

United States Patent [19]

Brautingham

[11]

4,189,779

[45]

Feb. 19, 1980

[54] PARAMETER INTERPOLATOR FOR
SPEECH SYNTHESIS CIRCUIT

[75] Inventor: George L. Brautingham, Lubbock,
Tex.

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

[21] Appl. No.: 901,394

[22] Filed: Apr. 28, 1978

[51] Int. Cl.² G01L 1/00; G06F 15/34

[52] U.S. Cl. 364/718; 179/1 SA;
179/1 SC; 179/1 SM; 179/15 AS; 364/513

[58] Field of Search 179/1 SA, 1 SC, 1 SM,
179/15 AS; 364/513, 718, 723

[56] References Cited

U.S. PATENT DOCUMENTS

3,974,334 8/1976 Cockerell 179/1 SA
3,982,070 9/1976 Flanagan 179/1 SM
4,076,958 2/1978 Fulghum 179/1 SA

Primary Examiner—Jerry Smith

Attorney, Agent, or Firm—William K. McCord; James T. Comfort; Melvin Sharp

[57] ABSTRACT

Disclosed is a parameter interpolator for a speech synthesis circuit. Using a parameter interpolator permits the data rate to the speech synthesis circuit to be lowered inasmuch as the incoming speech data is used to slowly charge the data previously inputted to the values of the incoming data. The speech synthesis circuit includes an input circuit for receiving the target values of the speech data and a memory for stored interpolated values of the speech data. The interpolator includes a circuit coupled to the input circuit and the memory which calculates the difference between the target values and the stored values. Another circuit is used to add a portion of the difference to the values stored in the memory; the particular portion of the difference is equal to $\frac{1}{2}N$ where $N=0, 1, 2 \dots$. Further, the interpolator is arranged to inhibit the normal interpolation upon certain conditions, such as changes from voiced speech to unvoiced speech, and visa versa.

8 Claims, 41 Drawing Figures

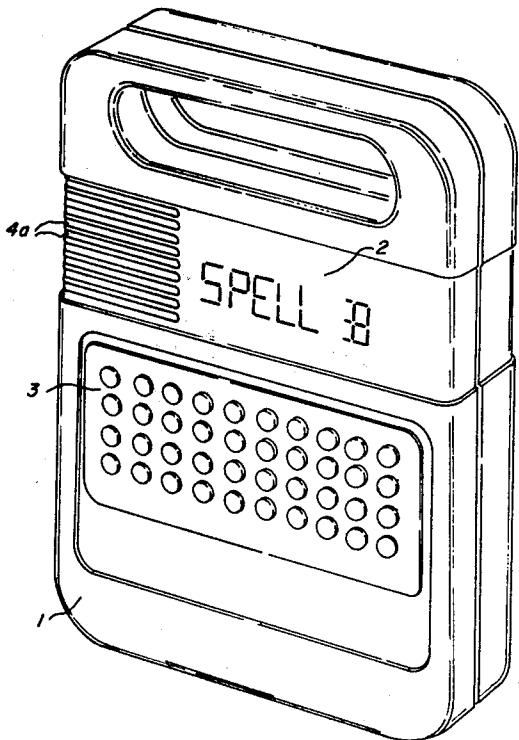


Fig. 1

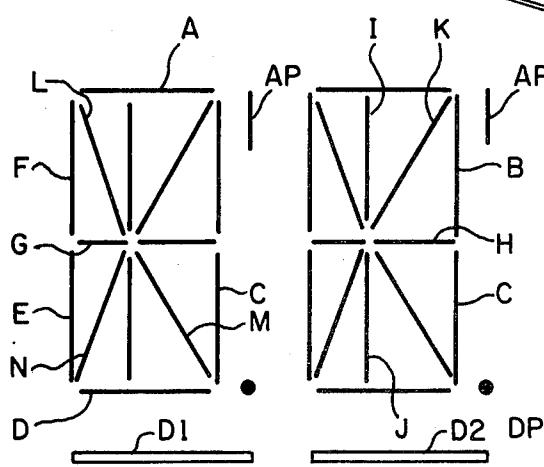
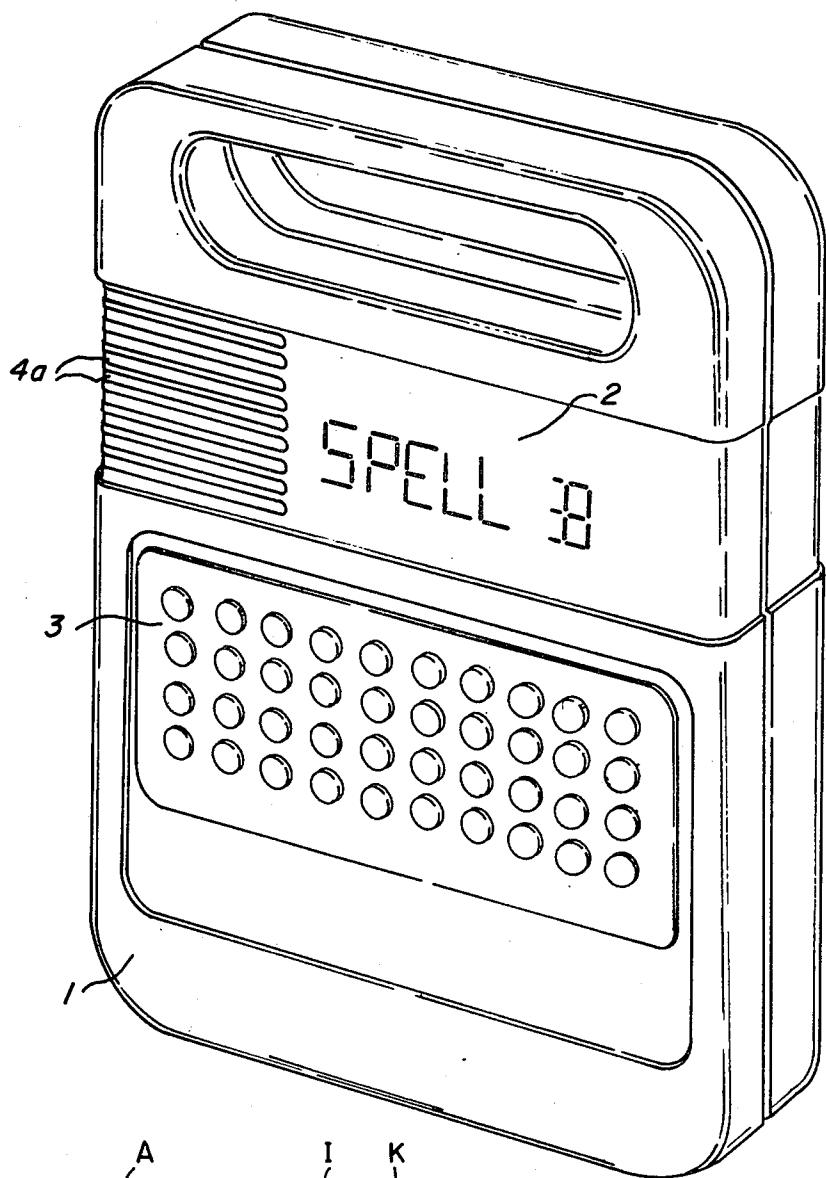


Fig. 2

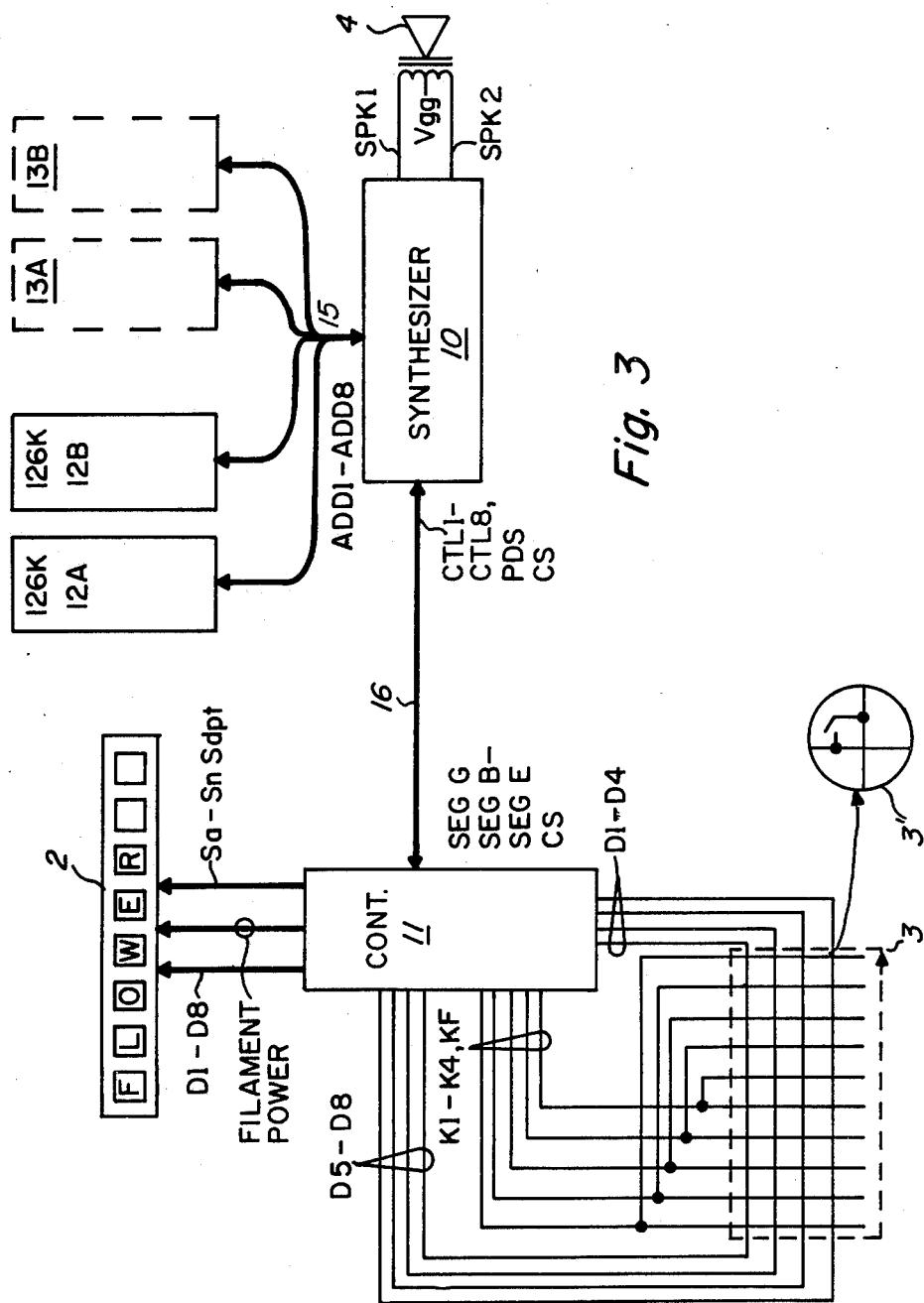


Fig. 3

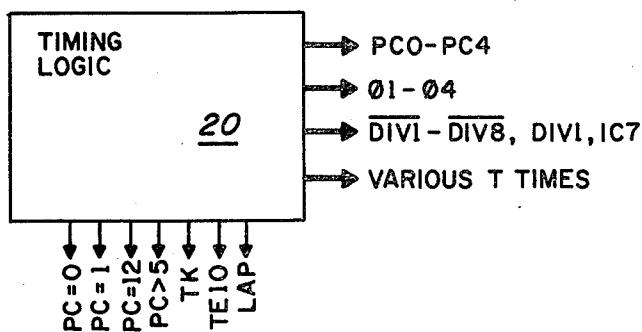
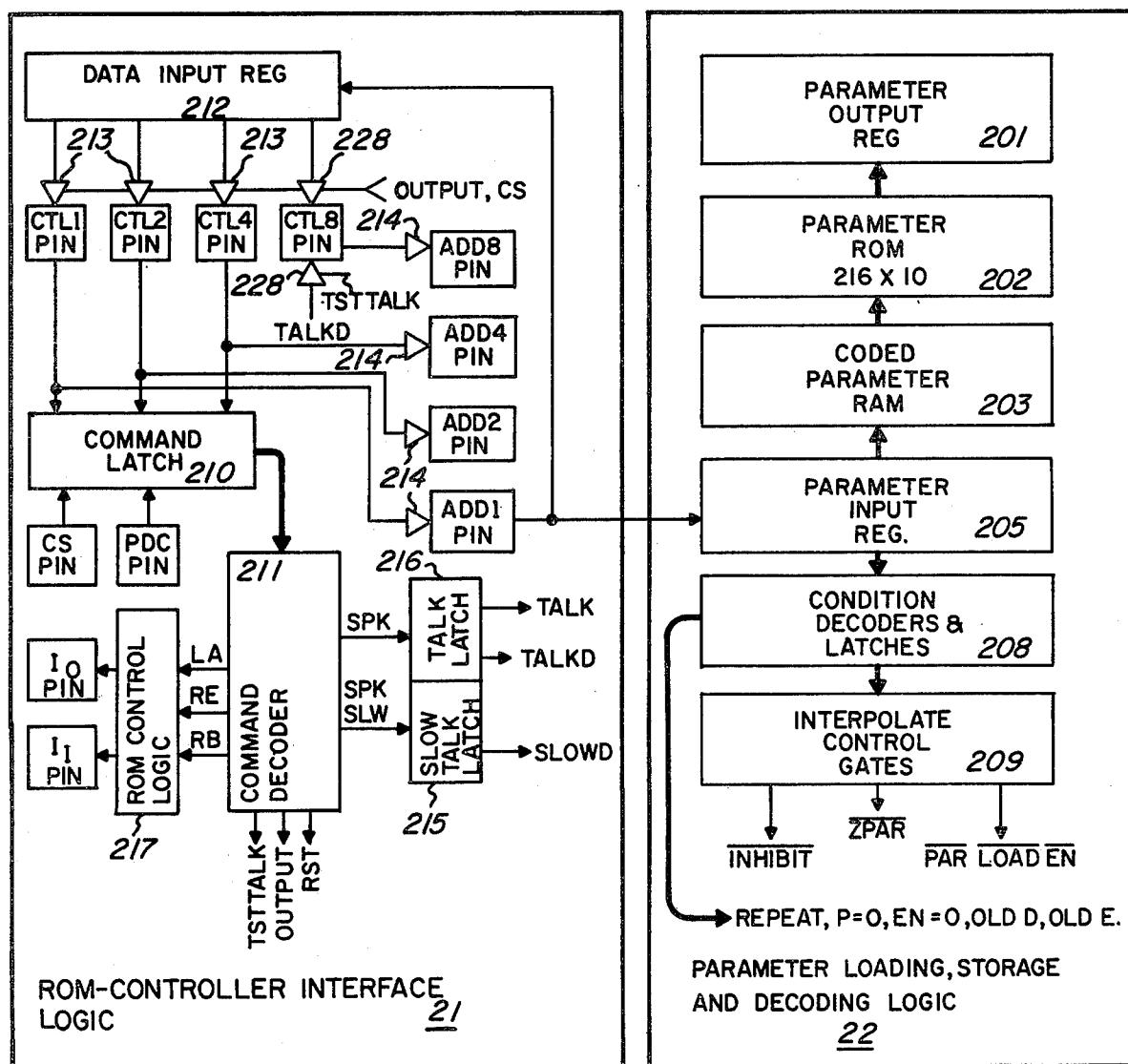
10

Fig. 4a



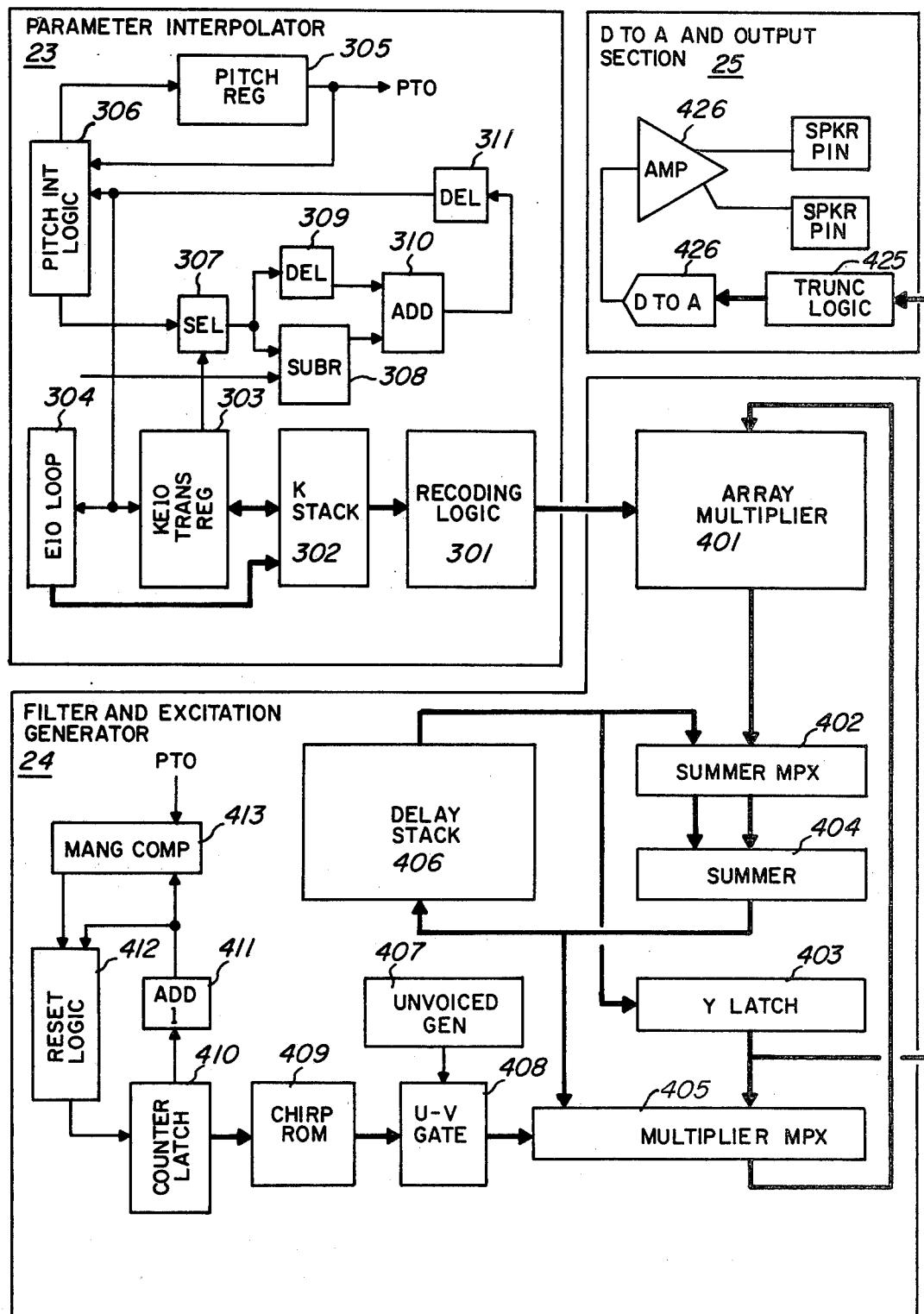
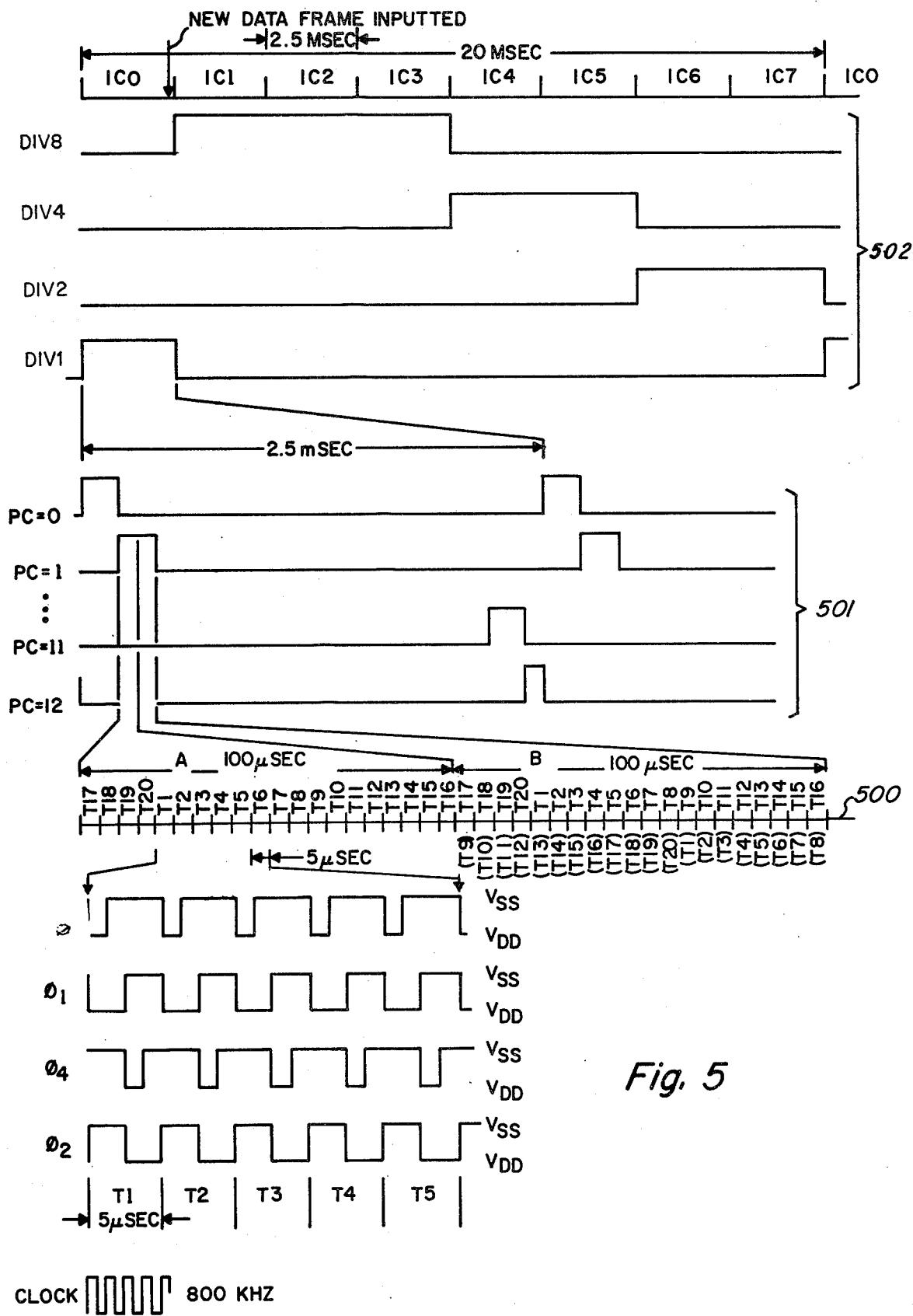


Fig. 4b



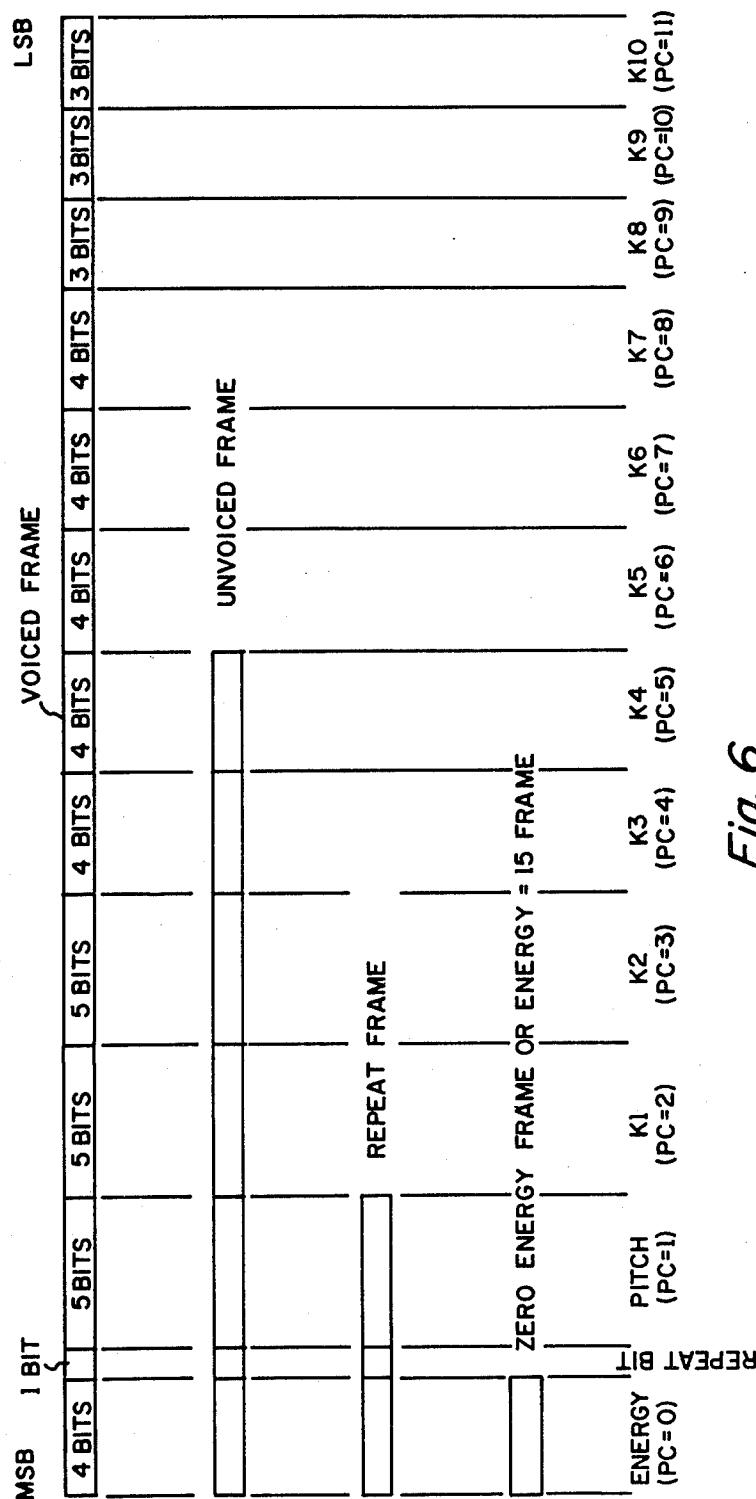
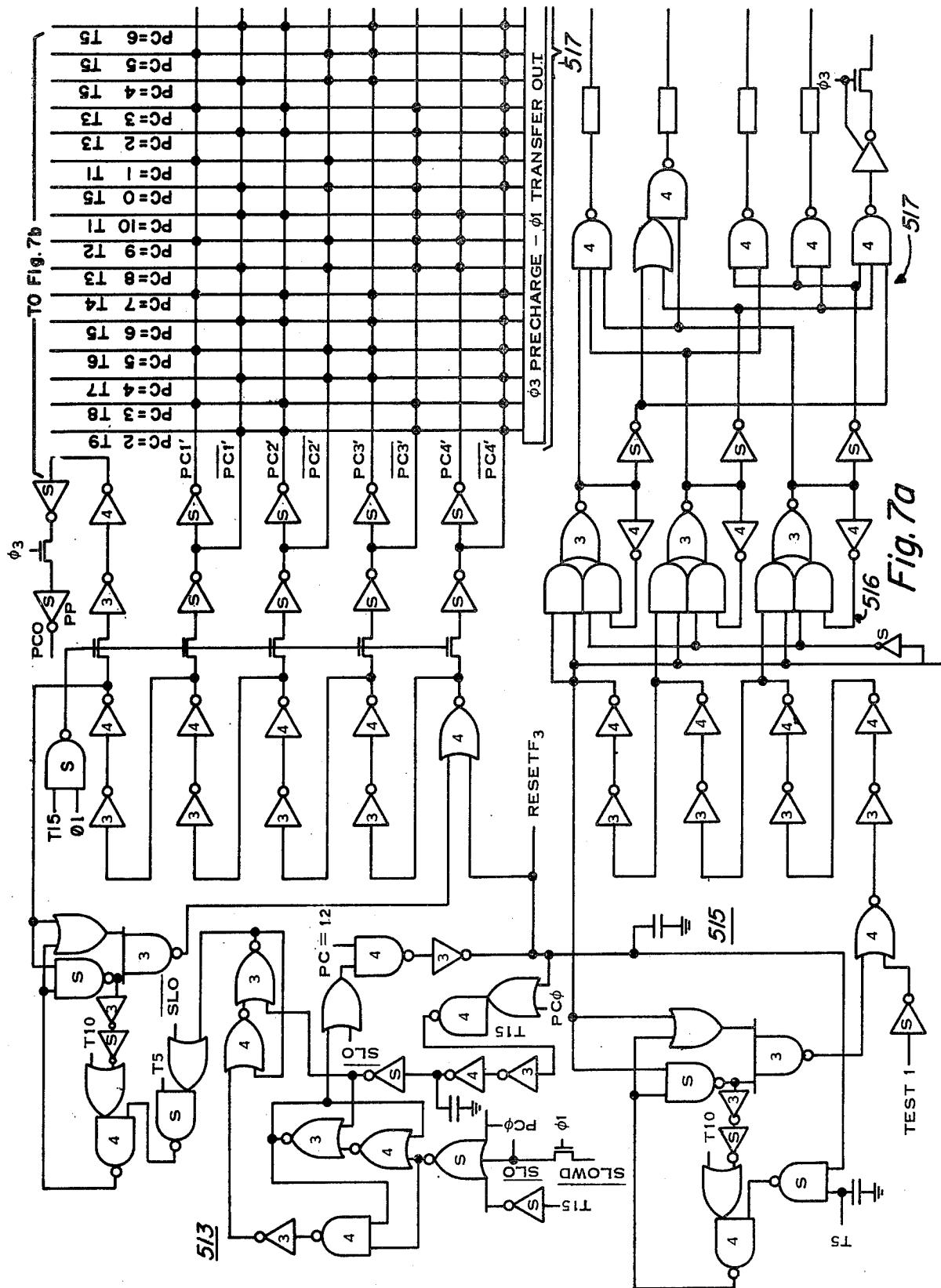
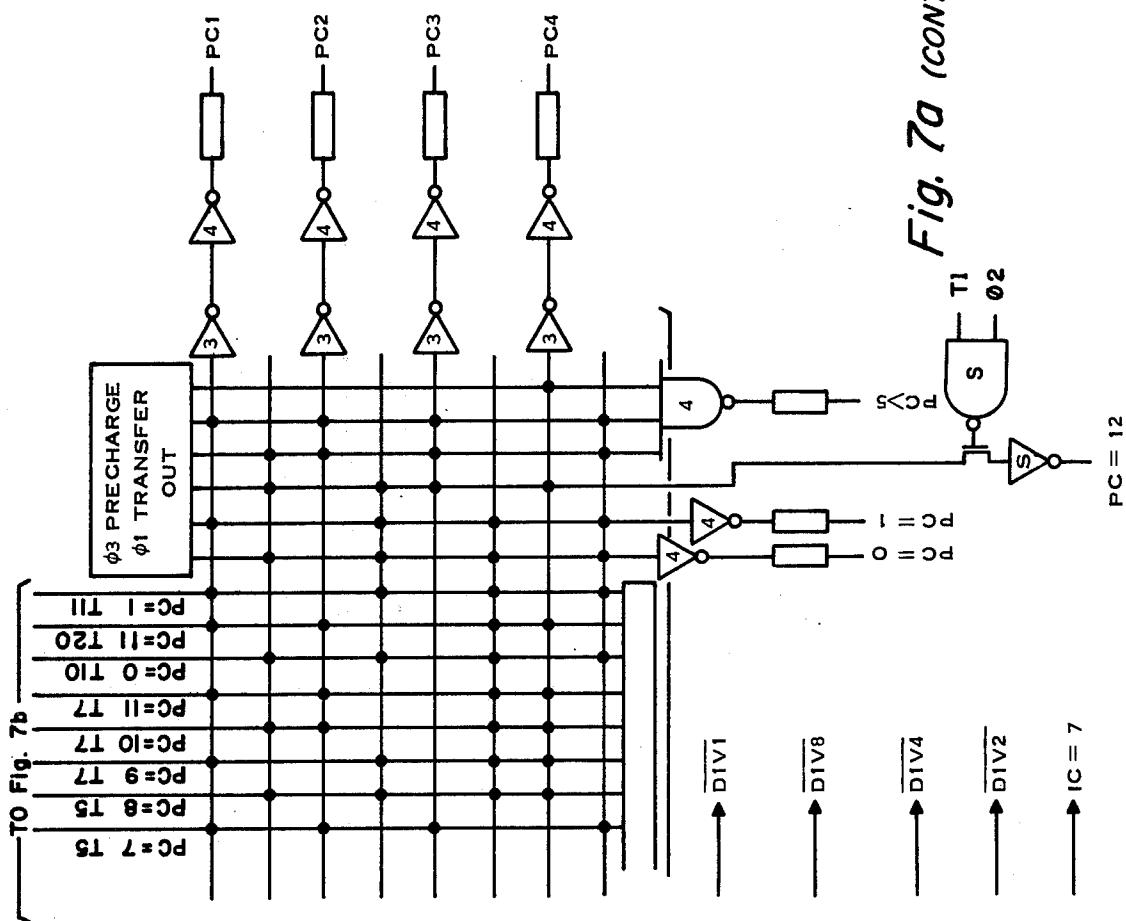


Fig. 6





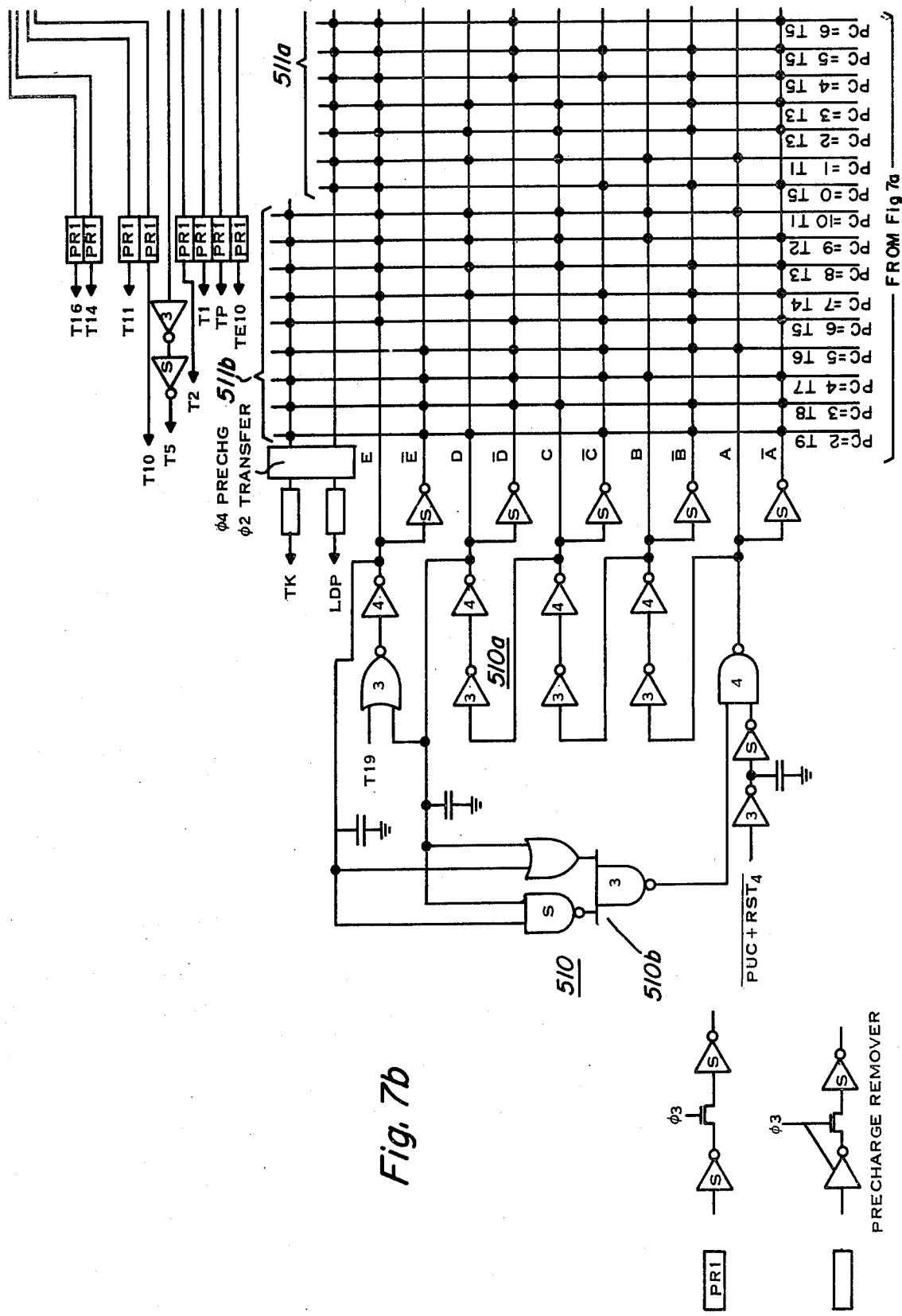
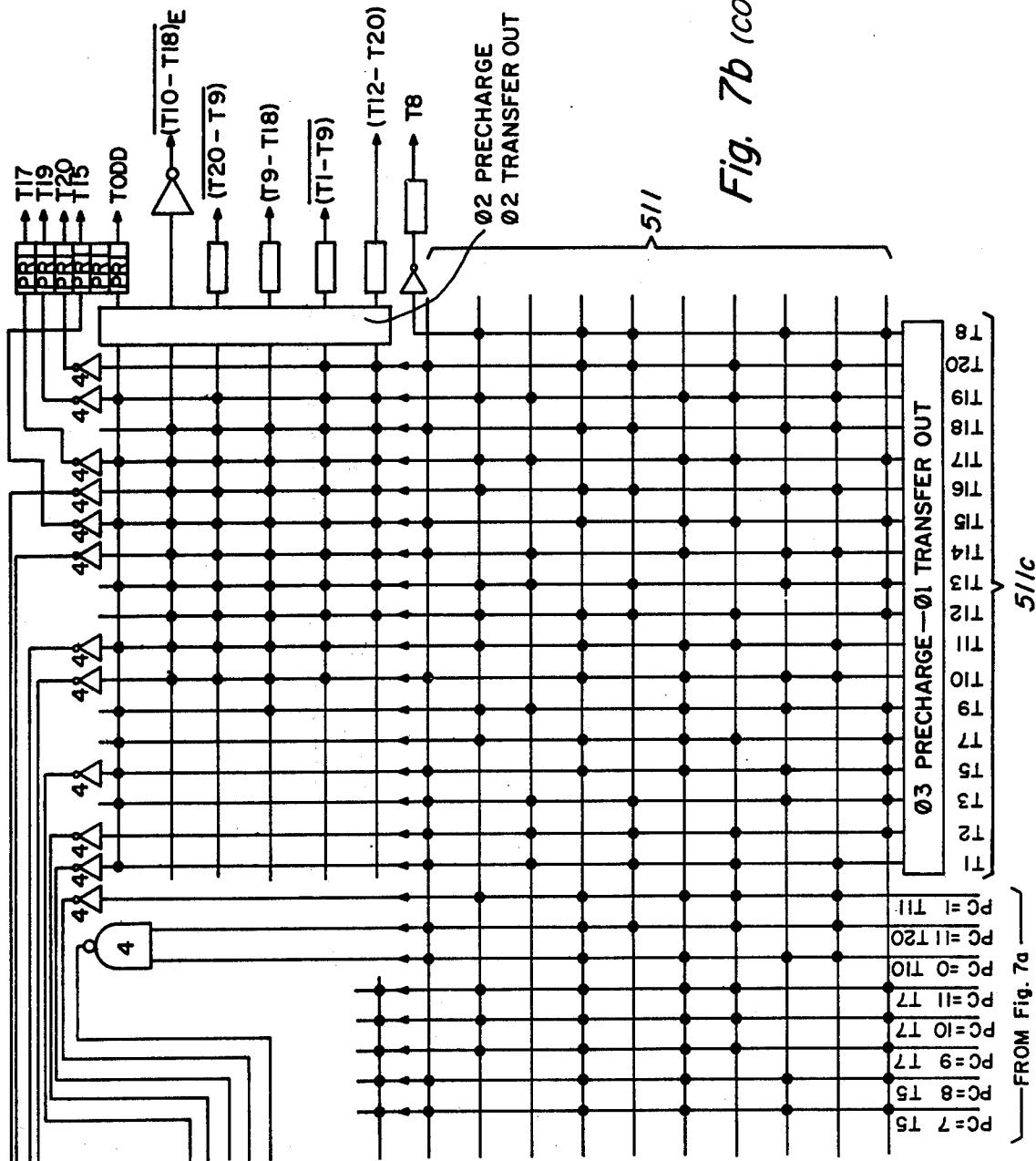


Fig. 7b (CONTINUED)



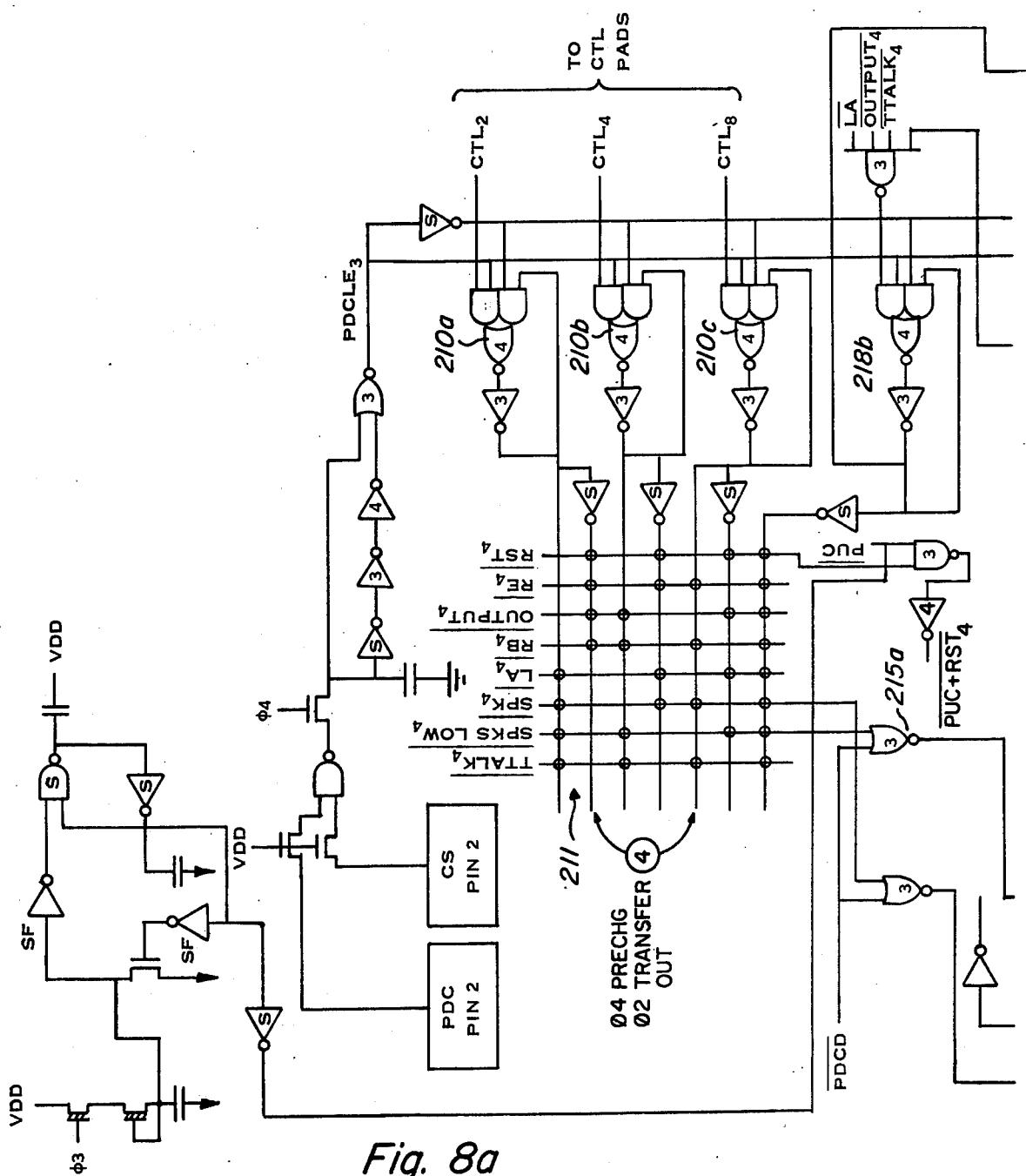
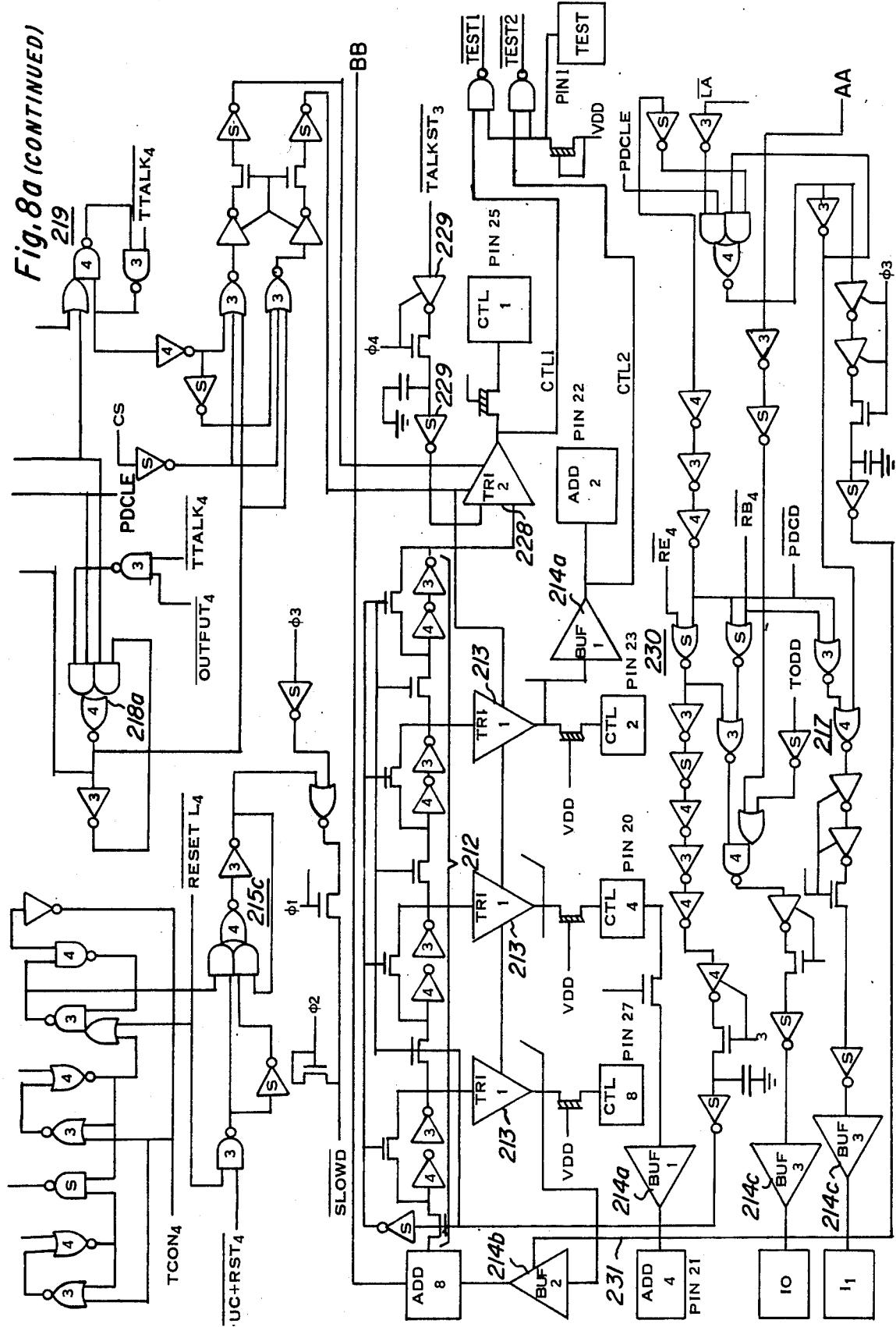


Fig. 8a

Fig. 8a (CONTINUED)



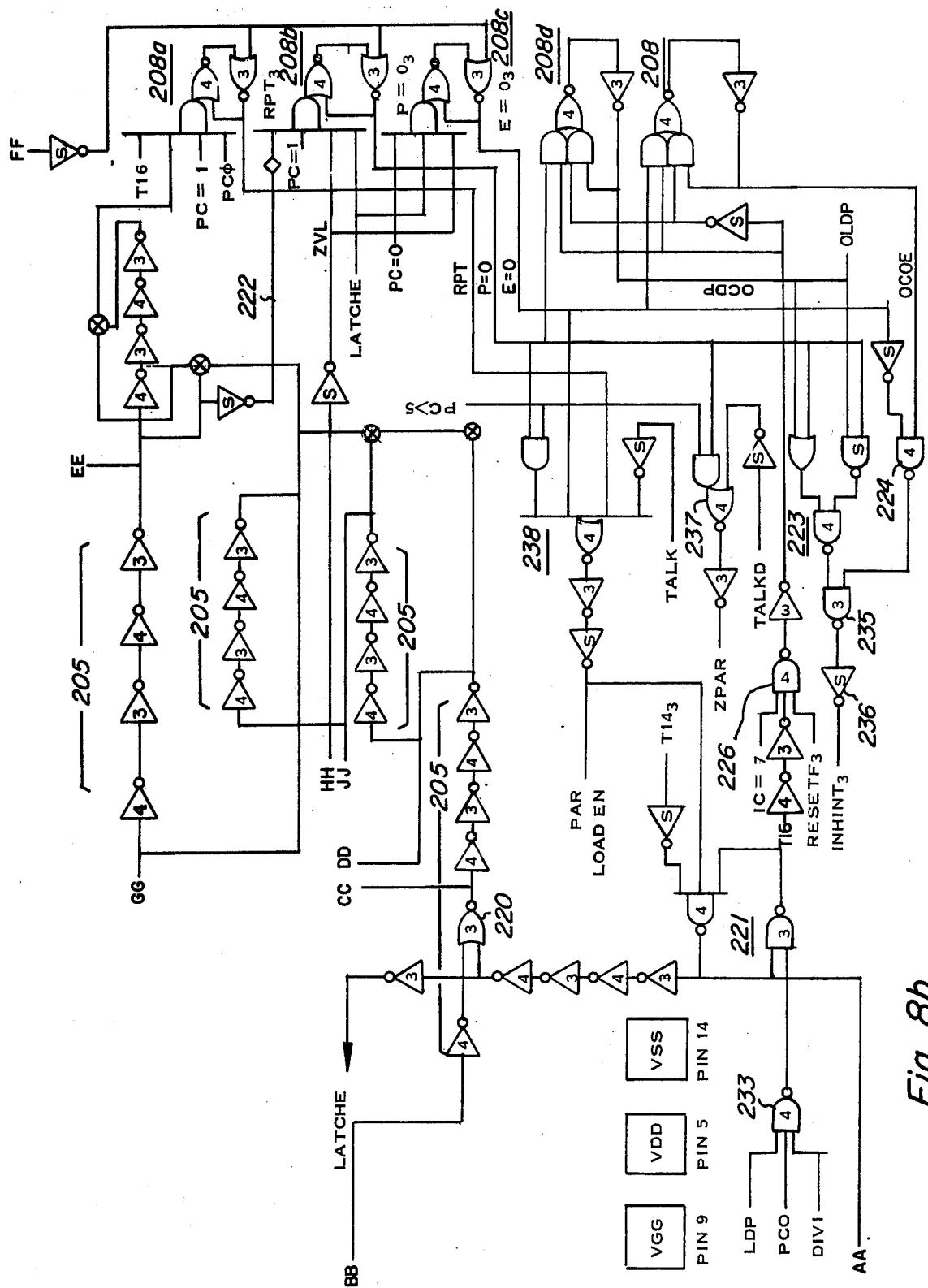


Fig. 8b

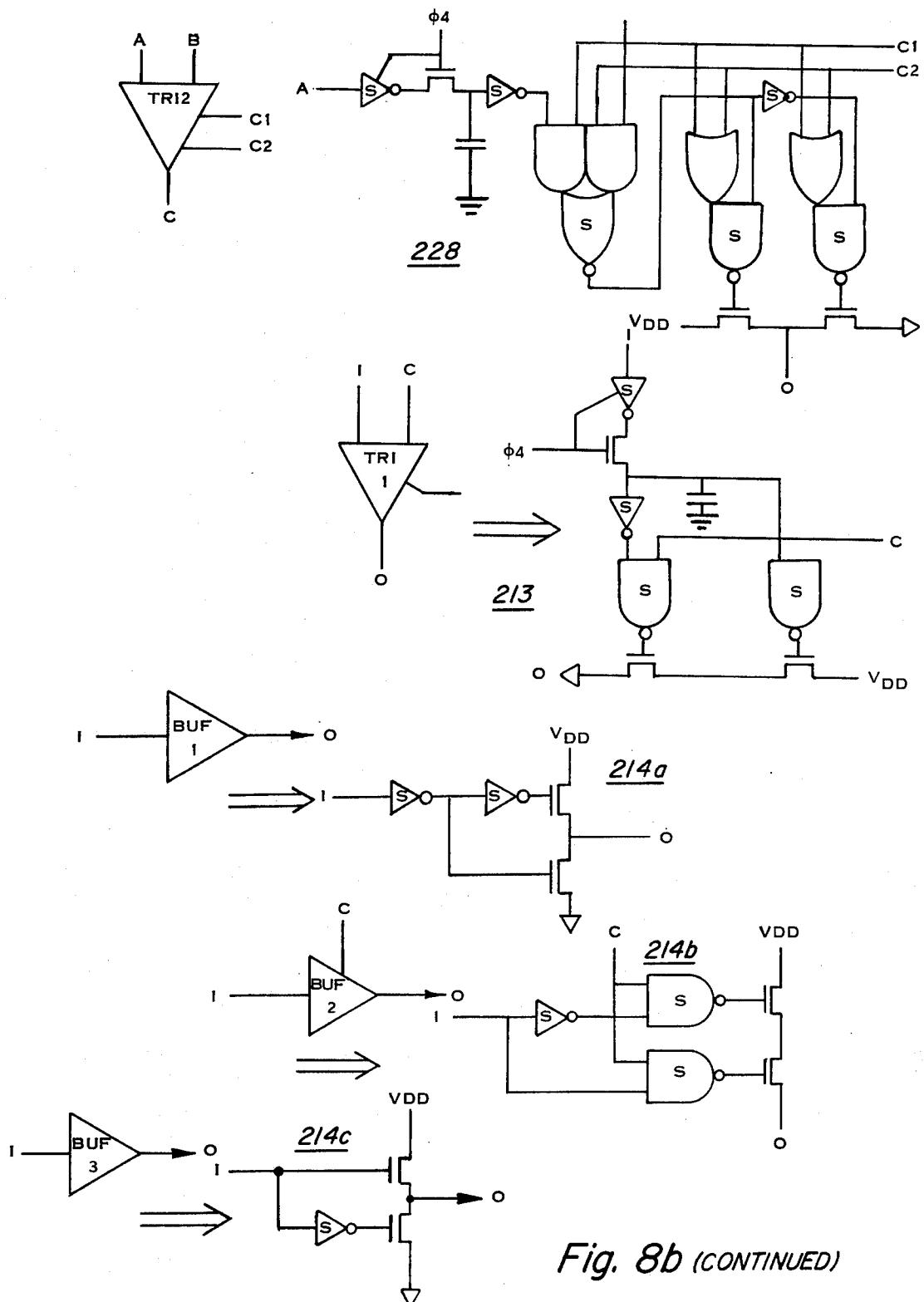


Fig. 8b (CONTINUED)

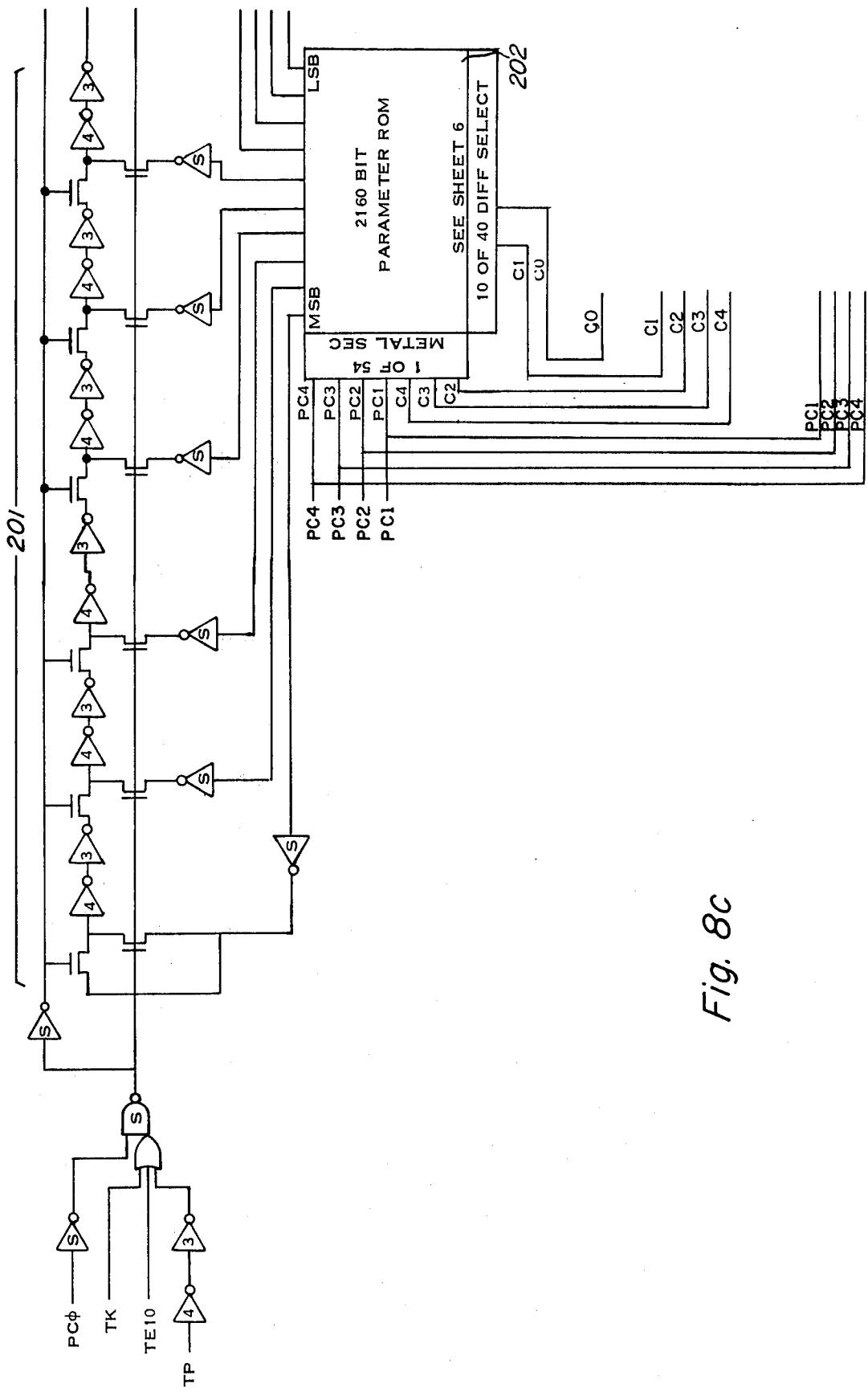
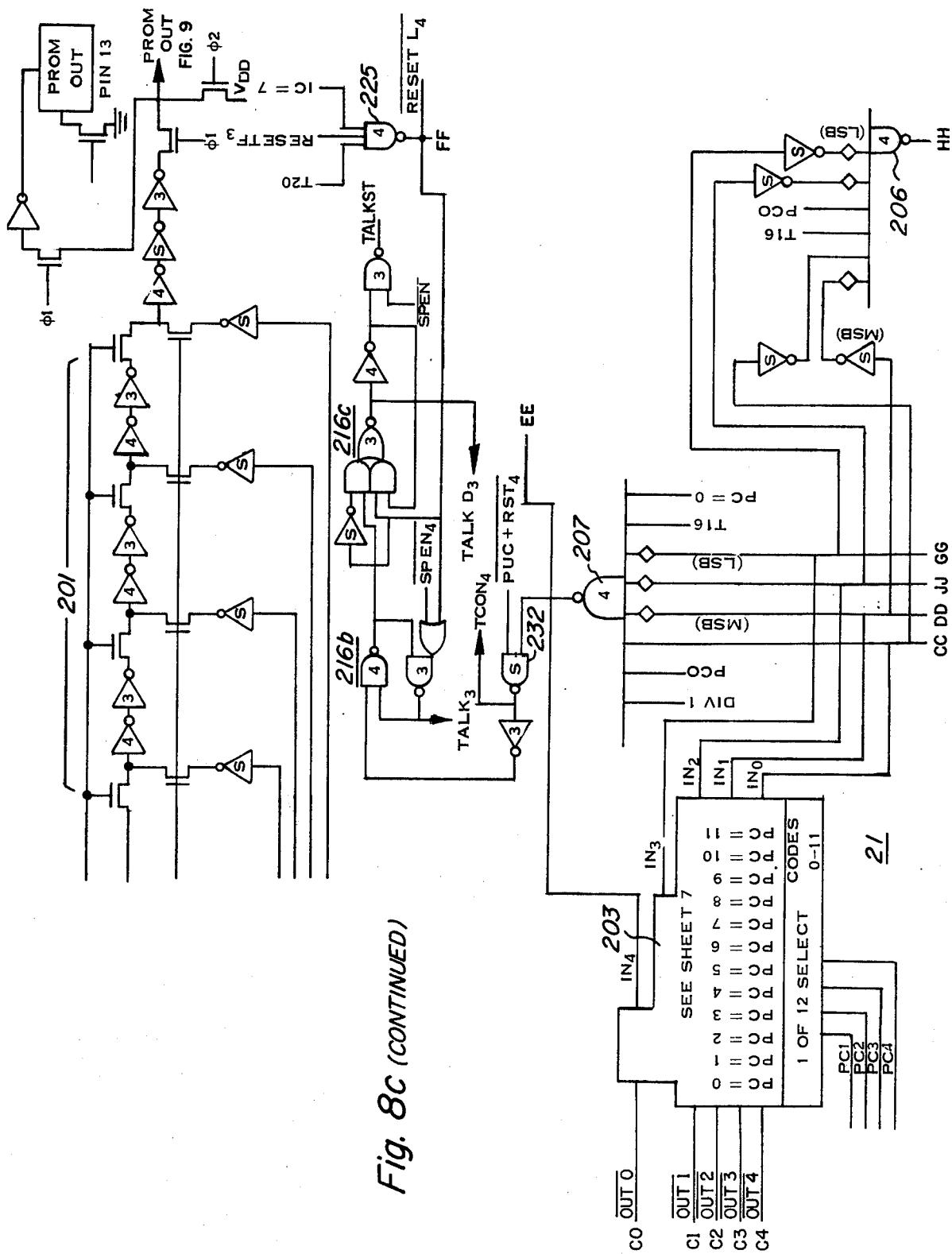


Fig. 8C



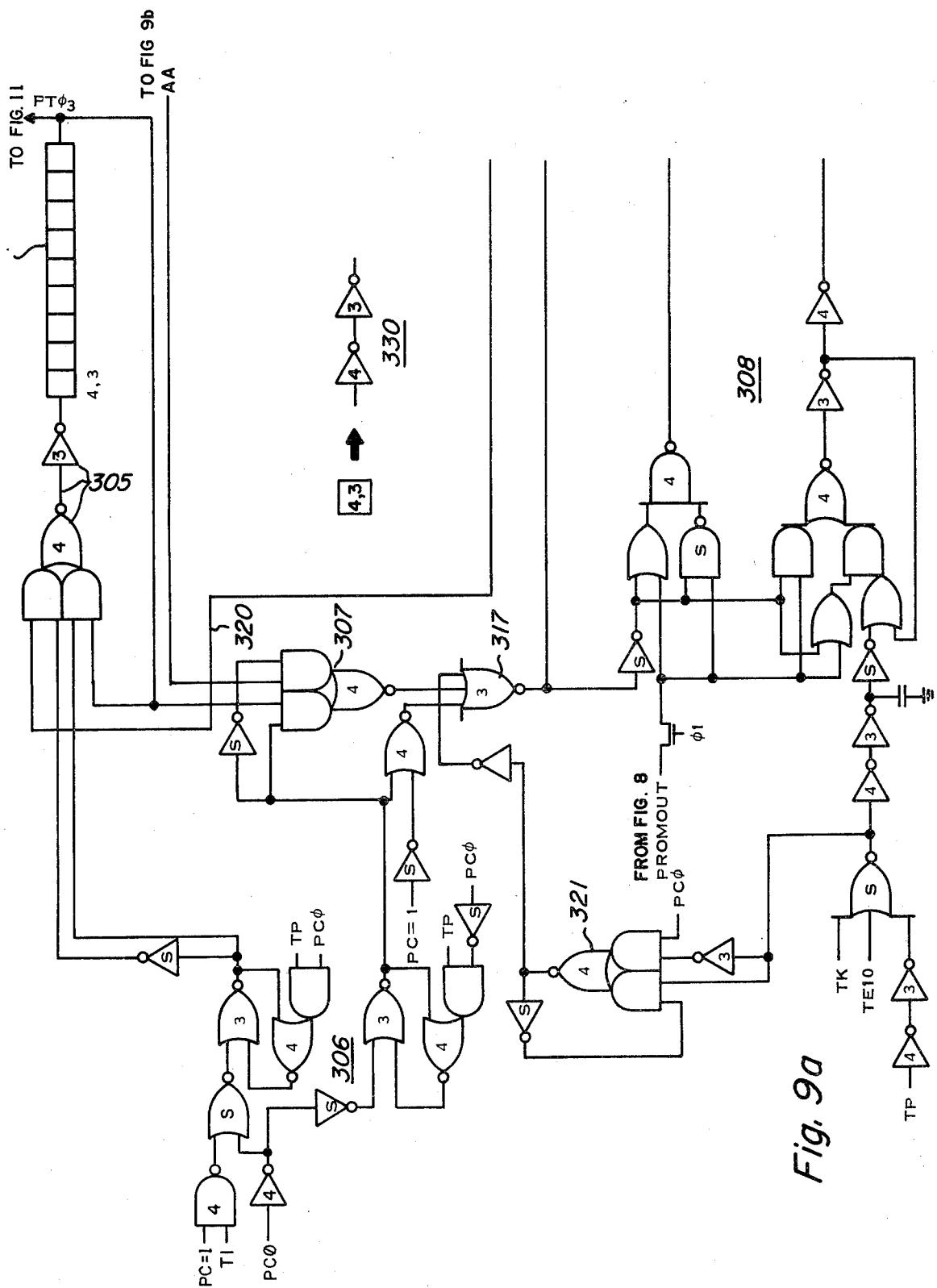


Fig. 9a

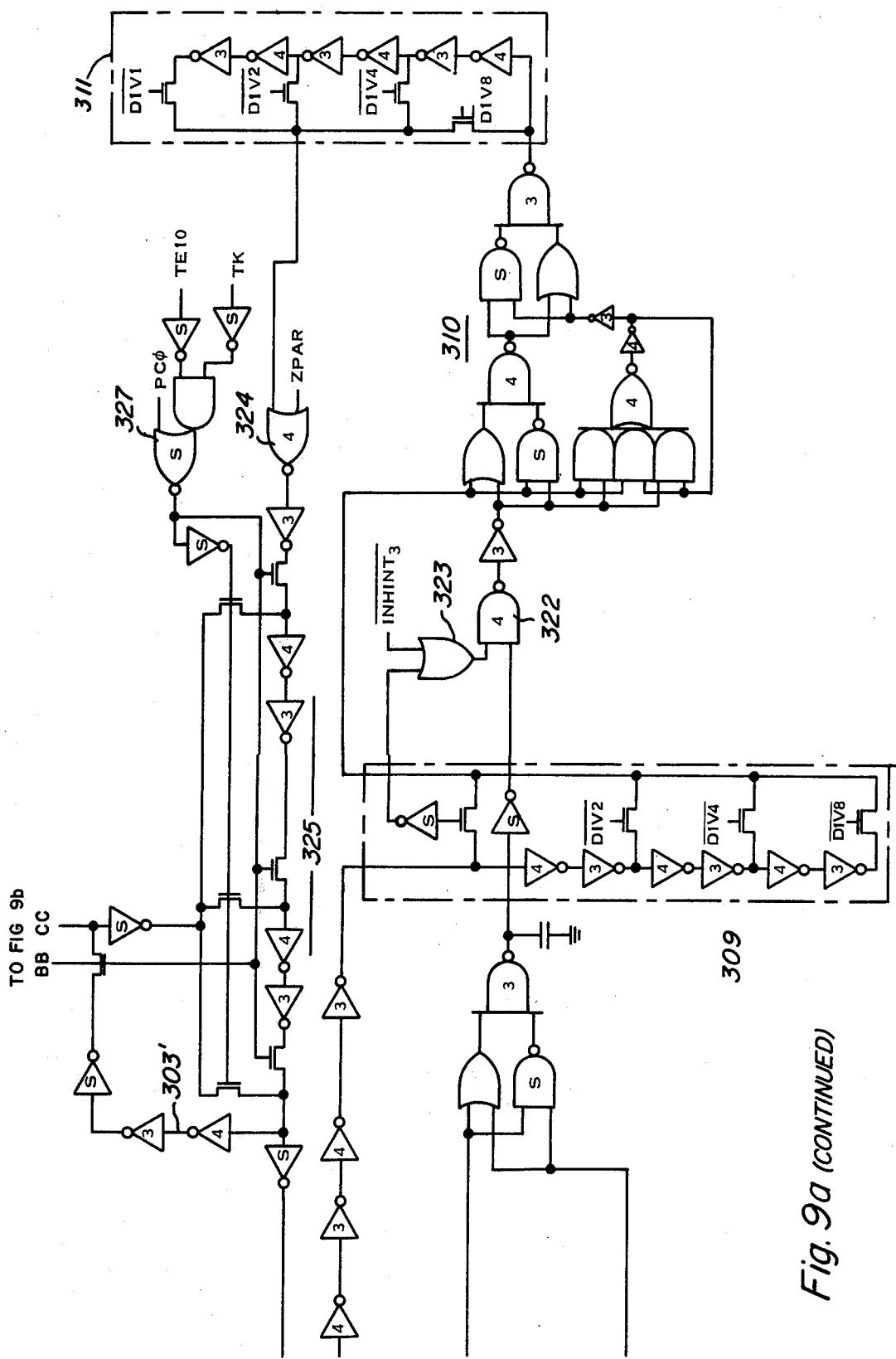


Fig. 9a (CONTINUED)

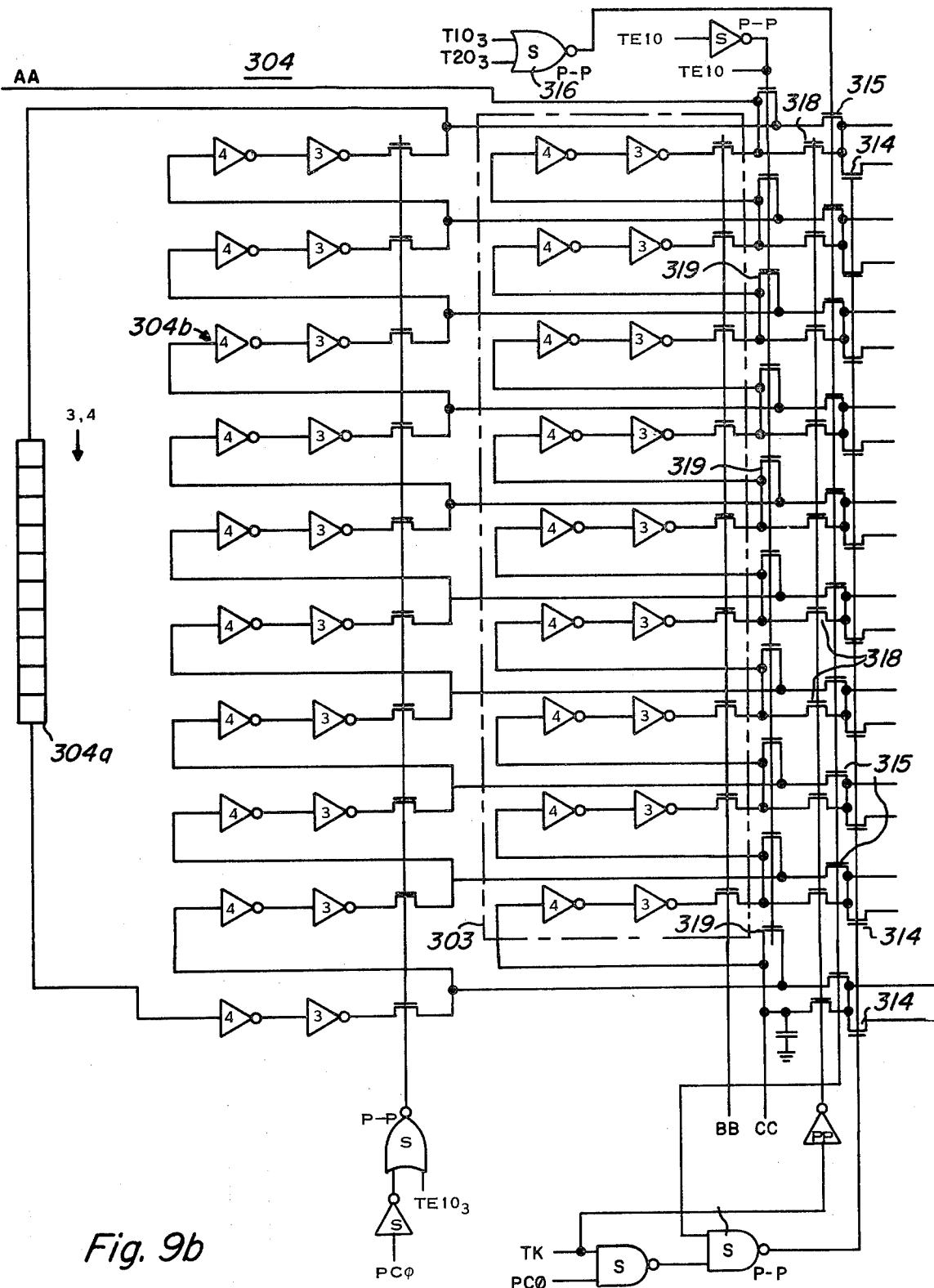
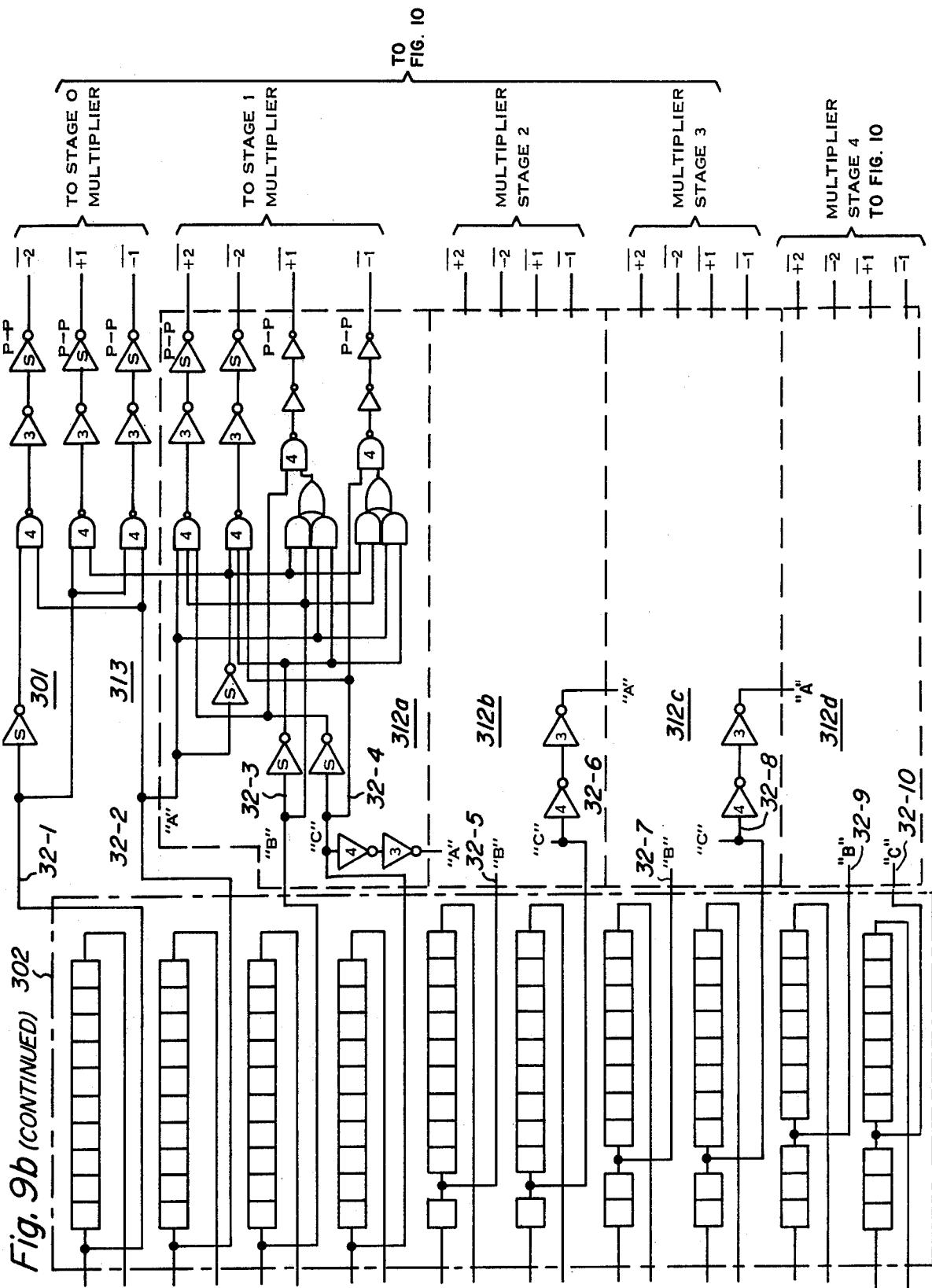


Fig. 9b



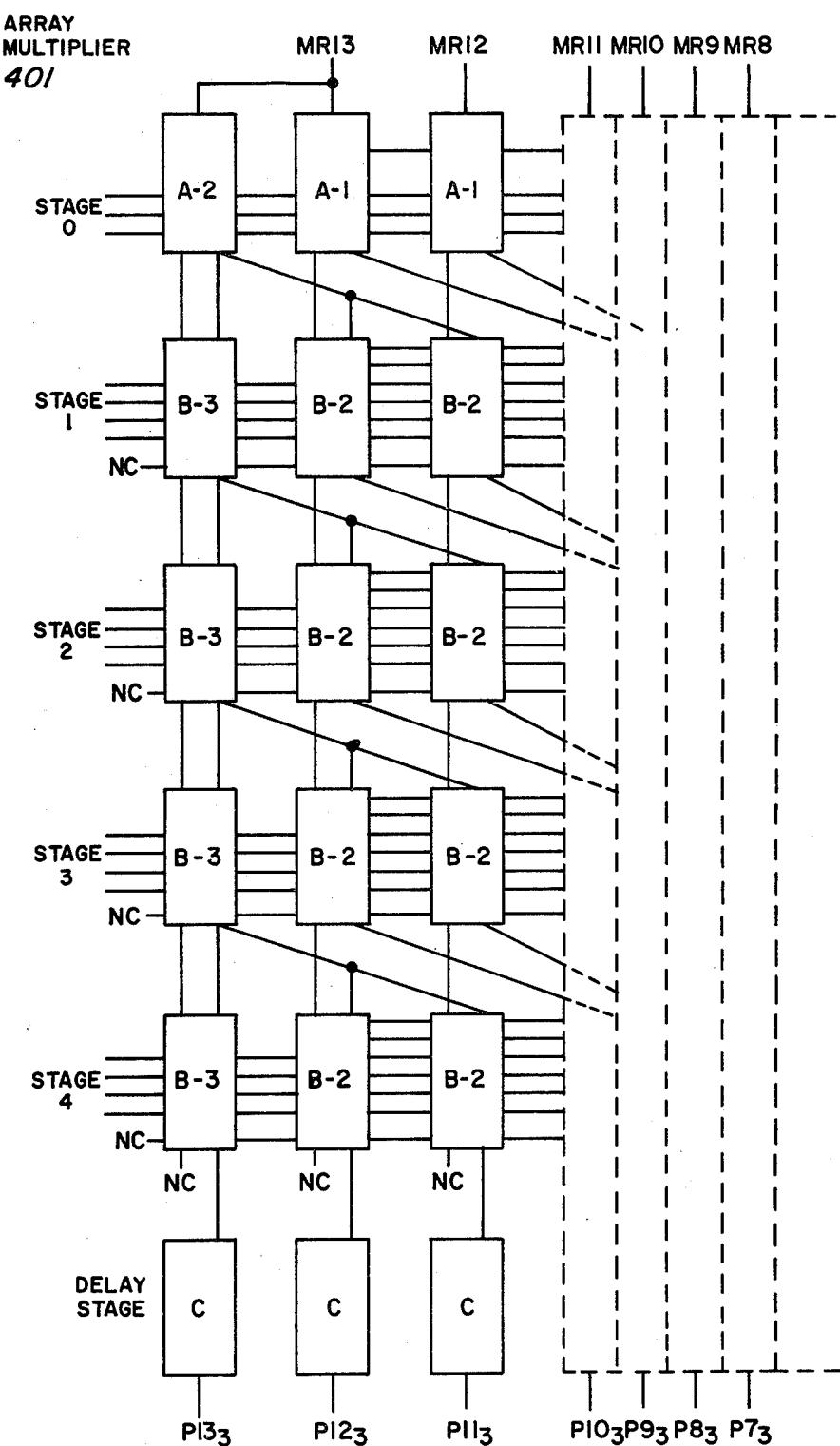


Fig. 10a

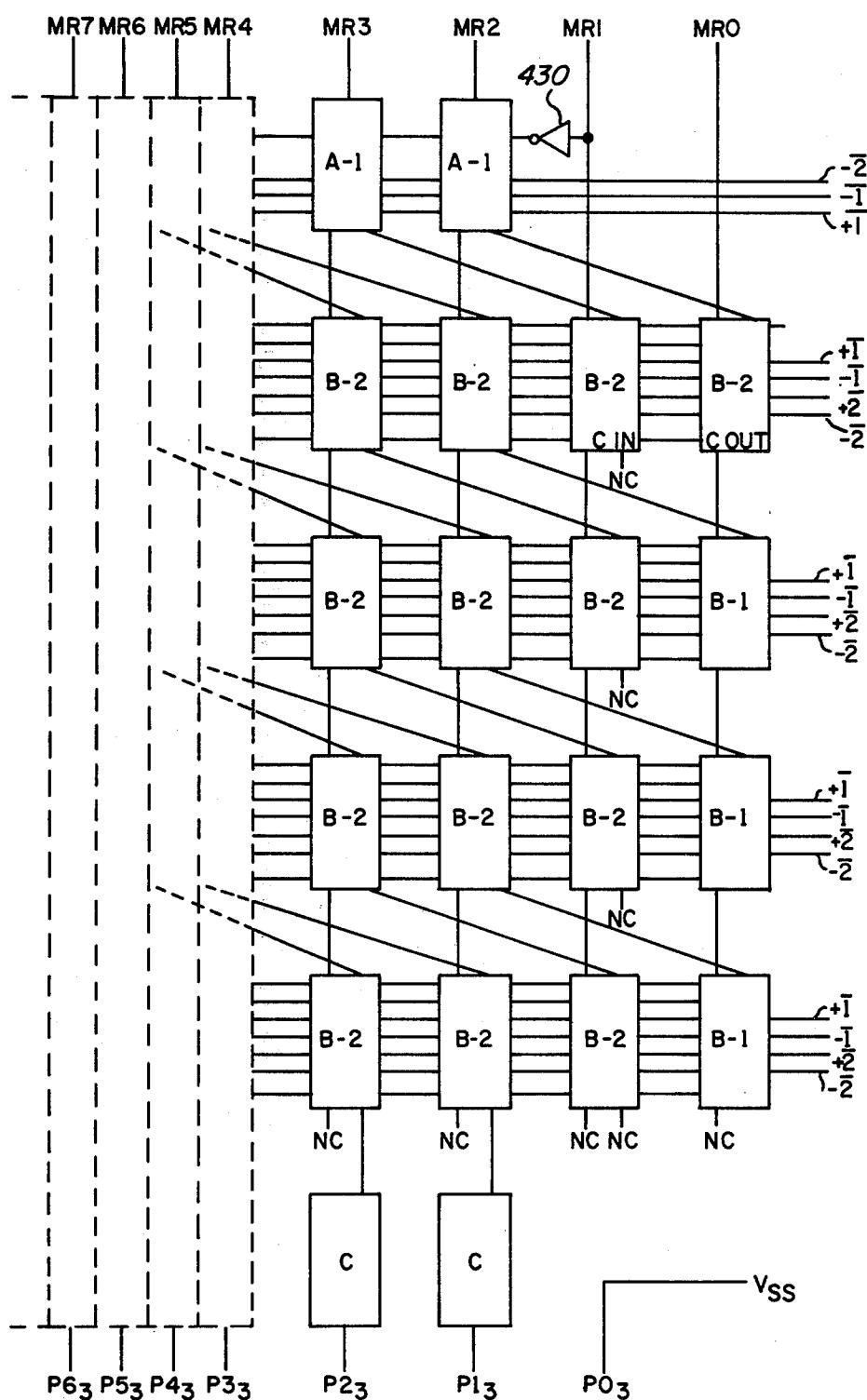
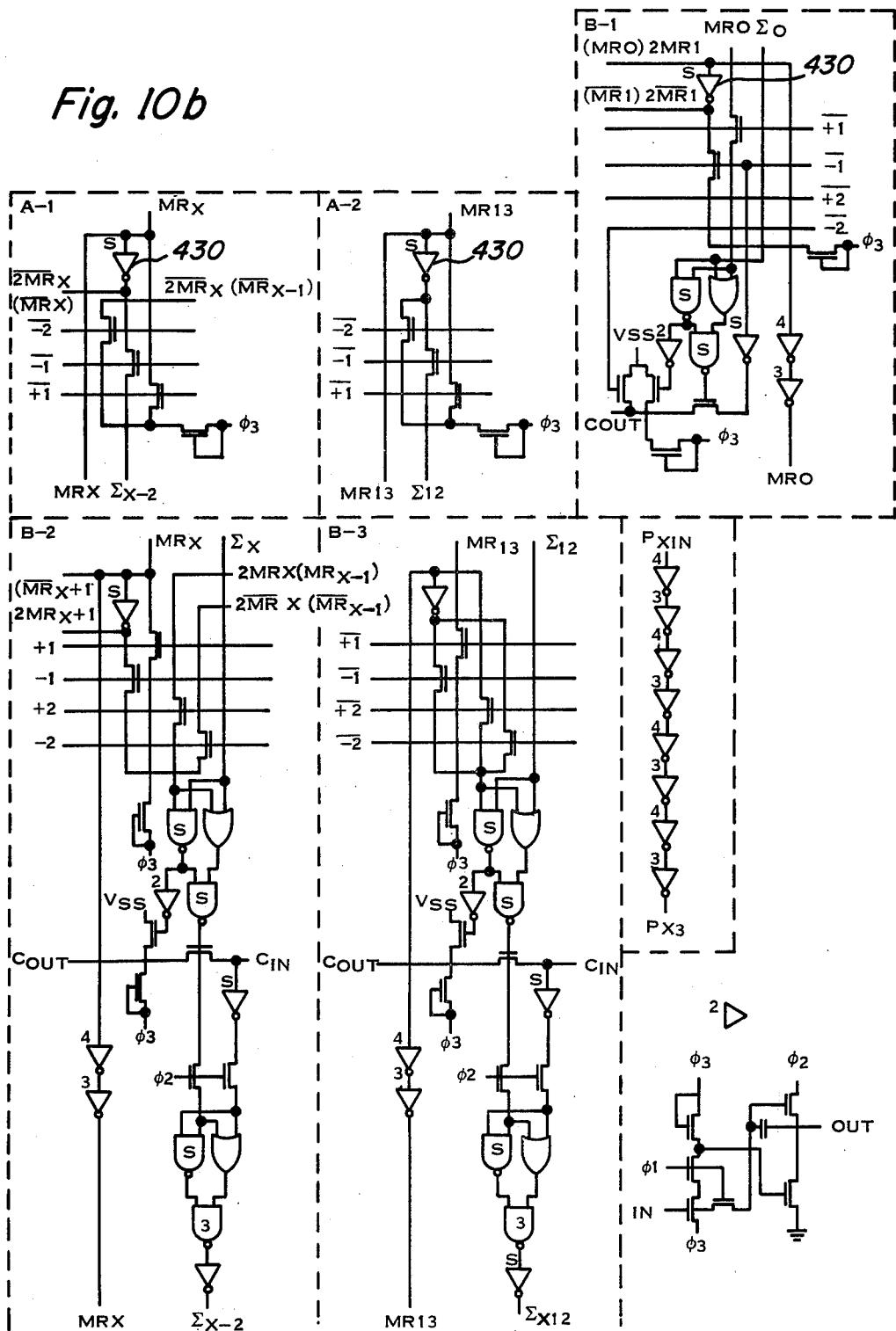


Fig. 10a (CONTINUED)

Fig. 10b



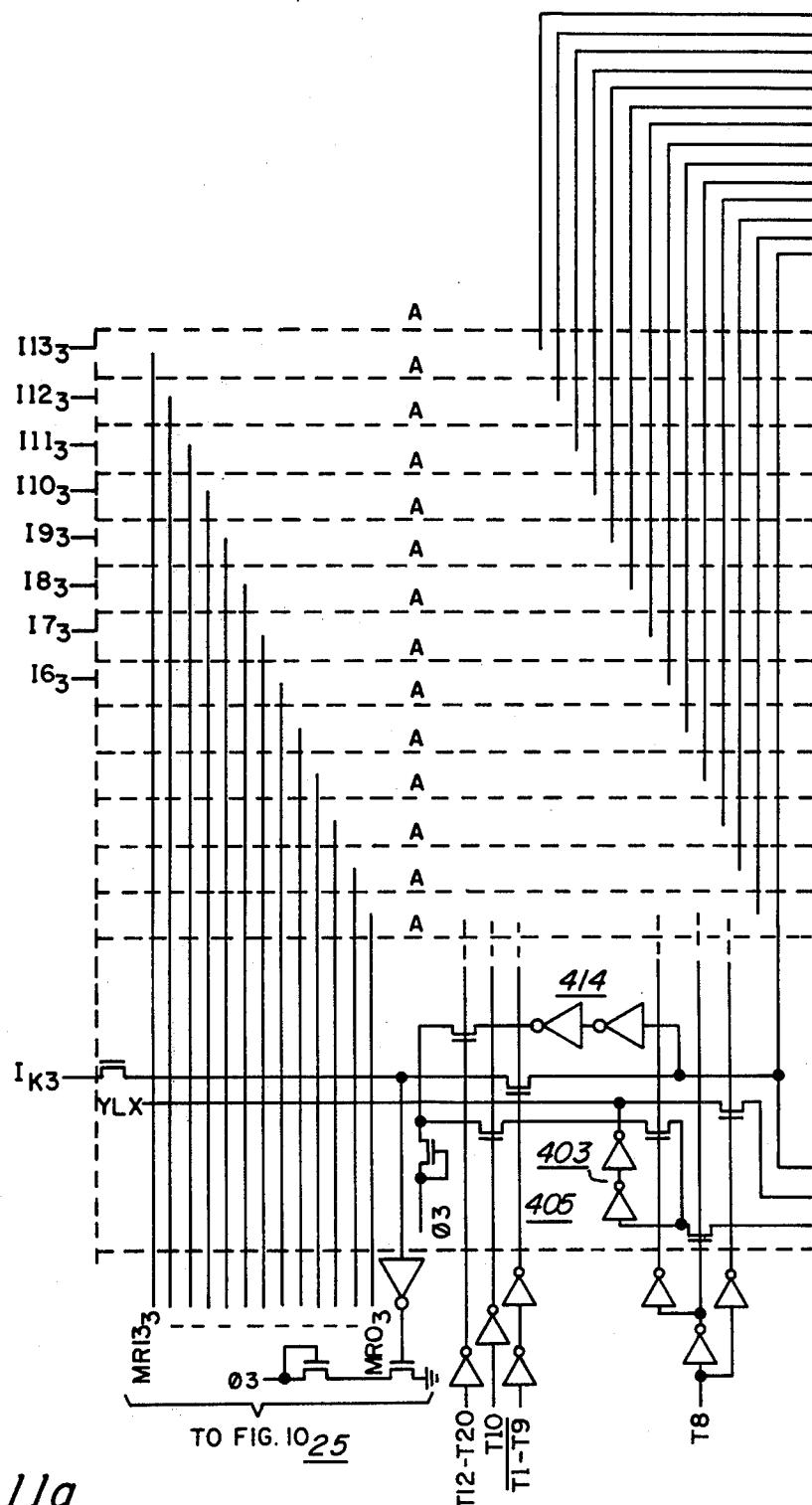
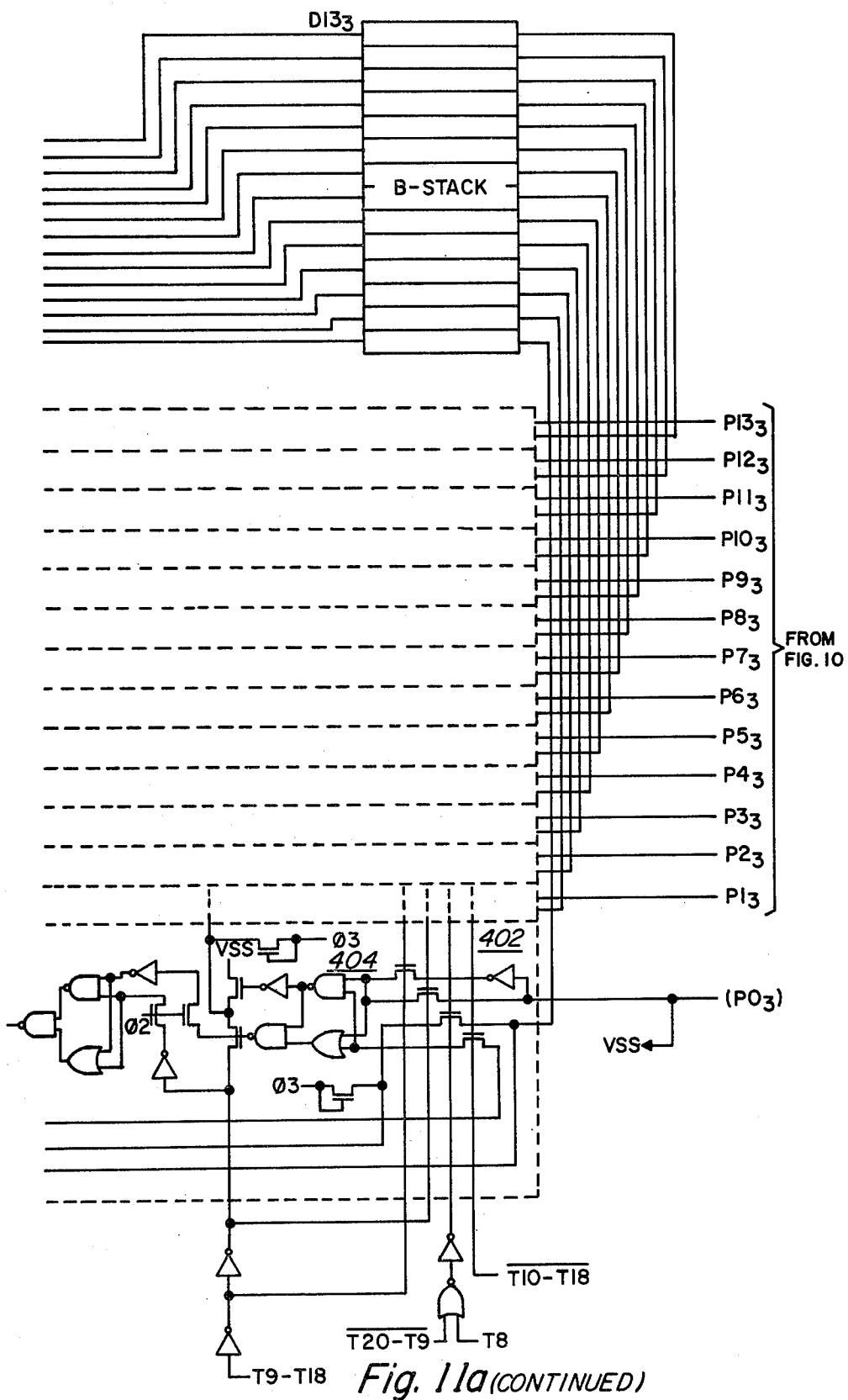


Fig. 11a



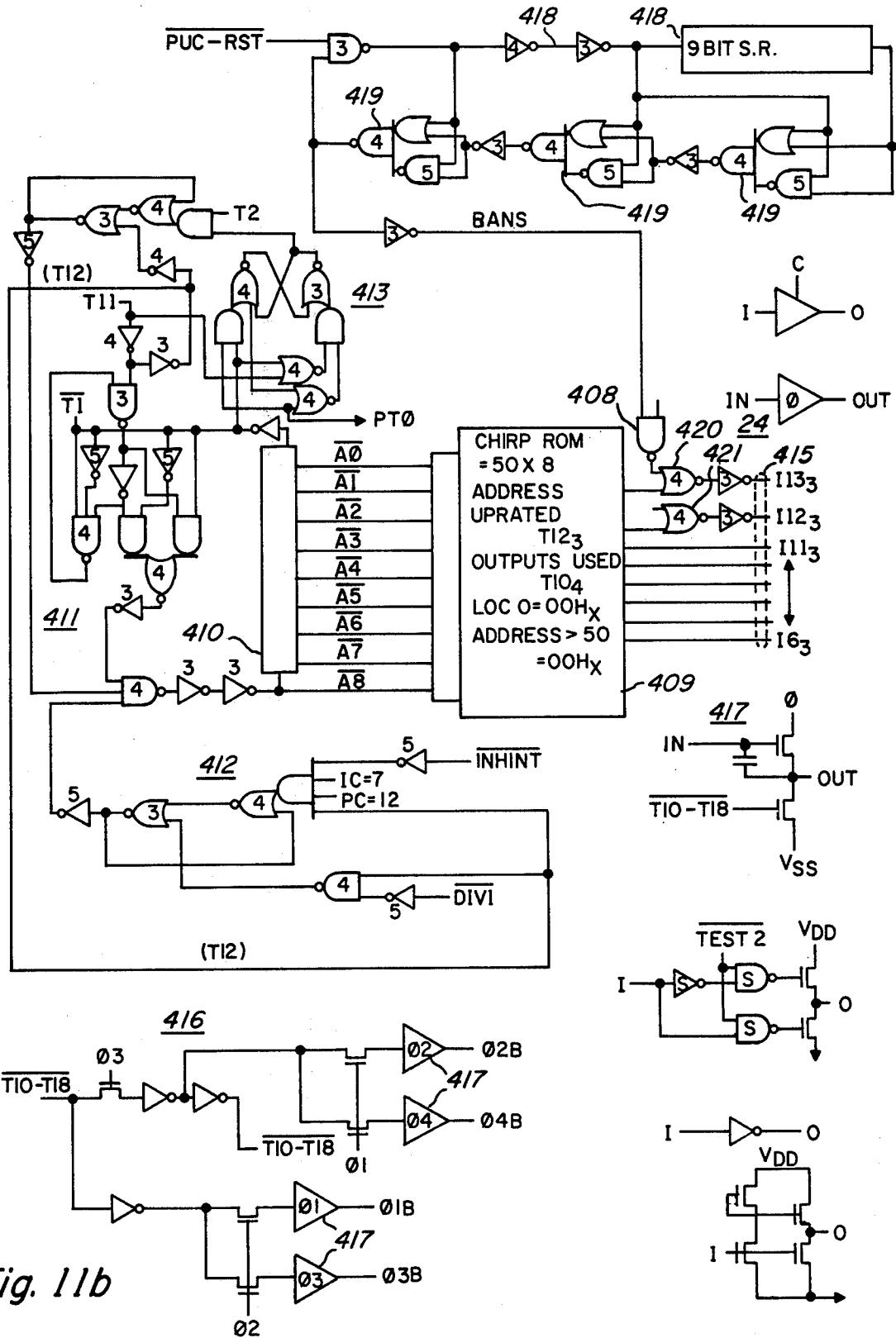


Fig. 11b

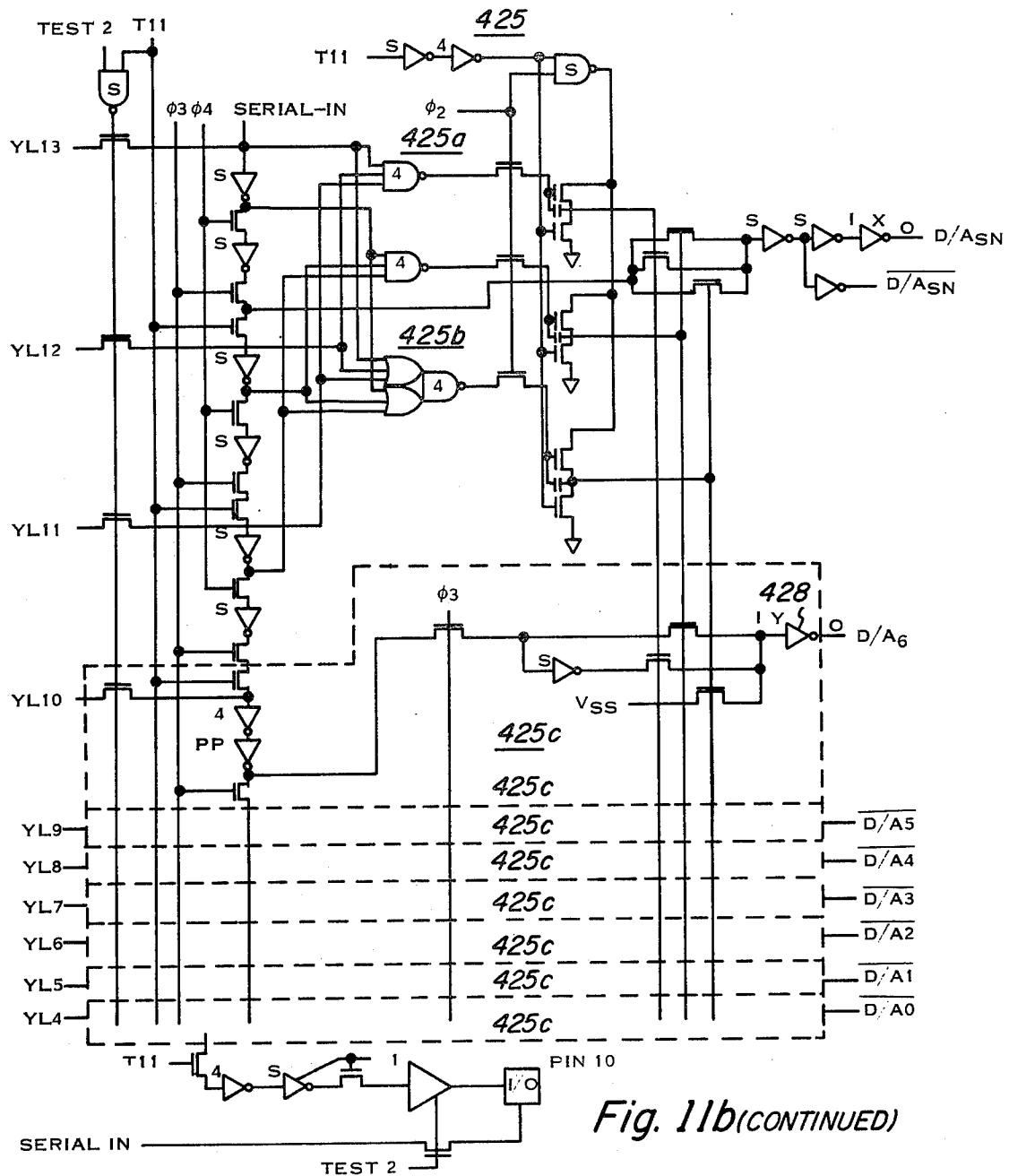
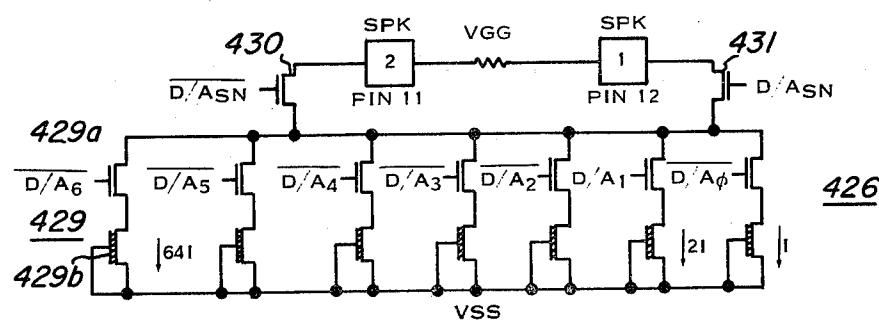
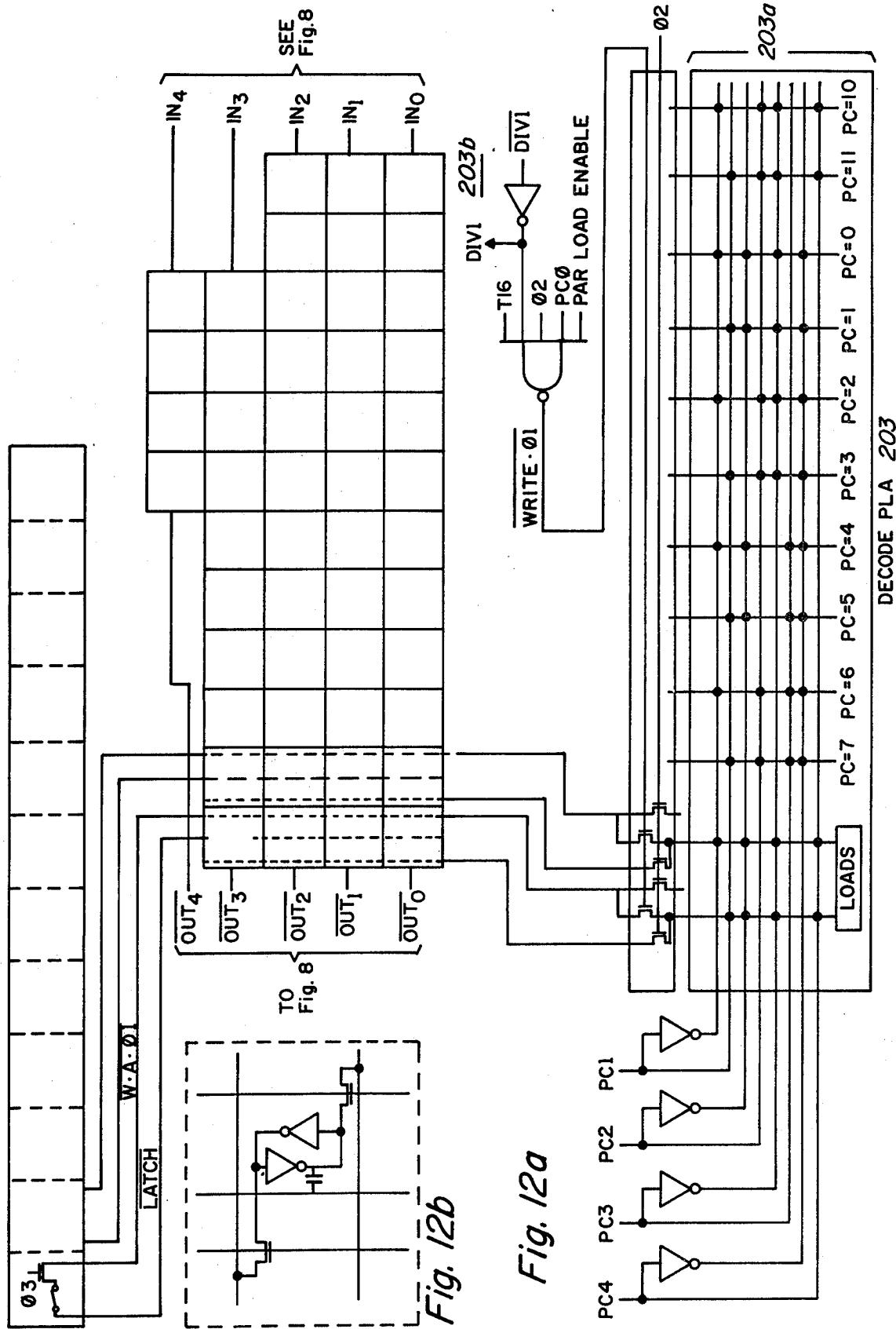


Fig. 11b (CONTINUED)





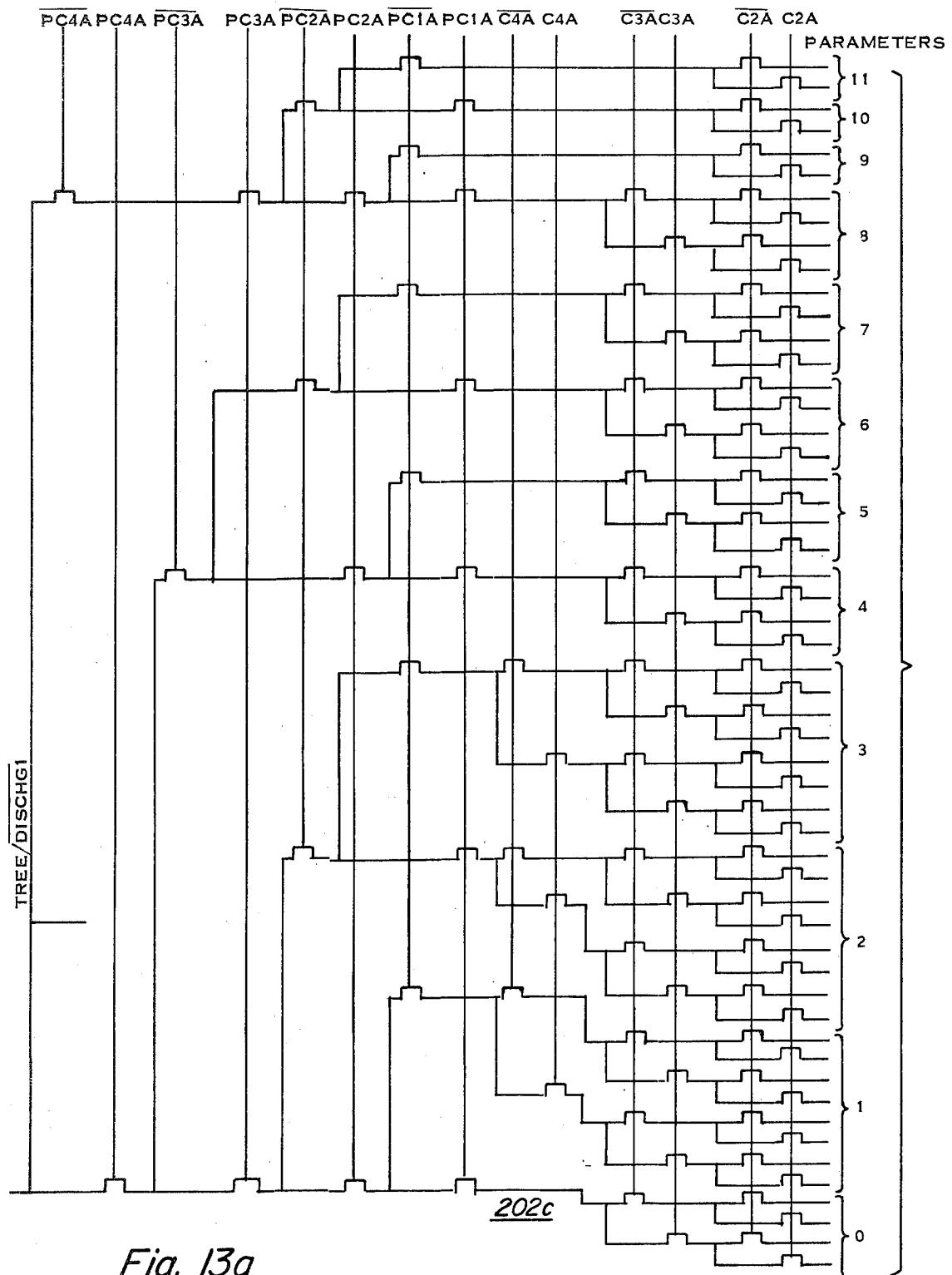
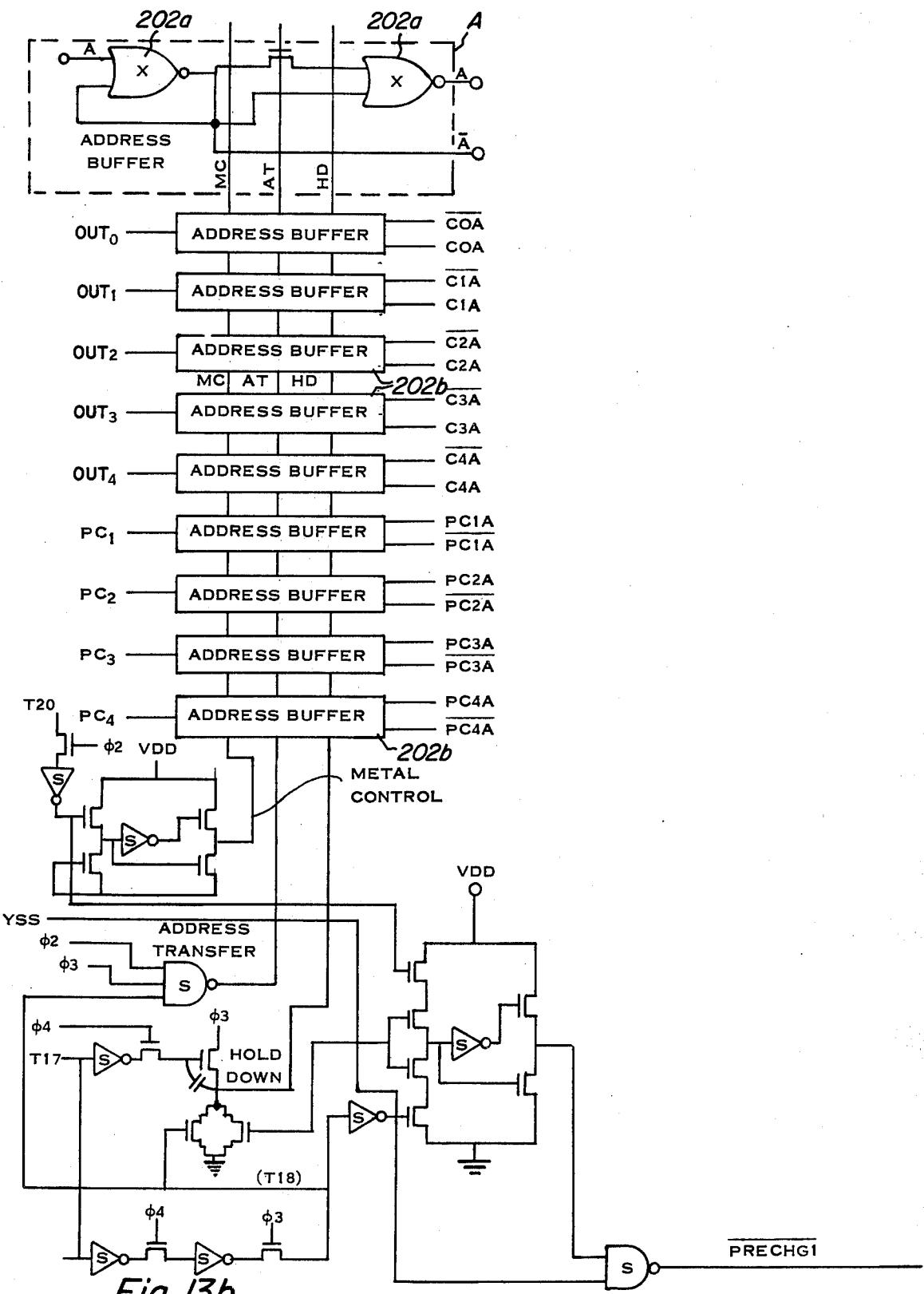
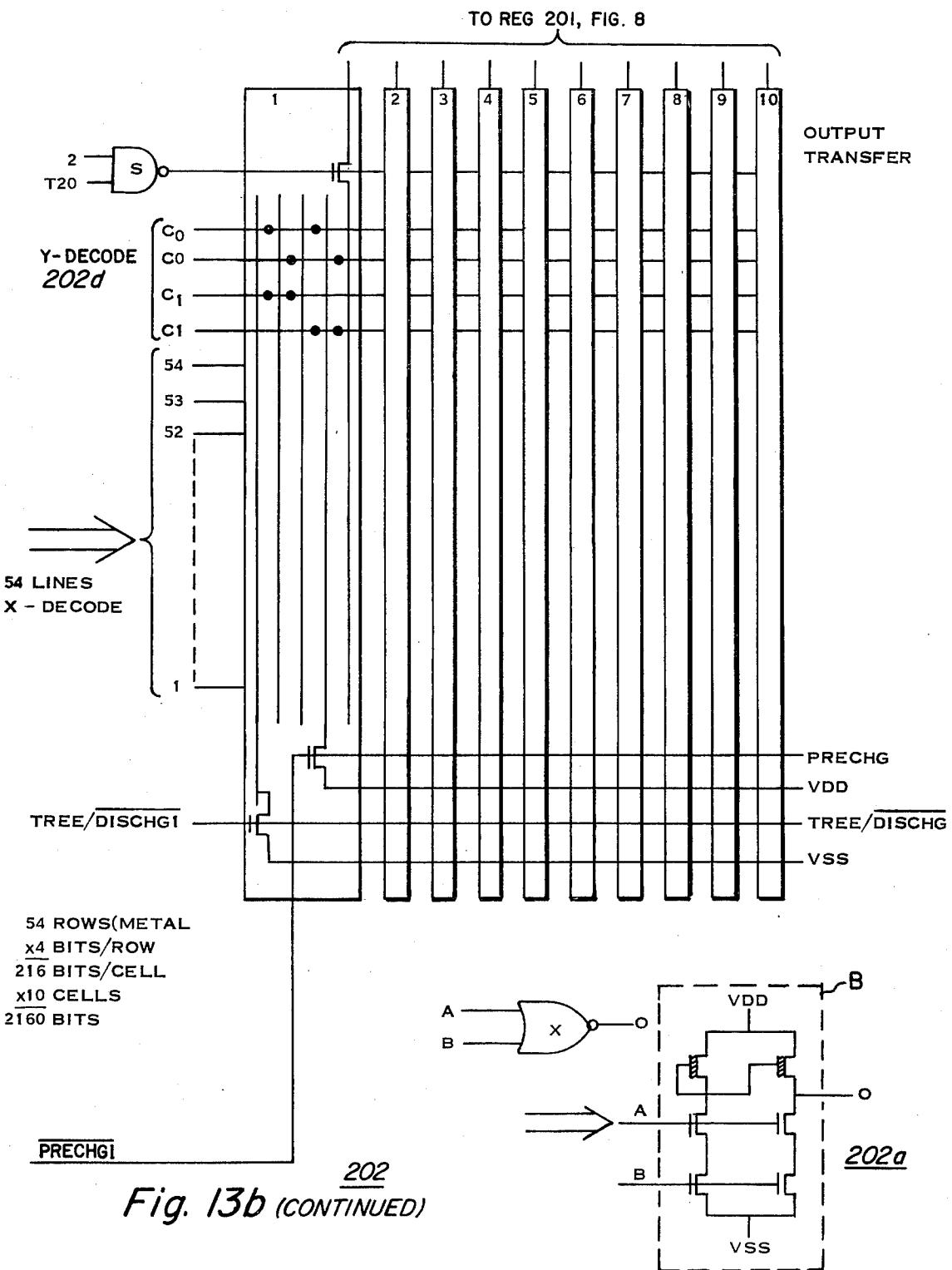
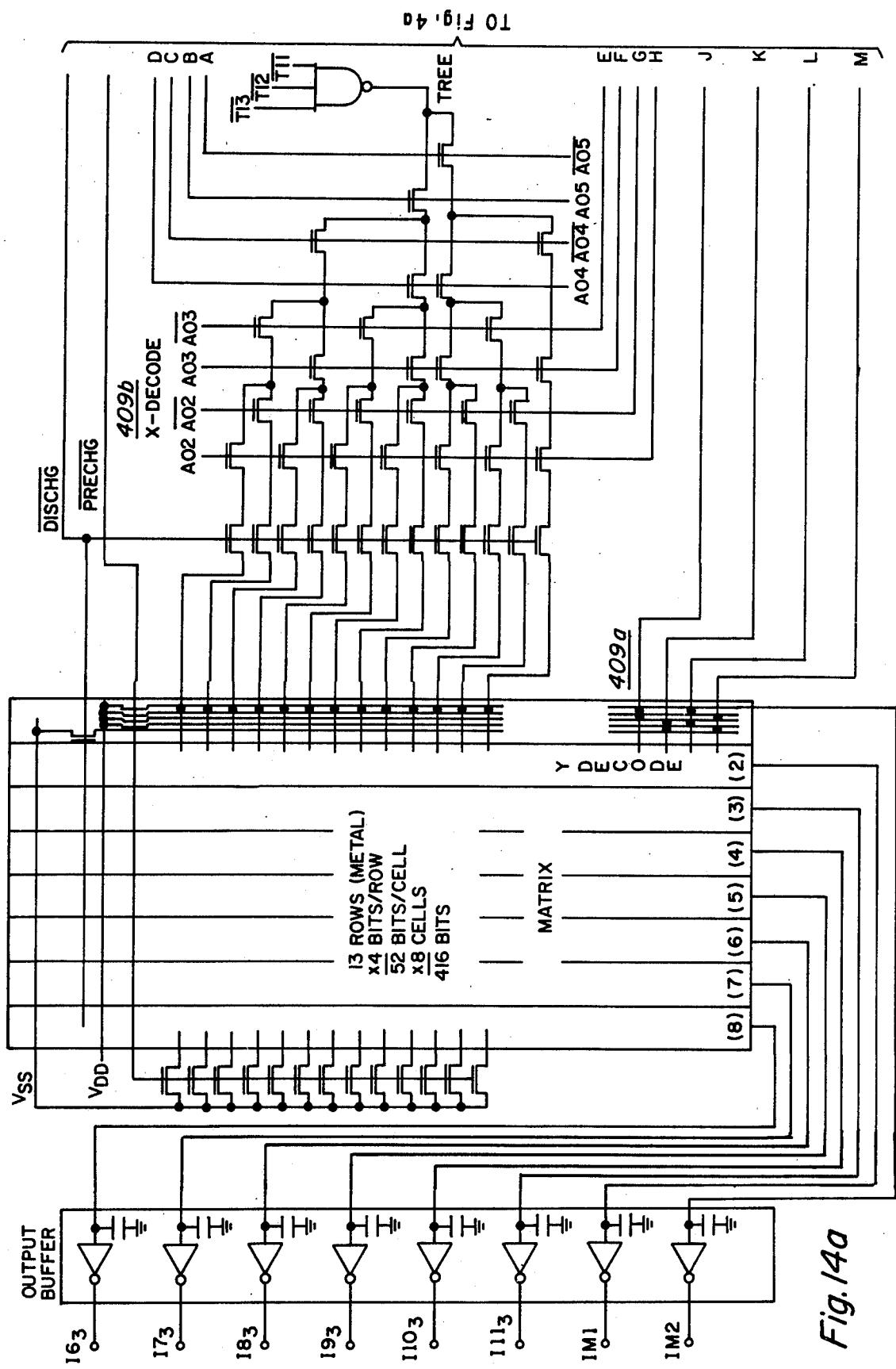
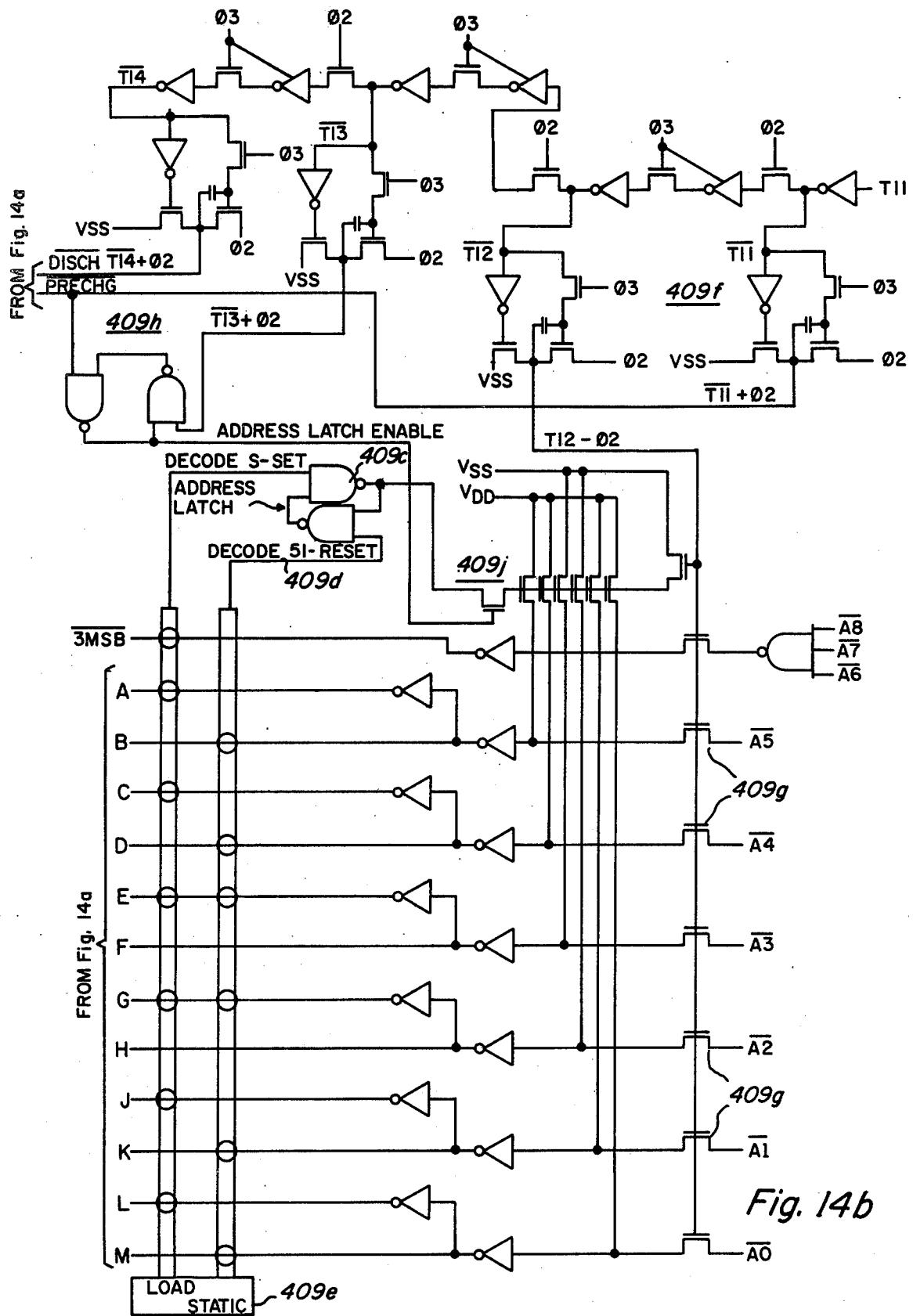


Fig. 13a









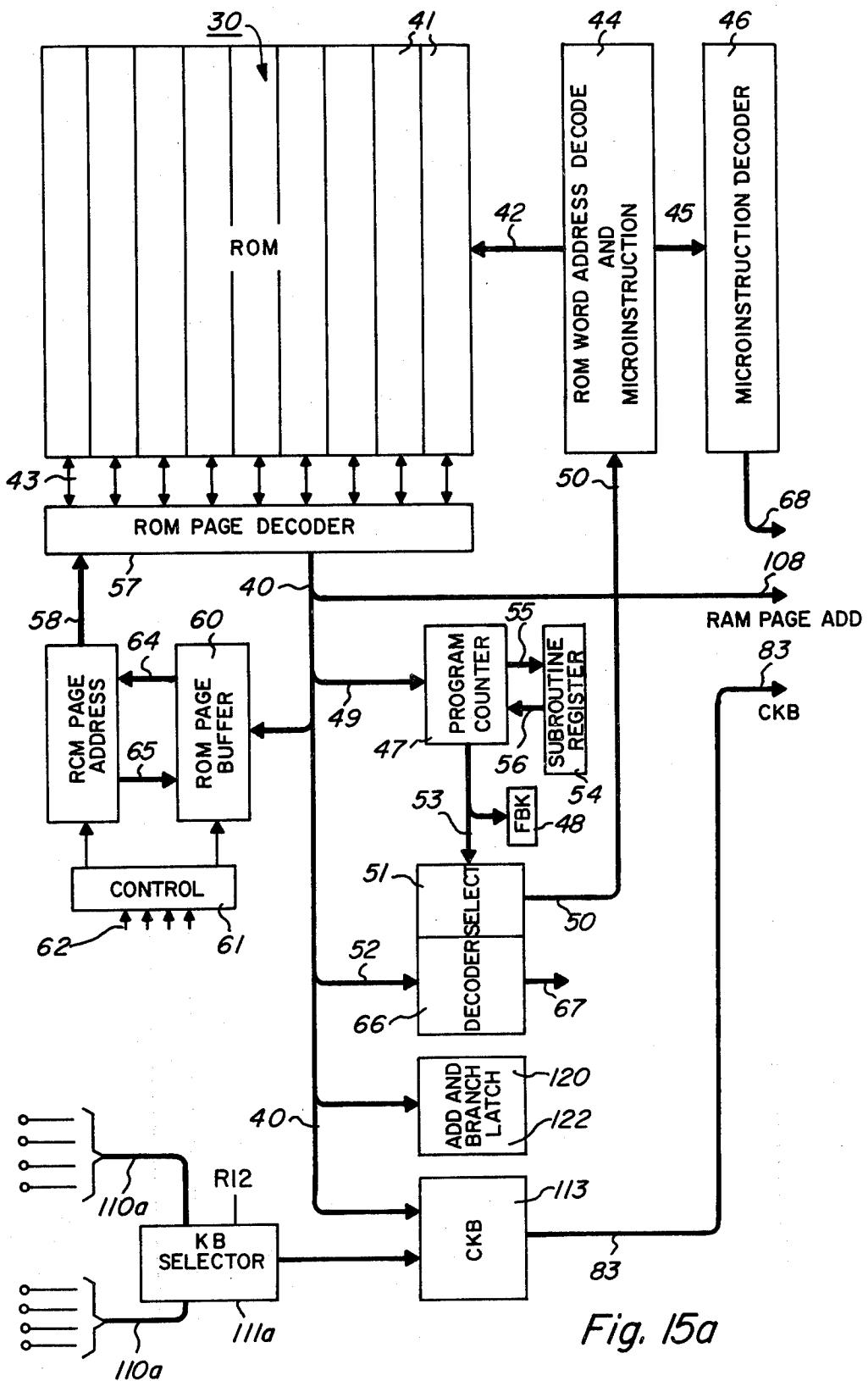
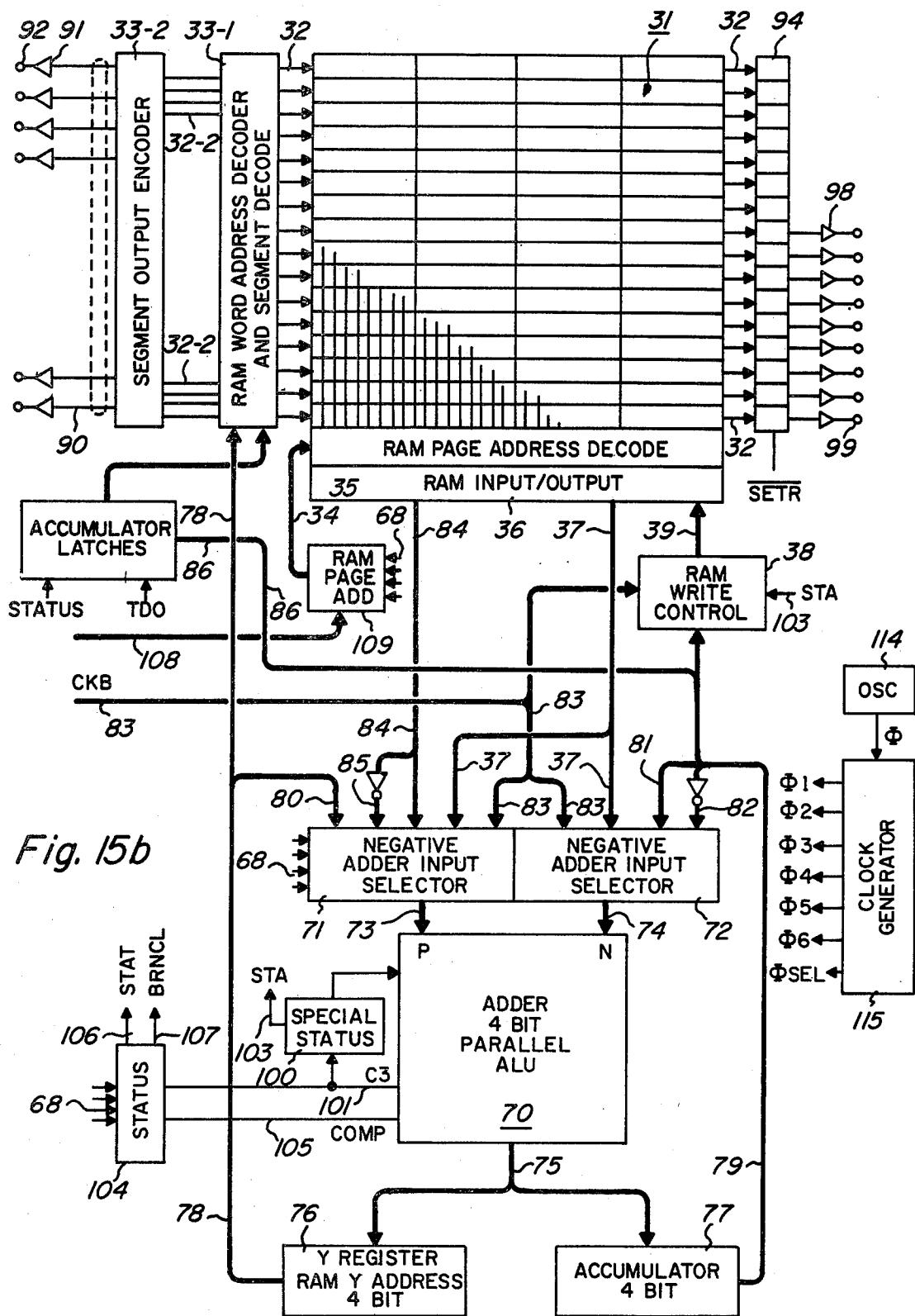


Fig. 15a



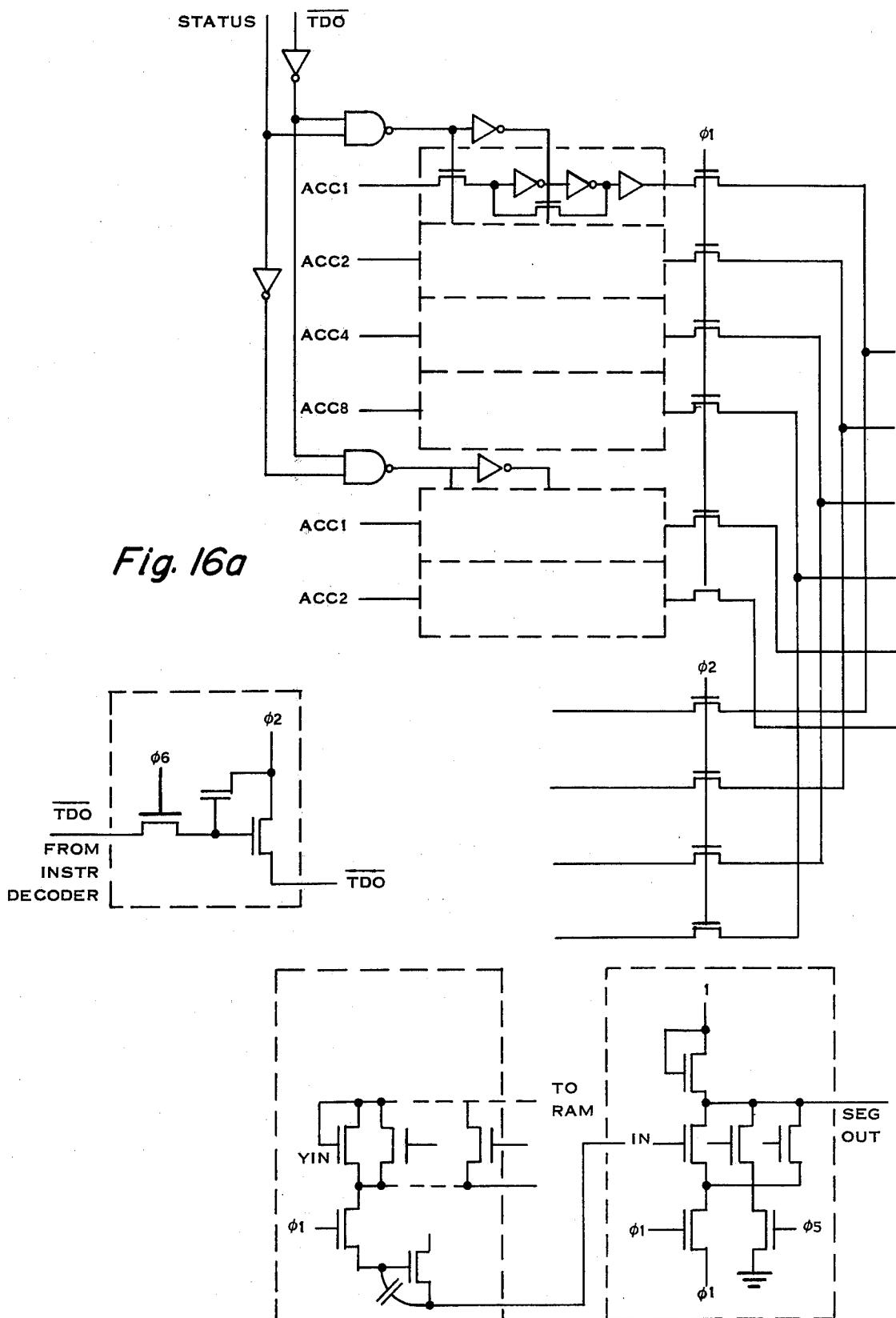
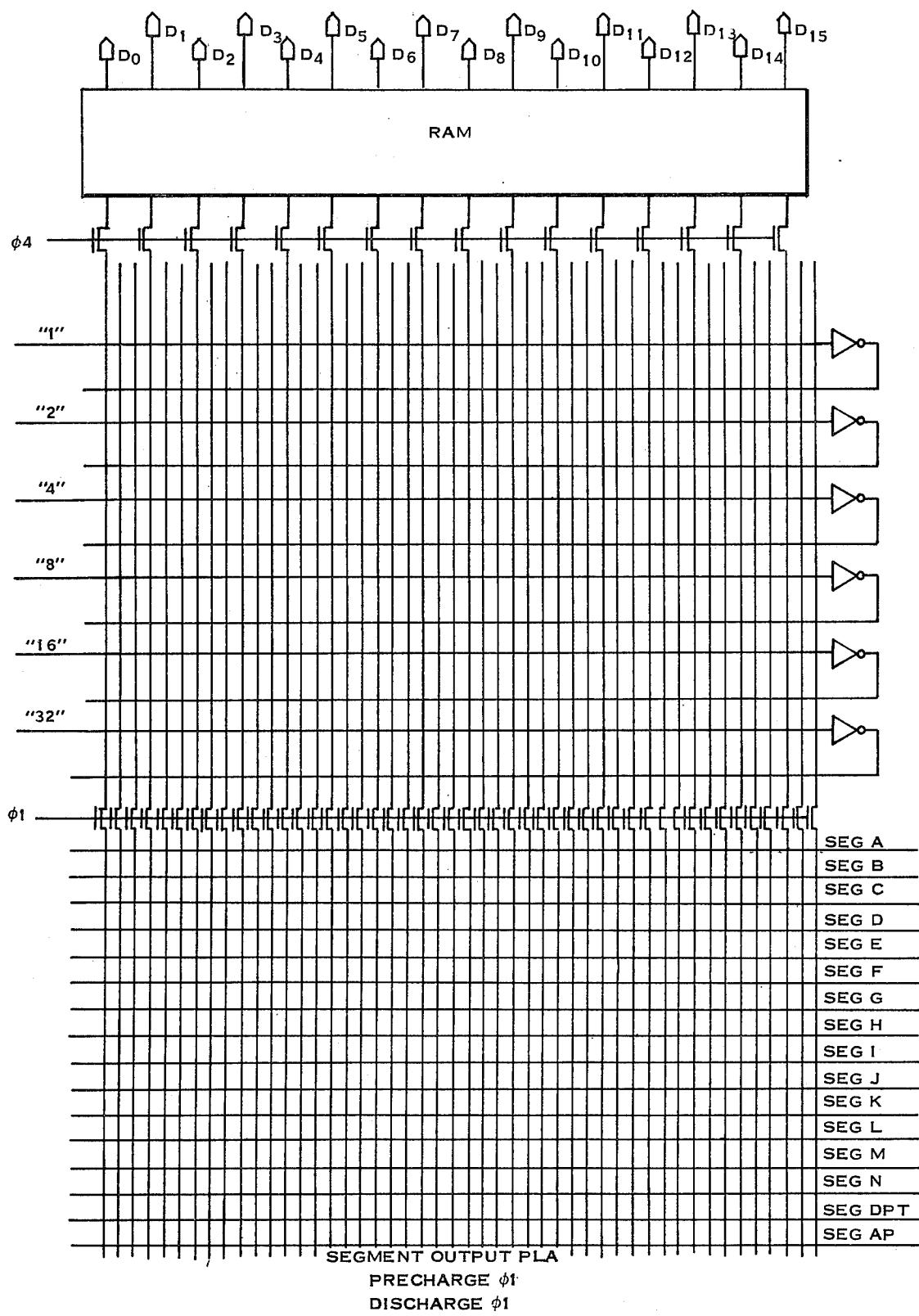


Fig. 16b

TO DIGIT LOGIC



RAM DECODE PLA
PRECHARGE ϕ_4
DISCHARGE ϕ_1

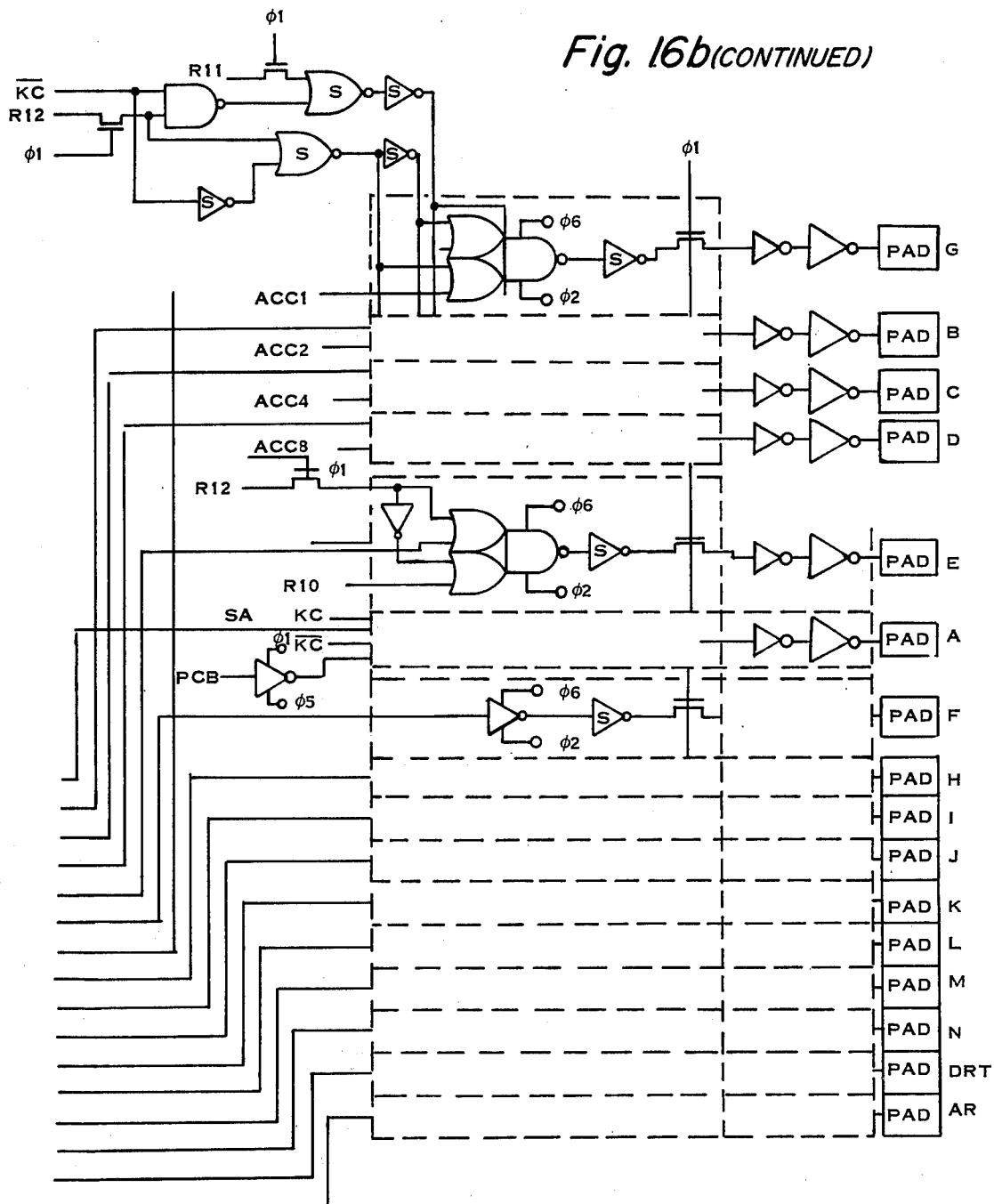


Fig. 16b (CONTINUED)

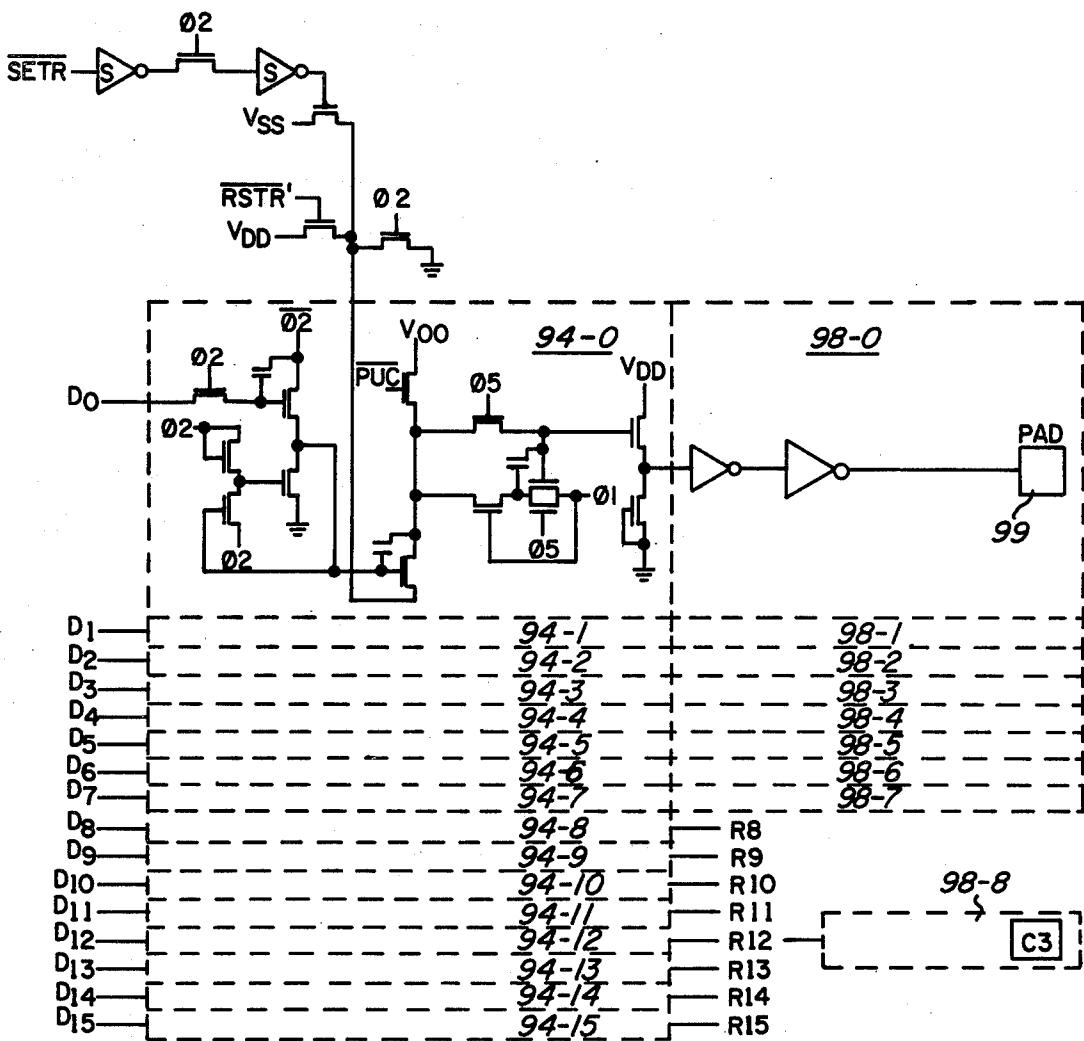


Fig. 17

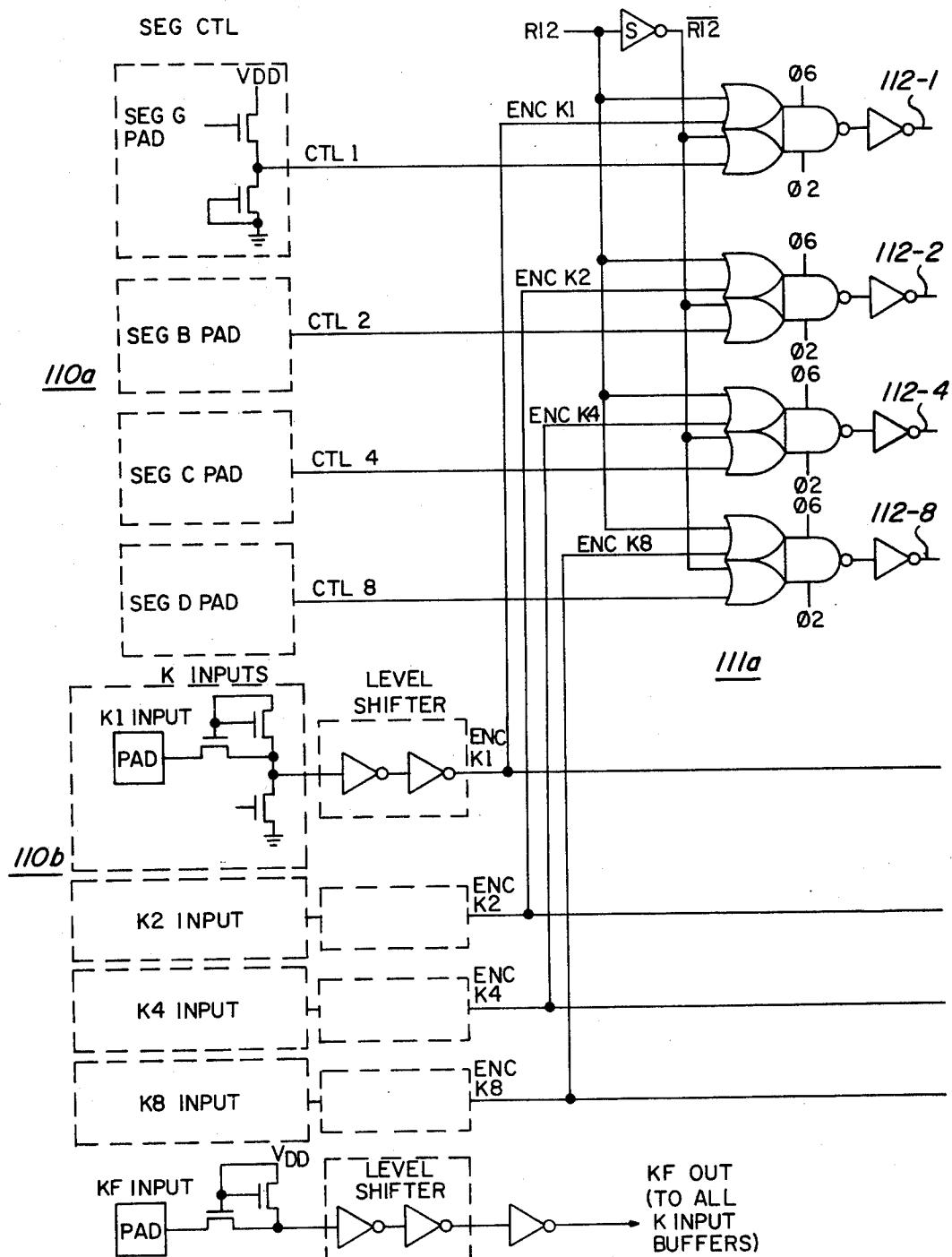


Fig. 18

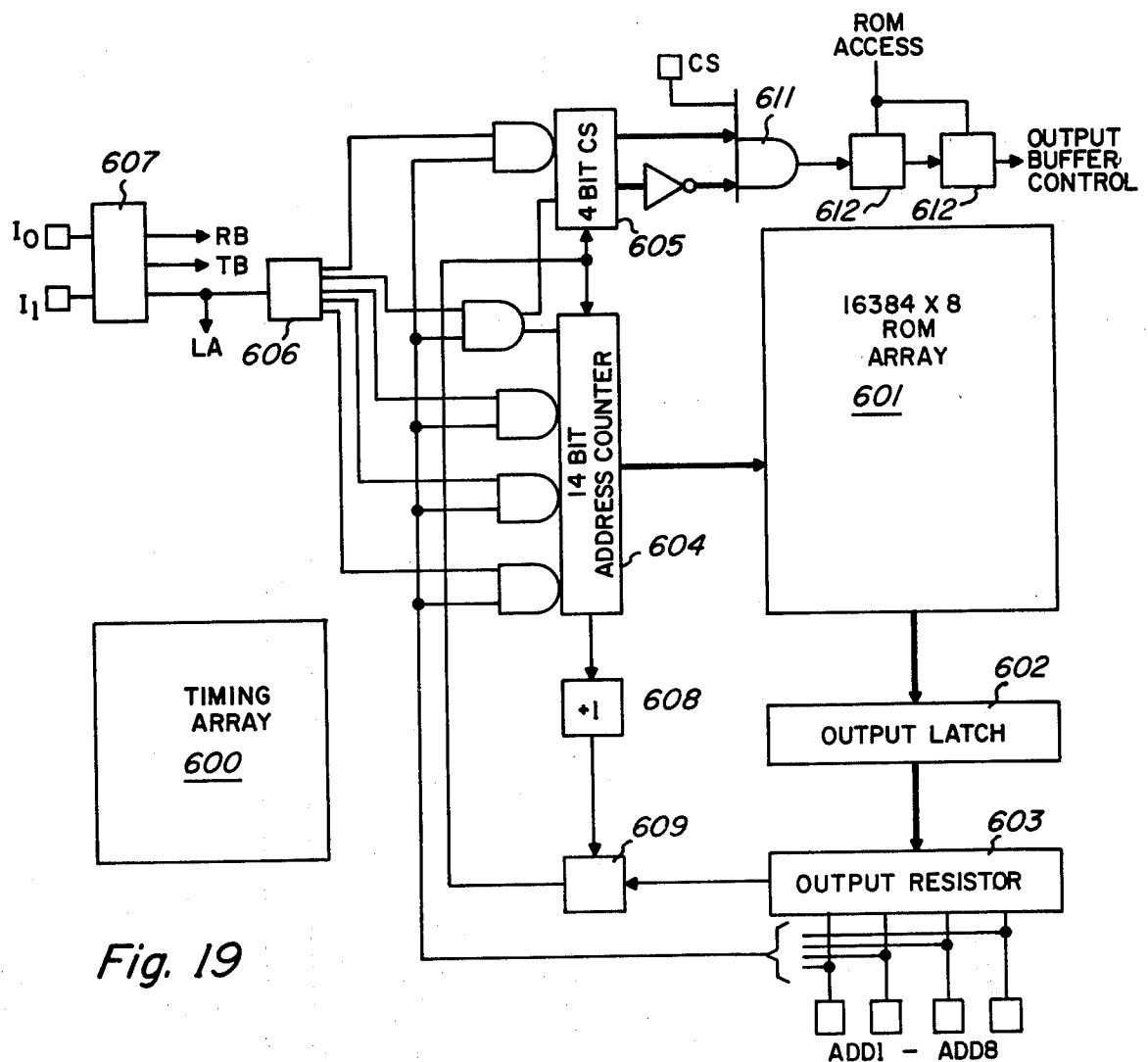
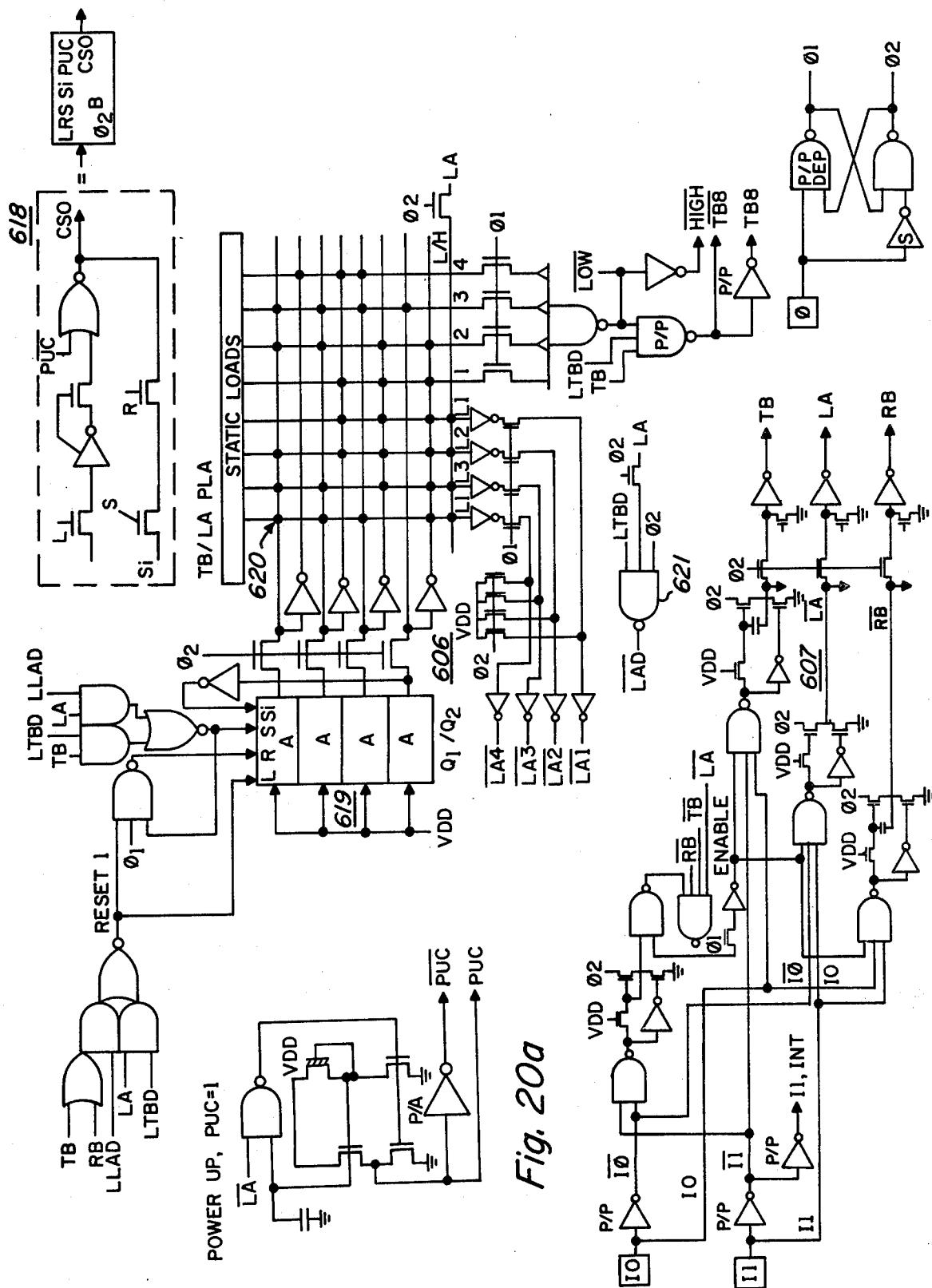


Fig. 19



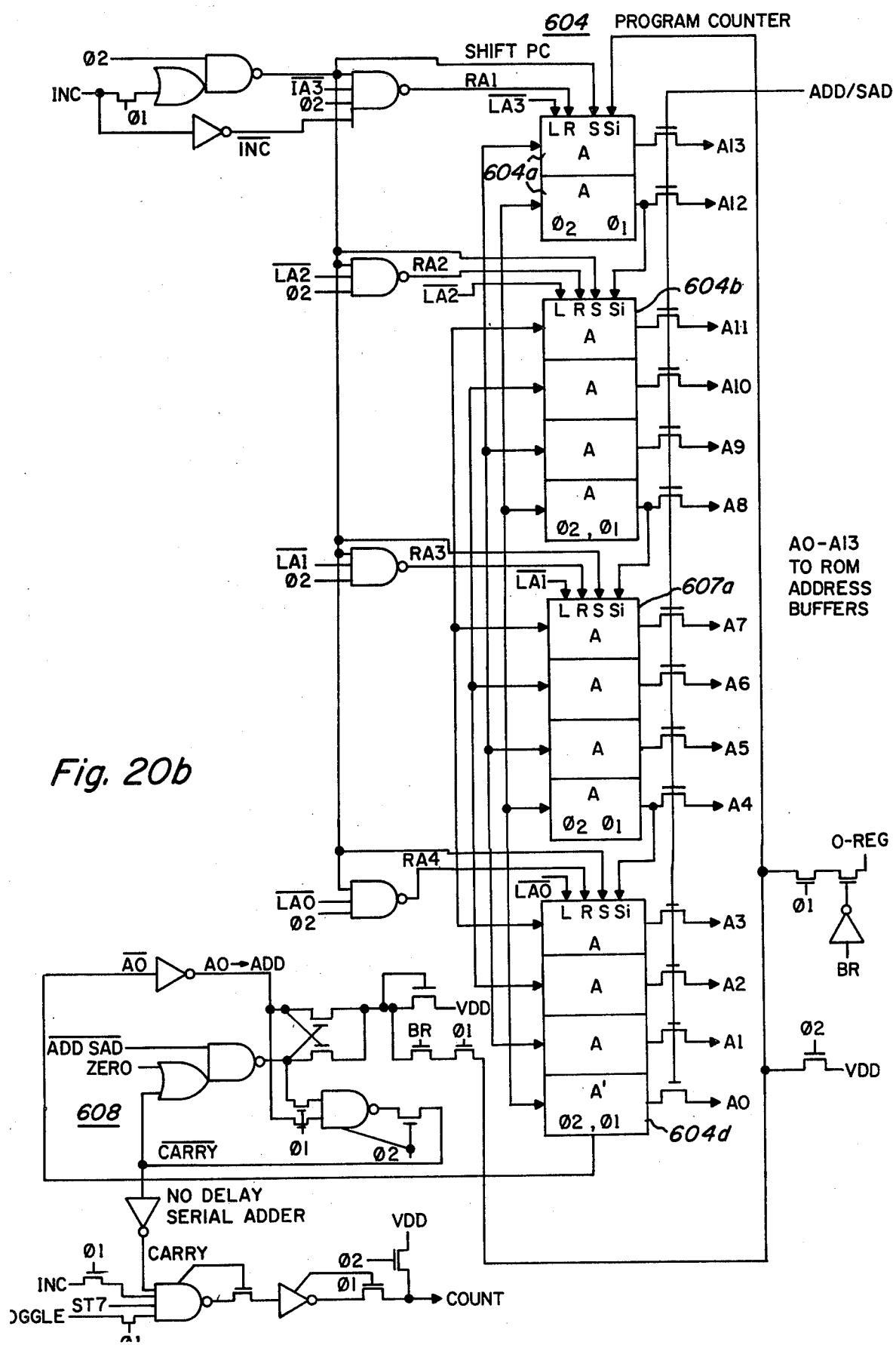
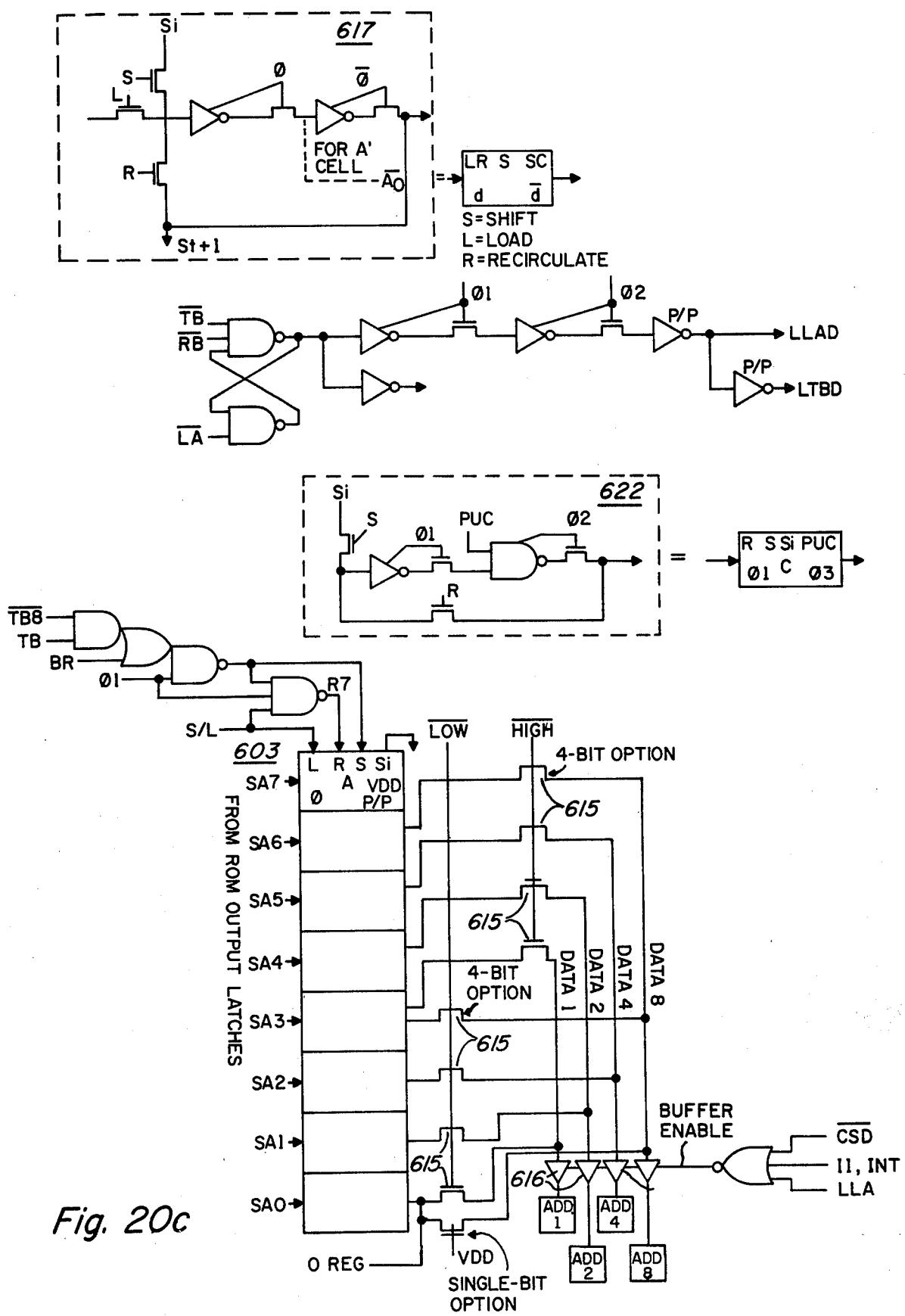


Fig. 20b



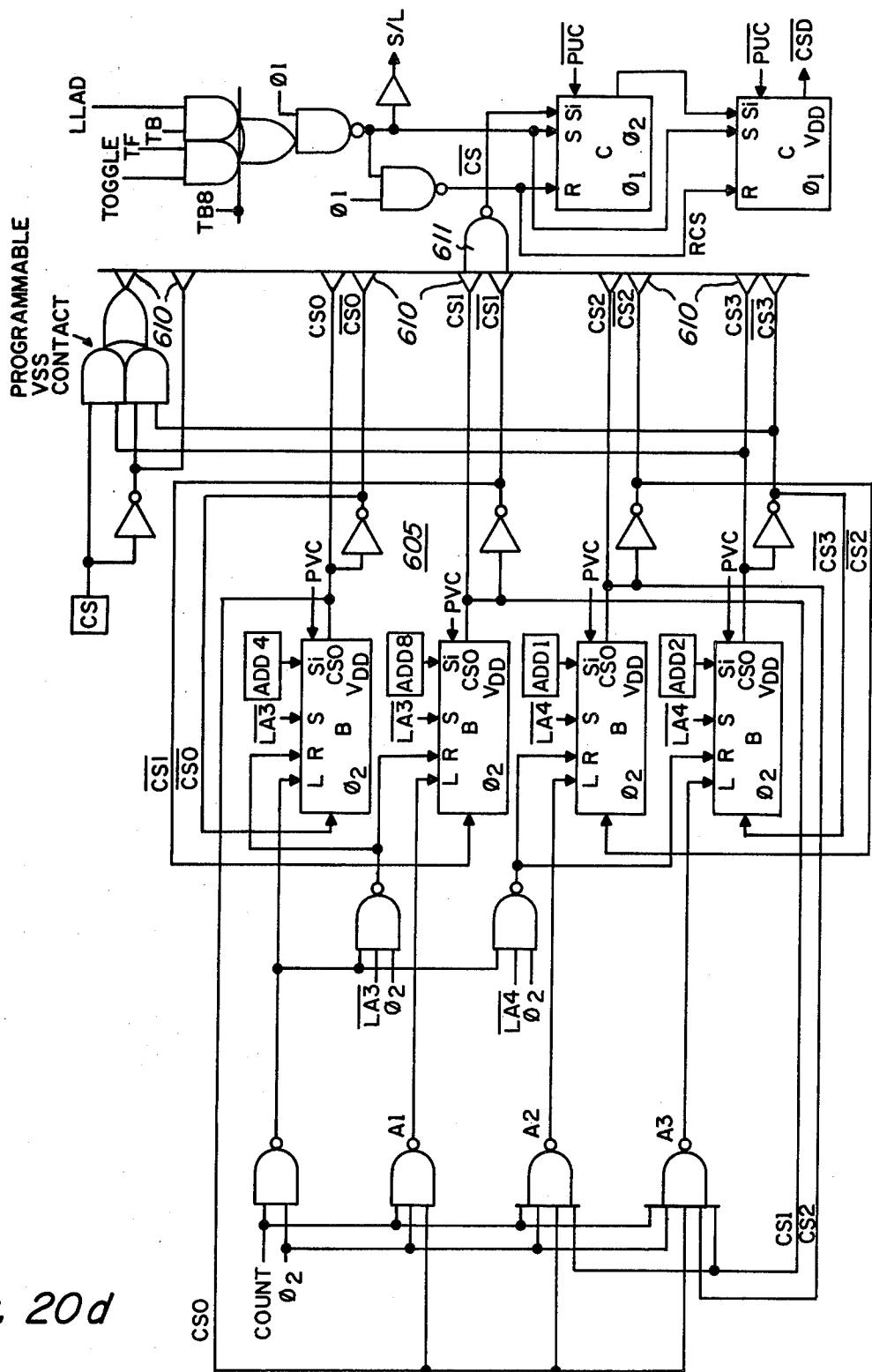


Fig. 20d

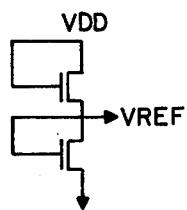
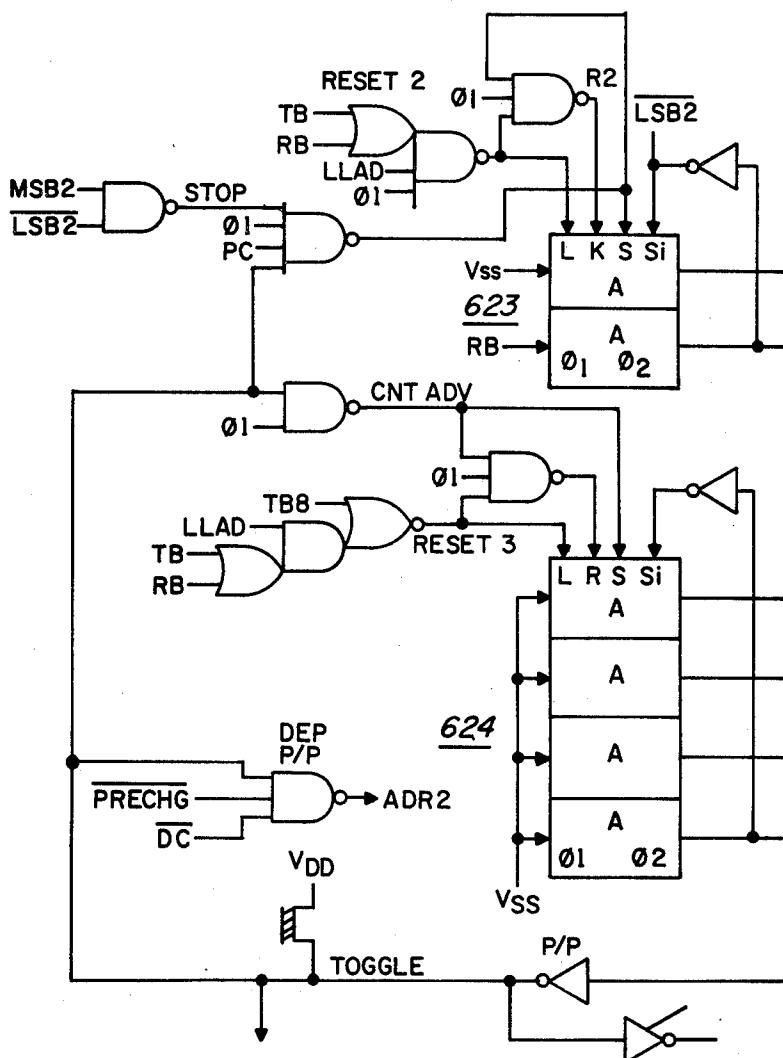


Fig. 20e

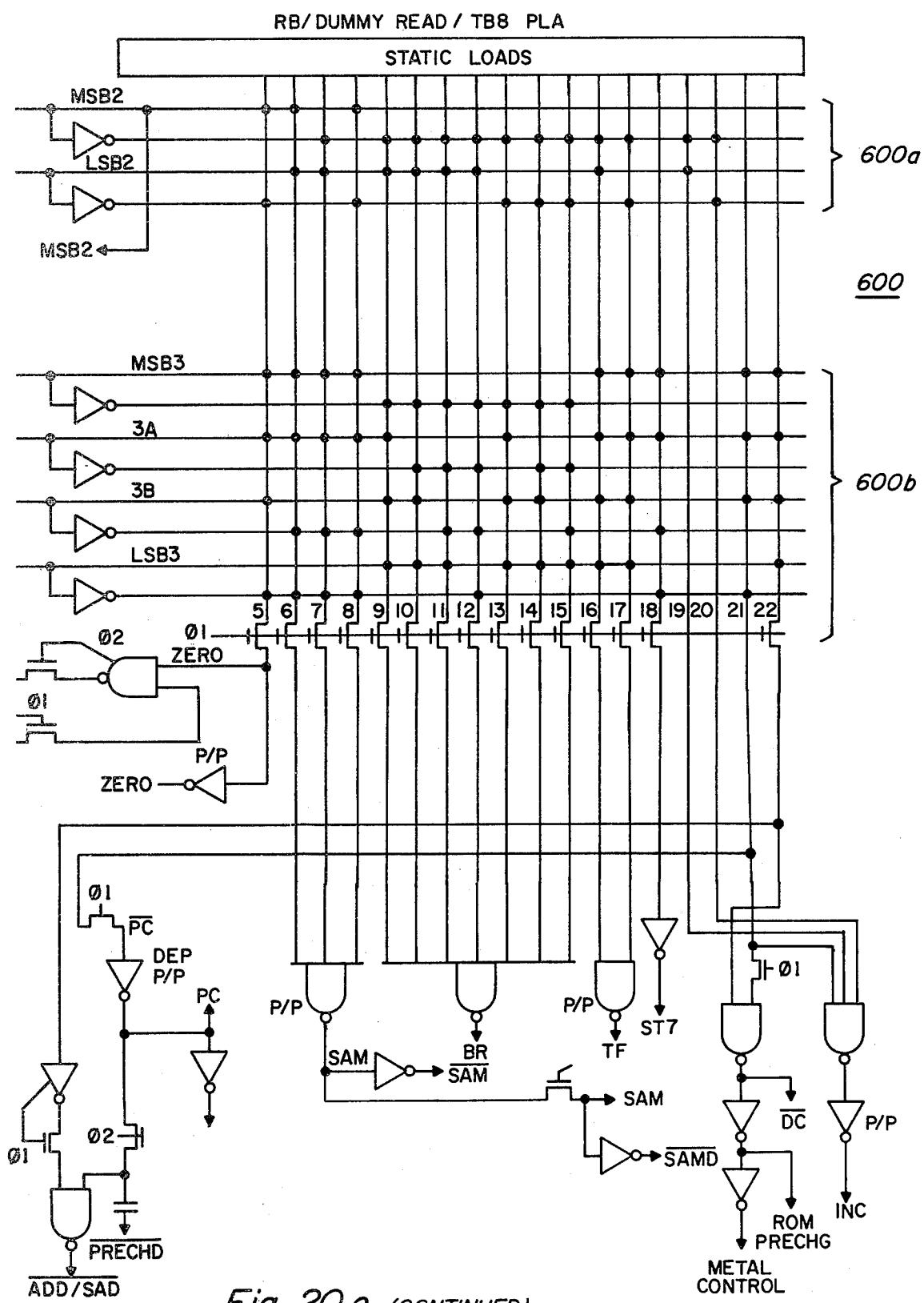


Fig. 20 e (CONTINUED)

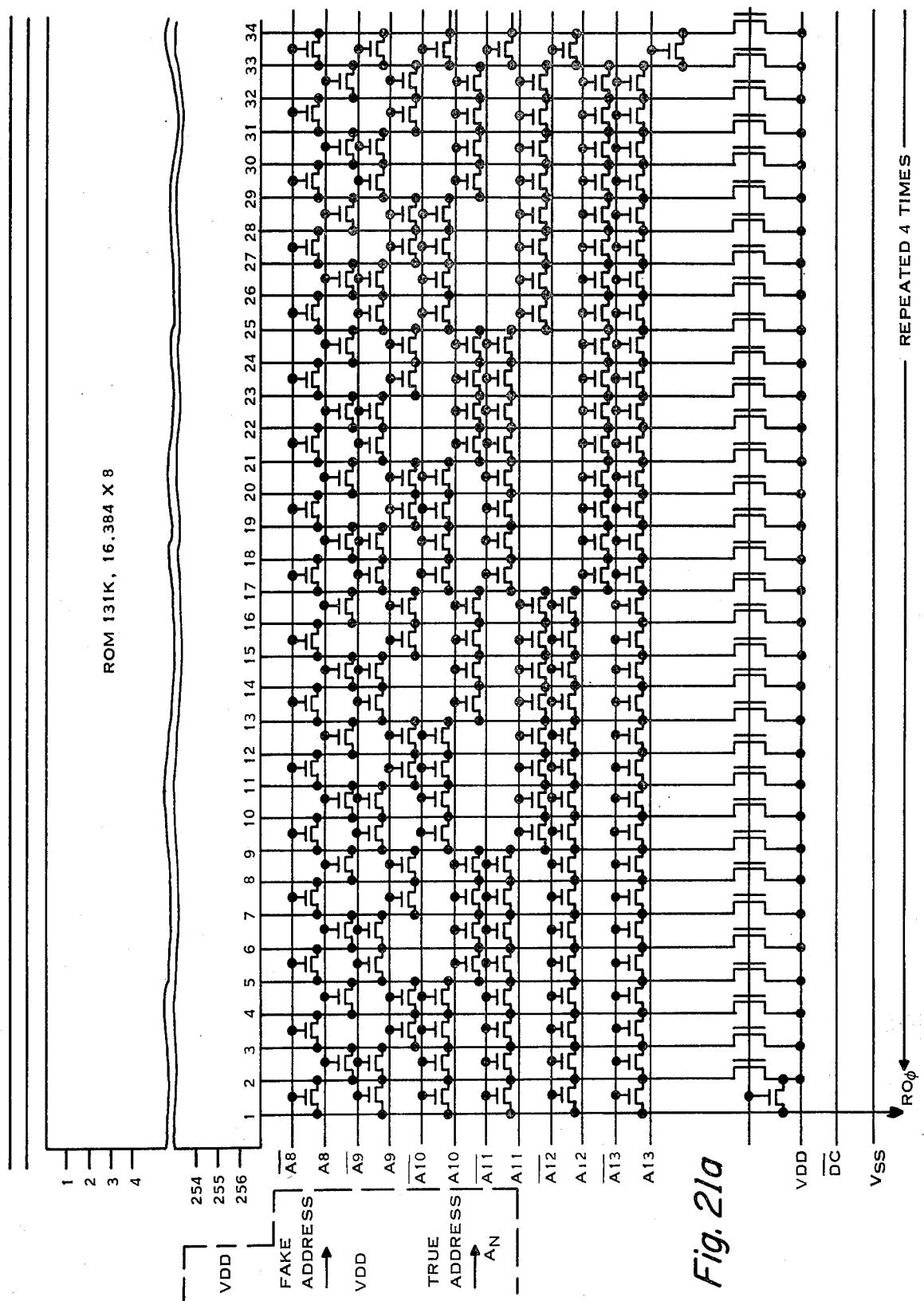


Fig. 21a

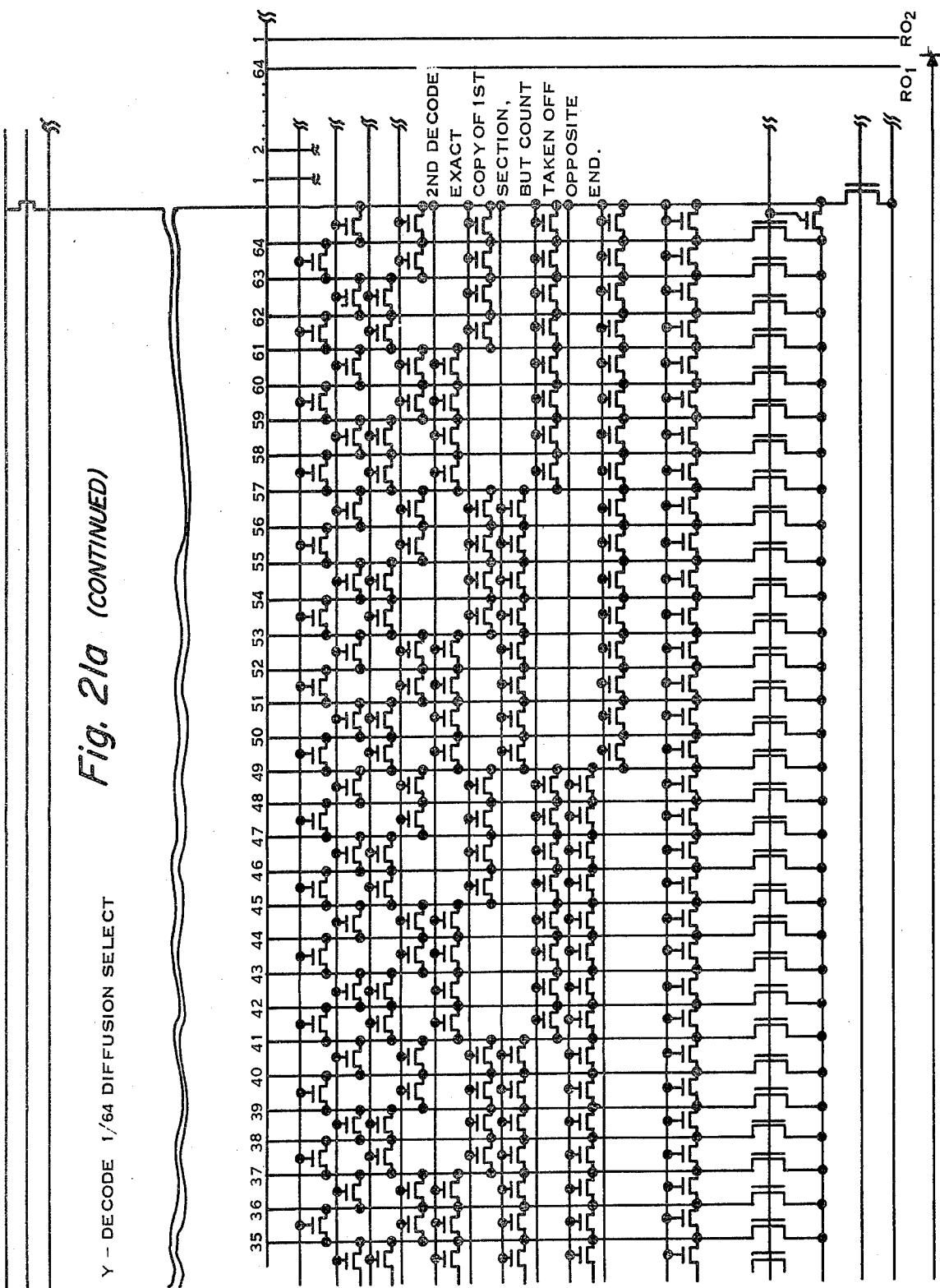
Fig. 2/a (CONTINUED)

Fig. 21b

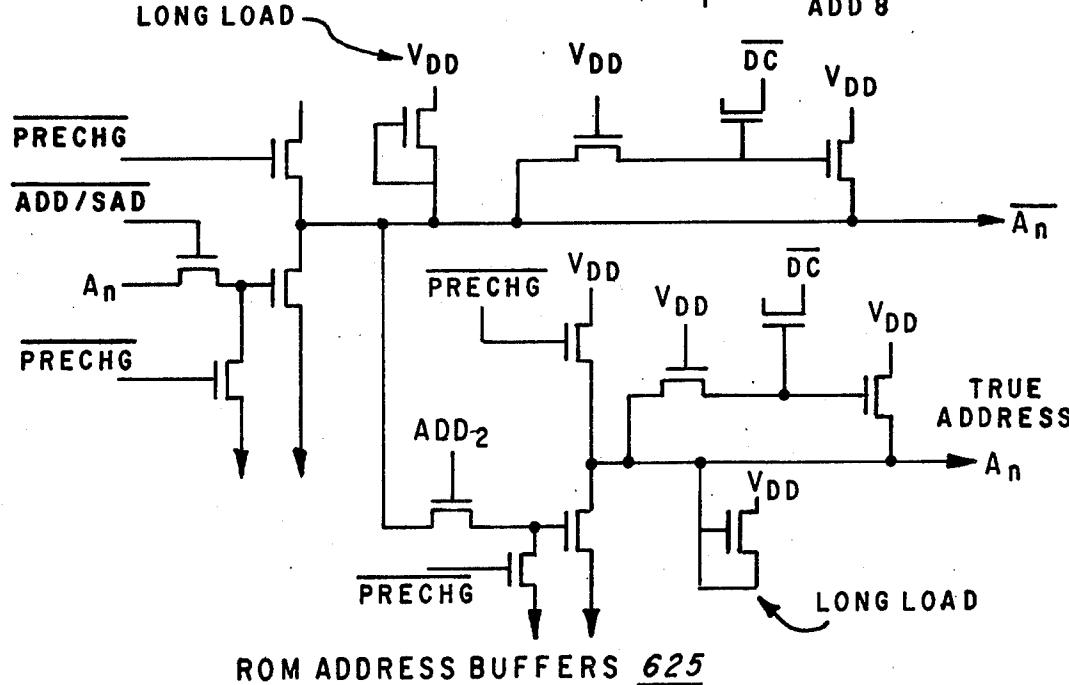
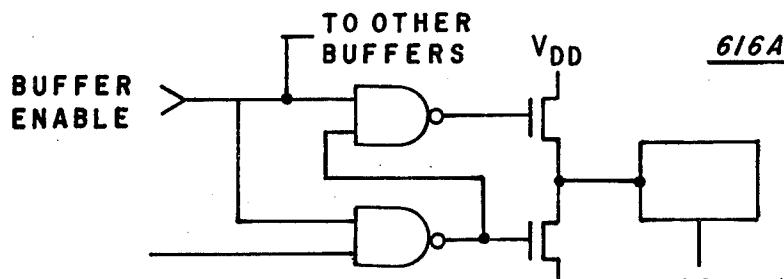
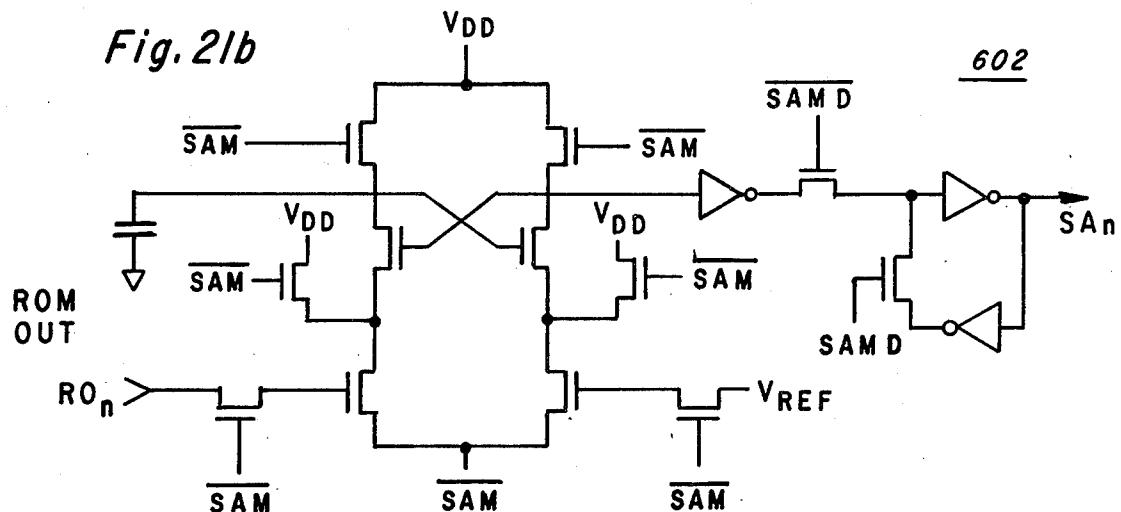
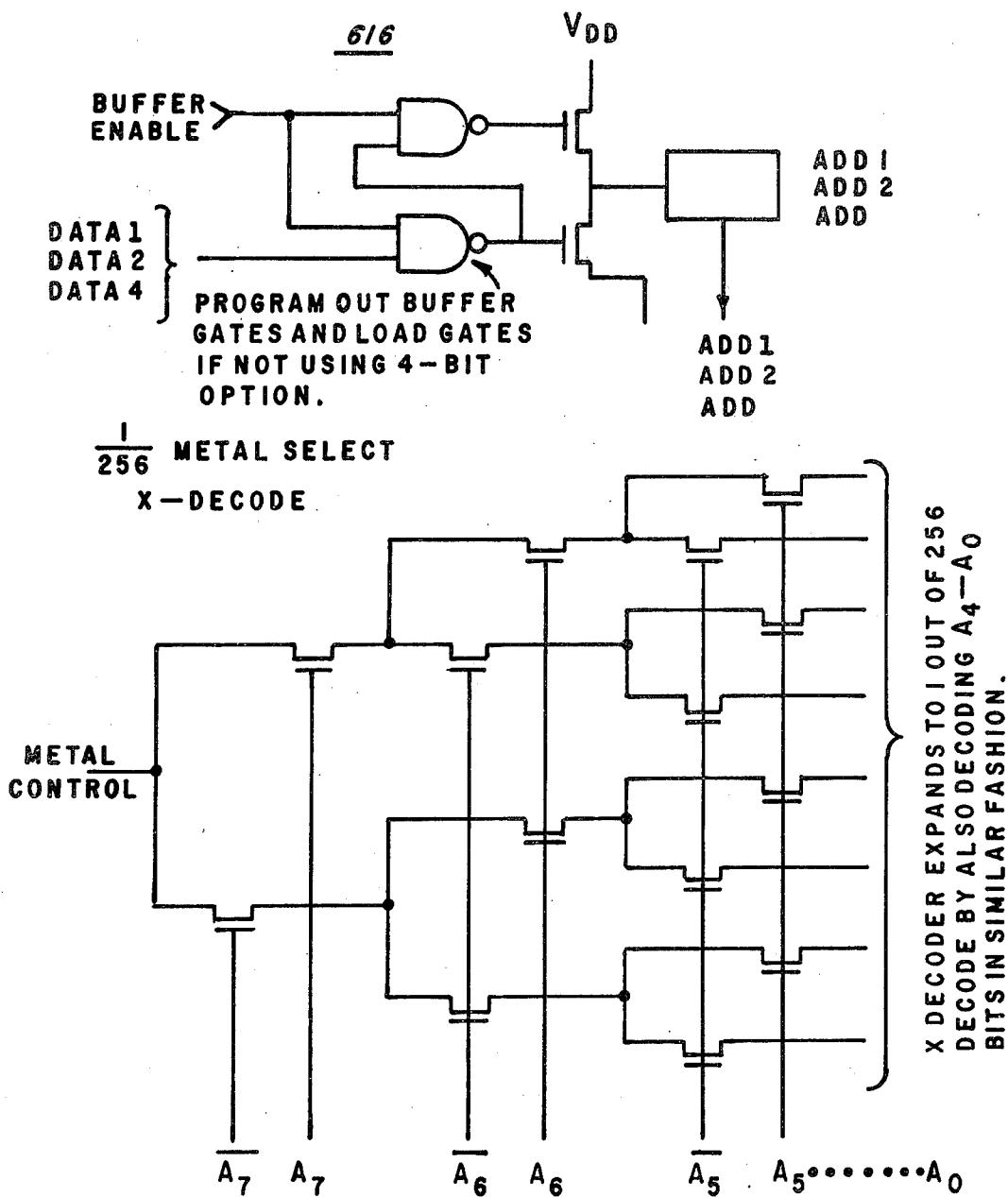


Fig. 21b (CONTINUED)



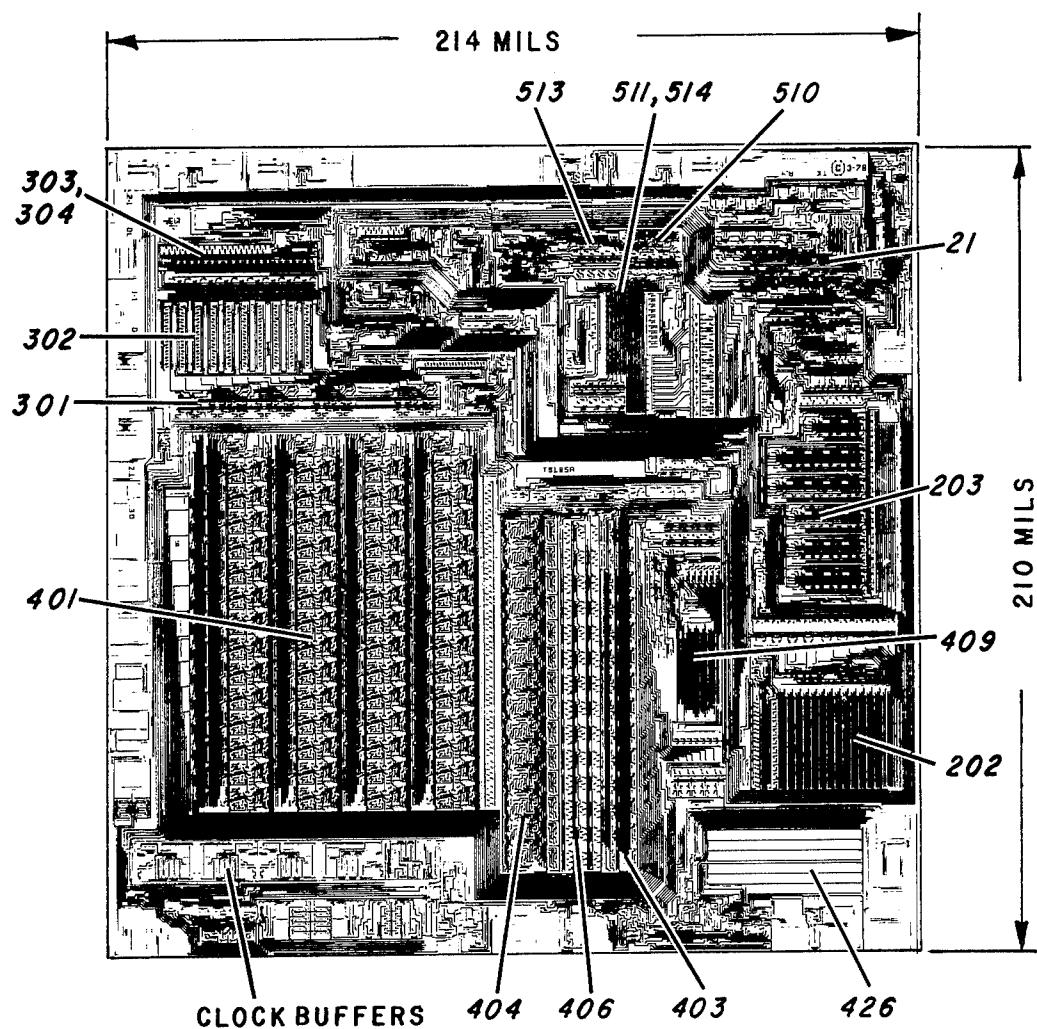
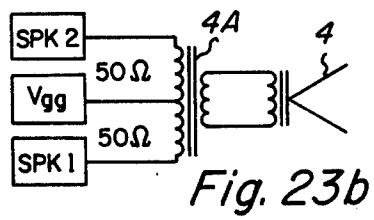
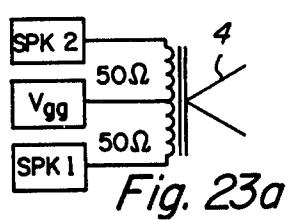


Fig. 22



**PARAMETER INTERPOLATOR FOR SPEECH
SYNTHESIS CIRCUIT**

BACKGROUND OF THE INVENTION

This invention relates to the interpolation of data in a speech synthesis circuit and especially to such speech synthesis circuits integrated on a semiconductor integrated circuit chip.

Several techniques are known in the prior art for digitizing human speech. For example, pulse code modulation, differential pulse code modulation, adaptive predictive coding, delta modulation, channel vocoders, cepstrum vocoders, format vocoders, voice excited vocoders and liner predictive coding techniques of speech digitalization are known. The techniques, are briefly explained in "Voiced Signals: Bit by Bit" on pages 28-34 of the October 1973 issue of IEEE Spectrum.

In certain applications and particularly those in which the digitized speech is to be stored in a memory tend to use the linear predictive coding technique because it produces very high quality speech using rather low data rates. Linear predictive coding systems usually make use of a multi-stage digital filter. In the past, the digital filter has typically been implemented by approximately programming a large scale digital computer. However, in U.S. Pat. application Ser. No. 807,461, filed June 17, 1977 and now abandoned, there is taught a particularly useful digital filter for a speech synthesis circuit, which digital filter may be implemented on an integrated circuit using standard MOS or equivalent technology. A theoretical discussion of linear predictive coding can be found in "Speech Analysis and Synthesis by Linear Predictive of the Speech Wave" at Volumn 50, number 2 (part 2) of The Journal of the Acoustical Society of America.

Disclosed herein is a talking learning aid which utilizes speech synthesis technology for producing human speech. A complete talking learning aid is disclosed, so, in addition to describing the speech synthesis circuits in detail, this patent also discloses the details of the learning aid's controller and the Read-Only-Memory devices used to store the digitized speech. Of course, those practicing the present invention may wish to practice the invention in conjunction with a talking learning aid, such as that described herein, other learning aids or in any other application wherein the generation of human speech from digital data is desirable. Using the techniques described in the aforementioned U.S. Pat. application Ser. No. 807,461 which is now abandoned and the teachings of this patent permit those desiring to make use of digital speech technology to do so with one, or a small number, of relatively inexpensive integrated circuit devices.

This invention relates to interpolation of data in a speech synthesis circuit, as aforementioned. By interpolating the speech data applied to the speech synthesis circuit the data rate required by the synthesis circuit to reproduce speech of a given quality level is effectively reduced. It was, therefore, one object of this invention to provide a speech data parameter interpolator for a voice synthesis circuit, and especially, an interpolator compatible with a synthesis circuit integrated on a semiconductor chip. It was yet another object of this invention to provide an interpolator having a small number of

components so as to take a minimum amount of surface area of the aforementioned chip.

The foregoing objects are achieved as is now described. The speech synthesis circuit includes an input circuit for receiving new target values of various speech parameters and a memory for storing the interpolated values of the parameters. The interpolator includes a subtractor circuit arranged to calculate the difference between the target values of the parameters and the stored values. A portion of the differences calculated are added back to the values stored in the memory, the particular portion being selected according to the formula $\frac{1}{2}N$ where $N=0, 1, 2, \dots$. In the embodiment disclosed, the circuit which performs this division is a delay circuit which preferably delays a serial train of data from the memory by a selectable amount before the difference is added thereto in an adder. The interpolator also preferably includes means for disabling the interpolation in response to changes from voiced to unvoiced speech and visa versa, for instance.

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 a preferred mode of use, 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 accompanying drawings, wherein:

FIG. 1 is a front view of a talking learning aid;

FIG. 2 depicts the segment details of the display;

FIG. 3 is a block diagram of the major components preferably making up the learning aid;

FIGS. 4a and 4b form a composite block diagram (when placed side by side) of the speech synthesizer chip;

FIG. 5 is a timing diagram of various timing signals preferably used on the synthesizer;

FIG. 6 pictorially shows the data compression scheme preferably used to reduce the data rate required by the synthesizer;

FIGS. 7a and 7b form a composite logic diagram of the synthesizer's timing circuits;

FIGS. 8a, 8b and 8c form a composite logic diagram of the synthesizer's ROM/Controller interface logics;

FIGS. 9a and 9b form a composite logic diagram of the interpolator logics;

FIGS. 10a-10b form a composite logic diagram of the array multiplier;

FIGS. 11a and 11b form a composite logic diagram of the speech synthesizer's lattice filter and excitation generator;

FIGS. 12a and 12b are schematic diagrams of the parameter RAM;

FIGS. 13a and 13b are schematic diagrams of the parameter ROM;

FIGS. 14a-14b form a composite diagram of the chirp ROM;

FIGS. 15a-15b form a composite block diagram of a microprocessor or which may be utilized as the controller;

FIGS. 16a and 16b form a composite logic diagram of the segment decoder of the microprocessor;

FIG. 17 depicts the digit output buffers and digit registers of the microprocessor;

FIG. 18 depicts the KB selector circuit of the microprocessor;

FIG. 19 is a block diagram of ROM's 12a, 12b, 13a or 3b;

FIGS. 20a-20e form a composite logic diagram of the control logic for ROMS 12a, 12b, 13a or 13b;

FIGS. 21a and 21b form a composite logic diagram of the X and Y address decoders and the array of memory cells;

FIG. 22 is a plan view of the synthesizer chip herein described, showing the metal mask or metal pattern, enlarged about fifty times;

FIGS. 23a-23b depict embodiments of the voice coil connection.

GENERAL DESCRIPTION

FIG. 1 is a front view of a talking learning aid of the type which may embody the present invention. The learning aid includes a case 1 which encloses electronic circuits preferably implemented on integrated circuits not shown in this figure). These circuits are coupled to display 2, a keyboard 3 and a speaker 4 or other voice coil means (also not shown in FIG. 1). However, the openings 4a are shown behind which speaker 4 is preferably mounted. The display is preferably of the vacuum fluorescent type in the embodiment to be described; however, it will be appreciated by those skilled in the art that other display means, such as arrays of light emitting diodes, liquid crystal devices, electrochromic devices, gas discharge devices or other displays means alternatively may be used if desired. Also, in this embodiment, as a matter of design choice, the display has eight character positions. The keyboard 3 of the learning aid of this embodiment has forty key switch positions, twenty-six of which are used to input the letters of the alphabet into the learning aid. Of the remaining fourteen key switch positions, five are utilized for mode keys (on/spelling mode, learn mode, word guesser game mode, code breaker mode and random letter mode), another five are used to control functions performed by the learning aid in its modes (enter, say gain, replay, erase and go) and the remaining four are used for an apostrophe key, a blank space key, a word list select key and an off key. The words spoken by the learning aid, as well as the correct spelling of those words, are stored as digital information in one or more Read-Only-Memories.

The learning aid depicted in FIG. 1 may be battery powered or powered from a source of external electrical power, as desired. The case is preferably made from injection molded plastic and the keyboard switches may be provided by two 5 by 8 arrays of key switches of the type disclosed in U.S. Pat. No. 4,005,293, if desired. Of course, other types of case materials or switches alternatively may be used.

Having described the outward appearance of the learning aid, the modes in which the learning aid may operate will be first described followed by a description of the block diagrams and detailed logic diagrams of the various electronic circuits used to implement the learning aid of FIG. 1.

MODES OF OPERATION

The learning aid of this embodiment has five modes of operation which will be subsequently described. It will be evident to those skilled in the art, however, that these modes of operation may be modified, reduced in number or expanded in capability. As a matter of design choice, the present talking and learning aid is provided with the following modes of operation.

The first mode, the spelling mode, is automatically entered when the "on" key is depressed. In the spelling mode the learning aid randomly selects ten words from a selected word list and at a selected difficulty category within the selected word list. The word list may be changed by depressing the "word list select" key which is coupled to a software implemented flip flop circuit which flips each time the "word list select" key is depressed. The word list select flip flop then determines, 10 as will be seen, which pair of read-only-memories from which the ten words will be randomly selected. Each word list preferably includes words arranged in four levels of difficulty. This embodiment of the learning automatically enters the least difficult level of difficulty. 15 The fact that the least difficulty level has been selected is shown by displaying "SPELL A" in display 2. The level difficulty may be increased by depressing the B, C or D keys, and display 2 will show, in response, "SPELL B", "SPELL C" or "SPELL D", respectively. Having selected the word list and level difficulty, the "go" key is depressed upon which the learning aid commences to randomly select ten words and to say the word "spell" followed by the first randomly selected word. A dash, that being segment D in display 20 2 (FIG. 2), comes up in the left hand most character position. At this time the student may either (1) enter his or her spelling of the word and then depress the "enter" key or (2) depress the "say again" key. The student may also depress the "erase" key if he or she realizes that the spelling being inputted is incorrect before having depressed the "enter" key; the student may then again try to input the correct spelling. The "say again" key causes the word to be spoken by the learning aid again. In some embodiments a subsequent depression of the "say again" key may cause the selected word to be repeated once more, however, then at a slower rate. As the student enters his or her spelling of the word using the alphabet keys at keyboard 3, the inputted spelling appears at display 2 and the shifts from left to right as the letters are inputted. Following the depression the "enter" key, the learning aid compares the student's spelling with a correct spelling, which is stored in one of the Read-Only-Memories, and verbally indicates to the student whether the student spelling was correct or 25 incorrect. The verbal response is also stored as digital information in a Read-Only-Memory. Of course, a visual response may likewise or alternatively be used, if desired. In this embodiment the student is given two opportunities to spell the word correctly and if the student has still failed to correctly spell the word, the learning aid then verbally (via speaker 4) and visually (via display 2) spells the word for the student and goes on to the next word from the group of ten randomly selected words. 30

At the end the test of the spelling of the ten randomly selected words, the learning aid then verbally and visually indicates the number of right and wrong answers. Further, in order to give the student additional reinforcement, the learning aid preferably gives a audible response which is a function of the correctness of the spellings. In this embodiment the learning aid plays a tune, the number of notes of which is a function of the correctness of the student's spellings for the group of selected words. The use of the "enter", "say again", "erase", and "go" function keys has just been described with reference to the spelling mode of operation. There is an additional function key, "replay", whose function has not yet been described. The "replay" key causes the 35

60

65

learning aid to repeat the group of ten randomly selected words after the group has been completed or causes the learning aid to start over with the first word of the group of ten words if it is depressed during the progression through the group. Alternatively, at the end of a group of ten words, the student may depress the "go" which initiates the random selection of another group of ten words from the selected word list.

An exemplary set of spell mode problems is shown in Table I; exemplary key depressions, which a student might make during the exemplary set of problems, are listed along with the responses made by the learning aid at display 2 and speaker 4.

The learn mode is entered by depressing the "learn" key. In the learn mode, after the "go" key is depressed 15 the learning aid randomly selects ten words from the selected word list at the selected difficulty level and then proceeds to display the first randomly selected word at display 2 and approximately one second later to speak "say it". Approximately two seconds thereafter 20 the learning proceeds to pronounce the word shown in display 2. During this interval the student is given the opportunity to try to pronounce the word spelled at display 2; the learning aid then goes on to demonstrate how the word should be pronounced. After going 25 through the ten randomly selected words the learning automatically returns to the aforementioned spell mode, but the ten words tested during the spell mode are the ten words previously presented during the learn mode. While in the learn mode the "say again", "erase", "repeat" and "enter" keys are invalid. The difficulty level is selected as in the spelling mode, but in the learn mode the learning aid displays the various levels as "SAY IT A", "SAY IT B", etc. Depressing the "go" key causes 30 the learning aid to select another group of ten words in the learn mode. An exemplary set of learn mode problems are set forth in Table II.

The word guesser mode is entered by depressing the "word guesser" mode key. In the word guesser mode the learning aid randomly selected a word from the 40 selected word list and displays dashes in a number of character positions at display 2, the number of character positions corresponding to the number of letters in the randomly selected word. Thus, if the learning aid randomly selects the word "course" for instance, then the 45 dashes will appear in six of the eight character positions in display 2, starting with the left most position and proceeding to the right for six character positions. The dash is shown in the characters of the display by energizing the D segments in those character positions (see FIG. 2). The child may then proceed to enter his or her 50 guesses of the letters in the randomly selected word by depressing the letter keys at keyboard 2. For a correct choice, the learning aid gives an audible response of four tones and shows every place the chosen letter occurs in the randomly selected word. Once letters have been correctly guessed, they remain in the display until the end of the game. For incorrect guesses the learning aid preferably makes no response, but may alternatively say something like "incorrect guess." In 55 60 this embodiment the child is given six incorrect guesses. Upon the seventh incorrect guess the learning says "I win". On the other hand, if the child correctly guesses all the letters before making seven incorrect guesses the learning aid speaks "you win" and gives an audible response of four tones. Thus in the word guesser mode, the learning aid permits the child to play the traditional spelling game known as "hangman" either by himself or

herself or along with other children. Exemplary word guesser problems are set forth in Table III.

The disclosed learning aid has another mode of operation known as "code breaker" which is entered by 5 depressing the "code breaker" mode key. In this mode the child may enter any word of his or her choice and upon depressing the "enter key" the letters in the display are exchanged according to a predetermined code. Thus, in the code breaker mode the learning aid may be used to encode words selected by the child. Further in the code breaker mode the learning aid may be used to decode the encoded words by entering the encoded word and depressing the "enter key".

Another mode with which the learning aid may be provided is the "random letter" mode which is entered by depressing the "random letter" key. In the random letter mode the learning automatically displays in response to depression of the "go" key a randomly selected letter of the alphabet in the first character position of display 2. The letters of the alphabet occur in approximate proportion to as they occur in the english language; thus, the more commonly letters are displayed more frequently than uncommonly used letters. If the "go" key is again depressed then another randomly selected letter is displayed in the first character position and the previously selected letter moves right to the second character position and so forth in response to further depressions of the "random letter" key.

Referring now to FIG. 2, there is shown a preferred arrangement of the segments of display 2. Display 2 preferably has eight character positions each of which is provided by a sixteen segment character having fourteen segments arranged somewhat like a "British flag" with an additional two segments for an apostrophe and a decimal point. In FIG. 2, segments a-n are arranged more or less in the shape of the "British flag" while segment approves apostrophe and segment dpt provides a decimal point. Segment conductors Sa through Sn, Sdp and Sap are respectively coupled to segments a through n, dpt and ap in the eight character positions of display 2. Also, for each character position, there is a common electrode, labeled as D1-D8. When display 2 is provided by a vacuum fluorescent display device, the segments electrodes are provided anodes in the vacuum fluorescent display device while each common electrode is preferably provided by a grid associated with each character position. By appropriately multiplexing signals on the segment conductors (Sa-Sn, Sdp and Sap) with signals on the character common electrodes (D1-D8) the display may be caused to show the various letters of the alphabet, a period, and an apostrophe and various numerals. For instance, by appropriately energizing segment conductors A,B,C,E and F when character common electrode D1 is appropriately energized the letter A is actuated in the first character position of display 2. Further, by appropriate strobing segment conductors A,B,C,D,H,I and J when character common electrode D2 is appropriately energized, the letter B is caused to be actuated in the second character position of display 2. It should be evident to those skilled in the art that the other letters of the alphabet as well as the apostrophe, period and numerals may be formed by appropriate energization of appropriate segment conductors and common electrodes. In operation, the character common electrodes D1-D8 are sequentially energized with an appropriate voltage potential as selected segment conductors are energized to their appropriate voltage potential to produce a display of characters at

display 2. Of course, the segment electrodes could alternatively be sequentially energized as the digit electrodes are selectively energized in producing a display at display 2.

BLOCK DIAGRAM OF THE LEARNING AID

FIG. 3 is a block diagram of the major components making up the disclosed embodiment of a speaking learning aid. The electronics of the disclosed learning aid may be divided into three major functional groups, one being a controller 11, another being a speech synthesizer 10, and another being a read-only-memory (ROM) 12. In the embodiment disclosed, these major electronic functional groups are each integrated on separate integrated circuit chips except for the ROM functional group which is integrated onto two integrated circuit chips. Thus, the speech synthesizer 10 is preferably implemented on a single integrated circuit denoted by the box labeled 10 in FIG. 3 while the controller is integrated on a separate integrated circuit denoted by a box 11 in FIG. 3. The word list for the learning aid is stored in the ROM functional group 12, which stores both the correct spellings of the words as well as frames of digital coding which are converted by speech synthesizer 10 to an electrical signal which drives speaker or other voice coil means 4. In the embodiment disclosed, ROM functional group 12 is preferably provided with 262,144 bits of storage. As a matter of design choice, the 262,144 bits of data is divided between two separate read-only-memory chips, represented in FIG. 3 at numerals 12a and 12b. The memory capacity of ROM functional group 12 is a design choice; however, using the data compression features which are subsequently discussed with reference to FIG. 6, the 262,144 bits of read-only-memory may be used to store on the order of 250 words of spoken speech and their correct spellings as well as various tones, praise phrases and correction phases spoken by the learning aid.

As is discussed with reference to FIG. 1, the "word list select" key causes the learning aid to select words from another word list. In FIG. 3, the basic word list used with the learning aid is stored in ROMs 12a and 12b along with their spellings and appropriate phrasology which the learning aid speaks during its different modes of operation. The second word list, which may be selected by depressing the "word list select" key, is preferably stored in another pair of ROMs 13a and 13b. In FIG. 3 these are depicted by dashed lines because these read-only-memories are preferably plugged into the learning aid by a person using the system (of course, when children use the system it is preferably that an adult change the read-only-memories since children may not have the required manual dexterity) rather than normally packaged with the learning aid. In this manner many different "libraries" of word lists may be made available for use with the learning aid.

Of course, the number of chips on which the learning aid is implemented is a design choice and as large scale integration techniques are improved (using electron beam etching and other techniques), the number of integrated circuit chips may be reduced from four to as few as a single chip.

Synthesizer chip 10 is interconnected with the read-only-memories via data path 15 and is interconnected with controller 11 via data path 16. The controller 11, which may be provided by an appropriately programmed microprocessor type device, preferably actuates display 2 by providing segment information on

segment conductors Sa-Sn, Sdpt and Sap along with character position information on connectors D1-D8. In the embodiment herein disclosed, controller 11 preferably also provides filament power to display 2 when a vacuum fluorescent device is used therefor. Of course, if a liquid crystal, electrochromographic, light emitting diode or gas discharge display were used such filament power would not be required. One technique for generating filament power on a controller chip is described in U.S. Pat. application Ser. No. 843,017 filed Oct. 17, 1977. Controller 11 also scans keyboard 3 for detecting key depressions thereat. Keyboard 3 has forty switch positions which are shown in representative form in FIG. 3, the switch locations occurring where the conductors cross within the dashed line at numeral 3 in FIG. 3. A switch closure causes the conductors shown as crossing in FIG. 3 to be coupled together. At numeral 3' the switch occurring at a crossing of conductors at numeral 3 is shown in detail. In addition to actuating display 2 and sensing key depression at keyboard 3, controller 11 also performs such functions as providing addresses for addressing ROMs 12a and 12b (via synthesizer 10), comparing the correct spellings from ROMs 12a or 12b with spellings inputted by a student at keyboard 3, and other such functions which will become apparent. Addresses from controller 11 are transmitted to ROMs 12a-b by synthesizer 10 because, as will be seen, synthesizer 10 preferably is equipped with buffers capable of addressing a plurality of read-only-memories. Preferably, only one of the pairs of ROMs will output information in response to this addressing because of a chip select signal which is transmitted from synthesizer 10 to all the Read-Only-Memories. Controller 11, in this embodiment, transmits addresses to the ROMs via synthesizer 10 so that only synthesizer 10 output buffers need be sized to transmit addresses to a plurality of ROMs simultaneously. Of course, controller 11 output buffers could also be sized to transmit information to a plurality of read-only-memories simultaneously and thus in certain embodiments it may be desirable to also couple controller 11 directly to the ROMs.

As will be seen, synthesizer chip 10 synthesizes human speech or other sounds according to frames of data stored in ROMs 12a-12b or 13a-13b. The synthesizer 10 employs a digital lattice filter of the type described in U.S. Pat. application Ser. No. 807,461, filed June 17, 1977. U.S. patent application Ser. No. 807,461, since abandoned and continued in U.S. patent application Ser. No. 905,328, filed May 12, 1978, is hereby incorporated herein by reference. The following discussion of the speech synthesizer assumes that the reader has a basic understanding of the operation of the lattice filter described in U.S. patent application Ser. No. 807,451, since abandoned and continued in U.S. patent application Ser. No. 905,328, filed May 12, 1978; therefore the reader is encouraged to read that patent before delving into the following detailed discussion of the speech synthesizer. As will also be seen, synthesizer 10 also includes a digital to analog (D to A) converter for converting the digital output from the lattice filter to analog signals for driving speaker 4 or other voice coil means with those analog signals. Synthesizer 10 also includes timing, control and data storage and data compression systems which will be subsequently described in detail.

SYNTHESIZER BLOCK DIAGRAM

FIGS. 4a and 4b form a composite block diagram of the synthesizer 10. Synthesizer 10 is shown as having six major functional blocks, all but one of which are shown in greater detail in block diagram form in FIGS. 4a and 4b. The six major functional blocks are timing logic 20; ROM-Controller interface logic 21; parameter loading, storage and decoding logic 22; parameter interpolater 23; filter and excitation generator 24 and D to A and output section 25. Subsequently, these major functional blocks will be described in detail with respect to FIGS. 5a-b, 6, 7a-b, 8a-c, 9a-b, 10a-d and 11a-b.

Rom/Controller Interface Logic

Referring again to FIGS. 4a and 4b, ROM/Controller interface logic 21 couples synthesizer 10 to read-only-memories 12a and 12b and to controller 11. The control 1-8 (CTL1-CTL8), chip select (CS) and processor data clock (PDC) pins are coupled, in this embodiment, to the controller while the address 1-8 (ADD1-ADD8) and instruction 0-1 (I0-I1) pins are connected to ROMs 12a and 12b (as well as ROMs 13a-13b, if used). ROM/Controller interface logic 21 sends address information from controller 11 to the Read-Only-Memories 12a-12b and preferably returns digital information from the ROMs back to the controller 12; logic 21 also brings data back from the ROMs for use by synthesizer 10 and initiates speech. A Chip Select (CS) signal enables tristate buffers, such as buffers 213, and a three bit command latch 210. A Processor Data Clock (PDC) signal sets latch 210 to hold the data appearing at CTL1-CTL4 pins from the controller. Command latch 210 stores a three bit command from controller 11, which is decoded by command decoder 211. Command decoder 211 is responsive to eight commands which are: speak (SPK) or speak slowly (SPKSLOW) for causing the synthesizer to access data from the Read-Only-Memory and speak in response thereto either at a normal rate or at a slow rate; a reset (RST) command for resetting the synthesizer to zero; a test talk (TTALK) so that the controller can ascertain whether or not the synthesizer is still speaking; a load address (LA) where four bits are received from the controller chip at the CTL1-CTL8 pins and transferred to the ROMs as an address digit via the ADD1-ADD8 pins and associated buffers 211; a read and branch (RB) command which causes the Read-Only-Memory to take the contents of the present and subsequent address and use that for a branch address; a read (RE) command which causes the Read-Only-Memory to output one bit of data on ADD1, which data shifts into a four bit data input register 212; and an output command which transfers four bits of data in the data input register 212 to controller 11 via buffers 213 and the CTL1-CTL8 pins. Once the synthesizer 10 has commenced speaking in response to a SPK or SPKSLOW command it continues speaking until ROM interface logic 21 encounters a RST command or an all ones gate 207 (see FIG. 7a-7b) 60 detects an "energy equal to fifteen" code and resets talk latch 216 in response thereto. As will be seen, an "energy equal to 15" code is used as the last frame of data in a plurality of frames of data for generating words, phases or sentences. The LA, RE and RB commands decoded by decoder 211 are re-encoded via ROM control logic 217 and transmitted to the read-only-memories via the instruction (I0-I1) pins.

The processor Data Clock (PDC) signal serves other purposes than just setting latch 210 with the data on CTL1-CTL4. It signals that an address is being transferred via CTL1-CTL8 after an LA or output command has been decoded or that the TTALK test is to be performed and outputted on pin CTL8. A pair of latches 218A and B (FIGS. 7a-7b) associated with decoder 211 disable decoder 211 when the aforementioned LA, TTALK and OUTPUT commands have been decoded and a subsequent PDC occurs so that the data then on pins CTL1-CTL8 is not decoded.

A TALK latch 216 is set in response to a decoded SPK or SPKSLW command and is reset: (1) during a power up clear (PUC) which automatically occurs whenever the synthesizer is energized; (2) by a decoded RST command or (3) by an "energy equals fifteen" code in a frame of speech data. The TALKD output is delayed output to permit all speech parameters to be inputted into the synthesizer before speech is attempted. 20 The talk slow latch 215 is set in response to a decoded SPKSLOW command and reset in the same manner as latch 216. The SLOWD output is similarly a delayed output to permit all the parameters to be inputted into the synthesizer before speech is attempted.

Parameter Loading, Storage and Decoding Logic

The parameter loading, storage and decoding logic 22 includes a six bit long parameter input register 205 which receives serial data from the read-only-memory via pin ADD1 in response to a RE command outputted to the selected read-only-memory via the instruction pins. A coded parameter random access memory (RAM) 203 and condition decoders and latches 208 are connected to receive the data inputted into the parameter input register 205. As will be seen, each frame of speech data is inputted in three to six bit portions via parameter input register 205 to RAM 203 in a coded format where the frame is temporarily stored. Each of the coded parameters stored in RAM 203 are converted to a ten bit parameter by parameter ROM 202 and temporarily stored in a parameter output register 201.

As will be discussed with respect to FIG. 6, the frames of data may be either wholly or partially inputted into parameter input register 205, depending upon the length of the particular frame being inputted. Condition decoders and latches 208 are responsive to particular portions of the frame of data for setting repeat, pitch equal zero, energy equal zero, old pitch and old energy latches. The function of these latches will be discussed subsequently with respect to FIGS. 7a-7b. The condition decoders and latches 208 as well as various timing signals are used to control various interpolation control gates 209. Gates 209 generate an inhibit signal when interpolation is to be inhibited, a zero parameter signal when the parameter is to be zeroed and a parameter load enable signal which, among other things, permits data in parameter input register 205 to be loaded into the coded parameter RAM 203.

Parameter Interpolator

The parameters in parameter output registers 201 are applied to the parameter interpolator functional block 23. The inputted K1-K10 speech parameters, including speech energy are stored in a K-stack 302 and E10 loop 304, while the pitch parameter is stored in a pitch register 305. The speech parameters and energy are applied via recoding logic 301 to array multiplier 401 in the filter and excitation generator 24. As will be seen, how-

ver, when a new parameter is loaded into parameter output register 201 it is not immediately inserted into K-stack 302 or E10 loop 304 or register 305 but rather the corresponding value in K-stack 302, E10 loop 304 or register 305 goes through eight interpolation cycles during which a portion of the difference between the present value in the K-stack, E10 loop 305 or register 05 and the target value of that parameter in parameter output register 201 is added to the present value in K-stack 203, E10 loop 304 or register 305.

Essentially the same logic circuits are used to perform the interpolation of pitch, energy and the K1-K10 speech parameters. The target value from the parameter output register 201 is applied along with the present value of the corresponding parameter to a subtractor 308. A selector 307 selects either the present pitch from pitch logic 306 or present energy or K coefficient data from KE10 transfer register 303, according to which parameter is currently in parameter output register 201, and applies the same to subtractor 308 and a delay circuit 309. As will be seen, delay circuit 309 may provide anywhere between zero delay to three bits of delay. The output of delay circuit 309 as well as the output of subtractor 308 is supplied to an adder 310 whose output is applied to a delay circuit 311. When the delay associated with delay circuit 309 is zero the target value of the articulatory parameter in parameter output register 201 is effectively inserted into K-stack 302, E10 loop 304 or pitch register 305, as is appropriate. The delay in delay circuit 311 is three to zero bits, being three bits when the delay in the delay circuit 309 is zero bits, whereby the total delay through selector, 307, delay 309 and 311, adder 310 and subtractor 308 is constant. By controlling the delays in delay circuit 309 and 311, either all, $\frac{1}{2}$, $\frac{1}{4}$ or of the difference outputted from subtractor 308 (that being the difference between the target value and the present value) is added back into the present value of the parameter. By controlling the delays in the fashion set forth in Table IV, a relatively smooth eight step parameter interpolation is accomplished.

U.S. patent application Ser. No. 807,461, since abandoned and continued in U.S. patent application Ser. No. 905,328, filed May 12, 1978, discusses with reference to FIG. 7 thereof a speech synthesis filter wherein speech coefficients K1-K9 are stored in the K-stack continuously, until they are updated, while the K10 coefficient and the speech energy (referred to by the letter A in U.S. patent application Ser. No. 807,461 since abandoned and continued in U.S. patent application Ser. No. 905,328, filed May 12, 1978) are periodically exchanged. The parameter interpolator 23, speech coefficients K1-K9 are likewise stored in stack 302, until they are updated, whereas the energy parameter and the K10 coefficient effectively exchange places in K-stack 302 during a twenty time period cycle of operations in the filter and excitation generator 24. To accomplish this function, E10 loop 304 stores both the energy parameter and the K10 coefficient and alternately inputs the same to the appropriate location in K-stack 302. KE10 transfer register 303 is either loaded with the K10 or energy parameter from E10 loop 304 or the appropriate K1-K9 speech coefficient from K-stack 302 for interpolation by logics 307-311.

As will be seen, recoding logic 301 preferably performs a Booth's algorithm on the data from K-stack 302, before such data is applied to array multiplier 401. Recoding logic 301 thereby permits the size of the array multiplier 401 to be reduced compared to the array

multiplier described in U.S. patent application Ser. No. 807,461, since abandoned and continued in U.S. patent application Ser. No. 905,328, filed May 12, 1978.

Filter and Excitation Generator

The filter excitation generator 24 includes the array multiplier 401 whose output is connected to a summer multiplexer 402. The output of summer multiplexer 402 is coupled to the input of summer 404 whose output is 10 coupled to a delay stack 406 and multiplier multiplexer 405. The output of the delay stack is applied as an input to summer multiplexer 402 and to Y latch 403. The output of Y latch 403 is coupled to an input of multiplier multiplexer 405 along with truncation logic 501. The 15 output of multiplier multiplexer 405 is applied as an input to array multiplier 401. As will be seen filter and excitation generator 24 make use of the lattice filter described in U.S. patent application Ser. No. 807,461, since abandoned and continued in U.S. patent application Ser. No. 905,328, filed May 12, 1978. Various minor interconnections are not shown in FIG. 4b for sake of clarity, but which will be described with reference to FIGS. 10a, 10b, 11a and 11b. The arrangement of the foregoing elements generally agrees with the arrangement shown in FIG. 7 of U.S. patent application Ser. No. 807,461, since abandoned and continued in U.S. patent application Ser. No. 905,328, filed May 12, 1978; thus array multiplier 401 corresponds to element 30', summer multiplexer 402 corresponds to elements 37b', 37c' and 37d', gates 414 (FIGS 11a and 11b) correspond to element 33', delay stack 406 corresponds to elements 34' and 35', Y latch 403 corresponds to element 36' and multiplier multiplexer 405 corresponds to elements 38a', 38b', 38c' and 38d'.

The voice excitation data is supplied from unvoiced/voice gate 408. As will be subsequently described in greater detail, the parameters inserted into parameter input gate 205 are supplied in a compressed data format. According to the data compression scheme used, when 40 the coded pitch parameter is equal zero in input register 205, it is interpreted as an unvoiced condition by condition decoders and latches 208. Gate 408 responds by supplying randomized data from unvoiced generator 407 as the excitation input on line 414. When the coded pitch parameter is of some other value, however, it is decoded by parameter ROM 202, loaded into parameter output register 201 and eventually inserted into pitch register 305, either directly or by the interpolation scheme previously described. Based on the period indicated by the number in pitch register 305, voiced excitation is derived from chirp ROM 409. As discussed in U.S. patent application Ser. No. 807,461, since abandoned and continued in U.S. patent application Ser. No. 905,328, filed May 12, 1978, the voiced excitation signal 45 may be an impulse function or some other repeating function such as a repeating chirp function. In this embodiment, a chirp has been selected as this tends to reduce the "fuzziness" from the speech generated (because it apparently more closely models the action of the vocal cords than does an impulse function) which chirp is repetitively generated by chirp ROM 409. Chirp ROM 409 is addressed by counter latch 410, whose address is incremented in an add one circuit 411. The address in counter latch 410 continues to increment in add one circuit 411, recirculating via reset logic 412 until magnitude comparator 413, which compares the magnitude of the address being outputted from add one circuit 411 and the contents of the pitch register 305,

indicates that the value in counter latch 410 then compares with or exceeds the value in pitch register 305, at which time reset logic 412 zeroes the address in counter 410. Beginning at address zero and extending through approximately fifty addresses is the chirp function in chirp ROM 409. Counter latch 401 and chirp ROM 409 are set up so that addresses larger than fifty do not cause any portion of the chirp function to be outputted from chirp ROM 409 to UV gate 408. In this manner the chirp function is repetitively generated on a pitch related period during voiced speech.

SYSTEM TIMING

FIG. 5 depicts the timing relationships between the occurrences of the various timing signals generated on synthesizer chip 10. Also depicted are the timing relationships with respect to the time new frames of data are inputted to synthesizer chip 10, the timing relationship with respect to the interpolations performed on the inputted parameters, the timing relations with respect to the foregoing with the time periods of the lattice filter and the relationship of all the foregoing to the basic clock signals.

The synthesizer is preferably implemented using precharged, conditional discharge type logics and therefore FIG. 5 shows clocks ϕ_1 - ϕ_4 which may be appropriately used with such precharge-conditional discharge logic. There are two main clock phases (ϕ_1 and ϕ_2) and two precharge clock phases (ϕ_3 and ϕ_4). Phase ϕ_3 goes low during the first half of phase ϕ_1 and serves as a precharge therefor. Phase ϕ_4 goes low during the first half of phase ϕ_2 and serves as a precharge transfer. A set of clocks ϕ_1 - ϕ_4 required to clock one bit of data and thus correspond to one time period.

The time periods are labeled T1-T20 and each preferably has a time period on the order of five microseconds. Selecting a time period on the order of five microseconds permits, as will be seen, data to be outputted from the digital filter at a ten kilohertz rate (i.e., at a 100 microsecond period) which provides for a frequency response of five kilohertz in the D to A output section 25 (FIG. 4b). It will be appreciated by those skilled in the art, however, that depending on the frequency response which is desired and depending upon the number of Kn speech coefficients used, and also depending upon the type of logics used, that the periods or frequencies of the clocks and clock phases shown in FIG. 5 may be substantially altered, if desired.

As is explained in U.S. patent application Ser. No. 807,461, since abandoned and continued in U.S. patent application Ser. No. 905,328, filed May 12, 1978, one cycle time of the lattice filter in filter excitation generator 24, preferably comprises twenty time periods, T1-T20. For reasons not important here, the numbering of these time periods differs between this application and U.S. patent application Ser. No. 807,461, since abandoned and continued in U.S. patent application Ser. No. 905,328, filed May 12, 1978. To facilitate the reader's understanding of the differences in the numbering of the time periods, both numbering schemes are shown at the time period time line 500 in FIG. 5. At time line 500, the time periods, T1-T20 which are not enclosed in parenthesis identify the time periods according to the convention used in this application. On the other hand, the time periods convention used in U.S. patent application Ser. No. 807,461, since abandoned and continued in U.S. patent application Ser. No.

905,328, filed May 12, 1978. Thus, time period T17 is equivalent to time period (T9).

At numeral 501 is depicted the parameter count (PC) timing signals. In this embodiment there are thirteen PC signals, PC=0 through PC=12. The first twelve of these, PC=0 through PC=11 correspond to times when the energy, pitch, and K1-K10 parameters, respectively, are available in parameter output register 201. Each of the first twelve PC's comprise two cycles, which are labeled A and B. Each such cycle starts at time period T17 and continues to the following T17. During each PC the target value from the parameter output register 201 is interpolated with the existing value in K-stack 302 in parameter interpolator 23. During the A cycle, the parameter being interpolated is withdrawn from the K-stack 302, E10 loop 304 or register 305, as appropriate, during an appropriate time period. During the B cycle the newly interpolated value is reinserted in the K-stack (or E10 loop or pitch register). The thirteenth PC, PC=12, is provided for timing purposes so that all twelve parameters are interpolated once each during a 2.5 milliseconds interpolation period.

As was discussed with respect to the parameter interpolator 23 of FIG. 4b and Table IV, eight interpolations are performed for each inputting of a new frame of data from ROMs 12a-b into synthesizer 10. This is seen at numeral 502 of FIG. 5 where timing signals DIV 1, DIV 2, DIV 4 and DIV 8 are shown. These timing signals occur during specific interpolation counts (IC) as shown. There are eight such interpolation counts, IC0-IC7. New data is inputted from the ROMs 12a-b into the synthesizer during IC0. These new target values of the parameters are then used during the next eight interpolation counts, IC1 through IC8; the existing parameters in the pitch register 305 K-stack 302 and E10 loop 304 are interpolated once during each interpolation count. At the last interpolation count, IC8, the present value of the parameters in the pitch register 305, K-stack 302 and E10 loop 304 finally attain the target values previously inputted toward the last IC8 and thus new target values may then again be inputted as a new frame of data. Inasmuch as each interpolation count has a period of 2.5 milliseconds, the period at which new data frames are inputted to the synthesizer chip is 20 microseconds or equivalent to a frequency of 50 hertz. The DIV 8 signal corresponds to those interpolation counts in which one-eighth of the difference produced by subtractor 308 is added to the present values in adder 310 whereas during DIV 4 one-fourth of the difference is added in, and so on. Thus, during DIV 2, $\frac{1}{2}$ of the difference from subtractor 308 is added to the present value of the parameter in adder 310 and lastly during DIV 1 the total difference is added in adder 310. As has been previously mentioned, the effect of this interpolation scheme can be seen in Table IV.

PARAMETER DATA COMPRESSION

It has been previously mentioned that new parameters are inputted to the speech synthesizer at a 50 hertz rate. It will be subsequently seen that in parameter interpolator and excitation generator 24 (FIG. 4b) the pitch data, energy data and K1-Kn parameters are stored and utilized as ten bit digital binary numbers. If each of these twelve parameters were updated with a ten bit binary number at a fifty hertz rate from an external source such as ROMs 12a and 12b, this would require $12 \times 10 \times 50$ or 6,000 hertz bit rate. Using the data com-

ression techniques which will be explained, we reduce his bit rate required for synthesizer 10 to on the order f 1,000 to 1,200 bits per second. And more importantly, has been found that the speech compression schemes herein disclosed do not appreciably degrade the quality f speech generated thereby in comparison to using the ata uncompressed.

The data compression scheme used is pictorially shown in FIG. 6. Referring now to FIG. 6, it can be seen that there is pictorially shown four different lengths of frames of data. One, labeled voiced frame, as a length of 49 bits while another entitled unvoiced frame, has a length of 28 bits while still another called "repeat frame" has a length of ten bits and still another which may be alternatively called zero energy frame or energy equals fifteen frame has the length of but four bits. The "voiced frame" supplies four bits of data for a coded energy parameter as well as coded four bits for each five speech parameters K3 through K7. Five bits of data are reserved for each of three coded parameters, pitch, K1 and K2. Additionally, three bits of data are provided for each of three coded speech parameters K8-K10 and finally another bit is reserved for a repeat t.

In lieu of inputting ten bits of binary data for each of the parameters, a coded parameter is inputted which is converted to a ten bit parameter by addressing parameter ROM 202 with the coded parameter. Thus, coefficient K1, for example, may have any one of thirty-two different values, according to the five bit code for K1, with one of the thirty-two values being a ten bit numerical coefficient stored in parameter ROM 202. Thus, the actual values of coefficients K1 and K2 may have one of thirty-two different values while the actual values of coefficients K3 through K7 may be one of sixteen different values and the values of coefficients K8 through K9 may be one of eight different values. The coded pitch parameter is five bits long and therefore may have up to forty-two different values. However, only thirty-one of these reflect actual pitch values, a pitch code of 00000 being used to signify an unvoiced frame of data. The coded energy parameter is four bits long and therefore could normally have sixteen available ten bit values; however, a coded energy parameter equal to 0000 indicates a silent frame such as occur as pauses in and between words, sentences and the like. A coded energy parameter equal to 1111 (energy equals fifteen), on the other hand, is used to signify the end of a segment of spoken speech, thereby indicating that the synthesizer is stop speaking. Thus, of the sixteen codes available for a coded energy parameter, fourteen are used to signify different ten bit speech energy levels.

Coded coefficients K1 and K2 have more bits than coded coefficients K3-K7 which in turn have more bits than coded coefficients K8 through K10 because coefficient K1 has a greater effect on speech than K2 which has a greater effect on speech than K3 and so forth through the lower order coefficients. Thus given the greater significance of coefficients K1 and K2 than coefficients K3 through K10, for example, more bits are used in coded format to define coefficients K1 and K2 than K3-K7 or K8-K10.

Also it has been found that voiced speech data needs more coefficients to correctly model speech than does unvoiced speech and therefore when unvoiced frames are encountered, coefficients K5 through K10 are not updated, but rather are merely zeroed. The synthesizer

realizes when an unvoiced frame is being outputted because the uncoded pitch parameter is equal to 00000.

It has also been found that during speech there often occur instances wherein the parameters do not significantly change during a twenty millisecond period; particularly, the K1-K10 coefficients will often remain nearly unchanged. Thus, a repeat frame is used wherein new energy and new pitch are inputted to the synthesizer, however, the K1-K10 coefficients previously inputted remain unchanged. The synthesizer recognizes the ten bit repeat frame because the repeat bit between energy and pitch then comes up whereas it is normally off. As previously mentioned, there occur pauses between speech or at the end of speech which are preferably indicated to the synthesizer; such pauses are indicated by a coded energy frame equal to zero, at which time the synthesizer recognizes that only four bits are to be sampled for that frame. Similarly, only four bits are sampled when an "energy equals fifteen." Using coded values for the speech in lieu of actual values, alone would reduce the data rate to 48×50 or 2400 bits per second. By additionally using variable frame lengths, as shown in FIG. 6, the data rate may be further reduced to on the order of one thousand to twelve hundred bits per second, depending on the speaker and on the material spoken.

The effect of this data compression scheme can be seen from Table V where the coding for the word "HELP" is shown. Each line represents a new frame of data. As can be seen, the first part of the word "HELP", "HEL", is mainly voiced while the "P" is unvoiced. Also note the pause between "HEL" and "P" and the advantages of using the repeat bit. Table VI sets forth the encoded and decoded speech parameter. The 3, 4 or 5 bit code appears as a hexadecimal number in the left-hand column, while the various decoded parameter values are shown as ten bit, two's complement numbers expressed as hexadecimal numbers in tabular form under the various parameters. The decoded speech parameter are stored in ROM 203. The repeat bit is shown in Table V between the pitch and K parameters for sake of clarity; preferably, according to the embodiment of FIG. 6, the repeat bit occurs just before the most significant bit (MSB) of the pitch parameter.

SYNTHESIZER LOGIC DIAGRAMS

The various portions of the speech synthesizer of FIGS. 4a and 4b will now be described with reference to FIGS. 7a through 14b which, depict, in detail, the logic circuits implemented on a semiconductor chip, for example, to form the synthesizer 10. The following discussion, with reference to the aforementioned drawings, refers to logic signals available at many points in the circuit. It is to be remembered that in P channel MOS devices a logical zero corresponds to a negative voltage, that is, Vdd, while a logical one refers to a zero voltage, that is, Vss. It should be further remembered that P-channel MOS transistors depicted in the aforementioned figures are conductive when a logical zero, that is, a negative voltage, is applied at their respective gates. When a logic signal is referred to which is unbarred, that is, has no bar across the top of it, the logical signal is to be interpreted as "TRUE" logic; that is, a binary one indicates the presence of the signal (Vss) whereas a binary zero indicates the lack of the signal (Vdd). Logic signal names including a bar across the top thereof are "FALSE" logic; that is, a binary zero (Vdd voltage) indicates the presence of the signal

whereas a binary one (V_{ss} voltage) indicates that the signal is not present. It should also be understood that a numeral three in clocked gates indicates that phase ϕ_3 is used as a precharge whereas a four in a clocked gate indicates that phase ϕ_4 is used as a precharge clock. An "S" in the gate indicates that the gate is statically operated.

Timing Logic Diagram

Referring now to FIGS. 7a and 7b, they form a composite, detailed logic diagram of the timing logic for synthesizer 10. Counter 510 is a pseudorandom shift counter including a shift register 510a and feed back logic 510b. The counter 510 counts into pseudorandom fashion and the TRUE and FALSE outputs from shift register 510a are supplied to the input section 511 of a timing PLA. The various T time periods decoded by the timing PLA are indicated adjacent to the output lines thereof. Section 511c of the timing PLA is applied to an output timing PLA 512 generating various combinations and sequences of time period signals, such as T odd, T10-T18, and so forth. Sections 511a and 511b of timing PLA 511 will be described subsequently.

The parameter count in which the synthesizer is operating is maintained by a parameter counter 513. Parameter counter 513 includes an add one circuit and circuits which are responsive to SLOW and SLOW D. In SLOW, the parameter counter repeats the A cycle of the parameter count twice (for a total of three A cycles) before entering the B cycle. That is, the period of the parameter count doubles so that the parameters applied to the lattice filter are updated and interpolated at half the normal rate. To assure that the inputted parameters are interpolated only once during each parameter count during SLOW speaking operations each parameter count comprises three A cycles followed by one B cycle. It should be recalled that during the A cycle the interpolation is begun and during the B cycle the interpolated results are reinserted back into either K-stack 302, E10 loop 304 or pitch register 305, as appropriate. Thus, merely repeating the A cycle has no effect other than to recalculate the same value of a speech parameter but since it is only reinserted once back into either K-stack 302, E10 loop 304 or pitch register 305 only the results of the interpolation immediately before the B cycle are retained.

Inasmuch as parameter counter 513 includes an add one circuit, the results outputted therefrom, PC1-PC4, represent in binary form, the particular parameter count in which the synthesizer is operating. Output PC0 indicates in which cycle, A or B, the parameter count is. The parameter counter outputs PC1-PC4 are decoded by timing PLA 514. The particular decimal value of the parameter count is decoded by timing PLA 514 which is shown in adjacent to the timing PLA 514 with nomenclature such as PC=0, PC=1, PC=7 and so forth. The relationship between the particular parameters and the value of PC is set forth in FIG. 6. Output portions 511a and 511b of timing PLA 511 are also interconnected with outputs from timing PLA 514 whereby the Transfer K (TK) signal goes high during T9 of PC=2 or T8 of PC=3 or T7 of PC=4 and so forth through T1 of PC=10. Similarly, a LOAD Parameter (LDP) timing signal goes high during T5 of PC=0 or T1 of PC=1 or T3 of PC=2 and so forth through T7 of PC=11. As will be seen, signal TK is used in controlling the transfer of data from parameter output register 201 to subtractor 308, which transfer occurs at different T times accord-

ing to the particular parameter count the parameter counter 513 is in to assure that the appropriate parameter is being outputted from KE10 transfer register 303. Signal LDP is, as will be seen, used in combination with the parameter input register to control the number of bits which are inputted therein according to the number of bits associated with the parameter then being loaded according to the number of bits in each coded parameter as defined in FIG. 6.

Interpolation counter 515 includes a shift register and an add one circuit for binary counting the particular interpolation cycle in which the synthesizer 10 is operating. The relationship between the particular interpolation count in which the synthesizer is operating and the DIV1, DIV2, DIV4 and DIV8 timing signals derived therefrom is explained in detail with reference to FIG. 6 and therefore additional discussion here would be superfluous. It will be noted, however, that interpolation counter 515 includes a three bit latch 516 which is loaded at T1. The output of three bit latch 516 is decoded by gates 517 for producing the aforementioned DIV1 through DIV8 timing signals. Interpolation counter 515 is responsive to a signal RESETF from parameter counter 513 for permitting interpolation counter 515 to increment only after PC=12 has occurred.

ROM/Controller Interface Logic Diagram

Turning now to FIGS. 8a, 8b and 8c, which form a composite diagram, there is shown a detailed logic diagram of ROM/Controller interface logic 21. Parameter input register 205 is coupled, at its input to address pin ADD8. Register 205 is a six bit shift register, most of the stages of which are two bits long. The stages are two bits long in this embodiment inasmuch as ROMs 12a and b output, as will be seen, data at half the rate at which data is normally clocked in synthesizer 10. At the input of parameter input register 205 is a parameter input control gate 220 which is responsive to the state of a latch 221. Latch 221 is set in response to LDP, PC0 and DIV1 all being a logical one. It is reset at T14 and in response to parameter load enable from gate 238 being a logical zero. Thus, latch 221 permits gate 220 to load data only during the A portion (as controlled by PC0) of the appropriate parameter count and at an appropriate T time (as controlled by LDP) of IC0 (as controlled by DIV1) provided parameter load enable is at a logical one. Latch 221 is reset by T14 after the data has been inputted into parameter register 205.

The coded data in parameter input register 205 is applied on lines IN0-IN4 to coded parameter RAM 203, which is addressed by PC1-PC4 to indicate which coded parameter is then being stored. The contents of register 205 is tested by all one's gate 207, all zeroes gate 206 and repeat latch 208a. As can be seen, gate 206 tests for all zeroes in the four least significant bits of register 205 whereas gate 207 tests for all ones in those bits. Gate 207 is also responsive to PC0, DIV1, T16 and PC=0 so that the zero condition is only tested during the time that the coded energy parameter is being loaded into parameter RAM 203. The repeat bit occurs in this embodiment immediately in front of the coded pitch parameter; therefore, it is tested during the A cycle of PC=1. Pitch latch 208b is set in response to all zeroes in the coded pitch parameter and is therefore responsive to not only gate 206 but also the most significant bit of the pitch data on line 222 as well as PC=1. Pitch latch

208b is set whenever the loaded coded pitch parameter is a 00000 indicating that the speech is to be unvoiced.

Energy=0 latch **208c** is responsive to the output of gate **206** and PC=0 for testing whether all zeroes have been inputted as the coded energy parameter and is set in response thereto. Old pitch latch **208d** stores the output of the pitch=0 latch **208b** from the prior frame of speech data while old energy latch **208e** stores the output of energy=0 latch **208c** from the prior frame of speech data. The contents of old pitch latch **208d** and pitch=0 latch **208b** are compared in comparison gates **223** for the purpose of generating an INHIBIT signal. As will be seen, the INHIBIT signal inhibits interpolations and this is desirable during changes from voiced to unvoiced or unvoiced to voiced speech so that the new speech parameters are automatically inserted into K-stack **302**, E10 loop **304** and pitch register **305** as opposed to being more slowly interpolated into those memory elements. Also, the contents of old energy latch **208e** and energy=0 latch **208c** is tested by NAND gate **224** for inhibiting interpolation for a transition from a non-speaking frame to a speaking frame of data. The outputs of NAND gate **224** and gates **223** are coupled to a NAND gate **235** whose output is inverted to INHIBIT by an inverter **236**. Latches **208a**-**208c** are reset by gate **225** and latches **208d** and **208e** are reset by gate **226**. When the excitation signal is unvoiced, the K5-K10 coefficients are set to zero, as aforementioned. This is accomplished, in part, by the action of gate **237** which generates a ZPAR signal when pitch is equal to zero and when the parameter counter is greater than five, as indicated by PC 5 from PLA **514**.

Also shown in FIGS. 8a-c is a command latch **210** which comprises three latches **210a**, **b**, and **c** which latch in the data at CTL2, 4 and 8 in response to a processor data clock (PDC) signal in conjunction with a chip select (CS) signal. The contents of command latch **210** is decoded by command decoder **211** unless disabled by latches **218a** and **218b**. As previously mentioned, these latches are responsive to decoded LA, output and TTALK commands for disabling decoder **211** from decoding what ever data happens to be on the CTL2-CTL8 pins when subsequent PDC signals are received in conjunction with the LA, output and TTALK commands. A decoded TTALK command set **45** TTALK latch **219**. The output of TTALK latch **219**, which is reset by a Processor Data Clock Leading Edge (PDCLE) signal or by an output from latch **218b**, controls along with the output of latch **218a** NOR gates **227a** and **b**. The output of NOR gate **227a** is a logical one if TTALK latch **219** is set, thereby coupling pins CTL1 to the talk latch via tristate buffer **228** and inverters **229**. Tristate latch **228** is shown in detail on the right side of FIGS. 8a-c. NOR gate **227b**, on the other hand, outputs a logical one if an output code has been detected, setting latch **228a** and thereby connecting pins CTL1 to the most significant bit of data input register **212**.

Data is shifted into data input register **212** from address pin **8** in response to a decoded read command by logics **230**. RE, RB and LA instructions are outputted to ROM via instruction pins **I₀-I₁** from ROM control logic **217** via buffers **214c**. The contents of data input register **212** is outputted to CTL1-CTL4 pins via buffers **213** and to the aforementioned CTL1 pin via buffer **228** when NOR gate **227b** inputs a logical one. CTL1-CTL4 pins are connected to address pins ADD-1-ADD4 via buffers **214a** and CTL8 pin is connected to

ADD8 pin **8** via a control buffer **214b** which is disabled when addresses are being loaded on the ADD1-ADD8 pins by the signal on line **231**.

The Talk latch **216** shown in FIGS. 8a-c preferably comprises, three latches **216a**, **216b** and **216c**. Latch **216a** is set in response to a decoded SPK command and generates, in response thereto, a speak enable (SPEN) signal. As will be seen, SPEN is also generated in response to a decoded SPKSLOW command by latch **215a**. Latch **216b** is set in response to speak enable during IC7 as controlled by gate **225**. Latches **216a** and **216b** are reset in response to (1) a decoded reset command, (2) an energy equals fifteen code or (3) on a power-up clear by gate **232**. Talk delayed latch **216c** is set with the contents of latch **216b** at the following IC7 and retains that data through eight interpolation counts. As was previously mentioned, the talk delayed latch permits the speech synthesizer to continue producing speech data for eight interpolation cycles after a coded energy=0 condition has been detected setting latch **208c**. Likewise, slow talk latch **215** is implemented with latches **215a**, **215b** and **215c**. Latch **215a** enables the speak enable signal while latches **215b** and **215c** enable the production of the SLOWD signal in much the same manner as latches **216b** **216d** **216c** enable the production of the TALKD signal.

Considering now, briefly, the timing interactions for inputting data into parameter input register **205**, it will be recalled that this is controlled chiefly by a control gate **220** in response to the state of a parameter input latch **221**. Of course, the state of the latch is controlled by the LDP signal applied to gate **233**. The PC0 and DIV1 signals applied to gate **233** to assure that the parameters are loaded during the A cycle of a particular parameter count during IC0. The particular parameter and the parameter T-Time within the parameter count is controlled by LDP according to the portion **511a** of timing PLA **511** (FIGS. 7a and 7b). The first parameter inputted (Energy) is four bits long and therefore LDP is initiated during time period **T5** (as can be seen in FIGS. 7a and 7b). During parameter count 1, the repeat bit and pitch bits are inputted, this being six bits which are inputted according to LDP which comes up at time period **T1**. Of course, there four times periods difference between **T1** and **T5** but only two bits difference in the length of the inputted information. This occurs because it takes two time periods to input each bit into parameter input register **205** (which has two stages per each inputted bit) due to the fact that ROMs **12a-12b** are preferably clocked at half the rate at that which synthesizer **10** is clocked. By clocking the ROM chips at half the rate, that the synthesizer **10** chip is clocked simplifies the addressing of the read-only-memories in the aforesaid ROM chips and yet, as can be seen, data is supplied to the synthesizer **10** in plenty of time for performing numerical operations thereon. Thus, in section **511a** of timing PLA **511**, LDP comes up at **T1** when the corresponding parameter count indicates that a six bit parameter is to be inputted, comes up at **T3** when the corresponding parameter count indicates that a five bit parameter is to be inputted, comes up at **T5** when the corresponding parameter count indicates that a four bit parameter is to be inputted and comes up at time period **T7b** when the corresponding parameter count (EG parameter counts **9**, **10**, and **11**) which correspond to a three bit coded parameter. ROMs **12a-b** are signaled that the addressed parameter ROM is to output information when signaled via **I₀** instruction pin, ROM con-

trol logic 217 and line 234 which provides information to ROM control logic 217 from latch 221.

Parameter Interpolator Logic Diagram

Referring now to FIGS. 9a and 9b, which form a composite diagram the parameter interpolator logic 23 is shown in detail. K-stack 203 comprises ten registers each of which store ten bits of information. Each small square represents one bit of storage, according to the convention depicted at numeral 330. The contents of each shift register is arranged to recirculate via recirculation gates 314 under control of a recirculation control gate 315. K-stack 302 stores speech coefficients K1-K9 and temporarily stores coefficient K10 or the energy parameter generally in accordance with the speech synthesis apparatus of FIG. 7 of U.S. patent application Ser. No. 807,461, since abandoned and continued in U.S. patent application Ser. No. 905,328, filed May 12, 1978. The data outputted from K-stack 302 to recoding logic 30 at various time periods is shown in Table VII. In Table III of U.S. patent application Ser. No. 807,461 since abandoned and continued in U.S. patent application Ser. No. 905,328, filed May 12, 1978, is shown the data outputted from the K-stack of FIG. 7 therof. Table VII of this patent differs from Table III of the aforementioned patent because of (1) recoding logic 301 receives the same coefficient on lines 32-1 through 32-4, on lines 32-5 and 32-6, on lines 32-7 and 32-8 and on lines 32-9 and 32-10 because, as will be seen, recoding logic 301 responds to two bits of information for each bit which was responded to by the array multiplier of the aforementioned U.S. Pat.; (2) because of the difference in time period nomenclature as was previously explained with reference to FIG. 5; and (3) because of the time delay associated with the recoding logic 301.

Recoding logic 301 couples K-stack 302 to array multiplier 401 (FIGS. 10a and 10b). Recoding logic 301 includes four identical recoding stages 312a-312d, only one of which, 312a, is shown in detail. The first stage of the recoding logic, 313, differs from stages 312a-312d basically because there is, of course, no carry, such as occurs on input A in stages 312a-312d, from a lower order stage. Recoding logic outputs ± 2 , ± 1 and ∓ 1 to each stage of a five stage array multiplier 401, except for stage zero which receives only ± 2 , ± 1 and ∓ 1 outputs. Effectively recoding logic 301 permits array multiplier to process, in each stage thereof, two bits in lieu of one bit of information, using Booth's algorithm. Booth's algorithm is explained in "Theory and Application of Digital Signal Processing", published by Prentice-Hall 1975, at pp. 517-18.

The K10 coefficient and energy are stored in E10 loop 304. E10 loop preferably comprises a twenty stage serial shift register; ten stages 304a of E10 loop 304 are preferably coupled in series and another ten stages 304b which are also coupled in series but also have parallel outputs and inputs to K-stack 302. The appropriate parameter, either energy or the K10 coefficient, is transferred from E10 loop 304 to K-stack 302 via gates 315 which are responsive to a NOR gate 316 for transferring the energy parameter from E10 loop 304 to K-stack 302 at time period T10 and transferring coefficient K10 from E10 loop 304 to K-stack 302 at time period T20. NOR gate 306 also controls recirculation control gate 315 for inhibiting recirculation in K-stack 302 when data is being transferred.

KE10 transfer register 303 facilitates the transferring of energy or the K1-K10 speech coefficients which are

stored in E10 loop 304 or K-stack 302 to adder 308 and delay circuit 309 via selector 307. Register 303 has nine stages provided by paired inverters and a tenth stage being effectively provided by selector 307 and gate 317 for facilitating the transfer of ten bits of information either from E10 loop 304 or K-stack 302. Data is transferred from K-stack 302 to register 303 via transfer gates 318 which are controlled by a Transfer K (TK) signal generated by decoder portion 511b of timing PLA 511 (FIGS. 7a and 7b). Since the particular parameter to be interpolated and thus shifted into register 303 depends upon the particular parameter count in which the synthesizer is operating and since the particular parameter available to be outputted from K-stack 302 is a function of particular time period the synthesizer is operating in, the TK signal comes up at T9 for the pitch parameter, T8 for the K1 parameter, T7 for the K2 parameter and so forth, as is shown in FIGS. 7a and 7b. The energy parameter or the K10 coefficient is clocked out of E10 loop 304 into register 303 via gates 319 in response to a TE10 signal generated by a timing PLA 511. After each interpolation, that is during the B cycle, data is transferred from register 303 into (1) K-stack 302 via gates 318 under control of signal TK, at which time recirculation gates 314 are turned off by gate 315, or (2) E10 loop 304 via gates 319.

A ten bit pitch parameter is stored in a pitch register 305 which includes a nine stage shift register as well as recirculation elements 305a which provide another bit of storage. The pitch parameter normally recirculates in register 305 via gate 305a except when a newly interpolated pitch parameter is being provided on line 320, as controlled by pitch interpolation control logics 306. The output of pitch 305 (PTO) or the output from register 303 is applied by selector 307 to gate 317. Selector 307 is also controlled by logics 306 for normally coupling the output of register 303 to gate 317 except when the pitch is to be interpolated. Logics 306 are responsive for outputting pitch to adder 308 and delay 309 during the A cycle of PC=1 and for returning the interpolated pitch value on line 320 on the B cycle of PC=1 to register 305. Gate 317 is responsive to a latch 321 for only providing pitch, energy or coefficient information to adder 308 and delay circuit 309 during the interpolation. Since the data is serially clocked, the information may be started to be clocked during an A portion and PC0 may switch to a logical one sometime during the transferring of the information from register 303 or 305 to adder 308 or delay circuit 309, and therefore, gate 317 is controlled by an A cycle latch 321, which latch is set with PC0 at the time a transfer coefficient (TK) transfer E10 (TE10) or transfer pitch (TP) signal is generated by timing PLA 511.

The output of gate 317 is applied to adder 308 and delay circuit 309. The delay in delay circuit 309 depends on the state of DIV1-DIV8 signals generated by interpolation counter 515 (FIGS. 7a and 7b). Since the data exits gate 317 least significant bit first, by delaying the data in delay circuit 309 a selective amount, and applying the output to adder 310 along with the output of subtractor 308, the more delay there is in circuit 309, the smaller the effective magnitude of the difference from subtractor 308 which is subsequently added back in by adder 310. Delay circuit 311 couples adder 310 back into register 303 and 305. Both delay circuits 309 and 303 can insert up to three bits of delay and when adder 309 is at its maximum delay 311 is at its minimum delay and visa-versa. A NAND gate 322 couples the output of

subtractor 308 to the input of adder 310. Gate 322 is responsive to the output of an OR gate 323 which is in turn responsive to INHIBIT from inverted 236 (FIGS. 8a-c). Gates 322 and 323 act to zero the output from subtractor 308 when the INHIBIT signal comes up unless the interpolation counter is at IC0 in which case the present values in K-stack 302, E10 loop 304 and P register 305 are fully interpolated to their new target values in a one step interpolation. When an unvoiced frame (FIG. 6) is supplied to the speech synthesis chip, coefficients K5-K10 are set to zero by the action of gate 324 which couples delay circuit 311 to shift register 325 whose output is then coupled to gates 305a and 303'. Gate 324 is responsive to the zero parameter (ZPAR) signal generated by gate 237 (FIGS. 8a-c).

Gate 326 disables shifting in the 304b portion of E10 loop 304 when a newly interpolated value of energy or K10 is being inputted into portion 304b from register 303. Gate 327 controls the transfer gates coupling the stages of register 303, which stages are inhibited from serially shifting data therebetween when TK or TE10 goes high during the A cycle, that is, when register 303 is to be receiving data from either K-stack 302 or E10 loop 304 as controlled by transfer gates 318 or 319, respectively. The output of gates 327 is also connected to various stages of shift register 325 and to a gate coupling 303' with register 303. Whereby up top the three bits which may trail the ten most significant bits after an interpolation operation may be zeroed.

Array Multiplier Logic Diagram

FIGS. 10a and 10b form a composite logic diagram of array multiplier 401. Array multipliers are sometimes referred to as Pipeline Multipliers. For example, see "Pipeline Multiplier" by Granville E. Ott, published by the University of Missouri.

Array multiplier 401 has five stages, stage 0 through stage 4, and a delay stage. The delay stage is used in array multiplier 41 to give it the same equivalent delay as the array multiplier shown in U.S. patent application Ser. No. 807,461, since abandoned and continued in J.S. patent application Ser. No. 905,328, filed May 12, 1978. The input to array multiplier 401 is provided by signals MR₀-MR₁₃, from multiplier multiplexer 405. MR₁₃ is the most significant bit while MR₀ is the least significant bit. Another input to array multiplier are the aforementioned +2, -2, +1 and -1 outputs from recoding logic 301 (FIGS. 8a-c). The output from array multiplier 401, P₁₃-P₀, is applied to summer multiplexer 402. The least significant bit thereof, P₀, is in this embodiment always made a logical one because doing so establishes the mean of the truncation error as zero instead of $-\frac{1}{2}$ LSB which value would result from a simple truncation of a two's complement number.

Array multiplier 401 is shown by a plurality of box elements labeled A-1, A-2, B-1, B-2, B-3 or B-C. The specific logic elements making up these box elements are shown on the right-hand side of composite FIGS. 10a-10b in lieu of repetitively showing these elements and making up a logic diagram of FIG. 401, for simplicity sake. The A-1 and A-2 block elements make up stage zero of the array multiplier and thus are each responsive to the -2, +1 and -1 signals outputted from decoder 13 and are further responsive to MR₂-MR₁₃. When multiplies occur in array multiplier 401, the most significant bit is always maintained in the left most column elements while the partial sums are continuously shifted toward the right. Inasmuch as each stage of array multi-

plier 401 operates on two binary bits, the partial sums, labeled Σ_n , are shifted to the right two places. Thus no A type blocks are provided for the MR₀ and MR₁ data inputs to the first stage. Also, since each block in array 5 multiplier 401 is responsive to two bits of information from K-stack 302 received via recoding logic 301, each block is also responsive to two bits from multiplier multiplexer 405, which bits are inverted by inverters 430, which bits are also supplied in true logic to the B 10 type blocks.

Filter and Excitation Generator Logic Diagram

FIGS. 11a-11b form a composite, detailed logic diagram of lattice filter and excitation generator 24 (other than array multiplier 401) and output section 25. In filter and excitation generator 24 is a summer 404 which is connected to receive at one input thereof either the true or inverted output of array multiplier 401 (see FIGS. 10a and 10b) on lines P₀-P₁₃ via summer multiplexer 402. The other input of adder 404 is connected via summer multiplexer 402 to receive either the output of adder 404 (at T10-T18), the output of delay stack 406 on lines 440-453 at T20-T7 and T9, the output of Y-latch 403 (at T8) or a logical zero from ϕ_3 precharge gate 420 (at T19 when no conditional discharge is applied to this input). The reasons these signals are applied at these times can be seen from FIG. 8 of the aforementioned U.S. patent application Ser. No. 807,461, since abandoned and continued in U.S. patent application Ser. No. 905,328, filed May 12, 1978; it is to be remembered of course, that the time period designations differ as discussed with reference to FIG. 5 hereof.

The output of adder 404 is applied to delay stack 406, multiplier multiplexer 405, one period delay gates 414 and summer multiplexer 402. Multiplier multiplexer 405 includes a one period delay gates 414 which are generally equivalent to one period delay 34' of FIG. 7 in U.S. patent application Ser. No. 807,461, since abandoned and continued in U.S. patent application Ser. No. 905,328, filed May 12, 1978. Y-latch 403 is connected to receive the output of delay stack 406. Multiplier multiplexer 405 selectively applies the output from Y-latch 403, one period delay gates 414, or the excitation signal on bus 415 to the input MR₀-MR₁₃ of array multiplier 401. The inputs D₀-D₁₃ to delay stack 406 are derived from the outputs of adders 404. The logics for summer multiplexer 402, adder 404, Y-latch 403, multiplier multiplexer 405 and one period delay circuit 414 are only shown in detail for the least significant bit as enclosed by dotted line reference A. The thirteen most significant bits in the lattice filter also are provided by logics such as those enclosed by the reference A line, which logics are denoted by long rectangular phantom line boxes labeled "A". The logics for each parallel bit being processed in the lattice filter are not shown in detail for sake of clarity. The portions of the lattice filter handling bits more significant than the least significant bit differ from the logic shown for elements 402, 403, 404, 405, and 414 only with respect to the interconnections made with truncation logics 501 and bus 415 which connects to UV gate 408 and chirp ROM 409. In this respect, the output from UV gate 408 and chirp ROM 409 is only applied to inputs I₁₃-I₆ and therefore the input labeled I_x within the reference A phantom line is not needed for the six least significant bits in the lattice filter. Similarly, the output from the Y-latch 403 is only applied for the ten most significant bits, YL₁₃ through YL₄, and therefore the connection labeled YL_x within the reference

line is not required for the four least significant bits in the lattice filter.

Delay stack 406 comprises 14 nine bit long shift registers, each stage of which comprise inverters clocked on ϕ_4 and ϕ_3 clocks. As is discussed in U.S. patent application Ser. No. 807,461, since abandoned and continued in U.S. patent application Ser. No. 905,328, filed May 12, 1978, the delay stack 406 which generally corresponds to shift register 35' of FIG. 7 of the aforementioned patent, is only shifted on certain time periods. This is accomplished by logics 416 whereby ϕ_B - ϕ_{4B} clocks are generated from T10-T18 timing signal from PLA 512 (FIGS. 7a and 7b). The clock buffers 417 in circuit 416 are also shown in detail in FIGS. 11a and 11b.

Delay stack 406 is nine bits long whereas shift register 35' in FIG. 7 of U.S. patent application Ser. No. 807,461, since abandoned and continued in U.S. patent application Ser. No. 905,328, filed May 12, 1978, was eight bits long; this difference occurs because the input to delay stack 406 is shown as being connected from the output of adder 404 as opposed to the output of one period delay circuit 414. Of course, the input to delay stack 406 could be connected from the outputs of one period delay circuit 414 and the timing associated therewith modified to correspond with that shown in U.S. patent application Ser. No. 807,461, since abandoned and continued in U.S. patent application Ser. No. 905,328, filed May 12, 1978.

The data handled in delay stack 406, array multiplier 401, adder 402, summer multiplexer 402, Y-latch 403, and multiplier multiplexer 405 is preferably handled in two's complement notation.

Unvoiced generator 407 is a random noise generator comprising a shift register 418 with a feedback term supplied by feedback logics 419 for generating pseudo-random terms in shift register 418. An output is taken therefrom and is applied to UV gate 408 which is also responsive to OLDP from latch 208d (FIGS. 8a and 8b). Old pitch latch 208d controls gate 408 because pitch=0 latch 208b changes state immediately when the new speech parameters are inputted to register 205. However, since this occurs during interpolation count IC0 and since, during an unvoiced condition the new values are not interpolated into K-stack 302, E10 loop 304 and pitch register 305 until the following IC0, the speech excitation value cannot change from a periodic excitation from chirp ROM 409 to a random excitation from unvoiced generator 407 until eight interpolation cycles have occurred. Gate 420 nols the output of gate 408 into the most significant bit of the excitation signal, I₁₃, thereby effectively causing the sign bit to randomly change during unvoiced speech. Gate 421 effectively forces the most significant bit of the excitation signal, I₁₂, to a logical one during unvoiced speech conditions. Thus the combined effect of gates 408, 420 and 421 is to cause a randomly changing sign to be associated with a steady decimal equivalent value of 0.5 to be applied to the lattice filter and Filtering Excitation Generator 24.

During voiced speech, chirp ROM 409 provides an eight bit output on lines I₆-I₁₃ to the lattice filter. This output comprises forty-one successively changing values which, when graphed, represent a chirp function. The contents of ROM 409 are listed in Table VIII; ROM 404 is set up to invert its outputs and thus the data is stored therein in complemented format. The chirp function value and the complemented value stored in the chirp ROM are expressed in two's complement hexadecimal notation. ROM 409 is addressed by an

eight bit register 410 whose contents are normally updated during each cycle through the lattice filter by add one circuit 411. The output of register 410 is compared with the contents of pitch register 305 in a magnitude comparator 403 for zeroing the contents of 410 when the contents of register 410 become equal to or greater than the contents of register 305. ROM 409, which is shown in greater detail in FIGS. 14a-14b, is arranged so that addresses greater than 110010 cause all zeroes to be outputted on lines I₁₃-I₆ to multiplier multiplexer 405. Zeros are also stored in address locations 41-51. Thus, the chirp may be expanded to occupy up to address location fifty, if desired.

Random Access Memory Logic Diagram

Referring now to FIGS. 12a-12b, there is shown a composite detailed logic diagram of RAM 203. RAM 203 is addressed by address on PC1-PC4, which address is decoded in a PLA 203a and defines which coded parameter is to be inputted into RAM 203. RAM 203 stores the twelve decoded parameters, the parameters having bit lengths varying between three bits and five bits according to the decoding scheme described with reference to FIG. 6. Each cell, reference B, of RAM 203 is shown in greater detail in FIG. 12b. Read-/Write control logic 203b is responsive to T1, DIV1, PC0 and parameter load enable for writing into the RAM 203 during the A cycle of each parameter count during interpolation count zero when enabled by parameter load enable from logics 238 (FIGS. 8a-c). Data is inputted to RAM 203 on lines IN0-IN4 from register 205 as shown in FIGS. 8a and 8b and data is outputted on lines OUT1-OUT5 to ROM 202 as is shown in the aforementioned Figures

Parameter Read-Only-Memory Logic Diagram

In FIGS. 13a-13b, there is shown a logic diagram of ROM 202. ROM 202 is preferably a virtual ground ROM of the type disclosed in U.S. Pat. No. 3,934,233. Address information from RAM 202 and from parameter counter 513 are applied to address buffers 202b which are shown in detail at reference A. The NOR gates 202a used in address buffers 202b are shown in detail at reference B. The outputs of the address buffers 202b are applied to an X-decoder 202c or to a Y-decoder 202d. The ROM is divided into ten sections labeled reference C, one of which is shown in greater detail. The outline for output line from each of the sections is applied to register 201 via inverters as shown in FIGS. 8a and 8b. X-decoder selects one of fifty-four X-decode lines while Y-decoder 202d tests for the presence or nonpresence of a transistor cell between an adjacent pair of diffusion lines, as is explained in greater detail in the aforementioned U.S. Pat. No. 3,934,233. The data preferably stored in ROM 202 of this embodiment is listed in TABLE VI.

Chirp-Read-Only-Memory Logic Diagram

FIGS. 14a-14b form a composite diagram of chirp ROM 409. ROM 409 is addressed via address lines A₀-A₈ from register 410 (FIGS. 11a-11b) and output information on lines I₆-I₁₁ to multiplier multiplexer 405 and lines I_{m1} and I_{m2} to gates 421 and 420, all which are shown in FIGS. 11a and 11b. As was previously discussed with reference to FIGS. 11a and 11b, chirp ROM outputs all zeros after a predetermined count is reached in register 410, which, in this case is the count equivalent to a decimal 51. ROM 409 includes a Y-

decoder 409a which is responsive to the address on lines \overline{A}_0 and \overline{A}_1 (and A_0 and A_1) in an X-decoder 409b which is responsive to the address on lines \overline{A}_2 through \overline{A}_5 (and A_2 - A_5).

ROM 409 also includes a latch 409c which is set when decimal 51 is detected on lines \overline{A}_0 - \overline{A}_5 according to line 409c from a decoder 409e. Decoder 409e also decodes a logical zero on lines A_0 - A_8 presetting latch 409c. ROM 409 includes timing logics 409f which permit data to be clocked in via gates 409g at time period T12. At this time decoder 409e checks to determine whether either a decimal 0 or decimal 51 is occ on address line \overline{A}_0 - \overline{A}_8 . If either condition occur, latch 409c, which is a static latch, is caused to flip.

An address latch 409h is set at time period T13 and reset at time period T11. Latch 409h permits latch 409c to force a decimal 51 onto lines \overline{A}_0 - \overline{A}_5 when latch 409c is set. Thus, for addresses greater than 51 address register 410, the address is first sampled at time period T12 to determine whether it has been reset to zero by reset logic 412 (FIGS. 12a-12b) for the purpose of resetting latch 409c and if the address has not been reset to zero then whatever address has been inputted on lines \overline{A}_0 - \overline{A}_8 is written over by logics 409j at T13. Of course, at location 51 in ROM 409 will be stored all zeros on the output lines I6-I11, IM1 and IM2. Thus by the means of logics 409c, 409h and 409j addresses of a preselected value, in this case a decimal 51, are merely tested to determine whether a reset has occurred but are not permitted to address the array of ROM cells via decoders 409a and 409b. Addresses between a decimal 0 and 50 address the ROM normally via decoders 409a and 409b. The ROM matrix is preferably of the virtual ground type described in U.S. Pat. No. 3,934,233. As aforementioned, the contents of ROM 409 are listed in Table VIII. The chirp function is located at addresses 00-40 while zeros are located at addresses 41-51.

Truncation Logic and Digital-To-Analog Converter

Turning again to FIGS. 11a and 11b, the truncation logic 425 and Digital-to-analog (D/A) converter is shown in detail. Truncation logic 425 includes circuitry for converting the two's complement data on YL_{1-3} - YL_{14} to sign magnitude data. Logics 425a test the MSB from Y-latch 403 on line YL_{13} for the purpose of generating a sign bit and for controlling the two's complement to sign magnitude conversion accomplished by logics 425c. The sign bit is supplied in true and false logic on lines D/Asn and D/Asn to D/A converter 426.

Logics 425c convert the two's complement data from Y-latches 403 in lines YL_{10} - YL_4 to simple magnitude notation on lines D/A₆-D/A₀. Only the logics 425c associated with YL_{10} are shown in detail for sake of simplicity.

Logics 425b sample the YL_{12} and YL_{11} bits from the Y-latches 403 and perform a magnitude truncation function thereon by forcing outputs D/A₆ through D/A₀ to a logical zero (i.e., a value of one if the outputs were in true logic) wherever either YL_{12} or YL_{11} is a logical one and YL_{13} is a logical zero, indicating that the value is positive or either YL_{12} or YL_{11} is a logical zero and YL_{13} is a logical one, indicating that the value is negative (and complemented, of course). Whenever one of these conditions occurs, a logical zero appears on line 427 and V_{ss} is thereby coupled to the output buffer 428 in each of logics 425c. The magnitude function effectively truncates the more significant bits on YL_{11} and YL_{12} . It is realized that this is somewhat unorthodox

truncation, since normally the less significant bits are truncated in most other circuits where truncation occurs. However, in this circuit, large positive or negative values are effectively clipped. More important digital speech information, which has smaller magnitudes, is effectively amplified by a factor of four by this truncation scheme.

The outputs D/A₆-D/A₀, along with D/Asn and D/Asn, are coupled to D/A converter 426. D/A converter 426 preferably has seven MOS devices 429 coupled to the seven lines D/A₆ through D/A₀ from truncation logics 425. Each device 429 preferably includes a MOS transistor whose gates are coupled to one of the lines D/A₆-D/A₀ and a series connected implanted load transistor 429b. Devices 429 are arranged, by controlling their length to width ratios, to act as current sources, the device 429 coupled to D/A₆ sourcing twice as much current (when on) as the device 429 coupled to D/A₅. Likewise the device 429 coupled to D/A₅ is capable of sourcing twice as much current as the device 429 coupled to D/A₄. This two to one current sourcing capability similarly applies to the remaining devices 429 coupled to the remaining lines D/A₃-D/A₀. Thus, device 429 coupled to D/A₁, is likewise capable of sourcing twice as much current as the device 429 coupled to D/A₀, but only one-half of that source by the device 429 coupled to D/A₂. All devices 429 are connected in parallel, one side of which are preferably coupled to V_{ss} and the other side is preferably coupled to either side of the speaker 4 via transistors 430 and 431. Transistor 430 is controlled by D/Asn which is applied to its gates; transistor 431 is turned off and on in response to D/Asn. Thus, either transistor 430 and 431 is on depending on the state of the sign bit, D/Asn. The voice coil of speaker 4 preferably has a 100 ohm impedance and has a center tap connected to V_{gg}, as shown in FIG. 23a. Thus, the signals on D/A₆-D/A₀ control the magnitude of current flow through the voice coil while the signals on lines D/Asn and D/Asn control the direction of that flow.

Alternatively to using a center-tapped 100 ohm voice coil, a more conventional eight ohm speaker may be used along with a transformer having a 100 ohm center tapped primary (connected to V_{gg} and transistors 430 and 431) and an eight ohm secondary (connected to the speaker's terminals, as shown in FIG. 23b).

It should now be appreciated by those skilled in the art that D/A converter 426 not only converts digital sign magnitude information on lines D/A₆-D/A₀ and D/Asn-D/Asn to an analog signal, but hastively amplified this analog signal to sufficient levels to permit a speaker to be driven directly from the MOS synthesis chip 10 (or via the aforementioned transformer, if desired). Of course, those skilled in the art will appreciate that simple D/A converters, such as that disclosed here, will find use in other applications in addition to speech synthesis circuits.

THE SPEECH SYNTHESIZER CHIP

In FIG. 22 a greatly enlarged plan view of a semiconductor chip which contains the entire system of FIGS. 4a and 4b is illustrated. The chip is only about two hundred fifteen mils (about 0.215 inches) on a side. In the example shown, the chip is manufactured by the P-channel metal gate process using the following design rules: metal line width 0.25 mil; metal line spacing 0.25 mil; diffusion line width 0.15 mil; and diffusion line spacing 0.30 mil. Of course, as design rules are tightened

with the advent of electron beam mask production or slice writing, and other techniques, it will be possible to further reduce the size of the synthesizer chip. The size of the synthesizer chip can, of course also be reduced by not taking advantage of some of the features preferably used on the synthesizer chip.

The total active area of speech synthesizer chip 10 is approximately 45,000 square mils.

It will also be appreciated by those skilled in the art, that other MOS manufacturing techniques, such as N-channel, complementary MOS (CMOS) or silicon gate processes may alternatively be used.

The various parts of the system are labeled with the same reference numerals previously used in this description.

CONTROLLER LOGIC DIAGRAMS

The controller used in the learning aid is preferably a microprocessor of the type described in U.S. Pat. No. 4,074,355, with modifications which are subsequently described. U.S. Pat. No. 4,074,355 is hereby incorporated herein by reference. It is to be understood, of course, that other microprocessors, as well as future microprocessors, may well find use in applications such as the speaking learning aid described herein.

The microprocessor of U.S. Pat. No. 4,074,355 is an improved version of an earlier microprocessor described in U.S. Pat. No. 3,991,305. One of the improvements concerned the elimination of digit driver devices so that arrays of light emitting diodes (LED's) forming a display could be driven directly from the microprocessor. As a matter of design choice, the display used with this learning aid is preferably a vacuum fluorescent (VF) display device. Those skilled in the art will appreciate that when LED's are directly driven, the display segments are preferably sequentially actuated while the display's common character position electrodes are selectively actuated according to information in a display register or memory. When VF displays are utilized, on the other hand, the common character position electrodes are preferably sequentially actuated while the segments are selectively actuated according to information in the display register or memory. Thus, the microprocessor of U.S. Pat. No. 4,074,355 is preferably altered to utilize digit scan similar to that used in U.S. Pat. No. 3,991,305.

The microprocessor of U.S. Pat. No. 4,074,355 is a four bit processor and to process alphanumeric information, additional bits are required. By using six bits, which can represent 2^6 or 64 unique codes, the twenty-six characters of the alphabet, ten numerals as well as several special characters can be handled with ease. In lieu of converting the microprocessor of U.S. Pat. No. 4,074,355 directly to a six bit processor, it was accomplished indirectly by software pairing the four bit words into eight bit bytes and transmitting six of those bits to the display decoder.

Referring now to FIGS. 15a-15b, which form a composite block diagram of the microprocessor preferably used in the learning aid, it should be appreciated that this block diagram generally corresponds with the block diagram of FIGS. 7a and 7b of U.S. Pat. No. 4,074,355; several modifications to provide the aforementioned features of six bit operation and VF display compatibility are also shown. The numbering shown in FIGS. 15a and 15b generally agrees with that of U.S. Pat. No. 4,074,355. The modifications will now be described in detail.

Referring now to the composite diagram formed by FIGS. 16a-16b, which replace FIG. 13 of U.S. Pat. No. 4,074,355, there can be seen the segment decoder and RAM address decoder 33-1 which decodes RAMY for addressing RAM 31 or ACC1-ACC8 for decoding segment information. Decoder 33-1 generally corresponds to decoder 33 in the aforementioned U.S. patent. The segment information is re-encoded into particular segment line information in output section 32-2 and outputted on bus 90 to segment drivers 91. Six bits of data from the processor's four bit accumulator 77 are decoded in decoder 33-1 as is now described. First, four bits on bus 86 are latched into accumulator latches 87-1 through 87-8 on a TDO (Transfer Data Out) instruction when status is a logical one. Then, two bits on bus 86 (from lines 86-1 and 86-2) are latched into accumulator latches 87-16 and 86-32, respectively, on another TDO instruction when status is a logical zero. Then the six bits in latches 87-1 through 87-32 is decoded in decoder 33-1. Segment drivers 91 may preferably be of one of three types, 91A, 91B or 91C as shown on FIGS. 16a-16b. The 91A type drivers permits the data on ACC1-ACC8 to be communicated externally via pins SEG G, SEG B, SEG C and SEG D. The 91B type driver coupled to pin SEG E permits the contents of digit register 94-10 to be communicated externally when digit register 94-12 is set. The 91B type driver coupled to pin SEG A permits the contents of the program counter to be outputted during test operations.

The digit buffers registers and TDO latches of FIG. 14 of U.S. Pat. No. 4,074,355 are also preferably replaced with the digit buffers registers of FIG. 17 herein inasmuch as (1) the DDIG signal is no longer used and (2) the digit latches (elements 97 in U.S. Pat. No. 4,074,355) are no longer used. For simplicity's sake, only one of the digit output buffer registers 94 is shown in detail. Further, since in this embodiment of the learning aid, display 2 preferably has eight character positions, eight output buffers 98-0 through 98-7 connect D₀D₇ to the common electrodes of display 2 via registers 94-0 through 94-7 are shown in FIG. 17. An additional output buffer 98-8 communicates the contents of registers 94-12, which is the chip select signal, to synthesizer 10.

To facilitate bi-directional communication with synthesizer 10, the microprocessor of U.S. Pat. No. 4,074,355 is preferably modified to permit bi-directional communication on pins SEG G, SEG B, SEG C and SEG D. Thus, in FIG. 18, these SEG pins are coupled to the normal K lines, 112-1 through 112-8, via an input selector 111a for inputting information when digit registers 94-12 (R12) is set. Further, these pins are also coupled to ACC1-ACC8 via segment drivers 91A when digit registers 94-12 (R12) and 94-11 (R11) are set for outputting information in accumulator 77.

Thus, when digit latch 94-12 (which communicates the chip select signal externally) is set, SEG E is coupled to R10 (digit registers 94-10) for communicating the PDC signal to synthesizer 10. Also, ACC1-ACC8 is outputted on SEG G and SEG B-SEG D, during the time R12 is set and R11 are set. When R11 is a logical 0, i.e., is reset, segment drivers 91A are turned off and data may be read into CKB circuit 113 for receiving data from ROMS 12a-12b via synthesizer 10, for instance, FIG. 18 replaces the keyboard circuit 111 shown in FIG. 22 of U.S. Pat. No. 4,064,554.

Preferably, pins SEG G and SEG B-SEG D are coupled to CTL1-CTL8 pins of synthesizer 10, while pin SEG E is coupled to the PDC pin of synthesizer 10.

In Table IX (which comprises Tables 0 through IX-15) is listed the set of instructions which may be stored in the main Read-Only Memory 30 of FIGS. 15a-15b to provide controller 11. Referring now to Table IX, there are several columns of data which are, reading from left to right: PC (Program Counter), INST (Instruction), BRLN (Branch Line), Line and Source Statement (which includes Name, Title and Comments). In U.S. Pat. No. 4,074,355, it can be seen that main Read-Only-Memory 30 is addressed with a seven bit address in program counter 47 and a four bit address in a buffer 60. The address in buffer 60 is referred to as a page address in the main Read-Only-Memory. The instructions listed on Table IX-0 correspond to page zero in the microprocessor while the instructions listed in Table IX-1 are those on page one and so forth through to the instructions in Table IX-15 which are stored on page fifteen in the microprocessor.

The program counter 47 of the aforementioned microprocessor is comprised of a feedback shift register and therefore counts in a pseudorandom fashion, thus the addresses in the left-hand column of Table IX, 25 which are expressed as a hexadecimal number, exhibit such pseudorandomness. If the instruction starting at page zero were read out sequentially from the starting position in the program counter (00) then the instructions would be read out in the order shown in Table IX. In the "Line" column is listed a sequentially increasing decimal number associated with each source statement and its instruction and program counter address as well as those lines in which only comments appear. The line number starts at line 55 merely for reasons of convenience not important here. When an instruction requiring either a branch or call is to be performed, the address to which the program counter will jump and the page number to which the buffer will jump, if required, is reflected by the binary code comprising the instruction or instructions performing the branch or call. For sake of convenience, however, the branch line column indicates the line number in Table IX to which the branch or call will be made. For example, the instruction on line 59 (page 0, Program Counter Address OF) 45 is a branch instruction, with a branch address of 1010111 (57 in hexadecimal). To facilitate finding the 57 address in the program counter, the branch line column directs the reader to line 80, where the 57 address is located.

READ-ONLY MEMORY LOGIC DIAGRAMS

Read-Only-Memories 12a or 12b or 13a or 13b are shown in FIGS. 19, 20a, 20b, 21a and 21b. FIG. 19 is a block diagram of any one of these ROMs. FIGS. 20a and 20b form a composite logic diagram of the control logic for the ROMs while FIGS. 20a and 20b form a composite logic diagram of the X and Y address decoders and pictorially show the array of memory cells.

Referring now to FIG. 19, the RAM array 601 is arranged with eight output lines, one output line from each section of 16,384 bits. The eight output lines from ROM array 601 are connected via an output latch 602 to an eight bit output register 603. The output register 603 is interconnected with pins ADD1-ADD8 and arranged either to communicate the four high or low order bits from output register 603 via the four pins ADD1-ADD8 or alternatively to communicate the bit

serially from output register 603 via pin ADD1. The particular alternative used may be selective according to mask programmable gates.

ROM array 601 is addressed via a 14 bit address counter 604. The address counter 604 has associated therewith a four bit chip select counter 605. Addresses in address counter 604 and chip select counter 605 are loaded four bits at a time from pins ADD1-ADD8 in response to a decoded Load Address (LA) command. The first LA command loads the four least significant bits in address counter 604 (bits A₀-A₃), and subsequent LA commands load the higher order bits, A₄-A₇, A₈-A₁₁ and A₁₂-A₁₃). During the fourth LA cycle the A₁₂ and A₁₃ bits are loaded at the same time the CS0 and 15 CS1 bits in chip select counter 605 are loaded. Upon the fifth LA command the two most significant bits in chip select counter 605 are loaded from ADD1 and ADD2. A counter 606 counts consecutively received LA commands for indicating where the four bits on ADD-1-ADD8 are to be inputted into counters 604 and/or 605.

Commands are sent to the ROM chip via I₀ and I₁ pins to a decoder 607 which outputs the LA command to TB (transfer bit) and a RB (read and branch) command.

Address register 604 and chip select register 605 have an add-one circuit 608 associated therewith for incrementing the address contained therein. When a carry occurs outside the fourteen bit number stored in address register 604 the carry is carried into shift select register 605 which may enable the chip select function if not previously enabled or disable the chip select function if previously enabled, for example. Alternatively, the eight bit contents of output register 603 may be loaded into address register 604 by means of selector 609 in response to an RB command. During an RB command, the first byte read out of array 601 is used as the lower order eight bits while the next successive byte is used for the higher order six bits in counter 604.

40 The output of chip select register 605 is applied via programmable connectors 610 to gate 611 for comparing the contents of chip select counter 605 with a preselected code entered by the programming of connectors 610. Gate 611 is also responsive to a chip select signal on the chip select pin for permitting the chip select feature to be based on either the contents of the four bit chip select register 605 and/or the state of the chip select bit on the CS pin. The output of gate 611 is applied to two delay circuits 612, the output of which controls the 50 output buffers associated with outputting information from output register 603 to pins ADD1-ADD8. The delay imposed by delay circuits 612 effect the two byte delay in this embodiment, because the address information inputted on pins ADD1-ADD8 leads the data outputted in response thereto by the time to require to access ROM array 601. The CS pin is preferably used in the embodiment of the learning aid disclosed herein.

A timing PLA 600 is used for timing the control signals outputted to ROM array 601 as well as the timing of other control signals.

Referring now to the composite drawing formed by FIGS. 20a and 20b output register 603 is formed by eight "A" bit latches, an exemplary one of which is shown at 617. The output of register 603 is connected in parallel via a four bit path controlled on LOW or HIGH signals to output buffers 616 for ADD1-ADD4 and 616a for ADD8. Buffers 616 and 616a are shown in detail on FIGS. 21a-21b.

Gates 615 which control the transferring of the parallel outputs from register 603 via in response to LOW and HIGH are preferably mask level programmable gates which are preferably not programmed when this chip is used with the learning aid described herein. Rather the data in register 603 is communicated serially via programmable gate 614 to buffer 616a and pin ADD8. The bits outputted to ADD1-ADD8 in response to a HIGH signal are driven from the third through sixth bits in register 603 rather than the fourth through seventh bits inasmuch as a serial shift will normally be accomplished between a LOW and HIGH signal.

Address register 604 comprises fourteen of the bit latches shown at 617. The address in address 604 on lines A₀-A₁₃ is communicated to the ROM X and Y address buffers shown on FIG. 21a-21b. Register 604 is divided into four sections 610a-601d, the 601d section loading four bits from ADD1-ADD8 in response an LA0 signal, the 610c section loading four bits from ADD1-ADD8 in response to an LA0 signal and likewise for section 601b in response to an LA2 signal. Section 601a is two bits in length and loads the ADD1 and ADD2 bits in response to an LA2 signal. The chip select register 605 comprise four B type bit latches of the type shown at 618. The low order bits, CS0 and CS1 are loaded from ADD4 and ADD8 in response to an LA3 signal while the high order bits CS2 and CS3 are loaded from ADD1 and ADD2 on an LA4 signal. The LA0-LA4 signals are generated by counter 606. Counter 606 includes a four bit register 619 comprised of four A bit latches 617. The output of the four bit counter 619 is applied to a PLA 620 for decoding the LA1-LA4 signals. The LA0 signal is generated by a NAND gate 621. As can be seen, the LA0 signal comes up in response to an LA signal being decoded immediately after a TB signal. The gate 621 looks for a logical one on the LA signal and a logical one on an LTBD (latched transer bit delay) signal from latch 622. Decoder 607 decodes the I₀ and I₁ signals applied to pins I₀ and I₁ for decoding the TB, LA and RB control signals. The signals on the I₀ and I₁ pins are set out in Table X. Latch circuit 622 is responsive to LA, RB and TB for indicating whether the previously received instruction was either an LA or a TB or RB command.

In addition to counting successive LA commands, four bit counter 609 and PLA 620 are used to count successive TB commands. This is done because in this embodiment each TB command transfers one bit from register 603 on pin ADD8 to the synthesizer chip 10 and output register 603 is loaded once each eight successive TB commands. Thus, PLA 620 also generates a TB8 command for initiating a ROM array addressing sequence. The timing sequence of counter 619 and PLA 620 are set forth in Table XI. Of course, the LA1-LA4 signal is only generated responsive to successive LA commands while the TB8 signals only generate in response to successive TB commands.

Add-one circuits 608 increments the number in program counter 604 in response to a TB command or an RB command. Since two successive bytes are used as a new address during an RB cycle, the card address and the present address incremented by one must be used to generate these two bytes. The output of add-one circuit 608 is applied via selector 609 for communicating the results of the incrementation back to the input of counter 604. Selector 609 permits the bits in output register 603 to be communicated to program counter

604 during an RB cycle as controlled by signal BR from array 600. Add-one circuit 608 is also coupled via COUNT to chip select counter 605 for incrementing the number stored therein whenever a CARRY would occur outside the fourteen bits stored in program counter 604. The output of chip select counter 605 is applied via programmable gate 610 to gate 611. The signal on the CS pin may also be applied to gate 611 or compared with the contents of CS3. Thus, gate 611 can test for either (1) the state of the CS signal, (2) a specific count in counter 605 or (3) a comparison between the state on the chip select and the state of CS3 or (4) some combination of the foregoing, as may be controlled by those knowledgeable in the art according to how programmable links 610 are programmed during chip manufacture. The output of gate 611 is applied via two bit latches of the C type, which are shown at 622. Timing array 600 controls the timing of ROM sequencing during RB and TB sequences. Array 600 includes PLA sections 600a and 600b and counter 623 and 624. Counter 623 is a two bit counter comprising two A type bit latches shown at 617. Counter 63 counts the number of times a ROM access is required to carry out a particular instruction. For instance, a TB command requires one ROM access while an RB command requires three ROM accesses. Counter 624, which comprises four "A" type bit latches of the type shown at 617, counts through the ROM timing sequence for generating various control signals used in accessing ROM array 601. The timing sequence for a TB command is shown in Table XI which depicts the states in counter 623 and 624 in the signals generated in response thereto. A similar timing sequence for an RB command is shown in Table XIII. The various signals generated by PLA 600a and 600b will now be briefly described. The BR signal controls the transfer of two serial bits from the output register 603 to the program counter 604. The TF signal controls the transfer of eight bits from the sense amp latch 602 (FIG. 21-21b) to output register 603 on lines SA0-SA7. INC controls the serial incrementing of the program counter, two bits for each INC signal generated. PC is the precharge signal for the ROM array and normally exists for approximately ten microseconds. The DC signal discharges the ROM 601 array and preferably lasts for approximately ten microseconds for each DC signal. This particular ROM array uses approximately seventy microseconds to discharge and thus seven DC signals are preferably generated during each addressing sequence. SAM gates the data outputted from the ROM into the sense amp latch 602 while SAD gates the address lines by gating the address from the program counter into the ROM address buffers 625 (FIGS. 21a-21b).

ALTERNATIVE EMBODIMENTS

Although the invention has been described with reference to a specific embodiment, this description is not meant to be construed in a limiting sense. Various modifications of the disclosed embodiment as well as alternative embodiments of the invention will become apparent to persons skilled in the art upon reference to the description of the invention. It is therefore contemplated that the appended claims will cover any such modifications or embodiments that fall within the true scope of the invention.

TABLE I

**THE FOLLOWING SEQUENCE IS AN EXAMPLE OF THE
LEARNING AID IN THE SPELLING MODE.**

KEY	DISPLAY	SPEAKER
OMPUSPELL		4 RANDOM TONES
B	SPELL A	
C	SPELL B	B
D	SPELL C	C
P	SPELL D	D
A	SPELL D	P
GO	SPELL A	A
D	—	SPELL DO AS IN DO NOT
O	D-	D
NTER	DO-	O
W	DO	THAT IS CORRECT, NOW SPELL WAS
U	—	
S	W-	W
RASE	WU-	U
W	WUS-	S
A	—	
S	W-	W
NTER	WA-	A
W	WAS-	S
A	WAS	THAT IS RIGHT, NEXT SPELL
N	—	ANY
I	A-	A
NTER	AN-	N
A	ANI-	I
N	ANI	TRY AGAIN, ANY
EPEAT	—	
EPEAT	—	ANY
E	—	ANY ($\frac{1}{2}$ SPEED)
N	E-	E
Y	EN-	N
NTER	ENY-	Y
E	ENY	THAT IS INCORRECT, THE CORRECT SPELLING OF ANY IS
A	A	A
AN	AN	N
ANY	ANY	Y
ANY	ANY	ANY
	—	NOW TRY
	FULL	FULL
J	F-	F
J	FU-	U
J	FUL-	L
J	FULL-	L
J	FULL	THAT IS CORRECT, TRY SHOE MEANING FOOTWEAR
I	S-	S
I	SH-	H
I	SHO-	O
I	SHOE-	E
ENTER	SHOE	YOU ARE CORRECT, SPELL COMB
C	—	
C	C-	C
C	CO-	O
C	COM-	M
C	COME-	E
ENTER	COME	TRY AGAIN, COMB
C	—	
C	C-	C
C	CO-	O
C	COM-	M
C	COMB-	E
ENTER	COMB	YOU ARE CORRECT, NOW SPELL FOUR AS IN THE NUMBER
C	—	
C	F-	F
C	FO-	O
C	FOU-	U
C	FOUR-	R
ENTER	FOUR	THAT IS CORRECT, NEXT SPELL WHO
N	—	
N	W-	W

TABLE I-continued

THE FOLLOWING SEQUENCE IS AN EXAMPLE OF THE
LEARNING AID IN THE SPELLING MODE.

LEARNING AID IN THE SPELLING MODE.			
	KEY	DISPLAY	SPEAKER
5	H	WH-	H
	O	WHO-	O
	ENTER	WHO	YOU ARE RIGHT, NOW TRY SOUP
10	S	S-	S
	O	SO-	O
	U	SOU-	U
	P	SOUP-	P
	ENTER	SOUP	THAT IS RIGHT. TRY MOST
15	M	M-	M
	O	MO-	O
	S	MOS-	S
	T	MOST-	T
	ENTER	MOST	YOU ARE CORRECT
20	+8 -2	4 TONES	
	+8 -2	4 TONES	
	+8 -2	HERE IS YOUR SCORE, EIGHT CORRECT, TWO DID NOT COMPUTE.	
25			
		TABLE II	
		<u>LEARN MODE</u>	
	KEY	DISPLAY	SPEAKER
30		BUSY	(1 SECOND PAUSE) SAY IT (2 SECOND PAUSE)
		MANY	BUSY (1 SECOND PAUSE) SAY IT (2 SECOND PAUSE)
35		CARRY	MANY (1 SECOND PAUSE) SAY IT (2 SECOND PAUSE)
40		YOUR	CARRY (1 SECOND PAUSE) SAY IT (2 SECOND PAUSE)
		WILD	YOUR (1 SECOND PAUSE) SAY IT (2 SECOND PAUSE)
45		LOVE	WILD (1 SECOND PAUSE) SAY IT (2 SECOND PAUSE)
50		BUSH	LOVE (1 SECOND PAUSE) SAY IT (2 SECOND PAUSE)
	REPEAT	IGNORED	BUSH (1 SECOND PAUSE) SAY IT (2 SECOND PAUSE)
55	REPEAT	EARN	IGNORED BUSH (1 SECOND PAUSE) SAY IT (2 SECOND PAUSE)
	REPEAT		EARN (1 SECOND PAUSE) SAY IT (2 SECOND PAUSE)
60	M	—	EARN SPELL MANY
	A	M-	M
	N	MA-	A
	Y	MAN-	N
	ENTER	MANY-	Y
		MANY	YOU ARE CORRECT, NOW SPELL EARN
65			

TABLE III

IN THE WORD GUESSER MODE THE LEARNING AID RANDOMLY SELECTS A WORD FROM LEVEL C OR D AND DISPLAYS DASHES TO REPRESENT THE NUMBER OF LETTERS IN THE CHOSEN WORD. THE USER TRIES TO GUESS THE WORD. THE USER MUST COMPLETE THE WORD BEFORE MAKING SEVEN INCORRECT GUESSES. THE FOLLOWING IS AN EXAMPLE OF THE FUNCTION OF THE LEARNING AID IN THE SPELLING MODE.

KEY	DISPLAY	SPEAKER	
HANGMAN	-----	4 TONES	
A	-----		10
E	E-E---E	4 TONES	
I	E-E---E		
O	E-E-O-E	4 TONES	15
U	E-E-O-E		
B	E-E-O-E		
C	E-E-O-E		
D	E-E-O-E		
F	E-E-O-E		
	EVERYONE	4 TONES, I WIN	20

A	-----		
E	----E	4 TONES	
I	----E		
O	-O--E	4 TONES	25
U	-OU-E	4 TONES	
B	-OU-E		
C	COU--E	4 TONES	
R	COUR-E	4 TONES	
S	COURSE	4 TONES	30
	COURSE	4 TONES, YOU WIN	

TABLE IV

The synthesizer 10 includes interpolation logics to accomplish a nearly linear interpolation of all twelve

speech parameters at eight points within each frame that is, once each 2.5 msec. The parameters are interpolated one at a time as selected by the parameter counter. The interpolation logics calculate a new value of a parameter from its present value (i.e. the value currently stored in the K-stack, pitch register or E-10 loop) and the target value stored in encoded form in RAM 203 (and decoded by ROM 202). The value computed by each interpolation is listed below.

Where

P_i is the present value of the parameter,

P_{i+1} is the new parameter value

P_t is the target value

N_i is an integer determined by the interpolation counter

The values of N_i for specific interpolation counts and the values

$$\frac{P_i - P_o}{P_t - P_o}$$

(P_0 is initial parameter value) are as follows:

	INTERPOLATION COUNT	N_i	$\frac{P_i - P_o}{P_t - P_o}$
			$P_i - P_o$
	1	8	0.125
	2	8	0.234
	3	8	0.330
	4	4	0.498
	5	4	0.623
	6	2	0.717
	7	2	0.859
	0	1	1.000

TABLE V

ENERGY	PITCH	REPEAT	"HELP"											
			K ₁	K ₂	K ₃	K ₄	K ₅	K ₆	K ₇	K ₈	K ₉	K ₁₀		
0000														
0100	00000	0	10011	01110	1001	0111								
0111	00000	1												
1101	10010	0	10000	10100	1000	0110	0111	1000	1010	100	101	010		
1101	10011	1												
1110	10011	1												
1101	10100	0	01101	01111	1010	1010	1001	0111	1000	100	101	101		
1101	10100	0	01110	01011	1000	1100	1101	1000	0100	100	011	101		
1101	10011	0	10001	01010	0110	1001	1111	1011	0101	010	000	110		
1011	10010	1												
1010	10010	0	01101	00111	1000	1100	1111	0111	0010	001	010	110		
1001	10000	1												
1000	01110	1												
0010	01101	1												
0000														
0000														
0111	00000	0	10100	01011	1011	1000								
0111	00000	0	10001	01011	1011	0110								
0101	00000	1												
0011	00000	0	10011	00111	1010	0110								
0010	00000	0	10010	00101	1011	0101								
0000														
1111														

HEL

P

TABLE VI

CODE	E	P	DECODED PARAMETERS											
			K ₁	K ₂	K ₃	K ₄	K ₅	K ₆	K ₇	K ₈	K ₉	K ₁₀		
00	000	000	208	2A3	273	28F	2C1	2DE	2DD	326	31F	34D		
01	000	02F	20F	2B8	293	282	2E2	304	300	37B	363	386		
02	001	02B	213	2CF	2B9	2D8	306	32F	328	3DA	3AF	3C3		
03	001	02D	218	2F8	2E6	30B	320	35D	352	038	3FD	001		
04	002	02F	229	304	31B	341	358	38E	380	098	04C	03F		

TABLE VI-continued

DECODED PARAMETERS												
CODE	E	P	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10
5	003	031	229	321	356	370	286	3C2	3B0	0EB	097	07B
6	005	033	234	340	398	3HD	286	3F7	3E1	131	0DC	0B3
7	007	035	242	362	3DC	3FF	3E7	02C	013	169	118	0E7
8	00A	037	255	384	023	040	018	061	045			
9	00F	03A	268	3A8	068	080	049	083	075			
A	015	03C	286	3CD	0A9	0BC	079	0C2	0A3			
B	01F	03F	2A8	3F2	DF4	DF3	0A7	DEF	0CF			
C	028	042	2CF	017	119	123	0D2	116	0F6			
D	03D	046	2FD	03C	146	14C	0F9	139	118			
E	056	048	332	061	16C	16F	11D	158	13C			
F	000	04C	360	085	18C	18D	13E	173	159			
0		04F	3AA	0A7								
1		053	3F8	0C7								
2		057	02D	0E6								
3		05A	06E	103								
4		05E	0AB	11F								
5		063	0F3	136								
6		067	115	14D								
7		068	147	162								
8		070	165	1754								
9		176	184	185								
A		018	19D	194								
B		081	182	1A1								
C		086	163	1AD								
D		080	1D0	187								
E		093	1DA	101								
F		099	1E2	1FA								

TABLE VII

K-STACK	DATA OUTPUTTED FROM K-STACK 302 TO RECODING LOGIC 301 BY TIME PERIODS																				
	OUTPUT	TIME PERIODS																			
BIT LINE		T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25	T26	T27
L-SB	K2	K1	A	K9	K8	K7	K6	K5	K4	K3	K2	K1	K10	K9	K8	K7	K6	K5	K4	K3	
32-1	K2	K1	A	K9	K8	K7	K6	K5	K4	K3	K2	K1	K10	K9	K8	K7	K6	K5	K4	K3	
32-2	K2	K1	A	K9	K8	K7	K6	K5	K4	K3	K2	K1	K10	K9	K8	K7	K6	K5	K4	K3	
32-3	K2	K1	A	K9	K8	K7	K6	K5	K4	K3	K2	K1	K10	K9	K8	K7	K6	K5	K4	K3	
32-4	K2	K1	A	K9	K8	K7	K6	K5	K4	K3	K2	K1	K10	K9	K8	K7	K6	K5	K4	K3	
32-5	K3	K2	K1	A	K9	K8	K7	K6	K5	K4	K3	K2	K1	K10	K9	K8	K7	K6	K5	K4	
32-6	K3	K2	K1	A	K9	K8	K7	K6	K5	K4	K3	K2	K1	K10	K9	K8	K7	K6	K5	K4	
32-7	K4	K3	K2	K1	A	K9	K8	K7	K6	K5	K4	K3	K2	K1	K10	K9	K8	K7	K6	K5	
32-8	K4	K3	K2	K1	A	K9	K8	K7	K6	K5	K4	K3	K2	K1	K10	K9	K8	K7	K6	K5	
32-9	K5	K4	K3	K2	K1	A	K9	K8	K7	K6	K5	K4	K3	K2	K1	K10	K9	K8	K7	K6	
MSB	32-10	K5	K4	K3	K2	K1	A	K9	K8	K7	K6	K5	K4	K3	K2	K1	K10	K9	K8	K7	K6

TABLE VIII

ADDRESS	CHIRP ROM CONTENTS			45
	CHIRP FUNCTION VALUE	STORED VALUE	(COMPLEMENTED)	
00	00	FF		
01	2A	D5		
02	D4	2B		
03	32	CD		
04	B2	4D		
05	12	ED		
06	25	DA		
07	14	EB		
08	02	FD		
09	E1	IE		
10	C5	3A		
11	02	FD		
12	5F	A0		
13	5A	A5		
14	05	FA		
15	0F	F0		
16	26	D9		
17	FC	03		
18	A5	5A		
19	A5	5A		
20	D6	29		

TABLE VIII-continued

ADDRESS	CHIRP ROM CONTENTS		
	CHIRP FUNCTION VALUE	STORED VALUE	(COMPLEMENTED)
21	DD	22	
22	DC	23	
23	FC	03	
24	25	DA	
25	2B	D4	
26	22	DD	
27	21	DE	
28	0F	F0	
29	FF	00	
30	F8	07	
31	EE	11	
32	ED	I2	
33	EF	10	
34	F7	08	
35	F6	09	
36	FA	05	
37	00	FF	
38	03	FC	
39	02	FD	
40	01	FE	

TABLE IX-0 LEARNING AID INSTRUCTION SET

TABLE IX-0 (Continued)

4,189,779

43

44

.65

TABLE IX-0 (Continued)

TABLE IX-0 (Continued)

		142	CALL	TRANSMIT?	GET SO OF RANDOM LETTER
0042	000000000000	0143	0143	0	* SILENCE
0043	000000000000	1423	0143	0	* SILENCE
0044	000000000000	1424	0144	0	* SILENCE
0045	000000000000	1425	0145	0	* SILENCE
0046	000000000000	1426	0146	0	* SILENCE
0047	000000000000	1427	0147	0	* SILENCE
0048	000000000000	1428	0148	0	* SILENCE
0049	000000000000	1429	0149	0	* SILENCE
0050	000000000000	1430	0150	0	* SILENCE
0051	000000000000	1431	0151	0	* SILENCE
0052	000000000000	1475	0191	0	* SAYS LETTER AND TRANSFER
0053	000000000000	0142	*	*	* PLITS IT IS DISPLAY

TABLE IX-1

		142	CALL	TRANSMIT?	GET SO OF RANDOM LETTER
0000	000000000000	0193	0193	1	0000
0001	000000000000	0194	0194	1	0001
0002	000000000000	0195	0195	1	0002
0003	000000000000	0196	0196	1	0003
0004	000000000000	0197	0197	1	0004
0005	000000000000	0198	0198	1	0005
001F	000000000000	0199	0199	1	001F
003F	000000000000	0200	0200	1	003F
007F	000000000000	0201	0201	1	007F
007F	000000000000	0202	0202	1	007F
007F	000000000000	0203	0203	0	007F
007F	000000000000	0204	0204	0	007F
007F	000000000000	0205	0205	0	007F
007F	000000000000	0206	0206	0	007F
007F	000000000000	0207	0207	0	007F
007F	000000000000	0208	0208	0	007F
007F	000000000000	0209	0209	0	007F
007F	000000000000	0210	0210	*	007F WHICH MODE YOUR IN AND BRANCHES
007F	000000000000	0211	0211	*	007F WHICH MODE YOUR IN AND BRANCHES
007F	000000000000	0212	0212	*	007F WHICH MODE YOUR IN AND BRANCHES

TABLE IX-1 (Continued)

NAME	FUNCTION	CODE	NAME	FUNCTION	CODE
SFT GO MODE FLAG		0753	0214	KEY IN	0215
TEST WHICH MODE		0673	0215	KEY OUT	0216
		0673	0216	TCV	0217
		0667	0217	TCV	0217
		064F	0218	TCV	0218
		060E	0219	SHIT	0219
		001E	0220	TCV	0220
		003D	0221	TCV	0221
		016104010	0222	LTX	0222
		007A	0223	TCV	0223
		016104010	0224	TCWY	0224
		0075	0225	CORR&SPL	0225
		003A	0226	CALL	0226
		116666666	0227	LDP	0227
		00970	0228	ALFC	0228
		016104010	0229	BRANCH	0229
		0061	0230	LDP	0230
		011101010	0231	ALFC	0231
		0003	0232	BRANCH	0232
		166666666	0233	LDP	0233
		0006	0234	ALFC	0234
		011101010	0235	BRANCH	0235
		0006	0236	TCV	0236
		011101010	0237	TCV	0237
		001A	0238	TCV	0238
		011101010	0239	TCV	0239
		0053	0240	TCV	0240
		011101010	0241	TCV	0241
		0034	0242	TCV	0242
		011101010	0243	TCV	0243
		0074	0244	TCV	0244
		011101010	0245	TCV	0245
		0069	0246	TCV	0246
		011101010	0247	TCV	0247
		0053	0248	TCV	0248
		011101010	0249	RPTV	0249
		0062	0250	TCV	0250
		011101010	0251	TCV	0251
		0045	0252	ENTER	0252
		011101010	0253	ENTER	0253
		000A	0254	PRINCESS	0254
		011101010	0255	ENTER	0255
		0015	0256	ENTER	0256
		011101010	0257	ENTER	0257
		0024	0258	ENTER	0258
		011101010	0259	ENTER	0259
		002C	0260	ENTER	0260

TABLE IX-1 (Continued)

55

60

65

TABLE IX-1 (Continued)

				SEARCH FROM CHASER	
				SEARCHED	SEARCHED
0016	1...1-1 111	0373	1205	LDR	1
0021	1...1-1 111	0297	1207	LDR	1
0042	1...1-1 111	0275	1208	LDR	1
0044	1...1-1 111	0299	1209	LDR	1
				SPACE?	1
				SEARCHES LOANS FROM ANGUS WITH SECOND MANNING RESPONSE	
				4301	
				4302	
				4303	
				4304	
				4305	
				4306	
				4307	
				4308	
				4309	
				4310	
				4311	
				4312	
				4313	
				4314	
				4315	
				4316	
				4317	
				4318	
				4319	
				4320	
				4321	
				4322	
				4323	
				4324	
				4325	
				4326	
				4327	
				4328	
				4329	
				4330	
				4331	
				4332	
				4333	
				4334	
				4335	
				4336	
				4337	
				4338	
				4339	
				4340	
				4341	
				4342	
				4343	
				4344	
				4345	
				4346	
				4347	
				4348	
				4349	
				4350	
				4351	
				4352	
				4353	
				4354	
				4355	
				4356	
				4357	
				4358	
				4359	
				4360	
				4361	
				4362	
				4363	
				4364	
				4365	
				4366	
				4367	
				4368	
				4369	
				4370	
				4371	
				4372	
				4373	
				4374	
				4375	
				4376	
				4377	
				4378	
				4379	
				4380	
				4381	
				4382	
				4383	
				4384	
				4385	
				4386	
				4387	
				4388	
				4389	
				4390	
				4391	
				4392	
				4393	
				4394	
				4395	
				4396	
				4397	
				4398	
				4399	
				4400	
				4401	
				4402	
				4403	
				4404	
				4405	
				4406	
				4407	
				4408	
				4409	
				4410	
				4411	
				4412	
				4413	
				4414	
				4415	
				4416	
				4417	
				4418	
				4419	
				4420	
				4421	
				4422	
				4423	
				4424	
				4425	
				4426	
				4427	
				4428	
				4429	
				4430	
				4431	
				4432	
				4433	
				4434	
				4435	
				4436	
				4437	
				4438	
				4439	
				4440	
				4441	
				4442	
				4443	
				4444	
				4445	
				4446	
				4447	
				4448	
				4449	
				4450	
				4451	
				4452	
				4453	
				4454	
				4455	
				4456	
				4457	
				4458	
				4459	
				4460	
				4461	
				4462	
				4463	
				4464	
				4465	
				4466	
				4467	
				4468	
				4469	
				4470	
				4471	
				4472	
				4473	
				4474	
				4475	
				4476	
				4477	
				4478	
				4479	
				4480	
				4481	
				4482	
				4483	
				4484	
				4485	
				4486	
				4487	
				4488	
				4489	
				4490	
				4491	
				4492	
				4493	
				4494	
				4495	
				4496	
				4497	
				4498	
				4499	
				4500	
				4501	
				4502	
				4503	
				4504	
				4505	
				4506	
				4507	
				4508	
				4509	
				4510	
				4511	
				4512	
				4513	
				4514	
				4515	
				4516	
				4517	
				4518	
				4519	
				4520	
				4521	
				4522	
				4523	
				4524	
				4525	
				4526	
				4527	
				4528	
				4529	
				4530	
				4531	
				4532	
				4533	
				4534	
				4535	
				4536	
				4537	
				4538	
				4539	
				4540	
				4541	
				4542	
				4543	
				4544	
				4545	
				4546	
				4547	
				4548	
				4549	
				4550	
				4551	
				4552	
				4553	
				4554	
				4555	
				4556	
				4557	
				4558	
				4559	
				4560	
				4561	
				4562	
				4563	
				4564	
				4565	
				4566	
				4567	
				4568	
				4569	
				4570	
				4571	
				4572	
				4573	
				4574	
				4575	
				4576	
				4577	
				4578	
				4579	
				4580	
				4581	
				4582	
				4583	
				4584	
				4585	
				4586	
				4587	
				4588	
				4589	
				4590	
				4591	
				4592	
				4593	
				4594	
				4595	
				4596	
				4597	
				4598	
				4599	
				4600	
				4601	
				4602	
				4603	
				4604	
				4605	
				4606	
				4607	
				4608	
				4609	
				4610	
				4611	
				4612	
				4613	
				4614	
				4615	
				4616	
				4617	
				4618	
				4619	
				4620	
				4621	
				4622	
				4623	
				4624	
				4625	
				4626	

4,189,779

55

56

TABLE IX-2 (Continued)

4,189,779

TABLE IX-2 (Continued)

		MISS1	MISS2	CLA	ACACC	12
		0425	0426	LDX	LDX	0
0050	0000000110		0427			
0059	0011100111		0428			
0032	0100000000		0429			
0064	0101111111		0430			
0049	0100000101	1546	0431	CALLI	CLEAR	
0012	10011001		0432			
0025	0100001000		0433	LDX	B	
0044	110111010	0236	0434	TCY	8	
0014	0100110001		0435	SHTL	2	
0020	0010000001		0436	RATT	1	
0052	0101000001		0437	TCY	4	
0026	0101001110		0438	LDX	0	
0044	0100100000		0439	TCMIV	2	
0010	0100110000000000		0440	TCY	7	
0021	0101001100		0441	TCMIV	2	
0042	0010001110		0442	LDX	1	
0004	0011000100		0443	TCY	0	
0009	0100111000		0444	TCMIV	13	
0015	0010000000		0445	TCY	6	
0027	0011010111		0446	TCMIV	14	
004F	001000110		0447	LDX	5	
001C	0011001111		0448	TCY	13	
0039	010011010		0449	TMA		
0072	0010001011		0450	LDX	1	
0065	0001001001		0451	TCY	7	
0043	0100110000		0452	JAM		
0016	001001110		0453	CALLI	FL2	
0020	000101111		0454			
005A	0100001110		0455	LDX	1	
0054	1100001100	1145	0456	TCY	1	
0065	0100110000		0457	TAR		
0067	0100110000		0458	HL	SCORE	
0051	0010001000		0459			
0022	0001011111		0460			
0044	0100001000		0461			
0004	1001000010	0314	0462	SPELL	LDX	6
			0463		TCY	7
			0464		TCMIV	0
			0465		BRANCH	SPELL9

4,189,779

59

60

TABLE IX-2 (Continued)

0019	010010001	0466	LEARN	LUX	7
0033	00101110	0467	TCV	7	
0069	00110000	0468	TCMV	2	
0030	010001111	0469	SPELL4	NSPT	
001A	10110000	2188	0470		
0035	001111100		0471	MISS3	3
0064	000101111		0472	TA4	4
2655	010000010		0473	LDP	4
0024	011100110		0474	ALEC	6
0054	100101100	04680	0475	NOSTRANS	
0024	010001009		0476	BRANCH	
0050	101101100		0477	HL	

TABLE IX-3

0000	01001100	0478	0479	GAMES1	3
0001	000000110	0480	0481	LOX	2
0003	001000011	0481	0482	CLA	2
0007	000100111	0483	0484	TCV	13
0009	010011100	0485	0486	TA4	13
001F	001100111	0484	0485	LDP	3
0035	000001011	0485	0486	TCMV	14
007F	000000000	0486	0487	TCV	10
007F	101111111	0486	0487	IK11	0
007F	001111000	0486	0487	BRANCH	HANG2
007F	001111000	0486	0487	ACACC	1
0078	001111000	0489	0490	HANG2	
0077	010011000	0490	0491	LIX	
0068	001001111	0491	0492	TCV	15
005F	000101111	0492	0493	TA4	15
003F	010010001	0493	0494	LIX	8
007C	001001110	0494	0495	TCV	7
0079	001101010	0495	0496	TCMV	5
0074	010001100	0496	0497	HL	CURLEVEL
0067	101101111	0497			

0476
CLEAR GUESS COUNTER
HANGMAN FLAG
TEST RANDOM COUNTER
BIT AND PUT 2 OR 3
IN ACC
STORE 2 OR 3 IN LEVEL
OF DIFFICULTY
NAME
SET HANGMAN MODE

TABLE IX-3 (Continued)

004F	010001000	0236	0502	TCV	A	
001F	110111010	0236	0505	DYN		
0050	001000001	0501	HANG	CALL	CLEAN	
007A	000900100	0504	HANGS			PUT BLANKS IN DISPLAY
0075	0100006100	0505				
0064	110101110	0374	0506	ALEC	0	
0057	011100000	0507		HANG3		
002E	101111010	0504	0508	BRANCH		
005C	010011000	0509		DIGIT THAT IS NOT A		
0034	000101100	0510		FINDS THE FIRST		
0079	100111000	0514	0515	BLANK, STARTING		
0361	000001101	0516	SONG	FROM THE RIGHT SIDE!		
0043	100100011	1657	0517	LDX		
0016	001001011			THE ROUTINE BELOW THEN PUTS CURSORS IN		
0037	001000001			THE DIGITS CORRESPONDING TO LETTERS		
000E	001010001			LDX		
0050	000000100			HANG4		
005A	100101101	0562	0527	TAMDN		
0074	001000001	0528		HANCH		
0059	010101101	0529	HANG10	TCV		
0053	000000100	0530	DYN	RHT	3	
0026	100100001	0529		HANCH		
004C	001000101	0531		HANG10		
0018	000100000	0532		TCV	13	
0051	101100011	0533		RHT	0	
0062	010011000	0555	0534	HANCH	HANG11	
0045	000100000	0535		LDX	?	
000A	000010111	0536		INAC		
						* ADD 1 TO INCORRECT
						* GUESS COUNTER

4,189,779

63

64

TABLE IX-3 (Continued)

00015	0100001111	0536	LDP	15	
00028	011100110	0539	ALEC	6	
00056	100101100	2219	0540	RWANCH	DISP/KH
0002C	010011000	0541	TWIN	10X	1
00058	001000101	0542		TCV	10
00030	001100000	0543		TCMIV	0
00060	001101110	0544		TCMIV	7
00041	001100000	0545	TWIN	TCMIV	6
00002	001100000	0546		TCMIV	0
00005	001100111	0547		TCV	15
00008	010010100	0548		LOX	2
00017	001110000	0549		TCMIV	0
0002F	010010001	0550		LOX	8
0005E	001100001	0551		TCV	6
0003C	010100110	0552		KBT	1
0007A	010000101	0553	HE	LOADDISP	
00071	101110001	1456	0554		
00083	000100010	0555	HANG11	TRIT	1
00047	101100001	0516	0556	RWANCH	SONG
0000E	001000101	0557	YOUWIN	TCV	10
0001D	010010000	0558		LOX	1
00038	001100100	0559		TCMIV	2
00076	001101110	0560		TCMIV	7
00060	101000001	0545	0561	RWANCH	TWIN1
00058	010000100	0562	HANG6	CALLL	SPLNTR#1
00036	100101110	0374	0563		
0006C	011100000	0564		ALEC	0
00059	101101110	0525	0565	RWANCH	HANG5
00032	001001111	0566		TCV	15
00049	000101001	0567	FLD01Y	TMA	
0004A	001000001	0568		TCV	
00010	000101111	0570		DYN	
00012	000000100	0569	HANG7	GRANCH	HANG7
00025	000100001	0570			
0004A	100000010	0560	0571		
00010	000101111	0572		HEIN	
00029	000101111	0573		TAN	
00052	001000111	0574		TCV	14
00024	010001100	0575		CALLL	FIND01
00044	111000100	0567	0576		LDP
	00010	0577			

,189,779

65

66

65

TABLE IX-3 (Continued)

0021	0001011111	0578	TAM	SPLNTR#1
0042	0100001000	0579	CALLI	CHECK TO SEE IF
0004	1101011110	0374	0580	NEW LETTER MATCHES
0009	0111000000	0581	ALEC	*
0013	1001011101	0591	HANGA	*
0027	0100011000	0582	BRANCH	*
004F	01-01010111	0583	LDX	DOES NOT MATCH
001C	0011000111	0584	TYA	*
0019	0100100000	0585	TCMIV	PUT BLANK BACK
0039	0010101000	0586	LDX	*
0072	0010010111	0587	TCY	*
0065	0101030910	0588	SHIT	IN DISPLAY
004B	0001010100	0589	TAY	*
0016	1001101010	0590	BRANCH	SET FLAG FOR WORD NOT COMPLETED
0020	00-01010111	0591	HANG8	BET
0054	0010010111	0592	TYA	CORRECT LETTER GUESS
0054	0101000000	0593	TCY	*
0069	0001010100	0594	SHIT	*
0051	1011011110	0595	TAY	CORRECT LETTER FLAG IF Y=13
0596	*	0596	BRANCH	HANG5
0597	*	0597	BRANCH	HANG5
0598	*	0598	NEXTWORD	RESETS FLAGS, INCREMENTS COUNTERS AND POINTERS
0022	0101100010	0599	COMB6	*
0044	0010001010	0600	TCY	INCREMENT PHRASE COUNTER
0008	0001010101	0601	TMA	*
0011	0011101000	0602	ACACC	*
0023	01110001	0603	ALFC	*
0046	1000110001	0604	BRANCH	*
000C	000006110	0605	CLAB	*
0019	0001011111	0606	NXT?	RESET HITS FLAG6
0055	0010001010	0607	TAM	*
0069	0101010000	0608	PHLT	*
0049	001010110	0609	WHIT	*
001A	0010000000	0610	TCY	*
0035	000110010	0611	IMAC	*
006A	0001011111	0612	TAN	*
0055	0001010100	0613	MY	*
002A	0100001000	0614	TOP	*
0054	0010101011	0615	YHFC	INCREMENT RHE PUNTER
0024	1000000111	0345	BRANCH	*
0050	0100001009	0616	HL	*
0020	1001000111	0431	F3	*
		0617		
		0618		

TABLE IX-4

		PUTS BLANKS AND CURSOR IN DISPLAY		4		CLEAR	
		0619	0620	GAME #2	ORGPG	CALL	CLEAR
0000	010001000	0236	0621		LDX	A	
0001	110111010				TCY	7	
0003	010001001		0622		TCMIV	6	
0007	001001110		0623		SFT	1	
000F	001100110		0624		AL	TONES	
001F	0101010010		0625				
003F	010001101		0626				
007F	101000111	1657	0627				
			0628	*	DIFFSLV	LDX	8
007E	010010001		0629		TCY	SEVEN	
007D	001001110		0630		IMA		
007B	000101001		0631		LDX	0	
0077	010010000		0632		TCY	0	
006F	001000000		0633				
005F	001101000		0634	BLANKH	TCMIV	1	
003F	001010001		0635		YNEC	8	
007C	101011111	0634	0636		BRANCH	BLANKH	
0079	001001000		0637		TCY	1	
0075	011100000		0638		ALEC	0	
0067	101000011	065A	0639		BRANCH	LZERO\$	
			0640	*			
004F	001100000		0641		TCMIV	0	
001E	001000010		0642		TCY	4	
003D	001100000		0643		TCMIV	0	
			0644	*	LDX	ONE	
007A	010011000		0645		TCY	DISPLAY	
0075	001100000		0646		TCMIV	S	
006A	001100100		0647		TCMIV	2	
0057	001100000		0648		TCMIV	0	
002E	001100001		0649		TCMIV	8	
005C	001101101		0650		TCMIV	11	
003A	001100001		0651		TCMIV	8	
0070	001101100		0652		TCMIV	3	
0061	101110100	0665	0653		BRANCH	BLANK	
			0654	*			
			0655	*			
0043	001100000		0656	LZERO\$	TCMIV	0	
0006	001011010		0657		YNEC	5	
0000	101000011	0656	0658		BRANCH	LZERO\$	
001H	010011000		0659		LDX	UNF	

0037	001000000	0661	TCY	DISPLAY
006E	001100100	0662	TCMIV	LSSDS
0050	001101111	0663	TCMIV	LSSDP
003A	001100010	0664	TCMIV	LSSSE
0074	001101101	0665	BLANK	TCMIV
0069	001010001	0666	YNEC	TCMIV
0053	101110100	0665	BRANCH	ALANK
0026	001001111	0668	PUTSLVL	TCV
004C	000101001	0669	THA	PUT LEVEL IN DISPLAY
001A	001001110	0670	TCV	AA
0031	000101111	0671	TAM	AA
0062	010000000	0672		
0045	001100000	0673		
0004	010110010	0674		
0015	001000001	0675		
002H	001100000	0676		
0056	910111111	0677		
002C	010010000	0678		
0058	001001111	0679		
0030	000101001	0680		
0060	010011000	0681		
0041	001000001	0682		
0002	001100000	0683		
0095	001110000	0684		
0008	000100110	0685		
0017	000010111	0686		
002F	010000000	0687		
005F	111011000	0690		
003C	010010000	0691		
0074	001001111	0692		
0071	000101001	0693		
0063	010011000	0694		
0047	001000101	0695		
000F	000100111	0696		
001D	010000000	0697		
003B	111011000	0712		
0076	000000010	0698		
0060	001110011	0699		CIA
0054	001000101	0700		ACACC
0036	010000000	0701		TCY
006C	111011000	0702		10
0059	010000010	0703		CALL
		0704		AUDCARRY
				MEMADUR

TABLE IX-4 (Continued)

			L ADDRESS	CALLL		
0032	111011000	1501	0705			
0064	010001100	0706				
0049	111000010	1121	0707			
0012	010000011	0708				
0025	100001010	2057	0709			
0044	0100010100	0710	RETNSHCM LDX	2		
0014	001001111	0711	TCV	15		
0029	000101001	0712	TMA			
0052	010001111	0713	LDP	15		
0024	011101000	0714	ALEC	1		
0048	100101100	2219	0715			
0010	010001101	0716	BRANCH	DISP/KB		
0021	011100100	0717	LUP	1		
0042	101000010	1680	0718	NXTTONE		
0004	010001100	0719	LDP	3		
0009	011101100	0720	ALEC	3		
0013	100100010	0599	0721	BRANCH	NXTWORD	
0027	010000101	0722	LDP	10		
0046	011100010	0723	ALEC	4		
001C	100001001	1540	0724	BRANCH	MSPEL3	
0039	010000001	0725	LDP	8		
0072	011101010	0726	ALEC	5		
0065	010110011	1232	0727	BRANCH	DISP+S	
004A	010001101	0728	LDP	9		
0016	011100110	0729	ALEC	6		
002D	101110110	1372	0730	BRANCH	LET+4	
0054	010001100	0731	LDP	3		
0034	011101110	0732	ALEC	7		
0068	100000110	0521	0733	BRANCH	HANG1	
0051	011100001	0734	ALEC	8		
0022	100000000	0479	0735	BRANCH	GAMES!	
0044	010000101	0736	LDP	10		
0004	011101001	0737	ALEC	9		
0011	101101010	1570	0738	BRANCH	ADDCTW2	
0023	010000001	0739	LDP	6		
0064	011101010	0740	ALEC	10		
000C	101100011	1232	0741	BRANCH	DISP+S	
		0742	*	VSTR12	-USED IN LOADING UNK/EDT TO TEST FOR 3 WORDS OF ZERO	
		0743	*	1 WORD OF 0001		
		0744	*	DAM REG		
		0745	TSTBLT2	COMMX		
		0746	TCV	2		
		0747	WHIT	1		
		0748	KBIT	2		
		0749	COMMX			
		0750	LET+			

TABLE IX-5

		0751	ORGPG	S	**
		0752	* STORE SEED NUMBER		
		0753	RANDOM	LDX	4
0000	0100010010	0754		TCY	10
0001	001000101	0755		CLA	
0003	000000110	0756		CALLL	FILSLOOP
0007	010001111	2183	0757		
000F	110101110		0758	TCY	7
001F	001001110		0759	LDX	8
003F	010010001		0760	LDP	7
007F	010001110		0761	TBIT	0
007E	000100000		0762	BRANCH	LDPREV
007D	101110011	1039	0763	SBIT	0
007B	010100000		0764	LDP	5
0077	010001010		0765	* CURLEVEL-->	
			0766	* STORES NUMBER OF ENTRIES IN CURRENT LEVEL	
			0767	* INTO RAM	
			0768	*	
006F	001000101		0769	CURLEVEL	TCY 10
005F	010011000		0770		LDX 1
			0771	* ZERO OUT ROM ADDR	
003E	001100000		0772		TCMIY 0
007C	001100000		0773		TCMIY 0
0079	001100000		0774		TCMIY 0
0073	001100000		0775		TCMIY 0
0067	001000101		0776		TCY 10
004F	010111111		0777		RETN
			0778	* FIND DIFFICULTY LEVEL	
001E	001001111		0779		TCY 15
0030	000101001		0780		TMA
007A	001000101		0781		TCY 10
0075	000101111		0782		TAM
0068	010000111		0783	CALLL	ADDR
0057	110001100	2139	0784		
002E	010000101		0785	CALLL	MEMADDR
005C	111011000	1501	0786		
			0787	* OUTPUT # OF ENTRIES IN THIS LEVEL	
0038	010001110		0788	CALLL	OUTADDR2
0070	111000001	1083	0789		
0061	001001111		0790		TCY 15
0043	010011010		0791		LDX 5
0006	000101111		0792		TAM
000D	010001110		0793	CALLL	OUTADDR2
0018	111000001	1083	0794		
0037	001001111		0795		TCY 15
006E	010010010		0796		LDX 4
005D	000101111		0797		TAM
003A	010011010		0798		LDX 5
0074	000000111		0799		DMAN
0069	100011000	0804	0800	BRANCH	DECMEM
0053	000101111		0801		TAM
0026	0100010010		0802		LDX 4
004C	000000111		0803		DMAN
0018	000101111		0804	DECMEM	TAM
0031	010011100		0805		LDX 3
0062	001000001		0806		TCY 8
0045	000101001		0807		TMA
000A	010011010		0808		LDX 5
0015	001000000		0809		TCY 0
0028	000101111		0810		TAM
0056	010011100		0811		LDX 3
002C	001000101		0812		TCY 9
0058	000101001		0813		TMA
0030	010010010		0814		LDX 4
0060	001000000		0815		TCY 0
0041	000101111		0816		TAM

TABLE IX-5 (Continued)

		0817	* DETERMINE IF SEED IS > NUMBER OF ENTRIES		
0002	001091111	0818	DECLOOP	TCY	15
0005	000000001	0819	ALEM		
0008	101111000	0820	BRANCH	RANOK	
0017	001000000	0821	TCY	0	
002F	001111100	0822	ACACC	3	
005E	000101111	0823	TAM		
003C	100000010	0824	BRANCH	DECLOOP	
0074	000001001	0825	RANOK	MNEA	
0071	101011001	0826	BRANCH	RANOK2	
0063	001000000	0827	TCY	0	
0047	010011010	0828	LDX	5	
000E	000101001	0829	TMA		
0010	001001111	0830	DECLOOPS	TCY	15
0038	000000001	0831	ALEM		
0076	101011001	0832	BRANCH	RANOK2	
0060	001000000	0833	TCY	0	
0058	001111100	0834	ACACC	3	
0036	000101111	0835	TAM		
006C	100011101	0836	BRANCH	DECLOOP3	
0059	010110010	0837	RANOK2	RCOMX8	
		0838	* ZERO RWE POINTER		
0032	001000000	0839	TCY	0	
0064	001100000	0840	TCMIY	0	
0049	010011010	0841	RPLLOOP	LDX	5
0012	010001101	0842	CALLL	RCOMX8	
0025	111001100	1631	0843		
004A	000101001	0844	TMA		
0014	000000101	0845	IYC		
0029	001111000	0846	ACACC	1	
0052	110101000	0847	CALL	INCARRY	
0024	000101100	0848	TAMDYN		
0048	010010010	0849	LDX	4	
0010	000101001	0850	TMA		
0021	000000101	0851	IYC		
0042	000010101	0852	AMAAC		
0004	000101111	0853	TAM		
0009	010001101	0854	RANARND	CALLL	RCOMX8
0013	111001100	1631	0855		
0027	000101001	0856	TMA		
004E	001001111	0857	TCY	15	
001C	000000001	0858	ALEM		
0039	101100101	0859	BRANCH	RANCNT	
0072	101000100	0860	BRANCH	ZRORAND	
0065	000001001	0861	RANCNT	MNEA	
004B	101100110	0878	0862	BRANCH	RANCOMP
0016	010001101	0863	LDX	5	
002D	010000101	0864	CALLL	RCOMX8	
0054	111001100	1631	0865		
0034	000101001	0866	TMA		
0068	001001111	0867	TCY	15	
0051	000000001	0868	ALEM		
0022	101100110	0878	0869	BRANCH	RANCOMP
0044	010000101	0870	ZRORAND	CALLL	RCOMX8
0008	111001100	1631	0871		
0011	001100000	0872	TCMIY	0	
0023	000000100	0873	DYN		
0046	010010010	0874	LDX	4	
000C	001100000	0875	TCMIY	0	
0019	001100000	0876	TCMIY	0	

TABLE IX-5 (Continued)

TABLE IX-6

65

					ORPGC	6
					CODE	BREAKER
					CALL	SPACE=3
0042	0043	0044	0045	0046	TXY	0
0892	0893	0894	0895	0896	LUX	0
					MNEZ	
					BRANCH	CHY2
					LDX	1
					IMAC	
					CPA12	
					HEIN	
					ALEC	9
					BRANCH	CHY3
					ACACC	6
					BRANCH	CHY6
					TAM	
					LDX	0
0047	0048	0049	004A	004B	CRY5	
004C	004D	004E	004F	004G	CRY6	
					TCIV	1
					YNEC	a
					BRANCH	CHY1
004B	004C	004D	004E	004F	SET MSH TO 1	
004F	0050	0051	0052	0053	ARE ALL LETTERS FINISHED?	
					NO, CONTINUE	

80

PAST 2 HAS BEEN
+ CREATED, ADD 6 TO GET A LETTER
RET
STORE COMPLEMENT OF LSD

SET MSR TO 1
ARE ALL LETTERS FINISHED?
NO, CONTINUE

* IF A CHARACTER CODE
* PAST 121 HAS BEEN
* CREATED, ADD 6 TO GET
* THE COMPLEMENT OF LS

TABLE IX-6 (Continued)

004F	010001101	0913	CRY12	HL	TONES	
001E	101000111	1657	0914	CALL	COMPL	
0050	010000110	0915	CRY2			* TEST FOR CODES OTHER * THAN LETTERS AND SKIP THEM
007A	110111111	0900	0916	ALEC	5	
0075	011101010	0917		RHANCH	CRY5	
0068	101111000	0909	0918	TAM		
0057	000101111	0919	CRY6			
002E	010010000	0920		IDX	0	
005C	001100900	0921		TCMIV	0	
0038	101110011	0911	0922	BRANCH	CRY4	
0070	010011000	0923	CLUE	LDX	3	
0061	001100001	0924		TCY	6	
0043	000101001	0925		TMA		GET HEX RANDOM NUMBER
0006	011101110	0926		ALEC	7	* IF NUMBER IS GREATER
0000	100110111	0929	0927	BRANCH	CLUE1	* THAN 7, ADD 8
001B	001110001	0928		ACACC	8	
0057	000101000	0929	CLUE1	TAY		SET Y RANDOMLY 0 = 7
006E	000100100	0930	CLUE2	DYN		* LOOK FOR FIRST
005D	101111000	0933	0931	BRANCH	YOK	
003A	001001110	0932		TCV	7	
0074	0100000100	0933	YOK	CALL	SPLNTR+1	* LETTER THAT HASN'T
0059	110101110	0374	0934			* BEEN CORRECTLY ENTERED
0053	011100000	0935		ALEC	0	YES
0026	101101110	0930	0936	RHANCH	CLUE2	NO
004C	0100110100	0937		LDX	2	* GET LSD OF LETTER
0018	000100000	0938		THIT	0	* FROM CORRECT SPELLING
0031	100006101	0952	0959	BRANCH	CLUES	* BUFFER AND PUT IT IN
0062	010011100	0940	GFTIT	LDX	3	* KEY CODE
0045	000101001	0941		TMA		
0004	010010000	0942		LUX	0	SET MSB#0
0015	001100111	0943		TCY	14	
002B	000101101	0944		TAMIV		
0056	010111111	0945		RETW		
002C	001100000	0946		TCMIV	0	
005A	001001011	0947	CLUE4	TCY	13	
0050	010010100	0948		LDX	2	
0060	000101001	0949		TMA		
0041	0100000100	0950		HL	MISS1	
0002	100110101	0471	0951			
0005	1111000910	0940	0952	CLUF3	CALL	GET IT
0003	001101000	0953		TCMIV	1	SET MSB#1
0017	101011000	0947	0954	RHANCH	CLUE4	HE1
002F	000101000	0955		TAY		

TABLE IX-6 (Continued)

					BRANCH	CRV12	
003C	1010011111	0913	0957		LDX	0	10
007A	0100100000		0958		TCY	5	
0071	0010010101		0959		TCMIV	1	
0063	0011010000		0960		TCMIV	2	
0047	0011001600		0961		TCMIV	2	
000E	0011001600		0962		TCMIV	2	
0010	0100110000		0963		LDX	1	
003A	0010010101		0964		TCY	5	
0076	0011001111		0965		TCMIV	14	
0060	0011001000		0966		TCMIV	1	
0058	0011000000		0967		TCMIV	0	
0036	1010011111	0913	0968		BRANCH	CRV12	
006C	0100110000		0969	F5	LDX	3	
0059	0010010111		0970		TCY	13	
0042	0011000000		0971		TCMIV	0	
0064	0100110000		0972	F2L00P	LDX	1	
0049	0010001011		0973		TCY	10	
0012	0011001000		0974		TCMIV	2	
0025	0011001010		0975		TCMIV	4	
004A	0011000000		0976		TCMIV	0	
0014	0011000000		0977		TCMIV	0	
0029	0100011010		0978		LDX	5	
0052	0010010111		0979		TCY	13	
0024	0001010001		0980		IMA		
004B	0000101011		0981		AMAAC		
0010	1010011011	1012	0982		BRANCH	HOF2	
0021	0100011000		0983		LUX	1	
0042	0010001011		0984		TCY	10	
0004	0000101011		0985		AMAAC		
0009	1010011010	1015	0986		BRANCH	HOF3	
0013	0010010111		0987		TAM		
0027	0101111111		0988		RETN		
004F	0100000101		0989		CALL		
001C	1110110000	1501	0990		CALL		
0039	0100011110		0991		LOAD ADDRESS		
0072	1110000110	1121	0992				
0065	0100110110		0993		LDX		
004H	0100000111		0994		CALL	TRANS=1	
0016	1101010011	1036	0995		BL	F4	
002D	0100010011		0996				

4,189,779

83

84

55

60

65

TABLE IX-6 (Continued)

005A	101001101	1422	0997			
0034	001000101		0998	FINL3	TCY	10
0068	010011000		0999	FINL6	LDX	1
0051	000101001		1000		TMA	
0022	010010010		1001		LDX	4
0044	000101101		1002		TAMIYC	
0008	001010111		1003		YNEC	14
0011	101101000	0999	1004		BRANCH	FINL6
0023	010001010		1005		CALL	CURLEVL
0046	111101111	0769	1006			
000C	001100010		1007		TCMIY	4
0019	001101110		1008		TCMIY	7
0033	010001000		1009		HL	SPR4
0066	101100101	0311	1010			
0040	010011000		1011			
001A	001000101		1012	NDF2	LDX	1
0035	000010101		1013		TCY	10
0064	000101101		1014		AMAAC	
0055	000110010		1015	NDF3	TAMIYC	
0024	000101111		1016		IMAC	
0054	100100111	0988	1017		TAM	
			1018		BRANCH	FINL2

TABLE IX-7

0000	001100000		1019		ORGPG	7
0001	001000101		1020	*	LOADED 10 VALUES ->STORE LAST VALUE	
0003	010011010		1021	RANSTOP	TCMIY	0
0007	000101001		1022		TCY	10
000F	001000111		1023		LDX	5
001F	000101111		1024		TMA	
003F	010010010		1025		TCY	14
007F	001000101		1026		TAM	
007E	000101001		1027		LDX	4
007D	001000111		1028		TCY	10
007B	000101111		1029		TMA	
0077	010011010		1030		TCY	14
006F	111110000	1052	1031		TAM	
005F	010010010		1032	RSCRAM2	LDX	5
003E	111110000	1052	1033		CALL	RSCRAM
007C	010001000		1034		LDX	4
0079	101110011	0215	1035		CALL	RSCRAM
			1036		BL	RANRTN
0073	001000111		1037		*	LDPREV--> LOADS NEXT VALUE INTO RWE
0067	010010010		1038			
004F	000101001		1039	LDPREV	TCY	14
001E	001000000		1040		LDX	4
003D	000101111		1041		TMA	
007A	001000111		1042		TCY	0
0075	010011010		1043		TAM	
006B	000101001		1044		TCY	14
0057	001000000		1045		LDX	5
002E	000101111		1046		TMA	
005C	010001010	0837	1047		TCY	0
0038	101011001	0837	1048		TAM	
			1049		LDP	5
0070	001000000		1050		BRANCH	RANOK2
0061	000101001		1051	*	SCRAMBLES RWE WORDS	
0043	001000110		1052	RSCRAM	TCY	0
0006	000000011		1053		TMA	
0000	001000000		1054		TCY	6
0018	000101101		1055		XMA	
0037	000101001		1056		TCY	0
			1057		TAMIYC	
			1058		TMA	

4,189,779

87

88

IX-7 (Continued)

100E	001001110	1059	TCY	7
1050	000000001	1060	XMA	
103A	001001000	1061	TCY	1
1074	000101101	1062	TAMIYC	
1069	000101001	1063	TMA	
1053	001001010	1064	TCY	5
1026	000000011	1065	XMA	
1040	001000100	1066	TCY	2
1018	000101101	1067	TAMIYC	
1031	000101001	1068	TMA	
1062	001000001	1069	TCY	8
1045	000000011	1070	XMA	
100A	001001100	1071	TCY	3
1015	000101101	1072	TAMIYC	
1028	000101001	1073	TMA	
1056	001001001	1074	TCY	9
102C	000000011	1075	XMA	
1058	001000010	1076	TCY	4
1030	000000011	1077	XMA	
1069	010111111	1078	RETN	

30

35

40

45

50

55

60

65

TABLE IX-7 (Continued)

```

1079   * OUTADDR2!          * OUTADDR2!           * OUTADDR2!
1080   * LOADS 4 BITS INTO K-LINES USING PDC AND OUTPUT 4 BITS    * CHIP SELECT
1081   * LOADS 4 BITS INTO K-LINES USING PDC AND OUTPUT 4 BITS    * L/R = 0
1082   * OUTADDR2 TCY      12      * ACC=OUTPUT! 4 BITS COMMAND
1083   * OUTADDR2 TCY      12      * ACC=OUTPUT! 4 BITS COMMAND
1084   * SETR               * SETR               * ACC=OUTPUT! 4 BITS COMMAND
1085   * SETR               * SETR               * ACC=OUTPUT! 4 BITS COMMAND
1086   * SETR               * SETR               * ACC=OUTPUT! 4 BITS COMMAND
1087   * SETR               * SETR               * ACC=OUTPUT! 4 BITS COMMAND
1088   * CLA                * CLA                * ACC=OUTPUT! 4 BITS COMMAND
1089   * ACACC              * ACACC              * ACC=OUTPUT! 4 BITS COMMAND
1090   * SETR               * SETR               * ACC=OUTPUT! 4 BITS COMMAND
1091   * RSTR               * RSTR               * ACC=OUTPUT! 4 BITS COMMAND
1092   * SETR               * SETR               * ACC=OUTPUT! 4 BITS COMMAND
1093   * RSTR               * RSTR               * ACC=OUTPUT! 4 BITS COMMAND
1094   * SETR               * SETR               * ACC=OUTPUT! 4 BITS COMMAND
1095   * RSTR               * RSTR               * ACC=OUTPUT! 4 BITS COMMAND
1096   * SETR               * SETR               * ACC=OUTPUT! 4 BITS COMMAND
1097   * RSTR               * RSTR               * ACC=OUTPUT! 4 BITS COMMAND
1098   * CLA                * CLA                * ACC=OUTPUT! 4 BITS COMMAND
1099   * ACACC              * ACACC              * ACC=OUTPUT! 4 BITS COMMAND
1100   * SETR               * SETR               * ACC=OUTPUT! 4 BITS COMMAND
1101   * RSTR               * RSTR               * ACC=OUTPUT! 4 BITS COMMAND
1102   * SETR               * SETR               * ACC=OUTPUT! 4 BITS COMMAND
1103   * RSTR               * RSTR               * ACC=OUTPUT! 4 BITS COMMAND
1104   * SETR               * SETR               * ACC=OUTPUT! 4 BITS COMMAND
1105   * RSTR               * RSTR               * ACC=OUTPUT! 4 BITS COMMAND
1106   * SETR               * SETR               * ACC=OUTPUT! 4 BITS COMMAND
1107   * ACACC              * ACACC              * ACC=OUTPUT! 4 BITS COMMAND
1108   * SETR               * SETR               * ACC=OUTPUT! 4 BITS COMMAND
1109   * RSTR               * RSTR               * ACC=OUTPUT! 4 BITS COMMAND
1110   * SETR               * SETR               * ACC=OUTPUT! 4 BITS COMMAND
1111   * TKA                * TKA                * LOAD INTO ACC
1112   * SETR               * SETR               * 3RD PDC DISCONNECTS SR
1113   * LDX                * LDX                * BRANCH
1114   * TBIT               * TBIT               * LSHIFT=1
1115   * BRANCH             * BRANCH             * END OF OUTADDR2 SUBROUTINE
1116   * RETN               * RETN               *
1117   * END OF OUTADDR2 SUBROUTINE
1118   * END OF OUTADDR2 SUBROUTINE
1119   * END OF OUTADDR2 SUBROUTINE
1120   * END OF OUTADDR2 SUBROUTINE

```

TABLE IX-7 (Continued)

		LOAD ADDRESS	TCY	11	
		1121	LDX	2	
		1122	SBIT	3	
0042	00100101101	1123	TCY	10	
0044	010010100	1124	CLA	3	
0049	010100011	1125	ACACC	3	
0053	0010000101	1126	LDX	2	MEMORY FOR LOOP
0057	000000110	1127	LOADR+1	1	
004E	010111100	1128	TAMZA		
001C	010010100	1129	LDX	1	
0039	000101110	1130	BRANCH		
0072	010011060	1131	LSHIFT#1	13	SHIFT ROUTINE
0065	101000501	1132	LDX	1	
0048	0010001011	1133	XMA		
0016	010011060	1134	DYN		
0020	0000000011	1135	YNEC	9	
005A	000000100	1136	BRANCH		
0034	001011001	1137	LSHIFT	10	TEST LOOP COUNT
0068	1001151161	1138	TCY	1	
0051	001000101	1139	LDX	2	
0022	010010100	1140	BRANCH		
0044	0000000111	1141	LOADR+1	11	
0008	100111061	1142	TCY	11	
0011	001001101	1143	RBIT	3	
0023	010106111	1144	REIN		
0046	010111111	1145	*		
000C	010011010	1146	LDX	5	
0019	001001011	1147	TCY	13	
0033	000000110	1148	CLA		
0066	00111101	1149	ACACC	10	
0040	000000011	1150	SAMAN		
001A	000110000	1151	TAM		
0035	000101111	1152	REIN		
0064	010111111	1153	ROM	6	
0055	001000110	1154	LDX	8	
002A	010010001	1155	TMA		
0054	000101001	1156	ACACCL	6	
0028	001110001	1157	TAM		
0050	000101111	1158	HI		DISP/RH
0020	010001111	2219	1159		
0040	100101160				

TABLE IX-8

	URGPGC	8	
1160			
1161	* CALADDR--> STICKS ADDRESS WANTED INTO LINK/FDT		
1162			
1163			
0000 001000111	CALADDN	TCY 14	
0001 00100101	1164	TCMIV 10	
0003 001001001	1165	TCV 9	
0007 010011000	1166	LDX 1	
000F 000101001	1167	TMA	
001F 001111111	1168	ACACC 15	
003F 001010010	1169	COMXB	
007F 000101011	1170	TAM	
007E 001000101	1171	TCV 10	
0070 000101001	1172	TMA	
0074 010011000	1173	TMAA	
0077 000101101	1174	1	
006F 010110010	1175	TAMIYC	
005F 001010111	1176	CUMXA	
003E 101111101	1177	YNEC 14	
007C 111001111	1178	BRANCH TMAA	
0079 010011110	1179	CALL+2 CALL CAL+1	
0073 000101110	1180	7	
0067 010110010	1181	TAM	
004F 001000111	1182	CUMXA	
001E 000101010	1183	CAL+1 TCY 14	
003D 000101111	1184	IMAC	
007A 000101010	1185	TAM	
0075 000100100	1186	THV	
006A 000101001	1187	DYN	
0057 001001001	1188	TMA	
002E 000101010	1189	OUTSRVN TCY 9	
005C 010110010	1190	TCV	
003E 010111111	1191	COMXB	
0070 010010110	1192	KETIN	
0061 000101111	1193	LDX 6	
0043 010110010	1194	TAM	
0006 001001001	1195	CUMXA	
0009 000101010	1196	TCY 9	
0014 000101111	1197	IMAC	
0037 001001111	1198	TAM	
006F 000101010	1199	TCV	
0050 000101011	1200	THV	
003A 100111110	1179	YNEC 14	
	1201	BRANCH CALL+2	
	1202	LOAD 2 MSW	

4,189,779

93

94

60

65

55

TABLE IX-8 (Continued)

0074	001000100	1203	ICY	2
0069	010100101	1204	RBT	2
0053	010011000	1205	LDX	1
0026	001001001	1206	ICY	9
004C	010000011	1207	BL	LNRN#12
0018	101010100	1798	1208	
0031	010011100	1209	ULRN#1	3
0062	001001011	1210	ICMIV	13
0045	001001010	1211	ICMIV	5
000A	010001010	1212	BL	CORR#1
0015	101111110	1213		
		1214		
		1215		
0024	001100000	1216	DISLP#1	ICMIV
0056	010000101	217	AL	LOADDISP
002C	101111001	1456	218	
0058	010000111	1219	DISLP7	CALLL
0050	110000001	2010	1220	SPEAK#1
0060	010000011	1221		TRANS=1
0041	110100011	1836	1222	CALLL
0002	010010100	1223	DISLP#2	LDX
0005	031001111	1224		2
0094	001101010	1225	ICMIV	15
0017	010001010	1226	CALLL	5
002F	111011111	0769	1227	CURRLVL
005E	001100111	1228	ICMIV	14
003C	001100110	1229	ICMIV	6
0074	010000010	1230	HL	ADDLRL
0071	101011001	076A		
0063	001001111	1232	DISLP#5	ICY
0047	010110010	1233	CUMX8	15
0068	001101111	1234	ICMIV	15
0010	010110001	1235	DISP#00P	CUMX8
0034	001000111	1236	TCY	14
0076	010011100	1237	10X	3
0060	010101000	1238	SBRJ	0
0053	010101111	1239	HL	DISP/KH
0036	100101100	2219	1240	
006C	010110016	1241	DISLP#1	CUMX8
0059	001010111	1242	ICY	15
0032	101000111	1243	DISP#1	ADDRESS DAM
				Loop
				#

60

65

TABLE IX-8 (Continued)

		TAM	MNEZ	BRANCH	DISPLOOP	* ELSE
0064	000101111	1244				
0049	000110011	1245				
0012	100111011	1235	1246			
0025	001000111	1247	TCY	14		
004A	010001100	1248	LDX	5		
0014	010100100	1249	RHTT	0		
0029	010010001	1250	LOX	8		
0052	010001001	1251	LDP	9		
0024	001000100	1252	TCY	2		
0048	000100011	1253	THIT	3		
0010	101010011	1341	BRANCH	LETA		
0021	000100001	1254	THIT	2		
0042	101001010	1385	1256	BRANCH	REFST02	
0004	010000001	1257	LDP	6		
0009	010010100	1258	LDX	2		
0013	001001111	1259	TCY	15		
0027	000101001	1260	TMA			
004E	011101001	1261	ALFC	9		
001C	101110010	1264	BRANCH	DISP8		
0039	101101000	1271	1263	BKANCH	DISPS	
0072	001100101	1264	DISP8	TCMIV	10	
0065	010011000	1265	LDX	1		
0048	010110010	1266	CUMXA			
0016	010000001	1267	DISP9	CALLL	TRANS-1	
0020	110100011	1436	1268	BL	AUDWUS2	
0054	010000011	1269				
0034	110001010	2057	1270	DISP5	CALLL	DELAY2
0068	010001001	1271				
0051	110100111	1396	1272	CUMXA		
0022	010110010	1273	TCY	0		
0044	001000000	1274				
0008	000110010	1275	IMAC			
0011	000101101	1276	TAMIVC	0		
0023	001100000	1277	TCMIV			
0046	011101001	1278	ALEC	9		
000C	101010000	1291	BRANCH	DISP6		
0019	001000000	1280	TCY	0		
0033	001100000	1281	TCMIV	0		
0066	001100000	1282	TCMIV			
004D	010011010	1283	LDX	5		
001A	010001110	1284	CALLL	RSCHAM		
0035	111110000	1452	1285			

4,189,779

97

98

TABLE IX-8 (Continued)

006A	010010010	1286	LDX	4
0055	010011110	1287	RSCKAM	
0024	111110000	1052	CALL	
0054	010000100	1289	H1.	
0024	100000111	0345	1290	0 ISPELL+1
0050	010001001	1291	DISP6	DELAY2
0020	110100111	1398	1292	ULRN+1
0040	100110001	1209	1293	BRANCH

TABLE IX-9

1294	ORGPG	9		
1295	*			
1296	*			
1297	*			
1298	*			
1299	LETTER	TCY	15	
		CLA		
0060	001001111	1300	CALL	
0061	000000110	1301	CALL	RETURN4
0063	010000111	2113	1302	CLEAR
0007	110000100	1303	1304	
000F	010001000	0236	1305	TCY
001F	110111010	0236	1306	COMX8
003F	001001000	1307	1307	
007F	010110010	1308	1308	TCMIV
007E	001100000	1309	1309	15
0079	001001111	1310	1310	TCIV
007A	001101000	1311	1311	TCIV
0077	010011100	1312	1312	LOAD LSW >>ACC
006F	001001000	1313	1313	
005F	010001101	1632	1314	
003E	110011000	1315	1315	
007C	006101001	1316	1316	THA
0079	010011110	1317	1317	LDX
0075	001000000	1318	1318	TCY
0067	000101111	1319	1319	1AM
004F	010010100	1316	1316	LDX
001E	001001000	1319	1319	TCY
0030	010001101	1320	1320	COMX8
				* STORE IN LNK/EDT
				* GET V POINTER
				* COMX8

TABLE IX-9 (Continued)

007A	110011060	1632	1321			LOAD MSW
0075	000101061	1322		TMA	10	
0068	010009101	1323		LUP	2	LAST LETTER?
0057	000100001	1324		TBIT	2	YES, SETBITS
002E	111010011	1485	1325	CALL	SETBITS	
005C	010001111	1326		LDP	15	
0034	000100011	1327		TRIT	3	SYLLABLE?
0070	111016101	2291	1328	CALL	SETBITS	SET SYLLABLE FLAG
0061	001000000	1329		TCY	0	*
0043	010010110		1330	LDX	6	*
0006	000101111		1331	LDP	H	*
0000	010100101		1332	TAM		
001A	010100111		1333	RBIT	2	
0057	001000160		1334	RBIT	3	
006E	010010061		1335	CALCULATE ADDRESS OF LETTER		
005D	010000001		1336	TCY	2	FLAG WORD
005A	000100001	1235		LDX	8	
0074	+100011101		1337	LDP	H	
0069	010001001		1338	TBIT	3	SYLLABLE?
0053	001000060		1339	BRANCH	DISPLDP	
0026	010001010		1340	LDP	9	
004C	000101001		1341	LET4	TCY	
0018	000010101		1342	LDX	6	
0031	000101111		1343	IMA		MULTIPLY BY 2
0062	010011110		1344	AMAAC		
0045	000101001		1345	TAM		
0004	000100101		1346	LDX	7	
0015	111000010	1394	1347	TMA		
002B	000101111		1348	AMAAC		
0050	010011110		1349	CALL	TLETTER	CARRY, GO TO TLETTER
002C	000101001		1350	TAM		
0058	001110011		1351	LDX	7	
0030	111000010	1394	1352	IMA		
0060	000101111		1353	ALACC	12	
0041	010000111		1354	CALL	TLETTER	
0092	110000001	2010	1355	TAM		
0005	010011100		1356	LOADS LETTER ADDRESS FROM ADOR AREA (RAM)		
0004	001000101			CAL11	SPEAK+1	
0017	001100011			LUX	2	FLAG
002F	010001000					

TABLE IX-9 (Continued)

005E	0011001111	1363	TCY	15
003C	0011001100	1364	TCM1Y	6
007A	0010010000	1365	TCY	1
0071	0010001101	1366	CALLL	COMX8
0063	0110011000	1362	1367	
0047	0100001011	1368	CALLL	DPLLOAD
000E	1111100111	1457	1369	
0010	0100000010	1370	BL	ADDCTR6
0034	0101011001	0704	1371	
0076	0010001000	1372	LET#4	TCY
0060	0101100100	1373	* SPEAKS LETTER	
005R	0101001111	1374	COMX8	
0036	0001000001	1375	RBT	3
006C	1000010010	1376	TBT	2
0059	0001001000	1365	1377	BRANCH
0052	0001100100	1378	TCY	HESTO
0064	0001011111	1379	IMAC	
0049	1011101111	1310	1360	TAM
0012	0100000001	1381	BRANCH	LETTER#1
0025	1011100011	1382	* RESTORE LNK/EDT POINTER AND RETURN TO CONTINUE SPEAKING	
004A	0101000101	1232	1383	RES10 HL
0014	0100010100	1384	REST02	RBT
0029	0010011111	1385	1346	LDX
0052	0011011000	1387	TCY	2
0024	0010001000	1388	TCM1Y	15
0048	0101100100	1389	TCY	3
0010	0100000101	1390	COMX8	
0021	1000000011	1439	1391	HL
0042	0001011111	1592	REPT2	
0004	0100010110	1393	* INCREMENT WHEN OVERFLOW OCCURS	
0009	0001100100	1394	LETTER TAM	
0013	0101111111	1395	LDX	6
0027	0000001100	1396	IMAC	
004E	0100001000	1397	HEIN	
001C	0010000001	1398	DELAY2	
0059	0011000000	1399	CLAD	
0072	0011000000	1400	LDX	2
0065	0011000000	1401	TCY	*
004B	0010000001	1402	TCM1Y	0
		1403	TCM1Y	6
		1404	TCY	A

4,189,779

103

104

60

65

TABLE IX-9 (Continued)

TABLE IX-10

1432	ORG#6							
1433	*	REPEAT ROUTINE	REPEATS PHRASE PREVIOUSLY SPOKEN					
1434	*	REPEAT ROUTINE	REPEATS PHRASE PREVIOUSLY SPOKEN					
1435	*	TWO REPEATS	OR MORE CAUSES PHRASE TO BE SPOKEN SLOWER					
1436	*							
0000	0100010160	1437	REPEAT	LDX	2			
0001	001001111	1438		TCY	15			
0003	001100000	1439	REPT2	TCMIV	0			
0007	010011000	1440		LOX	1			
000F	001000101	1441	RPT#1	TCV	10			
001F	010110010		COMXA					
003F	000101001	1443		IMA				
007F	010110101	1444		COMXA				
007E	000101101	1445		TAPIYC				
007D	001010111	1446		YNEC	14			
007B	100011111	1447	RANCH	RPT#1				
0077	010110010	1448		COMAR				
006F	001000100	1449		TCY	1			
005F	001100000	1450		TCMIV	0			
003E	010001111	1451		BL	ADDWSS2			
007C	100001010	2057						
1452	*	LOADDISP	TO DISPLAY WORD BEING USED IN LEARN MODE					
1453	*	LOADDISP	TO DISPLAY WORD BEING USED IN LEARN MODE					
1454	*	SUBROUTINE	TO DISPLAY WORD BEING USED IN LEARN MODE					
1455	*	LOADDISP	TCY	0				
1456	001000000	1457	OPLOAD	LOX	3			
0079	001001100	1458		THA				
0073	010001100	1459		LDX	1			
0067	000101001	1460		TAM				
004F	010011000	1461		LOX	2			
001E	000101111	1462		THA				
0030	010010100	1463		LOX	0			
007A	000101001	1464		TAM				
0075	010010000	1465		RETN				
006A	009101111	1466		TBIT	0			
0057	010111111	1467		BRANCH	LDONE			
002E	000100000	1470		TCMIV	0			
005C	101100001	1471		BRANCH	LDONE#1			
003A	001100000	1472		TCMIV	1			
0070	101000001	1473		BRANCH	LDONE#1			
0061	001101000	1474		TCMIV	0			
0043	001010000	1475		YNEC	8			
0006	101110011	1476		BRANCH	OPLOAD			
000D	010010001	1477		LDX	8			

TABLE IX-10 (Continued)

001B	0010011110		1474	TCY	7		
0037	00010101010		1475	TMY			
006E	0100000001		1476	LOP	8		
005D	00101101016		1477	VNEC	5		
003A	1010110000	1219	1478	BRANCH		DISLP7	
0074	01000000010		1479	BL		ADDCTR6	
0069	1010110001	0704	1480				
			1481	*			
			1482	*			
			1483	*			
			1484	*			DAM REG
0053	0101110010		1485	SETHIT2	COMX8		
0026	0010000100		1486	TCY	2		
004C	0101000001		1487	SBIT	2	TEST BIT 2	
0018	0010001000		1488	TCY	1		
0031	0001001010		1489	TMY			
0062	0101100010		1490	COMX8			
0045	0101111111		1491	RETN			
			1492	*			EXIT DAM
000A	0101100101		1493	SETHIT1	COMX8		
0015	0010000100		1494	TCY	2		
0028	0101000010		1495	SBIT	1		
0056	0101100010		1496	COMX8			
002C	0101111111		1497	RETN			
			1498	*			
1499	*	MEMLOOP= LOADS ADDRESS INTO ROM ADDRESS, 4 BITS AT A TIME					
1500	*	MEMADR TCY	12			CHIP SELECT	
005A	0010000011		1501	MEMADR SETH			
0050	000001101		1502	TCY	1	L/R = 1 (INPUT)	
0060	001001101		1503	SETR		R11 = 1	
0041	0000001101		1504				
0002	0010000101		1505	TCY	10		
0005	0000000110		1506	CLA			
000R	0011111100		1507	ACACC	3		
0017	0100010100		1508	LDX	2		
002F	000101110		1509	MEMLOOP	TAMZA		
005E	010011000		1510	LDX	1		
003C	001110100		1511	ACACC	1		
0078	0000001101		1512	SETR		LOADS COMMAND	
0071	0001110110		1513			* RS11	
0063	0001000101		1514	TMA		4 BITS OF ADDR => ACC	
0047	001110000		1515	ACACC	0		

4,189,779

109

110

	LOADS DATA	SETR	RSTH	TCY	13	SHIFT ROUTINE	SHIFT UP IN	SAME REGISTER	ORIGINAL WORD	REG 6	MEM1 => ACC LOOP
000E	0000061101	1516									
001D	0001101110	1517									
003H	0010010111	1518	SHIFTUP	XMA		DYN	9				
0076	0000060111	1519				YNEC					
006D	0000061000	1520				BRANCH	SHIFTUP				
005H	001611001	1521				TCY	10				
0036	1011101110	1519	1522			LDX	?				
006C	001090101	1523				DMAN					
0059	0100101010	1524				BRANCH	MEM1 => ACC	LOOP			
0032	000500111	1525									
0064	100101111	1509	1526								
0049	000101111	1527	TAM								
0012	001111100	1528	ACACC	3							
0025	000001101	1529	SETR								
0044	000110110	1530	RSTR								
0014	000000110	1531	CLA								
0029	000001101	1532	SETR								
0052	000110110	1533	RSTR								
0024	010011000	1534	MEMDRED	LDX	ONE						
0048	001000101	1535	TCY	TEN							
0010	001110001	1536	ACACC	EIGHT							
0021	000001101	1537	SETR								
0042	000110110	1538	RSTR								
0004	010111111	1539	RETN								
0009	010001000	1540	WSPELS	CALL	CLEAR						
0013	110111010	0236	1541	CALLL	DELAY2						
0027	010001001	1542									
004E	110100111	1398	1543	BRANCH	REPEAT						
001C	100600000	1437	1544								
0039	010111111	1545	* SPELLING IS INCORRECT								
0072	010110010	1546	MISSPELL RETN								
0065	001000110	1547	CUMKS								
0048	010001000	1548	TCY	6							
0016	000100010	1549	LOP	1							
0020	100001001	0304	1550	TBIT	1						
005A	010100010	1551	BRANCH	MOPHRASE							
		1552	SBIT	1							
		1553	* LOAD NEGATIVE RESPONSE INTO L/E								
		1554	SCMF	10							
0034	010011010	1555	TCY	13							
0068	001001011	1556	IMAC								
0051	000100101	1557	1AM								
0022	000101111										

TABLE IX-10 (Continued)

			LOX	13	5	FLAG
0044	0100111100		1554			
0008	0010010101		1559			
0011	0011001000	1560		TCMIV	2	CURLEVEL
0023	0100010100	1561		CALL		
0046	1111011111	0769	1562	TCMIV		
000C	0000000101		1563	TCMIV	6	
0019	0011001010		1564	TCMIV		
0033	0100010100		1565	LDX	2	
0066	0010001111		1566	TCY	15	
004D	0011000100		1567	TCMIV	4	
001A	0100010000		1568	BL		SPK4
0035	1011001011	0311	1569			
006A	0011000000		1570	ADDCTR2	TCMIV	
0055	0100111009		1571	LDX	3	
002A	0010001011		1572	TCY	13	
0054	0011000100		1573	TCMIV	4	
0028	0100001001		1574	BL		CORR+1
0050	1011111110		1590	1575		
			1576	*		

FOR RETNSBCH
*
*
*

TABLE IX-11

		ORGPG	11			
1577						
1578	*	POINTERS DAM=WORD 0*** RANDOM WORD ENTRY POINTER				
1579	*	POINTER DAM=WORD 1*** CORRECR SPELLING BUFFER POINTER				
1580	*	POINTER DAM=WORD 1*** CORRECR SPELLING BUFFER POINTER		DAM REG=POINTER		
1581	*			ZEROS OUT POINTER		
				OUT OF DAM REG		
0000	0101110010	1582	CORR\$PL CONXB			
0001	0010000000	1583	TCY	0		
0003	0011000000	1584	TCMIV	0		
0007	0011000000	1585	TCMIV	0		
000F	0011000000	1586	TCMIV	0		
001F	0011000000	1587	TCMIV			
003F	0101100100	1588	COMXB			
007F	0101111111	1589	RETN			
007E	0100010100	1590	CORR+1	CALL		
007D	1111011111	0769	1591			
007E	0011001111		1592	TCY		
0077	0001001001		1593	TNA		
006F	0000010101		1594	ANAAC		

TABLE IX-11 (Continued)

005F	001110010	1595		ACACC	4				
003E	001000191	1596		1CY	10				
007C	000101111	1597		1AM					
0079	010000111	1598		CALLE					
0073	110001100	2139	1599						
0067	010000101	1600		CALL					
004F	110110000	1501	1601						
001E	010001110	1602		CALL					
0030	111000010	1121	1603						
		1604	*						
		1605	*	RESIDENT?					
		1606	*	LOOP TO TRANSFER ADDRESS FROM RESIDENT (RAM) TO ADDRESS REGION (RAM)					
		1607	*						
007A	001001110	1608		RESIDENT	TCY	7			
0075	000000110	1610	CSB2	CA					
0068	001110000	1611		ACACC	1				
0057	010011000	1612		LDX	2				
002E	000101111	1613		TAM					
005C	010001100	1614		LDX	3				
0038	001110101	1615		ACACC	10				
0070	000100100	1616		TAM/DYN					
0061	101110101	1617		BRANCH					
0043	010011000	1618		LDX	1				
0006	001000001	1619		1CY	8				
000D	001110000	1620		TCY	2				
0018	010011010	1621	ADRSCLC	LDX	5				
0037	111001100	1631	1622	CALL					
006F	000101001	1623	ADD2ROM	TMA					
0050	010011000	1624		LDX	1				
003A	001000001	1625		1CY	10				
0074	010000000	1626		CALL					
0069	111011000	0112	1627						
0053	010010010	1628		LDX	4				
0026	001000000	1629		1CY	0				
		1630	*						
004C	001000000	1631	ACOMX8	1CY	0				
001A	010110010	1632	COM8	COM8					
0031	000101010	1633		1CY					
0062	010110110	1634		LDX					
0045	010111111	1635		RTIN					
000A	000101001	1636		1MA					

4,189,779

115

116

TABLE IX-11 (Continued)

ROM ADDR REGION				LOX	TCY	ICY	H
0015	0100011000	1637					1
002H	031001101	1638					11
0056	050000000	1639					11
002C	111011000	0112	1640	CALL	CALL	CALL	ADDCARRY
005A	061000001		1641				*
0030	080000111		1642	UMAN			ADDCROM TO BE EXECUTED TWICE
0060	080010111		1643	TAN			*
0041	090110011		1644	MNEZ			
0002	100011011	1621	1645	BRANCH	ADHSACLC		
0005	010000101		1646	MEMADDR	MEMADDR		
009B	111011000	1501	1647	CALL	CALL		
0017	010001110		1648	LOADRESS	LOADRESS		
002F	111000010	1121	1649	CALL	CALL		
005E	010000101		1650	MEMADDR	MEMADDR		
003C	111011000	1501	1651	CALL	CALL		
			1652				
0078	010000011		1653	BL	OUTADDR		
0071	100000000	1723	1654				
			1655	*			
0065	000101111		1656	TUNE22	TAM	TCMIV	
0047	010001010		1657	TONES	CALL	TCMIV	*
006E	111011111	6769	1658				
001D	001100001		1659				
0034	061101110		1660				
0076	010011100		1661				
0060	001000001		1662				
005H	010100100		1663				
0036	010101111		1664				
006C	060101001		1665				
0059	010011100		1666				
0032	001000101		1667				
0064	000010101		1668				
0049	100101000	1676	1669				
0012	060610111		1670	TONE3	TAM	TONCARRY	
0025	010010100		1671				
004A	061001111		1672				
0014	061101000		1673				
0029	010000910		1674				
0052	191011001	0704	1675				
0024	069101101		1676				
0048	000110101		1677				
0010	000101111		1678				

TABLE IX-11 (Continued)

				TONE3
0021	10001010	1670	1679	BRANCH TONE
0042	00100110		1680	1CY LDX
0004	01001001		1681	THY YNEC
0009	000101010		1682	BRANCH CRY24
0013	001011010	1669	1683	BRANCH CRY24
0027	101101010		1684	LDX 15
0045	010010106		1685	LDX 15
001C	001001111		1686	1CY 15
0059	001101110		1687	1CY 7
0072	101001011	1692	1688	BRANCH TONESCOR
0065	001100000		1689	CRY24 FCMIV 0
		1690	1691	RETURN TO ROUTINE
			1692	TONESCOR LUX 6
			1693	1CY 6
			1694	BIT 2
		1698	1695	BRANCH TON12
		1699	1696	DISP/KB
		2212	1697	BL
			1698	TON12 LDX 2
			1699	1CY 14
			1700	BRANCH
		1656	1701	BRANCH
			1702	LDX
			1703	TCY
			1704	BIT 2
			1705	LUX 5
			1706	TCY 15
			1707	THA
			1708	LDX 1
			1709	LDX 6
			1710	ALEC 9
		0019	001001011	BRANCH FS
		0033	000100001	CURLEV
		0066	010010000	CALL
		0046	01001001	
		0000C	010011010	
		001A	011101001	
		0055	101101000	
		0064	010000100	
		0055	111101111	
		002A	001100110	
		0054	001101110	
		0028	010000010	
		0050	101011001	
			0704	WPG 12
			1714	
			1715	
			1716	
			1717	
			1718	
			1719	
			1720	
			1721	
			1722	
			1723	
				OUTADDR LOADS CORRECT SPELLING BUFFER WITH ACTUAL SPELLING CODE
				OUTADDR CALLI.
				OUTADDR CALLI.
				OUTADDR CALLI.

TABLE IX-12

00001	1110000001	1083	1724	I.DX	3
00003	010001100	1725		TCY	1
00007	001001000	1726		CALL	COMX8
0000F	0110001101	1727			*
001F	110011000	1632	1728	TAM	PDC FOR OUTPUT COMMAND
003F	000010111	1729		CALL	
007F	0100001100	1730		LDX	
007E	1110000001	1085	1731	TCY	
0070	010001000	1732		CALL	
0078	0010001000	1733		LDX	
0077	0100001001	1734		TCY	
006F	110011000	1632	1735	CALL	COMX8
005F	0100001001	1736		CALL	
003F	000010111	1737		LDP	10
007C	0000100001	1738		TAM	
0079	1100010010	1493	1739	TBIT	2
0073	0100000001	1740		CALL	SETBIT1
0057	010011000	1741		LDP	12
004F	0010010000	1742		TCY	1
001E	0000100000	1743		IMAC	
003D	000010111	1744		TAM	
0074	0010001000	1745		TCY	
0075	0000100000	1746		TBIT	1
0069	100111000	1751	1747	BRANCH	LINKSET
0057	1000101110	1749	1748	BRANCH	EXDAM2
002E	0100110010	1749		COMX8	
005C	1000000000	1724	1750	BRANCH	OUTADDR
003A	0000000000	1724		CLA	
0070	0010001001	1751		TCY	9
0061	0000010001	1752		LDX	
0043	000010111	1753		CALL	
0006	000001100	1754	1754	LDX	OUTADDR2
0000	1110000001	1085	1755	TCY	PDC FOR OUTPUT 4 BITS
0016	0100001001	1756		CALL	
0037	0111000000	1757		LDP	10
005F	1110000001	1485	1759	ALEC	0
0056	0000000001	1760		CALL	SETHIT2
003A	0011000000	1761		LDP	12
0074	010001100	1760		ALEC	0
0069	0100000001	1763		BRANCH	LNKON
0053	011101000	1764		LDP	10
0026	110001010	1493	1765	ALEC	1
004C	0000000001	1766		CALL	SETHIT1
0018	11101110	1785	1767	CALL	LNKPTH2
0051	010000110				OUTADDR2

4,189,779

123

124

	111000001	1083	1769	
0062	0100000010	1770	LDP	4
0045	001111111	1771	ACACC	15
0064	010011091	1772	CALL	1\$RE112
0015	001111000	1773	ACACC	1
0024	001111000	1774	CALL	LINKPTR
0056	010000111	1775		
002C	111101000	2130		
0058	000110010	1776	IMAC	
0030	000101111	1777	TAN	
0060	010001110	1778	CALLL	
0041	111000001	1083	OUTADDR2	
0002	0100000010	1780	LDP	4
0005	001111111	1781	ACACC	15
0006	110011001	0745	CALL	1\$THIT2
0017	0100000011	1783	LDP	12
0024	001111000	1784	ACACC	1
005F	010011000	1785	LINKPTR2	LINK
003C	0011001001	1786	TCY	9
0078	000101010	1787	TCY	8
0071	010011110	1788	LINK	
0063	010101111	1789	TAN	
0047	0011000101	1790	TCY	10
0001	010111111	1791	RETN	
0010	0100091110	1792	CALLL	
0038	1110000011	1083	OUTADDR2	
0076	011100000	1793		
0060	111100000	1794	ALFC	0
0058	010000111	1801	BRANCH	LINKEND
0036	111101000	2130	LINKT	LINKPTR
006C	0001100010	1796	LINKNT2	IMAC
0055	1111000010	1815	BRANCH	ENDSPEL
0032	1101000011	1750	1800	BRANCH
0064	0101100101	1801	LINKEND	LINK\$ET+1
0049	0101000100	1802	CURR8	
0012	0001000100	1803	TCY	2
0025	100000100	1814	TBLT	1
0044	000100001	1804	BRANCH	ENDSPEL
0014	111100001	1805	TBLT	2
0029	101011011	1796	1806	BRANCH
0052	000100000	1807	LINK4	LINK4
0024	100010001	1833	1808	BRANCH
0049	010100000	1809	SHIT	F9
0010	010000001	1810	SHIT	0
0021	100000000	1811	AL	CALADDR
0042	010110010	1812		
		1813	ENDSPEL	COMM8
				ADDRESS DNM

55

60

65

TABLE IX-12 (Continued)

0604 001000100 1814 ENDSPFL1 TCY 2
 0009 001100000 1815 TGMIV 0
 0013 010011100 1816 LDX 3
 0027 001100101 1817 TCY 13
 0045 001110101 1818 THA
 0015 010001110 1819 LDP 14
 0016 010001110 1820 ALEC 3
 0039 011101100 1821 BRANCH
 0072 100000000 2009 SPEAK
 0065 010000011 1822 LDP 12
 0043 011100100 1823 ALEC 4
 0015 100101010 1847 USPELL3
 0020 010000001 1824 BRANCH
 0054 011101010 1825 LDP 8
 0034 100101011 1826 ALFC 5
 0064 010001100 1827 BRANCH
 0051 011100111 1828 LIP 3
 0022 101001111 0501 1829 ALFC 14
 0044 010001111 2219 1830 HANG
 0008 100101110 1831 BRANCH
 0011 010001010 1832 HL
 1853 F9 IDX
 1834 * TRANS-STORES CALCULATED ADDRESS IN DAM FOR
 1835 *
 0025 011000101 1836 TRANS-1 ICY 10
 0046 000016101 1837 TRANS THA
 000C 010111010 1838 CDRXH
 0019 000101010 1839 TMRYC
 0052 010110101 1840 COMRH
 0068 001010111 1841 YNEC 14
 0040 101000110 1837 1842 BRANCH
 0014 010111111 1843 HL
 0055 010110010 1844 CALL
 006A 001010111 1845 CALADDR
 0055 100000000 1164 1846 LOAD ACC
 0024 010000011 1847 USPELL3 CALL SPEAK+1
 0054 110000001 2010 1848
 0025 010000001 1849
 0050 111100011 1836 1850
 0020 010000111 1851 HL
 0040 100000000 2009 1852

4,189,779

125

126

1853 ORG#6 15

1854 *
 1655 *
 1856 *
 1857 * THE FOLLOWING ROUTINE DIRECTS THE PROGRAM FLOW ACCORDING TO THE
 KEY PRESSED.
 1858 *
 1859 *

0000 001000010	1860 KEY00	1861 KEY1
0001 00111011	KEY0	BRANCH KEY2
0003 010001000	1862	LDX H
0007 010000001	1863	TCV A
000F 000100010	1864	TAIT 1
001F 101011111	1875	BRANCH TRANSFER
003F 001001110	1865	TCV 7
007F 000100001	1866	THIT 2
007E 101011111	1875	BRANCH TRANSFER
0070 011101100	1870	ALFC 3
007B 101100001	1892	KEY12
0077 010000001	1871	BRANCH DIFFSLV
006F 101111110	1872	KEY13
0069 011111110	1873	BRANCH NOTRANS
005F 001001110	1870	BRANCH TRANSFER
003F 010010001	1875	TCV 7
007C 000101010	1876	LDX H
0079 001001101	1877	TMY
0073 100011110	1882	YHFC 5
0067 010000001	1880	BRANCH TRANS
004F 100101110	1880	ALFC 6
001E 001001111	1881	TRANS3 TCV 15
003D 010010000	1882	LDX 0
007A 000101001	1883	TCV
0075 001001101	1884	TMA
0068 000101010	1885	TCV 11
0057 010001000	1886	TMY
002F 001001001	1887	LDP 1
005C 100000000	0194	TRANS NOTFULL
0038 010001011	1889	BRANCH LDP 13
0070 100010100	1946	1891 BRANCH NOP
0061 001001111	1892	KEY12 TCV 15
0043 010011000	1893	LDX 1
0006 000101111	1894	TAN
000D 101101011	1872	BRANCH KEY13
0018 011100101	1896	ALFC 10
0037 101000111	1950	KEY1 BRANCH

4,189,779

127

128

TABLE IX-13 (Continued)

006E	011100111	1998	ALEC	14
005D	101101001	1902	KEY7	BRANCH
003A	010000016	1900		BL
0074	100000000	0620	KEY7	GAME#2
0069	011101011	1902	KEY7	KEY#1
0053	100011000	1906	1903	BRANCH
0026	010001100	1904		BL
004C	100000000	0479	1905	GAME#1
0018	010010001	1906	KEY8	KEY#1
0031	001001110	1907		BL
0062	000101010	1908	LDX	KEY#1
0045	011101101	1909	TCY	7
0004	101000001	1910	ALEC	KEY14
0015	001011010	1910	YNEC	5
0024	100101100	1911		BRANCH
0056	100010100	1914	1912	K10A
002C	001000001	1946	1915	NOP
0058	000100010	1914	K10A	A
0030	100000011	1921	1915	TEST
0060	100010100	1946	1917	1
0041	010001110	1916	KEY14	BRANCH
0002	101010101	1153	1919	BL
		1920	*	KEY#1
0005	011100011	1921	ALEC	12
0005	100010001	1974	1922	BRANCH
0017	001001110	1923	TCY	7
002F	000101010	1924		KEY#1
005F	001011110	1925	YNEC	7
003C	101110001	1928	1926	BRANCH
0078	100010100	1946	1927	K10P
0071	010001010	1928	KEY9	NOP
0063	101011000	0254	1929	ENTER
0047	000101011	1930	KEY15	TYA
000E	010001011	1931		BL
0010	100000011	1862	1932	KEY#1
003A	010010001	1933	KEY2	LDX
0076	001001110	1934	TCY	6
006D	01101100	1935	ALEC	MSD#2
005A	101010016	1949	1936	BRANCH
0036	011100110	1937	ALEC	KEY3
006C	101110010	1962	1956	BRANCH
0059	000101010	1939	1940	KEY6
0032	001011010	1940	YNFC	7
0064	100010100	1946	1941	BRANCH
0049	010000010	1942	1942	1DP

4,189,779

129

130

TABLE IX-13 (Continued)

4,189,779

131

132

TABLE IX-13 (Continued)

0054	001110001	1987	ACACC	8
0028	000001001	1988	MNTA	
0050	100010100	1946	HRANCH	
0020	010001000	1989	HOP	
0040	100101100	1990	K21	
		1991	HL	REPLAY

TABLE IX-14

		ORGPC	14	
1992				
1993		SPEAK		
1994		* ROUTINE TO CONTROL SPEECH TO AND FROM SYNTHESIZER		
1995				
1996				
1997		* IF SS==SET, SPEAK WAS CALLED * IF SS==RESET, HEMADDR WAS CALLED		
1998				
1999				
2000		* IF SS=1, ADDRESSES ARE TRANSFERRED FROM FILES 6 AND 7 TO FILE * 11, WORDS 10=13, ELSE IF SS=0, ADDRESS IS IN FILE 1 PRIOR TO CALL		
2001				
2002				
2003		* 2 POINTERS USED * 1) LINK/EDIT POINTER FOR WORDS IN FILES 6 AND 7 * 2) ROM ADDR POINTER FOR WORDS IN FILE 1.		
2004				
2005				
2006				
2007				
2008				
2009		SPEAK SFAC		
2010		SPEAK+1 LDX	1	
2011		TCY	8	
2012		ICMIV	10	INITIALIZE ROM ADDR POINTER
2013		TCMIV	0	INITIALIZE LNK/EDT POINTER
2014		SPKLOP=1 TCY	9	
2015		SPKLONP TMV		GET WORD FROM LNK/EDT
2016		LDX	7	LOAD WORD IN ACC
2017		TMA		
2018		LDX	1	
2019		TCY	8	
2020		TMV		
2021		TAM		
2022		TCY	8	
2023		IMAC		
2024		TAM		
2025		TCY	9	STORE WORD
2026		TY		RUMP POINTER
2027		CUMX		
2028		TMA		

4,189,779

135

0050	001000001	2030	TCY	H
007A	0010101010	2031	TCY	*
0075	000101111	2032	1AM	*
006H	001001001	2033	TCY	9
0057	000110010	2034	INAC	*
002E	10010001	2111	BRANCH	RETURN
005C	000101111	2036	1AM	*
0038	001000001	2037	TCY	*
0070	000110010	2038	IMAC	*
0061	000101110	2039	1AM24	*
0043	000101010	2040	THY	*
0006	001010111	2041	YNEC	*
0000	100011111	2014	2042	SPKLOP#1
001A	010111111	2043	RETN	*
0037	0010000101	2044	TCY	10
006E	010000111	2045	ADDWS	LDP
005D	0000010101	2046	AHAAC	*
003A	100001010	2057	2047	ADDWS#2
0074	010000111	2048	LUP	14
0069	000000101	2049	TCY	*
0053	001010111	2050	YNFC	14
0026	101101110	2045	2051	BRANCH
004C	011101000	2052	ALFC	1
001E	1001000001	2111	2053	RETURN
0031	010001001	2054	LDP	9
0062	011100100	2055	ALEC	2
0045	1000000000	1299	2056	LETTR
0004	0100000101	2057	ADDWS2	CALLL
0015	111011000	1501	MEMADR	*
		2059	*	MAIN ADDRESSING SUBROUTINE #
		2060	*	ASSUMES X AND Y HAVE BEEN DEFINED PRIOR TO CALLING
		2061	*	*
		2062	*	*
		2063	*	LOADS ADDRESS INTO ROM ADDRESS AREA
		2064	*	ALL R LINES, ETC... REMAIN THE SAME AS WHEN
		2065	*	ENTERING SUBROUTINE.
		2066	*	*
		2067	*	*
		2068	*	*
		2069	*	*
		2070	*	END OF ROUTINE
		2071	*	*
		2072	*	HEMADDR2 TCY
		2073	*	SETB
		2074	*	CLA
		2075	*	SPKREG ACACC
				TEN

136

002H	0010000011	2071	CS, GIVING SYN. COMMANDS
0056	0000001101	2072	R12 = 1
002C	0000000110	2073	*
0058	001110101	2074	*
		2075	*

TABLE IX-14 (Continued)

		TCY	10
0030	0010000101	2076	
0060	000001101	2077	SE1R
0041	00010110	2078	RSTW
0002	0000000110	2079	SPKREG#1 CLA
0005	0000000101	2080	TCY 12
0008	0000000101	2081	SETH
0017	001000101	2082	TCY 10
002F	001110111	2083	ACAC 14
005E	000001101	2084	SETH
003C	0000011010	2085	RSTR
0078	001000101	2086	TCY 11
0071	000110116	2087	RSTW
0063	001000101	2088	TCY 10
0047	000001101	2089	SETH
000E	000110110	2090	RSTW
0010	001110000	2091	ACAC 0
0038	0000001000	2092	TKA
0076	0000001101	2093	SETR
006D	000110110	2094	RSTR
005H	001000101	2095	TCY 11
0036	0000001101	2096	SETR
006C	010011100	2097	LDX 3
0059	001001111	2098	TCY 15
0032	000101111	2099	TAM
0064	000000000	2100	TAIT 0
0049	1010101010	2125	BRANCH HI7SE10
0012	010011000	2102	LDX 1
0025	001000001	2103	TCY 6
004A	001100101	2104	TCMIV 10
0014	0000001010	2105	CCLA
0029	011100000	2106	ALEC ZERO
0052	101091000	2109	BRANCH RETS
0024	100011111	2014	BRANCH SPKLUP=1
0048	610001000	2105	RETS IDX 1
0010	601000001	2110	TCY 8
0021	000100110	2111	RETURN TAMZA
0042	001001111	2112	TCY 15
0004	010010110	2113	RETURN4 LDX SIX
0009	000101111	2114	TAM
0013	010011110	2115	LDX SEVEN
0027	000101100	2116	TAMDN BRANCH
004E	1000000100	2114	RETURN4

60

65

4,189,779

137

138

1ST PDC LOADS COMMAND

2ND PDC APPLIES TALK TO CTLB

3RD PDC RELEASES OUTPUT

ACC = ZERO

TABLE IX-14 (Continued)

001C	010111111	2118	RETIN		
0039	0101110100	2119	RETURN+1 REAC		
0072	001001111	2120	RETURN+2 TCY	15	TALK BIT
0065	010011100	2121	LDX	3	*
004B	010100100	2122	RHI	0	*
0016	010000010	2123	RL		RETNSBCH
002D	101001010	2124			
005A	010001111	2125	BITSETU	LDP	
0054	100101100	2219	2126	HANCH	DISP/KB
		2127	*		
		2128	*		
		2129	*		
		2130	LNKPTR	LDX	1
		2131		TCY	9
		2132		THY	*
		2133		LDX	6
		2134		TAM	
		2135		LDX	1
		2136		TCY	9
		2137		RETN	
		2138	*		
		2139	ADD8	TCY	6
		2140		LDX	A
		2141		TBT	3
		2142		BRANCH	RADD8
		2143		BRANCH	RADD2
		2144	RADD8	LDX	
		2145		TCY	13
		2146		TCMY	A
		2147	RADD2	RETN	

4,189,779

139

140

POINTER FOR LNK/EDT

*

a

*

STORE WORD
POINTER

*

a

END OF SPECIFIC CONTROL SUBROUTINE

a

POINTER UP / CLEAR ROUTINE

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

a

TABLE IX-15 (Continued)

		TCY	10
0031	001000101	2205	
0062	000001101	2206	
0045	000110110	2207	
000A	010000101	2208	
0015	110100100	1534	2209
0028	010001101	2210	
0056	101000111	1657	2211
		2212	*
		2213	*
		2214	*
		2215	*
		2216	*
		2217	*
		2218	*
		2219	DISP/KH
		2220	LCX
		2221	TCY
		2222	ICM1Y
		2223	RSTR
		2224	ICM1Y
		2225	CLA
		2226	TCY
		2227	LDX
		2228	TAM1YC
		2229	ICM1Y
		2230	TCY
		2231	SETR
		2232	TCY
		2233	LDX
		2234	TCY
		2235	LDX
		2236	TMA
		2237	MNFA
		2238	TDD
		2239	SETR
		2240	TCY
		2241	RSTR
		2242	BL
		2243	TIMFLUP
		2244	*
		2245	DISP/R81
		2246	TCY
		2247	TMAC
		2248	TCY
		2249	SEIR
		2250	TAY

143

4,189,779

144

KEYBOARD SCAN / DISPLAY ROUTINE
 THIS ROUTINE DISPLAYS THE CONTENTS OF DISPLAY BUFFER AND
 CHECKS FOR A KEYPRESS.

TABLE IX-15 (Continued)

0014	000000100	2251	DYN
0029	000110110	2252	RS1K
0052	000000101	2253	TYC
0024	001010061	2254	YNEFC
0048	101110001	2252	DSP2
0010	001001111	2256	15
0021	000110110	2257	TCY
0042	010006000	2258	RSTK
0009	110101011	0106	TIMEUP1
		2259	CALLL
		2260	*
0009	010010000	2261	LDX
0013	001000101	2262	TCY
0027	000110010	2263	IMAC
004F	100111001	2266	HBRANCH
001C	000101111	2264	TAM
0039	001000001	2265	DSP3
0072	000110010	2266	TCY
0065	011100101	2267	IMAC
0048	100000101	2268	ALEC
0016	010000111	2269	LDP
002D	001001111	2270	DSP1
0054	010011100	2271	TCY
0034	000100000	2272	LUX
0068	101011000	2273	THIT
0051	010000001	2075	BRANCH
0022	001000111	2274	SPKREG + 1
0044	000101011	2275	LDP
0008	000100000	2276	A
0011	101101100	2277	TCY
0023	010001111	2278	THIT
0046	100000101	2279	BRANCH
000C	010010000	2280	LDP
0019	000100011	2281	BRANCH
0033	000101001	2282	KEYSEVL
0066	001001111	2283	TCY
0040	610001011	2284	IMA
0014	000100000	2285	TCY
0035	100011011	1896	LOP
0064	100000000	1860	BRANCH
0055	010010001	2286	KEY1
0024	001000100	2287	BIT
0054	010100011	2288	BRANCH
0024	010111111	2289	KEY00
		2290	*
		2291	SETHTIS
		2292	LOX
		2293	TCY
		2294	SMIT
		2295	RETN

4,189,779

145

146

TABLE X

<u>I₀/I₁ COMMANDS</u>		
I ₀	I ₁	
0	0	No Operation
0	1	Load Address (LA)
1	0	Transfer Bit (TB)
1	1	Read and Branch (RB)

TABLE XI

<u>Counter 619/PLA 620 Timing Sequence</u>		
STEP	COUNTER CONTENTS (HEX)	SIGNALS GENERATED
1	0	LA1, TB8
2	8	LA2
3	C	LA3
4	E	LA4
5	F	
6	7	
7	3	
8	1	

TABLE XII

<u>TB8 READ SEQUENCE</u>			
TEP	COUNTER 623 CONTENTS (BINARY)	COUNTER 624 CONTENTS (HEX)	SIGNALS GENERATED
1	10	F	SAD, INC
2	10	E	DC, INC
3	10	C	DC, INC
4	10	8	DC, INC
5	10	0	DC, INC
6	10	1	DC, INC
7	10	3	SAM, DC, INC
8	10	7	PC, ZERO

TABLE XIII

<u>TB8 READ SEQUENCE</u>			
TEP	COUNTER 623 CONTENTS (BINARY)	COUNTER 624 CONTENTS (HEX)	SIGNALS GENERATED
1	11	F	SAD, INC
2	11	E	DC, INC
3	11	C	DC, INC
4	11	8	DC, INC
5	11	0	DC, INC
6	11	1	DC, INC
7	11	3	SAM, DC, INC
8	11	7	PC
9	01	F	SAD, TF
0	01	E	BR, PC
1	01	C	BR, DC
2	01	8	BR, DC
3	01	0	BR, DC
4	01	1	DC
5	01	3	SAM, DC
6	01	7	PC
7	00	F	SAD, TF
8	00	E	BR
9	00	C	BR
0	00	8	BR
1	00	0	
2	00	1	
3	00	3	
4	00	7	PC
5	10	F	SAD, INC
6	10	E	DC, INC
7	10	C	DC, INC
8	10	8	DC, INC
9	10	0	DC, INC
0	10	1	DC, INC
1	10	3	SAM, DC, INC
2	10	7	PC, ZERO

What is claimed is:

1. A parameter interpolation for a speech synthesizer having an input means for receiving target values of speech parameters and a memory means for storing interpolated values of speech parameters, said parameter interpolator comprising:
- (a) first means coupled to said input means and said memory means for calculating the difference between the target values of the parameters and values of the parameters stored in said memory means;
 - (b) second means coupled to said first means and said memory means for adding a portion of differences calculated by said first means to values parameters stored in said memory means;
 - (c) third means for determining the particular portions of the differences to be added by said second means according to the formula $1/2^N$ where $N=0, 1, 2, \dots, N$; and
 - (d) fourth means for inserting the output of said second means into said memory means.
2. The parameter interpolator according to claim 1 wherein N equals the number three.
3. A parameter interpolator for a speech synthesizer having an input means for receiving a plurality of target values of speech parameters and a memory means for storing a plurality of values of speech parameters being utilized by said speech synthesizer, said parameter interpolator comprising:
- (a) timing means for generating eight interpolation cycles;
 - (b) subtractor means coupled to said input means and said memory means for calculating the difference between the target values of said parameters and the values of said parameters stored in said memory means during each interpolation cycle;
 - (c) adder means coupled to said subtractor means and to said memory means for adding a selected portion of the difference calculated by said subtractor means to the values of said parameters stored in said memory means during each interpolation cycle, said adder means adding one-eighth of differences during each of three successive interpolation cycles, adding one-fourth of the differences during each of two successive interpolation cycles, adding one-half of the differences during each of two another successive interpolation cycles and adding the entire differences during one of the eight interpolation cycles; and
 - (c) circuit means for replacing the values of the parameters stored in said memory means with the results of addition performed by said adder means during each interpolation cycle.
4. The interpolator according to claim 3, wherein said circuit means replaces each value of the parameters stored in said memory means after each value has been applied to said adder and subtractor means during each interpolation cycle and before the values in the memory means are output to the adder and subtractor means during the next successive interpolation cycle.
5. The interpolator according to claim 4, wherein said speech synthesizer is responsive to an excitation parameter which is indicative of voiced and unvoiced speech, and wherein said interpolator further includes a detector responsive to a change between voiced and unvoiced speech and means for disabling said adder from adding either one-eight, one-fourth, or one-half of the differences to the values stored in said memory means in

response to said detector detecting a change from voiced to unvoiced speech or unvoiced to voiced speech, whereby the values of the parameters in said memory means are not interpolated to the target values in eight steps but rather assume the target values in one step during changes from voiced to unvoiced speech or unvoiced to voiced speech.

6. The system according to claim 4, wherein the values of the parameters in said memory means and the target values of the parameters from said input means are applied in serial to said subtractor means and wherein said adder means includes means for delaying the output of the subtractor means by either zero, one, two or three bits whereby the portion of the differences added in the adder correspond to $1/2^N$ wherein N is equal to the number of bits of delay occurring in said delay means.

7. The system according to claim 6, wherein said circuit means includes a delay circuit for delaying the results of the addition by either zero, one, two or three bits, the delay circuit delaying:

- (i) three bits when the delay means is delaying zero bits,
- (ii) delaying two bits when the delay means is delaying one bit,

- (iii) delaying one bit when the delay means is delaying two bits, and
- (iv) delaying zero bits where the delay means is delaying three bits.

8. A speech parameter interpolator for a speech synthesis circuit having an input for receiving target values of digital speech parameters and a memory for storing values of said digital speech parameters used by said synthesis circuit in synthesizing speech, said interpolator comprising:

- (a) subtractor means coupled to said input means and to said memory for calculating the difference between said target values and the values stored in said memory;
- (b) first means for generating 2^N interpolation cycles, where N equal 0,1,2... N; and
- (c) means coupled to said subtractor means and said memory, and responsive to said first means for adding a selected portion of the difference calculated by said subtractor means, during each of said interpolation cycles, to the values of said digital speech parameters stored in said memory.

* * * *

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 4,189,779

DATED : February 19, 1980

INVENTOR(S) : George L. Brantingham

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Inventor's name should read -- George L. Brantingham --.

Signed and Sealed this

Twenty-seventh Day of May 1980

[SEAL]

Attest:

SIDNEY A. DIAMOND

Attesting Officer

Commissioner of Patents and Trademarks