------|
| <b>CRT:</b>   |                    |         |         |     |       |      |      |        |      |
| CRT IS  | 1                  | R       | —       | —   | —     | —    | —    | —      | —    |
| ALPHA/GRAPHICS  | alpha              | R       | R       | R   | R     | R    | —    | —      | R    |
| Current Display Line  | 2                  | —       | —       | —   | —     | —    | —    | —      | —    |
| CURSOR ON/OFF   | on                 | R       | R       | —   | —     | —    | —    | —      | —    |
| <b>Keyboard:</b>  |                    |         |         |     |       |      |      |        |      |
| Typing Aids   | default            | R       | —       | —   | —     | —    | —    | —      | —    |
| User-defined keys<br>(in programs)  | none               | R       | R       | —   | —     | —    | *    | *      | —    |
| ENABLE KYBD   | none               | R       | —       | —   | —     | —    | —    | —      | —    |
| Keyboard mode   | typewriter         | —       | —       | —   | —     | —    | —    | —      | —    |
| <b>Printer:</b>   |                    |         |         |     |       |      |      |        |      |
| PRINT ALL   | off                | R       | —       | —   | —     | —    | —    | —      | —    |
| Print Column  | 1                  | R       | R       | R   | R     | R    | —    | —      | R    |
| PRINTER IS  | 2                  | 2       | —       | —   | —     | —    | —    | —      | —    |
| <b>Variables:</b>   |                    |         |         |     |       |      |      |        |      |
| Program variables   | none               | —       | R       | R   | R     | †    | —    | —      | R    |
| Keyboard variables  | none               | R       | R       | R   | R     | R    | —    | —      | R    |
| COMmon variables  | none               | —       | R       | —   | —     | —    | —    | —      | R    |
| OPTION BASE   | 0                  | —       | R       | R   | R     | R    | *    | *      | R    |
| <b>Graphics:</b>  |                    |         |         |     |       |      |      |        |      |
| PLOTTER IS  | 1                  | R       | —       | —   | —     | —    | —    | —      | —    |
| Graphics display  | none               | R       | R       | R   | R     | R    | R    | R      | R    |
| GU's/UU's mode  | UU's               | UU's    | —       | —   | —     | —    | —    | —      | —    |
| Scaling units   | GU's               | GU's    | —       | —   | —     | —    | —    | —      | —    |
| Pen   | 1                  | 1       | —       | —   | —     | —    | —    | —      | —    |
| LINE TYPE   | 1                  | 1       | —       | —   | —     | —    | —    | —      | —    |
| CSIZE   | default            | default | —       | —   | —     | —    | —    | —      | —    |
| LORG  | 1                  | R       | —       | —   | —     | —    | —    | —      | —    |
| Graphics limits   | device limits      | R       | —       | —   | —     | —    | —    | —      | —    |
| Plotting area   | graphics<br>limits | R       | —       | —   | —     | —    | —    | —      | —    |
| Pen location  | lower-left         | R       | —       | —   | —     | —    | —    | —      | —    |
| PDIR  | 0                  | 0       | —       | —   | —     | —    | —    | —      | —    |
| LDIR  | 0                  | 0       | —       | —   | —     | —    | —    | —      | —    |
| * CALL suspends the program ON KEY# assignments and option base. They are restored by SUBEND.<br>† Allocated. |                    |         |         |     |       |      |      |        |      |

R returned to power-on state.  
 — no effect.

### 4-8 Reference Tables

## Reset Conditions (continued)

| Condition  | Power-on                                      | Reset                           | Scratch                         | Run                             | Chain                           | Init                            | Call                            | Subend                          | Load                            |
|--|---|---------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| <b>Math:</b><br>DEFAULT ON/OFF<br>Trigonometric mode<br>RANDOMIZE seed   | on<br>rad<br>default value                    | R<br>R<br>R                     | —<br>—<br>—                     | —<br>—<br>—                     | —<br>—<br>—                     | —<br>—<br>—                     | —<br>—<br>—                     | —<br>—<br>—                     | —<br>—<br>—                     |
| <b>Clock functions:</b><br>TIME<br>DATE  | —<br>—  | —<br>—                          | —<br>—                          | —<br>—                          | —<br>—                          | —<br>—                          | —<br>—                          | —<br>—                          | —<br>—                          |
| <b>Even-initiated branching:</b><br>ON EOT<br>ON ERROR<br>ON INTR<br>ON KEY#<br>ON KYBD<br>ON TIMEOUT<br>ON TIMER#   | off<br>off<br>off<br>off<br>off<br>off<br>off | —<br>—<br>—<br>—<br>—<br>—<br>— | R<br>R<br>R<br>R<br>R<br>R<br>R | R<br>R<br>R<br>R<br>R<br>R<br>R | R<br>R<br>R<br>R<br>R<br>R<br>R | —<br>R<br>R<br>R<br>R<br>R<br>R | *<br>*<br>*<br>*<br>*<br>*<br>* | *<br>*<br>*<br>*<br>*<br>*<br>* | —<br>—<br>—<br>—<br>—<br>—<br>— |
| <b>Mass Storage:</b><br>ASSIGN# buffers<br>MASS STORAGE IS<br>CHECK READ<br>READ#/PRINT#<br>pointer  | none<br>cwd†<br>off<br>none                   | R<br>—<br>—<br>—                | R<br>—<br>—<br>—                | R<br>—<br>—<br>—                | R<br>—<br>—<br>—                | R<br>—<br>—<br>—                | —<br>—<br>—<br>—                | —<br>—<br>—<br>—                | R<br>—<br>—<br>—                |
| <b>Tracing:</b><br>TRACE<br>TRACE VAR<br>TRACE ALL   | off<br>off<br>off                             | R<br>R<br>R                     | R<br>R<br>R                     | —<br>—<br>—                     | —<br>—<br>—                     | —<br>—<br>—                     | *<br>*<br>*                     | *<br>*<br>*                     | R<br>R<br>R                     |
| NPAR<br>READ/DATA pointer<br>Binary Programs<br>BASIC program<br>Subprogram(s)<br>flags  | 0<br>none<br>none<br>none<br>none<br>cleared  | R<br>R<br>—<br>—<br>—<br>—      | R<br>R<br>R<br>R<br>R<br>R      | R<br>R<br>—<br>—<br>—<br>R      | R<br>R<br>—<br>—<br>—<br>R      | R<br>R<br>—<br>—<br>—<br>—      | R<br>R<br>—<br>—<br>—<br>R      | —<br>‡<br>—<br>—<br>—<br>—      | R<br>R<br>R<br>—<br>R<br>R      |
| <p>* CALL disables tracing and even-initiated branching until control returns to the calling program. SUBEND disables tracing and branching in the subprogram.</p> <p>† Current working directory.</p> <p>‡ CALL saves the position of the DATA pointer; SUBEND restores its position.</p> |   |                                 |                                 |                                 |                                 |                                 |                                 |                                 |                                 |

R returned to power-on state.  
— no effect.



## Boundaries and Scaling

| Condition or Statement | Parameter Units | Effect on Mode<br>GU's vs. UU's | Effect on Scaling<br>Units                          | Effect on Graphics<br>Limits                           | Effect on Plotting<br>Boundaries                       |
|------------------------|-----------------|---------------------------------|---|--|--|
| Power-on or Reset      | —               | In UU's                         | UU's=GU's<br>Shortest dimension=<br>100 GU's        | Set to default graphics limits of the graphics display | Set to default graphics limits of the graphics display |
| PLOTTER IS             | —               | Set to UU's mode                | UU's=GU's   | Read from device                                       | Set to graphics limits                                 |
| LIMIT                  | mm              | Set to UU's mode                | UU's=GU's   | Set according to LIMIT parameters                      | Set to graphics limits                                 |
| LOCATE                 | GU's            | No effect                       | No effect   | No effect  | Set according to LOCATE parameters                     |
| CLIP                   | Current units   | No effect                       | No effect   | No effect  | Set according to CLIP parameters                       |
| UNCLIP                 | —               | No effect                       | No effect   | No effect  | Set to current graphics limits                         |
| SCALE                  | UU's            | Set to UU's mode                | Set according to SCALE parameters                   | No effect  | No effect  |
| SHOW                   | UU's            | Set to UU's mode                | Set in equal x,y units according to SHOW parameters | No effect  | No effect  |
| MSCALE                 | mm              | Set to UU's mode                | Set to mm units according to MSCALE parameters      | No effect  | No effect  |
| SETGU                  | —               | Set to GU's mode                | GU's  | No effect  | Temporarily set to graphics limits                     |
| SETUU                  | —               | Set to UU's mode                | UU's  | No effect  | Restores plotting boundaries                           |

## Reflecting Plots WITH LIMIT, LOCATE, SCALE and SHOW

| Order of Statement Parameters | Effect                        |
|-------------------------------|-------------------------------|
| x-max, x-min, y-min, y-max    | Reflects output across y-axis |
| x-min, x-max, y-max, y-min    | Reflects output across x-axis |
| x-max, x-min, y-max, y-min    | Reflects output across origin |

## Pen Up/Down Status

| Statement  | Pen Status after Execution |
|------------|----------------------------|
| AXES       | Up                         |
| DRAW       | Down                       |
| FRAME      | Up                         |
| GRID       | Up                         |
| IDRAW      | Down                       |
| IMOVE      | Up                         |
| IPLOT      | Determined by parameter    |
| LABEL      | Up                         |
| LAXES      | Up                         |
| LGRID      | Up                         |
| LIMIT      | Up                         |
| MOVE       | Up                         |
| PENUP      | Up                         |
| PLOT       | Determined by parameter    |
| PLOTTER IS | Up                         |
| RPLOT      | Determined by parameter    |
| XAXIS      | Up                         |
| YAXIS      | Up                         |

## Pen Control With PLOT, IPLOT, and RPLLOT

| Pen Control Parameter | Pen Action                 |
|-----------------------|----------------------------|
| Positive, even        | Pen moved and then lifted  |
| Positive, odd         | Pen moved and then lowered |
| Negative, even        | Pen lifted and then moved  |
| Negative, odd         | Pen lowered and then moved |

## Graphics Display Pen Numbers

| Pen Number   | Graphics Display Operation  |
|--------------|---|
| $\geq 1$     | Plots white dots.   |
| 0            | Pen is deactivated and does not plot.   |
| -1 or $< -2$ | Plots black dots.   |
| -2           | Performs an exclusive or, plotting white dots over black dots and black dots over white dots. |

## Pen Numbers With GCLEAR

| Pen Number      | Display Background Color After GCLEAR |
|-----------------|---------------------------------------|
| $\geq 1$        | Black                                 |
| -1 or $\leq -2$ | White                                 |
| 0               | Uses previous pen number              |

## Branch Precedence Table

Branch precedence indicates the order in which event-initiated branches are taken. Events with lower precedence can interrupt an active service routine. When two branches are pending, the one with the lower precedence number is taken first. When the first line of the service routine has been executed, the second pending branch is taken (unless the first line disables that branching).

| Priority | Branch Type | Select Code      |    |    |    |    |    |    |    |  |
|----------|-------------|------------------|----|----|----|----|----|----|----|--|
|          |             | 3                | 4  | 5  | 6  | 7  | 8  | 9  | 10 |  |
| 1        | ON ERROR    | ←----- 1 -----→  |    |    |    |    |    |    |    |  |
| 2        | ON INTR     | 2                | 3  | 4  | 5  | 6  | 7  | 8  | 9  |  |
| 3        | ON TIMEOUT  | 10               | 11 | 12 | 13 | 14 | 15 | 16 | 17 |  |
| 4        | ON EOT      | 18               | 19 | 20 | 21 | 22 | 23 | 24 | 25 |  |
| 5        | ON TIMER#   | ←----- 26 -----→ |    |    |    |    |    |    |    |  |
| 6        | ON KYBD     | ←----- 27 -----→ |    |    |    |    |    |    |    |  |
| 7        | ON KEY#     | ←----- 28 -----→ |    |    |    |    |    |    |    |  |

## HP-IB Control-Line Signals

| Mnemonic | Message Name               | Response   |
|----------|----------------------------|--|
| ATN      | Attention                  | The Controller Active device asserts ATN true to source commands on the data bus or, in conjunction with EOI, to do a parallel poll. When ATN is false, data may be sent over the data bus by a designated talker. |
| DAV      | Data Valid                 | Allows source to validate data lines.  |
| EOI      | End or Identify            | Terminates a flow of data, and can be used with ATN to do a parallel poll.   |
| IFC      | Interface Clear<br>(Abort) | The system controller uses this to place talkers and listeners in an unaddressed state. If control has been passed, the system controller again becomes active controller when it asserts IFC.                     |
| NDAC     | Not Data<br>Accepted       | Used by devices to inform the source that data has been accepted.  |
| NRFD     | Not Ready For<br>Data      | Used to inform the source that all listener devices are ready for data.  |
| REN      | Remote Enable              | Removes all devices from Local Lockout mode and causes all devices to revert to manual control. Any device that is addressed to listen while REN is true is placed in the REMOTE mode of operation.                |
| SRQ      | Service Request            | Indicates a device's need for interaction with the controller.   |

## HP-IB Multiple-Line Commands

| Mnemonic          | Message Name            | Decimal Value | Response   |
|-------------------|-------------------------|---------------|--|
| DCL               | Device Clear            | 20            | Causes all devices to be initialized to a predefined or power-up state.  |
| GET               | Group Execute Trigger   | 8             | Signals one or more devices to simultaneously initiate a set of device-dependent actions.                                |
| GTL               | Go To Local             | 1             | Causes selected device(s) to switch to local (front panel) control.  |
| LAG<br>(LA0-LA30) | Listen Address Group    | 32-62         | A group of 31 listen addresses, one of which corresponds to the listen address of the interface.                         |
| LLO               | Local Lockout           | 17            | Disables remote-mode override switch (the LOCAL button) on peripheral device(s).   |
| SCG<br>(SC0-SC31) | Secondary Command Group | 96-127        | A group of 32 commands that are only recognized if they immediately follow a talk or listen address.                     |
| SDC               | Selected Device Clear   | 4             | Causes a specified device to be initialized to a predefined or power-up state.   |
| SPD               | Serial Poll Disable     | 25            | Devices exit serial poll mode and are not allowed to send their status byte.   |
| SPE               | Serial Poll Enable      | 24            | Devices enter serial poll mode and are allowed to send their status byte when addressed to talk.                         |
| TAG<br>(TA0-TA30) | Talk Address Group      | 64-94         | A group of 31 talk addresses.  |
| TCT               | Take Control            | 9             | Passes bus controller responsibilities from the current controller to a device that can assume the bus supervisory role. |
| UNL               | Unlisten                | 63            | Device(s) become unaddressed to listen.  |
| UNT               | Untalk                  | 95            | Device(s) become unaddressed to talk.  |



# 5

## I/O Registers

---

### I/O Buffer Registers

#### Status Registers

| Register | Default Value | Function               |
|----------|---------------|------------------------|
| SR0      | 1             | Buffer empty pointer   |
| SR1      | 0             | Buffer fill pointer    |
| SR2      | 0             | Active-in select code  |
| SR3      | 0             | Active-out select code |

#### Control Registers

| Register | Default Value | Function             |
|----------|---------------|----------------------|
| CR0      | 1             | Buffer empty pointer |
| CR1      | 0             | Buffer fill pointer  |



---

## HP-IB Interface

### Status Register 0: Interface Identification

Status Register 0 always returns the value 1 (“00000001”), the identification code for an HP-IB interface.

### Control Register 0: Parity Control

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2     | Bit 1     | Bit 0     |
|-------------|------------|------------|------------|-----------|-----------|-----------|-----------|
| Not Used    |            |            |            | Odd       | Even      | Always 1  | Always 0  |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4 | Value = 2 | Value = 1 |

### Status Register 1: Interrupt Cause

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2      | Bit 1     | Bit 0     |
|-------------|------------|------------|------------|-----------|------------|-----------|-----------|
| IFC         | LA         | CA         | TA         | SRQ       | DCL or SDC | GET       | SCG       |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4  | Value = 2 | Value = 1 |

### Control Register 1: Interrupt Mask

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2      | Bit 1     | Bit 0     |
|-------------|------------|------------|------------|-----------|------------|-----------|-----------|
| IFC         | LA         | CA         | TA         | SRQ       | DCL or SDC | GET       | SCG       |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4  | Value = 2 | Value = 1 |

## Status Register 2: HP-IB Control Lines

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2     | Bit 1     | Bit 0     |
|-------------|------------|------------|------------|-----------|-----------|-----------|-----------|
| Not Used    | REN        | SRQ        | ATN        | EOI       | DAV       | NDAC      | NRFD      |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4 | Value = 2 | Value = 1 |

## Control Register 2: HP-IB Control Lines

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2     | Bit 1     | Bit 0     |
|-------------|------------|------------|------------|-----------|-----------|-----------|-----------|
| Not Used    | REN        | SRQ        | ATN        | EOI       | DAV       | NDAC      | NRFD      |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4 | Value = 2 | Value = 1 |

## Status Register 3: HP-IB Data Lines

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2     | Bit 1     | Bit 0     |
|-------------|------------|------------|------------|-----------|-----------|-----------|-----------|
| DIO8        | DIO7       | DIO6       | DIO5       | DIO4      | DIO3      | DIO2      | DIO1      |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4 | Value = 2 | Value = 1 |

## Control Register 3: HP-IB Data Lines

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2     | Bit 1     | Bit 0     |
|-------------|------------|------------|------------|-----------|-----------|-----------|-----------|
| DIO8        | DIO7       | DIO6       | DIO5       | DIO4      | DIO3      | DIO2      | DIO1      |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4 | Value = 2 | Value = 1 |

## Status Register 4: HP-IB Address/System Controller

| Bit 7       | Bit 6      | Bit 5             | Bit 4         | Bit 3     | Bit 2     | Bit 1     | Bit 0     |
|-------------|------------|-------------------|---------------|-----------|-----------|-----------|-----------|
| Not Used    |            | System Controller | HP-IB Address |           |           |           |           |
| Value = 128 | Value = 64 | Value = 32        | Value = 16    | Value = 8 | Value = 4 | Value = 2 | Value = 1 |

## Status Register 5: HP-IB State

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2        | Bit 1     | Bit 0     |
|-------------|------------|------------|------------|-----------|--------------|-----------|-----------|
| SC          | LA         | CA         | TA         | SPE       | Parity Error | REN       | LLO       |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4    | Value = 2 | Value = 1 |

## Status Register 6: Secondary Command Register

| Bit 7       | Bit 6      | Bit 5      | Bit 4             | Bit 3     | Bit 2     | Bit 1     | Bit 0     |
|-------------|------------|------------|-------------------|-----------|-----------|-----------|-----------|
| Not Used    |            |            | Secondary Command |           |           |           |           |
| Value = 128 | Value = 64 | Value = 32 | Value = 16        | Value = 8 | Value = 4 | Value = 2 | Value = 1 |

## Control Register 16: EOL Control

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3                                | Bit 2     | Bit 1     | Bit 0     |
|-------------|------------|------------|------------|--------------------------------------|-----------|-----------|-----------|
| EOI Enable  | Not Used   |            |            | Number of Characters in EOL Sequence |           |           |           |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8                            | Value = 4 | Value = 2 | Value = 1 |

## **Control Registers 17 through 23: EOL Sequence**

Control Registers 17 through 23 contain the decimal value of the characters sent as the EOL (end-of-line) sequence.

## Serial Interface

The various cables are indicated in the registers tables by the following superscripts:

- \* Modem Cable F/M.
- † Instrumentation Cable F/F.
- ‡ Printer Cable F/F.

### Status Register 0: Interface I.D.

This register always returns the value 2 ("00000010"), the identification code for a serial interface.

### Status/Control Register 1: Interrupt Mask

| Bit 7          | Bit 6         | Bit 5        | Bit 4                   | Bit 3         | Bit 2           | Bit 1                     | Bit 0         |
|----------------|---------------|--------------|-------------------------|---------------|-----------------|---------------------------|---------------|
| Break Received | Framing Error | Parity Error | Received Data Available | DCD*<br>RTS†‡ | Auto-disconnect | DSR*<br>RTS†<br>Not Used‡ | CTS*<br>DTR†‡ |
| Value = 128    | Value = 64    | Value = 32   | Value = 16              | Value = 8     | Value = 4       | Value = 2                 | Value = 1     |

## Status/Control Register 2: Modem Control Signals

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2                          | Bit 1          | Bit 0                     |
|-------------|------------|------------|------------|-----------|--------------------------------|----------------|---------------------------|
| Not Used    |            |            |            |           | DRS*<br>No Connection†<br>DSR‡ | RTS*<br>DCD† ‡ | DTR*<br>CTS /DSR†<br>CTS‡ |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4                      | Value = 2      | Value = 1                 |

## Status Register 3: Modem Status

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3          | Bit 2     | Bit 1                     | Bit 0          |
|-------------|------------|------------|------------|----------------|-----------|---------------------------|----------------|
| Not Used    |            |            |            | DCD*<br>RTS† ‡ | Not Used  | DSR*<br>DTR†<br>Not Used‡ | CTS*<br>DTR† ‡ |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8      | Value = 4 | Value = 2                 | Value = 1      |

## Control Register 3: Standard Baud Rates

Control Register 3 selects a standard baud rate for transmitted and received data.

**Table A-3. Standard Baud Rates**

|                  |      |      |      |       |      |      |      |      |
|------------------|------|------|------|-------|------|------|------|------|
| <b>Value</b>     | 0    | 1    | 2    | 3     | 4    | 5    | 6    | 7    |
| <b>Baud Rate</b> | 50   | 75   | 110  | 134.5 | 150  | 200  | 300  | 600  |
| <b>Value</b>     | 8    | 9    | 10   | 11    | 12   | 13   | 14   | 15   |
| <b>Baud Rate</b> | 1200 | 1800 | 2000 | 2400  | 2600 | 4800 | 7200 | 9600 |

## Status/Control Register 4: Line Characteristics

| Bit 7       | Bit 6      | Bit 5        | Bit 4       | Bit 3         | Bit 2     | Bit 1            | Bit 0     |
|-------------|------------|--------------|-------------|---------------|-----------|------------------|-----------|
| Not Used    | Set Break  | Force Parity | Even Parity | Enable Parity | Stop Bits | Character Length |           |
| Value = 128 | Value = 64 | Value = 32   | Value = 16  | Value = 8     | Value = 4 | Value = 2        | Value = 1 |

### Parity Selection

| Bit 5 | Bit 4 | Bit 3 | Parity Specified |
|-------|-------|-------|------------------|
| 0     | 0     | 0     | No Parity bit    |
| 0     | 0     | 1     | Odd parity       |
| 0     | 1     | 1     | Even parity      |
| 1     | 0     | 1     | Always 1         |
| 1     | 1     | 1     | Always 0         |

### Character Length

| Bit 1 | Bit 0 | Character Length |
|-------|-------|------------------|
| 0     | 0     | 5                |
| 0     | 1     | 6                |
| 1     | 0     | 7                |
| 1     | 1     | 8                |

## Status/Control Register 5: Modem Features

| Bit 7       | Bit 6      | Bit 5             | Bit 4              | Bit 3          | Bit 2     | Bit 1                      | Bit 0         |
|-------------|------------|-------------------|--------------------|----------------|-----------|----------------------------|---------------|
| Not Used    |            | Receive Handshake | Transmit Handshake | DCD*<br>RTS† ‡ | Not Used  | DSR*<br>DTR†‡<br>Not Used‡ | CTS*<br>DTR†‡ |
| Value = 128 | Value = 64 | Value = 32        | Value = 16         | Value = 8      | Value = 4 | Value = 2                  | Value = 1     |

## Status/Control Registers 6 and 7: Divisor

These registers specify the divisor used by the baud rate generator to establish a non-standard baud rate:

1. Divisor =  $115,200 \div (\text{Baud Rate})$
2. Status/Control Register 6 value = (Divisor) DIV 256
3. Status/Control Register 7 value = (Divisor) MOD 256

## Status/Control Register 8: Error Replacement Character

This register contains the decimal value of the ASCII character used as the parity and framing error replacement character.

## Status/Control Register 9: Transmitter/Receiver Control

| Bit 7              | Bit 6                  | Bit 5                | Bit 4                     | Bit 3                           | Bit 2               | Bit 1            | Bit 0           |
|--------------------|------------------------|----------------------|---------------------------|---------------------------------|---------------------|------------------|-----------------|
| Enable Transmitter | Strip Received Rubouts | Strip Received Nulls | Change Character if Error | Set Bit 7 of Character if Error | Reset Receive Queue | Auto-Echo Enable | Enable Receiver |
| Value = 128        | Value = 64             | Value = 32           | Value = 16                | Value = 8                       | Value = 4           | Value = 2        | Value = 1       |

## Status Register 10: Line Status

| Bit 7       | Bit 6      | Bit 5                   | Bit 4          | Bit 3         | Bit 2        | Bit 1     | Bit 0                   |
|-------------|------------|-------------------------|----------------|---------------|--------------|-----------|-------------------------|
| Not Used    |            | Transmit Register Empty | Break Received | Framing Error | Parity Error | Not Used  | Received Data Available |
| Value = 128 | Value = 64 | Value = 32              | Value = 16     | Value = 8     | Value = 4    | Value = 2 | Value = 1               |



## Status Register 11: I/O Termination Cause

| Bit 7                   | Bit 6                  | Bit 5           | Bit 4                   | Bit 3                   | Bit 2                   | Bit 1                   | Bit 0                    |
|-------------------------|------------------------|-----------------|-------------------------|-------------------------|-------------------------|-------------------------|--------------------------|
| End of Output Data List | End of Input Data List | COUNT Satisfied | CR15 Character Received | CR14 Character Received | CR13 Character Received | CR12 Character Received | DELIM Character Received |
| Value = 128             | Value = 64             | Value = 32      | Value = 16              | Value = 8               | Value = 4               | Value = 2               | Value = 1                |

## Control Register 11: Input Data Control

| Bit 7                | Bit 6                 | Bit 5      | Bit 4             | Bit 3             | Bit 2             | Bit 1             | Bit 0     |
|----------------------|-----------------------|------------|-------------------|-------------------|-------------------|-------------------|-----------|
| Enable Transmit Flag | Disable Transmit Flag | Not Used   | Terminate if CR15 | Terminate if CR14 | Terminate if CR13 | Terminate if CR12 | Not Used  |
| Value = 128          | Value = 64            | Value = 32 | Value = 16        | Value = 8         | Value = 4         | Value = 2         | Value = 1 |

## Control Registers 12 through 15: Termination Characters

These registers contain the decimal value of the input termination characters specified by Control Register 11.

## Control Register 16: End-of-Line Sequence

| Bit 7           | Bit 6                | Bit 5               | Bit 4      | Bit 3     | Bit 2     | Bit 1     | Bit 0     |
|-----------------|----------------------|---------------------|------------|-----------|-----------|-----------|-----------|
| Auto RTS Enable | EOL Transmit Disable | EOL Character Count |            |           |           |           |           |
| Value = 128     | Value = 64           | Value = 32          | Value = 16 | Value = 8 | Value = 4 | Value = 2 | Value = 1 |

## **Control Registers 17 through 23: EOL Sequence Characters**

Control Registers 17 through 23 contain the decimal value of the characters sent as the EOL (end-of-line) sequence.

---

## BCD Interface

### Status Register 0: Interface Identification

Status Register 0 always returns the value 3 ("00000011"), the interface identification code for a BCD interface.

### Status Register 1: Interrupt Cause

| Bit 7  | Bit 6         | Bit 5         | Bit 4         | Bit 3  | Bit 2        | Bit 1        | Bit 0        |
|--|---------------|---------------|---------------|--|--------------|--------------|--------------|
| Generated from<br>Function B<br>(Most Significant Digit) |               |               |               | Generated from<br>Function A<br>(Most Significant Digit) |              |              |              |
| Value<br>= 128   | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8   | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

### Control Register 1: Interrupt Mask

| Bit 7                                  | Bit 6         | Bit 5         | Bit 4         | Bit 3                                  | Bit 2        | Bit 1        | Bit 0        |
|--|---------------|---------------|---------------|--|--------------|--------------|--------------|
| Function B<br>(Most Significant Digit) |               |               |               | Function A<br>(Most Significant Digit) |              |              |              |
| Value<br>= 128                         | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8                           | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

## Status Register 2: Control Line Messages

| Bit 7          | Bit 6         | Bit 5         | Bit 4         | Bit 3        | Bit 2        | Bit 1        | Bit 0        |
|----------------|---------------|---------------|---------------|--------------|--------------|--------------|--------------|
| I/OA           | I/OB          | CTLA          | CTLB          | FLGA         | FLGB         | Not Used     |              |
| Value<br>= 128 | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8 | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

0=input; 1=output

0=ready; 1=busy

## Control Register 2: Handshake Lines and Port 10

| Bit 7          | Bit 6         | Bit 5         | Bit 4         | Bit 3                              | Bit 2        | Bit 1        | Bit 0        |
|----------------|---------------|---------------|---------------|------------------------------------|--------------|--------------|--------------|
| I/OA           | I/OB          | CTLA          | CTLB          | Port 10 Output<br>(when available) |              |              |              |
| Value<br>= 128 | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8                       | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

## Status/Control Register 3: Mantissa Digits

| Bit 7  | Bit 6         | Bit 5         | Bit 4         | Bit 3  | Bit 2        | Bit 1        | Bit 0        |
|--|---------------|---------------|---------------|--|--------------|--------------|--------------|
| Number of Digits<br>for Channel B<br>Mantissa (0-11) |               |               |               | Number of Digits<br>for Channel A<br>Mantissa (0-11) |              |              |              |
| Value<br>= 128                                       | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8   | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

### Status/Control Register 4: Exponent Digits

| Bit 7   | Bit 6         | Bit 5         | Bit 4         | Bit 3   | Bit 2        | Bit 1        | Bit 0        |
|---|---------------|---------------|---------------|---|--------------|--------------|--------------|
| Number of Digits<br>for Channel B<br>Exponent (0-3) |               |               |               | Number of Digits<br>for Channel A<br>Exponent (0-3) |              |              |              |
| Value<br>= 128                                      | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8  | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

### Status/Control Register 5: Function Digits

| Bit 7  | Bit 6         | Bit 5         | Bit 4         | Bit 3  | Bit 2        | Bit 1        | Bit 0        |
|--|---------------|---------------|---------------|--|--------------|--------------|--------------|
| Number of Digits<br>for Channel B<br>Function (0-11) |               |               |               | Number of Digits<br>for Channel A<br>Function (0-11) |              |              |              |
| Value<br>= 128                                       | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8   | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

### Status/Control Register 6: Decimal Point Placement

| Bit 7   | Bit 6         | Bit 5         | Bit 4         | Bit 3   | Bit 2        | Bit 1        | Bit 0        |
|---|---------------|---------------|---------------|---|--------------|--------------|--------------|
| Number of Mantissa Digits to the<br>Right of the Decimal Point<br>(Channel B) |               |               |               | Number of Mantissa Digits to the<br>Right of the Decimal Point<br>(Channel A) |              |              |              |
| Value<br>= 128  | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8  | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

## Status/Control Register 7: Control Sense and Handshake Mode

| Bit 7            | Bit 6            | Bit 5            | Bit 4            | Bit 3            | Bit 2            | Bit 1            | Bit 0            |
|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| Logic Sense I/OA | Logic Sense I/OB | Logic Sense CTLA | Logic Sense CTLB | Logic Sense FLGA | Logic Sense FLGB | Handshake Mode A | Handshake Mode B |
| Value = 128      | Value = 64       | Value = 32       | Value = 16       | Value = 8        | Value = 4        | Value = 2        | Value = 1        |

## Status/Control Register 8: Mantissa and Exponent Digit Sense

| Bit 7  | Bit 6      | Bit 5      | Bit 4      | Bit 3  | Bit 2     | Bit 1     | Bit 0     |
|--|------------|------------|------------|--|-----------|-----------|-----------|
| Logic Sense for Channel B Mantissa and Exponent Digits |            |            |            | Logic Sense for Channel A Mantissa and Exponent Digits |           |           |           |
| Value = 128  | Value = 64 | Value = 32 | Value = 16 | Value = 8  | Value = 4 | Value = 2 | Value = 1 |

## Status/Control Register 9: Function Digit Sense

| Bit 7                                     | Bit 6      | Bit 5      | Bit 4      | Bit 3                                     | Bit 2     | Bit 1     | Bit 0     |
|---|------------|------------|------------|---|-----------|-----------|-----------|
| Logic Sense for Channel B Function Digits |            |            |            | Logic Sense for Channel A Function Digits |           |           |           |
| Value = 128                               | Value = 64 | Value = 32 | Value = 16 | Value = 8                                 | Value = 4 | Value = 2 | Value = 1 |

## Status/Control Register 10: Handshake Lines and Port 10

| Bit 7                  | Bit 6                  | Bit 5                  | Bit 4                  | Bit 3                        | Bit 2     | Bit 1     | Bit 0     |
|------------------------|------------------------|------------------------|------------------------|------------------------------|-----------|-----------|-----------|
| Exponent B Logic Sense | Mantissa B Logic Sense | Exponent A Logic Sense | Mantissa A Logic Sense | Logic Sense for Port 10 Data |           |           |           |
| Value = 128            | Value = 64             | Value = 32             | Value = 16             | Value = 8                    | Value = 4 | Value = 2 | Value = 1 |

## GPIO Interface

### Status Register 0 : Interface Identification

Status Register 0 always returns the Value 4 (“00000100”), the identification code of the GPIO interface.

### Control Register 0: Parity Control

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2     | Bit 1      | Bit 0       |
|-------------|------------|------------|------------|-----------|-----------|------------|-------------|
| Not Used    |            |            |            | Odd       | Even      | Always One | Always Zero |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4 | Value = 2  | Value = 1   |

### Status Register 1: Interrupt Cause

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2     | Bit 1        | Bit 0     |
|-------------|------------|------------|------------|-----------|-----------|--------------|-----------|
| FLGD        | FLGC       | FLGB       | FLGA       | Not Used  |           | Parity Error | Not Used  |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4 | Value = 2    | Value = 1 |

### Control Register 1: Interrupt Mask

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2     | Bit 1        | Bit 0     |
|-------------|------------|------------|------------|-----------|-----------|--------------|-----------|
| FLGD        | FLGC       | FLGB       | FLGA       | Not Used  |           | Parity Error | Not Used  |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4 | Value = 2    | Value = 1 |



## Status Register 2: Line Status

| Bit 7          | Bit 6         | Bit 5         | Bit 4         | Bit 3        | Bit 2        | Bit 1        | Bit 0        |
|----------------|---------------|---------------|---------------|--------------|--------------|--------------|--------------|
| FLGD           | FLGC          | FLGB          | FLGA          | CTLD         | CTLB         | CTLC         | CTLA         |
| Value<br>= 128 | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8 | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

## Control Register 2: Assertion Control

| Bit 7          | Bit 6         | Bit 5         | Bit 4         | Bit 3        | Bit 2        | Bit 1        | Bit 0        |
|----------------|---------------|---------------|---------------|--------------|--------------|--------------|--------------|
| RESB           | RESA          | Not Used      |               | CTLD         | CTLB         | CLTC         | CTLA         |
| Value<br>= 128 | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8 | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

## Status/Control Register 3: Handshake Line Normalization

| Bit 7          | Bit 6          | Bit 5          | Bit 4          | Bit 3          | Bit 2          | Bit 1          | Bit 0          |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Invert<br>FLGD | Invert<br>FLGC | Invert<br>FLGB | Invert<br>FLGA | Invert<br>CTLD | Invert<br>CTLB | Invert<br>CTLC | Invert<br>CTLA |
| Value<br>= 128 | Value<br>= 64  | Value<br>= 32  | Value<br>= 16  | Value<br>= 8   | Value<br>= 4   | Value<br>= 2   | Value<br>= 1   |

## Status/Control Register 4: Data Normalization and Handshake Control

| Bit 7             | Bit 6         | Bit 5              | Bit 4         | Bit 3            | Bit 2            | Bit 1            | Bit 0            |
|-------------------|---------------|--------------------|---------------|------------------|------------------|------------------|------------------|
| Handshake<br>Mode |               | Flag<br>Transition | Always<br>1   | Invert<br>Port D | Invert<br>Port C | Invert<br>Port B | Invert<br>Port A |
| Value<br>= 128    | Value<br>= 64 | Value<br>= 32      | Value<br>= 16 | Value<br>= 8     | Value<br>= 4     | Value<br>= 2     | Value<br>= 1     |

## Status/Control Register 5: Primary Address and Trigger Action

| Bit 7       | Bit 6      | Bit 5      | Bit 4                | Bit 3           | Bit 2     | Bit 1     | Bit 0     |
|-------------|------------|------------|----------------------|-----------------|-----------|-----------|-----------|
| R7 > DATA   | R7 = DATA  | R7 < DATA  | Enable Auto Response | Primary Address |           |           |           |
| Value = 128 | Value = 64 | Value = 32 | Value = 16           | Value = 8       | Value = 4 | Value = 2 | Value = 1 |

## Status/Control Register 6: Strobe Control

| Bit 7                | Bit 6                 | Bit 5      | Bit 4      | Bit 3     | Bit 2     | Bit 1     | Bit 0     |
|----------------------|-----------------------|------------|------------|-----------|-----------|-----------|-----------|
| Resolution Increment | Strobe Pulse Duration |            |            |           |           |           |           |
| Value = 128          | Value = 64            | Value = 32 | Value = 16 | Value = 8 | Value = 4 | Value = 2 | Value = 1 |

## Status/Control Register 7: Trigger Character

Bits 0 through 7 contain the decimal Value of the trigger character.

## Status/Control Register 8: Output Enable

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2     | Bit 1           | Bit 0           |
|-------------|------------|------------|------------|-----------|-----------|-----------------|-----------------|
| Not Used    |            |            |            |           |           | Output Enable B | Output Enable A |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4 | Value = 2       | Value = 1       |

## Status/Control Register 9: Output Inhibit

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2     | Bit 1     | Bit 0                 |
|-------------|------------|------------|------------|-----------|-----------|-----------|-----------------------|
| Not Used    |            |            |            |           |           |           | Enable Output Inhibit |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4 | Value = 2 | Value = 1             |

## Control Register 16: EOL Control

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2                                | Bit 1     | Bit 0     |
|-------------|------------|------------|------------|-----------|--------------------------------------|-----------|-----------|
| Not Used    |            |            |            |           | Number of Characters in EOL Sequence |           |           |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4                            | Value = 2 | Value = 1 |

## Control Registers 17 through 23: EOL Sequence

Control Registers 17 through 23 contain the decimal Values of the characters sent as the EOL (end-of-line) sequence.

---

## IP-IL Interface

### Status Register 0: Interface Identification

Status register 0 always returns the value 5 ("00000101"), the identification code for an HP-IL interface.

### Status Register 1: Interrupt Cause

| Bit 7          | Bit 6         | Bit 5         | Bit 4         | Bit 3        | Bit 2        | Bit 1        | Bit 0        |
|----------------|---------------|---------------|---------------|--------------|--------------|--------------|--------------|
| IFC            | LA            | CA            | TA            | SRQ          | CLR          | GET          | DDC          |
| Value<br>= 128 | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8 | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

### Control Register 1: Interrupt Mask

| Bit 7          | Bit 6         | Bit 5         | Bit 4         | Bit 3        | Bit 2        | Bit 1        | Bit 0        |
|----------------|---------------|---------------|---------------|--------------|--------------|--------------|--------------|
| IFC            | LA            | CA            | TA            | SRQ          | CLR          | GET          | DDC          |
| Value<br>= 128 | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8 | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

### Status Register 2: Last Control Bits

| Bit 7          | Bit 6         | Bit 5         | Bit 4         | Bit 3        | Bit 2        | Bit 1        | Bit 0        |
|----------------|---------------|---------------|---------------|--------------|--------------|--------------|--------------|
| Not Used       |               |               |               |              | C2           | C1           | C0           |
| Value<br>= 128 | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8 | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

### Control Register 2: Next Control Bits

| Bit 7          | Bit 6         | Bit 5         | Bit 4         | Bit 3        | Bit 2        | Bit 1        | Bit 0        |
|----------------|---------------|---------------|---------------|--------------|--------------|--------------|--------------|
| Not Used       |               |               |               |              | C2           | C1           | C0           |
| Value<br>= 128 | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8 | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

### Status Register 3: Last Loop Byte

| Bit 7          | Bit 6         | Bit 5         | Bit 4         | Bit 3        | Bit 2        | Bit 1        | Bit 0        |
|----------------|---------------|---------------|---------------|--------------|--------------|--------------|--------------|
| D7             | D6            | D5            | D4            | D3           | D2           | D1           | D0           |
| Value<br>= 128 | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8 | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

### Control Register 3: Next Data Byte

| Bit 7          | Bit 6         | Bit 5         | Bit 4         | Bit 3        | Bit 2        | Bit 1        | Bit 0        |
|----------------|---------------|---------------|---------------|--------------|--------------|--------------|--------------|
| D7             | D6            | D5            | D4            | D3           | D2           | D1           | D0           |
| Value<br>= 128 | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8 | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

### Status Register 4: Loop Address

| Bit 7          | Bit 6         | Bit 5         | Bit 4         | Bit 3        | Bit 2        | Bit 1        | Bit 0        |
|----------------|---------------|---------------|---------------|--------------|--------------|--------------|--------------|
| AAD            | Not Used      |               | A4            | A3           | A2           | A1           | A0           |
| Value<br>= 128 | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8 | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

### Control Register 4: Loop Address

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2     | Bit 1     | Bit 0     |
|-------------|------------|------------|------------|-----------|-----------|-----------|-----------|
| AAD         | Not Used   |            | A4         | A3        | A2        | A1        | A0        |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4 | Value = 2 | Value = 1 |

### Status Register 5: Interface Status

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2     | Bit 1     | Bit 0     |
|-------------|------------|------------|------------|-----------|-----------|-----------|-----------|
| Tx          | LA         | CA         | TA         | SRQ       | EAR       | REN       | LLO       |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4 | Value = 2 | Value = 1 |

### Control Register 5: Interface Status

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2     | Bit 1     | Bit 0     |
|-------------|------------|------------|------------|-----------|-----------|-----------|-----------|
| Not Used    |            |            |            |           |           |           | EAR       |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4 | Value = 2 | Value = 1 |

### Status Register 6: Device-Dependent Commands

| Bit 7       | Bit 6      | Bit 5      | Bit 4      | Bit 3     | Bit 2     | Bit 1     | Bit 0     |
|-------------|------------|------------|------------|-----------|-----------|-----------|-----------|
| Not Used    |            | T/L        | C4         | C3        | C2        | C1        | C0        |
| Value = 128 | Value = 64 | Value = 32 | Value = 16 | Value = 8 | Value = 4 | Value = 2 | Value = 1 |

## Status Register 7: Responding Devices

| Bit 7          | Bit 6         | Bit 5         | Bit 4         | Bit 3        | Bit 2        | Bit 1        | Bit 0        |
|----------------|---------------|---------------|---------------|--------------|--------------|--------------|--------------|
| Not Used       |               |               | A4            | A3           | A2           | A1           | A0           |
| Value<br>= 128 | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8 | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

## Control Register 16: EOL Control

| Bit 7                  | Bit 6         | Bit 5         | Bit 4         | Bit 3        | Bit 2        | Bit 1        | Bit 0        |
|------------------------|---------------|---------------|---------------|--------------|--------------|--------------|--------------|
| End<br>Frame<br>Enable | Not Used      |               |               |              | EOL2         | EOL1         | EOL0         |
| Value<br>= 128         | Value<br>= 64 | Value<br>= 32 | Value<br>= 16 | Value<br>= 8 | Value<br>= 4 | Value<br>= 2 | Value<br>= 1 |

## Control Registers 17 through 23: EOL Sequence

Control registers 17 through 23 contain the decimal value of the characters sent as the EOL (end-of-line) sequence.

# 6

## Errors

The following table lists prerun (program initialization) and runtime error and warning conditions. Syntax errors are not listed. With DEFAULT ON, conditions 1 through 8 return a warning message and default value. With DEFAULT OFF, an error message is returned and execution halts.

Keep in mind that prerun errors occur before program execution begins, and therefore cannot be trapped by ON ERROR routines.

| Warning Number | Message          | Causes   |
|----------------|------------------|--|
| 1              |                  | Not used.  |
| 2              | OVERFLOW         | Overflow; returns maximum value for specified precision.<br>■ REAL, SHORT, or INTEGER value is out of range.<br>■ Division by 0. |
| 3              | COT/CSC=INF      | COT or CSC of $n \times 180^\circ$ ; returns INF   |
| 4              | TAN/SEC=INF      | TAN or SEC of $n \times 90^\circ$ ; returns INF  |
| 5              | 0 <sup>NEG</sup> | Zero raised to a negative power; returns INF   |
| 6              | 0 <sup>0</sup>   | Zero raised to the zero power; returns 1   |
| 7              | NULL DATA        | Executing AXAMCOL or AMINCOL for a vector; returns 0.  |
| 8              |                  | Not used.  |



| Error Number | Message             | Causes   |
|--------------|---------------------|--|
| 9            | NEG^NON-INT         | Negative value raised to a non-integer power.  |
| 10           | SQR(—)              | Square root of a negative number.  |
| 11           | ARG OUT OF RANGE    | Argument or parameter out of range:<br><ul style="list-style-type: none"> <li>■ ATN2(0,0).</li> <li>■ ASN or ACS with <math>n &gt; 1</math> or <math>n &lt; -1</math>.</li> <li>■ ON...GOTO/GOSUB parameter out of range.</li> </ul>   |
| 12           | LOG(0)              | Logarithm of zero.   |
| 13           | LOG(—)              | Logarithm of a negative number.  |
| 14           |                     | Not used.  |
| 15           | SYSTEM              | System error; attempt to save the current program in a new file. Report the error by contacting your dealer, sales representative, or Response Center.*  |
| 16           | CONTINUE BEFORE RUN | Program not allocated:<br><ul style="list-style-type: none"> <li>■ Program or subprogram was not allocated before executing CONT.</li> <li>■ The current (sub) program has been changed (deallocated) since the program was paused.</li> </ul>   |
| 17 and 18    |                     | Not used.  |
| 19           | MEM OVFL            | Memory overflow:<br><ul style="list-style-type: none"> <li>■ Attempting to initialize a program that requires more than existing memory.</li> <li>■ Attempting to load a program that requires more than existing memory.</li> <li>■ Insufficient memory to dynamically load a binary program.</li> <li>■ Attempting an operation for which insufficient memory is available; e.g., opening a file, concatenating a string, creating an I/O buffer.</li> </ul> |
| 20 and 21    |                     | Not used.  |
| 22           | SECURED             | Attempting to violate system file security; e.g.:<br><ul style="list-style-type: none"> <li>■ Attempting to overwrite a directory.</li> <li>■ Attempting to edit, list, store, or overwrite a secured BASIC/PROG file.</li> <li>■ Attempting to open a secured BASIC/DATA file.</li> <li>■ Attempting to access a file for which system permission is denied.</li> </ul>   |

\* In addition to Error 15, BASIC provides a series of messages in the form *Basic fault number xxx* in the event of a system failure. If you receive a Basic fault message, note the fault number and the conditions leading to the failure. Then, contact your dealer, sales representative, or Response Center.

| Error Number  | Message           | Causes   |
|---------------|-------------------|--|
| 23 and 24     |                   | Not used   |
| 25            | BAD BIN LOAD      | LOADBIN operation has failed:<br><ul style="list-style-type: none"> <li>■ The specified file does not exist.</li> <li>■ The specified file is not formatted properly.</li> </ul>   |
| 26 through 29 |                   | Not used.  |
| 30            | OPTION BASE       | OPTION BASE ERROR<br><ul style="list-style-type: none"> <li>■ More than one OPTION BASE statement.</li> <li>■ OPTION BASE statement follows an array declaration.</li> <li>■ OPTION BASE parameter is not 0 or 1.</li> </ul> |
| 31            |                   | Not used.  |
| 32            | COM MISMATCH      | Common variable mismatch.  |
| 33            | DATA TYPE         | Data type mismatch:<br><ul style="list-style-type: none"> <li>■ READ variable and DATA constant do not agree.</li> <li>■ Attempting to read a string into a READ# numeric variable.</li> </ul>                               |
| 34            | NO DATA           | Insufficient data:<br><ul style="list-style-type: none"> <li>■ The DATA list has been used.</li> <li>■ RESTORE has been executed with no DATA statement</li> </ul>   |
| 35            | DIM EXIST VRBL    | Attempting to dimension a variable that has previously been explicitly or implicitly dimensioned.  |
| 36            |                   | Not used.  |
| 37            | DUP FN            | Duplicate user-defined function name.  |
| 38            | NO FN END         | A second DEF FN statement has been executed before the first function was ended with FN END.   |
| 39            | FN MISSING        | Referencing a non-existent user-defined function:<br><ul style="list-style-type: none"> <li>■ Attempting to executed FN END with no matching DEF FN.</li> <li>■ Branching to the middle of a function.</li> </ul>            |
| 40            | FN PARAM          | Illegal function parameter; function parameter mismatch.   |
| 41            |                   | Not used.  |
| 42            | RECURSIVE FN CALL | Recursive user-defined function.   |

| <b>Error Number</b> | <b>Message</b>   | <b>Causes</b>   |
|---------------------|------------------|---|
| 43                  | NUMERIC INPUT    | Numeric input is required.  |
| 44                  | TOO FEW INPUTS   | Too few inputs for INPUT or MAT INPUT.  |
| 45                  | TOO MANY INPUTS  | More items were given than were requested by INPUT.   |
| 46                  | NEXT MISSING     | FOR with no matching NEXT.  |
| 47                  | NO MATCHING FOR  | NEXT with no matching FOR.  |
| 48 and 49           |                  | Not used.   |
| 50                  | BIN PROG MISSING | Binary program could not be found in memory.  |
| 51                  | RETURN W/O GOSUB | Attempt to execute RETURN before GOSUB.   |
| 52                  | IMAGE            | Illegal IMAGE format string: <ul style="list-style-type: none"> <li>■ Unrecognized image specifier.</li> <li>■ Illegal quotation marks around format string.</li> </ul> |
| 53                  | PRINT USING      | Illegal PRINT USING: <ul style="list-style-type: none"> <li>■ Data overflows image specifier.</li> <li>■ Data type does not match image specifier.</li> </ul>           |
| 54                  | TAB              | Illegal TAB argument; default=TAB(0) (no change in position).   |
| 55                  | SUBSCRIPT        | Array subscript out of dimensioned range.   |
| 56                  | STRING OVF       | String overflow; a string is too large for the length of a string variable.   |
| 57                  | MISSING LINE     | Referencing a nonexistent line.   |
| 58 and 59           |                  | Not used.   |
| 60                  | WRITE PROTECT    | <ul style="list-style-type: none"> <li>■ The medium is write-protected.</li> <li>■ The file is secured against overwriting.</li> </ul>                                  |
| 61 and 62           |                  | Not used.   |
| 63                  | DUP NAME         | Duplicate path name for RENAME, CREATE, or COPY.  |
| 64 and 65           |                  | Not used.   |

| Error Number         | Message       | Causes  |
|----------------------|---------------|---|
| 66                   | FILE CLOSED   | Attempting to access (by READ# or PRINT#) or close a closed file.   |
| 67                   | FILE NAME     | Incorrect file name or path name: <ul style="list-style-type: none"> <li>■ File with specified path name was not found.</li> <li>■ Path name not enclosed in quotes.</li> <li>■ Attempt to purge an open file.</li> </ul>   |
| 68                   | FILE TYPE     | File type mismatch: <ul style="list-style-type: none"> <li>■ Attempt to treat a program file as a data file, or vice versa.</li> <li>■ Attempt to SECURE a file with an inappropriate security type.</li> <li>■ Attempting to MERGE or FINDPROG a non-BASIC file.</li> </ul>  |
| 69                   | RANDOM OVF    | Random overflow: <ul style="list-style-type: none"> <li>■ Attempt a READ#/PRINT# beyond the existing number of bytes in logical record with random file access.</li> <li>■ Attempt to PRINT# a string to a logical record with fewer than 4 bytes available.</li> <li>■ UNIX kernel tables are full; a new drive cannot be loaded.</li> </ul> |
| 70                   | READ          | FAILURE by MERGE or FINDPROG to access the mass storage medium.   |
| 71                   | EOF           | End-of-file; attempting to PRINT#/READ# beyond the end of the file.   |
| 72                   | RECORD        | Attempting to READ#/PRINT# to a nonexistent record.   |
| 73<br>through<br>87  |               | Not used.   |
| 88                   | BAD STATEMENT | SUB statement must be first line of subprogram.   |
| 89                   | INVALID PARAM | Invalid parameter; parameter out of range.  |
| 90                   |               | Not used.   |
| 91                   | MISSING PARAM | Missing parameter.  |
| 92<br>through<br>100 |               | Not used.   |

Error numbers over 100 often have more than one message associated with them. In cases where no error message is displayed (for example, in an ON ERROR routine), the errors can be differentiated by the module number returned by the ERROM function.

| <b>Error Number</b>          | <b>Message</b>      | <b>ERROM Number</b> | <b>Causes</b>  |
|------------------------------|---------------------|---------------------|--|
| 101<br>102<br>through<br>108 | XFR                 | 192                 | Warning; program paused with an active TRANSFER.<br>Not used.  |
| 109                          | # DIMS              | 176                 | Incorrect number of dimensions in an array.  |
| 109                          | PRGM TYPE           | 232                 | Attempting to CALL a non-subprogram file.  |
| 110                          | NOT A 3-VECTOR      | 176                 | The specified vector does not have 3 elements.   |
| 111                          | DIM MISMATCH        | 176                 | Incorrect number of array elements.  |
| 111                          | I/O OPER            | 192                 | The I/O operation is invalid for the specified interface.  |
| 111                          | RECURSIVE           | 232                 | A subprogram attempts to CALL or SCRATCHSUB itself.  |
| 112                          | DETERMINATE IS 0    | 176                 | Determinate of a matrix is 0.  |
| 113                          | DIM SIZE            | 176                 | Dimension size:<br><ul style="list-style-type: none"> <li>■ Total number of redimension elements exceeds number originally dimensioned.</li> <li>■ Attempt to create an empty array with option base 0.</li> </ul>   |
| 113                          | INTERFACE-DEPENDENT | 192                 | Interface dependent error:<br><ul style="list-style-type: none"> <li>■ HP-IB: interface must be system controller.</li> <li>■ Serial: UART receiver overrun.</li> <li>■ BCD: port 10 is not currently available.</li> <li>■ GPIO: odd number of bytes was transferred in the 16-bit word configuration.</li> <li>■ HP-IL: take control message was ignored by the device.</li> </ul> |
| 113                          | PARAM MISMATCH      | 232                 | Mismatch between CALL and SUB parameters.  |

| <b>Error Number</b> | <b>Message</b>      | <b>ERROM Number</b> | <b>Causes</b>   |
|---------------------|---------------------|---------------------|---|
| 114                 | NOT SQUARE          | 176                 | Array is not square.  |
| 114                 | INTERFACE-DEPENDENT | 192                 | Interface-dependent error:<br><ul style="list-style-type: none"> <li>■ HP-IB, HP-IL: interface must be active controller.</li> <li>■ Serial: receiver buffer overrun.</li> <li>■ BCD: port 10 not currently available.</li> <li>■ GPIO: FHS transfer was aborted.</li> </ul>  |
| 115                 | NON-VECTOR          | 176                 | Array is not a vector.  |
| 115                 | INTERFACE-DEPENDENT | 192                 | Interface-dependent error:<br><ul style="list-style-type: none"> <li>■ HP-IB, HP-IL: interface must be addressed to talk.</li> <li>■ Serial: automatic disconnect forced.</li> <li>■ BCD: FHS transfer aborted.</li> <li>■ GPIO: configuration does not allow output enable or output operation on port A or port B.</li> </ul> |
| 115                 | SUB STMT MSG        | 232                 | SUB statement is missing in called subprogram.  |
| 116                 | INTERFACE-DEPENDENT | 192                 | Interface-dependent error:<br><ul style="list-style-type: none"> <li>■ HP-IB, HP-IL: interface must be addressed to listen.</li> <li>■ BCD: data direction mismatch has occurred.</li> <li>■ GPIO: CTL line is not in the proper state.</li> </ul>  |
| 117                 | INTERFACE-DEPENDENT | 192                 | Interface-dependent error:<br><ul style="list-style-type: none"> <li>■ HP-IB, HP-IL: interface must be non-controller.</li> <li>■ BCD: command was directed to nonexistent field.</li> </ul>  |
| 118                 | INTERFACE-DEPENDENT | 192                 | Interface-dependent error:<br><ul style="list-style-type: none"> <li>■ BCD: CTL is not in the proper state to start the operation.</li> <li>■ HP-IL: a transmission error or protocol violation occurred.</li> </ul>  |
| 119                 | INTERFACE-DEPENDENT | 192                 | Interface-dependent error:<br><ul style="list-style-type: none"> <li>■ BCD: data format does not match mode of the interface.</li> <li>■ HP-IL: addressed talker ignored the start of transmission message.</li> </ul>  |
| 120                 | NO M.S. DEVICE      | 208                 | No mass storage device is currently active.   |
| 121 through 123     |                     |                     | Not used.   |
| 124                 | ISC                 | 192                 | Illegal interface select code.  |

| <b>Error Number</b> | <b>Message</b> | <b>ERROM Number</b> | <b>Causes</b>   |
|---------------------|----------------|---------------------|---|
| 125                 | ADDR           | 192                 | Improper primary address.   |
| 125                 | VOLUME         | 208                 | The specified volume name (top-level directory) was not found.  |
| 126                 | BUFFER         | 192                 | I/O buffer problem:<br><ul style="list-style-type: none"> <li>■ Attempting to OUTPUT or TRANSFER data to a full buffer.</li> <li>■ Attempting to ENTER data from an empty buffer.</li> <li>■ The specified string variable is not a declared I/O buffer.</li> </ul> |
| 126                 | PLOTTER IS     | 1                   | The designated plotter does not respond.  |
| 126                 | MSUS           | 208                 | The specified device name was not found.  |
| 127                 | NUMBER         | 192                 | Invalid number:<br><ul style="list-style-type: none"> <li>■ Incoming character sequence is not a valid number.</li> <li>■ Number being output has exceeded the range specified by the "e" format.</li> </ul>  |
| 127                 | READ VFY       | 208                 | A read verify error has occurred.   |
| 128                 | EARLY TERM     | 192                 | Buffer was emptied before all enter fields were satisfied.  |
| 128                 | FULL           | 208                 | The directory or mass storage medium is full.   |
| 129                 | VAR TYPE       | 192                 | An ENTER variable does not match the image specified for that variable.   |
| 130                 | NO TERM        | 192                 | Required terminator was not received during ENTER.  |
| 130                 | DISC           | 208                 | Disc error:<br><ul style="list-style-type: none"> <li>■ The mass storage medium is not initialized or formatted.</li> <li>■ The mass storage device drive latch is open.</li> <li>■ The mass storage medium is damaged.</li> </ul>                                  |
| 131                 | TIMEOUT        | 192                 | An I/O timeout has occurred.  |

# 7

## Keyword Summary

---

### General Math Functions and Operators

|           |   |
|-----------|---|
| ABS       | Absolute value.                         |
| CEIL      | Smallest integer $\geq$ the argument.   |
| DIV       | Integer portion of a quotient.          |
| EPS       | Smallest machine number.                |
| EXP       | $e^x$                                   |
| FLOOR     | Largest integer $\leq$ the argument.    |
| FP        | Fractional part of the argument.        |
| INF       | Largest machine number.                 |
| INT       | Largest integer $\leq$ the argument.    |
| IP        | Integer part of a number.               |
| LET       | Variable assignment.                    |
| LGT       | Log to the base 10.                     |
| LOG       | Log to the base e.                      |
| MAX       | Larger of two values.                   |
| MIN       | Smaller of two values.                  |
| MOD       | Modulo operator; remainder of division. |
| PI        | $\pi$                                   |
| RANDOMIZE | Modifies the seed used by RND.          |
| RMD       | Remainder of division.                  |
| RND       | Random number.                          |
| SGN       | Sign of a number.                       |
| SQR       | Square root.                            |
| VAL       | Numeric equivalent of a string.         |



## **Trigonometric Functions and Operations**

|      |   |
|------|---|
| ACS  | Arccosine (in the 1st or 2nd quadrant). |
| ASN  | Arcsine (in 1st or 4th quadrant).       |
| ATN  | Arctangent in quadrants 1 or 4.         |
| ATN2 | Arctangent in quadrants 1, 2, 3, or 4.  |
| COS  | Cosine.                                 |
| COT  | Cotangent.                              |
| CSC  | Cosecant.                               |
| DEG  | Sets BASIC to degrees mode.             |
| DTR  | Converts angle in degrees to radians.   |
| GRAD | Sets BASIC to grads mode.               |
| RAD  | Sets BASIC to radians mode.             |
| RTD  | Converts angle in radians to degrees.   |
| SEC  | Secant.                                 |
| SIN  | Sine.                                   |
| TAN  | Tangent.                                |

## **Logical Operators**

|      |                                     |
|------|-------------------------------------|
| AND  | Logical and of two values.          |
| EXOR | Logical exclusive-or of two values. |
| NOT  | Logical complement of a value.      |
| OR   | Logical inclusive-or of two values. |

## Binary Functions

|         |   |
|---------|---|
| BINAND  | Bit-by-bit logical and of two values.   |
| BINCOMP | Bit-by-bit complement of a value.   |
| BINEOR  | Bit-by-bit exclusive-or of two values.  |
| BINIOR  | Bit-by-bit inclusive-or of two values.  |
| BIT     | Value of the specified bit.   |
| BTD     | Converts string containing 0's and 1's to a decimal number.                       |
| DTB\$   | Converts decimal value to a string containing its binary representation.          |
| DTH\$   | Converts decimal value to a string containing its hexadecimal representation.     |
| DTO\$   | Converts decimal value to a string containing its octal representation.           |
| HTD     | Converts a string contains digits and/or letters A through F to a decimal number. |
| OTD     | Converts a string containing digits 1 through 8 to a decimal number.              |

## String Operations

|          |   |
|----------|---|
| CHR\$    | Interprets a numeric value as a character code and returns the character. |
| FLAG\$   | Returns an 8-character string showing status of 64 flags.                 |
| HMS      | Converts a string (HH:MM:SS) to seconds.                                  |
| HMS\$    | Converts seconds to a string (HH:MM:SS).                                  |
| LEN      | Length of a string.   |
| LWC\$    | Converts all uppercase characters to lowercase.                           |
| MDY      | Converts a string (MM/DD/YYYY) to the Julian day.                         |
| MDY\$    | Converts the Julian day to a string (MM/DD/YYYY).                         |
| NUM      | Returns decimal code of first character in string.                        |
| POS      | Position of a character in a string.                                      |
| REV\$    | Returns a string in which characters are in reversed order.               |
| ROTATE\$ | Shifts characters left or right.  |
| RPT\$    | Repeats the character sequence in the string.                             |
| TRIM\$   | Removes leading and trailing blanks.                                      |
| UPC\$    | Converts all lowercase characters to uppercase.                           |
| VAL      | Returns the numeric equivalent of a string.                               |
| VAL\$    | Returns the string equivalent of a value.                                 |

## **Clock and Time Functions**

|         |  |
|---------|--|
| DATE    | Julian date (YYDDD).   |
| DATE\$  | Date in the form YY/MM/DD.                                   |
| HMS     | Converts a string (HH:MM:SS) to seconds.                     |
| HMS\$   | Converts seconds to a string (HH:MM:SS).                     |
| MDY     | Converts a string (MM/DD/YYYY) to the Julian day.            |
| MDY\$   | Converts the Julian day to a string (MM/DD/YYYY).            |
| READTIM | Number of seconds elapsed since setting a timer.             |
| TIME    | Number of seconds elapsed since midnight.                    |
| TIME\$  | Converts number of seconds past midnight to HH:MM:SS format. |

## **Program Entry and Editing**

|            |  |
|------------|--|
| AUTO       | Starts automatic line numbering.                           |
| DELETE     | Deletes program line(s).                                   |
| INIT       | Initializes the program.                                   |
| LIST       | Lists program lines to the display (CRT IS device).        |
| MERGE      | Merges a program in mass storage with one in BASIC memory. |
| PLIST      | Lists program lines to the system (PRINTER IS) printer.    |
| REN        | Renumbers program lines.                                   |
| REPLACEVAR | Changes the name of a variable throughout the program.     |
| SCAN       | Searches for all occurrences of a character string.        |
| XREF L     | Cross-references program lines.                            |
| XREF V     | Cross-references program variables.                        |

## **Debugging**

|            |   |
|------------|---|
| ERRL       | Line number of most recent error.           |
| ERRM       | Error message of most recent error.         |
| ERRN       | Error number of most recent error.          |
| ERROM      | Module number of most recent error.         |
| ERRSC      | Select code of most recent interface error. |
| NORMAL     | Stops tracing.                              |
| SINGLESTEP | Executes the current program line.          |
| TRACE      | Traces branches.                            |
| TRACE VAR  | Traces specified variables.                 |
| TRACE ALL  | Traces branching and all variables.         |

## **Variable Allocation**

|             |   |
|-------------|---|
| COM         | Reserves memory for common variables.                   |
| DIM         | Reserves memory for REAL arrays and strings.            |
| INIT        | Initializes the program.                                |
| INTEGER     | Reserves memory for INTEGER variables.                  |
| OPTION BASE | Declares lower bound of 0 or 1 for array variables..    |
| REAL        | Reserves memory for REAL variables.                     |
| SCRATCH     | Erases program, subprograms, and variables from memory. |
| SCRATCHSUB  | Erases specified subprogram from memory.                |
| SHORT       | Reserves memory for SHORT precision variables.          |

## Display Control

|                      |  |
|----------------------|--|
| ALPHA                | Displays alpha display.  |
| AREAD                | Reads contents of alpha display memory into a string variable. |
| AWRIT                | Writes value of a string variable to the alpha display.        |
| CLEAR (no parameter) | Clears the alpha display.                                      |
| CRT IS               | Declares device to receive displayed output.                   |
| CURSCOL              | Returns column location of cursor.                             |
| CURSROW              | Returns row location of the cursor.                            |
| DISP                 | Outputs items to the CRT IS device.                            |
| GCLEAR               | Clears all or portions of the graphics display.                |
| GRAPHICS             | Displays the graphics display.                                 |
| OFF CURSOR           | Turns the cursor off.  |
| ON CURSOR            | Turns the cursor on.   |
| TAB                  | Defines column position for DISP, LABEL, and PRINT.            |

## Program Control

|             |  |
|-------------|--|
| AREAD       | Reads contents of alpha display memory into a string variable. |
| AWRIT       | Writes value of a string variable to the alpha display.        |
| BEEP        | Produces an audible tone.                                      |
| CALLBIN     | Calls the specified binary entry point.                        |
| CFLAG       | Clears the specified flag.                                     |
| CHAIN       | Chains a program into memory.                                  |
| CONT        | Continues a paused program.                                    |
| CRT IS      | Designates the display device.                                 |
| CURSCOL     | Returns column location of cursor.                             |
| CURSROW     | Returns row location of the cursor.                            |
| DATA        | Specifies data items for READ.                                 |
| DEF FN      | Defines a user-defined function.                               |
| DEFAULT OFF | Turns off default for math errors.                             |
| DEFAULT ON  | Turns on default for math errors.                              |
| DISP        | Outputs items to the CRT IS device.                            |
| END         | Stops program execution.                                       |
| FLAG        | Returns status of specified flag.                              |
| FLAG\$      | Returns 8-character string showing status of 64 flags.         |
| FLIP        | Switches keyboard between BASIC and typewriter modes.          |
| FN          | User-defined function call.                                    |
| FOR...TO    | Defines the beginning of a FOR...NEXT loop.                    |
| GOSUB       | Causes branching to a subroutine.                              |
| GOTO        | Causes branching to the specified statement.                   |

## **Program Control (continued)**

|            |  |
|------------|--|
| IF...THEN  | Causes conditional branching.  |
| IMAGE      | Provides formats for DISP, PRINT, LABEL, ENTER, and OUTPUT.                        |
| INPUT      | Inputs data from the keyboard into program variables.                              |
| KEY LABEL  | Displays key labels for user-defined keys.   |
| LINPUT     | Inputs a character string from the keyboard.                                       |
| PAUSE      | Pauses the program.  |
| PRINT      | Outputs items to the PRINTER IS printer.   |
| PRINT ALL  | Sets system to print-all mode.   |
| PRINTER IS | Specifies device as the system printer.  |
| READ       | Reads items from DATA statements.  |
| REM        | Program comment.   |
| RESTORE    | Provides for reusing data statements.  |
| RETURN     | Transfers program from a subroutine to the statement following the invoking GOSUB. |
| RUN        | Begins program execution.  |
| SCRATCHBIN | Scratches the specified binary program.  |
| SFLAG      | Sets the specified flag(s).  |
| STOP       | Stops program execution.   |
| TAB        | Defines column position for DISP, LABEL, and PRINT.                                |
| WAIT       | Causes execution to wait the specified number of seconds.                          |



## **Subprogram Control**

|            |   |
|------------|---|
| CALL       | Calls a subprogram and optionally passes in parameters.         |
| DIRECTORY  | Displays a directory of the program and subprograms in memory.  |
| FINDPROG   | Makes a subprogram available for listing and editing.           |
| NPAR       | Returns the number of parameters passed into a subprogram.      |
| SCRATCHSUB | Scratches the specified subprogram.                             |
| SUB        | First statement of a subprogram; defines the formal parameters. |
| SUBEND     | Returns execution to the invoking (sub)program.                 |
| SUBEXIT    | Returns execution to the invoking (sub)program.                 |

## **Mass Storage**

|                   |   |
|-------------------|---|
| ASSIGN#           | Opens a data file.  |
| CAT               | Displays the specified directory.   |
| CHAIN             | Chains a program into BASIC memory.   |
| CHECK READ ON/OFF | Turns on and off data verification during PRINT# operations.                |
| COPY              | Copies the specified file(s).   |
| CREATE            | Creates a data file.  |
| GET               | Retrieves a text file and enters its contents into memory as program lines. |
| GLOAD             | Load a BASIC/GRAF file into the graphics display.                           |
| GSTORE            | Store the current graphics display into a BASIC/GRAF file.                  |

## **Mass Storage (continued)**

|                 |   |
|-----------------|---|
| LOAD            | Load a BASIC/PROG file.   |
| LOADBIN         | Loads the specified binary program.                                   |
| MASS STORAGE IS | Changes the current working directory                                 |
| PRINT#          | Writes items to a data file.  |
| PURGE           | Removes a BASIC file from its directory.                              |
| READ#           | Retrieves items from a data file.                                     |
| RENAME          | Changes the name of a BASIC, non-directory file.                      |
| SAVE            | Saves the program in memory as a text file.                           |
| SECURE          | Protects BASIC files against listing, editing, and being overwritten. |
| STORE           | Stores the program in memory.   |
| UNSECURE        | Removes file security previously established by SECURE.               |
| TYP             | Returns the data type of the next item in a data file.                |
| VOLUME IS       | Changes the volume label of a disc.                                   |

## **Graphics Boundaries, Scaling, and Control**

|               |   |
|---------------|---|
| CLIP          | Specifies plotting boundaries in current scale units. |
| DUMP GRAPHICS | Outputs the graphics display to the system printer.   |
| GCLEAR        | Clears all or portions of the graphics display.       |
| GRAPHICS      | Displays the graphics display.                        |
| LIMIT         | Specifies graphics limits in millimeter units.        |
| LOCATE        | Specifies the plotting boundaries in GUs.             |
| MSCALE        | Scales the plotting area in millimeter user units.    |

## **Graphics Boundaries, Scaling, and Control (continued)**

|            |   |
|------------|---|
| PLOTTER IS | Specifies the plotting device.                                |
| RATIO      | Returns the ratio of the graphics limits—horizontal/vertical. |
| SCALE      | Scales the plotting area by the specified user units.         |
| SETGU      | Sets the system to graphics units mode.                       |
| SETUU      | Sets the system to user units mode.                           |
| SHOW       | Scales the plotting area with equal x and y user units.       |
| UNCLIP     | Sets the plotting boundaries equal to the graphics limits.    |

## **Graphics Plotting**

|          |   |
|----------|---|
| AXES     | Plots x- and y-axes.  |
| BPLOT    | Plots groups of dots on the display.  |
| BREAD    | Reads the on/off status of dots on the display.   |
| CURSOR   | Reads the location and status of the physical pen.  |
| DIGITIZE | Halts program execution until the physical pen position and status is entered from the plotting device. |
| DRAW     | Draws a line to the specified point.  |
| FRAME    | Draws a frame around the plotting area.   |
| GCLEAR   | Clears all or portions of the graphics display.   |
| GRID     | Draws grid lines.   |
| IDRAW    | Draws a line incrementally to the specified point.  |
| IMOVE    | Lifts the pen and moves it incrementally to the specified point.  |
| IPLOT    | Moves the pen incrementally to the specified point with pen control.                                    |

## **Graphics Plotting (continued)**

|           |   |
|-----------|---|
| LAXES     | Draws and labels x- and y-axes.   |
| LGRID     | Draws and labels a grid.  |
| LINE TYPE | Specifies the line type used for lines, axes, and grids.                          |
| MOVE      | Lifts the pen and moves it to the specified point.                                |
| PDIR      | Establishes plotting direction for relative and incremental plotting.             |
| PEN       | Specifies the pen number.   |
| PENUP     | Lifts the pen.  |
| PLOT      | Moves the pen to the specified point with pen control.                            |
| RPLOT     | Moves the pen with pen control to a point specified relative to a movable origin. |
| WHERE     | Assigns the pen logical position to variables.                                    |
| XAXIS     | Draws an x-axis.  |
| YAXIS     | Draws a y-axis  |

## **Graphics Labeling**

|       |  |
|-------|--|
| CSIZE | Establishes character size and shape for labels.                     |
| FXD   | Formats labels for LAXES and LGRID.                                  |
| LABEL | Plots a label at the current pen position.                           |
| LAXES | Draw and labels x- and y-axes.                                       |
| LDIR  | Specifies label direction.   |
| LGRID | Draws and labels a grid.   |
| LORG  | Defines the position of labels relative to the current pen position. |

## **Event-Initiated Branching**

|             |  |
|-------------|--|
| ON ERROR    | Establishes an event-initiated branch to be taken when an error occurs.                          |
| OFF ERROR   | Cancels ON ERROR branching.  |
| ON EOT      | Establishes an end-of-line branch to be taken when a transfer terminates.                        |
| OFF EOT     | Cancels ON EOT branching.  |
| ENABLE INTR | Enables and disables interrupts specified by ON INTR.  |
| ON INTR     | Establishes end-of-line branching for interrupts at the specified interface.                     |
| OFF INTR    | Cancels ON INTR branching for the specified interface..  |
| ENABLE KBD  | Enables and disables portions of the keyboard.   |
| ON KEY#     | Establishes end-of-line branching for the specified user-defined key.                            |
| OFF KEY#    | Cancels ON KEY# branching for the specified user-defined key.                                    |
| ON KYBD     | Establishes end-of-line branching for the specified key(s).                                      |
| OFF KYBD    | Cancels ON KYBD branching for the specified keys.  |
| ON TIMEOUT  | Establishes end-of-line branching for timeouts at the specified interface.                       |
| OFF TIMEOUT | Cancels ON TIMEOUT branching for the specified interface.  |
| ON TIMER#   | Establishes end-of-line branching to be taken when the designated interval elapses on the timer. |
| OFF TIMER#  | Cancels ON TIMER# branching for the specified timer.   |

## Input/Output

|                              |  |
|------------------------------|--|
| ABORTIO                      | Terminates an active transfer.   |
| ASSERT                       | Sets and clears interface control lines.   |
| ASSIGN                       | Assigns a device/file selector to a device or file.                                  |
| CLEAR (with device selector) | Clears the interface or resets the device.   |
| CONTROL                      | Writes one or more control bytes to control registers.                               |
| CONVERT                      | Establishes a conversion table for OUTPUT or ENTER data.                             |
| CRT IS                       | Designates the system display device.  |
| DISP                         | Displays the specified items.  |
| ENTER                        | Enters data from the specified buffer or device.                                     |
| HALT                         | Stops I/O operations at the specified interface.                                     |
| IMAGE                        | Defines the format for formatted (with USING) DISP, PRINT, OUTPUT, ENTER, and LABEL. |
| IOBUFFER                     | Declares a string variable an I/O buffer.  |
| LOCAL                        | Returns devices to manual control.   |
| LOCAL LOCKOUT                | Prevents an instrument from being placed under manual control.                       |
| OUTPUT                       | Outputs data to the specified buffer or device.                                      |
| PASS CONTROL                 | Passes active controller status to a device.   |
| PPOLL                        | Returns the parallel poll response byte.   |
| PRINTER IS                   | Designates the system printer.   |
| REMOTE                       | Places devices under remote control of the active controller.                        |

## **Input/Output (continued)**

|             |  |
|-------------|--|
| REQUEST     | Used by the non-active controller to send a response byte to the active controller.                    |
| RESET       | Performs a hardware reset of the interface.  |
| RESUME      | Re-enables I/O operations after they have been disabled.   |
| SEND        | Sends the specified commands or data to devices.   |
| SET I/O     | Writes a byte to a control register.   |
| SET TIMEOUT | Sets the amount of time an interface will wait to complete a handshake.                                |
| SPOLL       | Returns the serial poll response byte.   |
| STATUS      | Returns the contents of a status register.   |
| TAB         | Defines column position for DISP, OUTPUT, and PRINT output.  |
| TRANSFER    | Moves data from a buffer to a device, or from device to a buffer, using interrupt or fast handshaking. |
| TRIGGER     | Sends Group Execute Trigger to a device.   |

## **Numeric Array Functions**

|          |  |
|----------|--|
| ABSUM    | Sum of the absolute value of the elements. |
| AMAX     | Largest element.                           |
| AMAXCOL  | Column containing the largest element.     |
| AMAXROW  | Row containing the largest element.        |
| AMIN     | Smallest element.                          |
| AMINCOL  | Column containing the smallest element.    |
| AMINROW  | Row containing the smallest element.       |
| CNORM    | Column norm.                               |
| CNORMCOL | Column containing the column norm.         |

## **Numeric Array Functions (continued)**

|          |   |
|----------|---|
| DET      | Determinant of a matrix.  |
| DETL     | Determinant of last matrix specified in MAT...INV or MAT...SYS. |
| DOT      | Dot product of two vectors.                                     |
| FNORM    | Euclidean (Frobenius) norm.                                     |
| LBND     | The lower bound (option base).                                  |
| MAXAB    | Largest absolute value.   |
| MAXABCOL | Column number of element with largest absolute value.           |
| MAXABROW | Row number of element with largest absolute value.              |
| RNORM    | Row norm.   |
| RNORMROW | Row containing the row norm.                                    |
| SUM      | Sum of the elements.  |
| UBND     | Upper bound of a subscript.                                     |

## **Numeric Array Operations**

|           |  |
|-----------|--|
| MAT=      | Arithmetic and scalar operations;<br>Matrix multiplication;<br>Array initialization;<br>Computation of identity, inverse, and transpose;<br>Copying arrays;<br>Solving linear equations;<br>Cross product. |
| MAT DISP  | Displays elements of the specified array(s).   |
| MAT INPUT | Inputs values into the specified array(s).   |
| MAT PRINT | Prints elements of the specified array(s).   |
| MAT READ  | Reads DATA statement items and enters them into the specified array(s).  |
| REDIM     | Redimension an array.  |





# Reader Comment Card

We welcome your evaluation of this manual. Your comments and suggestions help us improve our publications.

**Manual Title and/or Part number** \_\_\_\_\_

**Please circle one number for each of the statements below.**

|  | Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree |
|--|----------------|-------|---------|----------|-------------------|
| ■ The manual is well organized.                            | 1              | 2     | 3       | 4        | 5                 |
| ■ I can find the information I want.                       | 1              | 2     | 3       | 4        | 5                 |
| ■ The information in the manual is accurate.               | 1              | 2     | 3       | 4        | 5                 |
| ■ I can easily understand the instructions and procedures. | 1              | 2     | 3       | 4        | 5                 |
| ■ The manual is clearly written.                           | 1              | 2     | 3       | 4        | 5                 |
| ■ The manual contains enough examples.                     | 1              | 2     | 3       | 4        | 5                 |
| ■ The examples are appropriate and helpful.                | 1              | 2     | 3       | 4        | 5                 |
| ■ The layout and format are attractive and useful.         | 1              | 2     | 3       | 4        | 5                 |
| ■ The illustrations are clear and helpful.                 | 1              | 2     | 3       | 4        | 5                 |
| ■ I like the use of color in the manual.                   | 1              | 2     | 3       | 4        | 5                 |

Please feel free to write additional comments, particularly if you disagree with a statement above. Use additional pages if you wish—the more detailed your comments, the more useful they are to us.

**Comments:** \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Name: \_\_\_\_\_

Title: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_

City/State/Zip: \_\_\_\_\_

Phone: \_\_\_\_\_



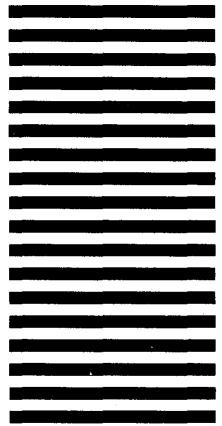


NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**  
FIRST CLASS    PERMIT NO. 33    CORVALLIS, OR

POSTAGE WILL BE PAID BY ADDRESSEE

Hewlett-Packard Company  
ATTN: Documentation Department  
Portable Computer Division  
1000 N.E. Circle Blvd.  
Corvallis, OR 97330, U.S.A.



.....  
Fold here

Tape

Please do not staple

Tape