United States Patent [19]

Poland

[11] E **Re. 30,671**

[45] Reissued **Jul. 7, 1981**

[54] **MICROPROCESSOR SYSTEM HAVING HIGH ORDER CAPABILITY**

[75] Inventor: **Sydney W. Poland,** Arlington, Tex.

[73] Assignee: **Texas Instruments Incorporated,** Dallas, Tex.

[21] Appl. No.: **55,888**

[22] Filed: **Jul. 9, 1979**

Related U.S. Patent Documents

Reissue of:

[64] Patent No.: **4,153,937**
Issued: **May 8, 1979**
Appl. No.: **783,903**
Filed: **Apr. 1, 1977**

U.S. Applications:

[63] Continuation-in-part of Ser. No. 714,464, Aug. 16, 1976.

[51] **Int. Cl.**[3] **G06F 3/08;** G06F 15/02

[52] **U.S. Cl.** **364/706;** 364/200; 364/900

[58] **Field of Search** 364/706, 200, 900

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,593,313	7/1971	Tomaszewski et al.	364/200
3,760,171	9/1973	Wang et al.	364/706
3,859,636	1/1975	Cook	364/200
3,942,156	3/1976	Mock et al.	364/200
3,953,833	4/1976	Shapiro	364/200
3,971,925	7/1976	Wenninger et al.	364/706

OTHER PUBLICATIONS

"National's Scientific Calculator has Long-Term All-Semi-Conductor Memory", *Electronics,* Jun. 10, 1976, pp. 29–30.

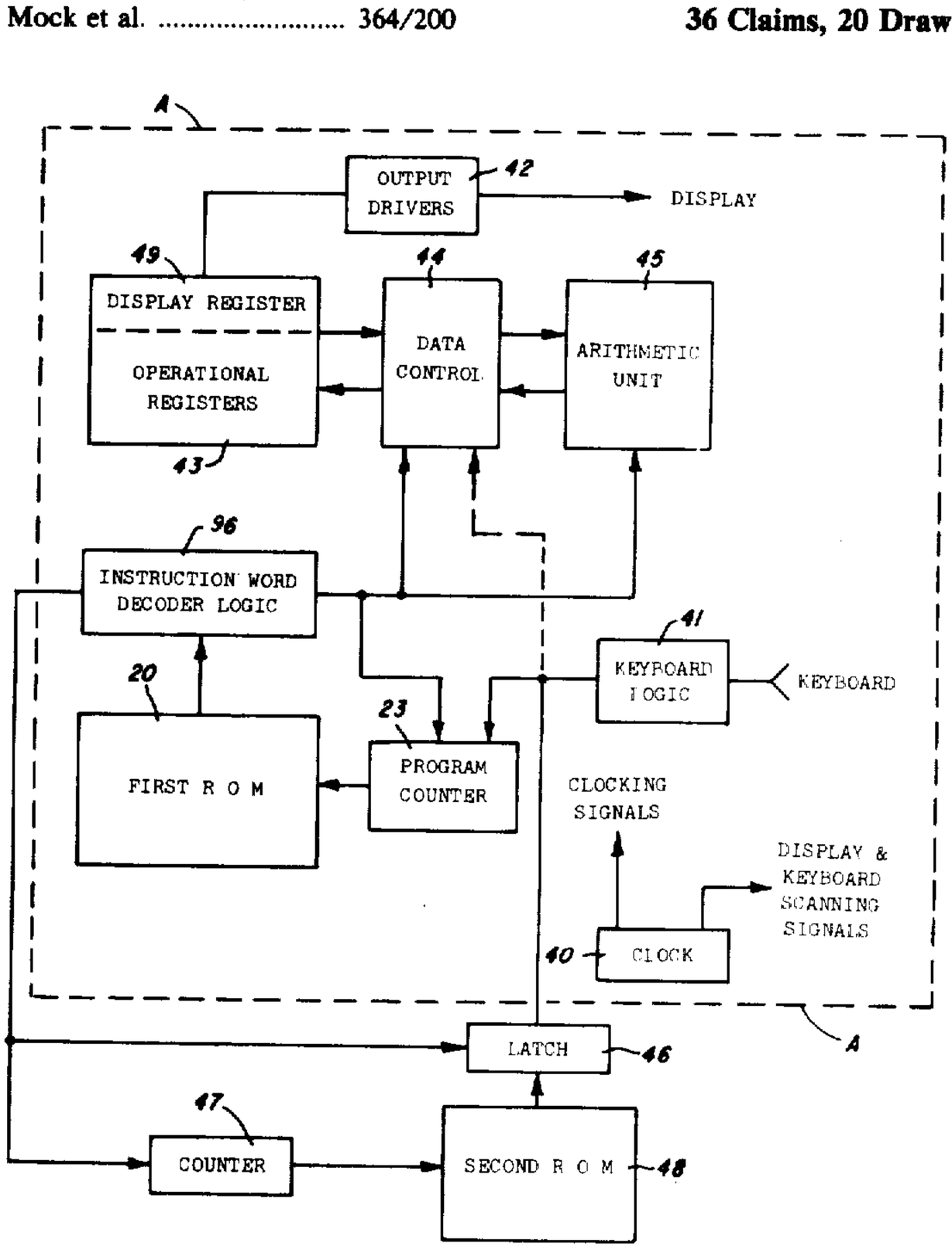
Primary Examiner—Jerry Smith
Attorney, Agent, or Firm—Robert D. Marshall, Jr.; Leo N. Heiting; Melvin Sharp

[57] **ABSTRACT**

A microprocessor system with high order capabilities is provided with the two non-volatile memories which are read-only-memories (ROMs) in the disclosed embodiment. A first ROM stores the microcode for controlling the operation of the microprocessor circuits. The second ROM, which is preferably disposed in a module or cartridge, stores a plurality of program codes which are used to address the first ROM. The second ROM's module may be inserted into a receptacle for interconnecting it with the remainder of the microprocessor system. Preferably, a plurality of such second ROMs are available for selectively plugging into the microprocessor system.

Further, a particular embodiment of the microprocessor system with high order capabilities for use as an electronic calculator with high order capabilities is disclosed in great detail.

36 Claims, 20 Drawing Figures



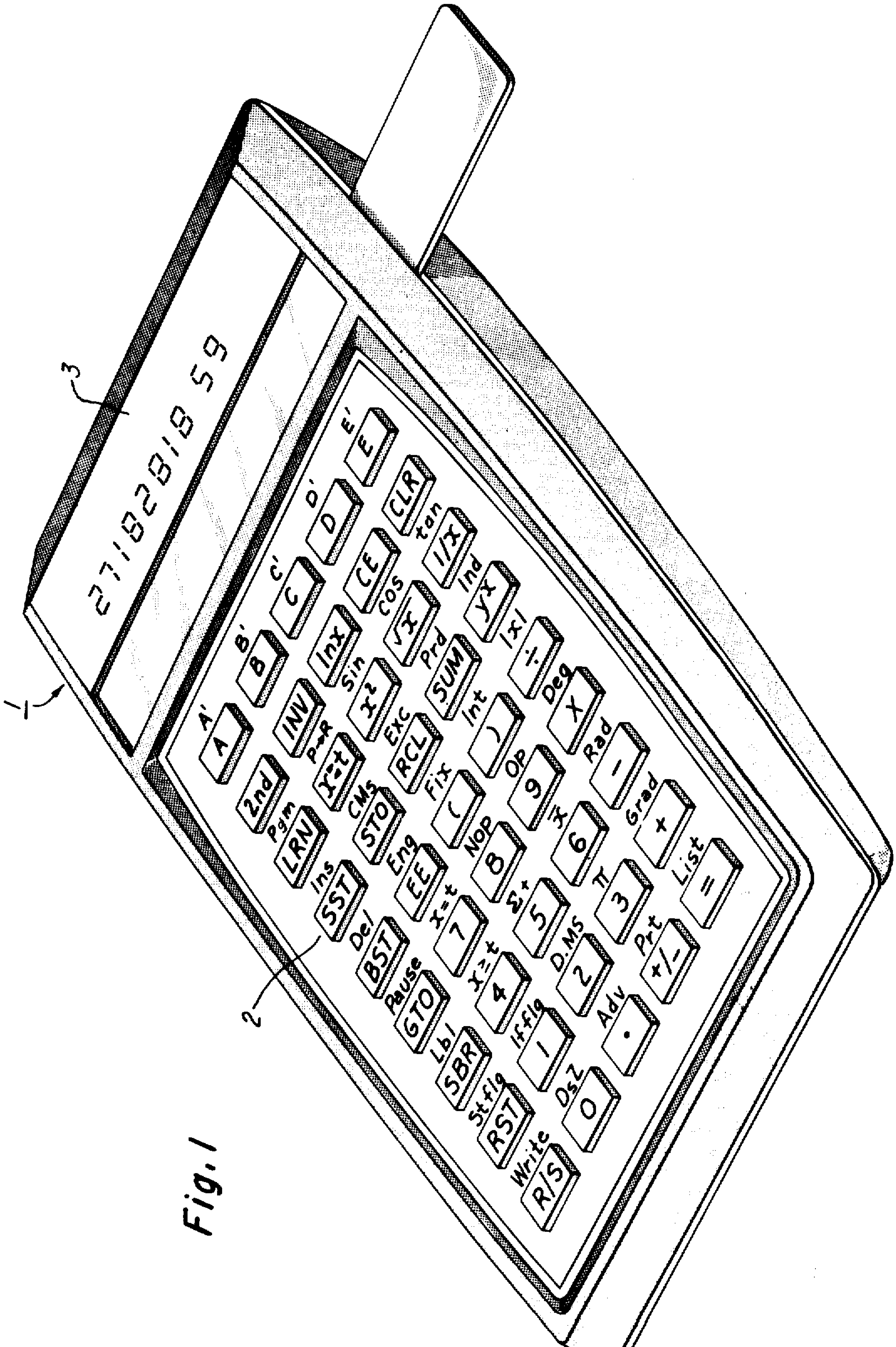


Fig. 1

I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I0	I1	
0	1	0	1	0	0	0	0	1	1	0	0	0	LOAD LSD OF KEYBOARD REG WITH R5 (R5 KR)
0	1	0	1	0	0	0	0	0	1	0	0	0	LOAD R5 WITH LSD OF KEYBOARD REG (KR R5)
0	1	0	1	0	0	0	0	0	1	1	0	0	LOAD KEYBOARD REG WITH EXT (EXT KR)
0	0	0	0	0	0	0	0	1	0	1	0	1	PREG
0	1	0	1	0	0	0	0	0	1	1	1	0	FETCH
0	1	0	1	0	0	0	1	1	1	1	1	0	FETCH HIGH
0	1	0	1	0	0	0	0	1	1	1	1	0	LOAD PC
0	1	0	1	0	0	0	1	0	1	1	1	0	UNLOAD PC

Fig. 5h

Fig. 7

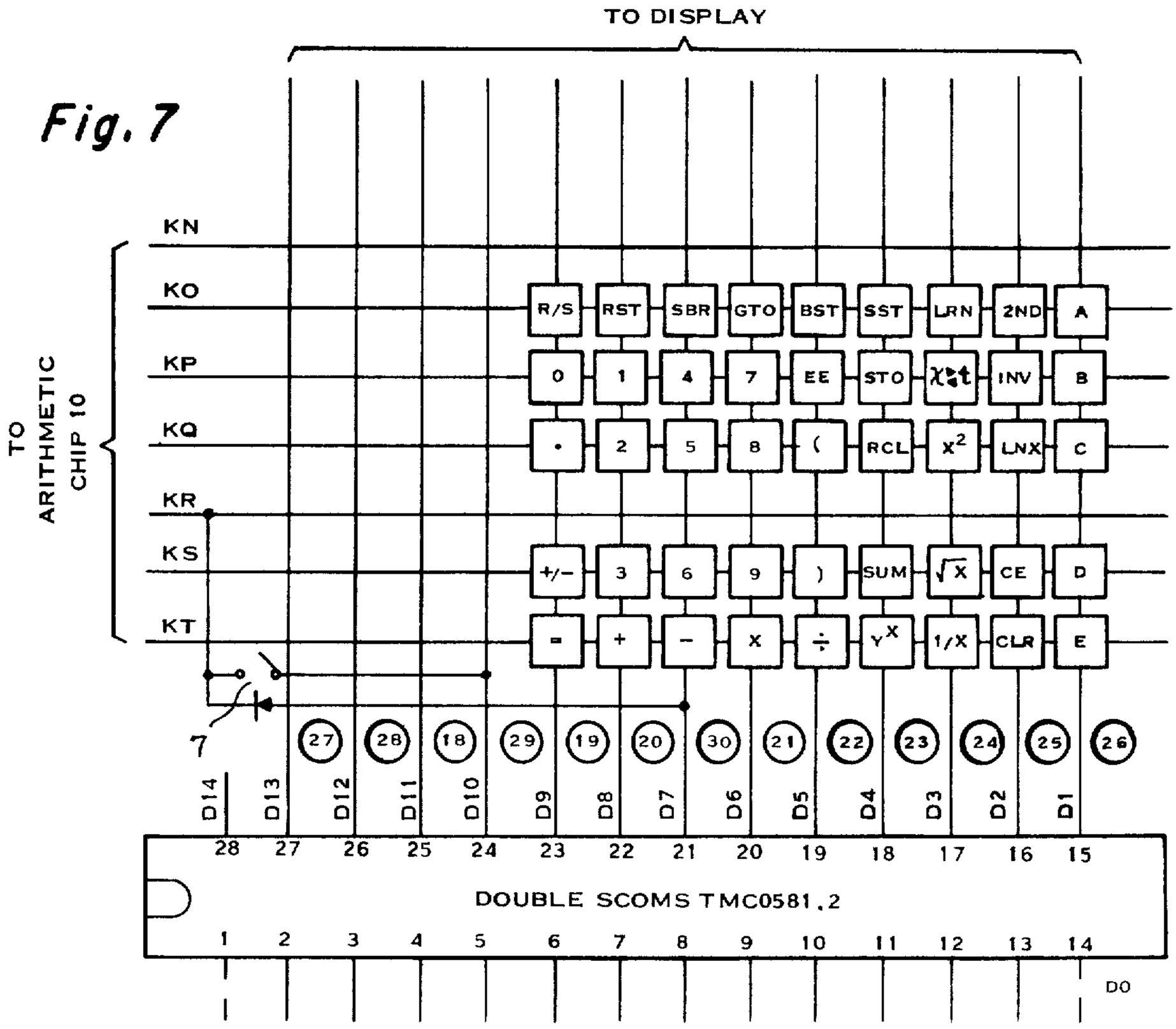


Fig. 11

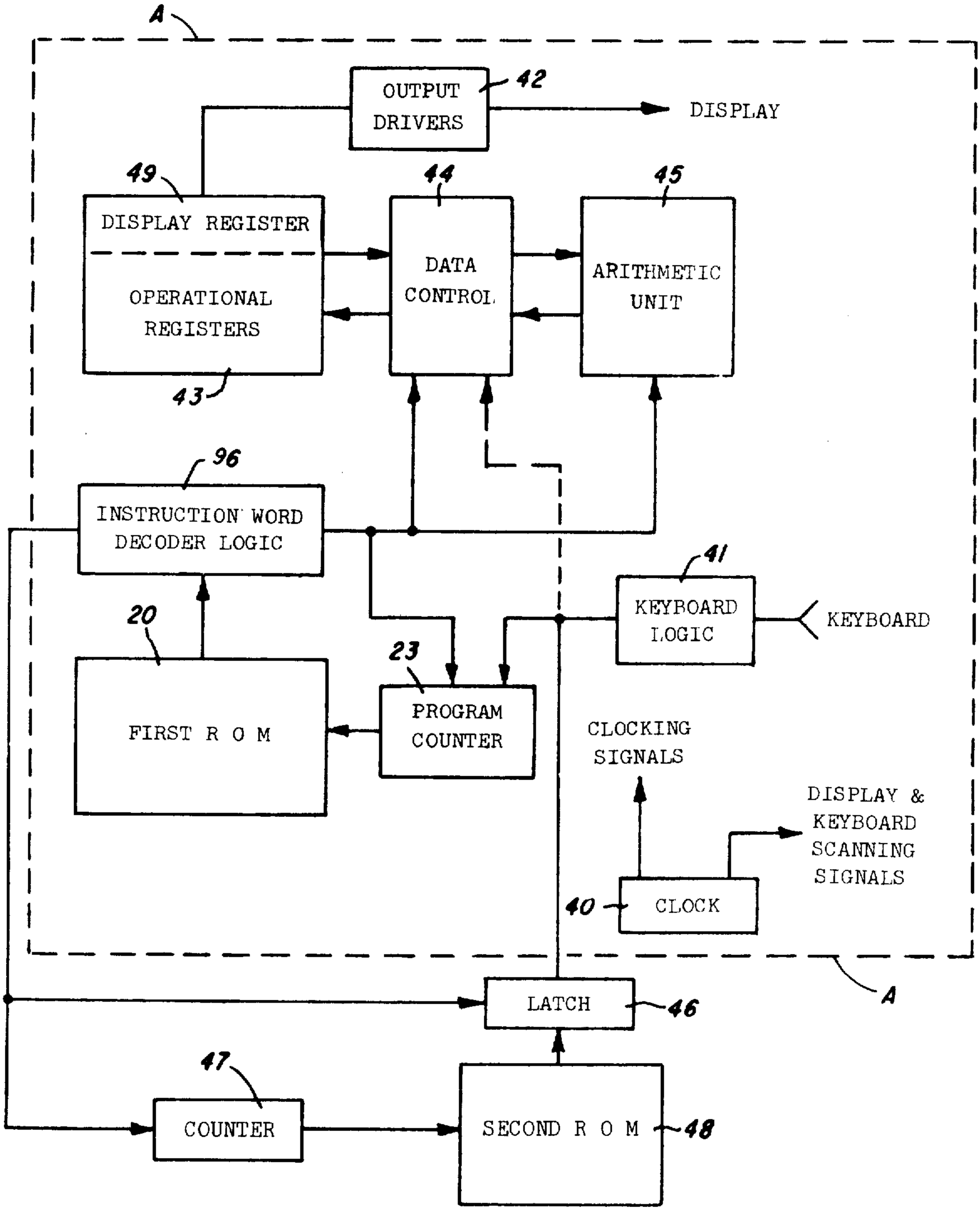
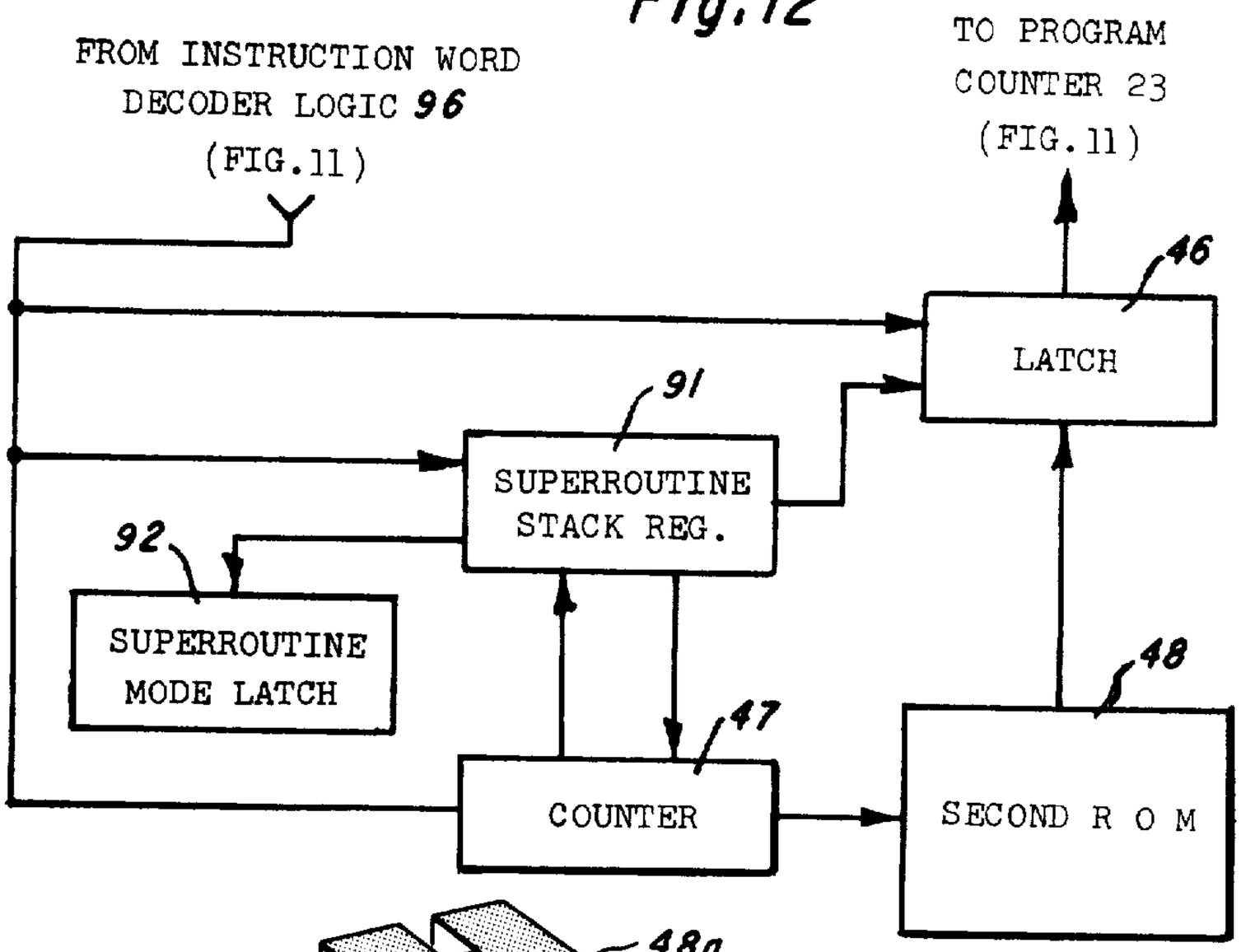

A
OUTPUT DRIVERS
42
DISPLAY
49
DISPLAY REGISTER
OPERATIONAL REGISTERS
43
44
DATA CONTROL
45
ARITHMETIC UNIT
96
INSTRUCTION WORD DECODER LOGIC
41
KEYBOARD LOGIC
KEYBOARD
20
FIRST R O M
23
PROGRAM COUNTER
CLOCKING SIGNALS
DISPLAY & KEYBOARD SCANNING SIGNALS
40
CLOCK
A
LATCH
46
47
COUNTER
SECOND R O M
48

Fig. 12


FROM INSTRUCTION WORD DECODER LOGIC 96 (FIG. 11)
TO PROGRAM COUNTER 23 (FIG. 11)
LATCH
46
91
SUPERROUTINE STACK REG.
92
SUPERROUTINE MODE LATCH
47
COUNTER
48
SECOND R O M

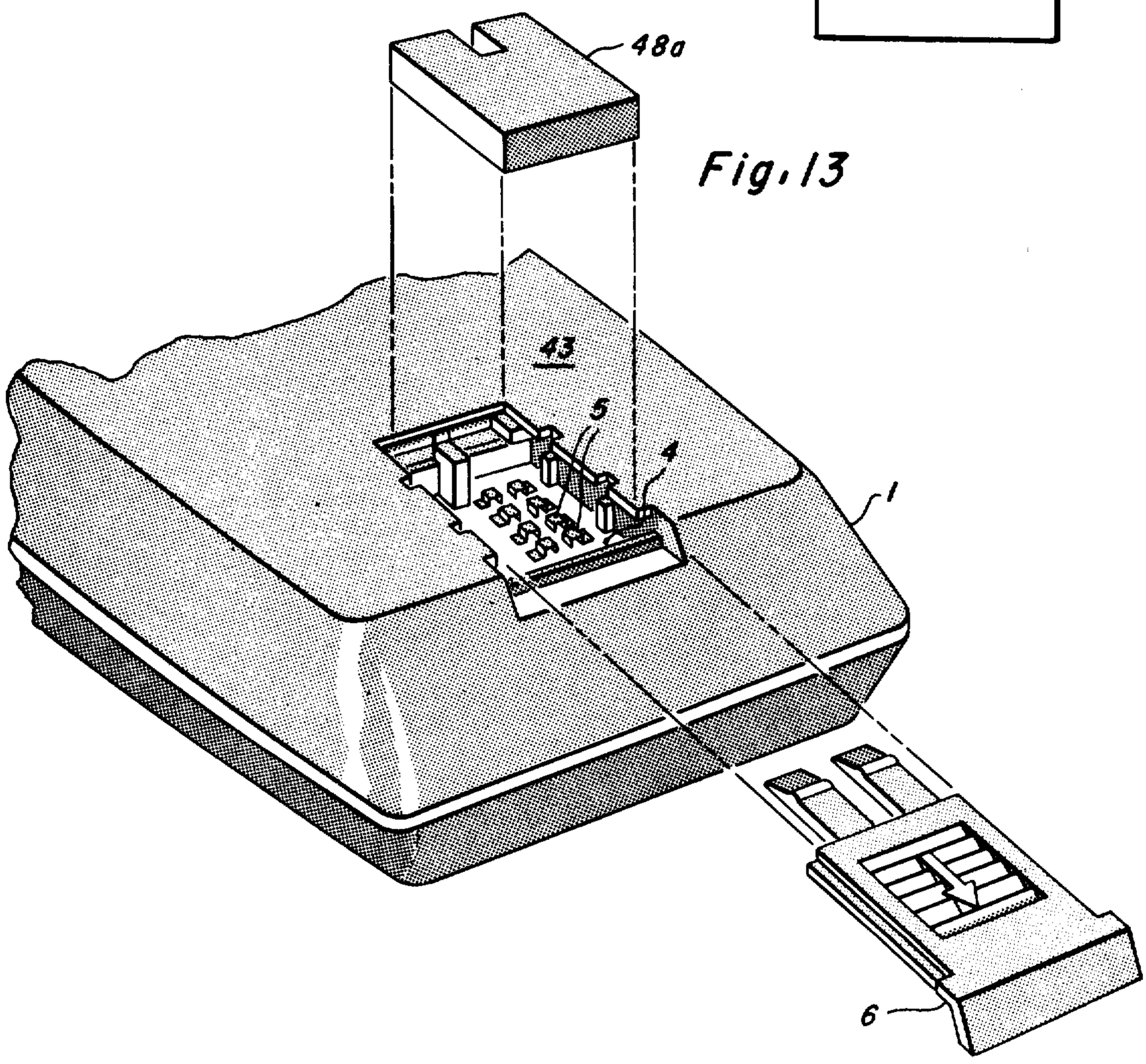

48a
43
5
4
1
6

Fig. 13

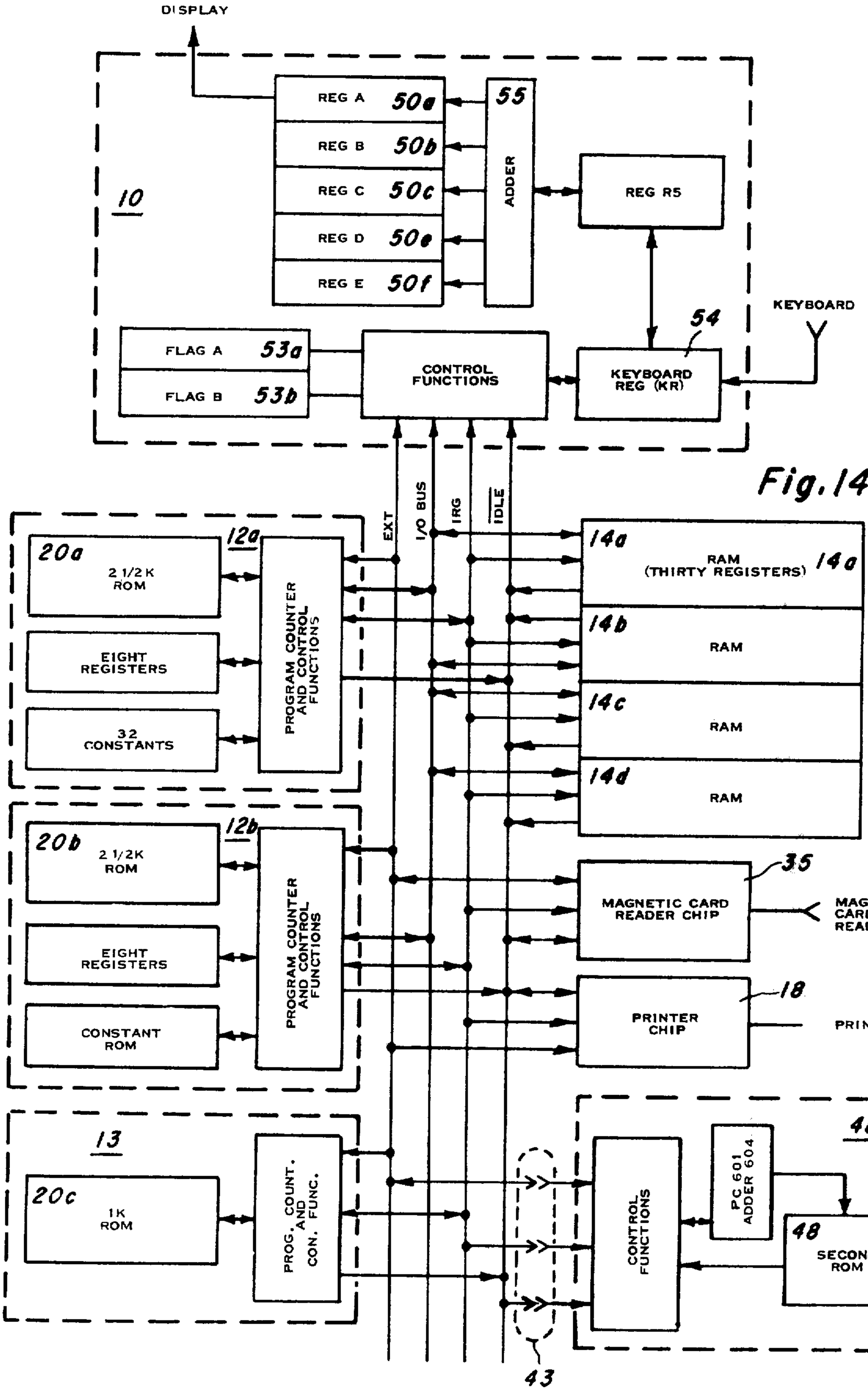

DISPLAY
10
REG A 50a
REG B 50b
REG C 50c
REG D 50d
REG E 50f
ADDER
55
REG R5
54
KEYBOARD
FLAG A 53a
FLAG B 53b
CONTROL FUNCTIONS
KEYBOARD REG (KR)
EXT
I/O BUS
IRG
IDLE
120
20a
2 1/2 K ROM
EIGHT REGISTERS
32 CONSTANTS
PROGRAM COUNTER AND CONTROL FUNCTIONS
14a
RAM (THIRTY REGISTERS)
14b
RAM
14c
RAM
14d
RAM
12b
20b
2 1/2K ROM
EIGHT REGISTERS
CONSTANT ROM
PROGRAM COUNTER AND CONTROL FUNCTIONS
35
MAGNETIC CARD READER CHIP
MAGNETIC CARD READER
18
PRINTER CHIP
PRINTER
13
20c
1K ROM
PROG. COUNT. AND CON. FUNC.
48'
CONTROL FUNCTIONS
PC 601 ADDER 604
48
SECOND ROM
43

Fig. 14

Fig. 15

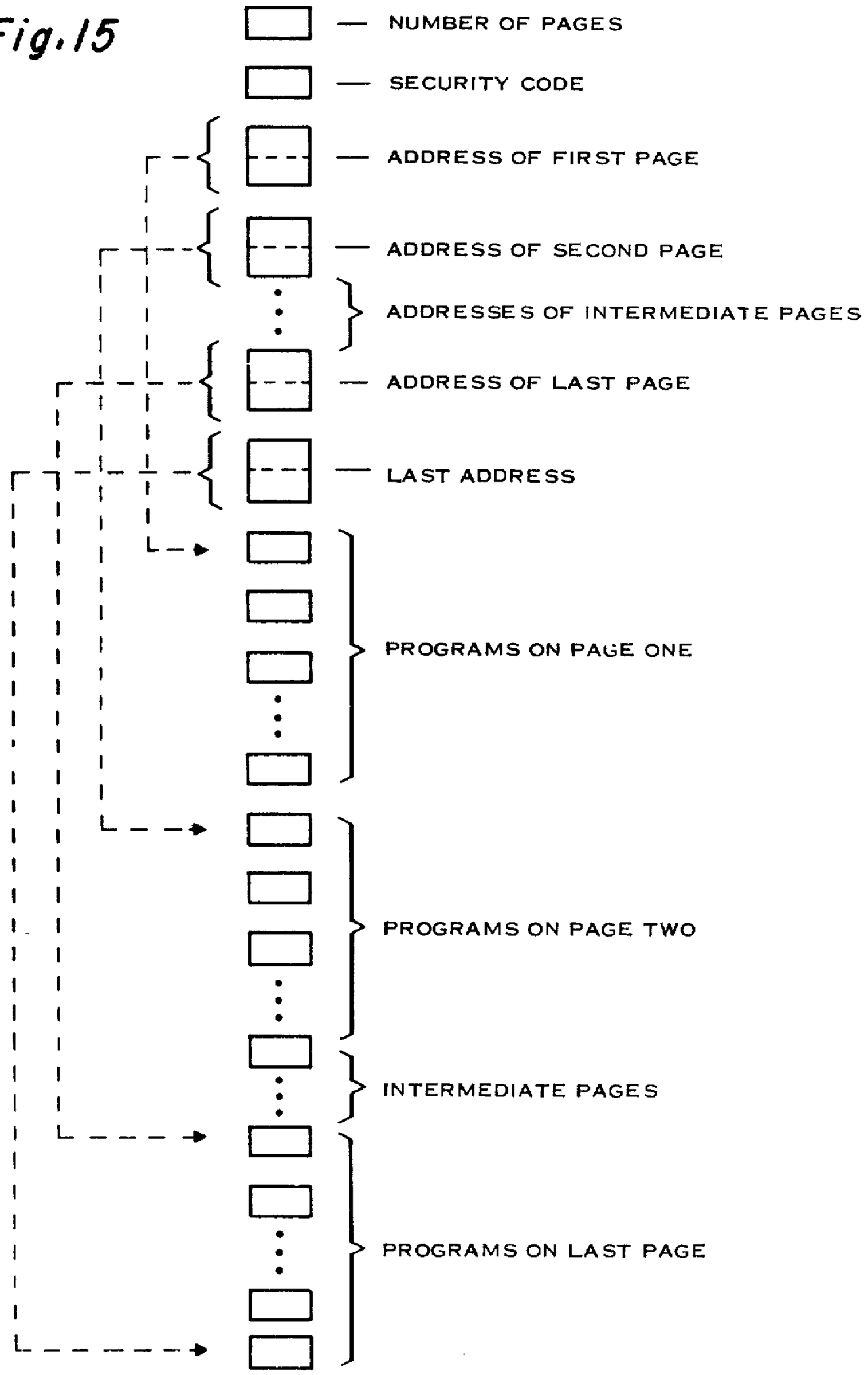
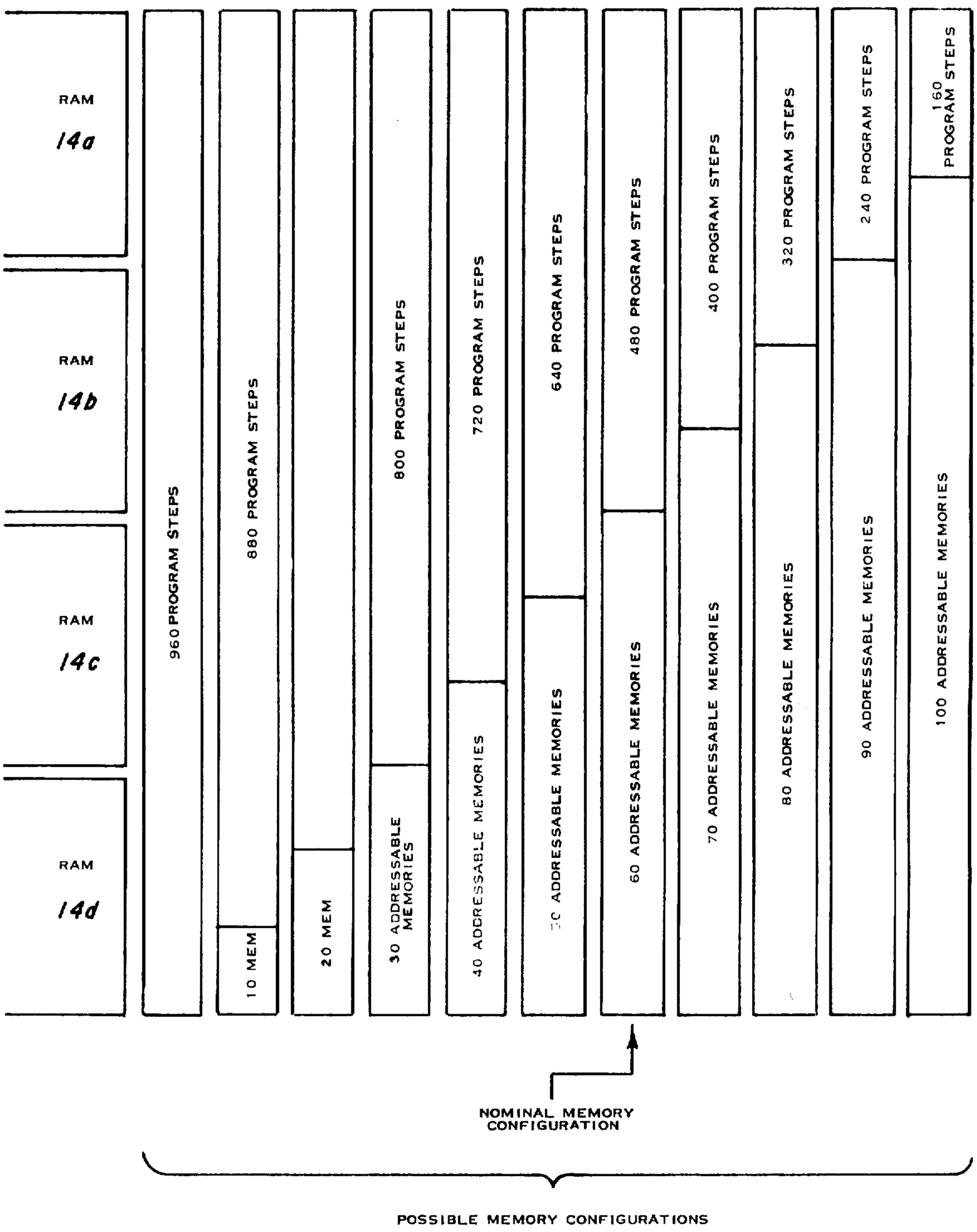

NUMBER OF PAGES
SECURITY CODE
ADDRESS OF FIRST PAGE
ADDRESS OF SECOND PAGE
ADDRESSES OF INTERMEDIATE PAGES
ADDRESS OF LAST PAGE
LAST ADDRESS
PROGRAMS ON PAGE ONE
PROGRAMS ON PAGE TWO
INTERMEDIATE PAGES
PROGRAMS ON LAST PAGE

Fig. 16


RAM
14a
RAM
14b
RAM
14c
RAM
14d
960 PROGRAM STEPS
880 PROGRAM STEPS
10 MEM
20 MEM
800 PROGRAM STEPS
30 ADDRESSABLE MEMORIES
720 PROGRAM STEPS
40 ADDRESSABLE MEMORIES
640 PROGRAM STEPS
50 ADDRESSABLE MEMORIES
480 PROGRAM STEPS
60 ADDRESSABLE MEMORIES
400 PROGRAM STEPS
70 ADDRESSABLE MEMORIES
320 PROGRAM STEPS
80 ADDRESSABLE MEMORIES
240 PROGRAM STEPS
90 ADDRESSABLE MEMORIES
160 PROGRAM STEPS
100 ADDRESSABLE MEMORIES
NOMINAL MEMORY CONFIGURATION
POSSIBLE MEMORY CONFIGURATIONS

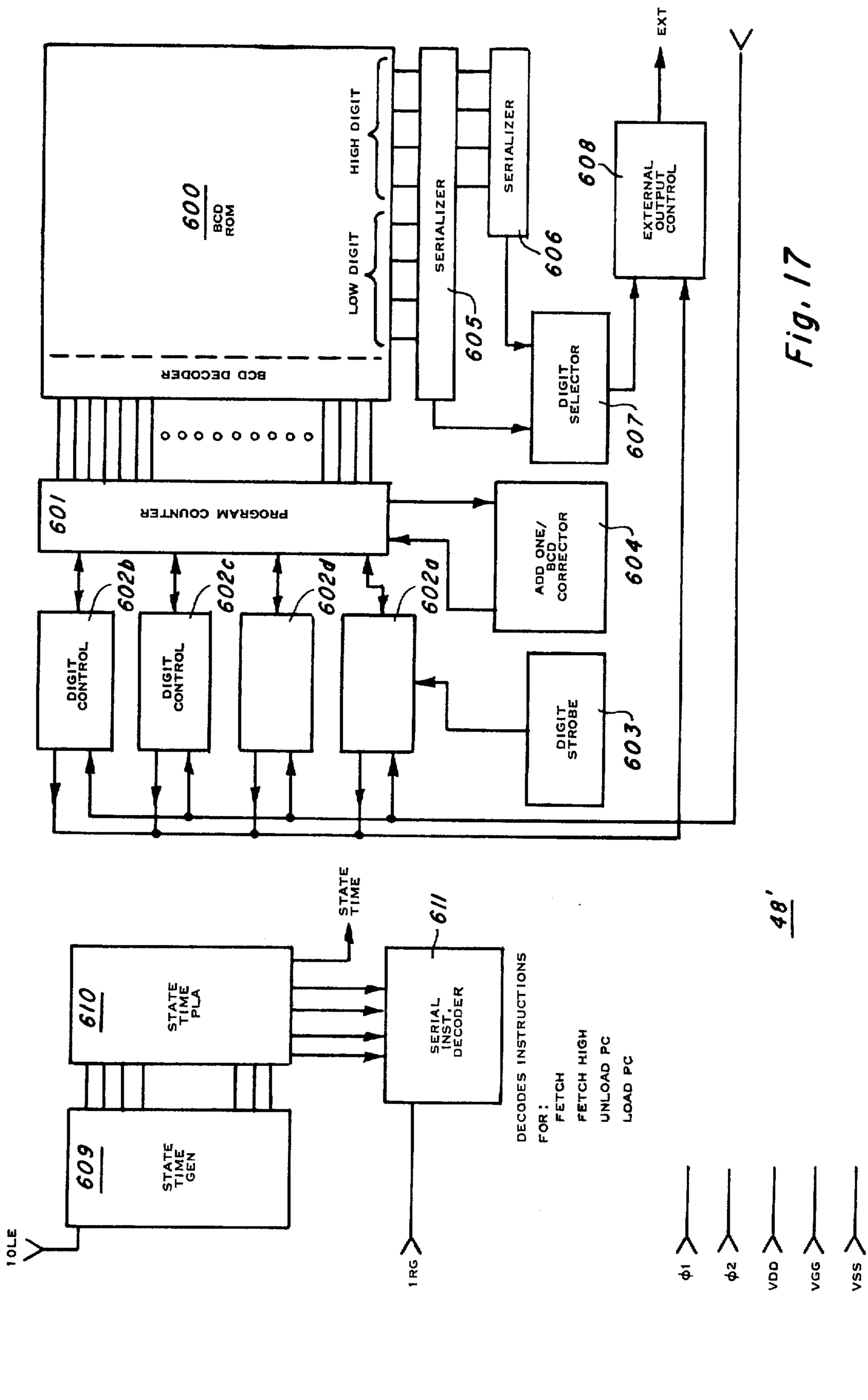

600
BCD ROM
HIGH DIGIT
LOW DIGIT
BCD DECODER
SERIALIZER
SERIALIZER
605
606
608
EXTERNAL OUTPUT CONTROL
EXT
DIGIT SELECTOR
607
601
PROGRAM COUNTER
ADD ONE/ BCD CORRECTOR
604
602b
602c
602d
602a
DIGIT CONTROL
DIGIT CONTROL
DIGIT STROBE
603
STATE TIME
611
610
STATE TIME PLA
SERIAL INST. DECODER
609
STATE TIME GEN
DECODES INSTRUCTIONS FOR:
FETCH
FETCH HIGH
UNLOAD PC
LOAD PC
10LE
1RG
φ1
φ2
VDD
VGG
VSS

Fig. 17

48'

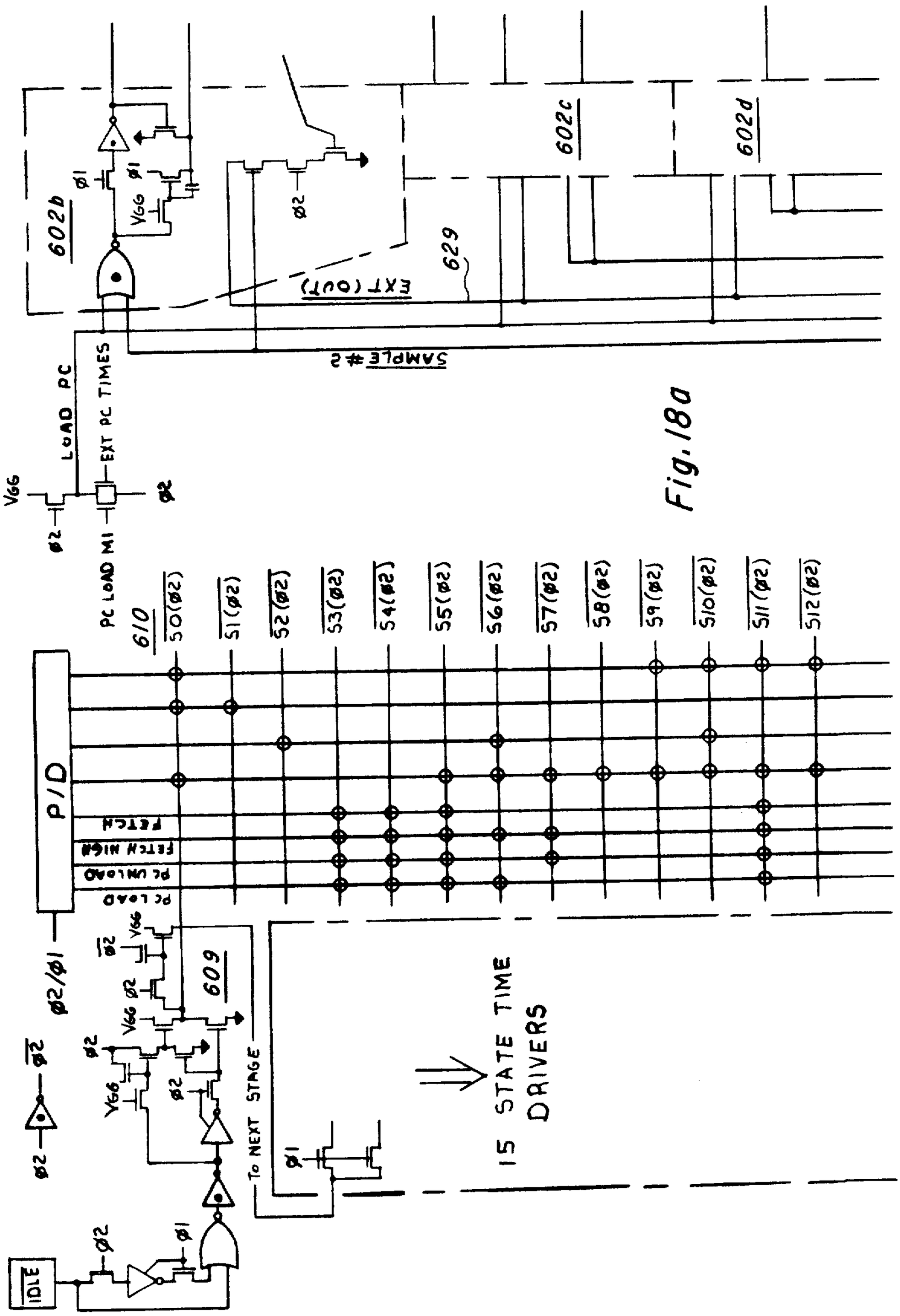

602b
602c
602d
629
SAMPLE #2
EXT (OUT)
LOAD PC
EXT PC TIMES
PC LOAD
P/D
FETCH
FETCH HIGH
PC UNLOAD
PC LOAD
610
609
15 STATE TIME DRIVERS
To NEXT STAGE
IDLE

Fig. 18a

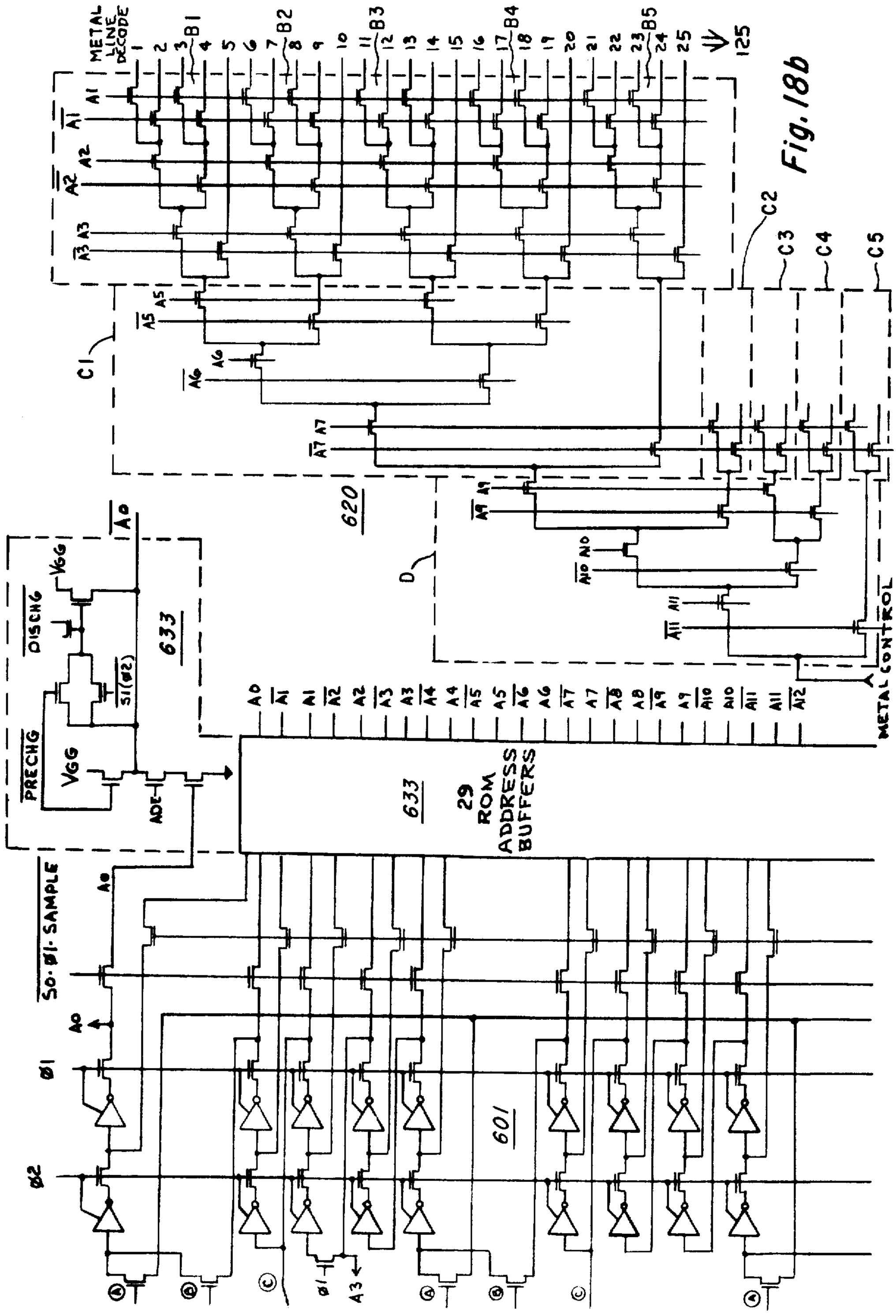
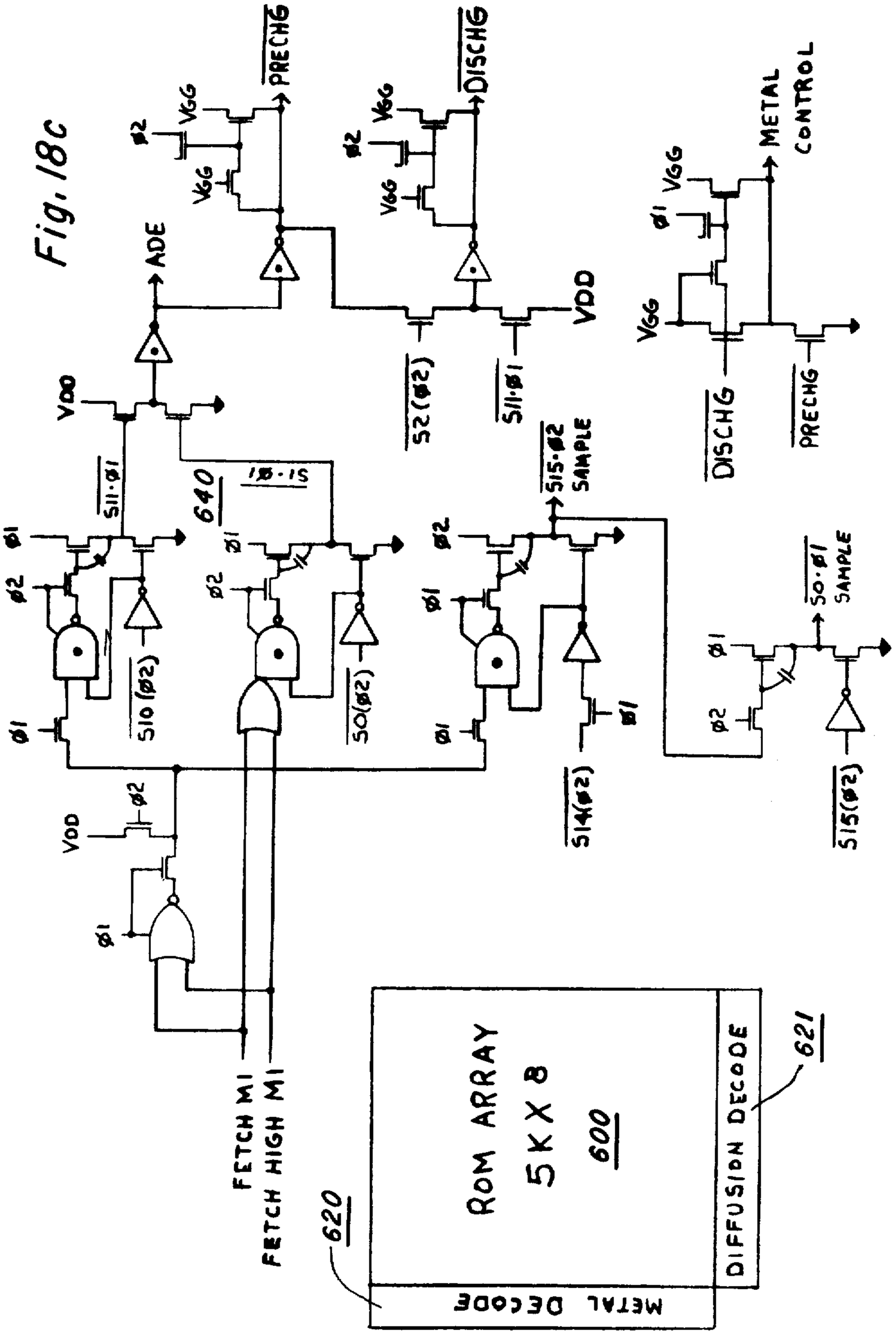

METAL LINE DECODE
B1
B2
B3
B4
B5
C1
C2
C3
C4
C5
620
633
29 ROM ADDRESS BUFFERS
601
DISCHG
PRECHG
VGG
ADE
SO·Ø1·SAMPLE
METAL CONTROL

Fig. 18b

Fig. 18c


PRECHG
DISCHG
ADE
VDD
VGG
φ2
φ1
S11·φ1
S10 (φ2)
S40
S8 · S5
S0(φ2)
S0·φ1
SAMPLE
S15·φ2
SAMPLE
S2(φ2)
S11·φ1
S14(φ2)
S15(φ2)
METAL
CONTROL
FETCH MI
FETCH HIGH MI
ROM ARRAY
5K X 8
600
620
621
METAL DECODE
DIFFUSION DECODE

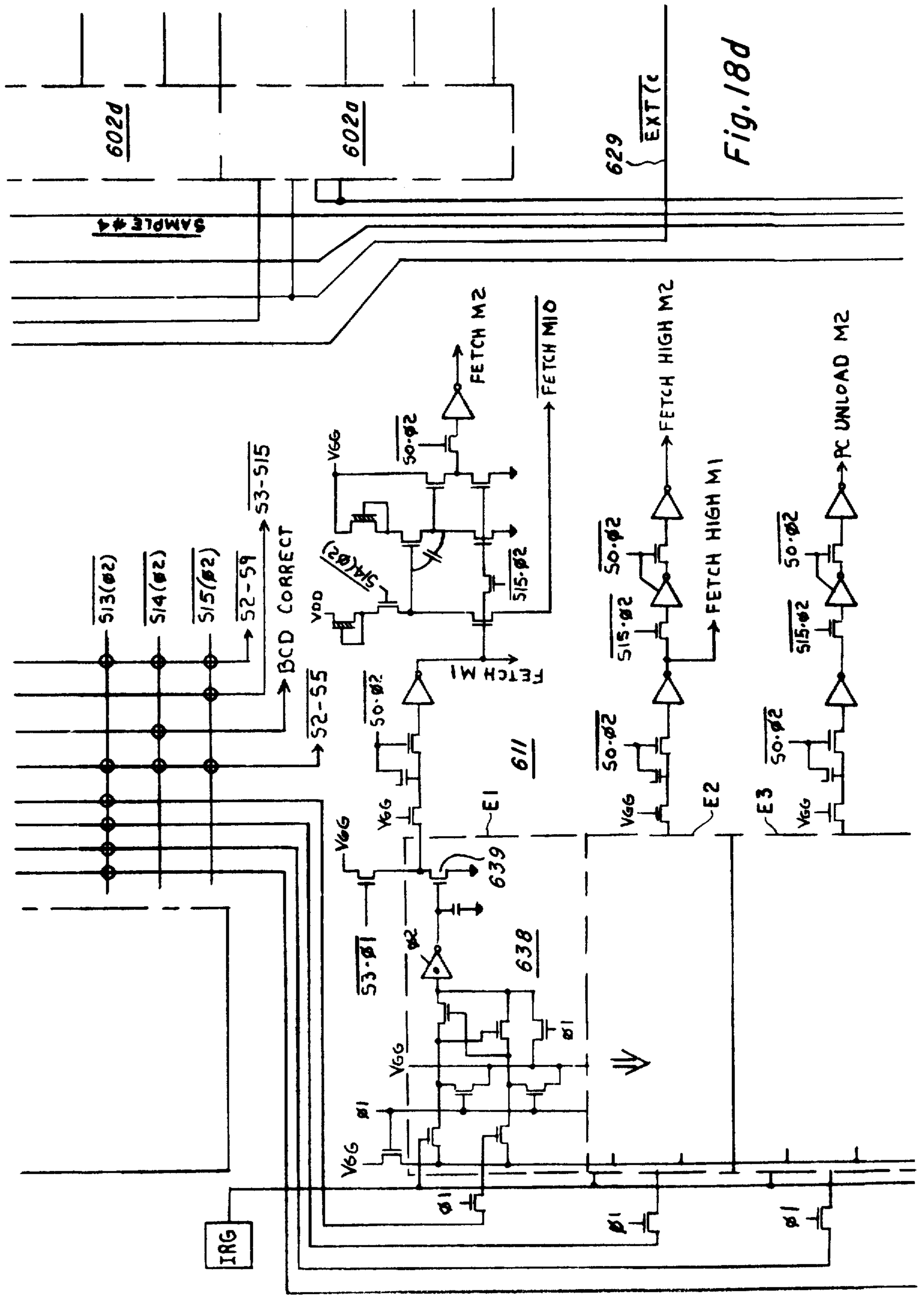
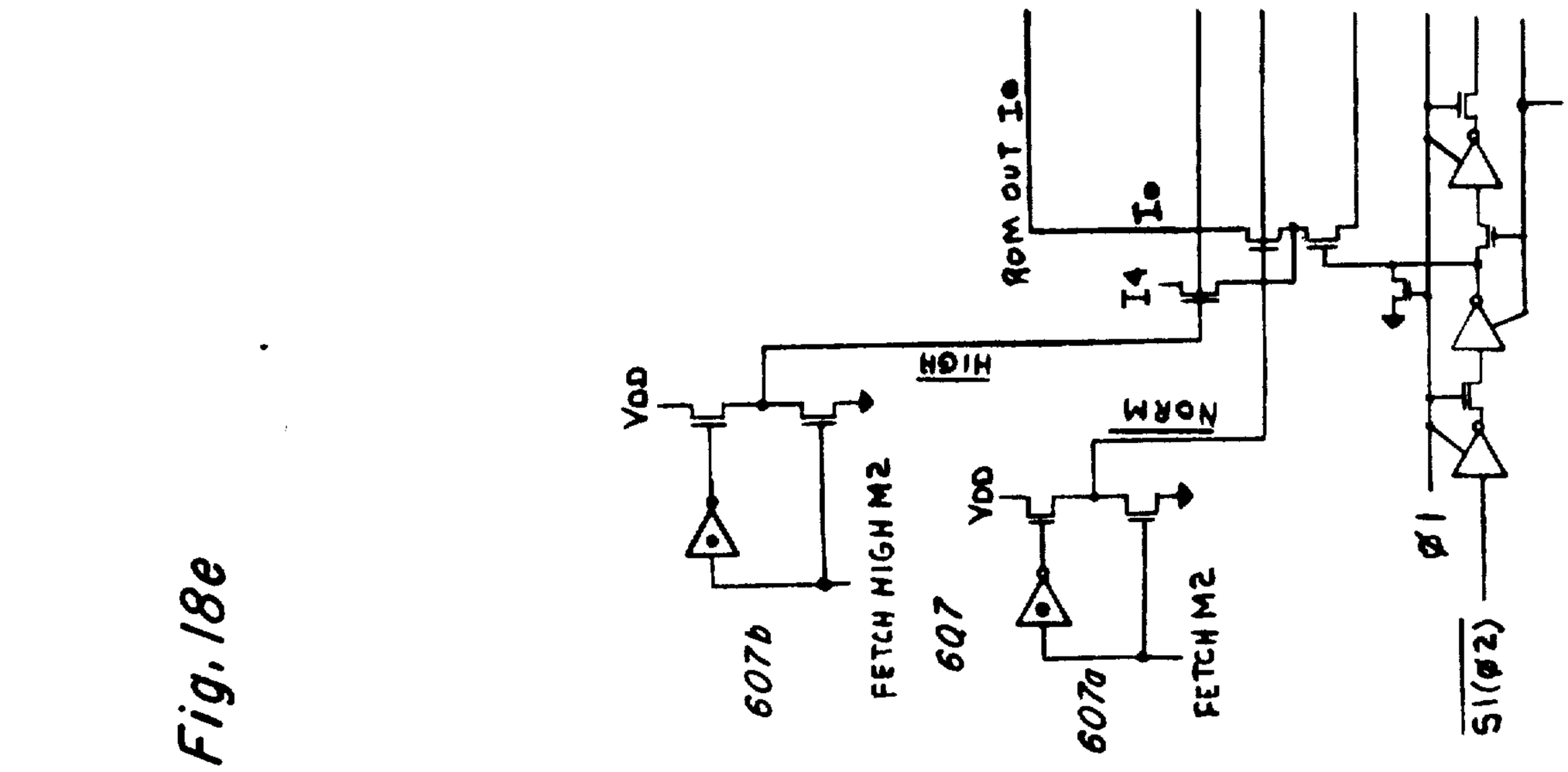
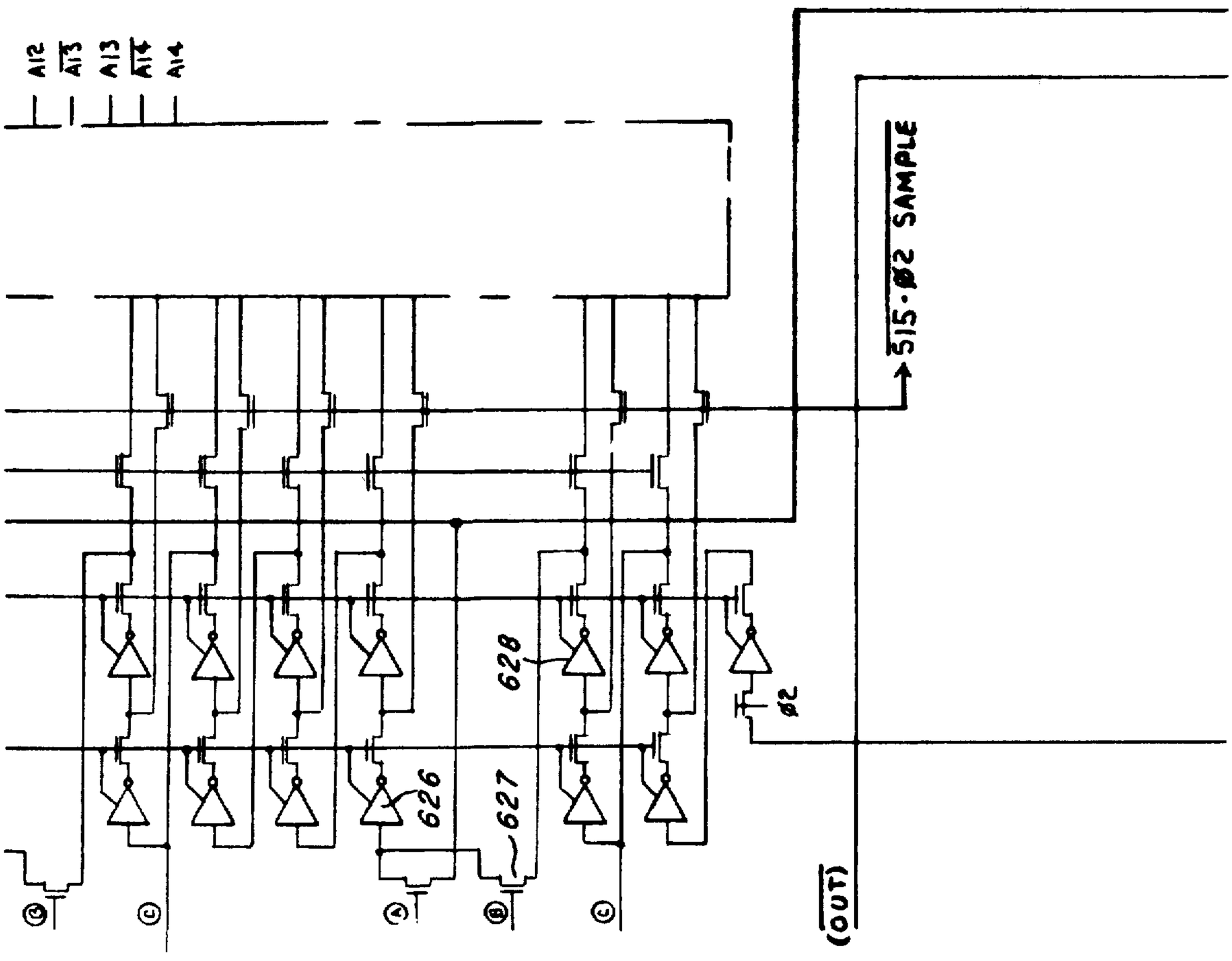

602d
602a
SAMPLE #4
629
EXT(c
FETCH M2
FETCH M10
FETCH HIGH M2
FETCH HIGH M1
PC UNLOAD M2
50·ø2
VGG
S3-S15
S13(ø2)
S14(ø2)
S15(ø2)
S2-S9
BCD CORRECT
S2-S5
VDD
S1A(ø2)
S15·ø2
FETCH M1
611
E1
E2
E3
639
638
53·ø1
ø2
ø1
IRG

Fig. 18d

Fig. 18e


ROM OUT I0
I0
I4
HIGH
NORM
VDD
FETCH HIGH M2
607b
607
VDD
FETCH M2
607a
ø1
S1(ø2)


A12
A13
A13
A14
A14
S15·ø2 SAMPLE
628
626
627
ø2
(OUT)

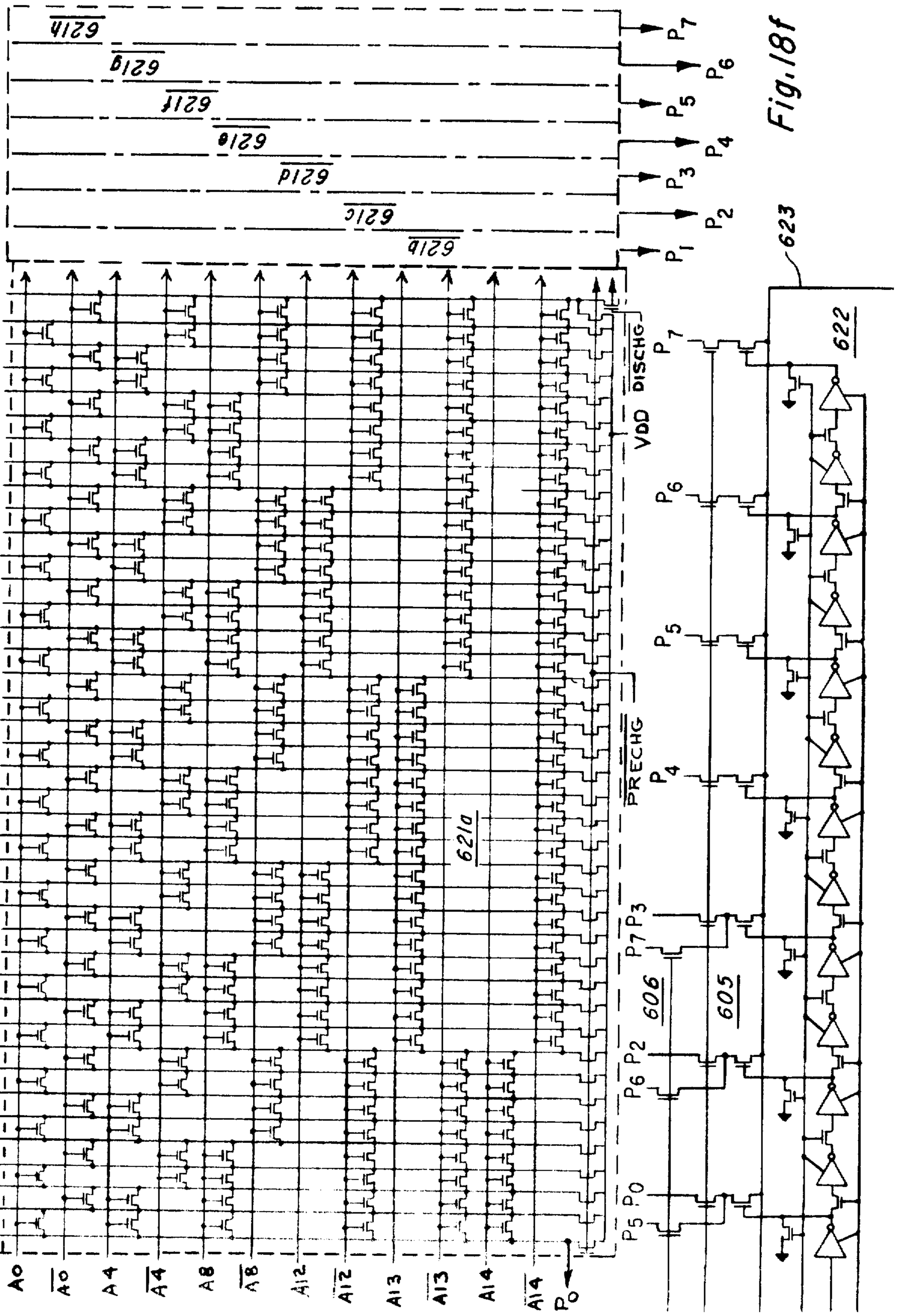

621h
621g
621f
621e
621d
621c
621b
P7
P6
P5
P4
P3
P2
P1
623
622
VDD DISCHG
PRECHG
621a
606
605
P7 P3
P6 P2
P5 P0
A0
A4
A8
A12
A13
A14
P0

Fig. 18f

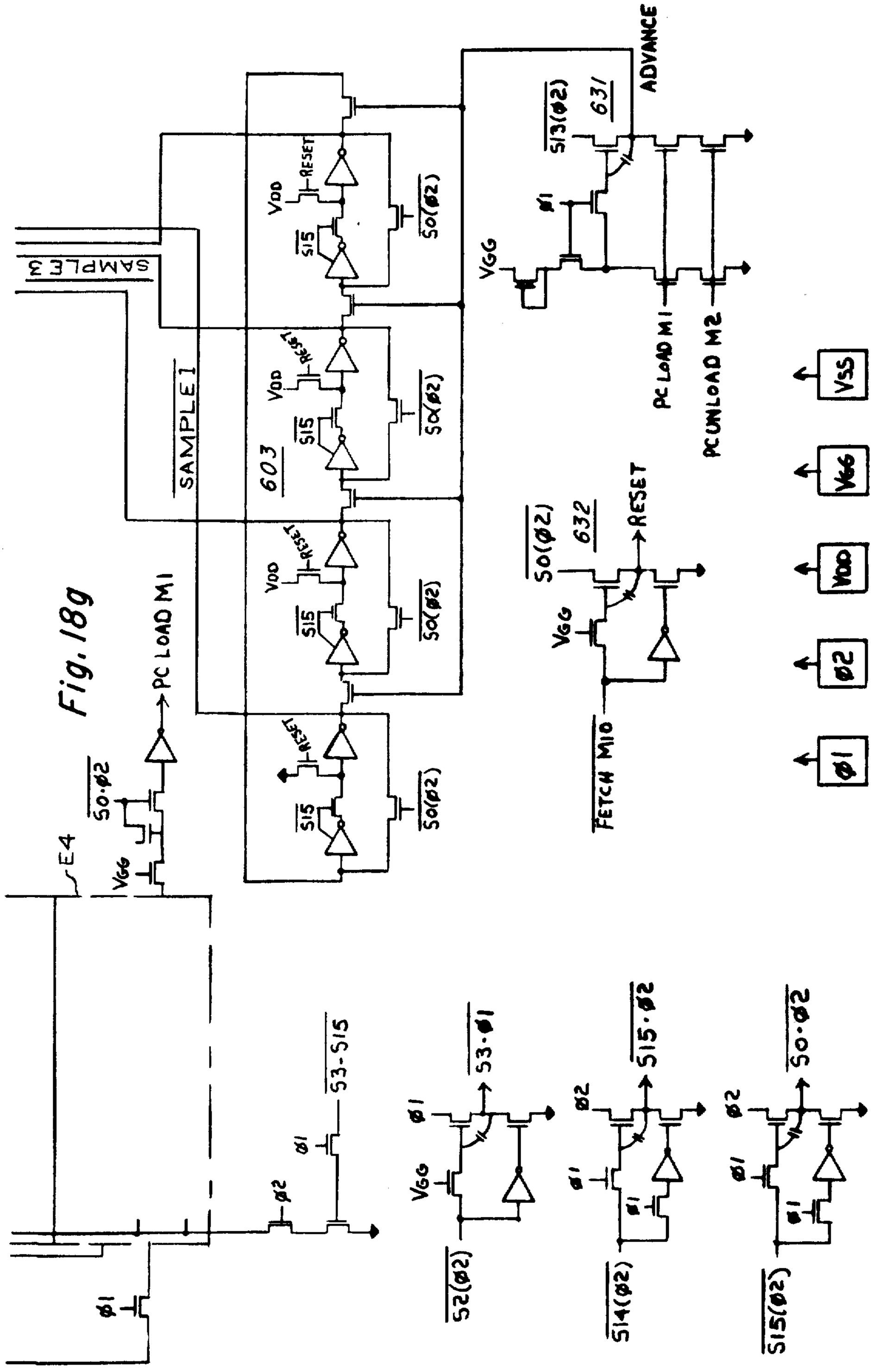
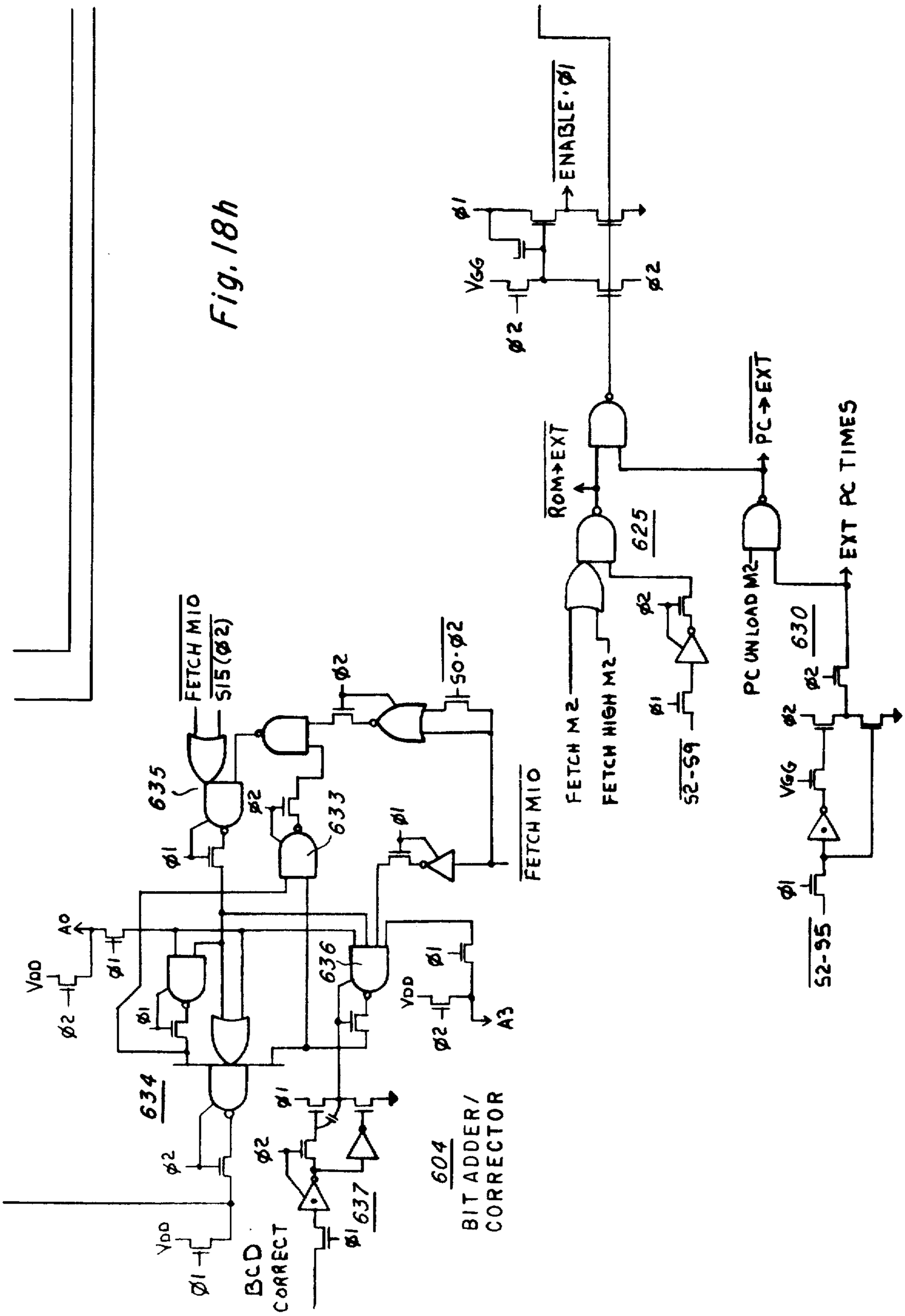

SAMPLE 3
SAMPLE 1
603
PC LOAD M1
E4
VGG
VDD
RESET
S15
S0(Ø2)
S0·Ø2
Ø1
Ø2
S3-S15
S3·Ø1
S15·Ø2
S0·Ø2
S2(Ø2)
S14(Ø2)
S15(Ø2)
631
ADVANCE
PC LOAD M1
PC UNLOAD M2
632
FETCH M10
VSS
VDD

Fig. 18g

Fig. 18h


BIT ADDER/ CORRECTOR
604
BCD CORRECT
634
637
635
633
636
FETCH M10
S15(Ø2)
50·Ø2
FETCH M10
A0
A3
VDD
Ø1
Ø2
625
ROM→EXT
PC→EXT
FETCH M2
FETCH HIGH M2
S2-S9
630
PC UNLOAD M2
E→EXT PC TIMES
VGG
S2-S5
E→ENABLE·Ø1

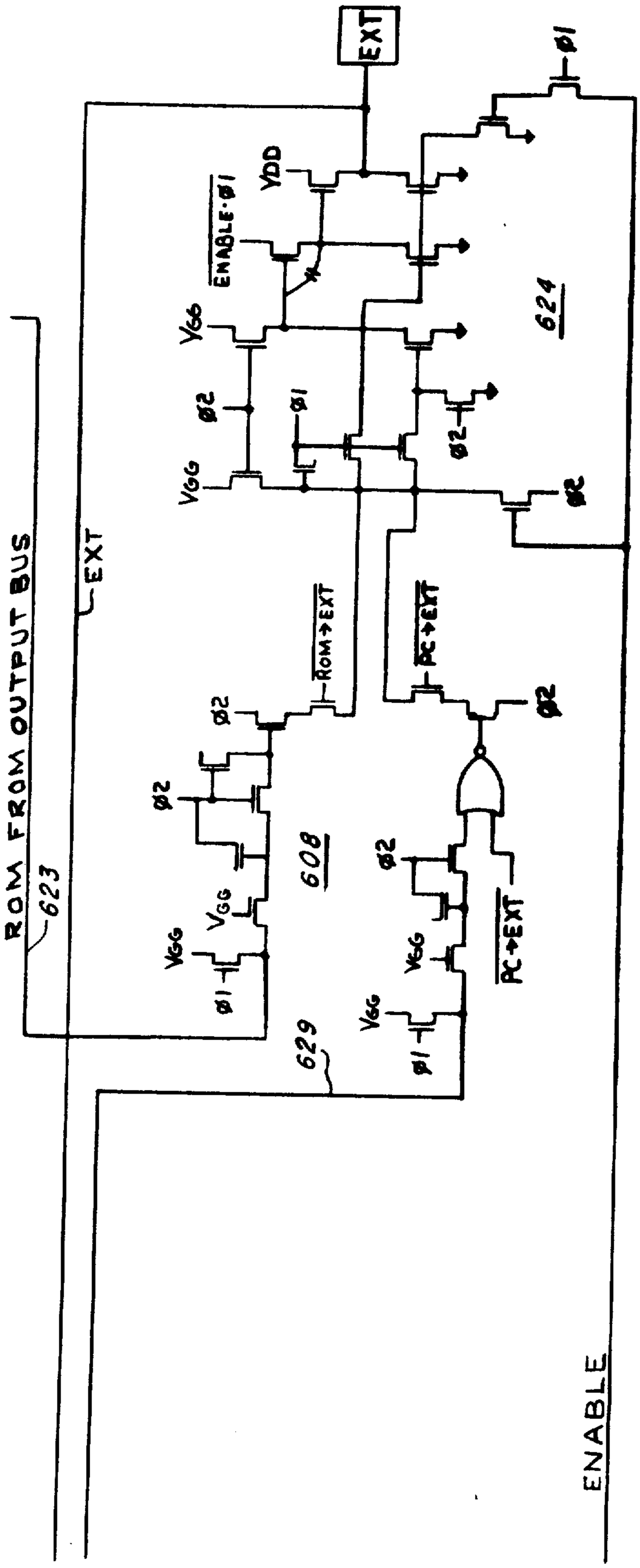
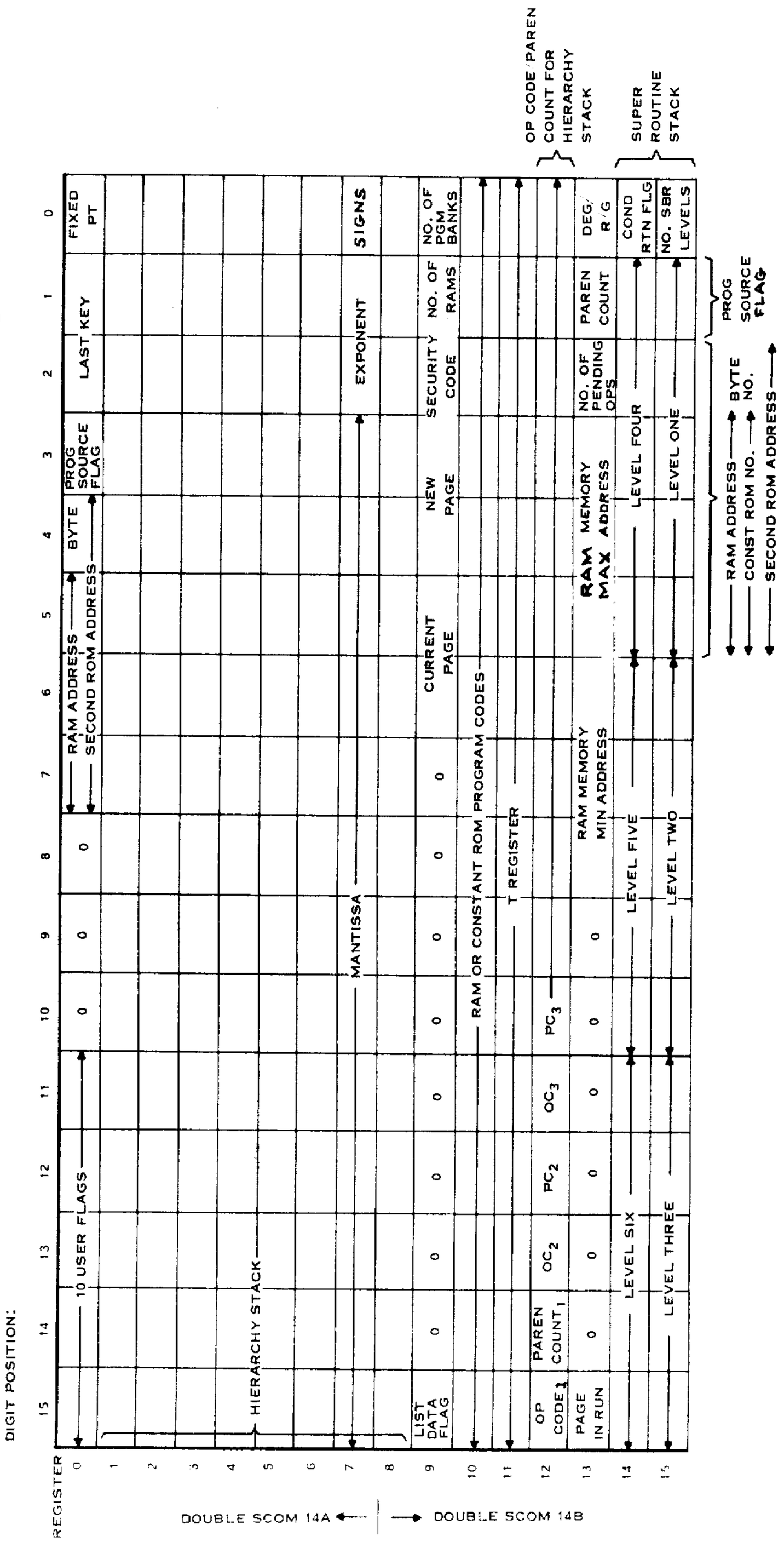

EXT
ROM FROM OUTPUT BUS
623
EXT
624
608
629
ENABLE
ENABLE·Ø1
VDD
VGG
Ø2
Ø1
ROM→EXT
PC→EXT
PC→EXT

Fig. 18i

Fig. 19


DIGIT POSITION:
REGISTER
DOUBLE SCOM 14A
DOUBLE SCOM 14B
RAM ADDRESS
SECOND ROM ADDRESS
BYTE
PROG SOURCE FLAG
LAST KEY
FIXED PT
EXPONENT
SIGNS
MANTISSA
CURRENT PAGE
NEW PAGE
SECURITY CODE
NO. OF RAMS
NO. OF PGM BANKS
RAM OR CONSTANT ROM PROGRAM CODES
T REGISTER
NO. OF PENDING OPS
PAREN COUNT
DEG/R/G
COND RTN FLG
NO. SBR LEVELS
RAM MEMORY MAX ADDRESS
RAM MEMORY MIN ADDRESS
LEVEL FOUR
LEVEL ONE
LEVEL FIVE
LEVEL TWO
LEVEL SIX
LEVEL THREE
10 USER FLAGS
HIERARCHY STACK
LIST DATA FLAG
OP CODE 1
PAGE IN RUN
PAREN COUNT1
OC2
OC3
PC2
PC3
OP CODE PAREN COUNT FOR HIERARCHY STACK
SUPER ROUTINE STACK
PROG SOURCE FLAG
RAM ADDRESS — BYTE
CONST ROM NO. — NO.
SECOND ROM ADDRESS

MICROPROCESSOR SYSTEM HAVING HIGH ORDER CAPABILITY

Matter enclosed in heavy brackets [ ] appears in the original patent but forms no part of this reissue specification; matter printed in italics indicates the additions made by reissue.

This application is a continuation-in-part of Ser. No. 714,464 filed Aug. 16, 1976.

BACKGROUND OF THE INVENTION

This invention relates to microprocessor systems and more specifically to electronic calculators having the capability of solving higher order or complex mathematic problems. It should become evident, moreover, that my invention has utility in other applications making use of microprocessor technology, such as video games, to increase the level of sophistication of the functions performed by the microprocessor.

Electronic calculators have evolved from comparatively simple machines which add, subtract, multiply and divide the data entered into the calculator to machines which can perform sophisticated financial and mathematical operations such as, for example, changing polar coordinates to rectangular coordinates or solving compound or annuity interest problems. The calculator systems developed to date have had a single or multiple read-only-memory (ROM's) in which groups of instruction words are stored as microcode. The different groups of instruction words stored in the ROM cause the calculator's arithmetic unit, memory and display to cooperate to perform desired mathematical operations when instruction words are read out of the ROM. The ROM is addressed or controlled by a keyboard or other input means associated with the electronic calculator or microprocessor system. Thus, the depression of a key causes a selected group of instruction words to be read out of the ROM and these instruction words are provided to circuits controlling the inputting of data, the storage of data and the manipulation of data to perform the operation or function invoked by the key depressed. Thus, the instruction words generated by the ROM cause data to be entered, stored, and manipulated using, for instance, an arithmetic unit to perform such functions as adding, subtracting, multiplying dividing or performing higher order or complex mathematical operations on the data.

Each instruction word typically has a length of eight to sixteen binary bits, although microprocessors having longer or shorter instruction words are well within the state of the art. The performance of even a relatively simple operation, such as adding two floating point numbers, requires a group of many instruction words. For instance, the addition of two floating point numbers may require as many as 75 instruction words having thirteen bits each, thus absorbing 975 bits of ROM area to be able of performing such a simple operation. Similarly, the subtraction operation has a comparable set of instruction words and so on for other operations and functions. Of course, portions of such sets may be shared for certain operations.

Modern electronic calculators now perform sophisticated arithmetic and financial computations such as, for example, squaring, taking square roots, converting from polar to rectangular coordinates, computing logorithms and trigometric relationships, compounding interest,

and other such computations. These computations have typically been implemented into the electronic calculator by increasing the size of the ROM to accommodate the larger number of instruction words associated with these higher order computations. While the size of commercially available ROM's has increased during the past several years, the library of computational programs desired to be implemented in an electronic calculator has grown at even a faster rate. For example, it is desirable to have an electronic calculator capable of performing an entire library of computations related to electrical engineering, mechanical engineering, surveying, or the like. However, an electronic engineering library of computations could comprise, for instance, 45 computational programs or more, each of which require as many as 600 instruction words implemented in a read-only-memory. Thus an entire computational library would include for example, on the order of 27,000 instruction words of thirteen bits each which would require many conventional chips to implement the electrical engineering library in a conventional calculator. Of course, using a large number of chips can significantly increase the cost of an electronic calculator as well as making the packaging for hand-held use more difficult and unduly increasing power consumption.

In the prior art it is also known that an electronic calculator may be provided with the ability to perform higher order calculators by making the calculator programmable and storing the program in a Random Access Memory (RAM) or on a magnetic tape or card. But, in this case, the program is not permanently stored in the calculator, but must be read into the calculator's memories at least each separate time the calculator is energized; thus, such a program is not directly accessible from the calculator's keyboard.

It is an object, therefore, of this invention to improve electronic calculators and microprocessors.

It is another object of this invention to increase the number of computational programs stored in an electronic calculator without correspondingly increasing the size of the ROM(s) implemented in the electronic calculator.

It is a further object of this invention to make such computational programs directly accessible from the calculator's keyboard.

It is another object of this invention to store a large number of computational programs in the hand-held electronic calculator using a small number of chips.

It is yet another object of this invention to selectively equip an electronic calculator or microprocessor with different libraries of higher order function, which functions may be accessed from the calculator's keyboard and/or from a program stored in a programmable calculator.

It is still another object of this invention that the particular library with which a calculator is equipped may be changed by the end user thereof.

The aforementioned objects are satisfied as is now described. Generally, and in accordance with the preferred embodiment of the invention, an electronic calculator is equipped with first and second ROM's. The first ROM stores a plurality of groups of instruction words, each group effective for controlling an arithmetic unit to perform basic arithmetic operations such as adding, subtracting, multiplying and dividing, taking square roots, forming logarithmic and trigonometric operations and so forth in response to the operation of the first set of keys on a calculator keyboard. Each of

these groups of these instruction words contains on the order of 75 to 200 instruction words. Thus, the first ROM is used as a main ROM as in a conventional calculator microprocessor. The second ROM stores a plurality of sets of program codes. Each set of program codes are capable of performing a higher order mathematical program and are read out of the second ROM in response to operation of a second set of keys on the calculator keyboard. Each program code (which comprises eight binary bits in the embodiment disclosed) is effective for addressing a group of instruction words stored in the first ROM in much the same manner as depression of a key in the first set of keys. Thus, a set of program codes may mimic the depression of a plurality of keys in the first set of keys. Therefore, each higher order mathematical program is preferably a series of the basic arithmetic operations stored in the first ROM combined with operations for entry of data and/or constants. The second ROM reads out program codes which serve to address the groups of instruction words stored in the first ROM. By using the second ROM to store such higher order calculational programs, a second set of keys can be used to input commands triggering a long chain of basic arithmetic operations, including data entry operations, without the chance of human error and at a much greater speed than a human operator. Thus, the second ROM, in the preferred embodiment, stores sets of program codes, each program code effective for addressing the first ROM in much the same manner as a single depression of a key in the first set of keys; therefore, a set of such program codes may be advantageously utilized for a large number of higher order calculational programs in an electronic calculator. Since the second ROM must only store on the order of eight bits, or so, to select or address an entire group of instruction words, it should be evident to one trained in the art by utilizing the second ROM herein disclosed that great economies can be effected in total ROM area, when compared with prior art techniques. While I have referred to first and second sets of keys on the calculator keyboard, it is well known that a single physical key may be used to perform several functions and therefore the keys referred to in the first and second sets may be physically the same keys.

In a further aspect, the second ROM is preferably provided by a chip or chips which may be temporarily plugged into the calculator or mocroprocessor system by the end user thereof. Preferably, the calculator or microprocessor system is operated with one or more such second ROM chips, which chips are selected from a group of chips for operation in the calculator or microprocessor system by the end user thereof, according to the end user's particular needs at any given time. In the embodiment disclosed, a major portion of the second ROM is provided by a plugged-in chip and minor portion is provided by a permanently wired-in chip. Thus, in the embodiment disclosed, the end user may select which higher order functions are performable according to which particular second ROM chip is plugged into the calculator while a few high order functions stored in the permanently wired chip is inherently a part of the calculator system disclosed. The second ROM is disposed in a module for ease of handling by the end user of the calculator or microprocessor.

While a calculator system is disclosed herein in detail, it should be evident to those skilled in the art that my invention may also be used in applications other than

calculators where a microprocessor with high order capability may be advantageously utilized.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as further objects and advantages thereof, will be best understood by reference to the following detailed description of an illustrative embodiment, when read in conjunction with the drawings, wherein:

FIG. **1** is a pictorial view of an electronic portable calculator of the type which may embody the invention;

FIG. **2** is a simplified block diagram of a multi-chip calculator system which may be utilized in practicing the present invention;

FIGS. **3a–3b** are detailed block diagrams of the arithmetic chip featured in FIG. **2**;

FIG. **4** is a detailed block diagram of the SCOM chip featured in FIG. **2**;

FIGS. **5a–5e** depict in representative form the instruction words decoded by the arithmetic and SCOM chips;

FIG. **5f** depicts the originization of the EXT signal;

FIG. **5g** depicts the first ROM address as stored in the address register;

FIG. **5h** depicts the instruction words decoded on the second ROM chip and selected instruction words decoded on the arithmetic chip, but which may be conveniently employed in connection with the utilization of second ROM chip;

FIGS. **6a–6b** are timing diagrams showing the timing of various parts of the multi-chip system;

FIG. **7** is a representation of the keyboard input matrix;

FIGS. **8a–8d** are a composite schematic diagram of the arithmetic chip of FIG. **2**;

FIGS. **9a–9e** are a composite schematic diagram of the SCOM chip of FIG. **2**;

FIGS. **10a–10r** are schematics of certain circuits used in FIGS. **8a–8d** and **9a–9e**;

FIG. **11** is a block diagram of a modern electronic calculator equipped with one embodiment of the invention;

FIG. **12** is a block diagram of another embodiment of the invention which may be utilized with a modern electronic calculator of the type depicted in FIG. **11**;

FIG. **13** is a pictorial view of an electronic calculator having an opening for removeably receiving a packaged second second ROM chip;

FIG. **14** is a simplified block diagram of a multi-chip calculator system utilizing the present invention;

FIG. **15** is a function diagram of the logical organization of data stored in the second ROM;

FIG. **16** depicts the variable boundary between data and program steps in the calculator's memory;

FIG. **17** is a block diagram of the second ROM chip;

FIGS. **18a–18i** form a composite schematic diagram of the second ROM chip; and

FIG. **19** is a representation of data stored in the memory registers on the double SCOM chips.

LOCATION OF THE DRAWINGS

FIGS. **1**, **5h**, **7**, and **11–19** accompany this patent. FIGS. **2**, **3a–3b**, **4**, **5a–5g**, **6a–6b**, **8a–8d**, **9a–9e** and **10a–10r** are hereby incorporated by reference from U.S. Pat. No. 3,900,722, entitled "Multi-Chip Calculator System Having Cycle and Subcycle Timing Genera-

tors", which issued on Aug. 19, 1975 to Michael J. Cochran and Charles P. Grant, Jr. and which is assigned to the assignee of this invention.

CONCEPTUAL DESCRIPTION

Referring to FIG. **1**, an electronic portable calculator of the type which may employ features of this invention is shown in pictorial form. The calculator **1** comprises the keyboard **2** and the display **3**. The display **3**, in one embodiment, consists of twelve digits or characters, each provided by an array of light emitting diodes or characters, such provided by an array of light emitting diodes, a liquid crystal display, gas discharge tube or other display means. The display is preferably implemented to having eight mantissa digits, two exponent digits, and two character places for negative signs, etc., (one for the mantissa and one for the exponent), thereby permitting outputting the data in scientific notation for instance. Of course, the type of display and the number of digits displayed is a design choice. Ordinarily, the diplay would be of the seven segment or eight segment variety, with provisions for indicating a decimal point for each digit. The display **2** includes a number of keys (0–9), a decimal point key, the conventional plus (+), minus (−), multiply (×), divide (÷), and equal (=) keys. Further the keyboard perferably includes keys for exponentation (Y^x and inverse Y^x) and trigonometric relationships (Sine X, Cosine X, and Tangent X). The calculator is further provided with OP Code Keys for performing special functions such as slope, intercept, plotting operations, alphanumeric operations and the like. Further, the calculator may be provided with keys for storing (STO) and recalling (RCL) data from memory, for clearing the calculator (CLR) and for clearing the last entry (CE). The keys used to access higher order functions will be described subsequently.

In FIG. **11**, there is shown in block diagram form, the basic elements of a modern electronic calculator implemented on one or more semiconductor chips. It is to be understood that the block diagram of FIG. **11** is not intended to represent the block diagram of a detailed representation of electronic calculators, but is merely intended to indicate how the additional elements of an electronic calculator system having higher order capability are incorporated into a typical electronic calculator. Subsequently, it will be explained in detail how my invention may be praticed with the multi-chip calculator system depicted in FIGS. **2–19**. The calculator of FIG. **11**, is shown with a clock **40** which provides clocking signals for transferring data throughout the electronic calculator and provides scanning signals for scanning the display **3** and keyboard **2** or other data entry means. The inputs for the keyboard **2** are provided to keyboard logic **41** which provides an address in response to the depression of a particular key to program counter **23**. It should be evident to one skilled in the art that keyboard logic **41**, as well as other logic circuitry, may be implemented in the calculator as the elements described or may be implemented as a part of read only memory **20** and instruction word decoder logic **96**.

The address received from keyboard logic **41** is inserted into program counter **23** and is utilized in addressing the First Read-Only-Memory (ROM) **20**. First ROM **20** contains the microcode for performing basic arithmetic operations and outputs an instruction word in response to the address contained in program counter **23**. Program counter typically includes an add-one circuit for incrementing the address in program counter **23**. Thus, program counter **23** causes a group of instruction words to be read out of First ROM **20** in response to the incrementing of program counter **23**, each instruction word being read out during an instruction cycle. The group of instruction words read out of First ROM **20** corresponds to the address received from keyboard logic **41**.

The instruction words read out of First ROM **20** are decoded by instruction word decoder logic **96** to provide instruction commands to program counter **23**, arithmetic unit **45** and data control **44**. The instruction commands provide to program counter **23** enable branches to be executed by inserting a new address into program counter **23** in response to a branch instruction command stored in First ROM **20**. Instruction commands provided to data control **44** and arithmetic unit **45** control the manipulation of numeric data in the calculator. Instruction word decoder **96** is also interconnected with a counter **47** and a latch **46** in my electronic calculator system having higher order math capability.

Data control **44** is interconnected with display register **49**, operational registers **43** and with the arithmetic unit **45**. Display register **49** stores the number displayed by the display **3** and has associated therewith a plurality of operational registers **43** which are used in conjunction with arithmetic unit **45** to perform arithmetic operations in response to particular instruction commands. Output drivers **42**, interconnect display register **49** with a display **3** for decoding the electrical signal, stored in display register **49** and for driving display **3**. Data control **44** comprises a series of selector gates for interconnecting the appropriate operational registers **43** and display register **49** with the arithmetic unit **45**, with portions of instruction words, (if need be), or with logic signals from keyboard **2** (if need be).

Numeric data is inputted into display register **49** from keyboard **2** either by a data path from keyboard logic **41** via data control **44** under the control of appropriate instruction commands or by inputting selected portions of an appropriate instruction word in response to selected instruction commands. The electronic calculator system hereinbefore described, that being the portion shown within the reference a dashed line in FIG. **11**, basically corresponds to the type of electronic calculators known in the prior art. Exemplary of the prior art calculators systems is the calculator system depicted in FIGS. **2–10**.

Also in FIG. **11**, there is shown a counter **47** and a latch **46** which is responsive to outputs from instruction word decoder logic **96**. The counter **47** has an output for addressing a Second ROM **48**. Second ROM **48** outputs a program code in response to the inputted address, the program codes being outputted via latch **46** to program counter **23**. When keyboard logic **41** decodes keyboard outputs indicating that a higher order math calculation is to be executed, the higher order math calculation being preferably a series of basis arithmetic functions and operations of the type implemented in first ROM **20**, keyboard **41** preferably input an address into program counter **23** which causes First ROM **20** to branch to a location therein for calling a program from Second ROM **48**. When a program is called from Second ROM **48**, instruction word decoder logic **96** first sets latch **46** to permit the program codes outputted by Second ROM **48** to be inputted into program counter **23**. The program codes outputted by Second ROM **48** effectively transmit an address into program

counter **23** for addressing First ROM **20**. The first such code preferably causes the First ROM **20** to branch to a location for performing the first basic arithmetic operation or function required by the Second ROM **48** program. The program codes may take the same logical format, for instance, as the output from keyboard logic **41**. When calling a program from Second ROM **48**, instruction word decoder logic **96** also transmits an address into counter **47**, the address being the first location in Second ROM **48** of the called program. It should be evident, moreover, that counter **47** could be loaded with an address directly from keyboard logic **41** in lieu of from instruction decoder logic **96**, this being essentially a design choice.

After the first program code is read out of Second ROM **48** via latch **46** and loaded into program counter **23** then First ROM **20** cycles through a group of instruction words to accomplish the indicated basic arithmetic operation or function. Of course, the number of instruction cycles required to accomplish the indicated operation or function depends on, for instance, a number of instructions contained for that basic operation or function in First ROM **20**. As is well known, those operations or functions which are accessible via keyboard logic **41** from keyboard **2**, usually contain instruction words for causing the display to be enabled at the end of the function or operation addressed in First ROM **20**. Since, however, another program code is to be read from Second ROM **48** and inserted into program **23** upon accomplishment of the indicated function or operation, counter **47** includes an add-one circuit which is responsive to, for instance, a display command or other such commands located near or at the end of a group of instruction words in First ROM **20** for accomplishing a basic arithmetic operation or a function. When the display command or other such command is decoded by instruction word decoder logic **96**, the add-one circuit in counter **47** increments and causes Second ROM **48** to read out the next program code of the called program via the set latch **46** to program counter **23**, which in turn causes First ROM **20** to cycle through another group of instructions to accomplish the function or operation indicated by the outputted program code. Again, towards the end of this next basic arithmetic function or operation, a display code or other such code will be decoded in instruction word decoder logic **96** causing the add-one circuit in counter **47** to increment counter **47**, the cycle repeating itself.

The advantages of Second ROM **48** and associated counter **47** and latch **45** should be evident to one trained in the art. This system permits equipping an electronic calculator with the capability of performing higher order calculation programs: for instance, changing polar coordinates to rectangular coordinates, doing financial calculations or solving complex engineering equations using significantly less total ROM area than would be required if such programs were implemented only in First ROM **20**. Additionally, it should be evident that while the foregoing discussion has suggested that a program code read from Second ROM **48** mimics keyboard logic outputs from keyboard logic **41**, the program codes read from Second ROM **48** could, in lieu thereof or in addition thereto, have codes which do not mimic the outputs from keyboard logic **41**, but rather, for instance, would cause the program counter **23** to branch to locations in First ROM **20** which are not directly accessible from the keyboard. Thus an output from Second ROM **48** may cause program counter **23** to branch to a location in First ROM **20** which could not be accessed directly from the keyboard **2** via keyboard logic **41**. One purpose for such a program code would be a program code in a called program to indicate that the end of the program had been reached. This program code, which I shall refer to as the "return" program code, preferably causes program counter **23** to branch to an address location in First ROM **20** which would contain a group of instructions for resetting latch **46** and for displaying the contents of display register **49**. The display instruction preferably follows the reset latch instruction, so that when the display command causes counter **47** to increment (if so used), no branching will occur in response thereto at program counter **23**. Also latch **46** inhibits outputs from Second ROM **48** from being inserted into program counter **23** whenever a display instruction or other such instruction is decoded by instruction word decoder logic **96** incrementing the add-one circuit in counter **47** when the calculator has not called a program from Second ROM **48**. Referring now to FIG. **12**, there is shown a partial block diagram of a second embodiment of my calculator system having higher math capability. The latch **46** and counter **47** are interconnected with program counter **23** and instruction word decoder **96** as done in the embodiment shown in FIG. **11**. In fact, this embodiment is similar to the embodiment in FIG. **11**, except that a superroutine stack register **91** and an associated superroutine latch **92** have been interconnected with counter **23**; stack **91** is responsive to outputs from instruction word decoder logic **96**. Thus, the program codes outputted from Second ROM **48** are passed to program counter **23** via latch **46**; counter **47** is used to address Second ROM **18** and is responsive to instruction word decoder logic **96** for inserting an initial address therein and for incrementing that address in response to decoded display commands, for instance. The superroutine stack **91** functions to either receive an address from counter **47** or to output an address to counter **47**, both functions being in response to outputs from instruction word decoder logic **96**. Superroutine stack **91** is a multi-level stack and functions in normal last-in-first-out mode. Superroutine stack **91** may be advantageously utilized in calculator systems with higher math capability so that the program codes stored in Second ROM **48**, in addition to prescribing addresses for performing basic arithmetic operations and functions according to microcode stored in First ROM **20**, may also use First ROM **20** for addressing Second ROM **48** itself. The advantages of this superroutine stack **91** may be best seen by example. For instance, Second ROM **48** may be implemented with program codes to perform the factorial function, and during the calculation of statistical programs, such as combinations and permutations, it is often advantageious to be able to use the factorial function. If a factorial function and the statistical combination function are both implemented in Second ROM **48**, then the combination function program may call the factorial function, if a superroutine stack **91** is utilized. The point of exit from the combination function program must be stored so that the program can return thereafter accomplishing the factorial function. Thus, the address in counter **47** to which the program must return in Second ROM **48** after accomplishing the factorial function is stored in superroutine stack **91** because a new set of addresses will be loaded into counter **47** when the factorial function is executed. Whenever an address has been stored in superroutine stack **91**, superroutine latch **92** is set. Further, as afore-

mentioned, the factorial program will preferably have a "return" program code loaded in Second ROM **48** at the end thereof. This return code normally causes latch **46** to be reset. However, the return code is inhibited from setting latch **46** when superroutine latch **92** has been set. Latch **46** is not reset at this time because the program codes being read from the Second ROM **48** must continue to be inserted in program counter **23** to carry out the main program, eg, the combination function program in the aforementioned example. Although the return code is not used to reset latch **16** if superroutine latch **92** is set, the return code is used to "pop" the address in the stack back into counter **47**. Since stack **91** is a multi-level stack, several levels of "superroutines" may be utilized.

Preferably, the second ROM **48** is implemented by a ROM chip which is provided with a package permitting it to be plugged into the aforementioned electronic calculator. The second ROM chip is preferably packaged in a form to facilitate handling by the end user of the calculator to permit easy installation of that chip into the electronic calculator. Preferably, the electronic calculator system of this invention receives, at any given time, one second ROM chip, but the operator thereof selects which particular second ROM chip is plugged into the calculator system. The operator ROM chips each programmed to perform different types of high order functions. For instance, one or a plurality of second ROM chips may be provided for performing statistical problems while another ROM chip or chips is provided for performing surveying problems; still yet another second ROM chip or chips may be provided for performing aviation or navigational problems, for instance. Thus the end user of the calculator can configure a basic calculator to perform many different types of high order functions depending upon the particular library available in the particular second ROM chip or chips plugged into the electronic calculator of this invention.

Referring to FIG. **13**, there is shown an electronic calculator having an opening **4** for exposing contacts **5**, which are connected to the electronics of the calculator. Opening **4** is preferably provided on the rear side of the calculator case **1** as shown in the FIG. **1** and forms, with contact **5**, a receptacle for receiving module **48**a. Opening **4** is adapted to removably receive the second ROM **48**, which is not shown in FIG. **13**, but which is disposed in module **48**a. Second ROM chip module **48**a has contacts (not shown) which mate with contacts **5** for connecting the second ROM **48** therein to the electronic calculator. Door **6** may be closed to retain module **48**a in opening **4** during normal operation.

THE SPECIFIC EMBODIMENT IN A PROGRAMMABLE CALCULATOR

Having described how the second ROM is advantageously used with an electronic calculator, a particular embodiment of the second ROM in a particular calculator is now described.

Referring now to FIG. **14**, there is shown a detailed block diagram of an specific embodiment of a programmable electronic calculator employing the second ROM **48** of this invention. In FIG. **14**, there is shown a plurality of chips (**48'**, **10**, **12**a, **12**b, **13**, **14**a–d and **18**). Chips **10**, **12**a, **12**b, **13**, and **18** have heretofore been described in some detail in prior U.S. Patents and Patent Applications and therefore reference will be made to U.S. Patents or U.S. Patent Applications, as the case may be, for a detailed description of these chips. The following discussion will basicly relate to how chips **10**, **12**a, **12**b and **13** cooperate with a second ROM chip **48'**; which is described hereafter in detail, to implement a calculator having high order capability.

The calculator's arithmetic chip **10** has a plurality of Registers **50**a–**50**e for storing numeric data, an Arithmetic Unit **55** for performing arithmetic operations on the data stored in Registers **50**a–e, Flag Registers **53**a–b for storing a plurality of flags, a keyboard register **54** which is (1) loadable with a decoded keyboard address derived from the calculator's keyboard, (2) loadable from a subroutine register or (3) loadable from the second ROM chip **48'**. Arithmetic chip **10** is described in detail in aforementioned U.S. Pat. No. 3,900,722 which issued to Michael J. Cochran and Charles P. Grant, Jr. on Aug. 19, 1975 and which is assigned to the assignee of this invention. Line 21, column 4 through line 31, column 44 of U.S. Pat. No. 3,900,722 is hereby incorporated herein by reference.

U.S. Pat. No. 3,900,722 discloses a multiple chip calculator system employing the aforementioned arithmetic chip **10** and a scanning and read-only-memory (SCOM) chip. U.S. Pat. No. 3,900,722 discloses that eight SCOM chips may be utilized in a single calculator system. Referring again to FIG. **14**, chips **12**a and **12**b are each double SCOM chips; a double SCOM chip is the equivalent to two SCOM chips of the type disclosed in U.S. Pat. No. 3,900,722 implemented on a single chip of silicon, with the F and G registers thereof replaced by a single eight register memory of the type disclosed in U.S. Patent Application Ser. No. 745,157 which was filed Nov. 26, 1976 and which is assigned to the assignee of this invention.

External ROM chip **13** provides for increased instruction word storage capacity. The ROMs **20**a and **20**b on double SCOM chips **12**a and **12**b and the ROM **20**c on chip **13** provide the first ROM **20** for storing the microcode which controls the operation of the calculator system. The microcode stored in ROM's **20**a–**20**c is listed in Tables IIa–IIc, respectively. ROM **20**c is a 1 K×13 bit ROM while ROMs **20**a–**20**b are each 2.5 K×13 bit ROMs.

Referring briefly to Tables IIa–IIc, the first column thereof is the hexidecimal address of the microcode instruction word appearing in the third column. The second column identifies the chip in which the microcode is stored. TMC-582 and TMC-583 are the two double SCOM chips **12**a and **12**b; TMC-571 is the external ROM chip **13**. The fourth through nineteenth columns contain instruction words whose addresses are incremented by one for each column, reading from left to right. Thus, in Table IIa, the seventeen instruction words in the first row, columns three through nineteen are located at hexidecimal addresses 0000 through 0010. The instruction words are in hexidecimal format also and correspond to the instruction words identified in FIGS. **5**a–**5**h.

As explained in U.S. Pat. No. 3,900,722, the arithmetic chip **10** and the double SCOM chips **12**a and **12**b are interconnected by lines for exchanging the following control signals: external (EXT), input/output (I/O), instruction words (IRG), and $\overline{\text{IDLE}}$. External is a serial data channel which may be used, for instance, for addressing ROMs **20**a–**20**c using an address stored in the keyboard register **54** when the PREG bit thereof is a logical one or for inputting or outputting serial data depending on the instruction word outputted on IRG

(when the PREG bit is a logical zero). I/O is a four bit parallel data channel for transferring data in bit parallel, digit serial fashion under control of construction words outputted from ROMs **20**a–**20**c. IRG is a serial channel for transmitting the instruction word from the particular ROM **20**a–**20**c controlling the oeration of the system.

In FIG. **14**, there are shown four multi-register chips **14**a–**14**d which are connected to the I/O, IRG, and $\overline{\text{IDLE}}$ lines. These multi-register chips are essentially random access memory (RAM) chips which are utilized for storing the data used by the calculator system and programmed functions. It should be evident that the numbers of such chips as well as the size of the RAMS thereon is a design choice.

The magnetic card reader chip **35** is responsive to EXT, IRG and $\overline{\text{IDLE}}$ for inputting digital information to the calculator system from magnetic cards or outputting digital information from the calculator system to magnetic cards. Chip **35** is described in greater detail in U.S. Pat. Application 622,288 filed Oct. 10, 1975 and now U.S. Pat. No. 4,006,455. Of course, the use of a card reader is a design choice. If chip **35** is not utilized, the diode and switch **7** shown in FIG. 7 should be omitted. Switch 7 closes in response to a card being inserted into the card reading mechanism associated with chip **35**.

Printer chip **18** may be used to provide the calculator of this invention with printing capability. It should be evident that the utilization of printer chip **18** is a design choice and further this chip may be either permanently installed in a printer calculator or may be installed in a print cradle, such as the PC 100a cradle manufactured by Texas Instruments Incorporated of Dallas, Texas which print cradle may be interfaced with a hand-held calculator provided with printing capability. Chip **18** is described in greater detail in U.S. Pat. Application Ser. No. 428,492 filed Dec. 26, 1973 and now U.S. Pat. No. 4,020,465.

The second ROM chip **48**′ is interconnected with the calculator system via external, IRG and $\overline{\text{IDLE}}$. Chip **48**′ includes a second ROM **48** of the type heretofore discussed plus various control circuits for interfacing it with the remainder of the calculator system disclosed. As previously mentioned, second ROM chip **48**′ is preferably removable from the calculator of this invention and therefore a plug assembly **43** is provided for ease of removal and insertion. Preferably, the calculator of this invention is provided with a plurality of such second ROM chips **48**′, at least any one of which may be connected into the calculator system at any given time. This plurality of ROM chips **48**′ are programmed to provide different types of problem solving capabilities. For instance, one chip **48**′ might be implemented with programs for solving statistical problems while another might solve financial, surveying, navigation, medical, mechanical or electrical engineering problems, or the like. Moreover, it should become evident to those skilled in the art, that a plurality of such chips **48**′ might be interfaced with a calculator at one time if such chips were provided with a chip selection means for identifying which second ROM chip **48**′ is being addressed at any given time. Such chip selection circuits, while not used in the embodiment herein disclosed, are well known in the art.

While the second ROM of this invention is described as a read-only-memory, it should be evident to those skilled in the art that second ROM might be an electrically alterable device, such as an EPROM, or the like. Similarly, a bubble memory or other such non-volatile memory means could also be utilized as a second ROM.

In Table VIII there is a listing of program codes used in a general purpose second ROM chip **48**′ to perform such operations as: performing a diagnostic checks, complex math operations, matrix math operations, matrix inversion, annuity and compound interest operations, permutation and combination calculations and the like. The program codes are listed in columns 3–19 of Table VII. The address of the program code in column three is given in column one and the addresses of the other program codes on the same line increment by one for each column reading from left to right.

ORGANIZATION OF PROGRAMS STORED IN THE SECOND ROM

As it has been previously discussed, the second ROM stores a plurality of program codes for performing high order functions. The organization of these program codes on chip **48**′ is now described in detail. In this embodiment, the program codes comprise a pair of four bit binary coded decimal (BCD) digits. Therefore, these codes may be any number between 00 and 99.

Referring now to FIG. **15**, there is shown a functional diagram of how the program codes are organized on the second ROM of chip **48**′ and preferably a second ROM implemented in a pluggable package. In this embodiment ROM **48** stores on the order of 5,000 eight bit program codes. Referring now to FIG. **15**, a rectangle thereon represents an eight bit code outputtable from ROM **48** in response to an address. Second ROM **48** stores a plurality of programs, which are for ease of addressing, arranged on "pages". Several programs are preferably allocated to each page. When a program is second ROM **48** is to be accessed from the calculator's keyboard, the operator depresses the "2ND" key and the "program" key (PGM) in this embodiment. The operator next enters a two digit number indicating the page upon which the program he or she wishes to access exists. For instance, if he or she wishes to access a program on page twelve, he or she would depress the one and two number keys. The operator knows upon which page the desired program exists because a program directory is preferably supplied along with a pluggable second ROM chip **48**′. The operator then preferably enters a label to uniquely identify the particular program which is desired on the page previously entered. This is done by depressing either a particular label key A—E or A′—E′ or the subroutine key (SBR) followed by a non-number key (e.g., SBR, =; or SBR, X^2; or the like). Depressing the subroutine key and entering a three digit address preferably causes a branch to the location equal to the sum of the inputted address plus the address of the first program code on the inputted page.

The calculator is preferably permanently programmed to first read out the program code at location 0000 which indicates the number of pages stored on that particular second ROM **48**. This number is compared with the inputting page number to assure that the inputted page number exists in second ROM **48**. Next the security code at location 0001 is preferably outputted for examination, the function of which will be later described. The next step is to address second ROM **48** with the entered page location, the address being derived by multiplying the inputted page number by two. For example, if page 02 is entered, then the address to

be used is 0004. At address 0004 is a top half of the address (the thousands and hundreds digits) for the beginning point of the second page. At address 0005 is the bottom half of the address (the tens and units digits). The program codes at locations 0004 and 0005 define the address where the second page begins in second ROM **48**. Locations 0006 and 0007 will also be read out to provide the address of the beginning point of the third page, which is indicative of the ending point of the programs stored on the second page. Thus the address of the second page derived from locations 0004 and 0005 is used as the starting point for a label search and the address of a third page is used to define the ending point of that search.

The program in second ROM **48** is caused to branch to the program code which occurs at the starting point of page two. At page two in second ROM **48**, the label search is commenced by reading out program codes sequentially until either the label being searched for is detected or the beginning point of page three is encountered, indicating that the label being searched for does not exist on the page selected. The label being searched for is either a particular label program code (Table III code 10–19) or the label program code (Table III, code 76) followed by a particular non-numeral program code. When the last page is selected, then the address of the last page, as well as the last address on that page are read out to fulfill the function of reading out the addresses of pages 2 and 3 in the foregoing example. This sequence of events is also diagrammatically depicted in FIG. **15**.

Referring now to Table III, there is shown a list of the program codes 00–99 preferably used in the calculator system of this invention along with the corresponding functions performed by these codes and the key sequences used to generate the codes when generated from the keyboard. As can be seen, certain program codes may not be directly generated from the calculator's keyboard. The functions performed by the program codes listed in Table III should be evident to those skilled in the art based on the description set forth in Table III. By way of further clarification, however, the inverse function key (INV) is used to perform the inverse of the function indicated for selected keys. For instance, the inverse function key when combined with the LNx key causes the calculator to take the number e^x in lieu of taking the natural logorithm of the number x. The indirect addressing key (IND, which must be used in combination with the 2ND key, of course) is used with the memory operation keys and "go to" or "conditional go to" keys (GTO, X×T or X≧T) to indicate that the number following the program code does not describe either the memory used (if a memory operation) or the branching location (if a go to or conditional go to operation), but rather identifies the particular memory whose contents define either the particular memory to be used (if a memory operation) or the branch address (if a go to type instruction).

Referring again to Table III, program codes 00–09 define the ten numeral keys and the remaining program codes are defined according to the following convention. The first number thereof identifies the keyboard row in which the key is located and the second number defines the keyboard column in which the key is located, for the basic functions which may be accessed by a single key push. For functions accessed by multiple key push sequences, selected merged program codes are utilized. For instance, when the 2ND key is combined with another key to perform the operations indicated, the number 5 is added to the basic program code (without a carry) to generate the merged program code. Thus, for example, the label A is stored as a program code "11" whereas the label A', which requires the 2ND key to be actuated before the A key, is stored as program code "16". Program codes which otherwise would define those keys performing the numeral functions (eg, 0–9), are reserved for selected merged program codes or for program code not directly generated at the keyboard. For example, program codes 62, 63, 64, 72, 73, 74, 83 and 84 are used for merged program codes wherein the IND key is used. Program codes **82** and **92**, which are not defined according to the foregoing convention, define a heirarchy address function and the "return" function. The "return" function has already been mentioned and the heirarchy address function is used to address the eight registers on one of the double SCOM chips, which are set aside for heirarchy control purposes. This calculator system utilizes the algebraic operating system disclosed in U.S. Patent Application Ser. No. 708,958 filed July 26, 1976, for heirarchy control purposes. The heirarchy address code (82) is followed by another program code to define the heirarchy register and operation involved or to define a conditional return, whose function will be mentioned later. The meaning of the program code following the heirarchy address code is set forth in Table IV.

The use of such codes which are not directly accessible from the keyboard permit the accomplishment to special functions or entry in to date areas which are normally isolated from the operator.

The operation code (OP) is used with a following program code for calling the special functions identified in Table V. These routines are implemented in this calculator either in microcode alone or by using second ROM addressing techniques. The second ROM are for such operation code functions is located in the constant ROM areas of double SCOM chips **12a** and **12b**. Approximately half of the constant ROM in double SCOM chip **12a** is used for storing constants in the manner contemplated by U.S. Pat. No. 3,900,722 while the other half of that constant ROM and all of the constant ROM in double SCOM **12b** is used for storing program codes, as defined in Table III, in the manner generally set forth in U.S. Patent Application Ser. No. 714,464, filed Aug. 16, 1976. The contents of the constant ROMs are listed in Table VI hereof. Eight two digit program codes are stored in each constant storage area in the constant ROMs. The codes are stored from right to left; thus the first program code in constant area sixteen is an 82. The addresses of these program codes for discussion purposes will be: Constant number hyphen one of eight locations. Thus, the address of the first program code on constant area sixteen (82) is 16-0, while the third program doce in constant area eighteen (43) is 18-2. Locations 16-0 through 24-2 contain a slope-intercept routine. The other routines in the constant ROM area are defined in Table VIa. As can be seen, the functions stored in the constant ROM areas are accessed either by OP codes or by normal keyboard entries; for example, the polar to rectangular conversion function is stored in the constant ROM areas and is accessed by depressing the P→R key on the keyboard.

The program codes in Table VI make use of the conditional return program code (82 followed by 20). The conditional return evokes a return function only when all the program codes for the accessed function

have been read out. For instance, if the variance is being calculated, the conditional return stored at locations **26-2** and **26-3** is ignored so that after having found the mean according to the program codes at locations **24-3** through **29-2** is performed. The hierarchy address function (program codes 82,-) used with several routines to address the hierarchy registers in order to maximize the addressable storage area available in RAM chips **14a–14d**.

STORING A KEYED-IN PROGRAM

When the operator desires to utilize his or her own program in lieu of a program stored in the first or second ROM's, he or she may do so by an appropriate key sequence for storing keyboard enterable program codes of Table III in RAMs **14a–14d**. RAMs **14a–14d** may also be used for storage of numeric data, i.e., the results of the calculations performed by this electronic calculator. Normally, RAMs **14a–14d** provide a storage for storing 480 program codes while RAMs **14c** and **14d** provide 60 addressable memory locations for storing numeric data. By depressing 2ND, OP, **1**, **6**, the operator may determine which data configuration RAMs **14a–14d** are in; in this case, the number 479.59 would be outputted. The number to the left of the decimal point is the maximum address in RAMs **14a–14d** for program steps, while the number to the right is the maximum address in RAMs **14a–14d** of memory registers.

By inputting a number between one and ten, 2ND, OP, **1**, **7**; the inputted number is used as the number of decodes of memory registers set aside in RAMs **14a–14d** and the resulting configuration is displayed in the aforementioned manner. For example, inputting 7, 2ND, OP, **1**, **7** results in RAMs **14a–14d** being repartitioned with seventy memory registers and 400 program step locations; also 399.69 would appear in the display.

RAMs **14a–14d** can store up to 120 sixteen digit words and can be partitioned to store as many as 960 program codes with no addressable memory registers to as few as 160 program codes with 100 addressable memories in this embodiment. As can be seen from FIG. **16**, the addressable memory locations may be traded at the rate of 10 for 80 program step locations when repartitioning takes place. Of course, the precise number of memories as well as the range of possible data configurations is a design choice.

The partitioning data is stored in digits **3–8** of register **13** on double SCOM chip **14b** (see FIG. **19**). Digits **8-6** holds the actual address in RAMs **14a–14d** defining the location of memory **00** while digits **5—5** hold the largest number assigned to a memory. Thus, in the normal 479.59 configuration, 060 is stored in digits **8-6** while 059 is stored in digits **5-3**. Attempting to branch to a program location equal to or greater than the contents of digits **8-6** causes an error condition, as does addressing a memory greater than the number in digits **5-3** of register **13**.

Referring again briefly to FIG. **14**, the down load operational code may be utilized to permit a page in second ROM **48** to be loaded into the program code storage area of RAMs **14a–14d**. The security code in the second ROM **48** to be down loaded must be a 00 to permit the down loading to occur. A 01 security code inhibits the down loading operation, thereby helping to maintain the secrecy of the programs stored in second ROM **48** should that be desired. If the security code bit is set, then the programs in the second ROM may be utilized to perform the function indicated but the series of program codes may not be read out of the calculator to the operator. After a program is down loaded into RAMs **14a–14d**, the operator has free access for examining the program and altering it as he or she sees fit.

ALPHANUMERIC PRINTING OPERATIONS

Operational (OP) codes 00–07 are used for alphanumeric printing operations when the calculator of this invention includes a printer. For example, a thermal printer in combination with printing chip **18**, such as that provided by the PC-100a disk unit manufactured by Texas Instruments Incorporated of Dallas, Tex. may be used. The twenty character position PC-100a printer may be utilized for printing a line of alphanumeric characters chosen by the operator by loading five two-digit character codes from the display register into four printing buffers. These two-digit alphanumeric character codes are listed in Table VII. The display register is first loaded with five two-digit codes from the keyboard, which are transferred to one of the buffers by OP01–OP04 codes; of course, each buffer stores one fourth of a line of characters.

The contents of the buffers are printed by a 2ND, OP, 0, 5 key sequence or by encountering 69, 05 program codes from either the RAMs **14a–14d** or second ROM **48**.

OP code 06 causes the printing of the contents of one buffer and the numeric contents of the display register. OP code 07 prints an asterisk in the column corresponding to the interger portion of the number then in the display register, provided that number is in the range of 0–19. Thus, OP code 07 is preferably used for plotting a series of answers obtained by the calculator. Of course, the range 0 to 19 is a design choice and, of course, to make better use of this plotting capability, the answers to be plotted are preferably first normalized to occur within the range 0 to 19.

DESCRIPTION OF THE SECOND ROM CHIP AND THE INTERFACE BETWEEN THE SECOND ROM CHIP AND THE OTHER CALCULATOR CHIPS

Referring now to FIG. **17**, there is shown a block diagram of second ROM chip **48'**. The second ROM **48** implemented thereon, is provided by binary coded decimal (BCD) ROM **600**. The output of BCD ROM **600**, which outputs the program codes listed in TABLE III, for instance, is connected to serializers **605** and **606**. Serializer **605** converts both the high digit and low digit of the program code to serial format and supplies it to a digit selector. The serializer **606** converts only the high digit of the program code to serial and also supplies it to digit selector **607**. Digit selector **607** provides the output of serializer **605** to external output control **608** in response to a decoded FETCH instruction or supplies the output of serializer **606** to external output control **608** in response to a decoded FETCH HIGH instruction.

BCD ROM **600** is addressed by an address in program counter **601**. The address in program counter **601** preferably is a BCD four digit numeral. The address in program counter **601** is incremented each time a FETCH (but not a FETCH HIGH) instruction is decoded and is maintained in BCD format by an one bit/BCD corrector **604**. Program counter **601** may be loaded one digit at a time with a digit appearing on External (EXT) via digit controls **602**a–**602**d. Digit strobe **603** controls which one of the digit control

602a–602d is enabled; digit strobe 603 sequentially enables digit controls 602a–602d in response to a decoded UNLOAD PC (unload program counter) or a LOAD PC (load program counter) instruction. Digit strobe 603 is reset by a decoded FETCH instruction. The digit control 602a–602d enabled by digit strobe 603 inputs a single digit (four bits) of the four digit number from the EXT bus in response to a decoded LOAD PC instruction, the number being obtained from the bits occurring during state times S3–S6 on the EXT bus. Also, the enabled digit control 602a–602d outputs a digit from program counter 601 to the EXT bus in response to a decoded UNLOAD PC instruction.

State time generator 609 outputs an indication to state time PLA 610 of which one of the 16 possible state times of an instruction cycle the calculator is in. State time generator 609 is responsive to IDLE for maintaining the state times generated thereby in phase with state times generator on the arithmetic chip 10 or double SCOM chips 12a and 12b, for instance. State time PLA 610 outputs selected timing signals to the various logic circuits and also to serial instruction decode 611. Serial instruction decode 611 is also responsive to the serial instruction words appearing on IRG. Serial instruction decode 611 decodes the aforementioned FETCH, FETCH HIGH, UNLOAD PC and LOAD PC instructions.

External output control 608 outputs the serialized low and high program code digits or the serialized high digit alone from digit selector 607 in response to the FETCH and FETCH HIGH instruction, respectively, and outputs the digit from the enabled digit control 602a–602d in response to an UNLOAD PC instruction, the output from external output control being provided to the EXTERNAL bus.

Considering FIGS. 3a and 3b with FIG. 17, the following discussion examines the flow of data between second ROM chips 48', arithmetic chip 10 and the main ROMs 20a–20c. When a program is to be called from second ROM chip 48' according to the aforementioned 2ND, PGM, page number and label key sequence, the location 0000 is first loaded into the program 601. This may be done for instance by (1) zeroing a selected register 50a–50e, which loads the zero in register R5 65. As disclosed in the incorporated by reference U.S. Pat. No. 3,900,722, register R5 65 is automatically loaded with the least significant digit after some arithmetic operation involving the arithmetic unit and Register R5 65 may be loaded with the least significant digit in keyboard board register 54 in response to a KRR5 instruction. Also the least significant digit of keyboard register 54 may be loaded with the contents of register R5 65 in response to a R5KR instruction. Now, the zero digit in register R5 65 is loaded into the least significant digit position of keyboard register 54. After having reset digit strobe with a FETCH instruction, the zero is loaded into each of the four digits of program counter 601 by four sequential LOAD PC instructions, thereby transferring the zero in the LSD of the keyboard register into all digit positions of program counter 601. Then a FETCH HIGH instruction is issued followed by a load keyboard with EXTERNAL (EXTKR) instruction for loading high digit of the program code outputted from ROM 600 into the keyboard register. The contents of keyboard register 54 is then loaded into register R5 65 and thence into a register 50a–50e using an R5→Adder instruction (FIG. 5b). This high digit may then be shifted in its register to the next more sigificant digit position by register shift instructions (FIG. 5b). A FETCH instruction then follows, which is followed by a EXTKR, KRR5 and R5→Adder instructions to load the low digit into the register in which the high digit had been previously loaded, thereby providing the high and low digits in one of the operational registers 50a–50e. This number may then be subtracted with the inputted page number to determine whether the inputted page number is available in ROM 600.

As can be seen, having an R5 register of a single digit length complicates the transferring of the low and high order digits of the program code into an operational register 50a–50e. It should be evident that an eight bit register R5 65 would simplify this process; the method herein described is used because register R5 in the preexisting arithmetic chip 10 only has four bit positions. Modification to the above described instruction sequence for a calculator system having an eight bit R5 register 54 should be evident to those skilled in the art.

It should be also evident to those skilled in the art that this addressing sequence could be simplified if the I/O bus were used in lieu of EXT. Then, however, additional connections would be necessary to interface the second ROM chip 48' with the rest of the calculator system, thereby complicating the plug assembly for interfacing chip 48' with the remainder of the calculator system.

Assuming that the inputted page number is available in ROM 600, the inputted page number is multiplied by two and the resulting address is loaded one digit at a time into program counter 601 via register R5 65, keyboard register 54 and the EXT bus. As has been previously discussed, the two program codes at that location and the following location in ROM 600 are read out thereof one digit at a time using a sequence of FETCH HIGH, EXTKR, KRR5, KR→Adder, SHIFT, FETCH, EXTKR, KRR5, R5→Adder, SHIFT instructions and so forth in the manner previously set forth for four digits. At this time, the four digit address of the first program code for the inputted page number has been loaded into a selected register 50a–50e. This process is then repeated for the next two program codes, which are loaded into another register, this address defining the first program code on the page immediately following the inputted page number. The difference of these two numbers is taken and stored and the address of the program code on the inputted page is then loaded one digit at a time into program counter 601 by generally reversing the above sequence and substituting the FETCH and FETCH HIGH instructions with LOAD PC instructions, thereby loading program counter 601 with the address of the first program code on the inputted page number. The program codes are then read out sequencially and compared with the inputted label, the calculator testing for a match thereof. For each FETCH operation accomplished during this label search, the aforementioned difference is decremented by one. If the decrementing difference becomes equal to zero before a match is found, the calculator generates an error condition inasmuch as the label being searched for does not exist on the particular page inputted by the operator. Once the inputted label is detected, the outputted program codes are loaded into program counter 54 which in turn is used to address main ROMs 20a–20c when a PREG instruction is generated telling main ROMs 20a–20c to branch on the address being outputted on EXT. The microcode at that address is then read out in the usual manner to accomplish the

function indicated by the outputted program code, that sequence of events taking between fifty and several thousand instruction cycles, for instance. At the end of this sequence of instructions, a flag is tested to determine whether another program code is to be read from second ROM **48**, or whether a program code is to be read from the program code storage area in RAMs **14**a–**14**c, whether a program code is to be read from the aforementioned constant ROM area in double SCOM chips **12**a or **12**b or whether the calculator is simply to await another keyboard input by the operator. This flag is a memory on one of the double SCOM chips; the control of which the above-mentioned elements control calculator operation will be subsequently discussed. Assuming control is to be directed back to second ROM **48**, FETCH and PREG instructions will again be issued to cause the calculator to branch to the location defined by the subsequent program code in second ROM **48**. This process will be repeated until a RETURN program code is loaded into keyboard register **54**, and main ROM **20**a–**20**c branches to the location defined thereby; there instructions transfer control back to normal keyboard inputs and the display is activated with the contents of register A, unless of course, the above-mentioned **2**ND, PRGM, page number, label key sequence occurred as a separate chain in a user inputted program in RAMs **14**a–**14**d, in which case, control would be then returned to the program thereat.

It has previously been indicated that after the page number has been inputted, a label search is commenced on that page by inputting an appropriate label, i.e. either the labels A-E or A′-E′ or the subroutine key (SBR) combined with a non-numeric key (indicating that such key is being used as a label) or a three digit number (which is treated as a relative address, as aformentioned). It should be appreciated that data may be entered into the memories, including the display register, after the page number is inputted and before the label or relative address is inputted. In fact, many of the programs in the embodiment of second ROM **48** programmed according to Table VIII assume that the number in the display at the time the label is inputted is to be used during the execution of the program identified by the label.

Referring now briefly to FIG. **19**, there is set out the utilization of the sixteen registers on the two double SCOM chips **12**a and **12**b. Each register can store up to sixteen digits, each of which has four bits, of course.

Registers **1**–**8** and **12**, as well as digits **1** and **2** in register **13** are reserved for heirarchy control, although registers **1**–**8** may be addressed from the second ROM using the aforementioned heirarchy address program code.

Register **0** is reserved for: (1) ten user flags, (2) the RAM/Constant ROM Program Counter and Program Source Flag, (3) last key entry and (4) a fixed point display indicator. The Program Source Flag in digit **3** and the digit **15** flag in Register B **53**b (FIG. **3**a) define where calculator control is to be passed after the present set of instruction words from first ROM **20** are executed. If flag B **15** is set, then the calculator is under control of a program either in (1) RAM's **14**a–d, (2) second ROM **48** or (3) the second ROM portion of the constant ROMs on double SCOM chips **12**a or **12**b. If flag B **15** is reset, the calculator is under normal keyboard control. The program Source Flag is a 0 if control is to be returned to RAM **14**d–d area; a 1 through 7 of control is to be passed to second ROM chip **48**′; or an 8 or 9 if control is to be passed to the constant ROM area.

Registers **14** and **15** are utilized to permit superroutining of the programs in RAMs **14**a–d or in second ROM chip **48**′. Thus after control passes to the element specified by the current program control flag, the stack implemented by registers **14** and **15** is popped and the location and new program control flag previously in level one thereof is inserted into digits **7**–**3** of register **0**. Of course, when the number of superroutine levels in the stack is one or greater, as indicated by the number in digit **0** of register **15**. The stack is pushed (i.e., another level of superroutining is added) when (1) a function stored in the constant ROM area (e.g. P→R) is encountered; (2) a label is encountered (e.g. A-E or A′E′); (3) a subroutine program code or key depression followed by either a three digit address or a non-numeral label address (which initiates either a branch or a label search in the element presently controlling operation and if the second ROM chip **48**′ is the controlling element, the label search is limited to the present page); or (4) program codes or key depressions for **2**ND, PGM, and then A-E′ or SBR and label (e.g. SBR X^2 or SBR=) or SBR and a three digit relative address (initiating either a label search on the indicated page or a branch to a specific program code on that page whose address is determined by adding the relative address to the address of the first program code on that page). As can be seen, six levels of such routines may be employed in this embodiment, with digit **0** of Register **15** indicating the number of levels actually being utilized at any given time.

Register **13** stores the addresses of the useable data storage area in RAMs **14**a–d, which may be varied or reconfigured by the operator inputting appropriate OP codes, as previously mentioned. Digit **0** of register **13** contains a flag indicating whether angular results are to be provided in degrees, radians or grads.

Register **11** is used as the T register, for the comparisons made in the "conditional go to" program codes, while register **10** stores eight program codes during operations under RAM **14**a–d or constant ROM control, at which time the eight programs stored in a constant area or register in RAM being accessed are temporarily stored in register **10** to simplify the extraction of the program code to be used to address first ROM **20**.

Register **9** contains the old and new page numbers which allows the user to change pages easily. Register **9** also contains information about the program size, size of the RAM and also the program's security flag, which may be stored in second ROM **48**, if desired, as forementioned.

Referring now to composite FIGS. **18**a–**18**i, there is shown a detailed logic diagram of second ROM chip **48**′. BCD ROM **600** is implemented as a conventional virtual ground type ROM of the type disclosed in U.S. Pat. No. 3,934,233, entitled "Read-Only-Memory For Electronic Calculator", which issued Jan. 20, 1976 and is assigned to the assignee of this invention. Decoders **620** and **621** used in addressing ROM **600** are important features of this invention which permit ROM **600** to be addressed using BCD data without wasted space within ROM **600**. The decoders heretofore known in the prior art, such as those exemplified by U.S. Pat. No. 3,934,233, decode either binary, octal, or hexadecimal data, as the case may be. These decoders may be used with a ROM to decode BCD data, of course; however, in that case, large portions of the ROM would go un-

used inasmuch as hexadecimal numbers 11 through 16 would be decodable, but have no need to be decoded. Using the addressing scheme herein disclosed, permits the addressing of ROM **600** with BCD data without the wasted space within the ROM which would otherwise result with conventional decoders.

ROM **600** is implemented as a 5000×8 bit array for storing 5000 eight bit program codes, the addresses thereof being the BCD encoded numerals **0000–4999**. These four numerals are stored in program counter **601**. Decoders **620** and **621** are able to decode these 5000 addresses without decoding the non-BCD codes often seen in the binary data contained in a register such as program counter **601**. As will be seen, the ROM metal line decoder **620** uniquely decodes the one of 125 metal lines while the ROM difusion line decoder **621** decodes one out of forty difusion lines for each bit in a program code. Inasmuch as 125×40 equals 5000, 5000 program codes in ROM **600** will be uniquely identified by metal decoder **620** and difusion decoder **621**.

Metal decoder **620** comprises a plurality of one of five decoders. These one of five decoders are provided in a three level cascaded arrangement inashuch as 5×5×5 equals 125. Referring to dashed lines **B1–B5**, it can be seen that the dashed line **B1** encompasses a one of five decoder, as do references dashed lines **B2–B5**. The inputs of one of five decoders **B1–B5** are connected to the outputs of one of five decoder **C1**, all of which are used to perform a one out of twenty-five decode. One of five decoders **C2–C5** are identical to one of five decoders **B1–B5**. The five outputs from one of five decoders **C2–C5** are each coupled to one of five decoders like **B1–B5** (not shown) for providing the decoding of metal lines **26–125**. The inputs of one of five decoders **C1–C5** are connected to the output of a third level one of five decoder identified by a reference line D. The B level decoders (e.g. **B1–B5**) are responsive to the **A1–A3** bits outputted from program counter **601**, while the C level decoders are responsive to the **A7–A5** bits and the D level decoder is responsive to the **A9–A11** bits from program counter **601**. The remaining bits, (e.g. **A0**, **A4**, **A8**, **A12**, **A13**, and **A14**) are decoded by diffusion decoder **621**, for providing a one out of forty decode. Diffusion decoder **621** is divided into eight sections **621a–621h** for addressing the eight bits of the addressed program code. As can be seen decoder section **621a** has one output line (P_0) one ground line and **39** intermediate diffusions, inasmuch as ROM **600** is of the virtual ground type. Decoder sections **621b-621h** are identical to section **621a**, but output the P_1-P_7 bits of the program code.

It should be evident that decoder **621** performs a one out of two decode for address lines **A0**, **A4** and **A8** and performs a one out of five decode for address lines **A12–A14**. It should be evident to one skilled in the art, moreover, that other configurations of decoders **620** and **621** could be utilized. For instance, metal decoder **620** could be arranged to perform a one out of 250 decode by doubling the size of that decoder and adding a one out of two decoder of line A_0, for instance, in front of the D level decoders, while diffusion decoder **621** would perform a one of twenty decode by deleting the A_0 address line 0 and collapsing the size thereof by deleting the odd numbered diffusion lines.

It should further be evident to those skilled in the art that similar BCD only addressing schemes may be used with other sizes of ROMs. For instance, a 7000 word ROM may be BCD addressed with an one out of 250 metal decoder (2×5×5×5) and an one out of twenty-eight difusion decoder (2×2×7). As can be seen; the numbers in the parenthesis are the prime number factors of 7000.

Diffusion decoders sections **621a–621h** output the P_0–P_7 bits of the program, respectively. The P_0–P_7 bits are outputted serially on external during state times S_3–S_{10} in response to a decoded FETCH instruction while bits P_4–P_7 are to be outputted serially on external during S_3–S_6 in response to decoder FETCH HIGH instruction, as aforementioned. The P_0–P_7 bits from decoder **621** are connected to bus **623** when strobed by shift register **622** beginning at **S2ϕ2** for the P_0 bit and when enabled by logic **607a** in response to a decoded FETCH instruction. Bits P_4–P_7 are conducted to bus **623** starting at state time **S2ϕ2** when strobed by shift register **622** provided logic **607b** receives a FETCH HIGH command. Bus **623** is coupled to the EXT line via external output buffer **624** when enabled by external output control **608**. Control **608**, in combination with logic **625**, couples bus **623** to buffer **624** in response to either a FETCH or FETCH HIGH command. The data on bus **623** is one-half bit early due to a one-half bit delay in buffer **624**.

Program counter **601** is provided by a 15½ bit shift register, comprising thirty-one inverters. The other half bit a delay occurs in the one bit adder/BCD corrector **604**, thereby providing sixteen bits of storage for storing four BCD numerals. The least significant digit in program counter **601** is loadable from a four bit serial numeral appearing on EXT during state times S_3–S_6 via digit control **602a** in response to a LOAD PC command. As can be seen from FIG. **18**, digit control **602a** inserts a new digit in the least significant digit position in program counter **601** by coupling the data on EXT to inverter **626** and open-circuiting transfer gate **627** connecting inverter **628** to inverter **626**. Inverters **626** and **628** are two of the thirty-one inverters in program counter **601**. Digit control **602a** also outputs to bus **629** serial data stored in program counter **601** when enabled by digit strobe **603**, the digit appearing on line **629** being one-half bit early compared to the time at which it will be outputted on EXT in response to a decoded UNLOAD PC instruction.

Digit controls **602b–602d** are identical to digit control **602a**, except that they are connected to program counter **601** at appropriate places in the shift register therein to interface with the next to the least significant digit through most significant digit positions during the S_3–S_6 state time period. Bus **629** is connected to external output control **608** which functions in combination with logic **630** for conducting the BCD digit outputted from the enabled digit control **602a–602d** through to EXT during S_3–S_6 in response to a decoded UNLOAD PC instruction. Bus **629** is one-half bit early compared to the data on EXT, because of the one-half bit delay associated with output buffer **624**.

Digit strobe **603** is comprised of a four bit ring shift register counter for sequentially enabling one of the digit controls **602a–602d**. Shift register **603** advances in response to a LOAD PC or UNLOAD PC command by logic **631**; shift register **603** for enabling the LSD digit decoder **602a** in response to a decoded FETCH instruction by a logic **632**.

The four digit BCD number stored in program counter **601** shifts through program counter **601** and from the output thereof back to the input thereof via the one bit adder/BCD corrector **604** each instruction cy-

cle. The contents of shift register **601** is coupled to thirty ROM address buffers **633** during $S_{15}.\phi 2$ through $S_0.\phi 1$. The output of the thirty address buffers **633** provide the A_0–A_{14} and $\overline{A_0}$–$\overline{A_{14}}$ outputs to the metal decoder **620** and difusion decoder **621**. Only fifteen stages of the shift register in program counter **601** are outputted to ROM **600** via buffers **633** and decoders **620** and **621**, inasmuch as the most significant bit of the most significant digit in program counter **601** need not be decoded when the largest address is 4999. It should be evident to one trained in the art, however, to utilize the outputs of all stages and to increase the size of the aforementioned decoder **620** or **621** to accomodate extra address lines for using larger addresses than 4999.

One bit adder/BCD corrector **604** is a single bit, serial adder that adds one to the stream of for BCD digits circulating through it from program counter **601** in response to a decoded FETCH instruction. During NON-FETCH (including FETCH HIGH) instructions, the four BCD digits circulate through one bit adder/BCD corrector **604** without the add one operation being performed. The one bit adder/BCD corrector **604** does a "look ahead" at the serial stream of data exiting from program counter **601** in order to determine if BCD correction is necessary when an add one operation is to be accomplished. When corrector **604** receives the least significant bit of a digit, it is also provided with the most significant bit of that digit for the "look ahead" operation. Corrector **604** is also provided with a clock signal, BCD CORRECT, for indicating to corrector **604** when the least significant bit of a digit is being received thereby. If during this time period the least significant digit is being inputted and a FETCH instruction has been decoded, one will be simply added to the least significant digit unless the first and last bits of that digit are both a logical one, (i.e., a decimal nine has been outputted), then in lieu of adding one, which would form the illegal BCD code 1010, a 0000 is outputted from corrector **604** in the corresponding four state times and an add one operation is accomplished on the next more significant digit via carry circuit **633** in corrector **604**. Should that digit contain a nine, the above operation repeats and, if not, a one is added to that digit.

In one bit adder/BCD corrector **604**, adder gates **634** perform the add operation when a one is outputted from add one insertion gate **635** unless inhibited by gate **636**. Gate **635** is responsive to a decoded FETCH instruction and state time $S15\phi 2$ for inserting a logical one into adder **634** in the least significant bit of the least significant digit or for inserting a one in the least significant bit when enabled by carry circuit **633**. Gate **636** is responsive to a decoded FETCH instruction, the A_0 and A_3 bits from program counter **601** and the output of gate **635** for generating four logical zeros when a BCD nine is to be incremented during a FETCH instruction. Gate **636** is enabled by timing signal BCD correct at $S_4\phi 1$, $S_8\phi 1$, $S_{12}\phi 1$ and $S_0\phi 1$, the BCD correct signal being delayed $1\frac{1}{2}$ state times by a logic **637**. Thus once gate **636** outputs a logical zero disabling the output from adder **634**, that condition remains for four state times until the BCD correct signal is again generated from PLA **610**. Carry circuit **633** is responsive to adder **634** for generating carries within a digit and to gate **636** for generating carries between digits.

The state time generator **609** comprises sixteen state time drivers, the first of which is responsive to a timing signal on $\overline{IDLE}$ for sequencing the state times generated thereby with the state times generated on the other chips in the calculator system. The outputs from the sixteen state time drivers are supplied to a PLA **610** for providing various timing signals used, for instance, for outputting the program code during S3–S10, receiving and outputting single digit addresses during S3–S6 and the like. PLA **610** also outputs in serial fashion four serial trains of digits mimicking the LOAD PC, UNLOAD PC, FETCH HIGH and FETCH instructions which are outputted from main ROM **20a–20c** or IRG. Of course, IRG also transmits many instructions which are not decoded on chip **48'**.

These four serial bit trains are supplied to four serial instruction decoder circuits E1–E4 in decoder **611**, along with the instruction words appearing on IRG. Each decoder in serial instruction decoder **611** performs an exclusive OR function on the data from IRG and one of the bit trains from PLA **610**. If the exclusive OR function is satisfied for all bit positions of the instruction word (indicating that there was a match between the instruction word outputted and the bit train from PLA **610**), then either a LOAD PC, UNLOAD PC, FETCH HIGH or FETCH instruction has been decoded, depending, of course, on which bit train is provided to the particular decoder for which the exclusive OR function was satisfied. Cross-coupled gates **638** perform exclusive OR function and discharge NODE **639** when a mismatch occurs anytime during the exclusive OR operation. The decoder circuitry encompassed by reference lines **E2–E4**.

It should be remembered that in the calculator system disclosed, an instruction is normally decoded during one instruction cycle and performed during the following instruction cycle. Inasmuch as the UNLOAD PC, FETCH HIGH AND FETCH operations output data on EXT and considering that the keyboard register **54** in arithmetic chip **10** must be sensitized to input this data by a EXTKR instruction, the data to appear on EXT from chip **48'** must appear one instruction cycle later than the normal instruction cycle for executing decoded instructions. That is, the FETCH, FETCH HIGH and UNLOAD PC instructions result in data being outputted on external during the second instruction cycle following decoding. ROM **600** is a relatively large ROM, so for FETCH, or FETCH HIGH instructions the precharge cycle begins during the beginning of the instruction cycle immediately following the decoding of the instruction (by the FETCH **M1** and FETCH HIGH **M1** signals) whereas the conditional discharge occurs about one instruction cycle later. Thus ROM percharge circuitry **640** is responsive to FETCH M1 and FETCH HIGH M1 whereas the logic **625**, **607a**, **607b**, add one/BCD corrector **604**, logic **632** and the like are sensitized to delayed FETCH instructions FETCH **M2** or FETCH $\overline{M10}$.

I have described my invention in connection with certain specific embodiments thereof. It is to be understood that modification may now suggest itself to those skilled in the art and that this invention is not limited to the specific embodiment disclosed, except as set forth in the appended claims.

TABLE IIa

HEX ADDRESS	CHIP	INSTRUCTION WORDS→																
0000	TMC582	0401	0404	0080	030C	01D8	0408	1002	0121	040F	0101	032C	1809	0402	0107	1FCF	0477	0085
0011	TMC582	1460	1C4C	1FDC	1FC8	1FFC	1D2C	1F44	1CF2	1FE2	0497	1820	04C7	1618	0407	1818	0477	18DE
0022	TMC582	1C84	1C45	1815	19C6	147F	1482	1486	1486	0457	0095	1424	0447	19C4	040C	04C7	184E	1882
0033	TMC582	1827	1851	19F0	145C	1440	1444	108F	0CD2	0213	0007	0088	1F34	1636	0081	18E0	1EEE	1814
0044	TMC582	1468	1436	1434	143F	1402	00F0	1482	0023	0024	0058	1520	10E8	0407	197C	1F94	1880	1900
0055	TMC582	0011	19C8	0011	1890	0497	185F	0111	040F	0121	0105	194C	0427	1843	1C0E	1F14	1C1F	1C14
0066	TMC582	1C40	1F18	180C	1FFC	0487	1897	0904	1994	04D7	1940	04E7	141C	19F4	1F0F	1C20	1C0E	1800
0077	TMC582	140E	19F0	0437	1C46	0447	1430	0022	0024	146C	04E7	141C	1832	185C	1895	1837	18C6	1FD4
0088	TMC582	10D4	1FFC	0407	14F7	0CD2	0213	0007	0088	1F78	1674	1FFF	1834	18ED	1828	18EE	190B	1F70
0099	TMC582	0041	1836	0487	0080	14E4	0487	14F0	0427	193F	10F4	18C7	1E94	1837	1C12	1FF8	1FF6	0410
00AA	TMC582	08F8	1814	0049	0051	0477	1E74	00C1	0080	0407	1284	0437	1F6C	1FCC	1474	1FF6	1FC0	1FC8
00BB	TMC582	1FF6	1484	1FF8	1F6C	1FF0	1480	1F60	1F84	1034	1D28	1034	1FAC	1899	1E08	192C	1E44	1990
00CC	TMC582	1934	1F9E	1446	1FC4	1F94	1849	1E94	18F6	1F90	1432	1F48	1E84	1FF8	18F8	1941	1FFE	1F80
00DD	TMC582	1F0F	0051	18F4	1F78	1F04	18F6	184F	1FFC	1F6F	1FCC	180C	1FD4	1F66	18FD	1F62	1FC0	1F88
00EE	TMC582	1F5C	1F8F	1E58	1468	1F70	1F52	19F4	18C0	1F64	10F6	1436	1F40	1F44	1F42	1F00	1987	1FCE
00FF	TMC582	14F7	1FC4	1F36	1F94	1FF8	1842	1C06	1008	1424	1F4E	1814	1F44	184C	1E20	1800	1F02	14F4
0110	TMC582	0487	0058	1206	0080	0052	16FC	005A	1806	0054	0049	00F9	0497	02F3	0289	1C44	0202	0F03
0121	TMC582	005A	0300	0409	1007	0202	045B	0089	1C36	00A8	1368	00F1	1864	04D7	02F3	0105	0111	041F
0132	TMC582	0104	0064	0724	0663	0619	1149	0099	1987	0467	1804	0447	0080	1804	0011	0091	0045	0065
0143	TMC582	0418	0015	0100	0003	0300	0101	041F	0603	0060	065A	0709	1894	0099	070A	0121	041F	0100
0154	TMC582	0148	02F3	0108	0121	040F	0101	0153	095B	187F	0427	184C	0018	0107	0102	1872	0400	0400
0165	TMC582	0404	0404	0404	0404	0404	0404	0404	000A	0008	145A	0308	0095	140C	0040	04C7	1412	005A
0176	TMC582	0052	0080	1210	0437	01F3	0153	0437	10C4	0447	0095	0085	187F	02D8	04E7	02F6	0131	041F
0187	TMC582	0C03	0131	040F	0111	0477	01F3	0080	1502	0467	06F3	0153	1062	0467	09F3	0060	1E72	0300
0198	TMC582	0101	041F	0603	065A	0708	105A	0404	0121	041F	0100	015B	0289	0153	1048	0C48	09F0	0980
01A9	TMC582	1046	065A	0708	0140	0902	0107	0121	041F	0100	0148	0148	0121	040F	0101	015A	0911	02F0
01BA	TMC582	02F3	040D	040C	0095	0045	0280	0418	0156	0111	041F	0103	000F	0014	0120	1412	0000	0980
01CB	TMC582	185A	065A	0140	0487	02F3	0289	10FE	0064	0300	0406	02F4	0101	040F	0121	0404	0121	041F
01DC	TMC582	0100	0140	0911	02F0	0440	0611	02F0	0121	040F	0101	0323	0111	041F	0100	0700	0600	0680
01ED	TMC582	1834	010E	0080	1522	0008	0008	1582	0107	0407	186C	015A	0308	1006	0140	0403	0407	02F3
01FE	TMC582	0111	041F	0100	0610	0919	1008	0648	0709	1807	0111	040F	0101	108F	0427	0080	1804	0457
020F	TMC582	0085	00E5	00C5	199F	0009	100C	180C	0020	1809	0009	1004	0091	07D8	0107	0201	0007	0400
0220	TMC582	1826	0090	0092	1804	0085	0107	0001	0427	000F	0008	1804	0437	00C9	01F0	06F0	0140	0107
0231	TMC582	0072	004A	17A8	00AF	0051	0058	1804	0054	0026	0036	0120	0009	1104	0709	1808	0032	0308
0242	TMC582	0009	1818	0309	1804	0300	0E09	1814	0940	0409	1008	0F09	18F4	0940	0030	1010	1806	0030
0253	TMC582	1804	0708	182F	0608	0033	0700	1837	00F0	1006	0099	0030	1808	0FD0	0708	10C2	0FD8	108F
0264	TMC582	0041	1881	0091	0093	0C93	006F	004A	1750	0049	005A	10F0	1449	0467	19F5	0081	0417	004A
0275	TMC582	1864	0085	00F5	00D5	1801	04F7	1C41	0447	0049	0051	180D	03D3	0447	06F3	1411	0082	1C14
0286	TMC582	04F7	0058	1017	0051	1840	0033	1840	0417	18F8	0004	0049	041F	01D6	090E	0089	040F	0131
0297	TMC582	04F7	02F3	0111	040F	0407	02F3	0111	041F	0106	090F	0111	040F	0131	0068	0044	10E3	0040
02A8	TMC582	1804	0448	0082	1843	0407	0095	103D	020F	18E8	0080	0079	0091	18FF	00F4	0074	1985	0030
02B9	TMC582	1138	0684	128C	0009	1950	1188	0104	0022	0032	005A	0000	1008	0104	002E	003F	0084	00F2
02CA	TMC582	040C	0085	0400	0050	0040	14F4	005A	0015	040D	0427	1880	0120	0416	02F0	0103	0091	1880
02DB	TMC582	0640	0658	0680	0008	1784	00F8	0078	0030	0042	0108	0046	0014	1008	0709	0437	1014	0FD6
02EC	TMC582	0030	0477	1804	0407	06F3	0618	1002	0457	02F3	0283	0020	0437	1804	0467	09F3	0206	0427
02FD	TMC582	02F0	0C76	0953	0300	0409	1806	0289	100D	0230	0048	0044	1008	0082	0989	15FF	0953	0976
030E	TMC582	1809	0100	0100	0100	0100	0100	0100	0100	0100	0100	0049	00F2	0085	00C8	1758	0048	1582
031F	TMC582	005A	1808	010A	0402	0054	0485	103C	0408	0FF3	0653	00F0	109F	095A	095A	095A	0447	02F3
0330	TMC582	0030	1006	0709	1004	095A	095A	0487	02F3	0980	1814	0289	1810	0048	1804	0300	0940	0908
0341	TMC582	0C53	0980	000A	1532	18C7	0011	00C3	0092	00C0	00F0	0093	1804	0011	0080	1808	0F03	0024
0352	TMC582	0034	0090	1008	0CD2	0024	0034	0010	1F04	0023	0012	003C	1948	068C	1015	0030	1808	0CD2
0363	TMC582	0024	0034	072C	1006	095A	1807	010C	002C	118C	0989	1017	0988	1954	0022	0080	1F50	0021
0374	TMC582	1F4C	010A	0484	0497	02F3	09F0	0300	04F8	0101	0101	0208	04F8	0101	0C03	0427	02F0	0CF6
0385	TMC582	04F6	0101	0467	01F4	0306	00F4	0064	0140	0140	0140	0140	0C19	0923	0130	1806	0186	0990
0396	TMC582	0111	040F	0131	0407	02F0	002C	0101	040F	0121	0040	1804	0448	04D7	01F0	0101	041F	0103
03A7	TMC582	0604	0101	040F	0111	010C	0074	0004	0004	0080	0404	0074	0004	0004	0084	040C	0400	040C
03B8	TMC582	0069	0058	0094	1189	0059	0084	1949	00F1	00C3	00C0	0092	0011	0133	000F	1808	0CD2	0024
03C9	TMC582	0034	0090	13E4	0010	1882	0011	0009	1874	0980	0026	1084	0099	0022	0E00	1902	004B	004A
03DA	TMC582	0042	0094	0004	0002	0042	1628	0089	0168	0007	0042	0004	0008	005A	1365	1470	0080	1744
03EB	TMC582	0447	0095	1880	0073	1808	0447	1804	1F06	000A	000A	1817	0001	1818	034A	1608	0009	0497
03FC	TMC582	02F3	0111	041F	0103													
0400	TMC582	0219	1839	0659	0619	103F	000A	185D	006E	0042	1F80	1FC0	0457	1841	0035	0114	0903	002C
0411	TMC582	0022	1004	0021	003C	181E	0680	1820	0041	00C8	13BD	040D	0404	040D	0030	1003	0042	188F
0422	TMC582	0033	06D2	0649	1007	0688	090A	06DC	0D29	1016	0010	183A	06D8	1804	0008	0C9C	1805	0C94
0433	TMC582	1F12	0F09	0012	1838	0C59	0940	0009	1815	0300	1819	096C	0029	1840	0C02	0F04	1808	0953
0444	TMC582	0300	1804	0948	052C	1017	0980	1807	0948	0958	0E00	0308	1811	0085	0051	010A	0467	041A
0455	TMC582	0022	0CC0	0054	1F74	0480	0408	13CC	0401	0081	040C	0487	006E	1F46	0080	10CF	0407	02F3
0466	TMC582	0F87	06F3	0153	0427	09F3	0153	0107	0150	0107	04D7	1909	0080	1046	0427	01F3	0477	06F3
0477	TMC582	0153	0503	1810	0107	020F	0500	1006	02F0	0306	0107	0208	041F	0FD6	060F	0133	1812	0410
0488	TMC582	08F8	179C	18C1	00F4	00F2	005A	1497	0051	1C81	0081	0417	0095	1009	0437	02F3	0417	1862
0499	TMC582	1F86	0051	0080	0133	000F	110B	1486	1FFF	0404	1804	0306	04D7	02F3	0111	041F	0CD6	1C11
04AA	TMC582	1F68	1C04	004C	1023	0448	0042	1020	0092	1F48	1FD4	0447	1C7F	0051	0406	02F0	0103	0107
04BB	TMC582	012C	0121	041F	0100	0007	0400	0050	1847	0121	040F	0111	1015	0209	0457	06F3	18C5	0C41
04CC	TMC582	0107	0417	1845	0046	0099	0404	1887	0051	1464	0406	02F0	0107	0108	000F	0107	0201	0007
04DD	TMC582	0403	0054	1CF4	0F03	1C7C	14F1	1049	1FF0	1FF0	040C	00C5	0619	1804	0940	0C40	069A	0404
04EE	TMC582	1005	0213	0410	0409	0404	0405	1805	0401	0619	1806	0C48	094B	0050	159E	00F8	1766	1028
04FF	TMC582	1FC4	19C9	1F84	1879	1E36	1488	1FF0	1990	19D7	0051	1C08	19DD	1CC4	1C22	1060	18F2	1F61
0510	TMC582	19F0	1F18	19E0	0051	1F2C	19F3	0051	190A	19F9	190A	1809	19FF	1C1C	1F08	1405	1FF2	1900
0521	TMC582	18F9	1F72	140F	1956	10D0	1949	1949	14CF	1F04	10C6	1983	1983	1403	1F98	1808	18C1	1004
0532	TMC582	0051	14F1	1431	1043	1FC4	1911	1FF4	1FF2	143D	18E1	1F0F	1443	1FEC	0427	0045	1F63	1927
0543	TMC582	1F94	1953	1FF8	0020	1F01	0FDC	17D5	0303	1F91	0040	004A	100C	041F	0903	06F3	0619	169C
0554	TMC582	0944	0914	0467	02F0	0103	0487	02F3	00F2	095A	095A	041F	0206	0050	1004	020F	00F8	007A
0565	TMC582	143C	0110	007A	1829	0437	06F6	0498	1803	0640	1002	0709	04D2	1016	0030	180C	0648	0402
0576	TMC582	0480	0648	1804	0648	0158	0248	0239	1824	0289	1026	0904	0308	1809	0030	1808	0900	0300
0587	TMC582	0709	1F94	0032	0232	1018	005A	1806	0509	1F96	0809	1702	005A	184C	094A	0308	0230	1004
0598	TMC582	0600	1806	0309	140F	0048	0044	1806	0050	1C9F	041F	0906	0905	0F2C	0F2C	0905	1806	0040
05A9	TMC582	0400	08F8	1808	0000	00C8	1E2F	0406	0000	0702	1804	0001	0007	0012	0042	040D	1E10	0800
05BA	TMC582	0048	0940	0400	0050	1F05	0485	15AF	0445	170C	0495	1F40	0475	1808	0405	0455	1834	0475
05CB	TMC582	1826	0455	0405	1824	132F	0051	0207	0407	02F0	0101	041F	0200	0309	1708	0009	1710	0207
05DC	TMC582	1883	041F	00F3	04F3	0410	118F	1E01	0465	0455	0405	179C	0082	0002	1FF1	1865	0455	1FC4
05ED	TMC582	0495	1F92	0080	0002	0082	1F98	0427	02F6	070F	17E4	177C	1E97	1F81	0100	0100	0100	0100
05FE	TMC582	0100	0100	0100	0100	0100	0C40	1F78	0011	1F08	1F41	13F1	1FC5	0619	1818	095A	095A	095A

TABLE IIa

HEX ADDRESS	CHIP	INSTRUCTION WORDS*																
060F	TMC5A2	0C53	0C53	0718	0C53	0618	1A06	0940	0C40	0309	1A04	005A	004A	1004	0208	0050	0052	1267
0620	TMC5A2	00EA	1257	0008	133C	00C6	0006	0213	0698	060A	1005	0610	0409	06F0	0620	1A0C	0605	0455
0631	TMC5A2	0465	0475	1019	04F0	0820	1A0C	0445	0455	0465	0475	1029	00C4	004F	04F0	0408	1ACF	009A
0642	TMC5A2	1A24	0401	0C05	0F24	0021	0418	0C05	0445	1018	0C05	0F24	0F24	0F24	0C05	0408	1A08	1008
0653	TMC5A2	0213	0410	0409	04F0	0820	1A19	04F0	0820	1A3F	0445	0455	0465	0475	1A49	0401	0619	1A06
0664	TMC5A2	0C48	0948	0475	1A16	0465	1A12	0045	004D	0085	0091	0495	0095	1A09	0045	0445	1A0C	0090
0675	TMC5A2	0485	0080	1A08	0085	0090	181E	0092	0485	0085	1A0C	0080	0495	0090	100C	0095	0445	00A5
0686	TMC5A2	1A06	004D	0085	004A	1A2A	0410	08F8	1A24	0497	02F6	0131	041F	0AD6	067E	041F	0906	02F6
0697	TMC5A2	0230	0630	1012	0400	08F8	1798	0905	0E2C	0E2C	0905	1FAE	0066	0069	0050	0052	1717	00C8
06A8	TMC5A2	1066	040D	0E1C	0085	0400	0475	0465	0455	1F8D	0015	040C	0F01	1EF6	00C0	0051	10C3	17F3
06B9	TMC5A2	041F	0903	04F3	0619	1F44	0050	17BF	00A8	0044	0042	130A	0082	1A02	040A	09F6	0002	0107
06CA	TMC5A2	0070	1A04	0309	1A06	0487	0CF0	0940	0980	0148	030A	0140	0300	114F	0107	004A	1173	1A45
06DB	TMC5A2	0407	02F3	0111	041F	0104	0105	0E2C	0105	1249	0485	0445	0495	0455	100E	0475	1F0F	0465
06EC	TMC5A2	1006	0445	1048	0111	040F	0131	18F6	005A	1004	0144	0406	02F4	0022	0095	00E5	0005	0400
06FD	TMC5A2	0032	1A6A	0010	1A06	0930	1AE7	0009	165F	00AD	010A	0CC3	0920	0030	15C8	090C	01AC	0729
070E	TMC5A2	1044	0C49	1559	0C48	0720	1A09	155A	010A	00A6	0009	177D	0030	1FCA	010A	1FA5	040D	04CC
071F	TMC5A2	0106	040A	02F6	0110	0120	0919	0939	1CAD	0009	13C5	0036	0D09	0009	15C0	002A	1A04	0099
0730	TMC5A2	0148	0022	0203	0104	0A0C	00AD	0404	0C0A	0048	1004	0C20	0ED4	0048	1012	0C23	0408	02F4
0741	TMC5A2	032C	0038	1144	0CC4	1940	0CC0	1A13	0404	0475	1A19	0048	1F10	0C23	0C18	0C5A	19A4	0C83
0752	TMC5A2	040C	0144	0214	0164	010A	01C0	0030	1742	0CA8	1F42	1D42	1D42	1F63	0465	1A0C	0455	1A0A
0763	TMC5A2	000A	000A	1969	0207	0309	0207	1A34	0000	0070	0002	00C4	1614	19C9	0051	0ED0	0034	0024
0774	TMC5A2	0013	0123	1F55	0445	0485	0475	0455	0445	1020	0495	1A31	0207	0309	0207	1143	041F	010A
0785	TMC5A2	017F	017F	00F9	0408	02F6	0176	0407	041A	04D5	1A0A	0495	1022	0075	0445	1A04	0045	0495
0796	TMC5A2	1A04	0055	0485	1A04	0045	040A	02F6	0239	1006	017F	0C76	00E9	1006	004F	1474	0176	0040
07A7	TMC5A2	140F	041F	0206	040F	0131	01D7	00C4	004A	1A02	0040	1754	0AC4	014E	0107	0070	16A8	040D
07B8	TMC5A2	0400	0015	040D	17C7	14C6	0130	0007	0400	1A97	0619	1A06	0C48	0948	041F	0906	06F6	0639
07C9	TMC5A2	11C4	00D4	0080	00C8	0000	11C6	0470	08FD	1A1A	0497	02F6	0131	041F	0606	067E	0639	1006
07DA	TMC5A2	0053	0050	15ED	0440	08FD	1016	0098	1940	0107	041F	0900	0140	0E0A	0E09	0107	1190	0008
07EB	TMC5A2	1008	0440	08FD	1009	0497	02F6	0131	041F	0103	0C5E	0C7E	000A	000A	1F76	0930	1744	000A
07FC	TMC5A2	1906	000A	000A	0072													
0800	TMC5A2	0082	17F5	18F3	0497	02F6	0107	0231	09F0	0107	0082	1097	1E7D	1085	0126	0406	02F6	0024
0811	TMC5A2	0034	0400	1FF5	1FF8	032C	198F	0C91	1FCD	1143	0082	00D1	1FF9	0051	0020	1A90	0099	010A
0822	TMC5A2	0F09	170A	0ED0	072C	150C	072C	1500	0700	1A0A	043F	040C	040A	04F6	040F	040C	040A	02F6
0833	TMC5A2	0102	0490	08FD	1447	0008	000A	1A0C	00CA	1262	0000	0002	1689	040C	0239	0231	041A	017F
0844	TMC5A2	1F86	0239	1F17	0445	1A04	0085	0455	1A04	0095	0465	1A04	0045	0475	1A04	0095	0231	041A
0855	TMC5A2	1A99	1070	1742	040C	0447	1CC5	1F42	008A	177A	000D	0107	041F	09D3	0002	1A0F	0CF6	0105
0866	TMC5A2	0724	0614	1FD2	00C3	0253	0953	0953	0949	1D3A	0487	09F3	0988	0024	1905	0983	0D2C	1002
0877	TMC5A2	0964	0944	04D7	02F3	0111	041F	0103	0008	041F	0100	1A06	0509	1A0C	095A	095A	095A	0900
0888	TMC5A2	1004	014A	095A	095A	095A	095A	095A	0953	0953	0953	0953	0953	0993	000A	1A04	060A	0C53
0899	TMC5A2	0910	040F	0101	0105	0089	0107	10CF	1F6E	0F21	1A04	0099	0030	000E	0402	0026	0006	0009
08AA	TMC5A2	040D	0059	1A08	0031	00EA	001A	1A86	0102	041F	0100	0948	041A	06F6	0405	111A	008A	15A6
08BB	TMC5A2	0467	09F6	05A9	0500	159F	0049	0000	0000	1D96	1A24	1F6C	095A	070A	1A05	0A0A	0407	041A
08CC	TMC5A2	0C06	040A	02F4	00A0	171C	0CFC	1F1A	0CE4	1420	0465	0404	1796	0080	1A1A	0C33	1F74	1FDF
08DD	TMC5A2	1FDF	0040	1A00	04C0	0457	041A	1F47	0107	0948	02F6	0447	09F6	0089	1FF4	0D0A	0009	0501
08EE	TMC5A2	1F9C	1790	0100	0100	0100	0100	0100	0100	0100	0100	0100	0100	0080	1A04	0051	0030	0009
08FF	TMC5A2	1F1C	0477	04F3	0689	1FFC	0090	11F1	060A	1F0A	1431	0051	040C	0108	01C9	0032	0022	1A37
0910	TMC5A2	0106	040A	02F6	0009	0012	0030	11E7	0940	072C	1A05	143F	0497	02F6	0131	041F	0103	0C5F
0921	TMC5A2	0C7F	000A	00C8	1D20	020A	0930	1436	0303	1442	1FF4	1FF0	0080	0052	1FF4	014F	041F	0103
0932	TMC5A2	017F	1FEA	0107	0101	140F	0107	0467	09F0	0107	00F5	00C5	0045	0457	041A	0440	0053	0050
0943	TMC5A2	1FCA	0040	1F70	164A	0030	1100	0709	1A04	0E0A	0009	0F00	11C6	0099	0051	147F	070A	0039
0954	TMC5A2	1A06	0948	1F4F	030A	0309	1F45	0036	1FFB	0050	1A04	0011	0477	041A	1A4D	0059	1A04	0011
0965	TMC5A2	0457	041A	010A	01C0	0031	0700	1A4A	0107	0176	04D7	02F6	0131	041F	0103	0102	0E98	1572
0976	TMC5A2	0131	040F	0101	090F	017F	00C4	0700	064A	1002	0709	1600	0D00	0477	06F0	030D	064E	02AA
0987	TMC5A2	1008	0709	0309	15FC	000A	0497	1A15	0107	1441	0485	1457	0495	1F93	0475	0445	1441	0465
0998	TMC5A2	1DFF	0455	0495	0048	1040	1009	00CA	17C1	0700	040C	0005	00F5	0095	00C8	130A	0430	0485
09A9	TMC5A2	104F	0445	144F	0495	1366	0475	1004	0465	1780	0455	1605	0082	010F	1C5D	1A0A	032C	1070
09BA	TMC5A2	0C64	0F09	1773	0C5A	0029	1561	0020	0032	1F61	0033	0023	1FB7	0485	15F4	0475	17CD	0465
09CB	TMC5A2	1F04	0455	1514	0082	1E3D	0445	0495	0475	103F	0465	1403	0485	1636	0455	104E	0445	1DFF
09DC	TMC5A2	01E0	1FED	1055	0C48	032C	1A05	0080	1221	0C80	1433	0C48	0C7E	0F00	0C5A	143D	175E	1FF7
09ED	TMC5A2	0C48	072C	1A05	040C	0CD2	0E09	1A06	0C53	0C40	0020	1A04	0CA8	0031	0724	17F7	0044	0620
09FE	TMC5A2	0C14	1D9F															

TABLE IIb

HEX ADDRESS	CHIP	INSTRUCTION WORDS*																
				1BF7	1BF0	1BF8	1BFD	1BFF	1C01	100F	1C05	1B0D	1B86	03D3	1FAC	0090	0051	1A95
0A0F	TMC5A3	170F	1FF6	1C08	1F7C	1A44	1DC2	1A17	00E9	1C29	0081	1FDE	0C48	0C48	0509	0140	1006	0427
0A20	TMC5A3	02F0	0C08	0D0A	0C48	0106	001A	11F6	0090	0092	1722	0070	1762	0107	0051	1002	00C0	00E1
0A31	TMC5A3	0106	0012	040A	02F6	1C81	0ED6	1085	0107	04F7	09F0	0940	000A	04A7	02F3	0082	0407	01F0
0A42	TMC5A3	0101	041F	0600	0C40	0400	1D34	0110	0107	0102	041F	010A	0C7A	0940	0201	09F6	0906	040F
0A53	TMC5A3	0131	094A	064F	0439	1640	0309	00CA	1004	000A	0009	00D9	1A10	0467	09F6	05A9	0500	1006
0A64	TMC5A3	0000	0000	0140	1002	0040	0041	1791	0043	1DC9	0054	0107	0091	1457	0445	1F07	0445	170A
0A75	TMC5A3	1F99	0082	0465	0455	11A9	1F93	0070	1354	0485	0475	1016	0445	1534	0495	100F	0465	1536
0A86	TMC5A3	0455	1A06	0445	17A1	0044	114F	0040	0042	1A04	044A	0001	0107	040A	0CF0	0407	041A	04A5
0A97	TMC5A3	1A0A	0495	1A1A	0075	0465	1A04	0065	0495	1A04	0055	0485	1A04	0045	040A	02F0	0140	1DE7
0AA8	TMC5A3	0030	1A04	0700	0948	0022	0148	0202	0148	0204	0C40	0C40	060C	1A24	0C5A	0C48	1A1C	0C9C
0AB9	TMC5A3	0C9C	0CA3	0C99	1A09	0102	0C8A	0102	0C94	1A02	0F29	1A1D	0C64	0C48	0F09	1A1D	0C10	0E09
0ACA	TMC5A3	1A06	0C40	1A07	0164	0214	0164	0620	1D7F	1CA7	0CA3	0F2C	1A05	0C64	0C48	0F09	1A0A	0140
0ADB	TMC5A3	0210	01A0	0910	1049	0400	1A41	153F	0C8A	1245	0E2C	1A07	0C60	0C7E	0F29	1A0A	0E09	1B06
0AEC	TMC5A3	0C40	0308	060A	0F09	0C14	1010	0903	090A	0F20	1F2F	0C60	0700	1A09	0032	0940	0700	181A
0AFD	TMC5A3	0F03	0C02	1FC7	0107	0001	0427	01F0	06F0	0140	0107	0072	00C9	14D3	0099	0004	1CF1	0405
0B0E	TMC5A3	12F0	14DC	1C96	1F92	1A47	1BF1	1DFA	1F52	0042	1F16	1772	0153	0153	0153	0153	01A3	1034
0B1F	TMC5A3	0100	1ED6	1093	1F06	040C	1F02	1F00	1FFF	1FFC	1FF6	1FFA	072C	0021	0CD2	072C	1010	0CA0
0B30	TMC5A3	1407	0C48	0C54	0F00	0304	1A11	0E09	1A06	0C40	0500	060A	1F51	00C9	0001	0107	0445	0477
0B41	TMC5A3	1014	0445	0455	0447	1A04	0485	1A0F	0407	1A04	0495	0487	1004	0427	02F0	0485	1A04	0300
0B52	TMC5A3	04E7	00CA	02F6	0131	041F	0CD0	0131	040F	0101	1424	1F43	1F57	174F	1749	0485	12F4	0475
0B63	TMC5A3	11DA	0465	12F4	0445	1367	0083	1A69	0901	09F0	0140	0140	04D7	02F6	0131	041F	0906	0974
0B74	TMC5A3	027F	097F	097F	097F	097F	09AF	1F24	0475	1A04	0903	014A	014A	0509	010E	1A82	0C40	0427

TABLE IIb

HEX ADDRESS	CHIP	INSTRUCTION WORDS→

TABLE IIb

HEX ADDRESS	CHIP	INSTRUCTION WORDS→																
1176	TMC583	0AF7	02F4	0AF3	0121	0A1F	0100	0308	1030	0A1F	0106	0103	042C	0320	0C40	0C40	097F	0308
1187	TMC583	1807	0C48	0C48	0976	032F	1809	0976	0986	0AF3	0AF0	0A0F	0131	0708	1EC6	090F	0A97	02F6
1198	TMC583	0131	0A1F	0100	0606	0C48	0C48	0930	0C40	0C40	0630	0131	0A0F	0101	0611	02F0	0658	0619
11A9	TMC583	1755	1F88	1D40	0631	0A18	0239	017E	0A75	0455	0A05	1176	0A65	1972	00CA	1974	0091	00C4
11BA	TMC583	1F43	1FF7	0120	0A06	02F0	0103	0107	0509	0501	1088	0948	02F6	0800	1008	0089	0D0A	0D08
11CB	TMC583	0C08	1018	0D08	10F9	0D0F	1005	0D0A	10FF	0A0D	0AF7	0A18	0A0D	1909	0059	0A16	02F4	0131
11DC	TMC583	0A0F	0121	0051	14C3	1F44	0105	0A1F	0103	0958	0953	0A0F	0111	0099	0AF7	02F3	0111	0A0F
11ED	TMC583	0008	0008	1768	1F72	0100	0100	0100	0100	0100	0100	0100	0100	0100	0A45	0910	1046	0207
11FE	TMC583	0A47	02F0	0101	0A1F	0103	0101	0A0F	0121	0107	0C5A	0C40	0308	100C	0153	0153	0C40	0C40
120F	TMC583	1800	0105	0080	172F	0920	019C	0153	0153	0194	0105	040D	0C48	0308	1805	09D7	0146	0107
1220	TMC583	0158	0509	01D8	153F	0140	0A27	09F3	0AF3	0988	0148	0508	0588	0A53	0C53	1555	0080	0105
1231	TMC583	1FF6	0140	040D	0C40	0308	1805	01AC	01AC	0966	01D2	0D08	0158	1830	0100	1E85	0082	017F
1242	TMC583	10AF	0002	0040	1004	0C02	0219	0173	1404	0103	0688	0619	0C02	15FF	0219	0105	1347	0102
1253	TMC583	0A87	02F0	0219	1006	0203	0030	0904	0288	09F3	1F09	0400	0A87	0AF0	0640	0C40	0C07	1C89
1264	TMC583	0040	0082	1491	0105	0A47	02F0	0101	0A1F	0104	1465	00CA	00C4	1094	0239	1006	0040	0091
1275	TMC583	0052	108F	1F8D	1454	040D	0308	1012	0708	1807	0948	0308	1805	0080	16C4	1F9F	0208	180F
1286	TMC583	0A97	02F0	0000	105C	0101	0A1F	090D	09AC	096C	0923	0900	0437	02F4	09AC	0A18	0A1E	032C
1297	TMC583	1809	0300	0708	010C	0153	043E	0A0C	0A08	02F0	0140	0A0F	0A0C	0A0A	02F4	0160	1017	0308
12A8	TMC583	1006	0123	1823	0980	0163	0153	0989	190D	0A37	02F0	0948	0A18	0A1F	0308	1809	0105	0107
12B9	TMC583	1FF8	1F67	1658	017E	0082	0040	1891	0000	1347	0A0C	070F	0C1A	0631	0A18	0A00	067E	0A6A
12CA	TMC583	1008	061A	0A98	0A98	0080	13F5	0700	0300	0A0C	0631	0A1A	0A04	0C7F	0A6A	0308	180F	0708
12DB	TMC583	1074	0A98	0A98	0A27	02F0	181D	0107	0A37	02F0	0158	0A2E	0A0C	0A0A	09F3	0308	180D	0A0F
12EC	TMC583	0A0C	0A37	02F0	0A0C	0A1F	0A1E	0A1F	0A1E	0A3F	0A0C	0A0A	09F6	0A0E	0A0C	0A0A	0AF6	0A76
12FD	TMC583	0174	0211	0A1A	0159	0A1F	0308	1A0A	094E	1006	0939	1224	0008	000A	1184	0A97	02F0	0101
130E	TMC583	0A1F	01D6	0906	0101	0A0F	0131	1889	0A8A	0080	0116	19F0	13F7	0120	0107	0050	1866	0A1F
131F	TMC583	0100	0A07	02F4	0AF4	0948	032C	1805	0940	072C	1805	0A0F	0101	0A47	02F0	0101	0A1F	0104
1330	TMC583	0049	1AED	1AF7	1F4E	0104	0328	0101	0A0F	0111	0008	000A	1693	0AF7	01F0	0101	0A0F	0111
1341	TMC583	0089	0A1F	0203	0A0F	0111	0A97	01F4	0121	0A1F	01F0	0640	0648	0121	0A0F	0101	0308	09F0
1352	TMC583	14C1	0A47	01F4	0C4C	0529	01DE	1AD7	0C48	0A27	09F6	06F6	094C	0C6C	052C	054F	0A7A	0C7A
1363	TMC583	16FD	0101	0A1F	0900	0964	0948	0903	0980	0A37	02F0	06F0	0948	0A18	0A1E	0308	1809	050D
1374	TMC583	0300	0033	010F	0A3F	0A0C	0A0A	02F6	0176	0A0F	0A0C	0A0A	02F6	0176	1015	0176	0176	0508
1385	TMC583	1823	0019	0948	0A18	0A1F	070A	1809	0C02	0990	1959	0003	0A3E	0A0C	0A0A	06F6	0A0F	0A0C
1396	TMC583	0A0A	02F6	1083	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04
13A7	TMC583	0A04	000D	1824	1802	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04
13B8	TMC583	0A04	0A04	0085	0090	0A68	1920	0A04	0A04	0A04	0A04	0A04	0404	0404	0A04	0A04	0A04	0A04
13C9	TMC583	0A04	0A04	0A04	0080	0095	1825	00C9	0320	0105	0101	0A1F	0104	0F24	0105	190D	0A57	0A18
13DA	TMC583	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0085	1831	0070	0072	1603	0040	0042
13EB	TMC583	1508	0A1F	0100	0148	0148	0948	0958	0958	0953	0953	0951	0988	0103	0980	0983	0980	0980
13FC	TMC583	0908	0153	0176	0986													

TABLE IIc

HEX ADDRESS	CHIP	INSTRUCTION WORDS→																
1400	TMC571	0006	103C	0A1F	0400	0C0F	0C04	1838	0A00	0AF8	1032	0A57	01F3	0A47	09F3	0976	0170	0140
1411	TMC571	0153	0153	0153	0153	0153	0400	00C8	1775	0A00	0015	0A0C	0099	0004	1AC0	0A40	0AF0	1045
1422	TMC571	0410	0AF8	1835	0498	0A08	1005	0A8A	0A0C	0133	0040	100C	0A78	0040	16F3	0068	12C3	0105
1433	TMC571	01AC	0A20	0A18	0A75	1804	00C5	0A45	1804	0085	0455	1800	0045	0A05	060C	1808	0A27	06F4
1444	TMC571	0095	0005	0A37	09F4	0223	0294	1804	0724	0294	1804	0724	002C	1026	0A24	0A18	0455	0090
1455	TMC571	1804	0095	0A45	0080	1804	0045	0224	0A18	0324	1804	0724	0A07	0015	0A18	0114	0105	0000
1466	TMC571	0040	1335	0080	1819	0A0C	0085	0065	0A6A	1401	0A47	02F0	0101	0A0F	0131	0A0A	02F0	0631
1477	TMC571	0CF6	0308	1304	0176	0176	1809	0417	0A18	0A1F	0A0C	0A1F	0A1F	0A1F	0090	0A0F	0A0C	0A08
1488	TMC571	02F3	0219	1A03	0A97	01F0	0C07	0101	0A1F	0000	0C48	0C48	090A	0C03	0C80	0A37	02F0	0C48
1499	TMC571	0A19	0A1F	0308	1809	0700	0300	0133	01DE	0153	0A3F	0A0C	040A	02F6	0176	0A0F	0A1C	0A0A
14AA	TMC571	02F6	0176	0308	1819	0708	1823	0176	0170	0948	01D2	0323	0111	0A1F	0106	0A37	02F0	0948
14BB	TMC571	0A18	0A1F	017F	0308	1804	0930	0980	0946	0980	0980	0989	1648	01D2	0094	0098	16C3	0A47
14CC	TMC571	02F0	0101	0A0F	0121	0A77	02F0	0AF0	01DC	0164	0A3E	0A0C	0A08	02F4	0164	0A0F	0A0C	0A08
14DD	TMC571	02F4	0008	1184	0308	1814	0176	0C6C	02F6	0176	016C	02F6	0708	180F	0107	0004	0148	0148
14EE	TMC571	1098	0100	0100	0100	0100	0100	0100	0100	0100	0100	0100	0708	1004	0009	0A27	0A18	0A1F
14FF	TMC571	0A0C	0A1F	0A1E	0A1F	0948	050A	0A0F	0131	0105	010C	0108	0308	043F	0A0C	0A08	02F3	0153
1510	TMC571	0A0F	0A0C	0A0A	02F3	0153	1015	0153	0153	0969	1006	0024	1825	0A37	02F0	0948	0A18	0A1F
1521	TMC571	0308	1809	0105	0121	0A1F	0103	02F0	0308	0703	1F29	0A00	0A0C	0303	0A97	02F6	0131	0A1F
1532	TMC571	01D6	067F	0059	19F4	0A00	0090	15F5	0A10	0AF4	1DF8	00A0	1A2C	0A1F	0996	02F6	0239	0459
1543	TMC571	1012	0105	0F2C	0F2C	0105	194E	0A00	04F4	1148	0A94	0A08	1005	0A8A	0A0D	0A78	1948	0659
1554	TMC571	1411	1A2F	00F8	00F4	0A19	0123	0111	0A1F	01D4	1062	0C48	0009	1004	0C00	0C48	0020	0C48
1565	TMC571	0C48	0C48	0A88	150F	02F6	0A06	0A1F	0900	0C48	0C48	0C48	0C48	0C48	0689	1000	0089	0C53
1576	TMC571	0A07	0010	1804	0A47	02F0	0C53	0296	0053	0056	0C76	0C76	0974	0020	0111	0A0F	0121	0A97
1587	TMC571	02F3	0111	0A1F	0C06	0111	0A0F	0131	0020	016C	016C	016C	096C	016F	0176	017F	0A6C	0024
1598	TMC571	0964	0966	0944	0944	002C	1800	090C	0729	0C64	1004	0F2C	096C	0C64	0C60	0A27	02F0	0022
15A9	TMC571	0054	0032	1FE1	0107	0920	090A	0158	0164	0164	0190	1F37	0051	0308	1119	0164	0164	1808
15BA	TMC571	0107	0A77	09F6	0A1F	0903	0901	00F0	0404	0940	0105	0004	1008	0808	0024	0704	0956	0969
15CB	TMC571	0008	1807	0902	0729	0948	0A18	0A1F	0100	0105	100F	0455	180F	0088	1F91	0948	180C	0A45
15DC	TMC571	1006	0088	1A40	0940	0106	1C9F	0099	1936	0A94	0A08	1C05	0A4A	0906	0A0C	0931	0A1A	0A04
15ED	TMC571	0976	0A68	0939	100F	0070	0062	180C	0A97	0045	0A18	0A60	0041	0A48	1FFF	01D2	0148	0102
15FE	TMC571	0148	017E	017F	017F	09DE	017F	0A08	02F0	0308	1564	0176	0176	1800	0105	0A19	1834	1741
160F	TMC571	0108	0F29	1000	0A98	0A08	1005	0A8A	0A1F	0A03	0704	0A0C	0A11	0A1A	0A08	0658	0A68	072C
1620	TMC571	180F	0A0D	0A18	0A00	0105	0A98	0A98	0106	0098	1810	0A0C	0A97	0A18	0085	0095	0AAA	1804
1631	TMC571	0A9A	0A0C	00F8	0078	1006	0709	1806	0A78	1F14	0219	0648	0A18	0A04	0A01	0A6A	1009	0030
1642	TMC571	0A0C	1800	0045	0A6A	0130	0308	0A0C	1A06	0095	0208	0A64	0A0C	0000	1806	0219	1F2F	0948
1653	TMC571	0A18	0A00	0219	0A4A	1021	0C08	1825	0103	0107	0901	02F6	0A18	0A75	1040	0A10	0AF8	1836
1664	TMC571	0A03	0A75	1A8C	0619	0030	0020	1111	0A67	09F0	0400	0948	1118	0A0C	0095	0075	0045	0A96
1675	TMC571	0A0A	1005	0A8A	000A	1006	0A98	1A07	0A6A	0A4A	0110	1949	0107	0A45	0A97	02F3	0111	0A1F
1686	TMC571	0103	06F6	0658	1267	0A40	0AF0	1005	0A08	116C	0619	1804	0031	0C48	0C48	0A07	0AF3	0A02
1697	TMC571	0098	0689	1222	0A14	0A90	0A44	020C	0A05	0A49	100C	0304	0D8A	1006	0030	1204	090C	0032
16A8	TMC571	0A14	0619	19E8	072C	0A2A	0A44	0A40	0140	0214	0126	0A40	04F0	11CA	0A80	0A08	1008	0A01
16B9	TMC571	0048	1806	0A54	1804	0AC8	0A40	0AF0	1142	0A0A	1009	0A37	09F0	0940	0008	0208	0A27	0AF0
16CA	TMC571	0048	100F	0A58	0A28	0A0C	010E	0A08	02F6	0240	0A58	0A28	0A0C	0C7F	0A0A	09F6	02F3	02A0
16DB	TMC571	1002	0708	1815	0619	1986	0148	0119	1988	0020	1806	0A85	182E	0130	0A0C	0099	0A40	0AF0
16EC	TMC571	1010	0A80	0A08	100F	0A5A	0A04	0AC5	1A07	0048	1146	1890	0020	1804	0085	0ACA	0A3A	1810

TABLE 11c

HEX ADDRESS	CHIP	INSTRUCTION WORDS→															
1AF0	THCS71	0F2A	03D6	1A4F	0A77	0A10	0C7A	0CAA	0A6A	0141	0113	0F7E	095E	097E	097E	097F	097F	097E
1B0F	THCS71	097F	097E	097F	097F	097E	0153	0153	0A0C	0931	0A1A	02F3	0445	1A08	00A5	0455	1A04	0095
1B1F	THCS71	0AA5	1A04	00A5	0A75	1A03	00A5	02A0	097F	0931	0A1A	02F3	0250	1002	0ACA	0A3A	0708	1A0F
1B30	THCS71	0FAC	0732	00D8	1A47	0A0C	0291	0A1A	0AFA	0A3A	0A6A	0A3A	0ACA	0107	0CAA	0CA3	0CA0	0CA0
1B41	THCS71	0CA0	0AD8	07D0	0A04	1002	00A4	00A4	0091	02D8	0090	1A45	0004	0A1F	0090	0529	134F	1FA7
1B52	THCS71	02A0	0231	1002	0A1A	0ACA	0A3A	0172	0708	1A11	0A0C	0105	02D4	0C7A	0A0E	010C	0107	0130
1BA1	THCS71	0107	0A77	0AF0	0094	1F41	0A5A	0A2A	0A0C	0153	0A08	02F3	0A07	0A1A	0AA5	1A04	00A5	0A95
1BF4	THCS71	1A04	0055	0AA5	1A04	00A5	04A5	1A06	0075	02A0	0153	0A08	02F3	02A0	1002	0708	1A37	0CA4
1B45	THCS71	0AFA	0141	0111	0CAC	0A5A	0A2A	0A0C	0732	00D4	0A77	0AF0	1A4A	0A0A	02F3	02AA	0210	0A4A
1B96	THCS71	1A4A	0A1A	0090	0AD2	0404	0C40	0C40	1A03	00AA	1A1A	0720	10E8	0140	02AE	0230	1170	0AA5
1BA7	THCS71	1053	0051	1057	0004	1047	0504	0A1A	0A07	02F0	1012	0A65	101F	02A6	0131	0A0E	0111	0107
1BA8	THCS71	1C1A	0A0C	0300	0101	0A0E	0A0A	0A45	1A04	1013	0A4A	0A0A	1005	0AAA	0A45	1A04	0011	0A87
1BC9	THCS71	02F0	0101	0A1F	0090	0A07	0AF0	0010	1A04	0720	0A0C	0904	0AFA	0904	0A1A	0154	1A04	0095
1BD8	THCS71	0AA5	1A04	0095	0A45	0A3A	0A1A	1A04	0075	0AAA	0720	1A25	0010	0051	0102	1383	0102	0304
1BE8	THCS71	02A0	1A45	0AAA	1A4F	0A90	1CC5	0030	19AC	0A07	02F3	0111	0A1F	0104	0C70	0504	0C7E	0111
1BFC	THCS71	0A0E	0131	0021	1047													

TABLE III

PROGRAM CODE	FUNCTION	KEY(s)
00	0	0
01	1	1
02	2	2
03	3	3
04	4	4
05	5	5
06	6	6
07	7	7
08	8	8
09	9	9
10	E^1	2nd, E
11	A	A
12	B	B
13	C	C
14	D	D
15	E	E
16	A^1	2nd, A
17	B^1	2nd, B
18	C^1	2nd, C
19	D^1	2nd, D
20	Clear	2ND, CLR
21	2nd	2nd
22	Inverse Function	INV
23	LNX	LNX
24	Clear Entry	CE
25	Clear	CLR
26	2nd	2nd, 2nd
27	Inverse Function	2nd, INV
28	log	2nd, log
29	Clear Program	2nd, CP
30	Tangent	2nd, TAN
31	Learn	LRN
32	Exchange display and T register	$X \rightleftarrows T$
33	X^2	X^2
34	$\sqrt{X}$	$\sqrt{X}$
35	1/X	1/X
36	Program Page	2nd, PGM
37	Polar to Rectangular	2nd, $P \rightarrow R$
38	Sine	2nd, Sin
39	Cosine	2nd, COS
40	Indirect Addressing	2nd, IND
41	Single Step	SST
42	Store in Memory #	STO
43	Recall from Memory #	RCL
44	Sum into Memory #	SUM
45	Y^x	Y^x
46	Insert Program Code	2nd, INS
47	Clear Memories	2nd, CMs
48	Exchange Display and Memory #	2nd, EXC
49	Multiply Display into Memory #	2nd, prod
50	Absolute Value	2nd, $\lvert x \rvert$
51	Back Step	BST
52	Exponent Entry	EE
53	(	(

TABLE III-continued

PROGRAM CODE	FUNCTION	KEY(s)
54	)	)
55	÷	÷
56	Delete Program Code	2nd, DEL
57	Engineering Notation	2nd, ENG
58	Fixed Point Notation	2nd, FIX
59	Integer	2nd, INT
60	Degree	2nd, DEG
61	Go To	GTO
62	Indirect Program Page #	2ND, PGM 2ND, IND
63	Exchange Indirect Memory # with display	2ND, EXC 2ND, IND
64	Multiply Display Into Indirect Memory #	2ND, PROD 2ND, IND
65	Multiply	X
66	Pause	2nd, PAUSE
67	Go To # if x = t	2nd, x = t
68	No Operation	2nd, NOP
69	Operation Code #	2nd, OP
70	Radians	2nd, RAD
71	Subroutine Call	SBR
72	Store in Indirect Memory #	STO, 2ND, IND
73	Recall Indirect Memory #	RCL, 2ND IND
74	Add Display into Indirect Memory #	SUM, 2ND IND
75	Minus	−
76	Lable Key	2nd, LBL
77	Go To # if $x \geqq t$	2nd, $x \geqq t$
78	Insert Data Point	2nd, $\Sigma+$
79	Mean	2nd, $\bar{X}$
80	Grad	GRD
81	Reset	RST
82	Hierarchy Address	Not directly accessible
83	Go to Indirect Address	GTO, 2ND IND
84	Operation Code Indirect #	2ND, OP 2ND, IND
85	Plus	+
86	Set Flag #	2nd, ST FLG
87	If Flag # Set, Go To #	2nd, IF FLG
88	Degrees, Minutes, Seconds	2nd, DMS
89	π	2nd, π
90	List Program	2nd, LIST
91	Run/Stop	R/S
92	Return	INV, SBR
93	Decimal Point	.
94	Change Sign	+/−
95	Equals	=
96	Write	2nd, Write
97	Decrement Register # and Go To # when zero	2nd, DSZ
98	Advance Paper	2nd, PAP
99	Print	2nd, PRT

TABLE IV

Program codes following an heirarchy address code (82)	
FIRST DIGIT	FUNCTION
0	store
1	recall
2	conditional return (second digit is ignored)
3	sum into
4	multiply into
5	subtract from
6	divide into
7	"
8	"
9	"
SECOND DIGIT	**HEIRARCHY REGISTER**
0	no operation
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	no operation

TABLE V

Codes Following OP Code (69)	
CODE	MEANING
	Initialize for alphanumeric printing
01	Fill far left quarter of print buffer
02	Fill next to left quarter of print buffer
03	Fill next to right quarter of print buffer
04	Fill far right quarter of print buffer
05	Print the buffer as filled with OPS 01-04
06	Print display plus contents of OP 04
07	Plot asterisk in column number contained in display register (0-19)
08	List labels
09	Download page
10	Signum
11	Variance
12	Slope, intercept
13	Correlation
14	y'
15	x'
16	See current partition RAM
17	Repartition RAM
18	If not error - Set flag 7
19	If error - Set flag 7
20 ' ' 29	Increment memory 0-9
30 ' ' 39	Decrement memory 0-9

TABLE VI

```
SCOMD 1,STOF=0A0F,RCLF=0A1F,REG=LO,DIGITS=USED
ADDRESS CONSTANT=1,2,3,5,6,7,8
CONSTANT   0=2302585092994046C  )
CONSTANT   1=0693147180559945   |
CONSTANT   2=0095310179804325   |
CONSTANT   3=0009950330853168   |
CONSTANT   4=0000999500333084   |  CONSTANTS
CONSTANT   5=0000099995000333   |
CONSTANT   6=0000009999950000   |
CONSTANT   7=0000000999999500   |
CONSTANT   8=0785398163397450   |
CONSTANT   9=0099668652491200   |
CONSTANT  10=0009999666686670   |
CONSTANT  11=0000999999666667   |
CONSTANT  12=0000099999999667   |
CONSTANT  13=157079632679501C   |
CONSTANT  14=314159265359000C   |
CONSTANT  15=572957795130801C   )
CONSTANT  16=4375064353530882   )
CONSTANT  17=5403435501436504   |
CONSTANT  18=8330043750543555   |
CONSTANT  19=2053535454034355   |  PROGRAM CODES
CONSTANT  20=0103859404033265   |
CONSTANT  21=5320825403435554   |
CONSTANT  22=3255541882859453   |
CONSTANT  23=8518826532539254   |
CONSTANT  24=4355044353925432   |
CONSTANT  25=4355014353325403   |
CONSTANT  26=8594533320825403   |
CONSTANT  27=3332540343550243   |
CONSTANT  28=0343550543859453   |
CONSTANT  29=7505435353923254   |
CONSTANT  30=5554034355530043   |
CONSTANT  31=0882540175034353   )

SCOMD 2,STOF=0A0F,RCLF=0A1F,REG=HI,DIGITS=UNUSED
ADDRESS CONSTANT=1,2,3,5,6,7,8
CONSTANT  32=7502435353323454   )
CONSTANT  33=5554034355530143   |
CONSTANT  34=4353539230540882   |
CONSTANT  35=5501436504437506   |
CONSTANT  36=0504353535554043   |
CONSTANT  37=5403035553004375   |
CONSTANT  38=3301437502435365   |
CONSTANT  39=9254305405034355   |
CONSTANT  40=3202403308820140   |
CONSTANT  41=4330782048820444   |
CONSTANT  42=0144204360281015   |  PROGRAM CODES
CONSTANT  43=2235202449433082   |
CONSTANT  44=9043307820882044   |
CONSTANT  45=3782034490410540   |
CONSTANT  46=4332178206401882   |
CONSTANT  47=3285330882539203   |
CONSTANT  48=5353323454330782   |
CONSTANT  49=8553922545055172   |
CONSTANT  50=2254017825518253   |
CONSTANT  51=3908825392245430   |
CONSTANT  52=9253325407823265   |
CONSTANT  53=5392540178265318   |
CONSTANT  54=0006655908825353   |
CONSTANT  55=0165592218825365   |
CONSTANT  56=6554590882540000   |
CONSTANT  57=5459221882850693   |
CONSTANT  58=8255359250460355   |
CONSTANT  59=6559221882855908   |
CONSTANT  60=5400000165580693   |
CONSTANT  61=1882855953530882   |
CONSTANT  62=9365540693655922   |
CONSTANT  63=0000000092540100   )
```

TABLE VIa

LOCATIONS	ROUTINE
16-0-21-6	m,b
21-7-23-1	X^1
23-2-24-2	Y^1
24-3-26-3	mean
26-4-29-2	variance
29-3-34-4	standard deviation
34-5-39-7	correlation coefficient(R)
40-0-42-3	$\Sigma+$
42-4-47-1	$\Sigma-$
47-2-51-2	$R \rightarrow P$
51-3-53-6	$P \rightarrow R$
53-7-58-4	D.MS (degrees, decimal point, minutes and seconds)
58-5-63-3	D.d (degrees, decimal point, fractional degree)

TABLE VII

ALPHANUMERIC CHARACTER	ALPHANUMERIC CODE
A	13
B	14
C	15
D	16
E	17
F	21
G	22
H	23
I	24
J	25
K	26
L	27
M	30
N	31
O	32
P	33
Q	34
R	35
S	36
T	37
U	41
V	42
W	43
X	44
Y	45
Z	46
0	01
1	02
2	03
3	04
4	05
5	06
6	07
7	10
8	11
9	12
blank	00
–	20
.	40
+	47
x	50
*	51
$\sqrt{}$	52
π	53
e	54
(	55
)	56
'	57
$\uparrow$	60
%	61
>	62
<	63
/	64
=	65
,	66
$\overline{x}$	67
2	70
?	71
÷	72
!	73
][	74
Δ	75
II	76
Σ	77

TABLE VIII

BCD ADDRESS	CHIP	PROGRAM CODES→
0	TMC501	25 00 00 5A 02 03 11 01 10 15 15 A2 17 01 19 51 20
17	TMC501	29 21 73 22 91 28 10 26 09 27 A0 29 52 30 95 32 31
38	TMC501	33 63 30 A0 36 51 02 00 00 31 05 36 0A 20 0A 72 07
51	TMC501	AA 0A 02 76 28 00 02 09 AA 5A 09 76 25 28 0A 02 01
AA	TMC501	00 72 01 07 01 00 15 02 76 05 71 28 05 32 03 00 37
A5	TMC501	76 22 37 76 A9 12 AA 76 A9 11 22 AA 22 76 A9 10 53
102	TMC501	28 75 36 15 71 AA 5A 52 22 52 32 03 07 07 03 02 05
119	TMC501	0A 00 09 05 0A A7 06 00 35 76 0A A9 00 01 03 05 06
134	TMC501	03 07 01 07 01 05 69 0A 03 00 A9 03 A9 05 01 00 02
153	TMC501	76 11 0A 00 A2 00 11 00 02 76 12 0A 00 A2 00 12 00
170	TMC501	02 76 13 0A 00 A2 00 13 00 02 76 10 0A 00 A2 00 10
187	TMC501	00 02 76 15 0A 00 A2 00 15 00 02 76 1A 0A 00 A2 00
200	TMC501	1A 00 02 76 17 0A 00 A2 00 17 00 02 76 1A 0A 00 A2
221	TMC501	00 1A 00 02 76 10 0A 00 A2 00 10 00 02 76 1A 0A 00
238	TMC501	A2 00 1A 00 02 76 10 05 53 28 75 01 50 A5 03 07 A5
255	TMC501	07 05 02 76 1A A1 0A 11 01 22 0A 01 03 07 22 0A 02
272	TMC501	02 22 07 05 00 51 75 01 0A 01 03 07 0A 02 73 01 A5
289	TMC501	73 02 A1 00 1A 02 22 07 05 00 A0 75 71 00 20 73 01
304	TMC501	A5 71 02 A1 00 52 02 76 11 02 07 00 0A 02 76 12 75
323	TMC501	32 01 05 A5 03 07 A5 0A 05 02 01 32 02 00 72 01 02
340	TMC501	32 01 0A 01 32 A1 00 03 76 13 03 07 02 05 A5 13 A5
357	TMC501	07 05 02 01 03 05 72 01 01 22 0A 01 07 05 01 1A 01
374	TMC501	02 0A 02 04 03 0A A5 07 05 02 01 75 03 07 75 0A 05
391	TMC501	02 05 03 0A 1A 02 02 73 02 50 32 22 07 05 01 0A 01
40A	TMC501	0A 02 73 02 50 22 77 01 50 32 03 07 A5 03 05 A5 0A
425	TMC501	05 02 03 A1 01 50 03 03 32 03 0A A5 07 05 A7 02 2A
442	TMC501	02 02 01 0A 00 0A 03 07 A5 01 05 02 05 73 02 A1 03
459	TMC501	72 02 03 07 0A 02 0A 03 07 05 02 12 03 0A 10 02 01
47A	TMC501	71 01 00 0A 20 A7 01 31 03 07 02 05 03 0A 10 02 02
491	TMC501	A5 03 05 75 03 0A 05 02 07 73 07 55 73 02 05 02 01
510	TMC501	72 03 03 07 0A 02 0A 03 33 A5 0A 05 32 03 03 77 02
527	TMC501	07 03 01 00 A5 73 02 05 7A 03 A1 02 A0 01 22 0A 05
544	TMC501	03 05 32 03 0A A7 03 12 A1 02 05 01 0A 0A 03 07 32

TABLE VIII

BCD ADDRESS	CHIP	PROGRAM CODES*																	
5A1	TMC5001	43	00	67	05	2A	A1	01	35	10	02	01	71	01	09	0A	03	0A
5?A	TMC5001	0A	09	0A	02	AA	AA	AA	AA	6A	AA	AA	AA	AA	AA	6A	7A	1A
505	TMC5001	42	05	02	02	01	03	07	33	A5	07	05	02	01	03	05	12	03
A12	TMC5001	07	22	77	01	5A	01	0A	01	73	01	22	A7	03	7A	43	07	4A
A20	TMC5001	01	01	4A	05	43	03	72	01	09	02	A1	05	55	7A	15	01	42
4AA	TMC5001	0A	03	07	A5	33	A5	07	05	42	01	43	0A	42	05	75	03	07
AA3	TMC5011	A5	07	05	42	02	00	71	00	30	A5	01	AA	01	73	01	05	72
AA0	TMC5001	01	01	4A	0A	43	0A	32	03	07	77	0A	0A	01	42	0A	07	A5
A07	TMC5001	53	43	07	A5	01	50	33	05	42	01	75	43	07	75	43	0A	42
710	TMC5001	05	05	42	02	00	71	00	52	A5	71	00	20	73	01	05	55	73
711	TMC5001	02	05	72	01	01	4A	0A	43	0A	32	43	07	77	0A	52	0A	01
70A	TMC5001	02	7A	1A	A5	32	07	A5	43	07	A5	53	20	A5	01	53	05	42
7A5	TMC5001	01	32	02	73	01	00	02	01	4A	01	A1	05	25	7A	17	01	42
7A2	TMC5001	0A	03	0A	10	05	42	01	73	01	35	72	01	01	4A	0A	03	0A
790	TMC5001	32	03	07	77	05	4A	01	0A	4A	0A	01	42	07	03	0A	10	42
81A	TMC5001	01	75	43	03	42	05	05	42	02	73	01	A5	73	02	0A	71	00
813	TMC5001	52	05	A5	71	00	20	73	02	05	72	01	01	4A	05	43	0A	32
850	TMC5001	03	03	22	A7	05	A9	01	42	03	22	4A	0A	43	0A	32	01	22
8A7	TMC5001	67	05	A9	03	0A	10	42	01	A5	43	03	42	05	05	42	02	73
8A8	TMC5001	02	0A	71	00	30	A5	01	4A	01	05	72	01	01	4A	03	43	03
901	TMC5001	32	03	07	75	43	0A	05	77	0A	27	01	0A	0A	02	03	43	07
91A	TMC5001	32	43	0A	22	A7	0A	27	01	42	9A	43	0A	42	05	10	42	03
915	TMC5001	43	05	10	42	01	43	03	42	02	43	07	75	43	05	A5	01	05
952	TMC5001	0A	05	32	73	01	0A	71	00	30	05	0A	72	03	43	07	0A	03
9A0	TMC5001	32	A5	01	05	42	05	32	77	0A	42	32	43	0A	A7	0A	0A	A5
9A4	TMC5001	01	05	42	05	43	0A	10	A5	01	05	42	03	43	05	10	42	02
1001	TMC5011	43	03	42	01	43	07	75	43	05	A5	01	05	0A	05	32	73	01
1020	TMC5001	65	73	02	0A	71	00	30	05	0A	72	03	01	4A	03	A5	32	05
1037	TMC5001	42	05	32	03	07	77	07	55	01	4A	0A	A1	0A	A5	0A	01	02
1058	TMC5011	42	03	32	43	07	22	77	0A	10	A5	42	05	33	A5	07	05	42
1071	TMC5001	01	00	02	00	73	01	67	0A	05	01	22	4A	01	07	05	0A	32
10AA	TMC5001	43	05	75	01	05	A5	43	07	A5	07	05	42	01	32	02	01	4A
11A5	TMC5001	01	4A	0A	73	01	00	02	03	0A	32	43	07	22	A7	0A	A0	01
1122	TMC5001	A5	03	03	05	1A	A1	0A	A0	7A	1A	13	2A	67	0A	05	17	43
1139	TMC5001	0A	02	7A	11	0A	0A	0A	03	43	0A	9A	00	02	7A	12	75	32
115A	TMC5001	01	05	A5	43	03	A5	0A	05	42	07	32	0A	02	76	50	72	07
1173	TMC5001	32	01	4A	07	32	00	02	A1	50	76	13	A5	32	03	0A	75	01
1190	TMC5001	05	A5	43	01	A5	0A	05	42	07	32	0A	02	A1	50	7A	1A	4A
12A7	TMC5001	0A	4A	05	43	0A	9A	00	02	76	15	07	42	01	A5	53	03	03
12?0	TMC5001	A5	43	01	50	42	07	05	42	02	7A	33	01	4A	01	4A	02	43
1201	TMC5001	05	A5	73	01	A5	43	0A	A5	73	02	05	72	01	07	07	33	01
125A	TMC5001	02	7A	1A	12	02	7A	05	73	07	00	02	01	0A	07	A1	05	76
1275	TMC5001	17	A5	32	43	03	A5	53	43	0A	A5	01	50	A5	0A	05	42	07
1292	TMC5001	32	0A	02	7A	A0	72	07	32	01	4A	07	32	00	02	A1	A0	7A
1309	TMC5001	1A	43	03	75	32	0A	05	0A	42	01	7A	52	43	03	65	53	43
132A	TMC5001	0A	42	07	A5	01	50	A5	0A	05	42	02	00	7A	50	A5	43	03
1343	TMC5001	4A	01	01	4A	02	73	01	A5	73	02	07	07	50	05	0A	01	4A
1360	TMC5001	01	0A	03	72	01	32	75	01	05	42	01	20	A7	25	0A	A5	0A
1377	TMC5001	05	0A	01	32	A1	52	7A	25	01	02	7A	10	A5	32	07	A5	43
1390	TMC5001	03	A5	43	0A	05	42	07	32	0A	02	7A	57	73	07	00	02	01
13A1	TMC5001	4A	07	A1	57	7A	10	43	01	0A	03	42	01	43	02	4A	00	42
13?A	TMC5001	02	00	02	7A	11	20	7A	0A	02	0A	01	43	02	24	02	7A	1A
13B5	TMC5001	4A	04	0A	03	43	00	24	02	7A	12	43	00	4A	02	43	02	32
13A2	TMC5001	43	03	4A	01	43	01	02	7A	17	01	0A	4A	03	0A	01	12	02
1379	TMC5001	7A	13	53	43	01	A5	43	03	75	43	02	A5	43	0A	50	32	53
1394	TMC5001	43	01	A5	43	0A	A5	43	02	A5	43	03	50	42	02	32	42	01
1413	TMC5001	02	7A	1A	01	0A	4A	0A	53	43	03	33	A5	43	0A	33	50	35
1430	TMC5001	4A	01	4A	02	15	02	7A	1A	10	36	05	12	20	A7	10	1A	05
1447	TMC5011	1A	13	3A	05	17	02	7A	19	3A	A5	1A	10	3A	05	1A	10	1A
145A	TMC5001	02	7A	15	10	3A	05	12	20	A7	10	3A	05	1A	1A	3A	05	17
1471	TMC5001	02	7A	10	53	53	43	02	22	23	A5	35	50	55	02	50	70	02
148A	TMC5001	7A	11	3A	0A	11	02	7A	12	70	03	01	32	43	02	22	37	32
1415	TMC5001	02	76	1A	12	23	02	01	32	42	02	32	02	7A	17	70	03	01
1432	TMC5001	22	23	32	43	02	37	42	02	32	42	01	02	7A	13	12	33	53
1449	TMC5001	32	65	7A	52	02	50	37	02	02	32	42	01	02	76	10	12	30
146A	TMC5001	53	32	55	A1	52	7A	15	01	42	03	00	42	0A	3A	0A	10	3A
1481	TMC5001	0A	1A	02	7A	1A	53	53	43	02	22	23	75	35	50	55	02	50
1?00	TMC5001	02	7A	1A	70	53	53	43	01	A5	01	50	33	A5	43	02	33	50
1717	TMC5001	30	02	7A	10	53	53	43	01	75	01	50	33	A5	43	02	33	50
1730	TMC5001	30	02	7A	15	53	53	1A	A5	10	50	55	02	A5	53	52	22	52
1751	TMC5001	33	75	01	50	30	50	23	02	7A	11	3A	0A	11	02	7A	12	70
17AA	TMC5001	53	43	01	3A	A5	3A	05	10	50	32	53	43	01	39	A5	3A	05
1?A5	TMC5001	1A	50	42	02	32	42	01	02	7A	13	70	53	43	01	39	A5	3A
1?02	TMC5001	A5	10	50	32	53	43	01	3A	00	A5	3A	05	1A	50	42	02	32
1?19	TMC5001	42	01	02	7A	10	43	01	02	03	43	02	02	0A	13	3A	0A	10
1?3A	TMC5001	12	3A	0A	1A	02	7A	17	53	53	1A	75	10	50	55	02	50	22
1?51	TMC5001	3A	32	15	02	02	32	42	01	02	7A	1A	53	53	10	75	10	50
1?70	TMC5001	55	02	50	22	30	32	15	00	42	02	32	42	01	02	7A	10	53
1?A7	TMC5001	53	01	75	43	01	33	75	43	02	33	50	32	53	02	A5	43	01
1?B4	TMC5001	50	22	37	55	02	50	32	53	53	53	43	01	33	A5	53	33	02
1?21	TMC5001	A5	01	53	33	53	55	53	43	01	33	A5	53	43	02	75	01	50
1?3A	TMC5001	33	50	50	23	55	00	50	42	02	32	42	01	02	7A	11	02	00
1?55	TMC5001	00	02	7A	12	53	20	A5	32	05	50	42	01	32	0A	02	76	50
1?72	TMC5001	72	01	32	01	40	01	32	00	02	A1	50	7A	13	0A	00	02	03
1?A0	TMC5001	51	43	00	42	02	A5	05	50	42	01	01	4A	02	73	01	7A	A0
2004	TMC5011	22	07	02	70	22	07	01	70	53	20	A5	43	01	A5	73	01	50
2021	TMC5001	A1	A0	7A	70	00	02	7A	11	42	01	02	7A	12	53	02	02	75
2040	TMC5011	32	43	01	51	42	03	03	00	01	42	0A	32	02	7A	13	02	01

TABLE VIII

BCD ADDRESS	CHIP	PROGRAM CODES→																
2057	TMC5401	92	7A	14	42	0A	92	7A	15	53	2A	A1	02	32	A3	01	77	50
2070	TMC5401	82	0A	A5	A3	03	50	82	01	82	05	3A	00	1A	82	07	51	A3
2081	TMC5401	0A	82	0A	36	00	1A	20	A7	70	A5	0A	07	50	77	15	7A	A0
2108	TMC5401	51	53	A3	01	A5	A3	05	50	55	02	50	82	0A	A3	0A	32	53
2125	TMC5401	11	05	75	03	08	50	22	77	70	51	A3	0A	3A	00	1A	A5	B3
2142	TMC5401	07	50	20	A7	70	77	A5	A3	0A	82	05	A1	A0	7A	A5	A3	0A
2159	TMC5401	82	0A	A1	A0	7A	70	A3	0A	92	76	50	00	35	92	7A	15	53
2174	TMC5401	A3	A1	A5	A3	05	A5	A3	03	50	82	7A	11	82	A1	82	7A	12
2191	TMC5401	82	02	82	7A	13	53	50	82	05	55	02	50	22	50	20	22	A7
2210	TMC5401	52	53	A3	05	3A	A5	53	A3	02	75	A3	A1	50	54	82	A3	82
2227	TMC5401	7A	52	00	35	92	7A	14	15	3A	00	1A	82	0A	7A	50	A1	22
2244	TMC5401	40	05	53	15	3A	00	1A	A5	0A	50	40	0A	22	97	05	B5	53
2261	TMC5401	15	3A	00	1A	A5	02	50	40	0A	A1	50	7A	A5	15	3A	00	1A
227A	TMC5401	40	0A	53	A3	03	55	A3	50	40	0A	A3	00	92	7A	11	53	50
2295	TMC5401	82	05	55	02	50	82	02	22	59	20	22	A7	52	A3	05	0A	92
2312	TMC5401	7A	52	00	35	92	7A	12	A2	A3	0A	92	7A	13	53	2A	A5	32
2329	TMC5401	0A	50	82	A1	32	0A	92	76	50	72	A1	32	A1	40	A1	32	00
2346	TMC5401	92	A1	50	7A	14	53	B3	A5	A5	0A	50	82	A1	73	A1	A2	0A
2363	TMC5401	7A	05	A1	22	80	A1	53	73	A1	A5	00	50	80	0A	A1	22	80
2380	TMC5401	A1	22	97	A2	33	53	73	A1	65	A2	50	80	0A	A1	05	7A	33
2397	TMC5401	73	A1	80	0A	53	B3	A3	55	A3	50	40	0A	B3	0A	0A	00	92
2A1A	TMC5401	7A	1A	53	53	B3	A6	33	A5	B3	A1	33	75	B3	A2	33	50	55
2A31	TMC5401	A2	55	33	0A	55	B3	A1	50	22	30	A7	00	97	B2	A5	0A	A0
2A4B	TMC5401	7A	57	B3	0A	80	A1	80	A2	82	0A	A1	1A	7A	97	A7	A1	80
2A65	TMC5401	82	00	8A	A1	A1	57	76	80	82	A3	A7	A2	90	7A	10	22	8A
2A82	TMC5401	00	22	8A	A1	22	A6	02	22	8A	A3	92	76	1A	53	53	B3	A2
2A99	TMC5401	3A	A5	B3	A1	55	B3	0A	50	22	52	52	22	52	22	3A	52	22
2A1A	TMC5401	52	A2	00	90	A5	A1	90	22	30	75	B3	A2	50	B2	A2	A2	A5
2A33	TMC5401	8A	A3	7A	14	53	B3	0A	33	A5	B3	A1	33	75	A2	A5	A3	0A
2A50	TMC5401	A5	B3	A1	A5	B3	A2	30	50	3A	A7	A3	8A	B2	A2	8A	00	8A
2A67	TMC5401	A2	A1	57	7A	90	B2	A5	B3	A1	A1	10	76	AA	B2	A2	A1	10
2A80	TMC5401	76	17	B3	00	92	7A	1A	B3	A5	92	7A	11	B2	0A	92	7A	12
2A91	TMC5401	B2	A1	92	76	13	B2	A2	92	7A	11	B2	A7	92	7A	12	B2	00
2AA8	TMC5401	92	7A	13	B2	A5	92	76	16	53	A1	90	22	30	75	B3	00	75
2AB5	TMC5401	B3	A5	50	B2	A3	A1	AA	7A	17	53	A1	90	22	30	75	A3	00
2AC2	TMC5401	82	A3	75	B3	A5	50	B2	00	AA	00	7A	AA	53	B3	A7	A5	B3
2AD0	TMC5401	00	3A	55	B3	A3	3A	50	B2	A1	53	B3	A7	A5	B3	A5	3A	55
2AEA	TMC5401	B3	A3	3A	50	B2	A2	A7	00	A0	B3	A3	92	7A	A0	B3	00	22
2B03	TMC5401	AA	00	92	7A	14	B3	A1	92	7A	15	B3	A2	92	7A	1A	53	53
2B20	TMC5401	B3	A7	A5	B3	A1	A5	B3	A2	50	55	A2	50	B2	0A	53	B3	0A
2B37	TMC5401	A5	53	B3	0A	75	B3	A7	50	A5	53	B3	0A	75	B3	A1	50	A5
2B50	TMC5401	53	B3	0A	75	B3	A2	53	50	30	92	7A	11	22	AA	00	22	AA
2B71	TMC5401	A1	20	B2	A1	3A	77	AA	23	7A	AA	B3	A1	92	7A	12	22	A6
2B8A	TMC5401	00	22	AA	A1	20	B2	A2	77	AA	23	B3	A2	7A	AA	92	76	13
2BA5	TMC5401	A2	A3	B2	7A	14	B2	0A	A6	00	AA	A1	B2	7A	1A	A7	00	57
2BB2	TMC5401	53	B3	A3	55	B3	A2	50	B2	A1	92	7A	57	7A	53	53	B3	0A
2BC9	TMC5401	55	A2	55	B3	A2	50	22	3A	A5	A2	50	B2	A1	92	7A	17	A7
2BDA	TMC5401	A1	AA	53	B3	A3	55	B3	A1	50	B2	A2	92	7A	AA	53	73	A9
2BF3	TMC5401	35	A5	B3	0A	50	B2	A2	92	7A	A0	70	53	53	B3	A1	55	A2
2C00	TMC5401	50	3A	65	A2	50	92	7A	1A	53	B3	A1	A5	B3	A2	50	92	7A
2C07	TMC5401	10	73	A0	A5	B3	A2	50	92	7A	15	53	B3	A2	33	A5	B3	A1
2C2A	TMC5401	55	A2	50	92	7A	10	70	53	B3	A1	A5	B3	A2	33	55	A2	75
2C41	TMC5401	B3	A2	33	55	A2	A5	B3	A1	3A	50	92	7A	11	22	AA	A1	20
2C5A	TMC5401	77	A7	90	AA	A1	7A	A7	53	B2	B3	33	22	23	A5	A2	A5	A0
2C73	TMC5401	50	30	35	A2	A1	92	76	12	93	A2	A3	A1	0A	A0	A1	A0	A0
2C92	TMC5401	B3	A1	80	B3	B3	B3	35	53	53	B2	A2	A5	80	A5	A1	93	B3
3000	TMC5401	B3	A0	A2	A7	A0	80	A2	A0	75	B3	A2	A5	B3	A5	A1	B3	AA
3026	TMC5401	A2	A1	A2	A5	A5	A9	A7	AA	A5	B3	A2	A5	A2	A5	A1	93	A7
3043	TMC5401	AA	A1	A0	A7	A7	A9	B3	A7	75	B1	A2	A5	93	B3	A5	7A	A5
3060	TMC5401	0A	B3	A7	8A	B2	A5	93	B3	A1	A9	B5	6A	A1	A5	B3	50	A5
3077	TMC5401	B3	A2	A5	B3	A1	50	A7	A1	A0	92	7A	A0	53	80	A5	A1	50
3094	TMC5401	92	7A	A8	53	53	A2	00	B2	A9	AA	A5	B3	A9	A5	A9	A9	A0
3111	TMC5401	A9	A1	50	55	A1	A9	A9	A0	A1	A7	B2	A7	50	53	53	53	22
312A	TMC5401	59	A5	B3	A7	50	B2	A9	55	B3	A7	A5	A5	22	2A	54	40	55
3145	TMC5401	A5	22	2A	50	92	7A	13	71	AA	53	20	A5	53	B3	11	75	B3
3162	TMC5401	10	50	7A	37	A5	B3	10	50	B2	A7	7A	B3	A7	92	7A	1A	70
3179	TMC5401	71	AA	B2	A9	71	AA	53	53	20	A5	B2	A5	A0	50	30	A5	53
3196	TMC5401	B3	AA	23	A5	B2	A0	50	30	A5	B3	11	A1	37	7A	10	3A	A1
32B3	TMC5401	71	25	92	7A	15	B2	A9	92	7A	11	B2	10	92	76	12	B2	11
32D0	TMC5401	92	7A	13	20	A7	70	7A	A0	B3	A1	A9	AA	A7	A1	A0	7A	3A
32E7	TMC5401	B3	AA	A7	A1	30	92	7A	10	20	A7	70	10	97	B2	13	A1	3A
32FE	TMC5401	7A	15	20	A7	70	10	B3	B2	22	A9	AA	A7	B2	15	A1	3A	7A
3315	TMC5401	70	A1	B2	AA	B3	3A	7A	11	32	22	8A	A1	A1	B2	A1	B2	B3
332C	TMC5401	32	7A	17	20	77	A7	8A	A1	50	7A	A7	72	A1	32	73	A3	50
3343	TMC5401	A7	AA	8A	A1	7A	AA	72	A3	92	7A	30	A0	35	B3	A1	92	7A
335A	TMC5401	10	B3	A1	A9	AA	A1	22	AA	A1	92	7A	12	32	B2	B2	A3	32
3371	TMC5401	17	32	B3	A1	77	50	A0	35	92	7A	30	B3	B2	92	7A	A7	B3
3388	TMC5401	B5	72	A1	AA	A1	A1	AA	A1	53	B3	A1	55	32	50	92	7A	12
339F	TMC5401	B2	A5	A7	A1	A0	A1	AA	A3	B3	A3	32	B3	B2	77	A7	8A	B2
33B6	TMC5401	A1	8A	A1	7A	A0	71	A1	22	AA	AA	B3	B5	AA	A3	72	A1	A1
33CD	TMC5401	B3	A1	B2	A1	32	51	B3	B2	A5	A5	50	77	77	8A	B2	A1	76
33E4	TMC5401	77	53	B3	AA	55	B3	B2	50	92	7A	70	A0	35	92	7A	10	22
33FB	TMC5401	8A	A1	8A	B2	A1	A0	B2	A3	B2	AA	92	7A	11	20	A7	70	22
3412	TMC5401	77	70	B2	B2	32	B3	B2	50	22	A7	70	92	7A	1A	53	53	B3
3429	TMC5401	A0	A5	B3	A1	75	B2	12	A1	50	55	B3	8A	50	B2	7A	11	53
3440	TMC5401	1A	A5	B3	A0	50	B2	7A	1A	53	1A	55	B3	12	50	92	7A	10
3457	TMC5401	1A	53	20	A5	B3	A0	50	92	7A	10	20	50	B2	B2	7A	71	53
346E	TMC5401	B3	AA	55	B3	B3	50	92	7A	25	53	71	71	21	55	11	A0	23
3485	TMC5401	50	B2	A1	92	7A	11	A7	25	B2	A1	92	7A	22	51	A1	8A	A0
349C	TMC5401	75	20	A5	71	71	22	A5	B3	A1	50	A0	A7	15	7A	12	A7	22

TABLE VIII

BCD ADDRESS	CHIP	PROGRAM CODES+
35A7	TMC501	7A 15 02 02 53 24 55 01 00 00 85 02 0A 01 50 02 00
36A8	TMC501	03 02 02 7A 23 53 03 08 55 03 09 05 03 01 50 02 03
3A21	TMC501	02 7A 13 A7 25 02 03 02 7A 24 53 03 03 A5 03 00 05
343A	TMC501	03 01 50 02 00 02 7A 14 A7 24 02 00 02 7A 11 A1 03
3655	TMC501	7A 7A 12 A1 02 02 7A 13 A1 00 03 7A 10 A1 05 01 7A
3672	TMC501	15 A1 05 39 01 02 09 07 90 22 2A 32 53 03 00 55 03
3A80	TMC501	03 50 02 10 87 03 01 36 A7 00 01 36 53 24 55 03 01
370A	TMC501	33 75 03 10 35 50 82 0A 00 09 3A 1A 1A 82 18 A7 01
3725	TMC501	00 80 53 24 65 03 09 50 53 24 75 03 10 50 02 13 53
3740	TMC501	03 10 55 03 0A 75 03 01 A5 03 12 55 03 09 55 03 0A
3757	TMC501	50 A7 01 01 20 53 24 A5 03 00 75 03 10 50 22 00 13
3774	TMC501	03 13 00 0A 00 09 50 77 00 A5 A1 02 A0 53 03 05 55
3701	TMC501	03 03 A5 02 10 03 01 00 10 75 03 10 50 02 0A 53 03
3A0A	TMC501	01 A5 33 55 02 A5 03 10 55 03 01 50 A7 03 01 A0 53
3A25	TMC501	03 11 A5 33 55 02 A5 03 10 55 03 01 50 A7 03 01 A0 53
3A42	TMC501	3A 1A 1A 02 0A A7 03 02 0A 53 24 A5 03 09 50 53 24
3A50	TMC501	A5 03 10 55 03 01 55 03 12 75 03 10 50 02 13 53 03
3A7A	TMC501	0A 55 03 0A 75 03 01 55 03 12 55 03 09 55 03 0B 50
3A93	TMC501	A7 03 02 55 53 24 A5 03 00 A5 03 0A 50 53 24 A5 03
3910	TMC501	10 55 03 12 55 03 00 50 22 00 13 03 13 00 0A 00 00
3927	TMC501	50 77 01 01 53 03 0A 65 01 00 00 50 20 A7 02 05 A7
3940	TMC501	00 25 5A 00 3A 1A 15 02 AA 03 03 02 11 02 10 03 00
3941	TMC501	A7 02 03 10 87 00 03 10 01 40 11 00 10 53 03 0A A5
397A	TMC501	03 0A 50 A7 01 03 50 A7 02 03 50 00 00 10 53 03 05
3995	TMC501	A5 03 0A 50 00 00 11 A1 03 56 00 11 03 11 25 02 11
3012	TMC501	03 10 23 00 00 11 03 00 23 55 00 11 03 11 A7 01 A1
3020	TMC501	A7 03 01 02 01 00 02 11 00 5A 00 0A 02 0A A7 01 00
30A4	TMC501	22 A7 02 00 16 A7 03 00 10 3A 1A 10 61 00 25 3A 1A
30A3	TMC501	1A A1 00 25 3A 1A 17 A1 00 25 3A 1A 1A 53 20 55 53
30A0	TMC501	03 00 75 03 05 55 03 12 50 50 35 A7 00 06 A7 03 01
3097	TMC501	02 03 5A 02 0A 02 0B A7 01 00 03 A7 02 00 77 A7 03
3114	TMC501	00 71 3A 1A 10 A1 00 0A 3A 1A 1A A1 00 06 3A 1A 17
3131	TMC501	61 00 0A 36 1A 1A 53 24 A5 03 03 A5 03 05 55 03 12
310A	TMC501	50 A7 05 00 A7 00 52 02 00 5A 02 0A 02 0A 36 1A 10
3125	TMC501	A7 00 05 21 3A 1A 1A 53 53 24 A5 03 03 75 03 00 50
3142	TMC501	00 A5 03 12 50 A7 05 02 A7 05 10 02 05 5A 02 0A 02
3190	TMC501	7A 1A 0A 01 02 7A 17 06 02 02 7A 1A 0A 03 02 7A 10
3216	TMC501	0A 00 02 7A 10 22 0A 01 22 0A 02 22 0A 03 22 0A 00
3213	TMC501	00 02 05 5A 00 20 02 7A 7A 53 03 00 65 03 01 05 02
3250	TMC501	03 03 50 50 0A A5 01 77 7A 70 00 35 02 7A 10 53 02
3247	TMC501	01 20 22 77 70 22 50 22 00 01 A5 00 22 20 50 02 03
32A0	TMC501	32 01 05 0A 01 77 70 03 02 32 53 53 03 01 55 01 00
3301	TMC501	00 50 02 01 22 50 22 00 01 A5 01 00 00 50 02 02 77
331A	TMC501	70 01 03 32 03 01 77 70 53 03 0A 05 A5 03 03 A5 03
3315	TMC501	02 A5 03 01 A5 03 01 75 03 01 A5 03 32 03 01 77 7A
3352	TMC501	01 22 00 03 7A 77 53 03 03 55 01 50 50 75 53 03 07
33A0	TMC501	05 A5 53 03 03 55 01 00 00 50 50 A5 03 07 05 50 50
33AA	TMC501	50 02 7A 11 10 02 00 00 02 7A 12 10 02 05 00 02 7A
3003	TMC501	13 53 03 05 75 03 00 50 02 7A 10 53 53 10 02 01 00
3020	TMC501	55 07 50 50 A5 07 A5 03 01 50 02 7A 13 53 01 00 00
3037	TMC501	53 32 75 03 03 50 02 05 53 35 50 A5 03 05 50 02 7A
3050	TMC501	10 00 7A 17 32 01 22 00 02 53 02 05 03 02 A5 32 50
3071	TMC501	50 02 7A 12 53 36 15 71 AA A5 01 00 02 03 B5 01 50
30AA	TMC501	50 02 03 00 02 00 02 7A 10 20 03 00 02 7A 1A 00 02
3105	TMC501	02 05 01 02 0A 02 7A 10 53 01 00 75 03 02 50 02 7A
3122	TMC501	11 02 00 02 7A 10 A7 00 05 05 A1 00 7A A5 04 7A 00
3130	TMC501	02 10 73 10 02 76 13 00 7A 12 53 20 A5 10 50 72 10
3156	TMC501	02 7A 11 10 02 7A 1A 22 7A 17 0A 00 5A 02 02 7A 1A
3173	TMC501	02 07 02 7A 10 53 35 A5 03 07 50 20 3A 1A 12 02 7A
3200	TMC501	10 20 3A 1A 11 03 00 02 03 20 3A 1A 71 21 02 04 02
3207	TMC501	7A 15 A7 00 30 02 05 02 7A 30 02 04 02 7A 12 53 AA
3220	TMC501	A5 7A 15 03 01 50 53 20 A5 53 20 55 50 50 20 A5 05
3241	TMC501	00 22 20 50 22 AA 5A 00 02 7A 13 53 20 A5 A1 15 7A
325A	TMC501	10 53 35 A5 61 15 7A 11 50 00 AA 02 01 02 7A 11 53
3275	TMC501	20 A5 02 03 05 00 50 02 7A 12 53 20 A5 03 03 00 00
3292	TMC501	00 50 02 7A 13 53 20 65 03 00 01 00 00 50 02 7A 10
3309	TMC501	53 20 65 01 03 00 00 00 03 00 00 50 02 7A 15 53 20
3326	TMC501	A5 03 00 00 00 00 07 00 02 00 50 02 7A 10 35 11 35
3343	TMC501	02 76 17 35 12 35 02 7A 10 35 13 35 02 76 10 35 10
3360	TMC501	35 02 7A 10 35 15 35 02 7A 11 53 53 20 75 03 02 50
3377	TMC501	65 05 55 00 50 02 7A 12 53 20 A5 03 00 02 00 05 07
3304	TMC501	03 05 02 00 00 50 02 76 13 53 20 A5 03 03 03 07 00
3411	TMC501	00 01 01 07 00 00 50 02 7A 10 53 20 A5 02 00 03 03
3428	TMC501	00 00 05 02 03 01 03 50 02 7A 15 53 20 A5 03 00 05
3445	TMC501	01 05 00 02 03 07 50 02 7A 10 53 20 A5 01 03 00 A5
3462	TMC501	03 02 50 02 7A 17 35 12 35 02 7A 10 35 11 35 02 76
3479	TMC501	10 35 10 35 02 7A 10 35 15 35 02 A1 11 02 02 02 02
3490	TMC501	02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
3413	TMC501	02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
3430	TMC501	02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
3447	TMC501	02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
345A	TMC501	02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
3461	TMC501	02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02 02
347A	TMC501	02 02

What is claimed is:

1. In an electronic microprocessor system, having a keyboard, output means for outputting data, a data memory for storing data, an arithmetic unit for performing arithmetic operations on the data stored in said data memory, and a first nonvolatile memory for storing

groups of instruction words for controlling the arithmetic operations performed by said arithmetic unit, the combination which comprises:

(a) a second non-volatile memory for storing a plurality of sets of program codes, each program code being effective for addressing a preselected group of instruction words stored in first non-volatile memory, said second non-volatile memory being disposed in a module having a plurality of electrical contacts;

(b) a receptacle for temporarily interconnecting the contacts on said module with said microprocessor system;

(c) keyboard logic means for decoding inputs received at said keyboard;

(d) means for addressing said second non-volatile memory to read out preselected sets of program codes, said means for addressing said second memory including a counter responsive to selected instructions outputted from said first memory;

(e) means for addressing said first non-volatile memory in response to the program codes read out of said second non-volatile memory, said means for addressing said first memory including a first program counter selectively responsive to said keyboard logic means and said program codes outputted from said second memory.

2. The electronic microprocessor system according to claim **1**, wherein said first and second non-volatile memory are first and second read-only-memories, respectively.

3. The electronic microprocessor system according to claim **1**, further including means for loading said counter in said means for addressing said second read-only-memory with a multibit number stored in said data memory.

4. The electronic microprocessor system according to claim **3**, wherein said data memory comprises a plurality of registers and a multibit register coupled to the output of said arithmetic unit.

5. The electronic microprocessor system according to claim **4**, wherein said keyboard logic means includes a keyboard register, the contents of which are loaded into said first program counter in response to a particular instruction word outputted from said first read-only-memory and loaded into said counter addressing second read-only-memory in response to another instruction word outputted from said first read-only-memory.

6. The microprocessor system according to claim **5**, further including means for loading said keyboard register means with the contents of said multibit register connected to the output of said arithmetic unit in response to a given instruction word outputted from said first read-only-memory, whereby said arithmetic unit, multibit register connected to the output thereof, keyboard register means, and means for loading said keyboard register means provides said means for loading said counter addressing said second read-only-memory with a multibit number stored in said data memory.

7. The electronic microprocessor system according to claim **6**, further including random access memory means for storing a user defined program comprising of a selected sequence of key pushes at said keyboard, said selected sequence of key pushes being stored as a sequence of program codes therein and means for loading said program codes stored in said random access memory means into said first program counter.

8. The electronic microprocessor system according to claim **7**, wherein said electronic microprocessor system is provided in a programmable calculator.

9. The electronic microprocessor system according to claim **1**, further including random access memory means for storing a user defined program comprising of a selected sequence of key pushes at said keyboard, said selected sequence of key pushes being stored as a sequence of program codes therein and means for loading said program codes stored in said random access memory means into said first program counter.

10. The electronic microprocessor system according to claim **9**, further including means for loading said counter in said means for addressing said second read-only-memory with a multibit number stored in said data memory.

11. The electronic microprocessor system according to claim **10**, wherein said keyboard logic means includes keyboard register means for loading said first program counter in response to a particular instruction word outputted from said first read-only-memory and for loading said counter addressing second read-only-memory in response to another instruction word outputted from said first read-only-memory.

12. The electronic microprocessor system according to claim **10**, wherein said keyboard logic means includes keyboard registers, the contents of which are loaded into said first program counter in response to a particular instruction word outputted from said first read-only-memory and loaded into said counter addressing second read-only-memory in response to another instruction word outputted from said first read-only-memory.

13. The electronic microprocessor system according to claim **1**, wherein said keyboard logic means includes keyboard register, the contents of which are loaded into said first program counter in response to a particular instruction word outputted from said first read-only-memory and loaded into said counter addressing second read-only-memory in response to another instruction word outputted from said first read-only-memory.

14. The electronic microprocessor system according to claim **13**, further including random access memory means for storing a user defined program comprising of a selected sequence of key pushes at said keyboard, said selected sequence of key pushes being stored as a sequence of program codes therein and means for loading said program codes stored in said random access memory means into said first program counter.

15. The electronic microprocessor system according to claim **1**, wherein a plurality of said second non-volatile memories are provided in a plurality of modules, each one of said plurality of second-non-volatile memories storing different sequences of program codes, at least one of said modules including the second non-volatile memory disposed therein, being receivable in said receptacle at any given time.

16. The electronic microprocessor system according to claim **15**, wherein said electronic microprocessor system is provided in a programmable calculator.

17. The programmable calculator according to claim **16**, wherein said calculator includes a case in which said data memory, arithmetic unit receptacle and first non-volatile memory are disposed, said case having an opening adjacent to said receptacle for temporarily receiving at least a selected one of said modules.

18. The programmable calculator according to claim **17**, wherein said first and second non-volatile memory are first and second read-only-memories, respectively.

19. The programmable calculator according to claim

17, further including a magnetic card reader for reading magnetic cards providing storage for programs including a plurality of program codes, a program memory for storing program codes read by said magnetic card reader, means for reading out of said program memory said program codes and means for addressing said first non-volatile memory in response to the program codes read out of said program memory.

20. The programmable calculator according to claim 19, wherein said input means includes a keyboard and wherein said calculator may be selectively placed in a learn mode or run mode at said keyboard and wherein said calculator further includes means for storing the program codes according to the keys depressed at said keyboard in said program memory when said calculator is in said learn mode.

21. An electronic microprocessor system, having input means for receiving data and for receiving input commands, said input means including a keyboard, output means for outputting data, a data memory for storing data received and data to be outputted, an arithmetic unit for performing arithmetic operations on data stored in said data memory, and a first non-volatile memory for storing groups of instruction words for controlling the arithmetic operations performed by said arithmetic unit, the combination which comprises:

(a) a second non-volatile memory for storing a plurality of sets of program codes, each program code being effective for addressing a preselected group of instruction words stored in said first non-volatile memory;

(b) keyboard logic means for decoding inputs received at said keyboard;

(c) means for addressing said second non-volatile memory to read-out preselected sets of program codes in response thereto, said means for addressing said second memory including a counter responsive to selected instructions outputted from said first memory;

(d) means for addressing said first non-volatile memory in response to the program codes read-out of said second non-volatile memory, said means for addressing said first memory including a first program counter selectively responsive to said keyboard logic and to said program codes outputted from said second memory;

(e) magnetic card reader means for receiving program codes stored on magnetic cards;

(f) program memory means for storing program codes received by said magnetic card reader means;

(g) means for addressing said first non-volatile memory in response to the program codes read-out of said program memory means; and

(h) wherein said means for addressing said first memory is further responsive to the program codes outputted from said program memory.

22. The system according to claim 21, wherein said first and second non-volatile memories are first and second read-only-memories respectively.

23. In an electronic microprocessor system of the type having a keyboard, an arithmetic unit for performing numerical operations on data, a first memory for storing a plurality of groups of instruction words, a first address register for addressing the first memory, keyboard logic means for inserting addresses into said first address register in response to key depressions at said keyboard and instruction word decoder means for controlling the arithmetic unit in response to instruction words outputted from the first memory, the combination which comprises:

(a) a second memory for storing a plurality of sets of program codes, each program code being effective for addressing a preselected group of instruction words stored in the first memory, the second memory being disposed in a module having a plurality of electrical contacts;

(b) a receptacle for temporarily interconnecting the contents on said module with said microprocessor system;

(c) a second address register for addressing the second memory;

(d) means for inserting a preselected address into said second address register in response to the depression of a "program" key at said keyboard;

(e) means for comparing the program code stored at said preselected address with a numerical value inputted at said keyboard after the depression of said "program" key; and

(f) means for displaying an error condition when the results of the comparison indicate that the numerical value inputted after the depression of the "program" key is greater than the numerical value of the program code stored at the preselected address.

24. The system as defined in claim 23, wherein said first and second memories are first and second read-only-memories.

25. An integrated circuit disposed in a module, the module being temporarily receivable in a receptacle of a calculator and the calculator including an instruction word memory for controlling the operations performed by the calculator, the integrated circuit comprising:

(a) a program memory for storing a plurality of program codes, each program code having two four-bit digits and being adapted for use by said calculator to address said instruction word memory;

(b) an address register for addressing the program memory;

(c) an instruction decoder responsive to bit serial instructions generated by said calculator;

(d) first means coupled to said program memory for outputting the four most significant bits of the addressed program code in serial to said calculator in response to said instruction decoder decoding a "FETCH HIGH" instruction from said calculator; and

(e) second means coupled to said program memory for outputting the four least significant bits of the addressed program code in serial to said calculator in response to said instruction decoder decoding a "FETCH" instruction from said calculator.

26. The system according to claim 25, further including means for incrementing the address in said address register in response to a decoded "FETCH" instruction.

27. The system according to claim 26, wherein said address register stores a four digit address and further including means for outputting to said calculator in serial the contents of a sequentially different digit position of said address register each time said instruction decoder decodes an "UNLOAD PC" instruction from said calculator.

28. A plug-in memory module for use in an electronic microprocessor system having a keyboard, logic means responsive to the keyboard for providing a predetermined

one of a plurality of program codes in response to the selective actuation of the keyboard, a data memory for storing data, an instruction memory for storing groups of instructions in a non-volatile form, program counter means coupled to the logic means and to the instruction memory for addressing a predetermined one of the groups of instructions in response to each of the program codes, an arithmetic unit for performing arithmetic operations on the data stored in the data memory in response to the addressed instructions, output means for indicating the results of the arithmetic operations, and receptacle means including contacts for temporarily receiving the memory module and for coupling the memory module to the instruction memory and to the program counter means, the memory module comprising:

a memory module housing having contacts connectable to the contacts of said receptacle means;

a program memory disposed within said memory module housing for storing at least one sequence of stored program codes in a non-volatile form;

counter means disposed within said memory module housing and connectable to the instruction memory via said contacts, said counter means being connected to the program memory for addressing a predetermined one of said stored program codes in response to selected ones of the addressed instructions; and

means disposed within said memory module housing for providing said addressed stored program codes to the program counter means via said contacts, whereby the instruction memory is also addressable by the program counter means in response to said addressed stored program codes.

29. A plug-in memory module for use in conjunction with a data processing system having a first memory means for storing a plurality of instruction codes and a first address generator for generating addresses of said instruction codes in said first memory means, said memory module comprising:

(a) A housing member;

(b) A plurality of electrical conductor means disposed at least partially within said housing member said conductor means for temporarily electrically coupling said plug-in module to said system;

(c) Second memory means disposed within said housing member and coupled to said conductor means, for storing a plurality of program codes and for providing said program codes to said system via said conductor means, each of said program codes being effective to cause said system to execute a plurality of said instruction codes; and

(d) Second address generator means disposed within said housing member, coupled to said second memory means and to said conductor means for selectively generating addresses of said program codes in said second memory means, said address generator means including counter means for incrementing the address of said second memory means.

30. The plug-in memory module according to claim 29 wherein said address counter is responsive to signals received from said system via said conductor means for setting the address of said second memory means to a selected address.

31. The plug-in memory module according to claim 30 wherein said selected address is a branch address.

32. The plug-in memory module according to claim 29 wherein said counter means receives incrementing signals from said system via said conductor means and increments the address of said second memory means in response to said incrementing signals.

33. The plug-in memory module according to claim 29 wherein said second memory means is a nonvolatile memory means.

34. The plug-in memory module according to claim 33 wherein said nonvolatile memory is a read only memory means.

35. The plug-in memory module according to claim 29 wherein said first memory means is a nonvolatile memory means.

36. The plug-in memory module according to claim 35 wherein said nonvolatile memory is a read only memory.

* * * * *