1 455 160

PATENT SPECIFICATION

(11) 1 455 160

(21) Application No. 48228/74 (22) Filed 7 Nov. 1974
(31) Convention Application No. 419 153
(32) Filed 26 Nov. 1973 in
(33) United States of America (US)
(44) Complete Specification published 10 Nov. 1976
(51) INT CL² G06F 15/02
(52) Index at acceptance
G4A 10G 16G 17X 5A 5B 6M1 8C KB

(19)



(54) CALCULATOR

(71) We, HEWLETT-PACKARD COMPANY, of 1501 Page Mill Road, Palo Alto, California 94304, United States of America, a corporation organized and existing under the laws of the State of California, United States of America, do hereby declare the invention, for which we pray that a patent may be granted to us, and the method by which it is to be performed, to be particularly described in and by the following statement:—

This invention is concerned with improvements in or relating to calculators.

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 for 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.

The present invention provides a calculator having electronic circuits, comprising: a keyboard having a plurality of keys including a user-definable function key; an encoder 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 for storing coded signals; computing means connected to the encoder 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; means whereby electrical power applying means can be connected to the electronic circuits including the encoder, the read-write memory and the computing means; and programming means connected to the read-write memory and to said electrical power applying means for establishing, in response to the application of power to the electronic circuits, a program of one or more coded signals in the read-write memory that can be processed in the computing means in response to the generation of the predetermined coded signal.

In a calculator as set forth in the last preceding paragraph preferably the program of coded signals stored in the read-write memory may be replaced with other coded signals by actuation of keys on the keyboard.

In a calculator as set forth in the last preceding paragraph preferably the program of coded signals represents a numerical constant.

In a calculator as set forth in either one of the two immediately preceding paragraphs preferably the keyboard includes a manuipulative function key having a corresponding first coded signal for causing the computing means to perform a first manipulative function; the keyboard includes a prefix key having a corresponding second coded signal for causing the computing means to perform a second manipulative function in response to receipt to 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.

A calculator as set forth in any one of the four immediately preceding paragraphs preferably further comprises a switch connected to the computing means for setting the calculator in a program mode in a first position and a run mode in a second position, the first position enabling establishment of another program of one or more coded signals in the read-write memory in 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.

In a calculator as set forth in the last preceding paragraph preferably the preprogramming 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.

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 predefines 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 must 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.

There now follows a detailed description which is to be read with reference to the accompanying drawings of a calculator according to the invention; it is to be clearly understood that this calculator has been selected for description to illustrate the invention by way of example and not by way of limitation.

In the drawings:—

Figure 1 shows a block diagram of an electronic calculator according to the preferred embodiment of the present invention; and

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

Figure 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 a 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. The keyboard encoder 14 is connected to a read-only memory (ROM) 16 and a central processing unit (CPU) 18. The ROM 16 contains instructions in binary form corresponding to the various key codes generated from the keyboard encoder 14. The CPU 18 responds to the instructions from the ROM 16 to perform various mathematical calculations and manipulative operations on data entered from the keyboard. The operation of such a ROM and CPU along with a keyboard and other calculator circuits is described in detail in the complete specification of our corresponding United Kingdom patent application (Serial No. 1,454,266) No. 50372/73, and also in the *Hewlett-Packard Journal* June 1, 1972, pages 1—9.

The calculator 10 also includes a read-write memory 20 which is connected to the keyboard encoder 14 and the CPU 18. The 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 the 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 the CPU 18 to select the mode of operation of the calculator.

In the Run mode, a key code from the keyboard encoder 14 is directed by the CPU 18 to establish an address in the ROM 16 which, in turn, causes the CPU to execute the instructions stored in the ROM 16 at that address. When the calculator is in the Program mode, the key codes are stored in the memory 20 for later use instead to being executed. The 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 to E on the keyboard 12 may be used for initiating subroutines stored on the 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 the switch 24 to the left, the following keys are depressed:

LBL

A

÷

1

0

0

X

RTN

The first key depressed, LBL, corresponds to a memory instruction that denotes the beginning of the user-definable function that will correspond to the user-definable key depressed, here the key A. When the calculator is later instructed to find the 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 the switch to the right and the 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 has stopped. The memory pointer can also be commanded to move to any particular word in memory or to find particular word in memory identified by a particular LBL.

When the calculator is in the Run mode and the 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 the complete specification of our United Kingdom patent No. 1,403,710.

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 Figure 1, in some calculators the user must actuate a shift key before certain functions can be initiated such as 1/X, $X^{1/2}$, Y^x 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 entered 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, $X^{1/2}$, Y^x, roll stack down (R↓) and exchange X and Y (X⇄Y) are shown associated with the user-definable function keys A to 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 to 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 programmed with the aforementioned functions using a program in the turn-on sequence illustrated by the flow diagram in Figure 2. This subroutine is programmed into a portion of the ROM 16 in the form of the sequence of steps or key codes to be programmed into the memory 20. The programmed sequence for the illustrated embodiment is as follows:

TABLE I.

LBL
A
Shift
7 (1/X)
RTN
LBL
B
Shift
8 ($\sqrt{x}$)
RTN
LBL
C
Shift
9 (Y^x)
RTN

TABLE I (Continued).

LBL
D
Shift
4 (R↓)
RTN
LBL
E
Shift
5 (X≤Y)
RTN

This same turn-on programming sequence could also be used to program one or more of the keys A to 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 the memory 20 does not divide 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 to E correspond to a particular fixed address or series of addresses in memory.

The foregoing programming sequence and the flow chart in Figure 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 same 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 may be such as those disclosed in the aforementioned patent application No. 50372/73 (Serial No. 1,454,266) for preforming mathematical functions when the user-definable keys are depressed.

TABLE II.

0	L12010:	...1111111→L2037	G	: GO TO DFLT0
1	L12011:	1.1.....	RSETP	: SEARCH FOR LABEL
2	L12012:	..1....111→L2041		GO TO DFLT3
3	L12015:	...1111111→L2037	RDN	: GO TO DFLT0
4	L12016:	...1111111→L2037	F	: GO TO DFLT0
5	L12021:	...1111111→L2037	EXCG	: GO TO DFLT0
6	L12022:	.....		NO OPERATION
7	L12023:	...1111111→L2037	YTX	: GO TO DFLT0
8	L12024:	...1111111→L2037	INX	: GO TO DFLT0
9	L12030:	...1111111→L2037	E	: GO TO DFLT0
10	L12032:	...1111111→L2037	D	: GO TO DFLT0

TABLE II (Continued).

11	L12033:	...1111111→L2037	C	:	GO TO DFLTØ
12	L12034:	...1111111→L2037	B	:	GO TO DFLTØ
13	L12036:	...1111111→L2037	A	:	GO TO DFLTØ
14	L12037:	..1.....	DFLTØ	:	MEMORY INSERT
15	L12040:	111.1....			1→F7
16	L12041:	1.111..1..	DFLTS	:	Ø→S11
17	L12042:	1.111.....	DFLT1	:	Ø→F5
18	L12043:	1.11.1.1..			IF S11≠1
19	L12044:	..1...1.11→L2042			THEN GO TO DFLT1
20	L12045:	1.111....			Ø→F5
21	L12046:	....11...			RETURN
22	L12052:	...1111111→L2037	RTN	:	GO TO DFLTØ
23	L12053:	...1111111→L2037	LBL	:	GO TO DFLTØ
24	L12062:	...1111111→L2037	SQT	:	GO TO DFLTØ
25	L12303:	111.1.1...			CLEAR REGISTERS
26	L12304:	.1..1....			1→F2
27	L12305:	.11.1....			1→F3
28	L12306:	....11.1..			CLEAR STATUS
29	L12307:	..1...11..			2→P
30	L12310:	..1..11...			LOAD CONSTANT 2
31	L12311:	..1..11...			LOAD CONSTANT 2
32	L12312:	...1.11...			LOAD CONSTANT 1
33	L12313:	1.111....			Ø→F5
34	L12314:	111.1....			1→F7
35	L12315:	..1.1.11.1→L2053			JSB LBL
36	L12316:	...1111..1→L2036			JSB A
37	L12317:	....1....1→L2010			JSB G
38	L12320:	...1.1...1→L2024			JSB INX
39	L12321:	..1.1.1..1→L2052			JSB RTN
40	L12322:	..1.1.11.1→L2053			JSB LBL
41	L12323:	...111...1→L2034			JSB B
42	L12324:	....111..1→L2016			JSB F

TABLE II (Continued).

43	L12325:	..11..1..1→L2062		JSB SQT
44	L12326:	..1.1.1..1→L2052		JSB RTN
45	L12327:	..1.1.11.1→L2053		JSB LBL
46	L12330:	...11.11.1→L2033		JSB C
47	L12331:	....1....1→L2010		JSB G
48	L12332:	...1..11.1→L2023		JSB YTX
49	L12333:	..1.1.1..1→L2052		JSB RTN
50	L12334:	..1.1.11.1→L2053		JSB LBL
51	L12335:	...11.1..1→L2032		JSB D
52	L12336:	....11.1.1→L2015		JSB RDN
53	L12337:	..1.1.1..1→L2052		JSB RTN
54	L12340:	..1.1.11.1→L2053		JSB LBL
55	L12341:	...11....1→L2030		JSB E
56	L12342:	...1...1.1→L2021		JSB EXCG
57	L12343:	..1.1.1..1→L2052		JSB RTN
58	L12344:	....1..1.1→L2011		JSB RSETP
59	L12345:	111.1.1..1→L2352		JSB CLRC0
60	L12352:	11....11..	CLRC0 :	12→P
61	L12353:	....1.111.	CLRC1 :	0→B[W]
62	L12354:	1...1.111.		B EXCHANGE C[W]
63	L12355:	....11....		RETURN

As an alternative to the above-disclosed method, the memory 20 could also be programmed by constructing a read-write memory with a read-only memory 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.

WHAT WE CLAIM IS:—

1. A calculator having electronic circuits, comprising: a keyboard having a plurality of keys including a user-definable function key; an encoder 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 for storing coded signals; computing means connected to the encoder 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; means whereby electrical power applying means can be connected to the electronic circuits including the encoder, the read-write memory and the computing means; and preprogramming means connected to the read-write memory and to said electrical power applying means for establishing, in

response to the application of power to the electronic circuits, a program of one or more coded signals in the read-write memory that can be processed in the computing means in response to the generation of the predetermined coded signal.

2. A calculator according to claim 1 wherein the program of coded signals stored in the read-write memory may be replaced with other coded signals by actuation of keys on the keyboard.

3. A calculator according to claim 2 wherein in program of coded signals represents a numerical constant.

4. A calculator according to either one of claims 2 and 3 wherein: the keyboard includes a manipulative function key having a corresponding first coded signal for causing the computing means to perform a first manipulative function; the keyboard includes a prefix key having a corresponding second coded signal for causing the computing 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 according to any one of the preceding claims which further comprises a switch connected to the computing means for setting the calculator in a program mode in a first position and a run mode in a second position, the first position enabling establishment of another program of one or more coded signals in the read-write memory in 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 according to claim 5 wherein the preprogramming 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.

7. A calculator according to claim 1 substantially as hereinbefore described with reference to the accompanying drawings.

ERIC POTTER & CLARKSON,
Chartered Patent Agents,
Kingsway House,
Kingsway,
London W.C.2B 6QX.

Printed for Her Majesty's Stationery Office by the Courier Press, Leamington Spa, 1976.
Published by the Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.

TABLE II (Continued).

43	L12325:	..11..1..1→L2062		JSB SQT
44	L12326:	..1.1.1..1→L2052		JSB RTN
45	L12327:	..1.1.11.1→L2053		JSB LBL
46	L12330:	...11.11.1→L2033		JSB C
47	L12331:	....1....1→L2010		JSB G
48	L12332:	...1..11.1→L2023		JSB YTX
49	L12333:	..1.1.1..1→L2052		JSB RTN
50	L12334:	..1.1.11.1→L2053		JSB LBL
51	L12335:	...11.1..1→L2032		JSB D
52	L12336:	....11.1.1→L2015		JSB RDN
53	L12337:	..1.1.1..1→L2052		JSB RTN
54	L12340:	..1.1.11.1→L2053		JSB LBL
55	L12341:	...11....1→L2030		JSB E
56	L12342:	...1...1.1→L2021		JSB EXCG
57	L12343:	..1.1.1..1→L2052		JSB RTN
58	L12344:	....1..1.1→L2011		JSB RSETP
59	L12345:	111.1.1..1→L2352		JSB CLRC0
60	L12352:	11....11..	CLRC0 :	12→P
61	L12353:	....1.111.	CLRC1 :	0→B[W]
62	L12354:	1...1.111.		B EXCHANGE C[W]
63	L12355:	....11....		RETURN

As an alternative to the above-disclosed method, the memory 20 could also be programmed by constructing a read-write memory with a read-only memory 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.

WHAT WE CLAIM IS:—

1. A calculator having electronic circuits, comprising: a keyboard having a plurality of keys including a user-definable function key; an encoder 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 for storing coded signals; computing means connected to the encoder 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; means whereby electrical power applying means can be connected to the electronic circuits including the encoder, the read-write memory and the computing means; and preprogramming means connected to the read-write memory and to said electrical power applying means for establishing, in