MODEL 30

HEWLETT-PACKARD 9830A CALCULATOR EXTENDED I/O ROM

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

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 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 Model 30's BASIC language, allowing a wide variety of devices to be attached to the calculator. Although the calculator recognizes ASCII* code, the I/O ROM enables data to be received in any form and converted to ASCII code.

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 ASCII Bus.

SUPPLIED EQUIPMENT

Two Operating Manuals, -hp- Part Number 09830-90007, are supplied with the Extended I/O ROM.

INSPECTION PROCEDURE

Refer to Appendix A in the 9830A Calculator Operating and Programming Manual for the procedures to verify operation of the ROM.

* American Standard Code for Information Interchange.

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 MODEL 30

The general I/O scheme for interfacing devices with the Model 30 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*.

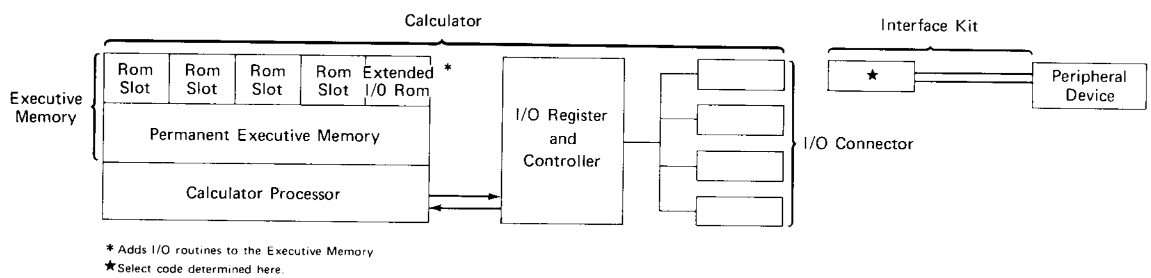


Figure 2-1. The Model 30 and the I/O Scheme

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.

INTERFACE KITS

Interface kits are available to permit a wide range of devices which are not 9800 Series peripherals to be interfaced to the Model 30. 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 Model 30 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.

Your nearest -hp- Sales and Service office can furnish you with data sheets listing specifications for available interface kits.

* Use of the Model 9868A I/O Expander permits up to thirteen devices to be connected to the calculator at one time.

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 Table 3-1 in Chapter 3.

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 Model 30 to send and receive data and to send commands using standard ASCII code. Programmable instructions are available for conversion from non-ASCII code to ASCII code at time of input, and from ASCII code to non-ASCII code at time of output. ASCII equivalent codes are given in Appendix A.

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. (Alternatively, information can be output in a 12-bit parallel character series by using the special I/O ROM statement, WBYTE.) 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.

There is no provision for system 'interrupt' operation when using the I/O ROM (i.e., the peripheral device cannot initiate or call for an input or output operation). 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.

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. Parameters which are underlined in the I/O statement syntax are explained in Table 3-1. Square brackets are used in syntax descriptions to enclose those parameters which are optional.

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. 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]) variable list [,FOR parameter]

```
10 ENTER (7,*)A,B,(FORI=1TO10,A[I])
20 ENTER (7,100)A,B,C
30 ENTER (9,100)(FORI=1TO9STEP3,A[I])
40 ENTER (7,*)(FORI=1TO10,(FORJ=1TO10,A[I,J]))
```

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 found. The data is then handled in the same way as during a standard READ or INPUT statement.

If a standard ASCII left arrow (←) 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) or an alternate mode (ALT) is encountered, a search is made for LF (end of record). The entire record containing ESC or ALT is ignored, and because the ENTER statement has not been terminated, the next record is accepted.

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

The OUTPUT statements, as well as the standard WRITE and PRINT statements, cause a carriage return (CR) and line feed (LF) to be automatically output at termination of the OUTPUT list. As in the standard WRITE and PRINT statements, however, the CR and LF can be suppressed by using a semicolon (;).

Data can be output with the standard leading blanks suppressed for devices, such as network analyzers, etc., requiring data in that form. See 'Leading Blanks', later in this chapter.

```
10 OUTPUT (15,*)A,B,C
20 OUTPUT (15,*,A)A,B,C
30 OUTPUT (9,35,T)"ABCDEFG",1,2,3
40 OUTPUT (9,35,A)A[5,6]
```

If a string name* is specified, internal conversion occurs. That is, data is not output but instead is moved to the string field. This enables you to perform additional processing of the string characters before output. For example, you can use a conversion table to generate even parity (See Chapter 4). In the following statements, S is a parity conversion table (A$ is the string name):

```
80 OUTPUT (A$,*)A,B,C
90 OUTPUT (15,*,S)A$
```

* Valid only when the String Variables ROM is installed in the calculator.

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

PARAMETER	EXPLANATION
select code	A numerical code, from 1 to 15, uniquely representing the peripheral device, as follows: select code 1 9869A Calculator Card Reader select code 2 through 9 User assignable select code 5 9865A Cassette Memory select code 10 Internal Cassette Memory select code 11 & 12 Reserved select code 13 11144A ASCII Bus select code 14 9862A Plotter select code 15 9861A Typewriter or 9866A Printer
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 String Variables ROM (either Option 274 or 11274B) 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	Input multiple data items from one record into an array; syntax as follows: (FOR var = exp TO exp [STEP exp],[(FOR var = exp TO exp [STEP exp],] array name (var_1 [,var_2]))). No NEXT statement is used with FOR in an ENTER statement.

THE CMD (COMMAND) STATEMENT

The CMD statement is a specialized output instruction used only to send commands to the -hp- Model 11144A ASCII Bus. This Bus is an interface system which interconnects electronic instruments which are ASCII compatible. The select code for the ASCII Bus is 13; the CMD statement cannot be used for any other select code or device.

Syntax:

CMD address [[,message],address] . . .

address – a literal field, enclosed in quotes, or a string name if the String Variables ROM is installed, gives the address of an ASCII Bus peripheral to be commanded.

message – a literal field, or string name, gives a command to be directed to the ASCII Bus peripheral.

Multiple address-message groups may be sent if desired, separated by commas.

For more information about the use of the CMD statement with the ASCII Bus, refer to the Operating Manual supplied with the 11144A ASCII Bus.

I/O FUNCTIONS

The following I/O Functions return a value to an expression, in the same way as, for example, the square root function returns a value. An example utilizing the INOR, ROT and RBYTE functions can be found on page 8-6.

The STAT (Status) Function

STAT select code

The STAT function returns a 4-bit code indicating the operational status (on, off, wait, etc.) of the device specified by the select code. In general, if the device is in a 'ready' condition, the code is 1, if not the code is 0. Experimentation with other conditions may prove useful for certain devices. Status codes for 9800 Series peripherals are listed in the Appendix.

The RBYTE (Read Byte) Function

RBYTE select code

The RBYTE function reads one byte (8 bits) of data from the device specified by the select code, regardless of the data structure.

> NOTE
>
> The following functions, ROT, INOR and BIAND, are performed on integer expressions. Full and split precision expressions are changed to integer expressions before the functions are performed. Thus if the absolute value of an expression is greater than 32767, the expression is evaluated as ±32767.

The ROT (Rotate) Function

ROT (expression 1, expression 2)

The ROT function performs right rotation on the binary equivalent of expression 1, the number of positions represented by expression 2. Refer to the note, above.

The INOR (Inclusive OR) Function

INOR (expression 1, expression 2)

The INOR function combines the binary equivalents of expression 1 and expression 2 in an 'inclusive or' logic operation. Refer to the note, above.

The BIAND (Binary AND) Function

BIAND (expression 1, expression 2)

The BIAND function combines the binary equivalents of expression 1 and expression 2 in an 'and' logic operation. Refer to the note, above.

OUTPUT CONTROL FUNCTIONS

The following functions are used with the WRITE, OUTPUT, or PRINT Statements.

The WBYTE (Write Byte) Function

WBYTE expression

The WBYTE function outputs the binary equivalent of an expression, length 8 bits. It is used to output to a binary storage device and also provides increased control of the output format.

The SPA (Space) Function

SPA expression

The SPA function advances the printer or typewriter carriage the number of character spaces represented by the expression. This number cannot exceed 72. The SPA function cannot be executed in a formatted output statement.

The LIN (Line Feed) Function

LIN expression

The LIN function advances the printer or typewriter the number of lines represented by the expression. If the expression is positive, the carriage returns and advances the number of lines specified. If the expression is negative, there is no carriage return; the printer or typewriter carriage advances the absolute value of the number of lines specified. If the expression is zero, the carriage returns but does not advance. LIN cannot be executed in a formatted output statement.

CONVERSION TABLES

A conversion table enables groups or characters of arrays of data to be converted from one code to another. As mentioned earlier, the calculator makes use of standard ASCII codes. Throughout this manual all non-ASCII representation codes used by printers, card readers, paper tape readers, punches, typewriters, etc., are referred to as 'foreign codes'.

To create a conversion table, an integer array is used. The decimal equivalents of the ASCII characters are stored as array elements. The decimal equivalents of the foreign code characters correspond to the array subscripts.

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

```
10 DIM AI[150],BI[2,256]
```

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.

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. When the conversion table is referenced in an OUTPUT statement, the contents of the conversion table are searched 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.

Here is an example. Suppose you have a paper tape reader which uses EIA* coded tape. X and Y values are contained on the tape, separated by commas. The chart below shows the tape punches, the represented symbols, and also the octal and decimal equivalent codes for the tape punches for the EIA codes contained on the tape.

EIA STANDARD TAPE CHANNEL NUMBERS									SYMBOL	EIA OCTAL EQUIVALENT	EIA DECIMAL EQUIVALENT
8	7	6	5	4	Feed	3	2	1			
		●			·				0	40	32
					·			●	1	1	1
					·		●		2	2	2
			●		·		●	●	3	23	19
					·	●			4	4	4
			●		·	●		●	5	25	21
			●		·	●	●		6	26	22
					·	●	●	●	7	7	7
				●	·				8	10	8
			●	●	·			●	9	31	25
		●	●	●	·		●	●	,	73	59
	●	●		●	·		●	●	.	153	107
●					·				CARR RET	200	128

* Electronic Industries Association Standard Code.

From the ASCII table in Appendix A, you can write a chart showing the ASCII decimal equivalents of the same symbols.

SYMBOL	EIA DECIMAL EQUIVALENT	ASCII DECIMAL EQUIVALENT
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
LF	128	10

In the ASCII chart, the Line Feed (LF) is used to correspond to the carriage return (CR) on the incoming tape. Now, a conversion table can be made directly from your EIA/ASCII chart. First, dimension an integer array for the conversion table:

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

The conversion table is dimensioned to have 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.

Statements like the following one can now be used for automatic conversion at input:

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

CONVERSION TABLES
(Continued)

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 statements 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 it 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. If the conversion table is used for output of foreign code, no two elements in the array should contain the same values. If they do, the code having the lowest-valued subscript is output.

TWO-DIMENSIONAL CONVERSION TABLES

A conversion table with more than 256 elements can be created by using a two-dimensional array. Conversion from ASCII code to a foreign code, or from a foreign code to ASCII code, does not require that many elements, so a two-dimensional conversion table is not needed. However, 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, in which case a two-dimensional conversion table can be used.

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	1	21
B	2	22
C	3	23
D	4	24
E	5	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, 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 a '4', in PQR code (see the above array), is input, the calculator simply goes to the fourth element in the array and finds, in this case, the value '24' in XYZ code. 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 '5'.

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 subscripts (I,J) is:

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

LEADING BLANKS

To suppress leading blanks in output data, a special E or F notation can be used in a FORMAT statement. Basically, this consists of specifying a field width between 1000 and 1014. For example:

```
40 FORMAT 3F1000.2
50 FORMAT E1000.8,F1010.8
```

Here is a program to enter A, B and C from the keyboard, to print the numbers and then suppress blanks using the OUTPUT statement.

```
10 DISP "A,B,C";
20 INPUT A,B,C
30 PRINT A,B,C
40 FORMAT 3F1000.2
50 OUTPUT (15,40)A,B,C
```

The FORMAT statement in line 40, above, causes the numbers entered from the keyboard to be printed to two decimal places.

To limit maximum field width of the output data, the F notation can be changed; the following statement specifies a maximum field width of 6:

```
40 FORMAT 3F1006.2
```

NOTES

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 9861A Typewriter, 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
```

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
70 FOR J=1 TO 7
80 IF ROT(I,J)>0 THEN 100
90 D=D+1
100 NEXT J
110 IF BIAND(D,1)#0 THEN 140
120 A[I]=I
130 GOTO 150
140 A[128+I]=I
150 NEXT I
```

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

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

PARITY GENERATION
(Continued)

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

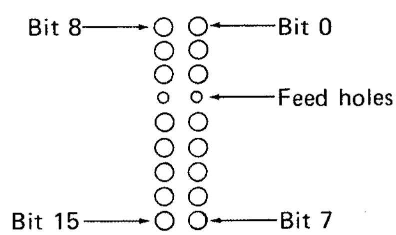
```
170 FOR I=1 TO 255 STEP 4
180 FOR J=1 TO I+3
190 PRINT "A("J")="A[J];
200 NEXT J
210 PRINT
220 NEXT I
```

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:

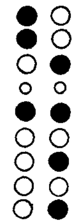
```
50 X=INOR(ROT(RBYTE9,8),RBYTE9)
```

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.

00001011 10101100

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

DIGITAL VOLTMETER INTERFACING

The 11203A, Revision C, BCD interface card makes the 3480A/B Digital Voltmeter and the 3485A Scanning Unit compatible with the Model 30 Calculator. Revision A of the 11203 card, however, is not compatible with the Model 30. Therefore, it is necessary to use the Extended I/O ROM to build a conversion table.

The following program builds a conversion table:

```
10 COM AI[63]
20 FOR I=1 TO 63
30 A[I]=I
40 NEXT I
50 A[61]=10
60 PRINT "CONVERSION TABLE IS IN MEMORY"
70 END
```

DIGITAL VOLTMETER INTERFACING
(continued)

To store the conversion table on a cassette, use the following program:

```
10 PRINT "WANT CONVERSION TABLE ON FILE=";
20 INPUT A
30 STORE  DATA A
40 PRINT "CONVERSION TABLE STORED ON FILE";A
50 END
```

The next program loads the conversion table from the cassette to the calculator memory.

```
10 DISP "CONVERSION TABLE ON FILE=";
20 INPUT A
30 LOAD  DATA A
40 PRINT "CONVERSION TABLE IS IN MEMORY"
50 END
```

The conversion table can be used as follows:

```
10 COM AI[63]
20 LOAD  DATA A
30 ENTER (3,*,A)A,B
40 PRINT A,B
50 GOTO 30
60 END
```

When connected to the Digital Voltmeter, the calculator can set range and filter conditions, set to home (channel 0) position, step to a specific channel or simply to the next channel, and read single or multiple data. Sample programs to perform these operations are printed below. Certain functions must be set up initially. These are as follows:

FNA (X) to set range and filter conditions.

```
10 DEF FNA(X)=INTLGTP2+P1+1+3*P1
```

FNB (X) to set to home (channel 0) postion.

```
10 DEF FNB(X)=INTLGTP2+P1+1*P1+8
```

FNC (X) to step to next channel.

```
10 DEF FNC(X)=INTLGTP2+P1+1*P1+16
```

FND (X) to set to desired channel.

```
10 DEF FND(X)
20 WRITE (1,*)WBYTEINTLGTP2+P1+1+3*P1+8;
30 A=A-1
40 IF A<1 THEN 70
50 WRITE (1,*)WBYTEINTLGTP2+P1+1+3*P1+16;
60 GOTO 30
70 RETURN INTLGTP2+P1+1+3*P1+16
```

FNE (X) to take readings.

```
10 DEF FNE(X)
20 DIM R[256]
30 WRITE (1,*)WBYTEINTLGTP2+P1+1+3*P1+8;
40 FOR I=1 TO T2
50 ENTER (3,*)R[I],R[I]
60 WAIT 300
70 IF T1#1 THEN 90
80 WRITE (1,*)WBYTEINTLGTP2+P1+1+3*P1+16;
90 NEXT I
100 RETURN T2
```

These functions can be loaded onto special function keys for easy access. Once these functions have been defined, the following sample programs can be used.

Set

```
10 PRINT "SET"
20 PRINT "P1=FILTER (1=IN;0=OUT) P2=VOLTAGE (.1, 1 OR 10)"
30 PRINT
40 DISP "P1,P2";
50 INPUT P1,P2
60 WRITE (1,*)WBYTEFNAP1;
70 PRINT "FILTER";P1;"VOLTAGE";P2
80 END
```

Home

```
10 PRINT "HOME"
20 WRITE (1,*)WBYTEFNBP1;
30 DISP "HOME";
40 END
```

Step to Next Channel

```
10 PRINT "STEP"
20 A=FNCP1
30 DISP "STEP"
40 END
```

DIGITAL VOLTMETER INTERFACING

(continued)

Step to Desired Channel

```
10 PRINT "SELECT CHANNEL"
20 DISP "DESIRED CHANNEL";
30 INPUT A
40 WRITE (1,*)WBYTEFNDP1;
50 DISP "DONE"
60 END
```

Single Reading

```
10 PRINT "SINGLE READING"
20 ENTER (3,*)A,B
30 PRINT A,B
40 END
```

Multiple Readings

```
10 PRINT "MULTIPLE READINGS"
20 REM IF 1 CHANNEL IS SPECIFIED, BURST MODE IS ASSUMED;
30 REM  OTHERWISE CHANNELS ARE SCANNED.
40 PRINT "HOW MANY CHANNELS";
50 INPUT A
60 IF A=1 THEN 100
70 T1=1
80 T2=A
90 GOTO 140
100 PRINT "HOW MANY READINGS";
110 INPUT A
120 T1=2
130 T2=A
140 A=FNEP1
150 PRINT A;"READINGS COMPLETED"
160 END
```

The above sample programs can be modified for the user's specific needs, but the functions – FNA (X) through FNE (X) – can be re-used for other programs indefinitely. They should be stored permanently on special function keys.

NOTES

Table 5-1. Typewriter Control Codes

Typewriter Key	Decimal Code	Typewriter Key	Decimal Code	Typewriter Key	Decimal Code
TAB SET	1	:	58	[	91
RED RIBBON	6	;	59		92
BLK RIBBON	7	<	60	]	93
BACKSPACE	8	=	61	^	94
TAB	9	>	62	—	95
LINE FEED	10	?	63	`	96
CLR ALL TABS	11	@	64	a	97
TAB CLR	12	A	65	b	98
CARR RET	13	B	66	c	99
SPACE	32	C	67	d	100
!	33	D	68	e	101
"	34	E	69	f	102
#	35	F	70	g	103
$	36	G	71	h	104
%	37	H	72	i	105
&	38	I	73	j	106
' (apost.)	39	J	74	k	107
(	40	K	75	l	108
)	41	L	76	m	109
*	42	M	77	n	110
+	43	N	78	o	111
, (comma)	44	O	79	p	112
-	45	P	80	q	113
.	46	Q	81	r	114
/	47	R	82	s	115
0	48	S	83	t	116
1	49	T	84	u	117
2	50	U	85	v	118
3	51	V	86	w	119
4	52	W	87	x	120
5	53	X	88	y	121
6	54	Y	89	z	122
7	55	Z	90	{	123
8	56			}	125
9	57			~	126

Chapter 5

TYPEWRITER CONTROL

The -hp- Model 9861A Typewriter can be used without the Extended I/O ROM. However, the Extended I/O ROM provides additional control with the special I/O functions WBYTE, LIN and SPA.

Below is a sample of typing which illustrates some of the control possible using the Extended I/O ROM.

```
With the 11274 EXTENDED I/O ROM the typewriter can be
changed to red ribbon and back to black.

                    Physical tabs can be set,
                    and the tab key can be operated.
                    The following special characters
                    can be typed:
                              [
                              ]
                              {
                              }
                              ~
                              "
                              `

Also you can cause the typewriter to
skip lines, backspace, and clear tabs.
```

THE TYPEWRITER FUNCTIONS

Table 5-1 shows the control keys and other characters on the Model 9861A Typewriter, and the decimal code associated with each key or character. Using the WBYTE function and the decimal code, you activate those control keys and special characters which do not have corresponding keys on the calculator. In these examples, the select code for the typewriter is assumed to be 15.

THE TYPEWRITER FUNCTIONS
(Continued)

Red Ribbon:

To set the typewriter to red ribbon, locate RED RIBBON in Table 5-1 (decimal code 6), and use the WBYTE function. For example, the following instruction sets red ribbon and types the word "YES".

```
20 WRITE (15,*)WBYTE6,"YES"
```

Black Ribbon:

To revert to black ribbon, find BLK RIBBON in Table 5-1 (decimal code 7), and use the WBYTE function. For example, the following instruction changes to black ribbon and types the word "NO":

```
20 WRITE (15,*)WBYTE7,"NO"
```

Clear All Tabs:

To clear all tabs on the typewriter, the carriage must first be moved to the extreme right. Then the CLR ALL TABS (decimal code 11) is used. In the following instruction, the SPA function is used to move the carriage 72 spaces to the right and then WBYTE 11 clears all tabs as the carriage returns.

```
20 WRITE (15,*)SPA72,WBYTE11
```

Set Tabs:

To set tabs (decimal code 1) at column 15 and at column 25, for example, use the following instruction:

```
20 WRITE (15,*)SPA15,WBYTE1,SPA10,WBYTE1
```

In the above WRITE statements, the '*' indicates that no FORMAT statement is referenced. In these cases, PRINT could have been used for the same result. Set or clear tabs cannot be executed in a formatted WRITE statement. Here, however, is a statement which references a FORMAT statement.

```
10 X=PI
20 WRITE (15,30)WBYTE91,X,WBYTE93
30 FORMAT 2F8.3
```

When the WRITE statement references a FORMAT statement, a carriage return and line feed (CR/LF) are automatically output after all the specifications in the FORMAT statement have been referenced. Therefore, in statement 30, above, an additional specification is intentionally included in the FORMAT statement to suppress the automatic CR/LF after the value of X has been printed.

Suppressing CR/LF:

The semicolon is used to suppress the CR/LF when no FORMAT statement is referenced:

```
10 WRITE (15,*)WBYTE6,"changed to red ribbon";
20 WRITE (15,*)WBYTE7," and back to black."
```

Carriage Return and Line Feed:

The LIN function can be used to skip lines (CR/LF). In the following WRITE statement, 10 lines are skipped.

```
10 WRITE (15,*)"abc",LIN10,"def"
```

The following program illustrates many of the possible controls you can use with the typewriter. The sample shown earlier was typed using this program. The lower case letters are obtained by typing the letters while the SHIFT key is pressed.

```
10 FIXED 0
20 FORMAT 2F1006.0
30 WRITE (15,*)SPA60,WBYTE11,WBYTE7
40 OUTPUT (15,20)"With the ",11274," EXTENDED I/O ROM the";
50 WRITE (15,*)" typewriter can be";WBYTE6
60 WRITE (15,*)"changed to red ribbon";WBYTE7;
70 WRITE (15,*)" and back to black."
80 WRITE (15,*)LIN2,SPA20,WBYTE1"Physical tabs";
90 WRITE (15,*)" can be set,"
100 WRITE (15,*)WBYTE9"and the tab key can be operated."
110 WRITE (15,*)WBYTE9"The following special characters "
120 WRITE (15,*)WBYTE9"can be typed:"
130 WRITE (15,*)SPA20,WBYTE12,SPA10,WBYTE91;
140 WRITE (15,*)LIN-1,WBYTE8,WBYTE93,LIN-1,WBYTE8,WBYTE123;
150 WRITE (15,*)LIN-1,WBYTE8,WBYTE125,LIN-1,WBYTE8,WBYTE126;
160 WRITE (15,*)LIN-1,WBYTE8,WBYTE34,LIN-1,WBYTE8,WBYTE96
170 WRITE (15,*)WBYTE11,LIN2,"Also you can ";
180 WRITE (15,*)"cause the typewriter to"
190 WRITE (15,*)"skip lines, backspace";
200 WRITE (15,*)LIN0,SPA12;
210 FOR I=1 TO 9
220 WRITE (15,*)WBYTE95;
230 NEXT I
240 WRITE (15,*)", and clear tabs."
250 END
```

NOTES

Chapter 6

TAPE READER CONTROL

The -hp- Model 9863A Tape Reader is controlled by using the ENTER statement. Numeric data can be used as input to the calculator, and, if the Model 11274 String Variables ROM is installed, alphabetic data can also be processed.

When the tape reader mode switch is set to NORMAL, the tape reader is controlled by the calculator. In the following examples, it is assumed that the tape reader is operating in NORMAL mode, and that the select code for the tape reader is 7.

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 (or symbols) ending with a line feed (LF). Here is a sequence of ASCII characters contained on a tape (two records are shown):

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

Here are instructions to read and print the numbers:

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

Another tape contains these three records:

872193425 (LF) 693345786 (LF) 667238425 (LF)

Here are instructions to read and print the numbers:

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

THE USE OF STRINGS

Alphabetic data can be input if you have a String Variables ROM installed in the calculator. The following tape contains character data only:

ROGERS,△JOHN△HT5–10,WT160 (LF) SMITH,△ARTHUR△HT6–2,WT230 (LF)

Here are instructions to read and print that character data:

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

This next tape contains strings of fixed length (in the following case, student names) and numeric data (grades), separated by commas.

ROGERS,△JOHN△△△△80,75,83 (LF) SMITH,△ARTHUR△△△90,93,88 (LF)

Here are instructions to read and print the student names and their average grades:

```
10 DIM A$[80]
20 FOR I=1 TO 2
30 ENTER (7,*)A$[1,16],A,B,C
40 FIXED 0
50 PRINT A,(A+B+C)/3
60 NEXT I
```

Chapter 7

DIGITIZER CONTROL

The -hp- Model 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 a set of X-Y coordinates from the digitizer:

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

When an ENTER statement is encountered, the calculator waits for a data sample to assign values to X and Y.

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
```

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.

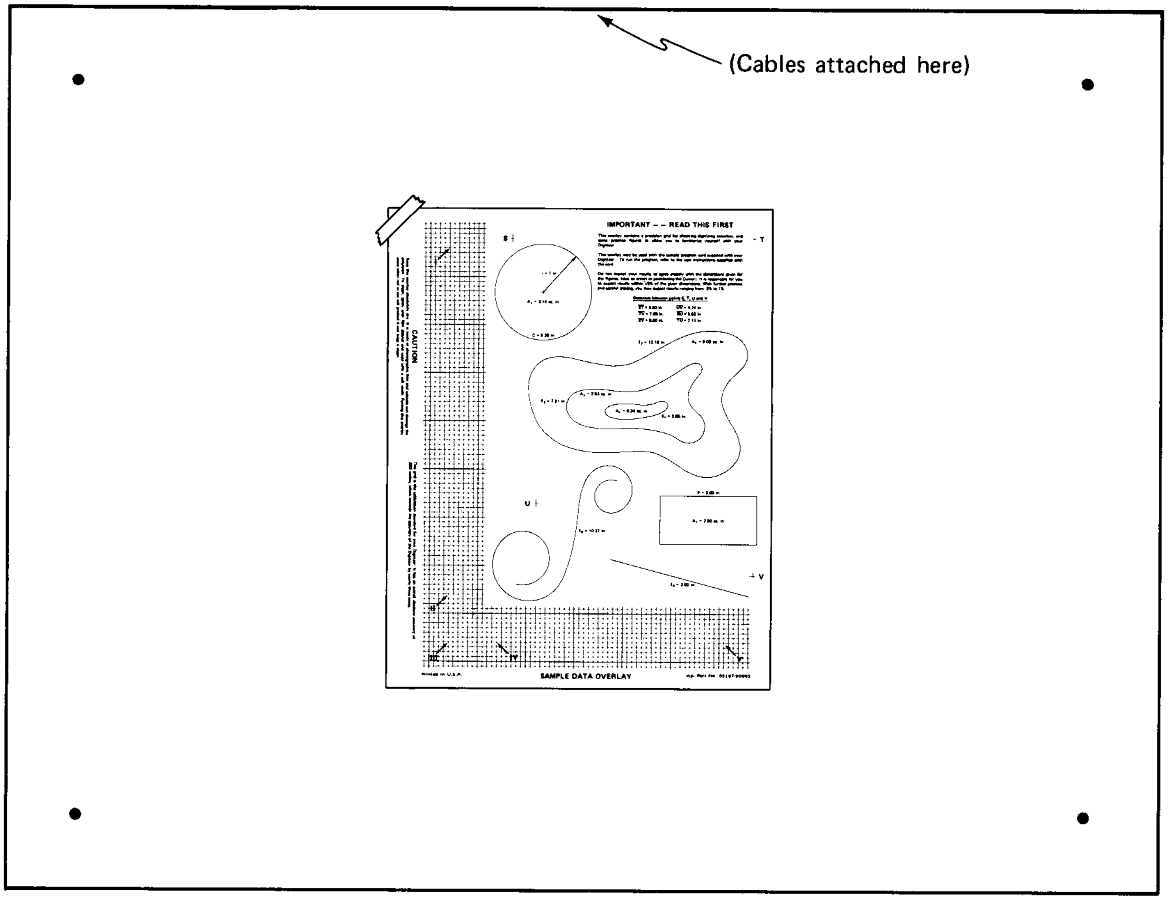
THE STOP KEY

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 7-1) and tape down all four corners of the overlay.

THE DIGITIZER PLATEN



(FRONT OF PLATEN)

Figure 7-1. 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 7-1.
2. Place the cursor cross-hairs over point I (the upper left-hand corner of the document) and press [O]. Slide the cursor over point III (the lower left-hand corner of the document) and, with the above program running in the calculator, position the cross-hairs exactly over point III; press [C]. If point III is positioned exactly over the X axis, the digitizer will 'beep' slowly. Whenever point III 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 I and III (return to step 2 if the X axis is not precisely aligned).

Chapter 8

CALCULATOR CARD READER CONTROL

The -hp- Model 9869A Calculator Card Reader can be used efficiently with the Extended I/O ROM (Option 272, 11272B or 11272F). In addition to the standard features of the Extended I/O ROM, however, the Batch ROM (Option 278, 11278B or 11278F) 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.

The reader inputs information which is stored on punched or marked data processing cards to the Model 9830A Calculator at a rate of up to 300 cards per minute. Accessibility to a key punch machine is a major determining 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 section below.

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 features called "buffers". When power is switched on, the buffers are empty. They can be loaded with the first WRITE input statement 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. 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]) variable list [,FOR parameter]

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 6-2 in this manual for information concerning the use of strings with the 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 reader operates in the DEMAND, CONTINUOUS, IMAGE or NORMAL mode until it is instructed to switch modes. Each of these modes is represented by an alpha control character which is programmed either from the keyboard or from program cards.

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 WRITE statement is used to indicate in which mode the reader is operating.

Syntax:

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

Demand

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 reader. Once the reader is activated, or "ready", it remains ready. The green ready light indicates the 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 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 reader's select code and the symbol, *, indicates a free input format. Notice that the control character, D, is written within quote marks to inform the 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 reader interpreting a lower case letter. The 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 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 yellow 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 write 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 9.99 E+99.

```
10 WRITE (1,*)"D"
20 ENTER (1,60)X,Y,Z
30 IF X=9.99E+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

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

The WRITE "C" statement switches the reader from any mode it is presently in to the CONTINUOUS mode and will activate 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"
```

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

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

In the IMAGE mode, the reader transmits the binary image of the cards in two 8-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.

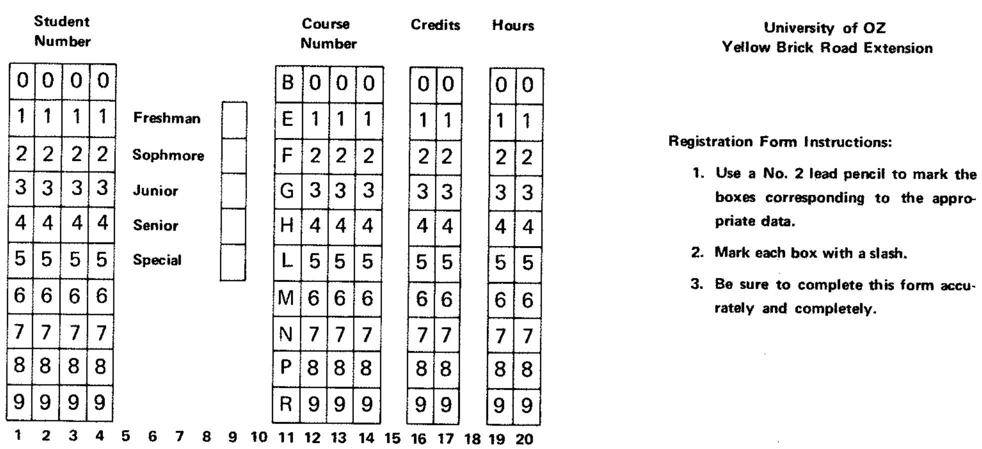
The alpha control character, I, must be used to indicate the IMAGE mode.

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 wanted to know how many hours of courses the average part-time student takes and how many the average full-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. Option 274, String Variables ROM is required to run this program.

OPERATING MODES
(continued)

- **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.

- **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
```

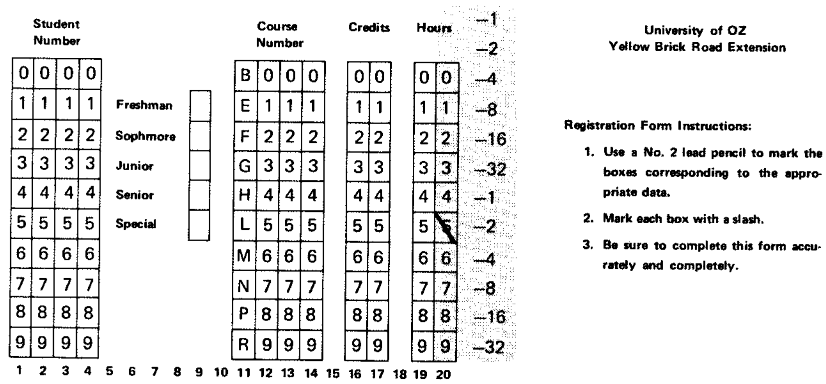
The above program segment picks a card and reads the first 20 columns, one by one. Line 120 combines two bytes of the 6 data bits to make 12 rows per column.†It then tests whether this is the last data card (i.e., whether 0, 1 and 2 are marked in column 1), searches for a mark on each column and assigns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

The program exists 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.

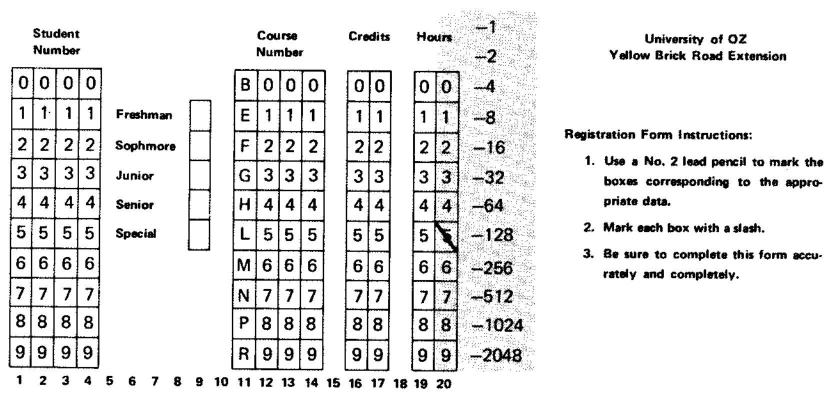
In binary code, marks in the rows of a column represent the following values:



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.

- **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, below.

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

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

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

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

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, above.

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). If the instruction is received after the contents on a card are read, it will abort the next card's data. In either case, 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

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

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 reader can also be used to input program cards, as well as data cards. When using it for this purpose, a PTAPE# statement must be keyed in and executed from the keyboard.

Syntax:

PTAPE# select code

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

ASCII CHARACTER CODES

ASCII Character	Decimal Equivalent	Octal Code	Binary Equivalent
NULL	0	000	0 000 000
LF	10	012	0 001 010
CR/LF	11	013	0 001 011
CR	13	015	0 001 101
△ (space)	32	040	0 100 000
!	33	041	0 100 001
"	34	042	0 100 010
#	35	043	0 100 011
$	36	044	0 100 100
%	37	045	0 100 101
&	38	046	0 100 110
' (apost.)	39	047	0 100 111
(	40	050	0 101 000
)	41	051	0 101 001
*	42	052	0 101 010
+	43	053	0 101 011
, (comma)	44	054	0 101 100
-	45	055	0 101 101
.	46	056	0 101 110
/	47	057	0 101 111
0	48	060	0 110 000
1	49	061	0 110 001
2	50	062	0 110 010
3	51	063	0 110 011
4	52	064	0 110 100
5	53	065	0 110 101
6	54	066	0 110 110
7	55	067	0 110 111
8	56	070	0 111 000
9	57	071	0 111 001
:	58	072	0 111 010
;	59	073	0 111 011
<	60	074	0 111 100
=	61	075	0 111 101

ASCII Character	Decimal Equivalent	Octal Code	Binary Equivalent
>	62	076	0 111 110
?	63	077	0 111 111
@	64	100	1 000 000
A	65	101	1 000 001
B	66	102	1 000 010
C	67	103	1 000 011
D	68	104	1 000 100
E	69	105	1 000 101
F	70	106	1 000 110
G	71	107	1 000 111
H	72	110	1 001 000
I	73	111	1 001 001
J	74	112	1 001 010
K	75	113	1 001 011
L	76	114	1 001 100
M	77	115	1 001 101
N	78	116	1 001 110
O	79	117	1 001 111
P	80	120	1 010 000
Q	81	121	1 010 001
R	82	122	1 010 010
S	83	123	1 010 011
T	84	124	1 010 100
U	85	125	1 010 101
V	86	126	1 010 110
W	87	127	1 010 111
X	88	130	1 011 000
Y	89	131	1 011 001
Z	90	132	1 011 010
[	91	133	1 011 011
\	92	134	1 011 100
]	93	135	1 011 101
↑	94	136	1 011 110
←	95	137	1 011 111

DIAGNOSTIC 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 statement.

PERIPHERAL STATUS CODES

It is often convenient or necessary to know whether a peripheral device is connected to the calculator and, if so, to determine its status.

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.

ELECTRONIC

SALES & SERVICE OFFICES

UNITED STATES

ALABAMA
P.O. Box 4207
2003 Byrd Spring Road S.W.
Huntsville 35802
Tel: (205) 881-4591
TWX: 810-726-2204

ARIZONA
2336 E. Magnolia St.
Phoenix 85034
Tel: (602) 244-1361
TWX: 910-951-1330

5737 East Broadway
Tucson 85711
Tel: (602) 298-2313
TWX: 910-952-1162

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: 910-328-6148

1101 Embarcadero Road
Palo Alto 94303
Tel: (415) 327-6500
TWX: 910-373-1280

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
7965 East Prentice
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

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

GEORGIA
P.O. Box 28234
450 Interstate North
Atlanta 30328
Tel: (404) 436-6181
TWX: 810-766-4890

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

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

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

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

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
2459 University Avenue
St. Paul 55114
Tel: (612) 645-9461
TWX: 910-563-3734

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

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

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

1060 N. Kings Highway
Cherry Hill 08034
Tel: (609) 667-4000
TWX: 710-892-4945

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

1219 Campville Road
Endicott 13760
Tel: (607) 754-0050
TWX: 510-252-0890

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

82 Washington Street
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) 454-2486
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
25575 Center Ridge Road
Cleveland 44145
Tel: (216) 835-0300
TWX: 810-427-9129

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

1120 Morse Road
Columbus 43229
Tel: (614) 846-1300

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
2500 Moss Side Boulevard
Monroeville 15146
Tel: (412) 271-0724
TWX: 710-797-3650

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

RHODE ISLAND
873 Waterman Ave.
East Providence 02914
Tel: (401) 434-5535
TWX: 710-381-7573

***TENNESSEE**
Memphis
Tel: (901) 274-7472

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

231 Billy Mitchell Road
San Antonio 78226
Tel: (512) 434-4171
TWX: 910-871-1170

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

VIRGINIA
P.O. Box 6514
2111 Spencer Road
Richmond 23230
Tel: (803) 285-3431
TWX: 710-956-0157

WASHINGTON
Bellefield Office Pk.
1203 - 114th SE
Bellevue 98004
Tel: (206) 454-3971
TWX: 910-443-2303

***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 . . . Paramus, New Jersey . . . Skokie, Illinois. Their complete addresses are listed above.
*Service Only

CANADA

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

BRITISH COLUMBIA
Hewlett-Packard (Canada) Ltd.
4608 Canada Way
North Burnaby 2
Tel: (604) 433-8213
TWX: 610-922-5059

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

NOVA SCOTIA
Hewlett-Packard (Canada) Ltd.
2745 Dutch Village Rd.
Suite 210
Halifax
Tel: (902) 455-0511
TWX: 610-271-4482

ONTARIO
Hewlett-Packard (Canada) Ltd.
1785 Woodward Dr.
Ottawa 3
Tel: (613) 255-6180, 255-6530
TWX: 610-562-8968

Hewlett-Packard (Canada) Ltd.
50 Galaxy Blvd.
Rexdale
Tel: (416) 677-9611
TWX: 610-492-4246

QUEBEC
Hewlett-Packard (Canada) Ltd.
275 Hymus Boulevard
Pointe Claire
Tel: (514) 697-4232
TWX: 610-422-3022
Telex: 01-20607

FOR CANADIAN AREAS NOT LISTED:
Contact Hewlett-Packard (Canada) Ltd. in Pointe Claire, at the complete address listed above.

CENTRAL AND SOUTH AMERICA

ARGENTINA
Hewlett-Packard Argentina S.A.C.e.I
Lavalle 1171 - 3°
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 1119
01307-**Sao Paulo**-SP
Tel: 288-7111, 287-5858
Telex: 309151/2/3
Cable: HEWPACK Sao Paulo

Hewlett-Packard Do Brasil
I.E.C. Ltda.
Praca Dom Feliciano, 78
90000-**Porto Alegre**-RS
Rio Grande do Sul (RS) Brasil
Tel: 25-8470
Cable: HEWPACK Porto Alegre

Hewlett-Packard Do Brasil
I.E.C. Ltda.
Rua da Matriz, 29
20000-**Rio de Janeiro**-GB
Tel: 266-2643
Telex: 210079 HEWPACK
Cable: HEWPACK Rio de Janeiro

CHILE
Héctor Calcagni y Cia, Ltda.
Casilla 16.475
Santiago
Tel: 423 96
Cable: CALCAGNI Santiago

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

ECUADOR
Laboratorios de Radio-Ingenieria
Calle Guayaquil 1246
Post Office Box 3199
Quito
Tel: 212-496; 219-185
Cable: HORVATH Quito

EL SALVADOR
Electronic Associates
Apartado Postal 1682
Centro Comercial Gigante
San Salvador, El Salvador C.A.
Paseo Escalon 4649-4° Piso
Tel: 23-44-60, 23-32-37
Cable: ELECAS

GUATEMALA
IPESA
5a via 2-01, Zona 4
Guatemala City
Tel: 63-6-27 & 64-7-86
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: 543-42-32
Telex: 017-74-507

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
Ave. Manuel Espinosa No. 13-50
Bldg. Alina
Panama City
Tel: 230833
Telex: 3481103, Curunda, Canal Zone
Cable: ELECTRON Panama City

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
Asuncion
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 Leon 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: 910-373-1267
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) 72 22 40
Cable: PALOBEN Brussels
Telex: 23 494 paloben bru

DENMARK
Hewlett-Packard A/S
Datavej 38
DK-3460 **Birkerod**
Tel: (01) 81 66 40
Cable: HEWPACK AS
Telex: 166 40 hp as

Hewlett-Packard A/S
Torvet 9
DK-8600 **Silkeborg**
Tel: (06) 82-71-66
Telex: 166 40 hp as
Cable: HEWPACK AS

FINLAND
Hewlett-Packard Oy
Bulevardi 26
P.O. Box 12185
SF-00120 **Helsinki** 12
Tel: (90) 13730
Cable: HEWPACKOY Helsinki
Telex: 12-15363 hel

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
4 Quai des Etroits
F-69321 **Lyon Cedex** 1
Tel: (78) 42 63 45
Cable: HEWPACK Lyon
Telex: 31617

Hewlett-Packard France
29 rue de la Gare
F-31700 **Blagnac**
Tel: (61) 85 82 29
Telex: 51957

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 **Böblingen**, Württemberg
Tel: (07031) 66 72 87
Cable: HEPAK Boblingen
Telex: 72 65 739 bbn

Hewlett-Packard GmbH
Vertriebsbüro Düsseldorf
Vogelsanger Weg 38
D-4000 **Düsseldorf**
Tel: (0211) 63 80 31/35
Telex: 85/86 533 hpdd d

Hewlett-Packard GmbH
Vertriebsbüro Hamburg
Wendenstr. 23
D-2000 **Hamburg** 1
Tel: (0411) 24 13 93
Cable: HEWPACKSA Hamburg
Telex: 21 63 032 hphh d

Hewlett-Packard GmbH
Vertriebsburo Hannover
Mellendorfer Strasse 3
D-3000 **Hannover** - Kleefeld
Tel: (0311) 55 06 26

Hewlett-Packard GmbH
Vertriebsbüro München
Unterhachinger Strasse 28
ISAR Center
D-8012 **Ottobrunn**
Tel: (0811) 601 30 61/7
Telex: 52 49 85
Cable: HEWPACKSA Müchen

(**West Berlin**)
Hewlett-Packard GmbH
Vertriebsbüro Berlin
Wilmersdorfer Strasse 113/114
D-1000 **Berlin W.** 12
Tel: (0311) 3137046
Telex: 18 34 05 hpbln d

GREECE
Kostas Karayannis
18, Ermou Street
GR-**Athens** 126
Tel: 3230-303, 3230-305
Cable: RAKAR Athens
Telex: 21 59 62 rkar gr

IRELAND
Hewlett-Packard Ltd.
224 Bath Road
GB-**Slough**, SL1 4 DS, Bucks
Tel: Slough (0753) 33341
Cable: HEWPIE Slough
Telex: 848413

Hewlett-Packard Ltd.
The Graftons
Stamford New Road
Altrincham, Cheshire, England
Tel: (061) 928-8626
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.
Piazza Marconi, 25
I-00144 **Rome** - Eur
Tel: (6) 5912544/5, 5915947
Cable: HEWPACKIT Rome
Telex: 61514

Hewlett-Packard Italiana S.p.A.
Vicolo Pastori, 3
I-35100 **Padova**
Tel: (49) 66 40 62
Telex: 32046 via Milan

Hewlett-Packard Italiana S.p.A.
Via Colli, 24
I-10129 **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: (03/02) 72 22 40
Cable: PALOBEN Brussels
Telex: 23 494

NETHERLANDS
Hewlett-Packard Benelux/N.V.
Weerdestein 117
P.O. Box 7825
NL-**Amsterdam**, Z1011
Tel: 020-42 77 77, 44 29 66
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 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: 1598

SPAIN
Hewlett-Packard Española, S.A.
Jerez No 8
E-**Madrid** 16
Tel: 458 26 00
Telex: 23515 hpe

Hewlett-Packard Española, S.A.
Milanesado 21-23
E-**Barcelona** 17
Tel: (3) 203 62 00
Telex: 52603 hpbe e

SWEDEN
Hewlett-Packard Sverige AB
Enighetsvägen 1-3
Fack
S-161 20 **Bromma** 20
Tel: (08) 98 12 50
Cable: MEASUREMENTS Stockholm
Telex: 10721

Hewlett-Packard Sverige AB
Hagakersgatan 9C
S-431 41 **Mölndal**
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 21/24
Cable: HPAG CH
Telex: 53933 hpag ch

Hewlett-Packard (Schweiz) AG
9, Chemin Louis-Pictet
CH-1214 Vernier—**Geneva**
Tel: (022) 41 4950
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.
224 Bath Road
GB-**Slough**, SL1 4 DS, Bucks
Tel: Slough (0753) 33341
Cable: HEWPIE Slough
Telex: 848413

Hewlett-Packard Ltd.
"The Graftons"
Stamford New Road
GB-**Altrincham**, Cheshire
Tel: (061) 928-8626
Telex: 668068

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

SOCIALIST COUNTRIES PLEASE CONTACT:
Hewlett-Packard Ges.m.b.H.
Handelskai 52/3
P.O. Box 7
A-1205 **Vienna**
Ph: (0222) 33 66 06 to 09
Cable: HEWPACK Vienna
Telex: 75923 hewpak a

ALL OTHER EUROPEAN COUNTRIES CONTACT:
Hewlett-Packard S.A.
Rue du Bois-du-Lan 7
P.O. Box 85
CH-1217 Meyrin 2 **Geneva**
Switzerland
Tel: (022) 41 54 00
Cable: HEWPACKSA Geneva
Telex: 2 24 86

AFRICA, ASIA, AUSTRALIA

ANGOLA
Telectra-Empresa Tecnica de Equipamentos Electricos SARL
Rua de Barbosa, Rodrigues, 42-1°, DT°
P.O. Box 6487
Luanda
Cable: TELECTRA Luanda

AUSTRALIA
Hewlett-Packard Australia Pty. Ltd.
22-26 Weir Street
Glen Iris, 3146
Victoria
Tel: 20-1371 (6 lines)
Cable: HEWPARD Melbourne
Telex: 31 024

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.
1st Floor, Suite 12/13
Casablanca Buildings
196 Adelaide Terrace
Perth, W.A. 6000
Tel: 25-6800
Cable: HEWPARD Perth

Hewlett-Packard Australia Pty. Ltd.
10 Woolley Street
P.O. Box 191
Dickson A.C.T. 2602
Tel: 49-8194
Cable: HEWPARD Canberra ACT

Hewlett-Packard Australia Pty. Ltd.
2nd Floor, 49 Gregory Terrace
Brisbane, Queensland, 4000
Tel: 29 1544

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

CYPRUS
Kypronics
19 Gregorios & Xenopoulos Road
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
1511, Prince's Building
15th Floor
10, Chater Road
Hong Kong
Tel: 240168, 232735
Telex: HX4766 SCHMCO
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 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: 463
Cable: BLUESTAR

Blue Star, Ltd.
Blue Star House
11/11A Magarath Road
Bangalore 560 025
Tel: 51473
Telex: 430
Cable: BLUESTAR

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: 2 39 55
Telex: 379
Cable: BLUESTAR

Blue Star, Ltd.
1B Kaiser Bungalow
Dindli Road
Jamshedpur 831 001
Tel: 38 04
Cable: BLUESTAR
Telex: 240

INDONESIA
Bah Bolon Trading Coy. N.V.
Djalah Merdeka 29
Bandung
Tel: 4915; 51560
Cable: ILMU
Telex: 08-809

IRAN
Multicorp 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
Kenya Kinetics
P.O. Box 18311
Nairobi, Kenya
Tel: 57726
Cable: PROTON

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

LEBANON
Constantin E. Macridis
P.O. Box 7213
RL-**Beirut**
Tel: 220846
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, Av. D. Luis
P.O. Box 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
TEIL (Masacon Division)
25 Moronu Street, Suru-Lere,
P.O. Box 5707
Lagos
Tel: 34545
Cable: THETEIL Lagos

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, 87-86-88, 87-18-45, 88-91-71, 83-81-12, 83-82-12
Cable: ELEMEX Manila

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

Hewlett-Packard Far East Area Office
P.O. Box 87
Alexandra Post Office
Singapore 3
Tel: 633022
Cable: HEWPACK SINGAPORE

SOUTH AFRICA
Hewlett Packard South Africa (Pty.), Ltd.
P.O. Box 31716
Braamfontein Transvaal
Milnerton
30 De Beer Street
Johannesburg
Tel: 725-2080, 725-2030
Telex: 0226 JH
Cable: HEWPACK Johannesburg

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

THAILAND
UNIMESA Co., Ltd.
Chongkoinee Building
56 Suriwongse Road
Bangkok
Tel: 37956, 31300, 31307, 37540
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
Co-ordination Office for Mediterranean and Middle East Operations
Piazza Marconi 25
I-00144 Rome-Eur, Italy
Tel: (6) 59 40 29
Cable: HEWPACKIT Rome
Telex: 61514

OTHER AREAS NOT LISTED, CONTACT:
Hewlett-Packard
Export Trade Company
3200 Hillview Ave.
Palo Alto, California 94304
Tel: (415) 326-7000
(Feb. 71 493-1501)
TWX: 910-373-1267
Cable: HEWPACK Palo Alto
Telex: 034-8300, 034-8493

I/O OPERATIONS

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 invoked through the use of 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.

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

THE OUTPUT STATEMENT

The OUTPUT statement is a general-purpose means 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 invoked through the use of the optional conversion table parameter.

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

THE CMD (COMMAND) STATEMENT

The CMD statement is a specialized output instruction used only to send commands to the ASCII Bus. Select code for the ASCII Bus is assumed to be 13, and the CMD statement cannot be used for any other select code or device.

CMD address [[, message] , address] . . .

THE STAT FUNCTION

The STAT function returns a 4-bit code of operational status (on, off, wait, etc.).

STAT select code

THE RBYTE (READ BYTE) FUNCTION

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

RBYTE select code

I/O OPERATIONS

THE ROT (ROTATE) FUNCTION

The ROT function performs right rotation on the binary equivalent of expression 1, the number of positions represented by expression 2.

ROT (expression 1, expression 2)

THE INOR (INCLUSIVE OR) FUNCTION

The INOR function combines binary equivalents of expression 1 and expression 2 in an 'inclusive or' logic operation.

INOR (expression 1, expression 2)

THE BIAND (BINARY AND) FUNCTION

The BIAND function combines binary equivalents of expression 1 and expression 2 in an 'and' logic operation.

BIAND (expression 1, expression 2)

THE WBYTE (WRITE BYTE) FUNCTION

The WBYTE function outputs the binary equivalent of an expression, length 8 bits. For output to binary storage device; also provides increased control of print format.

WBYTE expression

THE SPA (SPACE) FUNCTION

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

SPA expression

THE LIN (LINE FEED) FUNCTION

The LIN function advances printer or typewriter the number of lines represented by the expression. If expression $\geqslant 0$, the carriage returns; if expression <0, carriage does not return.

LIN expression

11272B & OPTION 272