1 403 706

PATENT SPECIFICATION

(11) 1 403 706

(21) Application No. 54042/74 (22) Filed 15 June 1972
(62) Divided out of No. 1 403 710
(31) Convention Application No. 153 437
(32) Filed 15 June 1971 in
(33) United States of America (US)
(44) Complete Specification published 20 Aug. 1975
(51) INT CL² G06F 9/06
(52) Index at acceptance
G4A 10EX 11C 15D 16G 5A 5B 9C 9F 9X

(19)



(54) CALCULATORS

(71) We, HEWLETT-PACKARD COMPANY, of 1501 Page Mill Road, Polo 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 relates generally to calculators.

The present application has been divided out of our Application No. 28012/72 (Serial No. 1403710) which contains a more detailed description of the calculator hereinafter described and which is therefore incorporated herein by reference. The invention is also related to those disclosed in our co-pending U.K. applications Nos. 50950/74, 51661/74. 54012/74, 54011/74, 54010/74, and 54009/74. (Serial Nos. 1403703, 1403709, 1403705, 1403707, 1403704 and 1403708).

In the complete specification of our co-pending application No. 28012/72 (Serial No. 1403710), there is disclosed an electronic calculator including an input unit for entering information into the calculator, a basic first memory unit into which information may be written and from which information may be read, a basic second memory unit storing routines and subroutines to be executed by the calculator in performing selected functions, processing means responsive to information from the input unit or the basic first memory unit and to operating states within the calculator itself for selectively executing one or more of the routines and subroutines stored in the basic second memory unit to perform selected functions employing information from one or both of the input and basic first memory units, and an output unit for providing an output indication of the functions performed when the calculator is in use, said calculator including means enabling a user to removably plug one or more additional second memory units into the calculator to increase the number of functions that may be performed by the calculator.

In the complete specification of our co-pending U.K. application No. 50950/74 (Serial No. 1403703) there is disclosed an electronic calculator comprising a keyboard input unit for entering information into the calculator, a first memory unit into which information may be written and from which information may be read, a second memory unit for storing routines and subroutines to be executed by the calculator in performing selected functions, processing means resonsive to information from the input unit or the first memory unit and to operating states within the calculator itself for selectively executing one or more of the routines and subroutines stored in the second memory unit to perform selected functions employing information from one or both of the input and first memory units, and an output unit for providing an output indication of the functions performed, said first memory unit including a plurality of working registers and a plurality of storage registers, and said keyboard input unit including one or more keys for conditioning the calculator to perform both direct and indirect addressing between the working registers and the storage registers.

In the complete specification of our co-pending U.K. application No. 51661/74 (Serial No. 1403709) there is disclosed an electronic calculator including a keyboard input unit for entering information into the calculator, a first memory unit into which information may be written and from which information may be read, a second memory unit for storing routines and subroutines to be executed by the calculator in performing

selected functions, processing means responsive to information from the input unit or the first memory unit and to operating states within the calculator itself for selectively executing one or more of the routines and subroutines stored in the second memory unit to perform selected functions employing information from one or both of the input and first memory units, an output printer unit for providing an output indication of the functions performed, and means for causing the output printer unit to print out a mnemonic representation of each key actuation of the keyboard input unit.

In the complete specification of our co-pending U.K. application No. 54012/74 (Serial No. 1403705) there is disclosed an electronic calculator comprising keyboard input means for entering information into the calculator, first memory means into which information may be written and from which information may be read, second memory means for storing routines and subroutines to be executed by the calculator in performing selected functions, processing means responsive to information from the keyboard input means or the first memory means and to operating states within the calculator itself for selectively executing one or more of the routines and subroutines stored in the second memory means to perform selected functions employing information from one or both of the keyboard input and first memory means, control means for redefining a plurality of nonalphabetic keys of the keyboard input means as alphabetic keys, and output means for providing an output indication of the functions performed.

In the complete specification of our co-pending U.K. application No. 54011/74 (Serial No. 1403707) there is disclosed a programmable calculator comprising keyboard input means for entering information into the calculator, first memory means into which information may be written and from which information may be read, second memory means for storing routines and subroutines to be executed by the calculator in performing selected functions, processing means responsive to information from the keyboard input means or the first memory means and to operating states within the calculator itself for selectively executing one or more of the routines and subroutines stored in the second memory means to perform selected functions employing information from one or both of the keyboard input and first memory means, magnetic reading and recording means for recording a program stored in the first memory means onto a magnetic record member associated with the calculator and for loading a program recorded on a magnetic record member associated with the calculator into the first memory means, first control means for designating any programme stored within the first memory means as being secure when that program is recorded onto a magnetic record member associated with the calculator, second control means for identifying a secure program loaded into the first memory means from a magnetic record member associated with the calculator and for thereupon preventing the calculator from listing or subsequently re-recording the secure program, and output means for providing an output indication of the functions performed by the calculator.

In the complete specification of our co-pending application No. 54010/74 (Serial No. 1403704) there is disclosed a prgrammable electronic calculator comprising keyboard input means for entering information into the calculator, first memory means into which information may be written and from which information may be read, second memory means for storing routines and subroutines to be executed by the calculator in performing selected functions, processing means responsive to information from the keyboard input means or the first memory means and to operating states within the calculator itself for selectively executing one or more of the routines and subroutines stored in the second memory means to perform selected functions employing information from one or both of the keyboard input and first memory means, magnetic reading and recording means for recording information stored in the first memory means onto a magnetic record member associated with the calculator and for loading information recorded on a magnetic record member associated with the calculator into the first memory means, control means for enabling the magnetic reading and recording means to separately record onto one or more magnetic record members associated with the calculator both program information and data stored in the first memory means and to thereafter separately reload the recorded program information and data back into the first memory means from the one or more magnetic record members, and output means for providing an output indication of the functions performed by the calculator.

In the complete specification of our co-pending U.K. application No. 54009/74 (Serial No. 1403708) there is disclosed a programmable electronic calculator comprising keyboard input means for entering information into the calculator, said keyboard input means including a first conversion control key, first memory means into which information may be written and

from which information may be read, second memory means for storing routines and subroutines to be executed by the calculator in performing selected functions, processing means responsive to information from the keyboard input means or the first memory means and to operating states within the calculator itself for selectively executing one or more of the routines and subroutines stored in the second memory means to perform selected functions employing information from one or both of the keyboard input and first memory means, logic means responsive to actuation of said first conversion control key for converting information provided by an angular measurement specified in decimal degrees to the equivalent degrees, minutes, and seconds, and output means for providing an output indication of the functions performed by the calculator.

In some conventional programmable calculators a program stored within the calculator may be edited by single stepping forward through the program while viewing an output display representing the last-encountered program step and its associated address and, in one case, also the presently-encountered program step and its associated address. However, these calculators typically cannot single step backward through the program or display the next program step to be encountered and its associated address. Moreover, they typically have no provision for inserting program steps into the program without reloading portions of the program and no provision for finding every occurrence of any designated program step. Such features would also be very helpful to the user in editing programs.

The present invention provides an electronic calculator comprising keyboard input means for entering information into the calculator, first memory means into which information may be written and from which information may be read, second memory means for storing routines and subroutines to be executed by the calculator in performing selected functions, processing means responsive to informing from the keyboard input means or the first memory means and to operating states within the calculator itself for selectively executing one or more of the routines and subroutines stored in the second memory means to perform selected functions employing information from one or both of the keyboard input and first memory means, insertion means for inserting a program step at a designated location within a series of program steps stored in the first memory means, and output means for providing an output indication of the functions performed.

In an electronic calculator as set forth in the last preceding paragraph preferably the insertion means includes an insert key of the keyboard input means, and logic means responsive to actuation of the insert key for conditioning the calculator to insert a program step at a designated location within a series of program steps stored in the first memory means.

In an electronic calculator as set forth in the last preceding paragraph preferably the logic means is further responsive to actuation of the insert key and one or more address designating keys for causing the calculator to insert a program step at the address designated by those address designating keys and for incrementing the address of each program step following the program step to be inserted.

An electronic calculator as set forth in any one of the last three preceding paragraphs preferably includes deletion means for deleting a program step at a designated location within a series of program steps stored in the first memory means.

In an electronic calculator as set forth in the last preceding paragraph preferably the deletion means includes a delete key of the keyboard input means, and logic means responsive to actuation of the delete key for conditioning the calculator to delete a program step at a designated location within a series of program steps stored in the first memory means.

In an electronic calculator as set forth in the last preceding paragraph preferably the logic means is further responsive to actuation of the delete key and one or more address designating keys for causing the calculator to delete the program step at the address designate by those address designating keys and for decrementing the address of each program step following the program step to be deleted.

An electronic calculator as in any of the last six preceding paragraphs but three preferably includes deletion means for deleting a user defined function stored in the first memory means.

In an electronic calculator as in the last preceding paragraph preferably the deletion means includes a delete key of the keyboard input means, and logic means responsive to actuation of the delete key for conditioning the calculator to delete a designated user defined function stored in the first memory means.

In an electronic calculator as in the last preceding paragraph preferably the logic means is further responsive to actuation of the delete key and one or more function designating keys for causing the calculator to delete the user defined function designated by those function designating keys.

In an electronic calculator as in any one of the last nine preceding paragraphs preferably the output means in an output display unit, and the calculator includes location means for causing the calculator to locate and display an indication of each occurrence and address of a specified program step within a series of program steps stored in the first memory means.

In an electronic calculator as in the last preceding paragraph preferably the location means includes a find key of the keyboard input means, and logic means responsive to actuation of the find key for conditioning the calculator to locate and display an indication of each occurrence and address of a specified program step within a series of program steps stored in the first memory means.

In an electronic calculator as in the last preceding paragraph preferably the logic means is further responsive to actuation of the find key and a program step designating key for locating and displaying an indication of each occurrence and address of the program step designated by that program step designating key.

An electronic calculator as in any one of the last twelve preceding paragraphs preferably includes backstep means operable, when the calculator is in a program checking mode, for stepping backward through the steps of a program stored in the first memory means.

In an electronic calculator as in the last preceding paragraph preferably the backstep means includes a backstep key of the keyboard input means, and logic means responsive to actuation of the backstep key for causing the calculator to step backward through the steps of a program stored in the first memory means.

In an electronic calculator as in the last preceding paragraph preferably the logic means is further responsive to actuation of the backstep key, when the calculator is in the program checking mode, for decrementing the contents of a program step pointer located in the first memory means and thereby stepping backward through the steps of a program stored in the first memory means.

An electronic calculator as in any one of the last fifteen preceding paragraphs but six preferably includes program indicating means operable when the calculator is in a program checking mode, for simultaneously indicating the program step presently encountered, the program step last encountered, the program step next-to-be encountered, and the addresses thereof.

In an electronic calculator as in the last preceding paragraph preferably the first memory means includes at least three display storage registers, the output means in an output display unit for displaying the contents of those display storage registers, the calculator includes a program control switch for setting the calculator to the program checking mode and a program step pointer for sequentially designating the address of each program step within a program stored in the first memory means, and the program indicating means is operable, when the calculator is in the program checking mode, for placing an indication of the address and program step then designated by the program step pointer in one of the display storage registers, for placing an indication of the next preceding address and program step in another of the display storage registers, and for placing an indication of the next following address and program step in still another of the display storage registers.

There now follows a detailed description which is to be read with reference to the accompanying drawings of an electronic programmable 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 is a front perspective view of an electronic programmable calculator according to the preferred embodiment of this invention;

Figures 2A—B are a simplified block diagram of the calculator of Figure 1;

Figure 3 is a memory map of the memory unit employed in the calculator of Figures 1 and 2A—B;

Figure 4 is a detailed memory map of the dedicated portion of the data storage section of the read-write memory employed in the memory unit of Figures 2A—B and 3;

Figure 5 is a plan view of the keyboard input unit employed in the calculator of Figures 1 and 2A—B; and

Figure 6 is a plan view of the definable keys of the keyboard input unit employed in the calculator of Figures 1 and 2A—B and the overlay associated with a definable functions plug-in read-only memory module that may be employed in the calculator.

CALCULATOR ARCHITECTURE

Referring to Figure 1, there is shown an electronic calculator 10 including both a keyboard input unit 12 for entering information into and controlling the operation of the calculator and a magnetic card reading and recording unit 14 for recording information stored within the calculator onto one or more external magnetic cards 16 and for subsequently loading the information recorded on these and other similar magnetic cards back into the calculator. The calculator also includes

(14)

a solid state output display unit 18 for displaying three lines of numeric information stored within the calculator and a group of indicator lights 19, serving as part of the keyboard input unit, for indicating the status of the calculator. It may also include an output printer unit 20 for printing out alphameric information on a strip of thermal-sensitive recording paper 22.

Referring to the simplified block diagram shown in Figures 2A—B, it may be seen that the calculator also includes an input-output control unit 44 (hereinafter referred to as the I/O control unit) for controlling the transfer of information to and from the input and output units, a memory unit 46 for storing and manipulating information entered into the calculator and for storing routines and subroutines of basic instructions performed by the calculator, and a central processing unit 48 (hereinafter referred to as the CPU) for controlling the execution of the routines and subroutines of basic instructions stored in the memory unit as required to process information entered into or stored within the calculator. The calculator also includes a bus system comprising an S-bus 50, a T-bus 52, and an R-bus 54 for transferring information from the memory and I/O control units to the CPU, from the CPU to the memory and I/O control units, and between different portions of the CPU. It further comprises a power supply for supplying DC power to the calculator and peripheral units employed therewith and for issuing a control signal $\overline{POP}$ when power is supplied to the calculator.

The I/O control unit 44 includes an input-output register 56 (hereinafter referred to as the I/O register), associated I/O gating control circuitry 58, and input-output control logic 60 (hereinafter referred to as the I/O control). I/O register 56 comprises a universal sixteen-bit shift register into which information may be transferred either bit-serially from CPU 48 via T-bus 52 or in parallel from keyboard input unit 12, magnetic card reading and recording unit 14, and peripheral input units 28 such as the marked card reader via twelve input party lines 62. Information may also be transferred from I/O register 56 either bit-serially to CPU 48 via S-bus 50 or in parallel to magnetic card reading and recording unit 14, solid state output display unit 18, indicator lights 19, output printer unit 20, and peripheral output units 28 such as the X—Y plotter or the typewriter via sixteen output party lines 64.

I/O gating control circuitry 58 includes control circuits for controlling the transfer of information into and out of I/O register 56 in response to selected I/O qualifier control signals from CPU 48 and selected I/O control instructions from I/O control 60. It also includes an interrupt control circuit 65, a peripheral control circuit 66, a magnetic card control circuit 67, a printer control circuit 68, a display control circuit 69, and an indicator control circuit 70 for variously controlling the input and output units and issuing control signals QFG and EBT to I/O control 60 via two output lines 71 and 72. These last mentioned control circuits variously perform their control functions in response to control signal $\overline{POP}$ from the power supply, I/O qualifier control signals from CPU 48, I/O control instructions from I/O control 60, and control signals from keyboard input unit 12. Interrupt control circuit 65 initiates the transfer of information into I/O register 56 from keyboard input unit 12 or interrupting peripheral input units 28 such as the marked card reader and issues a qualifier control signal QNR to CPU 48 via output lines 73. Peripheral control circuit 66 enables interface modules 30 plugged into the calculator to respond to information from I/O register 56, control associated peripheral units 28, transfer information to and/or receive information from associated peripheral units 28, and in some cases initiate the transfer of information to I/O register 56 from the interface modules themselves. Magnetic card control circuit 67 enables magnetic card reading and recording unit 14 to respond to information in I/O register 56 and either read information into I/O register 56 from a magnetic card 16 or record information onto a magnetic card 16 from I/O register 56. Printer control circuit 68, display control circuit 69, and indicator control circuit 70 enable output printer unit 20, output display unit 18, and indicator lights 19, respectively, to respond to information from I/O register 56.

When a basic I/O instruction obtained from memory unit 46 is to be executed, CPU 48 transfers control to I/O control 60 by issuing a pair of I/O microintructions $\overline{PTR}$ and $\overline{XTR}$ thereto. In response to these I/O microinstructions from CPU 48, control signal $\overline{POP}$ from the power supply, control signals QFG and EBT from I/O gating control circuitry 58, and I/O qualifier and clock control signals from CPU 48, I/O control 60 selectively issues one or more I/O control instructions to gating control circuitry 58 as required to execute the basic I/O instruction designated by CPU 48 and issues control signals $\overline{TTX}$, $\overline{XTR}$, QRD, and SCB to CPU 48 via output lines 74—77. The I/O qualifier control signals issued to I/O control 60 and gating control circuitry 58 by CPU 48 are derived from the basic I/O instruction to be executed. Those qualifier

control signals issued to I/O control 60 designate the specific I/O control instructions to be issued by I/O control 60, while those issued to gating control circuitry 58 designate selected control circuits to be employed in executing the basic I/O instruction.

Memory unit 46 includes a modular random-access read-write memory 78 (hereinafter referred to as the RWM), a modular read-only memory 80 (hereinafter referred to as the ROM), a memory address register 82 (hereinafter referred to as the M-register), a memory access register 84 (hereinafter referred to as the T-register), and control circuitry 85 for these memories and registers. RWM 78 and ROM 80 comprise MOS-type semiconductor memories. As shown in the memory map of Figure 3, they are organized into eight 1,024-word pages. The basic RWM 78 contains a data storage section 86 of 512 sixteen-bit words extending from address 1000 to address 1777 on page 0 and a separate program storage section 88 of 512 six-bit words extending from address 12000 to address 12777 on page 5. All addresses on the memory map are represented in octal form.

Data storage section 86 contains 49 four-word storage registers available to the user (as user addresses 000—048) for manipulating and storing data, 60 additional four-word storage registers that may be made available to the user (as user addresses 049—108) for the same purpose, and 76 words dedicated for use by CPU 48. The 60 additional four-word storage registers may be made available to the user by altering a control routine in the ROM section 80. This is accomplished by removing a top panel 90 of the calculator housing shown in Figure 1, removing a printed circuit board containing the control routine to be altered, and substituting another printer circuit board containing the altered control routine. Additional data storage modules made available to the user are automatically accommodated by the calculator.

As shown in the more detailed memory map of Figure 4, the dedicated portion of data storage section 86 includes 16 words (addresses 1750—1753, 1740—1743, and 1760—1763) employed as "x", "y", and "z" four-word working registers available to the user and 8 words (addresses 1764—1773) employed as "a" and "b" four-word storage registers available to the user. It also includes 8 words (addresses 1744—1747 and 1754—1757) employed as "AR1" and "AR2" four-word working registers for performing binary-coded-decimal arithmetic; 12 words (addresses 1664—1677) employed as three ("T1'", "T2'", and "T3'") or more temporary storage registers in connection with a definable section 91 of keyboard input unit 12 (see Figure 1); 16 words (addresses 1720—1737) employed as four ("T1", "T2", "T3", and "T4") or more temporary storage registers in connection with the remaining sections of the keyboard input unit; 7 words (addresses 1711—1717) employed as a variable-length "system subroutine stack" for storing return addresses required by programs stored in ROM 80 and as temporary storage for housekeeping information required by CPU 48; one word (address 1777) employed as a "system stack pointer"; 5 words (addresses 1702—1706) employed as a "user subroutine stack" for storing return addresses required by programs entered into program storage section 88 of RWM 78 by the user; 1 word (address 1710 employed as a "user stack pointer"; 1 word (address 1701) employed as a "user program counter" for program steps entered into program storage section 88 by the user; 1 word (address 1707) employed as an "address of user 0000" register for storing the first address available to the user in program storage section 88; 1 word (address 1700) employed as a "microprocessor store" register for storing housekeeping information required by CPU 48; 1 word (address 1774) employed as an "interrupt store" register for storing information displaced from CPU 48 by some keyboard entries; 1 word (address 1775) employed as an "input buffer" register for storing keyboard information awaiting entry into CPU 48, and 1 word (address 1776) employed as a "status word" register for storing information regarding the present state of the calculator.

As shown in the memory map of Figure 3, program storage section 88 of RWM 78 contains 500 one-word program step registers available to the user (as user addresses 0000—0500) for storing programs and 12 words dedicated for use by CPU 48. An additional 1,536 one-word program-step registers may be made available to the user (as user addresses 0511—2036) in steps of 512 words (addresses 13000—13777) and 1,024 words (addresses 14000 to 15777). This is accomplished by removing top panel 90 of the calculator housing shown in Figure 1 and plugging additional program storage modules into the calculator. Added program storage modules are automatically accommodated by the calculator.

As shown in the memory map of Figure 3, the basic ROM 80 contains 2,048 sixteen-bit words extending from address 0000 to address 0777 on page 0, from address 4000 to address 5777 on page 2, and from address 16000 to address 16777 on page 7. Routines and subroutines of basic instructions for

(16)

performing the basic functions of the calculator and constants employed by these routines and subroutines are stored in these portions of ROM 80. An additional 3,072 sixteen-bit words of ROM may also be added on pages 1, 3, and 4 in steps of 512 and 1,024 words. This is accomplished by simply inserting plug-in ROM modules 92 into receptacles 94 provided therefor in top panel 90 of the calculator housing as illustrated in Figure 1 by the partially-inserted plug-in ROM module on the left. As each plug-in ROM module 92 is inserted into one of these receptacles a spring-loaded door 95 at the entrance of the receptacle swings down allowing passage of the plug-in ROM module. Once the plug-in ROM module is fully inserted as illustrated by the plug-in ROM module on the right, a printed circuit terminal board 96 contained within the plug-in ROM module plugs into a mating edge connector mounted inside the calculator.

Routines and subroutines of basic instructions (and any needed constants) for enabling the calculator to perform many additional functions are stored in each plug-in ROM module 92. The user himself may therefore quickly and simply adapt the calculator to perform many additional functions oriented toward his specific needs by simply plugging ROM modules of his own choosing into the calculator. Added plug-in ROM modules are automatically accommodated by the calculator and associated with definable section 91 of keyboard input unit 12 or employed to expand the functions performed by this and other sections of the keyboard input unit.

Referring again to Figurs 2A—B, M-register 82 of the memory unit comprises a recirculating sixteen-bit serial shift register into which information may be transferred bit-serially from CPU 48 via T-bus 52 and out of which information may be transferred bit-serially to CPU 48 via $\overline{S}$-bus 50. Information shifted into M-register 82 may be employed to address any word in RWM 78 or ROM 80 via fifteen output lines 106.

T-register 84 of the memory unit comprises a recirculating sixteen bit serial shift register into which information may be transferred either bit-serially from CPU 48 via T-bus 52 or in parallel from any addressed word in RWM 78 and ROM 80 via sixteen parallel input lines 108. Information may be transferred from T-register 104 either bit-serially to CPU 48 via $\overline{S}$-bus 50 or in parallel to any addressed word in RWM 78 via sixteen parallel output lines 110. The four least significant bits of information contained in T-register 104 may comprise binary-coded-decimal information and may be transferred from the T-register in parallel to CPU 48 via three parallel output lines 112 taken with $\overline{S}$-bus 50.

The control circuitry 85 of the memory unit controls these transfers of information into and out of M-register 82 and T-register 84, controls the addressing and accessing of RWM 78 and ROM 80, and refreshes RWM 78. It performs these functions in response to memory microinstructions, memory clock pulses, and shift clock pulses from CPU 48.

CPU 48 includes a register unit 114, an arithmetic-logic unit 116 (hereinafter referred to as the ALU), a programmable clock 118, and a microprocessor 120. Register unit 114 comprises four recirculating sixteen-bit shift registers 122, 124, 126, and 128 and one four-bit shift register 130. Shift registers 122 and 124 serve as sixteen-bit serial accumulator registers (hereinafter referred to as the A-register and the B-register, respectively) into which information may be transferred bit-serially from ALU 116 via T-bus 52 and out of which information may be transferred bit-serially to ALU 116 via $\overline{R}$-bus 54. The four least significant bit positions of A-register 122 also serve as a four-bit parallel accumulator register into which four bits of binary-coded-decimal information may be transferred in parallel from ALU 116 via four parallel input lines 132 and out of which four bits of binary-coded-decimal information may also be transferred in parallel to ALU 116 via three parallel output lines 134 taken with $\overline{R}$-bus 54.

Shift register 126 serves as a sixteen-bit system program counter (hereinafter referred to as the P-register) into which information may be transferred bit-serially from ALU 116 via T-bus 52 and out of which information may be transferred bit-serially to ALU 116 via $\overline{R}$-bus 54. Information contained in the least significant bit position of P-register 126 may also be transferred as a qualifier control signal QPO to microprocessor 120 via output line 135.

Shift register 128 serves as a sixteen-bit qualifier register (hereinafter referred to as the Q-register) into which information may be transferred bit-serially from ALU 116 via T-bus 52 and out of which information may be transferred bit-serially to ALU 116 via $\overline{R}$-bus 54. Information contained in the five least-significant bit positions of Q-register 128 is transferred to I/O gating control circuitry 58 as five one-bit I/O qualifier control signals Q00—Q04 via five parallel output lines 136, and information contained in the six next-least-significant bit positions of the Q-register is transferred to I/O control 60 as six one-bit I/O qualifier

control signals Q05—Q10 via six parallel output lines 138. Similarly, information contained in the seven least-significant, the ninth and eleventh least-significant, and the most-significant bit positions of Q-register 128 and information derived from the thirteenth, fourteenth, and fifteenth least-significant bit positions of the Q-register may be transferred to microprocessor 120 as eleven one-bit microprocessor qualifier control signals Q00—Q06, Q08, Q10, Q15, and QMR via eleven output lines 140. Information contained in the twelfth through the fifteenth least-significant bit positions of Q-register 128 may be transferred to microprocessor 120 as a four-bit primary address code via four parallel output lines 142.

Shift register 130 serves as a four-bit serial extend register (hereinafter referred to as the E-register) into which informatiion may be transferred bit-serially either from ALU 116 via T-bus 52 or from the least-significant bit position of T-register 84 via input line 144. Information may also be transferred out of E-register 130 to ALU 116 via $\overline{R}$-bus 54.

Register unit 114 also includes control circuitry 146 for controlling the transfer of parallel binary-coded-decimal information into and out of A-register 122 and the transfer of serial binary information into and out of A-register 122, B-register 124, P-register 126, Q-register 128, and E-register 130. This is accomplished in response to register microinstructions from microprocessor 120, control signals $\overline{TTX}$ and $\overline{XTR}$ from I/O control 60, and shift clock control pulses from programmable clock 118. Control circuitry 146 includes a flip-flop 148 (hereinafter referred to as the A/B flip-flop) for enabling the transfer of information into and out of either the A-register 122 or the B-register 124 as determined by the state of the A/B flip-flop. The state of A/B flip-flop 148 is initially determined by information Q11 transferred to the A/B flip-flop from the twelth least-significant bit position of Q-register 128 but may be subsequently complemented one or more times by microinstruction CAB from microprocessor 120.

ALU 116 may perform either one-bit serial binary arithmetic on data received from T-register 84 or M-register 82 via $\overline{S}$-bus 50 and/or from any register of register unit 114 via $\overline{R}$-bus 54 or four-bit parallel binary-coded-decimal arithmetic on data received from T-register 84 via output lines 112 taken with $\overline{S}$-bus 50 and/or from A-register 122 via output lines 134 taken with $\overline{R}$-bus 54. It may also perform logic operations on data received from memory Unit 46 and/or register unit 114 via any of these lines. The arithmetic and logic operations performed are designated by ALU microinstructions from microprocessor 120 and are carried out in response to these microinstructions, shift clock control pulses from programmable clock 118, and control signal SCB from I/O control 60. Information is also transferred from ALU 116 to A-register 122 via output lines 132 or to I/O register 56, M-register 82, T-register 84, or any register of register unit 114 via T-bus 52 in response to microinstructions and control signals applied to these registers. If a carry results while ALU 116 is performing either one-bit serial binary arithmetic or four-bit parallel binary-coded-terminal arithmetic, the ALU issues a corresponding qualifier control signal QBC and QDC to microprocessor 120 via one of two output lines 152 and 154.

Programmable clock 118 includes a crystal-controlled system clock 156, a clock decoder and generator 158, and a control gate 160. System clock 156 issues regularly recurring clock pulses to clock decoder and generator 158 via output line 162. In response to these regularly recurring clock pulses from system clock 156 and to four-bit clock codes from microprocessor 120, clock decoder and generator 158 issues trains of n shift clock pulses to ALU 116, M-register 82, T-register 84, and all of the registers of register unit 114 via output line 164. These trains of n shift clock pulses are employed for shifting a corresponding number of bits of serial information into or out of any of these registers of for shifting a carry bit in the ALU. The number n of pulses in each of these trains may vary from one to sixteen as determined by the number of bits of serial information required during each operation to be performed. In response to a control signal CCO from microprocessor 120, control gate 160 prevents any shift clock pulses from being applied to the ALU or any of these registers. Upon completion of each train of n shift clock pulses, clock decoder and generator 158 issues a ROM clock pulse to microprocessor 120 via output line 166 and an I/O clock pulse to I/O control 60 via output line 168. In response to the regularly recurring clock signal from system clock 156, clock decoder and generator 158 also issues correspondingly regularly recurring memory clock pulses to memory unit 46 via output line 170.

Microprocessor 120 selectively issues two I/O microinstructions to I/O control 60 via two output lines 172, six memory microinstructions to memory unit 46 via six output lines 174, thirteen register microinstructions to register unit 114 via thirteen output lines 176, and five ALU microinstructions to ALU 116 via five output lines 178. It also issues a four-bit

clock code associated with each of these microinstructions to clock decoder 158 via four output lines 180. These microinstructions and associated clock codes are issued as determined by the control signal $\overline{POP}$ from the power supply the eleven microprocessor qualifier control signals from Q-register 128, the four-bit primary address codes from Q-register 128, and the five microprocessor qualifier control signals from I/O control 60, interrupt control 65, ALU 116, and P-register 126.

CALCULATOR OPERATION

All operations performed by the calculator may be controlled or initiated by the keyboard input unit and/or by keycodes entered into the calculator from the keyboard input unit, the magnetic card reading and recording unit, or peripheral input units such as the marked card reader and stored as program steps in the program storage section of the RWM. The calculator responds to keycodes in basically the same manner whether obtaining them from the keyboard input unit or from the program storage section of the RWM. An operational description of the keyboard input unit is therefore now given with specific reference to Figures 1 and 5 except as otherwise indicated.

An on-off line switch 182, which may be considered as part of the keyboard input unit, controls the application of power to the calculator and hence initiation of the control signal $\overline{POP}$ from the power supply. The indicator lights 19 serve as a pilot light since at least two of them are always turned on while power is applied to the calculator.

When the calculator is first turned on, it is automatically initialized and placed in a manual operating mode. If the calculator is switched to some other operating mode, it may subsequently be placed in the manual operating mode again by simply depressing the RUN mode key 284. In the manual operating mode, operation of the calculator is manually controlled by the user from the keyboard input unit. During this mode the output display unit displays a decimal numeric representation of the contents of the x-, y-, and z-registers (or of associated memory registers in which the actual or intended contents of the x-, y-, and z-registers are temporarily stored and, for simplicity of description, are then considered to be the contents of the x-, y-, and z-registers). The contents of the x-, y-, and z-registers are displayed adjacent to the corresponding register designators "keyboard x", "accumulator y", and "temporary z", respectively, in the display window. Depression of the RUN mode key 284 also conditions the calculator for operation in an automatic operating mode, a first key-log printing mode, a program-list printing mode, a magnetic card reading mode, and a magnetic card recording mode as determined by the other keys hereinafter explained. A run indicator light 19 positioned immediately below the RUN mode key 284 is turned on when the calculator is operating in any of the RUN modes.

The PRGM mode key 286 is depressed to place the calculator in a program entering mode. In this mode keycodes sequentially entered by the user from the keyboard input unit are stored as program steps in successive program-step registers of the program storage section of the RWM as specified by the user program counter. The program step register into which each program step is to be stored and from which each program step is to be obtained in any program-related operation is always specified by the user program counter. Thus, before entering a program or subprogram into the calculator, the user program counter must be set to the address of the program step register into which it is desired to store the initial program step of the program or subprogram to be entered (this address is hereinafter referred to as the desired starting address nnnn of the program). This may be accomplished, when the calculator is in a keyboard-controlled run mode, by depressing the GO TO key followed by the decimal-digit keys 0—9 specifying the desired starting address nnnn of the program. If the desired starting address is 0000, this may also be accomplished when the calculator is in the manual operating mode by simply depressing the END key.

Once the user program counter is set to the desired starting address nnnn, the user may proceed to enter the program or subprogram by sequentially performing basically the same key operations that he would normally perform in the manual operating mode. Thus no special language need be learned to program the calculator. During the program entering mode the output display unit displays a decimal numeric representation of the last-entered program step and its associated address and the addresses of the next two program steps to be entered and the present contents of those addresses.

Depression of the PRGM mode key 286 also conditions the calculator for operation in a second key-log printing mode and the program-list printing mode as determined by other keys hereinafter explained. A program mode indicator light 19 positioned immediately below the PRGM mode key 286 is turned on when the calculator is operating in any of the PRGM modes.

The KEY LOG mode key 288 is depressed, when the calculator is in the manual operating mode, to place the calculator in the first key-log printing mode. In this mode, the output printer unit prints out an octal numeric representation of each keyboard operation as it is performed by the user. This provides a permanent record of all keyboard operations (including regular data print-out operations), as illustrated by the following example:

Keyboard entry	Key log
1	01
2	02
↑	27
3	03
4	04
+	33
5	05
÷	35
PRINT / SPACE	45
	5.00000

If an alpha ROM module 92 enabling the calculator to print out every alphabetic character and many symbols individually or in messages is plugged into the calculator, the output printer unit also prints out a mnemonic representation of each keyboard operation as it is performed by the user. This is illustrated by the following example:

Keyboard entry	Key log	
1	1	01
2	2	02
↑	UP	27
3	3	03
4	4	04
+	+	33
5	5	05
÷	DIV	35
PRINT / SPACE	PNT	45
		5.00000

In the first key-log printing mode the output display unit displays the same information as during the manual operating mode.

The KEY LOG mode key 288 is depressed, when the calculator is in the program entering mode, to place the calculator in the second key-log printing mode. In this mode the output printer unit prints out an octal numeric representation of each keycode as it is entered into the calculator from the keyboard input unit and a decimal numeric representation of the address at which each such keycode is stored as a program step in the program storage section of the RWM. This provides a permanent record of all keyboard-entered program steps, as illustrated by the following example:

Keyboard entry	Key log	
CLEAR	0000	20
1	0001	01
2	0002	02
↑	0003	27
3	0004	03
4	0005	04
+	0006	33
5	0007	05
$x \rightleftharpoons y$	0008	30
PRINT / SPACE	0009	45
STOP	0010	41
END	0011	46

If the alpha ROM module 92 is plugged into the calculator, the output printer unit also prints out a mnemonic representation of each keyboard-entered program step. This is illustrated by the following example:

Keyboard entry	Key log	
CLEAR	0000 — CLR	20
1	0001 — 1	01
2	0002 — 2	02
↑	0003 — UP	27
3	0004 — 3	03
4	0005 — 4	04
+	0006 — +	33
5	0007 — 5	05
$x \rightleftharpoons y$	0008 — XEY	30
PRINT	0009 — PNT	45
STOP	0010 — STP	41
END	0011 — END	46

In the second key-log printing mode the output display unit displays the same information as during the program entering mode.

The KEY LOG mode key 288 is a toggling on-off key (i.e. repeated depressions of the key alternately switch the calculator in and out of either the first or the second key-log printing mode). A key-log indicator light key 19 positioned immediately below the KEY LOG mode key 288 is turned on when the calculator is operating in either key-log printing mode.

The LIST program key 290 is depressed, when the calculator is in the manual operating mode, first key-log printing mode, second key-log printing mode, or program entering mode, to place the calculator in the program-list printing mode. In this mode, the output printer unit prints out an octal numeric representation of keycodes then stored as program steps in the program storage section of the RWM and a decimal numeric representation of the addresses of these program steps. These program steps and addresses are printed out in a list beginning with the address initially specified by the user program counter and ending with the address specified by the user program counter when an END program step is encountered or the STOP key is depressed. The user may select the starting address nnnn of the list by depressing the GO TO key followed by the decimal digit keys (0—9) designating the desired starting address nnnn, or simply by depressing the END key if the starting address is 0000. Similarly, the user may terminate the list at any time by depressing the STOP key. When the program-list operation is terminated either by an END program step or by depression of the STOP key, the calculator reverts to its original manual operating, first or second key-log printing, or program entering mode. If the program-list operation is terminated by an END program step, the user program counter specifies the address of the END program step. However, if the program-list operation is terminated by depression of the STOP key, the user program counter specifies the address of the next program step to have been encountered had the STOP key not been depressed.

The list printed out by the output printer unit serves as a permanent record of a sequence of program steps stored as a program, subprogram, or part thereof in the program storage section of the RWM and the address of these program steps. If the alpha ROM module is plugged into the calculator, the output printer unit also prints out a mnemonic representation of each of these program steps.

The CONTINUE key is depressed, when the calculator is in the manual operating

mode or the first key-log printing mode, to start the automatic execution of a program or subprogram stored within the program storage section of the RWM. Automatic execution begins at the address specified by the user program counter. Thus, in order to execute a desired program or subprogram stored within the program storage section of the RWM:

1. The user program counter is set to the starting address nnnn of the desired program or subprogram. This is accomplished by depressing the GO TO key followed by those decimal digit keys designating the starting address nnnn of the desired program. If the starting address is 0000, this may be accomplished by simply depressing the END key.

2. The CONTINUE key is depressed, when the calculator is in the manual operating mode or the first key-log printing mode, to place the calculator in the automatic operating mode and begin automatic execution of the stored program or subprogram at the address nnnn specified by the user program counter. Automatic execution will continue unitl a STOP or END program step is encountered or a STOP key is depressed as described below.

The STOP key is depressed, when the calculator is in the automatic operating mode, to halt the automatic execution of a stored program or subprogram immediately after completion of the program step then being executed. Automatic execution of a stored program or subprogram is similarly halted when a STOP program step is encountered. In either case the calculator will thereupon revert to the original manual operating or first key-log printing mode with the program counter specifying the address of the next program step to be encountered. The user may then operate the calculator in the manual operating mode to enter data required by the program being executed or to perform other calculations. So long as the user does not depress the GO TO or END key or perform some other operation altering the last setting of the user program counter, he may resume automatic operation of the calculator at any time by simply depressing the CONTINUE key again.

As described above, the STOP key may also be depressed, when the calculator is in the program-list printing mode, to halt the program-list printing operation. The calculator will revert to the mode it was in immediately prior to the halted operation with the user program counter specifying the address of the next program step to have been printed, read, or recorded had the STOP key not been depressed.

The END key is depressed, when the calculator is in the manual operating mode or the first key-log printing mode, to set the user program counter to address 0000 in the program storage section of the RWM (this is equivalent to depressing the GO TO, 0, 0, 0, and 0 keys). An END program step terminates the automatic execution of a stored program by the calculator and resets the user program counter to the first available address 0000 in the program storage section of the RWM. The calculator thereupon reverts to the original manual operating or first key-log printing mode, from which automatic operation was initiated. Automatic execution may then be resumed by depressing the CONTINUE key, if the desired starting address is 0000 or by repeating steps 1 and 2 described above in connection with the CONTINUE key if the desired starting address is not 0000. An END program step also clears any subroutine return-address to which return has not by then been made. As described above, it also terminates the program listing operation.

The PAUSE key is typically used only as a program step. Automatic execution of a stored program or subprogram is automatically halted for a $\frac{1}{4}$ second pause interval whenever a PAUSE program step is encountered or the PAUSE key is depressed. This enables partial results of a calculation to be displayed by the output display unit during automatic execution of the stored program or subprogram. Successive PAUSE program steps may be employed to increase the duration of the pause interval by $\frac{1}{4}$ second increments. Automatic execution of the stored program or subprogram automatically resumes after the pause interval.

A PAUSE program step may also be used as a conditional stop permitting the user to stop the automatic operation of the calculator immediately after execution of any PAUSE program step. This is accomplished by simply depressing any key (other than STOP) during automatic execution of the program until the PAUSE program step has been executed. In other words, a PAUSE program step immediately followed by depression of any key other than STOP has the same effect as depressing STOP, with the advantage that automatic execution of a stored program or subprogram may thereby be precisely halted at one or more predetermined program steps if the user so desires. Automatic execution of the stored program may then be resumed by depressing the CONTINUE key.

The decimal digit keys 0—9 are depressed to enter numbers into the x-register. A number entered into the x-register is terminated as soon as any non-

data-entry key is depressed. Another number entered into the x-register will automatically replace a terminated number, but will become a part of any non-terminated number.

The decimal point (.) key is depressed to enter the decimal point into the x-register. For example, the number 1.234 is entered by sequentially depressing the 1, ., 2, 3, and 4 keys. Regardless of the display format, it is not necessary to use the decimal point key when entering integers. If the decimal point key is not used, the decimal point will be assumed to have followed the last-entered digit. When the fixed decimal point display format is used, the calculator automatically positions the decimal point. Similarly, when the floating decimal point display format is used the calculator automatically corrects the exponent according to the position of the decimal point.

The ENTER EXP key is depressed followed by one or two decimal digit keys to enter a one- or two-digit exponent (power of ten) into the x-register, the last digit entered becoming the least significant digit. If a third digit is entered, it will terminate entry of the exponent and begin a new numeric data entry. A non-terminated number in the x-register may be multiplied directly by successive powers of ten by simply entering successive exponents. If the ENTER EXP key is depressed as the first key of a numeric data entry (i.e. before any of the decimal digit keys have been depressed or following a terminated number), the number 1 is entered into the x-register. For example, the number 1×10^{16} may be entered by depressing the ENTER EXP, 1, and 6 keys.

The CHG SIGN key is depressed to changed the sign of any terminated or unterminated number in the x-register. If the CHG SIGN key is depressed as the first key of a numeric data entry, it changes the sign of the number then in the x-register (whatever the sign may be) and, once that number is replaced by the first digit of the new data entry, it prefaces the new data entry with a negative sign.

The CHG SIGN key is also depressed immediately following the ENTER EXP key (or the last-entered digit of the exponent) to enter negative exponents into the x-register.

The π key is depressed to enter the value of π (i.e. 3.14159265360) into the x-register.

The working register control keys (↑, ↓, ROLL ↑, $x \rightleftarrows y$) are used to reposition the contents of the x-, y-, and z-registers. They do not affect any other registers.

When the ↑ key is depressed, the contents of the y-register shift to the z-register and the contents of the x-register appear in both the x- and y-registers. The contents of the z-register are lost.

When the ↓ key is depressed, the contents of the y-register shift to the x-register and the contents of the z-register appear in both the y- and z-registers. The contents of the x-register are lost.

When the ROLL ↑ key is depressed, the contents of the x-register shift to the y-register, the contents of the y-register shift to the z-register, and the contents of the z-register shift to the x-register. No information is lost.

The $x \rightleftarrows y$ key is depressed to exchange the contents of the x- and y-registers. The contents of the z-register are unaffected by this operation.

The four arithmetic keys (+, −, ×, ÷) are used to perform working register arithmetic operations in which the contents of the x- and y-registers are employed as operands. The results of these arithmetic operations are stored in the y-register and the contents of the x-register remain unchanged by the arithmetic operations performed. These four keys do not affect any other registers.

The + key is depressed to add the number in the x-register to the number in the y-register, the sum appearing in the y-register.

The − key is depressed to subtract the number in the x-register from the number in the y-register, the difference appearing in the y-register.

The × key is depressed to multiply the number in the y-register by the number in the x-register, the product appearing in the y-register.

The ÷ key is depressed to divide the number in the y-register by the number in the x-register, the quotient appearing in the y-register.

The unary function keys ($\sqrt{x}$, x^2, 1/x, int x) are used to perform unary functions in which the contents of the x-register are employed as the argument. The results of the unary functions performed are placed in the x-register. These five keys do not affect any other registers.

The $\sqrt{x}$ key is depressed to calculate the square root of the number in the x-register. The x^2 key is depressed to calculate the square of the number in the x-register. The 1/x key is depressed to calculate the reciprocal of the number in the x-register. The int x key is depressed to truncate the fractional part of the number in the x-register. It does not affect the sign of the integer part of the number.

The data-storage and register-transfer keys (a, b, $x \rightarrow ()$, $y \rightarrow ()$, $x \leftarrow ()$, $y \rightleftarrows ()$, INDIRECT) keys are variously used to perform direct data storage and recall, direct storage-register arithmetic, indirect data storage and recall, and indirect storage-register arithmetic operations. The

a and b keys are depressed following any of the remaining five keys of this group to specify the a- and b-registers, respectively. The a and b keys may also be used to directly recall the contents of the a- and b-registers, respectively, to the x-register without changing the contents of the a- and b-registers themselves. Either of these functions of the a and b keys may be performed in response to a single keystroke thereof. For example, the contents of the b-register may be directly recalled to the x-register by simply depressing the b key alone. As noted above the contents of the b-register itself will remain unchanged by this recall operation.

All of the previously-described keys except the RUN, PRGM, KEY LOG, and LIST keys may be employed both for controlling the operation of the calculator during the manual operating and first key-log printing modes and for entering program steps into the calculator during the program entering the second key-log printing modes. When the calculator is in the manual operating mode, the user may continuously observe the output display of the contents of the x-, y-, and z-registers and, in accordance with his observations, make his own decisions about what to do next at any stage of the calculation. However, this is not possible when the calculator is executing a stored program at high speed in the automatic operating mode. The eight program-control keys

$$\left(\text{GO TO, IF } x<y, \text{ IF } x=y, \text{ IF } x>y, \text{ IF FLAG, SET FLAG, LABEL, } \frac{\text{SUB}}{\text{RETURN}}\right)$$

have therefore been provided to permit the calculator itself to test calculated quantities and make decisions based on those tests during the automatic execution of an internally-stored program. These eight keys may be used to permit unconditional branching (or transfers), conditional branching (or transfers), symbollic or labelled branching storage of "yes-no" information in a flag and conditional branching based on that "yes-no" information at a later stage in the execution of a program, and unconditional or conditional branches to pre-defined program routines with return to the main program sequence upon completion.

The STEP PRGM key is not programmable and is used, from the keyboard only, to single-step programs. When the calculator is in the manual operating mode the STEP PRGM key single-steps program execution. Each time STEP PRGM is depressed the program counter is incremented by one so that one program step is executed. When the calculator is in the program mode the STEP PRGM key enables the program steps stored in the program storage section of the RWM to be viewed. Each time the STEP PRGM key is depressed, the program counter is incremented by one, so that the address and program step displayed in the y-register shifts to the z-register, those in the x-register shift to the y-register and the next higher address and program step appears in the x-register.

The BACK STEP key is not programmable and is used to decrement the user program counter by one each time it is pressed. This backs up the output display (i.e. does just the opposite of the STEP PRGM key). It should only be used in the program mode. If the STEP PRGM and BACK STEP keys are depressed alternately, the same program step will be executed repeatedly. This key is extremely useful in editing and checking programs and when used with the PRGM STEP key permits the user to advance either forward or backward through a stored program one step at a time.

The half keys A—O comprising the group of definable keys 91 enable the calculator to be tailored to the special needs of the user. Operation of these keys is defined by the various plug-in ROM modules 92 that may be used with the calculator. Without these ROM modules the definable keys 91 serve no function and accidently depressing them, or encountering them in the execution of a stored program, will result in a completely non-destructive no-operation.

The plug-in ROM modules 92 include the alpha ROM module mentioned above, a definable functions ROM module and a typewriter ROM module. Both the alpha ROM module and the typewriter ROM module redefine nearly all of the keys of the keyboard as well as defining the definable keys 91 themselves. The definable functions ROM module uniquely defines the definable keys alone and may be used at the same time as the alpha ROM module or the typewriter ROM module.

An overlay 192 is associated with the definable functions ROM module and is employed with the definable keys 91 to identify the functions performed thereby when the definable functions ROM module is plugged into the calculator. The overlay 192 comprises a thin metal template that fits over the definable keys 91 and latches into a recess around them. The graphics on the template visually complete the key shapes and indicate the key function. A small tab positioned just above the nameplate releases the template, which then pops up to grasp. Three holes 196 near the top edge of the template allow direct viewing of three

light-emitting diode indicator lights used to indicate various operating conditions associated with the routines implemented by the definable functions ROM module. When the overlay 192 and its associated ROM module are not in use they may be secured together by a pair of tabs provided on the ROM module.

The alpha ROM module redefines the keyboard input unit as indicated by the letters printed on the tops of the definable keys 91 and the letters and symbols printed on the front sides of most of the other keys to provide an "alpha keyboard" (see Figure 5) containing 54 character-entry keys, 5 operational keys, and 16 "non-essential" keys (these non-essential keys are either inoperative or duplicate other keys during the alpha mode). During the alpha mode, the key-log feature is deactivated (thus, any keys pressed are not logged).

The 54 character-entry keys include all of the English alphabetic characters A—Z, all of the decimal numbers 0—9, and all of the following symbols @, $\sqrt{}$, / (printed by the ÷ key), ×, −, +, π, →, „, ., =, $, ?, (,), %, ", and ≠. Depressing any of these keys during the alpha mode, will cause the alphameric character or symbol indicated thereby to be printed out in line-printer fashion. The output printer unit operates as a line printer in that each character is not immediately printed out, but rather an entire line (16 characters) is first stored and then printed out. The print-out occurs as the 16th character is entered.

The 5 operational keys include the FMT key, the STOP key, a SPACE key (normally the CONTINUE key), a $\frac{\text{CLEAR}}{\text{RETURN}}$ key (normally the CLEAR key), and a PAPER key 292.

During the alpha mode, the following operational keys are depressed to perform various printing operations. Depressing the FMT key twice redefines the keyboard to the alpha mode, after which character keys may be depressed. After the last character is entered, depressing the FMT key causes a line print, a line feed, and returns the keyboard to normal operation. (The output display is blanked during the alpha mode, although the contents of the x-, y-, and z-registers remain unchanged.) For example, the alphabet may be printed by sequentially depressing the RUN, STOP, FMT, FMT, A through Z, and FMT keys.

Depressing the SPACE key insets a blank space in the printed line (similar in operation to the space bar on the typewriter).

CLEAR

Depressing the ——— key causes a

RETURN

line print and advances the printer to the next line (i.e. like a typewriter carriage-return and line feed operation). The alpha mode remains set after this instruction. Successive CLEAR instructions will cause the printer to advance, without printing, one line for each instruction.

Depressing the STOP key terminates the alpha mode without a line print or line feed. Any characters entered but not printed will be erased when STOP is pressed. This instruction is not programmable and should not be used while programming alpha messages.

The PAPER key 292 is a manual paper advance control. This operation is not programmable.

The 16 "non-essential" keys include the

PRINT SUB

↓, $x \rightleftharpoons y$, ↑, ———, ———, END,

SPACE RETURN

BACK STEP

———, ———, FLOAT 294, FIX () 296,

STEP PRGM

RUN, PRGM, KEY LOG, LIST, LOAD 198, and RECORD 300 keys. These keys are not essential for alpha printing operations. The non-essential keys which are programmable duplicate the SPACE key, while most of the non-programmable keys are "locked-out" (i.e. not operational) during alpha printing operations. Pressing BACK STEP or STEP PRGM will cause 1 or 0, respectively, to be printed.

All of the keys of the keyboard input unit and their associated octal keycodes and mnemonics are listed in Table D below. Every key has only one keycode and only one mnemonic (if it has a mnemonic), regardless of how many different functions it may be used to perform. The keycodes for the FLOAT, FIX ( ), RUN, PRGM, KEY LOG, LIST, LOAD, RECORD, BACK STEP, and STEP PRGM keys are applied to the CPU in seven-bit binary form. Since these keycodes are used only to control the mode of the calculator, are not programmable, and are never printed out by the output printer unit, they have no associated mnemonic. All of the remaining keycodes are applied to the CPU in six-bit binary form, are programmable, and have a mnemonic that may be printed out by the output printer unit.

TABLE D.

KEY	KEY CODE	MNEMONIC	KEY	KEY CODE	MNEMONIC
0	00	0	GO TO	44	GTO
1	01	1	PRINT / SPACE	45	PNT
2	02	2	END	46	END
3	03	3	CONTINUE	47	CNT
4	04	4	IF x=y	50	X=Y
5	05	5	LABEL	51	LBL
6	06	6	IF x<y	52	X<Y
7	07	7	IF x>y	53	X>Y
8	10	8	SET FLAG	54	SFL
9	11	9	K	55	K
x^2	12	XSQ	π	56	π
a	13	a	PAUSE	57	PSE
b	14	b	E	60	E
G	15	G	C	61	C
F	16	F	A	62	A
1/x	17	1/X	D	63	D
CLEAR	20	CLR	int x	64	INT
	21		I	65	I
ROLL↑	22	RUP	B	66	B
x→()	23	XTO	x←()	67	XFR
y=()	24	YE	M	70	M
↓	25	DN	O	71	O
ENTER EXP	26	EEX	L	72	L
↑	27	UP	N	73	N
x=y	30	XEY	H	74	H
INDIRECT	31	IND	J	75	J
CHG SIGN	32	CHS	$\sqrt{x}$	76	$\sqrt{\ }$

TABLE D Continued

KEY	KEY CODE	MNEMONIC	KEY	KEY CODE	MNEMONIC
+	33	+	SUB / RETURN	77	S/R
−	34	−	RECORD	102	
÷	35	DIV	LOAD	103	
×	36	×	LIST	104	
CLEAR x	37	CLX	KEY LOG	105	
y→()	40	YTO	PRGM	106	
STOP	41	STP	RUN	107	
FMT	42	FMT	FIX ()	110	
IF FLAG	43	IFG	FLOAT	110	

When the definable functions ROM module 92 is plugged into the calculator, the user may employ the definable keys 91 to perform the redefined functions indentified by the associated overlay shown in Figure 6. These functions include:

a. Defining special subroutines that can be called by a single key:
b. Protecting such subroutines against accidental erasure;
c. Deleting one, or more, of such subroutines when desired; and
d. Deleting, inserting, or searching for program steps (or keycodes) in any stored program.

Defining a Function.

A user definable function can be any sequence of program steps, beginning with the DEFINE and F(i) keys (or commands), where F(i) represents one of the F1 thru F9 keys (or commands). The execution of such a sequence of program steps must be terminated with the F-RET key (or command) in the same manner as the SUB/RET key (or command) in the case of subroutines.

Example.

The following is a function which puts π in the x, y, and z registers.

Program Step Address	Key/Program Step
0000	DEFINE
0001	F1
0002	π
0003	↑
0004	↑
0005	F-RET

The above function is identified by the F1 key (or command). To execute this function simply press the F1 key in the RUN mode or insert an F1 command in a program at the place where it should be executed.

Example.

The following is a call on the defined function F1 from another program (the program here happens to be another defined function F2).

Program Step Address	Key/Program Step
0000	DEFINE
0001	F2
0002	CLR
0003	F1
0004	F-RET

Notice that the same program step addresses have been assigned to both the defined functions F1 and F2. Every function definition is assumed to start at program step address 0000. All GO TO commands when present in a function are coded accordingly. This allows the user to define functions and delete them in a way independent of where they actually reside in the calculator memory. The DEFINE key (or command) in both of the above functions serves to mark the start of the definition of a function in the calculator memory. Actuation of the DEFINE key during the RUN mode is treated as an error. Attempted execution of the DEFINE command in a program is also treated as an error.

Example.

Program Step Address	Key/Program Step
0000	DEFINE
0001	F3
0002	CLR
0003	DEFINE←
0004	F2

The DEFINE command at program step address 0003 will be treated as an error when encountered during the execution of F3. The user here forgot to terminate the execution of F3 with F-RET. This error will cause the calculator to stop execution and turn on the STATUS light (the left-most indicator light 19 shown in Figure 1).

Protecting User Defined Functions.

The protection feature allows the user to designate areas of memory to be a part of the calculator's executive system.

i. Protecting A Single Function.

Assume that the definition of the function F1 has just been entered in memory (the same as function F1 discussed earlier). After the last program step in the definition, namely F-RET, has been entered, the program counter (y display) will point to program step address 0006. Program step address 0005, which contains F-RET, will be displayed in the z register. Switch to the RUN mode. Press the PROTECT key. The light above the PROTECT key (light 2 in Figure 6) will come on. The definition of the function F1 is now protected up to and including the F-RET program step. When the calculator is switched back to the PROGRAM mode, whatever was in program step address 0006 before will appear now to be at program step address 0000. All the program steps used in defining the function F1 have become "invisible". For all purposes these program steps are now a part of the calculator system. The function F1 can still be executed by pressing the corresponding key in the RUN mode or by giving an F1 command in another stored program or function.

ii. Protecting Several User Defined Functions.

Assume now that both the functions F1 and F2, described earlier, are to be protected. After protecting the function F1, switch to the PROGRAM mode and enter the definition of the function F2, namely:

Program Step Address	Key/Program Step
0000	DEFINE
0001	F2
0002	Clr
0003	F1
0004	F-RET

After entering the definition of the function F2 switch to the RUN mode. Press the DELETE and PROTECT keys. The protect light indicator (light 2 in Figure 6) will go off. Press the PROTECT key. The definitions of F1 and F2 are now protected.

iii. Effect of Protection on User Memory.

When protecting a certain number of program steps the calculator subtracts that number from the total number of program step locations available in user memory. The user can compute the number of program step locations available at any time as follows. In the RUN mode press the END, DELETE, and PROTECT keys. Then switch to the PROGRAM mode and

read the program step number in the y display (assume this number is p). The number of program step locations available to the user is given by the simple formula:

Available Program Step Locations = n − p

The variable n in the above formula is 500, 1012 or 2036 according to the amount of user memory included in the calculator. After finding the value of p, as illustrated above, the user can switch back to the RUN mode and press the PROJECT key again to re-establish the protection.

Deleting User Defined Functions.

To delete a given function simply press the DELETE key followed by the designating key (e.g. F1) of the function to be deleted. The DELETE key only applies to the unprotected area of user memory. This prevents accidental erasure of protected functions. Pressing the DELETE key causes the Insert/Delete light (light in Figure 6) to come on indicating that a memory modification operation is to be performed. If after pressing the DELETE key you find that you did not want to delete after all, you can press the CLEAR key to abort the delete operation. Pressing the CLEAR key at this time will cause the Insert/Delete light to go off. If the function to be deleted does not exist, the error light will come on and the delete operation will be aborted. The DELETE key must be pressed while the calculator is in the RUN mode. The DELETE key may be followed by a function designating key (F1 thru F9), the CLEAR key, or a digit key. Other keys following the DELETE key will turn on the error light and abort the delete operation. If encountered during the execution of a stored program, a DELETE command will cause the calculator to stop execution and turn on the error indicator.

The delete operation is designed to delete a portion of user memory which starts with the DEFINE command of the function to be deleted and ends either with the program step prior to the DEFINE command of the next function in memory or an END program step. To illustrate, consider the following memory map.

Program Step Address	Key/Program Step
0000	DEFINE←
0001	F1
0002	CLR
0003	F-RET←
0004	DEFINE
0005	F2
0006	EEX
0007	1
0008	-
0009	IF X=Y
0010	F-RET
0011	CONT
0012	CONT
0013	CONT
0014	GO TO
0015	4
0016	END
0017	...

When pressing the DELETE and F1 keys, the calculator will delete the memory portion delimited by the two arrows in the memory map. After this deletion, the DEFINE command at program step address 0004 will move to program step address 0000. The following program steps will move accordingly. Since the last function in a user memory is not likely to be followed by another DEFINE command, it is advisable to terminate that function with the program step END as shown in the memory map (program step address 0016).

If the last function in memory is not terminated with an END program step, then by deleting that function the user will also delete that portion of memory which follows that function and terminates with an END program step. When deleting the last function in memory, the delete operation does not delete the END program step. Thus, deleting F2 from the memory map shown earlier will result in the following memory map:

Program Step Address	Key/Program Step
0000	DEFINE
0001	F1
0002	CLR
0003	F-RET
0004	END

The program step END now becomes the terminator to the definition of the function F1.

General Editing Capabilities.

In addition to deleting functions the user of the definable functions ROM module has the capability of deleting, inserting, and searching for individual program steps in memory.

i. Deleting Individual Program Steps.

If the DELETE key is followed by a four digit address (see paragraph on automatic address termination for further details on address length), the calculator will delete the Program Step contained in that address and move all the following program steps accordingly.

Example.

Program Step Address	Key/Program Step
0000	CLR
0001	GO TO
0002	0
0003	5
0004	END
0005	1
0006	...

To delete the program step at program step address 0002 press the DELETE key in the RUN mode (the Insert/Delete light will come on), followed by the digit keys *0 0 0 2*. The resulting memory map will be:

Program Step Address	Key/Program Step
0000	CLR
0001	GO TO
0002	5
0003	END
0004	1
0005	...
0006	...

Notice that the program step previously at program step address 0003 is now at program step address 0002. All the program steps following the deleted program step up to the last program step in memory have been moved accordingly. Automatic Address Termination and Aborting a Delete: In the above example a four digit address was used to specify the program step to be deleted. The user can use a three, two, or one digit address if such an address is terminated by a non-numeric key other than CLEAR. The following delete commands are equivalent to that given in the last example:

DELETE	2	*STOP*
DELETE	02	*STOP*
DELETE	002	*STOP*

To abort a delete command before it is executed press the CLEAR key. The Insert/Delete light will then go off and the delete command will be cancelled. The CLEAR key can be pressed immediately after the DELETE key or during the entry of the address digits. Notice that the fourth digit in an address terminates that address. This digit signals to the calculator that the delete operation is to be performed. A CLEAR key following the fourth digit of an address will have no effect on a delete operation. After performing a delete operation the calculator switches off the Insert/Delete light. If the delete address is larger than 2035 the status light will come on and the delete operation will be aborted. This number represents the maximum available program step locations in the calculator.

ii. Inserting Individual Keystrokes.

The definable function block also gives the user the capability of inserting program steps at any arbitrary location in a program.

Example.

Consider the following memory map:

Program Step Address	Key/Program Step
0000	CLR
0001	GO TO ←
0002	9
0003	END
0004	...

To insert a program step before the program step at program step address 0002, press the INSERT key in the RUN mode (the Insert/Delete light will come on) followed by the digit keys *0 0 0 2*. Switch to the PROGRAM mode. The y display will point to program step address 0002 at which location a CONTINUE command is inserted by the calculator. The program step originally at program step address 0002 has now been moved to program step address 0003. All the program steps which follow this program step have been moved accordingly. To put a new program step at program step address 0002, simply press the new key while the calculator is still in the PROGRAM mode. The key pressed will replace the CONTINUE command inserted earlier by the calculator. Without replacing the CONTINUE command at program step address 0002 the new memory map is as shown below:

Program Step Address	Key/Program Step
0000	CLR
0001	GO TO
0002	CONT
0003	9
0004	END
0005	...

The address in an INSERT command may be automatically terminated in the same manner described earlier in connection with the DELETE command. The insert operation can also be aborted in the same fashion.

Notice that the insert operation effects the content of all the program steps following the one designated in that operation.

iii. Searching for a Given Program Step in Memory.

The ability to search for a given program step in memory is provided by the FIND key. By pressing the FIND key in the RUN mode followed by any other key on the keyboard, the user commands the calculator to search the memory for the program step designated by the last-mentioned key starting at the address currently designated by the program counter. If the designated program step is not found, the status light will come on. However, if the designated program step is found, the calculator will automatically switch to the PROGRAM mode, and the z register will contain the program step address at which the designated program step was found.

Example.

Consider protecting a function definition area in memory. This area is terminated with the program step END. Reset the program counter to 0000 by pressing the GO TO and 0 keys or the END key in the RUN mode. Now press the FIND and END keys. The z register will now contain the address of the terminating program step END in the function definition area. Press the RUN and PROTECT keys. The function definition area is now protected up to and including that terminating END program step.

We have found that the above-described electronic programmable calculator according to the invention is capable of editing programs stored within the calculator more efficiently than conventional programmable calculators. Furthermore a user of the calculator may single step either forward or backward through an internally stored program in a program entering mode to check the program and may single step forward through the program in a program-controlled operating mode to check the program execution. With the programmable calculator the last program step encountered, the program step presently encountered, and the next program step to be encountered, while single stepping either forward or backward through a program stored within the calculator, may be displayed. Additionally a user may insert program steps and automatically locate every occurrence of a designated program step in a program stored within the calculator.

WHAT WE CLAIM IS:—

1. An electronic calculator comprising keyboard input means for entering information into the calculator, first memory means into which information may be written and from which information may be read, second memory means for storing routines and subroutines to be executed by the calculator in performing selected functions, processing means responsive to information from the keyboard input means or the first memory means and to operating states within the calculator itself for selectively executing one or more of the routines and subroutines stored in the second memory means to perform selected functions employing information from one or both of the keyboard input and first memory means, insertion means for inserting a program step at a designated location within a series of program steps stored in the first memory means, and output means for providing an output indication of the functions performed.

2. An electronic calculator as in claim 1 wherein said insertion means includes an insert key of the keyboard input means, and logic means responsive to actuation of the insert key for conditioning the calculator to insert a program step at a designated location within a series of program steps stored in the first memory means.

3. An electronic calculator as in claim 2 wherein said logic means is further responsive to actuation of the insert key and one or more address designating keys for causing the calculator to insert a program step at the address designated by those address designating keys and for incrementing the address of each program step following the program step to be inserted.

4. An electronic calculator as in any one of the preceding claims 1 to 3 including deletion means for deleting a program step at a designated location within a series of program steps stored in the first memory means.

5. An electronic calculator as in claim 4 wherein said deletion means includes a delete key of the keyboard input means, and logic means responsive to actuation of the delete key for conditioning the calculator to delete a program step at a designated location within a series of program steps stored in the first memory means.

6. An electronic calculator as in claim 5 wherein said logic means is further responsive to actuation of the delete key and one or more address designating keys for causing the calculator to delete the program step at the address designated by those address designating keys and for decrementing the address of each program step following the program step to be deleted.

7. An electronic calculator as in any one of the preceding claims 1 to 3 including deletion means for deleting a user defined function stored in the first memory means.

8. An electronic calculator as in claim 7 wherein said deletion means includes a delete key of the keyboard input means, and logic means responsive to actuation of the delete key for conditioning the calculator to delete a designated user defined function stored in the first memory means.

9. An electronic calculator as in claim 8 wherein said logic means is further responsive to actuation of the delete key and one or more function designating keys for causing the calculator to delete the user defined function designated by those function designating keys.

10. An electronic calculator as in any one of the preceding claims 1 to 9 wherein said output means is an output display unit, and said calculator includes location means for causing the calculator to locate and display an indication of each occurrence and address of a specified program step within a series of program steps stored in the first memory means.

11. An electronic calculator as in claim 10 wherein said location means includes a find key of the keyboard input means, and logic means responsive to actuation of the find key for conditioning the calculator to locate and display in indication of each occurrence and address of a specified program step within a series of program steps stored in the first memory means.

12. An electronic calculator as in claim 11 wherein said logic means is further responsive to actuation of the find key and a program step designating key for locating and displaying an indication of each occurrence and address of the program step designated by that program step designating key.

13. An electronic calculator as in any one of the preceding claims 1 through 12 including backstep means operable, when the calculator is in a program checking mode, for stepping backward through the steps of a program stored in the first memory means.

14. An electronic calculator as in claim 13 wherein said backstep key of the keyboard input means, and logic means responsive to actuation of the backstep key for causing the calculator to step backward through the steps of a program stored in the first memory means.

15. An electronic calculator as in claim 14 wherein said logic means is further responsive to actuation of the backstep key, when the calculator is in the program

* C L M S *

checking mode, for decrementing the contents of a program step pointer located in the first memory means and thereby stepping backward through the steps of a program stored in the first memory means.

16. An electronic calculator as in any one of the preceding claims 1 to 9 including program indicating means operable when the calculator is in a program checking mode, for simultaneously indicating the program step presently encountered, the program step last encountered, the program step next-to-be encountered, and the addresses thereof.

17. An electronic calculator as in claim 16 wherein said first memory means includes at least three display storage registers, said output means is an output display unit for displaying the contents of those display storage registers, said calculator includes a program control switch for setting the calculator to the program checking mode and a program step pointer for sequentially designating the address of each program step within a program stored in the first memory means, and said program indicating means is operable, when the calculator is in the program checking mode, for placing an indication of the address and program step then designated by the program step pointer in one of the display storage registers, for placing an indication of the next preceding address and program step in another of the display storage registers, and for placing an indication of the next preceding address and program step in another of the display storage registers.

18. An electronic 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, WC2B 6QX.

Printed for Her Majesty's Stationery Office by the Courier Press, Leamington Spa, 1975.
Published by the Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.

* C L M S *