



MONROE

instruction
book

Model **300** I/O
WRITER



for use with
Monroe Series **1800**
electronic
programmable
printing
calculators

Monroe, The Calculator Company



MONROE

instruction
book

Model **300** **I/O**
WRITER

for use with
Monroe Series **1800**
electronic
programmable
printing

Monroe, The Calculator Company

ILLUSTRATIONS

	PAGE
1. Monroe Model 300 I/O Writer with 1800 Series Calculator	4
2. Convenience Power Supply	6
3. Power Line and I/O Cable Connections	8
4. Demonstration Printout	10
5. Demonstration Printout with Notations	12
6. Demonstration Data	18

CONTENTS

SECTION 1

	PAGE
INTRODUCTION	5

SECTION 2

USER INFORMATION	7
A. STARTUP	7
The System	7
Power Lines and I/O Cables	7
Connecting the Power Lines and I/O Cables	7
Turn On	9
B. OPERATION	9
Margin Alignment	9
Tab Clearance	9
Reading in the Program from Magnetic Cards	10
Starting the Program	11
C. MESSAGES	11
Programmed Messages	11
Entered Messages	11
Ready Signal for Entered Messages	12
How to Type Entered Messages	13
Entered Messages in the Demonstration Program	13
Correcting Entered Messages	13
Typed Information	15
Unscheduled Typed Information	15
Scheduled Typed Information	15
D. NUMERIC ENTRY	15
Numeric Entry from the Calculator	16
Correcting a Numeric Entry from the Calculator	16
Numeric Entry from the I/O Writer	16
Ready Signal for Numeric Entry from the Writer	17
Correcting a Numeric Entry from the Writer	17
E. NUMERIC OUTPUT	19
Output in Mixed Decimal Format	19
Output According to Calculator D. P.	20
Output in Exponential Format	21
Typed Record of Data Entered from the I/O Writer	21
F. CALCULATOR OPERATION WITH THE I/O WRITER OFF OR DISCONNECTED	24

SECTION 3

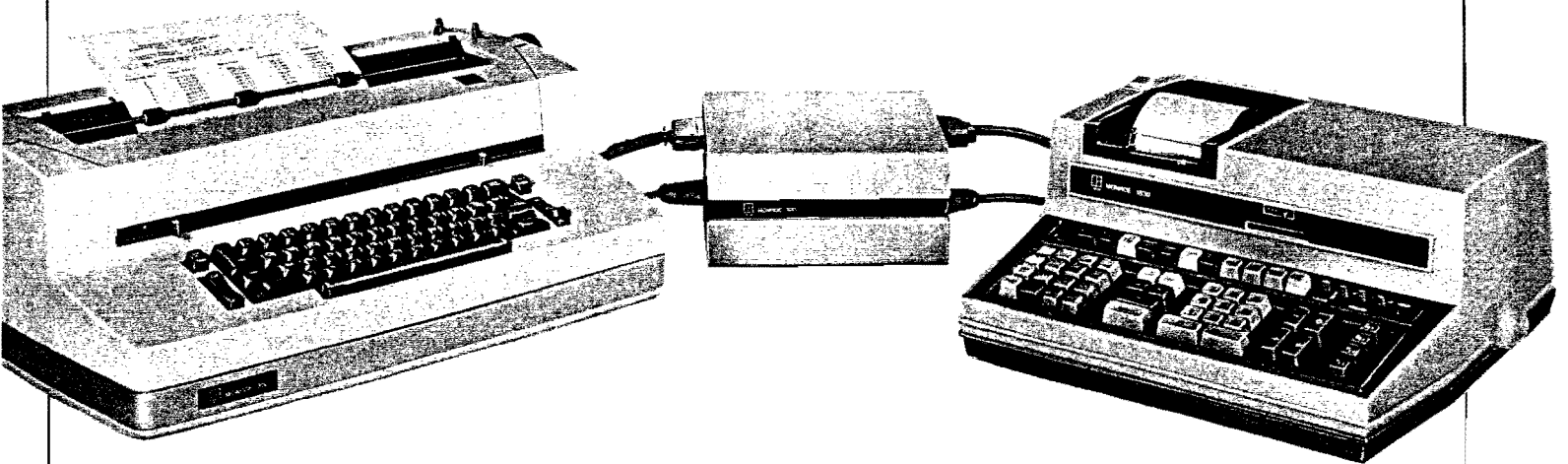
	PAGE
PROGRAMMING INFORMATION	25
A. I/O WRITER INTERFACE PROGRAM	25
B. SPECIAL INSTRUCTIONS	26
C. MESSAGES	26
Message Initialization	26
Storage of Messages	28
Limits on Total Message Length	29
Registers Used for Messages or Data	29
Message Initialization as Part of Calculational Program	30
Message Modification	30
Octal Numbering for Message Numbers	31
Message Augmentation	32
Entered Messages	32
Message Output	33
D. NUMERIC INSTRUCTIONS	33
Numeric Entry from the I/O Writer	33
Numeric Entry from the Calculator	34
Numeric Output from the I/O Writer; Mixed Decimal Format	34
Numeric Output from the I/O Writer According to Calculator D. P.	34
Numeric Output from the I/O Writer; Exponential Format	34

SECTION 4

PROGRAMMING TECHNIQUES	37
Steps in Programming	37
Preparing an I/O Writer Program for the 1810	38
Demonstration Printout Messages	39
Demonstration Program	40

APPENDIX A. 1810/1830 I/O WRITER INTERFACE ROUTINE	49
APPENDIX B. 1860/1880 I/O WRITER INTERFACE ROUTINE	57

Figure 1. Monroe Model 300 I/O Writer with 1800 Series Calculator



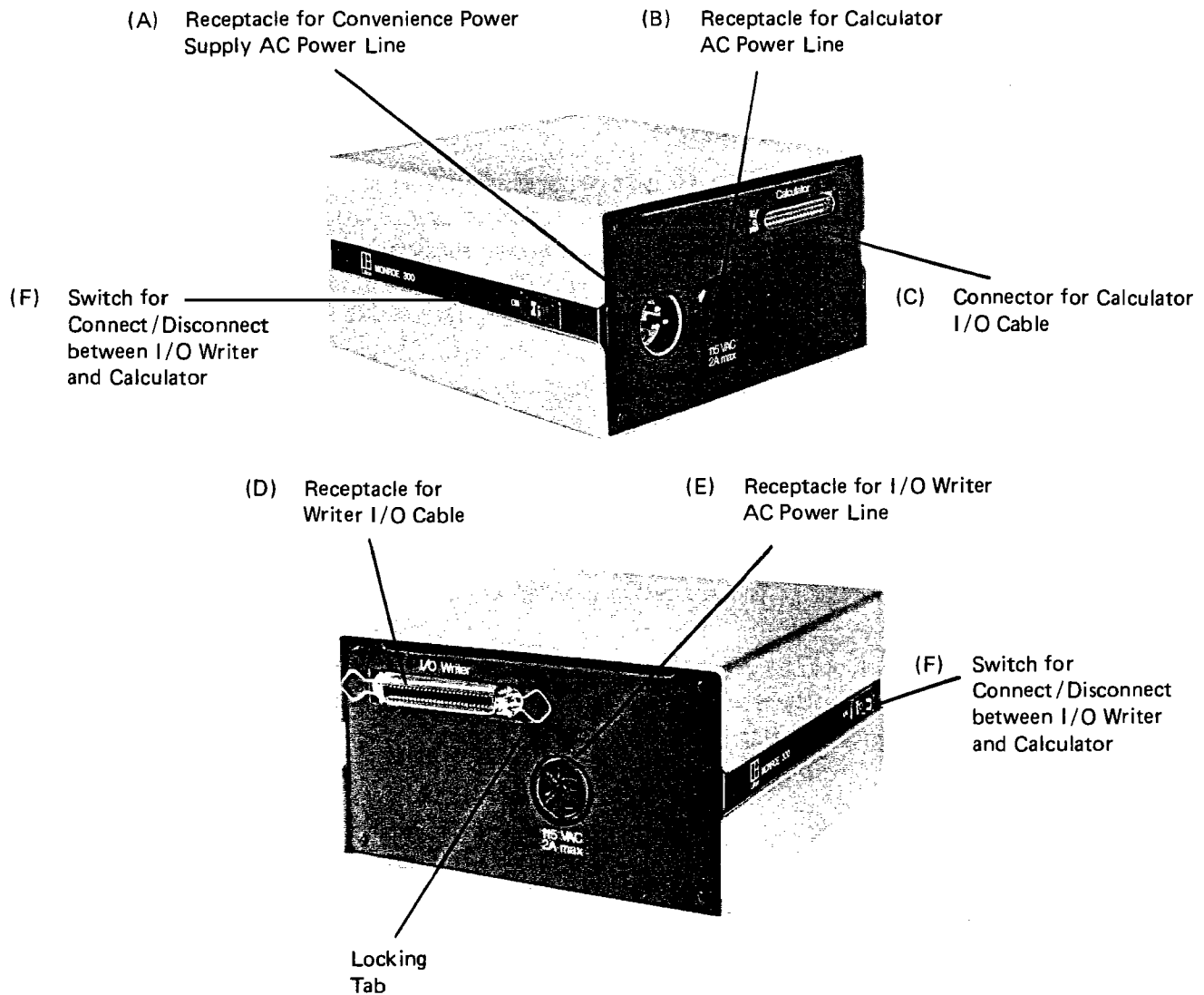
INTRODUCTION

The Monroe Model 300 Input/Output Writer (I/O Writer) used with any Monroe 1800 Series Programmable Calculator can perform many functions which expand the utility and scope of your 1800 calculator. Teamed together, as illustrated in figure 1, your calculator and I/O Writer can perform complex calculations and data manipulations and then print out the data and results in any format you select, with appropriate headings, messages, and identifiers. Numeric data can be entered from the Writer or the calculator, and worded (or numeric) messages can be entered from the Writer. Messages can be changed, added, or updated, as desired. Furthermore, the I/O Writer or Monroe 1800 can be used independently as a typewriter or a programmable calculator, respectively.

The I/O Writer is easy to use. For those who will not be preparing programs, Section 2 contains all the information needed to run the Writer. (Calculator keytop and magnetic card usage is discussed in the Operating Instructions Manual for your calculator.)

The I/O Writer is also simple and straightforward to program. The remainder of this manual, beyond Section 2, tells how to program the Monroe 1800 Series calculators for use with the I/O Writer.

Figure 2. Convenience Power Supply



USER INFORMATION

A STARTUP

The System

The Monroe Model 300 I/O Writer system consists of the I/O Writer and a Convenience Power Supply. These units are shown connected to a Monroe 1800 Series calculator in figure 1. The Convenience Power Supply is used to connect the calculator to the Writer, and to supply power to the calculator and Writer. The calculator-receptacle and Writer-receptacle sides of the Convenience Power Supply are shown in figure 2.

Power Lines and I/O Cables

Two AC power lines and two Input/Output (I/O) cables, are supplied with your Monroe 300 system:

- An I/O Writer AC power line (permanently connected to the Writer)
- A Writer I/O cable
- A Convenience Power Supply AC power line
- A calculator I/O cable

The AC power lines transmit electric power to the equipment; the I/O cables transmit information between the I/O Writer and calculator via electrical pulses.

Connecting the Power Lines and I/O Cables

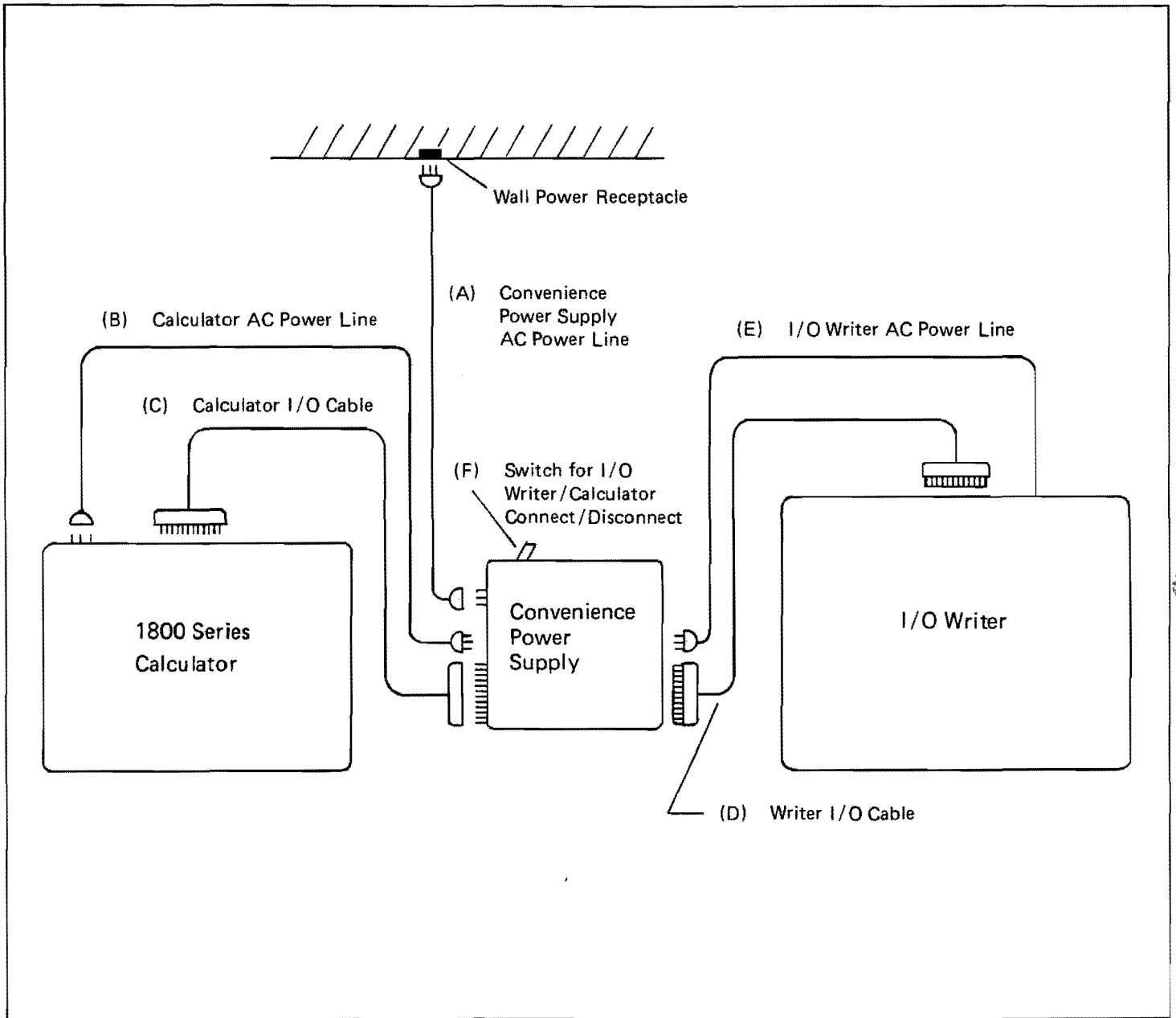
The power lines and I/O cables are connected as shown in figure 3. The following procedure should be followed in connecting power lines and I/O cables:

1. Make sure that the Convenience Power Supply's AC Power Line (A) is not connected to your wall outlet.
2. Check to see that the calculator and I/O Writer are both turned off.
3. Connect the I/O cables, (C) and (D). These cables are designed so that they cannot inadvertently be interchanged, one with the other. The calculator I/O cable has one male and one female connector; consequently, this cable can only connect to the equipment in one way. The Writer I/O cable has identical connectors on both ends, and can be connected either way. Finally, the connectors have shaped outer casings; this ensures that each connector can only be connected "right side up".

Note: The Writer I/O cable connectors must be locked into place using the locking tabs (see figure 2). After plugging in the connector, the tabs are lifted out and pressed into place in the notches on the connectors.

Continued

Figure 3. Power Line and I/O Cable Connections



4. Connect the calculator and I/O Writer power lines, (B) and (E). The calculator power line was supplied with your calculator. *Note that this line now connects to the Convenience Power Supply, not to your wall outlet.*
5. Connect the Convenience Power Supply AC Power line, (A), to your wall outlet.

Turn On

To turn on the system, simply turn on your calculator and I/O Writer. *If you want to use the calculator or I/O Writer separately, turn the switch, (F), on the Convenience Power Supply to the off position.* If you want to use the calculator and I/O Writer together, turn the switch to the on position.

B OPERATION

To illustrate the operation of the Monroe Model 300 I/O Writer, a demonstration I/O Writer printout will be used. This printout is shown in figure 4. Some of the terms in this printout may not be familiar, but all the terms will be discussed in the following pages.

Margin Alignment

In order to ensure proper alignment of typed numbers and messages, particularly on preprinted forms, it is necessary to place your sheet in the I/O Writer with care. The left Margin Stop location will be given on the program operating sheet. (In some cases, you will be told to align on a specific character or mark on the preprinted form; in other cases you would align on the left edge of the printed matter.) Once aligned, set the Paper Edge Guide to the proper position, to ensure the same alignment for additional sheets.

Vertical alignment would be performed in the normal way.

For programs which print out headings and other messages on blank sheets, alignment is somewhat less critical; nevertheless, some care must be taken to give a desired margin width. On some program operation sheets, the left Margin Stop location and the Paper Edge Guide location will be given. (For the example of figure 4, the Paper Edge Guide should be set on the fourth mark from the left. The left Margin Stop is set at 20.)

Tab Clearance

Since most I/O Writer programs automatically set tab stops and use these for formatting, you should always clear all tab stops before running your program.

If for any reason during operation you should reset your left margin after having run a printout on a sheet, you should clear all tabs before executing your next printout. (This is necessary because tabs set by the program are set in relation to the left margin; if tabs aren't cleared, you may have two sets of tab stops on your next program execution.)

Reading in the Program from Magnetic Cards

With the calculator on, the program is read from magnetic cards into the calculator as discussed in your calculator Operating Instructions Manual. Read the accompanying program operation sheet carefully for any special instructions such as SENSE switch placement, decimal point setting, or initializing steps.

Figure 4. Demonstration Printout

A Monroe Model 300 I/O Writer with an 1800 Series calculator can perform many functions. It can enter messages. Please type below:

Your Name: James Henry

Company: Washington Square, Inc.

It can handle numerical entry and output. Now enter a number on the calculator. Enter another number from the I/O Writer.

X	Y	X*Y	X/Y
123.456	2.0000000	246.9120	6.172800000 E 01
654.321	2.0000000	1308.6	3.271605000 E 02
123.456	2.0000000	246.912000	6.172800000 E 01
654.321	2.0000000	1308	3.271605000 E 02

Tabs for headings and data columns can be set automatically and then used as above. Note the different output formats. X has been printed according to a programmed format; X*Y according to calculator D.P.; and X/Y in scientific format.

Thank you, James Henry of Washington Square, Inc.

Starting the Program

Refer to your operating instruction sheets to determine the starting procedure. Depending upon your specific calculator (Model 1810, 1830, 1860 or 1880), you may be instructed to depress RESUME, to jump or branch to a starting address, or to depress a "Start" key or other user-definable key.

Once activated, the I/O Writer may print out a message, a line of tabulated headings, a set of data, or other matter.

In the demonstration printout of figure 5, the first message (identified as Message 0) includes all the typed matter through *Your Name:* (The technique for typing in a name will be discussed shortly.) Once the name is properly typed in, the next message is printed. This second message (Message 1) consists of the single word *Company:* Again, once the name of your company is correctly typed in, a third message is printed. This third message runs from *It can handle numerical . . .* through the last portion of the heading: *X/Y*. The fourth message isn't a printed message; rather, it executes a carriage return, space-up and tab to point "a", then stops, ready for entry of numerical data.

C MESSAGES

Programmed Messages

As you can see, a programmed message (called a message string) may consist of words, sentences, numbers, headings, symbols, carriage returns, horizontal tabulations, space-ups, and more. These messages are commonly stored on magnetic cards and entered into the calculator as part of the program. However, it is also possible for you to change some or all of the messages, or write additional messages. These capabilities will be discussed in Section 3, Programming Information.

Entered Messages

In addition to the programmed messages discussed above, programs can be prepared which permit the entry of messages from the I/O Writer keyboard at appropriate halts in the printout execution. These entered messages may be names, dates, identification numbers, product items, or any other desired information. The demonstration program has this capability. Your name (or any other message) may be entered at the pause which occurs after Message 0 is printed. Your company name (or any other message) may be entered after Message 1 (the word *Company:*) is printed.

Figure 4 shows a name (James Henry), and company (Washington Square, Inc.), typed out after Messages 0 and 1, respectively. But these entered messages are not only typed out, they are stored in calculator memory and may be recalled and printed any number of times on the sheet, or if the program is so written, on subsequent sheets. In figure 4, the name and company are typed again on the last line. Note that in the last line, the following words are *programmed* messages, just as Messages 0 and 1 were programmed messages (see also, figure 5):

Thank you,
of

Ready Signal for Entered Messages

When the program is ready for you to enter a message, a signal will often be programmed at that point: the ball-shaped typing element will momentarily shift, indicating entry readiness. The idle light on the calculator will be off, indicating that the calculator is in active communication with the I/O Writer, ready to receive a message.

It should be noted that the typing-element signal may not always be programmed; however, the content of the printout and the accompanying program operation sheet will indicate when a message is to be entered.

Figure 5. Demonstration Printout with Notations

Message 0 —

A Monroe Model 300 I/O Writer with an 1800 Series calculator can perform many functions. It can enter messages. Please type below:

Your Name:

Message 1 —

Company:

It can handle numerical entry and output. Now enter a number on the calculator. Enter another number from the I/O Writer.

X	Y	X*Y	X/Y
---	---	-----	-----

Ⓐ

Tabs for headings and data columns can be set automatically and then used as above. Note the different output formats. X has been printed according to a programmed format; X*Y according to calculator D.P.; and X/Y in scientific format.

Thank you, of

How to Type Entered Messages

There are only two requirements for typing in an entered message:

- Any limits on the permissible number of characters must be observed. This information will be provided on your program operation sheet.
- Margin Release must be depressed just prior to striking the last character of your message. Then with Margin Release down, the last character is depressed. This dual action informs the calculator that the entered message is ended. The program may immediately continue; it may type out the next programmed message, perform a calculation, or signal for data entry.

It is important to release the Margin Release key immediately after depressing the last character of your message. Otherwise, if the first portion of the next programmed message is a Carriage Return, the Carrier may move past the left margin on its return, since Margin Release is depressed. It may take a little practice to depress Margin Release, strike your last character key, then release the Margin Release key quickly.

Should you forget to use the Margin Release with the last character of your message, you can simply depress Margin Release with the space bar. (However, the space will be entered as a character in your message.)

Entered Messages in the Demonstration Program

In the sample demonstration of figure 4, *James Henry* would be typed in the normal manner except that the Margin Release must be depressed before striking the y of Henry. Again, the Margin Release must be depressed before striking the . in Inc.

The limit on permissible characters is determined, in this sample, by the fact that your name and company name must fit on a single line (bottom line) together with the programmed messages *Thank you*, and *of*. The limit on total characters, including spaces, in the two messages is approximately 46, dependent on location of the right Margin Stop.

Both entered messages are signaled in this sample program, by the momentary shift of the typing element.

Correcting an Entered Message

If you enter a message incorrectly it may be corrected by use of the Backspace key. Before describing the correction technique, a set of "special characters" must be defined. The following are considered special characters.

Carriage Return, Index, Space, Backspace, Tab, and Underscore.

In the following discussion, assume the entered message was a heading, as follows:

QUANTITY TOTAL PRICE

To clarify the discussion, each space between words will be designated by an asterisk:

QUANTITY*****TOTAL*PRICE

Continued

Typed Information

Unlike entered messages, typed information is not transmitted to the calculator, nor stored, for future use. It is a "one-time" message.

Unscheduled Typed Information

The I/O Writer can be operated as a typewriter by turning off the switch (switch F in figure 2) on the Convenience Power Supply. This may be done at any time during running of a program when the idle light on the calculator is on. (When the idle light is on, it indicates that the calculator isn't communicating with the typewriter or performing a computation, at the moment.) Once the desired information has been typed onto your sheet, the switch on the Convenience Power Supply should be turned on and the program continued with the next operation called for on your program operation sheets.

Scheduled Typed Information

In some programs, provision will have been made for entering typed information at certain pauses in program execution. In these instances, you do not have to turn the Convenience Power Supply switch off. Just type in your information as called for on the program operating sheets, and then continue the program with the next required operation.

Tab Setting Caution


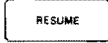
When using the I/O Writer for typing information, as above, you may perform any normal typing operation, with the exception of setting tabs or changing tab settings. Your prepared program may have executed a tab set instruction, or instructions, prior to your typed-information operation. These automatic instructions are often performed only once at the start of a program, and must hold throughout the program. If you change tab settings arbitrarily, you may have a combination of old and new settings after you've typed in your information. *Therefore, great caution should be used in changing or adding tab settings, if it is done at all.*

D NUMERIC ENTRY

Numeric data may be entered from the calculator, or from the I/O Writer keyboard, as called for by your program operation sheets. Such data can be used by the calculator in performing computations. The distinction should be noted between data entered for calculations and numbers entered as messages (via the above procedures). Numbers entered as messages are treated solely as messages, and cannot be used in computations. Your program operation sheets will distinguish between messages and data.

Continued

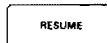


Numeric Entry from the Calculator

To enter a number from the calculator, proceed as you would for normal calculator operation. You may enter the number as a mixed decimal (or in scientific format using the  key, for Models 1860 and 1880). When the number has been entered, depress  to continue program operation. The number may be immediately typed out on the I/O Writer, if called for by the program.

If the number entered is to be typed out by the I/O Writer, the number must not exceed the limits (number of places, including sign and decimal point) programmed for the particular typed output. (See "Numeric Output")

If the number entered is not to be typed out by the I/O Writer, the limitations on numeric entry for your calculator hold.

Correcting a Numeric Entry from the Calculator

If you enter a number incorrectly from the calculator, so long as you have not yet depressed , depress  (1860 and 1880) or  (1810 and 1830), and re-enter the number as in normal calculator operation.

Numeric Entry from the I/O Writer

You may enter a number from the I/O Writer keyboard (when entry from the I/O Writer is called for) either as a mixed decimal or in exponential (scientific) format. This is true for all 1800 series calculators. *In either case, the Margin Release must be depressed just prior to striking the last digit to be entered for that number; the Margin Release should be released immediately following the entry of the final digit.* (This "entry termination" procedure is analogous to that for entered message termination, discussed under "How to Type Entered Messages".)

The entered number is limited to 13 digits, or 12 digits if the first character is a decimal point, as in entry of numbers from the calculator keyboard.

For those not familiar with exponential format, also called scientific format, a number in this format is represented by a single digit before the decimal point and a predetermined number of digits after the decimal point. This number may be positive or negative. To determine the actual location of the decimal point, the number is multiplied by a power of 10 (either positive or negative).

For example:

123.45 would be 1.2345×10^2 (i.e., 1.2345×100)

0.012345 would be 1.2345×10^{-2} (i.e., 1.2345×0.01)

In exponential format, 10^2 is printed as Exponential 02 or E 02, and 10^{-2} as E-02.

Therefore:

123.45 would be printed as 1.2345 E 02

0.012345 would be printed as 1.2345 E-02

123,450,000,000 would be printed as 1.2345 E 11

To enter a number from the I/O Writer in mixed decimal format (for example, 123.45) use the *period as a decimal*, and the *hyphen as a minus* for entering negative numbers.

In addition, for exponential format entry, use the *upper case E for the exponential*.

For example:

-0.01234 would be entered as a mixed decimal by striking:

Hyphen period 0 1 2 3 { Margin₄ Release }

-0.01234 would be entered in exponential format by striking:

Hyphen 1 period 2 3 4 E hyphen { MR₂ }

Notice that in exponential format, both the sign of the number and the sign of the exponent are negative, in this case. To enter exponents from 0 to 9, you need only type the single digit. For the above entered number, -1.234E-2 would be typed on your sheet. However, for an output in exponential format, -1.234 E-02 would be typed out. (The values under the X/Y column in figure 4 are outputs in exponential format. Output formats are discussed in Section E, "Numeric Output".)

Ready Signal for Numeric Entry from the Writer

When the program is ready for you to enter numbers from the I/O Writer, a signal will often be programmed at that point: the ball-shaped typing element will momentarily shift, indicating entry readiness. (This is the same signal as that for entered messages.) The idle light on the calculator will be off, indicating that the calculator is in active communication with the I/O Writer, ready to receive data.

It should be noted that the above typing-element signal may not always be programmed; however, the content of the printout and the accompanying program operation sheet will indicate when data is to be entered.

Correcting a Numeric Entry from the Writer

If you enter a number incorrectly from the I/O Writer, so long as you have not yet entered the last digit (or decimal point) using Margin Release, you can clear the *entire* number from the calculator Entry Register (E-Register) by *depressing backspace once*. You may then type the entire number correctly.

Though one backspace removes the number insofar as the calculator is concerned, you will probably want to erase or white-over the incorrect number and backspace to the first digit position, so that when you retype the corrected number it will appear in place on the sheet. (These additional backspaces clear the E-register repeatedly, but no harm results since the E-register was set to zero by the first backspace.)

Caution: When entering data in numeric entry mode from the I/O Writer keyboard, striking backspace prior to terminating that numeric entry (Margin Release plus final digit), will clear the entered data from the E-register.

During numeric entry mode from the I/O Writer, only 0 through 9, hyphen, period, upper case E, and backspace are used for data entry. Striking any other *character* key will not affect the data.

Figure 6. Demonstration Data

A Monroe Model 300 I/O Writer with an 1800 Series calculator can perform many functions. It can enter messages. Please type below:

Your Name: James Henry

Company: Washington Square, Inc.

It can handle numerical entry and output. Now enter a number on the calculator. Enter another number from the I/O Writer.

	X	Y	X*Y	X/Y
a)	123.456	1.00000000	123.4567	1.234567000 E 02
b)	12345.600	1.0000	12345.6000	1.234560000 E 04
c)	*****	1.0000	123456.7000	1.234567000 E 05
d)	0.0001	1.0000	0.0001	1.234000000 E-04
e)	0.0000001	1.0000	0.0000001	1.234000000 E-07
f)	*****	1.0000	0.00000001	1.234000000 E-08
g)	123.456	1.00E10	*****	1.234567000 E-08
h)	123.456	1.00E-11	0.000000001	1.234567000 E 13
i)	123.456	-1.00E-11	-0.000000001	-1.234567000 E 13
j)	123.456	1.00E-12		
k)	123.456	1.0000000000		123.4567

Tabs for headings and data columns can be set automatically and then used as above. Note the different output formats. X has been printed according to a programmed format; X*Y according to calculator D.P.; and X/Y in scientific format.

Thank you, James Henry of Washington Square, Inc.

E NUMERIC OUTPUT

The I/O Writer can output numeric data under any of three formats, as called for by the calculator program:

- Mixed Decimal Format
- Calculator Controlled Decimal Places (via D. P. Setting)
- Exponential (Scientific) Format

The program operation sheets will specify the output format and any limitations that apply, where necessary.

Output in Mixed Decimal Format

In mixed decimal format, data is typed from the I/O Writer according to a programmed number of decimal places, as well as a programmed total field length including sign, decimal point, and all digits. The maximum field length which can be specified is 12 places. Though a plus sign isn't typed out, the sign place must always be counted in determining field length. If the programmed instruction calls for zero decimal places, the decimal point will not be typed out, but the decimal point location must still be included in considering field length.

Example: 1234.5678 has a field length of 10, not 9, because the sign location, though not typed, must be considered; i.e.: □1234.5678.

All numbers printed in mixed decimal format will have at least one digit before the decimal. For example: 0.00124.

If the number to be printed in mixed decimal format has more digits after the decimal point than was specified by a program instruction, only the prescribed number of digits will print. For example, in figure 6, the X values are printed with three places after the decimal point and with a total field length of 10. In row (a), 123.4567 was input from the calculator; note that the printout under column X is limited to three decimal places: 123.456.


If the number to be printed has more digits before the decimal point than permissible, asterisks will print out, indicating overflow on the I/O Writer printout. But the data will be correct in the calculator and can be used for computations. For example, in row (b), the X value has five digits before the decimal point; it fills the specified field length of 10 (including non-printed plus sign). In row (c), the X value entered from the calculator had six digits before the decimal point (123456.780), requiring a total field length of 11. Consequently, a row of asterisks printed out, indicating Overflow. But the remaining operations (entry of Y; X multiplied by Y; X divided by Y) could be executed, since the entered X value was retained in the calculator.


If a decimal too small for the predetermined format is to be typed out, the programmed decimal point setting will temporarily be overridden and the decimal point will shift to the left a sufficient number of places to permit typing of the first significant decimal digit. 0.0001234 was entered as the X value for row (d); it may be

Continued

seen that the decimal point has shifted one place to the left to permit printing the first decimal digit. In row (e), with entry of a smaller decimal value, the decimal place has shifted left as far as possible for a field length of 10 (including non-printed plus sign). A further shift leads to typing of a line of asterisks (Underflow) as in row (f). But note that the entered number *was* retained in the calculator for further computations.

Output According to Calculator D. P.



The output format can be controlled by the  key (or equivalent code instruction) of the calculator, if so programmed. The field length for the typed output will always be 12 places, including sign and decimal point. (Again, the plus sign isn't typed, but the sign place must be counted as one of the 12 places.) All numbers printed according to D. P. will have at least one digit before the decimal point.







This format permits you to manually change the decimal setting during the course of a program execution, by depressing the  key and appropriate numeral key, where the program operation sheets permit such changes.

In the demonstration printout of figure 6, $X*Y$ is printed according to D. P. In row (a), an X value of 123.4567 was entered in the calculator. (As discussed above, the X value printed out as 123.456 since in the mixed decimal format used for printing X, only three decimal places were specified.) After multiplying X by a Y value of 1.0 (entered from the I/O Writer keyboard) the product is printed as 123.4567, since D. P. was set at 4.

If the number to be printed according to D. P. exceeds the permissible field length (Overflow) a row of asterisks will print out. The number is still retained by the calculator for further computations. For example, in row (g) of figure 6, the $X*Y$ value to be printed was 123.456 multiplied by 10^{10} . Notice that though the product wasn't printed, the computations continued, and X/Y did print.

If the number to be printed according to D. P. is too small to print a significant decimal digit, the D. P. setting will be temporarily overridden, and the decimal point will shift left a sufficient number of places to permit printing of the first significant decimal digit. In figure 6, row (e), the decimal point has shifted left three places beyond the setting of 4.

If the decimal point must shift left so many places that the total field length would be exceeded, *no printing occurs*, the idle light flashes on the calculator, and Overflow prints out on the calculator. To recover from this condition, depress  (1860 and 1880) or  (1810 and 1830). You must then execute an appropriate Jump or Branch instruction from the calculator keyboard. This instruction will be given on the program operating sheets, where necessary.

In the sample program of figure 6, row (h), the decimal point for the product $X*Y$ has shifted so far to the left that the entire field length of 12 is utilized. (Again, recall that a place is designated for the sign, though + is not printed. To show this, row (i) gives the same product, but with a - sign.) In row (j), the field has been exceeded and no printout occurs. In this sample program, after depressing  or , depressing , , ,  prepares the calculator for a new X entry.

Output in Exponential Format

The printed output will appear in exponential format when called for by the program. In this format, the total field length is 17. This includes the decimal point, and a two-digit exponent with its sign. (Again, the plus sign does not print; a space before the exponent implies a positive exponent.)

In exponential format, numbers with an exponent greater than +99 or less than -99 will not be printed. (This matches the calculator's limits; when numbers beyond these limits are attempted, the calculator will Overflow.)

In the sample printout, figure 6, the values of X/Y are printed in exponential format.

Typed Record of Data Entered from the I/O Writer

In the sample printout of figure 6, the values of Y were input from the I/O Writer keyboard as discussed under "Numeric Entry from the I/O Writer". The typed data is simply a record of the entered number; the format is typed as desired. But there is an important limitation: the typed data should not go beyond the next tab stop. If it does, any succeeding programmed tab instruction will seek the next tab stop beyond, and data may be printed out in the wrong column, or off the sheet on the platen. For example, in figure 6, row (k), a Y value of 1 was entered with so many zeroes after the decimal place, that the next tab stop (for X*Y printouts) was passed. Notice that the value 123.4567 was printed in the X/Y column. The value of X/Y was typed out on the platen.




Where such tab stop limitations are of importance, they will be included on the program operating sheet.

Table 1 summarizes the information on numeric entry and numeric output discussed in the preceding pages.

Continued

Numeric Entry

From Calculator:

- Enter as you would for normal calculator operation. Terminate entry with  .
- If the number is to be typed out by the I/O Writer, output format limits should be considered.
- To correct entry error, depress  (1880 and 1860) or  (1810 and 1830).

From I/O Writer:

- Enter number in either mixed decimal or exponential format.
- Terminate entry with Margin Release/Final Digit/Off-Margin Release.
- Upper case E gives exponential.
Hyphen gives minus sign.
Period gives decimal point.
- To correct entry error, depress backspace once. This clears number from the E-register. The entire number must be re-entered correctly. Further backspacing for correct placement on sheet will have no adverse effect.

Numeric Output from I/O Writer

Mixed Decimal Format:

- Total field length including sign and decimal point is programmed; number of decimal places is programmed. Maximum field length is 12 including sign and decimal point.
- Overflow: asterisks print out. Data is retained in calculator and can be used for computations.
- Underflow: asterisks print out. Data is retained in calculator and can be used for computations.

D. P. Format:

- Decimal places according to or equivalent programmed instruction. Manual change via possible, if permitted by program operating instructions.
- Field length, including sign and decimal point, is fixed at 12.
- Overflow: asterisks print out. Data is retained in calculator and can be used for computations.
- Underflow: no printout occurs. Idle light flashes on calculator and OVERFLOW prints. Data must be re-entered. To recover, depress (1810 and 1830) or (1860 and 1880). Then Jump or Branch as stated on the program operation sheets.

Exponential Format:

- Field length is 17, including sign, decimal point, and a two-digit exponent with its sign.
- Limitations: number must not have an exponent greater than +99 or less than -99.

F CALCULATOR OPERATION WITH THE I/O WRITER OFF OR DISCONNECTED

Programs prepared for the Monroe 300 I/O Writer can be used with your calculator even when the I/O Writer is turned off or is disconnected.

Operations which normally require the I/O Writer are handled as follows:

- Numeric output which was to be printed by the I/O Writer (in mixed decimal, D.P., or exponential format) will print on the calculator tape according to calculator D.P.
- Numeric data which was to be entered from the I/O Writer is entered from the calculator as if at a HALT. Depress to continue.
- If any messages were to be printed, or if messages were to be entered, the instruction will be bypassed and the calculator will continue the program. (Depression of is *not* required.)

PROGRAMMING INFORMATION

There are three major topics to be considered in programming for the 300 I/O Writer. These are:

The I/O Writer Interface Program which permits communication between the calculator and Writer. (This is also called the I/O Writer Software Interface.)

Special Instructions which permit input of programmed messages, addition and rewriting of messages, and entry and output of data according to a selected format.

Planning and Preparation of Messages

There are some differences in the details of these topics in programming for the Models 1810/1830, and programming for the 1860/1880. These differences will be clearly noted in the discussion.

A I/O WRITER INTERFACE PROGRAM

The I/O Writer Interface Program is a prepared program which you would not normally alter. The program contains 840 steps (843 steps for the 1810/1830) and it uses absolute, rather than symbolic, addressing. (However, your calculation program can use symbolic addressing.) The loading instructions follow:

For the 1860 or 1880

The Interface Program must always be loaded starting at branch point 00. The program runs from step 00 through step 839. Your calculation program can be loaded starting at step 840 (branch point 84), or at any point beyond branch point 84. The calculation program can be included on the last magnetic card containing the Interface Program, or it can be loaded from a separate card at step 840 or beyond.

For the 1810 or 1830



The Interface Program runs from step 00 through step 348, and from step 510 through step 1003. The Interface Program *must* be loaded at these locations. (Step 350 through 509, as well as steps 1004 on, may hold your calculation program, or may contain NOOP's.)

On the 1810, you must load your Interface Program and calculation program together in a single LOAD operation. Therefore, the magnetic cards must contain both programs in appropriate order.

On the 1830, you may load the Interface Program and calculation program in parts by branching to the appropriate branch points.

The I/O Writer Interface Program (I/O Writer Software Interface) code listings for the 1810/1830 and for the 1860/1880 calculators are presented in Appendices A and B, respectively. Many of the instruction codes in these listings may be unfamiliar. In fact, some codes which appear familiar will perform in a different

Continued

manner here than in normal programming, though they may still be entered by the appropriate key depression, or via . When loading the listed program into your calculator for ultimate writing onto a magnetic card, simply use  to enter any codes that you cannot associate with a keytop.

It may prove advantageous to write the Interface Program onto magnetic cards, and cut the A and B tabs of each card for a permanent set of I/O Writer Interface cards.

B SPECIAL INSTRUCTIONS

There are a number of special instructions which are required in your calculation program in order to perform the following operations: printing messages, entering data from the I/O Writer, and printing out data on the Writer according to one of three formats.

There are some further instructions which are used for creating programmed messages, modifying programmed messages, and adding new messages. These instructions may not always appear as steps in your calculational program.

In some cases, the instructions will vary depending on whether or not you want the I/O Writer to signal its readiness to accept input data (or messages). A signaled instruction executes the oscillation of the typing-element ball, referred to in Section 2.

The instruction codes are different for the 1810/1830 and the 1860/1880. Both sets of codes are given for each instruction, and a summary table of all the special instructions, Table 2, appears at the end of this Section.

C MESSAGES

Message Initialization

Once the I/O Writer Interface Program has been read into the calculator, you can create your set of programmed messages using the message input initialization instruction. This instruction need not be a part of the calculation program, and frequently is not. It can be entered and executed as a small, separate program. The instruction is always signaled; it is a double instruction (two-step instruction).

	For 1830	For 1860/1880
Message initialization,	202	201
signaled input.	146	302

The initialization instruction may be executed, and messages entered, by performing the following operations:

1. Load the I/O Writer Interface Program into the calculator.
2. Determine where an unoccupied location exists in program memory.

3. Depress , where nn is an unoccupied branch point.
4. Place the RUN/STEP/LOAD switch in LOAD.
5. Depress , or,
 , as appropriate.
6. Place the RUN/STEP/LOAD switch in RUN. Jump to nn and depress .
The idle light will go off and remain off, indicating that the calculator is communicating with the I/O Writer. The typing-element ball will oscillate, indicating readiness for entering programmed messages.
7. If more than one message is to be entered, the SENSE switch must be up. If only one message is to be entered, the SENSE switch must be down.
8. Messages are now entered from the I/O Writer. Type each message, terminating each by depressing Margin Release with the last character of that message, as described in Section 2 under "How To Type Entered Messages".
A message number is automatically assigned for each message, sequentially, starting with message number 0. (The message numbers are coded octally; this will be discussed under "Rewriting Messages".)
9. The SENSE switch remains up until the last message is to be entered. The SENSE switch is then moved down to indicate that the final message is being entered. The position of the SENSE switch for any message must be in the correct position at the time of Margin Release depression (i.e., termination) for that message.
10. The messages can be typed from the I/O Writer, just as if you were typing up a blank form, except that rules for SENSE switch position *and* message termination (via Margin Release) must be observed. There is also a limit on the total number of characters permitted for messages. (See "Limits on Total Message Length")
- 11a. *If messages are not to be stored on magnetic cards:*
When the last message has been entered, the program continues to the next instruction: HALT. Without depressing , enter your calculation program at an available branch point. The short program of step 5 can be overwritten, since it is no longer required. (But you must not enter your calculation program in locations reserved for the I/O Writer Interface Program.)
If your calculation program was already in program memory, simply Jump to the start of your program. (Or depress the appropriate user-definable "Start" key for the 1830.)
- 11b. *If messages are to be stored on magnetic cards:*
At the Halt instructions, depress for the 1860/1880, or for the 1830. Write the messages onto one or more magnetic cards in the same manner as for data, as described in your Programming Reference Manual. To find the number of registers to be written, use equation 2 under "Registers Used for Messages or Data". Then proceed as in 11a, if you wish to run the program.

Storage of Messages

Messages are stored in main data memory in a sequence of registers, starting at main data register 09 for the 1860/1880, or register 24 for the 1810/1830. Messages were written onto magnetic cards in operation 11b, starting from registers 00 or 15, because message-address information which must be saved is stored in registers 01 through 08 (1860/1880), or registers 16 through 23 (1810/1830). Further, a buffer register, 00 or 15, is required. This is summarized in the following tabulation.

Register Usage	Register Number	
	<u>1810/1830</u>	<u>1860/1880</u>
Buffer register	15	00
Message address pointer registers	16–23	01–08
Message registers (or data)	24 on	09 on

There are limits on the *total length* of all messages which can be stored; this limit is dependent on the size of the calculator's main data memory. There is also a limit on the *total number* of messages which can be stored, regardless of message length. In the present I/O Writer Interface Program configuration, the maximum number of messages (initial, added, or entered during a program) is 31.

A normal character in a message (any character except special characters) is stored as a three-digit code which occupies 1/8th of a main data register.

A message termination command is stored as a three-digit code which occupies 1/8th of a main data register.

A special character (carriage return, index, space, backspace, tab, and underscore) is stored as 2 three-digit codes which occupy 1/4th of a register. This is true for any number of identical special characters in a row, up to a limit of 127. For example, 1 space in a message, or 12 consecutive spaces, will both be stored in 1/4th of a register.

In addition to those registers used for message storage, 8 registers (registers 01 through 08 for the 1860/1880 or registers 16 through 23 for the 1810/1830) are reserved for storage of message-address pointers, as mentioned above. The details of these address pointers will not be given here; but it should be carefully noted that these registers are *not* available for storage of data from your calculation program.

Register 00 (1860/1880) or 15 (1810/1830) is available for data storage, but *only after* all messages have been read into the calculator, for a particular program execution. They are not available for message storage.

Limits on Total Message Length

The following expression gives the limits on the total number of characters permitted for *all* messages:

$$C + T + 2S \leq 8(N - 9 - D)$$

- Where:
- C = total number of normal characters
 - T = number of messages (i.e., number of Margin Release terminations)
 - S = number of special characters. (Any number of identical special characters in a row, up to a limit of 127, is considered as one special character.)
 - N = total number of main data registers in your calculator. (See Appendix B of your Operating Instructions Manual.)
 - D = number of main data registers required for storage in your calculational program. On the 1810/1830, D must always include 15 registers, whether used or not, which represent registers 00 through 14. These registers cannot be used for the storage of messages since they were made available for storing data associated with certain non-keyboard finance functions.

The terms on the left, which represent the sum of stored characters, can be equal to or less than 8 times the number of available registers, as indicated by the expression on the right.

If too many characters are used, your program will stop during a message printout, at the last allowable character. If such a stop occurs during a message printout, you should replan your messages and your format to reduce the total number of stored characters.

Registers Used for Messages or Data

Once you have planned and typed out a test copy of your format (or have planned out your messages for a prepared form), you can determine the following:

1. Total number of registers used for storing messages = $N_R = \left\lceil \frac{C + T + 2S}{8} \right\rceil$,
where C, T, and S are the number of normal characters, message terminations and special characters in the messages, and the bracketed value, if it contains a fraction, must be raised to the next highest whole number. For example: 18-3/8 becomes 19.
2. Total number of registers to be written onto magnetic cards = $N_R + 9$
3. Last register number containing a message = $23 + N_R$ (for 1810/1830)
= $8 + N_R$ (for 1860/1880)

Data may be stored in the register immediately following the last message register, so long as no further messages are to be appended later, and no messages are to be rewritten later that might give a greater message length than the current length.

Message Initialization as Part of Calculation Program

The short sequence of steps given in operation 5 under "Message Initialization" can be programmed as a permanent part of your calculation program. If most or all programmed messages will vary frequently, the messages can be initialized each time the program is run. Care should be taken, however, that the first register used for data lies outside the highest numbered register anticipated for message use.

Message Modification

You may wish to change some of the messages in the set of programmed messages, or you may wish to provide the user with the capability of entering messages, such as the entered messages of the sample printout. (Programming for entered messages will be discussed later.)

The following three-step instructions are used for message modification:

	<u>For 1830</u>	<u>For 1860/1880</u>
Message modification,	202	201
signaled input	122	256
	xxx	xxx
Message modification,	202	201
unsigned input	134	270
	xxx	xxx

As discussed under "Message Initialization", operation 8, messages are automatically numbered as you enter them. The xxx step of the message modification command is the number of the first message to be modified. (The xxx value must be entered octally; this will be discussed in detail later). From message xxx on, messages are modified in sequence. You cannot retain previous messages numbered higher than the modified message or messages; they will be distorted by the modification operation. (But you can write them in again, as part of your modified string of messages as in Sample 2 on page 31.)

The following samples of message modification show how the initial message group can be changed in a number of ways. Under each message number, a single letter represents a full message. For example, message 0 is "a". In each sample, the bracketed letters represent the changed messages.


In sample 1, messages 4 through 8 were modified. In sample 2, messages 4 through 8 were again modified, but some initial messages (h, i) were re-entered during the modification execution, since these were to be retained. In sample 3, the messages were terminated with message 5; the modified group has fewer messages than the initial group. In sample 4, the messages were continued beyond message 8, with v as message 9, and w as message 10; the modified group has more messages than the initial group. It is also possible to start message modification at the *first* message

beyond the initial message set, as shown in sample 5. But you cannot start modification beyond that point.

Message Number:	0	1	2	3	4	5	6	7	8		
Initial Messages:	a	b	c	d	e	f	g	h	i		
Modified Messages:											
Sample 1:	a	b	c	d	[q	r	s	t	u]		
2:	a	b	c	d	[q	r	s	h	i]		
3:	a	b	c	d	[j	k]					
4:	a	b	c	d	[q	r	s	t	u	v	w]
5:	a	b	c	d	e	f	g	h	i	[v	w]

Since the modified messages may contain more characters than the messages they replace, filling more data registers, you must make certain that you aren't encroaching on registers dedicated to data storage.

In the calculation program, messages are retrieved by message number for output by the I/O Writer. You must make sure that messages which have been modified still correspond to the message numbers you wish to access. Also, if you go beyond the last numbered message of the initial group (as in samples 4 and 5), you must either have appropriate instructions for retrieving these messages in your calculation program, or you must rewrite your program to include such instructions. Message output instructions will be discussed under "Message Output".

The message modification instructions can be entered into the calculator and executed in a manner analogous to the initialization instruction, operations 1 through 11. In operation 5, you would enter the appropriate three-step instruction via , followed by a HALT. You can select the instruction for signaling message-input readiness, or not, as desired. The same considerations hold for SENSE (up for more messages, down for last modified message) and Margin Release. Message numbers are automatically assigned, starting with the number of the first message to be modified.

Octal Numbering for Message Numbers

In the message modification instructions discussed previously, the xxx code represents the number of the first message to be modified, in octal notation. In the octal system, numbers run from 0 to 7, 10 to 17, 20 to 27, 30 to 37, and so forth. Digits eight and nine do not exist in the octal system. The following tabulation gives the equivalent octal number to be entered for each message up to the allowable maximum.

Decimal — Octal Equivalents

Decimal	Octal	Decimal	Octal	Decimal	Octal
0	000	11	013	22	026
1	001	12	014	23	027
2	002	13	015	24	030
3	003	14	016	25	031
4	004	15	017	26	032
5	005	16	020	27	033
6	006	17	021	28	034
7	007	18	022	29	035
8	010	19	023	30	036
9	011	20	024		
10	012	21	025		

Message Augmentation

You can add messages to your initial set of messages, or to your modified set. In message augmentation, the added messages start immediately after the last programmed (or entered) message, and the numbering continues from the number of that message.

The following two-step instructions are used for message augmentation:

	<u>For 1830</u>	<u>For 1860/1880</u>
Message augmentation, signaled input	202 167	201 324
Message augmentation, unsignaled input	202 201	201 336

A third step giving the message number isn't required, since message augmentation always starts after the last prior message. The technique for message augmentation is analogous to that given under "Message Initialization", operations 1 through 11, with the appropriate codes used in operation 5. Message input readiness for augmentation may be signaled or not, as desired. SENSE and Margin Release are utilized in the same way as for message initialization.

There is one important difference, however. *Message augmentation should only be performed when a message initialization or message modification has been executed, and following that, the calculator has neither been turned off nor used to execute any other calculation programs.*

Entered Messages

The modification and augmentation instructions can be used separately or together to create entered messages, which can be printed out as often as desired and then replaced.

James Henry and Washington Square, Inc. of the sample printout are examples of entered messages which utilize this technique. These will be discussed in detail under "Demonstration Program", Section 4, but the basic technique follows:

- **Entered Messages via Message Modification**

You can write a modification instruction into your program at an appropriate point, to add messages to your initial set (such as v and w of sample 5 under "Message Modification". The programmed modification instruction would contain a message number one greater than the last message number in the initial set. These modified messages may be typed in by the user. The message is stored; it may be recalled for output by the I/O Writer later in the program execution. When the program is re-executed, a new message (or messages) must be entered by the user at the same point in the program, since message modification is again required. In this way, names, addresses and dates may be entered on a form with new information entered on the next form.

- **Entered Messages via Message Modification and Augmentation**

If the entry of messages by the user must be interrupted for programmed message output, or for data manipulation, a somewhat different technique

can be used. This was the case in the Demonstration Program. The message *Company*: intervened between the entered message *James Henry* and the second entered message *Washington Square, Inc.*

To permit entry of messages which are separated as above, the first message (or group of messages) can be entered as before, using the message modification instruction. But the second message (or message group) and all subsequent messages can be entered using the message augmentation instruction, saving a program step each time. On the next loop through the program, the modification of the first entered message resets the message numbers following that first message, permitting new message augmentation.

For these techniques, the SENSE switch must be up or down, depending on whether the message being entered is the last in the current string. In the Demonstration Program, the SENSE switch had to be down for both entered messages, since each was the last (and only) one in its string.

Message Output

Your calculation program can execute the output of message number xxx on the I/O Writer with the following three-step instruction. (xxx is the message number in octal notation):

For 1810/1830/1860/1880

Message Output:	201
	074
	xxx

Messages can be called for printout in any desired order; and any message may be repeated. Simple messages, such as *tab* are often executed many times during a program.

D NUMERIC INSTRUCTIONS

Numeric Entry from the I/O Writer

Numbers to be used as calculation data (rather than messages) may be entered from the I/O Writer by executing the following instructions. (Readiness for data entry may be signaled or unsignaled:)

	<u>For 1810/1830</u>	<u>For 1860/1880</u>
Numeric Entry from I/O Writer; signaled	203 050	202 204
Numeric Entry from I/O Writer; unsignaled	203 062	202 216

Numeric Entry from the Calculator

Numeric entry from the calculator is programmed with Halts and/or User-Definable Keys (1810/1830), as discussed in your Programming Reference Manual. In fact, all calculator operations can be programmed in the normal manner.

Numeric Output from the I/O Writer; Mixed Decimal Format

To output numbers on the I/O Writer in mixed decimal format, the following 4-step instruction must be programmed:

	<u>For 1810/1830/1860/1880</u>
Output in Mixed Decimal	200
Format:	113
	YYY
	zzz

The value of yyy specifies the length of the field, including places for sign and decimal point. The value must be written octally; the tabulation of decimal/octal equivalents under "Octal Numbering for Message Numbers" can be used to determine the octal equivalent. The maximum permissible field length is 12 places (014 in octal numbering). The value of zzz specifies the number of places after the decimal point. It must also be entered in octal notation. Refer to "Output in Mixed Decimal Format" in section 2 for further details on this format.

Numeric Output from the I/O Writer, According to Calculator D.P.

To output numbers on the I/O Writer according to the current decimal point setting of the calculator, the following 2-step instruction is used:

	<u>For 1810/1830/1860/1880</u>
Output according to D.P.	200
	101

Refer to "Output According to Calculator D.P." in section 2 for further details on this format.

Numeric Output from the I/O Writer; Exponential Format

To output numbers on the I/O Writer in exponential format, the following 2-step instruction is used:

	<u>For 1810/1830</u>	<u>For 1860/1880</u>
Output in Exponential	203	202
Format:	241	375

Refer to "Output in Exponential Format" in section 2 for further details on this format.

Table 2. Summary of Special Instructions

Special Instructions	1810/1830	1860/1880
Messages		
Initialization, signaled input	202	201
	146	302
Modification, signaled input	202	201
	122	256
	xxx	xxx
unsigned input	202	201
	134	270
	xxx	xxx
Augmentation, signaled input	202	201
	167	324
unsigned input	202	201
	201	336
Message Output	201	201
	074	074
	xxx	xxx
Data		
Numeric Entry, signaled	203	202
	050	204
unsigned	203	202
	062	216
Numeric Output, Mixed Decimal Format	200	200
	113	113
	YYY	YYY
	ZZZ	ZZZ
Numeric Output, According to D.P.	200	200
	101	101
Numeric Output, Exponential Format	203	202
	241	375

xxx is the message number in octal notation.

yyy is the total field length, including sign and decimal point, in octal.

zzz is the number of places after the decimal point, in octal.

PROGRAMMING TECHNIQUES

Now that the special instructions used in preparing programs and messages for the I/O Writer have been discussed, the programming of the Demonstration Printout of figure 5 will be considered in detail. This should serve as a model for program and message preparation. But first, the basic steps in preparing a program will be discussed.

Steps in Programming

The broad steps in preparing a program for the I/O Writer are:

1. Type out your desired format, including headings, messages, underscores and other information on a trial sheet.
2. Decide on numeric output formats. For example, for large numbers or small fractions, exponential format might be convenient. For flexibility, output-according-to-D. P. can be used.
3. Decide whether data should be entered from the calculator or the I/O Writer.
4. Consider where tabs should be set for headings, indented matter, columnar numeric printouts, and similar material. This can be of particular importance for prepared forms; most of your messages may be tabs, spaces, carriage returns and line feeds (index). Precise alignments may be critical.
5. Will an entirely new set of tab settings be required at some point in your printout? If so, you can prepare a message to clear the old tab settings via [tab, clear, tab, clear, tab, clear . . .] and another message for the new tab settings. (You cannot clear tabs during a carriage return while under program control as you can in manual operation.)
6. Start dividing your printout into *programmed* messages and *entered* messages. Remember that carriage returns, spaces, space-ups, and the like, are all part of your messages. Will entered messages be different on each printout? Use the techniques discussed under "Entered Messages", Section 3. Will you be adding programmed messages or modifying programmed messages at any time? If so, how much longer will the total message string be? Type out a trial of the anticipated new format. Remember that your total message length is limited, as discussed under "Limits on Total Message Length", Section 3.
7. Can several programmed messages be combined into one? For example, a series of tab settings can be entered as part of another message line. In the Demonstration Printout, tabs for the data inputs and printouts were set with the first line of message 0. (Details of the procedure will be discussed later.) Also, tab clears (as in step 5 above) can be entered as part of another message line.
8. Once you know the number of messages required and the total length of the messages, you can write your calculation program. The first register available for data storage can be found as discussed in Section 3, "Registers Used for Messages or Data". (If in doubt for any reason, leave several registers blank as

Continued

"cushion". This is a recommended practice wherever possible; it will cover small errors in counting total characters.) Messages are recalled and output by placing the appropriate instruction codes (see "Message Output", Section 3) at each point in the program where a message is desired. Message Output instructions can be written into subroutines, if convenient. Also, messages can be called for output printing in any desired order, any number of times.

9. Check your completed program:
 - a) Store the I/O Writer Interface Program in the calculator.
 - b) Initialize your messages, as discussed under "Message Initialization", Section 3.
 - c) Store your calculation program in the calculator and execute the program.
 - d) Adjust or correct your messages, as necessary, and rerun the program. You can use the techniques discussed under "Message Modification", Section 3. If many corrections and changes are required, it may be simpler to redo the entire message initialization.
10. All necessary instructions and limitations should be noted, and given to the user. For example, the user should be informed about margin settings, and data field lengths and formats.

Preparing An I/O Writer Program for the 1810

The program for the 1810 must be prepared on the 1830, as discussed in the 1830 Programming Reference Manual. For messages to be read into the 1810, a Data-Read routine must first be loaded into the 1810 as discussed under "Data-Read Routine for the Model 1810", Section 7, 1830 Programming Reference Manual. The basic steps follow:

- Load the Data-Read Routine. Make certain that the initial "message" register, 15, is written in steps 1 and 2 of the Data-Read Routine. (See "Storage of Messages", Section 3.)
- Enter the messages from magnetic cards. The loading of the Data-Read Routine and loading of messages must be repeated for every set of 32 registers to be used. Remember that the appropriate next (new initial) message-register number must be written into the Data-Read Routine each time. (See Section 7, Programming Reference Manual, as cited above.)
- Enter the I/O Writer Interface Program with the calculation program, in a single Load-operation, as discussed under "I/O Writer Interface Program". (The Data-Read Routine can be written over, once the messages have been entered.)

As an alternative, the message initialization instruction can be read into the 1810 as an adjunct of your calculation program, or independently. Messages can then be initialized using the 1810, once the I/O Writer Interface Program has been entered. (Augmentation and modification can be performed similarly.)

If the initialization instruction was entered independently, the calculation program, with the I/O Interface Program, must be entered into the 1810 afterwards if program execution is desired.

Should you want to write the messages onto a magnetic card from the 1810, the Data-Write Routine must be loaded, as discussed in the 1830 Programming Reference

Manual. The I/O Interface Program will be written over by the Data-Write Routine. Therefore, after writing messages from the 1810 onto magnetic cards, the I/O Writer Interface program must be re-entered with the calculation program, if further execution of the calculation program is desired.

Demonstration Printout Messages

The Demonstration Printout used seven programmed messages (numbered 0 through 6) and two entered messages (messages 7 and 8). Message 0 included the tab settings for the data columns. (These tab settings were also of value in message 2 for speeding up the printout of headings by tabbing and spacing to each heading location, rather than just spacing to each heading location. See message 2 below.)

The messages follow. Spaces are indicated by Sp in locations other than the normal space locations between words and sentences. Carriage returns are indicated by CR. (CRCR results in a carriage return and a space-up.) Tab *settings* are shown by a T *above* the location at which the tab was set. A bar across characters such as SpSpSp indicates that a special character was stored in the calculator for that series of repeated characters.

Message 0:

T T T T

CRCR Monroe Model 300 I/O Writer with an 1800 Series calculator CRCR can perform many functions. It can enter messages. Please CRCR type below: CRCR
Your Name: SpSpSp

Message 1:

CRCR Company: SpSpSpSpSp (The two CR's take the typing-element from the entered name to the C of Company.)

Message 2:

CRCR It can handle numerical entry and output. Now enter a number CRCR on the calculator. Enter another number from the I/O Writer. CRCR
TabSpSpSpSpSpSpSpXTabSpSpSpSpSpSpYTabSpSpSpSpSpSpSpX*Y
TabSpSpSpSpSpSpSpSpX/Y

Message 3:

CRCR Tab (To go from end of a line of data to next line of data.)

Message 4:

Tab (To go to next data column.)

Message 5:

CRCR Tabs for headings and data columns scientific format. CRCR
Thank you, Sp

Message 6:

Sp of Sp (The word *of*, with spaces before and after.)

Message 7:

James Henry (This can be any entered message.)

Message 8:

Washington Square, Inc. (This can be any entered message.)

Demonstration Program

The program for the Demonstration Printout is shown in figure 7, for the 1810/1830 calculators. The only differences between this program and the equivalent program for the 1860/1880, shown in figure 8, is in the step locations and the special instruction codes. (See Table 2.) Therefore, 1860 and 1880 programmers should read this discussion of the program routine, keeping in mind those differences.

The program of figure 7 starts at step 350, defined as symbolic address 0. The instructions at steps 353 to 355 call for the output of programmed message 0. This is followed by a message modification instruction (signaled) for entry of the name (message 7). The SENSE switch must be down for entry of this message to indicate it is the last (only) message in the set. (Required positions of the SENSE switch throughout a program must be made known to the user.) Programmed message 1 [Company] is recalled for printing at steps 359 through 361, followed by a message augmentation instruction (signaled) at steps 362–363, for entering the name of the company (message 8). The SENSE switch must still be down for this last (only) message of *its* set.

The entry of messages 7 and 8 via a message modification instruction and a message augmentation instruction illustrates the second technique described under "Entered Messages" in Section 3.

At step 364, message 2 is recalled for printout. Symbolic address 1 is defined at step 367. Message 3 is executed at steps 369–371; this message takes the I/O Writer typing element from the last data-heading location to the location of the first data printout. At the Halt, (step 372), data is entered from the calculator, stored in scratch pad register 0, and printed on the calculator tape. The instruction to print this data (X-value) from the I/O Writer according to the programmed format is given at steps 376 through 379, where the total field width is 10 (octal 12) and the number of decimal places is three. Message 4 [Tab] is then executed, instructing the typing element to advance to the next tab setting. An instruction for numeric entry (signaled) from the Writer (Y-value) is written at steps 383–384, followed by instructions to print that entry on the calculator tape, store it in scratch pad register 1, and perform a register multiplication with the contents of scratch pad register 0, giving $X*Y$ in the E-register. Message 4 [Tab] is again executed (steps 391–393), advancing the typing-element to the next data printout location, under the $X*Y$ heading. At steps 394–395, the E-register contents are printed from the Writer according to calculator D. P. A tab (message 4) is executed again, to bring the typing-element to the X/Y data position. At steps 399 through 403, the contents of scratch pad register 0 (X) is recalled to the E-register and divided by the contents of register 1 (Y). The value of X/Y is printed by the Writer in exponential format (steps 404–405), completing one line of data.

The position of the SENSE switch is used in the demonstration program to determine whether additional data is to be entered and manipulated, (SENSE up), or the program is to continue on to the next message (SENSE down). If the SENSE switch is up, the program Jumps to symbolic address 1, and the data operations are repeated. If SENSE is down, the program prints out messages 5, 7, 6 and 8, to create the final message sequence. This SENSE switch information must be given to the user.

Depressing Resume at the Halt (step 423) will cause the Jump instruction (steps 424–426) to bring the program back to the start.

If underflow occurred in printing $X*Y$ according to D. P., a Jump to symbol 1 would have been required (entered via the calculator keyboard) for new data entry. At any point where the calculator and I/O Writer are idle, a Jump to symbol 0 may be made to start the program again.

A summation of message characters, total characters, and required registers is presented in the following tabulation. For entered messages 7 and 8, an approximate maximum of 40 (normal) characters was based on the requirement that both messages must fit on the last line, with programmed messages *Thank you* and *of*. Also, three special characters (spaces between words) were assumed for message 7 and three for message 8. In message 0, the four tab settings must be included in the (normal) character count.

<u>Message No.</u>	<u>Characters</u>	<u>Special Characters</u>	<u>Terminations</u>
0	122	25	1
1	8	2	1
2	109	30	1
3	0	2	1
4	0	1	1
5	206	42	1
6	2	2	1
7	{ 40 }	3	1
8	{ 40 }	3	1
Totals	487	110	9

$$\text{Total message registers} = N_R = \frac{487 + 9 + 2 (110)}{8} = 89\frac{1}{2}; \text{ use } 90 \text{ for } N_R.$$

$$\text{Number of registers to be written onto magnetic cards} = N_R + 9 = 99.$$

$$\begin{aligned} \text{Last register number containing a message} &= N_R + 23 = 113 \text{ (1810/1830)} \\ &= N_R + 8 = 98 \text{ (1860/1880)} \end{aligned}$$

DEMONSTRATION PROGRAMS

Figure 7. Demonstration Program, 1810/1830 (1 of 2)


TITLE _____				PROGRAMMER _____				 MONROE	
STEP				SYMBOL	COMMAND	CDMMENTS			
3 5 0				0	IND/SYMB	Symbolic Address 0			
1					0				
2					154	Print Disable			
3					201	} Print Message 0			
4					074				
5					000				
6					202	} Enter Message 7 (Name)			
7					122				
8					007				
9					201	} Print Message 1			
3 6 0					074				
1					001				
2					202	} Enter Message 8 (Company Name)			
3					167				
4					201				
5					074	} Print Message 2			
6					002				
7				1	IND/SYMB				Symbolic Address 1
8					1				
9					201	} Execute Message 3			
3 7 0					074				
1					003				
2					HALT	Numeric Entry from Calculator			
3					↓()				
4					0	X-value in Register 0			
5					060	Print Entry			
6					200	} Output Data according to Programmed Format			
7					113				
8					012				Field Width of 10 (octal 12)
9					003	} 3 Decimal Places			
3 8 0					201				
1					074				} Execute Message 4
2					004				
3					203	Numeric Entry from			
4					050	} Writer Print Entry			
5					060				
6					↓()				
7					1	Y-value in Register 1			
8					◇()	(↑() on 1860/1880)			
9					x	Multiply			

Figure 7. Demonstration Program, 1810/1830 (2 of 2)


TITLE				PROGRAMMER			 MONROE	
STEP				SYMBOL		COMMAND	COMMENTS	
3 9 0						0	Y * X	
1						201	} Execute Message 4	
2						074		
3						004		
4						200	} Output Data according to	
5						101		
6						201		
7						074	} Execute Message 4	
8						004		
9						◇()		
4 0 0						0	X-value	
1						◇()		
2						024	Divide	
3						1	X/Y	
4						203	} Output Data in Exponential	
5						241		
6						JUMP () ()	Jump to Symbol 1 if	
7						x	SENSE is up.	
8						IND/SYMB	Continue if SENSE is	
9						1	down.	
4 1 0						201	} Print Message 5	
1						074		
2						005		
3						201	} Print Message 7	
4						074		
5						007		
6						201	} Print Message 6	
7						074		
8						006		
9						201		
4 2 0						074	} Print Message 8 (octal 10)	
1						010		
2						155	Print Enable	
3						HALT		
4						JUMP () ()	} Jump to Start of Program	
5						IND/SYMB		
6						0		
7								
8								
9								

Figure 8. Demonstration Program, 1860/1880 (1 of 2)



TITLE _____				PROGRAMMER _____		 LITTON	MONROE
STEP	SYMBOL			COMMAND	COMMENTS		
8 4 0	0	IND/SYMB					
1				0			
2				154	Print Disable		
3				201	} Print Message 0		
4				074			
5				000			
6				201	} Enter Message 7 (Name)		
7				256			
8				007			
9				201			
8 5 0				074	} Print Message 1		
1				001			
2				201	} Enter Message 8 (Company Name)		
3				324			
4				201	} Print Message 2		
5				074			
6				002			
7	1	IND/SYMB			Symbolic Address 1		
8				1			
9				201			
8 6 0				074	} Execute Message 3		
1				003			
2				HALT	Numeric Entry from Calculator		
3				↓()			
4				0	X-value in Register 0		
5				PRINT X			
8				200	} Output Data according to Programmed Format		
7				113			
8				012			
9				003	Field Width of 10 (octal 12) 3 Decimal Places		
8 7 0				201	} Execute Message 4		
1				074			
2				004	} Numeric Entry from Writer		
3				202			
4				204			
5				PRINT X			
8				↓()			
7				1	Y-value in Register 1		
8				↑()			
9				x			

Figure 8. Demonstration Program, 1860/1880 (2 of 2)

TITLE _____				PROGRAMMER _____			 Litton	MONROE
STEP				SYMBOL	COMMAND	COMMENTS		
8 8 0					0	Y * X		
1					201	}		
2					074	} Execute Message 4		
3					004	}		
4					200	} Output Data according to		
5					101	} D.P. (X * Y)		
6					201	}		
7					074	} Execute Message 4		
8					004	}		
9					↑()			
8 9 0					0	X-value		
1					↑()			
2					÷			
3					1	X/Y		
4					202	} Output Data in Exponential		
5					375	} Format (X/Y)		
6					JUMP () ()	} Jump to Symbol 1 if SENSE		
7					x	} is up.		
8					IND/SYMB	} Continue if SENSE is		
9					1	} down.		
9 0 0					201	}		
1					074	} Print Message 5		
2					005	}		
3					201	}		
4					074	} Print Message 7		
5					007	}		
6					201	}		
7					074	} Print Message 6		
8					006	}		
9					201	}		
9 1 0					074	} Print Message 8 (octal 10)		
1					010	}		
2					155	Print Enable		
3					HALT			
4					JUMP () ()	}		
5					IND/SYMB	} Jump to Start of Program		
6					0	}		
7								
8								
9								

APPENDIX A 1810/1830 I/O WRITER INTERFACE ROUTINE

APPENDIX B 1860/1880 I/O WRITER INTERFACE ROUTINE

APPENDIX A

1810, 1830 I/O Writer Interface Routine

STEP	CODE	STEP	CODE	STEP	CODE
0000	302	0040	064	0080	334
1	245	1	262	1	372
2	007	2	017	2	340
3	247	3	243	3	340
4	000	4	061	4	340
5	220	5	267	5	160
6	031	6	003	6	201
7	247	7	260	7	065
8	013	8	062	8	040
9	371	9	341	9	364
0010	371	0050	371	0090	040
1	302	1	371	1	365
2	262	2	260	2	340
3	240	3	072	3	340
4	261	4	262	4	340
5	025	5	040	5	340
6	330	6	371	6	262
7	014	7	371	7	017
8	370	8	340	8	340
9	220	9	260	9	020
0020	000	0060	070	0100	131
1	262	1	262	1	040
2	240	2	120	2	257
3	220	3	371	3	215
4	001	4	371	4	345
5	201	5	266	5	345
6	123	6	014	6	345
7	306	7	163	7	261
8	247	8	273	8	166
9	042	9	347	9	040
0030	060	0070	322	0110	352
1	373	1	367	1	341
2	371	2	220	2	344
3	371	3	122	3	350
4	246	4	377	4	256
5	037	5	373	5	156
6	056	6	322	6	220
7	220	7	163	7	174
8	037	8	334	8	345
9	261	9	322	9	352

Continued

1810, 1830 I/O Writer Interface Routine

STEP	CODE	STEP	CODE	STEP	CODE
0120	350	0160	352	0200	326
1	256	1	325	1	345
2	153	2	266	2	262
3	040	3	337	3	020
4	365	4	256	4	261
5	265	5	227	5	364
6	340	6	365	6	266
7	325	7	264	7	004
8	365	8	012	8	321
9	365	9	327	9	151
0130	266	0170	345	0210	051
1	017	1	261	1	322
2	255	2	251	2	262
3	241	3	266	3	060
4	256	4	057	4	261
5	234	5	325	5	354
6	344	6	345	6	266
7	220	7	325	7	052
8	241	8	345	8	020
9	344	9	327	9	273
0140	353	0180	265	0220	345
1	260	1	340	1	322
2	213	2	325	2	272
3	257	3	266	3	076
4	223	4	377	4	271
5	220	5	325	5	310
6	201	6	273	6	031
7	256	7	345	7	060
8	330	8	322	8	321
9	220	9	340	9	060
0150	201	0190	340	0230	334
1	350	1	272	1	344
2	352	2	076	2	260
3	325	3	271	3	343
4	220	4	310	4	220
5	244	5	031	5	316
6	325	6	364	6	266
7	266	7	266	7	140
8	336	8	060	8	314
9	350	9	365	9	201

1810, 1830 I/O Writer Interface Routine

STEP	CODE	STEP	CODE	STEP	CODE
0240	060	0280	365	0320	065
1	051	1	360	1	202
2	221	2	064	2	063
3	111	3	221	3	373
4	262	4	042	4	322
5	020	5	340	5	334
6	365	6	260	6	372
7	243	7	046	7	202
8	377	8	266	8	035
9	321	9	055	9	322
0250	334	0290	321	0330	273
1	220	1	334	1	345
2	307	2	220	2	321
3	377	3	305	3	262
4	377	4	266	4	004
5	241	5	071	5	261
6	003	6	250	6	130
7	220	7	054	7	260
8	371	8	220	8	130
9	340	9	316	9	306
0260	260	0300	266	0340	267
1	035	1	056	1	375
2	266	2	221	2	221
3	040	3	042	3	070
4	321	4	306	4	322
5	334	5	267	5	200
6	266	6	077	6	047
7	001	7	221	7	221
8	064	8	070	8	376
9	365	9	306	9	377
0270	326	0310	263	0510	261
1	340	1	002	1	025
2	260	2	316	2	051
3	030	3	316	3	334
4	064	4	371	4	031
5	342	5	371	5	322
6	064	6	266	6	346
7	345	7	140	7	020
8	221	8	314	8	200
9	016	9	201	9	000

Continued

1810, 1830 I/O Writer Interface Routine

STEP	CODE	STEP	CODE	STEP	CODE
0520	273	0560	051	0600	000
1	345	1	371	1	266
2	322	2	371	2	016
3	315	3	306	3	315
4	360	4	263	4	202
5	060	5	300	5	115
6	250	6	221	6	373
7	004	7	070	7	322
8	051	8	247	8	334
9	334	9	074	9	372
0530	031	0570	371	0610	202
1	221	1	371	1	035
2	111	2	273	2	222
3	200	3	345	3	215
4	000	4	323	4	266
5	273	5	263	5	200
6	345	6	200	6	272
7	322	7	323	7	076
8	315	8	267	8	271
9	222	9	177	9	154
0540	020	0580	371	0620	321
1	347	1	371	1	361
2	263	2	306	2	272
3	200	3	267	3	020
4	361	4	077	4	266
5	272	5	263	5	300
6	020	6	200	6	321
7	032	7	221	7	334
8	322	8	070	8	266
9	273	9	266	9	020
0550	362	0590	141	0630	321
1	321	1	314	1	202
2	052	2	221	2	063
3	334	3	065	3	266
4	322	4	202	4	140
5	334	5	063	5	314
6	032	6	266	6	200
7	273	7	140	7	000
8	363	8	314	8	266
9	321	9	200	9	016

1810, 1830 I/O Writer Interface Routine

STEP	CODE	STEP	CODE	STEP	CODE
0640	315	0680	247	0720	223
1	202	1	361	1	003
2	115	2	200	2	020
3	272	3	000	3	261
4	076	4	305	4	334
5	271	5	202	5	262
6	154	6	070	6	010
7	322	7	273	7	260
8	272	8	345	8	334
9	020	9	250	9	322
0650	361	0690	322	0730	247
1	202	1	322	1	346
2	043	2	342	2	322
3	273	3	321	3	267
4	345	4	247	4	177
5	360	5	272	5	051
6	321	6	222	6	321
7	202	7	252	7	334
8	063	8	040	8	031
9	200	9	262	9	040
0660	000	0700	010	0740	222
1	305	1	260	1	231
2	202	2	320	2	051
3	070	3	261	3	336
4	020	4	320	4	322
5	051	5	322	5	020
6	321	6	346	6	334
7	334	7	346	7	031
8	031	8	267	8	273
9	200	9	177	9	345
0670	047	0710	321	0750	322
1	273	1	260	1	222
2	345	2	252	2	303
3	261	3	261	3	040
4	376	4	252	4	262
5	322	5	051	5	010
6	267	6	336	6	261
7	200	7	031	7	320
8	342	8	222	8	260
9	321	9	223	9	320

Continued

1810, 1830 I/O Writer Interface Routine

STEP	CODE	STEP	CODE	STEP	CODE
0760	360	0800	321	0840	066
1	321	1	300	1	040
2	051	2	244	2	262
3	336	3	046	3	060
4	222	4	221	4	261
5	314	5	123	5	151
6	322	6	222	6	243
7	247	7	215	7	131
8	012	8	202	8	040
9	222	9	106	9	262
0770	223	0810	266	0850	060
1	322	1	140	1	271
2	267	2	314	2	020
3	177	3	200	3	203
4	051	4	000	4	144
5	321	5	266	5	223
6	334	6	016	6	200
7	031	7	315	7	267
8	051	8	202	8	007
9	266	9	115	9	260
0780	004	0820	202	0860	137
1	321	1	106	1	223
2	334	2	360	2	120
3	031	3	024	3	344
4	362	4	273	4	260
5	052	5	345	5	200
6	321	6	321	6	223
7	273	7	200	7	120
8	363	8	000	8	272
9	322	9	305	9	050
0790	052	0830	202	0870	372
1	334	1	070	1	371
2	321	2	020	2	371
3	334	3	262	3	345
4	072	4	010	4	261
5	362	5	261	5	207
6	272	6	111	6	262
7	076	7	260	7	015
8	271	8	111	8	260
9	154	9	223	9	166

1810, 1830 I/O Writer Interface Routine

STEP	CODE	STEP	CODE	STEP	CODE
0880	266	0920	001	0960	334
1	013	1	223	1	266
2	271	2	235	2	300
3	131	3	326	3	364
4	223	4	365	4	364
5	125	5	321	5	203
6	262	6	334	6	233
7	003	7	371	7	364
8	260	8	371	8	344
9	200	9	201	9	367
0890	266	0930	065	0970	250
1	012	1	272	1	304
2	271	2	076	2	364
3	106	3	271	3	203
4	203	4	310	4	223
5	144	5	031	5	266
6	273	6	266	6	105
7	345	7	063	7	321
8	322	8	020	8	334
9	247	9	365	9	360
0900	044	0940	360	0980	326
1	223	1	326	1	241
2	073	2	240	2	333
3	262	3	264	3	203
4	145	4	203	4	223
5	260	5	223	5	223
6	200	6	223	6	337
7	261	7	270	7	266
8	200	8	266	8	055
9	266	9	055	9	321
0910	014	0950	321	0990	334
1	271	1	334	1	266
2	160	2	266	2	360
3	223	3	320	3	203
4	176	4	203	4	233
5	266	5	233	5	365
6	040	6	365	6	345
7	321	7	266	7	203
8	334	8	056	8	233
9	266	9	321	9	266
				1000	004
				1	321
				2	220
				3	354

APPENDIX B

1860, 1880 I/O Writer Interface Routine

STEP	CODE	STEP	CODE	STEP	CODE
0000	302	0040	064	0080	334
1	245	1	262	1	372
2	007	2	017	2	340
3	247	3	243	3	340
4	000	4	061	4	340
5	220	5	267	5	160
6	031	6	003	6	201
7	247	7	260	7	065
8	013	8	062	8	040
9	371	9	341	9	364
0010	371	0050	371	0090	040
1	302	1	371	1	365
2	262	2	260	2	340
3	240	3	072	3	340
4	261	4	262	4	340
5	025	5	040	5	340
6	330	6	371	6	262
7	014	7	371	7	017
8	370	8	340	8	340
9	220	9	260	9	020
0020	000	0060	070	0100	131
1	262	1	262	1	040
2	240	2	120	2	257
3	220	3	371	3	215
4	001	4	371	4	345
5	201	5	266	5	345
6	123	6	014	6	345
7	306	7	163	7	261
8	247	8	201	8	166
9	042	9	065	9	040
0030	060	0070	273	0110	352
1	373	1	347	1	341
2	371	2	322	2	344
3	371	3	220	3	350
4	246	4	143	4	256
5	037	5	373	5	156
6	056	6	322	6	220
7	220	7	163	7	174
8	037	8	334	8	345
9	261	9	322	9	352

Continued

1860, 1880 I/O Writer Interface Routine

STEP	CODE	STEP	CODE	STEP	CODE
0120	350	0160	352	0200	326
1	256	1	325	1	345
2	153	2	266	2	262
3	040	3	337	3	020
4	365	4	256	4	261
5	265	5	227	5	364
6	340	6	365	6	266
7	325	7	264	7	004
8	365	8	012	8	321
9	365	9	327	9	151
0130	266	0170	345	0210	051
1	017	1	261	1	322
2	255	2	251	2	262
3	241	3	266	3	060
4	256	4	057	4	261
5	234	5	325	5	354
6	344	6	345	6	266
7	220	7	325	7	052
8	241	8	345	8	020
9	344	9	327	9	273
0140	353	0180	265	0220	345
1	260	1	340	1	322
2	213	2	325	2	272
3	257	3	266	3	076
4	223	4	377	4	271
5	220	5	325	5	310
6	201	6	273	6	031
7	256	7	345	7	060
8	330	8	322	8	321
9	220	9	340	9	060
0150	201	0190	340	0230	334
1	350	1	272	1	344
2	352	2	076	2	260
3	325	3	271	3	343
4	220	4	310	4	220
5	244	5	031	5	316
6	325	6	364	6	266
7	266	7	266	7	140
8	336	8	060	8	314
9	350	9	365	9	201

1860, 1880 I/O Writer Interface Routine

STEP	CODE	STEP	CODE	STEP	CODE
0240	060	0280	365	0320	065
1	051	1	360	1	201
2	221	2	064	2	217
3	111	3	221	3	373
4	262	4	042	4	322
5	020	5	340	5	334
6	365	6	260	6	372
7	243	7	046	7	201
8	377	8	266	8	172
9	321	9	055	9	322
0250	334	0290	321	0330	273
1	220	1	334	1	345
2	307	2	220	2	321
3	377	3	305	3	262
4	377	4	266	4	004
5	241	5	071	5	261
6	003	6	250	6	130
7	220	7	054	7	260
8	371	8	220	8	130
9	340	9	316	9	306
0260	260	0300	266	0340	267
1	035	1	056	1	375
2	266	2	221	2	221
3	040	3	042	3	070
4	321	4	306	4	322
5	334	5	267	5	200
6	266	6	077	6	047
7	001	7	221	7	261
8	064	8	070	8	162
9	365	9	306	9	051
0270	326	0310	263	0350	334
1	340	1	002	1	031
2	260	2	316	2	322
3	030	3	316	3	346
4	064	4	371	4	020
5	342	5	371	5	200
6	064	6	266	6	000
7	345	7	140	7	273
8	221	8	314	8	345
9	016	9	201	9	322

Continued

1860, 1880 I/O Writer Interface Routine

STEP	CODE	STEP	CODE	STEP	CODE
0360	315	0400	263	0440	201
1	360	1	300	1	251
2	060	2	221	2	373
3	250	3	070	3	322
4	141	4	247	4	334
5	051	5	230	5	372
6	334	6	371	6	201
7	031	7	371	7	172
8	221	8	273	8	221
9	111	9	345	9	352
0370	200	0410	323	0450	266
1	000	1	263	1	010
2	273	2	200	2	272
3	345	3	323	3	076
4	322	4	267	4	271
5	315	5	177	5	154
6	221	6	371	6	321
7	155	7	371	7	266
8	347	8	306	8	110
9	361	9	267	9	272
0380	272	0420	077	0460	020
1	020	1	263	1	271
2	335	2	200	2	010
3	032	3	221	3	321
4	322	4	070	4	334
5	273	5	266	5	266
6	362	6	141	6	020
7	321	7	314	7	321
8	052	8	221	8	201
9	334	9	065	9	217
0390	322	0430	201	0470	266
1	334	1	217	1	140
2	032	2	266	2	314
3	273	3	140	3	200
4	363	4	314	4	000
5	321	5	200	5	266
6	051	6	000	6	016
7	371	7	266	7	315
8	371	8	016	8	201
9	306	9	315	9	251

1860, 1880 I/O Writer Interface Routine

STEP	CODE	STEP	CODE	STEP	CODE
0480	272	0520	200	0560	070
1	076	1	000	1	262
2	271	2	305	2	010
3	154	3	201	3	260
4	322	4	224	4	070
5	272	5	273	5	322
6	020	6	345	6	247
7	361	7	250	7	102
8	201	8	056	8	322
9	177	9	322	9	267
0490	273	0530	342	0570	177
1	345	1	321	1	051
2	360	2	247	2	321
3	321	3	030	3	334
4	201	4	222	4	031
5	217	5	010	5	040
6	200	6	040	6	221
7	000	7	262	7	366
8	305	8	010	8	051
9	201	9	260	9	336
0500	224	0540	137	0580	322
1	020	1	261	1	020
2	051	2	137	2	334
3	321	3	322	3	031
4	334	4	346	4	273
5	031	5	346	5	345
6	200	6	267	6	322
7	047	7	177	7	222
8	377	8	321	8	041
9	273	9	260	9	040
0510	345	0550	010	0590	262
1	261	1	261	1	010
2	132	2	010	2	261
3	322	3	051	3	137
4	267	4	336	4	260
5	200	5	031	5	137
6	342	6	221	6	360
7	321	7	360	7	321
8	247	8	020	8	051
9	115	9	261	9	336

Continued

1860, 1880 I/O Writer Interface Routine

STEP	CODE	STEP	CODE	STEP	CODE
0600	222	0640	221	0680	261
1	052	1	123	1	305
2	322	2	221	2	243
3	247	3	352	3	265
4	146	4	201	4	040
5	221	5	242	5	262
6	360	6	266	6	060
7	322	7	140	7	271
8	267	8	314	8	020
9	177	9	200	9	202
0610	051	0650	000	0690	300
1	321	1	266	1	222
2	334	2	016	2	334
3	031	3	315	3	267
4	051	4	201	4	007
5	266	5	251	5	260
6	004	6	201	6	273
7	321	7	242	7	222
8	334	8	360	8	254
9	031	9	024	9	344
0620	362	0660	273	0700	260
1	052	1	345	1	334
2	321	2	321	2	222
3	273	3	200	3	254
4	363	4	000	4	272
5	322	5	305	5	050
6	052	6	201	6	372
7	334	7	224	7	371
8	321	8	020	8	371
9	334	9	262	9	345
0630	072	0670	010	0710	261
1	362	1	261	1	343
2	272	2	245	2	262
3	076	3	260	3	015
4	271	4	245	4	260
5	154	5	222	5	322
6	321	6	222	6	266
7	300	7	040	7	013
8	244	8	262	8	271
9	202	9	060	9	131

1860, 1880 I/O Writer Interface Routine

STEP	CODE	STEP	CODE	STEP	CODE
0720	222	0760	365	0800	364
1	261	1	321	1	202
2	262	2	334	2	367
3	003	3	371	3	364
4	260	4	371	4	344
5	334	5	201	5	367
6	266	6	065	6	250
7	012	7	272	7	040
8	271	8	076	8	364
9	106	9	271	9	202
0730	202	0770	310	0810	357
1	300	1	031	1	266
2	273	2	266	2	105
3	345	3	063	3	321
4	322	4	020	4	334
5	247	5	365	5	360
6	200	6	360	6	326
7	222	7	326	7	241
8	227	8	240	8	067
9	262	9	020	9	202
0740	145	0780	202	0820	357
1	260	1	357	1	223
2	334	2	223	2	073
3	261	3	024	3	266
4	334	4	266	4	055
5	266	5	055	5	321
6	014	6	321	6	334
7	271	7	334	7	266
8	160	8	266	8	360
9	222	9	320	9	202
0750	261	0790	202	0830	367
1	266	1	367	1	365
2	040	2	365	2	345
3	321	3	266	3	202
4	334	4	056	4	367
5	266	5	321	5	266
6	001	6	334	6	004
7	222	7	266	7	321
8	371	8	300	8	220
9	326	9	364	9	354

instruction
book

Model **300** ^{I/O}
WRITER



MONROE

Monroe, The Calculator Company • Orange, New Jersey, U.S.A. • Toronto, Canada • Zurich, Switzerland

Sales and Service Throughout the World