United States Patent [19]

Poland

[11] **4,153,937**

[45] **May 8, 1979**

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

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

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

[21] Appl. No.: **783,903**

[22] Filed: **Apr. 1, 1977**

Related U.S. Application Data

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

[51] **Int. Cl.**[2] **G06F 3/08; G06F 15/02**

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

[58] **Field of Search** 235/156; 364/200, 900, 364/706

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,593,313	7/1971	Tomaszewski et al.	340/172.5
3,760,171	9/1973	Wang et al.	235/156
3,859,636	1/1975	Cook	340/172.5
3,942,156	3/1976	Mock et al.	340/172.5
3,953,833	4/1976	Shapiro	340/172.5
3,971,925	7/1976	Wenninger et al.	235/156

OTHER PUBLICATIONS

"National's Scientific Calculator HAS Long-Term All-Semiconductor Memory", *Electronics*, Jun. 10, 1976, pp. 29-30.

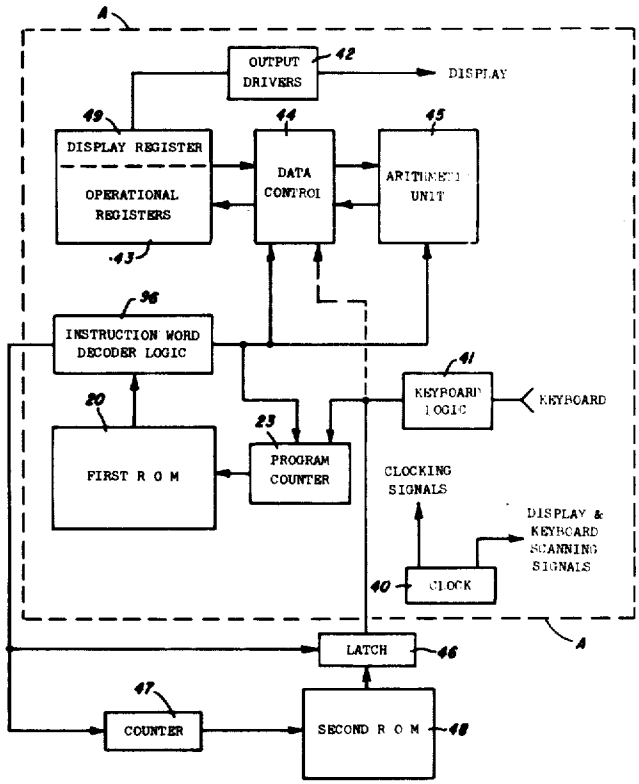
Primary Examiner—Jerry Smith
Attorney, Agent, or Firm—Stephen S. Sadacca; James T. Comfort

[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.

27 Claims, 20 Drawing Figures



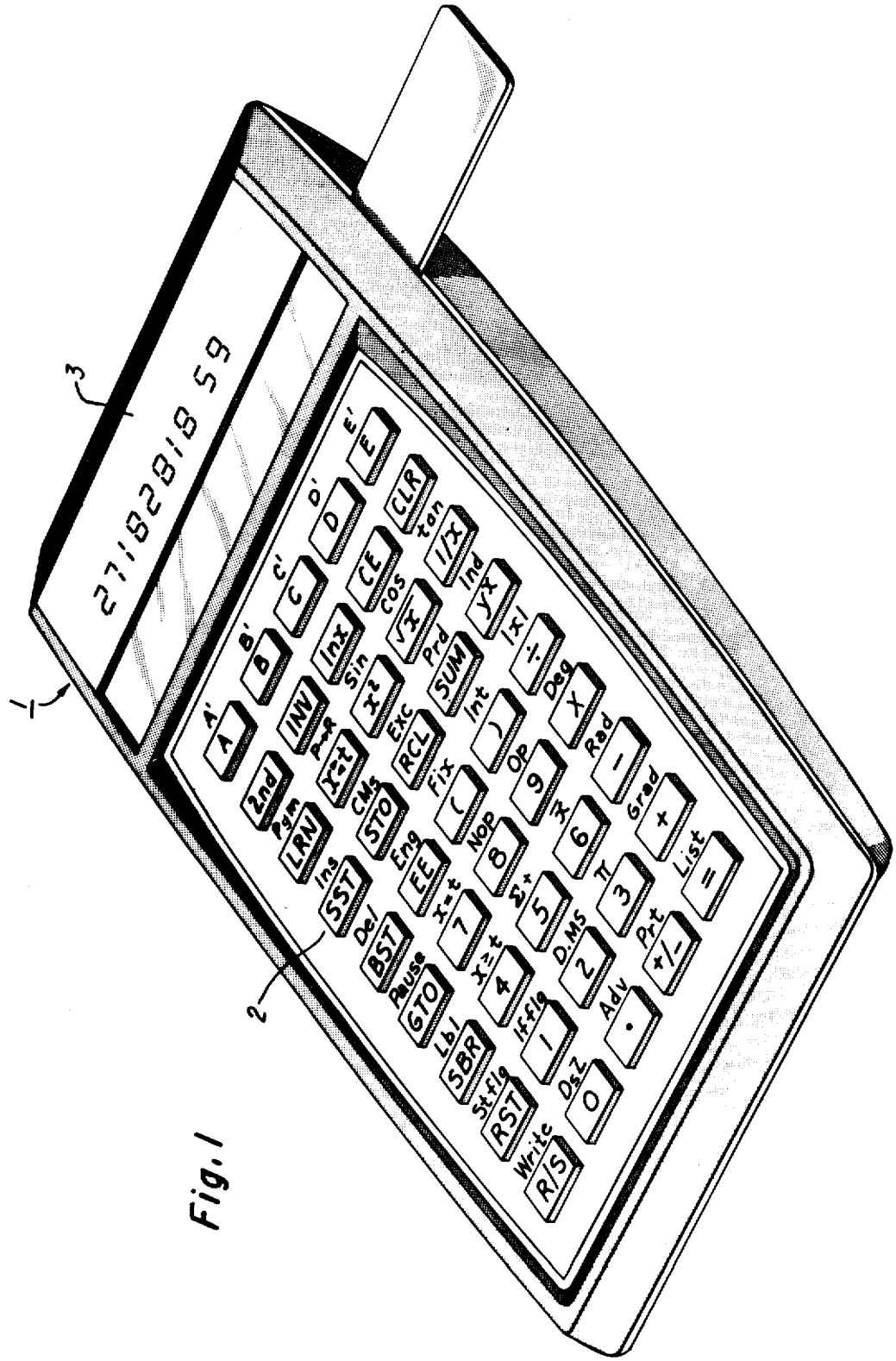

3
1
2
A' A
B' B
C' C
D' D
E' E
CLR
tan
1/x
Ind
yx
|x|
÷
Deg
X
Rad
−
Grad
+
List
=
INV
Pwr
2nd
Pgm
LRN
Ins
SST
Del
BST
Pause
GTO
Lbl
SBR
St flg
RST
Write
R/S
ln x
Sin
x²
Exc
RCL
Fix
(
Nop
STO
Eng
EE
x=t
7
x≥t
4
If flg
1
Dsz
0
CE
cos
√x
Prd
SUM
Int
)
OP
9
π
6
Σ+
5
D.MS
2
Adv
.
TT
3
Prt
+/−
27.182818 59

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

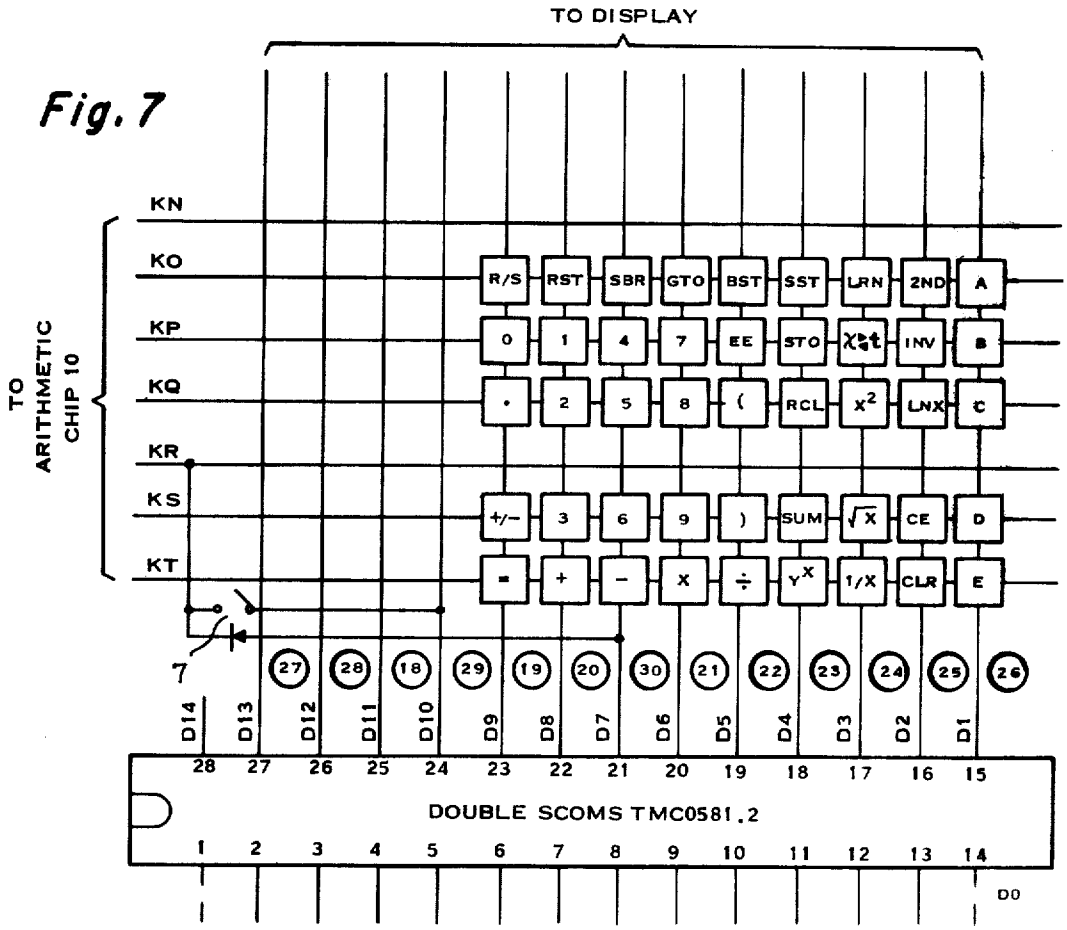


Fig. 11

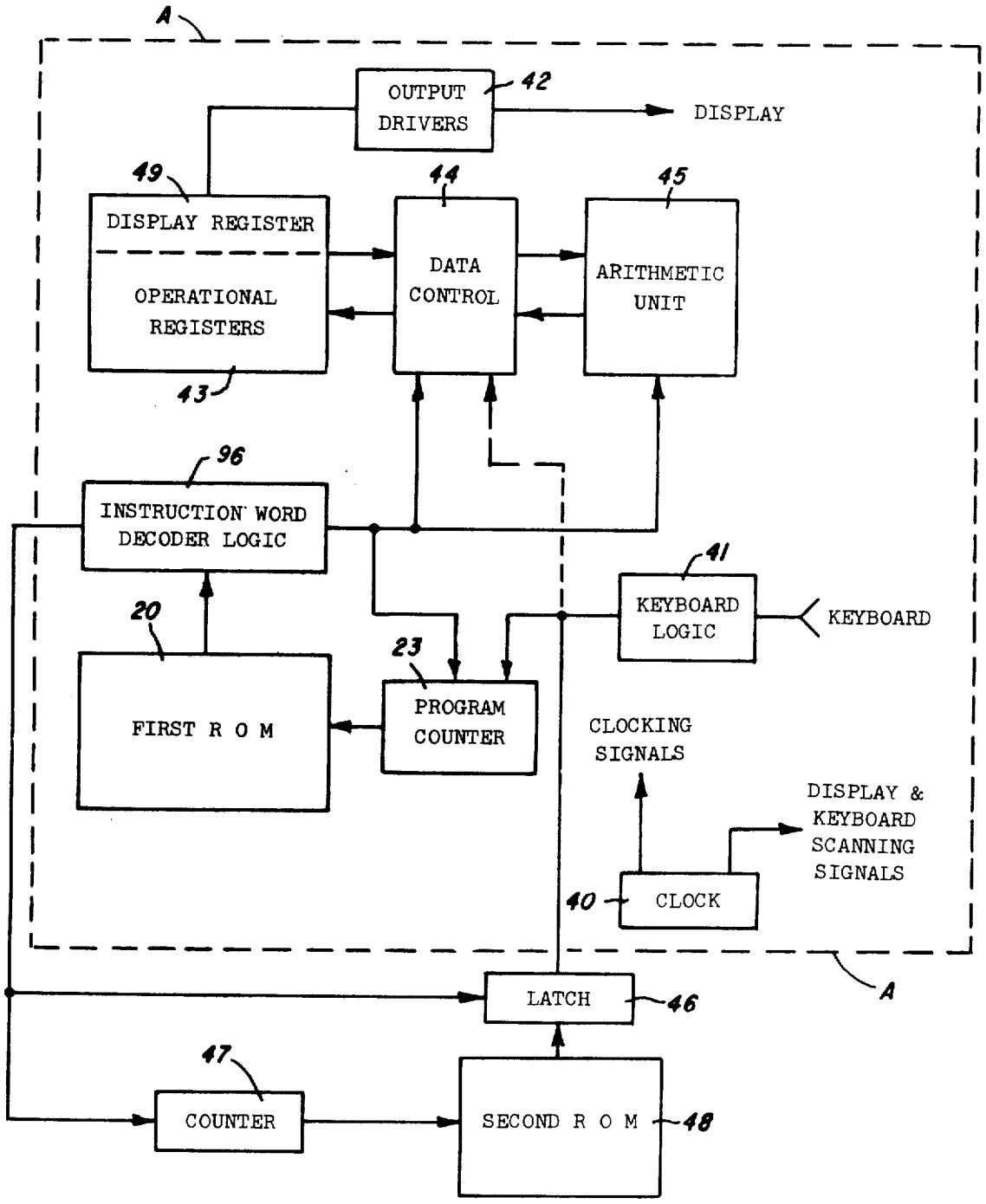

A
OUTPUT DRIVERS
42
DISPLAY
49
DISPLAY REGISTER
OPERATIONAL REGISTERS
43
44
DATA CONTROL
45
ARITHMETIC UNIT
96
INSTRUCTION WORD DECODER LOGIC
41
KEYBOARD LOGIC
KEYBOARD
20
FIRST R O M
23
PROGRAM COUNTER
CLOCKING SIGNALS
DISPLAY & KEYBOARD SCANNING SIGNALS
40
CLOCK
A
LATCH
46
47
COUNTER
SECOND R O M
48

Fig. 12

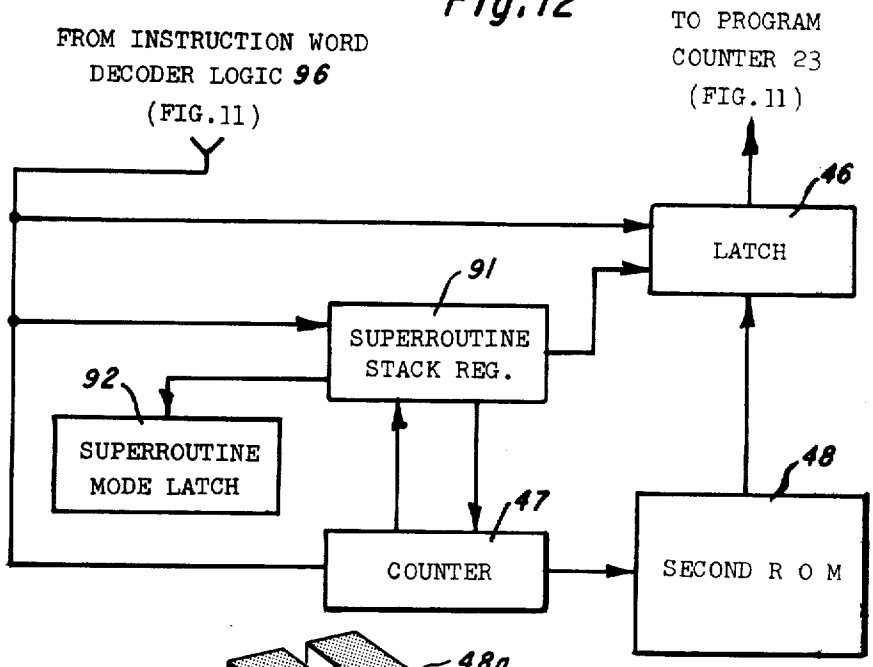
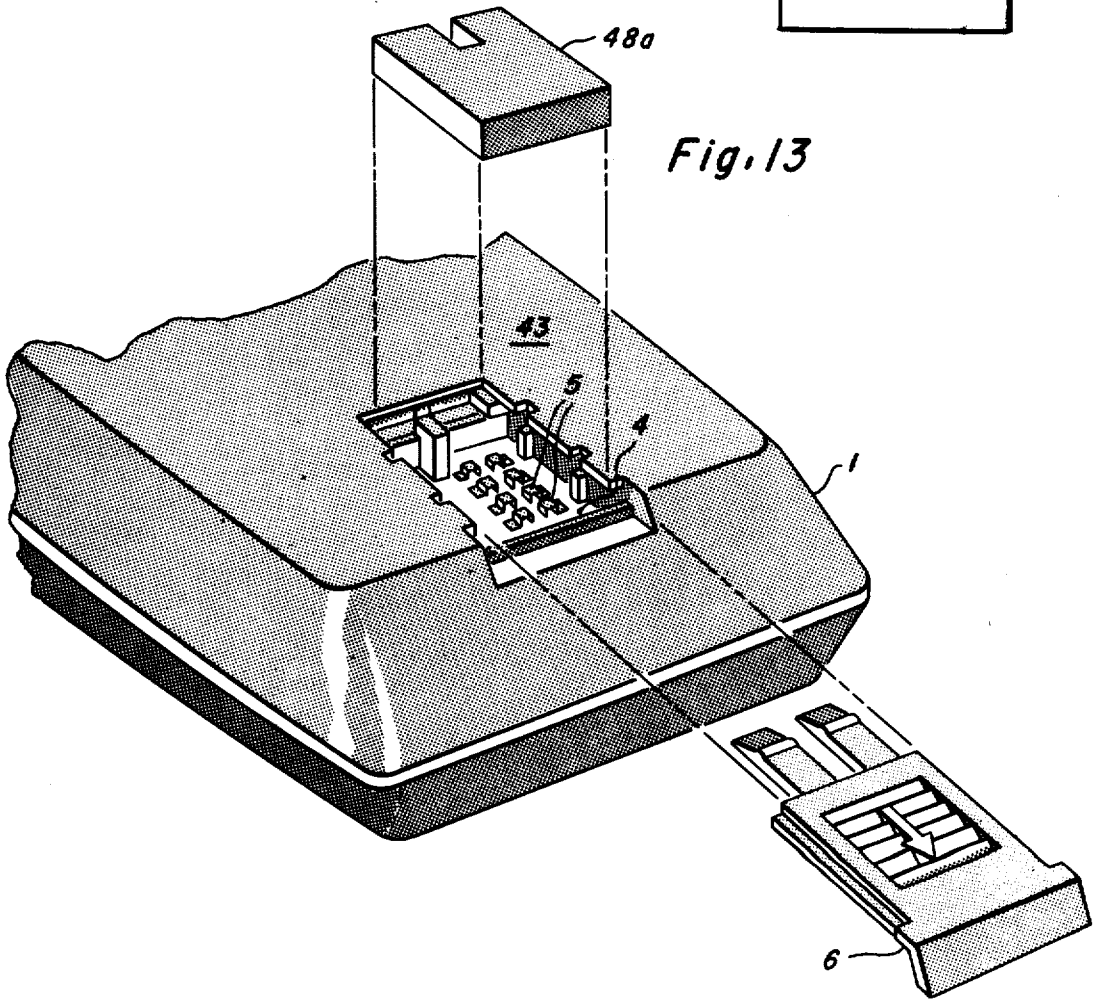

FROM INSTRUCTION WORD DECODER LOGIC 96 (FIG. 11)
TO PROGRAM COUNTER 23 (FIG. 11)
46
LATCH
91
SUPERROUTINE STACK REG.
92
SUPERROUTINE MODE LATCH
47
COUNTER
48
SECOND R O M

Fig. 13


48a
43
5
4
1
6

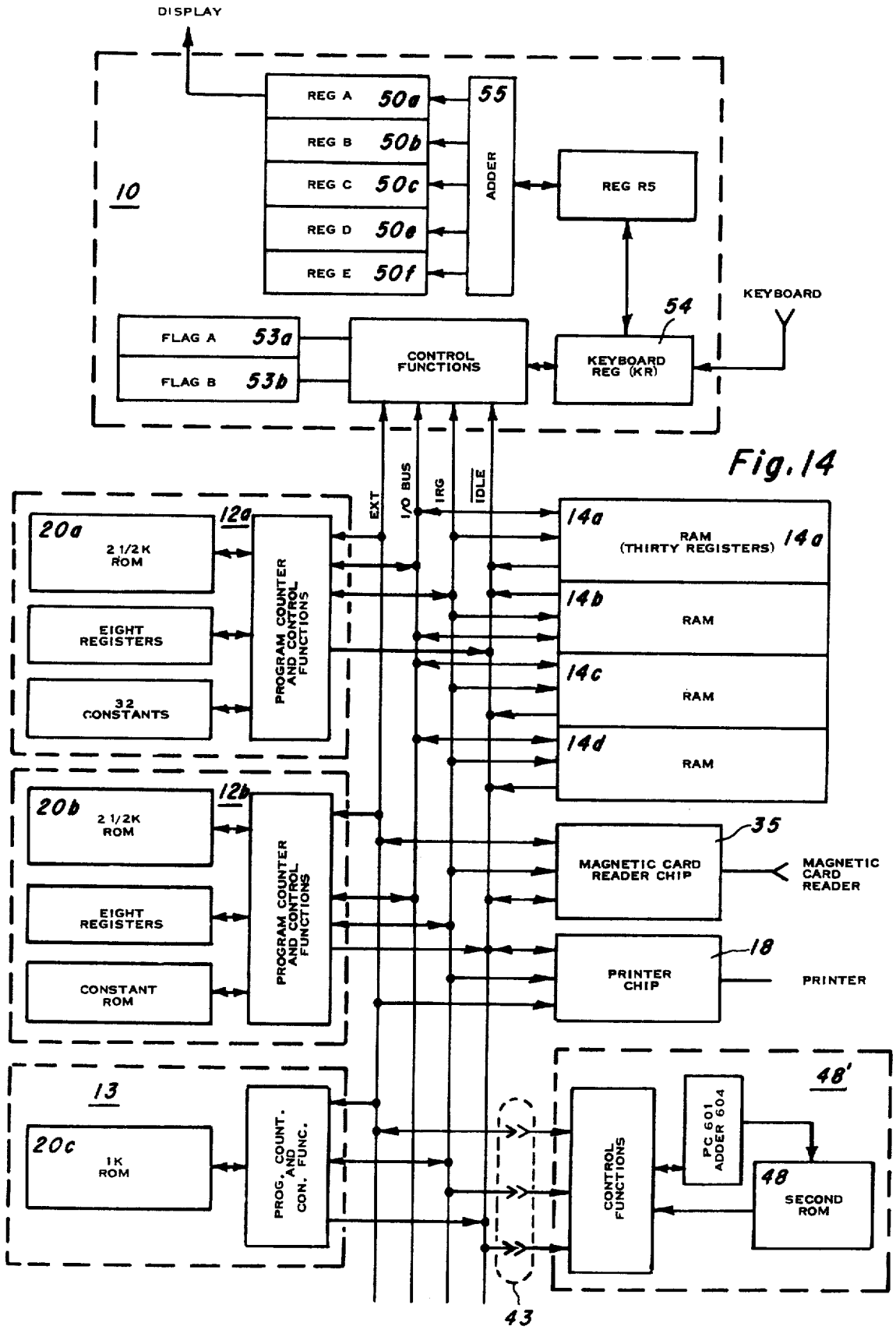

DISPLAY
10
REG A 50a
REG B 50b
REG C 50c
REG D 50e
REG E 50f
55
ADDER
REG R5
54
KEYBOARD
KEYBOARD REG (KR)
FLAG A 53a
FLAG B 53b
CONTROL FUNCTIONS
EXT
I/O BUS
IRG
IDLE
12a
20a 2 1/2K ROM
EIGHT REGISTERS
32 CONSTANTS
PROGRAM COUNTER AND CONTROL FUNCTIONS
14a RAM (THIRTY REGISTERS) 14a
14b RAM
14c RAM
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

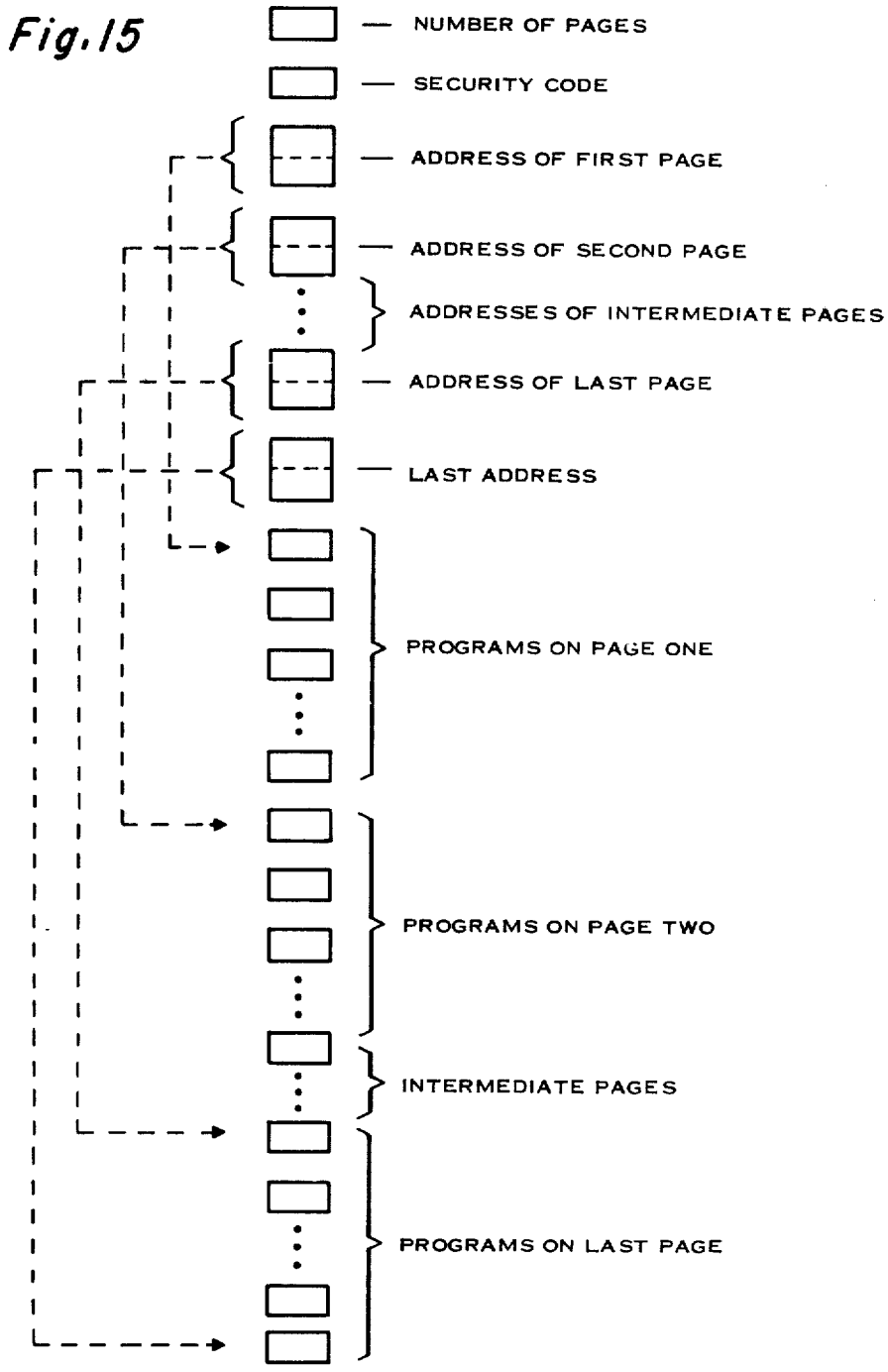
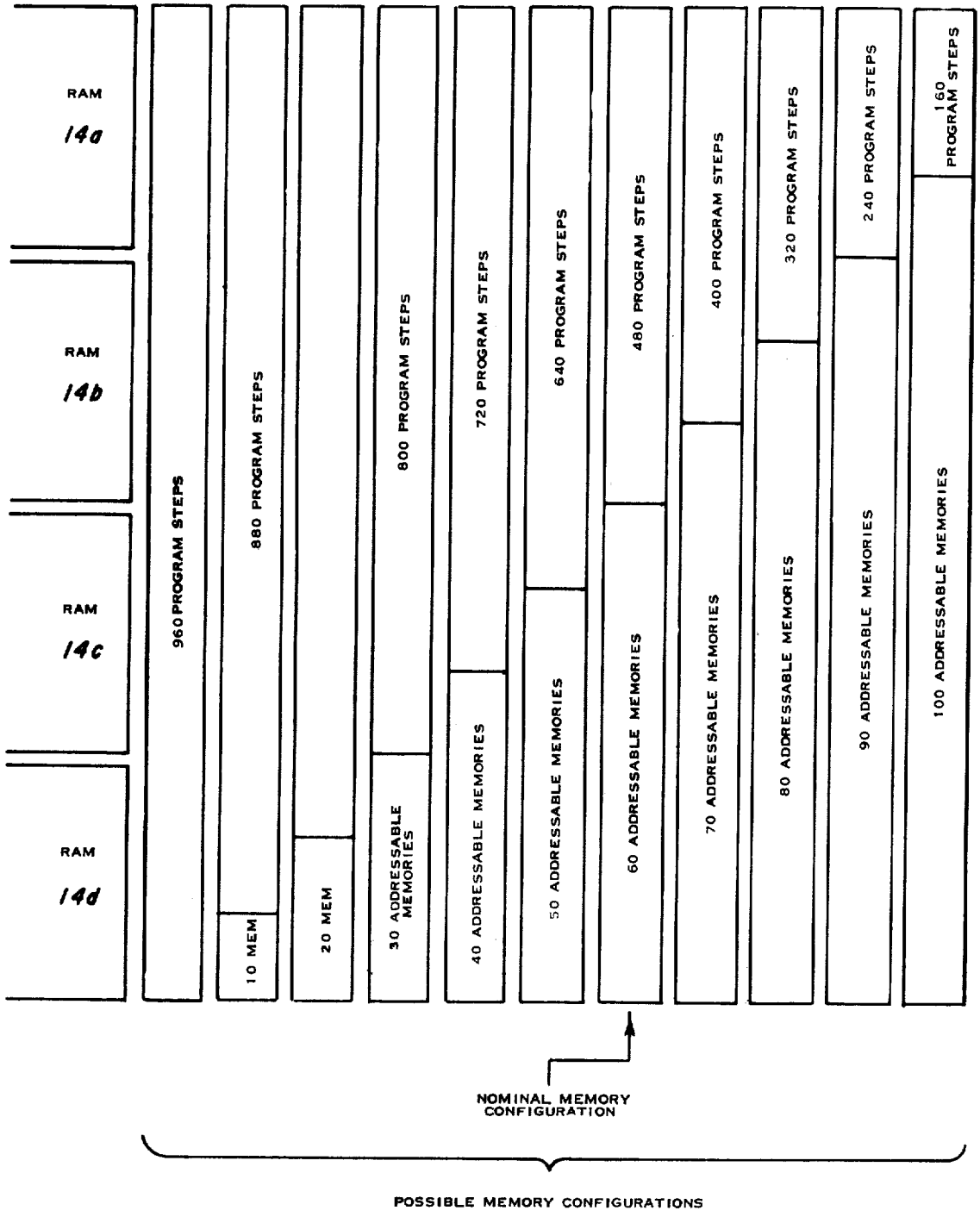

NUMBER OF PAGES
SECURITY CODE
ADDRESS OF FIRST PAGE
ADDRESS OF SECOND PAGE
ADDRESSES OF INTERMEDIATE PAGES
ADDRESS OF LAST PAGE
LAST ADDRESS
PROGRAMS ON PAGE ONE
PROGRAMS ON PAGE TWO
INTERMEDIATE PAGES
PROGRAMS ON LAST PAGE

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

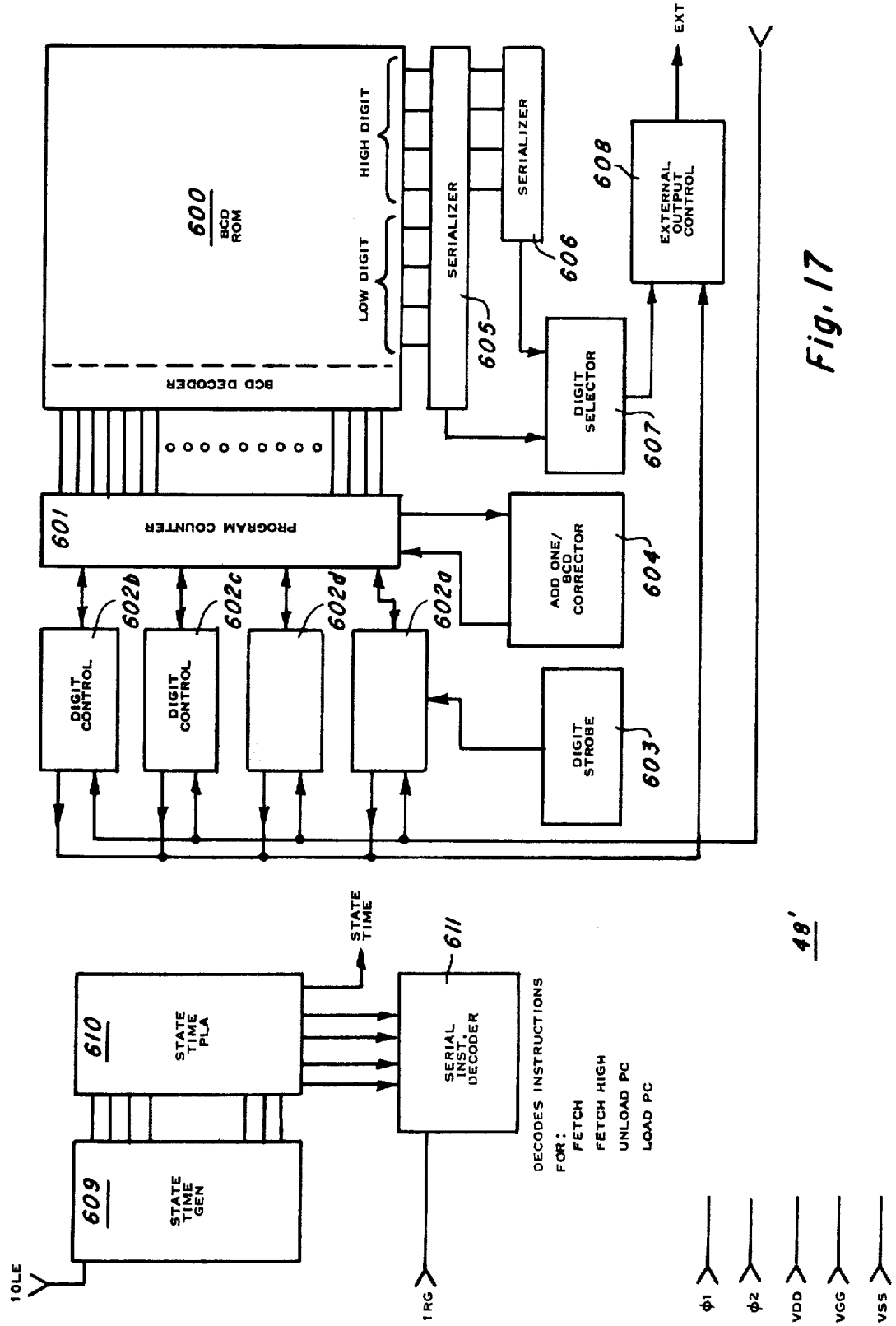

600
BCD ROM
BCD DECODER
HIGH DIGIT
LOW DIGIT
SERIALIZER
SERIALIZER
605
606
608
EXTERNAL OUTPUT CONTROL
EXT
DIGIT SELECTOR
607
601
PROGRAM COUNTER
ADD ONE/ BCD CORRECTOR
604
602b
602c
602d
602a
DIGIT CONTROL
DIGIT CONTROL
DIGIT STROBE
603
48'
STATE TIME
611
610
STATE TIME PLA
SERIAL INST. DECODER
609
STATE TIME GEN
DECODES INSTRUCTIONS FOR:
FETCH
FETCH HIGH
UNLOAD PC
LOAD PC
10LE
1RG
φ1
φ2
VDD
VGG
VSS

Fig. 17

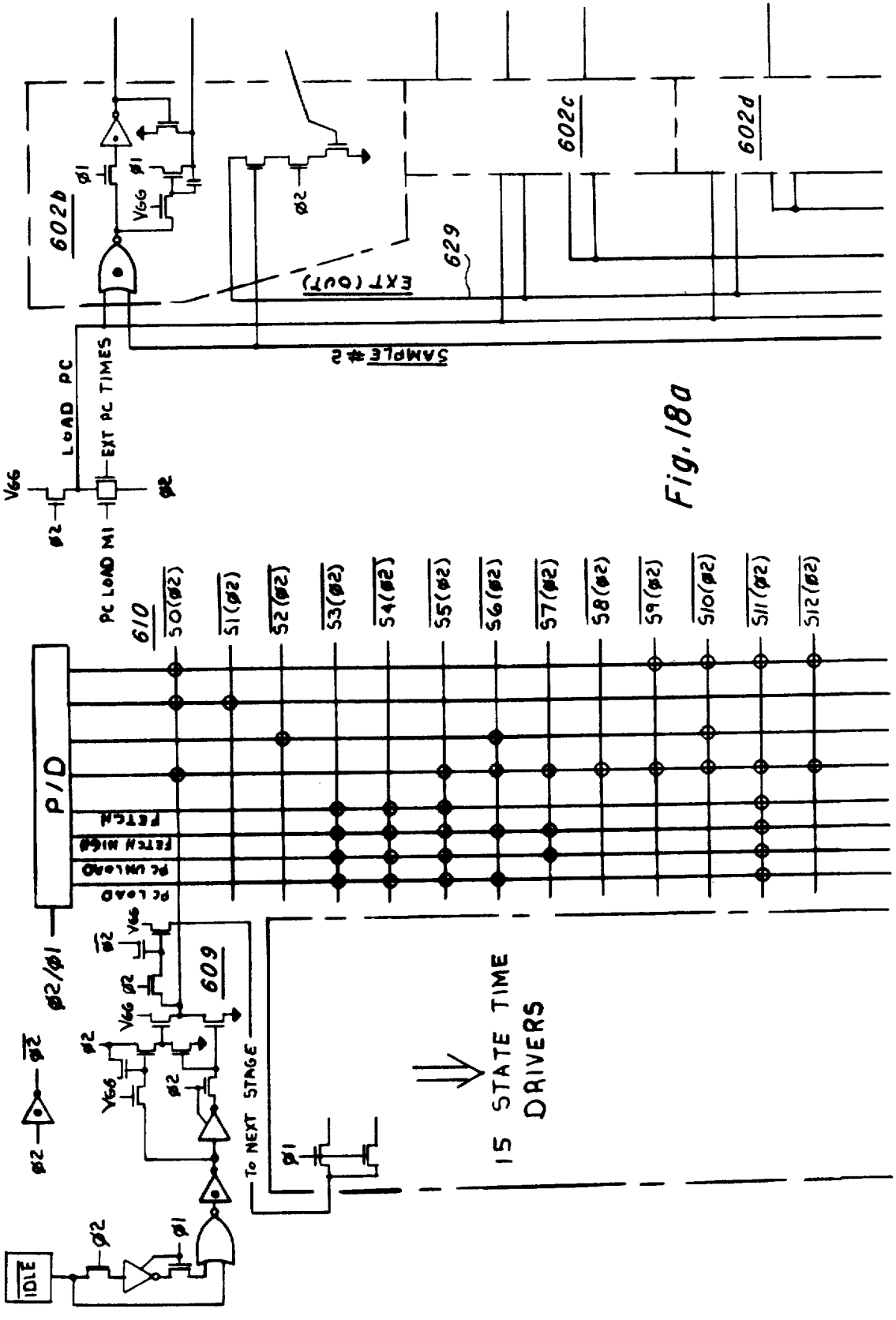

602b
602c
602d
629
EXT (OUT)
SAMPLE #2
LOAD PC
EXT PC TIMES
PC LOAD M1
610
P/D
PC LOAD
PC UNLOAD
FETCH HIGH
FETCH
609
TO NEXT STAGE
15 STATE TIME DRIVERS
IDLE

Fig. 18a

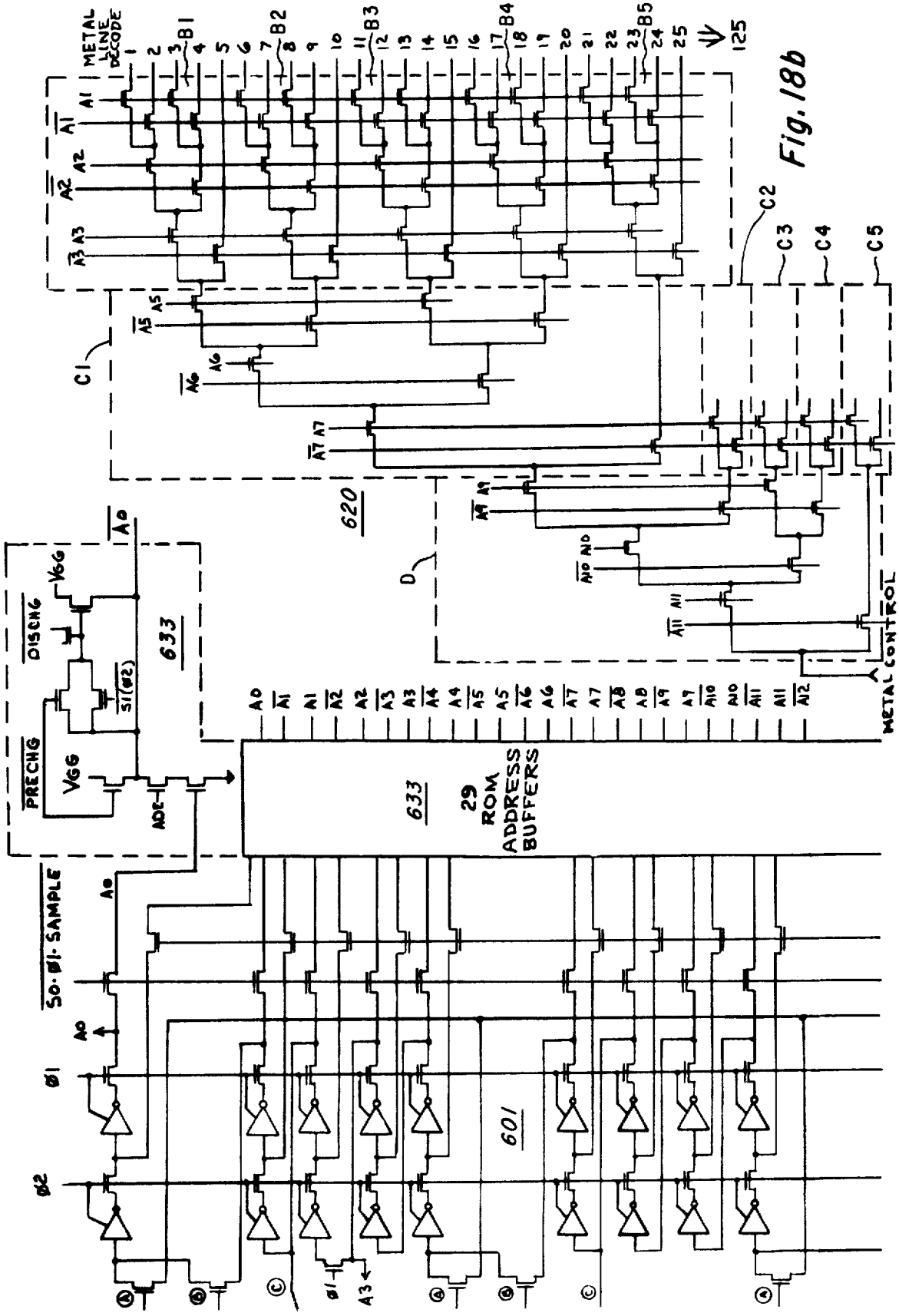

METAL LINE DECODE
B1
B2
B3
B4
B5
125
C1
C2
C3
C4
C5
620
D
633
DISCHG
PRECHG
VGG
ADE
29 ROM ADDRESS BUFFERS
METAL CONTROL
50-Ø1-SAMPLE
Ø1
Ø2
601
A3

Fig. 18b

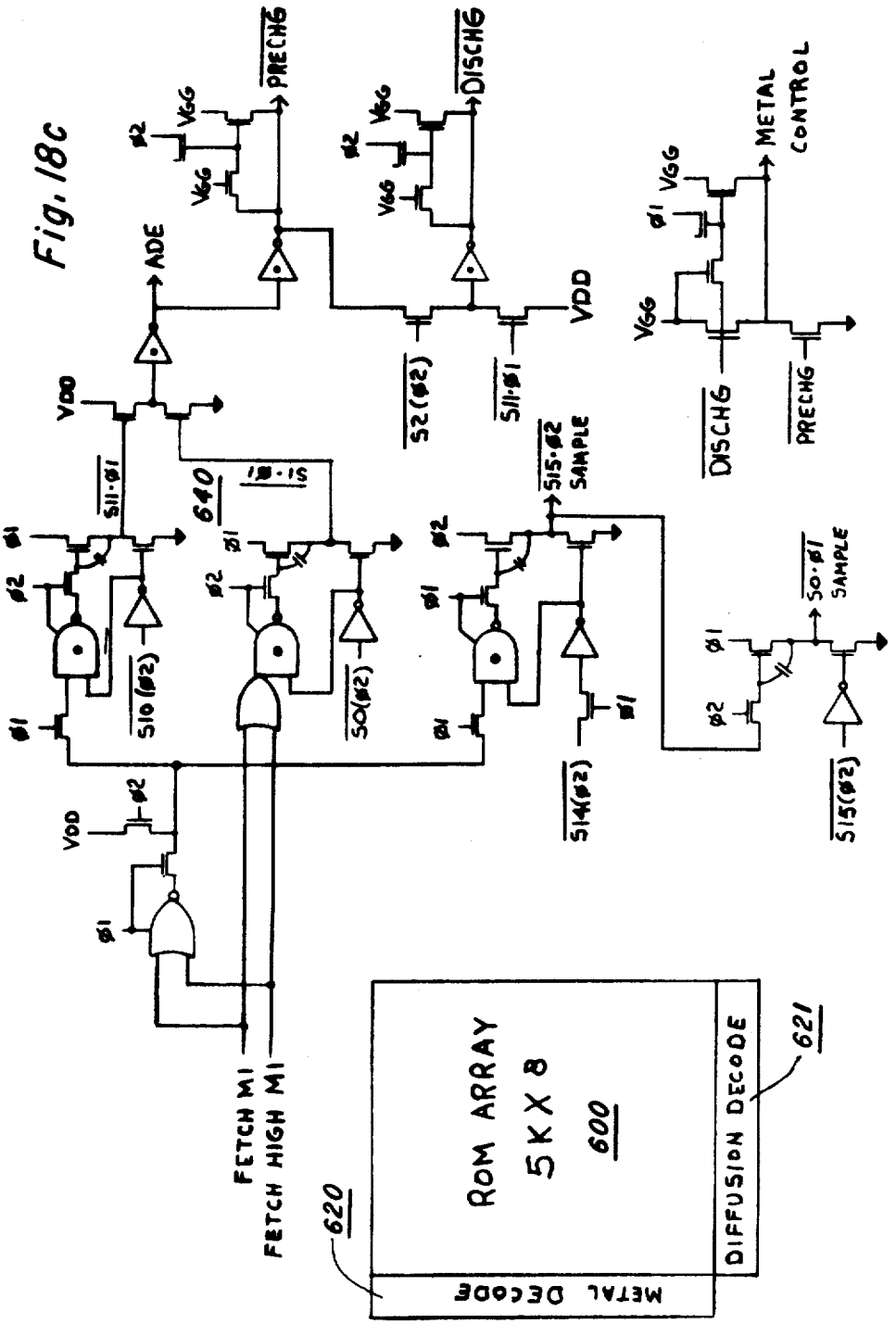


Fig. 18c
ROM ARRAY
5K X 8
600
METAL DECODE
620
DIFFUSION DECODE
621
640
ADE
VDD
VGG
PRECHG
DISCHG
METAL CONTROL
FETCH M1
FETCH HIGH M1
S11·φ1
S10(φ2)
S0(φ2)
S2(φ2)
S11·φ1
S15·φ2 SAMPLE
S14(φ2)
S0·φ1 SAMPLE
S15(φ2)

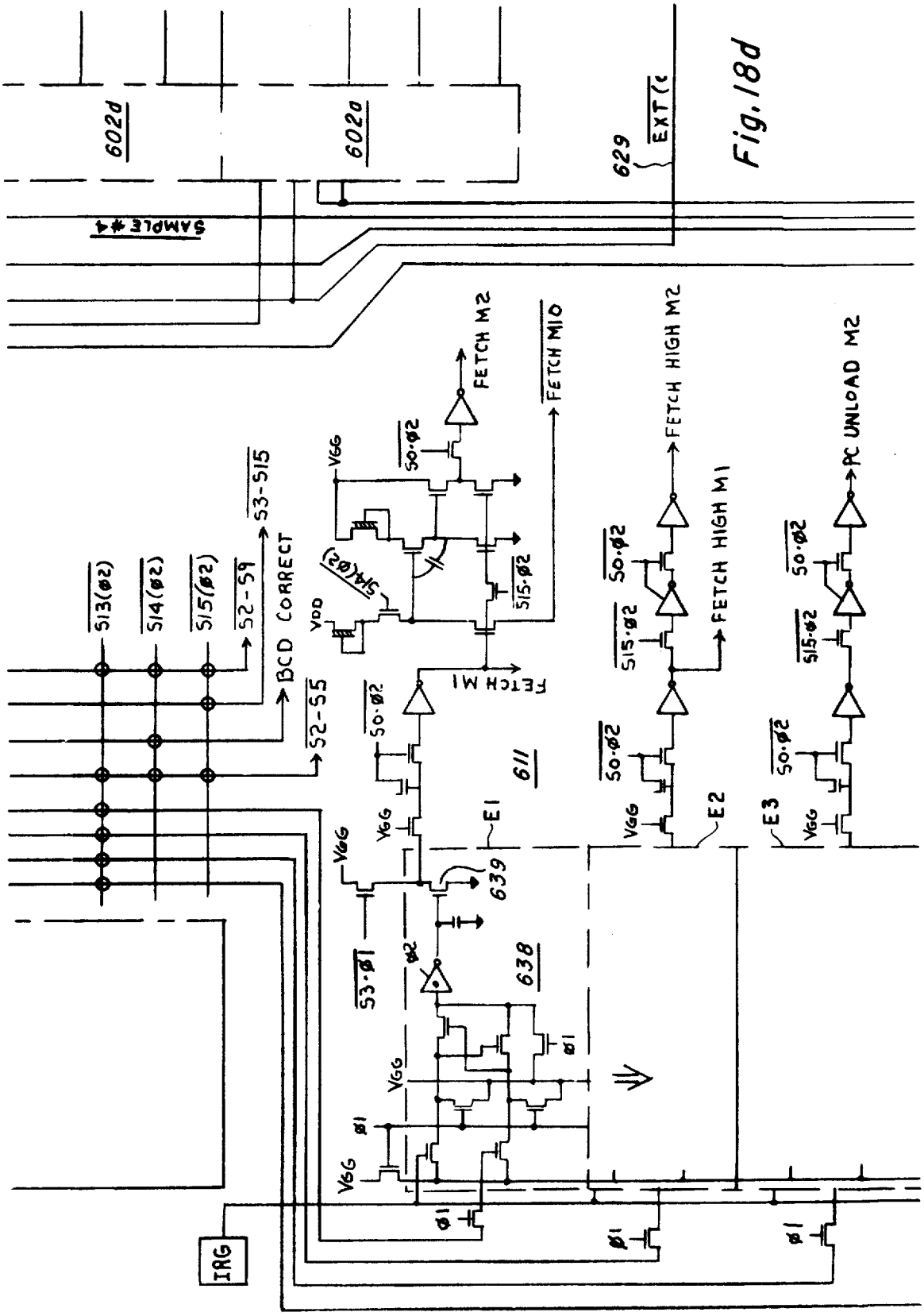
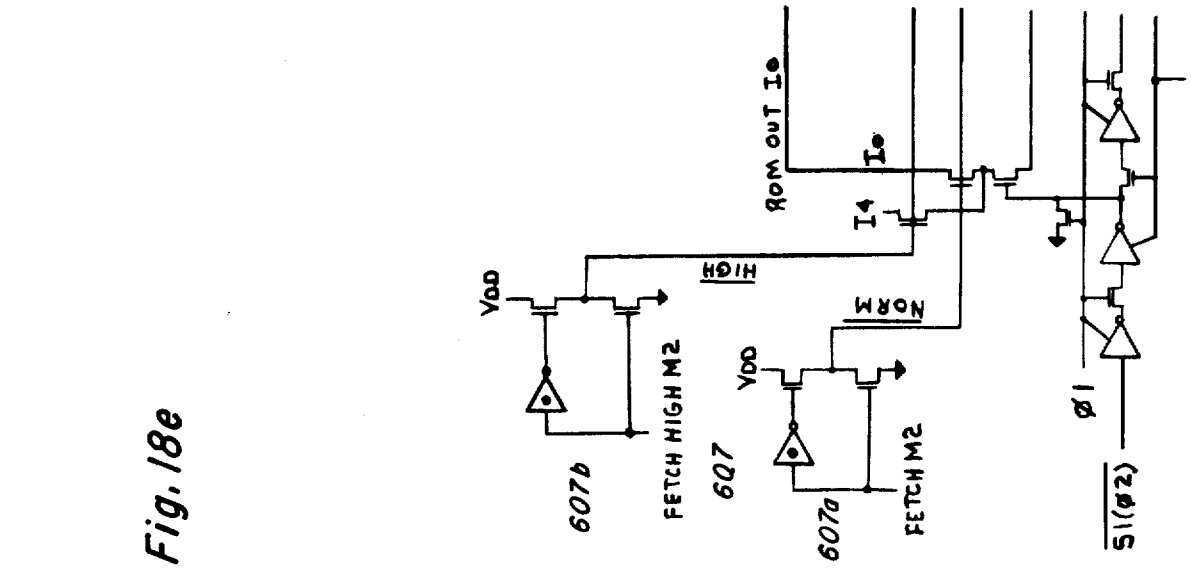
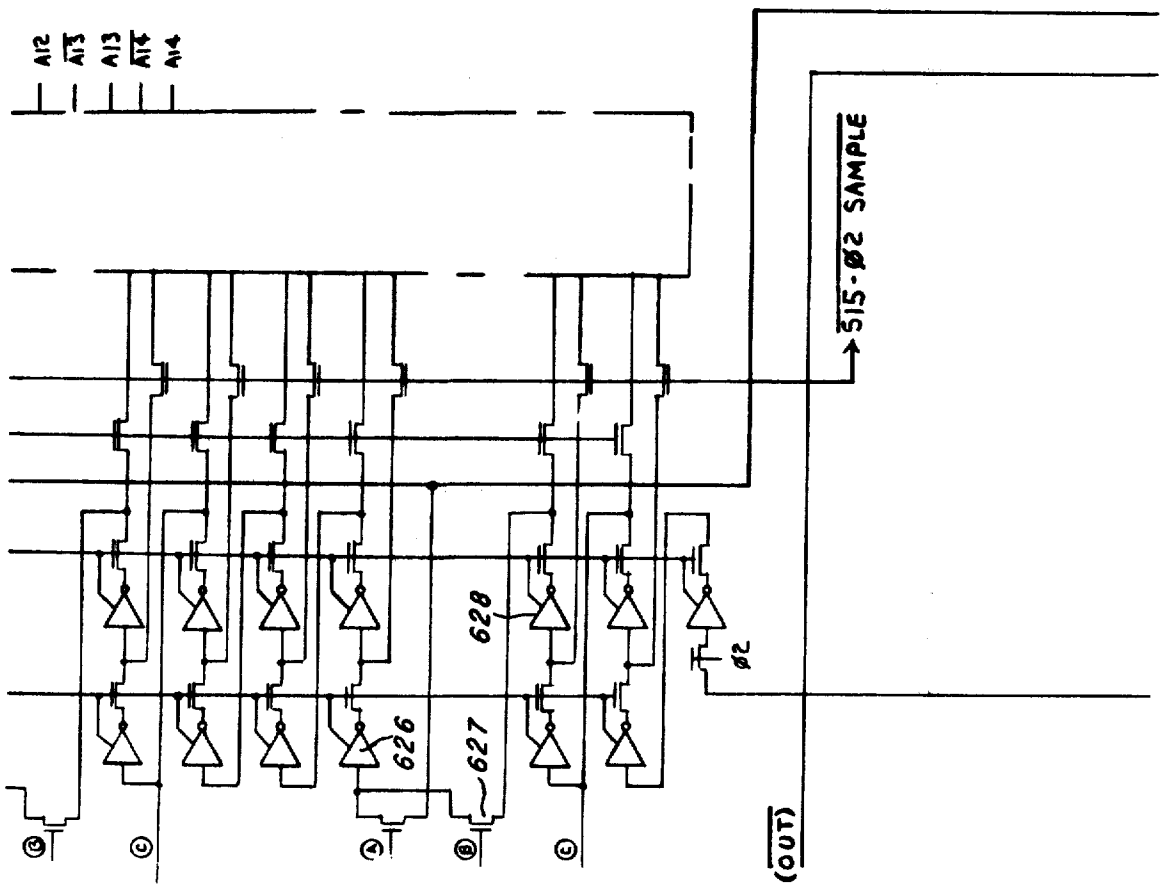

602d
602a
629
EXT(c
SAMPLE #4
FETCH M2
FETCH M10
FETCH HIGH M2
FETCH HIGH M1
PC UNLOAD M2
FETCH M1
BCD CORRECT
S3-S15
S2-S9
S2-S5
S13(ø2)
S14(ø2)
S15(ø2)
S0·ø2
S15·ø2
S3·ø1
VDD
VGG
E1
E2
E3
611
638
639
ø1
ø2
IRG

Fig. 18d

Fig. 18e


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


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

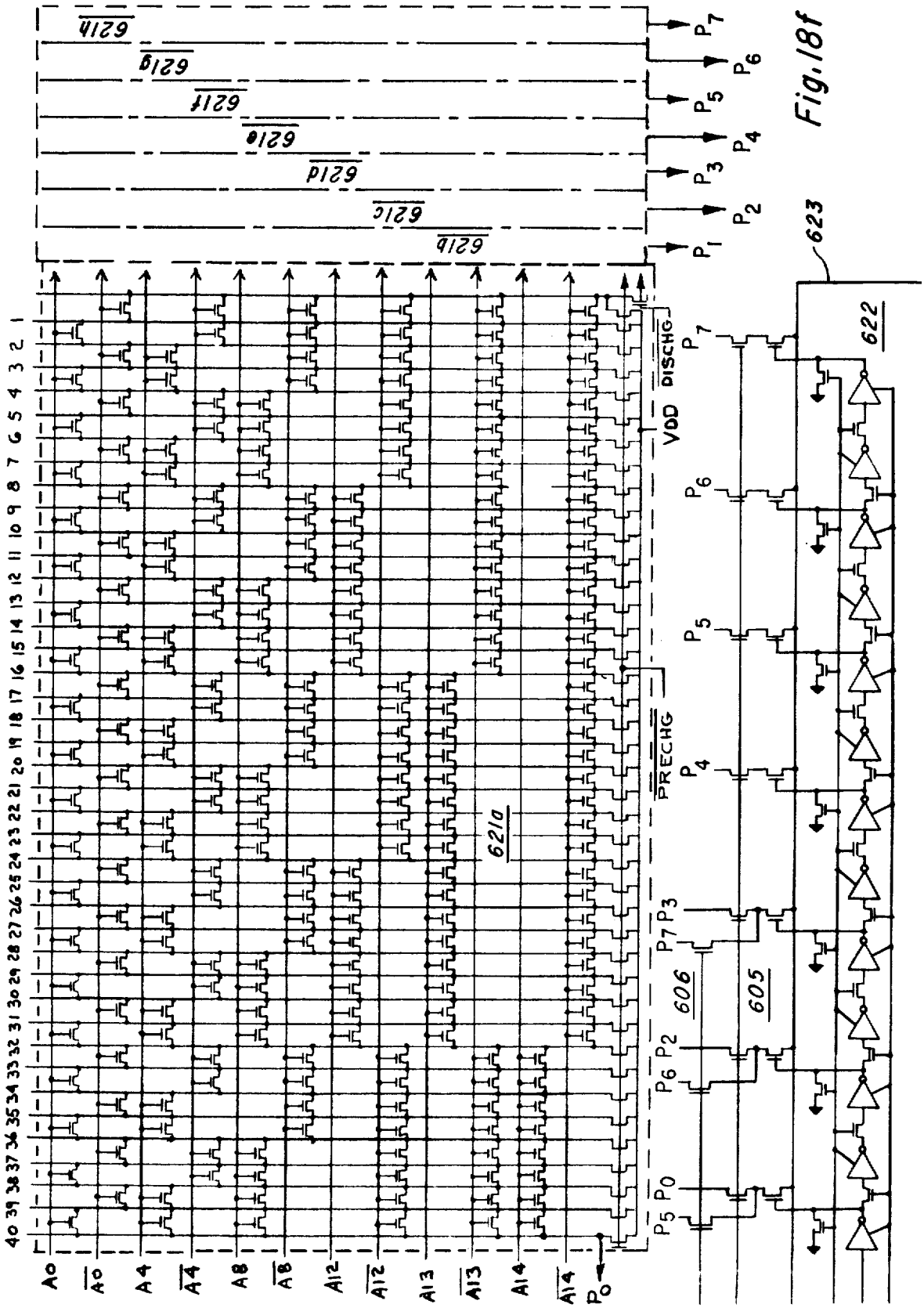

621h
621g
621f
621e
621d
621c
621b
P7
P6
P5
P4
P3
P2
P1
623
622
VDD
DISCHG
PRECHG
621a
606
605
A0
A4
A8
A12
A13
A14
P0

Fig. 18f

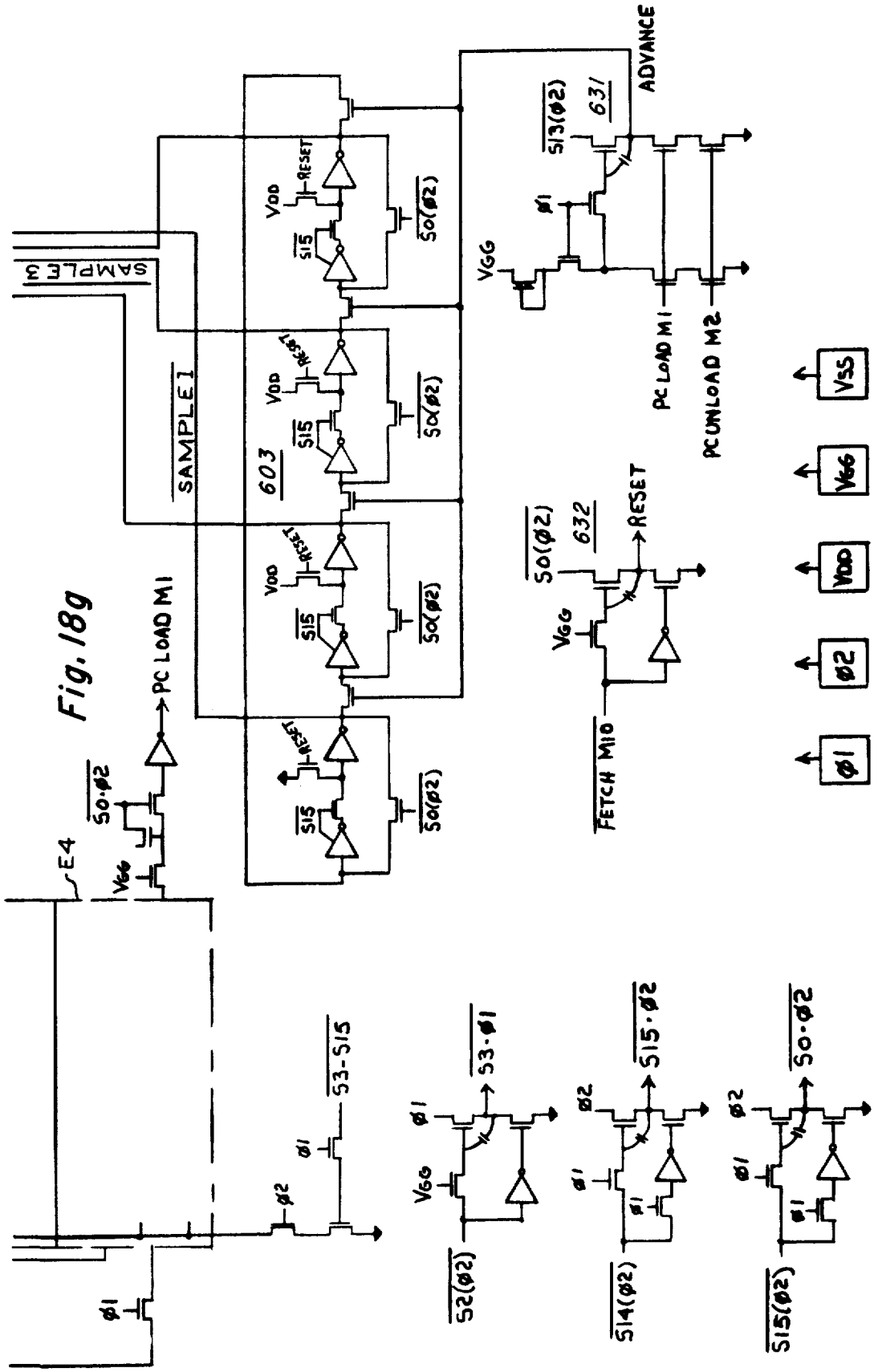

SAMPLE 3
SAMPLE 1
603
631
ADVANCE
PC LOAD M1
PC UNLOAD M2
632
RESET
FETCH M10
PC LOAD M1
E4
VDD
VGG
VSS
RESET
S15
S0(Ø2)
S13(Ø2)
S0·Ø2
S3-S15
S3-Ø1
S15·Ø2
S2(Ø2)
S14(Ø2)
S15(Ø2)
Ø1
Ø2

Fig. 18g

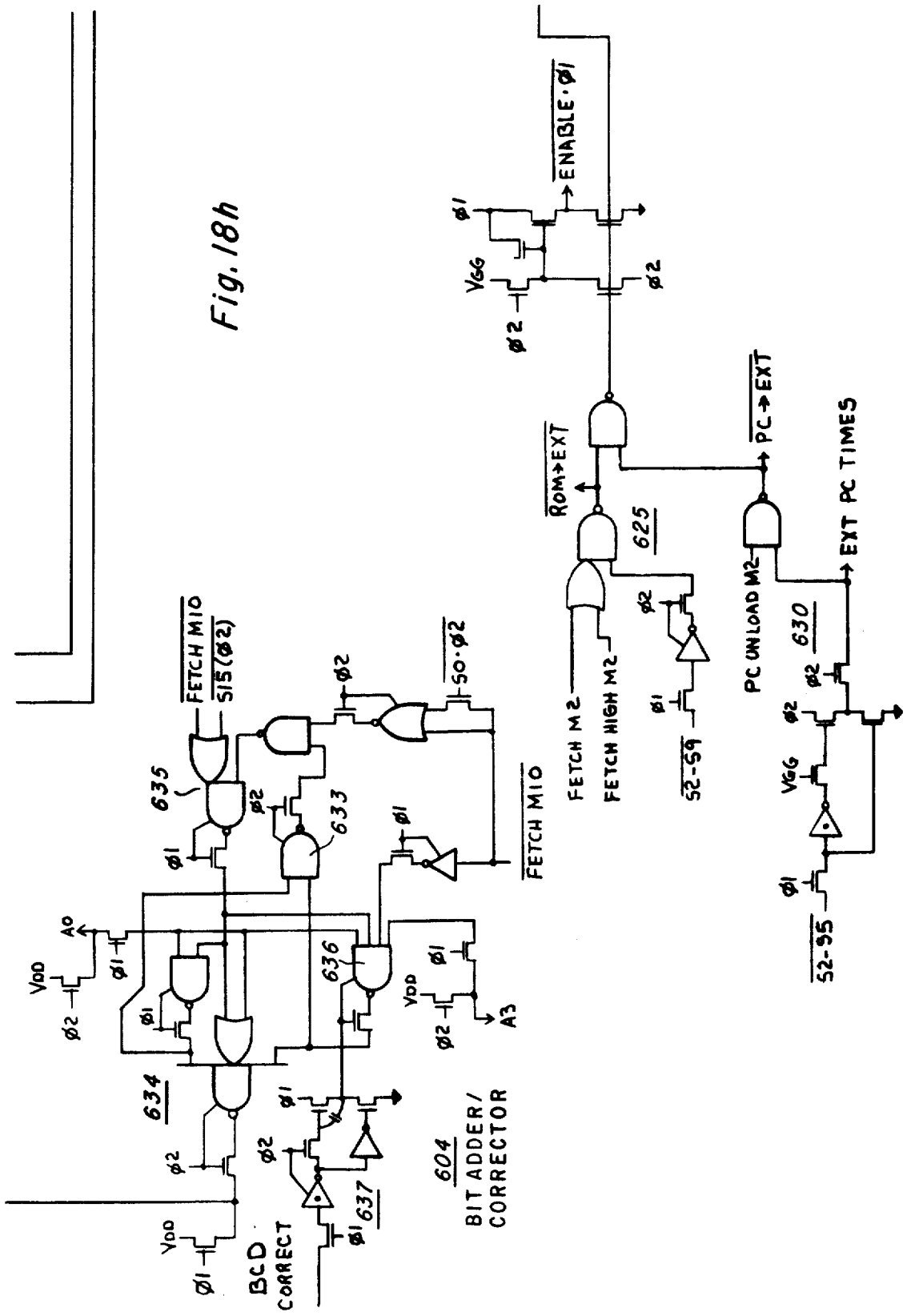


Fig. 18h
ENABLE·Ø1
Ø1
VGG
Ø2
ROM→EXT
PC→EXT
625
FETCH M2
FETCH HIGH M2
Ø2
Ø1
52-54
PC UNLOAD M2
EXT PC TIMES
630
52-55
FETCH M10
515(Ø2)
50·Ø2
635
633
FETCH M10
A0
VDD
636
A3
634
637
604
BIT ADDER/
CORRECTOR
BCD
CORRECT

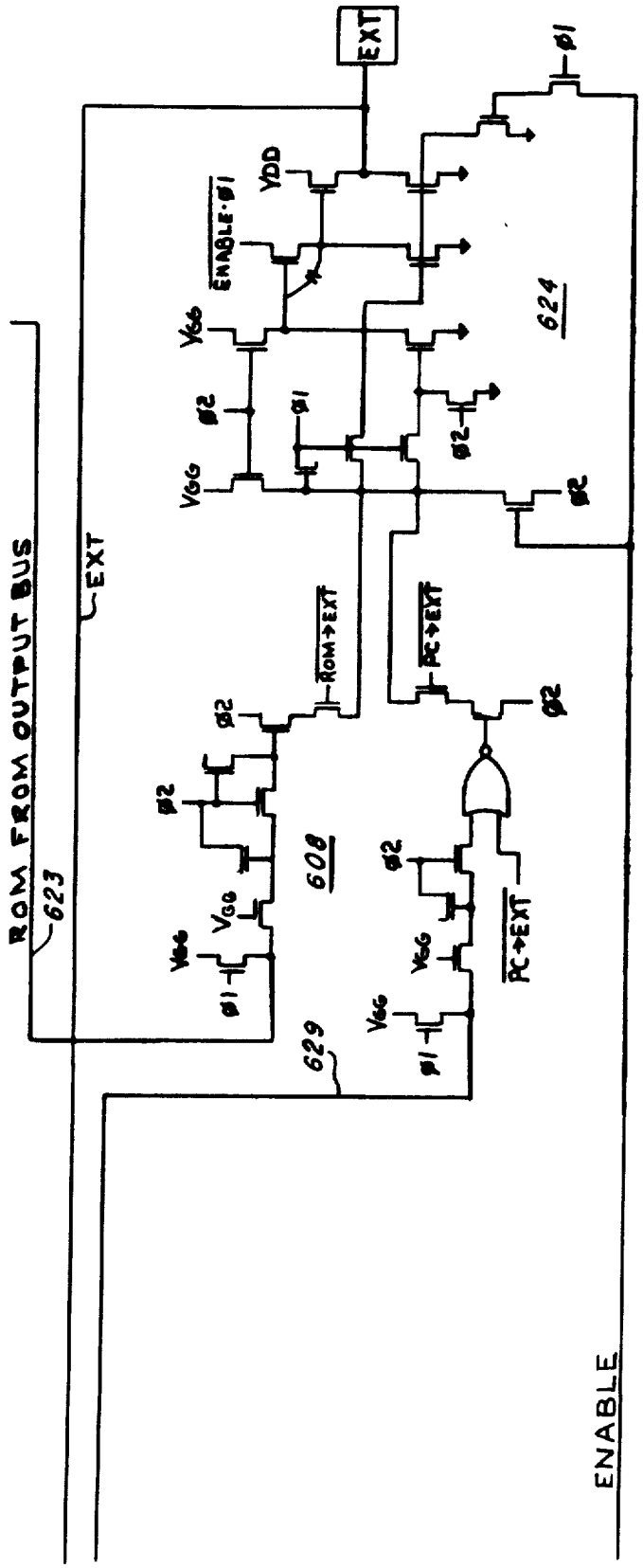
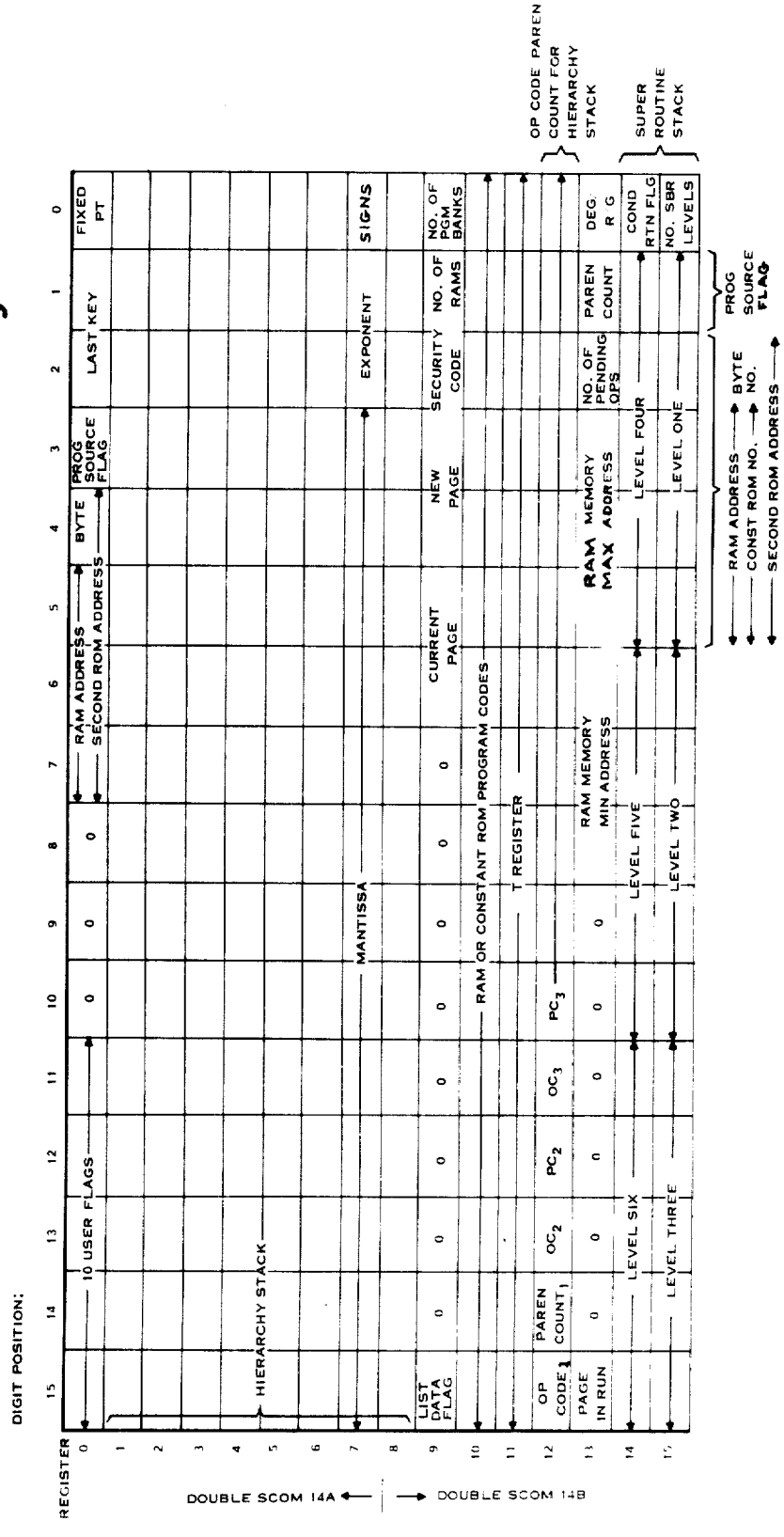

EXT
φ1
YDD
ENABLE·φ1
VGG
624
φ2
φ1
φ2
VGG
φ2
EXT
ROM FROM OUTPUT BUS
623
ROM→EXT
PC→EXT
φ2
φ2
φ2
608
VGG
VGG
φ1
φ2
VGG
PC→EXT
VGG
φ1
629
ENABLE

Fig. 18i

Fig. 19


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

MICROPROCESSOR SYSTEM HAVING HIGH ORDER CAPABILITY

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 accomodate 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. **3***a*–**3***b* 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. **5***a*–**5***e* depict in representative form the instruction words decoded by the arithmetic and SCOM chips;

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

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

FIG. **5***h* 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. **6***a*–**6***b* 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. **8***a*–**8***d* are a composite schematic diagram of the arithmetic chip of FIG. **2**;

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

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

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. **18***a*–**18***i* 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**, **5***h*, **7**, and **11**-**19** accompany this patent. FIGS. **2**, **3***a*–**3***b*, **4**, **5***a*–**5***g*, **6***a*–**6***b*, **8***a*–**8***d*, **9***a*–**9***e* and **10***a*–**10***r* are hereby incorporated by reference from U.S. Pat. No. 3,900,722, entitled "Multi-Chip Calculator System Having Cycle and Subcycle Timing Generators", 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 display 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 calculational 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 aforementioned, 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 aformentioned 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 desccribed hereafter in detail, to implement a calculator having high order capability.

The calculator's arithmetic chip **10** has a plurality of Registers **50***a–50e* 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–20c* is listed in Tables II*a–*II*c*, respectively. ROM **20***c* is a 1 K×13 bit ROM while ROMs **20***a–20b* are each 2.5 K×13 bit ROMs.

Referring briefly to Tables II*a–*II*c*, 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 II*a*, 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–5h*.

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–20c* 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 operation 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 100*a* 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 area for such operation code functions is located in the constant ROM areas of double SCOM chips **12***a* and **12***b*. Approximately half of the constant ROM in double SCOM chip **12***a* 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 **12***b* 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 **14***a*–**14***d*.

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 **14***a*–**14***d*. RAMs **14***a*–**14***d* may also be used for storage of numeric data, i.e, the results of the calculations performed by this electronic calculator. Normally, RAMs **14***a*–**14***d* provide a storage for storing 480 program codes while RAMs **14***c* and **14***d* provide 60 addressable memory locations for storing numeric data. By depressing 2ND, OP, 1, 6, the operator may determine which data configuration RAMs **14***a*–**14***d* 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 **14***a*–**14***d* for program steps, while the number to the right is the maximum address in RAMs **14***a*–**14***d* 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 **14***a*–**14***d* and the resulting configuration is displayed in the aforementioned manner. For example, inputting 7, 2ND, OP, 1, 7 results in RAMs **14***a*–**14***d* being repartitioned with seventy memory registers and 400 program step locations; also 399.69 would appear in the display.

RAMs 14*a*–**14***d* 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 **14***b* (see FIG. **19**). Digits 8–6 holds the actual address in RAMs **14***a*–**14***d* 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 **14***a*–**14***d*. 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 **14***a*–**14***d*, 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-100*a* disk unit manufactured by Texas Instruments Incorporated of Dallas, Tex. may be used. The twenty character position PC-100*a* 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 **14***a*–**14***d* 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 **602***a*–**602***d* is enabled; digit strobe **603** sequentially ena-

bles digit controls **602***a*–**602***d* 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 **602***a*–**602***d* 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 **602***a*–**602***d* 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 **12***a* and **12***b*, 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 **602***a*–**602***d* in response to an UNLOAD PC instruction, the output from external output control being provided to the EXTERNAL bus.

Considering FIGS. **3***a* and **3***b* with FIG. **17**, the following discussion examines the flow of data between second ROM chip **48'**, arithmetic chip **10** and the main ROMs **20***a*–**20***c*. When a program is to be called from second ROM chip **48'** according to the aformentioned 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 **50***a*–**50***e*, 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 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 **50***a*–**50***e* using an R5→Adder instruction (FIG. **5***b*). This high digit may then be shifted in its register to the next more significant digit position by register shift instructions (FIG. **5***b*). 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 **50***a*–**50***e*. 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 **50***a*–**50***e*. 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 pre-existing 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 **50***a*–**50***e*. 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 **20***a*–**20***c* when a PREG instruction is generated telling main ROMs **20***a*–**20***c* 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 aforementiond 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 2ND, 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 aforementioned). 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 2ND, 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_o) 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 $S2\phi2$ 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 $S2\phi2$ 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) instuctions, 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 1½ 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 $\overline{\text{IDLE}}$ 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 **20***a*–**20***c* 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**, **607***a*, **607***b*, add one/BCD corrector **604**, logic **632** and the like are sensitized to delayed FETCH instructions FETCH M2 or FETCH $\overline{\text{M10}}$.

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	0A01	0A0A	00B0	03DC	01DA	0A4B	1002	0121	0A0F	0101	032C	1B09	0A02	01D7	1FCF	0A77	0085
0011	TMC582	1A60	1CAC	1FD0	1FCA	1FFC	1D2C	1FA4	1CF2	1FE2	0A87	1A20	0AC7	1A1A	0A07	1A1A	0A77	1ADE
0022	TMC582	1C8A	1CAE	1A15	19D6	147F	1A82	1A86	1A8A	0A57	0095	1A2A	0A47	19C4	0A0C	0AC7	1A9E	1B82
0033	TMC582	1827	1831	19F0	1A5C	1A60	1A64	1D8E	0CD2	0213	0007	008A	1F34	1A3A	00B1	1AE0	1EEE	1B1A
0044	TMC582	1B6A	1A36	1A3A	1A3E	1A42	00F0	1AB2	0023	002A	005A	1520	1DE8	0A07	197C	1E9A	1B4D	19D0
0055	TMC582	0011	19CA	0011	1B90	0A97	1A5E	0111	0A0F	0121	01D5	1BAC	0A27	1AA3	1C0E	1F1A	1C1E	1C1A
0066	TMC582	1C40	1F18	180C	1FFC	0A47	1A97	090A	1984	0AD7	1944	0AE7	141C	1AEA	1F0E	1C20	1C0E	1B0D
0077	TMC582	1A0E	1BF0	0A37	1CA6	0A47	1A30	0022	002A	1B6C	0AE7	1A1C	1A32	1A5C	1A85	1A37	1AC6	1FD4
0088	TMC582	19D4	1FFC	0A07	1AF7	0CD2	0213	0007	008A	1F7A	167A	1FFF	1A3A	1AED	1A2B	1AEE	19D6	1F70
0099	TMC582	0041	1B36	0A87	0080	1AE4	0A87	1AE0	0A27	193E	1DFA	1AC7	1E94	1A37	1C12	1FF4	1FF6	0A10
00AA	TMC582	0AFA	1B14	00A9	0051	0A77	1E7A	00C1	0080	0A07	12A4	0A37	1F6C	1FCC	1A74	1FFA	1FC0	1EC4
00BB	TMC582	1FF4	1AB4	1FBA	1FBC	1FF0	1AB0	1F60	1E84	1934	1D2A	1934	1EAC	1A99	1E0A	192C	1E44	1990
00CC	TMC582	193A	1F9E	1A46	1FDA	1F9A	1A89	1E94	18F6	1F90	1A32	1F4A	1E8A	1FEA	1AFB	1941	1FFE	1F80
00DD	TMC582	1FDE	0051	1AF4	1E7A	1FD6	1AF6	1AAF	1FFC	1F6F	1FCC	1ADC	1FDA	1F66	1AFD	1E62	1FC0	1F8A
00EE	TMC582	1F5C	1FAF	1E5A	1A6A	1F70	1E52	19FA	18C0	1F6A	1DF6	1A36	1F40	1F44	1FA2	1FD0	1987	1FCE
00FF	TMC582	1AE7	1FCA	1E36	1F94	1FFA	1BA2	1CD6	1008	1A2A	1F4E	1A14	1F4A	194C	1E20	1AD0	1F42	1AF4
0110	TMC582	0AB7	0058	12D6	00B0	0052	16FC	005A	1B06	005A	0049	00F9	0A97	02F3	0289	1F4A	0202	0ED3
0121	TMC582	005A	0300	0409	1007	0202	095A	0989	1C36	00A8	1368	00F1	1B64	04D7	02F3	01D5	0111	0A1F
0132	TMC582	01D4	004A	0724	06A3	0619	11A9	0099	1987	0A67	1B0A	0AA7	00B0	1B04	0011	0091	0045	0065
0143	TMC582	0A1A	0015	0100	0DD3	0300	0101	0A1F	06D3	0060	0658	0709	109A	0099	070A	0121	0A1F	01D0
0154	TMC582	014A	02F3	014A	0121	0A0F	0101	0153	095A	187E	0427	18AC	0014	01D7	01D2	1A72	0A04	0A04
0165	TMC582	0A0A	0A0A	0A0A	0A0A	0A0A	0A0A	0A0A	000A	00DA	185A	0A0A	0095	1A0C	00A0	0AC7	1A12	005A
0176	TMC582	0052	00B0	1210	0A37	01F3	0153	0A37	10DA	0AA7	0095	0085	187E	02DP	0AE7	02F4	0131	0A1F
0187	TMC582	0CD3	0131	0A0F	0111	0A77	01F3	00A0	15D2	0AA7	06F3	0153	10A2	0AA7	09F3	00A0	1E72	0300
0198	TMC582	0101	0A1F	06D3	065A	0709	105A	04D0	0121	0A1F	01D0	015A	0289	0153	1048	0C4A	09F0	0980
01A9	TMC582	1046	065A	070A	0140	09D2	01D7	0121	0A1F	01D0	0148	0148	0121	0A0F	0101	015A	0911	02F0
01BA	TMC582	02F3	0A0D	0A0C	0085	00A5	02A0	0A1A	015A	0111	0A1F	01D3	000F	0014	0120	1A32	00D0	0980
01CB	TMC582	185A	065A	014A	0A87	02F3	0289	10FE	004A	0300	0A06	02F4	0101	0A0F	0121	04D4	0121	0A1F
01DC	TMC582	01D0	0140	0911	02F0	0140	0611	02F0	0121	0A0F	0101	0323	0111	0A1F	01D0	07D0	0640	0A80
01ED	TMC582	1834	01DE	0080	1522	000A	00D8	1582	01D7	0A07	186C	015A	0308	1A06	0140	0A03	0A07	02F3
01FF	TMC582	0111	0A1F	01D0	0A10	0919	100A	0A48	0709	18D7	0111	0A0F	0101	1D8E	0A27	0080	1A04	0A57
020F	TMC582	0085	00E5	00C5	199F	0D09	100C	180C	0020	1808	0D09	100A	0091	07DA	01D7	0201	0007	04D0
0220	TMC582	1824	0090	0092	1804	008A	01D7	0001	0A27	000B	00DB	1804	0A37	00C9	01E0	0AE0	0140	01D7
0231	TMC582	0072	00AA	1788	00BF	0051	0058	180A	005A	0626	003A	0120	0D09	1104	0709	1804	0032	030A
0242	TMC582	0409	181A	0303	1804	0300	0E09	1814	0940	0409	100B	0E09	18EA	0940	0030	1010	1806	0030
0253	TMC582	1804	070P	182F	0A0A	0033	0700	1837	00E0	1004	0099	0030	1808	0E00	070A	10C2	0E0B	10AE
0264	TMC582	0041	1881	0091	00B3	0C93	006E	00AA	1750	0049	0058	10E0	1AA9	0A67	19E5	0081	0A17	00AA
0275	TMC582	186A	0085	00E5	00D5	18D1	0AE7	1C81	0AA7	0049	0051	180D	03D3	0A47	06F3	1411	00B2	1C1A
0286	TMC582	0AE7	0058	1017	0051	1840	0033	1844	0417	18FB	000A	0049	041F	01D6	09DE	0111	0A0F	0131
0297	TMC582	0AE7	02F3	0111	0A0F	0A97	02F3	0111	0A1F	01D6	09DE	0111	0A0F	0131	00AB	00AA	10E3	00A0
02AB	TMC582	1804	0AAA	00B2	1883	0A07	0095	103D	02DF	1AEA	0080	0079	0091	18FF	00FA	007A	1905	0030
02B9	TMC582	113A	06A4	12BC	0D09	1950	114A	01DB	0022	0032	005B	04D0	1008	0104	002E	003E	00AA	00E2
02CA	TMC582	0A0C	00B5	0A0D	0050	00A0	14EA	005A	0015	0A0D	0A27	1880	0120	0A16	02F0	0103	0091	1880
02DB	TMC582	0640	065A	0A80	000A	17A4	00F8	007B	0030	00A2	01DB	00D6	001A	1008	0709	0A37	1014	0FD6
02EC	TMC582	0030	0A77	1B04	0A07	06F3	0A1B	1002	0A57	02F3	0283	0020	0A37	1804	0A67	09F3	0206	0A27
02FD	TMC582	02F0	0C76	0953	0300	0409	1806	0289	100D	0230	00AA	00AA	1004	0082	0989	15FE	0953	0976
030E	TMC582	1809	0100	0100	0100	0100	0100	0100	0100	0100	0100	0049	00F2	0085	00C8	1754	00AB	1582
031F	TMC582	005A	1B08	010A	0A02	005A	0AB5	103C	0A08	0EF3	0653	00F0	109E	095B	095B	0A57	0A47	02F3
0330	TMC582	0030	1006	0709	100A	095A	095B	0A87	02F3	0989	1814	0289	1810	00AB	1804	0300	0940	09DB
0341	TMC582	0C53	0980	000A	1532	1AC7	0011	00C3	0092	00C0	00B0	0093	1804	0011	00B0	1A0A	0EF3	002A
0352	TMC582	003A	0090	100A	0CD2	002A	003A	0010	1804	0023	0012	003C	194B	06BC	1015	0030	1A0A	0CD2
0363	TMC582	002A	003A	072C	1006	095A	1807	01DC	002C	11BC	0989	1017	0988	1954	0022	00B0	1F50	0021
0374	TMC582	1F4C	01DA	04AA	0A97	02F3	09F0	0300	0AFA	0101	0101	02DA	0AFA	0101	0CD3	0A27	02F0	0CFA
0385	TMC582	0AFA	0101	0AA7	01F4	030A	09F0	09A4	0140	0140	0140	0140	0C19	0923	0130	1A06	01A6	0990
0396	TMC582	0111	0A0F	0131	0AD7	02F0	0D2C	0101	0A0F	0121	0040	1804	0AAA	0AD7	01F0	0101	0A1F	01D3
03A7	TMC582	06DA	0101	0A0F	0111	01DC	0074	000A	000A	00B0	0A0A	0074	00D4	000A	00B4	0A0C	0A0D	0A0C
03B8	TMC582	0069	0058	009A	1188	0059	004A	1949	00E1	00C3	00C0	0092	0011	0133	000F	1A0A	0CD2	002A
03C9	TMC582	0034	0090	13E4	0010	1882	0011	0D09	1A7A	0989	0026	108A	0099	0022	0ED0	1902	00AB	00AA
03DA	TMC582	00A2	009A	000A	00D2	00A2	1628	0089	016B	0007	0042	04D4	00DB	005A	1365	147D	00A0	174A
03EB	TMC582	0A47	0095	1880	0073	1B09	0A47	180A	1ED6	000A	00DB	1817	0001	181B	00AA	1608	00D9	0A97
03FC	TMC582	02F3	0111	0A1F	01D3													
0400	TMC582	0219	1839	0659	0A19	103F	00D8	185D	004E	00A2	1FB0	1FC0	0A57	1831	0033	0114	0903	002C
0411	TMC582	0022	1004	0021	003C	181E	06A0	1820	0041	00CB	13BD	0A0D	0A04	0A0D	0040	1FD3	00A2	188F
0422	TMC582	0033	06D2	0A89	1007	06BA	09DB	06DC	0D29	1016	0010	1836	06DB	1804	0000	0E9C	1805	0C9A
0433	TMC582	1F12	0E09	0012	183A	0C5A	0940	0D09	1815	0300	1819	096C	0D29	104D	0CD2	0F0A	1808	0953
0444	TMC582	0300	1804	09AA	052C	1017	09B0	1807	09AA	095B	0E00	0308	1A11	0085	0051	01DA	0A47	0A1A
0455	TMC582	0022	0CC0	005A	1F7A	0A80	0A09	13CF	0A01	00B1	0A0C	0A87	00BE	1F4A	00B0	10CA	0A47	02F3
0466	TMC582	0A87	06F3	0153	0A27	09F3	0153	01D7	0150	01D7	0AD7	1909	0080	1046	0A27	01F3	0A27	06F3
0477	TMC582	0153	05D3	1A1C	01D7	02DE	0500	1006	02F0	0306	01D7	02DB	0A1F	0CDA	0A0F	0131	1812	0A10
0488	TMC582	0AFA	179C	1AC1	00FA	00F2	005A	1A97	0051	1CA1	0081	0A17	0095	1D09	0A37	02F3	0A17	1A62
0499	TMC582	1F8A	0051	00B0	0133	000F	110B	1A85	1FFF	04DA	1B04	03D6	0AD7	02F3	0111	0A1F	0CD6	1C11
04AA	TMC582	1FAA	1CD4	00A0	1923	00A2	1929	0092	1FAA	1FDA	0A47	1C7F	0051	00A6	02F0	0103	01D3	01D7
04BB	TMC582	032C	0121	0A1F	0190	0007	04D0	0050	1B47	0121	0A0F	0111	1D15	02DB	0A57	0AE3	1AC5	0A41
04CC	TMC582	01D7	0A17	1D45	004A	0099	04D4	1B87	0051	1B6A	0A06	02F0	01D7	0104	000F	01D7	0201	0007
04DD	TMC582	04D0	005A	1CFA	0ED3	1C7C	14F1	1A49	1FF0	1FF0	0A0C	00C5	0619	1B06	0940	0C40	0A98	0A0A
04EE	TMC582	1005	0213	0A10	0A09	0A04	0A05	1A05	0A01	0619	1B06	0C48	0948	0050	15BE	00EA	1746	1D2A
04FF	TMC582	1FCA	19C9	1FA4	1B79	1E3A	1AAA	1FF0	1990	19D7	0051	1CDA	19DD	1CCA	1C22	1D60	18F2	1F61
0510	TMC582	19E9	1F1A	19E0	0051	1C2C	19F3	0051	1B0A	19F9	19DA	1A09	19FF	1C1C	1FDA	1A05	1FE2	19D0
0521	TMC582	1AFA	1F72	1A0F	1954	1DD0	1949	1949	1ACA	1FD4	1DC6	1983	1983	1AD3	1F9A	1B0A	1BC1	1DDA
0532	TMC582	0051	1AE1	1A31	1DA3	1FCA	1911	1FFA	1FF2	1A3D	18E1	1FDF	1A43	1FEC	0A27	00A5	1F63	1927
0543	TMC582	1F94	1953	1FFA	0020	1F01	0FDC	17D5	0303	1F91	00A0	00AA	100C	0A1F	09D3	06F3	0619	169C
0554	TMC582	094A	094A	0AA7	02F0	0103	0AA7	02F3	00F2	0958	095B	0A1F	02D6	0050	1004	02DE	00FB	007A
0565	TMC582	1A3C	0110	007A	1B2A	0A37	06F4	0A8B	1A03	06B0	1002	0709	06D2	1016	0030	1B0C	0A8B	06D2
0576	TMC582	0A80	0AAA	1B0A	0A8B	015A	02AB	0239	1B2A	02B9	102A	094B	030B	1B09	0030	1B06	0940	0300
0587	TMC582	0709	1F94	0032	0239	101A	005A	1B04	0509	1F9A	0B09	1792	005A	1A0C	094A	030B	0239	100A
0598	TMC582	0509	1B0A	0309	1A0E	00AA	00AA	1A06	0059	1D0E	0A1E	09D6	09D5	0E2C	0E2C	09D5	1B0A	00A0
05A9	TMC582	0A00	0BEA	1B0A	0000	00CB	1E2E	0A0A	0000	0A02	1B0A	0001	0007	0012	0042	0A0D	1E1D	0B00
05BA	TMC582	094A	0940	0A0D	0059	1E05	0AB5	13AE	0AA5	179C	0A95	1FA0	0A75	1A0A	0AB5	0A55	1934	0A75
05CB	TMC582	1B2A	0A55	0AA5	1B2A	132E	0051	02D7	0AD7	02F0	0101	0A1F	02D0	0309	1708	0409	1710	02D7
05DC	TMC582	1987	0A1F	09D3	06F3	0A19	11AE	1E01	0AA5	0AA5	179C	0082	00D2	1E41	1B45	0A55	1FC4	
05ED	TMC582	0AA5	1F92	00B0	00D2	00B2	1F98	0A27	02F6	07DF	17EA	177C	1E97	1FB1	0100	0100	0100	0100
05FF	TMC582	0100	0100	0100	0100	0100	0C40	1F74	0011	1F08	1EA1	13E1	1EC5	0619	1A1A	095A	095A	095A

TABLE IIa

HEX ADDRESS	CHIP	INSTRUCTION WORDS→															
060F	TMC582	0C53	0C53	071A	0C53	0619	1806	0940	0C40	0309	180A	005A	0048	1004	020B	0050	0052	1267
0620	TMC582	00EA	1257	000A	133C	00D6	000A	0213	0A9A	0A0A	1005	0A10	0A09	04E0	0820	180C	0A05	0A55
0631	TMC582	0A65	0A75	1019	0AE0	0820	180C	0A05	0A55	0A65	0A75	1029	00CA	004A	0A80	0A0A	18CE	009A
0642	TMC582	1A2A	0A01	0CD5	0F24	0921	0A18	0CD5	0A45	101A	0CD5	0F24	0F24	0F24	0CD5	0A9A	0A0A	100A
0653	TMC582	0213	0A10	0A09	04E0	0820	1A39	04E0	0820	183F	0A05	0A55	0A65	0A75	1849	0A01	0619	180A
0664	TMC582	0C4A	094A	0A75	181A	0A45	1812	0045	006D	0085	0091	0A95	0095	1804	0045	0A45	180C	0090
0675	TMC582	0A85	0080	1004	0085	0090	181E	0092	0A85	0085	180C	008D	0A95	0090	100C	0095	0A45	0045
0686	TMC582	1806	004D	00B5	004A	1628	0A10	0AEA	1824	0A97	02F6	0131	0A1F	06D6	067E	041F	0906	02F6
0697	TMC582	0239	0639	1012	0A00	0AEA	1798	0905	0E2C	0E2C	0905	1FAE	0066	0069	0050	0052	1717	00CA
06A8	TMC582	1066	0A0D	0A0C	0085	0A0D	0A75	0A65	0A55	1FAD	0015	0A0C	0F01	1EF6	00C0	0051	10C3	17F3
06B9	TMC582	0A1F	09D3	04E3	0619	1FAA	0050	17BF	00AA	00A2	13D6	0082	18D2	0A08	09F6	0002	01D7	
06CA	TMC582	0070	180A	0309	1806	0A87	0CF0	0940	0980	014A	030A	0140	03D0	11AF	01D7	00AB	1173	1445
06DA	TMC582	0AD7	02F3	0111	041F	01D6	01D5	0E2C	01D5	1249	0AB5	0AA5	0A95	0455	100E	0A75	1F0F	0A65
06EC	TMC582	1006	0A45	108B	0111	040F	0131	18F6	005B	1004	0144	0A06	02F4	0022	0095	00E5	00D5	0A0D
06FD	TMC582	0032	1B4B	0010	180A	0939	1BE7	0DD9	165F	008D	01DA	0CC3	0920	0030	15C8	09DC	016C	0729
070F	TMC582	1D4A	0C89	1550	0C8A	072A	1809	1558	01DA	0086	0DD9	177D	0030	1FC6	01DA	1F85	0A0D	0A0C
071F	TMC582	0104	0A06	02F6	0114	0120	0919	0939	1C8D	0DD9	13C5	0036	0DD9	0049	15C0	0028	1A0A	0D99
0730	TMC582	014A	0022	0203	914A	060C	008D	0A04	0CD8	004A	1004	0C20	0ED4	004A	1012	0C23	0A08	02F4
0741	TMC582	032C	0048	114A	0CCA	19A0	0CC0	1813	0A04	0A75	1819	004A	1F10	0C23	0C1A	0C5A	19AA	0C83
0752	TMC582	0A0C	014A	021A	016A	010A	01C0	0030	17A2	0CBA	1FA2	1DA2	1DA2	1F63	0A65	180C	0455	180A
0763	TMC582	000A	000B	19A9	02D7	0309	02D7	1A34	0000	0070	0002	00CA	161A	19DD	0051	0ED4	003A	002A
0774	TMC582	0013	0123	1F55	0AA5	0A85	0A75	0A55	0A85	1020	0A95	1A31	02D7	0309	02D7	11A3	0A1F	01D6
0785	TMC582	017F	017E	00E9	0A0A	02F6	0174	0AD7	0A1A	0AB5	180A	0A95	1022	0075	0AA5	1804	0045	0A95
0796	TMC582	1804	0055	0AB5	1804	0045	0A0A	02F6	0239	1006	017E	0C7A	00E8	1006	004A	147A	0176	0040
0747	TMC582	16DF	0A1F	02D6	0A0E	0131	01D7	00CA	00AB	14D2	0040	175A	06DA	014E	01D7	0070	160B	0A0D
07BB	TMC582	0A04	0015	0A0D	17C7	16CA	0130	0007	04D0	1A97	0619	1806	0C4B	094A	0A1F	09D6	06F6	0639
07C9	TMC582	11CA	00DA	00A0	00CA	0000	11C6	0A70	0AED	1A1B	0A97	02F6	0131	0A1F	06D6	067E	0639	100A
07DA	TMC582	0053	0050	15ED	0AA0	0AED	101A	009A	1940	01D7	0A1F	09D0	0140	0E0A	0E09	01D7	1190	000A
07EB	TMC582	100A	0AA0	0AED	1009	0A97	02F6	0131	0A1F	01D3	0C5E	0C7E	000A	000A	1F76	0939	1748	00DA
07FC	TMC582	1006	000A	000A	0072													
0800	TMC582	00A2	17F5	1BF3	0A97	02F6	01D7	0231	09F0	01D7	00A2	1D97	1E7D	1D85	0126	0A04	02F6	0024
0811	TMC582	0034	0400	1FF5	1FFB	032C	19AF	0C91	1FCD	1143	00A2	00D1	1EF9	0051	0020	1804	0099	01DB
0822	TMC582	0F09	170B	0ED0	072C	150C	072C	1500	0700	1B0B	0A3F	0A0C	0A0B	06F6	0A0E	0A0C	0A0B	02F6
0833	TMC582	01D2	0A90	0AED	1A47	000B	000B	180C	00CB	1262	0000	0002	1689	0A0C	0239	0231	0A1B	017E
0844	TMC582	1FA4	0239	1FE7	0A05	1804	0085	0A55	1804	0095	0A45	1804	0045	0A75	1804	0095	0231	0A1A
0855	TMC582	1899	1D7A	17A2	0A0C	0AA7	1CC5	1F42	00BA	177A	0000	01D7	0A1F	09D3	0002	16DF	0CF6	01D5
0866	TMC582	072A	0619	1FD2	00D3	0953	0953	0953	0989	1D3A	0AB7	09F3	0988	0024	1505	0983	0D2C	1002
0877	TMC582	0960	0990	04D7	02F3	0111	041F	01D3	0008	041F	01D0	1806	0509	180C	095A	095A	095A	099A
0888	TMC582	1004	014A	095A	095A	095A	095A	095A	0953	0953	0953	0953	0953	0993	000A	1004	06DA	0C53
0899	TMC582	0910	0A0E	0101	01D5	00B9	01D7	10CF	1FAE	0F21	1804	0090	0030	000E	0AA2	0026	0006	0009
08AA	TMC582	0A0D	0059	1B0B	0031	00EA	001A	16B6	01D2	041F	01D0	0948	0A1A	06F6	0A45	111A	00BB	1586
08BB	TMC582	0A67	09F6	0589	0500	150E	00B9	0D00	0D00	1D96	162A	1F6C	095A	070A	1B05	06DA	0A07	0A1A
08CC	TMC582	0C0A	0A0B	02F4	00B0	171C	0CEC	1F1A	0CE4	1A20	0465	0A04	1786	00B0	181A	0C33	1F7A	1FDE
08DD	TMC582	1FDF	00B0	1AD0	0ACD	0A57	0A1A	1EA7	01D7	09AB	02F6	0A47	09F6	09B9	1FF4	0D0A	0D09	0501
08EE	TMC582	1F9C	1790	0100	0100	0100	0100	0100	0100	0100	0100	0100	0100	00B0	1B04	0051	0030	0D09
08FF	TMC582	1F1C	0A77	06F3	0689	1FFC	0090	11F1	06DA	1F0A	1631	0051	0A0C	01DA	01C0	0032	0022	1B37
0910	TMC582	0104	0A0A	02F6	0D09	0012	0030	11E7	0940	072C	1B05	183F	0A97	02F6	0131	0A1F	01D3	0C5E
0921	TMC582	0C7E	000A	00DA	1D20	02DA	0939	1A46	03D3	1A82	1FF4	1FF4	0080	0052	1FF4	014E	0A1F	01D3
0932	TMC582	017F	1FE0	01D7	0103	180E	01D7	0AA7	09F0	01D7	00F5	00C5	00A5	0A57	0A18	0A0D	0053	0050
0943	TMC582	1FCA	0049	1F7D	16AB	0030	11D0	0709	1B0A	0E0A	0009	0F00	11C6	0099	0051	1A7F	070A	0030
0954	TMC582	1B06	094B	1F9F	030A	0309	1F45	0036	1FEB	0050	1B04	0011	0A77	0A1B	1BAD	0050	1004	0011
0965	TMC582	0A57	0A1B	01DA	01C0	0031	0700	1BAB	01D7	0176	0AD7	02F6	0131	0A1F	01D3	01D2	0E4B	1572
0976	TMC582	0131	0A0F	0101	09DF	017E	00CA	07D0	0AAA	1002	0709	1A00	0DD0	0A77	06F0	03D0	06AB	02BA
0987	TMC582	100A	0709	0309	15FC	0D0A	0A97	1B15	01D7	1AA1	0A85	1457	0A95	1F93	0A75	0A45	1861	0A65
0998	TMC582	1DCF	0A55	0A95	00AA	1A6D	1DD9	00CA	17C1	0A0D	0A0C	00D5	00E5	0095	00CA	13DA	0A3D	0A85
09A9	TMC582	104F	0AA5	14AF	0A95	1366	0A75	100A	0AA5	17BD	0A55	1A05	0082	01DF	1C5D	1B0A	032C	1070
09BA	TMC582	0C64	0F09	1773	0C5A	0D29	1561	0020	0032	1FB1	0033	0023	1F87	0A85	15F4	0A75	17CD	0A65
09CB	TMC582	1F04	0A55	1514	00B2	1E3D	0AA5	0A95	0A75	103F	0A65	1A43	0A85	1636	0A55	104A	0AA5	1DFF
09DC	TMC582	0AB0	1FED	1055	0C4A	032C	1B05	00B0	1221	0CB0	1433	0C4A	0C7E	0F00	0C5A	1430	175E	1FE7
09ED	TMC582	0C4A	072C	1B05	0A0C	0CD2	0E09	1B06	0C53	0C40	0020	1B0A	0C4B	0031	072A	17F7	004A	0420
09FE	TMC582	0C1A	1D9F															

TABLE IIb

HEX ADDRESS	CHIP	INSTRUCTION WORDS→															
				1BF7	1BF9	1BFB	1BFD	1BFF	1C01	199F	1C05	1B0D	1FB6	03D3	1FBC	0099	0051	1B95
0A0F	TMC583	17DF	1FE6	1C0B	1F7C	1B4A	1DC2	1B17	00E9	1C29	00B1	1FDE	0C4B	0C4A	0509	0140	100A	0A27
0A20	TMC583	02F0	0D0A	0D0A	0C4A	010A	00AA	11F6	0090	0092	1722	0070	1762	01D7	0051	1D92	00C0	00E1
0A31	TMC583	0104	0012	0A06	02F4	1CB1	0ED6	1D85	01D7	0AF7	09F0	0940	0D0B	0A87	02F3	0082	0A97	01F0
0A42	TMC583	0101	0A1F	06D0	0C40	0B00	1D34	0110	01D7	01D2	041F	01D6	0C7A	0940	0201	09F6	0986	0A0F
0A53	TMC583	0131	094B	064F	0A39	1A40	0309	000B	100A	00DB	0009	00D9	1810	0A67	09F6	0589	0500	1006
0A64	TMC583	0D00	0D00	0140	1DD2	00A0	00A1	1791	0041	1DC9	005A	01D7	0091	1457	0A45	1F07	0A05	170B
0A75	TMC583	1F99	00B2	0A45	0A55	1189	1F93	0070	1354	0A85	0A75	1016	0AA5	1534	0A95	100F	0AA5	1536
0A86	TMC583	0A55	1B06	0A45	17B1	00AA	11AF	00A0	00A2	1B04	0AAA	0001	01D7	0A0B	0CF0	0A07	0A1B	0A85
0A97	TMC583	1B0B	0A95	101A	0075	0A45	1B04	0065	0A95	1B04	0055	0A85	1B04	0045	0A0A	02F0	0140	10E7
0AAB	TMC583	0030	1B04	0700	094B	0022	0148	02D2	0148	0204	0C40	0C40	06DC	1B2A	0C5B	0C4B	1B1C	0C9C
0AB9	TMC583	0C9C	0CF3	0C99	1B09	01D2	0CBB	01D2	0C94	1B02	0E29	1B1D	0C64	0C4B	0F09	1B1D	0C10	0E09
0ACA	TMC583	1B04	0C40	1B07	0168	0214	0164	0420	1D7F	1CB7	0CB3	0E2C	1B05	0C64	0C4B	0F09	1B0B	0140
0ADB	TMC583	0210	0140	0910	1099	0A0D	1B41	153F	0CBB	1245	0E2C	1B07	0C4A	0C7E	0F29	1B0B	0E09	1B06

TABLE IIb

HEX ADDRESS	CHIP	INSTRUCTION WORDS→															
0AEC	TMC583	0C40	0308	06DA	0E09	0C14	1010	0903	09D8	0E29	1F2F	0C64	0700	1809	0032	0940	0700	1818
0AFD	TMC583	0FD3	0CD2	1EC7	01D7	0001	0A27	01F0	06F0	0140	01D7	0072	00C9	1AD3	0099	000A	1CE1	0A45
0B0E	TMC583	12F0	1A0C	1C46	1F92	1867	18F1	1DFA	1F52	0042	1F1A	1772	0153	0153	0153	0153	0183	1D3A
0B1F	TMC583	0100	1ED6	1D93	1E06	0A0C	1E02	1F00	1EFE	1FFC	1FFA	1FFA	072C	0021	0CD2	072C	1010	0C80
0B30	TMC583	1A07	0C08	0C58	0E00	030A	1811	0E09	1806	0C40	0300	06DA	1E51	00C9	00D1	01D7	0A45	0A77
0B41	TMC583	1014	0A45	0A55	0A47	180A	0A85	180F	0A07	180A	0A85	0A87	1004	0A27	02F0	0A85	1804	0300
0B52	TMC583	0AE7	00C8	02F6	0131	0A1F	0CD0	0131	0A0F	0101	1428	1EA3	1F57	17AF	1749	0A85	12FA	0A75
0B63	TMC583	11DB	0A65	12EA	0A45	1367	00A3	1A69	0901	09F0	0140	0140	0AD7	02FA	0131	0A1F	09D6	0976
0B74	TMC583	097F	097E	097E	097F	097F	09BE	1F24	0A75	1804	0903	0148	0148	0509	01DE	16A2	0C40	0A27
0B85	TMC583	04F6	09F6	0988	0C48	0508	05BE	0A7A	0C76	1A9C	0980	0A0A	09F6	0980	0A0F	0101	0948	00B8
0B96	TMC583	00B8	1D0A	0A75	0148	0148	0148	1A65	0500	06B8	0500	1A05	0A0C	0A1A	0A65	1804	0A45	0A55
0BA7	TMC583	1804	0095	0A45	1804	00A5	0AB0	0A1A	01DE	01D5	01E4	01D5	1660	0009	09DE	0C76	0C76	0973
0BB8	TMC583	0953	01D2	0980	01D2	0131	0A0F	0111	0008	000A	1791	0939	1537	0009	01DB	04F7	02F6	0131
0BC9	TMC583	0A0F	0111	1FDB	01D7	01DB	01D7	1D4F	1F79	1FB6	02F3	0219	0A03	0A97	01F0	174C	0AD7	02F6
0BDA	TMC583	0131	0A1F	09D6	017E	017F	017E	017E	0990	1D10	0903	014B	0148	0509	01DE	16BA	0140	0A27
0BEB	TMC583	09F6	06F6	0988	014B	050B	05BE	0676	0C76	1A4A	0140	03D6	0AF8	01B1	0111	1E8F	16AD	1FB2
0BFC	TMC583	1F8A	0100	0100	0100													
0C00	TMC583	0100	0100	0100	0100	0100	0100	0100	0239	1F85	0095	1F71	0A1F	01D6	017E	00FA	1FA1	0AA7
0C11	TMC583	02F0	0101	0A0F	0131	00AB	142E	01D2	1FC3	0002	1EF7	0148	01D2	01B0	01D2	014B	09F6	06F6
0C22	TMC583	090B	0140	0939	1A63	050B	10AF	1A1C	00AB	1FAB	00C9	0A0D	1AB2	01D2	017E	0309	1A36	0070
0C33	TMC583	1032	0000	1012	0A0D	0A65	1808	0A75	16BE	0AB5	0A0D	1A1E	01D5	0AA7	06F3	0621	0CF4	0699
0C44	TMC583	0619	01D5	0239	1073	0000	1A06	0A39	157B	0A1F	01D3	0231	06F3	0A0F	0111	017E	06DE	00E9
0C55	TMC583	10AF	00B0	136A	1B01	0A45	10AD	0001	1CFF	0C4B	032C	1B05	0CD2	0C89	171B	0C8B	0A08	02F4
0C66	TMC583	1D00	01D7	00B6	014B	09DB	0140	0409	1A06	03D0	0B00	0D00	0D00	0A1F	09D6	097E	0DDB	1A05
0C77	TMC583	0931	0A1B	0309	1ADB	0A55	1004	0091	01D7	1D27	165C	0090	1B08	00C0	1E69	01DE	00C0	1499
0C88	TMC583	01DB	1A9D	0C4B	072C	1B05	0080	1F96	01DC	0030	0ED3	1F97	0E09	1F95	0CD2	1F95	0F24	0C88
0C99	TMC583	1B05	0C80	1AC3	0C64	0C58	0C19	100F	0C4C	01A3	00B0	1B04	0A06	0030	0007	0CC4	1004	0C98
0CAA	TMC583	00C0	1823	0020	1827	0022	0CE4	0C9A	14FF	00A0	1898	0C24	01DC	1B3D	0A0D	0A0C	0A1B	01D7
0CBB	TMC583	1D42	0D0B	0D09	0231	09F0	01D7	1B0D	00B1	1B11	0A75	1086	0A65	1AC3	0A55	1783	1E2D	187F
0CCC	TMC583	14FC	0A55	1E35	0445	101A	1CAF	0495	141B	0AB5	1443	0475	12AE	0A65	1E4B	0A55	0A45	0082
0CDD	TMC583	1841	0A55	144D	1F59	041F	09D0	0509	01D7	1F2D	000A	1F31	0A65	0A55	166D	0A0D	0A0C	00F5
0CEE	TMC583	00D5	00C5	0A0D	1E29	0A45	167F	1831	04D3	19BA	0239	01D5	01DB	1DF4	052C	0A67	09F3	15EC
0CFF	TMC583	04D4	0CDB	032C	052C	1008	0CB0	0E24	1A09	0C6C	0C4B	032C	1811	00C0	1804	0F36	0CB0	0F0A
0D10	TMC583	1467	00F1	0031	0CD5	1C61	00CB	00CA	16ED	01D7	01DE	18C8	0A95	1D25	0A45	1D29	0A85	0A55
0D21	TMC583	1D4F	1531	1C9A	1CF2	1DF2	162F	1E70	0497	02F3	0111	0A1F	0603	0A5B	0619	1216	0049	0041
0D32	TMC583	1A73	01D7	09D6	0A07	02F6	0131	0A1F	09D0	014B	0148	014B	09DB	0140	0140	0140	098B	0131
0D43	TMC583	0A1F	09D6	097E	097E	097E	097E	09B9	1E61	09D0	1A4C	0092	1BCF	1F81	0AF7	02F4	0121	0A1F
0D54	TMC583	01D6	0A57	02F0	0299	1042	0304	0308	0C76	0940	0308	1807	0C4B	032C	1805	0308	1B96	0121
0D65	TMC583	0A1F	09D0	0121	0A0F	0131	032C	0121	0A1F	01D6	0A47	02F0	0C4B	0C4A	0C76	030A	1809	0C86
0D76	TMC583	0121	0A0F	0131	04D4	01D2	1B61	0A06	01DB	017E	01D7	0070	1C22	0176	01D7	0630	01D7	00D0
0D87	TMC583	1C12	0AE7	02F6	0131	0A1F	0206	09F0	0D0E	0939	1F5C	0A77	0CF6	0B09	100C	02F4	0239	1F52
0D98	TMC583	0A67	0CF6	0A1F	01D0	09AA	0508	1A0B	0AC7	09F3	09B0	0A0F	0101	00D2	1B0A	00DB	1B0A	
0DA9	TMC583	0A27	06F0	0640	0A37	09F0	017E	0A2F	0A0C	0A0A	09F6	0D0A	1B0D	0176	0176	0174	0930	1FA7
0DBA	TMC583	01D5	0309	01DF	100A	0508	0467	09F6	1002	0140	0646	0C76	0AF8	01B1	01D0	00AB	171C	01D5
0DCB	TMC583	00D2	0201	0007	0054	1DF0	02DB	0AF7	02F6	0131	0A1F	0CD3	0131	0A0F	0111	01D7	01DB	1F3D
0DDC	TMC583	0072	0309	01DE	12B8	0508	0A67	09F6	12A0	1FBE	032C	0121	0A1F	01D6	0C30	0A47	02F4	0C7E
0DED	TMC583	0C5B	0C40	0C40	032C	1A08	09B3	0121	0A0F	0111	032C	0121	0A0F	0131	0089	00DB	1A08	030A
0DFF	TMC583	1804	000A	01D7	04D4	1F2E	0D29	1076	0ED4	003A	1B70	0098	1B67	0080	1015	01D7	0A77	09F0
0E0F	TMC583	01D7	1FA0	1910	190E	190C	190A	1908	1BF4	1BFE	1902	1900	0AA7	02F6	0131	0A1F	01D6	03D0
0F20	TMC583	1984	1BF0	1884	1BEC	1E2A	1BE8	1E24	1E22	1E20	1BE0	1778	0A67	1914	14A2	0A45	1E73	1297
0E31	TMC583	1BD0	1BCE	1BCC	1E0A	1BCA	1E04	1E02	1E00	1BC0	00AA	0042	00CA	0002	0091	0059	187E	1BB0
0E42	TMC583	1B8E	1B8C	1EE8	1B8A	1EE4	1EE2	1EE0	1BA0	01D7	0201	0007	04D0	0054	1F14	0001	1890	1B8E
0E53	TMC583	1B8C	1B8A	1B8A	1B86	1B84	1B82	1B80	00B9	0A1F	09D0	0A0C	0075	014B	10BF	1B70	186E	1EAA
0E64	TMC583	00B3	1B6A	1B6A	1B3E	0AC9	1B60	01D5	0092	00C4	0052	0002	1ECB	1E3D	1B50	1B10	18BC	1B4A
0E75	TMC583	1B4A	1B3A	1B32	0100	0051	1A25	019C	014B	014B	01D5	01B4	1A16	1B30	182E	182C	1E4B	180E
0E86	TMC583	1B2A	1B28	1B22	1B20	1E44	1DD4	00B0	1B0B	00B0	1014	0051	1B10	180E	180C	1E4B	1B0A	1E44
0E97	TMC583	1B04	1B02	00B2	0040	158C	0219	1342	017E	03DB	1956	0910	0D00	1D97	1FAB	01D7	0A1F	01D0
0EA8	TMC583	0A47	0AF6	0A70	0A0F	0101	0A0B	09F0	02F6	0D0B	0D0B	0D09	1131	090E	100C	0939	14C7	0A37
0EB9	TMC583	0AF6	0D00	0901	02F6	1CD3	00B9	0A1F	01D0	094B	0A1B	050B	1E6D	094B	014E	017E	017E	09DE
0ECA	TMC583	0139	1B06	0A37	09F6	0940	098B	1FA5	014B	014B	03DB	03DB	01DE	1285	00B1	0901	02F6	1AB7
0EDB	TMC583	0140	0AFB	01B1	01D6	0A75	014B	1632	03DB	1AA7	017E	017E	1B09	0139	0D09	100E	00BB	1B1B
0EEC	TMC583	0709	1B1B	0C19	1B1D	0AA7	02F3	0249	1B0B	0A0C	0045	0A6B	0A0C	00B5	0020	1B04	0A6B	04AA
0EFD	TMC583	00A2	0042	0002	0050	160F	1E65	0100	0100	0100	0100	0100	0100	0100	0100	0100	0100	0A75
0F0E	TMC583	16AC	0939	1D1B	0A1F	01D0	094B	00DB	00DB	00DB	0940	0A0F	0101	00B9	0040	11C5	00AB	177F
0F1F	TMC583	00DA	01D2	0099	1AA3	0140	0646	0C7A	0AFB	01B1	01D0	0A0B	02F6	0201	0A1B	0A55	0A65	0231
0F30	TMC583	0A1B	1F90	01DB	1F96	1FAF	17AC	0A01	0A0C	0A9B	0A0B	1005	0ABB	0A0D	0A55	1B0D	0A0D	0AFF
0F41	TMC583	0AA0	0AEF	1D19	1F55	0140	04FB	01B1	01DA	0A77	02F0	0090	0092	16DE	00CB	0050	14BB	0000
0F52	TMC583	15E2	0A0C	0231	0A1B	0239	017E	197F	0239	1983	00F5	00F5	00D5	0A75	1804	00B5	0A65	1804
0F63	TMC583	00A5	0A55	1804	0095	0A45	1804	00B5	0231	0A1B	0015	0A0C	0120	0633	0A0F	0111	0A06	02F0
0F74	TMC583	01D7	0A97	02F3	0111	0A1F	01D0	0F00	0111	0A0F	0101	0A0D	0A04	0A0D	0A1F	01D0	1B04	0140
0F85	TMC583	0303	0AFA	0181	0131	00AB	1F20	01D7	0A03	0D09	0045	1FFA	0A47	0AFA	0133	0A1F	09D0	094B
0F96	TMC583	094B	0A1B	1C9D	0A55	0A0D	16BB	1FA1	0A40	0FFD	0A1F	01D0	0700	1304	0A0F	0101	09DB	0C40
0FA7	TMC583	0C40	0091	0A41	0CD1	1CEF	1D4A	17AF	0497	02F3	0111	0A1F	01D0	094B	06F3	094B	09F0	0940
0FB8	TMC583	0C53	0C53	0980	0111	0A0F	0101	1EF3	1749	01D7	00B0	16F4	0AD7	02F0	0101	0A1F	09D0	094B
0FC9	TMC583	014B	014B	094B	095B	09DB	0C40	0C40	09BB	02F0	1351	1F00	1FAC	0CCB	00F0	1010	0023	0A57
0FDA	TMC583	0A1B	0CDB	00C0	1B04	0022	0CCC	1014	0CCB	00C0	00B0	1B04	0023	0CCC	1825	0CD2	0CC0	0CAC
0FEB	TMC583	0CDB	1FB7	0C64	0AFA	0141	03DB	00DB	1391	1FC4	0164	0663	0C53	03D3	0A16	02F6	0AFA	0191
0FFC	TMC583	0131	01D5	04D4	1EED													
1000	TMC583	1F50	1F4C	1F4C	1F4A	1F48	1F46	1F44	1F3C	1F40	1F3E	1F38	1F3A	1F38	1F2C	1F34	1F32	1F24
1011	TMC583	1F2E	1F2C	1F1C	1F28	1F26	1F14	1F22	1F20	1F0C	1F1C	1F1A	1F04	1F16	1DB6	1D49	1F10	1F7C
1022	TMC583	1EFF	1F0A	1EF2	1F06	1F04	1EFA	1F00	1EFE	1EE2	1EFA	1EFB	1ED4	1EF4	1EF2	1ED2	1EFF	1EFC
1033	TMC583	1F5A	1ED2	1EE6	1F52	1ECA	1EF0	1F0C	1EC2	1ED4	1F06	1EFA	0176	1F9A	1ED4	1EF4	1EB2	1E4F
1044	TMC583	1F04	1FC0	1F08	1F04	1F0C	1EFE	1FCF	1FD0	1FBA	1F9A	1F9A	1FBC	1FC4	1F94	1F90	1EBC	1EA2
1055	TMC583	1EFA	1EB6	1EBE	1EC0	1EAF	1EB2	1E9A	1EA6	1E7C	1A71	1DAA	1EA2	1E7A	1E92	1E9C	1E92	1E9C
1066	TMC583	1EBA	1EF6	1EC0	1E7F	1EF4	1E78	1EB6	1EAA	1EB2	1EE2	1EAA	1EB0	1EBA	1E76	1E70	1E04	
1077	TMC583	1E70	1E72	1E7A	1E64	1EA2	1EAA	1E3C	1600	16D5	1F34	1F60	1F6A	1E50	1E50	1E4E	1E54	1E4A
1088	TMC583	1F4A	1E54	1F22	1F44	1F4C	1E7C	1E3C	1E32	1F9C	1F6C	1F34	1E3C	1F44	1E2E	1E3A	1E08	1F06
1099	TMC583	1E5E	1DFE	1DFC	1E22	1F24	0910	1C22	1E2C	1E4A	1E2A	1E10	1E10	1DEE	1E04	1DE4	1DE2	1DFE
10AA	TMC583	1E24	1DFA	1DFA	1F22	1DF4	1DF2	1E54	1DFE	1E02	1DCC	1DCC	1E06	1DF6	1DC4	1E20	1DFA	1DFE
10BB	TMC583	1DF0	1DFA	1DF2	017E	1AF9	1DF0	1DB0	1DB0	1DDA	1DCE	1DE6	1DFA	1DF2	1DCE	1DF4	1DFC	1D9C

TABLE IIb

HEX ADDRESS	CHIP	INSTRUCTION WORDS→															
10CC	TMC583	1DFF	1D9A	1DC2	1DB2	1DEA	1D4E	1DF2	1D9A	1DB6	1DB2	1DB6	1DA2	1DAF	1DAE	1DAE	1D9A	1DA6
10DD	TMC583	1DA6	017E	1F21	1D74	1D7A	1D96	1D9A	1D70	108E	1D8A	1DBA	1D88	1DBA	1D62	1D84	1DAE	1DB6
10EE	TMC583	1D7C	1D72	1D9A	1D6E	1D82	1D52	1D7E	1DAA	1DBA	1DA2	1D60	1D9A	1DBC	1D5A	1DC2	1D56	0107
10FF	TMC583	1C59	1D34	1D5A	1D4C	1D50	1D50	1D50	1D4E	1D5F	1D5E	1D7A	1D4C	1D5A	1D6E	1D44	1D44	1D32
1110	TMC583	1D64	1D2E	1D4E	1D30	1D30	1D4B	1D46	1D3E	1D2A	1D32	1D00	1D1A	1D4A	1D16	0100	1E0A	1D1A
1121	TMC583	1D16	1D22	1D12	1D7A	1D10	1D16	1D08	1D0A	1CFE	1D44	1CFA	1D1C	1D6A	1D2E	1CF2	1D60	1CEE
1132	TMC583	1CF2	1CF4	1D2A	1D22	1CEC	1CC6	1D24	1CC2	1CC6	1CE0	1CE2	1CE4	0447	1804	0A37	02F6	0A87
1143	TMC583	1F55	00A2	0091	0A97	02F6	0131	0A1F	01D0	0E0A	0131	040F	0101	01D7	0051	1E00	0051	0131
1154	TMC583	0A1F	01D0	1F15	01D2	0140	0A0A	02F0	0116	0308	162A	017E	017E	1809	00D0	1089	0A0C	0116
1165	TMC583	1D9A	0A1F	09D0	0509	1455	0AE7	02F3	0111	0A1F	02D0	0409	19DA	0CD7	0309	04D0	1EAA	01D7
1176	TMC583	0AF7	02F4	06F3	0121	0A1F	01D0	0308	1030	0A1F	01D6	0103	042C	0320	0C40	0C40	097E	0308
1187	TMC583	1809	0C48	0C48	097A	032C	1809	097A	098A	06F3	06F0	040F	0131	0708	1ECA	09DE	0A97	02F6
1198	TMC583	0131	0A1F	01D0	0606	0C48	0C48	0930	0C40	0C40	0630	0131	040F	0101	0411	02F0	0658	0A19
11A9	TMC583	1755	1F8A	1DA4	0631	0A1A	0239	017E	0A75	0A55	0A05	1176	0A65	1972	00CA	197A	0091	00CA
11BA	TMC583	1F43	1EE7	0120	0A06	02F0	0103	01D7	0509	0501	108A	094A	02F6	0809	1008	0089	0D08	0D0A
11CB	TMC583	0D0A	101A	0D0A	10F9	0D0B	10D5	0D0B	10FF	0A0D	0AF7	0A1A	0A0D	1909	0059	0A16	02F4	0131
11DC	TMC583	040F	0121	0051	14C3	1FA4	01D5	0A1F	01D3	095A	0953	040F	0111	0099	04F7	02F3	0111	0A0F
11ED	TMC583	000A	000A	176A	1F72	0100	0100	0100	0100	0100	0100	0100	0100	0100	0A45	0910	1046	02D7
11FE	TMC583	0AA7	02F0	0101	0A1F	01D3	0101	040F	0121	0107	0C58	0C40	030A	100C	0153	0153	0C40	0C40
120F	TMC583	1800	01D5	0080	172F	0920	019C	0153	0153	0194	01D5	04D0	0C48	030A	1805	09D7	0146	01D7
1220	TMC583	015A	0509	01DB	153F	0140	0427	09F3	06F3	098A	0148	050A	05AB	0A53	0C53	1555	00B0	01D5
1231	TMC583	1FFA	016A	04D0	0C40	030A	1805	016C	016C	09B6	01D2	0D0B	015B	1B3D	0100	1EA5	00B2	017E
1242	TMC583	19BE	0002	0040	10D4	0CD2	0219	0173	1804	0303	06BB	0A19	0CD2	15FE	0219	01D5	1347	01D2
1253	TMC583	0AA7	02F0	0219	1004	0203	0D00	0904	02BB	09F3	1ED9	08D0	0A87	0AF0	0640	0C40	0CD7	1C89
1264	TMC583	0040	00B2	1A91	01D5	0AA7	02F0	0101	0A1F	01D4	1A65	00CA	00CA	109A	0239	1006	0040	0091
1275	TMC583	0052	10AE	1FBD	1B5A	04D0	0308	1012	0708	1807	094B	0308	1B05	00A0	16C4	1E9E	02D8	1B0E
1286	TMC583	0A97	02F0	0D09	105C	0101	0A1F	09D4	09AC	09BC	0923	099B	0A37	02F4	09AC	0A1A	0A1E	032C
1297	TMC583	1809	0300	07DA	01DC	0153	043E	0A0C	0A08	02F4	0164	0A0E	0A0C	0A08	02F4	0164	1017	030A
12A8	TMC583	100A	0123	1823	09A0	0163	0153	0988	199D	0A37	02F0	094B	0418	0A1F	0308	1809	01D5	01D7
12B9	TMC583	1FEB	1FA7	165A	017E	00B2	0040	1891	00D0	1347	0A0C	07DF	0C16	0431	0A1A	0A04	067E	0A6A
12CA	TMC583	100A	0616	0A9A	0A9A	00A0	13E5	07D0	03D0	0A0C	0631	0A1A	0A04	0C7E	0AAB	030A	180F	070A
12DB	TMC583	107A	0A9A	0A9A	0A27	02F0	181D	01D7	0A37	02F0	015B	0A2E	0A0C	0A0A	09F3	030A	180D	0A0E
12EC	TMC583	0A0C	0A37	02F0	0A0C	0A1F	0A1E	0A1F	0A1E	043F	0A0C	0A0A	09F6	0A0E	0A0C	0A0A	06F6	0476
12FD	TMC583	017A	0211	0A1A	015A	0A1F	0308	1808	09BE	100A	0939	122A	0008	00D8	118A	0A97	02F0	0101
130F	TMC583	0A1F	01D6	090A	0101	0A0F	0131	1889	0AA8	0040	0116	19E4	13F7	0120	01D7	0050	1B46	0A1F
131F	TMC583	01D0	0A47	02F4	04E8	094B	032C	1805	0940	072C	1805	0A0F	0101	0AA7	02F0	0101	0A1F	01D4
1330	TMC583	00B9	1AED	1AF7	1F4E	01DB	0328	0101	0A0F	0111	0008	00D8	1693	04F7	01F0	0101	0A0F	0111
1341	TMC583	00B9	0A1F	02D3	0A0F	0111	0A97	01F4	0121	0A1F	01D0	0640	0648	0121	0A0F	0101	030A	09F0
1352	TMC583	1AC1	0AA7	01F4	0C4C	0529	01DE	16D7	0C64	0A27	09F6	06F4	09AC	0C6C	052C	05AE	0676	0C76
1363	TMC583	1AED	0101	0A1F	09D0	094A	0948	0903	09B0	0A37	02F0	06F0	094B	0A1A	0A1E	030A	1809	05D0
1374	TMC583	030A	0933	01DF	043E	0A0C	0A0A	02F4	0176	0A0F	0A0C	0A0A	02F4	0176	1015	0174	0176	050A
1385	TMC583	1823	0919	094A	0A1A	0A1F	0708	1809	0CD2	09B9	1B59	0D03	0A3E	0A0C	0A0A	06F6	0A0E	0A0C
1396	TMC583	0A0A	02F6	10B3	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04
13A7	TMC583	0A04	008D	1824	1802	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04
13B8	TMC583	0A04	0A04	0085	009D	0A6B	1924	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04
13C9	TMC583	0A04	0A04	0A04	00BD	0095	1825	00C9	0320	01D5	0101	0A1F	01D4	0E24	01D5	1980	0A57	0A1A
13DA	TMC583	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A04	0A85	1831	0070	0072	1603	00A0
13EB	TMC583	1508	0A1F	01D0	01AB	014A	094B	0958	0958	0953	0953	0953	0988	0103	09A0	09B3	09A0	09A0
13FC	TMC583	09DB	0153	0176	09A6													

TABLE IIc

HEX ADDRESS	CHIP	INSTRUCTION WORDS→															
1400	TMC571	0040	103C	0A1F	09D0	0E0B	0E08	1A3A	0A00	0AEA	1032	0A37	01F3	0AA7	09F3	0976	0170	0140
1411	TMC571	0153	0153	0153	0153	0153	04D0	00CA	1775	0A00	0015	0A0D	0099	0004	1ACD	0AA0	0AED	1465
1422	TMC571	0A10	0AEA	1835	0A9B	0A0B	1005	0AA8	0A0C	0133	0040	100C	0A78	0040	16F3	004A	12C3	01D5
1433	TMC571	01AC	062A	0A1B	0A75	1804	00C5	0445	1804	00B5	0A55	1804	0045	0445	06DC	1808	0A27	06F4
1444	TMC571	0095	0005	0A37	09F4	0223	0294	1804	0724	0294	1804	0724	0D2C	1024	0624	0A1A	0A55	009D
1455	TMC571	1804	0095	0A45	00BD	1804	00B5	0224	0A1A	0324	1804	0724	0A07	0015	0A1A	0114	01D5	000D
1466	TMC571	0040	1335	00B0	1839	0A0C	00B5	0065	0AAA	1841	0A47	02F0	0101	040F	0131	040B	02F0	0631
1477	TMC571	0CF6	0308	1304	0176	0176	1809	0A17	0A1B	0A1F	0A0C	0A1F	0A1E	0A1F	0099	0A0F	0A0C	0A0B
1488	TMC571	02F3	0219	1643	0A97	01F0	0C07	0101	0A1F	09D0	0C4B	0C4B	090B	0C03	0CB0	0A37	02F0	0C4B
1499	TMC571	0A1B	0A1E	030B	1809	07D0	0300	0133	010E	0153	0A3E	0A0C	0A0B	02F4	0176	0A0E	0A0C	0A0B
14AA	TMC571	02F6	0176	030B	1819	070B	1823	0176	0170	098B	01D2	0323	0111	0A1F	01D6	0A37	02F0	090B
14BB	TMC571	0A1B	0A1E	017E	030B	180B	0930	09B0	09B6	09B0	09B0	09B9	16BB	01D2	009A	0D0B	16C3	0AA7
14CC	TMC571	02F0	0101	040F	0121	0A77	02F0	06F0	01DC	0164	0A3E	0A0C	0A0B	02F4	0164	0A0E	0A0C	0A0B
14DD	TMC571	02F4	0D0B	11A4	030B	181B	0176	0C4C	02F6	0176	014C	02F6	070B	180F	01D7	0D0A	014B	014B
14EE	TMC571	109B	0100	0100	0100	0100	0100	0100	0100	0100	0100	0100	070B	1004	00D9	0A27	0A1B	0A1F
14FF	TMC571	0A0C	0A1F	0A1E	0A1F	09AB	0506	040F	0131	01D5	01DC	01DB	030B	0A3E	0A0C	0A0B	02F3	0153
1510	TMC571	040F	0A0C	0A0B	02F3	0153	1015	0153	0153	0969	1006	0D2A	1825	0A37	02F0	09AB	0A1B	0A1F
1521	TMC571	030B	1809	01D5	0121	0A1F	01D3	02F0	030B	07D3	1F29	0A0D	0A0C	03D3	0A97	02F6	0131	0A1F
1532	TMC571	01DA	067E	0050	19B4	0A0D	0090	15E5	0A10	0AEA	1DEB	0040	182C	0A1F	09D6	02F4	0239	0639
1543	TMC571	1012	01D5	0F2C	0F2C	01D5	19BE	0A0D	0AEA	11B8	0A9A	0A0B	1005	0A8A	0A0D	0A7B	19AB	0639
1554	TMC571	1A11	162F	00EA	00EA	0619	0323	0111	0A1F	01D4	10A2	0C4B	0009	1004	0BD0	0C4B	0D20	0C4B
1565	TMC571	0C4B	0C4B	0A8B	150E	02F4	0606	0A1F	09D0	0C4B	0C4B	0C4B	0C4B	0A8A	0A89	1004	00B9	0C53
1576	TMC571	0A07	0919	1804	0AA7	02F0	0C53	02B6	0953	095A	0C7A	0C7A	097A	0D2C	0111	0A0F	0121	0A97
1587	TMC571	02F3	0111	0A1F	0CDA	0111	0A0F	0131	0D24	016F	016C	016C	096C	016E	017E	017E	064C	0924
1598	TMC571	09A4	09A6	09AA	09AA	0D2C	1804	09DC	0729	0C64	1004	0F2C	096C	0C64	0C60	0A27	02F0	0022
15A9	TMC571	005A	0032	1EE1	01D7	0920	09DB	015B	0164	0164	0194	1F37	0051	030A	1149	0164	0164	180B
15BA	TMC571	01D7	0A77	09F6	0A1F	09D3	0901	09F0	0B04	09B0	01D5	0DD4	100B	0B0A	0D24	07D4	0956	09A4
15CB	TMC571	0D0B	1807	09D2	0729	094A	0A1B	0A1F	01D0	01D5	100E	0A55	180E	00BA	1E41	09BB	180C	0A45
15DC	TMC571	100A	00BA	164D	0950	0104	1D9E	0099	193A	049B	0A0B	1005	0A8B	090A	0A0C	0931	0A1A	0A04
15ED	TMC571	097E	0A6B	0939	100E	0070	00A2	180C	0A97	0095	0A1A	0A6B	0041	0AAA	1EFF	01D2	01BA	01D2
15FF	TMC571	014E	017E	017E	017E	09DE	017E	0A0B	02F0	030B	154B	0176	0176	1809	01D5	0639	1834	1741

TABLE IIc

HEX ADDRESS	CHIP	INSTRUCTION WORDS→																
1600	TMC571	0105	0F29	1000	0A9A	0A0A	1005	0A8A	0A1F	0AD3	07D4	0A0C	0611	0A1A	0A04	0A5A	0A68	072C
1620	TMC571	180F	0A0D	0A1A	0A0D	0105	0A9A	0A98	0106	0098	1810	0A0C	0A97	0A1A	0085	0095	0AAA	1804
1631	TMC571	0A9A	0A0C	00EA	007B	100A	0709	1806	0A7A	1814	0219	064A	0A1A	0A04	0A01	0AAA	1009	0030
1642	TMC571	0A0C	1804	0085	0AAA	0130	0308	0A0C	1804	0095	02DA	0A6A	0A0C	0009	1806	0219	1F2F	094A
1653	TMC571	0A1A	0A04	0219	0AAA	1021	0CDA	1825	0103	01D7	0901	02FA	0A1A	0A75	1040	0A10	08EA	183A
1664	TMC571	0A04	0A75	1A8C	0A19	0030	0020	1111	0AE7	09E0	0A00	094A	111A	0A0C	0095	0075	0045	0A9A
1675	TMC571	0A0A	1005	0A8A	0D0A	1004	0A9A	1807	0A6A	0A4A	0110	1949	01D7	0A45	0A97	02F3	0111	0A1F
1686	TMC571	01D3	06FA	065A	1267	0AA0	08ED	1005	0A0A	116C	0619	1804	0031	0C4A	0C4A	0A47	06F3	06D2
1697	TMC571	009A	06A9	1222	0A14	0A9A	0A94	020C	0605	0A49	100C	03D4	0088	1006	0030	1204	090C	0032
16A8	TMC571	0A14	0619	19EA	072C	0A2A	0A44	0A40	0140	0210	0126	0AA0	08ED	11CA	0AB0	0A0A	100A	0A01
16B9	TMC571	00AA	1806	0A5A	1804	0ACA	0AA0	08ED	11A2	0A0A	1009	0A37	09E0	0940	0004	02DA	0A27	06E0
16CA	TMC571	00AA	190E	0A5A	0A2A	0A0C	01DE	0A0A	02E6	02B0	0A5A	0A2A	0A0C	0C7E	0A0A	09E6	02F3	02B0
16DB	TMC571	1002	0708	1815	0619	1984	01AB	0119	1988	0020	1806	0AB5	182E	0130	0A0C	0099	0AA0	08ED
16EC	TMC571	1010	0AB0	0A0A	100F	0A5A	0A04	0AC5	1A07	0004	1146	1890	0020	1804	0085	0ACA	0A3A	1810
16FD	TMC571	0C2A	0306	18BE	0A77	06F0	0C76	0C60	0AEA	01A1	01D3	0C7E	095E	097E	097E	097E	097F	097E
170E	TMC571	097E	097E	097E	097E	097E	0153	0153	0A0C	0931	0A1A	02F3	0A45	1804	0085	0A55	1804	0095
171F	TMC571	0AA5	1804	0085	0A75	1804	00A5	02B0	097E	0931	0A1A	02F3	0280	1002	0AC8	0A3A	0708	184F
1730	TMC571	0C6C	072A	000A	1A47	0A0C	0201	0A1A	0AC8	0A3A	0AC8	0A3A	0AC8	01D7	0C4A	0C03	0C90	0C80
1741	TMC571	0C80	06DA	0D00	0AAA	1002	0004	00BA	0091	02DA	009A	1845	000A	0A1F	0904	0529	134F	1F87
1752	TMC571	02B0	0231	1002	0A1A	0AC8	0A3B	017E	070A	1611	0A0C	01D5	02D6	0C76	06DE	01DC	0107	0130
1763	TMC571	01D7	0A77	0AE0	00AA	1AE1	0A5A	0A2A	0A0C	0153	0A0A	02F3	0A07	0A1A	0AA5	1804	0045	0A95
1774	TMC571	1804	0055	0AA5	1804	0065	0AA5	1804	0075	02B0	0153	0A0A	02F3	02B0	1002	0708	1837	0C6A
1785	TMC571	0AEA	01A1	0111	0CAC	0A5A	0A2A	0A0C	072A	000A	0A77	06E0	1A4B	0A0A	02F3	02BB	0219	0AAA
1796	TMC571	18BA	0A0A	0099	0A02	090A	0C40	0C40	1883	00BA	1816	070C	19EB	01A9	02AE	0239	117D	0A85
17A7	TMC571	1953	0031	1957	0049	1987	0508	0418	0A47	02E0	1012	0A65	101F	02B6	0131	0A0E	0111	01D7
17B8	TMC571	1C1B	0A0C	0300	0101	0A0E	0A04	0A65	1808	1013	0A9A	0A0B	1005	0ABB	0A45	1804	0011	0A87
17C9	TMC571	02E0	0101	0A1F	0900	0A47	06E0	0010	1804	070A	0A0C	094A	06E6	094A	0A1A	0A55	1804	0085
17DA	TMC571	0AA5	1804	0095	0AA5	0A3A	0A1A	1804	0075	0AAA	070A	1825	0010	0051	01D2	1383	01D2	030A
17EB	TMC571	02A9	1A45	0AAA	1A6E	0009	1CC5	0030	1960	0A97	02F3	0111	0A1F	01D6	0C76	0506	0C7E	0111
17FC	TMC571	0A0E	0131	0021	1987													

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 → 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

TABLE III-continued

PROGRAM CODE	FUNCTION	KEY(s)
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, |x|
51	Back Step	BST
52	Exponent Entry	EE
53	(	(
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 $x \geq t$	2nd, $x \geq t$
78	Insert Data Point	2nd, $\Sigma+$
79	Mean	2nd, $\bar{X}$
80	Grad	GRD
81	Reset	RST

TABLE III-continued

PROGRAM CODE	FUNCTION	KEY(s)
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

TABLE IV -continued

Program codes following an heirarchy address code (82)

SECOND DIGIT	HEIRARCHY REGISTER
3	3
4	4
5	5
6	6
7	7
8	8
9	no operation

TABLE V

Coces 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	Increment memory 0–9
.	
.	
29	
30	Decrement memory 0–9
.	
.	
39	

TABLE VI

```
SCOMD 1,STOF=0A0F,RCLF=0A1F,REG=LO,DIGITS=USED
ADDRESS CONSTANT=1,2,3,5,6,7,8
CONSTANT  0=2302585092994046
CONSTANT  1=0693147180559945
CONSTANT  2=0095310179804325
CONSTANT  3=0009950330854168
CONSTANT  4=0000999500333468
CONSTANT  5=0000099995000333
CONSTANT  6=0000009999950000
CONSTANT  7=0000000999999500
CONSTANT  8=0785398163397450
CONSTANT  9=0099668652491200
CONSTANT 10=0009999966668670
CONSTANT 11=0000099999666667
CONSTANT 12=0000009999999667
CONSTANT 13=1570796326794901
CONSTANT 14=3141592653590046
CONSTANT 15=5729577951308010
CONSTANT 16=4375068435530482
CONSTANT 17=5403435501436504
CONSTANT 18=3300437505435355
CONSTANT 19=2455353454034355
CONSTANT 20=0183598004433265
CONSTANT 21=5320825403435558
CONSTANT 22=3255541882859453
CONSTANT 23=8518826532539258
CONSTANT 24=4355044353925432
CONSTANT 25=4355014353325403
CONSTANT 26=8590453332082503
CONSTANT 27=3332540343550243
CONSTANT 28=0343550543859453
CONSTANT 29=7505435353923254
CONSTANT 30=5554034355330403
CONSTANT 31=0882540175034353
```

CONSTANTS

PROGRAM CODES

```
SCOMD 2,STOF=0A0F,RCLF=0A1F,REG=HI,DIGITS=UNUSED
ADDRESS CONSTANT=1,2,3,5,6,7,8
CONSTANT 32=7502435353238458
CONSTANT 33=5554034355330143
CONSTANT 34=4353539234541882
CONSTANT 35=5501436504437506
CONSTANT 36=0543535355580343
CONSTANT 37=5403435533004375
CONSTANT 38=3301437502835365
CONSTANT 39=0254385458034355
CONSTANT 40=3202843308820144
CONSTANT 41=4833078248820444
CONSTANT 42=0180834026410105
CONSTANT 43=2232028498350882
CONSTANT 44=4833078248820444
CONSTANT 45=3782034984010544
CONSTANT 46=4332178206481882
CONSTANT 47=3285330882539203
CONSTANT 48=5353321454330782
CONSTANT 49=4539225450551782
CONSTANT 50=2258178255188253
CONSTANT 51=3908825392245430
CONSTANT 52=8253325407823265
CONSTANT 53=5392541782453818
CONSTANT 54=0006655908825353
CONSTANT 55=0165592218825385
CONSTANT 56=6554598825400000
CONSTANT 57=5459221882850693
CONSTANT 58=8253539254060355
CONSTANT 59=4559221882855908
CONSTANT 60=5400000145540693
CONSTANT 61=1882855953530882
CONSTANT 62=9345540693455922
CONSTANT 63=0000000092540100
```

PROGRAM CODES

TABLE VIa

LOCATONS	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

TABLE VII-continued

ALPHANUMERIC CHARACTER	ALPHANUMERIC CODE
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
↑	60
%	61
>	
<	62
/	63
=	64
,	65
Σ	66
$\bar{x}$	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	50	02	43	11	01	14	15	15	82	17	01	19	51	20	
17	TMC501	29	21	73	22	91	24	14	26	09	27	64	29	52	30	95	32	31	
34	TMC501	33	63	34	80	36	51	42	00	44	31	05	26	46	20	06	72	47	
51	TMC501	68	08	92	76	24	00	42	09	60	58	09	76	25	29	06	42	01	
68	TMC501	00	72	01	97	01	00	15	92	76	95	71	24	05	32	03	00	37	
85	TMC501	78	22	37	78	69	12	88	78	69	11	22	88	22	78	69	14	53	
102	TMC501	24	75	36	15	71	88	54	52	22	52	32	03	07	07	93	02	05	
119	TMC501	08	00	09	05	04	67	96	00	35	76	96	69	00	01	03	03	06	
136	TMC501	03	07	01	07	03	05	69	04	03	00	69	03	69	05	01	99	92	
153	TMC501	76	11	98	99	62	00	11	99	92	76	12	98	99	62	00	12	99	
170	TMC501	92	76	13	98	99	62	00	13	99	92	76	14	98	99	62	00	14	
187	TMC501	99	92	76	15	98	99	62	00	15	99	92	76	16	98	99	62	00	
204	TMC501	16	99	92	76	17	98	99	62	00	17	99	92	76	18	98	99	62	
221	TMC501	00	18	99	92	76	19	98	99	62	00	19	99	92	76	10	98	99	
238	TMC501	62	00	10	99	92	76	19	85	53	24	75	01	50	65	43	07	85	
255	TMC501	07	05	92	76	18	61	08	11	01	22	44	01	43	07	22	44	02	
272	TMC501	92	22	97	05	00	51	75	01	44	01	43	07	44	02	73	01	65	
289	TMC501	73	02	61	00	30	92	22	97	05	00	69	75	71	00	20	73	01	
306	TMC501	65	73	02	61	00	52	92	76	11	02	07	99	98	92	76	12	75	
323	TMC501	32	01	95	65	43	07	85	08	95	02	01	32	92	99	72	01	92	
340	TMC501	32	01	44	01	32	61	00	93	76	13	43	07	42	05	85	33	85	
357	TMC501	07	05	42	01	43	05	72	01	01	22	44	01	97	05	01	18	01	
374	TMC501	42	04	42	06	43	04	85	07	95	42	03	75	43	07	75	08	95	
391	TMC501	42	05	43	04	19	42	02	73	02	50	32	22	97	05	01	88	01	
408	TMC501	44	02	73	02	50	22	77	01	59	32	43	07	85	43	05	85	08	
425	TMC501	95	42	03	61	01	59	43	03	32	43	04	95	07	95	67	02	28	
442	TMC501	42	02	01	94	09	06	43	07	85	01	95	42	05	73	02	63	03	
459	TMC501	72	02	43	07	44	02	44	03	97	05	02	12	43	04	19	42	01	
476	TMC501	73	01	49	06	29	67	03	31	43	07	42	05	43	04	19	42	02	
493	TMC501	85	43	05	75	43	04	95	42	03	73	03	55	73	02	95	42	01	
510	TMC501	72	03	43	07	44	02	44	03	33	85	08	95	32	43	03	77	02	
527	TMC501	97	43	01	94	65	73	02	95	74	03	61	02	69	01	22	44	05	
544	TMC501	43	05	32	43	04	67	03	12	61	02	45	01	44	04	43	07	32	

TABLE VIII

BCD ADDRESS	CHIP	PROGRAM CODES*																	
541	TMC501	43	04	67	03	24	61	01	35	19	42	01	73	01	49	06	43	06	
57A	TMC501	9A	99	9A	92	6A	6A	6A	6A	6A	6A	6A	6A	6A	6A	76	14		
595	TMC501	42	05	92	42	03	43	07	33	85	07	95	42	01	43	05	32	43	
612	TMC501	07	22	77	03	50	01	44	01	73	01	22	67	03	74	43	07	44	
629	TMC501	01	01	44	05	43	03	72	01	99	92	61	03	55	76	15	01	42	
646	TMC501	00	43	07	85	33	85	07	95	42	01	43	00	42	05	75	43	07	
663	TMC501	85	07	95	42	02	00	71	00	30	85	01	44	01	73	01	95	72	
680	TMC501	01	01	44	00	43	04	32	43	07	77	00	04	01	42	00	07	85	
697	TMC501	53	43	07	85	01	54	33	95	42	01	75	43	07	75	43	00	42	
714	TMC501	05	95	42	02	00	71	00	52	85	71	00	20	73	01	95	55	73	
731	TMC501	02	95	72	01	01	44	00	43	00	32	43	07	77	00	52	98	01	
748	TMC501	92	76	16	85	32	07	85	43	07	65	53	20	85	01	54	95	42	
765	TMC501	01	32	92	73	01	99	92	01	44	01	61	05	25	76	17	01	42	
782	TMC501	00	43	00	19	95	42	01	73	01	35	72	01	01	44	00	43	00	
799	TMC501	32	43	07	77	05	40	01	94	44	00	01	42	03	43	00	19	42	
816	TMC501	01	75	43	03	42	05	95	42	02	73	01	65	73	02	94	71	00	
833	TMC501	52	95	65	71	00	20	73	02	95	72	01	01	44	03	43	00	32	
850	TMC501	43	03	22	67	05	69	01	42	03	22	44	00	43	00	32	01	22	
867	TMC501	67	05	69	43	00	19	42	01	85	43	03	42	05	95	42	02	73	
884	TMC501	02	94	71	00	30	65	01	44	01	95	72	01	01	44	03	43	03	
901	TMC501	32	43	07	75	43	00	95	77	06	27	01	44	00	42	03	43	07	
918	TMC501	32	43	00	22	67	06	27	01	42	00	43	00	42	05	19	42	03	
935	TMC501	43	05	19	42	01	43	03	42	02	43	07	75	43	05	85	01	95	
952	TMC501	4A	05	32	73	03	94	71	00	30	95	94	72	03	43	07	44	03	
969	TMC501	32	85	01	95	42	05	32	77	06	92	32	43	00	67	0A	0A	85	
986	TMC501	01	95	42	05	43	00	19	85	01	95	42	03	43	05	19	42	02	
1003	TMC501	43	03	42	01	43	07	75	43	05	85	01	95	4A	05	32	73	01	
1020	TMC501	65	73	02	94	71	00	30	95	94	72	03	01	44	03	85	32	95	
1037	TMC501	42	05	32	43	07	77	07	55	01	44	04	61	06	85	9A	01	92	
1054	TMC501	42	03	32	43	07	22	77	08	10	85	42	05	33	85	07	95	42	
1071	TMC501	01	00	42	04	73	01	67	08	45	01	22	44	01	97	05	0A	32	
1088	TMC501	43	05	75	01	95	65	43	07	85	07	95	42	01	32	92	01	44	
1105	TMC501	01	44	04	73	01	99	92	43	04	32	43	07	22	67	08	60	01	
1122	TMC501	85	43	03	95	1A	61	08	60	76	10	13	29	67	08	95	17	43	
1139	TMC501	06	92	76	11	4A	04	4A	03	43	04	9A	99	92	76	12	75	32	
1156	TMC501	01	95	65	43	03	85	0A	95	42	07	32	9A	92	76	50	72	07	
1173	TMC501	32	01	44	07	32	99	92	61	50	76	13	85	32	43	04	75	01	
1190	TMC501	95	65	43	03	85	0A	95	42	07	32	9A	92	61	50	76	14	4A	
1207	TMC501	06	4A	05	43	06	9A	99	92	76	15	07	42	01	85	53	43	03	
1224	TMC501	65	43	04	54	42	07	95	42	02	76	33	01	44	01	44	02	43	
1241	TMC501	05	65	73	01	85	43	06	65	73	02	95	72	01	97	07	33	01	
1258	TMC501	92	76	1A	12	92	76	45	73	07	99	92	01	44	07	61	45	76	
1275	TMC501	17	85	32	43	03	65	53	43	04	85	01	54	85	0A	95	42	07	
1292	TMC501	32	98	92	76	60	72	07	32	01	44	07	32	99	92	61	60	76	
1309	TMC501	18	43	03	75	32	08	95	94	42	01	76	52	43	03	65	53	43	
1326	TMC501	04	42	07	85	01	54	85	08	95	42	02	00	76	59	85	43	03	
1343	TMC501	44	01	01	44	02	73	01	65	73	02	97	07	59	95	4A	03	44	
1360	TMC501	01	4A	03	72	01	32	75	01	95	42	01	29	67	25	94	85	0A	
1377	TMC501	95	4A	01	32	61	52	76	25	01	92	76	19	85	32	07	85	43	
1394	TMC501	03	65	43	04	95	42	07	32	9A	92	76	57	73	07	99	92	01	
1411	TMC501	44	07	61	57	76	10	43	01	4A	03	42	01	43	02	4A	04	42	
142A	TMC501	02	00	92	76	11	29	70	4A	02	4A	01	43	02	24	92	76	1A	
1445	TMC501	4A	04	4A	03	43	04	24	92	76	12	43	04	44	02	43	02	32	
1462	TMC501	43	03	44	01	43	01	92	76	17	01	94	49	03	49	04	12	92	
1479	TMC501	76	13	53	43	01	65	43	03	75	43	02	65	43	04	54	32	53	
1496	TMC501	43	01	65	43	04	85	43	02	65	43	03	54	42	02	32	42	01	
1513	TMC501	92	76	1A	01	94	49	04	53	43	03	33	85	43	04	33	54	35	
1530	TMC501	49	01	49	02	13	92	76	14	10	36	05	12	29	67	10	36	05	
1547	TMC501	16	13	36	05	17	92	76	19	36	05	16	10	36	05	1A	10	1A	
1564	TMC501	92	76	15	10	36	05	12	29	67	10	36	05	16	1A	36	05	17	
1581	TMC501	92	76	10	53	53	43	02	22	23	85	35	54	55	02	54	70	92	
159A	TMC501	76	11	36	04	11	92	76	12	70	43	01	32	43	02	22	37	32	
1615	TMC501	92	76	1A	12	23	42	01	32	42	02	32	92	76	17	70	43	01	
1632	TMC501	22	23	32	43	02	37	42	02	32	42	01	92	76	13	12	33	53	
1649	TMC501	32	65	76	52	02	54	37	42	02	32	42	01	92	76	10	12	34	
1666	TMC501	53	32	55	61	52	76	15	01	42	03	00	42	00	36	00	10	36	
1683	TMC501	04	1A	92	76	1A	53	53	43	02	22	23	75	35	54	55	02	54	
1700	TMC501	92	76	1A	70	53	53	43	01	85	01	54	33	85	43	02	33	54	
1717	TMC501	34	92	76	10	53	53	43	01	75	01	54	33	85	43	02	33	54	
1734	TMC501	34	92	76	15	53	53	16	85	10	54	55	02	85	53	52	22	52	
1751	TMC501	33	75	01	54	34	54	23	92	76	11	36	04	11	92	76	12	70	
176A	TMC501	53	43	01	3A	65	36	05	10	54	32	53	43	01	39	65	36	05	
1785	TMC501	1A	54	42	02	32	42	01	92	76	13	70	53	43	01	39	45	36	
1802	TMC501	05	10	54	32	53	43	01	38	94	65	36	05	1A	54	42	02	32	
1819	TMC501	42	01	92	76	14	43	01	42	03	43	02	42	04	13	36	04	10	
183A	TMC501	12	36	04	1A	92	76	17	53	53	16	75	10	54	55	02	54	22	
1853	TMC501	3A	32	15	42	02	32	42	01	92	76	1A	53	53	16	75	10	54	
1870	TMC501	55	02	54	22	39	32	15	94	42	02	32	42	01	92	76	19	53	
1887	TMC501	53	01	75	43	01	33	75	43	02	33	54	32	53	02	65	43	01	
1904	TMC501	54	22	37	55	02	54	32	53	53	53	43	01	33	85	53	43	02	
1921	TMC501	85	01	54	33	54	55	53	43	01	33	85	53	43	02	75	01	54	
193A	TMC501	33	54	54	23	55	04	54	42	02	32	42	01	92	76	11	42	04	
1955	TMC501	99	92	76	12	53	24	85	32	05	54	42	01	32	9A	92	76	50	
1972	TMC501	72	01	32	01	44	01	32	99	92	61	50	76	13	9A	99	42	03	
1989	TMC501	53	43	04	42	02	85	05	54	42	01	01	44	02	73	01	76	60	
2006	TMC501	22	97	02	70	22	97	01	70	53	24	65	43	03	85	73	01	54	
2023	TMC501	61	60	76	70	99	92	76	11	42	01	92	76	12	53	42	02	75	
2040	TMC501	32	43	01	54	42	03	93	00	01	42	0A	32	92	76	13	42	05	

TABLE VII

BCD ADDRESS	CHIP	PROGRAM CODES→															
2057	TMC541	92	76	14	42	08	92	76	15	53	24	43	02	32	43	01	77	50
2074	TMC541	42	04	A5	43	03	54	42	01	42	05	36	00	16	42	07	53	43
2091	TMC541	04	42	06	36	00	16	29	A7	70	A5	4A	07	54	77	15	76	A0
2108	TMC541	53	53	43	04	A5	43	05	54	55	02	54	42	06	43	0A	32	53
2125	TMC541	43	05	75	43	04	54	22	77	70	53	43	0A	36	00	16	A5	43
2142	TMC541	07	54	29	A7	70	77	45	43	06	42	05	A1	A0	76	45	43	06
2159	TMC541	42	04	61	A0	76	70	43	06	92	76	50	00	35	92	76	15	53
2176	TMC541	43	01	A5	43	05	A5	43	03	54	92	76	11	42	01	92	76	12
2193	TMC541	42	02	92	76	13	53	50	42	05	55	02	54	22	59	29	22	A7
2210	TMC541	52	53	43	05	35	A5	53	43	02	75	43	01	54	54	42	03	92
2227	TMC541	76	52	00	35	92	76	14	15	36	00	16	42	04	76	50	01	22
2244	TMC541	44	05	53	15	36	00	16	A5	04	54	44	04	22	97	05	A5	53
2261	TMC541	15	36	00	16	A5	02	54	44	04	A1	50	76	A5	15	36	00	16
2278	TMC541	44	04	53	43	03	55	03	54	49	04	43	04	92	76	11	53	50
2295	TMC541	42	05	55	02	54	42	02	22	59	29	22	A7	52	43	05	99	92
2312	TMC541	76	52	00	35	92	76	12	42	03	99	92	76	13	53	24	A5	32
2329	TMC541	06	54	42	01	32	98	92	76	50	72	01	32	01	44	01	32	99
2346	TMC541	92	A1	50	76	14	53	43	05	A5	06	54	42	01	73	01	42	04
2363	TMC541	76	A5	01	22	44	01	53	73	01	A5	04	54	44	04	01	22	44
2380	TMC541	01	22	97	02	33	53	73	01	65	02	54	44	04	A1	A5	76	33
2397	TMC541	73	01	44	04	53	43	03	55	03	54	49	04	43	04	98	99	92
2414	TMC541	76	16	53	53	43	06	33	A5	43	01	33	75	43	02	33	54	55
2431	TMC541	02	55	43	06	55	43	01	54	22	39	A7	00	97	42	05	A6	00
2448	TMC541	76	57	43	06	48	01	48	02	42	06	A1	16	76	97	A7	01	4A
2465	TMC541	42	04	A6	01	A1	57	76	6A	42	03	A7	02	90	76	10	22	A6
2482	TMC541	00	22	A6	01	22	A6	02	22	A6	03	92	76	14	53	53	43	02
2499	TMC541	34	65	43	01	55	43	06	54	22	52	52	22	52	22	3A	52	22
2516	TMC541	52	42	04	94	A5	01	94	22	39	75	43	02	54	42	02	42	05
2533	TMC541	A6	03	76	15	53	43	06	33	A5	43	01	33	75	02	65	43	06
2550	TMC541	A5	43	01	A5	43	02	39	54	34	A7	03	A8	42	02	A6	00	A6
2567	TMC541	02	A1	57	76	90	42	05	43	01	A1	10	76	A8	42	02	A1	10
2584	TMC541	76	17	43	04	92	76	18	43	05	92	76	11	42	06	92	76	12
2601	TMC541	42	01	92	76	13	42	02	92	76	11	42	07	92	76	12	42	04
2618	TMC541	92	76	13	42	05	92	76	16	53	01	94	22	39	75	43	04	75
2635	TMC541	43	05	54	42	03	A1	6A	76	17	53	01	94	22	39	75	43	04
2652	TMC541	42	03	75	43	05	54	42	04	A6	00	76	A8	53	43	07	A5	43
2669	TMC541	04	3A	55	43	03	3A	54	42	01	53	43	07	A5	43	05	3A	55
2686	TMC541	43	03	3A	54	42	02	A7	00	69	43	03	92	76	69	43	04	22
2703	TMC541	A6	00	92	76	14	43	01	92	76	15	43	02	92	76	1A	53	53
2720	TMC541	43	07	A5	43	01	A5	43	02	54	55	02	54	42	06	53	43	06
2737	TMC541	65	53	43	06	75	43	07	54	65	53	43	06	75	43	01	54	65
2754	TMC541	53	43	06	75	43	02	54	54	34	92	76	11	22	A6	00	22	A6
2771	TMC541	01	29	42	01	3A	77	4A	23	76	48	43	01	92	76	12	22	A6
2788	TMC541	00	22	A6	01	29	42	02	77	A8	23	43	02	76	A8	92	76	13
2805	TMC541	42	03	92	76	14	42	04	A6	00	A6	01	92	76	16	A7	00	57
2822	TMC541	53	43	03	55	43	02	54	42	01	92	76	57	70	53	53	43	04
2839	TMC541	55	02	55	43	02	54	22	3A	65	02	54	42	01	92	76	17	A7
2856	TMC541	01	4A	53	43	03	55	43	01	54	42	02	92	76	4A	53	71	69
2873	TMC541	35	A5	43	04	54	42	02	92	76	69	70	53	53	43	01	55	02
2890	TMC541	54	3A	65	02	54	92	76	1A	53	43	01	A5	43	02	54	92	76
2907	TMC541	19	71	69	A5	43	02	54	92	76	15	53	43	02	33	A5	43	01
2924	TMC541	55	02	54	92	76	10	70	53	43	01	A5	43	02	33	55	02	75
2941	TMC541	43	02	33	55	02	A5	43	01	3A	54	92	76	11	22	A6	01	29
2958	TMC541	77	A7	94	A6	01	7A	A7	53	42	03	33	22	23	A5	02	A5	A9
2975	TMC541	54	34	35	42	01	92	76	12	93	02	03	01	06	04	01	09	49
2992	TMC541	03	01	44	03	43	03	35	53	53	42	02	A5	04	A5	01	93	03
3009	TMC541	03	00	02	07	04	04	02	09	75	43	02	A5	03	A5	01	93	08
3026	TMC541	02	01	02	05	05	09	07	08	A5	43	02	A5	02	A5	01	93	07
3043	TMC541	08	01	04	07	07	09	03	07	75	43	02	A5	93	03	05	06	05
3060	TMC541	0A	03	07	08	02	A5	93	03	01	04	03	08	01	05	03	54	A5
3077	TMC541	43	02	A5	43	01	54	A7	01	A9	92	76	A9	53	94	A5	01	54
3094	TMC541	92	76	A8	53	53	02	04	02	09	08	A5	43	09	A5	09	09	09
3111	TMC541	09	01	54	55	01	09	09	00	01	07	42	07	54	53	53	53	22
3128	TMC541	59	A5	43	07	54	42	09	55	43	07	A5	05	22	28	54	59	55
3145	TMC541	05	22	28	54	92	76	13	71	A8	53	24	A5	53	43	11	75	43
3162	TMC541	10	54	76	37	A5	43	10	54	42	07	7A	43	07	92	76	1A	70
3179	TMC541	71	A8	42	08	71	A8	53	53	24	A5	02	A5	A9	54	39	A5	53
3196	TMC541	43	08	23	A5	02	94	54	34	A5	43	11	A1	37	76	10	36	01
3213	TMC541	71	25	92	76	15	42	09	92	76	11	42	10	92	76	12	42	11
3230	TMC541	92	76	13	29	A7	79	7A	A9	43	01	49	04	97	01	A9	7A	3A
3247	TMC541	43	04	A7	01	39	92	7A	14	29	A7	79	19	97	02	14	A1	3A
3264	TMC541	7A	15	29	A7	79	19	43	02	22	49	04	97	02	15	A1	3A	7A
3281	TMC541	79	01	42	04	A1	3A	7A	11	32	22	A6	01	01	42	04	42	03
3298	TMC541	32	7A	17	29	77	A7	A6	01	50	7A	A7	72	03	32	73	03	59
3315	TMC541	A7	A8	A6	01	7A	A8	72	03	92	76	39	00	35	43	04	92	76
3332	TMC541	19	43	01	49	04	01	22	44	01	92	7A	12	32	02	42	03	32
3349	TMC541	17	32	43	01	77	30	00	35	92	76	30	43	02	92	76	A7	43
3366	TMC541	05	72	01	44	04	01	44	01	53	43	04	55	32	54	92	76	12
3383	TMC541	42	05	A7	01	A9	01	44	03	43	03	32	43	02	77	A7	06	42
3400	TMC541	01	A6	01	7A	A9	73	01	22	44	04	43	05	44	04	72	01	01
3417	TMC541	44	01	43	01	32	53	43	02	A5	05	54	77	77	06	42	01	76
3434	TMC541	77	53	43	04	55	43	02	54	92	76	79	00	35	92	7A	10	22
3451	TMC541	A6	01	06	42	01	00	42	03	42	04	92	76	11	29	A7	79	22
3468	TMC541	77	79	42	02	32	43	02	59	22	A7	79	92	7A	16	53	53	43
3485	TMC541	09	A5	43	01	75	42	12	01	54	55	43	08	54	92	7A	17	53
3502	TMC541	16	A5	43	09	54	92	7A	18	53	16	55	43	12	54	92	7A	19
3519	TMC541	18	53	24	A5	43	09	54	92	76	10	29	58	02	92	76	71	53
3536	TMC541	43	04	55	43	03	54	92	76	25	53	71	71	21	55	43	09	23
3553	TMC541	54	42	01	92	76	11	A7	25	42	01	92	76	22	53	01	00	00
3570	TMC541	75	24	A5	71	71	22	A5	43	01	54	94	A7	15	76	12	A7	22

TABLE VIII

BCD ADDRESS	CHIP	PROGRAM CODES→																
3587	TMC541	76	15	42	02	53	24	55	01	00	00	85	42	08	01	54	42	09
3604	TMC541	43	02	92	76	23	53	43	04	55	43	09	45	43	01	54	42	03
3621	TMC541	92	76	13	67	23	42	03	92	76	24	53	43	03	65	43	09	45
3638	TMC541	43	01	54	42	04	92	76	14	67	24	42	04	92	76	11	61	03
3655	TMC541	78	76	12	61	02	92	76	13	61	04	43	76	14	61	05	01	76
3672	TMC541	15	61	05	39	01	42	09	07	94	22	28	32	53	43	04	55	43
3689	TMC541	03	54	42	10	87	03	01	36	87	04	01	36	53	24	55	43	01
3706	TMC541	33	75	43	10	35	54	42	08	44	09	36	18	16	42	14	87	01
3723	TMC541	00	80	53	24	65	43	09	54	53	24	75	43	10	54	02	13	53
3740	TMC541	43	14	55	43	08	75	43	01	65	43	12	55	43	09	55	43	08
3757	TMC541	54	87	01	01	20	53	24	65	43	09	75	43	14	54	22	49	13
3774	TMC541	43	13	44	08	44	09	50	77	00	65	61	02	80	53	43	05	55
3791	TMC541	43	03	85	42	14	43	01	49	14	75	43	10	54	42	08	53	43
3808	TMC541	01	85	33	55	02	85	43	14	55	43	01	54	87	03	01	84	53
3825	TMC541	43	11	85	33	55	02	85	43	14	54	22	49	08	43	08	44	09
3842	TMC541	36	18	18	42	06	87	03	02	06	53	24	65	43	09	54	53	24
3859	TMC541	85	43	14	55	43	01	55	43	12	75	43	10	54	42	13	53	43
3876	TMC541	06	55	43	08	75	43	01	55	43	12	55	43	09	55	43	08	54
3893	TMC541	87	03	02	55	53	24	65	43	09	85	43	06	54	53	24	85	43
3910	TMC541	14	55	43	12	55	43	09	54	22	49	13	43	13	44	08	44	09
3927	TMC541	50	77	01	91	53	43	08	65	01	00	00	54	29	67	02	95	67
3944	TMC541	00	25	58	04	36	18	15	92	68	43	03	42	11	42	10	43	09
3961	TMC541	87	02	03	19	87	04	03	19	01	49	11	49	10	53	43	04	65
3978	TMC541	43	08	54	87	01	03	54	87	02	03	54	94	44	10	53	43	05
3995	TMC541	65	43	08	54	94	44	11	61	03	56	44	11	43	11	23	42	11
4012	TMC541	43	10	23	94	44	11	43	09	23	35	49	11	43	11	67	03	81
4029	TMC541	67	03	01	42	01	94	42	11	94	58	09	68	92	68	87	01	04
4046	TMC541	22	87	02	04	16	87	03	04	10	36	18	19	61	04	25	36	18
4063	TMC541	18	61	04	25	36	18	17	61	04	25	36	18	16	53	24	55	53
4080	TMC541	43	04	75	43	05	55	43	12	54	54	35	67	04	46	67	03	91
4097	TMC541	42	03	58	02	68	92	68	87	01	04	83	87	02	04	77	87	03
4114	TMC541	04	71	36	18	19	61	04	86	36	18	18	61	04	86	36	18	17
4131	TMC541	61	04	86	36	18	16	53	24	65	43	03	85	43	05	55	43	12
4148	TMC541	54	67	05	04	67	04	52	42	04	58	02	68	92	68	36	18	19
4165	TMC541	87	04	05	21	36	18	18	53	53	24	65	43	03	75	43	04	54
4182	TMC541	94	65	43	12	54	67	05	42	67	05	10	02	05	58	02	68	92
4199	TMC541	76	16	86	01	92	76	17	86	02	92	76	18	86	03	92	76	19
4216	TMC541	86	04	92	76	10	22	86	01	22	86	02	22	86	03	22	86	04
4233	TMC541	00	42	05	58	09	29	92	76	78	53	93	04	65	43	01	85	02
4250	TMC541	93	03	54	59	94	85	61	77	76	79	00	35	92	76	10	53	42
4267	TMC541	01	29	22	77	79	22	59	22	44	01	65	04	22	28	58	02	03
4284	TMC541	32	01	05	08	01	77	79	03	02	32	53	53	43	01	55	01	00
4301	TMC541	00	54	42	01	22	59	22	44	01	65	01	00	00	54	42	02	77
4318	TMC541	79	01	03	32	43	01	77	79	53	03	06	05	65	43	03	85	43
4335	TMC541	02	85	03	01	65	43	01	75	03	01	85	03	32	43	01	77	78
4352	TMC541	01	22	44	03	76	77	53	43	03	55	04	54	59	75	53	93	07
4369	TMC541	05	85	53	43	03	55	01	00	00	54	59	65	93	07	05	54	59
4386	TMC541	54	92	76	11	10	42	08	00	92	76	12	10	42	05	00	92	76
4403	TMC541	13	53	43	05	75	43	04	54	92	76	14	53	53	10	42	01	94
4420	TMC541	55	07	54	59	65	07	85	43	01	54	92	76	13	32	01	44	04
4437	TMC541	53	32	75	43	03	54	42	05	53	35	50	65	43	05	54	92	76
4454	TMC541	18	94	76	17	32	01	22	44	02	53	02	45	43	02	85	32	54
4471	TMC541	50	92	76	12	53	36	15	71	88	65	01	00	02	03	85	01	54
4488	TMC541	59	42	03	00	42	04	92	76	14	24	43	04	92	76	16	09	42
4505	TMC541	02	05	01	02	98	92	76	19	53	01	00	75	43	02	54	92	76
4522	TMC541	11	42	09	92	76	10	87	00	45	05	61	44	76	45	06	74	44
4539	TMC541	42	10	73	10	92	76	13	94	76	12	53	24	85	10	54	72	10
4556	TMC541	92	76	11	10	92	76	16	22	76	17	86	00	58	02	92	76	18
4573	TMC541	42	07	92	76	19	53	35	65	43	07	54	29	36	18	12	92	76
4590	TMC541	14	29	36	18	11	43	06	42	03	29	36	18	71	24	42	06	92
4607	TMC541	76	15	87	00	34	42	05	92	76	34	42	06	92	76	12	53	88
4624	TMC541	85	76	15	43	01	54	53	24	85	53	24	55	50	54	24	65	05
4641	TMC541	94	22	28	54	22	88	58	04	92	76	13	53	24	65	61	15	76
4658	TMC541	14	53	35	65	61	15	76	11	58	09	88	42	01	92	76	11	53
4675	TMC541	24	65	02	93	05	04	54	92	76	12	53	24	65	93	03	00	04
4692	TMC541	08	54	92	76	13	53	24	65	93	09	01	04	04	54	92	76	14
4709	TMC541	53	24	65	01	93	06	00	09	03	04	04	54	92	76	15	53	24
4726	TMC541	65	93	08	06	08	09	07	06	02	04	54	92	76	16	35	11	35
4743	TMC541	92	76	17	35	12	35	92	76	18	35	13	35	92	76	19	35	14
4760	TMC541	35	92	76	10	35	15	35	92	76	11	53	53	24	75	03	02	54
4777	TMC541	65	05	55	09	54	92	76	12	53	24	65	93	00	02	09	05	07
4794	TMC541	03	05	02	09	06	54	92	76	13	53	24	65	03	93	07	08	05
4811	TMC541	04	01	01	07	08	04	54	92	76	14	53	24	65	02	08	93	03
4828	TMC541	04	09	05	02	03	01	03	54	92	76	15	53	24	65	93	04	05
4845	TMC541	03	05	09	02	03	07	54	92	76	16	53	24	65	01	93	08	85
4862	TMC541	03	02	54	92	76	17	35	12	35	92	76	18	35	13	35	92	76
4879	TMC541	19	35	14	35	92	76	10	35	15	35	92	61	11	92	92	92	92
4896	TMC541	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92
4913	TMC541	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92
4930	TMC541	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92
4947	TMC541	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92
4964	TMC541	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92
4981	TMC541	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92	92
4998	TMC541	92	92															

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 instructiion 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 money.

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 mircoprocessor 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.

* * * * *