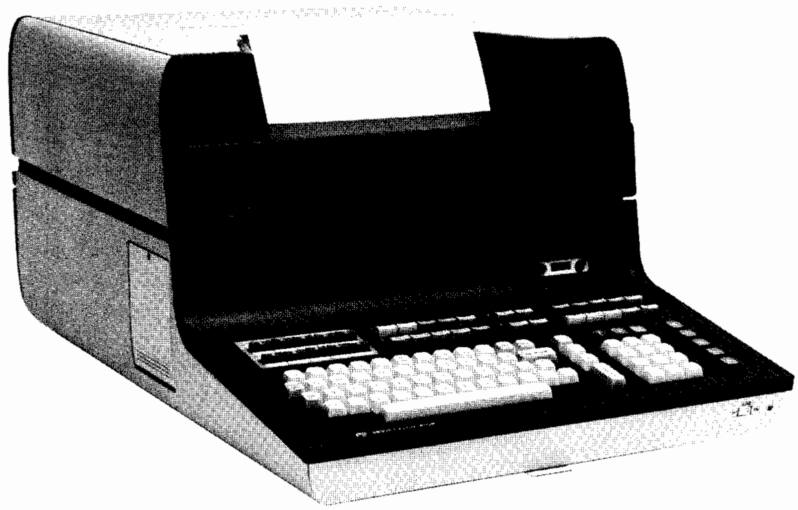
HEWLETT-PACKARD 9830A CALCULATOR
11272B EXTENDED I/O ROM

OPERATING MANUAL

OPERATING MANUAL

EXTENDED I/O ROM
11272B & OPTION 272



9830A CALCULATOR SHOWN WITH 9866A PRINTER

HEWLETT–PACKARD CALCULATOR PRODUCTS DIVISION
P.O. Box 301, Loveland, Colorado 80537, Tel. (303) 667-5000
(For World-wide Sales and Service Offices see rear of manual.)

TABLE OF CONTENTS

APPENDIX

PREFACE

The Extended I/O Read-Only-Memory (ROM) can be purchased as an accessory plug-in block or as an internal modification to the HP 9830A Calculator.

The Plug-in Version:

The 11272B Extended I/O ROM block is installable by the user. It plugs into any of the five slots behind the ROM door on the left side of the calculator.

The Calculator Modification:

The Option 272 Extended I/O ROM must be installed by qualified HP personnel. When it is installed, a decal showing the option number (Option 272) is attached to the inside of the ROM door.

Should you wish to add the option after you have received your calculator please order accessory number HP 11272F from the sales office nearest to you (see the back of this manual). The Option 272 will then be installed for you by our field personnel.

Once either version of the ROM (the plug-in block or the internal modification) has been installed, the operation is identical. Therefore, this manual makes no further distinction between the two types of ROM.

CHAPTER 1

GENERAL INFORMATION

INTRODUCTORY DESCRIPTION

The Extended I/O Read-Only-Memory (the Extended I/O ROM) provides additional statements to the HP 9830A's BASIC language, allowing a wide variety of devices to be attached to the calculator. Although the calculator recognizes ASCII* code, the Extended I/O ROM enables other codes to be input and output.

With access to data from a wide range of measuring devices, the calculator can be programmed to set experimental or test conditions and therefore can control research and production testing or sampling procedures.

Devices such as voltmeters, counters, teletypewriters, photoreaders and graphic plotters can be connected to the calculator. Direct conversion is provided at input from non-ASCII to ASCII code, and at output from ASCII to non-ASCII code. In addition, internal conversion from ASCII code is possible if the String Variables ROM is also installed in the calculator. Special functions are included for codes which cannot be immediately converted to ASCII code, such as codes of more than one byte length. Special functions may also be used for direct device-to-device transfer, for determining device status and for communication with the HP-IB (HP-Interface Bus).

SUPPLIED EQUIPMENT

One Operating Manual, HP Part Number 09830-90029, is supplied with the Extended I/O ROM.

INSPECTION PROCEDURE

Refer to the Complete Memory Test section in the 9830A Calculator System Test Instruction Booklet (HP Part No. 09830-90027) to verify operation of the ROM.

*American Standard Code for Information Exchange.

INSTALLING THE PLUG-IN ROM

The complete procedure to install a plug-in ROM is in the Operating and Programming Manual for the 9830A Calculator. Following are some reminders:

- The ROM can be installed in any of the five ROM slots.
- Switch the calculator off before installing or removing a ROM.
- The label on the ROM should be right-side-up and facing the ROM door when the ROM is properly installed.
- Ensure that the ROM is properly mated to the connector at the back of the slot before switching the calculator on.

OTHER REQUIREMENTS

It is assumed that you are familiar with the operating procedures and with the BASIC programming statements used with the 9830A Calculator. It is also assumed that you are familiar with the equipment which you plan to connect to the calculator, as outlined in the manuals supplied with that equipment.

CHAPTER 2

INTERFACE INFORMATION

INTERFACING THE HP 9830A CALCULATOR

The general I/O scheme for interfacing devices with the 9830A is shown in Figure 2-1. The four I/O connectors shown allow four peripheral devices to be connected to the calculator at one time.[1]

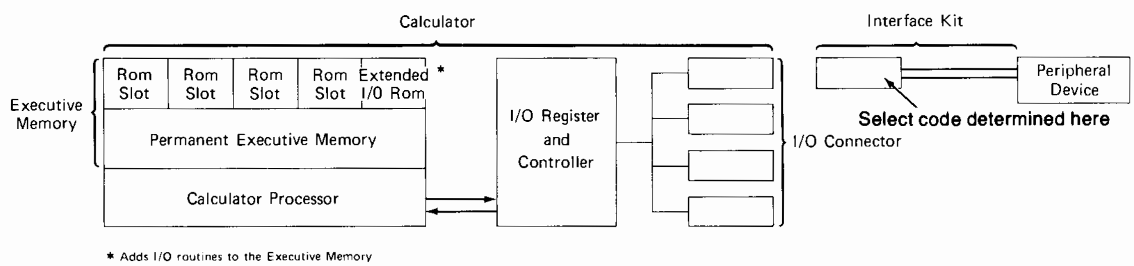


Figure 2-1. The 9830A Calculator and the I/O Scheme

Each device must be connected to the calculator with the appropriate interface card and cable. The interface card provides necessary electrical interface (signal conditioning, buffering, etc.) between the device and the calculator. Peripheral devices dedicated to the 9800 Series Calculators are supplied with all required interfacing hardware.

[1]Use of the HP 9868A I/O Expander permits up to thirteen devices to be connected to the calculator at one time.

INTERFACE KITS

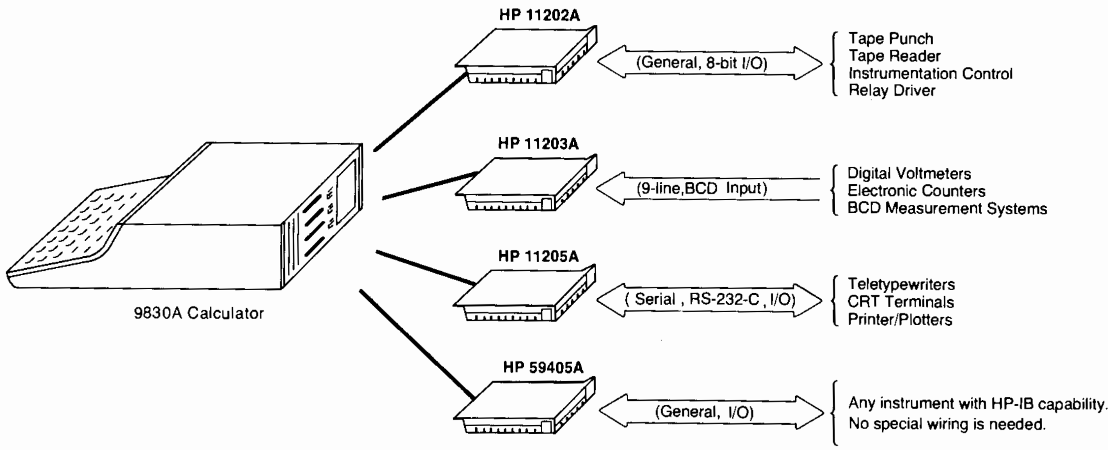


Figure 2-2. Interface Kits

The interface kits shown above allow non-9800 series peripherals to be interfaced to the 9830A. These kits are supplied with detailed hardware related information (electrical specifications, recommended interface circuits, etc.) enabling you either to interface your device directly to the 9830A Calculator or to build the necessary additional hardware required to interface the device. With the proper use of one of these interface kits, the operations described in this manual will enable the calculator to control your peripheral device. The calculator will then be able to send data to, or receive data from, your peripheral device, as appropriate.

A guide is available with detailed information on interfacing called "Selecting an HP Calculator Interface," HP Part No. 5951-7161. The HP Interface Bus makes it possible to plug-in up to 14 of the many HP-IB compatible peripherals to the 9830A. See the "Hewlett-Packard Interface Bus Users Guide", HP Part No. 59300-90002, for details.

PERIPHERAL SELECT CODES

Because several peripheral devices may be connected to the calculator at once, each device must have a unique address so that the calculator can specify the device which should respond to each operation. This address (or select code) consists of a one-digit or two-digit number and is determined on the interface card. The select codes for dedicated 9800 Series Peripherals are preset at the factory (e.g., the plotter's select code is set to 14); however, most of them can be changed, if desired, by the user. Interface kits contain switches permitting the user to set any one of nine select codes. A list of select code assignments is given in the Appendix.

Each peripheral I/O operation must specify a select code, so that the correct device responds to the operation, while all other devices ignore it.

INPUT-OUTPUT CODE

The I/O ROM enables the 9830A Calculator to send and receive data and to send commands using standard ASCII code. Conversion from non-ASCII code to ASCII code at input, and from ASCII code to non-ASCII code at output can be programmed. ASCII equivalent codes are given in the Appendix.

For input and output operations, all information exchange between the calculator and a peripheral device consists of data being transferred character-by-character in an 8-bit parallel fashion. The calculator generally sends and receives one 8-bit character at a time; if another character is to be sent or received, the calculator will wait until the device is ready.

SOFTWARE INTERRUPT

There is no provision for system interrupt operation when using the Extended I/O ROM (i.e., the peripheral device cannot initiate or call for an input or output operation). But, the status of a peripheral can be monitored using the STAT command (see Chapter 3) allowing a software interrupt. The calculator must be in complete control of each peripheral device while the device is involved in I/O operations. As mentioned earlier, if the device is not ready, the calculator will wait until the device is ready. However, the calculator can be taken out of the wait status by pressing STOP.

NOTES

CHAPTER 3

I/O OPERATIONS

This chapter describes the statements and functions available with the Extended I/O ROM. Applications and examples are shown in later chapters.

USAGE AND SYNTAX

The statements and functions provided in the Extended I/O ROM can be executed both in calculator mode and in program mode.

The following conventions are used throughout the remainder of this manual. The I/O statement syntax parameters are explained in Table 3-1. Square brackets are used in syntax descriptions to enclose the optional parameters. Colored words and symbols must appear in the program exactly as written in the syntax.

Table 3-1. Parameters Used in I/O Statements

Parameter	Explanation
select code	A numerical code, from 1 to 15, representing the address of the peripheral device (see the Appendix for a list of peripheral select codes).
string name	A single alphabetic character followed by a dollar sign ($); also, substring parameters can be specified, such as A$(5,10). Valid only when the 11274B String Variables ROM is installed in the calculator.
format	To reference a FORMAT statement, the line number of that statement; for free format data, an asterisk (*).
conversion table	The array name given to a conversion table. See 'Conversion Tables' later in this chapter.
variable list	A series of simple variables and arrays. Also, may include string names if the String Variables ROM is installed.
list	A series of variables, literals, expressions or numbers separated by commas.
FOR parameter (implied FOR...NEXT loop)	Input multiple data items from one record into an array.

THE ENTER STATEMENT

The ENTER statement enables the calculator to receive data from an external device. If the incoming data is not in ASCII code, conversion to ASCII code may be included in the ENTER statement through the use of an optional conversion table parameter described on page 3-11. Also, optional FOR parameters may be used to input multiple data items from one record into an array. A record is a sequence of characters ending with a line feed (LF).

Syntax:

ENTER (select code, format [, conversion table]) list

"Format" can be a line number reference to a FORMAT statement or an asterisk indicating free field format.

"List" can be simple variables, array variables, string variables, or an implied FOR...NEXT loop. The implied FOR...NEXT loop follows this form:

(FOR variable = $value_1$ TO $value_2$ [STEP $value_3$], array name (variable))

Example:

(FOR I = 1 TO 20 STEP 2, A(I)) } MAY ALSO BE NESTED TO 2 LEVELS

The ENTER statement causes the list of variables specified to be read from the device indicated by the select code. Character-by-character conversion to ASCII code is performed, if requested, until a standard ASCII line feed character (LF) is encountered.

Free-field Format

Free-field format is designated by an asterisk(*) in the ENTER Statement. Characters are entered as ASCII characters. All characters except 0 thru 9, +, −, and E are number delimiters. The line feed (LF) is the record delimiter. Spaces are ignored.

Example:

Data entered:

1.3,2-9,3,,4(LF)B1,2,3E5,+2(LF)

Program:

```
10 DIM A[5]
20 FOR I=1 TO 2
30 ENTER (1,*)(FORJ=1TO5,A[J])
40 PRINT A[1],A[2],A[3],A[4],A[5]
50 NEXT I
60 END
```

Output:

```
1.3        2.00000E-09     3        0        4
0          1               2        300000   2
```

When a + or − sign follows a number as "2-9" above, it is interpreted as exponential format. Two consecutive delimiters cause a zero to be entered as seen with the two consecutive commas in the example, and also the "LF" followed by "B".

FORMAT Specifications using the ENTER Statement

Fw.d Fixed-point format. Enters "w" number of characters with "d" number of decimal places. If the decimal point is part of the number being entered, then the "d" specification is ignored. If an "E" is in the field being entered, followed by one or two numbers, then the field is input using exponential format.

Example:

Data entered

123456E7-123987.(LF)

Program:

```
10 DIM C[4]
20 ENTER (1,30)(FORI=1TO4,C[I])
30 FORMAT 4F4.2
40 PRINT C[1],C[2],C[3],C[4]
50 END
```

Output:

```
12.34          560000000     -1.23          987
```

Notice in the example that no character delimiters are used. The "E" in the data is interpreted as exponential format; also note that the decimal point in "987." overrides the format specification.

Ew.d Exponential format. This format specification is the same as fixed-point format when entering data.

B Binary format. Each byte (8 bits) entered is stored as the decimal equivalent of a binary number, not as an ASCII character. For example, if the binary number 00100001 is entered, the decimal value 65 would be stored.

X Skip character format. This format specification causes individual characters to be skipped. For example, if four characters were to be skipped, 4X would be the format specification used.

/ Skip record format. This format specification causes all the characters between two consecutive line feeds to be skipped. A format specification of 3/, for example, would cause 3 records to be deleted.

" " Quote fields should not be used in FORMAT statements referenced by the ENTER statement. Error 84, improper format specification, will result.

If a standard ASCII left arrow (←) (decimal 95) is encountered, a backspacing operation is simulated. For example, if the input characters are 12←34, the resulting data values would be 134. If an escape (ESC)character (decimal 27) is encountered, a search is made for a line feed marking the end of the record. The ESC character is the same as the old alternate mode (ALT) character. The entire record containing ESC is ignored, and since the ENTER statement has not terminated, the next record is entered.

If the format of information to be entered from an external device is not known, the RBYTE function can be used to enter data byte-by-byte.

THE OUTPUT STATEMENT

OUTPUT is a general purpose statement used to send data or coded commands to an external device. If the external device requires non-ASCII code, conversion from ASCII to that code may be implemented through the use of the optional conversion table parameter.

Syntax:

OUTPUT(select code or string name, format [,conversion table]) list

"Format" can be a line number reference to a FORMAT statement or an asterisk for standard format. For information on the FORMAT statement, see the 9830A Operating and Programming Manual, part number 09830-90001.

"List" can be numbers, simple variables, array variables or string variables.

An automatic carriage return and line feed are output at the end of the OUTPUT statement. To suppress them, end the OUTPUT statement with a comma or semicolon.

Data can be output with the standard leading blanks suppressed. This is often required by devices such as network analyzers, digital voltmeters and others. The section on Suppressing Leading Blanks, later in this chapter, explains this subject.

Outputting into a String Variable

If a string variable[1] is specified instead of the select code, strings and variables can be output into a string variable. This is useful for generating characters which normally cannot be stored in the string directly. The conversion table parameter cannot be specified when outputting into a string variable.

Example:

The following program stores all of the values between 1 and 255 into the string variable, A$. Then, the string is printed on the calculator, 32 characters at a time.

Program listing:

```
10 DIM A$[255]
20 FOR I=1 TO 255
30 OUTPUT (A$[I,I],40)I;
40 FORMAT B
50 NEXT I
```

Output binary codes into A$, one character at a time.

```
60 I=1
70 OUTPUT (15,*)A$[I,I+31]
80 I=I+32
90 IF I+32<255 THEN 70
110 OUTPUT (15,*)A$[I,255]
120 END
```

Output A$, 32 characters at a time, on the 9866A printer.

[1]Valid only when the 11274B String Variables ROM is installed in the calculator.

Output:

```
*
!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑_@     } section 1
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑

!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑_@     } section 2
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑
```

Note that many characters are missing. This is because many ASCII characters are special characters. The 9866A printer does not have any characters corresponding to these special characters. Also, the output has two identical sections. This is because ASCII is an 8-bit code, but only 7 bits are data bits; the eighth bit is parity. Since there are 7 bits, there are 2^7 or 128 possible characters. Since the parity bit is ignored, the same characters are printed twice.

A different output device could be specified in lines 70 and 110 where the eighth bit is used for data. If that device uses the eighth bit, then up to 256 characters would be available.

THE CMD (COMMAND) STATEMENT

The CMD statement is a special output instruction used only to control instruments connected to the HP Interface Bus (HP-IB) at select code 13. For complete information, refer to the HP-IB User's Guide, HP Part No. 59300-90002.

I/O FUNCTIONS

Seven I/O functions allow data to be input, output, or manipulated on the bit level. Functions cannot stand alone with a line number; they return a value to an expression.

THE STAT (STATUS) FUNCTION

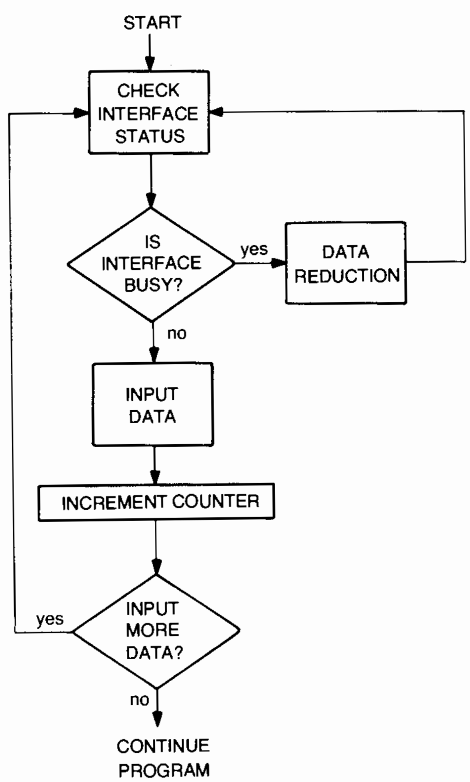
The STAT function interrogates the status of a peripheral. The result can be a value from 0 to 15, inclusive.

Syntax:

STAT select code

Generally, the status of a device is 1 when it is ready and 0 when it is busy. Status codes for most 9800 Series peripherals are listed in the Appendix. For the 9866A printer, the status remains at a fixed value once it is set, even though the peripheral may subsequently be turned off.

Although there is no provision for hard-wired peripheral interrupt operation, this flowchart and program show one method to provide software interrupt using the STAT function. In the following program, the calculator monitors an 11203A BCD Interface connected to a measurement device. When the interface is found to be busy (status bit is 0), the calculator performs some time-consuming data reduction. When the interface is ready (status bit is 1) the calculator goes thru the data-input routine.



Program:

```
110 FOR I=1 TO 20
120 IF STAT1=0 THEN 160
130 ENTER (1,*)A,B
140 NEXT I
150 GOTO 300
```

- •
- • Data Reduction Routine
- •

```
260 GOTO 120
```

THE ROTATE (ROT) FUNCTION

The ROTate function right-rotates a 16-bit integer value a specified number of places.

Syntax:

ROT (value, number of places)

The value is rotated the number of places specified. No bits are lost in the rotation. The usable range for value is between −32767 and +32767. Although the number of places rotated can be greater than 15, the actual rotation is from 0 to 15 locations.

Example:

```
10 A=5                               A = 0000 0000 0000 0101
20 B=ROT(A,2)                        B = 0100 0000 0000 0001
30 PRINT "ROT("A",2) = "B
40 END


ROT( 5     ,2) =   16385             result from line 30
```

THE INCLUSIVE OR (INOR) FUNCTION

The INclusive OR function performs a logic OR operation on the 16-bit binary equivalent of two decimal values.

Syntax:

INOR ($value_1$, $value_2$)

The two values have a usable range of integers from +32767 to −32767. The truth table for the OR function is:

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Example:

```
10 A=12
20 B=11
30 C=INOR(A,B)
40 PRINT A"OR"B" = "C
50 END
```

A = 0000 0000 0000 1100
B = 0000 0000 0000 1011
C = 0000 0000 0000 1111

```
 12   OR  11     =   15
```

result from line 40

THE BINARY AND (BIAND) FUNCTION

The BInary AND function performs the logic AND operation on the 16-bit binary equivalent of two decimal values.

Syntax:

BIAND ($value_1$, $value_2$)

Both values have a usable range of integers from +32767 to −32767. The truth table for the AND function is:

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Example:

```
10 A=20
20 B=24
30 C=BIAND(A;B)
40 PRINT A"AND"B" = "C
50 END
```

A = 0000 0000 0001 0100
B = 0000 0000 0001 1000
C = 0000 0000 0001 0000

```
 20   AND 24    =  16
```

result from line 40

THE WRITE BYTE (WBYTE) FUNCTION

The WBYTE function is used in the WRITE, PRINT, or OUTPUT statements to output a single 8-bit binary equivalent of a decimal value.

Syntax:

WBYTE value

The WRITE statement using FORMAT B is nearly the same as the WBYTE function, except that the final carriage return/line feed can be suppressed with the WRITE statement using FORMAT B and a comma or semicolon (, or ;) terminating the WRITE list.

Both examples print ASCII characters A through Z:

Program:

Automatically gives final CR/LF

```
10 FOR I=65 TO 90
20 PRINT WBYTEI;
30 NEXT I
40 END
```

; has no effect

Output:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Does not give final CR/LF

```
10 FORMAT B
20 FOR I=65 TO 90
30 WRITE (15,10)I;
40 NEXT I
50 END
```

Both required to suppress CR/LF (FORMAT B on line 10 and the ; on line 30)

There is no output.

THE READ BYTE (RBYTE) FUNCTION

The Read BYTE function is used to input 8 bits (1 byte) of data from an input device. This function is often used in the assignment statement (i.e., A = RBYTE 2) although it can be used in other statements.

Syntax:

RBYTE select code

Using this function in an assignment statement stores the decimal equivalent of the 8-bit value input from the select code specified.

Example:

```
10 A=RBYTE2      Stores the value from S.C.2 in A
20 PRINT A
30 GOTO 10
```

THE SPACE (SPA) FUNCTION

The SPAce function advances the printer or typewriter carriage the number of spaces specified, up to 71 spaces. Unlike the TAB function, which tabs to a specified column, the SPA function spaces from a previous output column the number of spaces specified.

Syntax:

SPA value

The space function works with the PRINT statement and the WRITE or OUTPUT statements when standard format (using an asterisk) is used [e.g., WRITE (15,*) SPA A"*"]. One space is given for the sign of the value.

Example:

```
10 FOR I=0 TO 5
20 PRINT SPAI;I
30 NEXT I
40 END
```

```
 0
  1
   2
    3
     4
      5
```

CONVERSION TABLES

A conversion table can be used in the ENTER and OUTPUT statements to convert groups of characters, or arrays of data from one code to another. As mentioned earlier, the calculator makes use of standard ASCII codes. Throughout this manual all non-ASCII codes (used by printers, card readers, paper tape readers, punches, typewriters, etc.) are referred to as foreign codes.

To create a conversion table, you have to store the codes in an integer array. The subscripts of the array correspond to the decimal equivalents of the characters in the foreign code. The decimal equivalents of the required ASCII characters are then stored in the appropriate array elements. The following is a simplified explanation of how a conversion table works.

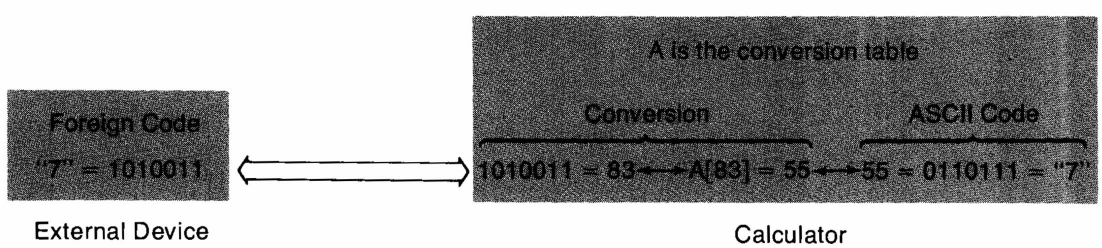
Assume that only the character "7" is to be input and output and that the external device uses some arbitrary foreign code. Suppose that in the foreign code the character "7" is represented by the binary number 1010011. Table x-x shows that 1010011 is equivalent to decimal 83, so 83 is used to determine the array element:

A[83]

In ASCII "7" is represented by the binary number 0110111; this is the bit-pattern which the calculator normally outputs (or receives) for the character "7". Table A-1 shows that 0110111 is equivalent to decimal 55, so you store the value 55 in the integer array, in the element to be determined by the foreign code as in the assignment statement:

A[83] = 55

As can be seen from the diagram below, when the external device inputs a "7" it sends bit-pattern 1010011. The calculator interprets this as 83, so it looks at element number 83 in the array and finds decimal 55; in the calculator the equivalent bit-pattern is 0110111 (see Table A-1), which is interpreted as the ASCII character "7". Conversely, when the calculator is to output the character "7," it looks for 55 in the array and finds it in the 83rd element. So the calculator outputs the bit-pattern for 83, which is 1010011. The external device interprets that bit-pattern as the character "7".



A conversion table must first be defined in a DIM statement as an integer array. For example,

```
10 DIM AI[150]
```

If you have a conversion table with more than 256 elements, you can use a two-dimensional array. Conversion tables using two-dimensional arrays are explained later in this chapter.

CONVERSION TABLES
(continued)

Conversion in the ENTER statement

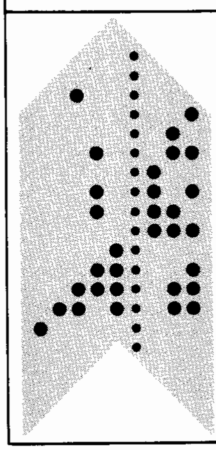
When a conversion table is referenced in an ENTER statement, the incoming foreign character-codes are used as subscripts to look up the ASCII equivalents in the conversion table. If a subscript is referenced which is not defined in a conversion table, ERROR 86 is displayed. Also, it is important that the line feed character be defined in the conversion table. For example A(108) = 10 would define the foreign code 108 as the ASCII line feed character 10.

Conversion in the OUTPUT statement

When the conversion table is referenced in an OUTPUT statement, the contents of the conversion table are searched sequentially for the outgoing ASCII character, and the subscript of the found element is the foreign code to be output. If a code is not found in the conversion table, no character is output.

Example of code conversion

Suppose you have a paper tape reader which uses EIA[2] coded tape. X and Y values are contained on the tape, separated by commas. The table below shows the tape punched with allowable codes, binary representation, the character, and the decimal equivalent codes for the EIA codes contained on the tape.

Eia Coded Tape	Binary	Character	Eia Decimal Code Equivalent
	00100000	0	32
	00000001	1	1
	00000010	2	2
	00010011	3	19
	00000100	4	4
	00010101	5	21
	00010110	6	22
	00000111	7	7
	00001000	8	8
	00011001	9	25
	00111011	,	59
	01101011	.	107
	10000000	Carr. Ret.	128

From the ASCII table in the Appendix, you can write a chart showing the ASCII decimal equivalents of the EIA codes.

[2]Electronic Industries Association standard code.

Character	EIA Decimal Equivalent (Subscript)	ASCII Decimal Equivalent (Value)
0	32	48
1	1	49
2	2	50
3	19	51
4	4	52
5	21	53
6	22	54
7	7	55
8	8	56
9	25	57
,	59	44
.	107	46
delimiter	128(CR)	10(LF)

Notice that the ASCII line feed (LF) is the delimiter used by the 9830A Calculator; this corresponds to the carriage return (CR) of the EIA code.

Now, a conversion table is made directly from the preceding EIA/ASCII chart. First, dimension an integer array for the conversion table:

```
10 DIM AI[128]
```

The conversion table must be dimensioned to have at least 128 elements because 128 is the largest subscript to be used, as can be seen from the decimal equivalent column in the EIA code chart.

Next the elements of the conversion table are defined:

```
20 A[32]=48
30 A[1]=49
40 A[2]=50
50 A[19]=51
60 A[4]=52
70 A[21]=53
80 A[22]=54
90 A[7]=55
100 A[8]=56
110 A[25]=57
120 A[59]=44
130 A[107]=46
140 A[128]=10
```

With the above instructions, definition of the conversion table is complete. Many elements in array A are undefined because the table does not have as many symbols as there are spaces for elements in the array.

CONVERSION TABLES
(continued)

The following statement inputs two characters and automatically converts the code:

```
150 ENTER (9,*,A)A,B
```

Remember, if any foreign code is not defined in the conversion table, ERROR 86 will occur at the ENTER statement.

Notice that the variable A and the conversion table A can appear in the same program. (Refer to "The ENTER Statement" for more discussion of the parameters shown.) With the above statement, EIA characters for the variables A and B are read from select code 9 (the paper tape reader) and converted to ASCII code by conversion table A.

Now assume that you have to output your variables to a tape punch with select code 3. To output variables A and B again, they must first be converted back to EIA code. The following statement will do this:

```
160 OUTPUT (3,*,A)A,B
```

Note that the same conversion table, A, is used in this case for both input and output.

When a conversion table is referenced in an ENTER statement, the calculator assumes that it has a foreign code and searches the subscripts of the conversion table array to find the element containing the equivalent ASCII code. The OUTPUT statement causes the calculator to assume that is has ASCII code and to search the values in the conversion table array to find the element whose subscript corresponds to the foreign code to be output. The output search is sequential, starting at the beginning of the array. Therefore, when the conversion table is used to output foreign code, and two or more elements in the array contain the same values, only the code having the lowest-valued subscript can be output.

TWO-DIMENSIONAL CONVERSION TABLES

A conversion table with more than 256 elements can be created by using a two-dimensional array. For example, conversion from one foreign code to another foreign code, such as device to device data transfer with code conversion, might require more than 256 elements.

The following example demonstrates the use of the two-dimensional conversion table. Rather than use a large array, such as an array of dimensions (2,256), this example uses an array of dimensions (2,3), for simplification. There are five symbols in this conversion table, and the decimal codes for these symbols in PQR code (which is being entered) and in XYZ code (which is to be output) are shown below.

Symbol	PQR Decimal Code	XYZ Decimal Code
A	2	21
B	3	22
C	4	23
D	5	24
E	6	25

The PQR/XYZ conversion table is defined in the following instruction.

```
10 DIM AI[2,3]
20 A[1,1]=21
30 A[1,2]=22
40 A[1,3]=23
50 A[2,1]=24
60 A[2,2]=25
```

The reason the two-dimensional conversion table works is because, at input instead of looking for an element with a particular subscript (as was assumed in the explanation of the one-dimensional table), the calculator looks for an element with a particular position in the array. Thus, if the PQR code for "5" is input (see above array), the calculator simply goes to the fourth element in the array and finds, in this case, the value "24" in XYZ code. In other words, the calculator doesn't look for element A(2,1); it looks for the fourth element. Conversely, if a "25", in XYZ code, is to be output, the calculator starts at the beginning of the array and counts the elements until it finds the value "25". It then outputs the number, in PQR code, corresponding to the number of elements counted, in this case "6".

Another way to express the relationship between the two subscripts and the decimal code that they represent is by using a formula. For any conversion table having dimensions (M,N), the decimal code, C, represented by the subscrips (I,J) is:

$$C = (I-1)*N+J$$

SUPPRESSING LEADING BLANKS

A special FORMAT specification causes leading blanks to be suppressed when using the OUTPUT statement. The FORMAT specifications Fw.d and Ew.d are used with w equal to 1000 and d equal to the number of decimal places to the right of the decimal point.

Example:

```
10 OUTPUT (15,20)-123.456789876
20 FORMAT F1000.9
30 END
```

```
-123.456789876
```

The value for the field width w can also be 1002 to 1014 to specify the maximum field width (2 to 14). If the value being output is too large for the field, "$" will fill the field.

Examples:

Values fit in field

```
10 OUTPUT (15,20)12345,-39
20 FORMAT 2F1007.0
30 END
```

```
12345-39
```

Values are too large for field

```
10 OUTPUT (15,20)12345,-39999
20 FORMAT 2F1005.0
30 END
```

```
$$$$$$$$$$
```

There is no field overflow when using 1000 for the field width.

This FORMAT specification works only with the OUTPUT statement.

CHAPTER 4

APPLICATIONS

This chapter contains some sample applications using the Extended I/O ROM. I/O techniques such as external data transfer, parity generation and absolute tape are explained in this chapter. 9800 Series peripherals such as the 9863A Tape Reader, the 9864A Digitizer, and the 9869A Calculator Card Reader are explained in later chapters.

EXTERNAL DATA TRANSFER

You can perform direct, byte-by-byte transfer of data from an input device to an output device without code translation.

Using an input device with select code 7 and an output device with select code 9,

```
10 WRITE (9,20)RBYTE7;
20 FORMAT B
30 GOTO 10
```

this program reads a byte of data from select code 7 and outputs the byte to select code 9 using binary format.

PARITY GENERATION

The following instructions create a parity conversion table which generates output with even parity.

```
10 DIM A[256]
20 FOR I=1 TO 256
30 A[I]=0
40 NEXT I
50 FOR I=10 TO 127
60 D=0  ←———— initialize bit counter
70 FOR J=1 TO 7
80 IF ROT(I,J)>0 THEN 100 ←— checks for one in most significant bit (neg. no)
90 D=D+1 ←———— add 1 to bit counter
100 NEXT J
110 IF BIAND(D,1)#0 THEN 140 check if D is odd or even
120 A[I]=I ←———— I has even parity
130 GOTO 150
140 A[128+I]=I←———— I has odd parity
150 NEXT I
```

PARITY GENERATION
(continued)

To use the conversion table, you would include an instruction such as the following:

```
160 OUTPUT (15,*,A)"ABCDEF12345!#$%"
```

The following instructions can be used to obtain a printout of the values in the parity conversion table.

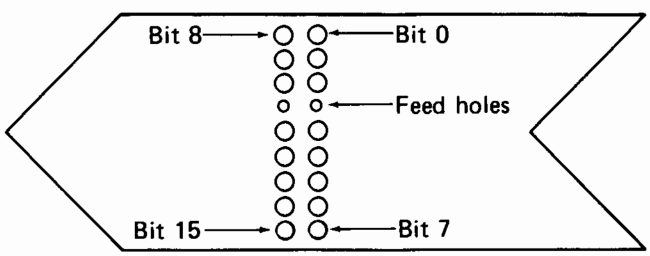
```
170 FOR I=1 TO 255 STEP 4
180 FOR J=I TO I+3
190 PRINT "A("J")="A[J];
200 NEXT J
210 PRINT
220 NEXT I
```

To create a conversion table for odd parity, statement 110 above should be changed to:

```
110 IF BIAND(D,1)=0 THEN 140
```

ABSOLUTE TAPE

Absolute tape contains instructions or data in word lengths, (2 bytes of 8 bits each) where each word is represented by two frames on the paper tape as follows:



Assuming select code 9 for the tape reader, the instruction for the calculator to read one word (two frames) from the tape into X would be:

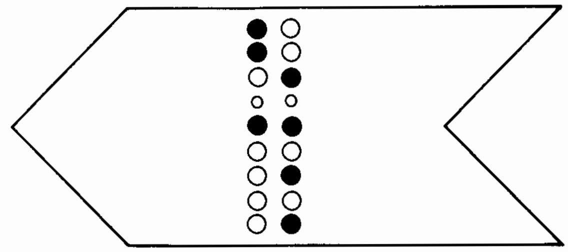
```
50 X=INOR(ROT(RBYTE9,8),RBYTE9)
```

To simplify explanation of the preceding statement, let us use A and B as temporary variables and separate the operations in the above instruction into separate program steps:

```
1 B=RBYTE9
2 A=ROT(B,8)
3 B=RBYTE9
4 X=INOR(A,B)
```

For the sample data word at right, the binary representation of B after step 1 would be:

```
  BYTE      BYTE
00000000  00001011
```



A Sample Data Word

One byte is read from the tape into the low-order bits of B. The high order bits (15-8) of B are initialized to zero.

The rotation, step 2, causes the following result in A.

00001011   00000000

When the second frame is read, the following result is contained in B.

00000000   10101100

The inclusive OR logic operation in step 4 yields the final result in X.

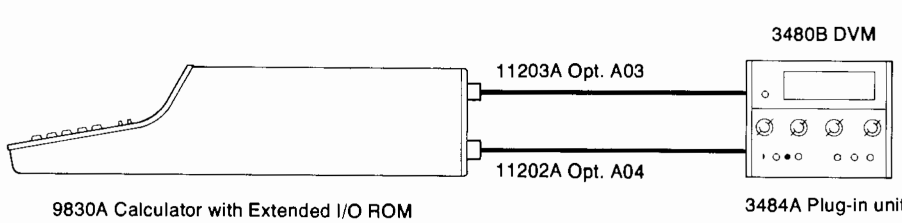
00001011   10101100

In octal code, this result is 005654; in decimal code, this is 2988.

EXTENDED I/O FOR INTERFACING

When entering data from peripherals such as voltmeters, frequency counters, and other measurement devices, the ENTER statement and RBYTE function are necessary. When outputting data to peripheral devices, the OUTPUT statement and WBYTE function are useful. The OUTPUT and WRITE statement are much alike, however, code conversion (see page 3-11) can only be done with the OUTPUT statement.

The following example illustrates the use of the ENTER statement for entering data samples from a digital voltmeter. The block diagram below shows the interface of a 9830A Calculator to an HP 3480B Digital Voltmeter and 3484A Multifunction Unit. The 11203A BCD Interface, which connects to the 3480B Digital Voltmeter, takes data from the 3480B and sends it to the calculator. The 11203A General Purpose Interface, which connects to the 3480A Plug-in unit, sends remote programming signals to the 3484A.



EXTENDED I/O FOR INTERFACING

(continued)

In the program section below, first an "@" character is sent to the 3484A to set the DC millivolt range. Then 5000 samples are taken. Line 60 sums the voltages for use in another part of the program.

The data-input rate for this program is about 21 samples per second. If a slower sampling rate is desired, a WAIT statement could be added to the program. For example, if about 10 samples per second is desired, WAIT 50 would be entered as line 55.

```
10 FORMAT B ◄────────── Dummy format used to suppress CR/LF in line 20
20 WRITE (1,10)"@";     { Sets millivolt range of 3484A plug-in unit at select code 1
                        { semicolon used to suppress CR/LF.
30 X=0 ◄─────────────── Initialize accumulator
40 FOR I=1 TO 5000
50 ENTER (2,*)A,B ◄──── Enter function code "A" and voltage sample "B"
60 X=X+B ◄───────────── Sum voltage
70 NEXT I
•
•
•
```

CHAPTER 5

9863A TAPE READER CONTROL

The HP Model 9863A Tape Reader is controlled by using the ENTER statement or RBYTE function described in Chapter 3. Numeric data can be used as input to the calculator, and, if the HP 11274B String Variables ROM is installed, alphabetic data can also be processed. Although the Extended I/O ROM is not necessary, programs can be entered from the 9863A Tape Reader.

NORMAL MODE

Although non-ASCII coded tape can be used with the 9863A Tape Reader, the examples in this section apply only to the use of ASCII coded tape. You can refer to "Conversion Tables" in Chapter 3 for information about the use of non-ASCII coded tape. Also, if you plan to use the tape reader for external data transfer, you can refer to "External Data Transfer" in Chapter 4.

Each ENTER statement causes a record to be read. A record is a sequence of data items separated by commas or semicolons, which ends with a line feed (LF). Commas and semicolons are called character delimiters; the line feed is called the record delimiter. For example, this sequence of ASCII characters is from a tape. Two records are shown:

12.8, −300, .99, 5E6 (LF) 15.0, 198, .32, 1.99E-6 (LF)

This short program reads and prints the individual data items:

Program:

```
10 DIM Y[4]
20 FOR I=1 TO 2
30 ENTER (7,*)(FORJ=1TO4,Y[J])
40 PRINT Y[1],Y[2],Y[3],Y[4]
50 NEXT I
60 END
```

Output:

```
12.8          -300          0.99          5000000
15             198          0.32          1.99000E-06
```

In the program, free-field format is used. Since commas are used to separate the data, there is no problem. But, to read a record which contains only numbers, such as:

972193425 (LF) 693345786 (LF) 667238425 (LF)

a different format specification would have to be used. If the numbers have 3 characters each, 3 numbers are contained in each record. To read the above 3 records, the following program could be used.

NORMAL MODE
(continued)

Program:

```
10 FOR I=1 TO 3
20 ENTER (7,30)A,B,C
30 FORMAT 3F3.0
40 PRINT A,B,C
50 NEXT I
60 END
```

Output:

```
972          193          425
693          345          786
667          238          425
```

In line 30, 3 characters are read; no numbers to the right of the decimal point are specified.

Using Strings

Alphabetic data can be entered if the 11274B String Variables ROM is installed in the calculator. This example shows how to enter string variable data into the calculator. The following alphanumeric characters are entered as strings:

ROGERS,ΔJOHNΔHT5-10,WT160 (LF) SMITH,ΔARTHURΔHT6-2,WT230(LF)

This program enters and prints the two records:

```
10 DIM A$[80]
20 FOR I=1 TO 2
30 ENTER (7,*)A$
40 PRINT A$
50 NEXT I
60 END
```

Output:

```
ROGERS, JOHN HT5-10,WT160
SMITH, ARTHUR HT6-2,WT230
```

The line feed (LF) is the string delimiter. Note that commas are interpreted as part of the string that is entered. To enter mixed string and numberic data, such as student names followed by grades:

ROGERS,ΔJOHNΔΔΔ80,75,83(LF)SMITH,ΔARTHURΔΔΔ90,93,88 (LF) this program can be used:

```
10 DIM A$[80]
20 PRINT "NAME"," ","AVG."
30 OUTPUT (15,40)
40 FORMAT 34"="
50 FOR I=1 TO 2
60 ENTER (7,*)A$[1,16],A,B,C
70 FIXED 0
80 PRINT A$,(A+B+C)/3
90 NEXT I
100 END
```

Output:

```
NAME                          AVG.
==================================
ROGERS, JOHN                  79
SMITH, ARTHUR                 90
```

In line 60, only 16 characters are entered into A$, then the numbers A, B, and C are entered. In line 80, the student name, A$, and the average of A, B, and C are printed.

PAPER TAPE PROGRAM STORAGE

The Extended I/O ROM is not needed to store programs on paper tape. Using the LIST# command, programs can be punched on paper tape. Then, with the 9863A tape reader, the PTAPE# command can be used to re-enter the program into the calculator. For example, store this short program in the calculator.

```
10 FOR I=1 TO 10
20 PRINT I
30 NEXT I
40 END
```

If a tape punch is at select code 2, then LIST#2 would store the program on paper tape. At a later time, the program can be re-entered into the calculator from the 9863A tape reader at select code 7. To do this, execute PTAPE# 7 when the tape reader is in NORMAL mode.

DATA MODE

The 9863A Tape Reader can be used in the data mode if a conversion table is used (see Conversion Tables in Chapter 3). A conversion table is necessary because the apostrophe (decimal 39) is sent as the end of character delimiter instead of the comma (decimal 44), and a comma (decimal 44) is sent to the calculator as the record delimiter instead of a line feed (decimal 10). The following lines generate the necessary conversion table:

```
10 DIM A[70]
20 FOR I=1 TO 70
30 A[I]=I
40 NEXT I
50 A[44]=10
```

In data mode, only numerics, plus and minus signs, and the letter E (decimal 69) for exponents are allowed. For this reason, only 70 characters are needed. In line 50, A(44) = 10 converts the decimal equivalent of a comma to the decimal equivalent of a line feed. The ENTER statement used would appear as follows:

```
60 ENTER (7,*,A)   (variables)
```

select code — free-field format — conversion table

The following example illustrates the use of the 9863A Tape Reader in data mode.

Data entered:

1.3,2-9,3,,4(LF)B1,2,3E5,+2(LF)

Program:

```
10 DIM A[70]
20 PRINT " A";" B";" C";" D";" E"
30 FOR I=1 TO 70
40 A[I]=I
50 NEXT I
60 A[44]=10
70 FOR I=1 TO 2
80 ENTER (7,*,A)A,B,C,D,E
90 PRINT A,B,C,D,E
100 NEXT I
110 END
```

Output:

```
A        B              C         D      E
1.3      2.00000E-09    3         4      0
1        2              300000    2      0
```

CHAPTER 6

9864A DIGITIZER CONTROL

The HP 9864A Digitizer is controlled by using the ENTER statement to accept data samples from the digitizer and by using the WRITE or OUTPUT statement to activate a BEEP sound. In these examples, we will assume the digitizer select code is 9.

THE "BEEP"

A WRITE or OUTPUT statement will cause the digitizer to sound its audible tone, which lasts about one-tenth of a second.

A series of these statements, when separated by WAIT or DISPLAY statements, produce a pattern of "beeps" which can be used to signal the operator during program operation.

The following statement causes a single "beep":

```
30 WRITE (9,*)
```

These program instructions produce a pattern of "beeps":

```
20 FOR I=1 TO 10
30 WRITE (9,*)
40 WAIT 100
50 NEXT I
```

DATA SAMPLES

The ENTER statement accepts X and Y coordinates from the digitizer and assigns the corresponding values to the variables specified.

```
60 ENTER (9,*)X,Y
```

When an ENTER statement is encountered, the calculator waits for a data sample to be sent from the digitizer.

If the digitizer is in the continuous mode when the ENTER statement is encountered, a data sample is sent immediately. However, if the digitizer is not in the continuous mode, then the data is not sent until either [S] or [C] (on the cursor) is pressed.

The maximum rate at which data samples can be transferred to the calculator is approximately 32 samples per second. Since the sample rate may be considerably slower because of program execution time, the operator must take care to move the cursor slowly in order to obtain the maximum practical sample density. The effects of sample rate and sample density are discussed in the Digitizer Peripheral Manual.

Example:

```
10 DIM A[256,2]
20 FOR I=1 TO 256
25 WRITE (9,*)
30 ENTER (9,*)A[I,1],A[I,2]
40 NEXT I
```

} takes approximately 13 seconds to run.

In the above example, data samples are stored in Array A. Column 1 of Array A contains X values, and column 2 contains Y values. A BEEP is added in line 25 to alert the operator that a data sample is about to be taken.

THE STOP KEY

If STOP is pressed while data samples are being taken from the digitizer, the ENTER operation is terminated and the continuous mode, if in effect, is deactivated.

THE SAMPLE DATA OVERLAY

Attach the Sample Data Overlay supplied with your digitizer to the platen (see Figure 6-1) and tape down all four corners of the overlay.

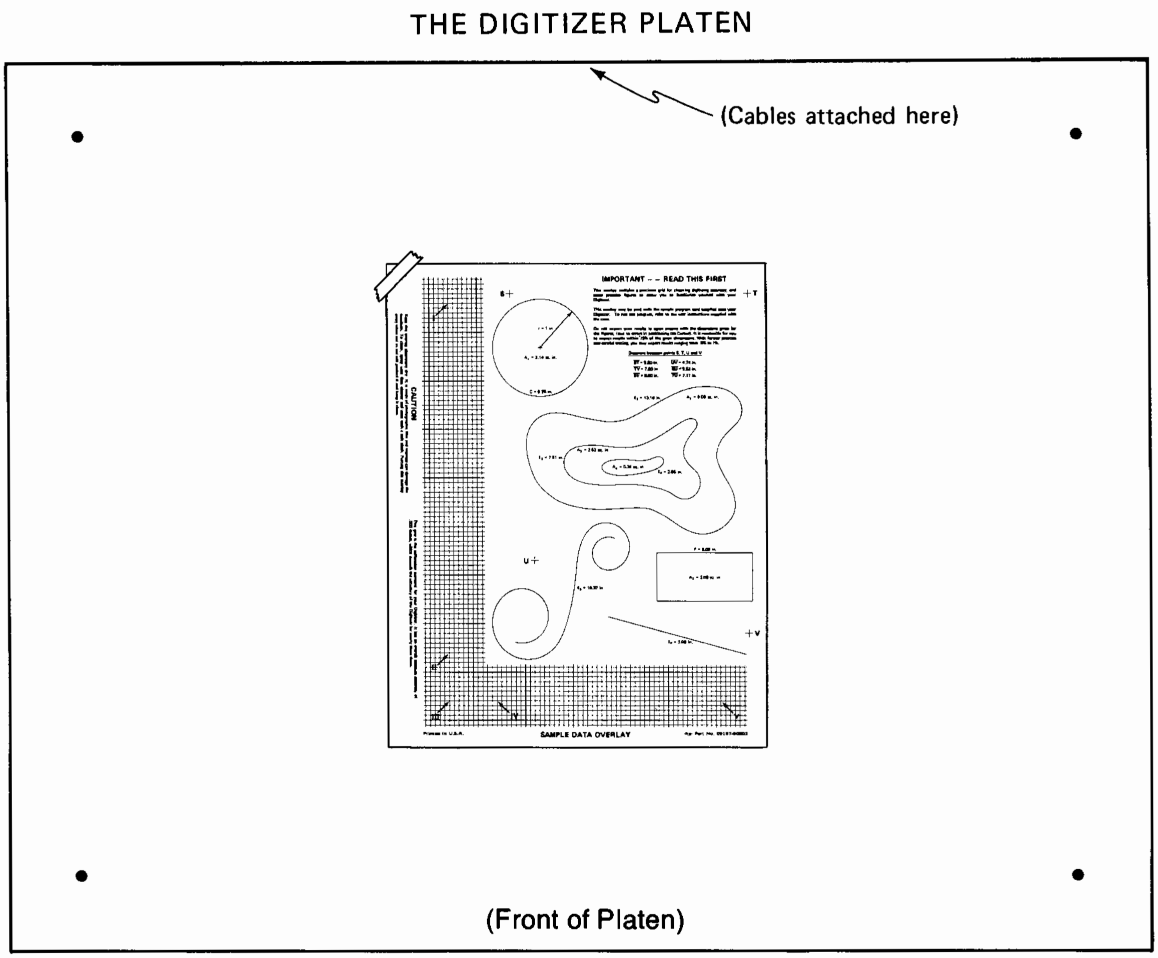


Figure 6-1. Document Alignment Procedure

To set the origin, press [O] on the cursor, with the cross-hairs over point U on the overlay.

Key in the following program:

```
10 WRITE (9,*)
20 ENTER (9,*)X,Y
30 PRINT X,Y
40 GOTO 10
```

Run the program by pressing RUN EXECUTE. The calculator is now waiting for a data sample from the digitizer. Press [S] several times; the digitizer supplies one data sample each time [S] is pressed.

To take continuous samples, first press [C] . Then slowly slide the cursor across the digitizing area (the corners of the digitizing area are indicated by the black dots on the platen). To stop continuous sampling, press [C] again.

To stop the program, press STOP.

DOCUMENT ALIGNMENT

The following program can be used to align a document on the digitizing surface.

```
10 ENTER (9,*)X,Y
20 IF X>0.1 THEN 80
30 IF X<-0.1 THEN 80
40 IF X#0 THEN 70
50 WRITE (9,*)
60 WAIT 100
70 WRITE (9,*)
80 DISP X;Y
90 GOTO 10
100 END
```

1. Attach the Sample Data Overlay (or any other document which is to be digitized) to the digitizing surface, as shown in Figure 6-1.

2. Place the cursor cross-hairs over point 1 (the upper left-hand corner of the document) and press [O] . Slide the cursor over to point 111 (the lower left-hand corner of the document) then, with the above program running in the calculator, position the cross-hairs **exactly** over point 111; press [C] . If point 111 is positioned exactly over the X axis, the digitizer will "beep" slowly. Whenever point 111 is positioned exactly over the X axis, the digitizer will "beep" more rapidly. Slowly move the cursor and the overlay (together) either right or left until the display equals .00 and the audible signal indicates alignment.

3. Without moving the overlay, tape the remaining three corners of the overlay (or document) to the platen. If necessary, retape the first corner.

4. Verify that the X axis of the document is in precise alignment with the platen by noting the display when the cross-hairs are positioned alternately over points 1 and 111 (return to step 2 if the X axis is not precisely aligned).

CHAPTER 7

9869A CARD READER CONTROL

The HP 9869A Card Reader can be used with the Extended I/O ROM. In addition to the standard features of the Extended I/O ROM, however, the Batch ROM, 11278B enables the reader to stack a number of separate programs and read educational basic data processing cards (HP 9320-2051). Also with the Batch ROM, fewer program statements are required (see the Batch Basic ROM Manual, part number 09830-90011).

The reader inputs information, which is stored on punched or marked data processing cards, to the 9830A Calculator at a rate of up to 300 cards per minute. Accessability to a key punch machine is a major factor in choosing which method of marking a card to use. See the 9869A Peripheral Manual for instructions on marking cards.

While the reader was developed primarily to read data from cards, it is possible to read program statements from cards. See "Running A Program", at the end of this chapter.

Although the reader reads the information on each data card only once before routing it to the output hopper, the data stored on one card can be used more than once because of two temporary memory storage locations called buffers. When power is switched on, the buffers are empty. They can be loaded with the first WRITE input statement (see page 7-2) and emptied with an ENTER statement, which transfers the information from the buffers to the calculator.

As mentioned in Chapter 3, I/O OPERATIONS, the ENTER statement enables the calculator to receive data from the reader. This empties the buffer automatically. If the incoming data is not in ASCII code, conversion to ASCII code may be included in the ENTER statement through the use of an optional conversion table parameter. Also, optional FOR parameters may be used to input multiple data from one data processing card into an array (see page 3-2). Each data processing card is considered a record and no line feed (LF) characters are needed to inform the calculator that a card (record) is complete.

Syntax:

ENTER (select code, format [,conversion table]) list

The ENTER statement causes the data on one card to be read from the reader. In the examples that follow, it is assumed that the reader select code is 1. See the 9869A Peripheral Manual for instructions on changing the select code. Also, see page 5-2 in this manual for information concerning the use of strings with the 9869A Card Reader.

OPERATING MODES

The reader responds to specific statement instructions by using operating modes. These modes instruct the reader how to pick and how to read cards. The card reader operates in a specified mode until it is instructed to switch modes. Each mode is represented by an alpha control character which is programmed either from the keyboard or from program cards.

The WRITE statement is used to set the operating mode of the card reader.

Syntax:

WRITE (select code,*)"alpha control character"

The following modes are represented by their respective alpha control characters:

Demand	D - to pick cards one at a time.
Continuous	C - to pick cards repeatedly.
Image	I - to transmit information from custom-made cards.
Normal	N - to transmit standard hollerith data cards.

The following functions are represented by their respective alpha control characters:

Reject/Select	J - to sort cards.
Bell	B - to ring the bell.
Abort	A - to abort the data on a card.
Stop	S - to deactivate the reader.
Retransmit	T - to re-use the data on a card.

See page 7-10 for more on the five functions above.

Demand Mode

In the DEMAND mode, cards are picked from the card deck one at a time.

The WRITE "D" statement switches the reader from any mode it is presently in to the DEMAND mode and activates the card reader. Once the card reader is activated, or ready, it remains ready. The green ready light indicates the card reader's status.

The WRITE statement is used as an input statement and the alpha control character, D, must be used to indicate the DEMAND mode.

The following statement causes the card reader to pick a single data card unless the input card hopper is empty, the output card hopper is full, or the buffer is full:

```
10 WRITE (1,*)"D"
```

In the statement above, the number 1 is the card reader's select code and the symbol, *, indicates a free-field input format. Notice that the control character, D, is written within quote marks to inform the card reader that the statement is an input instruction - in this case a DEMAND statement - not a simple WRITE output statement.

> **NOTE**
> Do not hold the SHIFT key down while pressing an alpha character key because this will result in the card reader interpreting a lower case letter. The card reader can differentiate between upper and lower case and it will ignore all statements with lower case letters. For example, in the statement above, the SHIFT key was used to enter both sets of quotation marks, but was released in order to enter the upper case D.

This statement must be executed each time a card is to be picked from the input card hopper. The card reader buffer must be empty before the WRITE "D" statement can be executed.

> **NOTE**
> The reader will try to pick a card when instructed to do so. If the reader does not sense that a card has been picked, it will try to pick two more times. If the reader fails to pick a card successfully by this time, it will become deactivated or "not ready". The status line will be set at 1 and the yellow "PICK FAIL" light will be switched on. The PICK FAIL light can be switched off by pressing the STOP key on the calculator keyboard. After removing the defective card, press the CONT key and the EXECUTE key on the calculator keyboard.

The following program causes the reader to transmit three data elements per card to the calculator. The calculator, in turn, instructs the printer to print this data after each card is read:

```
10 WRITE (1,*)"D"
20 ENTER (1,50)X,Y,Z
30 PRINT X,Y,Z
40 GOTO 10
50 FORMAT 3F10.2
60 END
```

It is necessary to instruct the reader to pick another data card after each ENTER statement (line 20) empties the buffer. This is accomplished with the GOTO statement (line 40).

> **NOTE**
> The display will show: ERROR 83 IN LINE 20 when the reader is finished reading cards (i.e., when the input hopper is empty or the output hopper is full).

OPERATING MODES
(continued)

The ERROR 83 display message can be avoided by including a test and a dummy data card for the last card read. The following program tests for the dummy data card on which the data is 1E99.

```
10 WRITE (1,*)"D"
20 ENTER (1,60)X,Y,Z
30 IF X=1E+99 THEN 70
40 PRINT X,Y,Z
50 GOTO 10
60 FORMAT 3F10.2
70 END
```

If a dummy data card is not used and additional cards are to be input, load the cards and press the CONT key and the EXECUTE key.

Continuous Mode

The CONTINUOUS mode is used to pick cards from the input card hopper continuously without being further instructed.

The WRITE (1,*)"C" statement switches the reader from any mode it is presently in to the CONTINUOUS mode and activates the reader. The green ready light indicates when the reader is activated.

The following statement causes the reader to pick data cards continuously until the input hopper is empty, the output hopper is full, or the buffer is full:

```
10 WRITE (1,*)"C"
```

The CONTINUOUS mode is faster than the DEMAND mode for inputting data. But a data card can be retransmitted, rejected, or aborted in the DEMAND mode only.

The following program causes the reader to transmit three data items on cards to the calculator. The calculator, in turn, instructs the printer to write this data:

```
10 WRITE (1,*)"C"
20 ENTER (1,50)X,Y,Z
30 PRINT X,Y,Z
40 GOTO 20
50 FORMAT 3F10.2
60 END
```

Notice that once the reader has been instructed to pick cards in the CONTINUOUS mode (line 10), there is no need to instruct it to pick individual cards again. The ENTER statement, however, must be re-executed for each card to clear the first buffer. This is accomplished by the GOTO statement (line 40).

Image Mode

In the IMAGE mode, the reader transmits the binary image of the cards in two 6-bit bytes directly to the calculator, without first converting it to ASCII code.

This mode makes it possible to input custom-made cards as well as standard hollerith cards. The number of possible combinations in this mode is 4,096 per column.

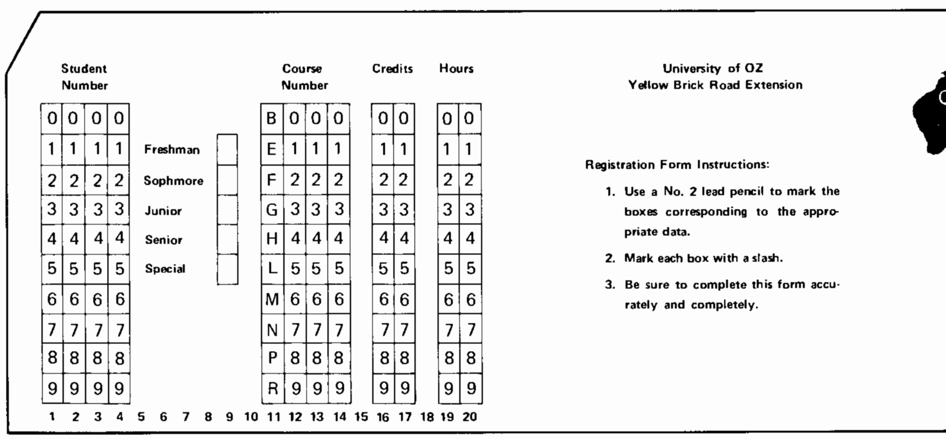
The alpha control character, I, must be used to indicate the IMAGE mode.

The following program causes the reader to switch to the IMAGE mode and pick custom-made cards one at a time.

Example:

The registrar of a small university wants to know how many hours of courses the average part-time student takes per week in one semester.

The registrar designed his own 40-column class registration cards to look like this:



The following program finds the average number of hours per week taken by part-time and full-time students. The program also lists cards in a reject column on which one column is marked more than once or on which the total number of hours per week is greater than 40. The 11274B String Variables ROM is required to run this program.

- **Initialize Program**

```
10 DIM A$[40],B$[13],C$[13],D$[12]
20 A=A0=E=F1=F2=P1=P2=0
30 B$=" +-0123456789"
40 C$="   BEFGHLMNPR"
50 D$="  FRSOJUSESP"
60 PRINT "CARDS:"
70 PRINT "REJECTED"TAB50"REPORTED"
80 PRINT
90 WRITE (1,*)"I"
```

The above program segment dimensions and defines the strings which will be used, sets the counters and flags to zero, prints headings, and sets the card reader to the IMAGE mode.

OPERATING MODES

(continued)

- **Read Card**

```
100 WRITE (1,*)"D"
110 FOR J=1 TO 20
120 X=INOR(ROT(RBYTE1,10),RBYTE1)
130 IF X=28 THEN 530
140 FOR I=11 TO 0 STEP -1
150 IF X >= 2↑I THEN 170
160 NEXT I
170 IF X=2↑I OR I=-1 THEN 220
180 A$[J,J]="$"
190 I=-1
200 E=1
210 GOTO 240
220 IF J=9 OR J=11 THEN 260
230 A$[J,J]=B$[I+2]
240 NEXT J
```

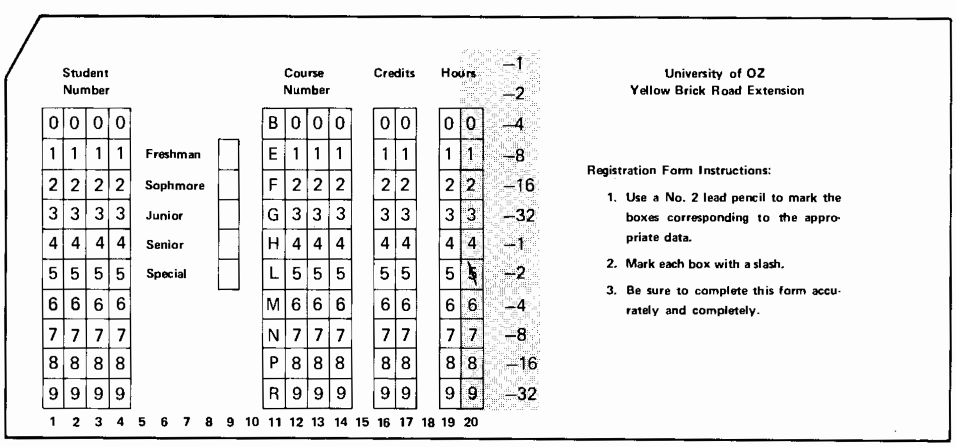
The above program segment picks a card and reads the first 20 columns, one by one. Line 120 combines two bytes of the 6 data bits to make 12 rows per column (see explanation below). It then tests whether this is the last data card (i.e., whether 0, 1, and 2 are marked in column 1), searches for a mark on each column and assigns a value from 2^0 to 2^{11} for each mark.

The program exits from the loop when a mark is found in the Jth column. The digit that matches the mark in each column is stored in the string, A$, as it is interpreted. The program will store a dollar sign ($) in the Jth column if there are two or more marks in it. If no mark is found, the error flag is set to its "on" position and variable I is set to −1 (a "no mark" value). At this point, the reader is instructed to read the card's next column. If a mark was made on the 9th or 11th column, a special character string is inserted instead of a digit.

Line 120 [X = INOR(ROT(RBYTE1,10),RBYTE1)] can be understood more easily when broken down as follows:

The reader reads two 6-bit bytes per column. The first byte consists of marks in rows 4 through 9 and the second byte includes the marks on the remaining rows in the column.

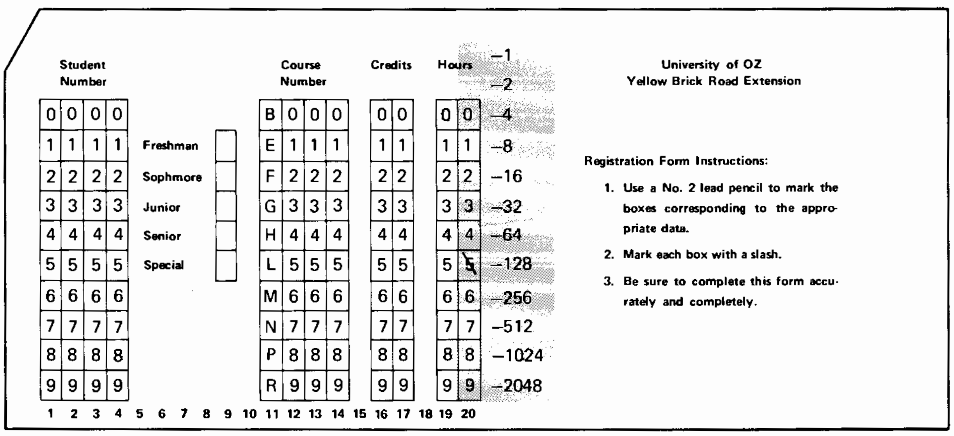
In binary code, marks in the rows of a column represent the following values:



When row 5 is marked on column 20 (above), the following intermediate results are obtained:

Operation	Resulting Intermediate Value
RBYTE1	2
ROT(2,10)	128
RBYTE1	0
INOR(128,0)	128

Notice that ROT(2,10) changes the value of the mark in row 5 from 2 to 128, as follows:



Line 120 defines X = 128 in this example. This value represents a binary code which is unique for any possible combination of marks on the column. In effect, line 120 has transformed a mark on row 5 to number 128.

OPERATING MODES

(continued)

• **Check Data**

```
250 GOTO 330
260 IF J=11 THEN 310
270 IF I>2 AND I<8 THEN 290
280 I=2
290 A$[7,9]=D$[(I-1)*2-1,(I-1)*2]
300 GOTO 240
310 A$[J,J]=C$[I+2]
320 GOTO 240
330 IF E THEN 460
340 IF A$[19,20]<"00" THEN 490
350 IF A$[19,20]>"40" THEN 490
360 PRINT TAB50,A$
370 WRITE (1,*)"A"
380 A=A+1
390 IF VAL(A$[19,20])<12 THEN 430
400 F1=F1+1
410 F2=F2+VAL(A$[19,20])
420 GOTO 100
430 P1=P1+1
440 P2=P2+VAL(A$[19,20])
450 GOTO 100
```

The above program segment instructs the reader to continue the program after the 20th column has been read. It checks for whether "Freshman," "Sophomore," "Junior," "Senior," or "Special" has been marked, sets column 11 to the proper course letter and determines whether the number of hours per week is greater than 40 or is not properly marked.

Time can be saved by instructing the reader to disregard the last 20 columns on each card since no information is included there. For this reason, an abort statement is used (lines 370, 500, 630). See Abort section, later in this chapter.

This program segment also performs statement 400 for cards which represent full-time students (12 or more hours per week) or statement 430 for part-time students. These statements keep a running total of full-time hours and part-time hours as well as the number of full-time and part-time students. After this information is stored, the reader picks another card.

• **Print Data**

```
460 PRINT "***";A$;"*** DOUBLE MARKINGS ***"
470 E=0
480 GOTO 500
490 PRINT "***";A$;"*** ERROR IN COL. 19-20 ***"
500 WRITE (1,*)"A"
510 A0=A0+1
520 GOTO 100
530 PRINT
540 PRINT "TOTAL CARDS READ=";A+A0;"REPORTED=";A
550 PRINT
560 IF  NOT P1 THEN 580
570 PRINT "PART-TIME=";P1;"HRS=";P2;"AVERAGE/STUDENT=";P2/P1
580 IF  NOT F1 THEN 600
590 PRINT "FULL-TIME=";F1;"HRS=";F2;"AVERAGE/STUDENT=";F2/F1
600 PRINT
610 PRINT "TOTAL STUDENT CLASS HOURS=";F2+P2
620 PRINT
630 WRITE (1,*)"A"
640 PRINT "DONE"
650 END
```

The above program segment instructs the printer to print a diagnostic message (either "DOUBLE MARKINGS" or "ERROR IN COL. 19-20") for rejected cards on the left side of the page and prints accepted cards on the right side of the page. the printer also records the number of cards read and accepted, the number of part-time and full-time students, the total number of hours per week for full-time and part-time students, the average number of hours per student per week and, finally, the total number of hours per week.

The reader will remain in the IMAGE mode until a NORMAL statement is received or the calculator STOP key is pressed.

Normal Mode

In the NORMAL mode, the reader converts the data on 128 character hollerith cards to ASCII code, before transmitting them to the calculator. Standard data processing cards can be used in this mode.

The reader is initialized to the NORMAL mode when the calculator STOP key is pressed or when power is turned on. The NORMAL mode can be set also with an appropriate statement.

The alpha control character, N, must be used to indicate the NORMAL mode.

The following statement causes the reader to switch to the NORMAL mode:

```
10 WRITE (1,*)"N"
```

CARD READER FUNCTIONS

Once one or two operating modes have been determined, it is possible to instruct the reader to perform a number of other functions. Each function must be re-executed each time it is to be performed.

The following functions are represented by their respective alpha control characters:

Reject/Select	J - to sort cards.
Bell	B - to ring the bell.
Abort	A - to abort the data on a card.
Stop	S - to deactivate the reader.
Retransmit	T - to re-use the data on a card.

Once again, the WRITE statement is used to indicate which function the reader is performing.

Syntax:

WRITE (select code, *) "alpha control character"

Reject/Select Function

The WRITE "J" statement is used to sort data cards and can be executed only if the reader includes Option 002, reject hopper.

REJECT/SELECT can be used only when the reader is in the DEMAND mode.

The following program causes the reader to sort all data cards on which variable X is negative or zero:

```
10 WRITE (1,*)"D"
20 ENTER (1,60)X,Y,Z
30 IF X>0 THEN 50
40 WRITE (1,*)"J"
50 GOTO 10
60 FORMAT 3F10.2
70 END
```

The display will show: ERROR 83 IN LINE 20 when the input hopper is empty or the reject hopper is full (approximately 40 cards). Unload the reject hopper and press the CONT key and the EXECUTE key on the calculator keyboard to continue the current program.

> **NOTE**
> Since the last data processing card cannot be rejected under any circumstances, a blank dummy card should be added to the end of the card deck. In this way, the last actual data card can be rejected and the dummy card will be routed to the output hopper automatically.

Bell Function

The WRITE "B" statement is used to signal the operator that a condition of the system has been reached; it can be executed only if the reader includes Option 004, bell.

The following program causes the reader to ring its bell after each group of 10 data cards has been read:

```
10 WRITE (1,*)"C"
20 FOR I=1 TO 10
30 ENTER (1,70)X,Y,Z
40 NEXT I
50 WRITE (1,*)"B"
60 GOTO 20
70 FORMAT 3F10.2
80 END
```

Abort Function

The WRITE "A" statement is used to ignore the data stored on a data card under certain conditions. See the sample program in the IMAGE section.

The ABORT function can be executed at any time in the program, but only when the reader is in the DEMAND mode.

The following program causes the reader to abort the data on a card if variable X is negative and print the data if the first data element is positive:

```
10 WRITE (1,*)"D"
20 ENTER (1,80)X,Y,Z
30 IF X<0 THEN 60
40 PRINT X,Y,Z
50 GOTO 10
60 WRITE (1,*)"A"
70 GOTO 10
80 FORMAT 3F10.2
90 END
```

NOTE

If the ABORT function is received before the contents of a card are completely read, the remaining data on the card is aborted (i.e., not transmitted). The entire aborted card's data is stored in the second buffer and can be re-used if needed. (See Retransmit section, below.)

CARD READER FUNCTIONS
(continued)

Stop Function

The WRITE "S" statement can be executed at any time in the program. It is used to deactivate the reader and clear the buffers.

In the DEMAND mode, the reader will finish transmitting the present card's data and then stop. In the CONTINUOUS mode, picking stops, but transmission of data continues until the buffer is empty.

The following program stops the reader and clears the buffers when the first negative data element appears in variable X and prints the data when variable X is equal to or greater than zero.

```
10 WRITE (1,*)"C"
20 ENTER (1,70)X,Y,Z
30 IF X<0 THEN 60
40 PRINT X,Y,Z
50 GOTO 20
60 WRITE (1,*)"S"
70 FORMAT 3F10.2
80 END
```

Retransmit Function

The WRITE "T" statement is used to enter the data from a particular card more than one time. It can re-use data on a card after the card is aborted or after other data on the same card has been used since the data remains in the second buffer until another card is read.

RETRANSMIT can be used only when the reader is in the DEMAND mode. It will re-use the data on the last card only.

The following program instructs the reader to retransmit and print the data on each card five times before reading the next data card.

```
10 WRITE (1,*)"D"
20 ENTER (1,100)X
30 FOR I=1 TO 5
40 WRITE (1,*)"T"
50 ENTER (1,*)Y
60 PRINT Y
70 NEXT I
80 GOTO 10
90 END
100 FORMAT F5.0
```

Two or more alpha control characters can be included in the same WRITE statement providing the characters do not cancel each other out. It is possible, for example, to write a program which would reject a card and simultaneously ring the bell when the first data element on that card is negative or zero by using a WRITE(1,*) "JB" statement, as follows:

```
10 WRITE (1,*)"D"
20 ENTER (1,60)X,Y,Z
30 IF X>0 THEN 50
40 WRITE (1,*)"JB"
50 GOTO 10
60 FORMAT 3F10.2
70 END
```

Conflicting alpha control characters ("IN" or "DC") cannot be contained in one WRITE statement in this manner.

RUNNING A PROGRAM

If the reader is to be used only to transfer data from cards to the calculator, the program can be input from the calculator keyboard. Be sure the data cards are stacked in the input hopper of the reader. Once the program is keyed in, press the RUN key and the EXECUTE key to pick and read the data cards and run the program.

The card reader can also be used to input program cards, as well as data cards. When using it to input program cards, the PTAPE# command is keyed in and executed.

Syntax:

PTAPE# select code

Each card can contain only one program statement with a line number because a line feed code (decimal 10) is sent at the end of each card. This code is the same as if the END OF LINE key was pressed from the calculator keyboard.

The CONTINUOUS PICK button on the front of the reader must be pressed after the PTAPE# statement is executed to pick the first card. Be sure the RUN statement is on the last card so that the program will be executed automatically.

NOTES

APPENDIX

STANDARD SELECT CODES

These select codes are the ones set by the factory.

Model #	Description	9830A Select Code
2570A	Coupler/Controller	4
9860A	Card Reader	12
9861A	Typewriter (11201A Interface)	15*
9862A	Plotter	14
9863A	Tape Reader	7*
9864A	Digitizer	9*
9865A	Cassette Memory	5
9866A	Thermal Printer	15
9867A/B	Mass Memory	11
9869A	Hopper Card Reader	1* 2
9870A	Card Reader	12
11202A	TTL Interface	1
11203A	BCD Input Interface	3*
11205A	Serial I/O Interface, RS-232-C	15*
11206A	Modem Interface	4*
11282A	Incremental Plotter Interface	14*
11284A	Data Communications Interface	1
59405A	HP Interface Bus	13
09830-67960	Internal Cassette Memory (9830A)	10

*These select codes are resettable.

Exceptions:

9861A - Select code 15 is hardwired on 11201A's with Serial Numbers lower than 1142A-00321.

9866A - Select code 15 is hardwired in the 9830A. No interface card is necessary to connect the 9830A to the 9866A.

11282A - Select code 14 is on position 0 of a select code switch. The 11282A can therefore have alternate select codes of one through nine.

Table A-1. ASCII character codes.

ASCII Char.	EQUIVALENT FORMS			KEY
	Binary	Octal	Dec [2]	9830A
NULL	00000000	000	0	—
SOH	00000001	001	1	—
STX	00000010	002	2	—
ETX	00000011	003	3	—
EOT	00000100	004	4	—
ENQ	00000101	005	5	—
ACK	00000110	006	6	—
BELL	00000111	007	7	—
BS	00001000	010	8	—
HTAB	00001001	011	9	—
LF	00001010	012	10	—
VTAB	00001011	013	11	—
FF	00001100	014	12	—
CR	00001101	015	13	—
SO	00001110	016	14	—
SI	00001111	017	15	—
DLE	00010000	020	16	—
DC_1	00010001	021	17	—
DC_2	00010010	022	18	—
DC_3	00010011	023	19	—
DC_4	00010100	024	20	—
NAK	00010101	025	21	—
SYNC	00010110	026	22	—
ETB	00010111	027	23	—
CAN	00011000	030	24	—
EM	00011001	031	25	—
SUB	00011010	032	26	—
ESC	00011011	033	27	—
FS	00011100	034	28	—
GS	00011101	035	29	—
RS	00011110	036	30	—
US	00011111	037	31	—

ASCII Char.	EQUIVALENT FORMS			KEY [1]
	Binary	Octal	Dec [2]	9830A
space	00100000	040	32	[Space Bar]
!	00100001	041	33	[SHIFT] [!/1]
"	00100010	042	34	—
#	00100011	043	35	[SHIFT] [#/3]
$	00100100	044	36	[SHIFT] [$/4]
%	00100101	045	37	[SHIFT] [%/5]
&	00100110	046	38	[SHIFT] [&/6]
'	00100111	047	39	[SHIFT] ['/7]
(	00101000	050	40	[(]
)	00101001	051	41	[)]
*	00101010	052	42	[*]
+	00101011	053	43	[+]
,	00101100	054	44	[,]
−	00101101	055	45	[−]
.	00101110	056	46	[.]
/	00101111	057	47	[/]
Ø	00110000	060	48	[0]
1	00110001	061	49	[1]
2	00110010	062	50	[2]
3	00110011	063	51	[3]
4	00110100	064	52	[4]
5	00110101	065	53	[5]
6	00110110	066	54	[6]
7	00110111	067	55	[7]
8	00111000	070	56	[8]
9	00111001	071	57	[9]
:	00111010	072	58	[*/:]
;	00111011	073	59	[+/;]
<	00111100	074	60	[SHIFT] [</,]
=	00111101	075	61	[=]
>	00111110	076	62	[SHIFT] [>/.]
?	00111111	077	63	[SHIFT] [?//]

ASCII Char.	EQUIVALENT FORMS Binary	Octal	Dec [2]	KEY [1] 9830A
@	01000000	100	64	(SHIFT) (RESULT)
A	01000001	101	65	(A)
B	01000010	102	66	(B)
C	01000011	103	67	(C)
D	01000100	104	68	(D)
E	01000101	105	69	(E)
F	01000110	106	70	(F)
G	01000111	107	71	(G)
H	01001000	110	72	(H)
I	01001001	111	73	(I)
J	01001010	112	74	(J)
K	01001011	113	75	(K)
L	01001100	114	76	(L)
M	01001101	115	77	(M)
N	01001110	116	78	(N)
O	01001111	117	79	(O)
P	01010000	120	80	(P)
Q	01010001	121	81	(Q)
R	01010010	122	82	(R)
S	01010011	123	83	(S)
T	01010100	124	84	(T)
U	01010101	125	85	(U)
V	01010110	126	86	(V)
W	01010111	127	87	(W)
X	01011000	130	88	(X)
Y	01011001	131	89	(Y)
Z	01011010	132	90	(Z)
[	01011011	133	91	—
\	01011100	134	92	—
]	01011101	135	93	—
^	01011110	136	94	(↑)
_	01011111	137	95	—

ASCII Char.	EQUIVALENT FORMS Binary	Octal	Dec [2]	KEY [1] 9830A
	01100000	140	96	—
a	01100001	141	97	(SHIFT) (A)
b	01100010	142	98	(SHIFT) (B)
c	01100011	143	99	(SHIFT) (C)
d	01100100	144	100	(SHIFT) (D)
e	01100101	145	101	(SHIFT) (E)
f	01100110	146	102	(SHIFT) (F)
g	01100111	147	103	(SHIFT) (G)
h	01101000	150	104	(SHIFT) (H)
i	01101001	151	105	(SHIFT) (I)
j	01101010	152	106	(SHIFT) (J)
k	01101011	153	107	(SHIFT) (K)
l	01101100	154	108	(SHIFT) (L)
m	01101101	155	109	(SHIFT) (M)
n	01101110	156	110	(SHIFT) (N)
o	01101111	157	111	(SHIFT) (O)
p	01110000	160	112	(SHIFT) (P)
q	01110001	161	113	(SHIFT) (Q)
r	01110010	162	114	(SHIFT) (R)
s	01110011	163	115	(SHIFT) (S)
t	01110100	164	116	(SHIFT) (T)
u	01110101	165	117	(SHIFT) (U)
v	01110110	166	118	(SHIFT) (V)
w	01110111	167	119	(SHIFT) (W)
x	01111000	170	120	(SHIFT) (X)
y	01111001	171	121	(SHIFT) (Y)
z	01111010	172	122	(SHIFT) (Z)
{	01111011	173	123	—
¦	01111100	174	124	—
}	01111101	175	125	—
~	01111110	176	126	—
DEL	01111111	177	127	—

11202A-C-50673 T-0001

[1] When (SHIFT) is shown, the SHIFT key is held down while the following key is pressed. Where a key is not shown in the 9830A column, use a WRITE (FORMAT B) statement to output the decimal-equivalent number.

[2] Decimal numbers are used with WRITE BYTE, and WRITE (FORMAT B) statements.

To display the status code of a peripheral device, execute the STAT statement using the select code of the peripheral device.

Syntax:

STAT select code

Following is a table of status codes and their corresponding meanings for various peripheral devices.

Peripheral Device	Status Code	Meaning
9861A Typewriter	0	Switched OFF or not connected.
	1	Switched ON.
9862A Plotter (select code 14)	0	Not connected to calculator.
	9	Switched OFF.
	3	Ready, pen UP.
	11	Ready, pen DOWN.
9863A Tape Reader	0	Switched OFF or not connected.
	1	Switched ON.
9864A Digitizer	0	Switched OFF or not connected.
	1	Switched ON.
9865A Cassette Memory	0	Not connected to calculator.
	1	Switched ON; tape ready and unprotected.
	3	Switched ON; protected tape cassette.
	5	Switched ON; tape on clear leader and unprotected.
	7	Switched ON; tape on clear leader & protected tape cassette.
	11	Door open; tape is in but not on clear leader.
	15	Switched ON; door open or ajar; no tape in or tape on clear leader.
	14	Switched OFF.
9869A Card Reader	0	Not connected to calculator.
	1	Connected to calculator.

BINARY CODING AND CONVERSIONS

This section is a simplified explanation of binary number representations. For in depth information, refer to any text on the subject.

Binary is the base 2 number system, in which only 1's and 0's are used. By giving the 1's and 0's positional value, any decimal number can be represented. Integer-precision numbers are represented by binary in the 9830A. For example:

decimal 41 = binary 101001

Decimal	
10^1	10^0
10	1
4	1

Binary					
2^5	2^4	2^3	2^2	2^1	2^0
32	16	8	4	2	1
1	0	1	0	0	1

Binary-Decimal Conversions

To convert from binary to decimal, the positional values for ones are added up. From the above example, this would be

$$2^5 + 2^3 + 2^0 = 32 + 8 + 1 = 41$$

To convert from decimal to binary, the decimal number is repeatedly divided by 2. The remainder provides the binary equivalent. For example:

	Remainder (read up)
2) 41	1
2) 20	0
2) 10	0
2) 5	1
2) 2	0
2) 1	1

ASCII code

Binary is often used as a code to represent not only numbers, but also alphanumeric characters such as: A or , or ? or x or 2. One of the most common binary codes used is ASCII[1]. ASCII is an eight-bit code — seven data bits and one parity bit. The 9830A Calculator uses ASCII for alphanumeric character representation. The parity bit is not used by the 9830A, however, it is available.

Examples:

Character	ASCII Binary Code	ASCII Decimal Code
A	01000001	65
B	01000010	66
?	00111111	63

For a complete list of ASCII characters and their equivalent binary and decimal representations, see Table A-1.

[1]ASCII is an acronym for American Standard Code for Information Interchange.

Binary-Coded Decimal

Another often used code for representing numeric values is Binary Coded Decimal (BCD). BCD is a four-bit binary code. Each four bits represents a decimal digit from 0 through 9.

For example to convert BCD 010000110110 to decimal:

BCD representation	0100	0011	0110
Decimal Value	4	3	6

The equivalent decimal number is found by breaking up the binary number into groups of 4 bits (starting from the right) and converting these groupings into decimal. Full-precision and split-precision numbers are represented in BCD by the 9830A.

Octal-Binary Conversions

Octal is the base 8 number system. Octal numbers are often used since conversion from binary to octal and vice-versa is very easy.

To convert from binary to octal, the binary number is broken up into groups of three bits (starting from the decimal point to the left). The groupings of 3 bits represent an octal number.

For example to convert binary 10110100011001 to octal:

Binary Number	10	110	100	011	001
Octal Number	2	6	4	3	1

Note that only values from 0 through 7 are used in octal.

To convert from octal to binary, the reverse is done:

Octal Number	1	4	0	7	2	6
Binary Number	001	100	000	111	010	110

Other Codes

There are many other binary codes which are possible. Some of the more popular codes are Electronics Industries Association Code (EIA), Extended Binary Coded Decimal Interchange Code (EBCDIC), Baudot Code, and 6-bit Transcode. To do a conversion to ASCII, which is used by the 9830A Calculator, use the conversion table option in the ENTER or OUTPUT statements.

NOTES

WORLD WIDE SALES & SERVICE OFFICES

UNITED STATES

ALABAMA
8290 Whitesburg Dr., S.E.
P.O. Box 4207
Huntsville 35802
Tel: (205) 881-4591
TWX: 810-726-2204

***Birmingham**
Medical Service only
Tel: (205) 879-2081

ARIZONA
2336 E. Magnolia St.
Phoenix 85034
Tel: (602) 244-1361
TWX: 910-951-1331

2424 East Aragon Rd.
Tucson 85706
Tel: (602) 889-4661

CALIFORNIA
1430 East Orangethorpe Ave.
Fullerton 92631
Tel: (714) 870-1000
TWX: 910-592-1288

3939 Lankershim Boulevard
North Hollywood 91604
Tel: (213) 877-1282
TWX: 910-499-2170

6305 Arizona Place
Los Angeles 90045
Tel: (213) 649-2511
TWX: 910-328-6147

***Los Angeles**
Tel: (213) 776-7500

3003 Scott Boulevard
Santa Clara 95050
Tel: (408) 249-7000
TWX: 910-338-0518

***Ridgecrest**
Tel: (714) 446-6165

2220 Watt Ave.
Sacramento 95825
Tel: (916) 482-1463
TWX: 910-367-2092

9606 Aero Drive
P.O. Box 23333
San Diego 92123
Tel: (714) 279-3200
TWX: 910-335-2000

COLORADO
5600 South Ulster Parkway
Englewood 80110
Tel: (303) 771-3455
TWX: 910-935-0705

CONNECTICUT
12 Lunar Drive
New Haven 06525
Tel: (203) 389-6551
TWX: 710-465-2029

FLORIDA
P.O. Box 24210
2806 W. Oakland Park Blvd.
Ft. Lauderdale 33307
Tel: (305) 731-2020
TWX: 510-955-4099

***Jacksonville**
Medical Service only
Tel: (904) 725-6333

P.O. Box 13910
6177 Lake Ellenor Dr.
Orlando 32809
Tel: (305) 859-2900
TWX: 810-850-0113

21 East Wright St.
Suite 1
Pensacola 32501
Tel: (904) 434-3081

GEORGIA
P.O. Box 28234
450 Interstate North
Atlanta 30328
Tel: (404) 434-4000
TWX: 810-766-4890

HAWAII
2875 So. King Street
Honolulu 96814
Tel: (808) 955-4455

ILLINOIS
(Calculators Only)
100 S. Wacker Drive
Suite 1100
Chicago 60606
Tel: (312) 346-9701

5500 Howard Street
Skokie 60076
Tel: (312) 677-0400
TWX: 910-223-3613

***St. Joseph**
Tel: (217) 469-2133

INDIANA
3839 Meadows Drive
Indianapolis 46205
Tel: (317) 546-4891
TWX: 810-341-3263

IOWA
1902 Broadway
Iowa City 52240
Tel: (319) 338-9466
Night: (319) 338-9467

***KANSAS**
Derby
Tel: (316) 267-3655

LOUISIANA
P.O. Box 840
3239 Williams Boulevard
Kenner 70062
Tel: (504) 721-6201
TWX: 810-955-5524

MARYLAND
6707 Whitestone Road
Baltimore 21207
Tel: (301) 944-5400
TWX: 710-862-9157

4 Choke Cherry Road
Rockville 20850
Tel: (301) 948-6370
TWX: 710-828-9685
710-828-0487

P.O. Box 1648
2 Choke Cherry Road
Rockville 20850
Tel: (301) 948-6370
TWX: 710-828-9684

MASSACHUSETTS
32 Hartwell Ave.
Lexington 02173
Tel: (617) 861-8960
TWX: 710-326-6904

MICHIGAN
23855 Research Drive
Farmington 48024
Tel: (313) 476-6400
TWX: 810-242-2900

MINNESOTA
2400 N. Prior Ave.
Roseville 55113
Tel: (612) 636-0700
TWX: 910-563-3734

MISSISSIPPI
***Jackson**
Medical Service only
Tel: (601) 982-9363

MISSOURI
11131 Colorado Ave.
Kansas City 64137
Tel: (816) 763-8000
TWX: 910-771-2087

148 Weldon Parkway
Maryland Heights 63043
Tel: (314) 567-1455
TWX: 910-764-0830

NEBRASKA
(Medical Only)
11902 Elm Street
Suite 4C
Omaha 68144
Tel: (402) 333-6017

***NEVADA**
Las Vegas
Tel: (702) 382-5777

NEW JERSEY
W. 120 Century Rd.
Paramus 07652
Tel: (201) 265-5000
TWX: 710-990-4951

NEW MEXICO
P.O. BOx 8366
Station C
6501 Lomas Boulevard N.E.
Albuquerque 87108
Tel: (505) 265-3713
TWX: 910-989-1665

156 Wyatt Drive
Las Cruces 88001
Tel: (505) 526-2485
TWX: 910-983-0550

NEW YORK
6 Automation Lane
Computer Park
Albany 12205
Tel: (518) 458-1550
TWX: 710-441-8270

New York City
Manhattan, Bronx
Contact Paramus, NJ Office
Tel: (201) 265-5000
Brooklyn, Queens, Richmond
Contact Woodbury, NY Office
Tel: (516) 921-0300

201 South Avenue
Poughkeepsie 12601
Tel: (914) 454-7330
TWX: 510-248-0012

39 Saginaw Drive
Rochester 14623
Tel: (716) 473-9500
TWX: 510-253-5981

5858 East Molloy Road
Syracuse 13211
Tel: (315) 455-2486
TWX: 710-541-0482

1 Crossways Park West
Woodbury 11797
Tel: (516) 921-0300
TWX: 510-221-2168

NORTH CAROLINA
P.O. Box 5188
1923 North Main Street
High Point 27262
Tel: (919) 885-8101
TWX: 510-926-1516

OHIO
16500 Sprague Road
Cleveland 44130
Tel: (216) 243-7300
Night: 243-7305
TWX: 810-423-9431

330 Progress Rd.
Dayton 45449
Tel: (513) 859-8202
TWX: 810-459-1925

1041 Kingsmill Parkway
Columbus 43229
Tel: (614) 436-1041

OKLAHOMA
P.O. Box 32008
Oklahoma City 73132
Tel: (405) 721-0200
TWX: 910-830-6862

OREGON
17890 SW Boones Ferry Road
Tualatin 97062
Tel: (503) 620-3350
TWX: 910-467-8714

PENNSYLVANIA
111 Zeta Drive
Pittsburgh 15238
Tel: (412) 782-0400
Night: 782-0401
TWX: 710-795-3124

1021 8th Avenue
King of Prussia Industrial Park
King of Prussia 19406
Tel: (215) 265-7000
TWX: 510-660-2670

SOUTH CAROLINA
6941-0 N. Trenholm Road
Columbia 29260
Tel: (803) 782-6493

TENNESSEE
***Memphis**
Medical Service only
Tel: (901) 274-7472

***Nashville**
Medical Service only
Tel: (615) 244-5448

TEXAS
P.O. Box 1270
201 E. Arapaho Rd.
Richardson 75080
Tel: (214) 231-6101
TWX: 910-867-4723

P.O. Box 27409
6300 Westpark Drive
Suite 100
Houston 77027
Tel: (713) 781-6000
TWX: 910-881-2645

205 Billy Mitchell Road
San Antonio 78226
Tel: (512) 434-8241
TWX: 910-871-1170

UTAH
2890 South Main Street
Salt Lake City 84115
Tel: (801) 487-0715
TWX: 910-925-5681

VIRGINIA
***Norfolk**
Medical Service only
Tel: (804) 497-1026

P.O. Box 9854
2914 Hungary Springs Road
Richmond 23228
Tel: (804) 285-3431
TWX: 710-956-0157

WASHINGTON
Bellefield Office Pk.
1203-114th SE
Bellevue 98004
Tel: (206) 454-3971
TWX: 910-443-2446

***WEST VIRGINIA**
Charleston
Tel: (304) 345-1640

WISCONSIN
9431 W. Beloit Road
Suite 117
Milwaukee 53227
Tel: (414) 541-0550

FOR U.S. AREAS NOT LISTED:
Contact the regional office nearest you: Atlanta, Georgia... North Hollywood, California... Rockville, (4 Choke Cherry Rd.) Maryland...Skokie, Illinois. Their complete addresses are listed above.

***Service Only**

CANADA

ALBERTA
Hewlett-Packard (Canada) Ltd.
11748 Kingsway Ave.
Edmonton T5G 0X5
Tel: (403) 452-3670
TWX: 610-831-2431

Hewlett-Packard (Canada) Ltd.
915-42 Avenue S.E. Suite 102
Calgary T2G 1Z1
Tel: (403) 287-1672

BRITISH COLUMBIA
Hewlett-Packard (Canada) Ltd.
837 E. Cordova Street
Vancouver V6A 3R2
Tel: (604) 254-0531
TWX: 610-922-5059

MANITOBA
Hewlett-Packard (Canada) Ltd.
513 Century St.
St. James
Winnipeg R3H 0L8
Tel: (204) 786-7581
TWX: 610-671-3531

NOVA SCOTIA
Hewlett-Packard (Canada) Ltd.
800 Windmill Road
Dartmouth B3C 1L1
Tel: (902) 469-7820

ONTARIO
Hewlett-Packard (Canada) Ltd.
1785 Woodward Dr.
Ottawa K2C 0P9
Tel: (613) 225-6530
TWX: 610-562-8968

Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive
Mississauga L4V 1L9
Tel: (416) 678-9430
TWX: 610-492-4246

QUEBEC
Hewlett-Packard (Canada) Ltd.
275 Hymus Blvd.
Pointe Claire H9R 1G7
Tel: (514) 697-4232
TWX: 610-422-3022
TLX: 05-821521 HPCL

Hewlett-Packard (Canada) Ltd.
2376 Galvani Street
Ste-Foy G1N 4G4
Tel: (418) 688-8710

FOR CANADIAN AREAS NOT LISTED:
Contact Hewlett-Packard (Canada) Ltd. in Mississauga.

CENTRAL AND SOUTH AMERICA

ARGENTINA
Hewlett-Packard Argentina S.A.C.e.I
Lavalle 1171-3° Piso
Buenos Aires
Tel: 35-0436, 35-0627, 35-0341
Telex: 012-1009
Cable: HEWPACK ARG

BOLIVIA
Stambuk & Mark (Bolivia) Ltda.
Av. Mariscal, Santa Cruz 1342
La Paz
Tel: 40626, 53163, 52421
Telex: 3560014
Cable: BUKMAR

BRAZIL
Hewlett-Packard Do Brasil I.E.C. Ltda.
Rua Frei Caneca, 1.152-Bela Vista
01307-**São Paulo**-SP
Tel: 288-71-11, 287-81-20, 287-61-93
Telex: 309151/2/3
Cable: HEWPACK São Paulo

Hewlett-Packard Do Brasil I.E.C. Ltda.
Praca Dom Feliciano, 78-8° andar (Sala 806/8)
9000-**Pôrto Alegre**-RS
Tel: 25-84-70-DDD (0512)
Cable: HEWPACK Pôrto Alegre

Hewlett-Packard Do Brasil I.E.C. Ltda.
Rua Siqueira Campos, 53, 4° andar Copacabana
2000-**Rio de Janeiro**-GB
Tel: 257-80-94-DDD (021)
Telex: 2100 79 HEWPACK
Cable: HEWPACK Rio de Janeiro

CHILE
Calcagni y Metcalfe Ltda.
Calle Lira 81, Oficina 5
Casilla 2118
Santiago
Tel: 398613
Cable: CALMET

COLOMBIA
Instrumentación
Henrik A. Langebaek & Kier S.A.
Carrera 7 No. 48-59
Apartado Aéreo 6287
Bogota, 1 D.E.
Tel: 45-78-06, 45-55-46
Cable: AARIS Bogota
Telex: 44400INSTCO

COSTA RICA
Lic. Alfredo Gallegos Gurdián
Apartado 10159
San José
Tel: 21-86-13
Cable: GALGUR San José

GUATEMALA
IPESA
Avenida La Reforma 3-48, Zona 9
Guatemala
Tel: 63627, 64736
Telex: 4192 TELTRO GU

MEXICO
Hewlett-Packard Mexicana, S.A. de C.V.
Torres Adalid No. 21, 11° Piso
Col. del Valle
Mexico 12, D.F.
Tel: (905) 543-42-32
Telex: 017-74-507

Hewlett-Packard Mexicana, S.A. de C.V.
Ave. Constitución No. 2184
Monterrey, N.L.
Tel: 48-71-32, 48-71-84

NICARAGUA
Roberto Terán G.
Apartado Postal 689
Edificio Terán
Managua
Tel: 3451, 3452
Cable: ROTERAN Managua

PANAMA
Electrónico Balboa, S.A.
P.O. Box 4929
Calle Samuel Lewis
Cuidad de Panama
Tel: 64-2700
Telex: 3431103 Curunda, Canal Zone
Cable: ELECTRON Panama

PARAGUAY
Z.J. Melamed S.R.L.
Division: Aparatos y Equipos Medicos
Division: Aparatos y Equipos Scientificos y de Investigacion
P.O. Box 676
Chile, 482, Edificio Victoria
Asunción
Tel: 4-5069, 4-6272
Cable: RAMEL

PERU
Compañia Electro Médica S.A.
Ave. Enrique Canaual 312
San Isidro
Casilla 1030
Lima
Tel: 22-3900
Cable: ELMED Lima

PUERTO RICO
San Juan Electronics, Inc.
P.O. Box 5167
Ponce de León 154
Pda. 3-PTA de Tierra
San Juan 00906
Tel: (809) 725-3342, 722-3342
Cable: SATRONICS San Juan
Telex: SATRON 3450 332

URUGUAY
Pablo Ferrando S.A.
Comercial e Industrial
Avenida Italia 2877
Casilla de Correo 370

Montevideo
Tel: 40-3102
Cable: RADIUM Montevideo

VENEZUELA
Hewlett-Packard de Venezuela C.A.
Apartado 50933
Edificio Segre
Tercera Transversal
Los Ruices Norte
Caracas 107
Tel: 35-00-11
Telex: 21146 HEWPACK
Cable: HEWPACK Caracas

FOR AREAS NOT LISTED, CONTACT:
Hewlett-Packard
Inter-Americas
3200 Hillview Ave.
Palo Alto, California 94304
Tel: (415) 493-1501
TWX: 910-373-1260
Cable: HEWPACK Palo Alto
Telex: 034-8300, 034-8493

EUROPE

AUSTRIA
Hewlett-Packard Ges.m.b.H.
Handelska 52/3
P.O. Box 7
A-1205 **Vienna**
Tel: (0222) 33 66 06 to 09
Cable: HEWPAK Vienna
Telex: 75923 hewpak a

BELGIUM
Hewlett-Packard Benelux S.A./N.V.
Avenue de Col-Vert, 1, (Groenkraaglaan)
B-1170 **Brussels**
Tel: (02) 672 22 40
Cable: PALOBEN Brussels
Telex: 23 494 paloben bru

DENMARK
Hewlett-Packard A/S
Datavej 52
DK-3460 **Birkerød**
Tel: (01) 81 66 40
Cable: HEWPACK AS
Telex: 166 40 hp as

Hewlett-Packard A/S
Navervej 1
DK-8600 **Silkeborg**
Tel: (06) 82 71 66
Telex: 166 40 hp as
Cable: HEWPACK AS

FINLAND
Hewlett-Packard Oy
Nahkahousuntie 5
P.O. Box 6
SF-00211 **Helsinki** 21
Tel: 6923031
Cable: HEWPACKOY Helsinki
Telex: 12-15363

FRANCE
Hewlett-Packard France
Quartier de Courtaboeuf
Boite Postale No. 6
F-91401 **Orsay**
Tel: (1) 907 78 25
Cable: HEWPACK Orsay
Telex: 60048

Hewlett-Packard France
Agence Régional
Chemin des Mouilles
Boite Postale No. 12
F-69130 **Ecully**
Tel: (78) 33 81 25, 83 65 25
Telex: 31 617

Hewlett-Packard France
Agence Régionale
Zone Aéronautique
Avenue Clément Ader
F-31770 **Colomiers**
Tel: (61) 78 11 55
Telex: 51957

Hewlett-Packard France
Agence Régionale
Centre d'aviation générale
F-13721 **Aéroport de Marignane**
Tel: (91) 89 12 36
TWX: 41770 F

Hewlett-Packard France
Agence Régionale
63, Avenue de Rochester
F-35000 **Rennes**
Tel: 74912 F
Telex. 74 912.F

Hewlett-Packard France
Agence Régionale
74, Allée de la Robertsau
F-67000 **Strasbourg**
Tel: (88) 35 23 20 21
Telex: 89141
Cable: HEWPACK STRBG

GERMAN FEDERAL REPUBLIC

Hewlett-Packard GmbH
Vertriebszentrale Frankfurt
Bernerstrasse 117
Postfach 560 140
D-6000 **Frankfurt** 56
Tel: (0611) 50 04-1
Cable: HEWPACKSA Frankfurt
Telex: 41 32 49 fra

Hewlett-Packard GmbH
Vertriebsbüro Böblingen
Herrenbergerstrasse 110
D-7030 **Boblingen**, Württemberg
Tel: (07031) 66 72 87
Cable: HEPAK Böblingen
Telex: 72 65 739 bbn

Hewlett-Packard GmbH
Vertriebsbüro Düsseldorf
Vogelsanger Weg 38
D-4000 **Düsseldorf**
Tel: (0211) 63 80 31 5
Telex: 85/86 533 hpdd d

Hewlett-Packard GmbH
Vertriebsbüro Hamburg
Wendenstrasse 23
D-2000 **Hamburg** 1
Tel: (040) 24 13 93
Cable: HEWPACKSA Hamburg
Telex: 21 63 032 hphh d

Hewlett-Packard GmbH
Vertriebsbüro Hannover
Mellendorfer Strasse 3
D-3000 **Hannover-Kleefeld**
Tel: (0511) 55 60 46
Telex. 092 3259

Hewlett-Packard GmbH
Vertriebsbüro Nuremberg
Hersbruckerstrasse 42
D-8500 **Nuremberg**
Tel: (0911) 57 10 66
Telex: 623 860

Hewlett-Packard GmbH
Vertriebsbüro München
Unterhachinger Strasse 28
ISAR Center
D-8012 **Ottobrunn**
Tel: (089) 601 30 61/7
Telex: 52 49 85
Cable: HEWPACKSA München

(West Berlin)
Hewlett-Packard GmbH
Vertriebsbüro Berlin
Keith Strasse 2-4
D-1000 **Berlin** 30
Tel: (030) 24 90 86
Telex: 18 34 05 hpbln d

GREECE

Kostas Karayannis
18, Ermou Street
GR-**Athens** 126
Tel: 3230-303 Sales/SVC
3230-305 Adm. Order Proc.
Cable: RAKAR Athens
Telex: 21 59 62 rkar gr

Hewlett-Packard S.A.
Mediterranean & Middle East Operations
35 Kolokotroni Street
Platia Kefallariou
Gr-Kifissia-**Athens**
Tel: 8080337, 8080359, 8080429, 8018693
Telex: 21 6588
Cable: HEWPACKSA Athens

IRELAND

Hewlett-Packard Ltd.
King Street Lane
Winnersh, Workingham
GB-**Berkshire** RG11 5AR
Tel: Wokingham 784774
Telex: 847178/848179

Hewlett-Packard Ltd.
"The Graftons"
Stamford New Road
GB-**Altrincham**, Cheshire
Tel: (061) 928-9021
Telex: 668068

ITALY

Hewlett-Packard Italiana S.p.A.
Via Amerigo Vespucci 2
I-20124 **Milan**
Tel: (2) 6251 (10 lines)
Cable: HEWPACKIT Milan
Telex: 32046

Hewlett-Packard Italiana S.p.A.
Via Pietro Maroncelli 40
(ang. Via Visentin)
I-35100 **Padova**
Tel: 66 40 62/66 31 88
Telex: 32046 via Milan

Hewlett-Packard Italiana S.p.A.
Via Medaglie d'Oro, 2
I-56100 **Pisa**
Tel: (050) 500022
Telex. 32046 via Milan

Hewlett-Packard S.p.A.
Via G. Armellini 10
I-00143 **Rome**-Eur
Tel: (6) 5912544/5
Telex: 61514
Cable: HEWPACKIT Rome

Hewlett-Packard Italiana S.p.A.
Via San Quintino, 46
I-10121 **Turin**
Tel: (11) 53 82 64
Telex: 32046 via Milan

LUXEMBURG

Hewlett-Packard Benelux S.A./N.V.
Avenue de Col-Vert, 1,
(Groenkraaglaan)
B-1170 **Brussels**
Tel: (02) 672 22 40
Cable: PALOBEN Brussels
Telex: 23 494

NETHERLANDS

Hewlett-Packard Benelux N.V.
Weerdestein 117
P.O. Box 7825
NL-**Amsterdam**, 1011
Tel: (020) 5411522
Cable: PALOBEN Amsterdam
Telex: 13 216 hepa nl

NORWAY

Hewlett-Packard Norge A/S
Nesveien 13
Box 149
N-1344 **Haslum**
Tel: (02) 53 83 60
Telex: 16621 hpnas n

PORTUGAL

Telectra-Empresa Técnica de Equipamentos Eléctricos S.a.r.l.
Rua Rodrigo da Fonseca 103
P.O. Box 2531
P-**Lisbon** 1
Tel: (19) 68 60 72
Cable: TELECTRA Lisbon
Telex: 12598

SPAIN

Hewlett-Packard Española, S.A.
Jerez No. 3
E-**Madrid** 16
Tel: 458 26 00
Telex: 23515 hpe

Hewlett-Packard Española, S.A.
Milanesado 21-23
E-**Barcelona** 17
Tel: (3) 2036200-08, 2044098/9
Telex: 52603 hpbe e

Hewlett-Packard Española, S.A.
Av Ramon y Cajal, 1
Edificio Sevilla I, planta 9°
E-**Seville**
Tel: 64 44 54/58

Hewlett-Packard Española S.A.
Edificio Albia II 7° B
E-**Bilbao**
Tel: 23 83 06/23 82 06

SWEDEN

Hewlett-Packard Sverige AB
Enighetsvägen 1-3
Fack
S-161 20 **Bromma** 20
Tel: (08) 730 05 50
Cable: MEASUREMENTS Stockholm
Telex: 10721

Hewlett-Packard Sverige AB
Hagakersgatan 9C
S-431 41 **Molndal**
Tel: (031) 27 68 00/01
Telex: Via Bromma

SWITZERLAND

Hewlett-Packard (Schweiz) AG
Zürcherstrasse 20
P.O. Box 64
CH-8952 Schlieren **Zurich**
Tel: (01) 98 18 21
Cable: HPAG CH
Telex: 53933 hpag

Hewlett-Packard (Schweiz) AG
9, chemin Louis-Pictet
CH-1214 Vernier-**Geneva**
Tel: (022) 41 49 50
Cable: HEWPACKSA Geneva
Telex: 27 333 hpsa ch

TURKEY

Telekom Engineering Bureau
Saglik Sok No. 15/1
Ayaspasa-Beyoglu
P.O. Box 437 Beyoglu
TR-**Istanbul**
Tel: 49 40 40
Cable: TELEMATION Istanbul

UNITED KINGDOM

Hewlett-Packard Ltd.
King Street Lane
Winnersh, Wokingham
GB-**Berkshire** RG11 5AR
Tel: Workingham 784774
Telex: 847178/848179

Hewlett-Packard Ltd.
"The Graftons"
Stamford New Road
GB-**Altrincham**, Cheshire
Tel: (061) 928-9021
Telex: 668068

Hewlett-Packard Ltd.
c/o Makro
South Service Wholesale Centre
Amber Way
Halesowen Industrial Estate
GB-**Halesowen**, Worcs
Tel: Birmingham 7860

Hewlett-Packard Ltd.
4th Floor
Wedge House
799, London Road
GB-**Thornton Heath** CR4 6XL, Surrey
Tel: (01) 684 0105
Telex: 946825

Hewlett-Packard Ltd.
c/o Makro
South Service Wholesale Centre
Wear Industrial Estate
Washington
GB-**New Town**, County Durham
Tel: Washington 464001 ext. 57/58

Hewlett-Packard Ltd.'s registered address for V.A.T. purposes only:
70, Finsbury Pavement
London, EC2A1SX
Registered No. 690597

USSR

Hewlett-Packard
Representative Office USSR
Hotel Budapest/Room 201
Petrovskie Linii 2/18
Moscow
Tel: 221-79-71

YUGOSLAVIA

Iskra-Standard/Hewlett-Packard
Topniska 58/3
61000 **Ljubljana**
Tel: 314561 or 314927
Telex: 31300

SOCIALIST COUNTRIES PLEASE CONTACT:

Hewlett-Packard S.A.
7, rue du Bois-du-Lan
P.O. Box 349
CH-1217 Meyrin 1 **Geneva**
Switzerland
Tel: (022) 41 54 00
Cable: HEWPACKSA Geneva
Telex: 2 24 86

AFRICA, ASIA, AUSTRALIA

ANGOLA

Telectra
Empresa Técnica de Equipamentos Eléctricos, S.A.R.L.
R. Barbosa Rodrigues, 42-1°DT.°
Caixa Postal, 6487-**Luanda**
Tel: 35515/6
Cable: TELECTRA Luanda

AUSTRALIA

Hewlett-Packard Australia Pty. Ltd.
31-41 Joseph Street
Blackburn, Victoria 3130
Tel: 89-6351, 89-6306
Telex: 31-024
Cable: HEWPARD Melbourne

Hewlett-Packard Australia Pty. Ltd.
31 Bridge Street
Pymble,
New South Wales, 2073
Tel: 449-6566
Telex: 21561
Cable: HEWPARD Sydney

Hewlett-Packard Australia Pty. Ltd.
97 Churchill Road
Prospect 5082
South Australia
Tel: 44 8151
Cable: HEWPARD Adelaide

Hewlett-Packard Australia Pty. Ltd.
141 Stirling Highway
Nedlands, W.A., 6009
Tel: 86 5455

Hewlett-Packard Australia Pty. Ltd.
121 Wollongong Street
Fyshwick, A.C.T., 2609
Tel: 95 3733

Hewlett-Packard Australia Pty. Ltd.
5th Floor
Teachers Union Building
495-499 Boundary Street
Spring Hill, 4000 Queensland
Tel: 29-1544
Telex: AA-42133

CEYLON

United Electricals Ltd
P.O. Box 681
60, Park St.
Colombo 2
Tel: 26696
Cable: HOTPOINT Colombo

CYPRUS

Kypronics
19 Gregorios & Xenopoulos Rd.
P.O. Box 1152
CY-**Nicosia**
Tel: 45628/29
Cable: KYPRONICS PANDEHIS

ETHIOPIA

African Salespower & Agency Private Ltd., Co.
P.O. Box 718
58/59 Cunningham St.
Addis Ababa
Tel: 12285
Cable: ASACO Addisababa

HONG KONG

Schmidt & Co.(Hong Kong) Ltd.
P.O. Box 297
Connalight Centre
39th Floor
Connaught Road, Central
Hong Kong
Tel: 240168, 232735
Telex: HX4766
Cable: SCHMIDTCO Hong Kong

INDIA

Blue Star Ltd.
Kasturi Buildings
Jamshedji Tata Rd.
Bombay 400 020
Tel: 29 50 21
Telex: 3751
Cable: BLUEFROST

Blue Star Ltd.
Sahas
414/2 Vir Savarkar Marg
Prabhadevi
Bombay 400 025
Tel: 45 78 87
Telex: 4093
Cable: FROSTBLUE

Blue Star Ltd.
Band Box House
Prabhadevi
Bombay 400 025
Tel: 45 73 01
Telex: 3751
Cable: BLUESTAR

Blue Star Ltd.
14 40 Civil Lines
Kampur 208 001
Tel: 6 88 82
Cable: BLUESTAR

Blue Star Ltd.
7 Hare Street
P.O. Box 506
Calcutta 700 001
Tel: 23-0131
Telex: 655
Cable: BLUESTAR

Blue Star Ltd.
Blue Star House.
34 Ring Road
Lajpat Nagar
New Delhi 110 024
Tel: 62 32 76
Telex: 2463
Cable: BLUESTAR

Blue Star Ltd.
Blue Star House
11/11A Magarath Road
Bangalore 560 025
Tel: 55668
Telex: 430
Cable: BLUESTAR

Blue Star Ltd.
Meeakshi Mandiran
xxx/1678 Mahatma Gandhi Rd.
Cochin 682 016 Kerala

Blue Star Ltd.
1-1-117/1
Sarojini Devi Road
Secunderabad 500 003
Tel: 7 63 91, 7 73 93
Cable: BLUEFROST
Telex: 459

Blue Star Ltd.
23/24 Second Line Beach
Madras 600 001
Tel: 23954
Telex: 379
Cable: BLUESTAR

Blue Star Ltd.
Nathraj Mansions
2nd Floor Bistupur
Jamshedpur 831 001
Tel: 38 04
Cable: BLUESTAR
Telex: 240

INDONESIA

BERCA Indonesia P.T.
P.O. Box 496
1st Floor JL. Cikini Raya 61
Jakarta
Tel: 56038, 40369, 49886
Telex: 2895 Jakarta

IRAN

Multi Corp International Ltd
Avenue Soraya 130
P.O. Box 1212
IR-**Teheran**
Tel: 83 10 35-39
Cable: MULTICORP Tehran
Telex. 2893 mci tn

ISRAEL

Electronics & Engineering Div. of Motorola Israel Ltd.
17 Aminadav Street
Tel-Aviv
Tel: 36941 (3 lines)
Cable: BASTEL Tel-Aviv
Telex: 33569

JAPAN

Yokogawa-Hewlett-Packard Ltd.
Ohashi Building
1-59-1 Yoyogi
Shibuya-ku, **Tokyo**
Tel: 03-370-2281/92
Telex: 232-2024YHP
Cable: YHPMARKET TOK 23-724

Yokogawa-Hewlett-Packard Ltd.
Nisei Ibaragi Bldg.
2-2-8 Kasuga
Ibaragi-Shi
Osaka
Tel: (0726) 23-1641
Telex: 5332-385 YHP OSAKA

Yokogawa-Hewlett-Packard Ltd.
Nakamo Building
No. 24 Kamisasazima-cho
Nakamura-ku, **Nagoya** City
Tel: (052) 571-5171

Yokogawa-Hewlett-Packard Ltd.
Nitto Bldg.
2-4-2 Shinohara-Kita
Kohoku-ku
Yokohama 222
Tel: 045-432-1504
Telex: 382-3204 YHP YOK

Yokogawa-Hewlett-Packard Ltd.
Chuo Bldg.
Rm. 603 3,
2-Chome
IZUMI-CHO
Mito, 310
Tel: 0292-25-7470

KENYA

Technical Engineering Services
P.O. Box 18311
Nairobi, Kenya
Tel: 57726
Cable: PROTON

KOREA

American Trading Company Korea
I.P.O. Box 1103
Oae Kyung Bldg., 8th Floor
107 Sejong-Ro.
Chongro-Ku, **Seoul**
Tel: (4 lines) 73-8924-7
Cable: AMTRACO Seoul

KUWAIT

Al-Khaldiya Trading & Contracting Co.
Al Soor Street
Michaan Bldg. No. 4
Kuwait
Tel: 42 99 10
Cable: VISCOUNT

LEBANON

Constantin E. Macridis
Clemenceau Street 34
P.O. Box 7213
RL-**Beirut**
Tel: 220846
Telex: 21114 Leb
Cable: ELECTRONUCLEAR Beirut

MALAYSIA

MECOMB Malaysia Ltd.
2 Lorong 13/6A
Section 13
Petaling Jaya, **Selangor**
Cable: MECOMB Kuala Lumpur

MOZAMBIQUE

A.N. Goncalves, Lta.
162, 1° Apt. 14 Av. D. Luis
Caixa Postal 107
Lourenco Marques
Tel: 27091, 27114
Telex: 6-203 Negon Mo
Cable: NEGON

NEW ZEALAND

Hewlett-Packard (N.Z.) Ltd.
94-96 Dixon Street
P.O. Box 9443
Courtenay Place,
Wellington
Tel: 59-559
Telex: 3898
Cable: HEWPACK Wellington

Hewlett-Packard (N.Z.) Ltd.
Pakuranga Professional Centre
267 Pakuranga Highway
Box 51092
Pakuranga
Tel: 569-651
Cable: HEWPACK, Auckland

NIGERIA

The Electronics Instrumentatiofns Ltd.
N6B/770 Oyo Road
Oluseun House
P.M.B. 5402
Ibadan
Tel: 22325
Cable: THETEIL Ibadan

The Electronics Instrumentations Ltd. (TEIL)
16th Floor Cocoa House
P.M.B. 5402
Ibadan
Tel: 22325
Cable: THETEIL Ibadan

PAKISTAN

Mushko & Company, Ltd.
Oosman Chambers
Abdullah Haroon Road
Karachi 3
Tel: 511027, 512927
Cable: COOPERATOR Karachi

Mushko & Company, Ltd.
38B, Satellite Town
Rawalpindi
Tel: 41924
Cable: FEMUS Rawalpindi

PHILIPPINES

Electromex, Inc.
6th Floor, Amalgamated Development Corp. Bldg.
Ayala Avenue, Makati, Rizal
C.C.P.O. Box 1028
Makati, Rizal
Tel: 86-18-87, 87-76-77,
Cable: ELEMEX Manila

SINGAPORE

Mechanical & Combustion Engineering Company Pte., Ltd.
10/12, Jalan Kilang
Red Hill Industrial Estate
Singapore, 3
Tel: 647151 (7 lines)
Cable: MECOMB Singapore

Hewlett-Packard Singapore (Pte.) Ltd.
Blk. 2, 6th FLOOR, Jalan Bukit Merah
Redhill Industrial Estate
Alexandra P.O. Box 87,
Singapore 3
Tel: 633022
Telex: HPSG RS 21486
Cable: HEWPACK, Singapore

SOUTH AFRICA

Hewlett-Packard South Africa (Pty.), Ltd.
Hewlett-Packard House
Daphne Street, Wendywood
Sandton, Transvaal 2001
Tel: 802-1040
Telex: SA43-4782JH
Cable: HEWPACK

Hewlett-Packard South Africa (Pty.), Ltd.
Breecastle House
Bree Street
Cape Town
Tel: 2-6941/2/3
Cable: HEWPACK Cape Town
Telex: 0006 CT

Hewlett-Packard South Africa (Pty.), Ltd.
641 Ridge Road, Durban
P.O. Box 99
Overport, Natal
Tel: 88-6102
Telex: 567954
Cable: HEWPACK

TAIWAN

Hewlett-Packard Taiwan
39 Chung Shiao West Road
Sec. 1 Overseas Insurance Corp. Bldg. 7th Floor
Taipei
Tel: 389160,1,2, 375121, Ext. 240-249
Telex: TP824 HEWPACK
Cable: HEWPACK Taipei

Hewlett-Packard Taiwan
38, Po-Ai Lane, San Min Chu,
Kaohsiung
Tel: 297319

THAILAND

UNIMESA Co., Ltd.
Elsom Research Building
Bangjak Sukumvit Ave.
Bangkok
Tel: 932387, 930338
Cable: UNIMESA Bangkok

UGANDA

Uganda Tele-Electric Co., Ltd.
P.O. Box 4449
Kampala
Tel: 57279
Cable: COMCO Kampala

VIETNAM

Peninsular Trading Inc.
P.O. Box H-3
216 Hien-Vuong
Saigon
Tel: 20-805, 93398
Cable: PENTRA, SAIGON 242

ZAMBIA

R.J. Tilbury (Zambia) Ltd.
P.O. Box 2792
Lusaka
Zambia, Central Africa
Tel: 73793
Cable: ARJAYTEE, Lusaka

MEDITERRANEAN AND MIDDLE EAST COUNTRIES NOT SHOWN PLEASE CONTACT:

Hewlett-Packard S.A.
Mediterranean and Middle East Operations
35, Kolokotroni Street
Platia Kefallariou
GR-Kifissia-**Athens**
Telex: 21-6588
Cable: HEWPACKSA Athens

OTHER AREAS NOT LISTED, CONTACT:

Hewlett-Packard
Export Trade Company
3200 Hillview Ave.
Palo Alto, California 94304
Tel: (415) 493-1501
TWX: 910-373-1267
Cable: HEWPACK Palo Alto
Telex: 034-8300, 034-8493

SYNTAX SUMMARY

The ENTER Statement

ENTER (select code, format [,conversion table]) list [FOR parameter]

The ENTER statement enables the calculator to receive data from an external device. If the incoming data is not in ASCII code, conversion to ASCII code can be performed using the optional conversion table parameter. The ENTER statement causes one record to be read from the device indicated by the select code. Character by character conversion to ASCII code is performed, if requested. The data is then handled in the same way as during a standard READ or INPUT statement.

The OUTPUT Statement

OUTPUT (select code or string name, format [,conversion table]) list

The OUTPUT statement is a general-purpose statement for sending data or coded commands to an external device. If the external device requires non-ASCII code, conversion from ASCII to that code is possible using the optional conversion table parameter.

The CMD (COMMAND) Statement

The CMD statement is a specialized output instruction used only to control instruments connected to the HP Interface Bus (HP-IB).

The STAT Function

STAT select code

The STAT function returns a value from 0 to 15 inclusive representing the status (on, off, wait, etc.) of a peripheral.

The RBYTE (READ BYTE) Function

RBYTE select code

The RBYTE function reads one byte of data from the device specified by the select code, regardless of the data structure.

The ROT (ROTATE) Function

ROT (value, number of places)

The ROT function performs right rotation on the binary equivalent of the value, the number of places specified.

The INOR (INCLUSIVE OR) Function

INOR ($value_1$, $value_2$)

The INOR function combines binary equivalents of $value_1$ and $value_2$ in an "inclusive or" logic operation.

The BIAND (BINARY AND) Function

BIAND ($value_1$, $value_2$)

The BIAND function combines binary equivalents of $value_1$ and $value_2$ in an AND logic operation.

The WBYTE (WRITE BYTE) Function

WBYTE value

The WBYTE function outputs the binary equivalent of the value length 8 bits. It is used in the WRITE, PRINT, or OUTPUT statements.

The SPA (SPACE) Function

SPA value

The SPA function advances the printer or typewriter carriage the number of spaces represented by the value.

ERROR NOTES

Indication	Meaning
ERROR 83	End of data reached or data contains more than ten blanks in a row.
ERROR 84	Invalid format specification.
ERROR 85	Numeric input has syntax error: multiple decimal points, more than one E, or other non-numerical input.
ERROR 86	Conversion table or code not found. Check for integer initialization in DIM or COM statement.

PART NO. 09830-90029
MICROFICHE NO. 09830-99029

PRINTED IN U.S.A.
APRIL 3, 1975