A calculator has one or more keys which initiate the performance of user-defined subroutines. When the calculator is turned on, the calculator memory is programmed with a predetermined subroutine which may subsequently be changed or replaced by the user.

6 Claims, 2 Drawing Figures
POWER ON

CLEAR STACK

ESTABLISH KEY CODE IN CENTRAL PROCESSING UNIT

INSERT KEY CODE INTO MEMORY

NEXT KEY CODE?

YES

NO

RETURN MEMORY POINTER TO TOP OF MEMORY

READY

FIGURE 2
CALCULATOR HAVING PREPROGRAMMED USER-DEFINABLE FUNCTIONS

BACKGROUND AND SUMMARY OF THE INVENTION

When an electronic calculator is first turned on and power is supplied to its memory and computational circuits, it is usually necessary to clear random bits out of the memory before the calculator is ready to use. Thus, in prior art programmable calculators, a power on or turn-on program is usually provided to clear all memory registers and insure that the calculator is ready for the first computation. Some programmable calculators have one or more keys on the keyboard which, when depressed, will initiate the performance of a subroutine which the user may define in the calculator memory. These user-definable keys provide a time saving feature for users who have frequently used subroutines comprising a number of calculator functions that require the sequential operation of a number of keys to perform. For example, the calculator keyboard may provide keys enabling the user to perform simple mathematical functions such as add, subtract, multiply, divide, square root, and so forth. A user may desire to perform a more complex function on a repeated basis such as the variance or standard deviation of data he is analyzing. User-definable keys allow the user to program these subroutines into memory and perform the desired operation on entered data with the depression of a single key. In the past it was necessary to redefine the functions associated with user-definable keys each time the calculator was turned on, since the turn-on program would clear the memory associated with the user-definable keys along with other parts of the calculator memory.

According to the preferred embodiment of the present invention, a calculator having user-definable keys is provided with a routine in its turn-on program which predetermines the user-definable keys with commonly used functions. Thus, these keys are immediately usable by the user after the calculator is turned on, without having to program the memory associated with the keys. However, if the user desires to have other functions associated with the user-definable keys he may reprogram the keys with as much ease as user-definable keys were programmed in prior art calculators. The user-definable keys may be predefined with functions normally accessible from the keyboard or with functions that are stored in a read-only memory or other hardware in the calculator, but are not normally accessible from the keyboard.

DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a block diagram of an electronic calculator according to the preferred embodiment of the present invention.

FIG. 2 shows a flow diagram for preprogramming user-definable keys in the calculator shown in FIG. 1.

DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 illustrates the block diagram of a programmable electronic calculator 10 with a keyboard 12 having a plurality of keys for initiating the performance of various computational and manipulative operations. The keyboard is connected to keyboard encoder 14 which generates various key codes in response to the depression of the various keys on the keyboard. For example, a two-digit octal number may be associated with each key on the keyboard and the encoder will generate a six digit binary number in response to the depression of each key. The keyboard and the keyboard encoder may be implemented, for example, with an array of single pole, single throw switches, each switch corresponding to a key and a common diode encoding matrix connected to the keys. Keyboard encoder 14 is connected to a read-only memory 16 and a central processing unit (CPU) 18. ROM 16 contains instructions in binary form corresponding to the various key codes generated from the keyboard 12. CPU 18 responds to the instructions from ROM 16 to perform various mathematical calculations and manipulative operations on data entered from the keyboard. The operation of such ROM and CPU along with a keyboard and other calculator circuits is described in detail in a patent application by Francé Rode, et al. entitled "Improved Business Calculator" Ser. No. 302,371, filed Oct. 30, 1972, now U.S. Pat. No. 3,863,060, and also in the Hewlett-Packard Journal June 1, 1972, pages 1-9.

Calculator 10 also includes a read-write memory 20 which is connected to keyboard encoder 14 and central processing unit 18. Read-write memory 20 may be used for storing information entered from the keyboard or from auxiliary devices such as a magnetic card reader. The key codes stored in read-write memory 20 are data and various mathematical manipulative operations stored in coded form for later use as a program or subroutine for performing calculations with data entered from the keyboard. In order to store a program in the read-write memory the calculator must be preconditioned to store rather than execute the key codes as they are generated. Thus, a Program/run switch 24 is connected to CPU 18 to select the mode of operation of the calculator.

In the Run mode, a key code from keyboard encoder 14 is directed by CPU 18 to establish an address in ROM 16 which, in turn, causes the CPU to execute the instructions stored in ROM 16 at that address. When the calculator is in the Program mode, the key codes are stored in memory 20 for later use instead of being executed. Memory 20 may be any one of several common types of read-write memories such as a delay line, a shift register or an array of magnetic cores.

User-definable keys A through E on the keyboard 12 may be used for initiating subroutines stored in read-write memory 20. As mentioned before, the subroutines may be entered from the keyboard and this will be illustrated with a program for computing the ratio of two numbers that have been entered from the keyboard and giving the answer in percent. After the calculator has been placed in the Program mode by moving switch 24 to the left, the following keys would be depressed:

<table>
<thead>
<tr>
<th>LBL</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>X</td>
<td>RTN</td>
</tr>
</tbody>
</table>

The first key depressed, LBL, is a memory instruction that denotes the beginning of the user-definable function that will correspond to the user-definable key depressed, here key A. When the calculator is later instructed to find user-definable function A, it will look.
for the notation LBL A in memory and then execute the following program steps. Data from the keyboard of the calculator is entered into an operational stack and the data in the bottom two positions will be divided in the next program step. Following that, new data, the number 100, will be entered and the result of the previous division will be pushed up to the next position in the stack. Next the numbers in the two bottom positions of the stack will be multiplied yielding the answer in percent. The last instruction, RTN, tells the calculator that the subroutine is finished and that the calculator should return to the manual operational mode. The calculator will perform this subroutine when it is placed in the Run mode by moving switch 24 to the right and key A is depressed.

In the preferred embodiment, the calculator uses a dynamic memory implemented in shift registers. Each piece of information in such a memory is located relative to other pieces of information but has no absolute address as it would in a core type memory. A memory pointer is used as an index to indicate that word in memory that is currently accessible. When information is read from memory, the memory will start the read-out at the memory pointer and then move to each successive memory word until told to stop, at which time the memory pointer will then point to the word where read out stopped. The memory pointer can also be commanded to move to any particular word in memory or to find a particular word in memory identified by a particular LBL.

When the calculator is in the Run mode and key A is depressed, the memory pointer will first be moved to the memory position immediately after LBL A. Next, the calculator will execute all instructions until it encounters a RTN instruction, when it will stop and display the results. The memory pointer will then remain at the RTN instruction until further instruction is given. A more detailed disclosure of the operation of user-definable keys along with an exemplary implementing structure may be found in U.S. Patent application by Robert E. Watson, entitled "Improved Programmable Calculator." Ser. No. 153,437 filed June 15, 1971, now U.S. Pat. No. 3,859,635.

As mentioned previously, prior art calculators having user-definable keys have no usable functions associated with these keys after the machine is turned on until the keys are programmed by the user, either from the keyboard or by the use of some auxiliary programming device such as a magnetic card. Since the calculator may have more than one user-definable key and the user may desire only one or two programs of his own, some of the keys may thus remain unprogrammed and unused. It is thus highly advantageous to have the user-definable function keys programmed with commonly used functions when the calculator is turned on, so that the user may use these if he does not desire to reprogram the calculator with his own subroutines. It may be desirable for example, to preprogram user-definable keys with functions that are accessible from the keyboard but that require more than a single key stroke to initiate.

As illustrated in FIG. 1, in some calculators the user must actuate a shift key before certain functions can be initiated such as 1/X, X^2, Y^2 and so forth. These functions are associated with the keys that are normally used for entering numbers; thus, when the 7 key is depressed the 7 is enterred into memory. However, when the shift key is depressed and then the 7 key is depressed the calculator will take the inverse of the number in the bottom register of the memory stack. In the illustrated embodiment the functions 1/X, X^2, Y^2, roll stack door (R_), and exchange X and Y (X=Y) are shown associated with the user-definable function keys A through E, so that those functions, which normally require the use of the shift key, may be initiated with a single key stroke. When the calculator is turned on, the memory is automatically programmed to associate those functions with keys A through E; but the user has the option of reprogramming the memory so that those keys will initiate whatever subroutine the user wishes.

The memory can be preprogrammed with the aforementioned functions using a program in the turn-on sequence illustrated by the flow diagram in FIG. 2. This subroutine is programmed into a portion of ROM 16 in the form of the sequence of steps or key codes to be programmed into memory 20. The programmed sequence for the illustrated embodiment is as follows:

<table>
<thead>
<tr>
<th>Table I</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBL</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>Shift</td>
</tr>
<tr>
<td>7 (1/X)</td>
</tr>
<tr>
<td>RTN</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>Shift</td>
</tr>
<tr>
<td>8 (V)</td>
</tr>
<tr>
<td>RTN</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>Shift</td>
</tr>
<tr>
<td>9 (a)</td>
</tr>
<tr>
<td>RTN</td>
</tr>
<tr>
<td>D</td>
</tr>
<tr>
<td>Shift</td>
</tr>
<tr>
<td>4((−1))</td>
</tr>
<tr>
<td>RTN</td>
</tr>
<tr>
<td>E</td>
</tr>
<tr>
<td>Shift</td>
</tr>
<tr>
<td>5(X=Y)</td>
</tr>
<tr>
<td>RTN</td>
</tr>
</tbody>
</table>

This same turn on programming sequence could also be used to program one or more of keys A through E with functions that are not ordinarily accessible from the keyboard such as cube root, for example, which does not appear on the illustrated keyboard. Because memory 20 is not divided up into discrete areas corresponding to each of the keys, a function corresponding to a user-definable key may be as short as one step or as long as the whole memory. It will be understood, of course, that another type of memory could be used in which each key A through E corresponded to a particular fixed address or series of addresses in memory.

The foregoing programming sequence and the flow chart in FIG. 2 can be implemented in a ROM using the bit pattern and programming steps shown in Table II, below. This ROM program writes the same set of subroutines into memory 20 that would be written if the sequence shown in Table I were entered from the keyboard. Thus the information stored in the key codes that establish addresses in other ROM's such as those disclosed in the aforementioned U.S. Pat. No. 3,863,060 for performing mathematical functions when the user-definable keys are depressed.

<table>
<thead>
<tr>
<th>Table II</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>L12010:</td>
</tr>
<tr>
<td>1111111</td>
</tr>
<tr>
<td>L2037</td>
</tr>
<tr>
<td>G</td>
</tr>
<tr>
<td>→ GOTO</td>
</tr>
<tr>
<td>DFLTO</td>
</tr>
</tbody>
</table>
As an alternative to the above-disclosed method, memory 20 could also be preprogrammed by constructing a read-write memory with a read-only memory 50 within it or by configuring the memory elements so that they turn on in a preferential state, i.e. 1 or 0. One type of memory that could be suitable for such a use is suggested in an IBM Technical Disclosure Bulletin, Volume 14, No. 9, February 1972, pages 2601–2.

We claim:

1. A calculator having electronic circuits comprising:
a keyboard having a plurality of keys including a user-definable function key;
encoder means connected to the keyboard for generating a plurality of coded signals representing manipulative functions and alphanumeric characters including those functions and characters corresponding to the keys on the keyboard;
a read-write memory connected to the encoder means for storing coded signals;
computing means connected to the encoder means and the read-write memory for processing coded signals stored in the read-write memory in response to a predetermined coded signal corresponding to the user-definable function key;

2. A calculator as in claim 1 wherein the program of coded signals is stored in the read-write memory.

3. A calculator as in claim 2 wherein the program of coded signals represents a numerical constant.
4. A calculator as in claim 2 wherein:
the keyboard includes a manipulative function key
having a corresponding first coded signal for caus-
ing the computing means to perform a first manipu-
larative function;
the keyboard includes a shift key having a corre-
sponding second coded signal for causing the com-
puting means to perform a second manipulative
function in response to receipt of the first coded
signal subsequent to receipt of the second coded
signal by the computing means; and
the program of coded signals includes the second and
then the first coded signals.

5. A calculator as in claim 1 further comprising
switch means connected to the computing means for
setting the calculator in a program mode in a first posi-
tion and a run mode in second position, the first position
enabling establishment of another program of one or
more coded signals in the read-write memory in re-
response to actuation of keys on the keyboard, and the
second position enabling processing of coded signals by
the computing means in response to actuation of keys
on the keyboard.

6. A calculator as in claim 5 wherein the preprogram
means includes a read-only memory containing the
program of one or more coded signals for transferring
the coded signals to and establishing the coded signals
in the read-write memory in response to the application
of power to the electronic circuits, independent of the
position of the switch means.
UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 4,037,092
DATED : 7/19/77
INVENTOR(S) : Thomas E. Osborne, Richard Kent Stockwell

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

Column 3, line 49, "calculator" should read --calculator--;
line 63, "X 1/2" should read --X 1/2--;
Column 4, line 5, "X 1/2" should read --X 1/2--;
line 5, "door" should read --down--.

Signed and Sealed this Twenty-second Day of November 1977

[SEAL]

Attest:

RUTH C. MASON
Attesting Officer

LUTRELLE F. PARKER
Acting Commissioner of Patents and Trademarks