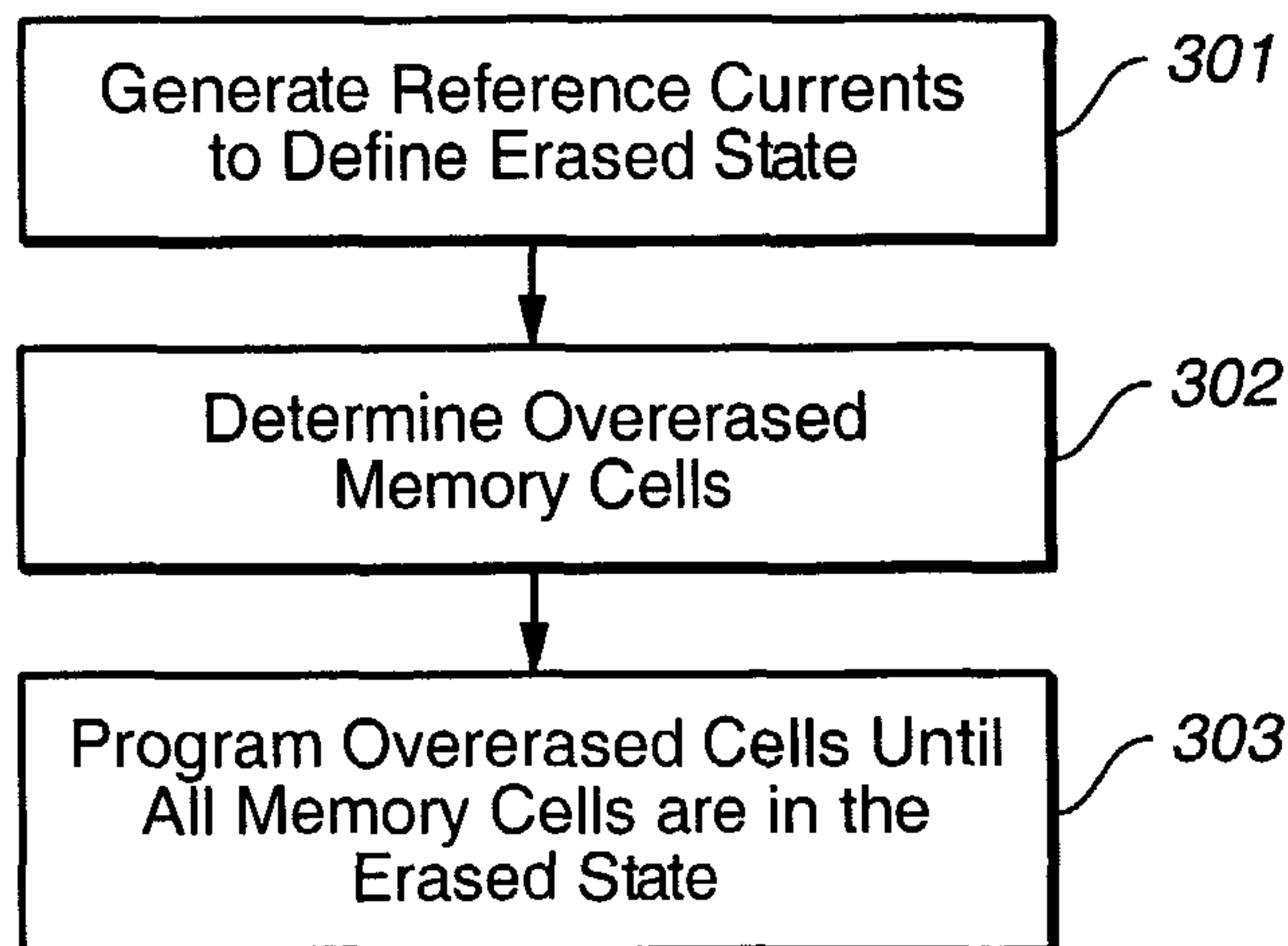




(86) Date de dépôt PCT/PCT Filing Date: 2003/07/30  
 (87) Date publication PCT/PCT Publication Date: 2004/03/18  
 (85) Entrée phase nationale/National Entry: 2005/03/02  
 (86) N° demande PCT/PCT Application No.: US 2003/023881  
 (87) N° publication PCT/PCT Publication No.: 2004/023239  
 (30) Priorité/Priority: 2002/09/04 (10/235,265) US

(51) Cl.Int.<sup>7</sup>/Int.Cl.<sup>7</sup> G11C 16/04, G11C 16/06  
 (71) Demandeur/Applicant:  
ATMEL CORPORATION, US  
 (72) Inventeur/Inventor:  
MANEA, DANUT I., US  
 (74) Agent: SMART & BIGGAR

(54) Titre : PROCÉDE DE RECUPERATION DE BITS ECRASES DANS UNE MEMOIRE  
 (54) Title: METHOD OF RECOVERING OVERERASED BITS IN A MEMORY DEVICE



(57) **Abrégé/Abstract:**

A method (Fig. 1) of recovering overerased bits in a memory cell. In the method, a pair of reference currents ( $I_{11H}$ ,  $I_{11L}$ ) are internally generated (301) to define a current window corresponding to the erased state ("11") of the memory cell. The first reference current defines the highest current of the current window (12) and the second reference current defines the lowest current of the current window. Then, it is determined (302) which of the memory cells in a memory array are in an overerased state (13) by having an amount of charge on its floating gate that corresponds to a conduction current during a read operation that is greater than the first reference current. Then, the overerased cells are programmed (303; Fig. 8) until the cells are in the erased state.

## (12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property  
Organization  
International Bureau



(43) International Publication Date  
18 March 2004 (18.03.2004)

PCT

(10) International Publication Number  
**WO 2004/023239 A3**

(51) International Patent Classification<sup>7</sup>: **G11C 16/04**,  
16/06

MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG,  
SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN,  
YU, ZA, ZM, ZW.

(21) International Application Number:  
PCT/US2003/023881

(84) Designated States (*regional*): ARIPO patent (GH, GM,  
KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),  
Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),  
European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE,  
ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO,  
SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM,  
GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(22) International Filing Date: 30 July 2003 (30.07.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
10/235,265 4 September 2002 (04.09.2002) US

**Declarations under Rule 4.17:**

- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii)) for all designations
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for all designations

(71) Applicant: **ATMEL CORPORATION** [US/US]; 2325  
Orchard Parkway, San Jose, CA 95131 (US).

**Published:**

- with international search report
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

(72) Inventor: **MANEA, Danut, I.**; 20012 A Rodrigues Av-  
enue, Cupertino, CA 95014 (US).

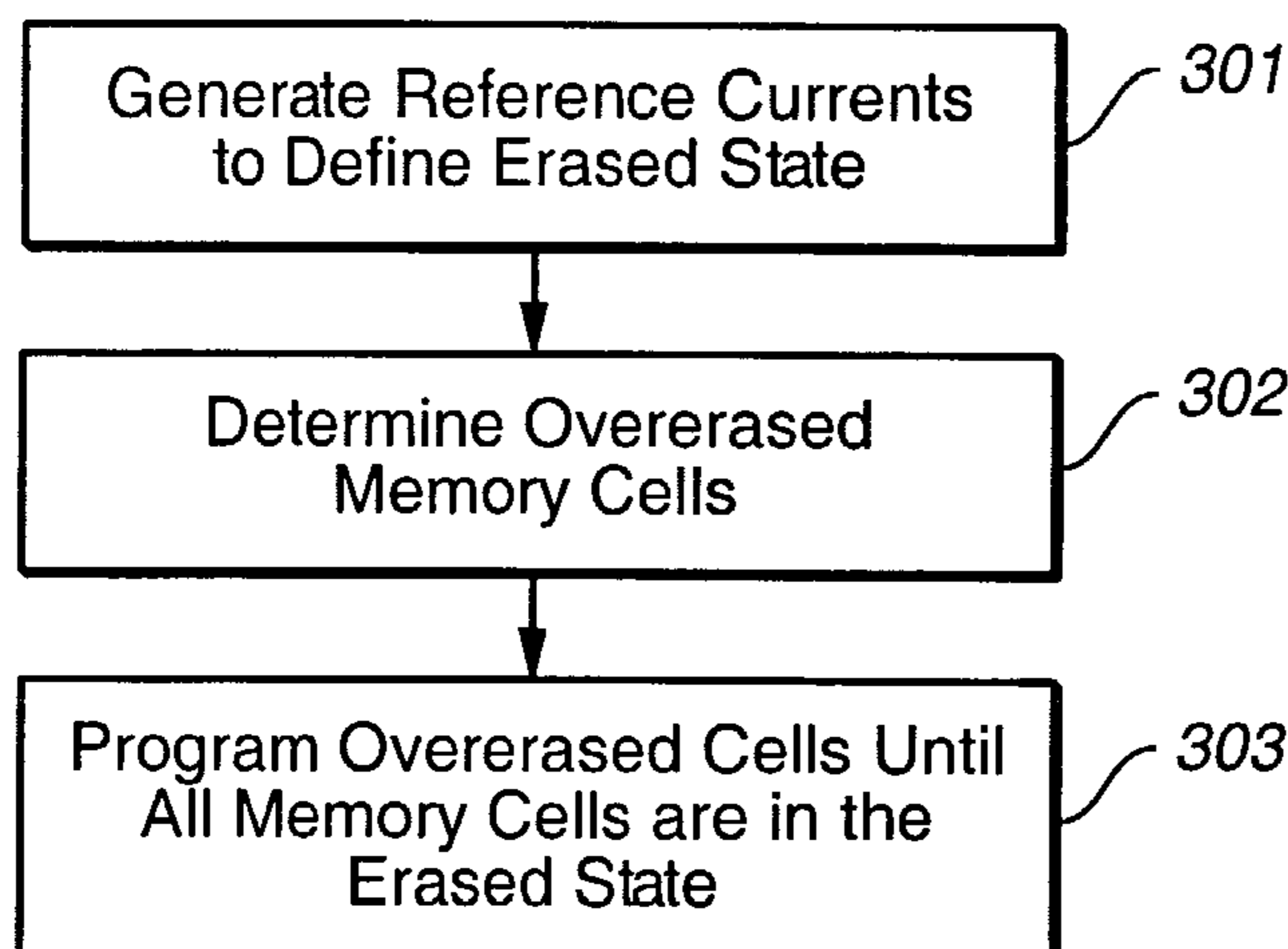
(74) Agent: **SCHNECK, Thomas**; Law Offices of Thomas  
Schneck, P.O. Box 2-E, San Jose, CA 95109-0005 (US).

(88) Date of publication of the international search report:  
6 May 2004

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU,  
AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU,  
CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH,  
GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC,  
LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW,

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD OF RECOVERING OVERERASED BITS IN A MEMORY DEVICE



(57) Abstract: A method (Fig. 1) of recovering overerased bits in a memory cell. In the method, a pair of reference currents ( $I_{11H}$ ,  $I_{11L}$ ) are internally generated (301) to define a current window corresponding to the erased state ("11") of the memory cell. The first reference current defines the highest current of the current window (12) and the second reference current defines the lowest current of the current window. Then, it is determined (302) which of the memory cells in a memory array are in an overerased state (13) by having an amount of charge on its floating gate that corresponds to a conduction current during a read operation that is greater than the first reference current. Then, the overerased cells are programmed (303; Fig. 8) until the cells are in the erased state.

WO 2004/023239 A3

-1-

## Description

METHOD OF RECOVERING OVERERASED  
BITS IN A MEMORY DEVICE

5

## TECHNICAL FIELD

This invention relates to semiconductor memory devices, and more particularly to a method of recovering overerased bits in memory devices.

10

## BACKGROUND ART

In conventional single bit per cell memory devices, the memory cell assumes one of two information storage states, either an on-state or an off-state. This combination of either on or off defines one bit of information. In bi-level memories, since the cells can only have two different values of threshold voltage,  $V_t$ , during the reading operation, it is only necessary to sense whether or not the addressed transistor is

conductive. This is generally done by comparing the current flowing through the memory transistor biased with predetermined drain-to-source and gate-to-source voltages with that of a reference transistor under the same bias conditions, either directly through current-mode sensing or after a current-to-voltage conversion through voltage-mode sensing.

25

In programming a typical single-bit-per-cell flash memory cell, a high potential (such as, for example, approximately 9-12 volts) is applied to the control gate of the cell, the source terminal is grounded, and the drain terminal is connected to a voltage of about 5 volts. This operation can be performed in an array by selectively applying the pulse to the word line which connects the control gates, and biasing the bit line which connects the drains. This is

35

-2-

commonly known in the art as the hot electron injection method of programming flash memory cells. Hot electron injection is used to move charge in the floating gate, thus changing the threshold voltage of the floating gate transistor. By placing the high voltage on the control gate, this generates electrons to flow in the channel and some hot electrons are injected on to the floating gate and change the potential of the floating gate to be more negative. Therefore, injection tends to saturate and the threshold voltage of a floating gate transistor follows the same trend. The state of the memory cell transistor is read or sensed by placing an operating voltage (for example, approximately 4-6 volts) on its control gate and 0.5 - 1 volts on the drain, and then detecting the level of current flowing between the source and drain to determine which memory state the cell is in.

Programming and sensing schemes for multi-level memory devices are more complex, typically requiring  $2^n - 1$  voltage references, where  $n$  is the number of bits stored in the cell. With reference to Fig. 9, an example of a prior art multi-level memory device is shown having two bits per cell which corresponds to four memory levels having three voltage references. A first memory level 121, represented by the binary number 11, is the state in which the memory cell has no charge. The memory level 124 in which the memory cell is fully charged is represented by the binary number 00. (The terms "no charge" and "fully charged" are used herein, and throughout this discussion, for the purposes of explanation and are not intended to be limiting. For example, the (11) state could have a slight amount of charge and the (00) state could have an amount of charge less than the absolute maximum amount of charge.) In between the uncharged state (11) 121 and the fully charged state (00) 124 are a first intermediate level

-3-

122, represented by the binary number 10, in which the memory cell has a small amount of charge, and a second intermediate level 123, represented by the binary number 01, in which the memory cell has more charge than the 10 state but is not fully charged. The threshold voltages (Vt) shown in between each of the memory states of the memory cell represent the threshold voltages needed to transition between memory cell states. As discussed, for a two-bit cell having four memory levels, there are three voltage references 111, 112, 113. For example, at the threshold voltage of 2.5 volts, the memory state is at the reference level 111 where the state of the cell will transition from the 11 state to the 10 state. At a voltage threshold  $V_t = 3.5$  volts, the memory cell is at the reference level 112 where the state of the cell will transition from the 10 state to the 01 state. And at the voltage threshold of  $V_t = 4.5$  volts, the memory cell is at the reference level 113 where the state of the cell will transition from the 01 state to the 00 state. The threshold voltage values shown in Fig. 9 are merely illustrative and the actual values of  $V_t$  will depend on the construction of the memory cell.

One of the main difficulties in implementing multi-level nonvolatile memory cells is being able to accurately program the cell, i.e. to place just the amount of charge on the floating gate of the cell transistor that is required to obtain the target value of the threshold voltage. The usual manner that is used in the prior art to deal with the problem of accurate charge placement is by using a cell-by-cell program and verify approach. In the program and verify approach, the programming operation is divided into a number of partial steps and the cell is sensed after every step to determine whether or not the target threshold voltage is achieved, so as to continue the programming if this is

-4-

not the case. As each cell is independently controlled during programming, this technique allows simultaneous programming of a whole byte or even a number of bytes. This procedure ensures that the target  $V_t$  is reached, with the accuracy allowed by the quantization inherent in the use of finite programming steps. However, this process can be very long and must be controlled by on-chip logic circuitry.

A typical program and verify technique is illustrated in Fig. 10. As shown in Fig. 10, the programming of the memory cell is implemented by an alternating sequence of programming and verifying voltage pulses. The voltage of each programming pulse incrementally increases with respect to time until the desired target voltage is reached. The voltage level of the verify pulse remains constant throughout the programming process. For example as shown, after a first verify pulse 151, a first programming pulse 141 is implemented, and then a verify pulse 152 follows. A next programming pulse 142 of an incrementally increased potential is applied, followed by a verify pulse 153, followed by a third programming pulse 143 which is increased in voltage from the previous programming step, followed by a next verify pulse 154 and so on, until the final programming pulse 147 is applied to allow the cell to reach the threshold voltage of the desired memory state. As can be seen in Fig. 10, the shape of the graph resembles a staircase, and this programming method is generally known in the art as staircase gate voltage ramp programming. This staircase method is described in numerous patents, including, for example, U.S. Pat. Nos. 5,043,940; 5,268,870; 5,293,560; and 5,434,825.

The electrical erase of a flash memory cell is usually a global operation that is applied to entire sections of a memory array. Each sector has its own

-5-

internal source line and its own circuitry used to switch this line. To perform the erase, a high electric field is provided between the source and the floating gate of the cell, causing the extraction of negative charge from the floating gate by means of Fowler-Nordheim tunneling. Typically, the erase operation is accomplished by placing a large negative voltage, such as -10V, on the floating gate and a positive voltage, such as 6V on the source.

Fig. 11 illustrates the ideal threshold voltage distribution for a multi-level memory device. The  $V_t$  distribution for each memory level is in the typical bell-shaped curve, with greatest number of cells at the target  $V_t$  at the center of the cell distribution curve and the number of cells decreasing as the voltage moves away, on both sides, from the target  $V_t$ . The cell distribution curves 172, 173, 174 for the states 10, 01 and 00 are similar to each other and are much tighter than the curve 171 for the 11 state. This is because the curve 171 for the 11 state is created by the erase mechanism, since the 11 state is the erased state, while the other states 10, 01 00 are created by the programming algorithm. This  $V_t$  distribution is similar for single-bit memory devices, except that there is one erased state and only one programming state.

One of the problems in erasing flash memory cells is the fact that the cell current distribution is not consistent. Not every memory cell erases at the same threshold voltage due to variations in oxide thicknesses, the location of the die in relation to the center of the wafer, and other such factors. Therefore, because some memory cells are erased faster than others, the memory cells that become erased more quickly can become overerased. This is because the erase mechanism in flash memories does not stop by itself; as long as the erasing voltage is applied to the memory cell, electrons are

-6-

continuously removed from the floating gate and the erasing pulse is not stopped at the right time. Since the erase operation continues until all of the memory cells in the sector being erased have been erased, some  
5 of the memory cells will be overerased.

Fig. 12 illustrates this concept of overerased bits. Cell distribution curve 171 represents an ideal curve in which the target threshold voltage 186 (or the threshold voltage at which most of the cells end up  
10 following the erase operation) is 2V. The undererase threshold voltage 185, or the maximum threshold voltage at which a cell is considered to be erased, is shown to be 2.5 volts. The overerase threshold voltage 184 is shown to be 1.5 volts. As explained above, it is  
15 difficult to control the cell distribution when erasing a large array of memory cells. Often times, the distribution curve would be more like curve 181. In this case, a number of memory cells 183 have been overerased due to the cells having been erased faster than other  
20 memory cells in the array. Because the erase mechanism operates until the last memory cell in the array is erased to beyond the undererase threshold 185, the faster cells 183 become overerased.

Overerasing of the memory cells is not  
25 desirable and can be a potential cause of failure because the memory array cannot be read correctly if it contains depleted cells, as all of the cells connected to the same bitline of a depleted cell would be read as a "1", regardless of the actual content of the cell.

30 In the prior art, different soft-programming techniques have been developed to solve the problem by recovering overerased bits. For example, a small constant voltage can be applied on the wordline, such as for example 1.5V, and then a soft-programming operation  
35 can be done to program cells below 1.5V back up to the

27-

overerase threshold. However, all of the proposed soft-programming recovery techniques are time and power consuming, which is undesirable. Other techniques for recovering overerased bits involve generating a separate reference current. This is also not desirable because it does not use the existing reference cell and thus adds more complexity and power consumption.

It is the object of the present invention to provide a method of recovering the overerased bits in a memory cell which does not require soft-programming techniques and does not require the generation of a separate reference current.

It is a further object of the invention to provide a method of recovering the overerased bits in a memory cell which is faster and consumes less power.

#### SUMMARY OF THE INVENTION

The above objects have been achieved by a method of recovering overerased bits in a memory device, in particular a multi-level memory device, in which a pair of reference currents are generated internally to define a current window corresponding to the erased or "11" state. The first reference current defines the highest current of the current window and the second reference current defines the lowest current of the current window. Then, it is determined which of the memory cells in the memory array have an amount of charge on its floating gate that corresponds to a conduction current that is greater than the first reference current. This group of memory cells is in an overerased state. Then, a plurality of alternating programming and verifying pulses are applied to each of the group of overerased memory cells in order to program the overerased cells to the erased state. The programming pulses are at the same voltage levels as is used to

-8-

program the memory cells to the other memory states (such as "10", "01", "00"). Therefore, the method uses the existing programming approach and programs the "11" state in the same manner as the other memory states. This results in a fast and more precise recovery of overerased bits. The method also does not require soft-programming or the generation of separate reference currents, which helps to conserve power and to simplify the process.

#### 10 BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a flow chart of the method of the present invention.

Fig. 2 is a graph of memory cell conduction current levels  $I_{SD}$  for respective reference, read and verify, with the read and verify current levels being generated in the present invention from the reference current levels.

Fig. 3 is a graph of memory cell conduction current levels for the overerased and erased memory levels.

Fig. 4 is a schematic circuit diagram of a circuit generating the reference current levels used in the method of the present invention.

Fig. 5 is a schematic circuit diagram of a detailed circuit portion of Fig. 4 for optionally generating the reference current levels from a single reference memory cell.

Fig. 6 is schematic circuit diagram illustrating a portion of the analog circuit block in Fig. 4 for generating the read and verify current levels from the supplied reference currents.

Fig. 7 is a schematic circuit diagram showing in further detail one element of the analog circuit block generating fractional currents of a supplied reference current with a selected margin value  $m$  for subsequent use

-9-

according to the scheme of Fig. 5 in generating read and verify current levels.

Fig. 8 is a graph showing the voltage to time relationship for the program and verify pulse steps used in the present invention.

Fig. 9 is a representation of a two-bit memory cell having four memory states as is known in the prior art.

Fig. 10 is a graph showing the voltage to time relationship for the program and verify pulse steps of the staircase programming method, as is known in the prior art.

Fig. 11 is a graph of the threshold voltage distribution for a four state memory cell as is known in the prior art.

Fig. 12 is a graph of the threshold voltage distribution curve for the erased state "11", in which some cells are overerased, as is known in the prior art.

## BEST MODE OF CARRYING OUT THE INVENTION

With reference to Fig. 1, the first step 301 in the method of the present invention is to generate a pair of reference currents to define a current window corresponding to the state in which the memory cell is fully erased. Fig. 2 shows the reference, read and verify current levels for a four level memory cell array. The present invention is not limited to four level memory cell arrays, and can also be used in single-bit memory cells, as well as in other types of memory cell arrays, such as memory cell arrays with 8, 16 or other numbers of memory cell states. Four reference current levels  $I_R$  are provided, one for each memory state. These are designated  $I_{R00}$ ,  $I_{R01}$ ,  $I_{R10}$  and  $I_{R11}$  for the four memory states "00", "01", "10" and "11". State "11" typically corresponds to the fully erased state, state "00" to the

-10-

fully programmed state, and "01" and "10" to intermediate states.

The fully erased state has the least charge (substantially zero) stored on the floating gate of the nonvolatile memory cell transistor structure, and has the lowest threshold voltage and largest conduction current of the four memory states. The fully programmed state has the most charge stored on the floating gate, and has the highest threshold voltage (often larger than the applied gate voltage during a read operation) and lowest conduction current (possibly zero) of the four states. The intermediate states have stored charge amounts that are between that of the fully erased and fully programmed states, and thus have intermediate threshold voltage and conduction current levels.

The reference current levels  $I_{R00}$ ,  $I_{R01}$ ,  $I_{R10}$  and  $I_{R11}$  are set by programmed reference memory cells at the center of the respective states, furthest from the read boundaries with adjacent states.

For a four state memory cell, three read boundaries exist between adjacent states. These are read current levels  $I_L$ ,  $I_M$  and  $I_H$  (low, medium and high).  $I_L$  is the boundary between adjacent memory states "00" and "01",  $I_M$  is the boundary between adjacent memory states "01" and "10", and  $I_H$  is the boundary between adjacent memory states "10" and "11". These boundary levels are preferably defined in the present invention as the arithmetic average between the corresponding adjacent states.

30

$$\begin{aligned} I_L &= (0.50) I_{R00} + (0.50) I_{R01}; \\ I_M &= (0.50) I_{R01} + (0.50) I_{R10}; \text{ and} \\ I_H &= (0.50) I_{R10} + (0.50) I_{R11}. \end{aligned}$$

-11-

Read current levels somewhat above or below those preferred values may be used, as the margin need not be exactly 0.50. However, a margin of substantially 50%, i.e. between about 40% and 60%, is generally desired.

5 For example, if a margin of 45% is used,  $I_L = (0.55) I_{R00} + (0.45) I_{R01}$ , etc. and the read currents will be slightly lower than it would be with an exactly 50% margin. (Also, the margins need not be exactly the same for all three state boundaries  $I_L$ ,  $I_M$ ,  $I_H$ .)

10 During a typical memory cell read operation, sense amplifiers and a comparison circuit compare the conduction current of a selected memory cell with the respective read boundary current levels to establish which of the four states that memory cell is in. This read operation is not part of

15 the present invention and can be done by any known method using the (n-1) boundary current levels supplied by the present invention to determine one of the n memory states for a selected multilevel memory cell.

A typical program operation consists of a

20 sequence of short program and verify steps until a selected memory cell arrives at the desired programmed state. For programming, a margin value  $m$  is used to create windows 17 of allowable current values for each state. Regions 15 between the windows 17 are current

25 levels that are forbidden as final program states. The forbidden zones 15 include the read boundary levels and a suitable margin to account for operational variations in memory cell conditions, such as temperature. Thus, in the present invention, upper and lower window levels are

30 established for most memory states for the program verify operation. For the fully erased state "11", the upper and lower window levels are only used to verify overerase recovery. For the fully programmed state "00", a lower window level is not needed, as there is no over-

35 programmed condition. Thus, a total of seven verify

-12-

current levels  $I_{11H}$ ,  $I_{11L}$ ,  $I_{10H}$ ,  $I_{10L}$ ,  $I_{01H}$ ,  $I_{01L}$ , and  $I_{00H}$  are defined for the four memory states.

$$\begin{aligned}
 & I_{11H} = (1+m) I_{R11}; \\
 5 \quad & I_{11L} = (1-m) I_{R11} + (m) I_{R10}; \\
 & I_{10H} = (1-m) I_{R10} + (m) I_{R11}; \\
 & I_{10L} = (1-m) I_{R10} + (m) I_{R01}; \\
 & I_{01H} = (1-m) I_{R01} + (m) I_{R10}; \\
 & I_{01L} = (1-m) I_{R01} + (m) I_{R00}; \text{ and} \\
 10 \quad & I_{00H} = (1-m) I_{R00} + (m) I_{R01} .
 \end{aligned}$$

The margin value(s)  $m$  is less than the margin used to define the read boundaries (less than 50%, i.e. less than 0.5 if the preferred read margin of 50% is used for  $I_L$ ,  $I_M$  and  $I_H$ ), and is typically between 0.05 and 0.375. Values  
 15 between 0.10 and 0.25 are preferred. Smaller margin values  $m$  (with smaller allowed program windows 17 and larger forbidden zones 15) are preferred when large operational variations are expected, whereas most often a  
 20 value of about 0.25 will be adequate. A preferred embodiment of the analog circuit block used in the present invention for establishing the verify levels allows any of the values 0.125, 0.25, or 0.375 to be selected for the margin  $m$ . The margin value can differ  
 25 for the different states.

There are many possible ways to generate the reference currents used in the present invention. In one embodiment of the invention, the step of generating the reference currents focuses on defining the upper and  
 30 lower current window levels for the fully erased state "11". However, the circuits for generating the reference currents are used to define all of the states of the memory cells and are described in detail below with reference to Figs. 4-7. With reference to Fig. 3, a  
 35 first reference current  $I_{11H}$  18 is generated to define the

-13-

highest current of a current window 12, and a second reference current  $I_{11L}$  19 is generated to define the lowest current of a current window 12. Current window 12 defines a first reference level corresponding to the fully erased state "11". Any memory cell having an amount of charge on its floating gate that corresponds to a conduction current during a read operation that is greater than the first reference current is in the overerased state 13. Any memory cell having an amount of charge on its floating gate corresponding to a conduction current that is between the first reference current and the second reference current is in the erased state 12. Zone 16 is a forbidden zone between the erased state and the first programmed state, as discussed above. By defining the reference voltage  $I_{11H}$  18 and  $I_{11L}$  19, this provides a target window for determining which memory cells are overerased. The techniques for generating the reference currents 18, 19 are described below with respect to generating reference currents for the entire memory device.

With reference to Fig. 4, the reference currents  $I_{R00}$ ,  $I_{R01}$ ,  $I_{R10}$  and  $I_{R11}$  are generated using reference memory cells in a reference array 21. There are different options, including using a single reference cell to generate all reference currents and using a separate reference cell to generate each of the reference currents. (An intermediate option where two or more cells generate multiple reference currents is also possible, especially where there are 8 or more states.) Where separate cells generate the several reference currents, these currents are supplied by the reference memory cells along lines 23A-23D through switches 25 to the reference current output lines 27A-27D and then to sense amplifiers 51 associated with an analog circuit block 53. The reference memory cells are pre-programmed

-14-

by the manufacturer to the respective states "11", "10", "01" and "00". These define the center of each state's read and program window.

When a reference memory cell is used to  
5 generate multiple reference currents, including the case where a single reference memory cell generates all of the reference currents for all of the states to be defined, the current conducted by that reference memory cell is supplied over an alternative line 33 to a reference sense  
10 amplifier 35 to produce a BIAS voltage for a current mirror circuit 37. For a four state per cell memory, the single reference memory cell corresponds to the fully erased "11" state (minimum floating gate charge, minimum threshold voltage, maximum conduction current). The  
15 various reference currents produced by the current mirror circuit 37 are supplied on lines 45A-45D through a set of switches 47 to the reference current output lines 27A-27D and then to the sense amplifiers 51 associated with the  
aforementioned analog circuit block 53. Only one set of  
20 switches 25 and 47 are set to pass current to the sense amplifiers 51, as determined by whether one or all reference memory cells are used to generate the reference currents.

With reference to both Figs. 4 and 5, the  
25 reference sense amplifier 35 and current mirror circuit 37 generate reference currents for multiple memory states from a single reference memory cell of state "11" in the reference array 21. The cell conduction current  $I_{cell}$  flows, through y-select pass transistors 31, along line  
30 33 from the sense amplifier 35 to that reference memory cell. A typical sense amplifier can include a diode-connected p-channel transistor 61, an n-channel transistor, and an inverter 63, connected to the  
reference memory cell 21 ("11"), so that the current  $I_{cell}$   
35 produces a BIAS voltage at the node between the p-channel

-15-

and n-channel transistors 61 and 62, with the BIAS voltage creating a mirror current  $I_{\text{cell}}$  through p-channel pull-up transistor 39 (because both transistors 61 and 39 have the same drain-to-gate voltage drop). A diode-connected n-channel transistor 41 connected between the

5 transistor 39 and ground has a gate-to-source voltage drop  $V_{\text{GS}}$  determined by the mirror current  $I_{\text{cell}}$  conducted through transistor 41. This voltage drop  $V_{\text{GS}}$  is applied to the gates of four other n-channel transistors 43A-43D.

10 The transistor 41 has a specific channel size (for convenience represented by the numeral 8). Transistors 43A-43D have different channel sizes, at most one (but not necessarily any) of which may be the same size as transistor 41 (here represented by numerals 8, 6, 4 and

15 2). The same applied voltage drop  $V_{\text{GS}}$  generates different currents proportional to their channel sizes. In the example used here, conduction current  $I_{\text{R11}}$  through transistor 43A of size 8 (equal in this example to the channel size of transistor 41) is equal to  $I_{\text{cell}}$ .

20 Conduction current  $I_{\text{R10}}$  through transistor 43B of channel size 6 is  $\frac{3}{4} I_{\text{cell}}$ . Conduction current  $I_{\text{R01}}$  through transistor 43C of channel size 4 is  $\frac{1}{2} I_{\text{cell}}$ . Finally, conduction current  $I_{\text{R00}}$  through transistor 43D of size 2 is  $\frac{1}{4} I_{\text{cell}}$ . The resulting conduction currents  $I_{\text{R11}}$ ,  $I_{\text{R10}}$ ,

25  $I_{\text{R01}}$  and  $I_{\text{R00}}$  are placed on lines 45A-45D as previously mentioned. These sizes and current values are representative, and can be selected by manufacturing transistors 43A-43D of any desired set of sizes. The reference currents need not have a linear relationship.

30 Indeed, it might be desired to have the separation between  $I_{\text{R11}}$  and  $I_{\text{R10}}$  greater than the separation between other reference currents due to the rapidity at which charge is initially injected into fully erased memory floating gates compared to the slower injection of

35 partially charged floating gates. In that case the

-16-

relative sizes of transistors 43A-43D might be approximately 8, 5, 3 and 1.

With reference to Fig. 6, for any given reference current  $I_{Ri}$ , a fractional current  $(1-m)I_{Ri}$  for  
5 some margin value  $m$  can be produced using a current mirror technique like that used to generate multiple reference currents from a single current in Figs. 4 and 5 above. Likewise, for any given reference current  $I_{Rj}$ , a fractional current  $(m)I_{Rj}$  can also be produced. In this  
10 technique the diode-connected p-channel transistors 71 and 72 of a first channel size (designated by numeral 1) produce a voltage drop dependent upon the input reference current  $I_{Ri}$  or  $I_{Rj}$ . P-channel transistors 73 and 74 have their gates connected to the gate-source of respective  
15 transistors 71 and 72. Transistors 73 and 74 have respective channel sizes of  $(1-m)$  and  $(m)$  relative to that of transistors 71 and 72. With the same drain-gate voltage drop across transistors 73 and 74 as across corresponding transistors 71 and 72, the transistors 73 and 74 conduct respective  $(1-m)I_{Ri}$  and  $(m)I_{Rj}$  currents.  
20 These currents can be combined with a simple connection to generate a current  $(1-m)I_{Ri} + (m)I_{Rj}$ . This combined current can likewise be sensed using a current mirror and sense amplifier elements 75-77 whose operation is like  
25 that described above for elements 35 and 39 in Fig. 5. The resulting current output may be compared during a read or verify step with a current from an addressed memory cell. For reads,  $m = 0.5$ , while for program verify operations  $m < 0.5$ . (Fig. 7 shows a preferred  
30 embodiment that allows a user to select  $m$  from any of 0.125, 0.25, 0.375 or 0.5, with the latter used only during read operations.) Also for verify, the lower bound of the allowed current of a state uses  $j = i-1$ . As previously discussed, there is no lower current boundary  
35 for the fully programmed "00" state. The upper bound of

-17-

the allowed current of a state during a program verify operation uses  $j = i+1$ , except for the fully erased state "11". For the fully erased state "11",  $I_{11H} = (1+m)I_{R11}$  is generated by summing three currents,  $(1-m)I_{R11}$ ,  $(m)I_{R11}$  and again  $(m)I_{R11}$ .

With reference to Fig. 7, one element of the analog circuit block 53 of Fig. 4 receives a reference current  $I_{Ri}$  for a state  $i$  and generates fractional currents  $(1-m)I_{Ri}$  and  $(m)I_{Ri}$  for a selected margin value  $m$  for use as in Fig. 6 to generate the read and verify currents set forth in Fig. 2 as defined above. The element shown in Fig. 7 is repeated so that each of the reference currents  $I_{R11}$ ,  $I_{R10}$ ,  $I_{R01}$  and  $I_{R00}$  has its own fractional current generating element. A single set of fractional currents  $(1-m)I_{R00}$  and  $(m)I_{R00}$  is needed from reference current  $I_{R00}$ . Two of each of the fractional currents  $(1-m)I_{R01}$ ,  $(m)I_{R01}$ ,  $(1-m)I_{R10}$ ,  $(m)I_{R10}$ , and  $(1-m)I_{R11}$  are needed to be generated from the reference currents  $I_{R01}$ ,  $I_{R10}$  and  $I_{R11}$ . Three sets of the fractional current  $(m)I_{R11}$  are needed, one for use in the verify current  $I_{11L}$  and two for use in the verify current  $I_{11H}$  in order to get  $(1+m)I_{R11}$ . Since read and verify current levels are never needed at the same time, the read current levels  $I_H$ ,  $I_M$  and  $I_L$  can share the same circuit elements as that used to generate the verify levels without additional overhead. The margin value  $m$  is merely switched to 0.50 when a read operation is indicated.

In Fig. 7, a sense amplifier including a diode-connected p-channel transistor 81 conducts the supplied reference current  $I_{Ri}$  and generates a drain-gate voltage drop and a corresponding BIAS voltage on line 82 that drives the gates of transistors  $83_1-83_8$ ,  $84_1-84_8$ ,  $85_1-85_4$  and  $86_1-86_4$ . The transistor 81 has a channel size, represented here by the numeral 8, that corresponds to the denominator of the fraction for the generated

-18-

fractional currents. Each of the two dozen other transistors 83-86 has a channel size, represented by the numeral one, that has in this example one-eighth of the current conduction  $I_{Ri}$  of the sense amplifier's transistor 81. The p-channel sources of each set of transistors 83<sub>1</sub>-83<sub>8</sub>, 84<sub>1</sub>-84<sub>8</sub>, 85<sub>1</sub>-85<sub>4</sub> and 86<sub>1</sub>-86<sub>4</sub> are connected to generate a cumulative fractional current  $(1-m)I_{Ri}$  or  $(m)I_{Ri}$  according to the number of active transistors in a particular set, where  $m$  is either 1/8, 2/8, 3/8 or 4/8 in this example. A set of transistors 91<sub>1</sub>-91<sub>4</sub>, 92<sub>1</sub>-92<sub>4</sub>, 93<sub>1</sub>-93<sub>4</sub> and 94<sub>1</sub>-94<sub>4</sub> serve as switches that selectively connect or disconnect certain transistors 83<sub>5</sub>-83<sub>8</sub>, 84<sub>5</sub>-84<sub>8</sub>, 85<sub>1</sub>-85<sub>4</sub>, and 86<sub>1</sub>-86<sub>4</sub> to the power supply line. These switches are controlled by control signals ADD0-ADD3 and SUB0-SUB3 provided to their transistors. Additional transistors 89<sub>1</sub>-89<sub>4</sub> and 90<sub>1</sub>-90<sub>4</sub> are provided for symmetry so that all paths contributing to the current output are identical, but these additional transistors have their gates tied to ground instead of receiving control signals. ADD0-ADD3 control the switches 93<sub>1</sub>-93<sub>4</sub> and 94<sub>1</sub>-94<sub>4</sub> in the opposite manner from the corresponding SUB0-SUB3 control signals' control of switches 91<sub>1</sub>-91<sub>4</sub> and 92<sub>1</sub>-92<sub>4</sub>. Thus, when only four transistors 83<sub>1</sub>-83<sub>4</sub> are active in the first set for  $(1-m) = 4/8$ , all four transistors 85<sub>1</sub>-85<sub>4</sub> in the corresponding set will be active for  $m = 4/8$ ; when five transistors 83<sub>1</sub>-83<sub>5</sub> are active for  $(1-m) = 5/8$ , only three transistors 85<sub>2</sub>-85<sub>4</sub> will be active for  $m = 3/8$ ; etc. This ensures that  $(1-m) + m = 1$  in actual implementation. The fractional currents are then combined as discussed above with reference to Fig. 6. The second bank of transistors in Fig. 7, 84<sub>1</sub>-84<sub>8</sub>, 86<sub>1</sub>-86<sub>4</sub>, 92<sub>1</sub>-92<sub>4</sub>, and 94<sub>1</sub>-94<sub>4</sub>, is not needed for reference current  $I_{R00}$ , as only one set of fractional currents is needed.

For reference current  $I_{R11}$ , the fractional current generating circuit element in Fig. 7, is modified

-19-

to include a third partial bank driven by the BIAS voltage and controlled by ADD0-ADD3 in a manner like transistors  $85_1-85_4$  and  $86_1-86_4$ . This modification provides three fractional current versions of  $(m)I_{R11}$ ,  
5 needed to produce  $I_{11H} = (1+m)I_{R11} = (1-m)I_{R11} + (m)I_{R11} + (m)I_{R11}$ . Alternatively, the fractional current generating circuit element in Fig. 7 can be modified so that control signals SUB0-SUB3 may be independently supplied to the two banks (rows) of transistors, whereby in the (erase) verify mode of operation, the transistors  $83_1-83_8$  supply  
10  $(1)I_{R11}$  (here  $i="11"$ ) instead of  $(1-m)I_{R11}$ , while transistors  $84_1-84_8$  still provide current  $(1-m)I_{R11}$ . This enables the current level  $I_{11H} = (1+m)I_{R11} = (1)I_{R11} + (m)I_{R11}$  to be provided by the first bank or row of the circuit,  
15 while  $I_{11L}$  and  $I_{10L}$  are still supplied with the current contributions  $(1-m)I_{R11}$  and  $(m)I_{R11}$ , respectively, from the second bank or row of that circuit.

Referring back to Fig. 1, the next step 302 in the method of the present invention is to determine which  
20 of the memory cells in the memory array have been overerased. This is done by reading the contents of the cell in the usual manner to determine the memory cell states of the memory cells. Again, if the conduction current read during the read operation exceeds that of  
25 the highest reference current of the erased state current window, then the memory cell is considered to be overerased.

Then, in step 303, the overerased cells are programmed until all of the overerased memory cells are  
30 in the erased state. With reference to Fig. 8, the programming method used can be the staircase programming method described above with reference to Fig. 10. As shown in Fig. 8, an alternating sequence of programming  
241, 242, 243 and verifying 251, 252, 253 pulses are  
35 applied to the memory cell. The voltage 230 of each

-20-

programming pulse incrementally increases with respect to time 232 overerased state 250 to the erased state 260.

One advantage of the present invention is that by defining the erased state to be like one of the programming states, the erased state can be programmed using normal multi-level programming methods and programming voltages. Additionally, speed and precision are increased, as using the regular programming voltage is faster than soft-programming techniques.

-21-

## Claims

1. A method of recovering overerased bits in a memory device following an erase operation, said method  
5 comprising:

generating a first reference current in the memory device, said first reference current defining a highest current of a current window corresponding to a first reference level defining a state in which a memory  
10 cell is in an erased state,

determining from a plurality of memory cells a first group of memory cells that have an amount of charge on its floating gate corresponding to a conduction current that is greater than the first reference current,  
15 said first group of memory cells being in an overerased state; and

applying a plurality of programming pulses to each of the first group of memory cells to place an amount of charge on the floating gate of said cell until  
20 the conduction current is less than the first reference current, such that each of the first group of memory cells is programmed from the overerased state to the erased state.

25

2. The method of Claim 1 further comprising:

applying a plurality of verify pulses to the specified memory cell, each one of said verify pulses being applied after each one of said programming pulses  
30 in an alternating manner.

3. The method of Claim 1 wherein said programming pulses increase incrementally in voltage to increase the amount  
35 of charge on the floating gate at a steady rate.

-22-

4. The method of Claim 1 further comprising:

generating a second reference current in the memory device, said second reference current defining a lowest current of a current window corresponding to the first reference level defining the state in which a memory cell is in the erased state.

5. The method of Claim 1 wherein the memory cell has two memory states per cell.

6. The method of Claim 1 wherein the memory cell has four memory states per cell.

7. The method of Claim 1 wherein the memory cell is a multi-level memory cell.

8. The method of Claim 1 wherein the step of determining the first group includes reading the plurality of memory of cells and comparing the conduction current of each memory cell read to the first reference current.

9. An apparatus for recovering overerased bits in a memory device following an erase operation, comprising:

means for generating a first reference current in the memory device, said first reference current defining a highest current of a current window corresponding to a first reference window corresponding to a first reference level defining a state in which a memory is in an erased state;

-23-

means for reading a plurality of memory cells to determine a first group of memory cells that have an amount of charge on its floating gate corresponding to a conduction current that is greater than the first  
5 reference current, said first group of memory cells being in an overerased state; and

means for applying a plurality of programming pulses to each of the first group of memory cells to place an amount of charge on the floating gate of said  
10 cell until the conduction current is less than the first reference current, such that each of the first group of memory cells is programmed from the overerased state to the erased state.

15

10. The apparatus of Claim 9 further comprising:

means for applying a plurality of verify pulses to each of the first group of memory cells to determine whether each of said cells is still in an overerased  
20 state, each one of said verify pulses being applied after each one of the said programming pulses in an alternating manner.

25

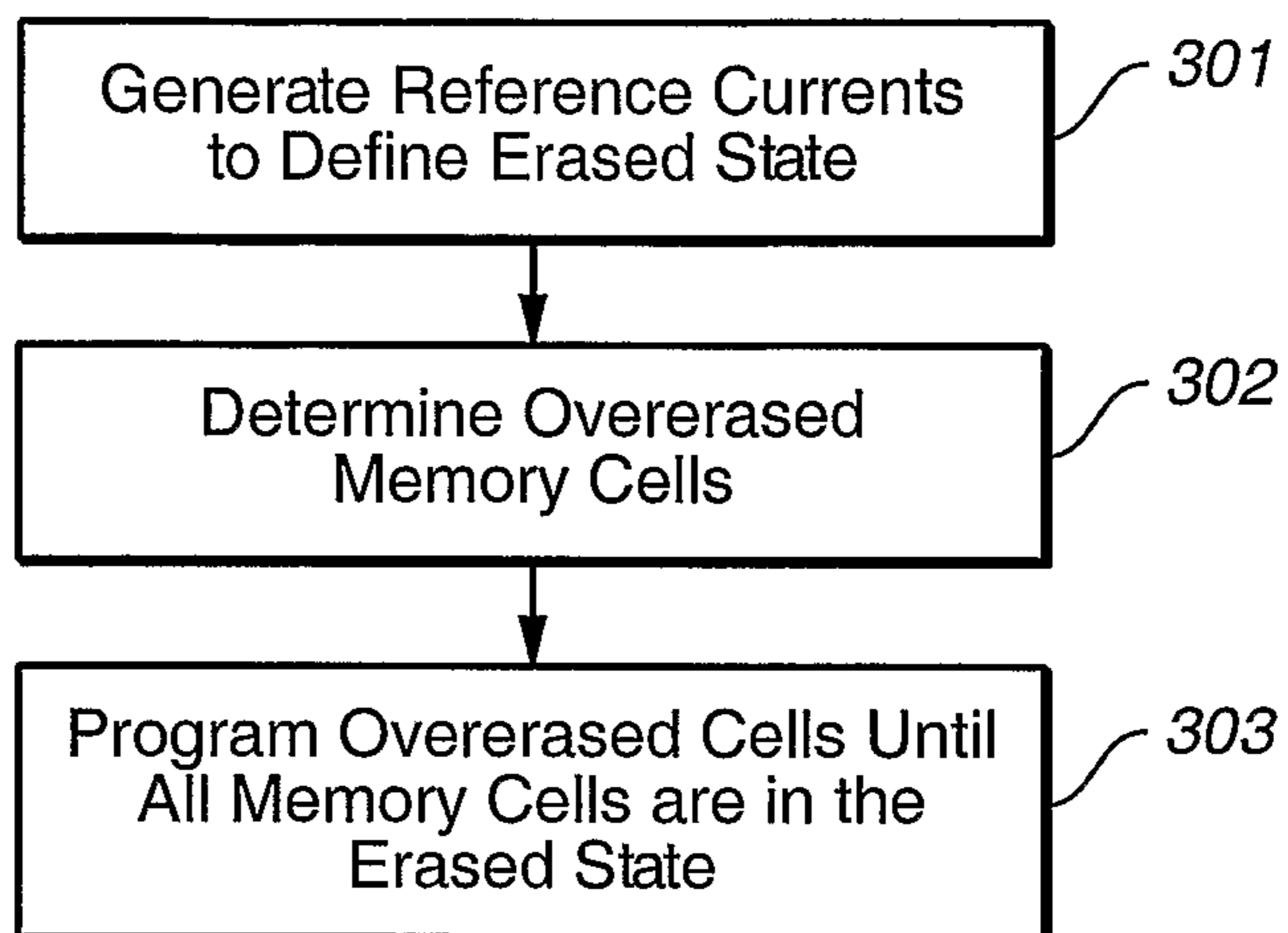
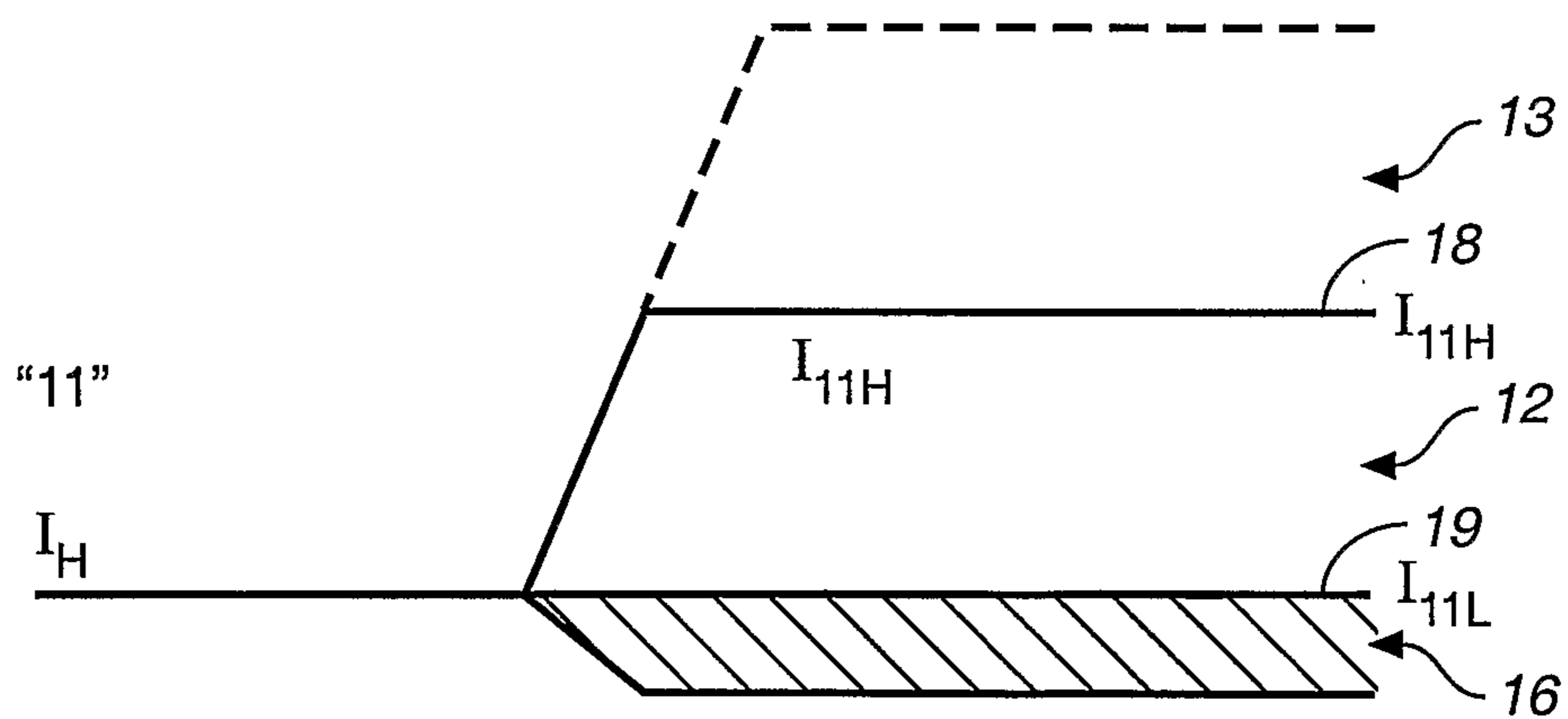
11. The apparatus of Claim 10 further comprising:

means for generating a second reference current in the memory device, said second reference current defining a lowest current of a current window  
30 corresponding to the first reference level defining the state in which a memory cell is in the erased state.

-24-

12. The apparatus of claim 9 wherein said means for  
applying programming pulses includes means for  
incrementally increasing the voltage of said programming  
pulses so as to increase the amount of charge on the  
5 floating gates of said first group of memory cells at a  
steady rate.

1 / 9

**FIG. 1****FIG. 3**

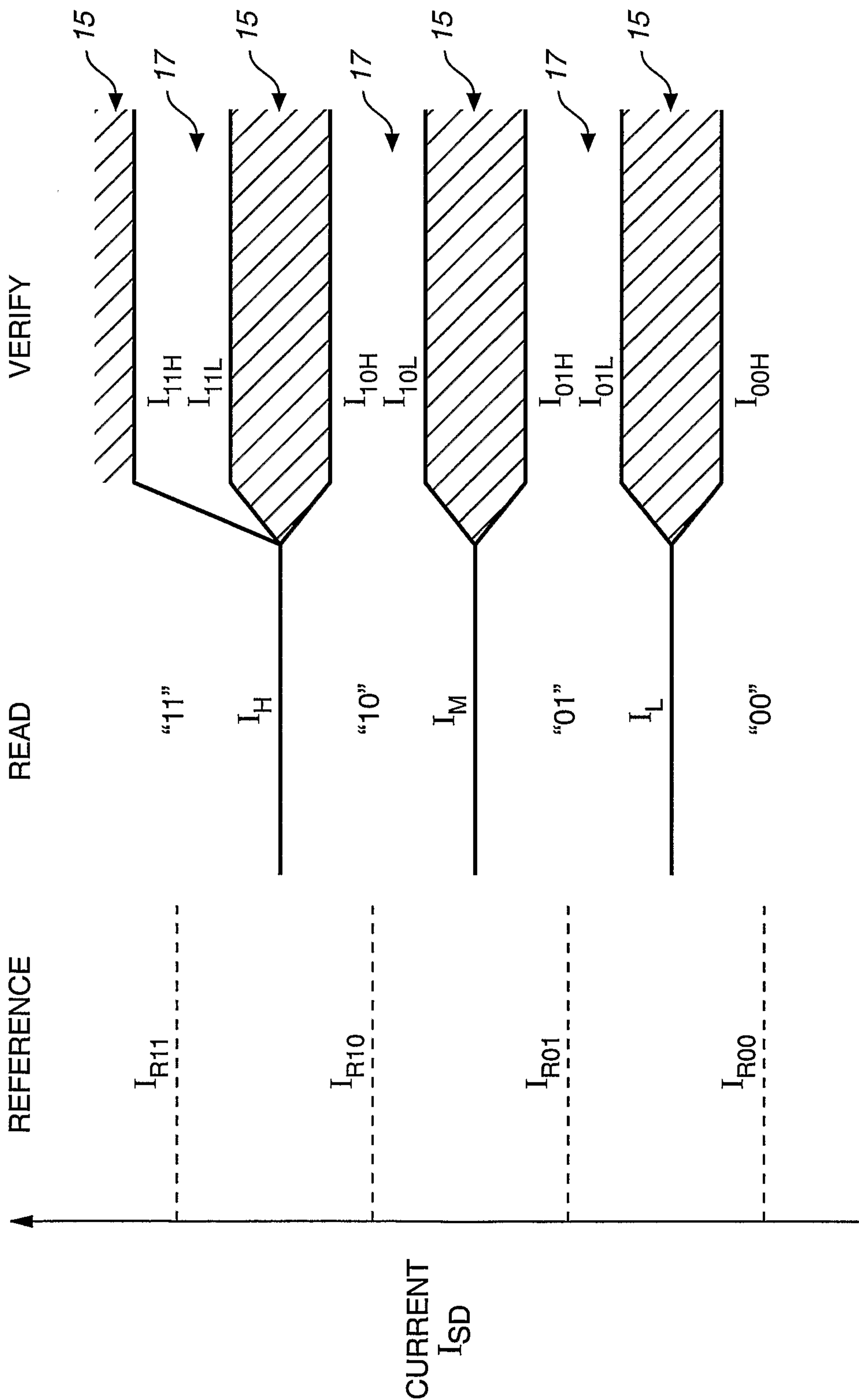


FIG.-2

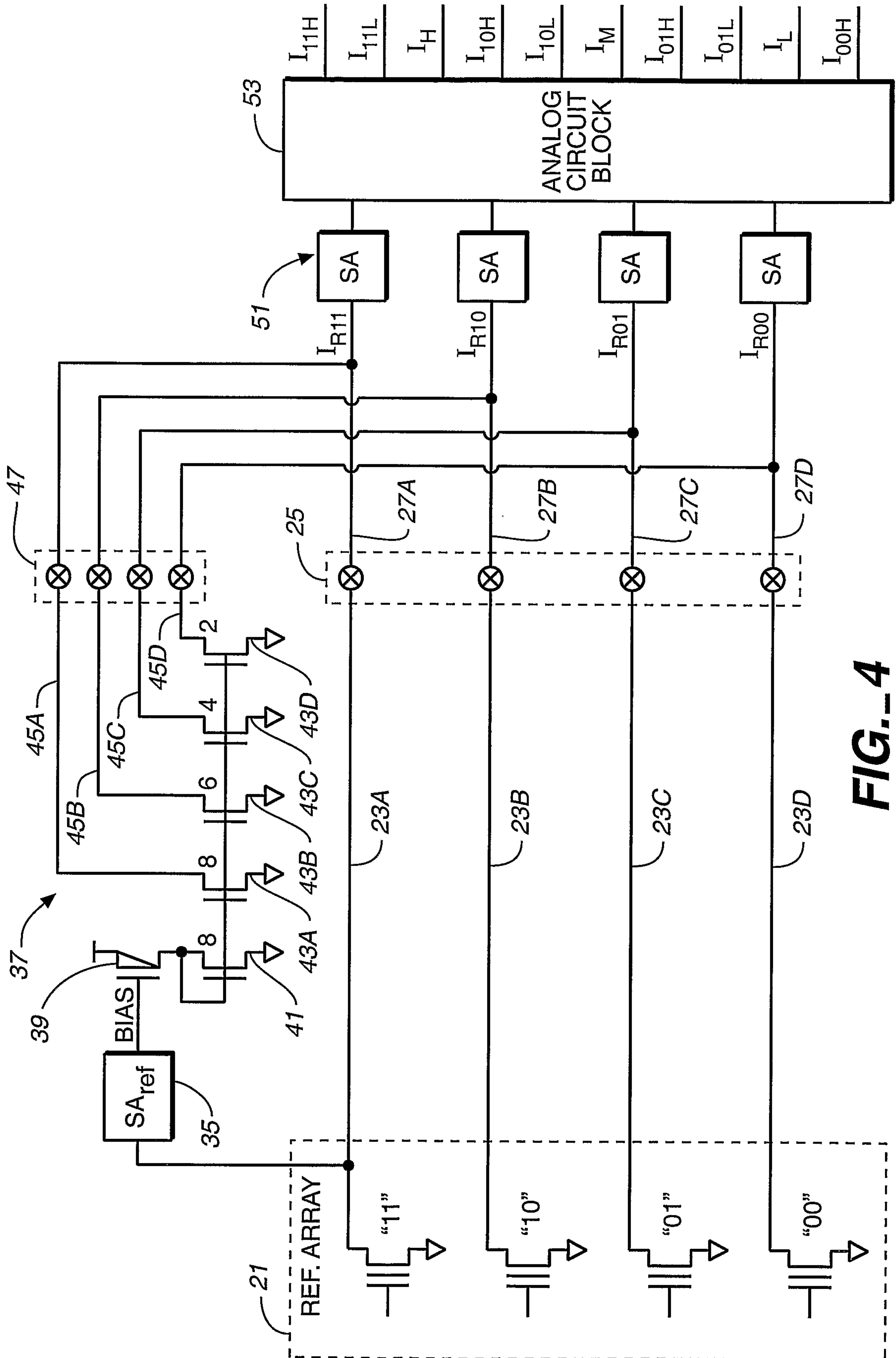
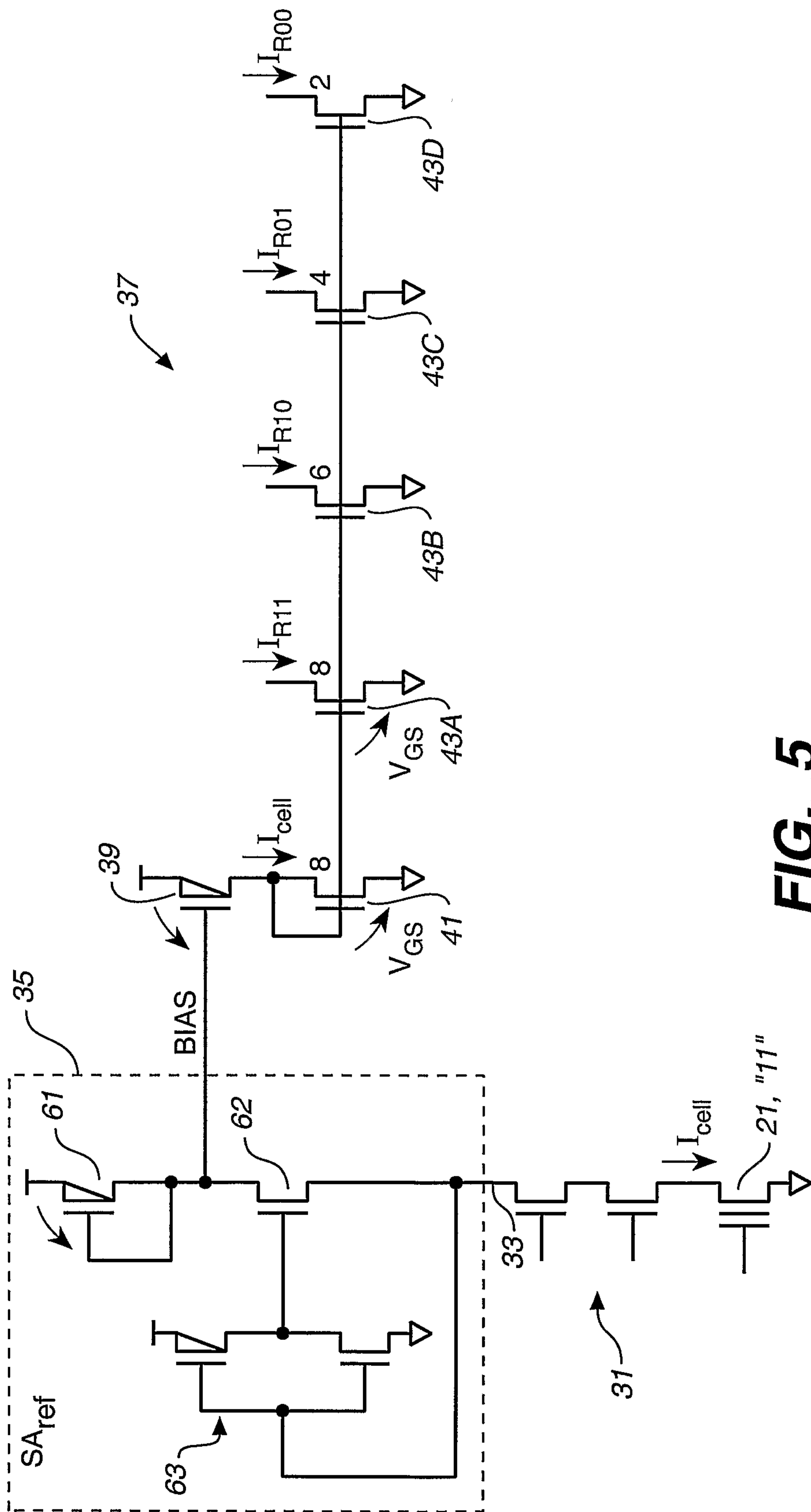
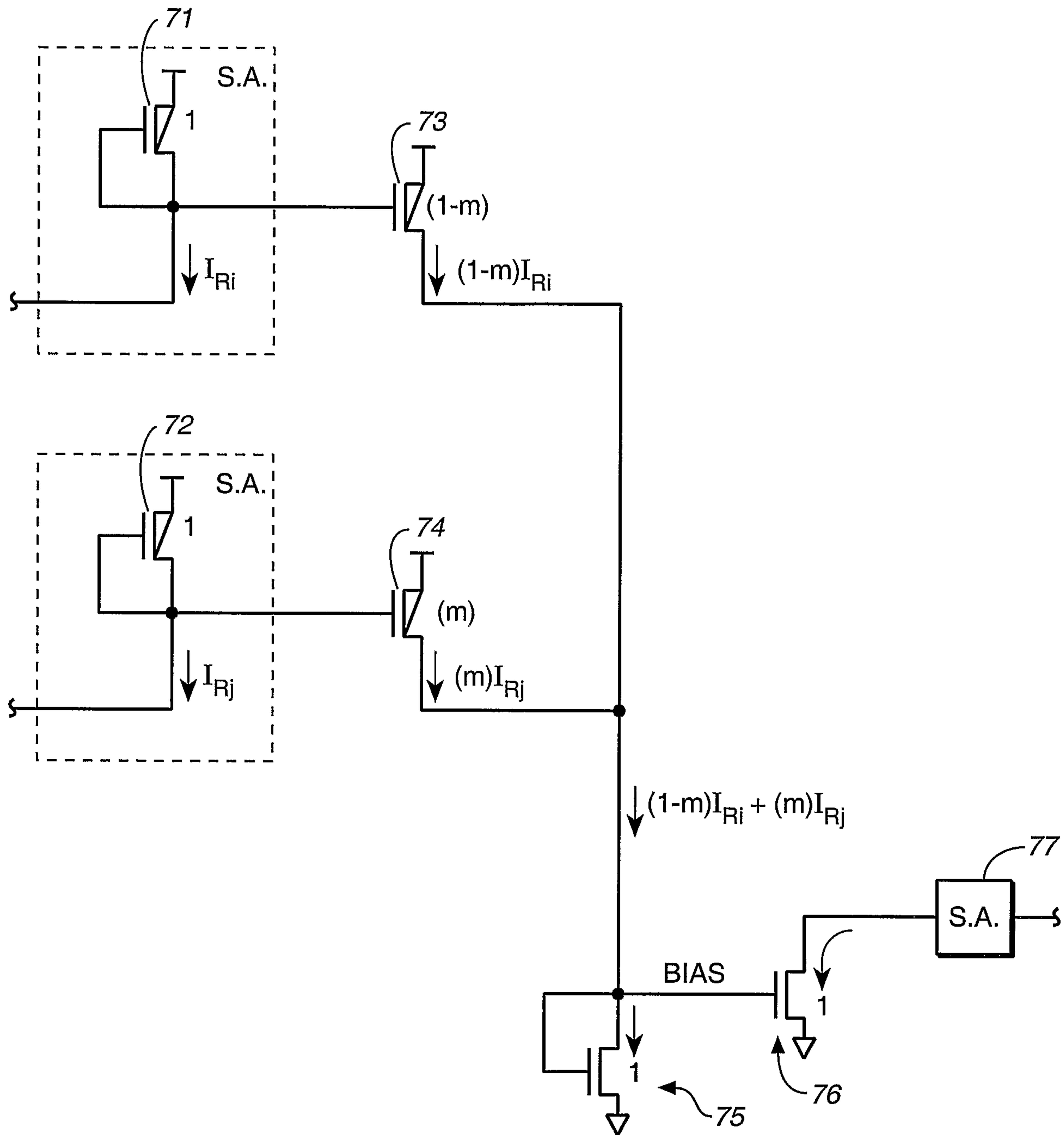


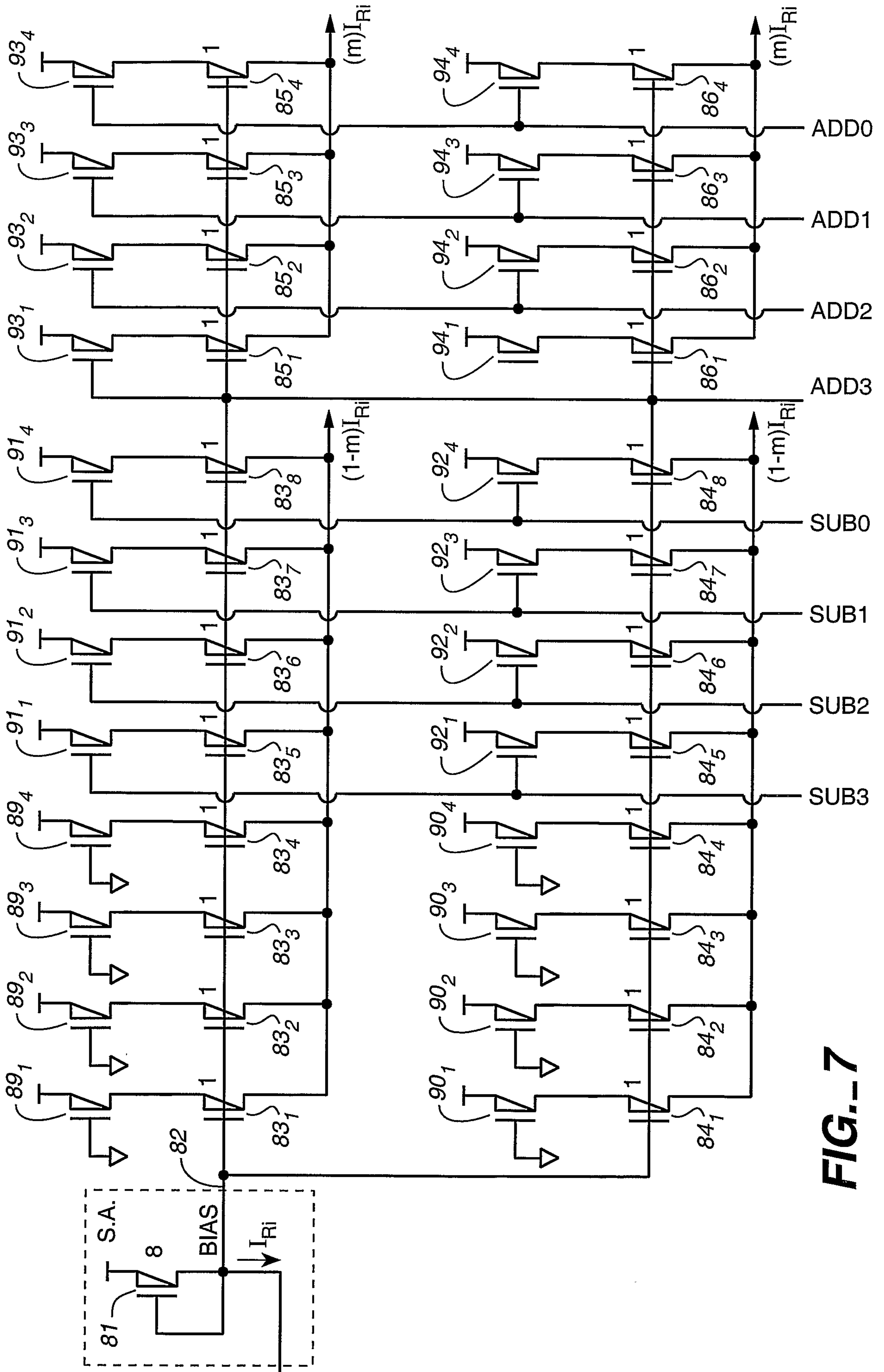
FIG. 4



**FIG. 5**

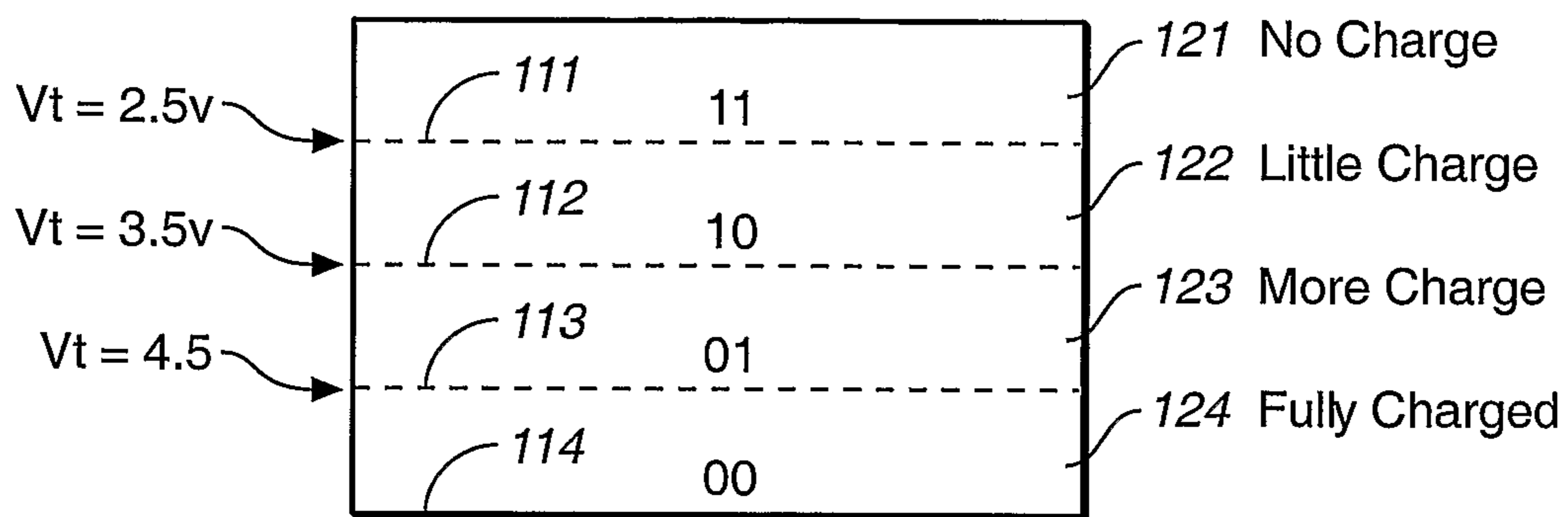
5 / 9

**FIG. 6**

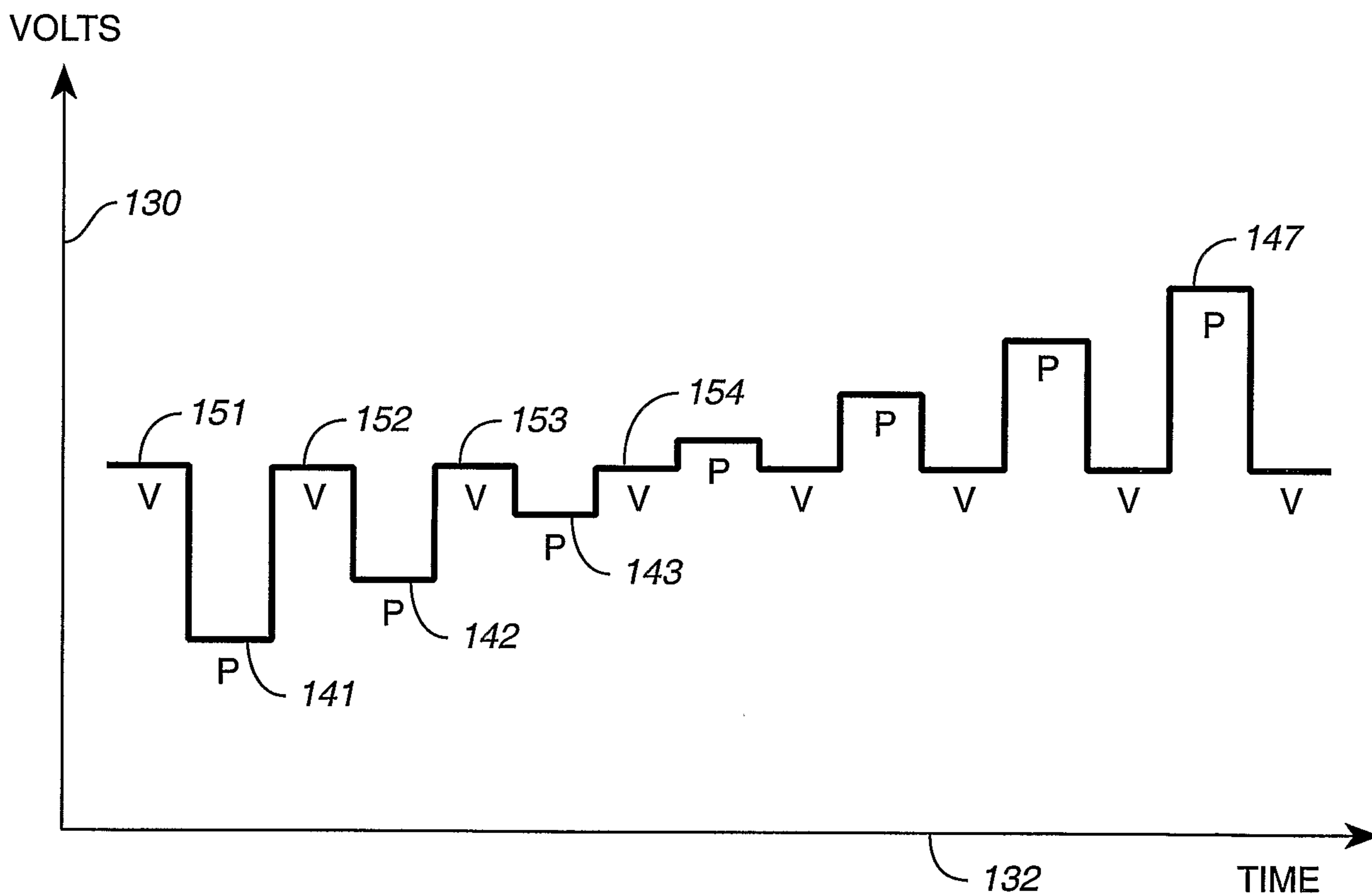


**FIG. 7**

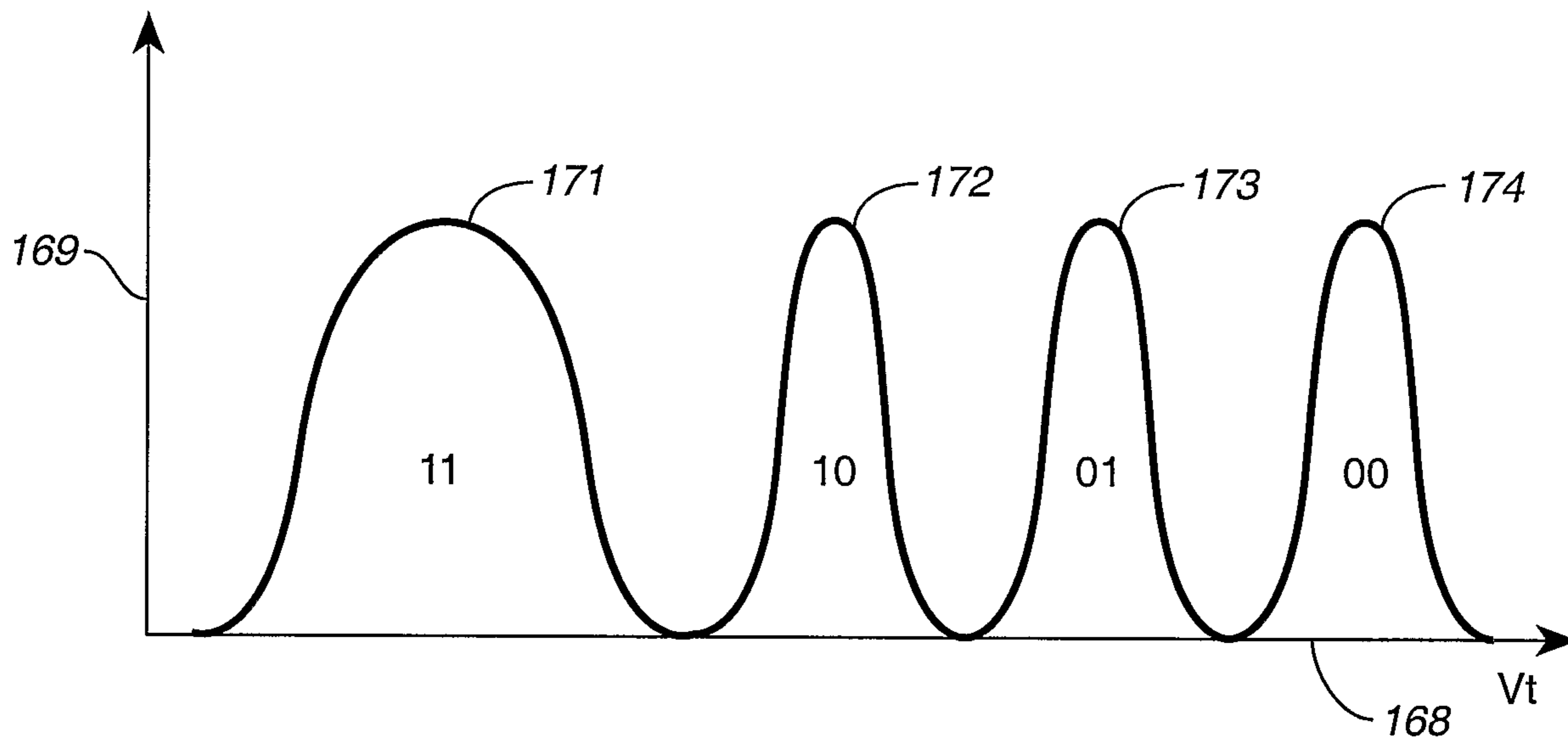




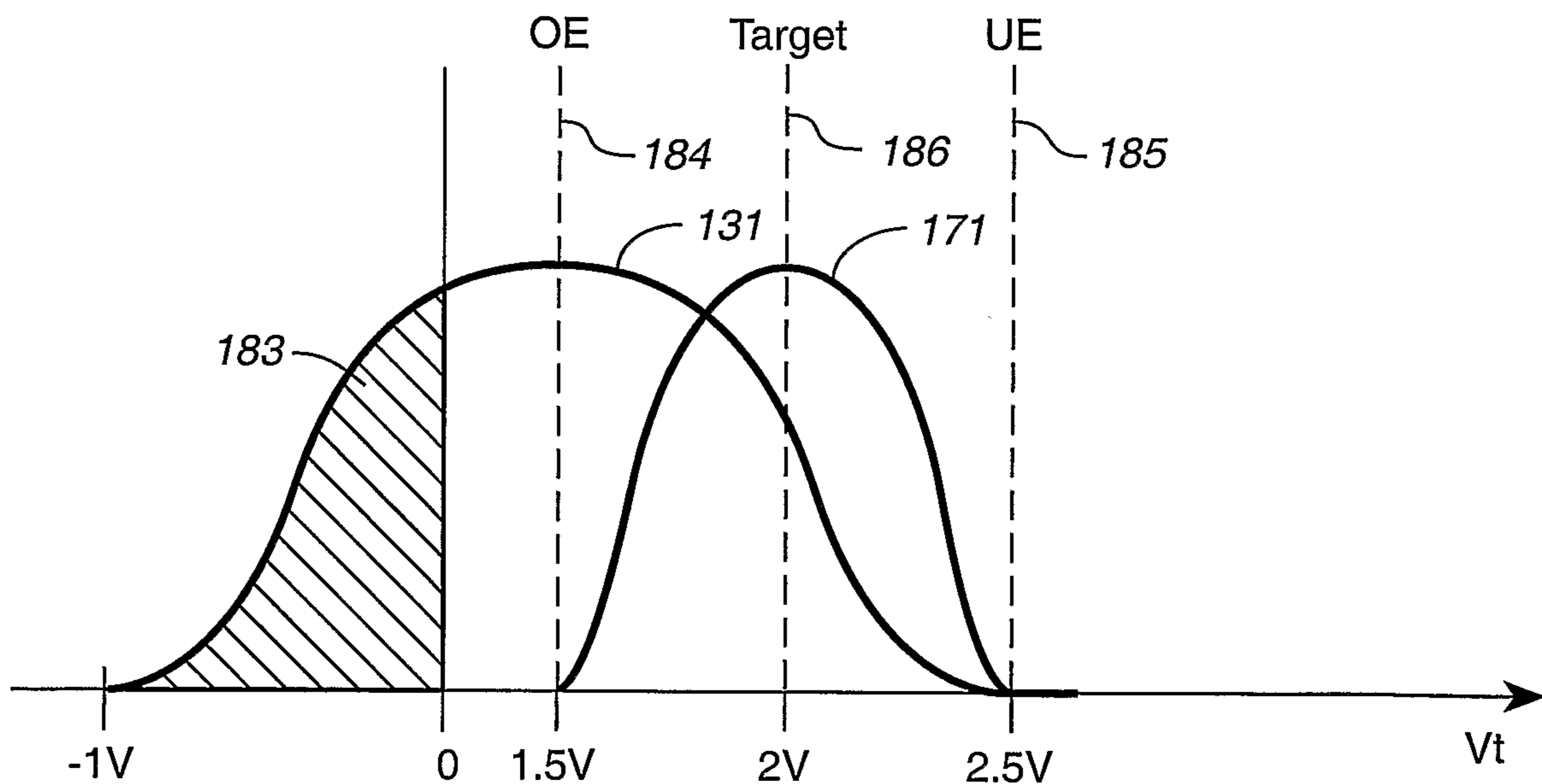
**FIG. 9**  
(PRIOR ART)



**FIG. 10**  
(PRIOR ART)



**FIG. 11**  
(PRIOR ART)



**FIG. 12**  
(PRIOR ART)

