United States Patent [19]

Puri

[11] **4,335,277**

[45] **Jun. 15, 1982**

[54] **CONTROL INTERFACE SYSTEM FOR USE WITH A MEMORY DEVICE EXECUTING VARIABLE LENGTH INSTRUCTIONS**

[75] Inventor: **Ajay K. Puri**, Lubbock, Tex.

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

[21] Appl. No.: **250,982**

[22] Filed: **Apr. 3, 1981**

Related U.S. Application Data

[63] Continuation of Ser. No. 36,928, May 7, 1979, abandoned.

[51] Int. Cl.³ **G10L 1/00**
[52] U.S. Cl. .... **179/1 SM; 364/900**
[58] Field of Search .... 179/1 SM, 1 SG, 1 SB; 364/200 MS File, 900 MS File

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,575,555 4/1971 Schanne .... 179/1 SM
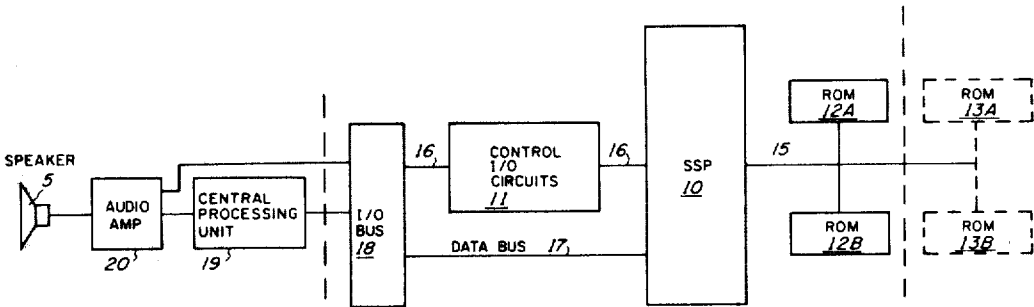3,662,348 5/1972 Weiss .... 179/1 SM

OTHER PUBLICATIONS

D. Cannon, et al. "Designing with Microprocessors", Texas Instruments Co., 1975, pp. 112, 113.
Mozer, "Talking Board", Computer Design, Jul. 1976, pp. 46, 48.

Primary Examiner—Joseph F. Ruggiero
Assistant Examiner—E. S. Kemeny
Attorney, Agent, or Firm—William E. Hiller; Melvin Sharp; James T. Comfort

[57] **ABSTRACT**

A control interface system for use with a memory device executing variable length instructions and implementable utilizing integratable field effect semiconductor devices is disclosed. One example of such a system is a speed synthesis device controlled by the central processing unit of a computer. A control latch is utilized to release the control device, such as a central processing unit, after an instruction has been initiated. A plurality of latches, each set by the conclusion of the execution of a previous instruction, are utilized to reset the control latch. An attempted second access of the memory device during the execution of a previous command will result in suspending the operation of the control device until such time as the previous instruction is concluded and the new instruction is initiated.
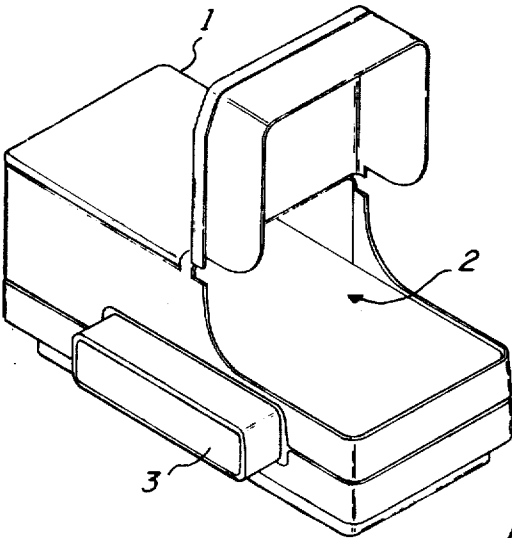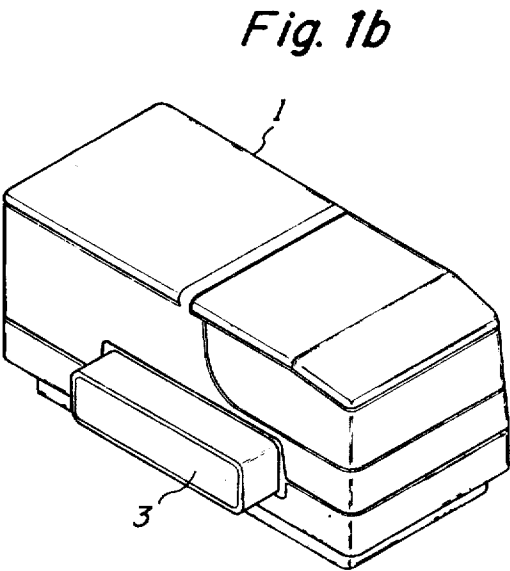
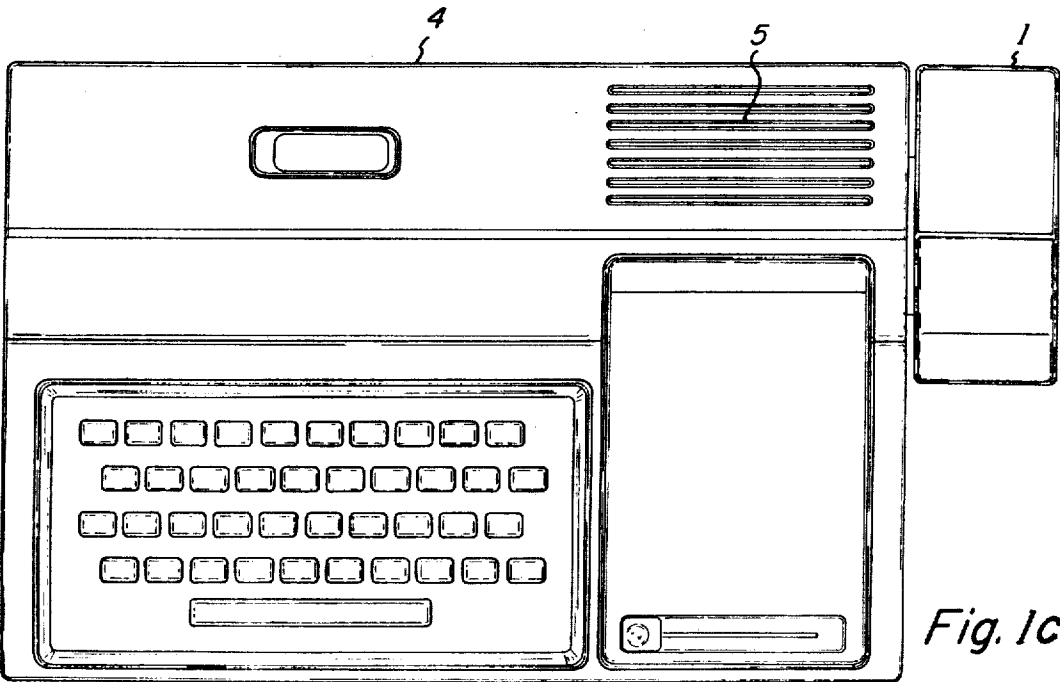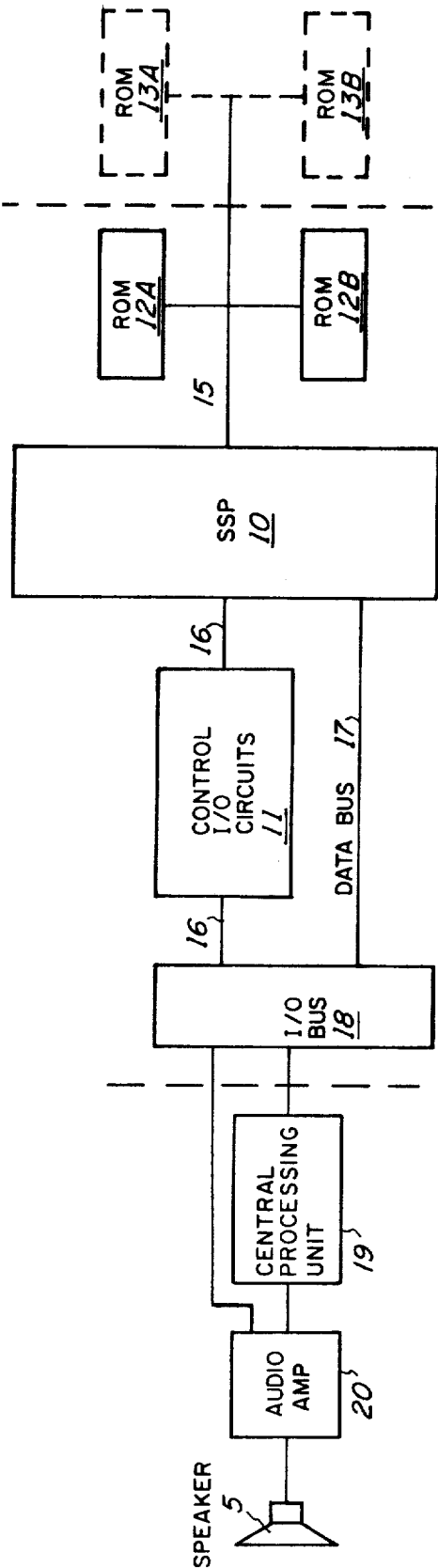**6 Claims, 43 Drawing Figures**

Fig. 1b

Fig. 1a

Fig. 1c

Fig. 2
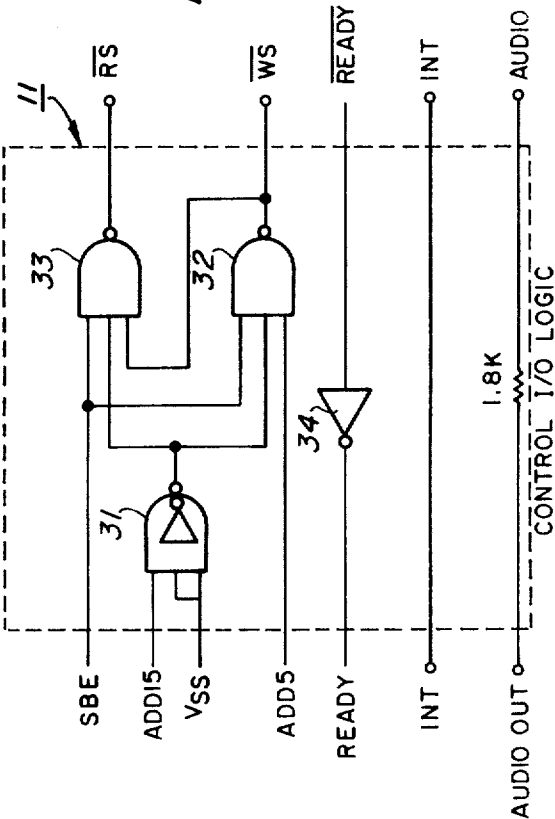
Fig. 3

Fig. 4a

Fig. 4b

Fig. 5

Fig. 6

Fig.7a

Fig. 7b

Fig. 7c

Fig. 7d

Fig. 8a

Fig. 8b

Fig. 8c

Fig. 8d

Fig.8e

Fig. 8f

Fig. 8g

Fig. 8h

Fig. 8i

JOINS Fig. 8h

Fig. 8i (CON'T) FIFDSO

Fig. 8j

FIFO CNTR

Fig. 8i (con't) 2250

CNT SEQUENCE

RST	I	I	I	I
	0	I	I	I
(INC	0	0	I	I
ENABLED	0	0	0	I
ON SSE	0	0	0	0
	I	0	0	0
	I	I	0	0
	I	I	0	0
	I	I	I	I

LOAD SPEECH LOGIC

Fig. 8k

Fig. 8l

Fig. 9a

Fig. 9b

Fig. 9c

Fig. 9d
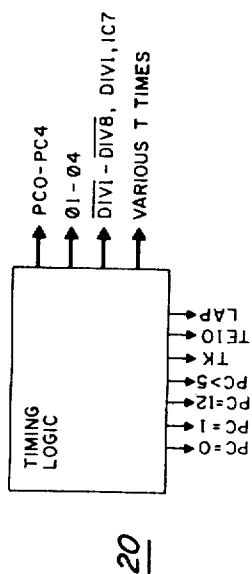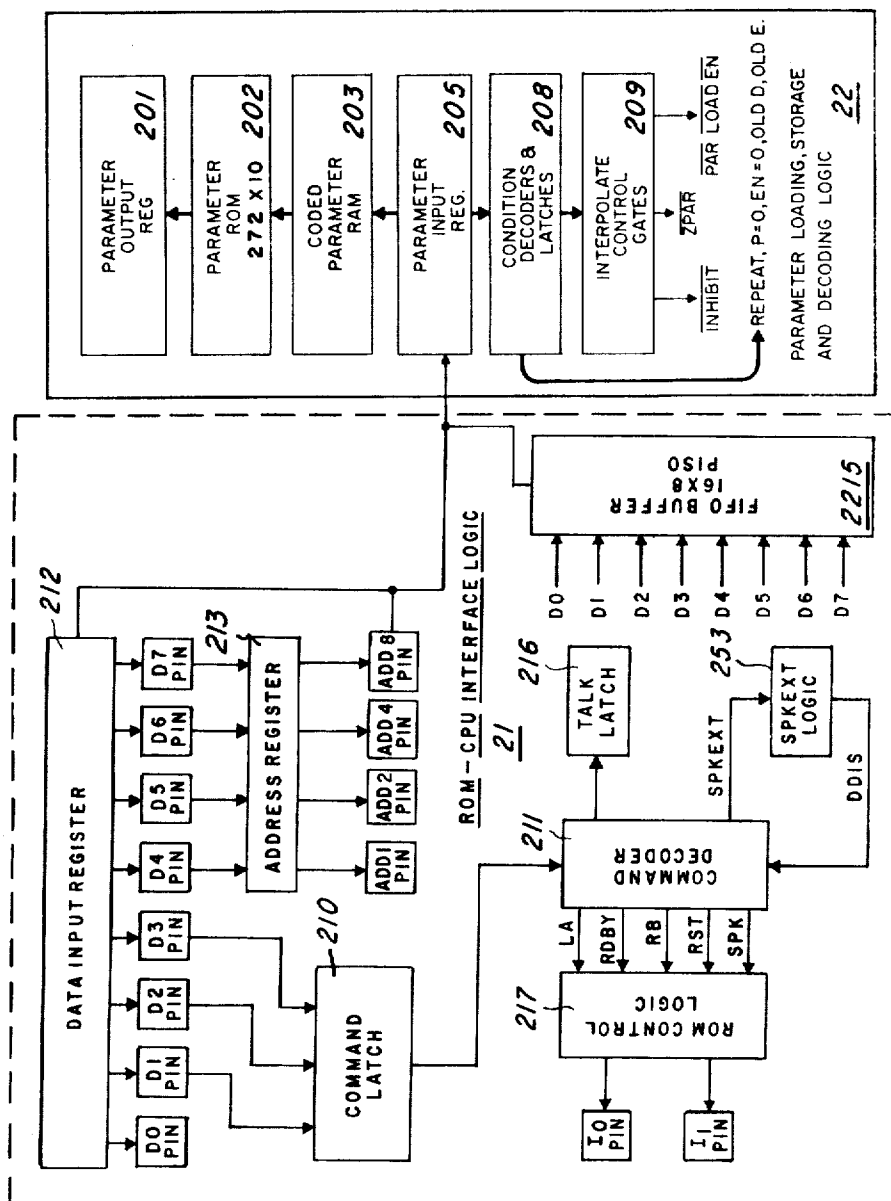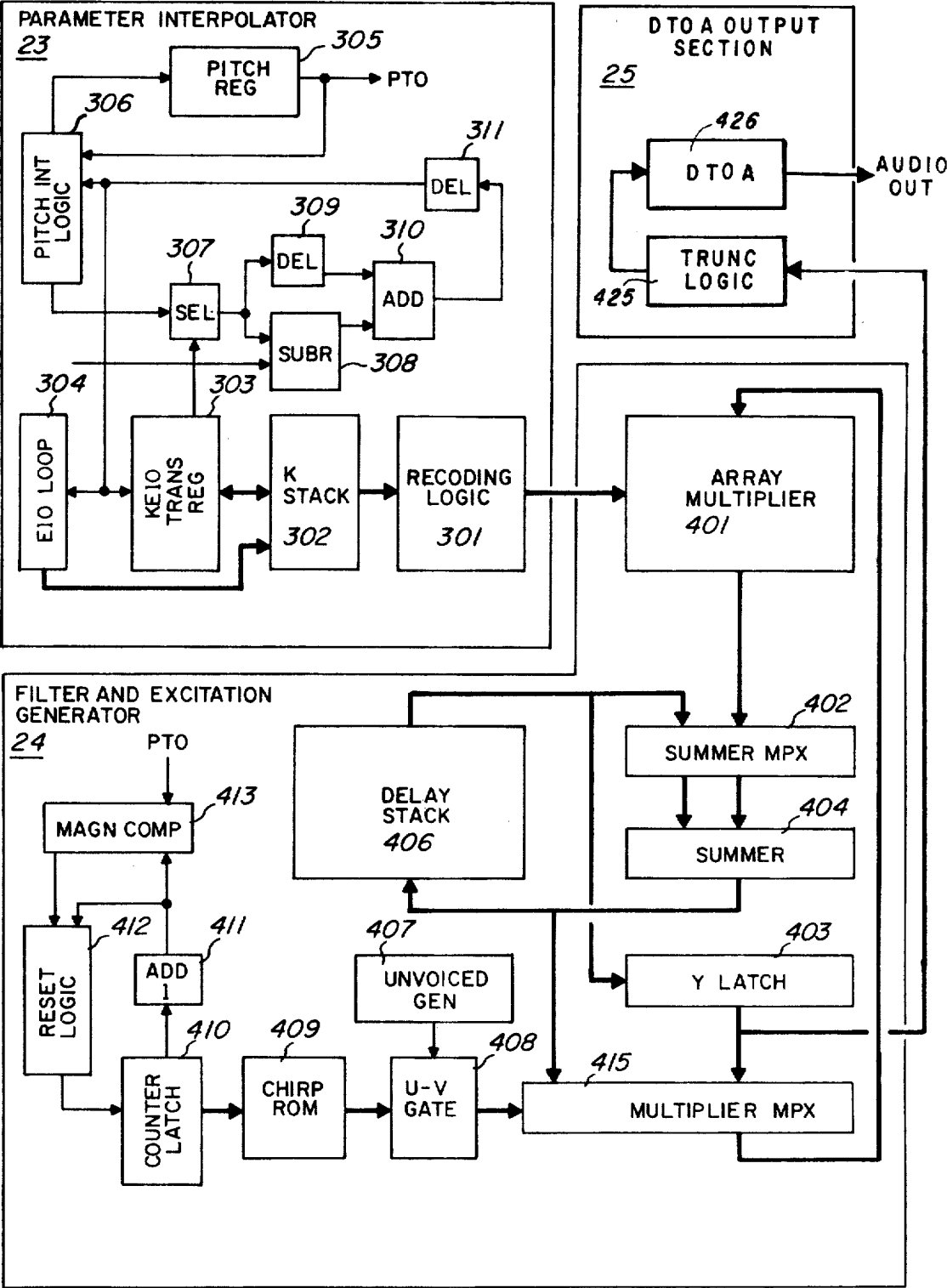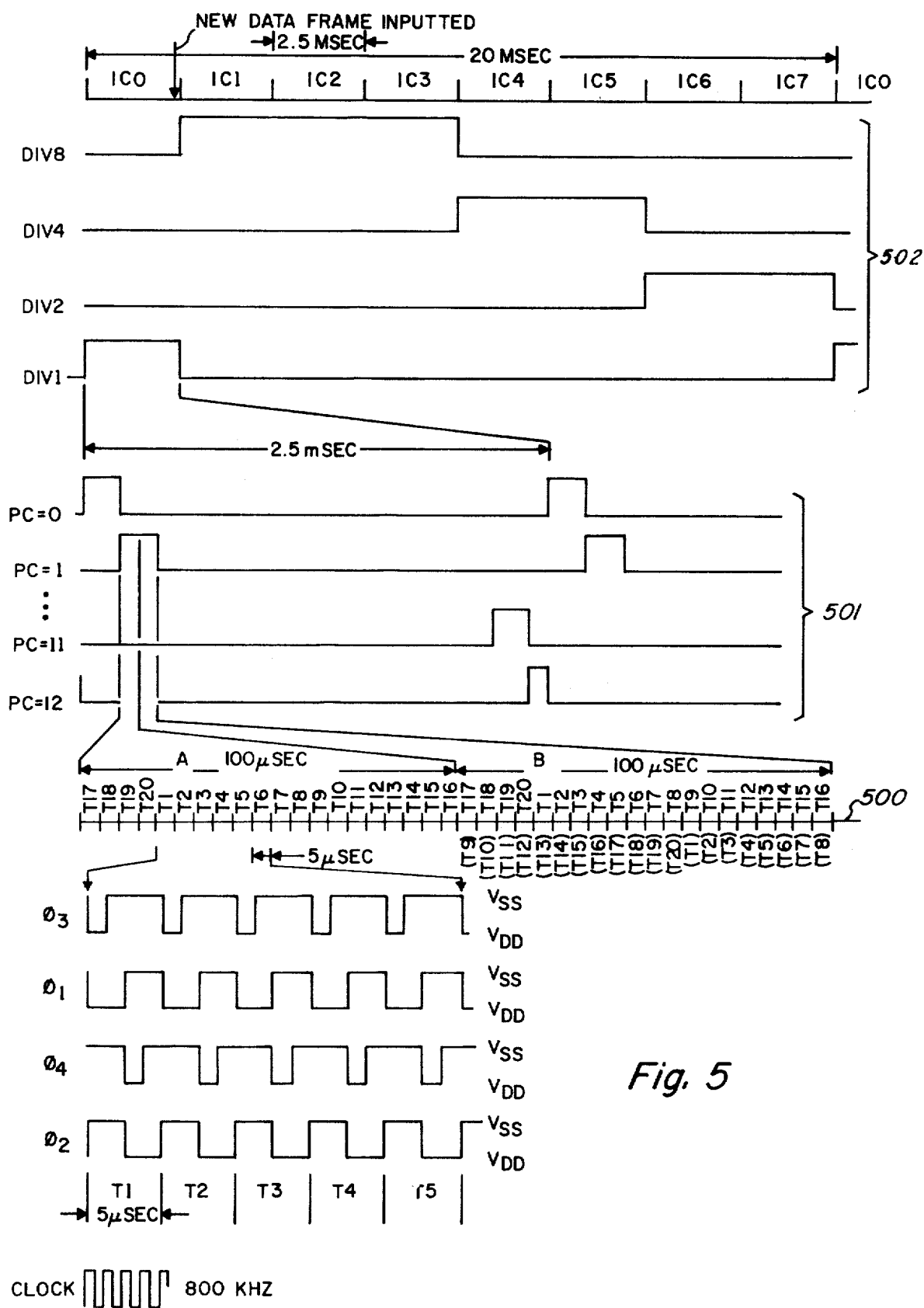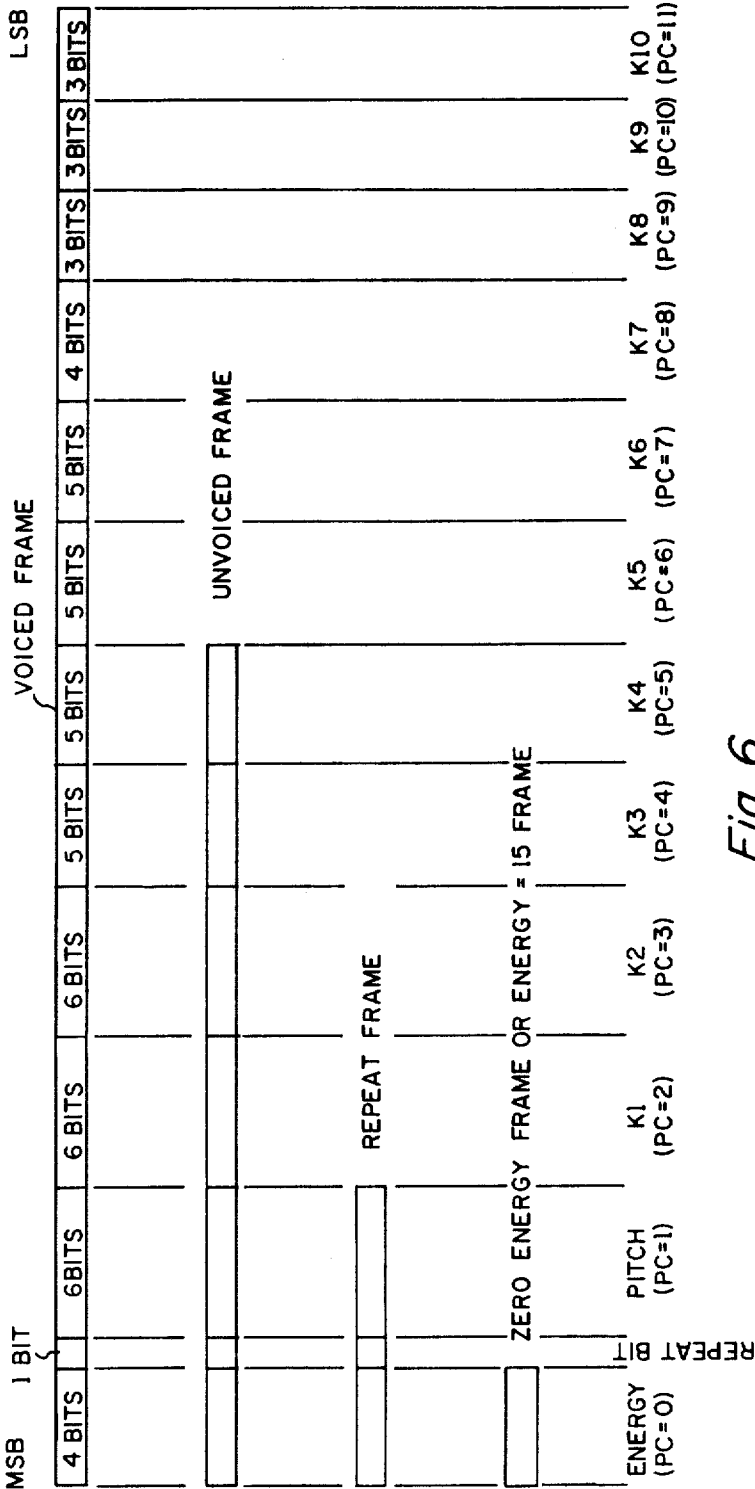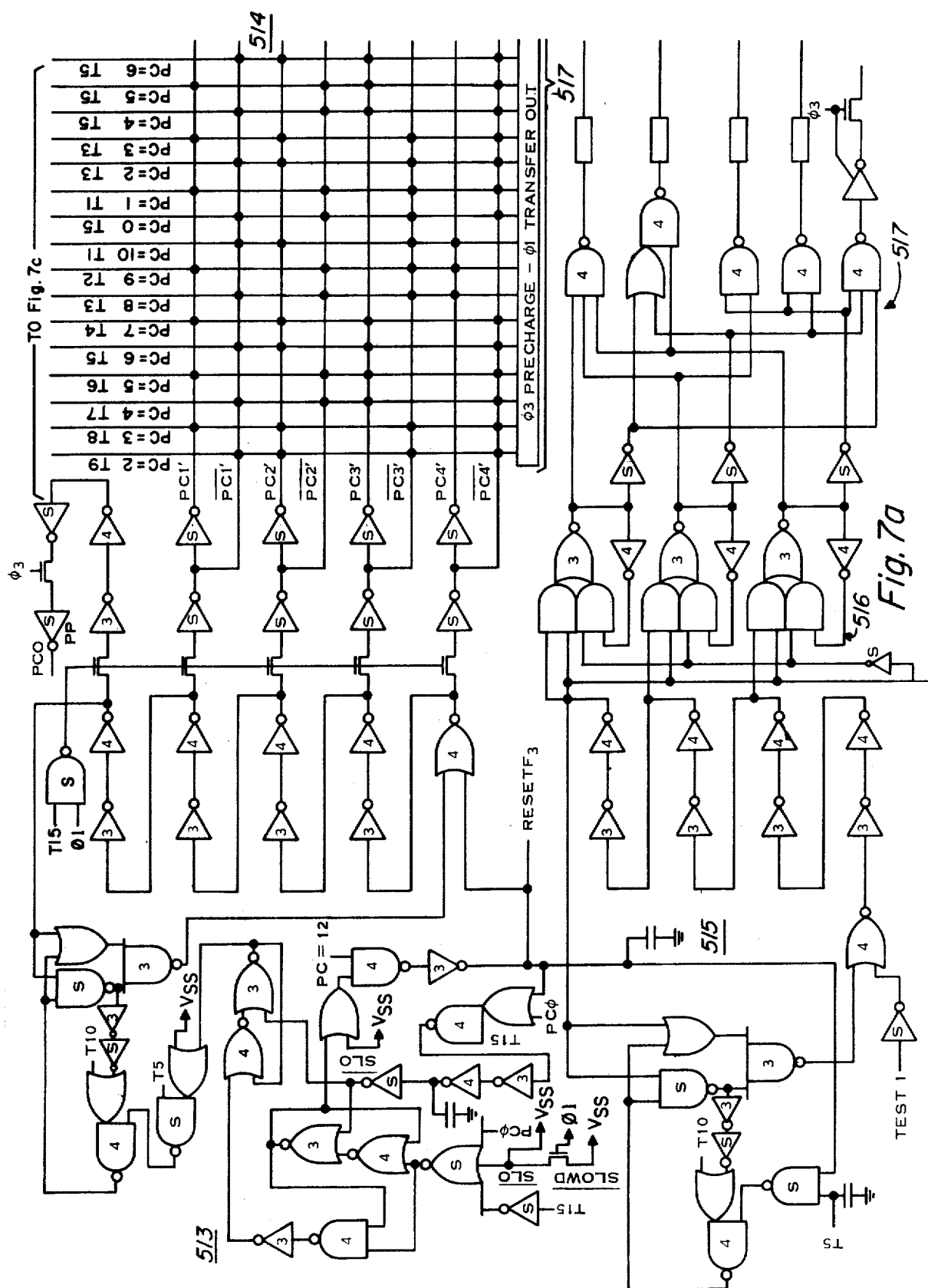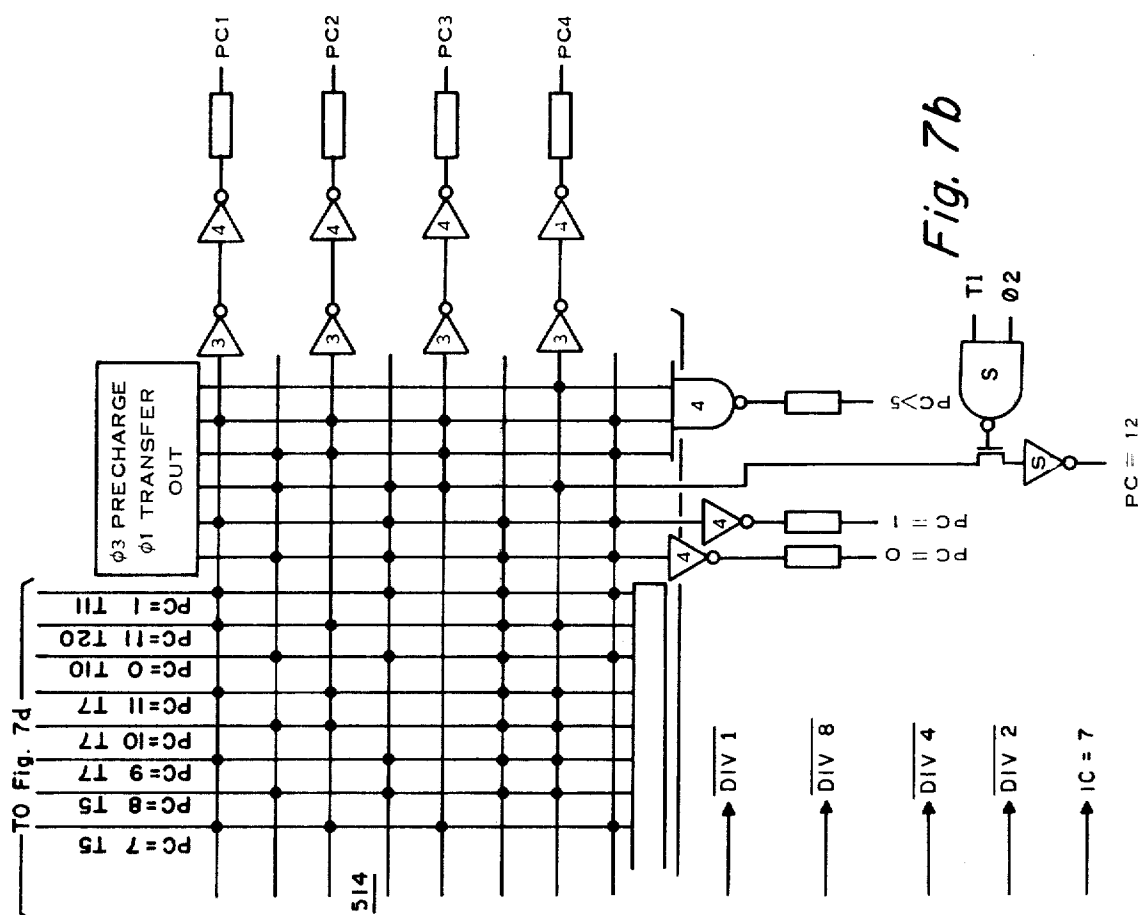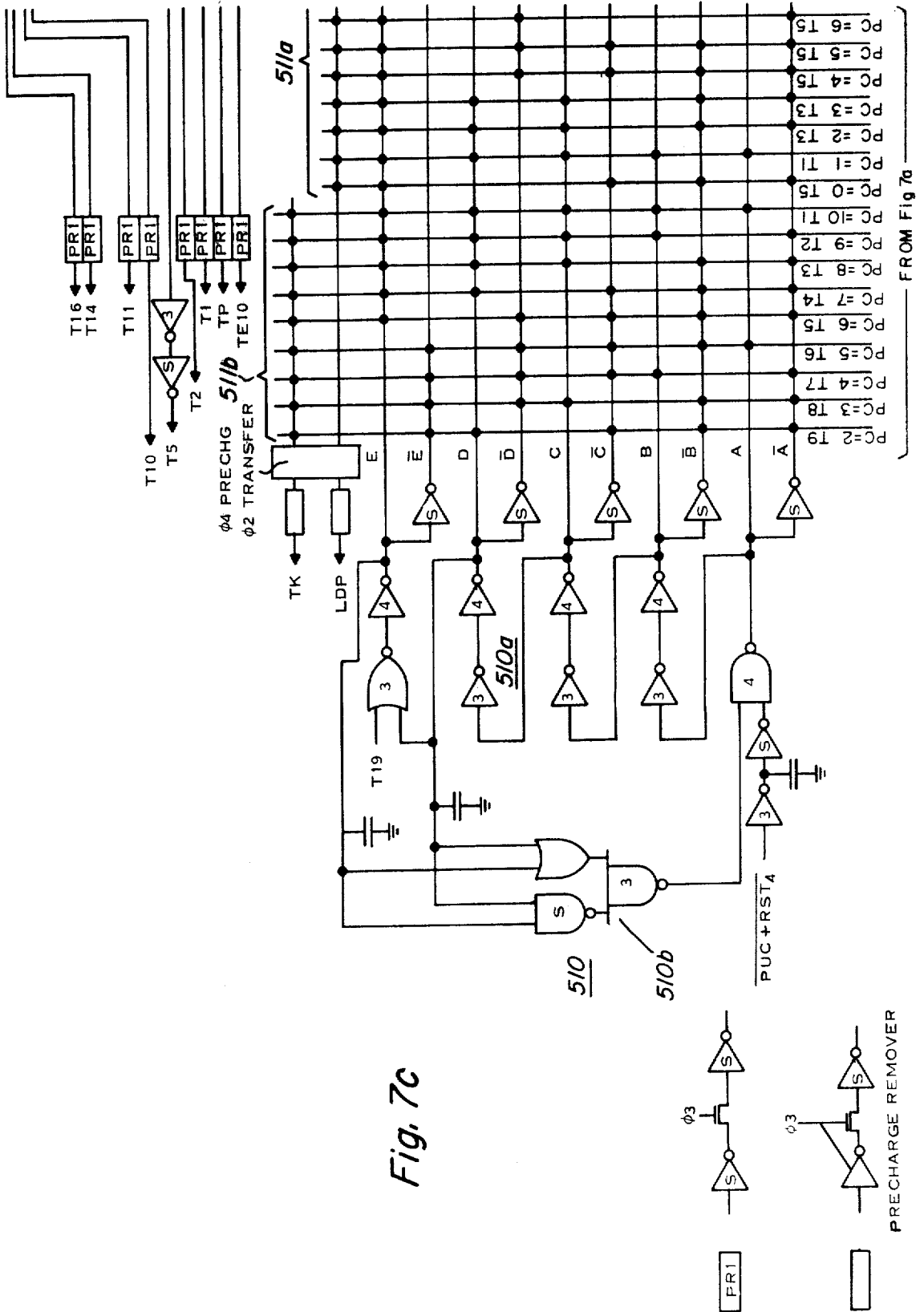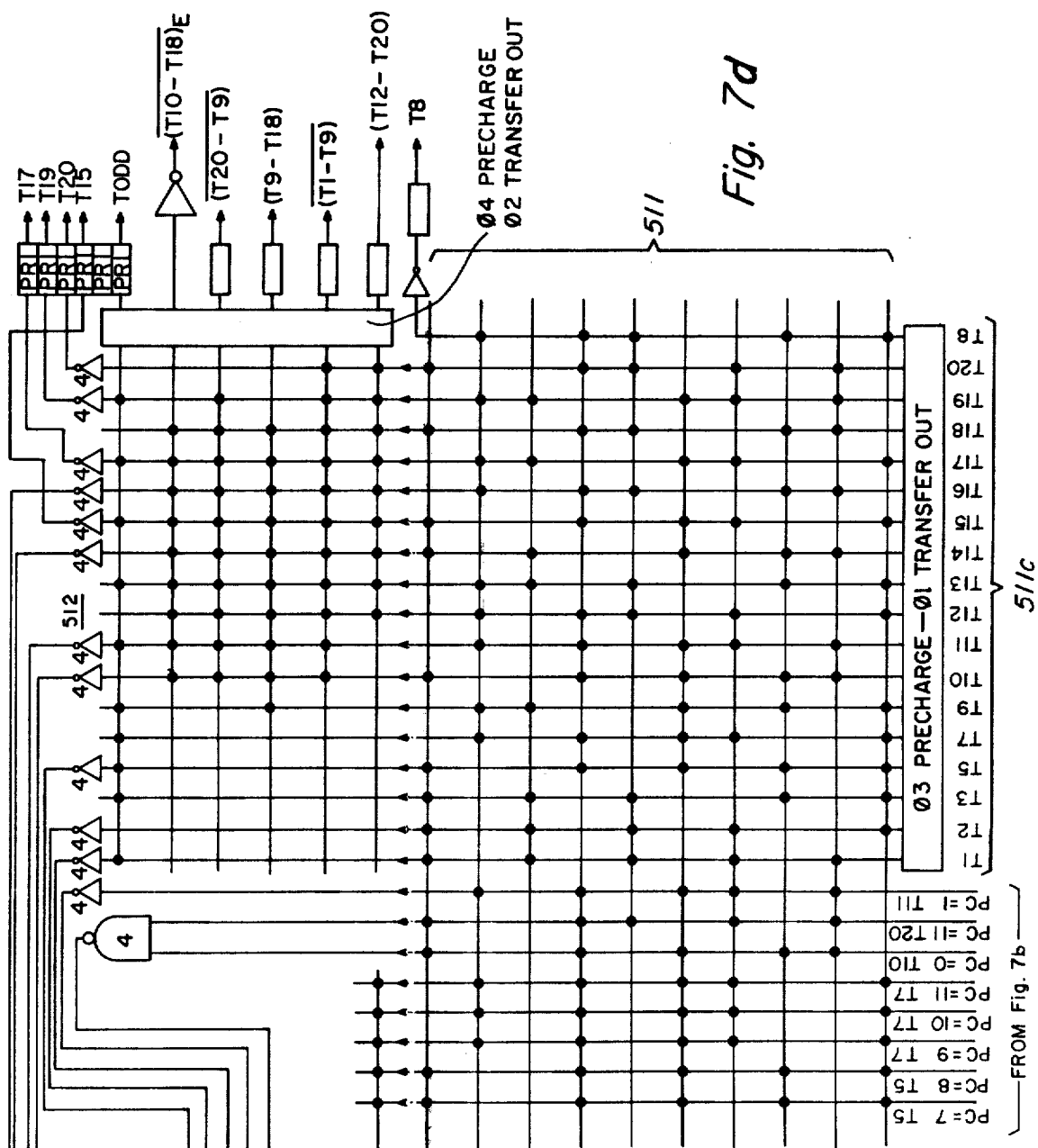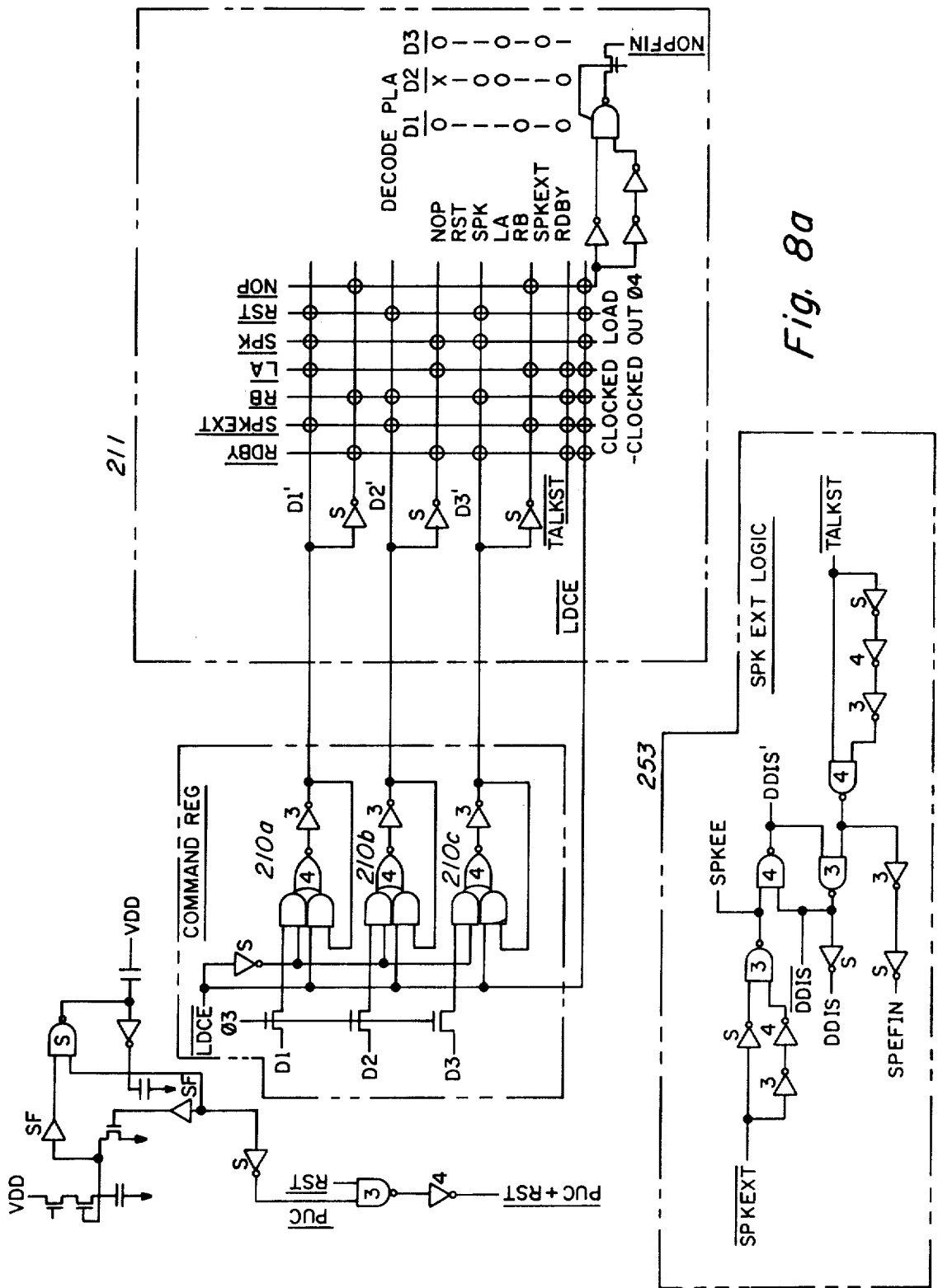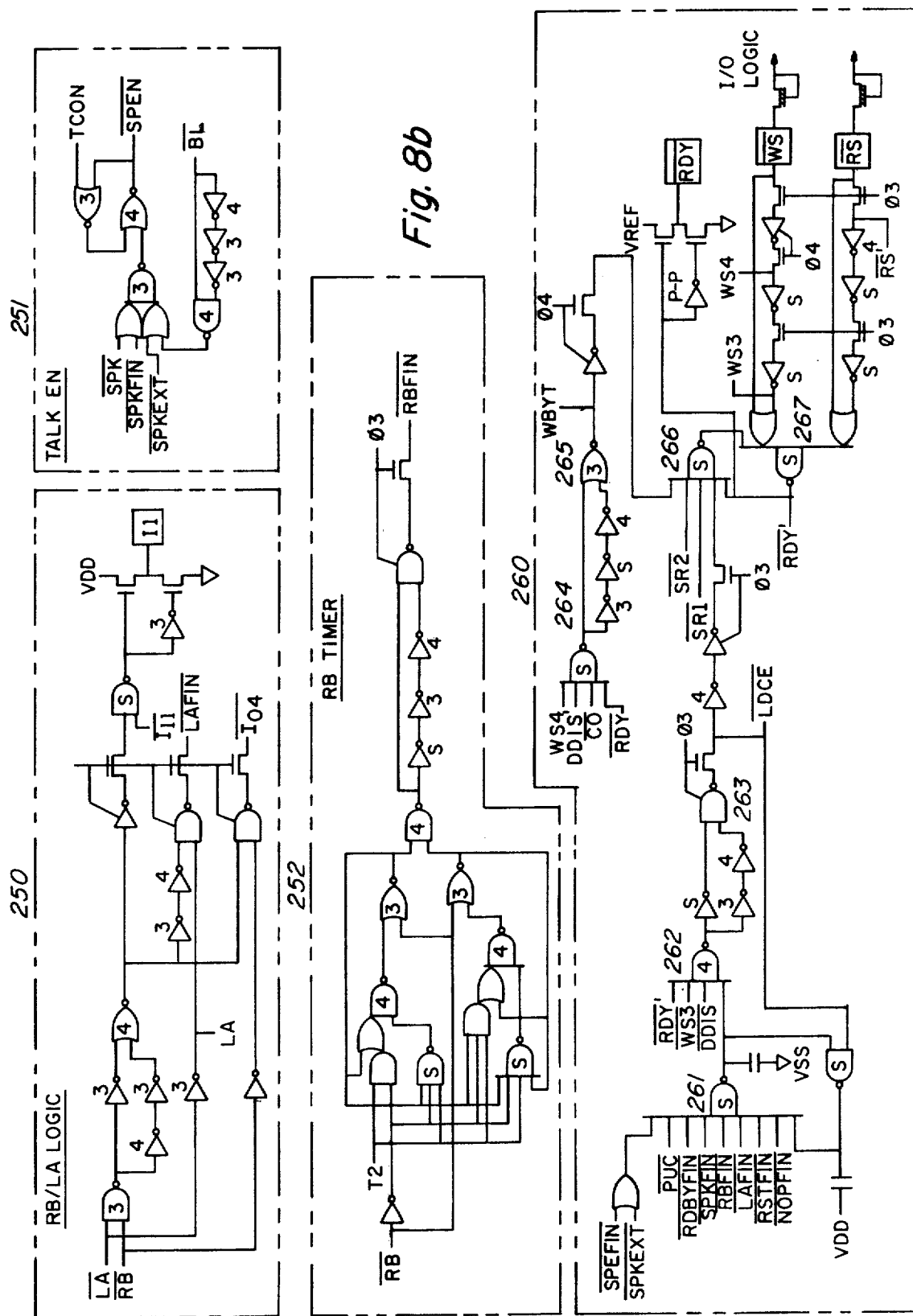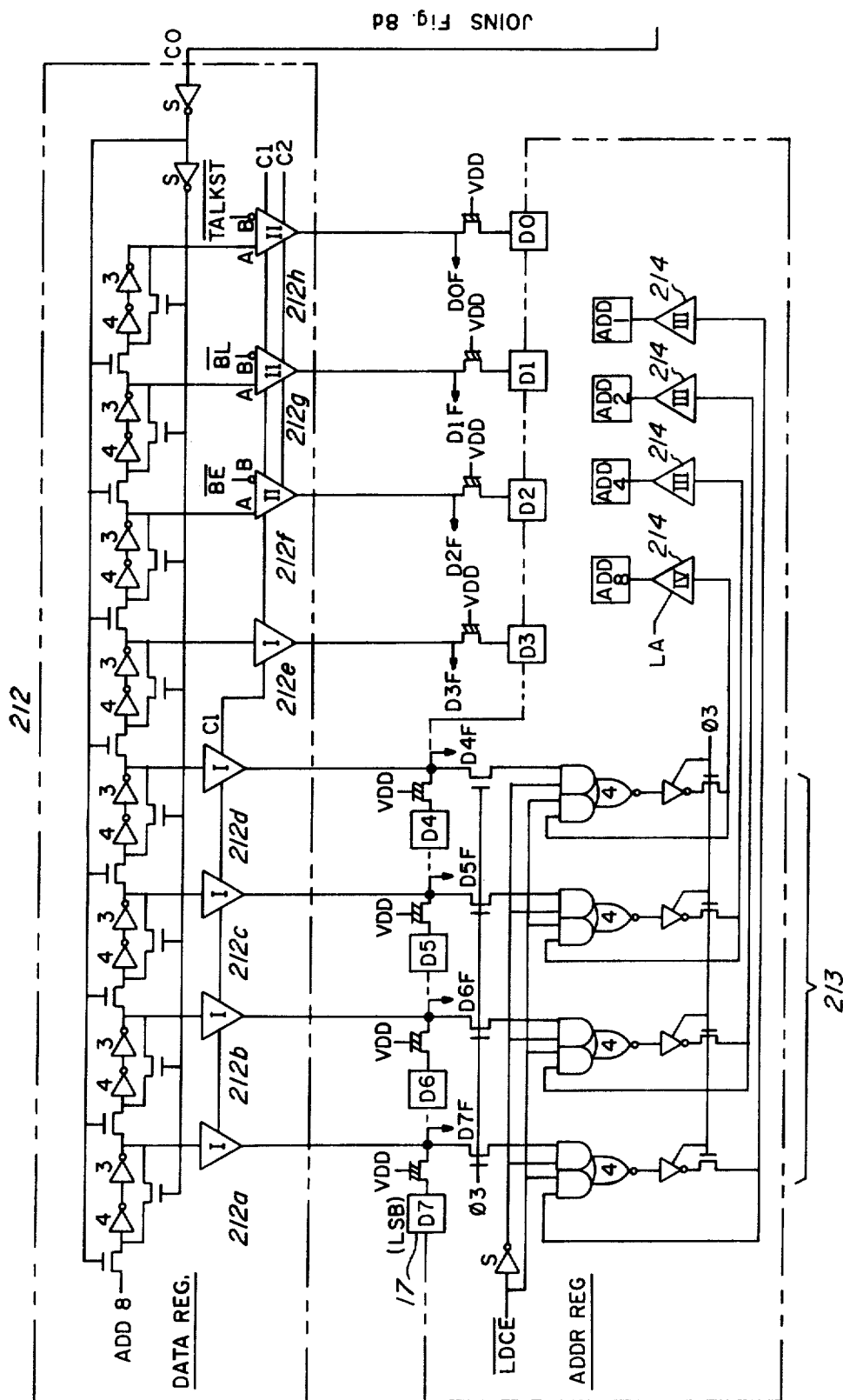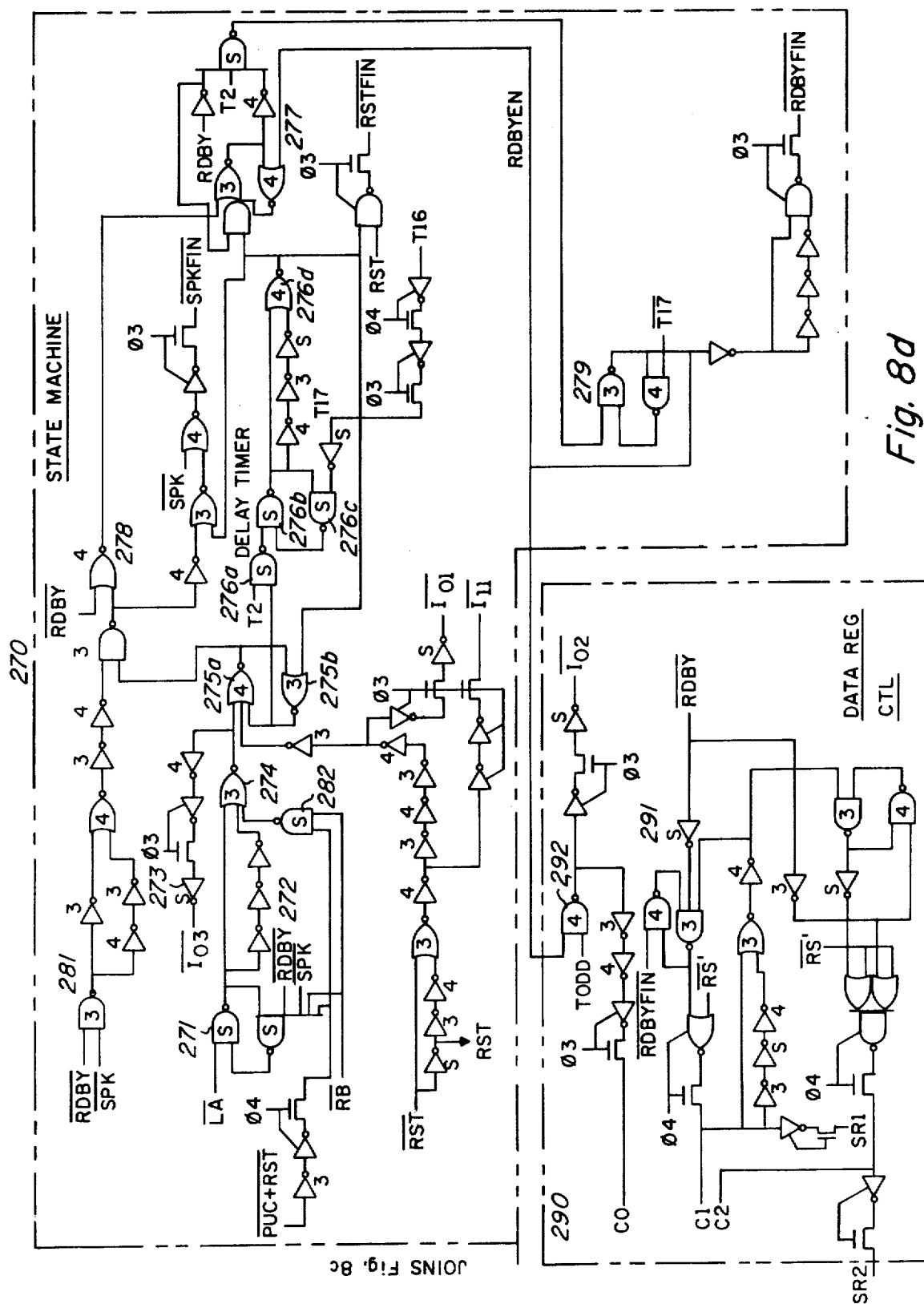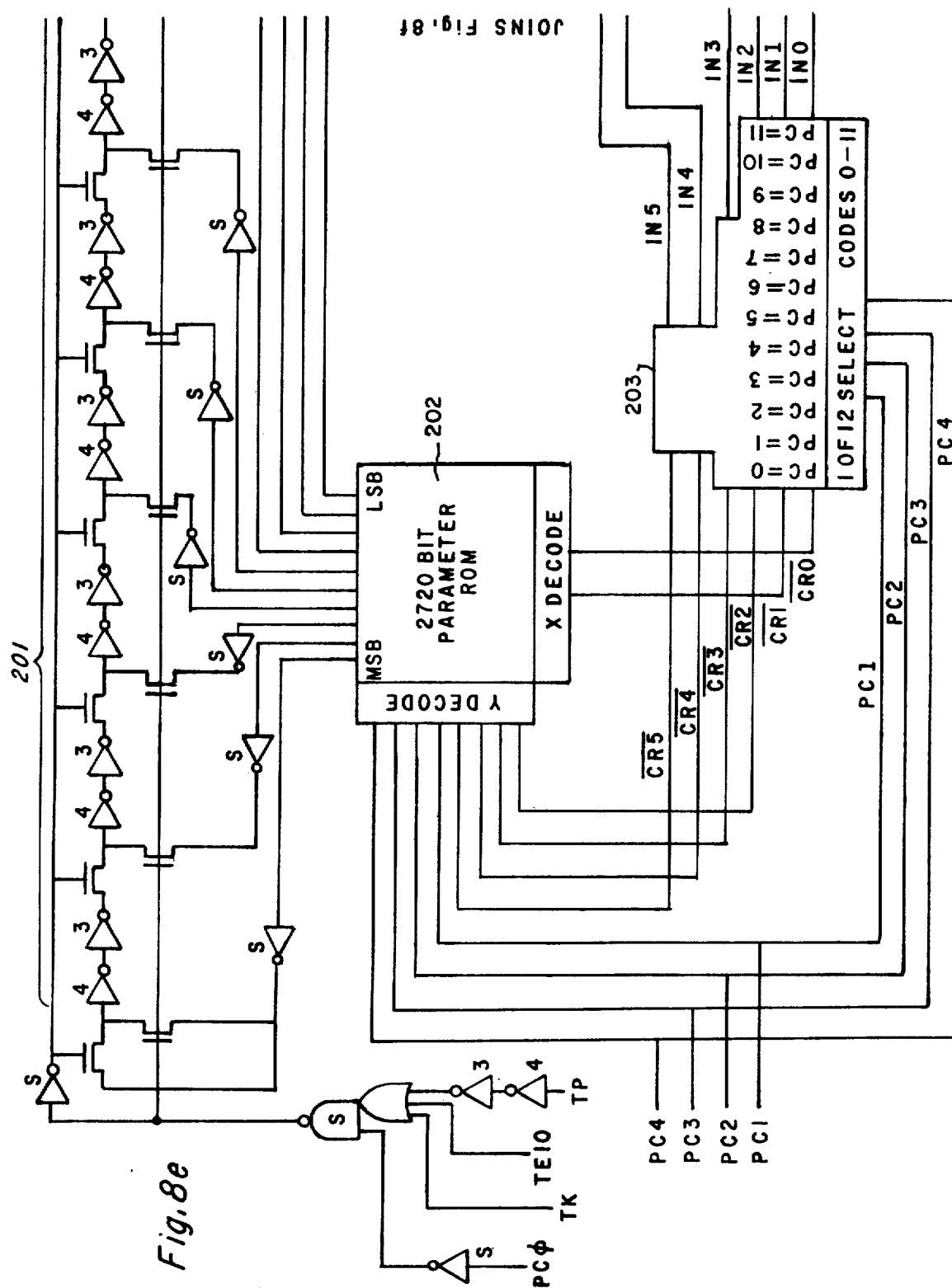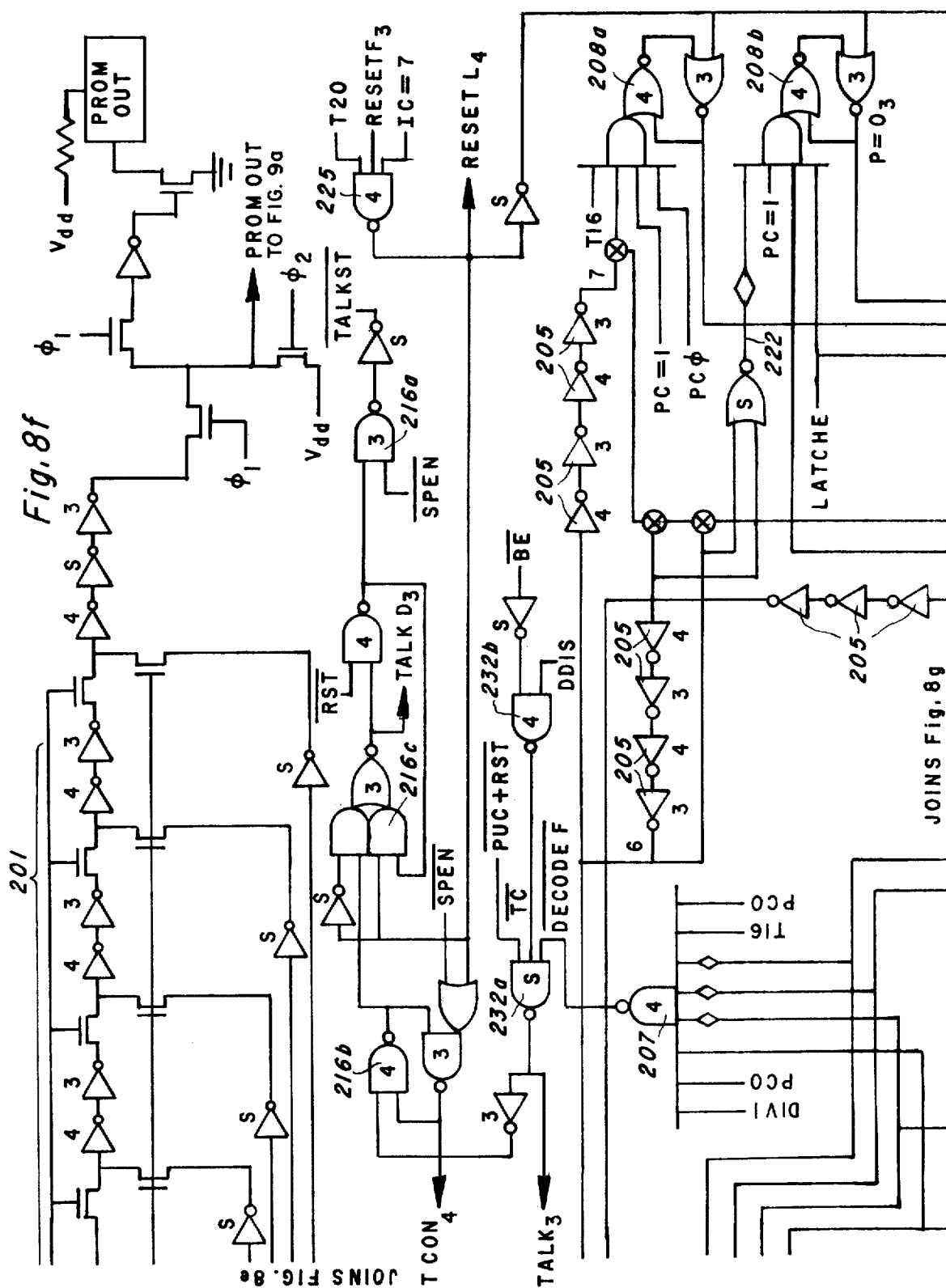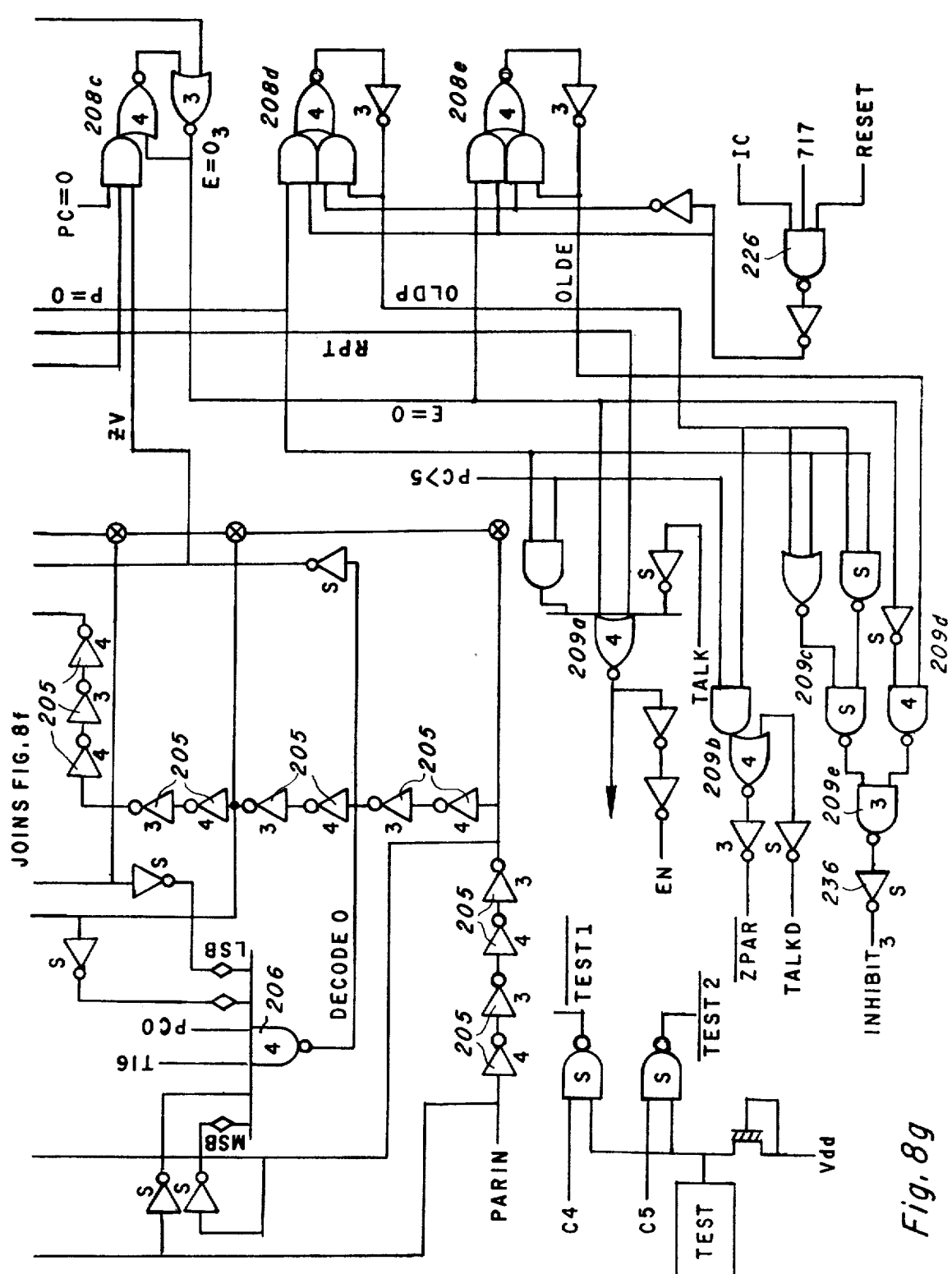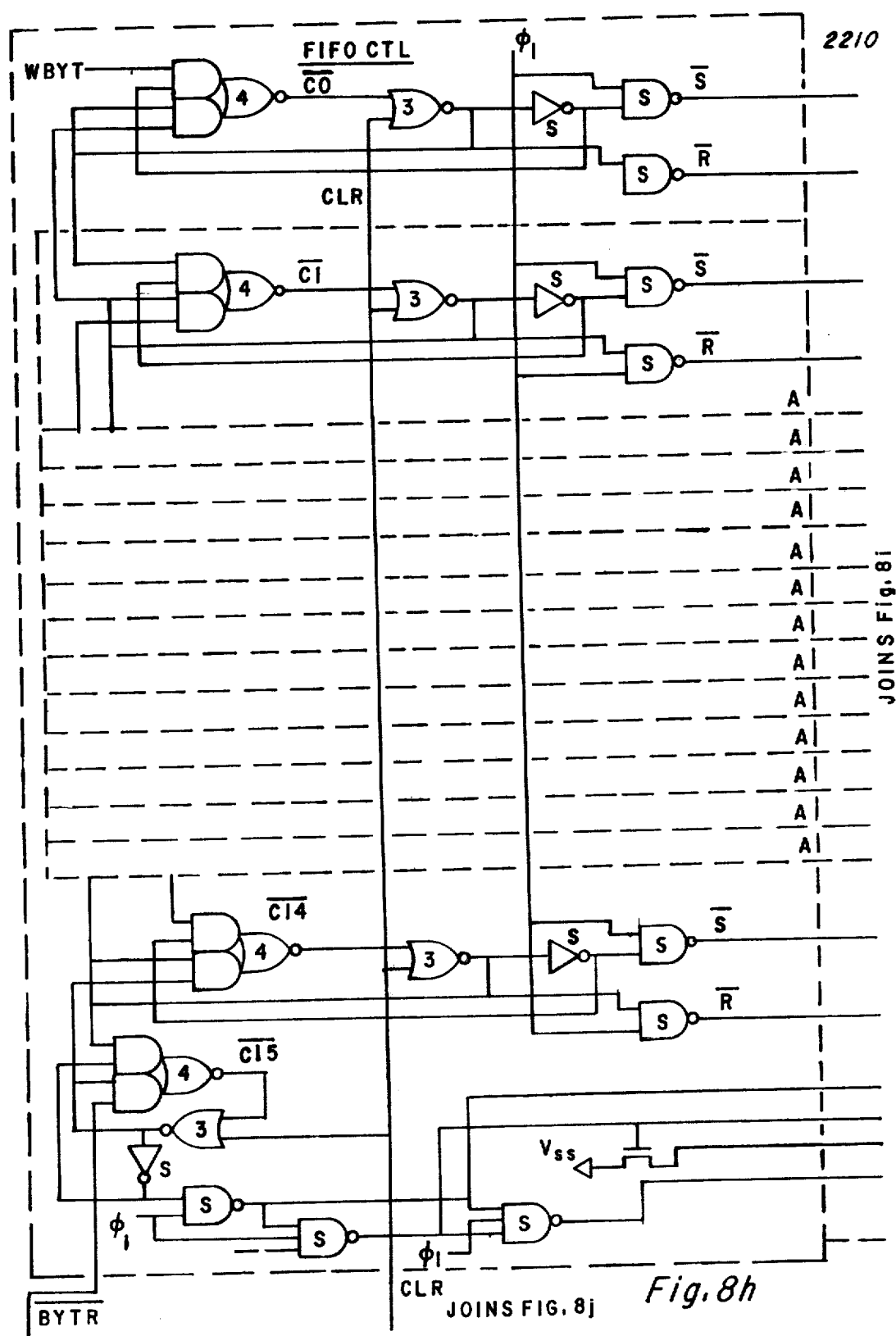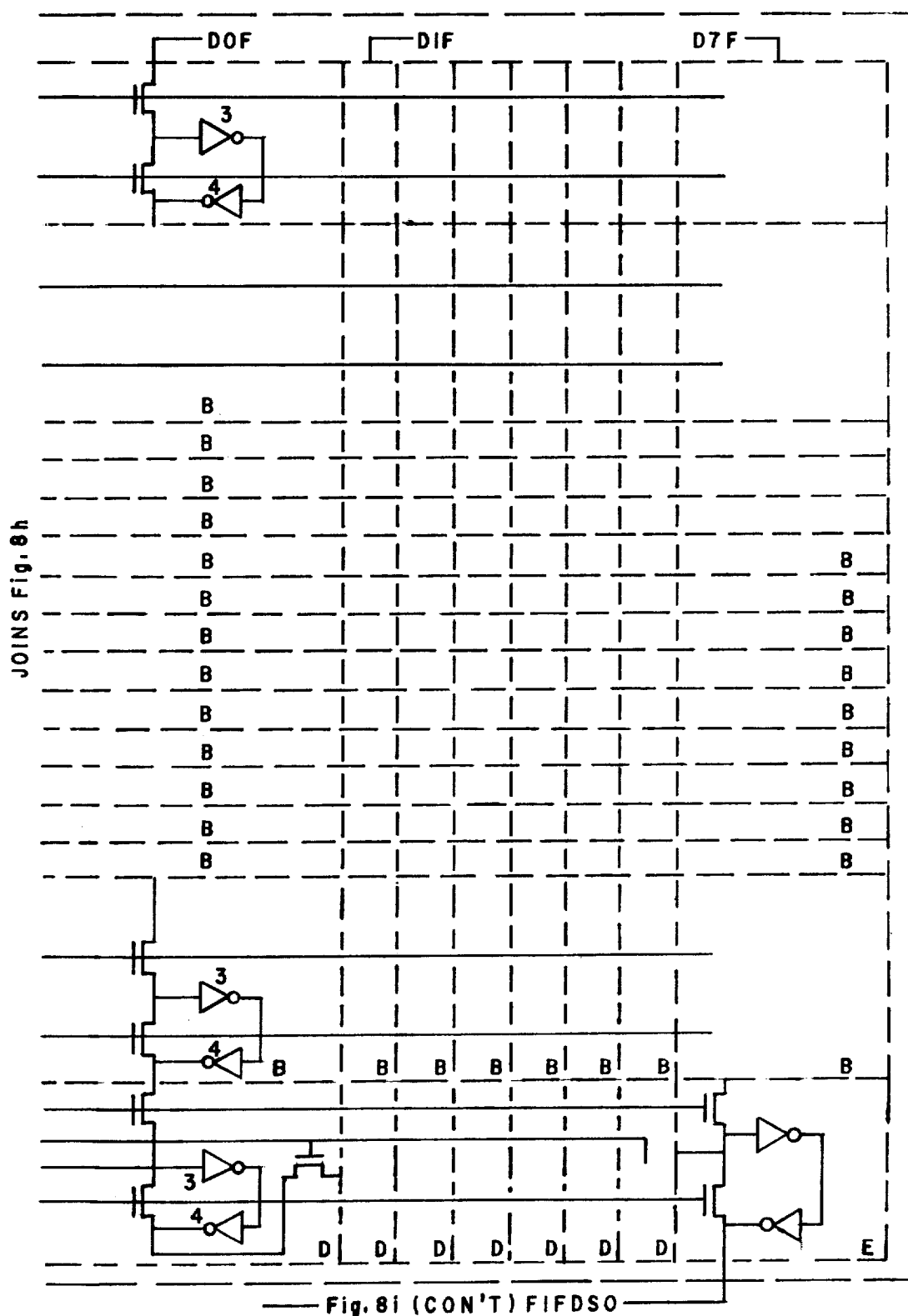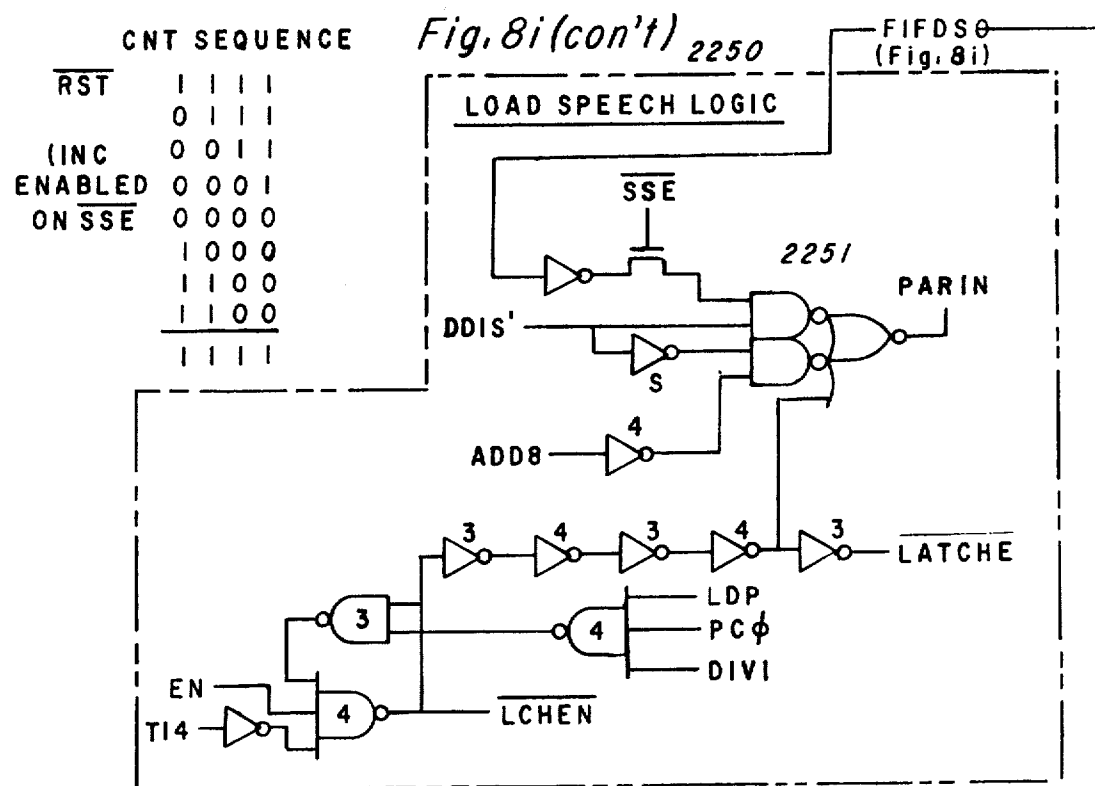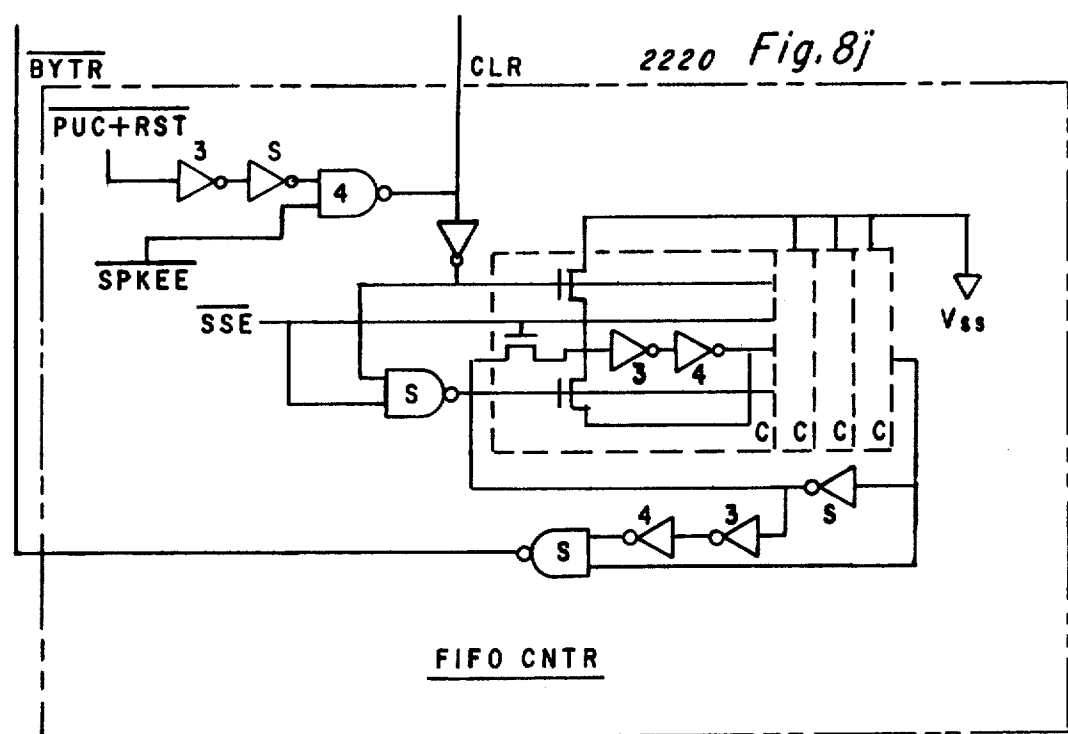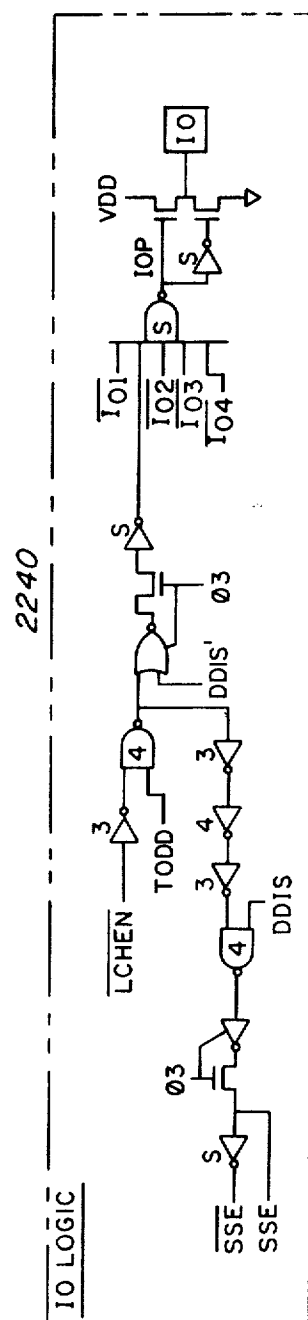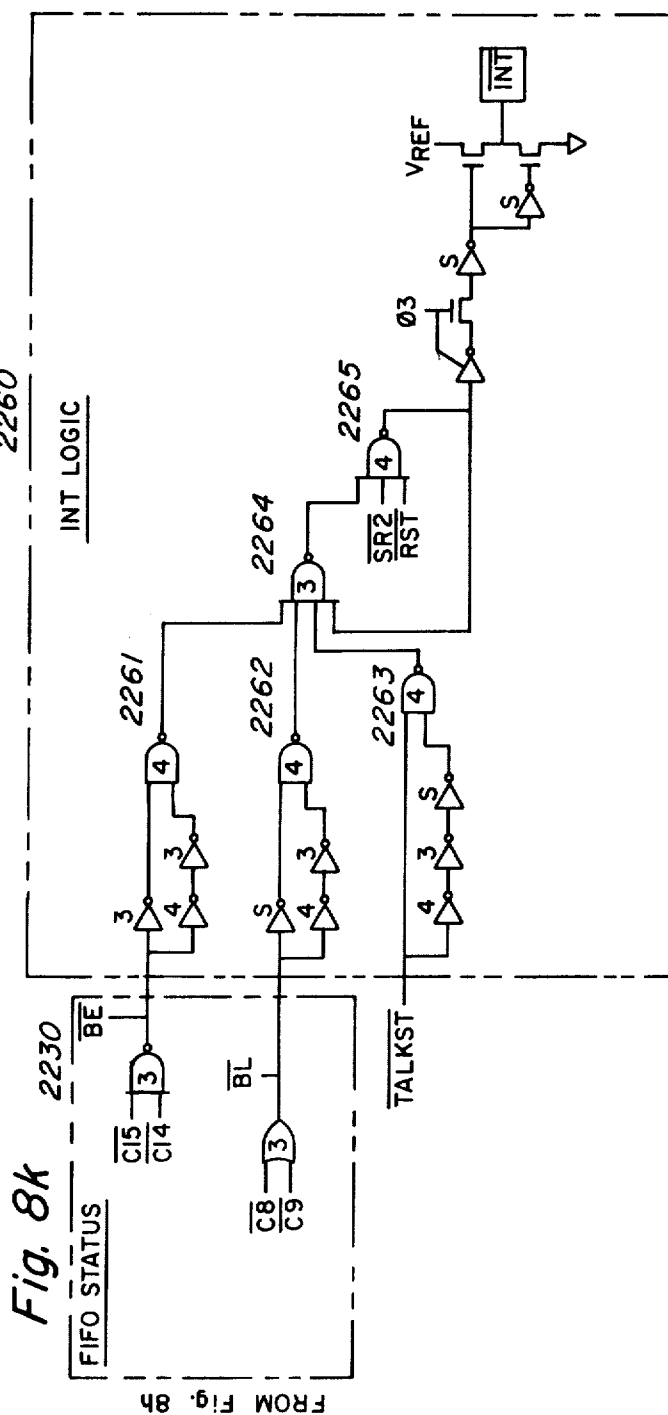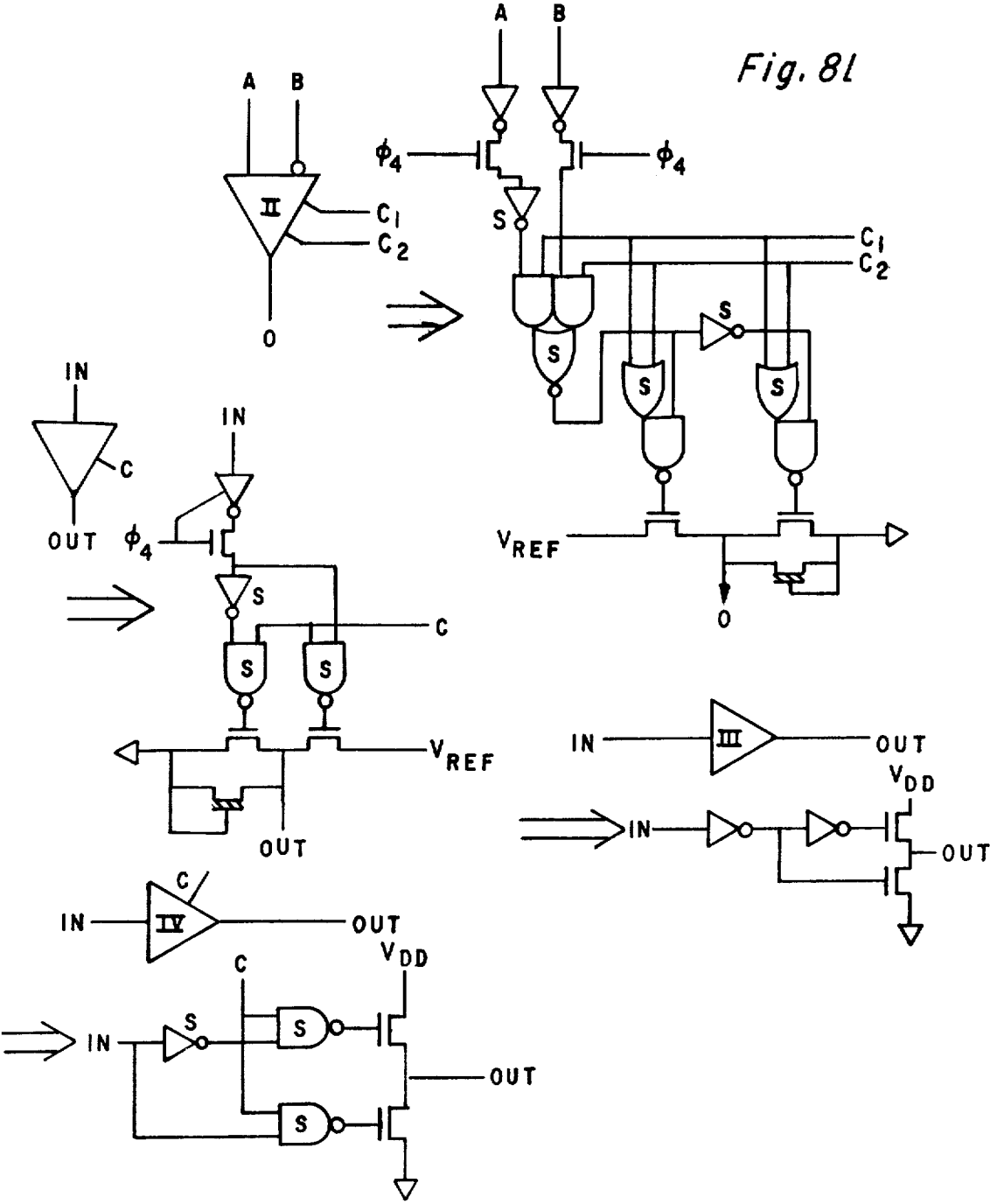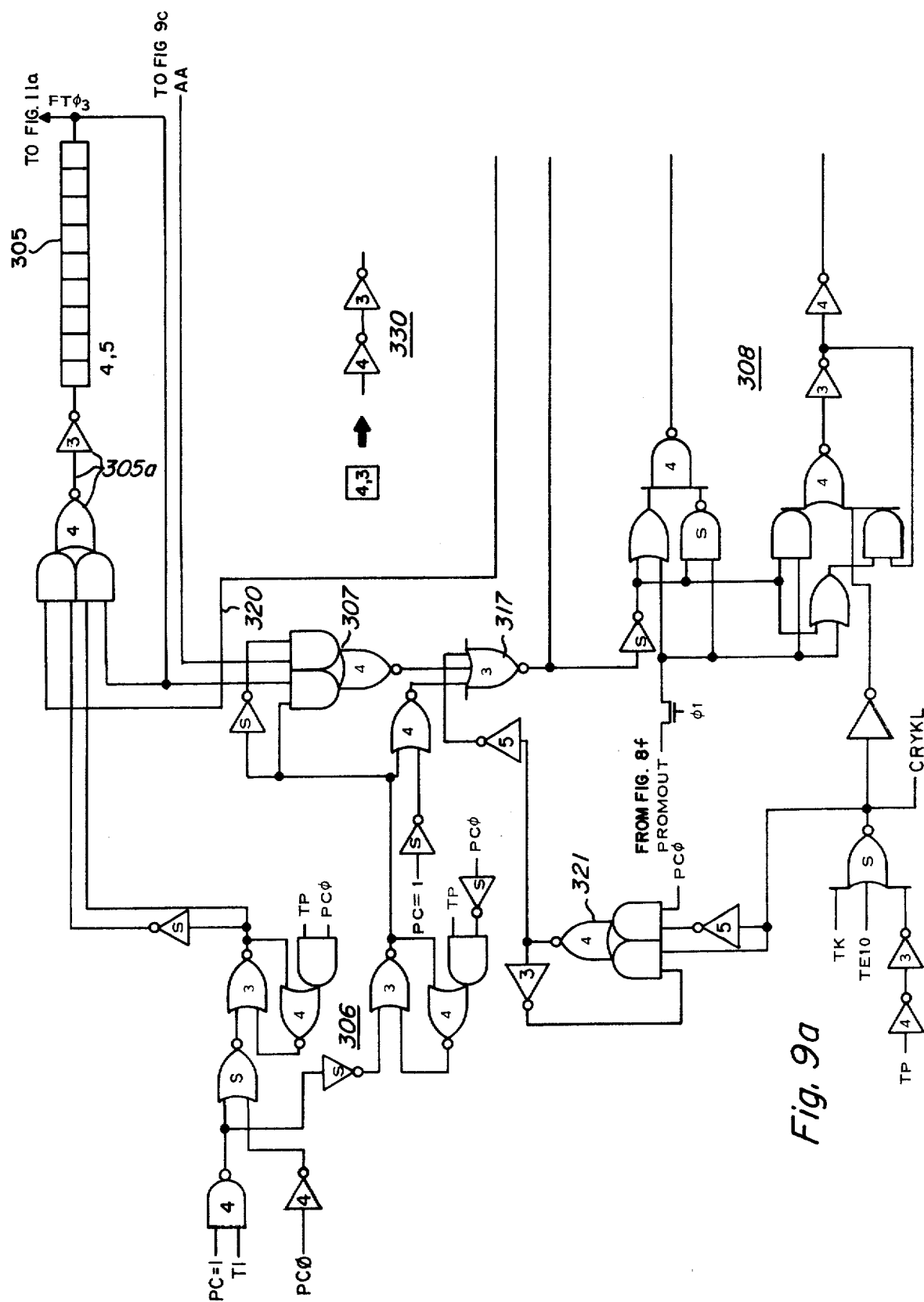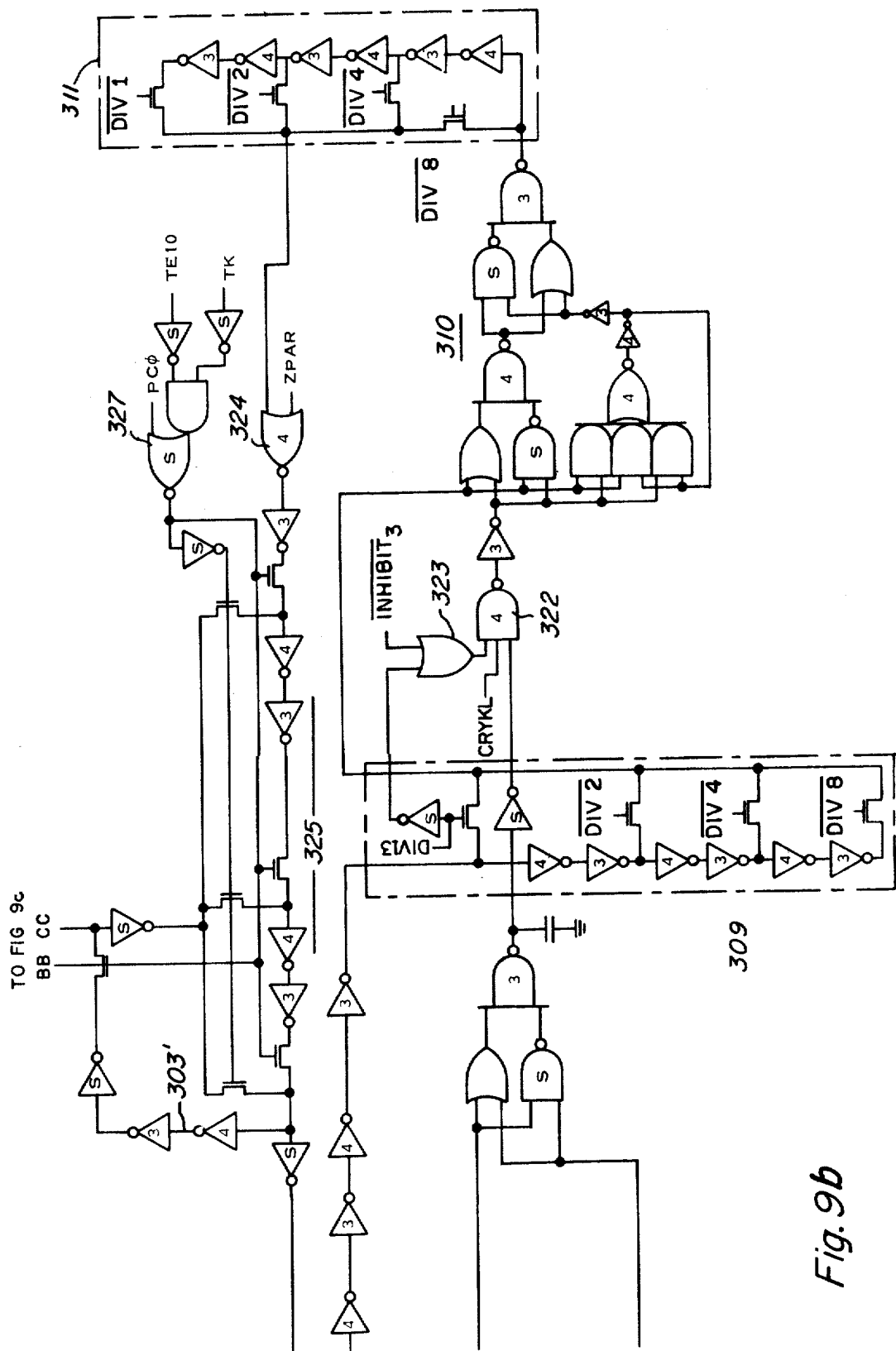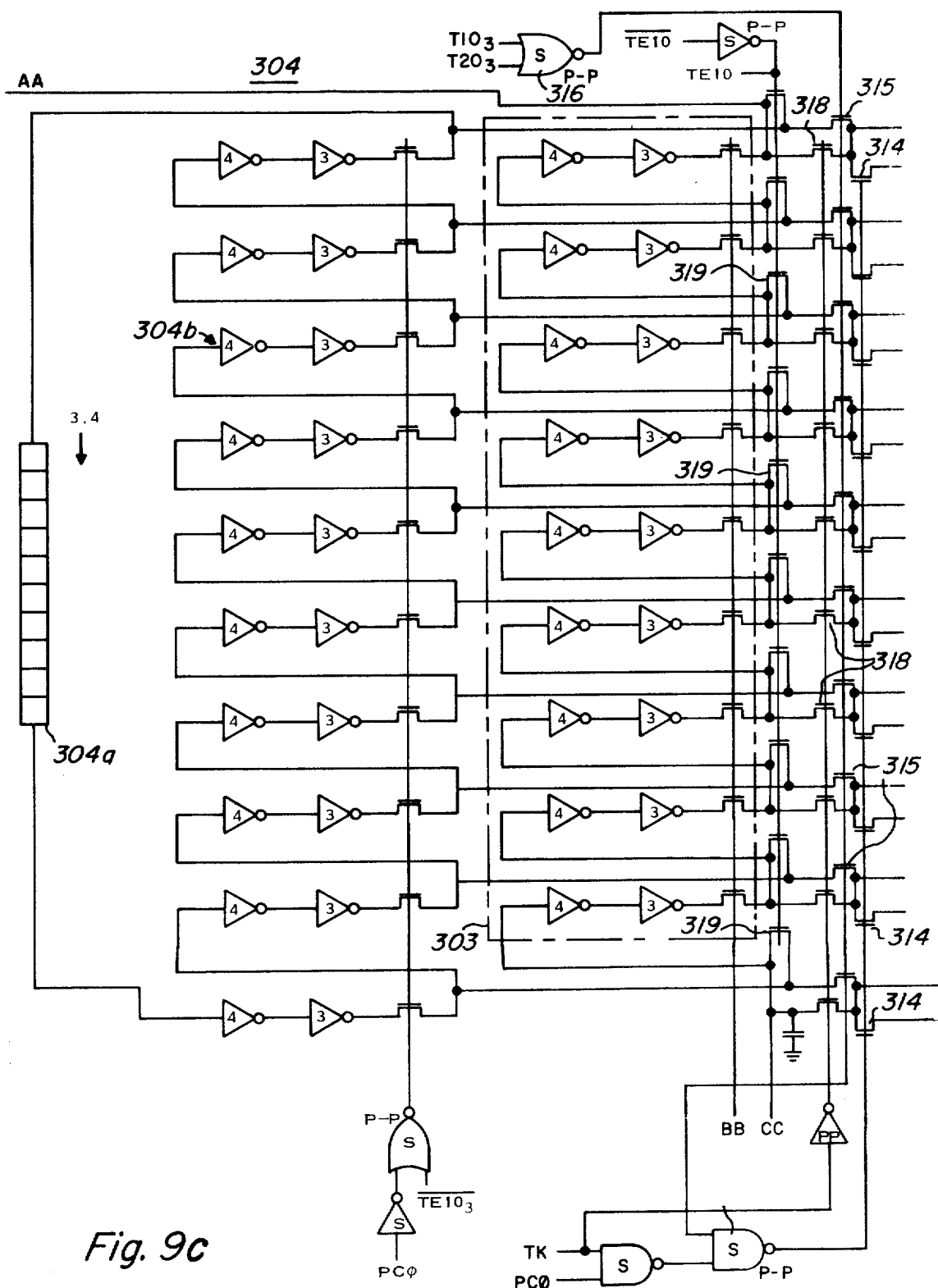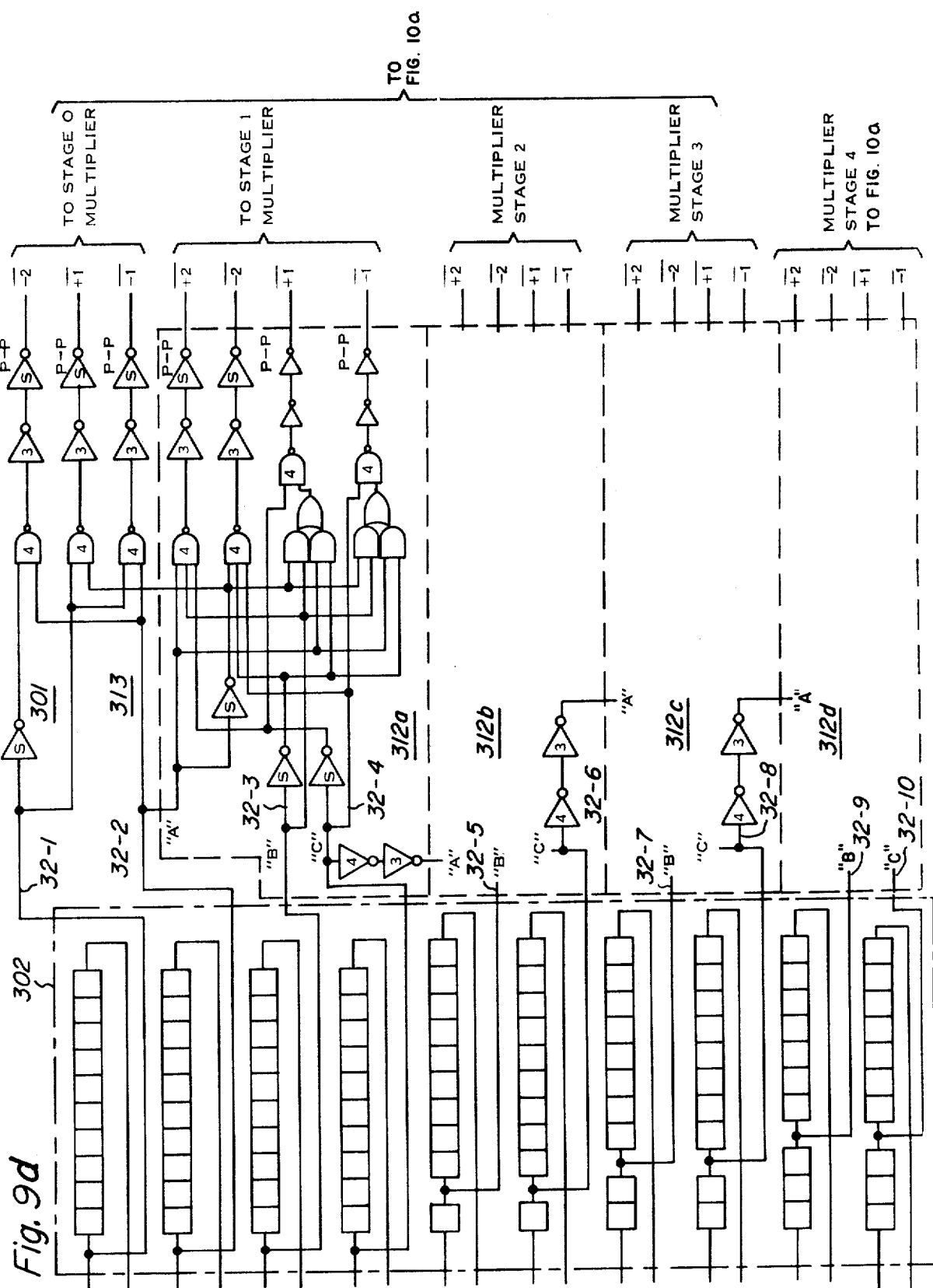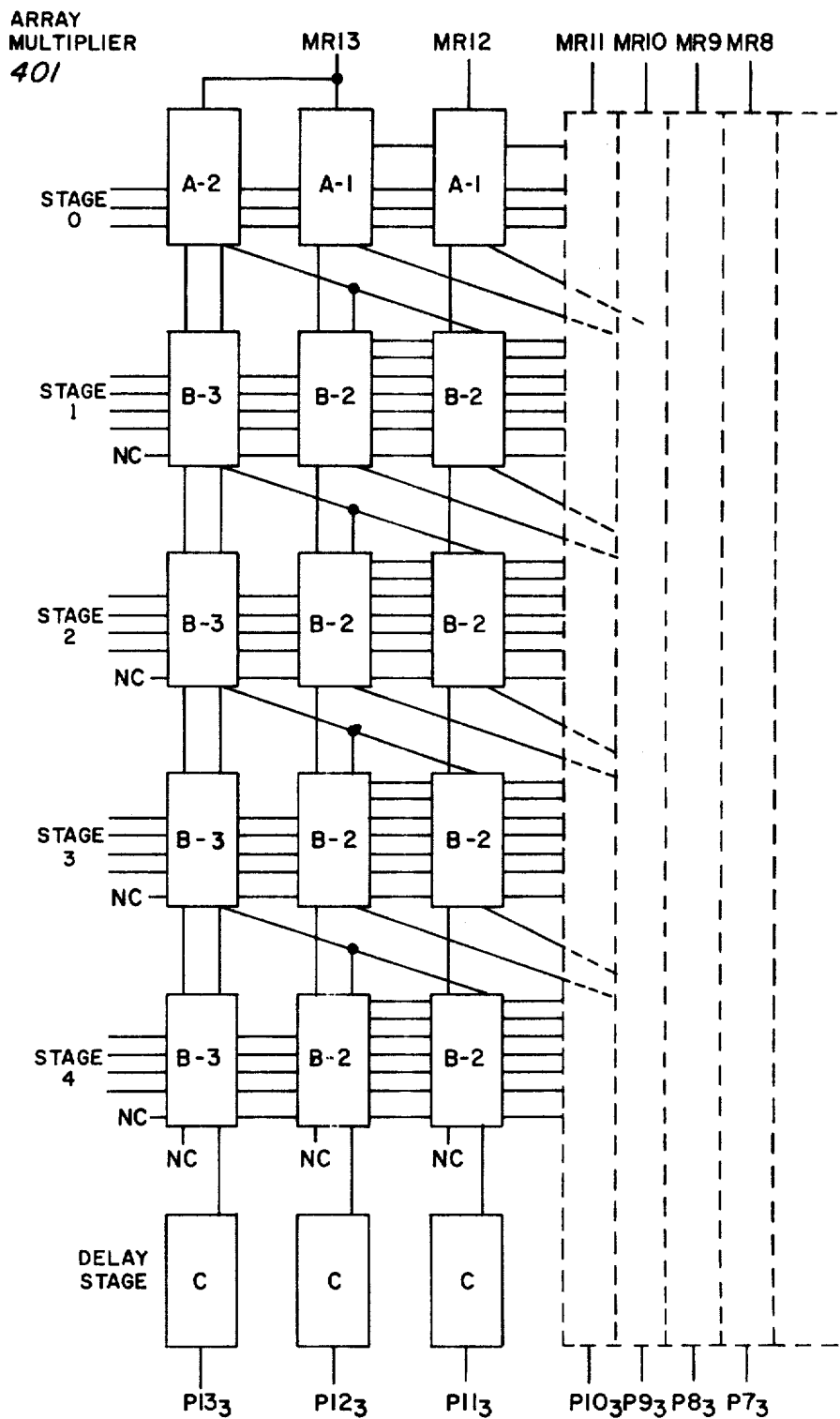
Fig. 10a

Fig. 10b

Fig. 10c

Fig. 11a
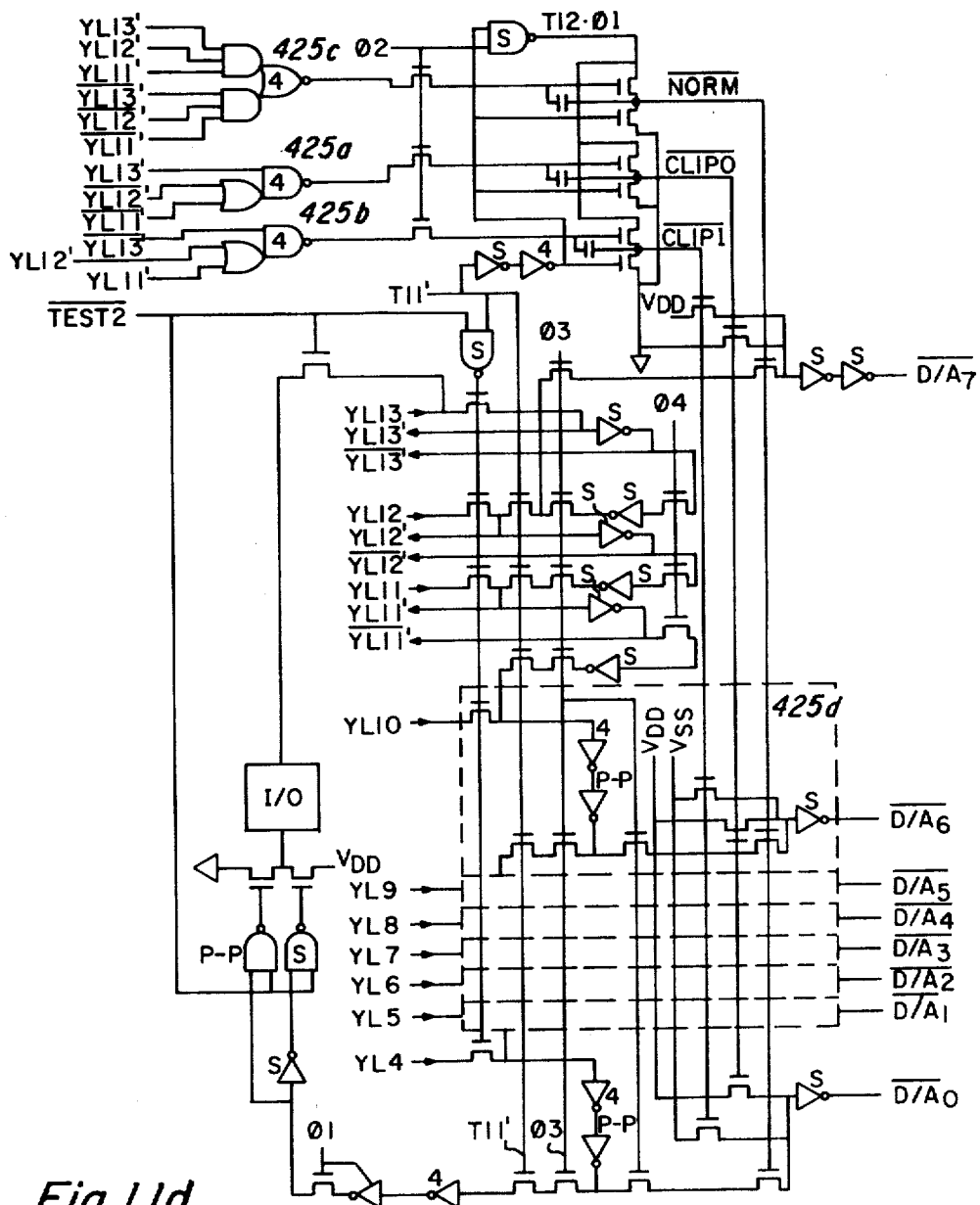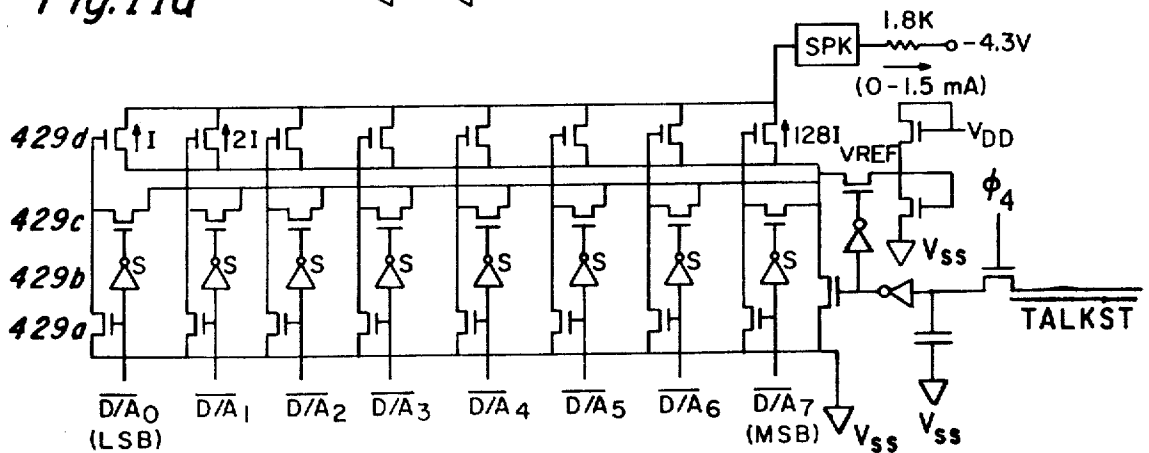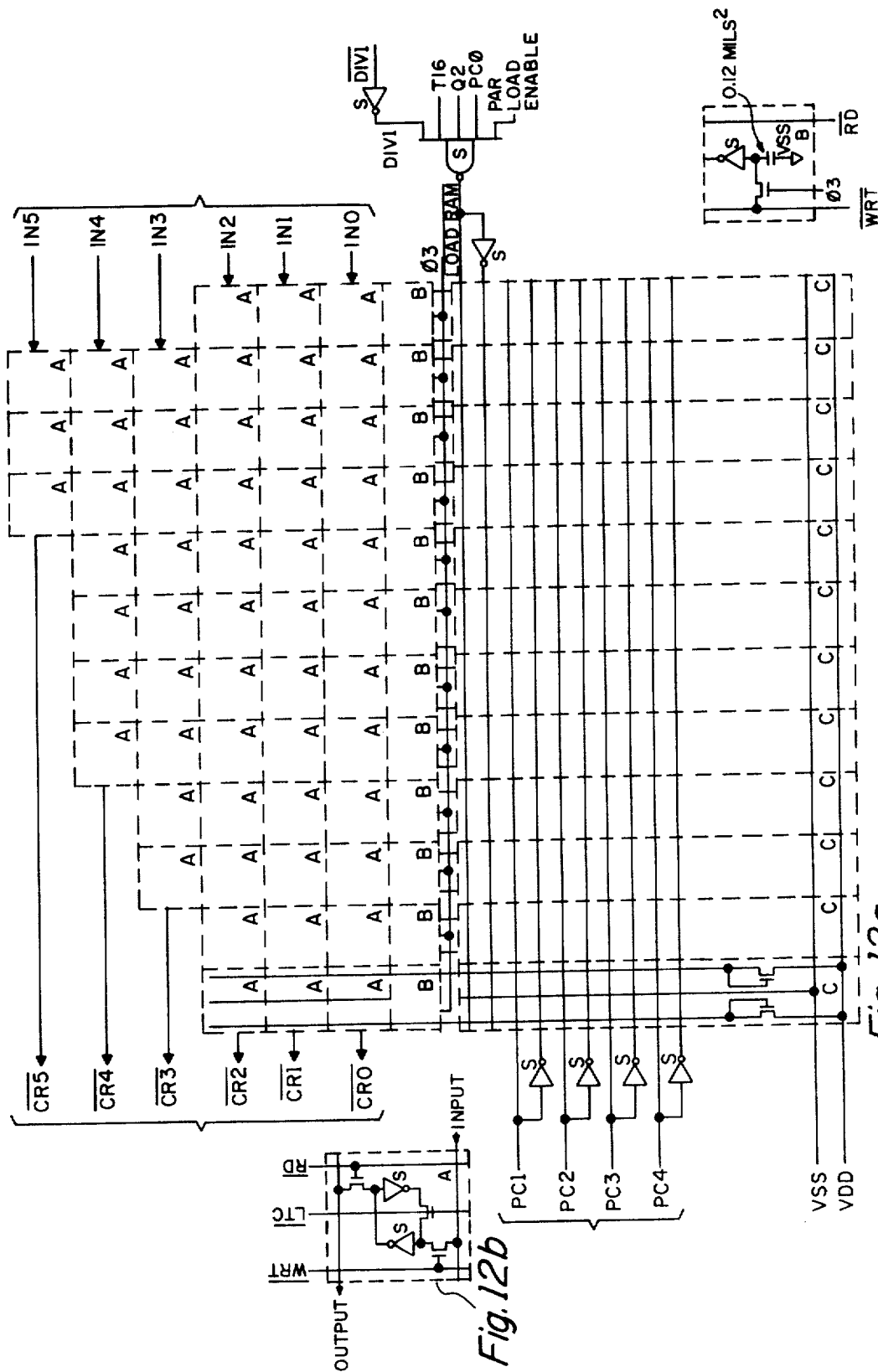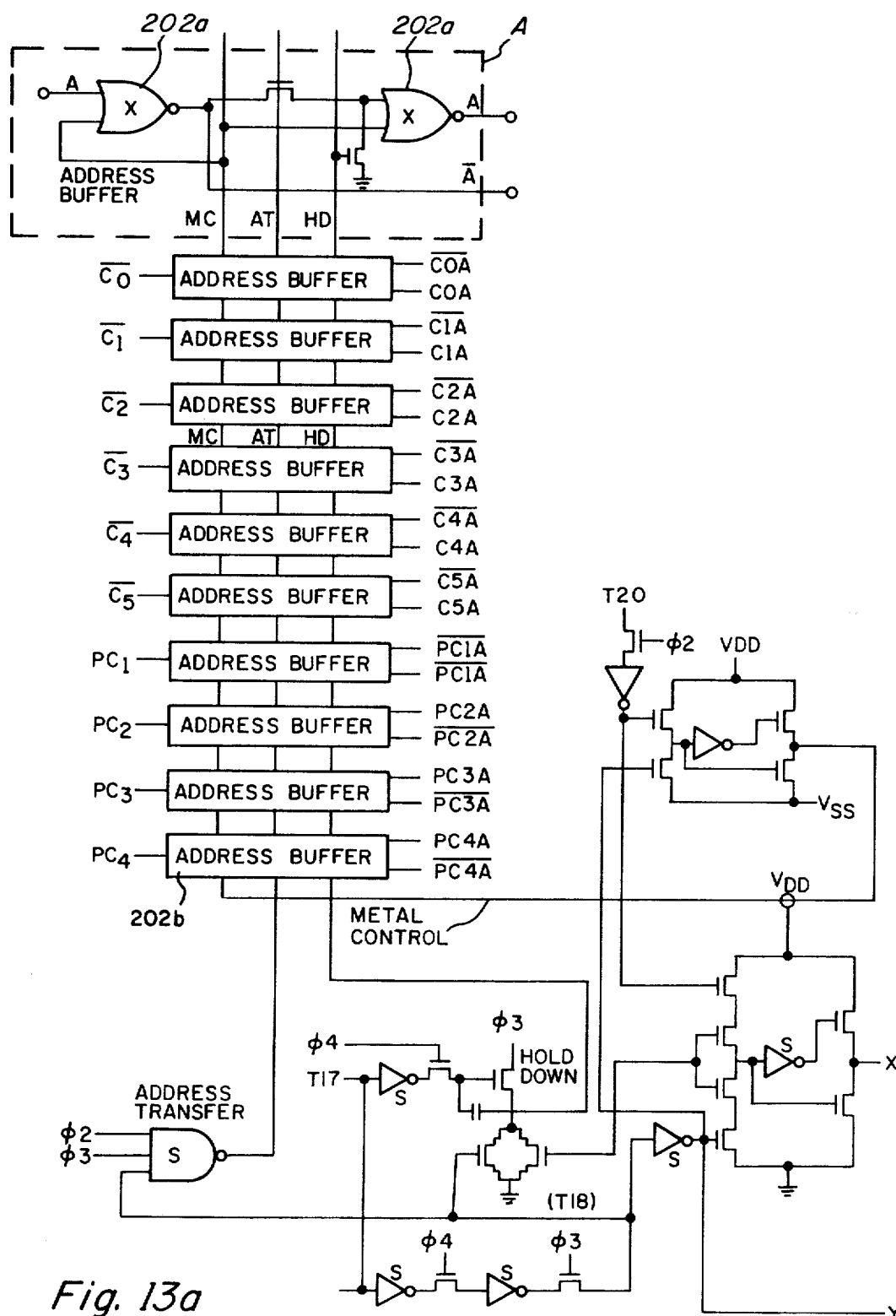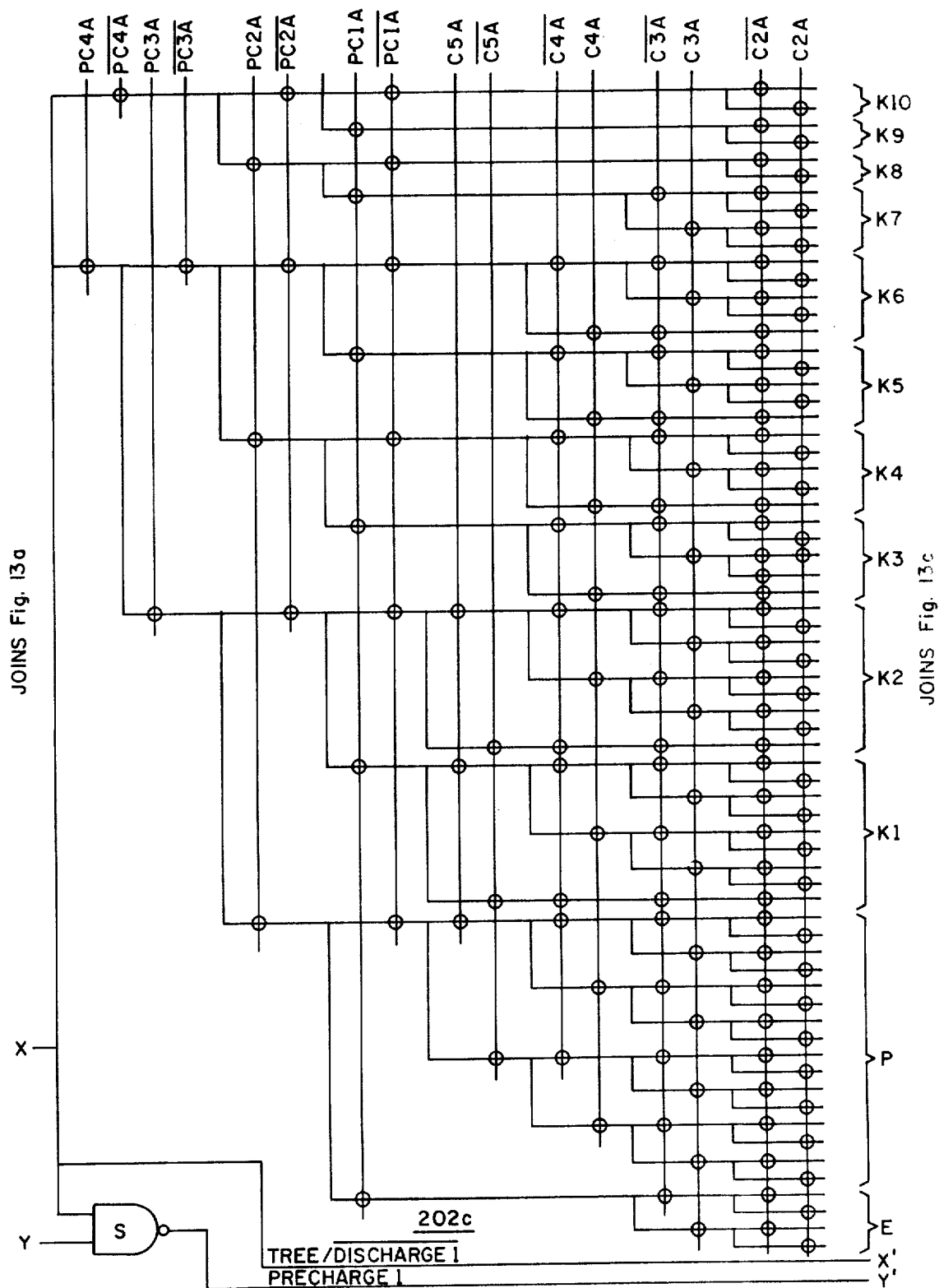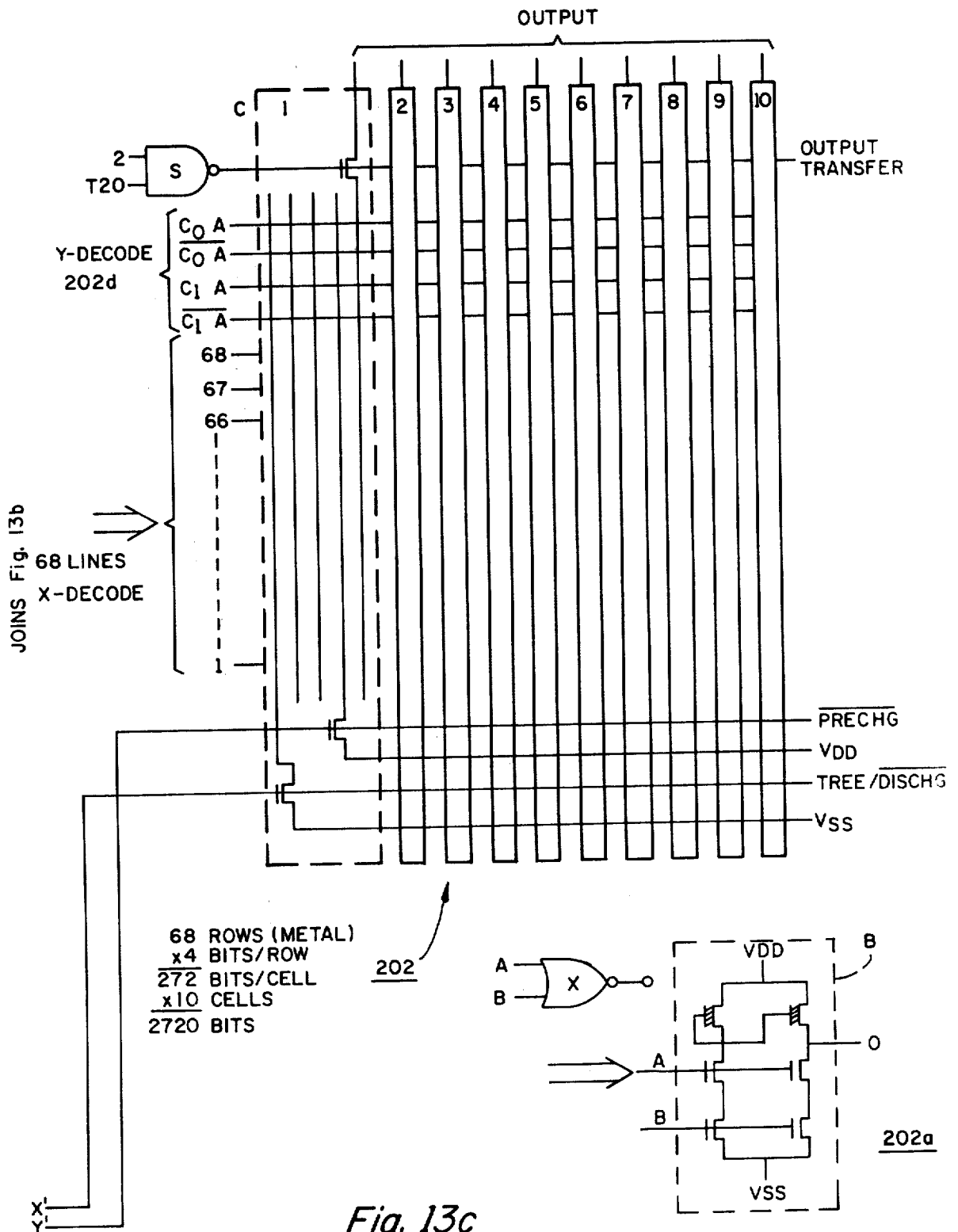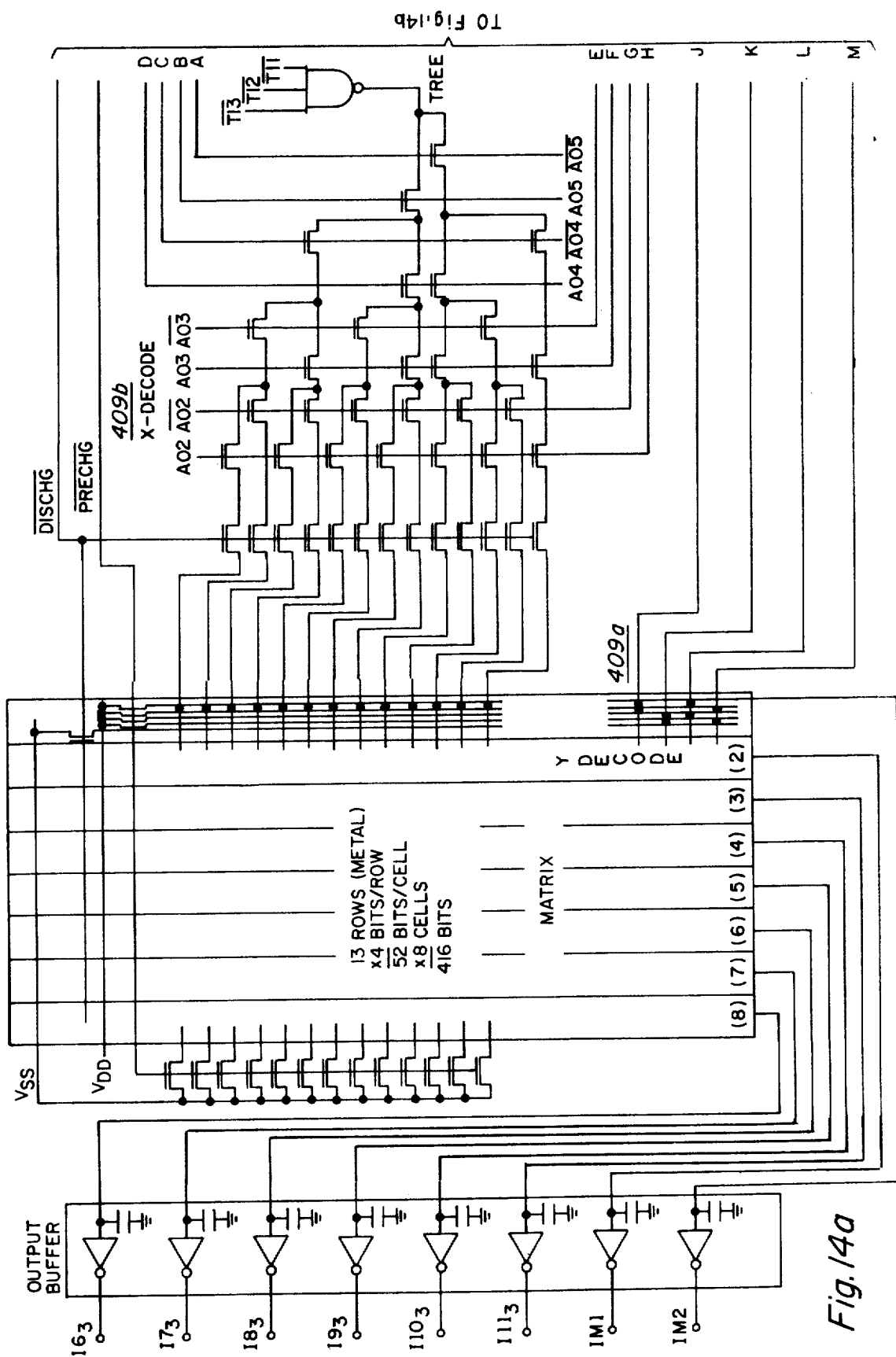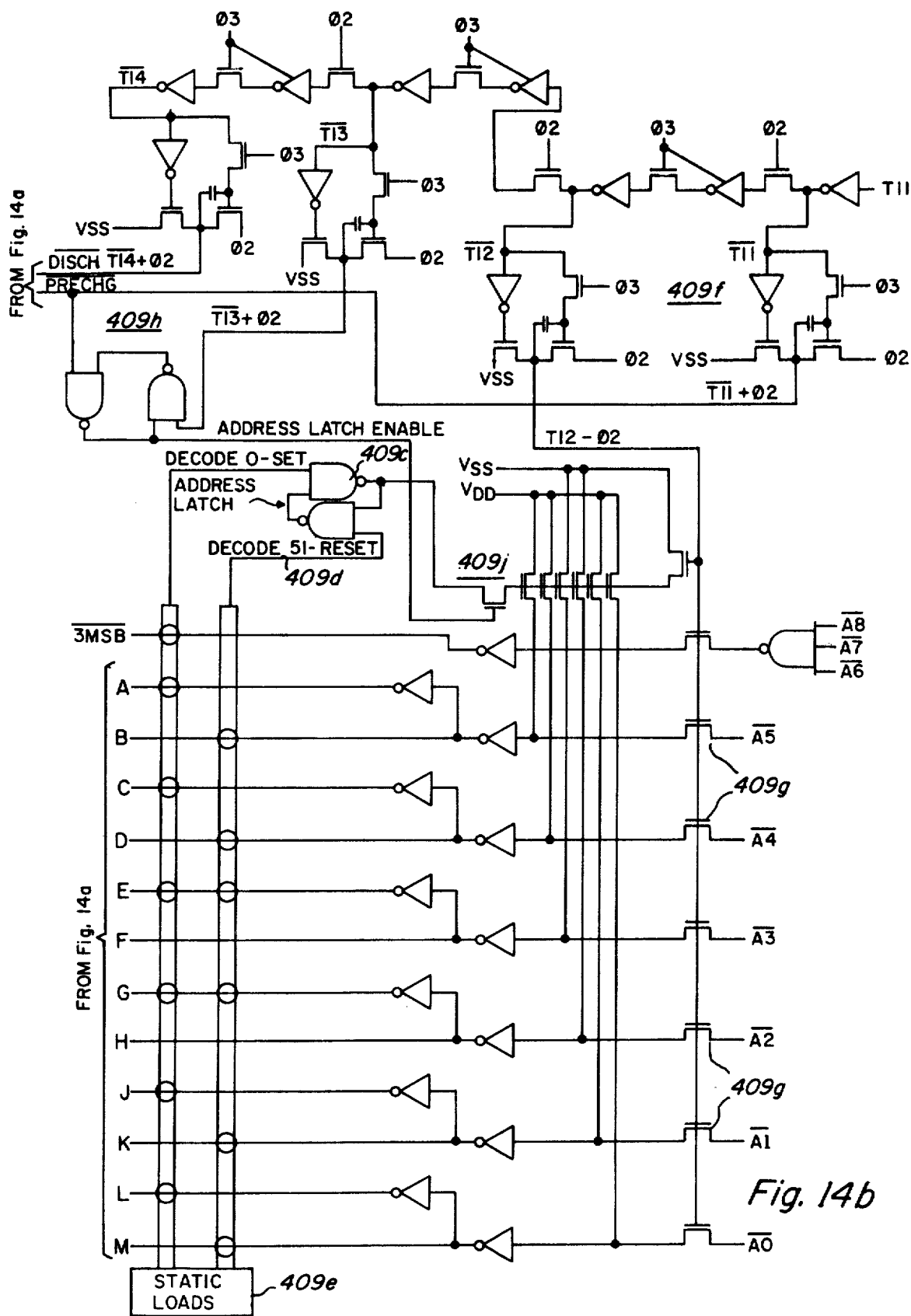
25

TO FIGS. 10a, 10b

Fig. 11b

Fig. 11c

Fig.11d

Fig. 12a

Fig. 12b

Fig. 13a

Fig. 13b

Fig. 13c

Fig. 14a

Fig. 14b

1

CONTROL INTERFACE SYSTEM FOR USE WITH A MEMORY DEVICE EXECUTING VARIABLE LENGTH INSTRUCTIONS

This is a continuation of application Ser. No. 036,928, filed May 7, 1979, now abandoned.

BACKGROUND

This invention relates to a control interface system for use with a memory device executing variable length instructions, such as a speech synthesis device, and more specifically to such a control interface system in which the circuits included therein are implementable in integratable field effect semiconductor devices.

Interface circuits which connect a control device, typically a microprocessor, to a memory device, are known in the prior art. It is typical for the control device to be kept executing "wait" states until the memory device has executed the instruction. An example of this solution may be found on pages 4-15 through 4-18 of Texas Instrument's 9900 Family Systems Design (1st Edition).

Disclosed herein is a speech synthesizer which is preferably controlled by the central processing unit of a home-type computer. It will, of course, be appreciated by those skilled in the art, that any interface between a memory device and a control device wherein the memory device cannot properly respond within the minimum access time determined by the system clock, may incorporate the interface system of this invention. Additionally, it is preferable that the interface between the speech synthesizer and the central processing unit be more efficient that those solutions known in the prior art.

It was, therefore, one object of this invention that the circuits included in a control interface system for use with memory devices be implemented in integratable field effect semiconductor devices.

It was another object of this invention to provide an improved control interface system for controlling a memory device with a control device.

It was yet another object of this invention to provide an improved control interface system for implementing variable length instructions with a control device.

It was yet another object of this invention to provide an improved control interface system for use with a speech synthesis device and a central processing unit, wherein the execution time of which the speech synthesis device is capable is relatively slow as compared to that of the central processing unit.

The foregoing objects are achieved as is now described. A control latch is utilized to release the control device after an instruction has been initiated. A plurality of latches, each set by the conclusion of the execution of a previous instruction, are utilized to reset the control latch. An attempted access of the memory device during the execution of a previous command will result in suspending the operation of the control device until such time as the previous instruction is concluded and the new instruction is initiated.

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 best be understood by reference to the following detailed de-

2

scription of an illustrated embodiment when read in conjunction with the accompanying drawings, wherein:

FIG. 1a is a perspective view of a Speech Module showing the case thereof in an open position for receiving an additional memory unit;

FIG. 1b is a perspective view of the Speech Module similar to FIG. 1a, but showing the case in closed position;

FIG. 1c is a front elevation view of a computer showing the Speech Module of FIGS. 1a and 1b connected thereto;

FIG. 2 is a block diagram of the major components preferably making up the interface system between a computer and the Speech Module;

FIG. 3 is a logic diagram of the Input/Output Circuitry included in the interface system shown in FIG. 2;

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

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

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

FIGS. 7a-7d form a composite logic diagram of the timing circuits of the speech synthesizer;

FIGS. 8a-8l form a composite logic diagram of the speech synthesizer/computer CPU interface logics;

FIGS. 9a-9d form a composite logic diagram of the interpolator logics;

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

FIGS. 11a-11d form a composite logic diagram of the lattice filter and excitation generator of the speech synthesizer;

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

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

FIGS. 14a and 14b form a composite diagram of the chirp ROM.
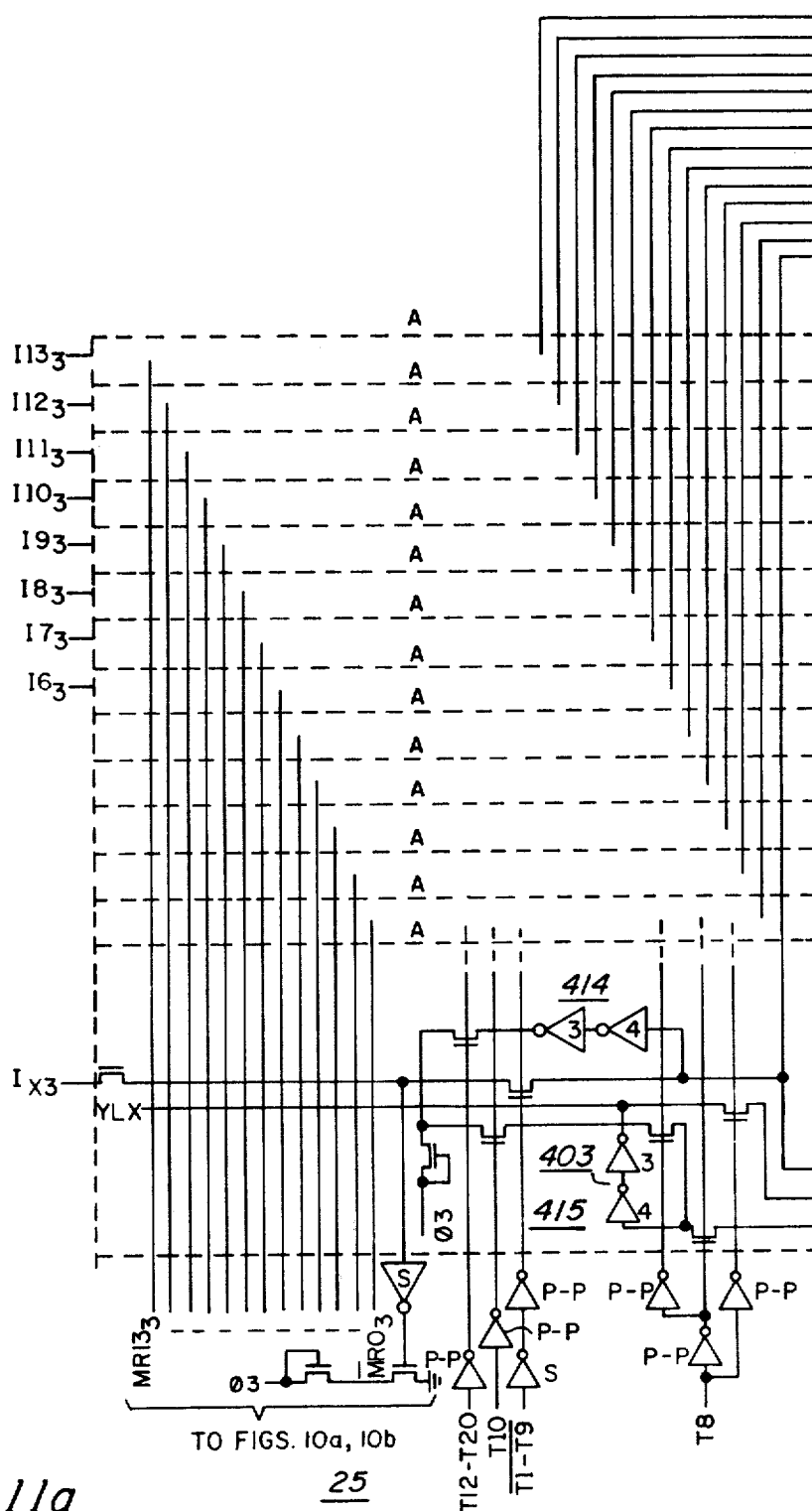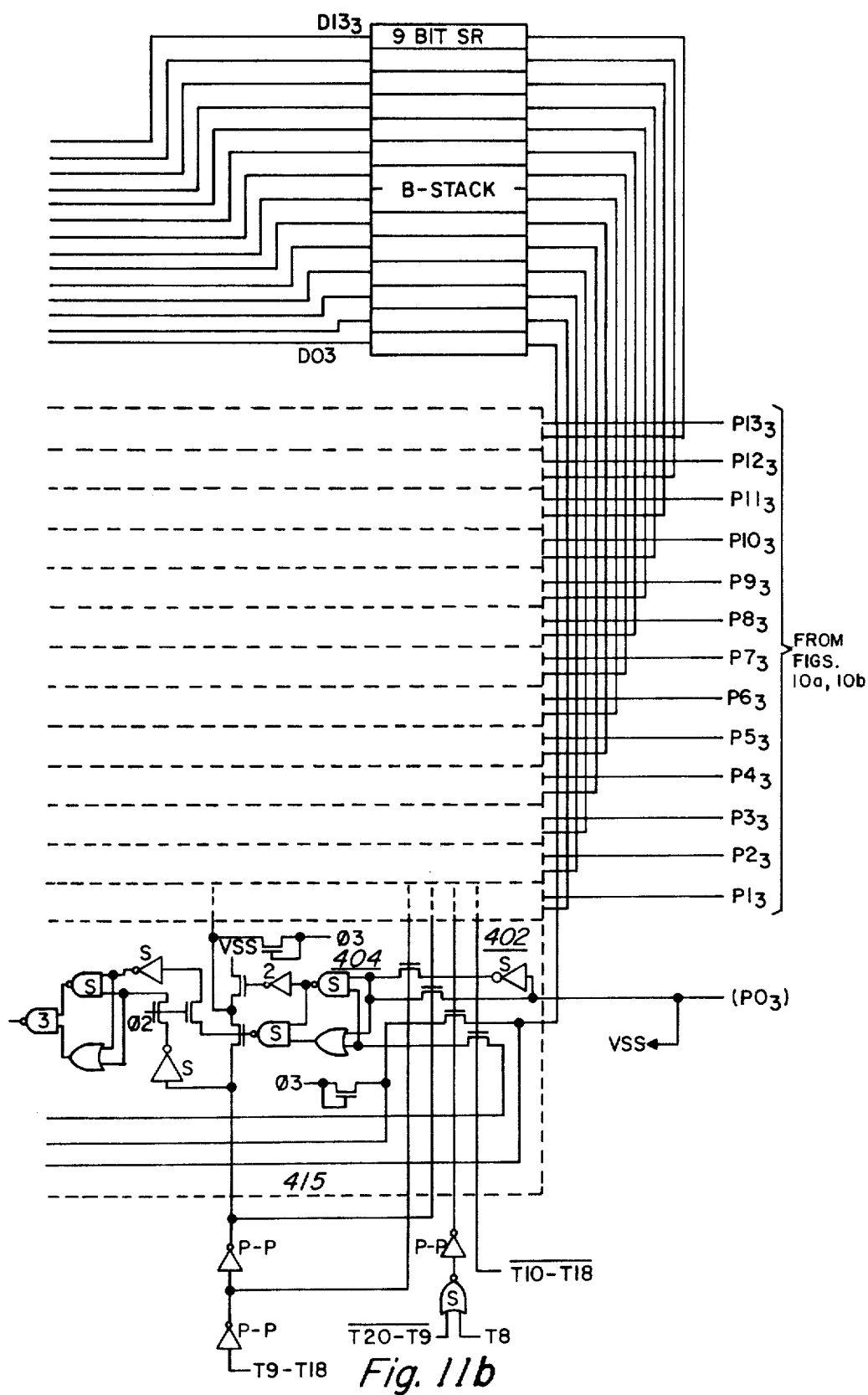
GENERAL DESCRIPTION

FIGS. 1a and 1b are perspective views of a Speech Module of a type which may be connected to a control device providing variable length instructions via an interface system in accordance with the present invention. The Speech Module includes a case 1 (shown in open position for receiving an additional memory unit in FIG. 1a and in closed position in FIG. 1b) which encloses electronic circuits, preferably implemented as integrated circuits (not shown in this figure). Also shown is access slot 2 in FIG. 1a into which additional memory units may be placed to supplement the installed memory circuits. These circuits are coupled through pin connector 3 to a commercial or home-type computer, electronic toys for children, or any product wherein a verbal instructional or informational response is desired. It will, of course, be appreciated by those skilled in the art, that alternative means of connection may be used if desired. FIG. 1c shows an embodiment wherein the Speech Module is connected via pin connector 3 to a control device in the form of a computer 4, which includes a speaker 5. The computer 4 may be a home-type computer having design characteristics specifically applicable to use in the home by operators having limited skills concerning computer use. FIG. 2 shows the major blocks of the speech synthesizer sys-

3

tem, including some blocks within the computer, namely the Central Processing Unit 19, the audio amplifier 20 and the speaker system 5, which are required to operate the Speech Module.

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

MODES OF OPERATION

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

The first mode, the Speak mode, utilizes coded speech parameters contained in a phrase Read-Only Memory (ROM) within the Speech Module. The coded parameters are inputted to the Speech Synthesizer Processor (SSP) chip, where they are decoded and used to construct a time varying model of the vocal tract. This model is used to produce a synthetic speech wave form.

In the second mode of operation, the Speak External mode, the coded speech parameters are provided from an external source, preferably the Central Processing Unit (CPU) of a commercial or home-type computer. The coded speech parameters are inputted through an input buffer to the Speech Synthesizer Processor (SSP) chip, where they are decoded and used to produce synthetic speech.

BLOCK DIAGRAM OF THE SPEECH MODULE

FIG. 2 is a block diagram of the major components making up the disclosed embodiment of a speech synthesizer system. The electronics of the disclosed Speech Module may be divided into three major functional groups, one being the Speech Synthesizer Processor 10, another being the Control Input/Output Circuits package 11 and another being Read-Only-Memories 12A and 12B. In the embodiment disclosed, these major functional groups are each integrated on a separate integrated circuit chip, except for the ROM functional group 12A and 12B which is integrated onto two integrated circuit chips. The coded speech parameters for the desired speech outputs are stored in the ROM functional group 12A and 12B. Additionally, other coded speech parameters may be stored in separate "dictionary modules" shown as read-only-memories 13A and 13B which may be connected to the Speech Module in a manner similar to that described in U.S. patent application Ser. No. 003,449, filed Jan. 15, 1979, which is assigned to the assignee of this invention. These additional read-only-memories 13A and 13B are depicted by dashed lines, since they will be plugged into the Speech Module by the operator, rather than normally packaged with the system.

The Speech Synthesizer Processor 10 is interconnected with the read-only-memories 12A and 12B via data path 15 and is connected to the input/output bus 18 via data path 16 and control input/output circuitry package 11. In a preferred embodiment, addresses of coded speech parameters are transmitted by a Central Processing Unit (CPU) 19 of a home or commercial type computer, to the Read-Only-Memories 12A and

4

12B by Speech Synthesizer Processor 10 because, as will be seen, Speech Synthesizer Processor 10 is preferably equipped with buffers capable of addressing a plurality of Read-Only-Memories. Of course, a Central Processing Unit with appropriately sized buffers could transmit addresses to a plurality of Read-Only-Memories and thus, in certain embodiments, it may be desirable to also couple the input from a Central Processing Unit directly to the Read-Only-Memories.

As will be seen, the Speech Synthesizer processor 10 synthesizes human speech or other sounds according to frames of data stored in Read-Only-Memories 12A and 12B or 13A and 13B. The Speech Synthesizer Processor employs a parameter interpolator of the type described in U.S. patent application, Ser. No. 901,394, filed Apr. 28, 1978, now U.S. Pat. No. 4,189,779 issued Feb. 19, 1980, which is hereby incorporated herein by reference. The Speech Synthesizer Processor 10 also utilizes a digital filter of the type described in U.S. patent application Ser. No. 905,328, filed May 12, 1978, now U.S. Pat. No. 4,209,844 issued June 24, 1980. U.S. Pat. No. 4,209,844 is hereby incorporated herein by reference. As will be seen, the Speech Synthesizer Processor 10 includes a digital-to-analog "D to A" converter for converting the digital output of the digital filter to analog signals capable of driving a sound amplifier and speaker system. Speech Synthesizer Processor 10 also includes timing, control and data storage, and data compression systems which will be subsequently described in detail.

CONTROL INPUT OUTPUT LOGIC CIRCUITRY

FIG. 3 shows the control input/output circuitry package 11. The control input/output circuitry 11 comprises three, 3-input NAND gates 31, 32 and 33, with open collectors. These logic gates may be similar to the SN74LS10 chip manufactured by Texas Instruments Incorporated of Dallas, Tex. Two of the inputs to the NAND gate 31 are connected to Vss. The third input is address bit 15 (ADD15) from the Central Processing Unit 19. Since two of its inputs are always high, NAND gate 31 effectively acts as an inverter and its output is $\overline{\text{ADD15}}$. NAND gate 32 has as its inputs, the $\overline{\text{ADD15}}$ from the output of NAND gate 31, the speech block enable signal SBE, and address bit 5 (ADD5). Therefore, the output of NAND gate 32 is a function of SBE, ADD5 and $\overline{\text{ADD15}}$. This output is designated WRITE SELECT, ($\overline{\text{WS}}$) and is coupled to the Speech Synthesizer Processor 10. A WRITE SELECT command from the centrl processing unit 19 allows the Speech Module to accept 8 bits of data via the bi-directional data bus 17. NAND gate 33 has as its inputs $\overline{\text{ADD15}}$ from the output of NAND gate 31, the speech block enable signal (SBE) and $\overline{\text{ADD5}}$, coupled from the output of NAND gate 32. Therefore, the output of NAND gate 33 is a function of SBE, $\overline{\text{ADD5}}$ and $\overline{\text{ADD15}}$. This output is designated READ SELECT ($\overline{\text{RS}}$) and is coupled to the Speech Synthesizer Processor 10. A READ SELECT command from the central processing unit 19 allows the Speech Module to output 8 units of data via the bi-directional data bus 17 or, causes the Speech Module to generate certain status signals at preselected points along data bus 17.

Additionally, the Speech Synthesizer Processor 10 can generate an interrupt signal (INT) which indicates to the Central Processing Unit 19 some change in Speech Synthesizer Processor status which may require Central Processing Unit attention. The particular status

changes which can cause an interrupt (INT) signal to be generated will be discussed at length herein. Inverter **34** inverts the READY signal at its input to provide the READY signal for the Central Processing Unit **19**. When READY is high, the Central Processing Unit **19** is locked to the speech synthesizer **10**.

SPEECH SYNTHESIZER BLOCK DIAGRAM

FIGS. **4a** and **4b** form a composite block diagram of the Speech Synthesizer Processor **10**. Speech Synthesizer Processor **10** is shown as having 6 major functional blocks, all but one of which are shown in greater detail in block diagram form in FIGS. **4a** and **4b**. The 6 major functional blocks are timing logic **20**; ROM-CPU interface logic **21**; parameter loading, storage and decoding logic **22**; parameter interpolator **23**; filter and excitation generator **24** and digital-to-analog conversion and output section **25**. Subsequently, these major functional blocks will be described in detail with respect to FIGS. **5-14b**.

ROM/CPU Interface Logic

Referring again to FIGS. **4a** and **4b**, ROM/CPU interface logic **21** couples the speech synthesizer **10** to Read-Only-Memories **12A** and **12B** and to the Central Processing Unit **19** (not shown). The 8 bit bi-directional data bus **17** (D0–D7) is coupled, in this embodiment, to the Central Processing Unit **19** and to the inputs of FIFO buffer **2215**, while the address **1-8** (ADD-1–ADD8) and instruction 0-1 (I_0–I_1) pins are connected to ROMS **12A** and **12B** (as well as ROMS **13A** and **13B**, if used). ROM/CPU interface logic **21** sends address information from the Central Processing Unit **19** to the Read-Only-Memories **12A** and **12B** through address register **213** to address pins ADD1–ADD8. Command register **210** stores a 3 bit command from the central processing unit **19**, which is decoded by command decoder **211**. Command decoder **211** is responsive to 6 commands: SPEAK (SPK), for causing the synthesizer **10** to access data from the Read-Only-Memory **12A** or **12B** and speak in response thereto; a RESET (RST) command for resetting the synthesizer to zero; a LOAD ADDRESS (LA) command, wherein 4 bits are received from the Central Processing Unit **19** at the D4–D7 pins and transferred to the Read-Only-Memories **12A** and **12B** as address digits via address register **213** and the ADD1–ADD8 pins; a READ AND BRANCH (RB) command which causes the Read-Only-Memory **12A** or **12B** to take the contents of the present and subsequent addresses and use it for a branch address; a READ BYTE (RDBY) command which allows the Central Processing Unit **19** to access data stored in the Read-Only-Memory **12A** or **12B** via address pin ADD8 and data input register **212**; and a SPEAK EXTERNAL (SPKEXT) command which causes the speak external logic circuit **253** to generate a DECODER DISABLE (DDIS) signal, which disables command decoder **211**, and allows the Central Processing Unit **19** to input 8 bits of data into FIFO buffer **2215** via inputs D0–D7. Once synthesizer **10** has commenced speaking in response to a SPK command, it continues speaking until ROM interface logic **21** encounters an RST command, or gate **207**, (see FIG. **8f**) detects an "energy equal to 15" code and resets talk latch **216** in response thereto. Once synthesizer **10** has commenced speaking in response to a SPKEXT command, it continues speaking until gate **207** detects an "energy equal to 15" code or a buffer empty (BE) command is generated

by FIFO status logic **2230** (see FIG. **8k**) 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, phrases or sentences. The LA, RB and RDBY commands are decoded by command decoder **211** and are re-encoded via ROM control logic **217** and transmitted to the Read-Only-Memories **12A** and **12B** via the instruction pins I_0 and I_1.

Talk latch **216** is set in response to a decoded SPK or SPKEXT 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; (3) by an "energy equals 15" code in a frame of speech data or (4) by a BE command from FIFO status logic **2230**. The TALKD output is a delayed output to permit all speech 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 7 bit long parameter input register **205** which receives serial data from the Read-Only-Memories **12A** and **12B** via Load Speech logic **2250** (see FIG. **8i** con't) from gate **2251**, which has as its input the data from pin ADD8, in response to an RDBY command outputted to the selected Read-Only-Memory **12A** and **12B** via the instruction pins I_0 and I_1. 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 random access memory (RAM) **203**, in a coded format, where the frame is temporarily stored. Each of the coded parameters stored in random access memory (RAM) **203** is converted to a 10 bit parameter by parameter read-only-memory **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. **8a–8l**. The condition decoders and latches **208**, as well as various timing signals are used to control the 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 random access memory **203**.

PARAMETER INTERPOLATOR

The parameters in parameter output register **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 factor are applied via recoding logic **301** to an array multiplier **401** in the filter and excitation generator **24**. As will be seen, however, when a new parameter is loaded into parame-

7

ter output register 201' it is not immediately inserted into K-stack 302 or E10 loop 304 or pitch register 305' but rather the corresponding value in K-stack 302, E10 loop 304 or pitch register 305 goes through 8 interpolation cycles during which a portion of the difference between the present value in the K-stack 302, E10 loop 304 or pitch register 305 and the target value of that parameter in parameter output register 201 is added to the present value in K-stack 302, E10 loop 304 or pitch 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 delay circuit 309. As will be seen, delay circuit 309 may provide anywhere from zero delay to 3 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 particular parameter in the 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 the selector 307, delay circuits 309 and 311, adder 310 and subtractor 308 is constant. By controlling the delays in delay circuits 309 and 311, either all, one half, one fourth or one eighth 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 I, a relatively smooth 8-step parameter interpolation is accomplished.

U.S. Pat. No. 4,209,844 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. Pat. No. 4,209,844) are periodically exchanged. In parameter interpolator 23, speech coefficients K1-K9 are likewise stored in K-stack 302, until they are updated, whereas the energy parameter and the K10 coefficient effectively exchange places in the 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 into the appropriate location in the K-stack 302. KE10 transfer register 303 is either loaded with the K10 coefficient 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, recording logic 301 preferably performs a Booth's algorithm on the data from K-stack 302, before such data is applied to array multiplier 401. Recording logic 301 thereby permits the size of the array multiplier 401 to be reduced compared to the array multiplier described in U.S. Pat. No. 4,209,844.

8

FILTER AND EXCITATION GENERATOR

The filter and 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 coupled to a delay stack 406 and a multiplier multiplexer 415. The output of the delay stack 406 is applied as an input to the summer multiplexer 402 and to Y latch 403. The output of Y latch 403 is coupled to an input of multiplier multiplexer 415 along with truncation logic 425. The output of multiplier multiplexer 415 is applied as an input to array multiplier 401. As will be seen, filter and excitation generator 24 makes use of the digital filter described in U.S. Pat. No. 4,209,844. Various minor interconnections are not shown in FIG. 4b for the sake of clarity, but which will be described with reference to FIGS. 10a-10c and 11a-11d. The arrangement of the foregoing elements generally agrees with the arrangement shown in FIG. 7 of U.S. Pat. No. 4,209,844; thus, array multiplier 401 corresponds to element 30', summer multiplexer 402 corresponds to elements 37B', 37C' and 37D', gates 414 (FIGS. 11a-11d) correspond to element 33', delay stack 406 corresponds to elements 34' and 35', Y latch 403 corresponds to element 36' and multiplier multiplexer 415 corresponds to elements 38A', 38B', 38C' and 38D'.

The voiced excitation data is supplied from unvoiced/voice gate 408. As will be subsequently described in greater detail, the parameters inserted into parameter input register 205 are supplied in a compressed data format. According to the data compression scheme used, when the coded pitch parameter is equal to 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. 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. Pat. No. 4,209,844, the voiced excitation signal 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). The 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 latch 410. Beginning at address zero and extending through approximately fifty addresses is the chirp function in chirp ROM 409. Counter latch 410 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 unvoiced 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 with respect to the speed synthesizer 10. Also depicted are the timing relationships with respect to the time new frames of data are inputted to the speech synthesizer 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 $\phi1-\phi4$ which may be appropriately used with such precharge-conditional discharge logic. There are two main clock phases ($\phi1$ and $\phi2$) and two precharge clock phases ($\phi3$ and $\phi4$). Phase $\phi3$ goes low during the first half of phase $\phi2$ and serves as a precharge therefor. A set of clocks $\phi1-\phi4$ is required to clock one bit of data and thus corresponds 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 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. Pat. No. 4,209,844, one cycle time of the digital 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. Pat. No. 4,209,844. To facilitate the an 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 parentheses identify the time periods according to the convention used in this application. On the other hand, the time periods enclosed in parentheses identify the time periods according to the convention used in U.S. Pat. No. 4,209,844. Thus, time period T17 is equivalent to time period (T9) .

At numeral 501 are 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 comprises two cycles, which are labeled A and B. Each such cycle starts at time period T17 and continues to the following time period 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 pitch register 305, as appropriate, during an appropriate time period. During the B cycle, the newly interpolated value is reinserted in the K-stack 302 (or E10 loop 304, or pitch register 305). The thirteenth PC, PC=12, is provided for timing purposes so that all twelve parameters are interpolated once each during a 2.5 millisecond interpolation period.

As was discussed with respect to the parameter interpolator 23 of FIG. 4b and Table I, eight interpolations are performed for each inputting of a new frame of data from Read-Only-Memories 12A and 12B into synthesizer 10. This is seen at numeral 502 of FIG. 5 where timing signals DIV1, DIV2, DIV4 and DIV8 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 Read-Only-Memories 12A and 12B into the synthesizer during IC0. These new target values of the parameters are then used during the next eight interpolation counts, IC1 through IC0; 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, IC0, the present values 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 IC0 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 is 20 milliseconds or equivalent to a frequency of 50 hertz. The DIV8 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 DIV4 one-fourth of the difference is added in, and so on. Thus, during DIV2, one-half of the difference from subtractor 308 is added to the present value of the parameter in adder 310 and lastly during DIV1 the total difference is added in adder 310. As has been previously mentioned, the effect of this interpolation scheme can be seen in Table I.

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 23 and excitation generator 24 (FIG. 4b), the pitch data, energy data and K_1-K_{10} 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 Read-Only-Memories 12A and 12B, this could require a $12 \times 10 \times 50$ or 6,000 hertz bit rate. Using the data compression techniques which will be explained, the bit rate required for synthesizer 10 is reduced to on the order of 1,000 to 1,200 bits per second. And more importantly, it has been found that the speech compression schemes herein disclosed do not appreciably degrade the quality of speech generated thereby in comparison to using the data uncompressed.

The data compression scheme used is pictorially shown in FIG. 6. Referring now to FIG. 6, it can be seen that there are pictorially shown four different lengths of frames of data. One, labeled voice frame, has a length of 56 bits while another entitled unvoiced frame, has a length of 33 bits while still another, called "repeat frame", has a length of eleven bits and still another which may be alternatively called zero energy

11

frame or energy equals 15 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 parameter K7. Six bits of data are reserved for each of three coded parameters, pitch, K1 and K2. Five bits of data are reserved for parameters K3 through K6. 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 bit.

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-six different values, according to the six bit code for K1, each one of the thirty-six 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-six different values while the actual values of coefficients K3 through K6 may be one of twenty different values. Coefficient K7 may be one of sixteen different values and the values of coefficients K8 through K10 may be one of eight different values. The coded pitch parameter is six bits long and therefore may have up to sixty-four different values. However, only sixty-three of these reflect actual pitch values, a pitch code of 000000 being used to signify an unvoiced frame of data. The coded energy parameter is four bits long and therefore would normally have sixteen available ten bit values; however, a coded energy parameter equal to 0000 indicates a silent frame such as occurs during 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 to stop speaking. Thus, of the sixteen codes available for the 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-K6 which in turn have more bits than coded coefficients K7 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 K8 through K10, for example, more bits are used in coded format to define coefficients K1 and K2 than K3-K6 or K7-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 encoded pitch parameter is equal to 000000.

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 prefera-

12

bly 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" frame is encountered. Using coded values for the speech in lieu of actual values, alone would reduce the data rate to 55×50 or 2750 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.

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 the 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 logic 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 (Vss voltage) indicates that the signal is not present. It should also be understood that a numeral three in clocked gates indicates that phrase $\phi 3$ is used as a precharge whereas a four in a clocked gate indicates that phase $\phi 4$ 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–7d, 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 in 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, $\overline{T10}$–$\overline{T18}$, 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 may be responsive to SLOW and SLOW D in an alternative embodiment. 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. Therefore, in an alternative embodiment, the speech module may be instructed to speak at a slower than normal rate. In the present embodiment, however, this capability is not desired and thus the SLOW and SLOW D inputs are tied to Vss.

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 PCO indicates in which cycle, A or B, the parameter count is. The parameter decimal value of the parameter count is decoded by timing PLA 514 which is shown 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 according to the particular parameter count the parameter counter 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 205 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 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. 5 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/CPU Interface Logic Diagram

Turning now to FIGS. 8a-8l, which form a composite diagram, there is shown a detailed logic diagram of ROM/CPU interface logic 21. Parameter input register 205 is a seven bit shift register, most of the stages of which are two bits long. The stages are two bits long in this embodiment, inasmuch as Read-Only-Memories 12A and 12B output, as will be seen, data at half the rate at which data is normally clocked in synthesizer 10.

The coded data in parameter input register 205 is applied on lines IN0-IN5 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 are 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 4 least significant bits of register 205 whereas gate 207 tests for all one's 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 register 205. 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 not only to gate 206 but also to the two most significant bits of the pitch data on line 222 as well as PC=1. Pitch latch 208b is set whenever the coded pitch parameter is a 000000 indicating that the speech is to be unvoiced.

Energy equals zero 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 equals zero latch 208b from the prior frame of speech data while old energy latch 208e stores the output of the energy equals zero latch 208c from the prior frame of speech data. The contents of old pitch latch 208d and pitch equals zero latch 208b are compared in comparison gates 209c 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 equals zero latch 28c are tested by NAND gate 209d for inhibiting interpolation for a transition from a non-speaking frame to a speaking frame of data. The outputs of NAND gate 209d and gates 209c are coupled to a NAND gate 209e 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 209b which generates a ZPAR signal when pitch is equal to zero and when the parameter counter is greater than 5, as indicated by PC5 from PLA 514.

Also shown in FIGS. 8a-8l is a command register 210 which comprises 3 latches 210a, b and c, which latch in the data at D1, D2 and D3 in response to a LOAD COMMAND ENABLE (LDCE) signal. The contents

of command register 210 are decoded by command decoder 211.

When command decoder 211 decodes an LA command, the 4 bits of data on pins D7, D6, D5 and D4 of data bus 17 are latched into address register 213. The nybble of address contained in address register 213 is then coupled through buffers 214 to the ADD1–ADD8 pins to Read-Only-Memories 12A and 12B. Additionally, the LA command is coupled to RB/LA logic 250 where it is used to generate the I1 instruction pin signal to control Read-Only Memories 12A and 12B. RB/LA logic 250 also generates the LAFIN signal to indicate the end of an LA command.

When command decoder 211 decodes a READ-BYTE (RDBY) command, the data stored in the Read-Only-Memories 12A and 12B is accessible to an external Central Processing Unit 19. The READ BYTE command causes the next 8 bits of data to be read from Read-Only-Memories 12A and 12B into data input register 212. The RDBY command is inputted into gate 291 of data register control circuit 290. The output of gate 291 is used to control buffers 212a and to output the data contained in data input register 212 to pins D0–D7 on data bus 17. If the RDBY command is immediately preceded by an LA command at gates 271 and 272 of State Machine 270, the resulting signal which passes through gate 274, generates an I03 instruction pin signal at gate 273. This output, I03, is used to initialize the counter in Read-Only-Memories 12A and 12B. The RDBY command is then delayed after passing through gates 275a and 275b by delay timer latch 276a, b and c. Delay timer latch is set at time T2 and reset at time T17. This delay allows sufficient time for the counter in Read-Only-Memories 12A and 12B to initialize. The RDBY signal is also applied to gate 278 of State Machine 270. The output of gate 278 is applied to gates 277 and used to generate the READ BYTE ENABLE (RDBYEN) signal at the output of gate 279. The RDBYEN signal is applied to gate 292 in data register control logic 290 together with the odd T times and used to generate the I02 instruction pin signals which clock data out of ROMS 12A and 12B and into data input register 212. If the RDBY command is not immediately preceded by an LA command (when the counter in Read-Only-Memories 12A and 12B is already initialized) then the RDBY command is inputted to gate 281 of State Machine 270 and the I03 instruction pin signal and the corresponding delay generated by delay timer latch 276 are not utilized.

If the command decoder 211 decodes a READ and BRANCH (RB) command, the synthesizer 10 may indirectly address areas of Read-Only-Memories 12A and 12B. This is accomplished by having the RB command applied to RB/LA logic 250 which generates the I1 and I04 instruction pin signals which are transmitted to Read-Only-Memories 12A and 12B. Additionally, the RB command is applied to RB timer 252 which delays 240 microseconds and then generates READ AND BRANCH FINISH (RBFIN) signal. The RBFIN signal indicates that the READ AND BRANCH instruction has been executed by Read-Only-Memories 12A and 12B. The RB command is also applied to State Machine 270 at gates 272 and 282; however, since the Read-Only-Memories 12A and 12B generate an internal I0 instruction pin signal during READ AND BRANCH operation, gate 282 acts through gate 274 to disable the I0 instruction pin signal normally generated by State Machine 270.

When command decoder 211 decodes a RESET (RST) command, the RST command is used extensively either alone or in combination with the power-up-clear (PUC) signal to initialize or reset various functions throughout synthesizer 10.

When command decoder 211 decodes a SPEAK (SPK) command, the synthesizer 10 generates synthetic speech utilizing coded speech parameters stored in Read-Only-Memories 12A and 12B. This is accomplished by talk enable logic 251 which generates a SPEAK ENABLE (SPEN) signal which is used to set talk latches 216a, b and c. Talk latch 216a generates a TALK STATUS (TALKST) signal which is used extensively throughout synthesizer 10 to indicate that speech is being generated. Talk latches 216a, b and c remain set unless reset by latch 232a or b in the event of: (1) a power-up-clear (PUC) and/or a reset (RST); (2) an "energy equals 15" detected by gate 207; or (3) during the speak external mode (which will be subsequently discussed) a signal is generated indicating that the buffer is empty and that command decoder 211 is disabled. The SPK command is also applied to gate 281 of State Machine 270 wherein it is utilized to generate a SPEAK FINISHED (SPKFIN) signal.

When command decoder 211 detects a SPEAK EXTERNAL (SPKEXT) command, the synthesizer shifts to the speak external mode of operation. In the speak external mode of operation, coded speech parameters from an external source, preferably the Central Processing Unit 19 of a commercial or home-type computer, are inputted on D0–D7 pins of data bus 17. The coded speech parameters at pins D0–D7 are inputted into a first-in-first-out (FIFO) buffer memory 2215, which is organized as a 16×8 parallel-in, serial-out (PISO) memory. The coded speech parameters are inputted into FIFO buffer memory 2215 through FIFO control 2210. FIFO control 2210 inputs one byte of data each time a WRITE BYTE (WBYT) signal is generated by Input-/Output logic 260. The speech data in FIFO buffer memory 2215 is serially inputted to parameter input register 205 during the speak external mode of operation and speech synthesis takes place. The speak external mode of operation is accomplished in the following manner. Speak external logic 253, which has SPKEXT as its input, generates a DECODE DISABLE (DDIS) signal which disables command decoder 211, thereby ensuring that the data on pins D0–D7 will be treated as speech data, rather than instruction data. Speak external logic 253 also generates a SPEAK EXTERNAL EDGE (SPKEE) signal which purges FIFO buffer memory 2215 by causing FIFO counter 2220 to initialize and to generate a clear (CLR) signal to FIFO control 2210. FIFO buffer memory 2215 also has associated with it FIFO status logic 2230, which generates two signals. The BUFFER LOW (BL) signal is generated whenever the FIFO buffer memory 2215 is half full. This signal is utilized to notify the Central Processing Unit 19 that the synthesizer may require servicing. FIFO status logic 2230 also generates a BUFFER EMPTY (BE) signal which indicates that the FIFO buffer memory 2215 is empty. The BE signal is used to reset talk latch 216, through gate 232b. The DDIS signal is also utilized by I0 logic 2240 to generate a serial shift enable (SSE) signal which allows FIFO control 2210 to serially shift speech data out of FIFO buffer memory 2215 and through load speech logic 2250 and into parameter input register 205. Also associated with ROM/CPU interface logic 21 are Input/Output logic

260 and Interrupt logic 2260. Input/Output logic 260 generates the LOAD COMMAND ENABLE (LDCE) command which allows command register 210 to latch commands in. This is accomplished by latch 261 which is set by the power up clear (PUC) or the "FINISH" signals for the various commands, and latch 262 which is set by the output of latch 261, and the Decoder Disable (DDIS), WRITE SELECT (WS) and READY signals. Therefore, the Load Command Enable signal is generated at the output of latch 263 when: 1. No command is currently being executed; 2. The command decoder 211 is not disabled; 3. A WRITE SELECT signal is present; and 4. the synthesizer 10 has just detected the WRITE SELECT signal ($\overline{READY}$ is high). Input/Output logic 260 also generates the WRITE BYTE (WBYT) signal which enables FIFO control 2210 to load an 8 bit byte of coded speech parameters into the top level of FIFO buffer memory 2215. This is accomplished utilizing latch 264 which is set by a WRITE SELECT (WS) command when the following conditions exist: 1. command decoder 211 is disabled by a DECODE DISABLE signal (DDIS), indicating a SPEAK EXTERNAL command has been executed; 2. the C0 level of FIFO buffer memory 2215 is empty; and 3. synthesizer 10 is not still executing a previous command (the $\overline{READY}$ signal is high). The WRITE BYTE (WBYT) signal is then generated at the output of gate 265. Input/Output logic circuit 260 also generates the READY signal at the output of gate 267 in response to a READ SELECT or WRITE SELECT input signal from the Central Processing Unit 19. When the READY signal is high, the Central Processing Unit 19 is tied to the Speech Module until such time as the READY signal is reset by gate 266. Gate 266 resets the READY signal to zero whenever any of the following signals occur: 1. a WBYT signal is generated at the output of gate 265 indicating that the byte of data on data bus 17 has been read into FIFO buffer memory 2215; 2. the SR2 signal is generated by buffers 212 f–g of data input register 212 through data register control 290, indicating that the status signals generated by a READ SELECT command have been generated; 3. the SR1 signal is generated by buffers 212 a–h of data input register 212 through data register control 290 indicating that the 8 bit byte called for by a Read Select signal preceded by a Read Byte signal has been generated; or 4. the LDCE command generated by gate 263 is inputted to gate 266 indicating that a command has been latched into command register 210. Interrupt logic 2260 generates the Interrupt (INT) signal to advise the Central Processing Unit 19 of a change in the status of the synthesizer 10. The three status signals monitored by the Central Processing Unit 19 are BUFFER EMPTY (BE), BUFFER LOW (BL), and TALK STATUS (TALKST). The BE and BL signals are generated by FIFO status circuit 2230 and outputted via buffers 212f and 212g, respectively, of data input register 212. TALKST is generated by Talk Latch 216a and is outputted via buffer 212h. A change in the status of the synthesizer 10 which results in a BE, BL or TALKST change will be detected by gates 2261, 2262 and 2263 of Interrupt Logic 2260 and will result in an INTERRUPT SIGNAL (INT) being generated through gates 2264 and 2265. Gate 2265 is utilized to reset the INT after receipt of a SR2 signal indicating that the status contained in buffers 212 f–h has been read by the Central Processing Unit 19 or that a RESET signal has been received.

Parameter Interpolator Logic Diagram

Referring now to FIGS. 9a–9d, which form a composite diagram the parameter interpolator logic 23 is shown in detail. K-stack 302 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 are 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. Pat. No. 4,209,844. The data outputted from K-stack 302 to recoding logic 301 at various time periods is shown in Table II. In Table III of U.S. Pat. No. 4,209,844 is shown the data outputted from the K-stack of FIG. 7 thereof. Table II of this patent application differs from Table III of the aforementioned U.S. patent because: (1) recoding logic 301 receives the same coefficient on lines 32-1 through 32-4, on line 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. No. 4,209,844; (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–10c). Recoding logic 301 includes four identical recoding stages 312a–312d, only one of which, 312a, is shown in detail. The first stages of the recoding logic 313 differ 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, −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 304 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 are also coupled in series but also have parallel outputs and inputs to K-stack 302. The appropriate parameter, either energy of 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 316 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 of the K1–K10 speech coefficients which are stored in E10 loop 304 or K-stack 302 to subtractor 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 infor-

mation 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-7d). 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 the 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-7d. 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 register 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 subtractor 308 and delay circuit 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 pitch register 305. Gate 317 is responsive to a latch 321 only for providing pitch, energy or coefficient information to subtractor 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 PCO may switch to a logical one sometime during the transferring of the information from register 303 or 305 to subtractor 308 or delay circuit 309, and therefore, gate 317 is controlled by an A cycle latch 321, which latch is set with PCO at the time a TRANSFER (TK) transfer E10 (TE10) or transfer pitch (TP) signal is generated by timing PLA 511.

The output of gate 317 is applied to subtractor 308 and delay circuit 309. The delay in delay circuit 309 depends on the state of DIV1-DIV8 signals generated by interpolation counter 515 (FIG. 7a). Since the data exits gate 317 with the 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 delay 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 registers 303 and 305. Both delay circuits 309 and 311 can insert up to three bits of delay and when delay circuit 309 is at its maximum, delay circuit 311 is at its minimum delay and vice-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 inverter 236 (FIGS. 8g and 9b). 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 pitch 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 K5-10 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 209b (FIG. 8g).

Gate 326 disables shifting in the 304b portion of E10 loop 304 when an 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 gate 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 to the three bits which may trail the ten most significant bits after an interpolation operation may be zeroed.

Array Multiplier Logic Diagram

FIGS. 10a-10c form a composite logic diagram of array mutliplier 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 input to array multiplier 401 is provided by signals MR_0-MR_{13}, from multiplier multiplexer 415. MR_{13} is the most significant bit while MR_0 is the least significant bit. Another input to array multiplier are the aforementioned +2, −2, +1 and −1 outputs from recording logic 301 (FIG. 9d). The output from array multiplier 401, P_{13}-P_0, is applied to summer multiplexer 402. The least significant bit thereof, P_0, is in this embodiment always made a logical one because doing so establishes the mean of the truncation error as zero instead of ±½ 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 in FIG. 10c in lieu of repetitively showing these elements and making up a logic diagram of array multiplier 401, for simplicity's 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 313 and are further responsive to MR_2-MR_{13}. When multiplies occur in array multiplier 401, the most significant bit is always maintained in the leftmost column elements while the partial sums are continuously shifted toward the right. Inasmuch as each stage of array multiplier 401 operates on two binary bits, the partial sums 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 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 415, which

bits are inverted by inverters **430**, which bits are also supplied in true logic to the B type blocks.

Filter and Excitation Generator Logic Diagram

FIGS. 11a–11d form a composite, detailed logic diagram of 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–10c) 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 φ3 precharge gate **420** (at T19 when no conditional discharge is applied to this input). The reason these signals are applied at these times can be seen from FIG. 8 of the aforementioned U.S. Pat. No. 4,209,844; 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 **415**, one period delay gates **414** and summer multiplexer **402**. Multiplier multiplexer **415** includes one period delay gates **414** which are generally equivalent to one period delay **34'** of FIG. 7 in U.S. Pat. No. 4,209,844. Y-latch **403** is connected to receive the output of delay stack **406**. Multiplier multiplexer **415** selectively applies the output from Y-latch **403**, one period delay gates **414**, or the excitation signal on bus **405** to the input MR0–MR13 of array multiplier **401**. The inputs D0–D13 to delay stack **406** are derived from the outputs of adder **404**. The logics for summer multiplexer **402**, adder **404**, Y-latch **403**, multiplier multiplexer **415** and one bit 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 filter also are provided by logics such as those enclosed by the reference line A, which logics are denoted by long rectangular phantom line boxes labeled "A". The logics for each parallel bit being processed in the filter are not shown in detail for sake of clarity. The portions of the filter handling bits more significant than the least significant bit differ from the logic shown for elements **402**, **403**, **404**, **415** and **414** only with respect to the interconnections made with truncation logics **425** and bus **405** 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 Ix within the reference A phantom line is not needed for the six least significant bits in the 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 YLx within the reference line A is not required for the four least significant bits in the filter.

Delay stack **406** comprises 14 nine bit long shift registers, each stage of which comprises inverters clocked on φ4 and φ3 clocks. As is discussed in U.S. Pat. No. 4,209,844, 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 φ1B–φ4B clocks are generated from T10–T18 timing signal from PLA **512** (FIGS. 7a–7d). The clock buffers **417** in circuit **416** are also shown in detail in FIG. 11c.

Delay stack **409** is nine bits long whereas shift register **35'** in FIG. 7 of U.S. Pat. No. 4,209,844 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. Pat. No. 4,209,844.

The data handled in delay stack **406**, array multiplier **401**, adder **404**, summer multiplexer **402**, Y-latch **403**, and multiplier multiplexer **415** 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** (FIG. 8g). 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 RAM **409** to a random excitation from unvoiced generator **407** until eight interpolation cycles have occurred. Gate **420** NORS 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 of 0.5 to be applied to the filter in Filter and Excitation Generator **24**.

During voiced speech, chirp ROM **409** provides an eight bit output on lines I6–I13 to the 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 III; ROM **409** 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 filter by add one circuit **411**. The output of register **410** is compared with the contents of pitch register **305** in a magnitude comparator **413** 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 **415**. Zeroes 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 ad-

dress 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 six 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 209a (FIG. 8g). Data is inputted to RAM 203 on lines IN0-IN5 from register 205 as shown in FIGS. 8e and 8f and data is outputted on lines CR0-CR5 to ROM 202 as is shown in the aforementioned figures.

Parameter Read-Only-Memory Logic Diagam

In FIGS. 13a-13c, 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 ROM 202 and from parameter counter 513 is 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 seconds 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. 8e and 8f. X-decoder selects one of sixty-eight 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 IV.

Chirp Read-Only-Memory Logic Diagram

FIGS. 14a-14b form a composite diagram of chirp ROM 409. ROM 409 is addressed via address lines A_0-A_8 from register 410 (FIG. 11c) and output information on lines I_6-I_{11} to multiplier multiplexer 415 and lines IM_1 and IM_2 to gates 421 and 420, all of which are shown in FIG. 11c. As was previously discussed with reference to FIGS. 11a-11d, chirp ROM outputs all zeroes 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) and 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 $\overline{A_0}-\overline{A_8}$ for resetting latch 409c. ROM 409 includes timing logic 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 occurring on address lines $\overline{A_0}-\overline{A_8}$. If either condition occurs, 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 in 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 (FIG. 11c) 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 zeroes 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 decoder 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 III. The chirp function is located at addresses 00-40 while zeroes are located at addresses 41-51.

Truncation Logic and Digital-To-Analog Converter

Turning again to FIGS. 11a-11d, the truncation logic 425 and Digital-To-Analog (D/A) converter are shown in detail. Truncation logic 425 includes circuitry for converting the two's complement data on $YL_{13}-YL_4$ to offset binary data. Logics 425a and 425b test the most significant bit from Y-latch 403 on line YL_{13} for the purpose of determining the sign bit and for generating the truncation signals CLIP0 and CLIP1. Logics 425a generate the CLIP0 signal and drive all of the inputs to D/A converter 426 to zero whenever YL_{13} is a logic one and either YL_{12} or YL_{11} are logic zeroes. Logics 425b generate the CLIP1 signal and drive all of the inputs to D/A converter 426 to one whenever YL_{13} is a logic zero and either YL_{12} or YL_{11} are logic ones. Logics 425c test $YL_{13}-YL_{11}$ for the opposite conditions from the conditions just enumerated and generate the NORM signal when no truncation is to take place. This magnitude truncation function effectively truncates the more significant bits on YL_{11} and YL_{12}. It is realized that this is a 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. It has been found that more important digital speech information, which has a smaller magnitude, is effectively amplified by a factor of four by this truncation scheme. Logics 425d convert the two's complement data from Y-latches 403 in lines $YL_{10}-YL_4$ to simple magnitude information on lines $\overline{D/A_6}-\overline{D/A_0}$. Line $\overline{D/A_7}$ is connected to YL_{12}, since when conditions are such that no truncation is occurring, YL_{12} and YL_{11} are identical.

The effects of the truncation scheme utilized are demonstrated in Table V. When the outputs $YL_{13}-YL_4$ result in a decimal number greater than +127, the D/A converter inputs are all driven to logical ones, and the output current is zero. When the outputs $YL_{13}-YL_4$ result in a decimal number less than −128, the D/A converter inputs are all driven to logical zeroes, and the output current is 1500 microamps. The midpoint occurs when $YL_{13}-YL_4$ is equivalent to a −1 in decimal notation, and the D/A output current is equal to 250 microamps. Thus, D/A converter 426 generates an analog output which varies about a static level (750 microamps in this embodiment). Additionally, when the Speech Module has ceased talking, the TALKST signal is utilized to drive output current to zero, in order to conserve power consumption.

The outputs $\overline{D/A_7}$–$\overline{D/A_0}$ are coupled to D/A converter **426**. $\overline{D/A_7}$–$\overline{D/A_0}$ are preferably connected to the gates of eight MOS switching devices **429a**. $\overline{D/A_7}$–$\overline{D/A_0}$ are also connected through inverters **429b** to the gates of eight MOS switching devices **429c**. The sources of switching devices **429a** are connected to the Vss and the sources of switching devices **429c** are connected to the Vref. Vref is a predetermined voltage calculated to bias current sources **429d** into the saturated mode of operation. The drains of switching devices **429a** and **429c** are connected at a common point in each leg of D/A converter **426** and tied to the gates of current source devices **429d**. Current sources **429d** have their current carrying electrodes coupled in parallel with the source of each current device connected to Vss. The drains of current devices **429d** are connected to an output pin through a 1.8K ohm resistor to an audio amplifier and speaker circuit contained in a commercial or home-type computer.

It should be appreciated by those skilled in the art that D/A converter **426** has effectively converted the sign data and magnitude data contained in YL_{13}–YL_4 to an analog signal, which can be characterized as an alternating signal with a fixed component. And, it should be apparent that D/A converters, such as disclosed here will find use in other embodiments in addition to speech synthesis circuits.

Read-Only-Memories

Read-Only-Memories **12A** and **12B** are preferably of the type shown and described in U.S. Pat. No. 4,189,779.

plated that the appended claims will cover any such modifications or embodiments that fall within the true scope of the invention.

TABLE I

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 for specific interpolation counts and the values $\dfrac{P_i - P_o}{P_t - P_o}$ (P_0 is initial parameter value) are as follows:

INTERPOLATION COUNT	N_i	$\dfrac{P_i - P_o}{P_t - P_o}$
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 II

DATA OUTPUTTED FROM K-STACK 302 TO RECODING LOGIC 301 BY TIME PERIODS

K-STACK OUTPUT BIT	LINE	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17
LSB	32-1	K_2	K_1	A	K_9	K_8	K_7	K_6	K_5	K_4	K_3
	32-2	K_2	K_1	A	K_9	K_8	K_7	K_6	K_5	K_4	K_3
	32-3	K_2	K_1	A	K_9	K_8	K_7	K_6	K_5	K_4	K_3
	32-4	K_2	K_1	A	K_9	K_8	K_7	K_6	K_5	K_4	K_3
	32-5	K_3	K_2	K_1	A	K_9	K_8	K_7	K_6	K_5	K_4
	32-6	K_3	K_2	K_1	A	K_9	K_8	K_7	K_6	K_5	K_4
	32-7	K_4	K_3	K_2	K_1	A	K_9	K_8	K_7	K_6	K_5
	32-8	K_4	K_3	K_2	K_1	A	K_9	K_8	K_7	K_6	K_5
	32-9	K_5	K_4	K_3	K_2	K_1	A	K_9	K_8	K_7	K_6
MSB	32-10	K_5	K_4	K_3	K_2	K_1	A	K_9	K_8	K_7	K_6

K-STACK OUTPUT BIT	LINE	T18	T19	T20	T21	T22	T23	T24	T25	T26	T27
LSB	32-1	K_2	K_1	K_{10}	K_9	K_8	K_7	K_6	K_5	K_4	K_3
	32-2	K_2	K_1	K_{10}	K_9	K_8	K_7	K_6	K_5	K_4	K_3
	32-3	K_2	K_1	K_{10}	K_9	K_8	K_7	K_6	K_5	K_4	K_3
	32-4	K_2	K_1	K_{10}	K_9	K_8	K_7	K_6	K_5	K_4	K_3
	32-5	K_3	K_2	K_1	K_{10}	K_9	K_8	K_7	K_6	K_5	K_4
	32-6	K_3	K_2	K_1	K_{10}	K_9	K_8	K_7	K_6	K_5	K_4
	32-7	K_4	K_3	K_2	K_1	K_{10}	K_9	K_8	K_7	K_6	K_5
	32-8	K_4	K_3	K_2	K_1	K_{10}	K_9	K_8	K_7	K_6	K_5
	32-9	K_5	K_4	K_3	K_2	K_1	K_{10}	K_9	K_8	K_7	K_6
MSB	32-10	K_5	K_4	K_3	K_2	K_1	K_{10}	K_9	K_8	K_7	K_6

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 described 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 contem-

TABLE III

	CHIRP ROM CONTENTS	
ADDRESS	CHIRP FUNCTION VALUE	STORED VALUE (COMPLEMENTED)
00	00 ·	FF
01	2A	D5
02	D4	2B

TABLE III-continued

CHIRP ROM CONTENTS

ADDRESS	CHIRP FUNCTION VALUE	STORED VALUE (COMPLEMENTED)
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	AO
13	5A	A5
14	05	FA
15	0F	F0
16	26	D9
17	FC	03
18	A5	5A
19	A5	5A
20	D6	29
21	DD	22

TABLE III-continued

CHIRP ROM CONTENTS

ADDRESS	CHIRP FUNCTION VALUE	STORED VALUE (COMPLEMENTED)
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	12
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 IV

DECODED PARAMETERS

CODE	E	P	K1	K2	K3	K4	K5	K6	K7	K8	K9
00	000	000	205	2DA	23F	1EF	28B	32B	20A	33D	386
01	001	00E	207	2FG	250	2F2	2A5	350	2F8	38B	3C9
02	002	00F	209	315	265	324	2C2	377	318	3E0	00F
03	003	010	20B	336	27F	35C	2E4	3A0	33A	036	053
04	004	011	20F	359	29E	39F	309	3CA	35F	089	095
05	006	012	213	37E	263	3D8	332	3F5	386	005	0D2
06	008	013	218	3A4	2EF	019	3SE	021	3AE	117	108
07	00B	014	21F	3CC	321	059	38D	04B	3D7	14F	137
08	010	015	227	3F4	359	096	3BF	075	001		
09	017	016	231	01C	395	0CF	3F1	090	02B		
0A	021	017	23E	044	305	103	024	OC3	054		
0B	02F	018	24E	06C	016	130	056	0E7	07D		
0C	03F	019	262	091	057	157	087	108	0A3		
0D	055	01A	27A	0B6	094	178	0B5	126	OC8		
0E	072	01B	296	008	0CE	193	0E0	142	0EA		
0F	000	01C	2B8	0F8	102	1A9	107	15B	10A		
10		01D	2E0	116	202	202	202	202			
11		01E	30E	131	20A	20A	20A	20A			
12		01F	341	14A	139	139	139	139			
13		020	379	160	13E	13E	13E	13E			
14		022	3B5	174							
15		024	3F3	186							
16		026	031	196							
17		028	06E	1A4							
18		029	0A8	1B0							
19		02B	0DD	1BB							
1A		02D	10D	1C5							
1B		030	137	1C0							
1C		031	15C	1D4							
1D		033	17B	1DA							
1E		036	194	1DF							
1F		037	1AA	1E6							
20		039	202	202							
21		03C	20A	20A							
22		03E	139	139							
23		040	13E	13E							
24		044									
25		048									
26		04A									
27		04C									
28		051									
29		055									
2A		057									
2B		05A									
2C		060									
2D		063									
2E		067									
2F		06B									
30		070									
31		075									
32		07A									
33		07F									
34		085									

TABLE IV-continued

CODE	E	P	K1	K2	K3	K4	K5	K6	K7	K8	K9
		DECODED PARAMETERS									
35		08B									
36		091									
37		097									
38		09D									
39		0A4									
3A		0AB									
3B		0B2									
3C		0BA									
3D		0C2									
3E		0CA									
3F		0D3									

TABLE V

	Y LATCH OUTPUT				D/A	ANALOG OUTPUT
	Y_{L13}	Y_{L12}	Y_{L11}	$Y_{L10}\,Y_{L4}$	INPUT	μ AMPS
	0	1	0	X	11111111	0
> +127	0	1	0	X	11111111	0
	0	0	1	X	11111111	0
127	0	0	0	1111111	11111111	0
126	0	0	0	1111110	11111110	5.86
+1	0	0	0	0000001	10000001	738
0	0	0	0	0000000	10000000	744
*−1	1	1	1	1111111	01111111	750
−2	1	1	1	1111110	01111110	755.8
−128	1	1	1	0000000	00000000	1500
< −128	1	1	0	X	00000000	1500
	1	0	1	X	00000000	1500
	1	0	0	X	00000000	1500

*NO OUTPUT, RESTING LEVEL

What is claimed is:

1. An interface system for selectively interactively interfacing between a control device providing variable length instructions and a memory device which initiates and executes respective ones of said variable length instructions over a longer time period as compared to the time period required for the control device to provide a variable length instruction to the memory device, said interface system comprising:

first circuit means interposed between said control device and said memory device for enabling said control device to provide a variable length instruction for reception by said memory device;

second circuit means operatively coupled to said first circuit means and providing a control signal changeable from a first state to a second state in response to said memory device receiving said variable length instruction from said control device to inform said control device that execution of the variable length instruction by said memory device has not been completed, the second state of said control signal being returned to the first state of said control signal in response to completed reception of the variable length instruction by said memory device from said control device while said memory device continues to execute the variable length instruction;

gating circuit means for selectively electrically coupling said control device to said memory device in order to provide said variable length instructions to said memory device, said control device being released from active interaction with said memory device in response to said control signal changing from its second state to its first state to free said control device to perform other control tasks independent of said memory device while said memory device continues to execute the variable length instruction to its completion following which said memory device is available to receive a subsequent variable length instruction from said control device; and

said second circuit means being responsive to an attempt at access of said memory device from said control device during the execution of a variable length instruction by said memory device to change the first state of said control signal to the second state thereof to inhibit interaction between said control device and said memory device and to inhibit action by said control device independent of said memory device until said memory device has completed execution of the current variable length instruction with which said memory device is concerned.

2. An interface system as set forth in claim 1, wherein said control device is a central processing unit of a computer.

3. An interface system as set forth in claim 1, wherein said memory device is included within speech synthesizer means.

4. A computer system capable of providing synthetic speech comprising:

a central processing unit for providing variable length speech instructions;

speech synthesizer means for initiating and executing said variable length speech instructions and for producing an analog speech signal in accordance therewith;

speaker means for converting said analog speech signal to audible sound;

interfacing means interposed between said central processing unit and said speech synthesizer means for selectively interactively interfacing therebetween, said interfacing means comprising:

first circuit means interposed between said central processing unit and said speech synthesizer means for enabling said central processing unit to provide a variable length speech instruction for reception by said speech synthesizer means,

second circuit means operatively coupled to said first circuit means and providing a control signal changeable from a first state to a second state in response to said speech synthesizer means receiving said variable length speech instruction

31

from said central processing unit to inform said central processing unit that execution of the variable length speech instruction by said speech synthesizer means has not been completed, the second state of said control signal being returned to the first state of said control signal in response to completed reception of the variable length speech instruction by said speech synthesizer means from said central processing unit while said speech synthesizer means continues to execute the variable length speech instruction,

gating circuit means for selectively electrically coupling said central processing unit to said speech synthesizer means in order to provide said variable length speech instructions to said speech synthesizer means, said central processing unit being released from active interaction with said speech synthesizer means in response to said control signal changing from its second state to its first state to free said central processing unit to perform other processing tasks independent of said speech synthesizer means while said speech synthesizer means continues to execute the variable length speech instruction to its completion following which said speech synthe-

32

sizer means is available to receive a subsequent variable length speech instruction from said central processing unit, and

said second circuit means being responsive to an attempt at access of said speech synthesizer means from said central processing unit during the execution of a variable length speech instruction by said speech synthesizer means to change the first state of said control signal to the second state thereof to inhibit interaction between said central processing unit and said speech synthesizer means and to inhibit action by said central processing unit independent of said speech synthesizer means until said speech synthesizer means has completed execution of the current variable length speech instruction with which said speech synthesizer means is concerned.

5. A computer system as set forth in claim 4, further including user entry means operatively associated with said central processing unit for receiving inputs from an operator to selectively control said central processing unit.

6. A computer system as set forth in claim 5, wherein said user entry means comprises a keyboard.

* * * * *

30

35

40

45

50

55

60

65