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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 January 2006 (05.01.2006)

PCT

(10) International Publication Number
WO 2006/001979 A1

(51) International Patent Classification[7]: G11C 16/04, 16/10

(21) International Application Number: PCT/US2005/019393

(22) International Filing Date: 2 June 2005 (02.06.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/868,147 15 June 2004 (15.06.2004) US

(71) Applicant (*for all designated States except US*): **SANDISK CORPORATION** [US/US]; 140 Caspian Court, Sunnyvale, CA 94089 (US).

(72) Inventor; and

(75) Inventor/Applicant (*for US only*): **GUTERMAN, Daniel, C.** [US/US]; 305 Jacaranda Drive, Fremont, CA 94539 (US).

(74) Agent: **MAGEN, Burt**; Vierra Magen Marcus Harmon & DeNiro, LLP, Suite 540, 685 Market Street, San Francisco, CA 94105 (US).

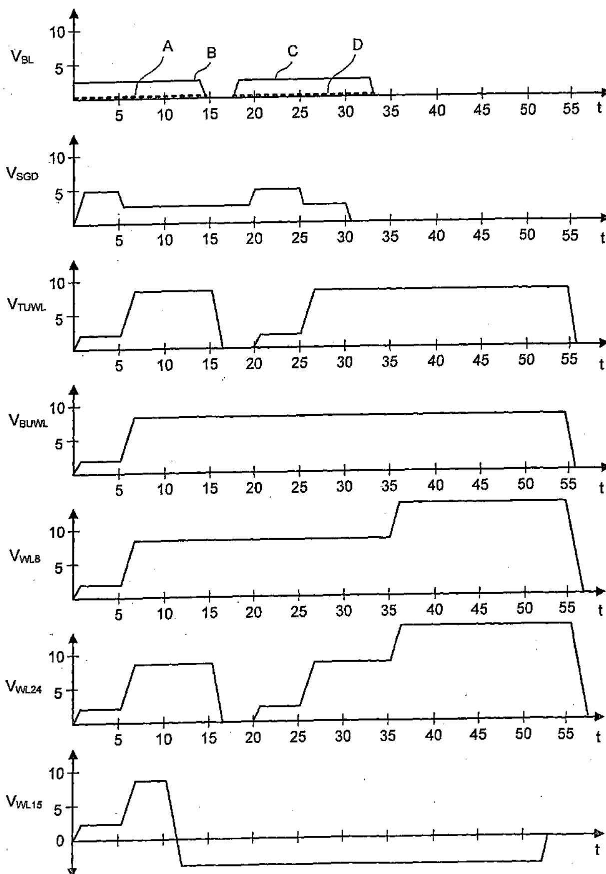
(81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

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

Published:
— *with international search report*

[Continued on next page]

(54) Title: CONCURRENT PROGRAMMING OF NON-VOLATILE MEMORY



(57) Abstract: One embodiment of the present invention includes applying a first value to a bit line (A or B. Fig. 12), boosting word lines (WLO/WL31 on Fig 9) associated with the bit line (DRAIN of Tr. 566 on Fig 9) and a common selection line (source of Tr. 500 on Fig 9) to create a first condition based on the first value, and cutting off a boundary non-volatile storage element (e.g. 532 on Fig 9) associated with the common selection line to maintain the first condition for a particular non-volatile storage element (e.g. 518 on Fig 9) associated with the bit line and common selection line. A second value is applied to the bit line and at least a subset of the word lines are boosted to create a second condition for a different non-volatile storage element (e.g. 550 on Fig 9) associated with the bit line and common selection line. The second condition is based on the second value. The first condition and the second condition overlap in time. Both non-volatile storage elements are programmed concurrently, based on their associated conditions.

WO 2006/001979 A1

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

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.

CONCURRENT PROGRAMMING OF NON-VOLATILE MEMORY

5

CROSS-REFERENCE TO RELATED APPLICATIONS

This Application is related to the following United States Patent
10 Applications, all of which are incorporated herein by reference in their entirety:

"Boosting To Control Programming Of Non-Volatile Memory," Application Serial No. 10/839,764, filed on May 5, 2004, Inventors Daniel C. Guterman, Nima Mokhlesi and Yupin Fong;

"Bitline Governed Approach For Program Control of Non-Volatile
15 Memory," Application Serial No. 10/839,806, filed on May 5, 2004, Inventors Daniel C. Guterman, Nima Mokhlesi and Yupin Fong; and

"Latched Programming Of Memory And Method," Application Serial No. 10/842,941, filed May 10, 2004, Inventor Raul-Adrian Cernea.

20

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] The present invention relates to technology for programming non-volatile memory.

25

Description of the Related Art

[0002] Semiconductor memory devices have become more popular for use in various electronic devices. For example, non-volatile semiconductor memory is used in cellular telephones, digital cameras, personal digital assistants, mobile computing devices, non-mobile computing devices and other

devices. Electrical Erasable Programmable Read Only Memory (EEPROM) and flash memory are among the most popular non-volatile semiconductor memories.

[0003] Both EEPROM and flash memory utilize a floating gate that is positioned above and insulated from a channel region in a semiconductor substrate. The floating gate is positioned between source and drain regions. A control gate is provided over and insulated from the floating gate. The threshold voltage of the transistor is controlled by the amount of charge that is retained on the floating gate. That is, the minimum amount of voltage that must be applied to the control gate before the transistor is turned on to permit conduction between its source and drain is controlled by the level of charge on the floating gate.

[0004] Some EEPROM and flash memory devices have a floating gate that is used to store two ranges of charges and, therefore, the memory cell can be programmed/erased between two states (an erased state and a programmed state). When programming an EEPROM or flash memory device, such as a NAND flash memory device, typically a program voltage is applied to the control gate and the bit line is grounded. Electrons from the channel are injected into the floating gate. When electrons accumulate in the floating gate, the floating gate becomes negatively charged and the threshold voltage of the memory cell is raised so that the memory cell is in the programmed state. More information about programming can be found in U.S. Patent Application 10/379,608, titled "Self Boosting Technique," filed on March 5, 2003; and in U.S. Patent Application 10/629,068, titled "Detecting Over Programmed Memory," filed on July 29, 2003, both applications are incorporated herein by reference in their entirety.

[0005] A multi-state flash memory cell is implemented by identifying multiple, distinct allowed programmed threshold voltage ranges separated by

forbidden voltage ranges. Each distinct threshold voltage range corresponds to a predetermined value for the set of data bits.

[0006] Typically, the program voltage applied to the control gate is applied as a series of pulses. The magnitude of the pulses is increased with each successive pulse by a predetermined step size (e.g. 0.2v, 0.4v, or other). In the periods between the pulses, verify operations are carried out. As the number of programmable states increase, the number of verify operations increases and more time is needed. One means for reducing the time burden of verifying is to use a more efficient verify process, such as the process that is disclosed in U.S. Patent Application Serial No. 10/314,055, "Smart Verify for Multi-State Memories," filed December 5, 2002, incorporated herein by reference in its entirety. However, there is a consumer demand for memory devices that program as fast as possible. For example, a user of a digital camera that stores images on a flash memory card does not want to wait between pictures.

[0007] Therefore, there is further need to reduce the amount of time needed to program non-volatile memory.

SUMMARY OF THE INVENTION

[0008] The present invention, roughly described, pertains to technology for reducing the time needed to program non-volatile memory.

[0009] One embodiment of the present invention includes programming a first non-volatile storage element of a group of non-volatile storage elements and programming a second non-volatile storage element of the group of non-volatile storage elements. The programming of the first non-volatile storage element overlaps in time with the programming of the second non-volatile storage element. In one example implementation, the first non-volatile storage

element and the second non-volatile storage element are NAND flash memory elements on the same NAND string.

[0010] Some embodiments of the present invention include establishing a first program condition for a first non-volatile storage element and establishing a second program condition for a second non-volatile storage element while the first program condition persists. The first program condition can be different than the second program condition. The first non-volatile storage element and the second non-volatile storage element are part of a group of non-volatile storage elements associated with a common source/drain control line. The first non-volatile storage element is programmed using the first program condition and the second non-volatile storage element is programmed using said second program condition.

[0011] One example implementation includes applying a first value to a bit line, boosting word lines associated with the bit line to create a first condition based on the first value and cutting off a boundary non-volatile storage element associated with the bit line to maintain the first condition for a particular non-volatile storage element associated with the bit line. A second value is applied to the bit line and at least a subset of the word lines associated with the bit line are boosted to create a second condition for a different non-volatile storage element. The first condition and the second condition exist during overlapping times. Both non-volatile storage elements are programmed based on the associated conditions.

[0012] Various embodiments of the present invention include programming one or more non-volatile storage elements. For example, the present invention can be used to program an array of flash memory devices (or other types of non-volatile storage elements). One embodiment utilizes NAND flash memory. In some example implementations, the programming of the one or more non-volatile storage elements is performed by or at the direction of a

control circuit. The components of the control circuit may differ based on the particular application. For example, a control circuit may include any one of the following components or any combination of two or more of the following components: controller, command circuits, state machine, row control, column control, source control, p-well or n-well control, or other circuits that perform similar functionality.

[0013] These and other objects and advantages of the present invention will appear more clearly from the following description in which the preferred embodiment of the invention has been set forth in conjunction with the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] Figure 1 is a top view of a NAND string.

[0015] Figure 2 is an equivalent circuit diagram of the NAND string.

[0016] Figure 3 is a cross sectional view of the NAND string.

[0017] Figure 4 is a block diagram of one embodiment of a non-volatile memory system in which the various aspects of the present invention are implemented.

[0018] Figure 5 illustrates an example of an organization of a memory array.

[0019] Figure 6 shows threshold voltage distributions for a multi-state non-volatile memory device.

[0020] Figure 7 depicts a programming voltage signal, which includes a series of programming pulses with magnitudes increasing over time.

[0021] Figure 8 depicts three of the programming pulses from the signal of Figure 7, and the verification pulses between the programming pulses.

[0022] Figure 9 depicts a NAND string.

[0023] Figure 10 is a flow chart describing one embodiment of a process
5 for programming flash memory.

[0024] Figure 11 is a flow chart describing one embodiment of a process for establishing program conditions.

[0025] Figure 12 is a timing diagram describing the behavior of various signals during the program process.

10 [0026] Figure 13 is a flow chart describing one embodiment of a process for establishing program conditions.

[0027] Figure 14 is a flow chart describing one embodiment of a process for verifying.

DETAILED DESCRIPTION

15 [0028] The invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings in which like references indicate similar elements. It should be noted that references to an or one embodiment in this disclosure are not necessarily the same embodiment, and such references mean at least one.

20 [0029] In the following description, various aspects of the present invention will be described. However, it will be apparent to those skilled in the art that the present invention may be practiced with only some or all of the aspects of the present disclosure. For purposes of explanation, specific numbers, materials and configurations are set forth in order to provide a
25 thorough understanding of the invention. However, it will be apparent to one

skilled in the art that the present invention may be practiced without all of the specific details. In other instances, well known features are omitted or simplified in order not to obscure the present invention.

[0030] Various embodiments will be described as multiple discreet steps in turn, in a manner that is most helpful in understanding the present invention. However, the order of this description should not be construed as to imply that these operations are necessarily order dependent.

[0031] One example of a memory system suitable for implementing the present invention uses the NAND flash memory structure, which includes arranging multiple transistors in series between two select gates. The transistors in series and the select gates are referred to as a NAND string. Figure 1 is a top view showing one NAND string. Figure 2 is an equivalent circuit thereof. The NAND string depicted in Figures 1 and 2 includes four transistors 100, 102, 104 and 106 in series and sandwiched between a first select gate 120 and a second select gate 122. Select gate 120 connects the NAND string to bit line 126. Each of the channels for the transistors on the NAND string are effected by the bit line 126. Select gate 122 connects the NAND string to source line 128. Select gate 120 is controlled by the applying appropriate voltages to control gate 120CG. Select gate 122 is controlled by applying the appropriate voltages to control gate 122CG. Each of the transistors 100, 102, 104 and 106 has a control gate and a floating gate. Transistor 100 has control gate 100CG and floating gate 100FG. Transistor 102 includes control gate 102CG and floating gate 102FG. Transistor 104 includes control gate 104CG and floating gate 104FG. Transistor 106 includes a control gate 106CG and floating gate 106FG. Control gate 100CG is connected to word line WL3, control gate 102CG is connected to word line WL2, control gate 104CG is connected to word line WL1, and control gate 106CG is connected to word line WL0. In one embodiment, transistors 100, 102, 104 and 106 are each memory cells. In other embodiments, the memory

cells may include multiple transistors or may be different than that depicted in Figs 1 and 2. Select gate 120 is connected to select line SGD. Select gate 128 is connected to select line SGS.

[0032] Figure 3 provides a cross-sectional view of the NAND string described above. As depicted in Figure 3, the transistors of the NAND string are formed in p-well region 140. Each transistor includes a stacked gate structure that consists of the control gate (100CG, 102CG, 104CG and 106CG) and a floating gate (100FG, 102FG, 104FG and 106FG). The floating gates are formed on the surface of the p-well on top of an oxide or other dielectric film. The control gate is above the floating gate, with an inter-polysilicon dielectric layer separating the control gate and floating gate. The control gates of the memory cells (100, 102, 104, 106) form the word lines. N+ doped layers 130, 132, 134, 136 and 138 are shared between neighboring cells, whereby the cells are connected to one another in series to form a NAND string. These N+ doped layers form the source and drain of each of the cells. For example, N+ doped layer 130 serves as the drain of transistor 122 and the source for transistor of 106, N+ doped layer 132 serves as the drain for transistor 106 and the source for transistor 104, N+ doped region 134 serves as the drain for transistor 104 and the source for transistor 102, N+ doped region 136 serves as the drain for transistor 102 and the source for transistor 100, and N+ doped layer 138 serves as the drain for transistor 100 and the source for transistor 120. N+ doped layer 126 connects to the bit line for the NAND string, while N+ doped layer 128 connects to a common source line for multiple NAND strings.

[0033] Note that although Figures 1-3 show four memory cells in the NAND string, the use of four transistors is only provided as an example. A NAND string can have less than four memory cells or more than four memory cells. For example, some NAND strings will include 8 memory cells, 16

memory cells, 32 memory cells, etc. The discussion herein is not limited to any particular number of memory cells in a NAND string.

[0034] Each memory cell can store data represented in analog or digital form. When storing one bit of digital data, the range of possible threshold 5 voltages of the memory cell is divided into two ranges which are assigned logical data "1" and "0." In one example of a NAND type flash memory, the voltage threshold is negative after the memory cell is erased, and defined as logic "1." The threshold voltage after a program operation is positive and defined as logic "0." When the threshold voltage is negative and a read is 10 attempted, the memory cell will turn on to indicate logic one is being stored. When the threshold voltage is positive and a read operation is attempted, the memory cell will not turn on, which indicates that logic zero is stored. A memory cell can also store multiple states, for example, thereby storing multiple bits of digital data. In the case of storing multiple states of data, the 15 range of possible threshold voltages is divided into the number of states. For example, if four states are used, there will be four threshold voltage ranges assigned to the data values "11", "10", "01", and "00." In one example of a NAND type memory, the threshold voltage after an erase operation is negative and defined as "11". Positive threshold voltages are used for the states of 20 "10", "01", and "00."

[0035] Relevant examples of NAND type flash memories and their operation are provided in the following U.S. Patents/Patent Applications, all of which are incorporated herein by reference in their entirety: U.S. Pat. No. 5,570,315; U.S. Pat. No. 5,774,397; U.S. Pat. No. 6,046,935; U.S. Pat. No. 25 5,386,422; U.S. Pat. No. 6,456,528 and U.S. Pat. Application. Ser. No. 09/893,277 (Publication No. US2003/0002348). Other types of non-volatile memory can also be used with the present invention.

[0036] Fig. 4 is a block diagram of one embodiment of a flash memory system that can be used to implement the present invention. Memory cell array 302 is controlled by column control circuit 304, row control circuit 306, c-source control circuit 310 and p-well control circuit 308. Column control circuit 304 is connected to the bit lines of memory cell array 302 for reading data stored in the memory cells, for determining a state of the memory cells during a program operation, and for controlling potential levels of the bit lines to promote the programming or to inhibit the programming. Row control circuit 306 is connected to the word lines to select one of the word lines, to apply read voltages, to apply program voltages and to apply an erase voltage. C-source control circuit 310 controls a common source line (labeled as "C-source" in Fig. 5) connected to the memory cells. P-well control circuit 308 controls the p-well voltage.

[0037] The data stored in the memory cells are read out by the column control circuit 304 and are output to external I/O lines via data input/output buffer 312. Program data to be stored in the memory cells are input to the data input/output buffer 312 via the external I/O lines, and transferred to the column control circuit 304. The external I/O lines are connected to controller 318.

[0038] Command data for controlling the flash memory device is input to controller 318. The command data informs the flash memory of what operation is requested. The input command is transferred to state machine 316, which controls column control circuit 304, row control circuit 306, c-source control 310, p-well control circuit 308 and data input/output buffer 312. State machine 316 can also output status data of the flash memory such as READY/BUSY or PASS/FAIL.

[0039] Controller 318 is connected or connectable with a host system such as a personal computer, a digital camera, personal digital assistant, etc. Controller 318 communicates with the host in order to receive commands from

the host, receive data from the host, provide data to the host and provide status information to the host. Controller 318 converts commands from the host into command signals that can be interpreted and executed by command circuits 314, which is in communication with state machine 316. Controller 318 typically contains buffer memory for the user data being written to or read from the memory array.

[0040] One exemplar memory system comprises one integrated circuit that includes controller 318, and one or more integrated circuit chips that each contain a memory array and associated control, input/output and state machine circuits. The trend is to integrate the memory arrays and controller circuits of a system together on one or more integrated circuit chips. The memory system may be embedded as part of the host system, or may be included in a memory card (or other package) that is removably inserted into the host systems. Such a removable card may include the entire memory system (e.g. including the controller) or just the memory array(s) and associated peripheral circuits (with the Controller or control function being embedded in the host). Thus, the controller can be embedded in the host or included within a removable memory system.

[0041] In some implementations, some of the components of Figure 4 can be combined. In various designs, one or more of the components of Fig 4, other than memory cell array 302, can be thought of as a control circuit.

[0042] With reference to Fig. 5, an example structure of memory cell array 302 is described. As one example, a NAND flash EEPROM is described that is partitioned into 1,024 blocks. The data stored in each block is simultaneously erased. In one embodiment, the block is the minimum unit of cells that are simultaneously erased. In each block, in this example, there are 8,512 columns that are divided into even columns and odd columns. The bit lines are also divided into even bit lines (BLe) and odd bit lines (BLo). Figure

5 shows four memory cells connected in series to form a NAND string. Although four cells are shown to be included in each NAND string, more or less than four memory cells can be used. One terminal of the NAND string is connected to corresponding bit line via a first select transistor SGD, and 5 another terminal is connected to c-source via a second select transistor SGS.

[0043] During one embodiment of read and programming operations, 4,256 memory cells are simultaneously selected. The memory cells selected have the same word line and the same kind of bit line (e.g. even bit lines or odd bit lines). Therefore, 532 bytes of data can be read or programmed 10 simultaneously. These 532 bytes of data that are simultaneously read or programmed form a logical page. Therefore, one block can store at least eight logical pages (four word lines, each with odd and even pages). When each memory cell stores two bits of data (e.g. a multi-level cell), one block stores 16 logical pages. Other sized blocks and pages can also be used with the present 15 invention. Additionally, architectures other than that of Figs. 4 and 5 can also be used to implement the present invention.

[0044] Memory cells are erased in one embodiment by raising the p-well to an erase voltage (e.g. 20 volts) and grounding the word lines of a selected block while the source and bit lines are floating. Due to capacitive coupling, 20 the unselected word lines, bit lines, select lines, and c-source are also raised to 20V. A strong electric field is thus applied to the tunnel oxide layers of selected memory cells and the data of the selected memory cells are erased as electrons of the floating gates are emitted to the substrate side. As electrons are transferred from the floating gate to the p-well region, the threshold voltage 25 of a selected cell becomes negative. Erasing can be performed on the entire memory array, separate blocks, or another unit of cells.

[0045] In the read and verify operations, the select gates (SGD and SGS) of a selected block are raised to one or more select voltages and the unselected

word lines (e.g., WL0, WL1 and WL3) of the selected block are raised to a read pass voltage (e.g. 4.5 volts) to make the transistors operate as pass gates. The selected word line of the selected block (e.g. WL2) is connected to a reference voltage, a level of which is specified for each read and verify
5 operation in order to determine whether a threshold voltage of the concerned memory cell has reached such level. For example, in a read operation for a two level memory cell, the selected word line WL2 may be grounded, so that it is detected whether the threshold voltage is higher than 0v. In a verify operation for a two level memory cell, the selected word line WL2 is
10 connected to 2.4v, for example, so that as the programming progresses it is verified whether the threshold voltage has reached at least 2.4v. The source and p-well are at zero volts during verify. The selected bit lines (BLe) are pre-charged to a level of, for example, 0.7v. If the threshold voltage is higher than the read or verify level on the word line, the potential level of the concerned bit
15 line (BLe) maintains the high level because of the non-conductive memory cell. On the other hand, if the threshold voltage is lower than the read or verify level, the potential level of the concerned bit line (BLe) decreases to a low level, for example less than 0.5V, because of the conductive memory cell. The state of the memory cell is detected by a sense amplifier that is connected to
20 the bit line.

[0046] The erase, read and verify operations described above are performed according to techniques known in the art. Thus, many of the details explained can be varied by one skilled in the art. Other read and verify techniques known in the art can also be used.

25 [0047] Figure 6 illustrates threshold voltage distributions for memory cells storing two bits of data (e.g., four data states). In one embodiment, distribution 460 represents a distribution of threshold voltages of cells that are in the erased state (e.g., storing "11"), having negative threshold voltage levels. Distribution 462 represents a distribution of threshold voltages of cells that are

storing "10." Distribution 464 represents a distribution of threshold voltages of memory cells storing "00." Distribution 466 represents a distribution of threshold voltages of cells that are storing "01." In other embodiments, each of the distributions can correspond to different data states than described above. In some implementations, these data values (e.g. logical states) are assigned to the threshold ranges using a gray code assignment so that if the threshold voltage of a floating gate erroneously shifts to its neighboring physical state, only one bit will be affected. The specific relationship between the data programmed into the memory cell and the threshold voltage ranges of the cell depends upon the data encoding scheme adopted for the memory cells. For example, U.S. Patent No. 6,222,762 and U.S. Patent Application No. 10/461,244, "Tracking Cells For A Memory System," filed on June 13, 2003, both of which are incorporated herein by reference in their entirety, describe various data encoding schemes for multi-state flash memory cells. Additionally, the present invention can work with memory cells that store more than two bits of data.

[0048] In one implementation, a memory cell in the erased state (e.g., distribution 460) can be programmed to any of the program states (distributions 462, 464 or 466). In another embodiment, memory cells in the erased state are programmed according to a two-step methodology. In this two-step methodology, each of the bits stored in a data state correspond to different logical pages. That is, each bit stored in a memory cell has a different logical page address, pertaining to a lower logical page and an upper logical page. For example, in state "10," the "0" is stored for the lower logical page and the "1" is stored for the upper logical page. In a first programming step, the cell's threshold voltage level is set according to the bit to be programmed into the lower logical page. If that bit is a logic "1," the threshold voltage is not changed since it is in the appropriate state (e.g. distribution 460) as a result of having been earlier erased. However, if the bit is to be programmed to a

logic "0," the threshold level of the cell is increased to be within the threshold voltage distribution 462.

[0049] In the second programming step, the memory cell's threshold voltage level is set according to the bit being programmed into the upper logical page. If the upper logical page bit is to be logic "1," then no further programming occurs since the cell is in one of the states corresponding to the threshold voltage distribution 460 or 462, both of which carry an upper page bit of "1." If the upper logical page bit is to be logic "0" and the first step resulted in the cell remaining in the erased state corresponding to threshold 460, then the second step of the programming process includes raising the threshold voltage to be within threshold distribution 466. If the upper logical page bit is to be logic "0" and the cell had been programmed into the state corresponding to threshold distribution 462 as a result of the first programming step, then the second step of the programming process includes raising the threshold voltage to be within threshold voltage distribution 464. The two step process is just one example of a methodology for programming multi-state memory. Many other methodologies, including a one step process or more than two steps can be used. Although Fig. 6 shows four states (two bits), the present invention can also be used with other multi-state structures including those that include eight states, sixteen states, thirty-two states, and others.

[0050] In one embodiment, memory cells that use an erased state and only one programmed state would only use two threshold voltage distributions. For example, threshold voltage distribution 460 could be used to represent the erased state and threshold voltage distribution 462 could be used to represent the programmed state. Other state assignments can also be used with the present invention.

[0051] When programming an EEPROM or flash memory device, such as a NAND flash memory device, typically a program voltage is applied to the

control gate and the bit line is grounded. Electrons from the channel are injected into the floating gate. When electrons accumulate in the floating gate, the floating gate becomes negatively charged and the threshold voltage of the memory cell is raised to one of the threshold voltage distribution described above. Typically, the program voltage applied to the control gate is applied as a series of pulses. In one embodiment, the magnitude of the pulses is increased with each successive pulse by a predetermined step size (e.g. 0.4v, 0.2v, or others). Figure 7 shows a program voltage signal Vpgm applied to the control gates of flash memory cells.

[0052] In the periods between the program pulses, verify operations are carried out. That is, the programming level of each cell of a group of cells being programmed in parallel is read between successive programming pulses to determine whether it is equal to or greater than a verify level to which it is being programmed. For arrays of multi-state flash memory cells, the memory cells will perform a verification step for each state to determine which state the memory cell is within. For example, a multi-state memory cell capable of storing data in four states may need to perform verify operations for three compare points. Figure 8 shows three programming pulses 10a, 10b and 10c (each of which are also depicted in Fig. 7). Between the programming pulses are three verify pulses in order to perform three verify operations. Based on the three verify operations, the system can determine whether or not the threshold compare point of the data state associated with each memory cell of the population of memory cells being concurrently programmed has been reached.. Note that one of the verify pulses is at 0 volts.

[0053] The present invention relates to technology for reducing the overall time needed to program a given amount of data into a non-volatile memory. One embodiment provides for faster overall programming by concurrently programming multiple pages within a single erasable block, thereby increasing write performance without increasing erase block size. For example, multiple

memory cells on the same NAND string can be programmed concurrently. In one implementation, various data conditional channel programming potentials are established in spaced apart memory cells within the NAND string, using intervening memory cells (referred to as boundary memory cells) as isolation there-between. Setting up the various internal potentials can be accomplished by a sequential data load process followed by an isolation (voltage trapping) operation. The data load process, to be explained in more detail below, essentially includes causing channels of addressed memory cells selected for programming to be at voltage potentials appropriate for programming and causing channels of memory cells not selected for programming to be at a voltage potential appropriate for inhibiting programming. Once this "data loading" sequence is completed, all the selected control gates (e.g., the control gates for the addressed memory cells to be data conditionally programmed) are ramped to their programming voltage (e.g., a pulse that can have a magnitude up to ~20v, in one embodiment) to effect the data conditional programming.

[0054] The magnitude of this programming and resulting threshold voltage shift depends both on the channel potential and on the relative storage strength (i.e. relative capacitance) of the underlying channel and source/drain reservoir. The underlying channel and source/drain reservoir is used to hold the boosting voltage for memory cells to be inhibited. The underlying channel and source/drain reservoir is also used to absorb the voltage applied to word lines so that the channel remains at or near 0 volts (or other target potential) for memory cells being programmed. If the underlying channel and source/drain reservoir is too small (e.g. limited cathode programming charge scenario), then very little programming will occur, requiring many repetitions to get useful programming. Use of substantially higher voltages may help a bit, but is unattractive, putting more burden on the process and circuitry to support such higher voltage, increasing vulnerability to programming disturbs as well as degrading overall reliability.

[0055] If one cell's channel and neighboring source/drains relative capacitance are not adequate to support this reservoir cathode function, then more storage elements will need to be strung together to hold the appropriate voltage levels. For example, this can be accomplished by setting up every fourth memory cell (i.e. row) for concurrent programming, with one of the intervening memory cells serving as isolation and the other two intervening memory cells providing additional cathode reservoir capacitance for data conditional programming. In the case of a sixteen element NAND string this would include four pages being concurrently programmed, allowing programming of the set of sixteen pages (in a binary flash device) in four data write operations. If the NAND string length is doubled to thirty two, then eight pages can be concurrently programmed, approximately further doubling the effective programming speed. However, the erase block size also doubles, increasing the garbage collection area in the same proportion. Furthermore, the actual write speed increase depends on the proportion of time required for verify, which does not change since separate verify operations must be performed on each of the concurrently programmed pages.

[0056] To increase the relative capacitance in order to better support this reservoir cathode function, every eight or every sixteenth memory cell can be concurrently programmed, with one of the intervening memory cells serving as isolation and the other intervening memory cells providing additional cathode reservoir capacitance for data conditional programming. Note that the number of memory cells programmed concurrently depends on the number of intervening memory cells providing additional cathode reservoir capacitance. The number of intervening memory cells required in order to provide the necessary additional cathode reservoir capacitance depends on the device physics of the transistors and neighboring source/drain junctions.. What is important is that there are enough intervening memory cells to provide the necessary additional cathode reservoir capacitance.

[0057] Figure 9 depicts a NAND string with thirty two memory cells: 502-564. The NAND string also includes a source side select gate 500 and a drain side select gate 566. Connected to the control gate for the source side select gate 500 is a select signal SGS. Connected to the control gate for the drain side select gate 566 is a select signal SGD. Each memory cell of Fig. 9 is connected to a word line. Memory cell 502 is connected to word line WL0. Memory cell 504 is connected to word line WL1. Memory cell 506 is connected to word line WL2. Memory cell 508 is connected to word line WL3. Memory cell 510 is connected to word line WL4. Memory cell 512 is connected to word line WL5. Memory cell 514 is connected to word line WL6. Memory cell 516 is connected to word line WL7. Memory cell 518 is connected to word line WL8. Memory cell 520 is connected to word line WL9. Memory cell 522 is connected to word line WL10. Memory cell 524 is connected to word line WL11. Memory cell 526 is connected to word line WL12. Memory cell 528 is connected to word line WL13. Memory cell 530 is connected to word line WL14. Memory cell 532 is connected to word line WL15. Memory cell 534 is connected to word line WL16. Memory cell 536 is connected to word line WL17. Memory cell 538 is connected to word line WL18. Memory cell 540 is connected to word line WL19. Memory cell 542 is connected to word line WL20. Memory cell 544 is connected to word line WL21. Memory cell 546 is connected to word line WL22. Memory cell 548 is connected to word line WL23. Memory cell 550 is connected to word line WL24. Memory cell 552 is connected to word line WL25. Memory cell 554 is connected to word line WL26. Memory cell 556 is connected to word line WL27. Memory cell 558 is connected to word line WL28. Memory cell 560 is connected to word line WL29. Memory cell 562 is connected to word line WL30. Memory cell 564 is connected to word line WL31. The NAND string of Figure 9 will be used to explain the present invention. However, it is noted that the present invention may be used with other types of non-volatile storage.

[0058] Assume, for purposes of example, an embodiment where two memory cells on the NAND string of Fig. 9 are programmed concurrently. In one implementation, the NAND string is divided into two regions. For example, assume a bottom region includes memory cells 502-532 and a top region includes memory cells 534-564. During a programming process, one memory cell from the top region will be programmed concurrently with one memory cell from the bottom region. There are many suitable methods for choosing which memory cell from the bottom region is paired with which memory cell from the top region. In one example, the memory cell from the top region is 16 memory cells (total cells on NAND string divided by number of groups) away from the memory cell of the bottom regions. This means that memory cell 518 would be concurrently programmed with memory cell 550, memory cell 520 would be concurrently programmed with memory cell 552, etc.

[0059] Figure 10 is a flow chart describing one embodiment of a process for programming using the above described technology. In step 602, the portion of the memory to be programmed is selected. In one implementation, this can be one or more write units appropriate to the memory structure. One example of a write unit is referred to as a page. In other embodiments, other units and/or structures can also be used. In step 604, a pre-programming process is sometimes used wherein the addressed memory cells are given non-data dependent programming to level out storage element wear and provide a more uniform starting point for the subsequent erase. In step 606, an erase process is performed, as appropriate for the type of storage element being used. One example of a suitable smart erase process is described in U.S. Patent No. 5,095,344, incorporated herein by reference in its entirety. Step 608 includes an optional soft programming process designed to put the threshold voltages of erased memory cells into a more uniform starting range for the actual write phase. In one embodiment, if any of the memory cells fail to verify during

erase (or during soft programming), they can be mapped out of the logical address space. At this point the memory is ready for the data conditional programming phase.

[0060] In step 610, the program voltage (Vpgm) is set to an initial value. 5 For example, in some embodiments, the waveform of Fig. 7 is used and step 610 includes setting the initial pulse. Also, in step 610, a program counter (PC) is initialized to zero.

[0061] In step 618, the various programming conditions are established. In the example described above which includes two groups on a 32 cell NAND 10 chain, there are two conditions set up: a first programming condition for the bottom group and a second programming condition for the top group. If there are more than two groups (e.g., 4, 5, 6, etc.), then (in one embodiment) a programming condition is set up for each group. In one implementation, the programming conditions are set up sequentially, starting with the group closest 15 to the source line. Although the various programming conditions may be established at different times, they all will persist for at least a common portion of time prior to the programming pulse being applied. For example, after step 618 is completed, the programming conditions for all of the groups will be set.

[0062] In step 620, program pulses are concurrently applied. If two 20 memory cells are being programmed concurrently, then two program pulses are applied: one program pulse is applied to the first memory cell being programmed and a second program pulse is concurrently applied to the second memory cell being programmed. If four memory cells are being programmed concurrently, then four program pulses are applied.

25 [0063] In step 622, a verification process is performed to the memory cells being concurrently programmed. During the verification process of step 622, if a memory cell being programmed has attained its target threshold voltage condition, then it is inhibited from further programming during the

remained of the data programming session.. Two memory cells that are being programmed concurrently may reach their target threshold voltage conditions at different times causing the memory cells to being inhibited at different times. As such, there may be a time when one of the memory cells is being programmed while the other memory cell is inhibited. Although such a situation can occur, the programming processes for the memory cells still overlap in time.

[0064] In step 624, it is determined whether each of the memory cells have verified that their threshold voltages are at the target threshold voltage condition for that memory cell. If so, the programming process is completed successfully (status = pass) in step 626 breaking out of the programming loop of Figure 10. If all of the memory cells are not all verified, then it is determined whether the program counter (PC) is less than 20. If the program counter (PC) is not less than 20 (step 628), then the program process indicates a status of "fail," again braking out of the program loop of Figure 10. (step 630). If the program counter (PC) is less than 20, then the program counter (PC) is incremented by 1 and the program voltage is stepped up to the next pulse in step 632. Subsequent to step 632, the process loops back to step 620 and the set of programming conditions is applied to the memory cells.

[0065] Figure 11 is a flow chart describing more details of one embodiment of the process for establishing program conditions (step 618 of Fig. 10). The steps of Fig. 11 implement an example that divides a NAND string into two groups, as discussed above. For ease of description, assume that memory cells 518 and 550 of Fig. 9 are to be programmed concurrently. Note that although much of the examples pertain to the NAND string of Fig. 9, in many implementations there will be many NAND strings being programmed simultaneously (e.g., as described above all even or all odd bit lines for a block may be programmed simultaneously).

[0066] In step 700, the drain side select gate 556 is turned on. For example, Vdd can be applied to the signal SGD. In step 702, data for memory cells in the bottom group is applied to the bit lines. For example, bit lines connected to NAND strings of bottom group memory cells that are not to
5 receive programming may receive an inhibit voltage, such as Vdd (e.g. ~2.5volts), and bit lines connected to NAND strings that are to receive programming may receive a voltage that enables programming (e.g., ~0 volts). In other embodiments, the bit line may receive an intermediate voltage (e.g., 1.5v or other values) that allows partial or reduced speed programming. For
10 example, the intermediate voltage may be used to retard programming as part of a coarse/fine programming methodology. More information about coarse/fine programming methodologies can be found in the following patent documents that are incorporated herein by reference in their entirety: U.S. Patent Application No. 10/766,217, "Efficient Verification for Coarse/Fine
15 Programming of Non-Volatile Memory" filed January 27, 2004; U.S. Patent Application No. 10/051,372, "Non-Volatile Semiconductor Memory Device Adapted to Store A Multi-Valued Data in a Single Memory Cell," filed January 22, 2002; U.S. Patent 6,301,161; U.S. Patent 5,712,815; U.S. Patent No. 5,220,531; and U.S. Patent No. 5,761,222.

20 **[0067]** The data applied to the bit lines in step 702 is for the memory cell in the bottom group. For example, the data is for memory cell 518.

[0068] In step 704, one or more boosting voltages are applied to the word lines connected to the NAND string. Those NAND strings receiving 0 volts at the bit line will dissipate the boosting voltage so that the channel region(s) for
25 those NAND strings (including the bottom group) will be at or near 0 volts.. Those NAND strings receiving 2.5 volts at the bit line will have channel regions boosted, for example, to approximately 7.5 volts in the case of a 9 volt boosting voltage applied to the word lines.

[0069] Consider that a transistor in a NAND string will turn on if V_G-V_S > V_{TH}, where V_G is the voltage applied to the gate, V_S is the voltage at the source and V_{TH} is threshold voltage of the transistor. The NAND transistor is symmetrical in that either side could be the source or the drain. The side with the lower voltage is typically referred to as the source. Thus, as voltages change, which side is the source and which side is the drain may also change. If V_G is less than V_{TH} than the transistor is cut off (no conduction between source and drain). If both V_S and V_D are increased relative to a given V_G, so that V_G-V_S < V_{TH} (remember that V_D > V_S), then the device is also cut-off.

[0070] To inhibit a NAND string from programming, the to be inhibited bit lines (termed unselected bit lines) are raised to Vdd (e.g., ~2.5 volts). In one embodiment, the control gate of the drain side select gate is also at Vdd, causing the select gate to conduct. The NAND strings on the unselected bit lines are then boosted by the boosting voltages applied to the word lines, which raises the voltage in the NAND string. When the voltage in the NAND string reaches V_G-V_{TH} (of the select gate), then the select gate will cut-off, which isolates the NAND string from the bit line so that the voltage on the NAND string will not be dissipated into the bit line. The voltage on the NAND string will then continue to increase so that it is higher then V_G, but because the bit line potential is greater than V_G-V_{TH}, the select gate will remain cut-off and the voltage in the NAND string will continue to increase in concert with the increasing boosting voltage, for example, to about 7.5 volts. When the voltage in the channel is at this boosted potential (e.g., 7.5 volts), the differential across the tunnel dielectric is not sufficient to allow for tunneling of electrons into the floating gate during the time of programming which could cause data state failure.

[0071] In step 706, the boundary cells between the bottom group and the top group will be cut off for each for the NAND strings. In one embodiment, the boundary cells are the memory cells that are at the borders between the

groups. In the above example, the boundary cell between the bottom group and the top group could be memory cell 532 or memory cell 534. In another embodiment, the boundary cell can be the memory cell that is midway between the two memory cells being programmed. In other embodiments, the boundary cell could be some other memory cell between the two memory cells being programmed. In one implementation, the boundary cell is cut-off by apply a voltage to its control gate that is less than the threshold voltage for that boundary cell. In one example, it may be possible for an erased memory cell to have a negative threshold voltage; therefore, negative voltage (e.g., –4 volts) is applied to the control gate of the boundary cells in order to insure cut-off the boundary cells.

[0072] Note that in some embodiments, the boundary cells are memory cells that also may need to be programmed. When it is time to program a memory cell that is a boundary cell, a different memory cell will become the boundary cell. For example, an adjacent memory cell or memory cell two or more away in the NAND string, and separating the memory cells being programmed, can become the new boundary cell.

[0073] In step 708, the word lines for the top group are reset (e.g., to 0 volts). In step 710, data is applied to the bit lines for the top group. In step 712, one or more boosting voltages are applied to the word lines for the top group, while leaving the voltages already applied to those of the bottom group unchanged. Those NAND strings receiving 0 volts at the bit line in step 710 will dissipate the boosting voltage applied to the top group in step 712 so that the channel region(s) are at or near 0 volts. Those NAND strings receiving 2.5 volts at the bit line in step 710 will have channels (for the top group) boosted, for example, to approximately 7.5 volts for a 9 volt boosting voltage applied to the word lines. In step 714, the drain side select gate is optionally cut-off (e.g., by lowering its control gate voltage to 0 volts).

[0074] Figure 12 is a timing diagram that explains the behavior of seven signals (V_{BL}, V_{SGD}, V_{TUWL}, V_{BUWL}, V_{WL8}, V_{WL24} and V_{WL15}) operating according to he process of Figure 11. The signals of Fig. 12 implement both steps 618 and 620 of Fig. 10. Note that Fig. 12 plots voltage versus time for
5 various signals. To improve the explanation, numbers are assigned to the time units. In one example implementation, the time units may correspond to μsec. However, the invention is in no way limited to any particular timing and the time units are chosen for example purposes only.

[0075] The graph for the bit line voltage V_{BL} shows four possible bit line
10 voltage signals: A, B, C and D. Signal A, at approximately 0 volts until 14 time units, is applied to a bit line to allow a memory cell of the bottom group to be programmed. Signal B, at approximately 2.5 volts until 14 time units, is applied to a bit line to inhibit a memory cell of the bottom group. Signal C, at approximately 2.5 volts starting from 18 time units until optionally 33 time
15 units, is applied to a bit line to inhibit a memory cell of the top group. Signal D, at approximately 0 volts starting from 18 time units and optionally held there until 33 time units, is applied to a bit line to allow a memory cell of the top group to be programmed. These optional conditions are the optional waveform considerations of V_{SGD} described below. Either signal A or signal B
20 will be asserted as part of step 702 of Figure 11. Either signal C or signal D will be asserted as part of step 710. Thus, when programming, there are four possible forms of bit line voltage V_{BL}: (1) signal A followed by signal C for programming the memory cell in the bottom group while inhibiting the memory cell in the top group, (2) signal A followed by signal D for
25 programming the memory cell in the bottom group and programming the memory cell in the top group, (3) signal B followed by signal C for inhibiting the memory cell in the bottom group and inhibiting the memory cell in the top group, and (4) signal B followed by signal D for inhibiting the memory cell in the bottom group and programming the memory cell in the top group. Which

of the four waveforms is applied to the bit line depends on the data to be stored and the current threshold voltage of the memory cells.

[0076] The voltage V_{SGD} applied to the drain side select gate is raised to approximately 5 volts at time 0, as part of step 700. At time unit 5, V_{SGD} is lowered to 2.5 volts. At time 20 it is raised to 5 volts, and then subsequently lowered to 2.5 volts at time 25. At time 30, V_{SGD} is optionally lowered to 0 volts as per step 714 of Figure 11. Note that optionally bringing V_{SGD} to 0v at time unit 30 frees up the bit lines, allowing V_{BL} to change so that the bit lines can be used for other purposes. Alternatively, V_{SGD} can be left at the selected voltage level of 2.5 volts for the duration of that programming step (e.g., until time 58) and shut-off thereafter. In that case, the bit line voltage conditions V_{BL} of signals C and D should also be maintained for the duration (e.g., to approximately time 58). The period when V_{SGD} is at 5 volts is used to pre-charge the bit line before each boosting phase.

[0077] The signal V_{TUWL} is the voltage on the unselected word lines associated with the memory cells in the top group (Top Unselected Word Lines). The signal V_{BUWL} is the voltage on the unselected word lines connected to the memory cells in the bottom group (Bottom Unselected Word Lines). The signal V_{WL8} is the voltage on the selected word line WL8 connected to the memory cell 518 of Fig. 9 selected for programming in the bottom group. The signal V_{WL24} is the voltage on the selected word line WL24 connected to the memory cell 550 selected for programming in the top group. The signal V_{WL15} is the voltage on the word line WL15, connected to the boundary memory cell 532. The signals V_{TUWL}, V_{BUWL}, V_{WL8}, V_{WL24} and V_{WL15} are raised to approximately 1.5 volts during the initial period when V_{SGD} is at 5 volts in order to pre-charge the bit line/NAND string. At time 5, signals V_{TUWL}, V_{BUWL}, V_{WL8}, V_{WL24} and V_{WL15} are raised to approximately 9.5 volts (could also be 9 volts or other levels), as per step 704 of Fig. 11, to provide a boosting voltage. Those NAND strings receiving signal B on the bit line will be

boosted and those NAND strings receiving signal A on the bit line will remain at or near 0 volts. At time 10, the boundary cell will be cut off (as per step 706), by lowering V_{WL15} to approximately – 4 volts. The word lines for the top group are reset in step 708 by lowering signals V_{TUWL} and V_{WL24} to 0 volts at time 15. V_{BUWL} remains at 9.5 volts until time 55. V_{WL8} remains at 9.5 volts until the start of the programming, at time 35.

[0078] At this point a first programming condition has been established and is maintained for the bottom group of memory cells. In one embodiment, the channel of bottom group transistors will be at or near 0 volts for programming or at or near 7.5 volts for inhibiting programming.

[0079] Either signal C or signal D will be asserted on V_{BL} as part of step 710. The signals V_{TUWL} and V_{WL24} are raised to approximately 1.5v at time 20 to allow NAND string pre-charging, and to approximately 9.5 volts at time 25 to provide boosting for the top group in step 712, and remain at 9.5 volts until time 55. The control gate voltage V_{SGD} for the drain side select transistor is optionally lowered to 0 volts at time 30 in order to cutoff the select transistor, as per step 714. At this point a second programming condition has established and is maintained for the top group of memory cells. In one embodiment, the channel of top group transistors will be at or near 0 volts for programming or at or near 7.5 volts for inhibiting programming. Note that the first program condition and the second programming condition both persist at this time. The pre-charge condition associated with raising V_{SGD} to 5v at time intervals 0 to 5 and 20 to 25, described in this example waveform, is optional. In other embodiments, no such pre-charge operation is used and V_{SGD} is maintained at the 2.5v level during those time intervals.

[0080] At time 35, the program pulses are applied to V_{WL8} and V_{WL24}. In one embodiment, the magnitude of the pulses can vary between 12 and 20 volts. Thus, both V_{WL8} and V_{WL24} are raised at time 35 to the magnitude of the

desired program pulse voltage and both memory cells 518 and 550 are concurrently programmed as dictated by the just described bit line related data programming setup. The program pulse lasts until time 55, at which time V_{TUWL}, V_{BUWL}, V_{WL8}, V_{WL24} and V_{WL15} (and optionally V_{BL} and V_{SGD}) are all
5 brought to 0 volts.

[0081] Figure 13 is a flow chart describing more details of another embodiment of the process for establishing program conditions (step 618 of Fig. 10). The steps of Fig. 13 implement an example that divides a NAND string into four groups so that four memory cells on a common NAND string
10 are concurrently programmed. In one such implementation using the NAND string of Fig. 9, the first group corresponds to memory cells 502-516, the second group corresponds to memory cells 518-532, the third group corresponds to memory cells 534-548 and the fourth group corresponds to memory cells 550-564. One example set of the boundary cells includes
15 memory cells 516, 532 and 548. Other memory cells can also serve as boundary cells.

[0082] In step 750 of Figure 13, the drain side select gate 556 is turned on. For example, Vdd can be applied to the signal SGD. In step 702, data is applied to the bit lines. The data applied to the bit lines in step 752 is for the
20 memory cell in the first group that is being programmed. For example, the data is for memory cell 508. In step 754, one or more boosting voltages are applied to the word lines connected to the NAND string. In step 756, the boundary cells between the first group and the second group will be cut off. In step 758, the word lines for the groups not cut off (e.g., groups 2-4) are reset.
25 Group one word lines remain at the boosting voltage. In step 760, data for the second group is applied to the bit lines. In step 762, one or more boosting voltages are applied to the word lines for the groups not cut-off (groups 2-4). In step 764, the boundary cell between the second group and the third group is cut-off.

[0083] In step 766, the word lines for the groups not cut-ff (e.g., groups 3-4) are reset. Word lines for group one and two remain at the boosting voltage. In step 768, data for the third group is applied to the bit lines. In step 770, one or more boosting voltages are applied to the word lines for the groups not cut-
5 off (groups 3-4). In step 772, the boundary cell between the third group and the fourth group is cut-off.

[0084] In step 774, the word lines for the groups not cut-ff (e.g., group 4) are reset. Group one, two and three word lines remain at the boosting voltage. In step 776, data for the fourth group is applied to the bit lines. In step 778,
10 one or more boosting voltages are applied to the word lines for the group not cut-off (group 4). In step 780, the drain side select gate is optionally cut-off. The considerations for this optional select gate cutoff are analogous to those described earlier for the two group case. Note that, in one embodiment, the process of Fig. 13 is performed simultaneously for many NAND strings.

15 [0085] The process of Figure 13 describes the use of four groups so that four memory cells on a NAND string can be simultaneously programmed. The process of Figure 13 can be adapted to be used with more than four groups so that more than four memory cells on a NAND string can be simultaneously programmed. For example, steps 758-764 (with the iteration of step 760
20 applying the appropriate data and step 764 cutting-off the appropriate boundary cell) can be repeatedly performed for each additional group.

[0086] Figure 14 is a flow chart describing one embodiment of a process for verifying. In one example implementation, the process of Figure 14 is performed as part of step 622 of Figure 10 for a NAND string divided into two
25 groups. Note that although multiple memory cells are programmed simultaneously, in one embodiment the verification process is performed sequentially. In step 820, pass voltages are applied to the unselected word lines in regard to the bottom group. That is the word lines for all of the memory

cells, except the memory cell selected for programming in the bottom group, receive a pass voltage. The pass voltage (e.g. 4.5 volts) is designed to make sure that each of the unselected memory cells is sufficiently turned on. In step 822, one or more verify pulses (appropriate to the type of data being programmed) are applied to the word line associated with the memory cell selected for programming in the bottom group. Step 822 may also include pre-charging the bit line, as discussed above. The data is sensed for each of the verify pulses. One verify pulse is used for binary memory storage, and multiple verify pulses (e.g, total number of states –1, as per Fig. 8) are used for multi-state memory storage. In step 824, the system determines whether the memory cell has reached its target threshold voltage condition. If a memory cell has reached its target threshold voltage condition, then that memory cell is locked out from further programming in step 826 (e.g., by raising its bit line voltage to Vdd), thereby terminating programming to that memory cell for the duration of that programming session.

[0087] In step 828, pass voltages are applied to the unselected word lines in regard to the top group. That is, the word lines for all of the memory cells, except the memory cell selected for programming in the top group, receives a pass voltage. The pass voltage (e.g. 4.5 volts) is designed to make sure that each of the unselected memory cells are sufficiently turned on. In step 830, one or more verify pulses are applied to the word line associated with the memory cell selected for programming in the top group. Step 830 may also include pre-charging the bit line, as discussed above. The data is sensed for each of the verify pulses. In step 832, the system determines whether the memory cell has reached its target threshold voltage. If a memory cell has reached its target threshold voltage, then that memory cell is locked out from further programming (e.g., by raising its bit line voltage to Vdd) in step 834 (analogous to step 826). Note that the process of Fig. 14 is performed on multiple NAND strings simultaneously. Furthermore, the process of Fig. 14

can be adapted to be used for more than two groups by repeating step 820-826 for each of the additional groups.

[0088] The foregoing detailed description of the invention has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. The described embodiments were chosen in order to best explain the principles of the invention and its practical application, to thereby enable others skilled in the art to best utilize the invention in various embodiments and with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the claims appended hereto.

CLAIMS

I claim:

1. A method for programming non-volatile storage, comprising:

5 programming a first non-volatile storage element on a first NAND string; and

programming a second non-volatile storage element on said first NAND string, said programming of said first non-volatile storage element overlaps in time with said programming of said second non-volatile storage element.

10

2 A method according to claim 1, wherein:

said step of programming a first non-volatile storage element includes applying a first program pulse to a control gate for said first non-volatile storage element; and

15 said step of programming a second non-volatile storage element includes applying a second program pulse to a control gate for said second non-volatile storage element while applying said first program pulse to said control gate for said first non-volatile storage element.

20 3. A method according to claim 1, wherein:

said step of programming a first non-volatile storage element includes establishing a first program condition for said first non-volatile storage element; and

said step of programming a second non-volatile storage element includes

25 establishing a second program condition for said second non-volatile storage element, said second program condition is separate from said first program condition.

4. A method according to claim 3, wherein:

30 said first program condition includes a channel voltage at or near 0 volts;

and

said second program condition includes a channel voltage at or near an inhibit level.

5 5. A method according to claim 3, wherein:

said first program condition includes a channel voltage at or near 0 volts; and

said second program condition includes a channel voltage at or near a level that retards but does not inhibit programming.

10

6. A method according to claim 1, further comprising:

programming a third non-volatile storage element on a second NAND string while programming said first non-volatile storage element; and

programming a fourth non-volatile storage element on said second 15 NAND string while programming said second non-volatile storage element, said programming of said third non-volatile storage element overlaps in time with said programming of said fourth non-volatile storage element.

7. A method according to claim 1, further comprising:

20 programming a third non-volatile storage element on said first NAND string, said programming of said third non-volatile storage element overlaps in time with said programming of said first non-volatile storage element.

8. A method according to claim 7, further comprising:

25 programming a fourth non-volatile storage element on said first NAND string, said programming of said fourth non-volatile storage element overlaps in time with said programming of said first non-volatile storage element.

9. A method according to claim 1, wherein:

30 said first NAND string includes a first group of non-volatile storage

elements and a second group of non-volatile storage elements;

said first group includes said first non-volatile storage element;

said second group includes said second non-volatile storage element;

said first NAND string includes a boundary storage element which
5 serves at a boundary between said first group and said second group for at least a period of time; and

said first NAND string includes a select gate.

10. A method according to claim 9, wherein said steps of
10 programming a first non-volatile storage element and programming a second non-volatile storage element comprise:

applying a first value to a bit line associated with said first NAND string;

boosting word lines associated with said first NAND string to create a
15 first condition for said first group based on said first value;

cutting off said boundary storage element to maintain said first condition for said first group;

applying a second value to said bit line; and

boosting at least a subset of said word lines associated with said first
20 NAND string to create a second condition for said second group based on said second value, said first condition and said second condition overlap in time.

11 A method according to claim 10, wherein:

said step of programming a first non-volatile storage element includes
25 applying a first program pulse to a control gate for said first non-volatile storage element; and

said step of programming a second non-volatile storage element includes applying a second program pulse to a control gate for said second non-volatile storage element while applying said first program pulse to said control gate for
30 said first non-volatile storage element.

12. A method according to claim 1, wherein:

said first non-volatile storage element and said second non-volatile
5 storage element are multi-state NAND flash memory transistors.

13. A method according to claim 1, wherein:

said first NAND string is on a removable memory card.

10 14. A method for programming non-volatile storage, comprising:

programming a first non-volatile storage element; and

programming a second non-volatile storage element while programming said first non-volatile storage element, said first non-volatile storage element and said second non-volatile storage element are part of a group of non-volatile
15 storage elements having channels associated with a common selection line.

15. A method according to claim 14, wherein:

said programming of said first non-volatile storage element includes establishing a first program condition for said first non-volatile storage element;
20 and

said programming of said second non-volatile storage element includes establishing a second program condition for said second non-volatile storage element, said second program condition is separate from said first program condition.

25

16. A method according to claim 14, further comprising:

programming a third non-volatile storage element, said third non-volatile storage element is part of said group of non-volatile storage elements, said programming of said third non-volatile storage element overlaps in time
30 with said programming of said first non-volatile storage element; and

programming a fourth non-volatile storage element, said fourth non-volatile storage element is part of said group of non-volatile storage elements, said programming of said fourth non-volatile storage element overlaps in time with said programming of said first non-volatile storage element.

5

17. A method according to claim 14, wherein said steps of programming a first non-volatile storage element and programming a second non-volatile storage element comprise:

applying a first value to a common bit line;

10 boosting word lines associated with said group to create a first condition for said first non-volatile storage element based on said first value;

cutting off a boundary storage element associated with said group to maintain said first condition for said first non-volatile storage element;

applying a second value to said common bit line; and

15 boosting at least a subset of said word lines associated with said group to create a second condition for a second non-volatile storage element based on said second value, said first condition and said second condition overlap in time.

18 A method according to claim 17, wherein:

20 said first non-volatile storage element and said second non-volatile storage element are NAND flash memory devices; and

said common selection line is a drain side select gate control line.

19. A method according to claim 14, wherein:

25 said first non-volatile storage element and said second non-volatile storage element are multi-state NAND flash memory devices;

said programming of said first non-volatile storage element includes raising a threshold voltage of said first non-volatile storage element to a first programmed state; and

30 said programming of said second non-volatile storage element includes

raising a threshold voltage of said second non-volatile storage element to a second programmed state.

20. A non-volatile storage system, comprising:

5 a group of non-volatile storage elements associated with a common source/drain control line and a common selection line, said group includes a first non-volatile storage element and a second non-volatile storage element; and

a control circuit, said control circuit causes programming of said first 10 non-volatile storage element, said control circuit causes programming of said second non-volatile storage element during said programming of said first non-volatile storage element.

21. A non-volatile storage system according to claim 20, wherein:

15 said control circuit establishes a first program condition for said first non-volatile storage element and establishes a second program condition for said second non-volatile storage element, said second program condition is separate from said first program condition.

20 22. A non-volatile storage system according to claim 20, wherein:

said group includes a third non-volatile storage element and a fourth non-volatile storage element;

said control circuit causes programming of said third non-volatile storage element during said programming of said first non-volatile storage 25 element; and

said control circuit causes programming of said fourth non-volatile storage element during said programming of said first non-volatile storage element.

30 23. A non-volatile storage system according to claim 20, wherein

said control circuit performs a method comprising:

applying a first value to said common source/drain control line;

boosting word lines associated with said group to create a first condition for said first non-volatile storage element based on said first value;

5 cutting off a boundary storage element associated with said group to maintain said first condition for said first non-volatile storage element;

applying a second value to said common source/drain control line; and

boosting at least a subset of said word lines associated with said group to create a second condition for a second non-volatile storage element based on

10 said second value, said first condition and said second condition overlap in time.

24. A non-volatile storage system according to claim 23, wherein

said control circuit programs said first non-volatile storage element by applying a first program pulse to a control gate for said first non-volatile storage

15 element; and

said control circuit programs said second non-volatile storage element by applying a second program pulse to a control gate for said second non-volatile storage element while applying said first program pulse to said control gate for said first non-volatile storage element.

20

25. A non-volatile storage system according to claim 24, wherein:

said group of non-volatile storage elements include NAND flash memory devices on a NAND string;

said common source/drain control line is a bit line for said NAND

25 string; and

said common selection line is a drain side select gate control line for said NAND string.

26. A non-volatile storage system according to claim 20, wherein:

30 said control circuit includes a controller, a state machine, decoders, and

sense amplifiers.

27. A method for programming non-volatile storage, comprising:
establishing a first program condition for a first non-volatile storage
5 element;
establishing a second program condition for a second non-volatile storage element while said first program condition persists, said first program condition is different than said second program condition so that said first non-volatile storage element programs to a different level than said second non-
10 volatile storage element, said first non-volatile storage element and said second non-volatile storage element are part of a group of non-volatile storage elements associated with a common source/drain control line and a common selection line; and
programming said first non-volatile storage element using said first
15 program condition and said second non-volatile storage element using said second program condition.

28. A method according to claim 27, wherein:
said programming includes applying a program pulse to said first non-
20 volatile storage element and applying a program pulse to said second non-volatile storage element.

29. A method according to claim 27, wherein:
said establishing a first program condition includes applying a first value
25 to said common source/drain control line, boosting word lines associated with said group, and cutting off a boundary non-volatile storage element associated with said group; and
said establishing a second program condition includes applying a second value to said common source/drain control line, and boosting at least a subset of
30 said word lines associated with said common source/drain control line.

30. A method according to claim 29, wherein:

said first non-volatile storage element and said second non-volatile storage element are NAND flash memory devices on a NAND string;

5 said common source/drain control line is a bit line connected to said NAND string;

said common selection line is a drain side select gate control line for said NAND string; and

said programming of said first non-volatile storage element overlaps in 10 time with said programming of said second non-volatile storage element.

31. A non-volatile storage system, comprising:

a group of non-volatile storage elements associated with a common source/drain control line and a common select line, said group includes a first 15 non-volatile storage element and a second non-volatile storage element; and

a control circuit, said control circuit establishes a first program condition for said first non-volatile storage element and establishes a separate second program condition for said second non-volatile storage element while said first program condition persists, said control circuit causes programming of said first 20 non-volatile storage element based on said first program condition and programming of said second non-volatile storage element based on said second program condition.

32. An apparatus according to claim 31, wherein:

25 said programming of said first non-volatile storage element includes applying a program pulse to said first non-volatile storage element and said programming of said second non-volatile storage element includes applying a program pulse to said second non-volatile storage element.

30 33. An apparatus according to claim 31, wherein:

said establishing of a first program condition includes causing a first value to be applied to said common source/drain control line, causing word lines associated with said group to receive one or more boosting voltages, and causing a boundary non-volatile storage element associated with said group to
5 cut-off; and

said establishing of a second program condition includes causing a second value to be applied to said common source/drain control line, and causing at least a subset of said word lines associated with said group to receive boosting voltage.

10

34. An apparatus according to claim 31, wherein:

said first non-volatile storage element and said second non-volatile storage element are NAND flash memory devices on a NAND string;

said common source/drain control line is a bit line connected to said
15 NAND string;

said common selection line is a drain side select gate control line for said NAND string; and

said programming of said first non-volatile storage element overlaps in time with said programming of said second non-volatile storage element.

20

35. An apparatus according to claim 34, wherein:

said programming of said first non-volatile storage element includes applying a program pulse to said first non-volatile storage element and said programming of said second non-volatile storage element includes applying a
25 program pulse to said second non-volatile storage element;

said establishing of a first program condition includes causing a first value to be applied to said common source/drain control line, causing word lines associated with said group to receive one or more boosting voltages, and causing a boundary non-volatile storage element associated with said group to
30 cut-off; and

said establishing of a second program condition includes causing a second value to be applied to said common source/drain control line, and causing at least a subset of said word lines associated with said group to receive boosting voltage.

5

36. A method for programming non-volatile storage, comprising:

establishing a first program condition for a first non-volatile storage element on a first NAND string;

establishing a second program condition for a second non-volatile storage element on said first NAND string while said first program condition persists, said first program condition is separate from said second program condition; and

10

programming said first non-volatile storage element using said first program condition and said second non-volatile storage element using said second program condition.

15

37. A method for programming non-volatile storage, comprising:

applying a first value to a bit line;

boosting word lines associated with a group of non-volatile storage elements to create a first condition based on said first value, said group of non-volatile storage elements are associated with said bit line;

20

cutting off a first non-volatile storage element of said group to maintain said first condition for a second non-volatile storage element of said group;

applying a second value to said bit line;

boosting at least a subset of said word lines to create a second condition for a third non-volatile storage element of said group based on said second value, said first condition and said second condition overlap in time; and

25

programming said second non-volatile storage element and said third non-volatile storage element.

30

38. A method according to claim 37, wherein:

said programming of said second non-volatile storage element and said third non-volatile storage element includes applying a program pulse to said second non-volatile storage element and applying a program pulse to said third
5 non-volatile storage element.

39. A method according to claim 37, wherein:

said group of non-volatile storage elements is a NAND string.

10 40. A method according to claim 37, wherein:

said first non-volatile storage element, said second non-volatile storage element and said third non-volatile storage element are multi-state NAND flash memory devices on a NAND string.

Fig. 1

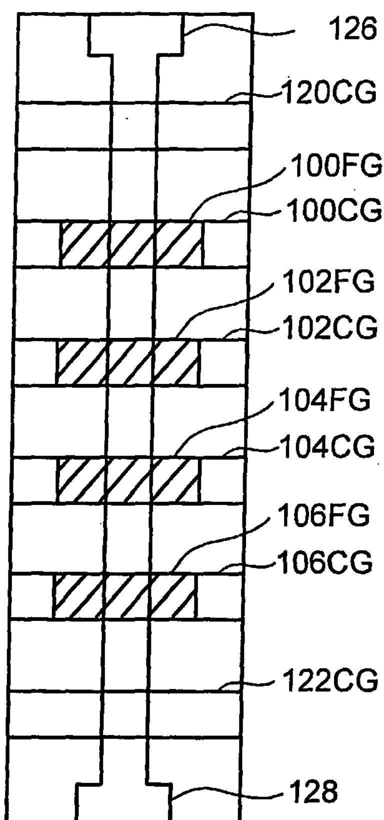


Fig. 2

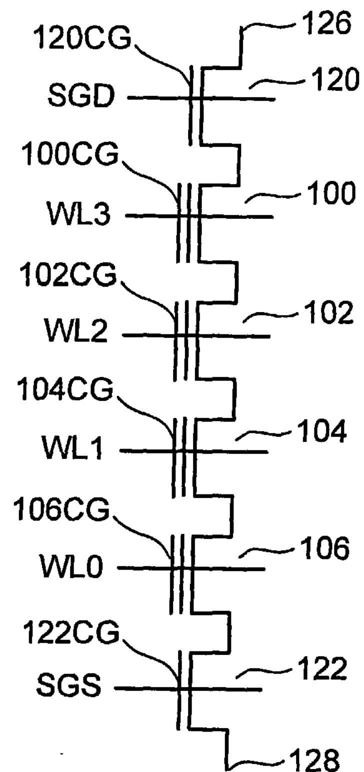
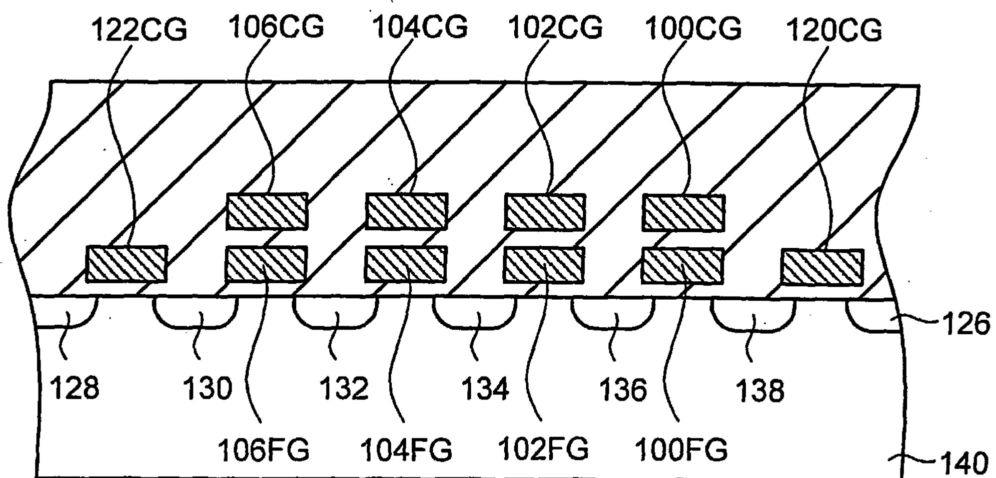


Fig. 3



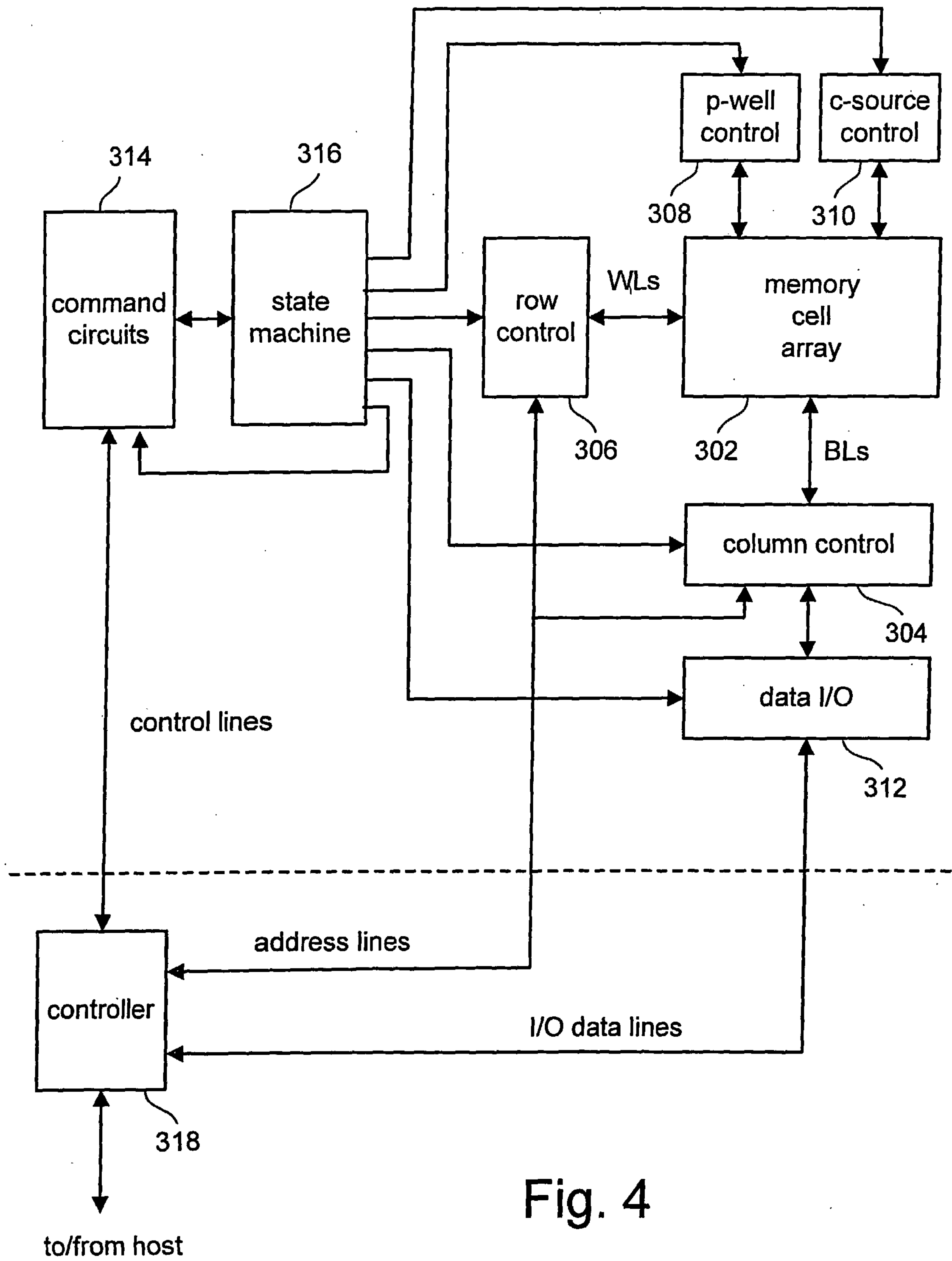
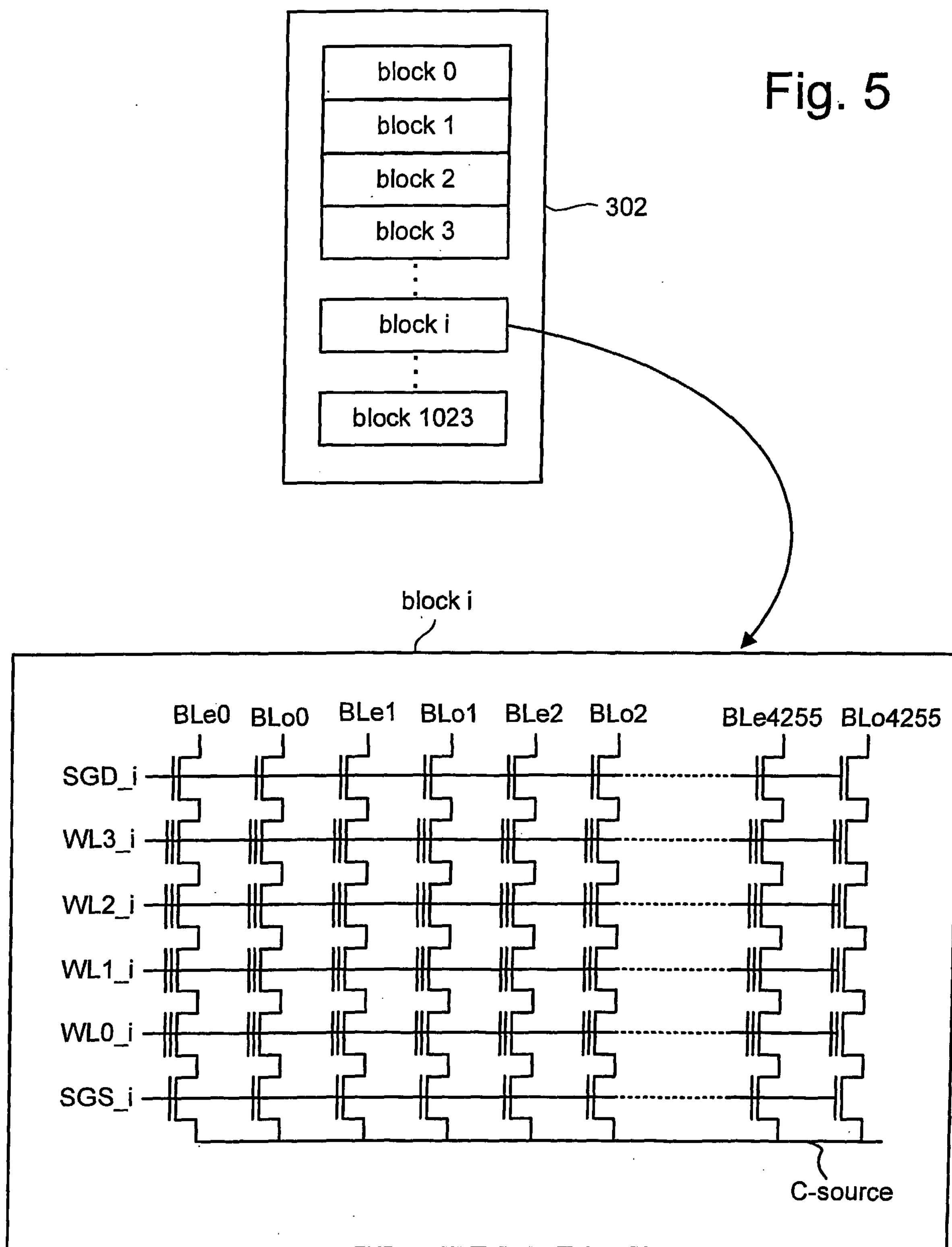
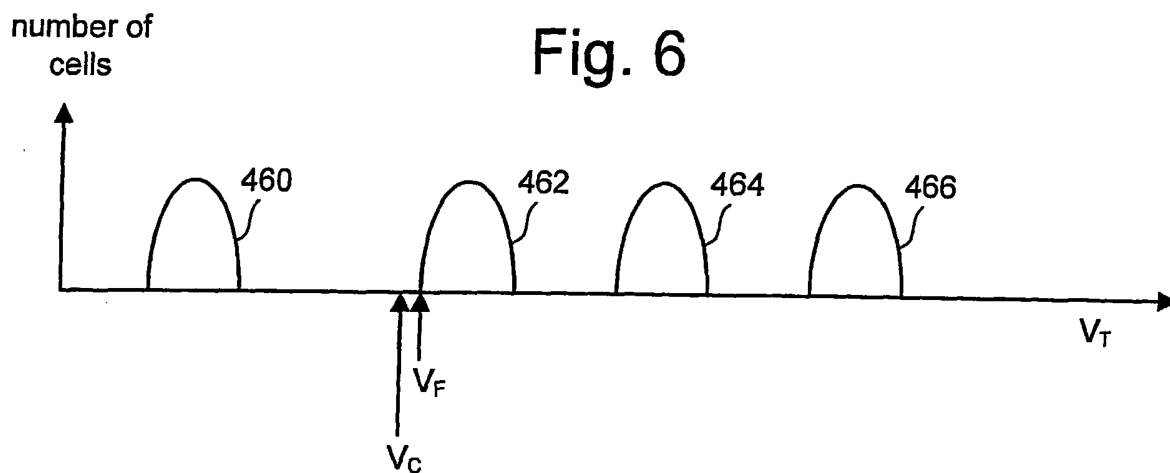


Fig. 4


block 0
block 1
block 2
block 3
block i
block 1023
302
Fig. 5
block i
BLe0
BLo0
BLe1
BLo1
BLe2
BLo2
BLe4255
BLo4255
SGD_i
WL3_i
WL2_i
WL1_i
WL0_i
SGS_i
C-source


number of cells
Fig. 6
460
462
464
466
V_T
V_F
V_C

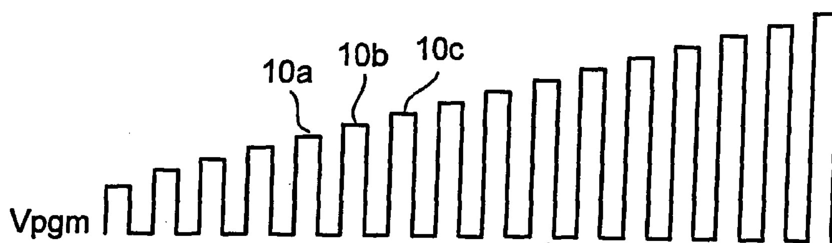


Fig. 7
10a
10b
10c
Vpgm

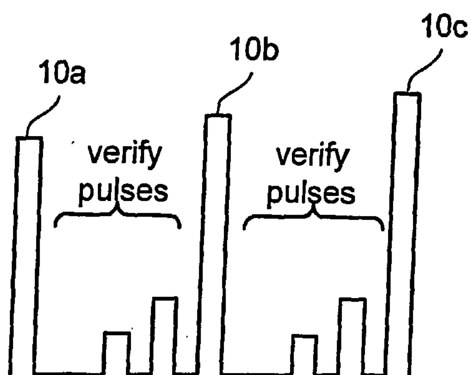
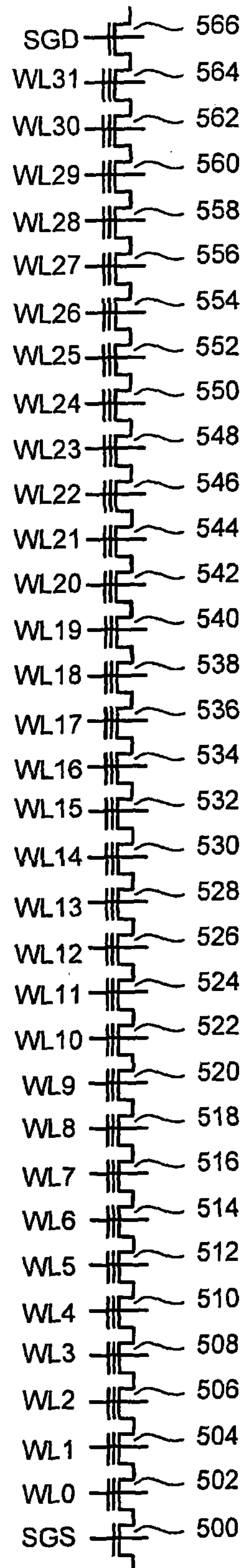


Fig. 8
10a
10b
10c
verify pulses
verify pulses

Fig. 9



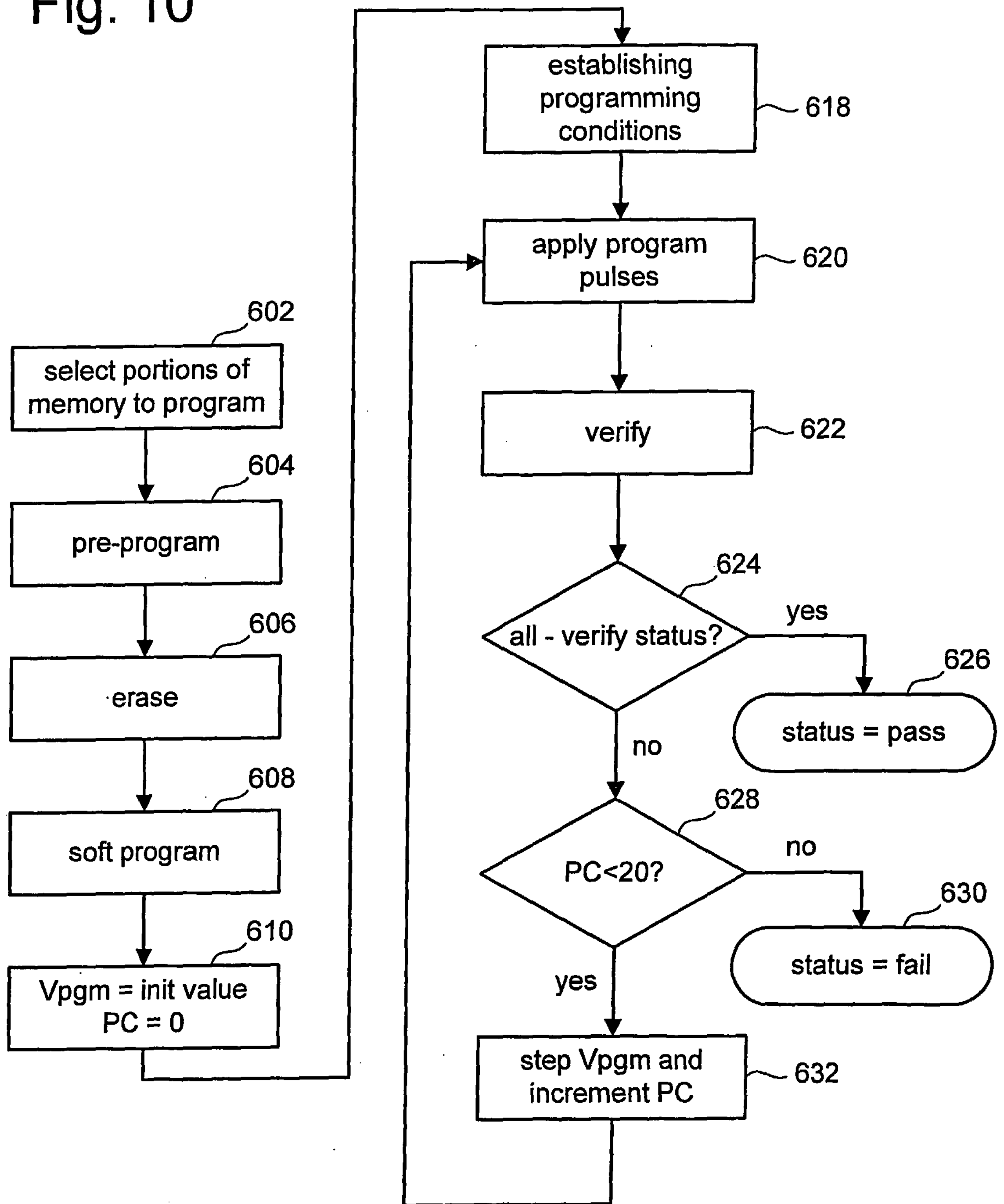


Fig. 10
select portions of memory to program
602
pre-program
604
erase
606
soft program
608
Vpgm = init value
PC = 0
610
establishing programming conditions
618
apply program pulses
620
verify
622
all - verify status?
624
yes
status = pass
626
no
PC<20?
628
no
status = fail
630
yes
step Vpgm and increment PC
632

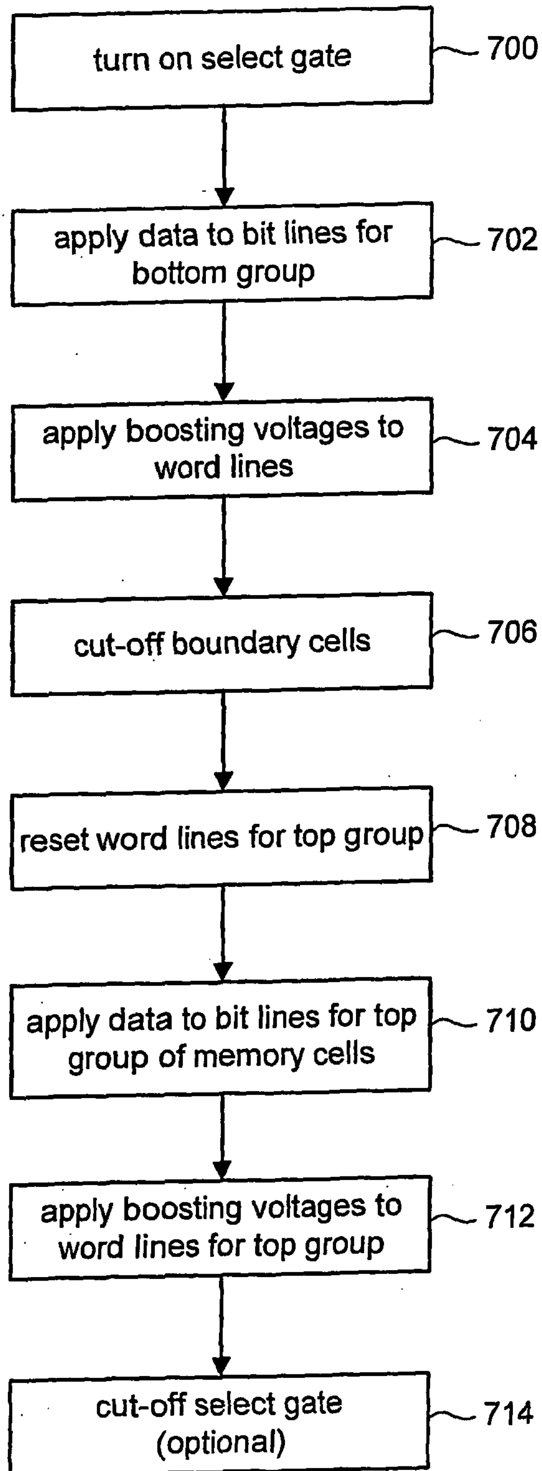


Fig. 11
turn on select gate
700
apply data to bit lines for bottom group
702
apply boosting voltages to word lines
704
cut-off boundary cells
706
reset word lines for top group
708
apply data to bit lines for top group of memory cells
710
apply boosting voltages to word lines for top group
712
cut-off select gate (optional)
714

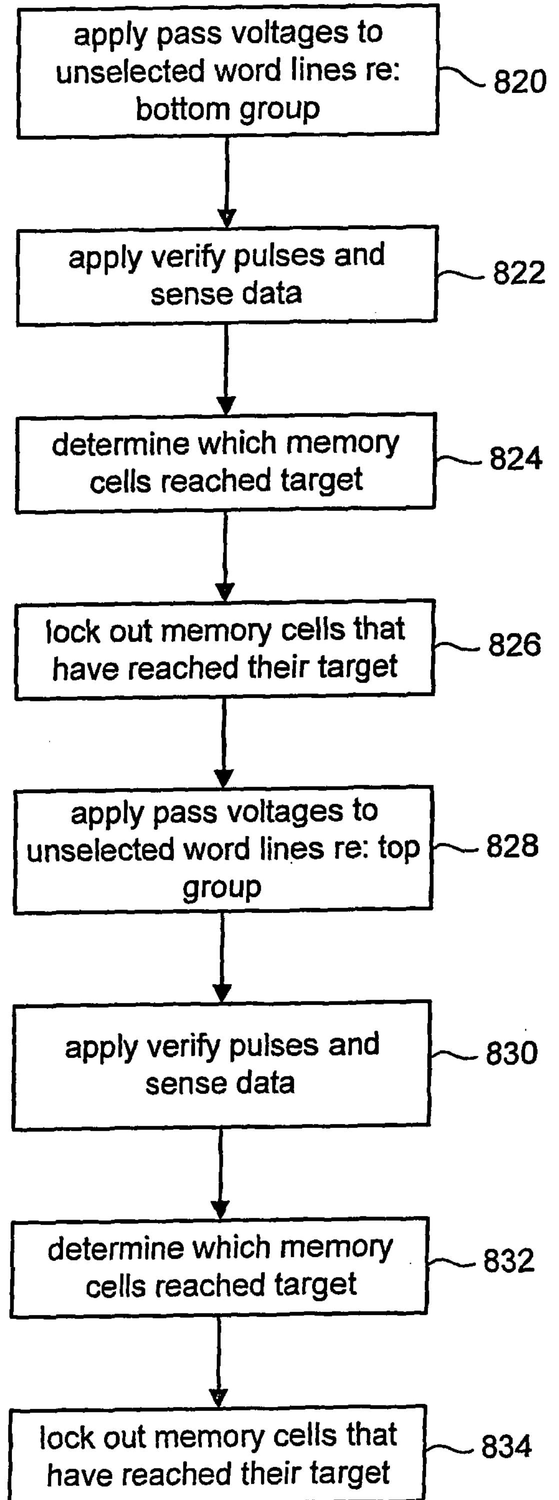


Fig. 14
apply pass voltages to unselected word lines re: bottom group
820
apply verify pulses and sense data
822
determine which memory cells reached target
824
lock out memory cells that have reached their target
826
apply pass voltages to unselected word lines re: top group
828
apply verify pulses and sense data
830
determine which memory cells reached target
832
lock out memory cells that have reached their target
834

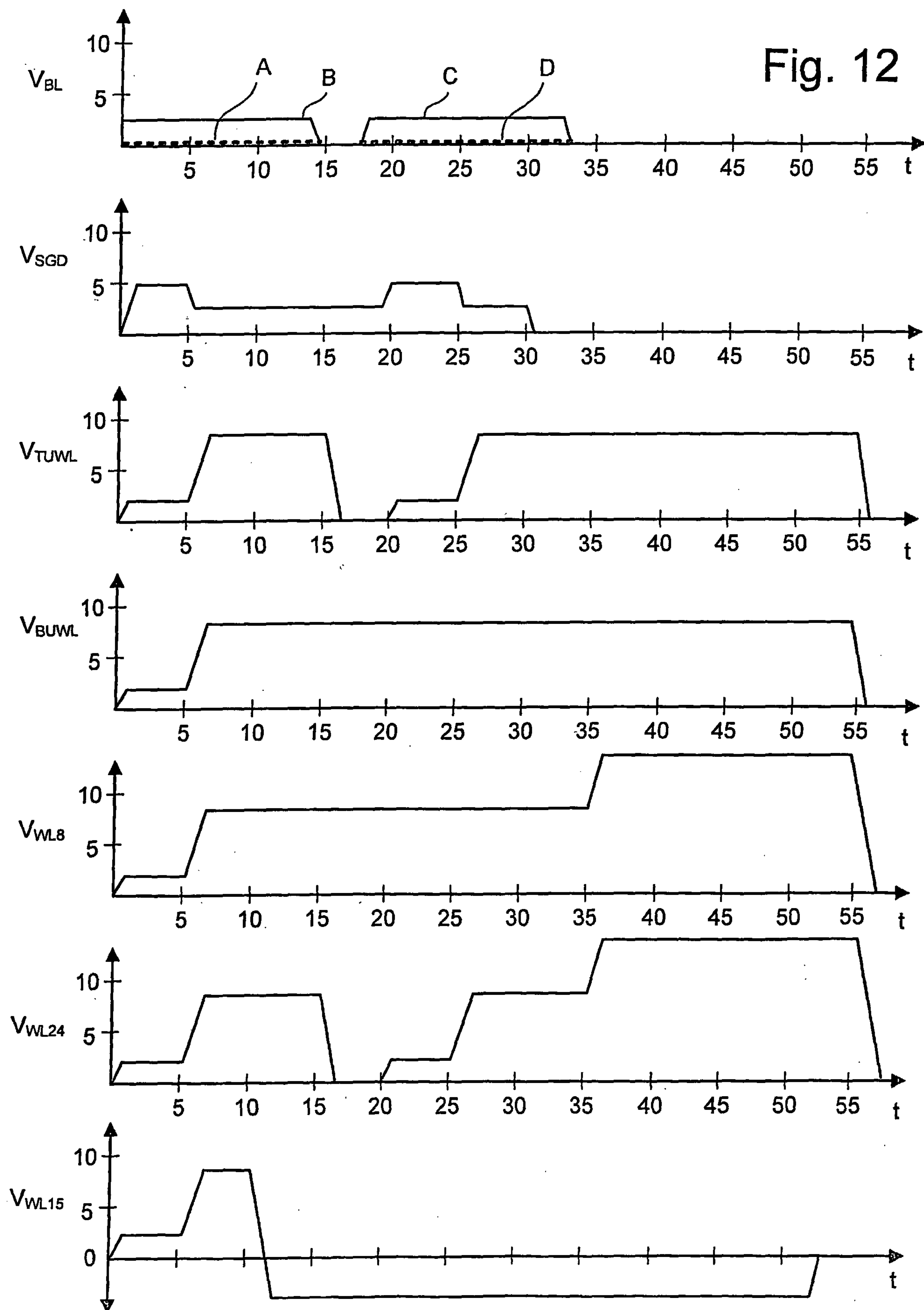
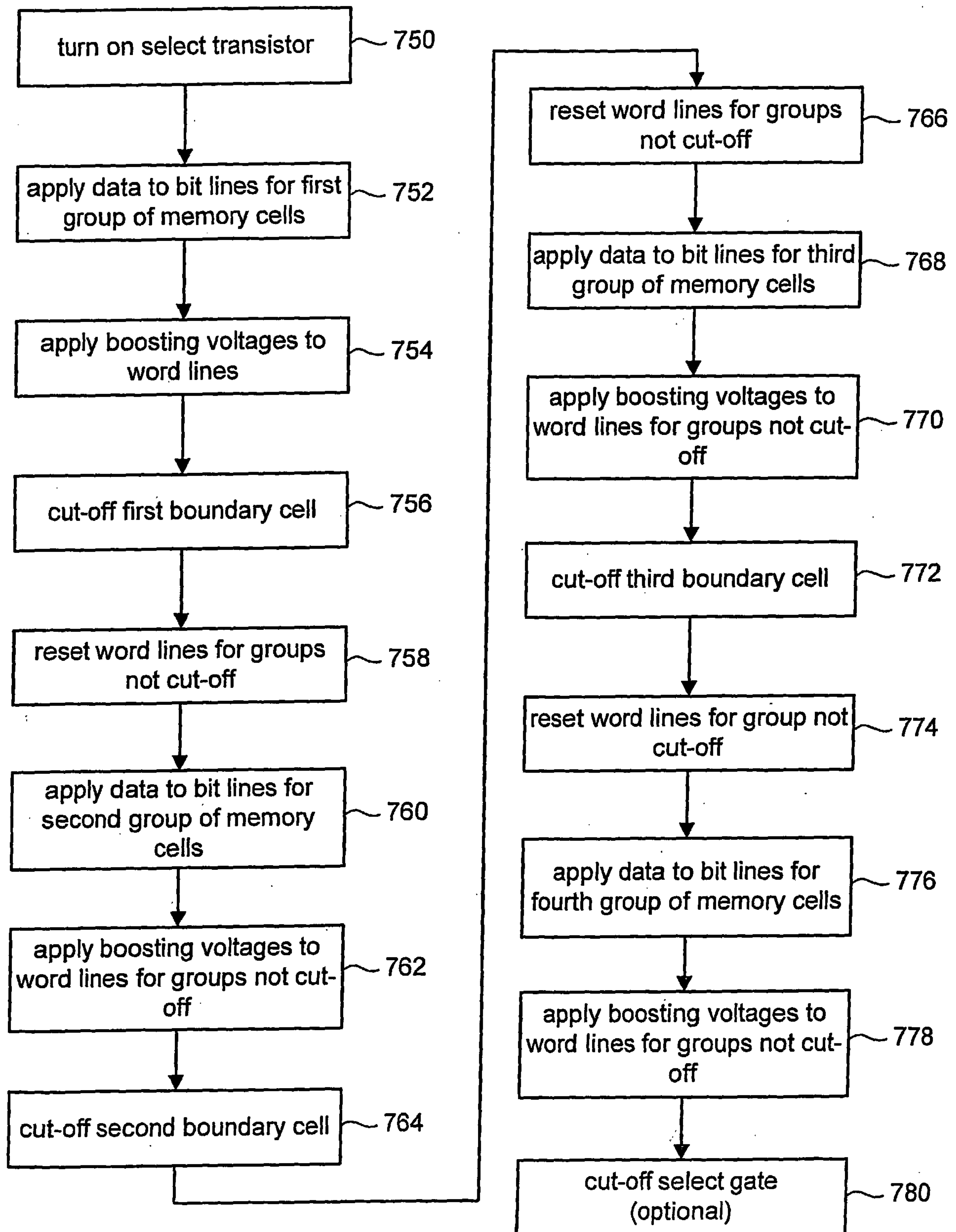


Fig. 12
V_{BL}
A
B
C
D
V_{SGD}
V_{TUWL}
V_{BUWL}
V_{WL8}
V_{WL24}
V_{WL15}
t

Fig. 13



INTERNATIONAL SEARCH REPORT

International Application No
PCT/US2005/019393

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G11C16/04 G11C16/10

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G11C

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 529 410 B1 (HAN MICHAEL ET AL) 4 March 2003 (2003-03-04)	1-9, 12-16, 19-22, 26-28, 31,32, 34,36
	abstract; claims 6,13; figure 3 column 5, paragraph 2-4	
A		10,11, 17,18, 23-25, 29,30, 33,35, 37-40
A	US 6 091 633 A (CERNEA ET AL) 18 July 2000 (2000-07-18) the whole document	1-40

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
10 November 2005	16/11/2005

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Trifonov, A

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US2005/019393

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
US 6529410	B1	04-03-2003	NONE	
US 6091633	A	18-07-2000	NONE	