ARTISAN Electronics

MICROCALCULATOR MODEL 85 HANDBOOK

	SHIFT DSP 9D	STO INV A1	RCL HYP A5	CLR CLX 8B
$\overline{X}$,S Σ+ 95	LOG Ln 9A	→POL SIN 9E	→RAD COS A2	RAD TAN A6
$\sqrt{X}$ Y^X 84	% X⇄Y 8C	Δ% ↓ 90	X! CHS 94	1/X EE 88
C→F ÷ 91	7 93	8 98		9 A4
LTR→GAL X 8D	4 97	5 9C		6 A0
CM→IN − 89	1 8F	2 9F		3 A3
KG→LB + 85	LST X 0 87	π · 9B		SCI ENT A7

PUBLICATION NUMBER MC-85-01

MICROCALCULATOR MODEL-85

HANDBOOK

By: J. W. PIECZYNSKI

FIRST EDITION

ARTISAN ELECTRONICS CORPORATION
5 EASTMANS ROAD
PARSIPPANY, NEW JERSEY 07054

INTRODUCTION

The Microcalculator Model 85 is an advanced scientific calculator without a keyboard. The keyboard has been replaced with control and interface circuitry to permit operation from an 8 bit microprocessor system.

Instead of pressing a key to enter a number or operation, the microprocessor delivers an 8 bit coded instruction which represents the desired key input. The Microcalculator performs the operation and displays the answer on the fourteen (14) digit display. At this time, the microprocessor may continue to deliver instructions to the Microcalculator, or the displayed answer may be read back into the microprocessor for display on a TTY or CRT terminal.

By using the microprocessor's RAM memory and storing complex calculator routines on tape, a programmable scientific calculator system may be developed with instruction steps limited only to the amount of RAM within the microprocessor system.

Nine general purpose storage registers within the Microcalculator permit the storage and recall of intermediate results or answers. The Microcalculator Model 85 operation is based on the Reverse Polish Notation (RPN) method, employing a four level stack for operation.

WIRING THE MICROCALCULATOR

All power, Input and Output connections are made thru J1, a 14 pin IC socket. Refer to figure 1 for description of signals on J1. Refer to Appendix E to determine the relationship of the ribbon cable to the IC pin numbers. Plus 5 volts is connected to pins 2 & 13 with the 5 volt ground connected to pins 1 & 14.

Eight pins are connected to the microprocessor's peripheral I/O port. This I/O port must be capable of having individual pins configured as either Input or Output, such as the Motorola 6820, Mostek 3820, MOS Technology 6530 or equivalent. Pins 3 & 4 are not used. See Appendix D to determine the location of pin 1 prior to inserting the IC plug.

Apply power to the system. The Microcalculator display should indicate 0.00, if the power connections are correct. If the display does not indicate properly, recheck the wiring of the +5V DC and ground.

The 14 digit display on the Microcalculator may be removed and mounted in a location remote to the PC board. If the display is to be mounted in a remote location, observe the wiring between the display and the Microcalculator board.

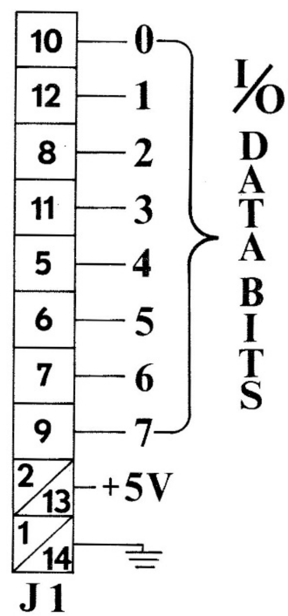


FIGURE 1

HOW THE MICROCALCULATOR WORKS

The Microcalculator Model 85 is an 8 bit peripheral device to the microprocessor system. Instructions to and answers back from the Microcalculator are over the 8 bit I/O bus. This bus is labeled bit 0 thru bit 7, with bit 0 being the least significant bit, and bit 7 the most significant bit.

Figure 2 represents the 8 bits when the Microcalculator is receiving microprocessor instructions. Figure 3 represents the 8 bits when the microprocessor is reading the 14 digit display from the Microcalculator. Instructions from the microprocessor to the Microcalculator is referred to as, the Write Instruction Mode.When the microprocessor is reading the 14 digit display, this is referred to as, the Read Answer Mode.

Notice that in both the Read and Write modes bit 7 is configured as output from the microprocessor. It is bit 7 that determines the state of the tri-state input buffers that are located within the Microcalculator. When bit 7 is set to a 1, the Microcalculator will accept instructions from the microprocessor; and send the 14 digit display back to the microprocessor, when bit 7 is set to a 0.

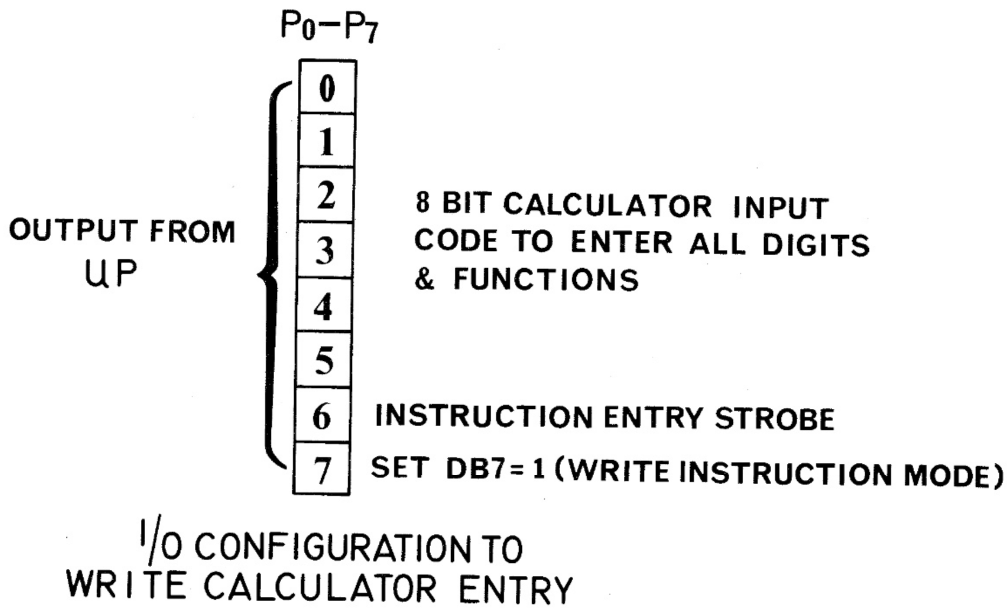


FIGURE 2

WRITE INSTRUCTION MODE

The operation of a standard scientific calculator calls for all entries to be made by key switches by the operator. To enter a number an operator would merely press the desired key, lift his finger and press the next key. With the Microcalculator there are no keys to press. To enter the required number, or function, the microprocessor will send an 8 bit coded instruction to the Microcalculator. Before another instruction may be sent by the microprocessor, another coded instruction must be sent which simulates the lifting of the finger from the keyboard. This is necessary, in order for the Microcalculator to distinguish between input instructions.

During any Write instruction to the Microcalculator, bit 7 must be set to a 1, this will insure that the instruction will be received by the Microcalculator. The instruction must be delivered to the Microcalculator for at least 40 milliseconds. After 40 milliseconds the instruction C0 is issued by the microprocessor. This is the instruction which simulates the lifting of the finger from the calculator keyboard.

The keyboard of our scientific calculator has been replaced with a set of instructions which simulates the desired key entry. Figure 3 lists the instruction set for all the key inputs of the Microcalculator. Notice that instruction 85 is listed as the Add instruction (+), and also the KG→LB conversion instruction. Most scientific calculators make use of a SHIFT key to permit multiple use of the same key. If you wanted to enter the + instruction to the Microcalculator, the instruction 85 would be all that was required. Should you want to enter the conversion instruction KG→LB, the SHIFT instruction would have to be entered first. That is, the SHIFT (9D) instruction would precede the (85) instruction.

This technique of using the same key for dual instructions is standard on most scientific calculators. All instructions in this manual are given in hexadecimal notation. The instruction "+" is (85), which in 8 bit machine language, is equal to 10000101.

At this time a functional test of the Microcalculator accepting instructions from the microprocessor should be performed.

Set up the I/O port servicing the Microcalculator for 8 bit output. The following program enters the number 45, takes the sine, and displays the results in scientific notation, with the sign of the answer changed to minus (-).

EQUIVALENT KEY OPERATION	MICROPROCESSOR INSTRUCTION	DISPLAY READING
Enter 4	(97)	b b b b b b b b b b b b 4 b
Release Key	(C0)	b b b b b b b b b b b b 4 b
Enter 5	(9C)	b b b b b b b b b b b 4 5 b
Release Key	(C0)	b b b b b b b b b b b 4 5 b
Enter SIN (take the sine of 45)	(9E)	b b b b b b b b b b 0.7 1 b
Release Key	(C0)	b b b b b b b b b b 0.7 1 b
Enter SHIFT (set up shift mode)	(9D)	b b b b b b b b b b 0.7 1 b
Release Key	(C0)	b b b b b b b b b b 0.7 1 b
Enter SCI (display scientific notation)	(A7)	b 7.0 7 1 0 6 7 8 1 1 - 0 1
Release Key	(C0)	b 7.0 7 1 0 6 7 8 1 1 - 0 1
Enter CHS (change sign of answer)	(94)	- 7.0 7 1 0 6 7 8 1 1 - 0 1
Release Key	(C0)	- 7.0 7 1 0 6 7 8 1 1 - 0 1

You will note that each instruction entered was followed by code (C0); this is the instruction that simulates the release of the input key, and permits the Microcalculator to accept another input instruction. In an actual program this code would be placed in a loop and would not take up more than one byte of memory

MICROCALCULATOR INSTRUCTION SET

CALCULATOR OPERATOR OR OPERAND		µ-P HEX CODE
NORMAL KEY	SHIFT KEY	
Ø	LST X	87
1		8F
2		9F
3		A3
4		97
5		9C
6		AØ
7		93
8		98
9		A4
+	KG→LB	85
−	CM→IN	89
X	LTR→GAL	8D
÷	C→F	91
•	π	9B
ENT	SCI	A7
Y^X	$\sqrt{X}$	84
X⇄Y	%	8C
↓	Δ%	9Ø
CHS	X!	94
EE	$1/x$	88
Σ+	$\bar{X}$,S	95
L_N	LOG	9A
SIN	→POL	9E
COS	→RAD	A2
TAN	RAD	A6
INV	STO	A1
HYP	RCL	A5
CLX	CLR	8B
DSP	SHIFT	9D

FIGURE 3

The Microcalculator instruction set is shown in figure 3. As with most scientific calculators certain keys perform two functions. In the Microcalculator certain instructions perform two operations. Note that there are two columns which list the key functions (instructions). One column is titled "NORMAL KEY", and the other is titled "SHIFT KEY". Should you want to use an instruction listed under the NORMAL KEY heading, merely write the instruction code listed under the column titled "u - P HEX CODE". If the instruction that you want to use falls under the SHIFT KEY column, the u -P HEX CODE for that instruction must be preceded by the SHIFT instruction code (9D). Using the SHIFT (9D) instruction is required before all instructions listed under the SHIFT KEY column except the LST X, STO, and RCL instructions. Should you precede these three instructions with the SHIFT instruction the Microcalculator would ignore the SHIFT instruction. From figure 7 you will find that the input instructions for 65.3 SIN require no SHIFT instruction; but the SCI instruction to change the display to scientific notation is preceded by the SHIFT instruction code (9D). The coded instruction listed under the u - P HEX CODE column represents the eight (8) bit code to the Microcalculator from the microprocessor.

For example:

The "+" key = Instruction Code 85
85 in hexadecimal code = 10000101
in machine language.

READ ANSWER MODE

A thorough understanding of the operation of the Microcalculator is required to read the displayed answer back into the microprocessor. The Read mode is required at two seperate times within an active program. The first is testing the busy status of the Microcalculator; the second is reading the fourteen (14) digit display back into the microprocessor.

Figure 4 shows the configuration of the I/O port when the microprocessor is set to read the Microcalculator display. Bit 7 is set for output operation and is equal to a 0, which sets up the Read mode. All remaining bits (0-6) are input data to the microprocessor from the Microcalculator. Bits 0 - 5 represent the display characters including the digits 0 - 9, blanks, the minus (-) sign, and the decimal point. This code is represented in figure 5.

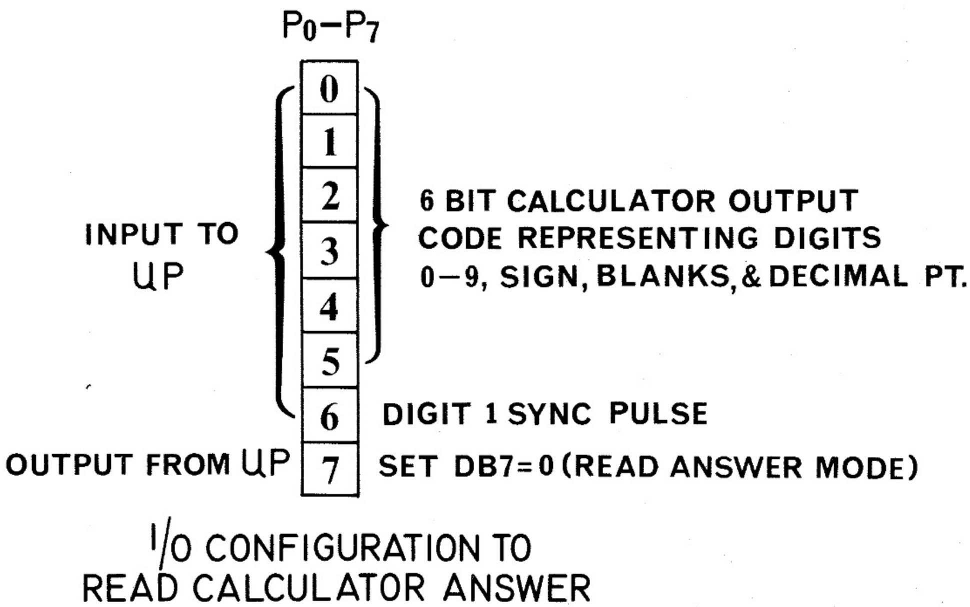


FIGURE 4

The Microcalculator has the capacity to display up to fourteen (14) characters made up of digits, blanks, and the minus (-) sign. These characters are time-division multiplexed over the bits 0 - 5. In order to determine which character position is present at the input to the microprocessor, requires the use of a sync pulse and knowledge of the time-division multiplex relationship. Figure 6 shows the relationship between the sync pulse (Digit 1), and the 14 digits which make up the 14 digit Microcalculator display.

From figure 4 we find that bit 6 is labeled DIGIT 1 SYNC PULSE. This means that each time digit 1 is being read by the microprocessor, bit 6 will be equal to a 1. If we had another bit which we could use for digit 2 sync pulse it would be relatively simple to read digit 2, and so on. However, we only have the digit 1 sync pulse to work with. Therefore, in order to know when digits 2 - 14 are being sent requires the need to simulate the respective digit sync pulses thru software timing.

OUTPUT CODE FOR DISPLAY DIGITS 1 – 14						
1-14	B_5	B_4	B_3	B_2	B_1	$B_Ø$
0	X	0	0	1	1	1
1	X	0	0	0	1	0
2	X	0	1	1	0	1
3	X	0	1	1	1	1
4	X	1	0	0	1	0
5	X	1	0	1	0	1
6	X	1	1	1	0	1
7	X	0	0	0	1	1
8	X	1	1	1	1	1
9	X	1	0	1	1	1
—	X	0	1	0	0	0
BLANK	X	0	0	0	0	0
DEC. PT.	1	WITH ANY VALID DIGIT 0 THRU 9				

FIGURE 5

OUTPUT CODE FOR DISPLAY DIGITS 1-14

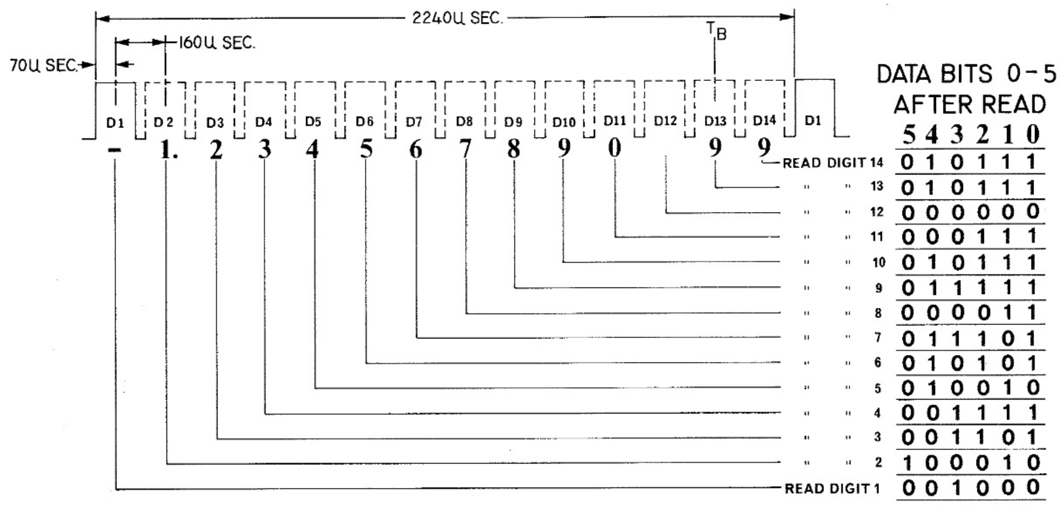
The Microcalculator displays up to 14 characters which are numbered 1 - 14 from left to right. Each time the microprocessor reads the Microcalculator, bits 0 - 5 contain the code which determines whether the character is a digit, a blank, or a minus (-) sign. Also the code determines where in the string of 14 characters the decimal point appears. Figure 5 shows the corresponding character to bits 0 - 5, after they are read into the microprocessor. The table does not convert the code directly to the ASCII equivalent, but will be used as an index to a table which will.

As an example let us set bits 4 - 0 to 10111 and determine what character is being read. From figure 5 it can be seen that when this code exists the character is a "9.".

If bit 5 for the same read were a 0, there would be no decimal point following the 9.

Each time bit 5 is a 1, the character read in will be followed by a decimal point.

Figure 7 lists the 8 bits read in for a given example, and should be studied to see the relationship between the bit pattern read and the corresponding character it represents.



T_B = DIGIT 13 SYNC TIME = 1990 MICROSECONDS

FIGURE 6

Figure 6 illustrates the time-division multiplex relationship with respect to the sync pulse and all fourteen (14) digits.

As discussed earlier, the only time we need to read the Microcalculator is:to determine when it is in a busy mode, and reading the final answer back into the microprocessor for terminal display.

To determine the busy status of the Microcalculator, digit 13 is tested. Digit 13 was selected because it is the one digit that always has a character in it besides a blank. To read digit 13:the microprocessor program waits until the sync pulse (bit 6) goes to a 1, and then creates a 1990 microsecond delay. At the end of the 1990 microsecond delay period, a read instruction will read in digit 13. The actual determination of busy status is covered in the explanation on the program flow charts.

To read in the entire 14 digit display the microprocessor must wait until the sync pulse arrives and then begin a 70 microsecond delay; at the end of which digit 1 is read in. The 70 microsecond delay is created to put the read command in the center of the digit 1 time frame. From this point a 160 microsecond delay will be initiated. At the end of the 160 microsecond delay, digit 2 will be read in. Twelve more 160 microsecond delay/read loops will have then read in all 14 of the Microcalculator's digits. The Microcalculator is shipped with the 14 digit cycle time set for 2240 microseconds. A potentiometer (R1) is provided to permit adjustment should the program require timing adjustment after operation.

Figure 6 illustrates the READ of the Microcalculator displaying -1.234567890 99, and the corresponding values of bits 0 - 5. To determine your understanding of the coded bits 0 - 5, and their actual display value, compare the data bits 0 - 5 from figure 6 with the chart of the output code for display digits 1 - 14 found in figure 5.

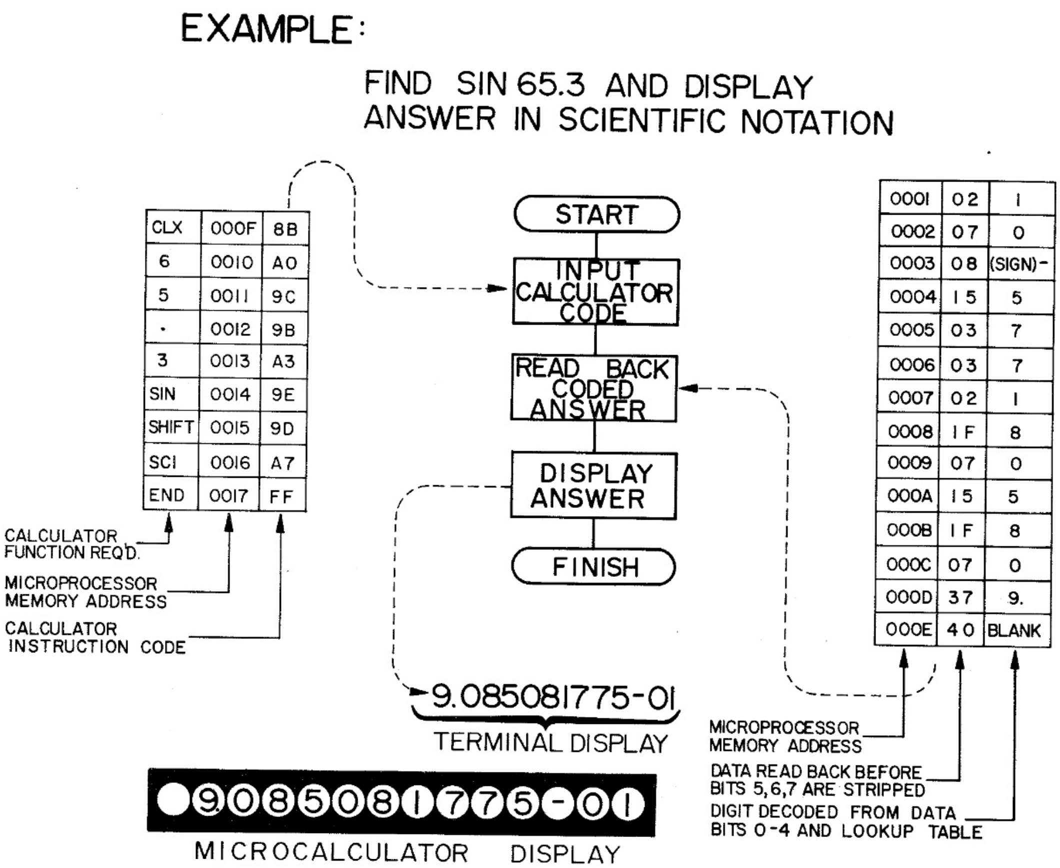


FIGURE 7

Figure 7 illustrates the input code to the Microcalculator from the microprocessor, and the output code from the Microcalculator to the microprocessor for a given problem.

I/O CODE FOR SYSTEM EXAMPLE (Figure 7 Explained)

Input code to the Microcalculator begins with a clear instruction (CLX), which from figure 4 is listed under the column headed NORMAL KEY, and is coded 8B. The Release key instruction C0 is not shown in this example, but is included in the program as part of the input routine. You will note that the instruction SCI, which displays the answer in scientific notation, is listed under the SHIFT KEY column and must therefore be preceded by the SHIFT instruction code 9D. The only exception to this is the LST X, STO, and the RCL instructions. You will also note that the input program ends with an END instruction, code FF. The END instruction (FF) is not part of the Microcalculator's instruction set, but is used to signal the microprocessor that all instructions to the Microcalculator have been entered.

The fourteen (14) digit display, which is read back into the microprocessor, resides in memory locations 0001 - 000E. Digit 1 is the first digit that will be read, and will be placed in memory at address 000E. The last digit read is digit 14, and will reside in memory at address 0001.

After the problem has been written to the Microcalculator, the display will be read back by the microprocessor. The first digit read is digit 1, and is read back as code 40. This breaks down as follows:

		b7	b6	b5	b4	b3	b2	b1	b0
DIGIT 1 = 40 (Hex)	=	0	1	0	0	0	0	0	0

Bit 7 = 0 — Bit 7 is the bit which is always configured as output, and since the microprocessor is READING the answer, bit 7 was set to a 0. A read of the I/O port will always read back the state of the bit as set in the output routine.

Bit 6 = 1 — Bit 6 is the DIGIT 1 SYNC PULSE, and since we are reading digit 1, bit 6 will be equal to a 1. In the remaining thirteen digits bit 6 will always be a 0.

Bit 5 = 0 — Bit 5 is reserved for decimal point location. From figure 5 we find that when bit 5 is a 1 a decimal point follows the digit read. Since bit 5 is equal to 0 there is no decimal point following the digit.

Bits 4 - 0 = 00000
From figure 5 this indicates a blank is being read for digit 1.

NOTE: Digit 2 (memory location 000D) has bit 5 set to a 1. This means that the digit decoded from bits 0-4 will have a decimal point following it. Digit 2 is equal to "9.".

MICROCALCULATOR PROGRAM FLOWCHART

A complete program to operate the Microcalculator may be divided into four parts:

1. CALCULATOR INPUT (Input of calculator instructions)
2. CALCULATOR BUSY ROUTINE (Determines busy status of Microcalculator)
3. READ DISPLAY (Reads the 14 digit display back into the microprocessor)
4. PRINT ANSWER (Prints the answer on a TTY or CRT terminal)

Should the microcalculator display be the only source required for the displayed answer, it will not be necessary to read back the 14 digit Microcalculator display, and parts 3 & 4 would not be required.

Each of the four parts shown in flowchart have a number in a circle located to the left of each step. This number refers to the corresponding numbered paragraphs of their respective sections. The address above each step refers to the program listed in this handbook.

CALCULATOR INPUT (Figure 8)

1. Microcalculator instructions are placed sequentially in memory within the microprocessor system. They may be as short as a single instruction, or as long as the microprocessor system may provide. Before the Microcalculator begins to accept these instructions an index pointer must be initialized to zero. The microprocessor memory locations which hold the Microcalculator instructions will be referred to as the Calculator Instruction Table (CITAB).

2. Input instructions to the Microcalculator require that the microprocessor I/O port be put into an output mode. This step sets up all 8 bits of the I/O port as outputs. Always remember that bit 7 will always be configured as output for all routines. When bit 7 is set to a 1, the Microcalculator will be set to receive instructions from the microprocessor. When bit 7 is set to a 0, the Microcalculator will be sending the fourteen digit display back to the microprocessor.

3. This step loads the Accumulator with the Calculator Instruction Table indexed by the pointer initialized in step 1 above.

4. After the Accumulator has been loaded with the Microcalculator instruction, that instruction is then tested to determine if it is last instruction. The last instruction is called the terminator instruction, and serves the program notice that all of the Microcalculator instructions have been written. At this time the program may terminate, and the display on the Microcalculator provide the answer on the fourteen digit display; or the display may be read back into the microprocessor for terminal display, or stored for use at a later time. The Microcalculator has nine (9) registers which may also be used to store the display.

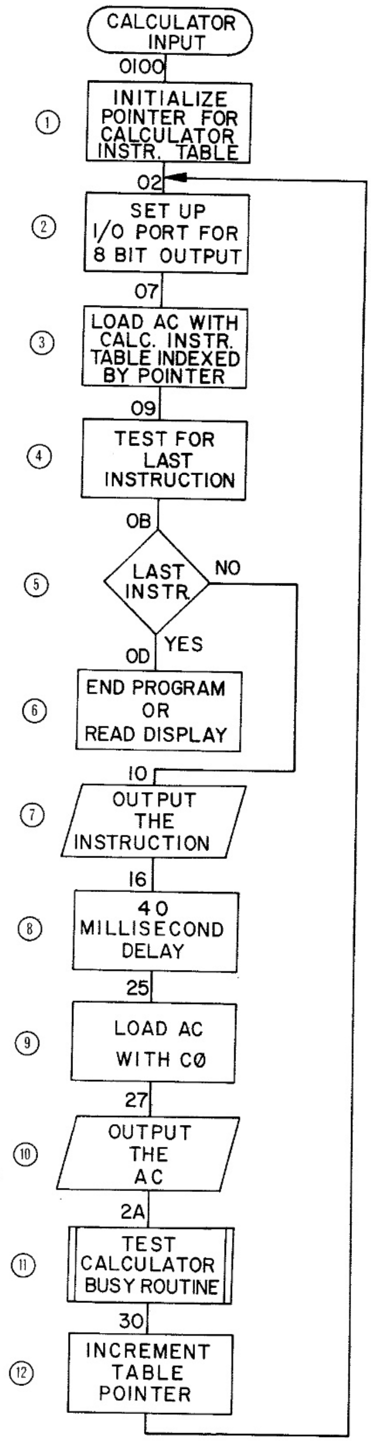


FIGURE 8

5. If the terminator (code FF) is found, the program will terminate, or go into the READ DISPLAY routine in which the Microcalculator's fourteen digit display is read back into the microprocessor. If the terminator is not found, a branch will occur which will output the instruction to the Microcalculator.

6. This part of the program is entered after the last Microcalculator instruction has been entered. As noted above, the program may then terminate, or read back the 14 digit display.

7. At this time the Microcalculator instruction residing in the Accumulator is written to the output port for interpretation by the Microcalculator.

8. Once the Microcalculator is given the instruction by the microprocessor, a delay of forty (40) milliseconds is required, in order that the Microcalculator circuits have time to examine, and accept the instruction.

9. After the forty millisecond delay is over, an instruction will be given which will act as a spacer between instructions. This simulates the lifting of the finger from the keyboard of the calculator. This is accomplished by sending the Microcalculator the instruction code "C0".

10. The instruction C0 is placed in the output port for the Microcalculator to read.

11. At this time, the Microcalculator has been given an instruction; accepted that instruction, and is in the process of performing that instruction. A routine must now be entered to determine the busy status. We cannot enter another instruction until the present one has finished execution.

12. After returning from the CALCULATOR BUSY ROUTINE, the Calculator Instruction Table pointer must be incremented to point to the next instruction. After the pointer has been incremented, the program branches back to step 2.

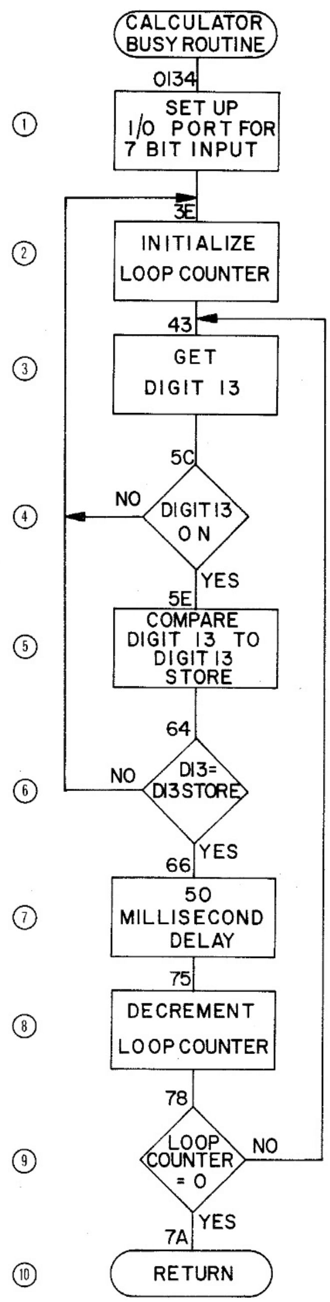


FIGURE 9

CALCULATOR BUSY ROUTINE

(Figure 9)

Various instructions to the Microcalculator may require different execution times. This routine tests the Microcalculator to determine the busy status, and return to the main program when the execution of a given instruction has been completed. The routine listed in this handbook will not return if the current instruction given the Microcalculator caused an overflow, underflow, or was an illegal instruction. If this occurs, the display will flash all nines, or zeroes. Should the user desire to detect an error condition, a five second timer might be set prior to calling the CALCULATOR BUSY ROUTINE. When an error condition did, occur the timer would run out and alert the system. Five seconds is longer than the longest execution time of any instruction in the Microcalculator's instruction set. Remember to reset the five second timer each time the busy routine returns.

1. The I/O port is configured to an input mode for bits 0 - 6, and an output mode for bit 7. Bit 7, as you will remember, is always set for output mode. When bit 7 is set equal to 1, the Microcalculator accepts instructions from the microprocessor; when set to a 0, the microprocessor reads the display of the Microcalculator.

2. The display turns on and off up to three times for various instructions, and changes the display many times during a calculation before the final answer is arrived at. An algorithm was selected that requires the display to be constant (same digit for digit 13) for six consecutive loops after the display turns back on. Each time the display turns off, the loop counter will be reset to six. This step initializes the loop counter.

3. Digit 13 is the only digit that must be on after any given calculation, or entry. This is the reason digit 13 has been selected for test of busy status. This step brings digit 13 into the Accumulator.
   Figure 10 illustrates step 3 in detail.

4. Digit 13 is tested to determine if it is turned on. This is accomplished by ANDing the Accumulator with hex 1F (00011111). Should digit 13 be turned off, the five digit code returned, and now residing in the Accumulator will be XXX00000. ANDing this with 1F will leave the zero flag set, if the result is zero (digit 13 off). If digit 13 is off, the program will branch back to step 2 and reset the loop counter; if it is on, the program steps to the next operation.

5. If digit 13 was turned on, it must now be compared to the value of digit 13 from the last time the loop was performed. Note: The program does not initialize the digit 13 store location.

6. Digit 13 is compared to the value of digit 13 from the digit 13 stored previously. If digit 13, and the stored digit 13 are not equal, the program will branch back to step 2 and once again reset the loop counter. If digit 13 is equal to the value of the stored digit 13, the program will pass to step 7.

7. A delay of fifty (50) milliseconds permits digit 13 to change, should the Microcalculator be in a calculation mode. Digit 13 must be found equal to the value of the stored digit 13 for six loops before the routine returns.

8. After the delay has been completed, the loop counter is decremented.

9. The loop counter is tested to determine if it is equal to zero. If the loop counter is zero, the busy routine terminates, and returns to the main program. If the loop counter is not equal to zero, the program branches back to step 3 to get another digit 13 for test.

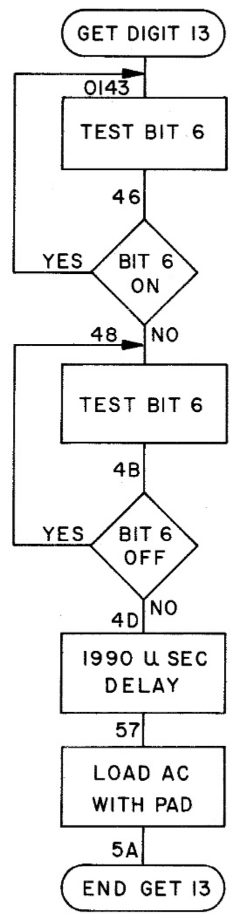


FIGURE 10

READ DISPLAY (Figure 11)

This part of the program is not a seperate subroutine, but is listed seperately to give the user a better idea of the structure of the program that is required to operate the Microcalculator.

This part of the program reads the 14 digit display back from the Microcalculator, and stores them in memory locations 0001 - 000E. These 14 storage locations are referred to as the MICROCALCULATOR DISPLAY TABLE (MDT).

1. For the fourteen digits that will be read into RAM memory, a table pointer is initialized to 14.

2. To read the digits from the Microcalculator, bit 7 of the I/O port must be set for output operation, and set equal to 0. Bits 0 - 6 are set for input.

3. Bit 6 of the incoming data is the SYNC pulse, digit 1. To accurately sync on the leading edge of the SYNC pulse calls for a routine that will discard the pulse should it be read during the period that it is a 1. This is required to assure that the next time bit 6 goes to a 1 it will be the leading edge of the SYNC pulse.

4. Once the SYNC pulse has been detected, a delay of seventy (70) microseconds will be initiated. This will place the first read operation at the center of digit 1.

5. To read the incoming digits from the Microcalculator, the Accumulator is loaded with the I/O port data.

6. During the reading of the 14 display digits a tempo- storage location (0001 - 000E) is loaded with the incoming digits. The first digit will be stored at location 000E, and the last, or 14th, will be stored at location 0001.

7. After each digit is read into the Accumulator and stored in the MDT, the table pointer which loads the digits into memory is decremented.

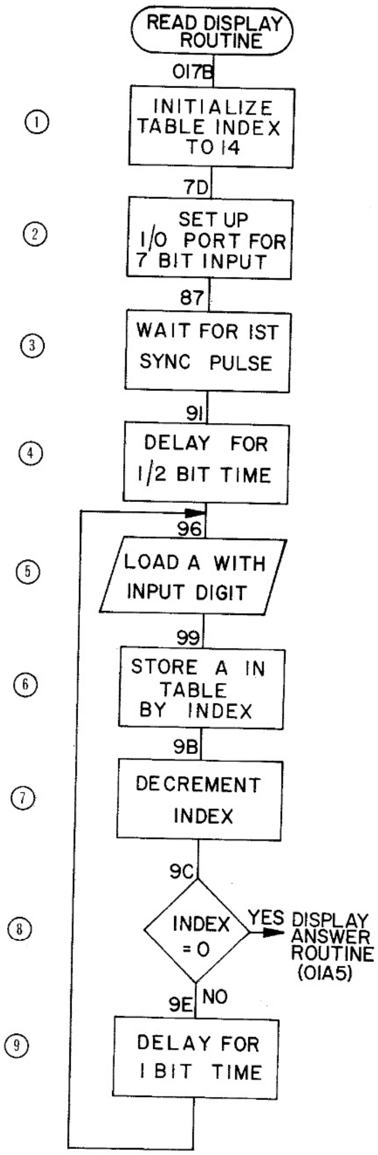


FIGURE 11

8. This instruction tests the table pointer to determine if all 14 digits have been read. If they have, the program will branch to the PRINT ANSWER portion of the program; if they have not, the program will advance to the next step.

9. To read the next digit, a delay of 160 microseconds is required. This 160 microsecond delay must include all machine cycle delays from the time the Accumulator loaded the input data at address 0196. After the delay, the program branches back to step 5 to read the next digit.

PRINT ANSWER (Figure 12)

The PRINT ANSWER portion of the program is the final part of the program. It is this part of the program that delivers the six bit ASCII code for digits 0 -9, the minus (-) sign and the blank space character to a routine which may display the answer on a CRT terminal or print the answer on a TTY.

1. The 14 digit display (the answer) resides in memory locations 0001 - 000E (the MDT), with the first digit in location 000E. To load the Accumulator with the MDT, a pointer is initialized to 14. This pointer will be referred to as the MDT pointer.

2. Each time a digit is retrieved from the MDT, two operations must be performed. The first will be to load the Accumulator with the coded byte for conversion to ASCII equivalent; the second will be to determine if the character to be displayed should be followed by a decimal point.

3. The code which represents the character of the display digit is contained in bits 0 - 4. To isolate these bits, the Accumulator is ANDed with hex 1F. This operation will leave the Accumulator with bits 0 - 4 equal to the character code, and bits 5 - 7 equal to 0. The value in the Accumulator (charac-code bits 0 - 4) will be used as an index to the ATABLE, which holds the value of the character in the ASCII code. To reduce the range of the table pointer, and thereby save on memory, a technique is devised that subtracts hex 11 from the code if bit 4 is equal to a 1.

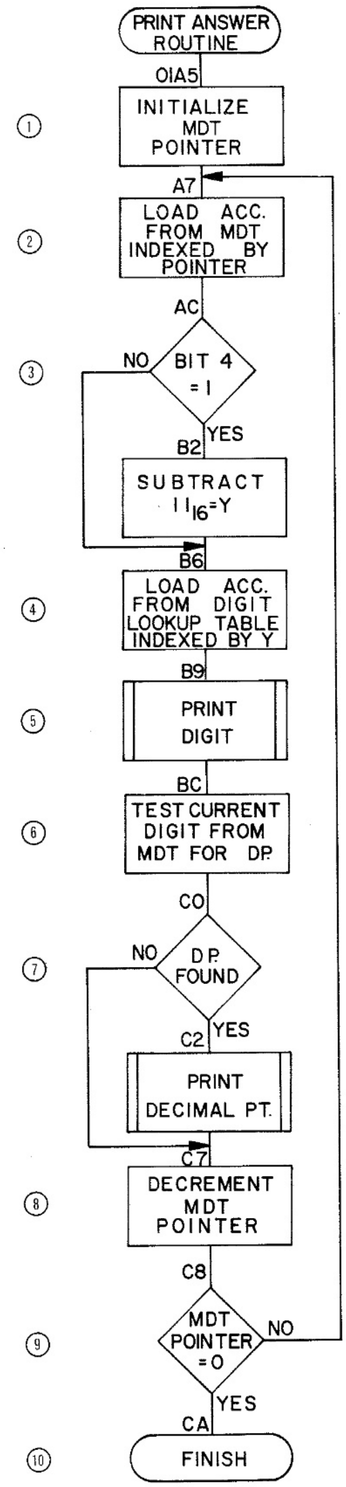


FIGURE 12

4. Using the ATABLE pointer (register Y), the ASCII value of the coded character is loaded into the Accumulator.

5. The ASCII equivalent of the display digit now resides in the Accumulator, and may be delivered to a routine which outputs the character to a TTY or CRT terminal.

6. At address 01A7 the value of the MDT pointer was saved by placing it into address 01BF. This value is the location in the MDT of the character that was just printed or displayed. At this time, bit 5 is tested to determine if it is a decimal point.

7. If bit 5 is equal to a 1, a decimal point is now printed. If bit 5 is equal to a 0, a branch is taken to address 01C7.

8. The MDT pointer is decremented.

9. If the MDT pointer is not equal to 0, the program goes back for another digit. If the MDT pointer is equal to 0 the program advances to the next step.

10. At this time the program is complete. This step can be a return instruction, which would make the entire program a subroutine; or it may jump to any location the user requires.

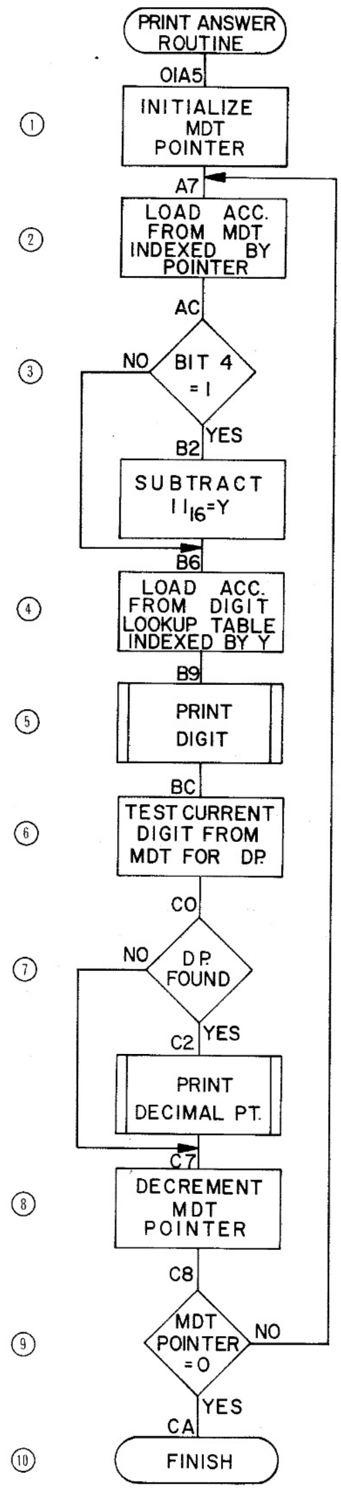


Figure 12 (Repeated for Reference)

MICROCALCULATOR MODEL 85 SOFTWARE EXAMPLE

The following program will serve as an example of the software required to operate the Microcalculator. The program was written for the 6502 microprocessor, and is supported in this example by the MOS Technology Inc., KIM Microcomputer.

The Microcalculator may be operated in two seperate modes, with respect to the display of the answer. The first mode uses the fourteen digit display as the source of the answer and does not read the display back into the microprocessor system. This mode of operation requires the program listed from address 0100 thru017A.The second mode reads the display back into the microprocessor for display on a TTY or CRT terminal. To support this mode of operation the entire program from address 0100 thru 01DF is required, along with the subroutine to output a character in the Accumulator to the TTY or CRT terminal. A thorough understanding of the operation of the Microcalculator will permit the user to modify the following program to suit his application.

```
                ;       MICROCALCULATOR MODEL 85 PROGRAM
                ;       FOR THE 6502 MICROPROCESSOR
                ;       STORAGE REQUIREMENTS : 237 BYTES
                ;       MICROCOMPUTER: MOS TECH  KIM-1
                ;       I/O INTERFACE: MICROCALCULATOR MODEL 85
                ;                      AND OPTIONAL TTY OR CRT TERMINAL
                ;       REQUIRED SUPPORTING SOFTWARE: KIM-1
                ;       $ INDICATES NUMBER IS IN HEXADECIMAL
                ;       # INDICATES IMMEDIATE ADDRESSING MODE
                ;       ' INDICATES ASCII LITERAL
                ;       EQUATES
                ;       SET UP FOR THE KIM 6530-003 I/O
                ;
                PAD     =$1700      6530 A DATA
                PADD    =$1701      6530 A DATA DIRECTION
                TIMERI  =$1707      DIVIDE BY 1024 TIME
                TIMERO  =$1707      READ TIME OUT BIT
                ;
                ;       SUBROUTINE TO OUTPUT AN ASCII CHARACTER IN THE
                ;       ACCUMULATOR TO A TTY OR CRT TERMINAL
                ;       PART OF THE KIM SYSTEM
                ;
                OUTCH   =$1E20
                ;
                ;
                ;
                ;       MICROCALCULATOR PROGRAM FIXED  AREA IN PAGE 0
                ;
0001            MDT     *=*+14      MICROCALCULATOR DISPLAY TABLE
000F            CITAB   *=*+224     CALCULATOR INSTRUCTION TABLE
                ;       THIS PROGRAM RUNNING IN THE KIM SYSTEM  MAY
                ;       HAVE UP TO 224 BYTES IN PAGE 0 TO USE FOR INPUT
                ;       INSTRUCTIONS TO THE MICROCALCULATOR
                ;
```

```
0100                      *=$0100
0100  A2 00     INIT      LDX #$00    INITILIZE POINTER TO ZERO
0102  A9 FF     NEXT      LDA #$FF    SET UP I/O PORT FOR 8 BIT OUTPUT
0104  8D 01 17            STA  PADD
0107  B5 0F               LDA  CITAB,X LOAD A WITH INPUT INSTRUCTION
0109  C9 FF               CMP #$FF    LOOK FOR END OF INSTRUCTIONS
010B  D0 03               BNE  OUT1   BRANCH TO OUT1 IF NOT LAST INSTR
010D  4C 7B 01            JMP  READ   JUMP TO READ IF END OF INSTRUCTIONS
0110  8E DF 01  OUT1      STX  SAVX   SAVE POINTER
0113  8D 00 17            STA  PAD    OUTPUT INSTR TO MICROCALCULATOR
0116  A2 02     STROB     LDX #$02    SET UP 40 MILLISECOND DELAY
0118  A9 13     LTIME     LDA #$13
011A  8D 07 17            STA  TIMERI SET TIMER FOR $13 X 1024 CYCLES
011D  2C 07 17  TIME      BIT  TIMERO TEST TIMER FOR OUTPUT BIT
0120  10 FB               BPL  TIME
0122  CA                  DEX         DECREMENT X
0123  D0 F3               BNE  LTIME  IF X NOT 0 BRANCH BACK TO LTIME
0125  A9 C0               LDA #$C0    LOAD END OF INSTRUCTION
0127  8D 00 17            STA  PAD    OUTPUT C0 TO MICROCALCULATOR
012A  20 34 01            JSR  BUSY   CALL BUSY STATUS SUBROUTINE
012D  AE DF 01            LDX  SAVX   RESTORE POINTER
0130  E8                  INX         INCREMENT POINTER FOR NEXT INSTR
0131  4C 02 01            JMP  NEXT   GO BACK AND GET NEXT INSTRUCTION
                ;
                ;
                ;         BUSY SUBROUTINE
                ;
                ;
0134  A9 80     BUSY      LDA #$80    SET UP I/O PORT FOR BIT 7 = OUTPUT
0136  8D 01 17            STA  PADD   BITS 0 - 6 SET TO INPUT MODE
0139  A9 00               LDA #$00    SET BIT 7 = 0 (INPUT MODE)
013B  8D 00 17            STA  PAD
013E  A9 06     ILOOP     LDA #$06    INITILIZE LOOP COUNTER TO SIX
0140  8D DD 01            STA  LCOUNT SAVE LOOP COUNTER VALUE
0143  2C 00 17  GET13     BIT  PAD    THIS LOOP TEST DIGIT 13 FOR BUSY STATE
0146  70 FB               BVS  GET13  DISCARD SYNC PULSE IF = 1
0148  2C 00 17  SYNC      BIT  PAD    WAIT FOR NEXT SYNC PULSE
014B  50 FB               BVC  SYNC
014D  A0 B5               LDY #$B5    BEGIN TB DELAY TO CENTER OF DIGIT 13
014F  A2 01     T1        LDX #$01    DELAY BEGINS AT START OF DIGIT 1 SYNC
```

```
0151  CA          T2     DEX           PULSE AND IS EQUAL TO REGISTER Y(181)
0152  D0 FD              BNE  T2       TIMES LOOP = 1991 MICROSECONDS
0154  88                 DEY
0155  D0 F8              BNE  T1
0157  AD 00 17           LDA  PAD      ACCUMULATOR NOW HAS DIGIT 13
015A  29 1F              AND #$1F      TEST DIGIT 13 TO SEE IF IT IS ON
015C  F0 E0              BEQ  ILOOP    BRANCH TO ILOOP IF DIGIT 13 IS OFF
015E  CD DE 01           CMP  DSTORE   COMPARE DIGIT 13 TO LAST DIGIT 13
0161  8D DE 01           STA  DSTORE   STORE NEW DIGIT 13
0164  D0 D8              BNE  ILOOP    BRANCH TO ILOOP IF DIGIT 13 = DSTORE
0166  A2 18              LDX #$18      BEGIN A 50 MILLISECOND DELAY
0168  A9 02       T3     LDA #$02
016A  8D 07 17           STA  TIMERI   SET TIMER FOR 2 TIMES 1024 CYCLES
016D  2C 07 17    T4     BIT  TIMERO   TEST TIMER OUTPUT BIT
0170  10 FB              BPL  T4       BRANCH TO T4 IF NOT END OF DELAY
0172  CA                 DEX
0173  D0 F3              BNE  T3       IF X NOT 0, BRANCH TO T3
0175  CE DD 01           DEC  LCOUNT   DECREMENT THE LOOP COUNT
0178  D0 CC              BNE  GET13    BRANCH TO GET13 IF LOOP COUNT NOT 0
017A  60                 RTS           RETURN
                  ;
                  ;
                  ;      READ DISPLAY
                  ;      HERE TO END IS REQUIRED IF MICROPROCESSOR IS
                  ;      TO READ THE 14 DIGIT DISPLAY FROM THE MICRO-
                  ;      CALCULATOR AND DISPLAY ON TTY OR CRT
                  ;
017B  A2 0E       READ   LDX #$0E      SET MDT INDEX POINTER TO 14
017D  A9 80              LDA #$80      SET UP I/O PORT FOR BIT 7 = OUTPUT
017F  8D 01 17           STA  PADD     BITS 0 - 6 TO INPUT MODE
0182  A9 00              LDA #$00      SET BIT 7 = 0 (INPUT MODE)
0184  8D 00 17           STA  PAD
0187  2C 00 17    T5     BIT  PAD      SYNC ON DIGIT 1
018A  70 FB              BVS  T5
018C  2C 00 17    T6     BIT  PAD
018F  50 FB              BVC  T6
0191  A0 0B              LDY #$0B      BEGIN 70 MICROSECOND DELAY TO THE
0193  88          T7     DEY           CENTER OF DIGIT 1
0194  D0 FD              BNE  T7
0196  AD 00 17    LOAD   LDA  PAD      ACCUMULATOR HAS CODED INPUT DIGIT
0199  95 00              STA  MDT,X    STORE DIGIT IN MDT INDEXED BY X
019B  CA                 DEX           DECREMENT MDT TABLE POINTER
019C  F0 07              BEQ  DSPLAY   BRANCH TO DSPLAY IF 14 DIGITS READ
019E  A0 1E       T8     LDY #$1E      BEGIN 160 MICROSECOND DELAY
01A0  88                 DEY
01A1  D0 FD              BNE  T8
```

```
01A3  F0 F1               BEQ  LOAD   BRANCH TO LOAD FOR NEXT DIGIT
                    ;
                    ;
                    ;
01A5  A2 0E      DSPLAY   LDX #$0E    SET MDT POINTER TO 14
01A7  8E BF 01   NEXTD    STX  NDEX   STORE MDT POINTER AT ADDRESS 01BF
01AA  B5 00               LDA  MDT, X LOAD A WITH MDT INDEXED BY X
01AC  29 1F               AND #$1F    STRIP OFF BITS 5 - 7
01AE  C9 10               CMP #$10    TEST BIT4 FOR A DECIMAL POINT CODE
01B0  90 03               BCC  LASCII BRANCH TO LASCII IF BIT 4 NOT A 1
01B2  38                  SEC         SET THE CARRY
01B3  E9 11               SBC  #$11   SUBTRACT $11 FROM CODE
01B5  A8         LASCII   TAY         SET UP Y AS INDEX FOR ATABLE
01B6  B9 CD 01            LDA  ATABLE,Y
01B9  20 A0 1E            JSR  OUTCH  PRINT OR DISPLAY CHARACTER
01BC  A9 20               LDA #$20    LOAD    A    WITH BIT 5 MASK
01BE  24 00               BIT  MDT    TEST DIGIT IN TABLE FOR DECIMAL POINT
01C0  F0 05               BEQ  NODEC  BRANCH TO NODEC IF NO DECIMAL FOUND
01C2  A9 2E               LDA #' .    LOAD A WITH ASCII DECIMAL POINT
01C4  20 A0 1E            JSR  OUTCH  PRINT OR DISPLAY DECIMAL POINT
01C7  CA         NODEC    DEX         DECREMENT MDT POINTER
01C8  D0 DD               BNE  NEXTD  BRANCH TO NEXTD IF 14 DIGITS NOT OUT
01CA  4C 4F 1C            JMP  KIM    JUMP TO KIM SYSTEM
                    ;
                    ;     ADDRESS 1C4F IS THE START ADDRESS OF THE KIM
                    ;     SYSTEM.....REPLACE WITH THE ADDRESS OF
                    ;     OWN SYSTEM
                    ;
                    ;     ATABLE
                    ;     THIS TABLE HOLDS THE DIGITS 0 - 9, THE MINUS (-)
                    ;     SIGN AND A BLANK SPACE IN ASCII CODE. THE FOUR
                    ;     LOCATIONS LOADED WITH 00 ARE NOT USED AS PART
                    ;     OF THE TABLE
                    ;
                    ;          ' ', '4 , ' 1 , ' 7 , '5 , $00, '9 , '0
01CD  20         ATABLE   .BYTE $20,$34,$31,$37,$35,$00,$39,$30
01CE  34
01CF  31
01D0  37
01D1  35
01D2  00
01D3  39
01D4  30
```

```
                ;
                ;
                ;                   ' - , $00,$00,$00,'6 ,'2 , '8 , '3
01D5  2D                      .BYTE $2D,$00,$00,$00,$36,$32,$38,$33
01D6  00
01D7  00
01D8  00
01D9  36
01DA  32
01DB  38
01DC  33
                ;
                NDEX     =$01BF      INDEX VALUE FOR DIGIT UNDER TEST
01DD            LCOUNT   *=*+1       SAVE AREA FOR LOOP COUNTER
01DE            DSTORE   *=*+1       SAVE AREA FOR DIGIT 13
01DF            SAVX     *=*+1       SAVE AREA FOR INDEX REGISTER X
                ;
                ;
                ;        END
```

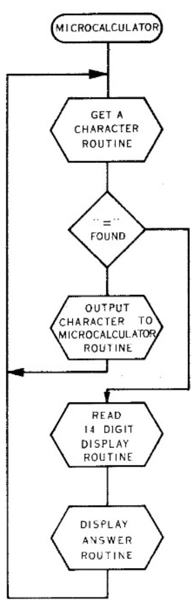


FIGURE 13

USING A TERMINAL AS A SCIENTIFIC CALCULATOR

Thru software, a TTY or CRT terminal may be used as a keyboard for the Microcalculator. The keys of the keyboard would be assigned designations to coorespond to the keyboard of the Microcalculator. In the program (see figure 13) the key would be tabled to the exact instruction required by the Microcalculator. This instruction would be issued to the Microcalculator via the Input routine. The program would test Busy status and return control to the keyboard. If the results were to be displayed back at the terminal, the "=" key could be used to indicate read the Microcalculator display back.

In this manner the manner the Microcalculator would respond as if it were a conventional scientific calculator. Another approach would load up the input table to the Microcalculator; and on receipt of the "=" key input, the entire table of input instructions would be delivered to the Microcalculator.

MICROCALCULATOR MODEL 85 INSTRUCTION SET

As with a standard scientific calculator, most keys control two functions. One of the functions operates in the normal mode of control, and is generally the function printed on the surface of the key. The other function is printed above the key, and operates when preceded by a shift key input. The Microcalculator model 85 also requires the service of a shift instruction to get the second function from a given instruction. Figure 3 outlines the complete instruction set for the model 85. The left hand column indicates the function that will be performed without the use of the SHIFT instruction (9D). The right hand column indicates the functions that will be performed when preceded by the SHIFT instruction (9D). The LST X (87), STO (A1), and RCL (A5) do not require the SHIFT instruction.

The following instructions describe each instruction operating in the NORMAL and the SHIFT modes. The characters in parenthesis represent that instruction in hexadecimal notation.

0...........(87)....
1...........(8F)....
2...........(9F)....
3...........(A3)....
4...........(97)....
5...........(9C)....
6...........(A0)....
7...........(93)....
8...........(98)....
9...........(A4)....

These are the standard digit keys and are used to enter all numbers, decimal point positions, register store and recall locations, and exponents.

+..........(85)....
−..........(89)....
X ....(8D)....
÷ ....(91)....

ADD, SUBTRACT, MULTIPLY, and DIVIDE functions. To use any of these instructions both numbers must be entered first.

Example: To multiply 5 by 4
Sequence of instructions:

5.......................(9C)
ENT..................(A7)
4.......................(97)
X.......................(8D) DISPLAYED ANSWER 20.00

(.)..........(9B) DECIMAL POINT.

ENT......(A7)...............ENTER. The Model 85 has no = key instruction, instead there is the ENT instruction. This instruction operates in conjunction with a parenthesis free logic, called Reverse Polish Notation (RPN).

Y^X ....(84).....RAISE Y TO THE X POWER. This instruction raises the number in the Y register to the power in the X, or display register.

EXAMPLE: Find 4^6
Sequence of instructions:

4.....................(97)
ENT.................(A7)
6.....................(A0)
Y^X..................(84) DISPLAYED ANSWER 4096.00

X ⇆ Y...........(8C).....EXCHANGE REGISTERS X AND Y. This instruction causes the contents of registers X and Y to be exchanged. See the section on the operation of the stack for details.

↓ ....(90).....ROLL STACK DOWN. The contents of the four (4) operational registers may be reviewed at any time. Each time this instruction is entered the contents of the stack shift downward one register. See the section on the operation of the stack for details.

CHS.................(94).....CHANGE SIGN. Each time this instruction is entered the sign of the number displayed (X register) is changed. To enter a negative number, enter the number and the CHS instruction. Should the CHS instruction be entered again the sign of the number will change back to positive (No indication).

EE...................(88).....ENTER EXPONENT. Numbers having exponents may be entered with this instruction. The instruction enters the exponent as a power of ten.

EXAMPLE: 20 X 10^6
Sequence of instructions:

2.....................(9F)
0.....................(87)
EE...................(88)
6.....................(A0)
ENT.................(A7) DISPLAYED ANSWER 20000000.00

Σ+.................(95).....SUMMATION. This instruction is used for statistical calculations. Before starting the summation enter SHIFT(9D),CLR(8B) to clear the previous contents of the stack registers.

EXAMPLE: Find the arithematic mean and standard deviation for the following series: 1, 3, 5, 7
Sequence of instructions:

SHIFT.............(9D)
CLR.................(8B)
1.....................(8F)
Σ+.................(95)
3.....................(A3)
Σ+.................(95)

5......................(9C)
Σ+.................(95)
7.....................(93)
Σ+..................(95)
SHIFT............(9D)
$\bar{x}$, S................(95) DISPLAYED ANSWER 4.00 Mean
X ⇆ Y............(8C) DISPLAYED ANSWER 2.58 Standard Dev.

Storage register 7 contains the total number of samples entered.
Storage register 8 contains the sum of squares of entries.
Storage register 9 contains the sum of the X register entries.

Ln...................(9A)....NATURAL LOG. This instruction takes the log of the value in the display register to base e (2.71828.......).

EXAMPLE: Find the natural log of 3.2 to 9 decimal places.
Sequence of instructions:
DSP.................(9D)
9......................(A4)
3......................(A3)
(.)....................(9B)
2......................(9F)
Ln...................(9A) DISPLAYED ANSWER 1.163150810

SIN................ (9E)....
COS................ (A2)... TRIGONOMETRIC FUNCTIONS. These instructions perform the Sine, Cosine,
TAN............... (A6) and Tangent of a given angle.

EXAMPLE: Find the Sine of 30°
Sequence of instructions:
3......................(A3)
0..................... (87)
SIN................. (9E) DISPLAYED ANSWER 0.50

INV.................(A1).... INVERSE FUNCTION. This instruction when preceeding an applicable instruction will perform the inverse of that function.

EXAMPLE: Find Sin^{-1} 0.5 in degrees
Sequence of instructions:
(.).................. (9B)
5.................... (9C)
INV............... (A1)
SIN................ (9E) DISPLAYED ANSWER 30.00

THE FOLLOWING INSTRUCTIONS ARE PERFORMED USING THE INV INSTRUCTION

Sin^{-1} ....
Cos^{-1} ... ARC SINE, ARC COSINE, ARC TANGENT.These functions require the
Tan^{-1} ... use of the INV instruction.

EXAMPLE: Find the Sin^{-1} .5

Instruction sequence:

(.)................... (9B)
5..................... (9C)
INV................ (A1)
SIN................. (9E) DISPLAYED ANSWER 30.00

$Sinh^{-1}$ ....
$Cosh^{-1}$ ... HYPERBOLIC ARC SINE, HYPERBOLIC ARC COSINE, HYPERBOLIC
$Tanh^{-1}$ .. ARC TANGENT. These functions require the use of the INV instruction.

EXAMPLE: Find the $Cosh^{-1}$ 15.6

Sequence of instructions:

1..................... (8F)
5..................... (9C)
(.).....................(9B)
6..................... (A0)
INV.................(A1)
HYP................(A5)
COS................ (A2) DISPLAYED ANSWER 3.44

$\sqrt[X]{Y}$ Xth ROOT OF Y. This function requires the use of the INV instruction.

EXAMPLE: Find the 5th root of 81. (X = 5, Y = 81)

Sequence of instructions:

8..................... (98)
1..................... (8F)
ENT................ (A7)
5..................... (9C)
INV.................(A1)
Y^X .... (84) DISPLAYED ANSWER 2.41

Σ- SUMMATION MINUS. This instruction may be used to correct an entry of Σ+ which was entered in error.

EXAMPLE: Enter Σ-

INV................ (A1)
Σ+ .... (95) Σ- entered.

e^x BASE e RAISED TO THE X POWER. This instruction uses the INV instruction.

EXAMPLE: Raise base e to the 5th power.

Sequence of instructions:

5.................... (9C)
INV............... (A1)
Ln.................. (9A) DISPLAYED ANSWER 148.41

10^x 10 TO THE X POWER. This instruction uses the INV instruction.

EXAMPLE: Raise 10 to the 6th power.

Sequence of instructions:

6.....................(A0)
INV............... (A1)
SHIFT.............(9D)
LOG............... (9A) DISPLAYED ANSWER 1000000.00

F° → C° FARHENHEIT TO CENTIGRADE CONVERSION. This instruction uses the INV instruction.

EXAMPLE: Convert 47°F to Centigrade.

Sequence of instructions:

4.................... (97)
7.................... (93)
INV............... (A1)
SHIFT.............(9D)
C → F.............. (91) DISPLAYED ANSWER 8.33 (°C)

GAL→LTR GALLONS TO LITERS CONVERSION. This instruction uses the INV instruction.

EXAMPLE: Convert 7 gallons to liters.

Sequence of instructions:

7.................... (93)
INV............... (A1)
SHIFT.............(9D)
LTR→GAL.......(8D) DISPLAYED ANSWER 26.50 (Ltr)

IN→CM INCHES TO CENTIMETERS CONVERSION. This instruction uses the INV instruction.

EXAMPLE: Convert 8.6 inches to centimeters.

Sequence of instructions:

8.................... (98)
(.).................. (9B)
6.................... (A0)
INV............... (A1)
SHIFT.............(9D)
CM→IN........... (89) DISPLAYED ANSWER 21.84 (Cm)

LB→KG POUNDS TO KILOGRAMS CONVERSION. This instruction uses the INV instruction.

EXAMPLE: Convert 8 pounds to kilograms.

Sequence of instructions:

8..................... (98)
INV................ (A1)
SHIFT.............(9D)
KG→LB........... (85) DISPLAYED ANSWER 3.63 (Kg)

X^2 SQUARE X. This instruction uses the INV instruction.

EXAMPLE: Square 45.

Sequence of instructions:

4......................(97)
5.................... (9C)
INV................ (A1)
SHIFT............ (9D)
$\sqrt{x}$.................. (84) DISPLAYED ANSWER 2025.00

RAD→ CONVERT RADIANS TO DEGREES. This instruction will convert the display in radians to degrees and uses the INV instruction.

EXAMPLE: Convert 4.3 radians to degrees.

Sequence of instructions:

4..................... (97)
(.)................... (9B)
3..................... (A3)
INV................ (A1)
SHIFT.............(9D)
→RAD............ (A2) DISPLAYED ANSWER 246.37 (°)

SPECIAL CASES INVOLVING THE USE OF THE INV INSTRUCTION

CONVERT POLAR COORDINATES TO RECTANGULAR COORDINATES.

EXAMPLE: Convert polar coordinates 5, 36.87° to rectangular coordinates.

Sequence of instructions:

3..................... (A3)
6..................... (A0)
(.)....................(9B)
8..................... (98)
7.................... (93)
ENT............. (A7)
5..................... (9C)
INV................ (A1)
SHIFT............ (9D)
→POL...............(9E) DISPLAYED ANSWER 4.00 = X
X⇆Y ....(8C) DISPLAYED ANSWER 3.00 = Y

X and Y are the coordinates in rectangular format.

CONVERT HYPERBOLIC POLAR COORDINATES TO RECTANGULAR COORDINATES

EXAMPLE: Find the rectangular coordinates for the hyperbolic polar coordinates 5, 1.5 to nine decimal places.

Sequence of instructions:

DSP................(9D)
9..................... (A4)
1..................... (8F)
(.)...................(9B)
5..................... (9C)
ENT................ (A7)
5..................... (9C)
INV.................(A1)
HYP................(A5)
SHIFT.............(9D)
→POL...............(9E) DISPLAYED ANSWER 11.762048076 = X
X ⇆ Y............(8C) DISPLAYED ANSWER 10.646397276 = Y

CONVERT FROM SCIENTIFIC NOTATION DISPLAY BACK TO BUSINESS DISPLAY

EXAMPLE: The display is indicating π as 3.141592653 00 which is in scientific notation, convert the display to business display.

Sequence of instructions:

INV.................(A1)
SHIFT.............(9D)
SCI..................(A7) DISPLAYED ANSWER 3.14159265359

THIS CONCLUDES THE USE OF THE INV INSTRUCTION

HYP ....(A5).... HYPERBOLIC INSTRUCTION. This instruction is used to find the hyperbolic sine, cosine, and tangent.

EXAMPLE: Find the Sinh 30 and display in scientific notation.

Sequence of instructions:

3..................... (A3)
0......................(87)
HYP.................(A5)
SIN................. (9E)
SHIFT..............(9D)
SCI................. (A7) DISPLAYED ANSWER 5.343237290 12

CLX................ (8B).... CLEAR X INSTRUCTION. This instruction has the following functions:

1. Clearance of the flashing display in case of overflow or underflow.
2. Clearance of the two shift flags STO/INV and RCL/HYP.
3. Clearance of the display, which is also the X register.

DSP/SHIFT.....(9D).... DECIMAL SELECT & SHIFT. This instruction serves two functions. One is to select the number of decimal positions, and the other is to set up the SHIFT instruction to permit dual function instructions. Each time the (9D) instruction is issued, the next instruction that follows will be the instruction from the right hand column of figure 3. Note that there are no instructions in the right hand column opposite the digits 1 thru 9. This means that each time the (9D) instruction is issued, and followed by a digit 1 thru 9, the number of decimal places will be established equal to the digit that follows the (9D) instruction.

EXAMPLE: Set up the display for six place decimal operation.
Sequence of instructions:
DSP.................(9D)
6..................... (A0) DISPLAYED ANSWER 0.000000
(Note: All problems worked will display the answer to 6 decimals)

THE FOLLOWING INSTRUCTIONS REQUIRE THAT THE SHIFT (9D) INSTRUCTION PRECEDE THEM. THEY ARE THE INSTRUCTIONS IN THE RIGHT COLUMN OF FIG. 3

KG→LB............(85)... KILOGRAMS TO POUNDS CONVERSION. This instruction converts the number in the display to pounds.
EXAMPLE: Convert 3 kilograms to pounds.
Sequence of instructions:
3..................... (A3)
SHIFT.............(9D)
KG→LB........... (85) DISPLAYED ANSWER 6.61 (Lb)

CM→IN.............(89)... CENTIMETERS TO INCHES CONVERSION. This instruction converts the number in the display to inches.
EXAMPLE: Convert 7 centimeters to inches.
Sequence of instructions:
7......................(93)
SHIFT.............(9D)
CM→IN.............(89) DISPLAYED ANSWER 2.76 (In)

LTR→GAL.......(8D).... LITERS TO GALLONS CONVERSION. This instruction converts the number in the display to gallons.
EXAMPLE: Convert 4 liters to gallons.
Sequence of instructions:
4......................(97)
SHIFT.............(9D)
LTR→GAL.......(8D) DISPLAYED ANSWER 1.06 (Gal)

C→F................ (91).... CENTIGRADE TO FARHENHEIT CONVERSION. This instruction converts the number in the display to farhenheit.

EXAMPLE: Convert 9°C to farhenheit.

Sequence of instructions:

9......................(A4)
SHIFT.............(9D)
C→F................ (91) DISPLAYED ANSWER 48.20 (°F)

π..................... (9B).... Pi CONSTANT. This instruction calls the constant π from memory and places it in the display (X register).

EXAMPLE: Display π in scientific notation.

Sequence of instructions:

SHIFT.............(9D)
π .... (9B) DISPLAYED ANSWER 3.14
SHIFT.............(9D)
SCI................. (A7) DISPLAYED ANSWER 3.141592653 00

SCI..................(A7).... SCIENTIFIC NOTATION DISPLAY. This instruction will convert a number in the display to scientific notation. (See π instruction above)

$\sqrt{X}$................. (84).... SQUARE ROOT OF X. This instruction performs a square root operation on the number in the X register, which must always be positive. (The X register is also the display)

EXAMPLE: Find the square root of 49.

Sequence of instructions:

4......................(97)
9......................(A4)
SHIFT.............(9D)
$\sqrt{X}$................. (84) DISPLAYED ANSWER 7.00

%......................(8C).... PERCENTAGE. This instruction is used to determine the percentage of a number entered.

EXAMPLE: Find 35% of 45.

Sequence of instructions:

4..................... (97)
5..................... (9C)
ENT................ (A7)
3......................(A3)
5......................(9C)
SHIFT.............(9D)
%.................... (8C) DISPLAYED ANSWER 15.75

Δ%...................(90).... PERCENTAGE DIFFERENCE. This instruction returns in percent the difference between the number entered into the Y register and the last entry (X register). The answer represents the percentage difference from the Y register entry.

EXAMPLE: An increase from 90 to 100 is what percentage increase ?
Sequence of instructions:

9......................(A4)
0..................... (87)
ENT................ (A7)
1..................... (8F)
0..................... (87)
0..................... (87)
SHIFT.............(9D)
Δ%.................. (90) DISPLAYED ANSWER 11.11 (%)

X!....................(94).... FACTORIAL. This instruction calculates the factorial of the number displayed.

EXAMPLE: Find the factorial of 8.
Sequence of instructions:

8......................(98)
SHIFT.............(9D)
X ! ....(94) DISPLAYED ANSWER 40320.00

1/X................. (88).... RECIPROCAL OF X. This instruction inverts the number displayed.

EXAMPLE: Find the reciprocal of 12.
Sequence of instructions:

1......................(8F)
2......................(9F)
SHIFT.............(9D)
1/X..................(88) DISPLAYED ANSWER 0.08

$\bar{X}$, S.................(95).... DISPLAY MEAN INSTRUCTION. This instruction will display the arithmetic mean when performing a statistical problem. See Σ+ instruction for example of usage.

LOG............... (9A).... LOGARITHM. This instruction will calculate the logarithm of the number that is displayed. (X Register).

EXAMPLE: Find the log of 2.
Sequence of instructions:

2..................... (9F)
SHIFT.............(9D)
LOG................(9A) DISPLAYED ANSWER 0.30

→ POL..............(9E).... POLAR CONVERSION. This instruction converts two values X and Y representing the rectangular coordinates to polar coordinates; r (magnitude) and θ (angle). In performing a calculation with this instruction, always enter the Y coordinate first, then the X coordinate.

EXAMPLE: Convert rectangular coordinates (4, 3) to polar coordinates with the angle θ expressed in degrees.

Sequence of instructions:

3..................... (A3)
ENT................ (A7)
4..................... (97)
SHIFT.............(9D)
→ POL..............(9E) DISPLAYED ANSWER 5 (magnitude)

(To display the angle θ in degrees to four decimal places)

X ⇆ Y............ (8C)
DSP.................(9D)
4..................... (97) DISPLAYED ANSWER 36.8699 (θ)

→ RAD.............(A2).... CONVERT TO RADIANS. This instruction will convert the number displayed to radians.

EXAMPLE: Convert 180° to radians.

Sequence of instructions:

1..................... (8F)
8......................(98)
0......................(87)
SHIFT.............(9D)
→ RAD.............(A2) DISPLAYED ANSWER 3.14 (radians)

RAD................(A6).... RADIAN MODE. This instruction will place the calculator in a radian mode of operation . This mode of operation will be indicated by a decimal point , which will be positioned at the extreme right of the display .

EXAMPLE: Set calculator for radian mode of operation.

Sequence of instructions:

SHIFT............ (9D)
RAD.............. (A6) DISPLAY 0.00 .

To return calculator to the non-radian mode of operation, perform the following sequence of instructions:

INV.................(A1)
SHIFT.............(9D)
RAD............... (A6) DISPLAY 0.00

CLR................ (8B).... CLEAR STACK AND LAST X. This instruction will clear the four stack registers W, Z, Y, and X. The LAST X is residing in register 0 and register 0 is also cleared. The remaining nine registers are not cleared. Refer to HOW THE STACK WORKS for details of the stack operation.

EXAMPLE: Clear the stack and the last X.
Sequence of instructions:

SHIFT............. (9D)
CLR................ (8B) Stack and Last X are now clear (= 0)

THE FOLLOWING THREE INSTRUCTIONS APPEAR IN THE SHIFT COLUMN OF FIG. 3 BUT DO NOT REQUIRE THE SERVICE OF THE SHIFT INSTRUCTION TO FUNCTION.

LST X............. (87).... LAST X. This instruction when preceded by the RCL (A5) instruction will recall the last X entry made prior to performing a calculation. This is a very useful instruction to reverse an entry should an error be made in the routine.

EXAMPLE: Illustrate the LAST X in error correction.
Sequence of instructions:

2...................... (9F)
ENT................ (A7)
3..................... (A3)
X..................... (8D) error made should be +
RCL................ (A5)
LST X............. (87)
÷..................... (91) reverse calculation of X
3..................... (A3)
+..................... (85) DISPLAYED ANSWER 5.00

STO................ (A1).... STORE . This instruction will store the display (X register) in any one of the nine general purpose registers. See STORING and RECALLING for details.

RCL.................(A5).... RECALL. This instruction will recall any of the nine general purpose registers to the display (X register). See STORING and RECALLING for details.

HOW THE STACK OPERATES

The Microcalculator Model 85 has fourteen memory registers. Four of these registers are referred to as the STACK. Of the remaining ten registers,one register (Register 0) is used to store the last number entered prior to a functional operation. This register (Register 0) is referred to as the LST X (Last X). The four STACK registers and register 0 (LST X) are used during the operation of a problem. The STACK registers are labeled W, Z, Y, and X.

The primary purpose of the STACK registers is to permit the solving of problems by using the Reverse Polish Notation (RPN). RPN is a procedure which permits the solving of problems without the use of parenthesis, and is characterized by the absence of an "=" sign.

All entries are made to the X register, which is also the display register. During a chain of calculations, using the RPN method, numbers which are entered via the ENT (A7) instruction are pushed down on the STACK. Each time an operation is performed such as multiply, add, subtract etc., the number that is used for the operation is retrieved from the Y register. After an operation is completed the Y register is replaced by the contents of the Z register; the contents of the Z register is replaced by the contents of the W register. During calculations register 0 holds the entry made to the X register prior to performing an operation, and may be recalled by executing the RCL (A5), and the LST X (87) instruction.

STORING AND RECALLING

In addition to the STACK and LST X (register 0), there are nine (9) general purpose registers, or user's registers which may be used to store and recall numbers. This feature permits the storing of intermediate data or answers in the Microcalculator, and after the program is complete calling them back. Each register holds a fourteen digit number . This amounts to an additional 126 bytes of storage capacity for the microprocessor.

All nine (9) registers may be used at any time except in problems involving the Σ + instruction. When the Σ + instruction is used registers 7, 8, and 9 are required in order to store the results. Refer to the Σ + instruction for details.

To store a number or result of a calculation issue the STO (A1) instruction followed by the number of the register to be used as storage.

EXAMPLE: Store the display (register X) in register 5, the display is indicating 345.78

Sequence of instructions:

STO................ (A1)
5..................... (9C) Register 5 now holds 345.78

(The display will still have the original 345.78 in it)

To recall any register, the number of the register that is to be recalled must be preceded by the recall instruction RCL (A5).

EXAMPLE: Recall register 5 from the example above, the display is indicating 22.67

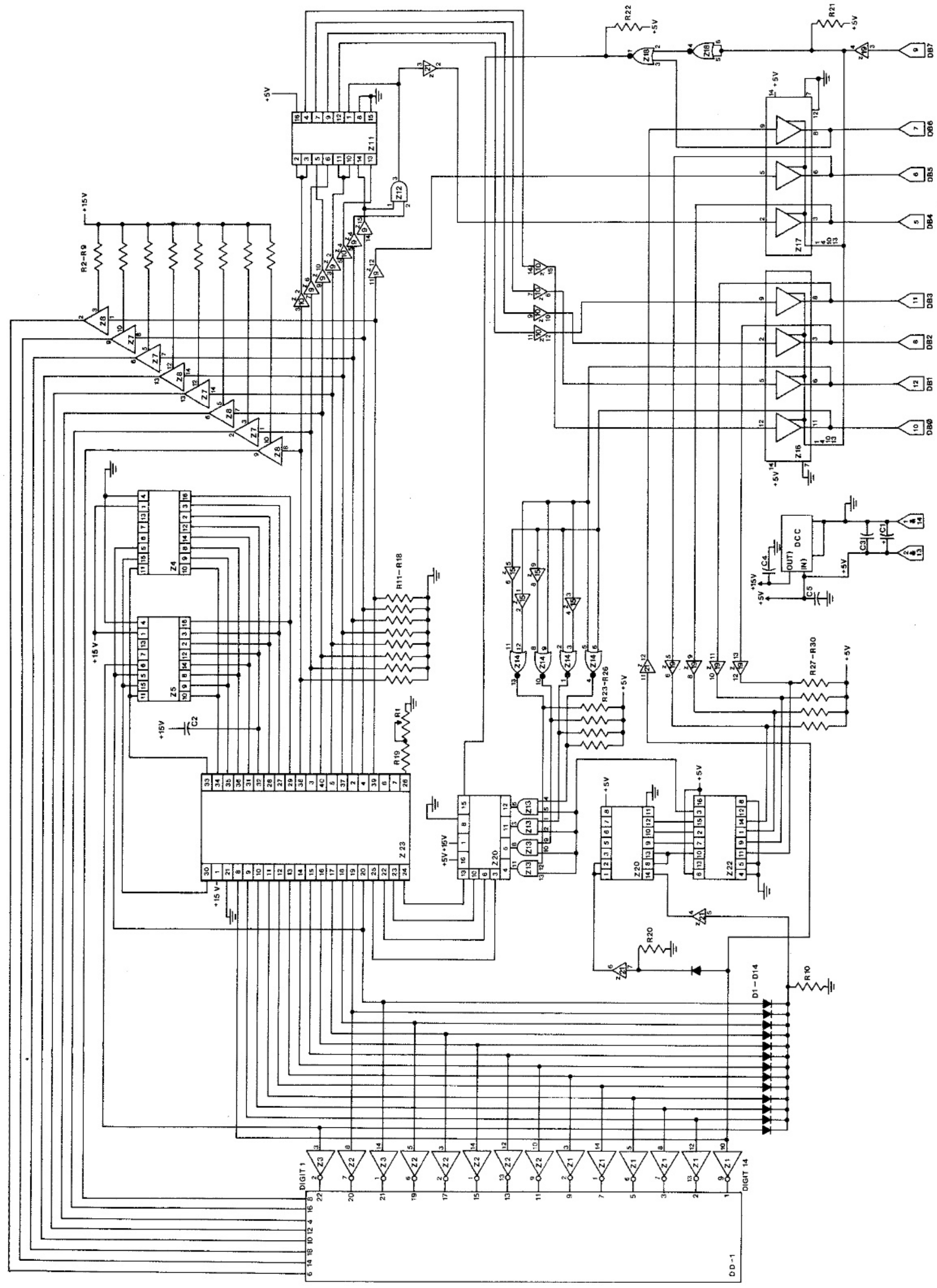
Sequence of instructions:

RCL................ (A5)
5..................... (9C) DISPLAY ANSWER 345.78

(The number 22.67 now resides in the Y register)

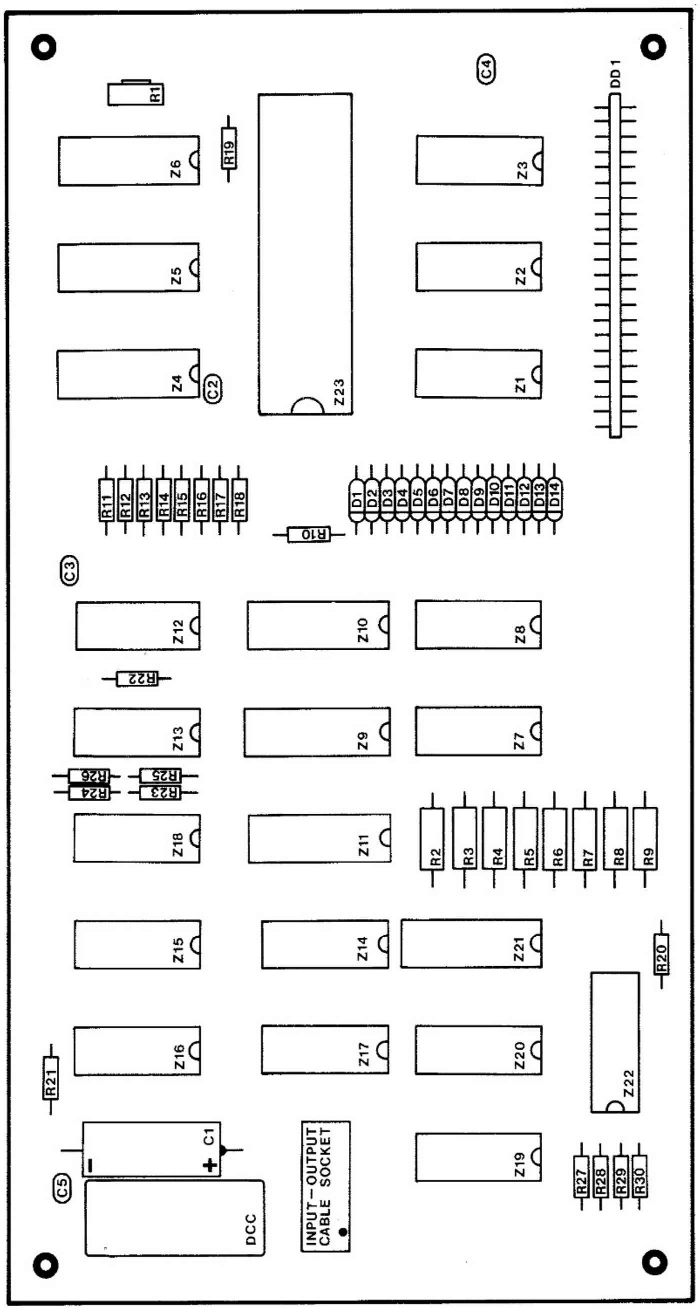
FLASHING DISPLAY

A flashing display indicates that there has been an overflow, underflow, or an illegal argument has been entered. To clear the problem the CLX (8B) instruction must be issued twice. The first time CLX is issued the display will stop flashing; the second time it is issued the display will be cleared.



MICROCALCULATOR MODEL 85 SCHEMATIC

APPENDIX A



MICROCALCULATOR MODEL 85 PARTS LAYOUT

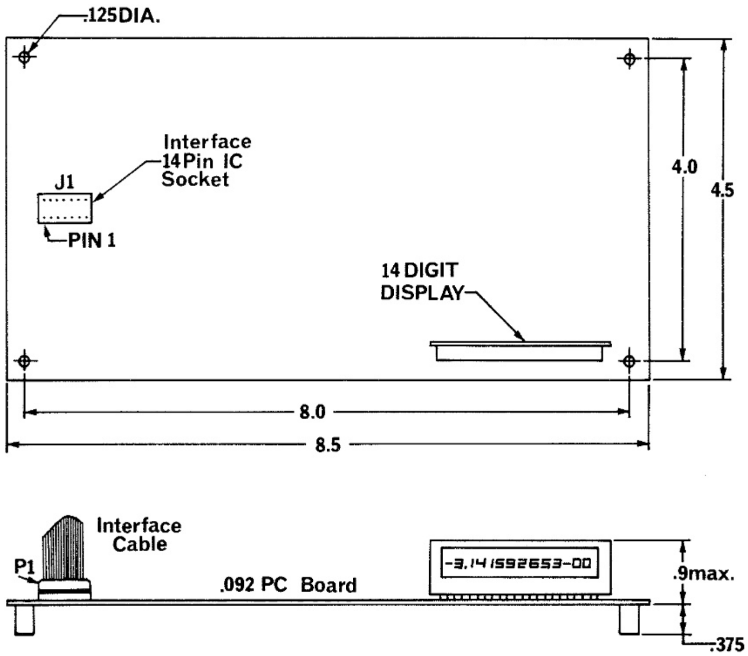
APPENDIX B

MICROCALCULATOR MODEL 85 PARTS LIST

REF	DESCRIPTION	RECOMMENDED VENDOR	PART NUMBER
PCB	Printed Circuit Board	Artisan Electronics Corp	PCB-85A
IFC	Interface Cable Assembly	Artisan Electronics Corp	IFC-85A
ICS-14	14 Pin Integrated Circuit Socket	Scanbe Corporation	US-2-14-160-G-B
ICS-16	16 Pin Integrated Circuit Socket	Scanbe Corporation	US-2-16-160-G-B
ICS-40	40 Pin Integrated Circuit Socket	Scanbe Corporation	US-2-40-160-G-B
DCC	DC-DC Converter 5 - 15 VDC	Elexon Power Systems	DC515
DD-1	14 Digit Display	Hewlett-Packard	5082-7447
D1 - D14	Silicon Diode	ITT	1N914A
C1	Capacitor, 220Mfd @ 16VDC	ICC	TDA-220-M16
C2	Capacitor, 220 Pf	Erie Corporation	8101100W5R0221K
C3 - C5	Capacitor, 0.1 Mfd	Erie Corporation	8121050651104M
R1	Potentiometer, 100K	Bournes Corporation	3389-100K
R2 - R9	Resistor, 2K @ ½W	Allen Bradley	RC20GF202J
R10 -R18	Resistor, 10K @¼W	Allen Bradley	RC07GF103J
R19 -R20	Resistor, 20K @¼W	Allen Bradley	RC07GF203J
R21	Resistor, 4.7K @¼W	Allen Bradley	RC07GF472J
R22 - R30	Resistor, 2K @¼W	Allen Bradley	RC07GF202J
Z1 - Z3	Integrated Circuit	ITT	ITT-500-5N
Z4	Integrated Circuit	Artisan Electronics Corp	ICA-85A
Z5	Integrated Circuit	Artisan Electronics Corp	ICB-85A
Z6	Integrated Circuit	Artisan Electronics Corp	ICC-85A
Z7 - Z8	Integrated Circuit	ITT	ITT-501-5N
Z9,10,21	Integrated Circuit	RCA	CD4050AE
Z11	Integrated Circuit	RCA	CD40257BE
Z12,13	Integrated Circuit	National Semiconductor	MM74C08N
Z14,18	Integrated Circuit	National Semiconductor	DM7402N
Z15	Integrated Circuit	National Semiconductor	DM7404N
Z16,17	Integrated Circuit	National Semiconductor	DM8093N
Z19	Integrated Circuit	National Semiconductor	DM7407N
Z20	Integrated Circuit	National Semiconductor	MM74C93N
Z22	Integrated Circuit	National Semiconductor	MM74C85N
Z23	Integrated Circuit	Artisan Electronics Corp	ICD-85

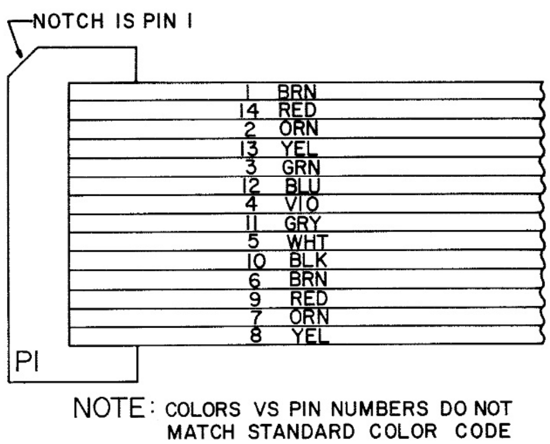
APPENDIX C

OUTLINE DRAWING OF MICROCALCULATOR MODEL 85



APPENDIX D

INTERFACE CABLE WIRING IDENTIFICATION



APPENDIX E

COMPLETE INSTRUCTION SET TEST PROGRAM

THE FOLLOWING SEQUENCE OF INSTRUCTIONS WILL EXERCISE THE MICROCALCULATOR THRU EACH INSTRUCTION IN THE MICROCALCULATORS INSTRUCTION SET. THROUGHOUT THE ROUTINE THE FIRST INPUT NUMBER IS RUN THRU ALL CALCULATIONS AND ALL REGISTERS ARE CHECKED. THE FINAL ANSWER IS -5.375970393 98

LOAD INSTRUCTIONS INTO MEMORY LOCATIONS 000F - 00D9 AND EXECUTE AN INPUT PROGRAM TO THE MICROCALCULATOR. (THE PROGRAM IN THIS BOOK STARTING AT ADDRESS 0100).

TEST TIME: 145 SECONDS.

000F	8B	9D	8B	9D	97	8B	A3	87	9D	A2	9D	A6	9E	9D	A1	A6
001F	A7	A1	9E	A2	A7	A1	A2	A6	A7	A1	A6	9D	88	85	85	85
002F	9B	97	93	A0	93	89	9F	87	8D	8F	87	91	A5	9E	A1	A5
003F	9E	A5	A2	A1	A5	A2	A5	A6	A1	A5	A6	A1	9D	9A	9D	9A
004F	9A	A1	9A	9D	A1	84	9D	84	97	9D	9E	8C	9C	9D	A1	9E
005F	98	84	98	A1	84	A5	87	A1	A5	9D	9E	8B	8F	95	A3	95
006F	A4	95	A1	95	98	95	9D	95	A5	93	A1	8F	A5	98	A1	9F
007F	A5	A4	A1	A3	A1	97	90	A1	9C	90	A1	A0	A5	8F	85	A5
008F	9F	85	A5	A3	89	A5	97	89	A5	9C	89	94	A5	A0	85	A5
009F	93	85	A5	98	85	A5	A4	89	8F	87	87	9D	91	9D	8D	9D
00AF	89	9D	85	9D	A1	85	9D	A1	89	9D	A1	8D	9D	A1	91	9F
00BF	87	A7	9C	87	9D	90	98	9D	8C	9F	A1	9D	8C	88	A3	8D
.00CF	A0	A4	9D	94	9D	9B	8D	9D	A7	94	FF					

APPENDIX F

IN-WARRANTY SERVICE

All Microcalculator Model 85's are warranted by Artisan Electronics Corporation against defects in workmanship and materials for a period of ninety (90) days from the date of delivery. During the warranty period Artisan Electronics Corporation will repair, or at its option replace at no charge components that prove to be defective; provided that the Microcalculator is returned, shipping prepaid to:

ARTISAN ELECTRONICS CORPORATION
5 EASTMANS ROAD
PARSIPPANY, NEW JERSEY 07054

This warranty does not apply if the Microcalculator has been damaged by accident or misuse, or as a result of repairs or modifications made by other than authorized personnel at Artisan.

No other warranty is expressed or implied. Artisan Electronics Corporation is not liable for consequential damages.

OUT-OF-WARRANTY SERVICE

Beyond the warranty period Artisan Electronics Corporation will repair the Microcalculator Model 85 for a reasonable service charge. All service work performed by Artisan beyond the warranty period is warranted for an additional ninety (90) days after shipment of the repaired product.

SHIPPING INSTRUCTIONS

It is the customer's responsibility to return the Microcalculator Model 85 with all shipping charges prepaid to the above address. For in-warranty service the Microcalculator Model 85 will be returned to the customer, shipping prepaid, by the fastest economical carrier.

For out-of-warranty service the customer will pay for shipping charges both ways. When returning the Microcalculator Model 85 be certain to use the anti-static shipping bag.

ARTISAN ELECTRONICS CORPORATION
5 EASTMANS ROAD
PARSIPPANY, NEW JERSEY 07054