TEKTRONIX®

# APPLICATION LIBRARY PROGRAM

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |
| | **MEMORY REQUIREMENT** |
| | 16K |
| **AUTHOR** | **PERIPHERALS** |
| Brian Diehm | |

**ABSTRACT**

INTRODUCTION

A. This program allows the 4051 GCS to be used in a manner similar to several hand-held calculators currently on the market, specifically those operating in reverse "Polish" notation with a 4 place operational stack, for example, the Hewlett Packard series.

B. The input to the program is completely accomplished by use of the function keys. Input is interactive with the user; just like a hand-held calculator.

C. Output is placed on the screen. As each new display is presented, it is written on the screen below the previous one. Also, a record of the input keys is provided on the screen.

| TITLE | ABSTRACT NUMBER |
| --- | --- |
| HAND CALCULATOR | 51/00-6001/0 |

HARDWARE REQUIREMENTS

The program requires a 4051 with 16K or more of memory. The proper overlay is required for the function keys.

PROGRAM LIMITATIONS

While the program is protected against invalid operations such as divide by zero, operations resulting in a result too large for the machine may result in an error message. The user may continue and disregard the message with no effect.

METHODOLOGY

The calculator program operates a stack, and problems are solved by utilizing reverse "Polish" logic. The stack has 4 positions.

OPERATING HINTS

N/A

REFERENCES

N/A

DESCRIPTION OF THE MAN/MACHINE INTERFACE

The user communicates with the program by using the function keys for input and the screen for readout. The manner of operation is similar to that of a hand-held calculator, except that previous results are saved on the screen as well as the directives used.

OPERATING INSTRUCTIONS

Loading - Place the program tape in the drive and press the "Auto-load" key. When the program clears the screen and prints "CLEAR" followed by a zero, the program is loaded and running.

Running the program with examples -

ACCESSING FUNCTIONS:

There are 10 function keys physically placed on the 4051 keyboard. The number of different entries is increased to 20 by the use of the shift keys.

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

This program further increases the number of available functions to 39 by the use of a "shi t" function. On the overlay it will be noted that each of the 20 key labels except one have two labels, one in parentheses. By pressing that function key, the function printed without parentheses will be performed. By pressing the parenthesis key (the only one without two functions), and then pressing any function key, the function printed in parentheses will be performed.

KEYING IN NUMBERS

Each time a number key is pressed, the number appears on the screen, to the right of the last digit, if any. Note that a decimal point key is included; it must be pressed if it is part of the number. To enter the number 692.7, for example, one would press the keys 6, 9, 2, ., and 7 in that order. On the screen would appear 692.7 as the digits are pressed.

THE DISPLAY

This program, because it simulates a hand-held calculator, constantly displays the current calculator display on the screen. Each new value or result is displayed on successively lower lines. Since the screen is a storage CRT, a running record of intermediate results and subtotals is thus kept. To make these intermediate results meaningful, then, just before displaying the newest value, the name of the key used to achieve that result is also printed after the last number, if any. The program therefore is more like a printing calculator in that a running record of all transactions is kept. For example, the following sequence of keys:

CLEAR 4 5 STO 1 SIN RCL ÷ 1

will result in the following display:

```
CLEAR
0
45 STO 1
45 SIN
0.707106781187
RCL ÷ 1
0.0157134840264
```

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

Note that all but one line is indented by one space. This one line is not indented because it is an actual number being entered into the calculator, not a directive or function key. Any number that is indented is either the result of an operation or the result of reprinting the previous value. (ENTER↑ is an example of reprinting a previous value.)

OPERATIONAL STACK

Stack Registers

The hand calculator program uses an operational stack and reverse "Polish" (Lukasiewicz) notation to evaluate mathematical expressions.

The four temporary memory locations (stack registers arranged vertically) are called X (bottom register), Y, Z, and T (top register), respectively. The contents of the X register are always displayed.

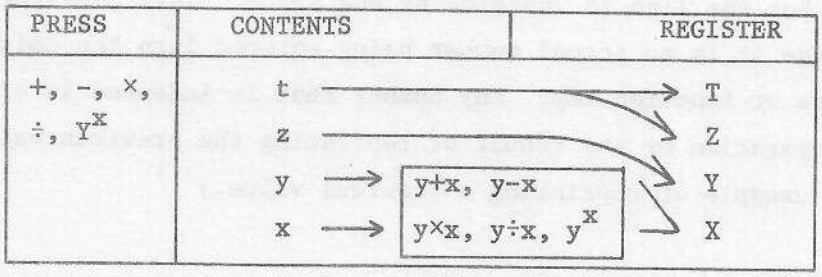
To avoid confusion between the name of a register and its contents, the register name is designated by a capital letter and the register contents by a lower case letter. Thus, register X contains the value x, which is always displayed.

When you key in a number, it goes into the X register and is displayed. When the ENTER↑ key is pressed, this number is duplicated into the Y register. At the same time, y is moved up to Z and z is moved up to T.

| PRESS | CONTENTS | REGISTER |
|---|---|---|
| ENTER↑ | t | (contents lost) |
| | z | T |
| | y | Z |
| | x | Y |
| | | X |

When + is pressed, x is added to y and is displayed in X, and the rest of the stack drops down one position. The same goes for −, ×, ÷, and $Y^X$.

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |




Observing the stack operation during a chain calculation can help intuitively understand the concepts involved. For example, in calculating (4x2)+(3-(5x2)) the following analysis will explain the necessary keystrokes. Directly above the keys pressed, the information in each of the stack registers is shown, as they are after the key is pressed.

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T | | | | | | | | 8 | 8 | 8 | 8 | 8 | |
| Z | | | | | | 8 | 8 | 3 | 3 | 8 | 8 | 8 | |
| Y | | 4 | 4 | | 8 | 3 | 3 | 5 | 5 | 3 | 8 | 8 | |
| X (Display) | 4 | 4 | 2 | 8 | 3 | 3 | 5 | 5 | 2 | 10 | -7 | 1 | |
| Press → | 4 | ↑ | 2 | × | 3 | ↑ | 5 | ↑ | 2 | x | - | + | |

Note: ↑ = Enter↑

4 is in X (Display)
X is duplicated into Y
2 is in X (Display)
4 x 2 is in X (Display)
3 is in display, automatic enter performed first
Enter↑ pushes everything up, still shows 3
5 overwrites 3 in X since it immediately follows Enter↑
Enter↑ pushes everything up, still shows 5
2 overwrites 5 in X since it immediately follows Enter↑
Result of 5x2 displayed, everything else moves down
Result of 3-(5x2) displayed, everything else moves down
Result of (4x2)+(3-(5x2)) displayed

Note that numbers were keyed into the machine in the same order they were encountered in the expression, no pre-planning is required. Note also that all intermediate results were stored in the stack where they would be ready exactly when needed. No intermediate results needed to be written down and re-entered later.

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

Manipulating the Stack

The R↓ key "rolls down" the stack and lets you review the contents (in last-in first-out order). It may be used to reposition data within the stack. The R↓ directive performs as follows:

| PRESS | CONTENTS | REGISTER |
|---|---|---|
| R↓ | t → X<br>z → Y<br>y → Z<br>x → T | |

The X↕Y key exchanges x and y as follows:

| PRESS | CONTENTS | REGISTER |
|---|---|---|
| X↕Y | t → T<br>z → Z<br>y → X<br>x → Y | |

The X↕Y function is often useful before −, ÷, and $Y^X$ operations. For example, if you are paying off a loan at 7% annual interest, you have $7000 principle left to pay, and your monthly payments are $105.65, how much of your payment goes to interest, how much towards principle, and what is the remaining principle? To do this:

| PRESS | SEE | |
|---|---|---|
| 7 → | 7 | (Interest Rate) |
| ENTER↑ → | 7 | |
| 12 → | 12 | (Divide by 12 months) |
| ÷ → | .583333333333 | |
| 100 → | 100 | For an easier way to do this, |
| ÷ → | 0.00583333333333 | see the % key. |
| 7000 → | 7000 | |
| × → | 40.8333333333 | (Amount to Interest) |

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

| PRESS (cont) | | SEE (cont) | |
|---|---|---|---|
| 105.65 | ⟶ | 105.65 | (Amount of Payment) |
| X↑↓Y | ⟶ | 40.8333333333 | |
| - | ⟶ | 64.8166666667 | (Amount to Principle) |
| 7000 | ⟶ | 7000 | (Old Principle) |
| X↑↓Y | ⟶ | 64.8166666667 | |
| - | ⟶ | 6935.18333333 | (Remaining Principle) |

Performing Combined Arithmetic Processes

The foregoing explanation of the operational stack enables the user to perform the following calculations.

Serial Calculation

Any time a new number is entered after any calculation, the calculator performs an automatic ENTER↑ on the result of the calculation. This permits a serial calculation to be performed without being encumbered by extra ENTER↑ requirements, and allows the user to operate the calculator in an easy, natural manner. Also, the user never need copy down intermediate results. For example, to solve the following equation:

$$A = 1 + 3 - 5 \div 10$$

| PRESS | | SEE |
|---|---|---|
| 1 | ⟶ | 1 |
| ENTER↑ | ⟶ | 1 |
| 3 | ⟶ | 3 |
| + | ⟶ | 4 |
| 5 | ⟶ | 5 |
| - | ⟶ | -1 |
| 10 | ⟶ | 10 |
| ÷ | ⟶ | -0.1 |

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

Chained Calculation

Chained calculations can be used to find the sums of products or the products of sums. For example, if a grocer sold 7 oranges at 18¢ each, 5 lb. of sugar at 67¢/lb., and 20 jars of jam at $1.17 per jar, the total sale price is:

$$(7\times.18) + (5\times.67) + (20 \times 1.17)$$

| PRESS | SEE |
|---|---|
| 7 ENTER↑ .18 × ⟶ | 1.26 |
| 5 ENTER↑ .67 × ⟶ | 3.35 |
| + ⟶ | 4.61 |
| 20 ENTER↑ 1.17 × ⟶ | 23.4 |
| + ⟶ | 28.01 Total Sale Price |

Note also how all the intermediate values are displayed.

Mixed Chain Calculations

Chained calculations may use any arithmetic operator. Additionally, a problem may be calculated using any combination of arithmetic operators in both nested and linked operations. For example, to calculate

$$(((8\times5)-12) + ((44\div2)\div2)) \times (13\times5\div.75) \div 2$$

| PRESS | SEE |
|---|---|
| 8 ENTER↑ 5 × ⟶ | 40 |
| 12 - ⟶ | 28 |
| 44 ENTER↑ 2 ÷ ⟶ | 22 |
| 2 ÷ ⟶ | 11 |
| + ⟶ | 39 |
| 13 ENTER↑ 5 × ⟶ | 65 |
| .75 ÷ ⟶ | 86.6666666667 |
| × ⟶ | 3380 |
| 2 ÷ ⟶ | 1690 |

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

CLEARING THE CALCULATOR

To clear the display (and zero the X register), press CLX. To clear the entire stack, including memory registers 5 through 8, press CLEAR. To restart the program (clear the stack, all the memory registers, and reset the angle mode to degrees), press the Break key and type RUN and press RETURN. More will be explained about memory registers and angle modes later. Note that CLX may be used to delete a number during keying if a keystroke is incorrect. The number may then be re-entered correctly.

Keying in Negative Numbers:

To enter a negative number, key in the number and then press CHS. The negative number will be displayed. For example, to enter -12:

| PRESS | | SEE |
|---|---|---|
| 12 | ⟶ | 12 |
| CHS | ⟶ | -12 |
| ENTER↑ | ⟶ | -12 |

or

| PRESS | | SEE |
|---|---|---|
| 1 | ⟶ | 1 |
| CHS | ⟶ | -1 |
| 2 | ⟶ | -12 |
| ENTER↑ | ⟶ | -12 |

If CHS is pressed immediately after a calculation or ENTER↑, the number in the X register is negated and redisplayed.

Keying in Exponents

To key in a number with an exponent, use the EEX key. For example, to key in $1.56 \times 10^{23}$, press:

| PRESS | | SEE |
|---|---|---|
| 1.56 | ⟶ | 1.56 |
| EEX | ⟶ | 1.56E |
| 23 | ⟶ | 1.56E23 |
| ENTER↑ | ⟶ | 1.56E+23 |

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

To enter negative exponents, press CHS any time after EEX. For example, to key in $28.45 \times 10^{-35}$, press:

| PRESS | | SEE |
|---|---|---|
| 28.45 | → | 28.45 |
| EEX | → | 28.45E |
| 35 | → | 28.45E35 |
| CHS | → | 28.45E-35 |
| ENTER↑ | → | 2.845E-34 |

or

| PRESS | | SEE |
|---|---|---|
| 28.45 | → | 28.45 |
| EEX | → | 28.45E |
| CHS | → | 28.45E- |
| 35 | → | 28.45E-35 |
| ENTER↑ | → | 2.845E-34 |

To enter a negative number with an exponent, be sure to press CHS before EEX. If you forget and if it won't disturb the stack (this is a first entry), you may press CHS after the ENTER↑. Entering $-285 \times 10^{-10}$ will illustrate both these points:

| PRESS | | SEE |
|---|---|---|
| 285 | → | 285 |
| CHS | → | -285 |
| EEX | → | -285E |
| CHS | → | -285E- |
| 10 | → | -285E-10 |
| ENTER↑ | → | -2.85E-8 |
| CHS | → | 2.85E-8 |
| CHS | → | -2.85E-8 |

If no mantissa is entered before EEX is pressed, a mantissa of 1 is assumed. If no exponent is entered after the EEX, an exponent of 0 is assumed.

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

| PRESS | SEE |
|---|---|
| CLEAR | 0 |
| EEX | 1E |
| ENTER↑ | 1 |

Entering π

π is a fixed constant available to the user. To enter the value of π, press the π key. An automatic ENTER↑ is performed before the value of π is placed in the X register and displayed.

| PRESS | SEE |
|---|---|
| 2 ENTER↑ | 2 |
| π | 3.14159265359 |
| 3 | 3 |
| × | 9.42477796077 |
| × | 18.8495559215 |

PERFORMING MONADIC FUNCTIONS

Several monadic, or single operator, functions are included in the hand calculator program, including square root, reciprocals, natural and base 10 logs and natural and base antilogs. These functions replace the contents of the X register with the value of the function of x, leaving the rest of the stack alone.

| PRESS | CONTENTS | REGISTER |
|---|---|---|
| $1/x$, $\sqrt{x}$ | t | T |
| $10^x$, LOG | z | Z |
| $e^x$, LN | y | Y |
| | x → f(x) | X (displayed) |

Reciprocals

To calculate the reciprocal of a displayed value, press 1/x:

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

| PRESS | | SEE |
|---|---|---|
| 5 | → | 5 |
| 1/x | → | 0.2 |
| 1/x | → | 5 |

Square Root

To calculate the square root of a displayed value, press x:

| PRESS | | SEE |
|---|---|---|
| 81 | → | 81 |
| $\sqrt{x}$ | → | 9 |
| $\sqrt{x}$ | → | 3 |
| $\sqrt{x}$ | → | 1.73205080757 |

Natural Logs and Antilogs:

To find the natural log of a displayed value, press LN. To find the natural antilog of a displayed value, press $e^x$.

| PRESS | | SEE |
|---|---|---|
| 2.02 | → | 2.02 |
| LN | → | 0.703097511413 |
| $e^x$ | → | 2.02 |

Base 10 Logs and Antilogs

To find the base 10 log of a displayed value, press LOG. To find the base 10 antilog of a displayed value, press $10^x$.

| PRESS | | SEE |
|---|---|---|
| 2.02 | → | 2.02 |
| LOG | → | 0.305351369447 |
| $10^x$ | → | 2.02 |

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

TRIGONOMETRIC FUNCTIONS

Three trigonometric functions are provided by the SIN, COS, and TAN keys. These are extended to six functions by first pressing the ARC key, to provide ARC SIN, ARC COS, and ARC TAN functions. The trig functions are monadic functions. For example, to find the sin, cos, and tan of 45 degrees:

| PRESS | | SEE |
|---|---|---|
| 45 | → | 45 |
| SIN | → | 0.707106781187 |
| ARC | → | ARC |
| SIN | → | 45 |
| COS | → | 0.707106781187 |
| ARC | → | ARC |
| COS | → | 45 |
| TAN | → | 1 |
| ARC | → | ARC |
| TAN | → | 45 |

When first run, the program is set in degrees mode. The program may be set in radians, gradians, or back to degrees mode by briefly stepping outside the program. To do this, press the BREAK key. The error message PROGRAM INTERRUPTED PRIOR TO LINE 300 will be printed and the cursor will appear. Then, type in the appropriate command:

| | |
|---|---|
| SET DEG | for Degrees mode |
| SET RAD | for Radians mode |
| SET GRAD | for Gradians mode |

followed by a carriage return. Then type RUN 300 followed by a carriage return. DO NOT type just RUN, or the stack and memory registers will be cleared and the program will be set in degrees mode again. When this is done, the program is where it was before the mode change, with the stack and memory registers unaffected. As an example, let us see how many radians are equivalent to 45 degrees. Also, let us express the number of radians in relation to $\pi$, as $\pi/8$ or $\pi/2$ radians. Assuming the program is in degrees mode:

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

| PRESS | | SEE |
|---|---|---|
| 45 | ⟶ | 45 |
| SIN | ⟶ | 0.707106781187 |
| BREAK | ⟶ | "PROGRAM INTERRUPTED PRIOR TO LINE 300" |

SET RAD↵*

RUN 300↵

| | | |
|---|---|---|
| ARC | ⟶ | ARC |
| SIN | ⟶ | 0.785398163397 (number of radians equivalent to $45^{o}$) |
| $\pi$ | ⟶ | 3.14159265359 |
| ÷ | ⟶ | 0.25 |
| 1/x | ⟶ | 4 |

The result is that 45 degrees is equivalent to $\pi/4$ radians.

DYADIC FUNCTIONS

Several dyadic, or two operator, functions are included in the hand calculator program. These are add, subtract, multiply, and divide, as well as exponentiation. For an explanation of the way these functions utilize the stack, see the section entitled "OPERATIONAL STACK".

Addition

Pressing the + key places the result y + x into the X register, and moves the stack up, propagating the T register.

| PRESS | CONTENTS | REGISTER |
|---|---|---|
| + | t → T<br>z → Z<br>y → Y<br>x, y+x → X | |

Subtraction

Pressing the - key places the result of y-x into the X register, and moves the rest of the stack according to dyadic operator rules.

*↵ means carriage return

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

Multiplication

Pressing the × key places the result of y×x into the X register, and moves the rest of the stack according to dyadic operator rules.

Division

Pressing the ÷ key places the result of y÷x into the X register, and moves the rest of the stack according to dyadic operator rules.

Exponentiation

Pressing the $Y^x$ key places the result of $y^x$ into the X register, moving the rest of the stack according to dyadic operator rules. For example, to find 2 raised to the 16th power:

| PRESS | SEE |
|---|---|
| 2 ⟶ | 2 |
| ENTER↑ ⟶ | 2 |
| 16 ⟶ | 16 |
| $Y^x$ ⟶ | 65536 |

When $Y^x$ is used in conjunction with the reciprocal key, n-th roots may be extracted. To continue the above example, find the 16th root of 65536:

| PRESS | SEE |
|---|---|
| | 65536 (from above) |
| 16 ⟶ | 16 |
| 1/x ⟶ | 0.0625 |
| $Y^x$ ⟶ | 2 |

As can be seen, 2 is the 16th root of 65536.

The fact that the contents of the T register is unaffected by dyadic operators implies that this last number propagates throughout any number of dyadic operations. This is correct, and sometimes useful. For example, finding the powers of 2 using the $Y^x$ key would involve each time entering 16 and then keying the next power and pressing $Y^x$. However, it can be done with the multiply key by completely filling the stack with 2's and allowing the stack to propagate them during the desired number of multiplies. To find the powers of 2, one would:

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

| PRESS | | SEE |
|---|---|---|
| 2 | → | 2 |
| ENTER↑ | → | 2 |
| ENTER↑ | → | 2 |
| ENTER↑ | → | 2 (The stack now contains all 2's) |
| × | → | 4 (or $2^2$) |
| × | → | 8 (or $2^3$) |
| × | → | 16 (or $2^4$) |
| × | → | 32 (or $2^5$) |
| × | → | 64 (or $2^6$) |
| × | → | 128 (or $2^7$) |
| × | → | 256 (or $2^8$) |

etc.

If desired, verification of the earlier result of 2 raised to the 16th power using the $Y^x$ key can be accomplished with this method.

CALCULATING PERCENTAGE PROBLEMS

The hand calculator program simplified the calculation of percentage problems because conversion of percents to their decimal equivalents is performed automatically, just press the % key after keying in the percent value. Three types of problems are handled:

- Finding percentage of number (base × rate)
- Finding net amount (base + or - percentage)
- Finding percent difference between a number and a base

$$\frac{\text{number - base}}{\text{base}}$$

Finding Percentage

To find the percentage of a number, key in the base number and press ENTER↑. Then key in the percent value and press %. For example, to find 25% of 256:

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

| PRESS | | SEE |
|---|---|---|
| 256 ENTER↑ | ⟶ | 256 base |
| 25 | ⟶ | 25 percent value |
| % | ⟶ | 64 percentage |

Finding net amount

An additional feature is that the original base number is not destroyed during a percentage calculation. From this may be calculated the net amount simply by pressing + or −, as applicable:

| PRESS | | SEE |
|---|---|---|
| 256 ENTER↑ | ⟶ | 256 base |
| 25 % | ⟶ | 64 percentage |
| + | ⟶ | 320 net amount (base + percentage) |

or

| PRESS | | SEE |
|---|---|---|
| 256 ENTER↑ | ⟶ | 256 base |
| 25 % | ⟶ | 64 percentage |
| − | ⟶ | 192 net amount (base − percentage) |

Finding percent difference between two numbers

To find the percent difference between a number and a base, key in the base number and press ENTER↑. Key in the second number and press Δ%. For example, if you wish to find the rate of increase in your rent ($200 last month and $225 now),

| PRESS | | SEE |
|---|---|---|
| 200 ENTER↑ | ⟶ | 200 base number |
| 225 | ⟶ | 225 second number |
| Δ% | ⟶ | 12.5 percent increase |

to verify, continue:

| | | |
|---|---|---|
| X↕Y | ⟶ | 200 base number |
| X↕Y | ⟶ | 12.5 percentage |
| % | ⟶ | 25 increase amount (second number − base) |
| + | ⟶ | 225 second number |

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

DATA REGISTERS

In addition to the operational stack, the hand calculator program provides 9 registers for user storage.

Unrestricted Storage

Registers $R_1$ through $R_4$ and register $R_9$ can be used for temporary storage without restriction. Values stored in these registers are not affected by calculations or by clearing operations. New values are entered by writing over the old contents; that is, by storing a new number. The contents are lost, of course, whenever the program is started (RUN).

Restricted Storage

Registers $R_5$ through $R_8$ are zeroed by the CLEAR function. This is the only difference between restricted and unrestricted storage.

Storing and Recalling Data

To store a value appearing on the display (whether the result of a calculation or a key entry), press STO followed by the number key (1 through 9) specifying the indicated storage register. To retrieve the value at any time, press RCL followed by the applicable number key. A copy of the recalled value appears on the display (X register); the storage register is unaffected. The stack is pushed up (an automatic ENTER↑ is performed) before the recall unless the last operation was a CLX or ENTER↑. These keys do not cause the stack to be pushed up by the last data entry. For example, when calculating the amount of interest and the amount from principle in a loan payment above, the original principle amount was keyed in twice. Using a register can avoid this needless delay. Assuming a principle amount of $7000, monthly payments of $105.65, and a 7% annual interest rate, find the parts of the first <u>two</u> payments credited to interest and credited to principle, and the remaining principle after these two payments.

| PRESS | | SEE | |
|---|---|---|---|
| 7 ENTER↑ 12 ÷ | → | 0.583333333333 | (monthly percent) |
| STO 1 | ——→ | 0.583333333333 | |
| 105.65 | ——→ | 105.65 | (monthly payment) |
| STO 2 | ——→ | 105.65 | |

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

| PRESS (cont) | SEE (cont) | |
|---|---|---|
| 7000 ⟶ | 7000 | (principle) |
| STO 3 ⟶ | 7000 | |
| RCL 1 ⟶ | 0.583333333333 | |
| % ⟶ | 40.8333333333 | (amount of interest in first payment) |
| RCL 2 ⟶ | 105.65 | |
| X↕Y ⟶ | 40.8333333333 | |
| − ⟶ | 64.8166666667 | (amount of first payment credited to principle) |
| RCL 3 ⟶ | 7000 | |
| X↕Y ⟶ | 64.8166666667 | |
| − ⟶ | 6935.18333333 | (amount of principle remaining after first payment) |
| STO 3 ⟶ | 6935.18333333 | (replace old principle) |
| RCL 1 % ⟶ | 40.4552361111 | (amount of second payment credited to interest) |
| RCL 2 X↕Y − ⟶ | 65.1947638889 | (amount of second payment credited to principle) |
| RCL 3 X↕Y − ⟶ | 6869.98856944 | (remaining principle after second payment) |

Note that one can loop back and perform this process repeatedly. Note also that only the values at the first were keyed in, all intermediate values are stored internally, thus reducing time required and also the chance for error.

Performing Register Arithmetic

Arithmetic operations (+, −, ×, ÷) can be performed between a data storage register and the X register (display). To modify the contents of a storage register, press STO followed by the applicable operator key (+, −, ×, ÷), then the number key specifying the storage register. For example, to store 6 in register $R_1$ and then increment it by 2:

| PRESS | SEE |
|---|---|
| 6 STO 1 ⟶ | 6 |
| 2 ⟶ | 2 |
| STO + 1 ⟶ | 2 |

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

To see what is now in $R_1$,

RCL 1 ⟶ 8

Now, to subtract 3 from the contents of $R_1$,

3 ⟶ 3
STO - 1 ⟶ 3
RCL 1 ⟶ 5

Conversely, to alter the X register (display) without affecting the data storage register or any other stack registers, press RCL, then the applicable operator (+, −, ×, ÷), followed by the number of the desired register. For example:

| PRESS | SEE |
|---|---|
| 4 STO 9 | 9 |
| CLX | 0 |
| 7 | 7 |
| RCL + 9 | 11 |
| RCL - 9 | 7 |
| RCL - 9 | 3 |
| RCL ÷ 9 | 0.75 |
| RCL 9 | 4 |

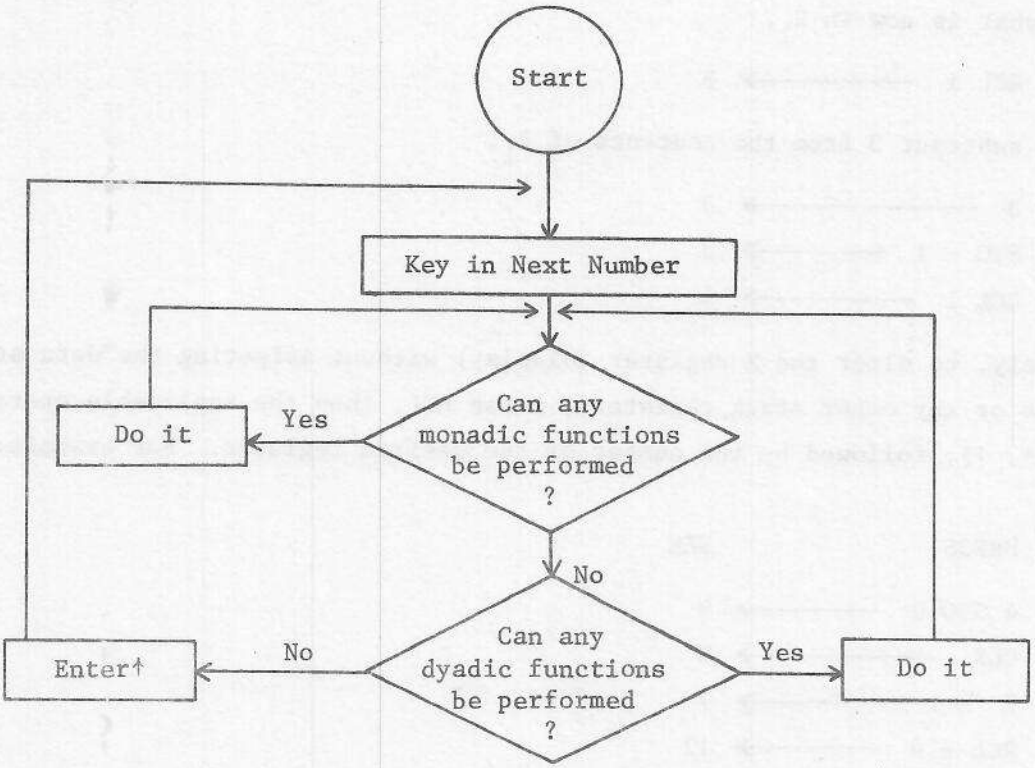
It is worth noting that a RCL ÷ 2 (for example) does exactly the same thing as a RCL 2 ÷, except that the stack is not moved up a position. This extends the number carrying capability of the stack.

APPENDIX

The flowchart which follows allows evaluation of any expression on a calculator using an operational stack and reverse "Polish" (Lukasiewicz) notation. Although the general case solution requires an operational stack of infinite length, the four register stack of the hand calculator program allows solutions of most expressions of normal complexity. Before using the algorithm, write the expression in serial form. For example $\frac{7}{32 - \frac{31}{(6)}}$ can be rewritten as

7/(32 - (31/(6))).

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |




DATA STRUCTURES

Memory Allocation to Variables

736 Bytes

Memory Allocation for Program

8371 Bytes

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

**VARIABLE MAP**

| VARIABLE | TAPE FILE | USAGE |
|---|---|---|
| F | | Used in RCL and STO operations to determine whether register arithmetic is desired and if so, what type:<br>F = 0 No register arithmetic<br>F = 1 +<br>F = 2 -<br>F = 3 ×<br>F = 4 ÷ |
| G | | Used to determine mode of operation.<br>G = 1 Regular Function<br>G = 2 Alternate (Colored) Function<br>G = 3 STO Operation<br>G = 4 RCL Operation<br>G = 5 ARC Operation |
| I1 | | Work Variable |
| K | | On some keys, used to store key number struck. |
| M(9) | | The nine storage registers M(1) is register 1, etc. |
| S(4) | | The four-place Stack. S(1) = X register, S(2) = Y register, S(3) = Z register, S(4) = T register. |
| B$ | | Work String (72) |
| C$ | | Work String (72) |
| D$ | | Work String (72) |
| V$ | | Contains Backspace Character (1) |
| X$ | | Contains Current Number Entry String (72) |
| Z$ | | Contains current function name (7) |

| TITLE | ABSTRACT NUMBER |
|---|---|
| HAND CALCULATOR | 51/00-6001/0 |

**USER PROGRAMMABLE KEYS**

| KEY | LABEL | TAPE FILE | FUNCTION |
|---|---|---|---|
| 1 | 1($Y^x$) | | The digit 1 and the function $Y^x$. |
| 2 | 2($\sqrt{x}$) | | The digit 2 and the function $\sqrt{}$. |
| 3 | 3($10^x$) | | The digit 3 and the base 10 antilog function. |
| 4 | 4(Δ%) | | The digit 4 and the function "change in percent". |
| 5 | 5(%) | | The digit 5 and the percent function. |
| 6 | 6(1/x) | | The digit 6 and the reciprocal function. |
| 7 | 7(ARC) | | The digit 7 and the ARC trig function shift. |
| 8 | 8(SIN) | | The digit 8 and the SIN function. |
| 9 | 9(COS) | | The digit 9 and the COS function. |
| 10 | 0(TAN) | | The digit 0 and the TAN function. |
| 11 | .(LOG) | | The decimal point and the Base 10 Logarithm function. |
| 12 | CHS(LN) | | The "change sign" directive and the natural logarithm function. |
| 13 | EEX($e^x$) | | The "enter exponent" directive and the natural antilog function. |
| 14 | CLX(CLEAR) | | The clear x function and the clear calculator directive. |
| 15 | ( ) | | The color shift (alternate function) key. |
| 16 | +(X↑↓Y) | | The Add function and the X-Y swap directive. |
| 17 | -(R↓) | | The Subtract function and the roll stack directive. |
| 18 | ×(STO) | | The Multiply function and the store register shift key. |
| 19 | ÷(RCL) | | The Divide function and the recall register shift key. |
| 20 | ENTER↑(π) | | The Enter directive and the π constant. |

| TITLE | ABSTRACT NUMBER |
| --- | --- |
| HAND CALCULATOR | 51/00-6001/0 |

```
 1 GO TO 100
 4 K=1
 5 GO TO G OF 520,1890,2860,3000,520
 8 K=2
 9 GO TO G OF 520,1950,2860,3000,520
 12 K=3
 13 GO TO G OF 520,1850,2860,3000,520
 16 K=4
 17 GO TO G OF 520,2000,2860,3000,520
 20 K=5
 21 GO TO G OF 520,2210,2860,3000,520
 24 K=6
 25 GO TO G OF 520,2250,2860,3000,520
 28 K=7
 29 GO TO G OF 520,2390,2860,3000,2390
 32 K=8
 33 GO TO G OF 520,2300,2860,3000,2340
 36 K=9
 37 GO TO G OF 520,2430,2860,3000,2470
 40 K=0
 41 GO TO G OF 520,2520,520,520,2560
 44 GO TO G OF 600,1580,600,600,600
 48 GO TO G OF 890,1730,890,890,890
 52 GO TO G OF 760,1810,760,760,760
 56 GO TO G OF 1300,1210,1300,1300,1300
 60 IF G=5 THEN 280
 61 G=2
 62 GO TO 280
 64 K=16
 65 GO TO G OF 1350,2050,2730,2730,1350
 68 K=17
 69 GO TO G OF 1400,2140,2730,2730,1400
 72 K=18
 73 GO TO G OF 1450,2600,2730,2730,1450
 76 K=19
 77 GO TO G OF 1500,2660,2730,2730,1500
 80 GO TO G OF 420,470,420,420,420
 100 INIT
 110 SET KEY
 120 SET DEGREE
 130 PAGE
 140 DIM S(4),M(9),V$(1),Z$(7)
 150 V$=CHR(8)
 160 Z$=""
 170 S=0
```

| TITLE | ABSTRACT NUMBER |
| --- | --- |
| HAND CALCULATOR | 51/00-6001/0 |

```
180 M=0
190 X$=""
200 G=1
210 PRINT " CLEAR"
220 PRINT S(1)
230 GO TO 300
240 X$=""
250 G=1
260 PRINT " ";Z$
270 PRINT S(1)
280 Z$=""
290 RETURN
300 GO TO 300

310 REM PUSH SUBROUTINE ----------

320 S(4)=S(3)
330 S(3)=S(2)
340 S(2)=S(1)
350 RETURN

360 REM POP SUBROUTINE ----------

370 S(1)=S(2)
380 S(2)=S(3)
390 S(3)=S(4)
400 RETURN

410 REM ENTER↑ SECTION ----------

420 GOSUB 320
430 X$=" "
440 Z$="ENTER↑"
450 GO TO 250

460 REM PI SECTION ----------

470 GOSUB 320
480 S(1)=PI
490 Z$="PI"
500 GO TO 240

510 REM NUMBER SECTION ----------

520 IF LEN(X$)>0 THEN 540
```

| TITLE | ABSTRACT NUMBER |
| --- | --- |
| HAND CALCULATOR | 51/00-6001/0 |

```
530 GOSUB 320
540 B$=CHR(48+K)
550 X$=X$&B$
560 S(1)=VAL(X$)
570 PRINT B$;
580 GO TO 280

590 REM . SECTION ------------------

600 I1=POS(X$,".",1)
610 IF I1=0 THEN 630
620 GO TO 730
630 I1=POS(X$,"E",1)
640 IF I1<>0 THEN 730
650 IF LEN(X$)>0 THEN 700
660 GOSUB 320
670 X$="."
680 S(1)=0
690 GO TO 720
700 X$=X$&"."
710 S(1)=VAL(X$)
720 PRINT ".";
730 G=1
740 GO TO 280

750 REM EEX SECTION ------------------

760 I1=POS(X$,"E",1)
770 IF I1<>0 THEN 860
780 IF LEN(X$)>0 AND S(1)<>0 THEN 830
790 X$=X$&"1E0"
800 PRINT "1E";
810 S(1)=1
820 GO TO 860
830 IF X$=" " THEN 790
840 X$=X$&"E"
850 PRINT "E";
860 G=1
870 GO TO 280

880 REM CHS SECTION ------------------

890 Z$="CHS"
900 IF LEN(X$)=0 THEN 920
910 IF X$<>" " THEN 940
```

| TITLE | ABSTRACT NUMBER |
| --- | --- |
| HAND CALCULATOR | 51/00-6001/0 |

```
920 S(1)=S(1)*-1
930 GO TO 250
940 I1=POS(X$,"E",1)
950 IF I1=0 THEN 1050
960 C$=SEG(X$,1,I1)
970 D$=SEG(X$,I1+1,LEN(X$)-I1)
980 B$=SEG(D$,1,1)
990 IF B$="-" THEN 1020
1000 D$="-"&D$
1010 GO TO 1030
1020 D$=SEG(D$,2,LEN(D$)-1)
1030 X$=C$&D$
1040 GO TO 1100
1050 B$=SEG(X$,1,1)
1060 IF B$="-" THEN 1090
1070 X$="-"&X$
1080 GO TO 1100
1090 X$=SEG(X$,2,LEN(X$)-1)
1100 IF X$="." THEN 1180
1110 IF X$="-." THEN 1180
1120 IF X$="-" THEN 1180
1130 S(1)=VAL(X$)
1140 PRINT " ";Z$
1150 PRINT " ";X$;
1160 G=1
1170 GO TO 280
1180 S(1)=0
1190 GO TO 1140

1200 REM CLEAR SECTION ----------

1210 S=0
1220 M(5)=0
1230 M(6)=0
1240 M(7)=0
1250 M(8)=0
1260 PAGE
1270 Z$="CLEAR"
1280 GO TO 240

1290 REM CLX SECTION ----------

1300 S(1)=0
1310 X$=" "
1320 Z$="CLX"
```

| TITLE | ABSTRACT NUMBER |
| --- | --- |
| HAND CALCULATOR | 51/00-6001/0 |

```
1330 GO TO 250

1340 REM + SECTION ------------------

1350 S(1)=S(1)+S(2)
1360 GOSUB 380
1370 Z$="+"
1380 GO TO 240

1390 REM - SECTION ------------------

1400 S(1)=S(2)-S(1)
1410 GOSUB 380
1420 Z$="-"
1430 GO TO 240

1440 REM x SECTION ------------------

1450 S(1)=S(1)*S(2)
1460 GOSUB 380
1470 Z$="x"
1480 GO TO 240

1490 REM / SECTION ------------------

1500 Z$=V$
1510 Z$=Z$&":"
1520 Z$="-"&Z$
1530 IF S(1)=0 THEN 1650
1540 S(1)=S(2)/S(1)
1550 GOSUB 380
1560 GO TO 240

1570 REM LOG SECTION ----------------

1580 Z$="LOG"
1590 IF S(1)<=0 THEN 1660
1600 IF S(1)=1 THEN 1630
1610 S(1)=LGT(S(1))
1620 GO TO 240
1630 S(1)=0
1640 GO TO 240

1650 REM ERROR ROUTINE --------------
```

| TITLE | ABSTRACT NUMBER |
| --- | --- |
| HAND CALCULATOR | 51/00-6001/0 |

```
1670 X$=""
1680 PRINT " ";Z$
1690 PRINT " ERROR! --- ";0
1700 S(1)=0
1710 GO TO 280

1720 REM LN ROUTINE ----------------

1730 Z$="LN"
1740 IF S(1)<=0 THEN 1660
1750 IF S(1)=1 THEN 1780
1760 S(1)=LOG(S(1))
1770 GO TO 240
1780 S(1)=0
1790 GO TO 240

1800 REM e↑X SECTION ----------------

1810 Z$="e↑X"
1820 S(1)=2.71828182846↑S(1)
1830 GO TO 240

1840 REM 10↑X SECTION ----------------

1850 Z$="10↑X"
1860 S(1)=10↑S(1)
1870 GO TO 240

1880 REM Y↑X SECTION ----------------

1890 Z$="Y↑X"
1900 IF S(2)<=0 THEN 1660
1910 S(1)=S(2)↑S(1)
1920 GOSUB 380
1930 GO TO 240

1940 REM SQRT SECTION ----------------

1950 Z$="SQRT"
1960 IF S(1)<0 THEN 1660
1970 S(1)=S(1)↑0.5
1980 GO TO 240

1990 REM DELTA % SECTION ----------
```

| TITLE | ABSTRACT NUMBER |
| --- | --- |
| HAND CALCULATOR | 51/00-6001/0 |

```
2000 Z$="DELTA %"
2010 IF S(2)=0 THEN 1660
2020 S(1)=(S(1)-S(2))/S(2)*100
2030 GO TO 240

2040 REM X<>Y SECTION ----------------

2050 Z$=CHR(127)
2060 Z$=Z$&V$
2070 Z$=Z$&"↑Y"
2080 Z$="X"&Z$
2090 I1=S(2)
2100 S(2)=S(1)
2110 S(1)=I1
2120 GO TO 240

2130 REM ROLL STACK SECTION -----------

2140 Z$=CHR(127)
2150 Z$="R"&Z$
2160 I1=S(1)
2170 GOSUB 370
2180 S(4)=I1
2190 GO TO 240

2200 REM % SECTION --------------------

2210 Z$="%"
2220 S(1)=S(1)*S(2)/100
2230 GO TO 240

2240 REM 1/X SECTION ----------------

2250 Z$="1/X"
2260 IF S(1)=0 THEN 1660
2270 S(1)=1/S(1)
2280 GO TO 240

2290 REM SIN SECTION ----------------

2300 Z$="SIN"
2310 S(1)=SIN(S(1))
2320 GO TO 240
```

| TITLE | ABSTRACT NUMBER |
| --- | --- |
| HAND CALCULATOR | 51/00-6001/0 |

```
2340 Z$="SIN"
2350 IF S(1)>1 THEN 1660
2360 S(1)=ASN(S(1))
2370 GO TO 240

2380 REM ARC SECTION ----------------

2390 G=5
2400 PRINT " ARC";
2410 GO TO 280

2420 REM COS SECTION ----------------

2430 Z$="COS"
2440 S(1)=COS(S(1))
2450 GO TO 240

2460 REM ARC COS SECTION ------------

2470 Z$="COS"
2480 IF S(1)>1 THEN 1660
2490 S(1)=ACS(S(1))
2500 GO TO 240

2510 REM TAN SECTION ----------------

2520 Z$="TAN"
2530 S(1)=TAN(S(1))
2540 GO TO 240

2550 REM ARC TAN SECTION ------------

2560 Z$="TAN"
2570 S(1)=ATN(S(1))
2580 GO TO 240

2590 REM STO SECTION ----------------

2600 G=3
2610 F=1
2620 X$=""
2630 PRINT " STO";
2640 GO TO 280
```

| TITLE | ABSTRACT NUMBER |
| --- | --- |
| HAND CALCULATOR | 51/00-6001/0 |

```
2650 REM RCL SECTION ----------------

2660 G=4
2670 F=1
2680 IF X$=" " THEN 2700
2690 X$=""
2700 PRINT " RCL";
2710 GO TO 280

2720 REM STO,RCL FUNCTION SECTION -----

2730 F=K-14
2740 IF X$=" " THEN 2760
2750 X$=""
2760 GO TO F OF 280,2770,2790,2810,2830
2770 PRINT " +";
2780 GO TO 280
2790 PRINT " -";
2800 GO TO 280
2810 PRINT " x";
2820 GO TO 280
2830 PRINT " -";V$;":";
2840 GO TO 280

2850 REM STO REG SECTION -------------

2860 Z$=CHR(K+48)
2870 GO TO F OF 2880,2900,2920,2940,2960
2880 M(K)=S(1)
2890 GO TO 240
2900 M(K)=M(K)+S(1)
2910 GO TO 240
2920 M(K)=M(K)-S(1)
2930 GO TO 240
2940 M(K)=M(K)*S(1)
2950 GO TO 240
2960 IF S(1)=0 THEN 1660
2970 M(K)=M(K)/S(1)
2980 GO TO 240

2990 REM RCL REG SECTION -------------

3000 Z$=CHR(K+48)
3010 GO TO F OF 3020,3060,3080,3100,3120
```

| TITLE | ABSTRACT NUMBER |
| --- | --- |
| HAND CALCULATOR | 51/00-6001/0 |

```
3020 IF X$=" " THEN 3040
3030 GOSUB 320
3040 S(1)=M(K)
3050 GO TO 240
3060 S(1)=S(1)+M(K)
3070 GO TO 240
3080 S(1)=S(1)-M(K)
3090 GO TO 240
3100 S(1)=S(1)*M(K)
3110 GO TO 240
3120 IF M(K)=0 THEN 1660
3130 S(1)=S(1)/M(K)
3140 GO TO 240
```