PATENT SPECIFICATION

(11) 1 403 703

(21) Application No. 50950/74 (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 17BX 17P 3W10W2 3W10Y 5A 5B ND

1 403 703

(19)



(54) ELECTRONIC CALCULATOR

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

This application is concerned with electronic calculators and is especially concerned with calculators of the type disclosed in our co-pending U.K. Patent applications Nos. 28012/72, 51661/74, 54042/74, 54012/74, 54011/74, 54010/74, and 54009/74 (Serial Nos. 1,403,710, 1,403,709, 1,403,706, 1,403,705, 1,403,707, 1,403,704 and 1,403.708).

In the complete specification of our co-pending U.K. Patent application No. 28012/72 (Serial No. 1,403,710) 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. 51661/74 (Serial No. 1,403,709) 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 of 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. 54042/74 (Serial No. 1,403,706) 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, 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 the complete specification of our co-pending U.K. application No. 54012/74 (Serial No. 1,403,705) 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 mey 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. 1,403,707) 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 magnetic record member associated with the calculator into the first memory means, first control means for designating any program 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. 1,403,704) there is disclosed a programmable 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. 1,403,708) 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

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.

Conventional electronic calculators typically cannot perform indirectly addressed numeric data register transfers and arithmetic without utilizing available working registers for addresses rather than data. This seriously limits their ability to efficiently perform complex operations such as file manipulations or matrix arithmetic. Moreover, they typically have a very limited capability for performing direct arithmetic between working and storage memory registers and little or no capability for performing indirect arithmetic between working and storage memory registers.

Accordingly one object of the present invention is the provision of an electronic calculator capable of employing extensive indirect addressing to permit efficient manipulation of files and matrix operations. Another object of the present invention is the provision of an electronic calculator capable of performing both direct and indirect storage, recall, and exchange between working and storage memory registers. Still another object of the present invention is the provision of a programmable calculator capable of performing both direct and indirect arithmetic bidirectionally between working and storage memory registers.

The present invention provides 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, and an output unit for providing an output indication of the functions performed, the first memory unit including a plurality of working registers and a plurality of storage registers, and the 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 a calculator set forth in the last preceding paragraph preferably the keyboard input unit includes one or more working register transfer keys each operable with one or more storage register designating keys for causing the calculator to directly store the contents of a specified one of the working registers in any one of the storage registers specified by those storage register designating keys, and further includes an indirect control key operable with each of the working register transfer keys and one or more storage register designating keys for causing the calculator to indirectly store the contents of a specified one of the working registers in any one of the storage registers specified by the contents of any other one of the storage registers specified by those storage register designating keys.

In a calculator as set forth in the last preceding paragraph preferably the contents of the specified one of the working registers and the last-mentioned one of the storage registers remain unchanged.

In a calculator as set forth in any one of the last two preceding paragraphs preferably the indirect control key is multiply operable with each of the working register transfer keys and one or more storage register designating keys for causing the calculator to indirectly store the contents of a specified one of the working registers in any one of the storage registers multiply indirectly specified by the contents of others of the storage registers a first of which is specified by those storage register designating keys.

In a calculator as set forth in any one of the last three preceding paragraphs preferably the keyboard input unit includes one or more working register transfer keys each operable with one or more storage register designating keys for causing the calculator to directly recall the contents of any one of the storage registers specified by those storage register designating keys into a specified one of the working registers, and the indirect control key is operable with each of the last-mentioned working register transfer keys and one or more storage register designating keys for causing the calculator to indirectly recall the contents of any one of the storage registers specified by the contents of any other one of the storage registers specified by those storage register designating keys into a specified one of the working registers.

In a calculator as set forth in the last preceding paragraph preferably the contents of all the mentioned storage registers remain unchanged.

In a calculator as set forth in any one of the last two preceding paragraphs preferably the indirect control key is

operable with each of the last-mentioned working register transfer keys and one or more storage register designating keys for causing the calculator to indirectly recall the contents of any one of the storage registers multiply indirectly specified by the contents of others of the storage registers a first of which is specified by those storage registers designating keys into a specified one of the working registers.

In a calculator as set forth in any one of the last six preceding paragraphs preferably the keyboard input unit includes a working register transfer key operable with one or more storage register designating keys for causing the calculator to directly exchage the contents of a specified one of the working registers with the contents of any one of the storage registers specified by those storage register designating keys, and the indirect control key is operable with the last-mentioned working register transfer key and one or more storage register designating keys for causing the calculator to indirectly exchange the contents of a specified one of the working registers with the contents of any one of the storage registers specified by the contents of any other one of the storage registers specified by those storage register designating keys.

In a calculator as set forth in the last preceding paragraph preferably the contents of the last-mentioned one of the storage registers remains unchanged.

In a calculator as set forth in any one of the last two preceding paragraphs preferably the indirect control key is operable with the last-mentioned working register transfer key and one or more storage register designating keys for causing the calculator to indirectly exchange the contents of a specified one of the working registers with the contents of any one of the storage registers multiply indirectly specified by the contents of others of the storage registers a first of which is specified by those storage register designating keys.

In a calculator as set forth in any one of the last nine preceding paragraphs preferably the keyboard input unit includes a plurality of arithmetic operator keys each operable for conditioning the calculator to perform a specified arithmetic operation, each of the working register transfer keys is operable with each of the arithmetic operator keys in combination with one or more storage register designating keys for causing the calculator to perform a specified arithmetic operation employing the contents of a specified one of the working registers and the contents of any one of the storage registers specified by those storage register designating keys, and the indirect control key is operable with each of the working register transfer keys in combination with any one of the arithmetic operator keys and one or more storage register designating keys for causing the calculator to perform a specified arithmetic operation employing the contents of a specified one of the working registers and the contents of any one of the storage registers specified by the contents of any other one of the storage registers specified by those storage register designating keys.

In a calculator as set forth in the last preceding paragraph preferably the result of the specified arithmetic operation is stored in one of the registers whose contents are employed in the specified arithmetic operation without changing the contents of the other registers mentioned.

In a calculator as set forth in any one of the last two preceding paragraphs preferably the direct control key is operable with each of the working register transfer keys in combination with any one of the arithmetic operator keys and one or more storage register designating keys for causing the calculator to perform a specified arithmetic operation employing the contents of a specified one of the working registers and the contents of any one of the storage registers multiply indirectly specified by the contents of others of the storage registers a first of which is specified by the storage register designating keys.

There now follows a detailed description which is to be read with reference to the accompanying drawings of an electronic 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 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; and

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

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 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 $\overline{S}$-bus 50, a T-bus 52, and an $\overline{R}$-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 $\overline{S}$-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 microinstructions $\overline{PTR}$ 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 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 plut-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 Figures 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 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 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 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 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 ALU116 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 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 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 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 information may be transferred bit-serially either from ALU 116 via T-but 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 or 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 $\overline{CCO}$ 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.

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 change 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 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 register.

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 data-storage and register-transfer keys (a, b, x→(), y→(), x←(), y⇄(), 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. As illustrated below, 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.

The x→(), y→(), x←(), and y⇄() keys are used to control the transfer of numeric data from the x- and y-registers to the a- and b-registers and 49 additional storage registers available to the user (at user addresses 000—048) in the data storage section of the RWM and from these storage registers to the x- and y-registers. If the 60 optional storage registers included in the data storage section of the RWM (at user addresses 049—108) are made available to the user, these same four keys may also be used to control the transfer of numeric data from the x- and y-registers to these optional storage registers and from them to the x- and y-registers.

The specific storage register to or from which numeric data is transferred by these four keys is specified by the key or keys depressed immediately following them. Accordingly, the a key is used to specify the a-register, the b key to specify the b-register, and the decimal digit keys 0—9 to selectively specify any of the available storage registers at user addresses 000—108 of the data storage section of the RWM. For simplicity of description, the available storage registers at user addresses 000—108 will hereinafter be referred to by their addresses (i.e the storage register at any available address nnn will be referred to a register nnn). If a non-existant or non-available storage register is designated, the STATUS indicator light is turned on, no data transfer or arithmetic operation is performed, and the operation of the calculator halts. (If such a register is designated while the calculator is automatically executing a stored program, the program-counter specifies the program step immediately following the improper operation).

The x→() key is depressed followed by the a key, the b key, or decimal digit keys, n, n, n to directly store the contents of the x-register in the a-register, the b-register, or register nnn respectively. In any case, the contents of the x-register remain unchanged by this operation. For example, π may be stored directly in the b register by sequentially depressing the π, x→(), and b keys. Similarly, π may be stored directly in register 027, by sequentially depressing the π, x→(), 0, 2, and 7 keys. In each of these examples, π will also remain in the x-register.

The y→() key is depressed followed by the a key, the b key, or decimal digit keys n, n, n to directly store the contents of the y-register in the a-register, the b-register, or register nnn, respectively. In any case, the contents of the y-register remain unchanged by this operation. For example, a number contained in the y-register may be stored directly in register 000, without changing the contents of the y-register, by sequentially depressing the y→(), 0, 0, and 0 keys.

The x←() key is depressed followed by the a key, the b key, or decimal digit keys n, n, n, to directly recall the contents of the a-register, the b-register, or register nnn, respectively, to the x-register. In any case, the contents of the recalled register remain unchanged by this operation. For example, the contents of register 012 may be recalled to the x-register, without changing the contents of register 012, by sequentially depressing the x←(), 0, 1, and 2 keys. Recall from the a- or b-register to the x-register may be accomplished in the same manner as described above, by simply depressing the a or b key alone.

The y⇄() key is depressed followed by the a key, the b key, or decimal digit keys n, n, n, to exchange the contents of the y-register with the contents of the a-register, the b-register, or register nnn, respectively. For example, a number in the y-register may be exchanged with a number in register 048 by depressing the y⇄(), 0, 4, and 8 keys.

The direct storage and recall operations performed by the x→(), y→(), x←(), and y=() keys may be conveniently summarized by employing the following notation:

$$\left\{\begin{array}{l} x\rightarrow() \\ y\rightarrow() \\ x\leftarrow() \\ y=() \end{array}\right\} \qquad \left\{\begin{array}{l} a \\ b \\ nnn \end{array}\right\}$$

In this notation each pair of braces implies that any one of the enclosed group of key operations or storage registers may be selected. The order of the successive pairs of braces from left to right specifies the key-sequence required to perform the selected key operation with the selected storage register. Thus, any of the key operations enclosed in the left-hand pair of braces may be employed with any of the storage registers enclosed in the right hand pair of braces by first depressing the key performing the selected key operation and then depressing the key or keys designating the selected storage register. The usefulness of these direct storage and recall operations is illustrated by the following method of multiplying a series of numbers n_1, n_2, etc. by a constant K, where $n_1 = 3$, $n_2 = 11.2$, etc. and $K = 1.684$:

KEYS DEPRESSED	CONTENTS OF x-REGISTER	CONTENTS OF y-REGISTER	CONTENTS OF a-REGISTER
CLEAR	0	0	0
1, ., 6, 8, 4,	1.684	0	0
y=(), a	1.684	0	1.684
3	3	0	1.684
↑	3	3	1.684
a	1.684	3	1.684
x	1.684	5.0520	1.684
1, 1, ., 2	11.2	5.0520	1.684
↑	11.2	11.2	1.684
a	1.684	11.2	1.684
x	1.684	18.8608	1.684

The answers, appearing in the y-register, are underlined.

The x→(), y→(), x←(), and y=() keys may also be employed with the +, −, ×, and ÷ arithmetic keys to perform direct register-arithmetic operations in which the contents of the x-register or the y-register are employed as one operand and the contents of the a-register, the b-register, or register nnn are employed as the other operand. These direct register-arithmetic operations may be summarized as follows by using the above-described notation:

$$\left\{\begin{array}{l} x\rightarrow() \\ y\rightarrow() \\ x\leftarrow() \\ y=() \end{array}\right\} \qquad \left\{\begin{array}{l} + \\ - \\ \times \\ \div \end{array}\right\} \qquad \left\{\begin{array}{l} a \\ b \\ nnn \end{array}\right\}$$

Thus, the x→() key may be employed to directly perform any of the arithmetic operations enclosed by the second pair of braces upon the contents of the x-register and the contents of any selected one of the storage registers enclosed by the third pair of braces and store the result in the selected storage register without recalling the contents of the selected storage register and without changing the contents of the x-register. For example, the contents of the x-register may be subtracted from the contents of the b-register by sequentially depressing the x→(), −, and b keys. The difference is stored in the b-register, and the contents of the x-register remain unchanged by the operation. Similarly, the contents of register 042 may be divided by the contents of the x-register by sequentially depressing the x→(), ÷, 0, 4, and 2 keys. The quotient is stored in register 042, and the contents of

the x-register itself remain unchanged by the operation.

The y→() key may be similarly employed to directly perform any of the arithmetic operations enclosed by the second pair of braces upon the contents of the y-register and any of the storage registers enclosed by the third pair of braces and store the result in the selected storage register without recalling the contents of the selected storage register and without changing the contents of the y-register. For example, the contents of the y-register may be added to the contents of the a-register by sequentially depressing the y→(), +, and a keys. The sum is stored in the a-register, and the contents of the y-register remain unchanged by the operation. Similarly, the contents of register 038 may be multiplied by the contents of the y-register by sequentially depressing the y→(), x, 0, 3, and 8 keys. The product is stored in register 038, and the contents of the y-register remain unchanged by the operation.

The x←() key may be employed to directly perform any of the arithmetic operations enclosed by the second pair of braces upon the contents of the x-register and the contents of any of the storage registers enclosed by the third pair of braces and store the result in the x-register without changing the contents of selected storage register. For example, the contents of the a-register may be added to the contents of the x-register by sequentially depressing the x←(), +, and a keys. The sum is stored in the x-register, and the contents of the a-register remain unchanged by the operation. Similarly, the contents of the x-register may be multiplied by the contents of register 022 by sequentially depressing the x←(), x, 0, 2, and 2 keys. The product is stored in the x-register, and the contents of register 022 remain unchanged by the operation.

The y⇄() key may be employed to directly perform any of the arithmetic operations enclosed by the second pair of braces upon the contents of the y-register and the contents of any of the storage registers enclosed by the third pair of braces and store the result in the y-register without changing the contents of the selected storage register (i.e. the y⇄() may be used as though it were a y←() key in performing register-arithmetic operations).

For example, the contents of the b-register may be subtracted from the contents of the y-register by sequentially depressing the y⇄(),–, and b keys. The difference is stored in the y-register, and the contents of the b-register remain unchanged by the operation. Similarly, the contents of the y-register may be divided by the contents of register 039, by sequentially depressing the y⇄(), ÷, 0, 3, and 9 keys. The quotient is stored in the y-register, and the contents of register 039 remains unchanged by the operation.

The INDIRECT key is used with the above-described data-storage and register-transfer keys to perform indirect data-storage and recall and indirect register-arithmetic operations, in which the contents of a directly-addressed storage register (e.g. the a-register, the b-register or any of the registers nnn) are employed as the address of an indirectly-addressed storage register (e.g. any of the other storage registers nnn) to be used in these operations. When indirectly addressing any of the storage registers 000 through 048 or 108, care must be taken to insure that the directly-addressed storage register contains the proper address nnn of the indirectly-addressed storage register nnn. The indirect address used in the absolute value of the integer part of the contents of the directly-addressed storage register. Thus, 1.732 will be treated as the address of register 001, —6.99 as the address of register 006, 0.999 as the address of register 000, and 106,75 as the address of register 106. Since the storage registers may only contain numeric data, the a- and b-registers may not be used as the indirectly-addressed storage register in these operations.

The indirect data storage and recall operations may be summarized as follows by again using the above-described notation without braces for the INDIRECT key:

$$\left\{\begin{matrix} x\rightarrow() \\ y\rightarrow() \\ x\leftarrow() \\ y\rightleftarrows() \end{matrix}\right\} \quad \text{INDIRECT} \quad \left\{\begin{matrix} a \\ b \\ nnn \end{matrix}\right\}$$

Thus, the x→() and y→() keys may be employed with the INDIRECT key to indirectly store the contents of the x- and y-registers, respectively in any storage register nnn designated by the contents of any of the other storage registers. Moreover, this may be accomplished without changing the contents of the x- and y-registers or of the directly addressed storage register. For example, the contents of the x-register may be indirectly stored in the storage register designated by the contents of the a-register by sequentially depressing the x→(), INDIRECT, and a keys. The contents of the x-register and the a-register remain unchanged by this operation. Similarly, the contents of the y-register may be indirectly stored in the storage register designated by the contents of register 002 by sequentially depressing the y→(), INDIRECT, 0, 2, and 2 keys. The

contents of the y-register and register 022 remain unchanged by this operation.

The x←() key may be similarly employed with the INDIRECT key to indirectly recall to the x-register the contents of any storage register nnn designated by the contents of any of the other storage registers. This is accomplished without changing the contents of either the directly-or indirectly-addressed storage register. For example, the contents of the register nnn designated by the contents of the b-register may be indirectly recalled to the x-register by sequentially depressing the x←(), INDIRECT, and b keys. The contents of the b register and of the indirectly-addressed register remain unchanged by this operation.

The y⇄() key may be employed with the INDIRECT key to indirectly exchange the contents of the y-register with the contents of any storage register nnn designated by the contents of any of the other storage registers. This is accomplished without changing the contents of the directly-addressed storage register. For example, the contents of the y-register may be indirectly exchanged with the contents of the register designated by register 041 by sequentially depressing the y⇄(), INDIRECT, 0, 4, and 1 keys. The contents of register 041 remain unchanged by this operation.

The indirect register-arithmetic operations may be summarized as follows:

$$\left\{\begin{matrix} x\rightarrow() \\ y\rightarrow() \\ x\leftarrow() \\ y\rightleftarrows() \end{matrix}\right\} \text{INDIRECT} \left\{\begin{matrix} + \\ - \\ \times \\ \div \end{matrix}\right\} \left\{\begin{matrix} a \\ b \\ nnn \end{matrix}\right\}$$

In connection with the indirect register arithmetic operations it should be noted that the INDIRECT key may also be depressed immediately after the selected arithmetic key. The indirect register-arithmetic operations may therefore also be summarized as follows:

$$\left\{\begin{matrix} x\rightarrow() \\ y\rightarrow() \\ x\leftarrow() \\ y\rightleftarrows() \end{matrix}\right\} \left\{\begin{matrix} + \\ - \\ \times \\ \div \end{matrix}\right\} \text{INDIRECT} \left\{\begin{matrix} a \\ b \\ nnn \end{matrix}\right\}$$

Thus, any of the x→(), y→(), x←(), and y⇄() keys may be employed with the INDIRECT key and any of the arithmetic keys to indirectly perform register-arithmetic operations employing the contents of either the x- or the y-register as one operand and the contents of any of the storage registers nnn designated by the contents of any of the storage registers as the other operand. In the case of the x→() and y→() keys, the results of these operations are stored in the indirectly addressed storage register nnn, and the contents of the x- and y-registers and the directly-addressed registers are not changed. Similarly, in the case or the x←() and y⇄() keys, the results of these operations are stored in the x- and y-registers, respectively, and the contents of the directly and indirectly addressed registers are not changed.

For example, the contents of the x-register may be added to the contents of the register nnn designated by the contents of the a-register by sequentially depressing the x→(), INDIRECT, +, and a keys. The sum is stored in the indirectly-addressed register nnn, and the contents of the x-register and the a-register remain unchanged by this operation. Similarly, the contents of the storage register nnn designated by the contents of register 014 may be multiplied by the contents of the y-register by sequentially depressing the y→(), x, INDIRECT, 0, 1, and 4 keys. The product is stored in the indirectly-addressed storage register nnn, and the contents of the y-register and register 014 remain unchanged by the operation. Similarly, the contents of the register nnn designated by the contents of the b-register may be subtracted from the contents of the x-register by sequentially depressing the x←(), INDIRECT, —, and b keys. The difference is stored in the x-register, and the contents of the b-register and the indirectly-addressed register remain unchanged by the operation. As a last example, the contents of the y-register may be divided by the contents of the register nnn designated by the contents of register 008 by sequentially depressing the y⇄(), INDIRECT, ÷, 0, 0, and 8 keys. The quotient is stored in the y-register, and the contents of register 008 and the indirectly-addressed register nnn remain unchanged by this operation.

Multiple-level indirect addressing may be performed to any desired level in the above described indirect data storage and recall and indirect register arithmetic operations by sequentially depressing the INDIRECT key one for each desired level. Multiple-level indirect data storage and recall may be summarized as follows:

$$\left\{\begin{array}{l} x\rightarrow() \\ y\rightarrow() \\ x\leftarrow() \\ y\rightleftarrows() \end{array}\right\} \text{ INDIRECT .... INDIRECT } \left\{\begin{array}{l} a \\ b \\ nnn \end{array}\right\}$$

Thus, for example, the contents of the x-register may be indirectly stored in the contents of register 017 designated by the contents of register 003 in turn designated by the contents of the a-register by sequentially depressing the x→(), INDIRECT, INDIRECT, and a keys. The contents of the x-register, the a-register, and register 003 remain unchanged by this operation. Similarly, the contents of register 046 designated by the contents of register 031 in turn designated by the contents of register 000 in turn designated by the contents of register 025 may be exchanged with the contents of the y-register by sequentially depressing the y⇄(), INDIRECT, INDIRECT, INDIRECT, 0, 2, and 5 keys. The contents of registers 025, 000 and 031, remain unchanged by this operation.

Multiple level indirect register-arithmetic may be similarly summarized as follows:

$$\left\{\begin{array}{l} x\rightarrow() \\ y\rightarrow() \\ x\leftarrow() \\ y\rightleftarrows() \end{array}\right\} \text{ INDIRECT .... INDIRECT } \left\{\begin{array}{l} + \\ - \\ \times \\ \div \end{array}\right\} \left\{\begin{array}{l} a \\ b \\ nnn \end{array}\right\}$$

or

$$\left\{\begin{array}{l} x\rightarrow() \\ y\rightarrow() \\ x\leftarrow() \\ y\rightleftarrows() \end{array}\right\} \left\{\begin{array}{l} + \\ - \\ \times \\ \div \end{array}\right\} \text{ INDIRECT .... INDIRECT } \left\{\begin{array}{l} a \\ b \\ nnn \end{array}\right\}$$

Thus, assuming, for example, that the 60 optional storage registers 049—108 are available to the user, the contents of the y-register may be subtracted from the contents of register 108 designated by the contents of register 082, in turn designated by the contents of register 049 by sequentially depressing the y→(), INDIRECT, INDIRECT, —, 0, 4, and 9 keys. The difference is stored in register 108 and the contents of the y-register, register 049, and register 082 remain unchanged by this operation. Similarly, the contents of the x-register may be divided by the contents of register 106 designated by the contents of register 056 in turn designated by the contents of the b-register by sequentially depressing the x←(), ÷, INDIRECT, INDIRECT, INDIRECT, and b keys. The results are stored in the x-register, and the contents of the b-register and registers 003, 056, and 106 remain unchanged by this operation.

As described above, each of the available numerically-addressed storage registers 000 through 048 or 108 employed in the foregoing data-storage and register-transfer operations is properly addressed by selectively depressing the decimal digit keys to specify its three-digit address. Upon entry of the third digit, the numeric address is automatically terminated. Any immediately following digit entries are therefore not interpreted as part of the numeric address, but rather as the beginning of a new data entry. Thus for example, if the x→(), 1, 0, 3, 2, and 5 keys are sequentially depressed, the contents of the x-register are stored in register 103, and the number 25 is entered into the x-register. Similarly, if the y→(), +, 0, 0, 2, 1, and + keys are sequentially depressed, the contents of the y-register are added to the contents of register 002, and the number 1 is entered into the x-register and thereupon added to the contents of the y-register.

WHAT WE CLAIM IS:—

1. An electronic calculator comprising a keyboard input unit for entering information into the calculator, a first

$$\left\{\begin{array}{l} x\rightarrow() \\ y\rightarrow() \\ x\leftarrow() \\ y=() \end{array}\right\} \text{INDIRECT} \ldots \text{INDIRECT} \left\{\begin{array}{l} a \\ b \\ nnn \end{array}\right\}$$

Thus, for example, the contents of the x-register may be indirectly stored in the contents of register 017 designated by the contents of register 003 in turn designated by the contents of the a-register by sequentially depressing the x→(), INDIRECT, INDIRECT, and a keys. The contents of the x-register, the a-register, and register 003 remain unchanged by this operation. Similarly, the contents of register 046 designated by the contents of register 031 in turn designated by the contents of register 000 in turn designated by the contents of register 025 may be exchanged with the contents of the y-register by sequentially depressing the y=(), INDIRECT, INDIRECT, INDIRECT, 0, 2, and 5 keys. The contents of registers 025, 000 and 031, remain unchanged by this operation.

Multiple level indirect register-arithmetic may be similarly summarized as follows:

$$\left\{\begin{array}{l} x\rightarrow() \\ y\rightarrow() \\ x\leftarrow() \\ y=() \end{array}\right\} \text{INDIRECT} \ldots \text{INDIRECT} \left\{\begin{array}{l} + \\ - \\ \times \\ \div \end{array}\right\} \left\{\begin{array}{l} a \\ b \\ nnn \end{array}\right\}$$

or

$$\left\{\begin{array}{l} x\rightarrow() \\ y\rightarrow() \\ x\leftarrow() \\ y=() \end{array}\right\} \left\{\begin{array}{l} + \\ - \\ \times \\ \div \end{array}\right\} \text{INDIRECT} \ldots \text{INDIRECT} \left\{\begin{array}{l} a \\ b \\ nnn \end{array}\right\}$$

Thus, assuming, for example, that the 60 optional storage registers 049—108 are available to the user, the contents of the y-register may be subtracted from the contents of register 108 designated by the contents of register 082, in turn designated by the contents of register 049 by sequentially depressing the y→(), INDIRECT, INDIRECT, —, 0, 4, and 9 keys. The difference is stored in register 108 and the contents of the y-register, register 049, and register 082 remain unchanged by this operation. Similarly, the contents of the x-register may be divided by the contents of register 106 designated by the contents of register 056 in turn designated by the contents of the b-register by sequentially depressing the x←(), ÷, INDIRECT, INDIRECT, INDIRECT, and b keys. The results are stored in the x-register, and the contents of the b-register and registers 003, 056, and 106 remain unchanged by this operation.

As described above, each of the available numerically-addressed storage registers 000 through 048 or 108 employed in the foregoing data-storage and register-transfer operations is properly addressed by selectively depressing the decimal digit keys to specify its three-digit address. Upon entry of the third digit, the numeric address is automatically terminated. Any immediately following digit entries are therefore not interpreted as part of the numeric address, but rather as the beginning of a new data entry. Thus for example, if the x→(), 1, 0, 3, 2, and 5 keys are sequentially depressed, the contents of the x-register are stored in register 103, and the number 25 is entered into the x-register. Similarly, if the y→(), +, 0, 0, 2, 1, and + keys are sequentially depressed, the contents of the y-register are added to the contents of register 002, and the number 1 is entered into the x-register and thereupon added to the contents of the y-register.

WHAT WE CLAIM IS:—

1. An electronic calculator comprising a keyboard input unit for entering information into the calculator, a first

* C L M S *

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, 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.

2. An electronic calculator as in claim 1 wherein said keyboard input unit includes one or more working register transfer keys each operable with one or more storage register designating keys for causing the calculator to directly store the contents of a specified one of the working registers in any one of the storage registers specified by those storage register designating keys, and further includes an indirect control key operable with each of the working register transfer keys and one or more storage register designating keys for causing the calculator to indirectly store the contents of a specified one of the working registers in any one of the storage registers specified by the contents of any other one of the storage registers specified by those storage register designated keys.

3. An electronic calculator as in claim 2 wherein the contents of the specified one of the working registers and the last-mentioned one of the storage registers remain unchanged.

4. An electronic calculator as in claim 2 or 3 wherein said indirect control key is multiply operable with each of the working register transfer keys and one or more storage register designating keys for causing the calculator to indirectly store the contents of a specified one of the working registers in any one of the storage registers multiply indirectly specified by the contents of others of the storage registers a first of which is specified by those storage register designating keys.

5. An electronic calculator as in any one of the preceding claims 2 through 4 wherein said keyboard input unit includes one or more working register transfer keys each operable with one or more storage register designating keys for causing the calculator to directly recall the contents of any one of the storage registers specified by those storage register designating keys into a specified one of the working registers, and said indirect control key is operable with each of the last-mentioned working register transfer keys and one or more storage register designating keys for causing the calculator to indirectly recall the contents of any one of the storage registers specified by the contents of any other one or the storage registers specified by those storage register designating keys into a specified one of the working registers.

6. An electronic calculator as in claim 5 wherein the contents of all the mentioned storage registers remain unchanged.

7. An electronic calculator as in claim 5 or 6 wherein said indirect control key is operable with each of the last mentioned working register transfer keys and one or more storage register designating keys for causing the calculator to indirectly recall the contents of any one of the storage registers multiply indirectly specified by the contents of others of the storage registers a first of which is specified by those storage registers designating keys into a specified one of the working registers.

8. An electronic calculator as in any one of the preceding claims 2 through 7 wherein said keyboard input unit includes a working register transfer key operable with one or more storage register designating keys for causing the calculator to directly exchange the contents of a specified one of the working registers with the contents of any one of the storage registers specified by those storage register designating keys, and said indirect control key is operable with the last-mentioned working register transfer key and one or more storage register designating keys for causing the calculator to indirectly exchange the contents of a specified one of the working registers with the contents of any one of the storage registers specified by the contents of any other one of the storage registers specified by those storage register designating keys.

9. An electronic calculator as in claim 8 wherein the contents of the last-mentioned one of the storage registers remains unchanged.

10. An electronic calculator as in claim 8 or 9 wherein said indirect control key is operable with the last-mentioned working register transfer key and one or more storage register designating keys for causing the calculator to indirectly exchange the contents of a specified one of the working registers with the contents of any one of the storage registers multiply indirectly specified by the contents of others of the storage registers a first of which is specified by those storage register designating keys.

* C L M S *

11. An electronic calculator as in any one of the preceding claims 2 through 10 wherein said keyboard input unit includes a plurality of arithmetic operator keys each operable for conditioning the calculator to perform a specified arithmetic operation, each of the working register transfer keys is operable with each of the arithmetic operator keys in combination with one or more storage register designating keys for causing the calculator to perform a specified arithmetic operation employing the contents of a specified one of the working registers and the contents of any one of the storage registers specified by those storage register designating keys, and said indirect control key is operable with each of the working register transfer keys in combination with any one of the arithmetic operator keys and one or more storage register designating keys for causing the calculator to perform a specified arithmetic operation employing the contents of a specified one of the working registers and the contents of any other one of the storage registers specified by those storage register designating keys.

12. An electronic calculator as in claim 11 wherein the result of the specified arithmetic operation is stored in one of the registers whose contents are employed in the specified arithmetic operation without changing the contents of the other registers mentioned.

13. An electronic calculator as in claim 11 or 12 wherein said direct control key is operable with each of the working register transfer keys in combination with any one of the arithmetic operator keys and one or more storage register designating keys for causing the calculator to perform a specified arithmetic operation employing the contents of a specified one of the working registers and the contents of any one of of the storage registers multiply indirectly specified by the contents of others of the storage registers a first of which is specified by the storage register designating keys.

14. 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.

