US 20250266094A1

(19) **United States**

(12) **Patent Application Publication**

Prakash et al.

(10) **Pub. No.: US 2025/0266094 A1**

(43) **Pub. Date: Aug. 21, 2025**

(54) **REAL TIME RAMP RATE ADJUSTMENT FOR BETTER PERFORMANCE AND CURRENT CONSUMPTION TRADEOFF**

(71) Applicant: **Western Digital Technologies, Inc.**, San Jose, CA (US)

(72) Inventors: **Abhijith Prakash**, Milpitas, CA (US); **Parth Amin**, Fremont, CA (US); **Anubhav Khandelwal**, San Jose, CA (US)

(21) Appl. No.: **18/442,709**

(22) Filed: **Feb. 15, 2024**

Publication Classification

(51) **Int. Cl.**

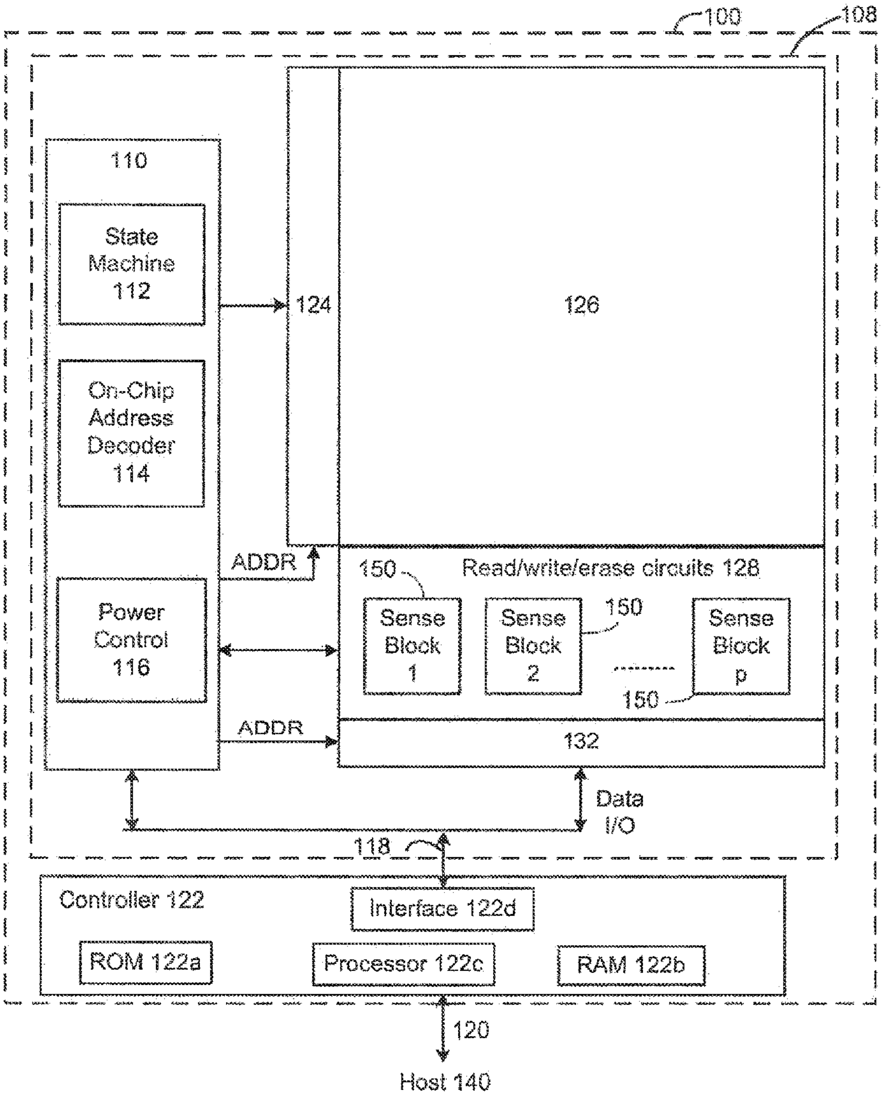
G11C 16/10	(2006.01)
G11C 16/08	(2006.01)
G11C 16/32	(2006.01)

(52) **U.S. Cl.**

CPC *G11C 16/102* (2013.01); *G11C 16/08* (2013.01); *G11C 16/32* (2013.01)

(57) **ABSTRACT**

A memory apparatus includes memory cells each connected to one of a plurality of word lines and configured to store a threshold voltage corresponding to one of a plurality of data states. The memory apparatus also includes a control means configured to identify ones of the plurality of word lines as slow word lines. The control means is also configured to ramp at least one program pulse applied to the slow word lines to a program kick voltage higher in magnitude than a program voltage for a program kick period of time during at least one program loop of a program operation.



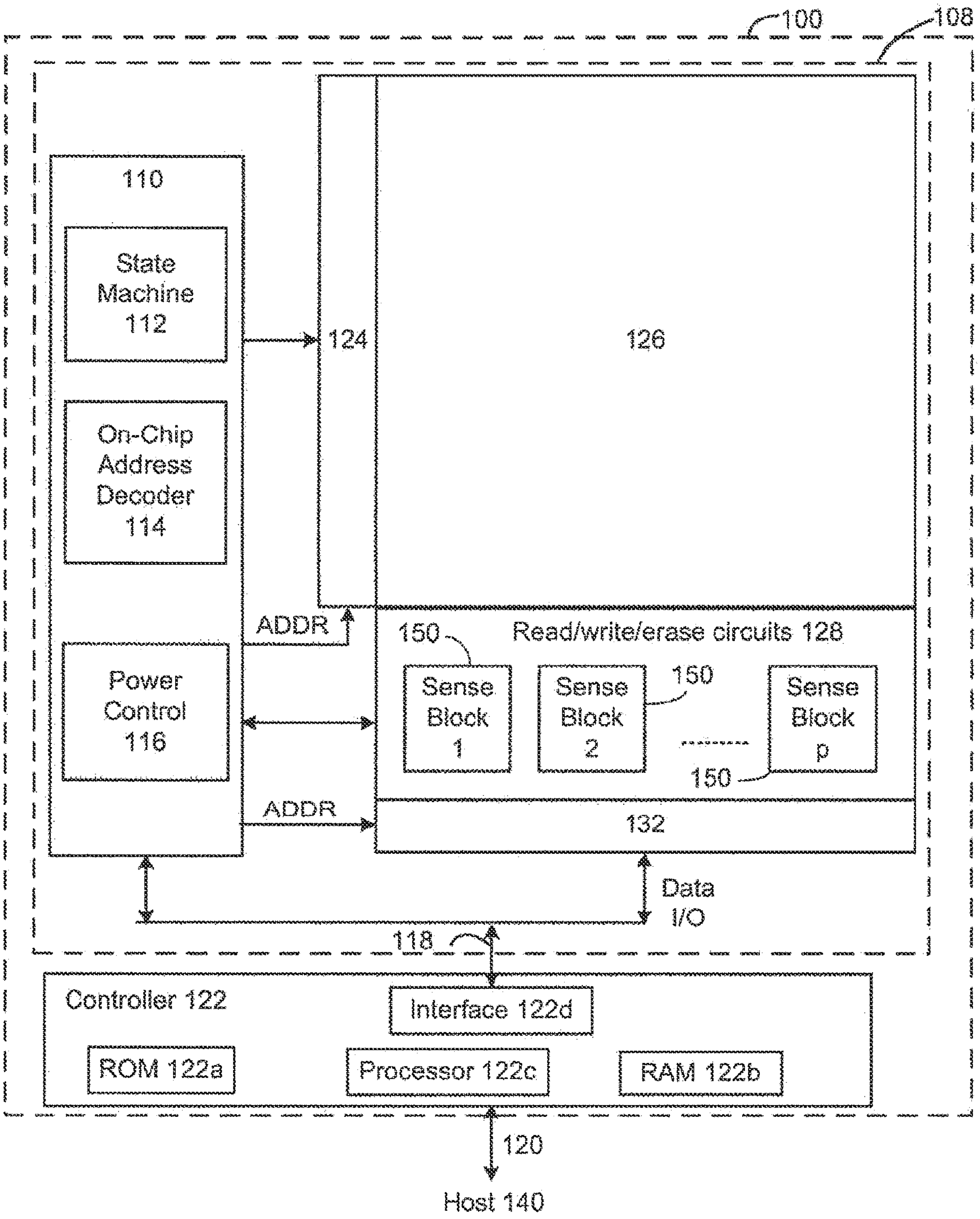

100
108
110
State Machine 112
On-Chip Address Decoder 114
Power Control 116
124
126
ADDR
Read/write/erase circuits 128
150
Sense Block 1
Sense Block 2
Sense Block p
150
150
ADDR
132
Data I/O
118
Controller 122
Interface 122d
ROM 122a
Processor 122c
RAM 122b
120
Host 140

FIG. 1

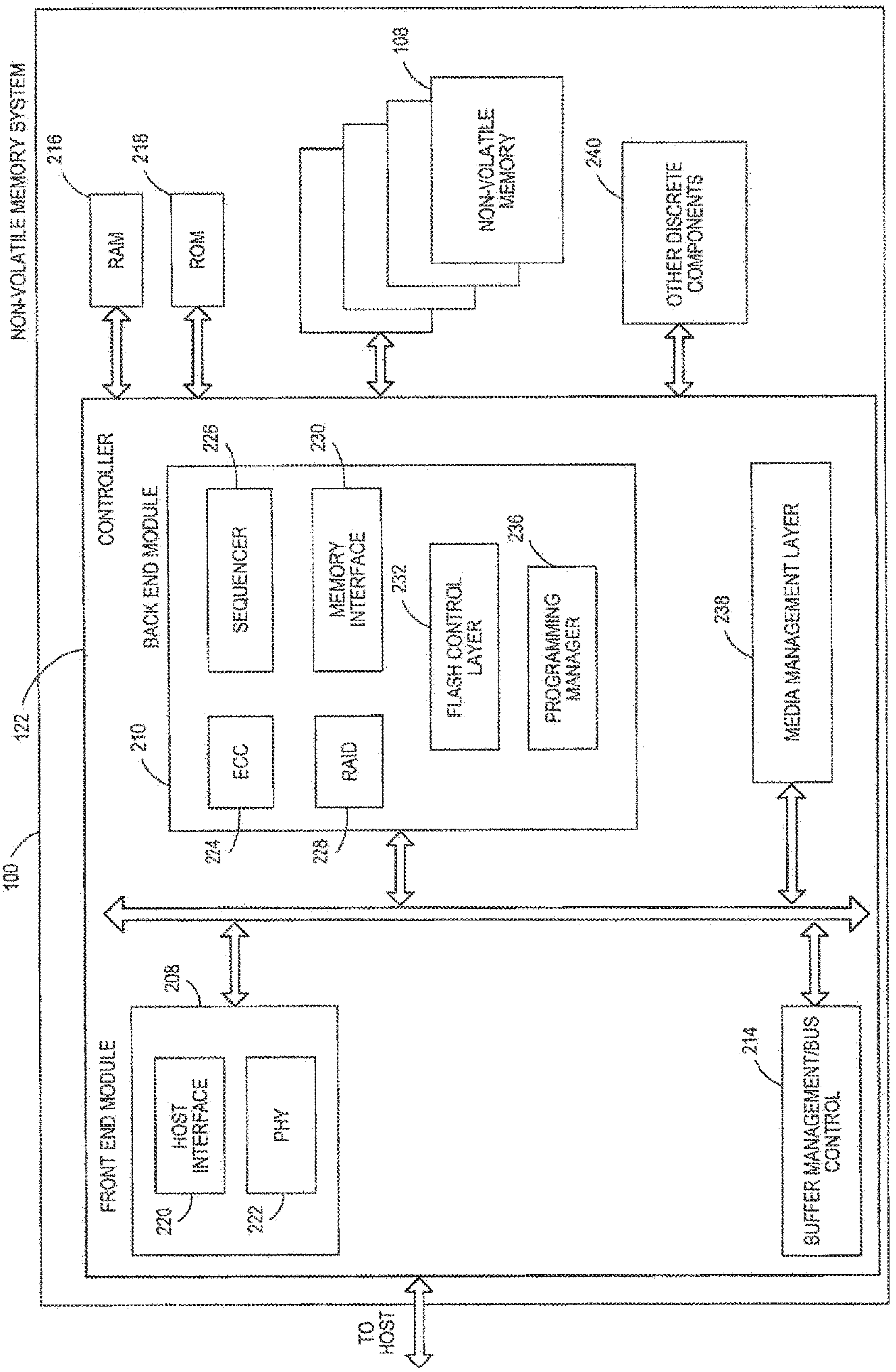

NON-VOLATILE MEMORY SYSTEM
100
122
216
RAM
218
ROM
108
NON-VOLATILE MEMORY
240
OTHER DISCRETE COMPONENTS
CONTROLLER
BACK END MODULE
210
226
SEQUENCER
230
MEMORY INTERFACE
232
FLASH CONTROL LAYER
236
PROGRAMMING MANAGER
ECC
224
RAID
228
238
MEDIA MANAGEMENT LAYER
FRONT END MODULE
208
HOST INTERFACE
220
PHY
222
214
BUFFER MANAGEMENT/BUS CONTROL
TO HOST

FIG. 2

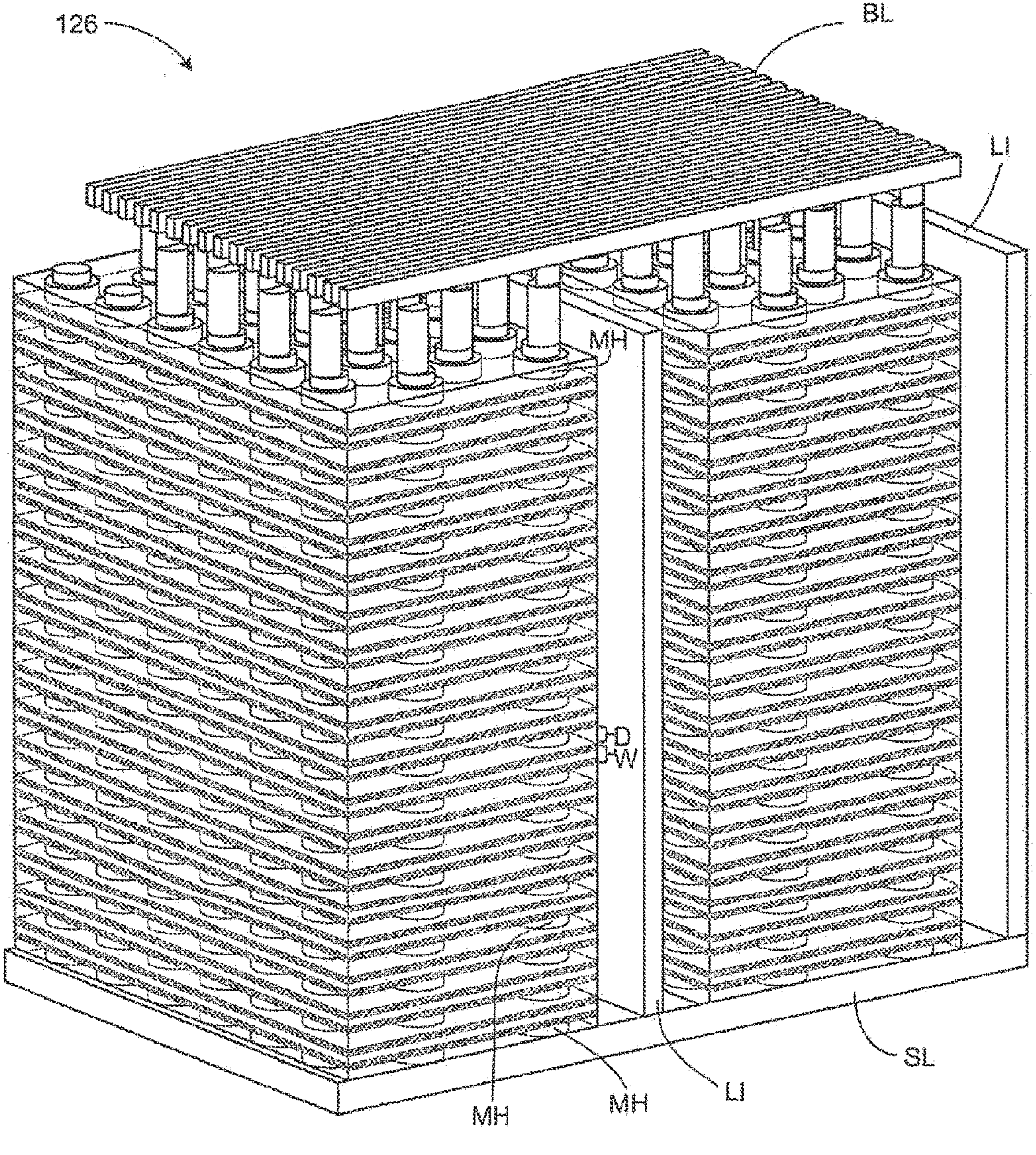


FIG. 3

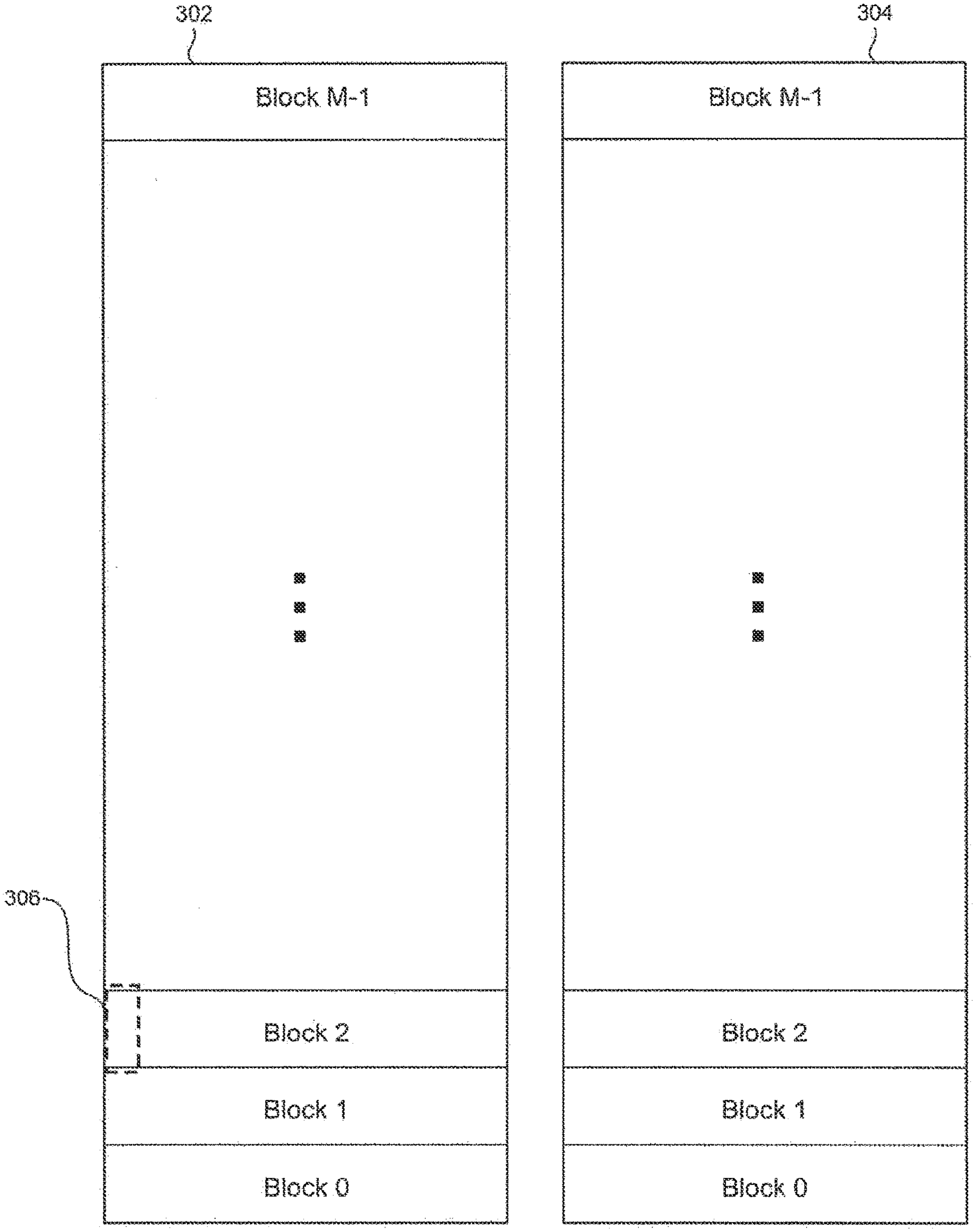

302
304
Block M-1
Block M-1
306
Block 2
Block 2
Block 1
Block 1
Block 0
Block 0

FIG. 4A

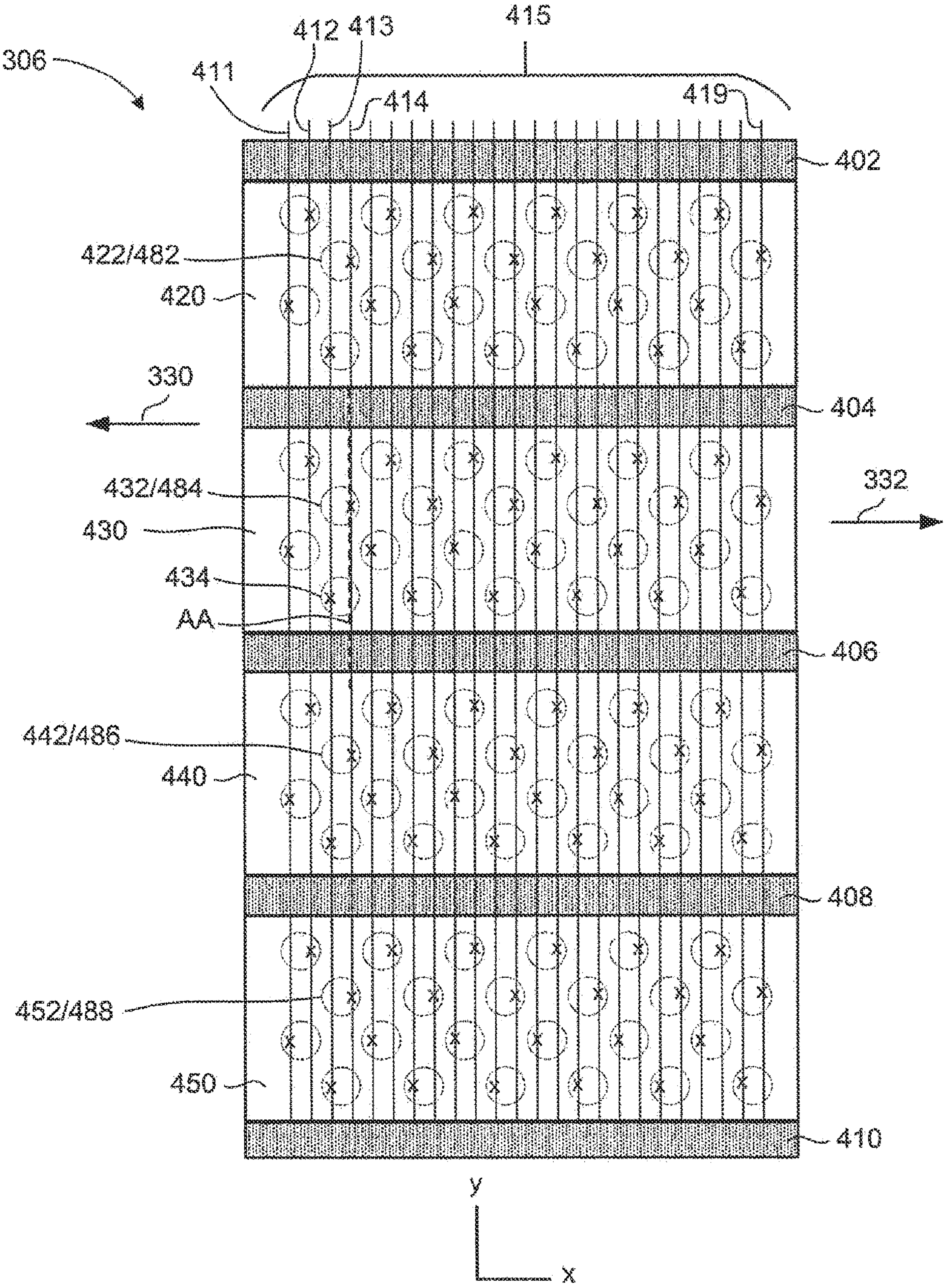

306
415
412
413
411
414
419
402
422/482
420
330
404
432/484
332
430
434
AA
406
442/486
440
408
452/488
450
410
y
x

FIG. 4B

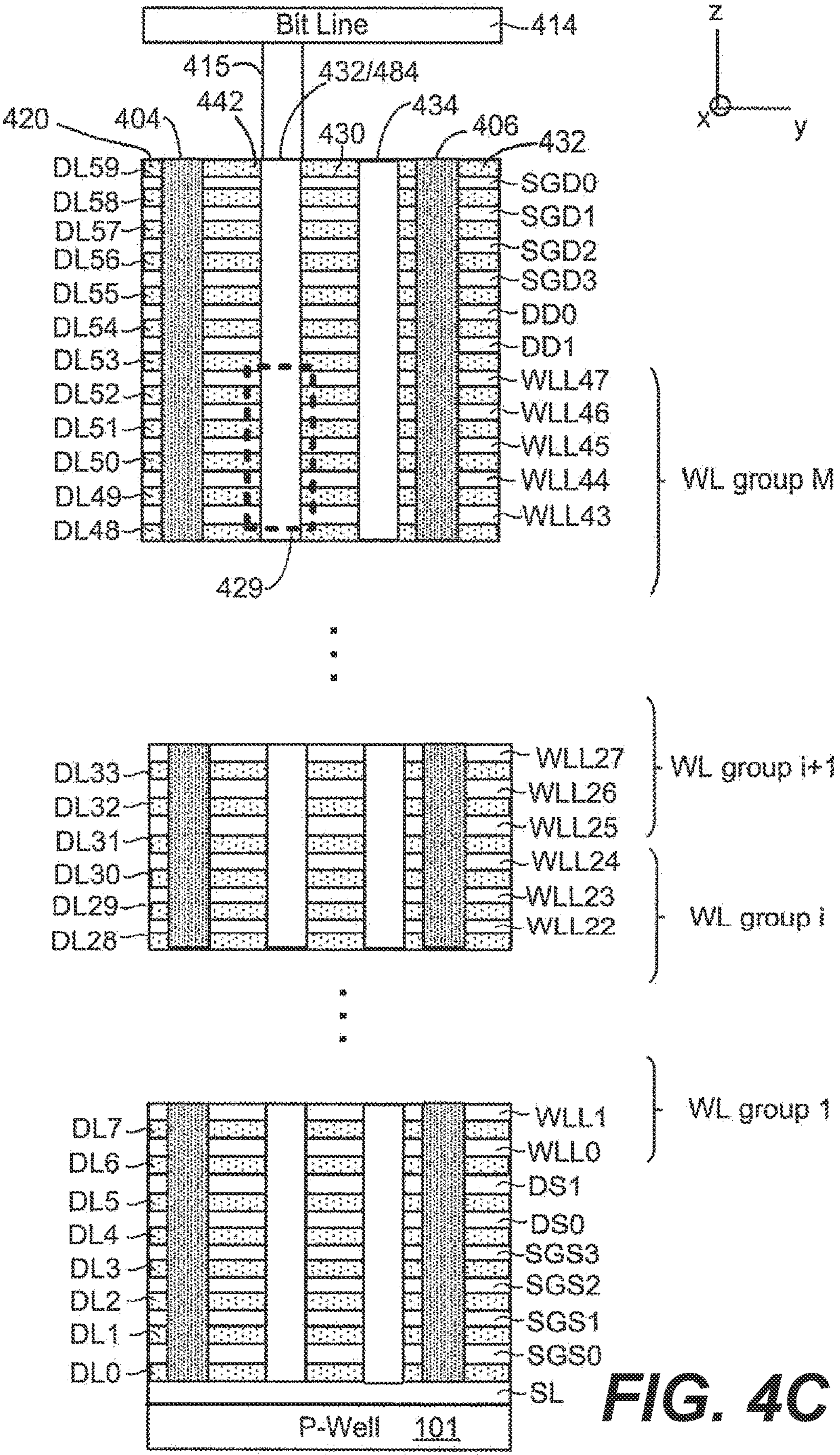

Bit Line
414
415
432/484
442
434
420
404
430
406
432
z
x
y
DL59
DL58
DL57
DL56
DL55
DL54
DL53
DL52
DL51
DL50
DL49
DL48
SGD0
SGD1
SGD2
SGD3
DD0
DD1
WLL47
WLL46
WLL45
WLL44
WLL43
WL group M
429
DL33
DL32
DL31
DL30
DL29
DL28
WLL27
WLL26
WLL25
WLL24
WLL23
WLL22
WL group i+1
WL group i
DL7
DL6
DL5
DL4
DL3
DL2
DL1
DL0
WLL1
WLL0
DS1
DS0
SGS3
SGS2
SGS1
SGS0
SL
WL group 1
P-Well 101

FIG. 4C

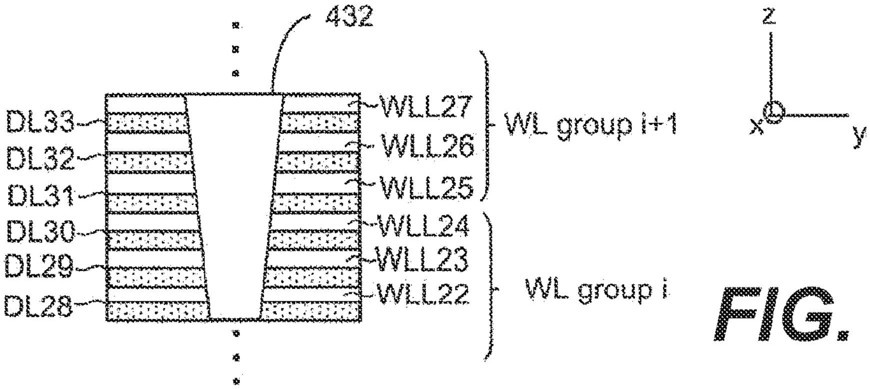

432
DL33
DL32
DL31
DL30
DL29
DL28
WLL27
WLL26
WLL25
WLL24
WLL23
WLL22
WL group i+1
WL group i
z
x
y

FIG. 4D

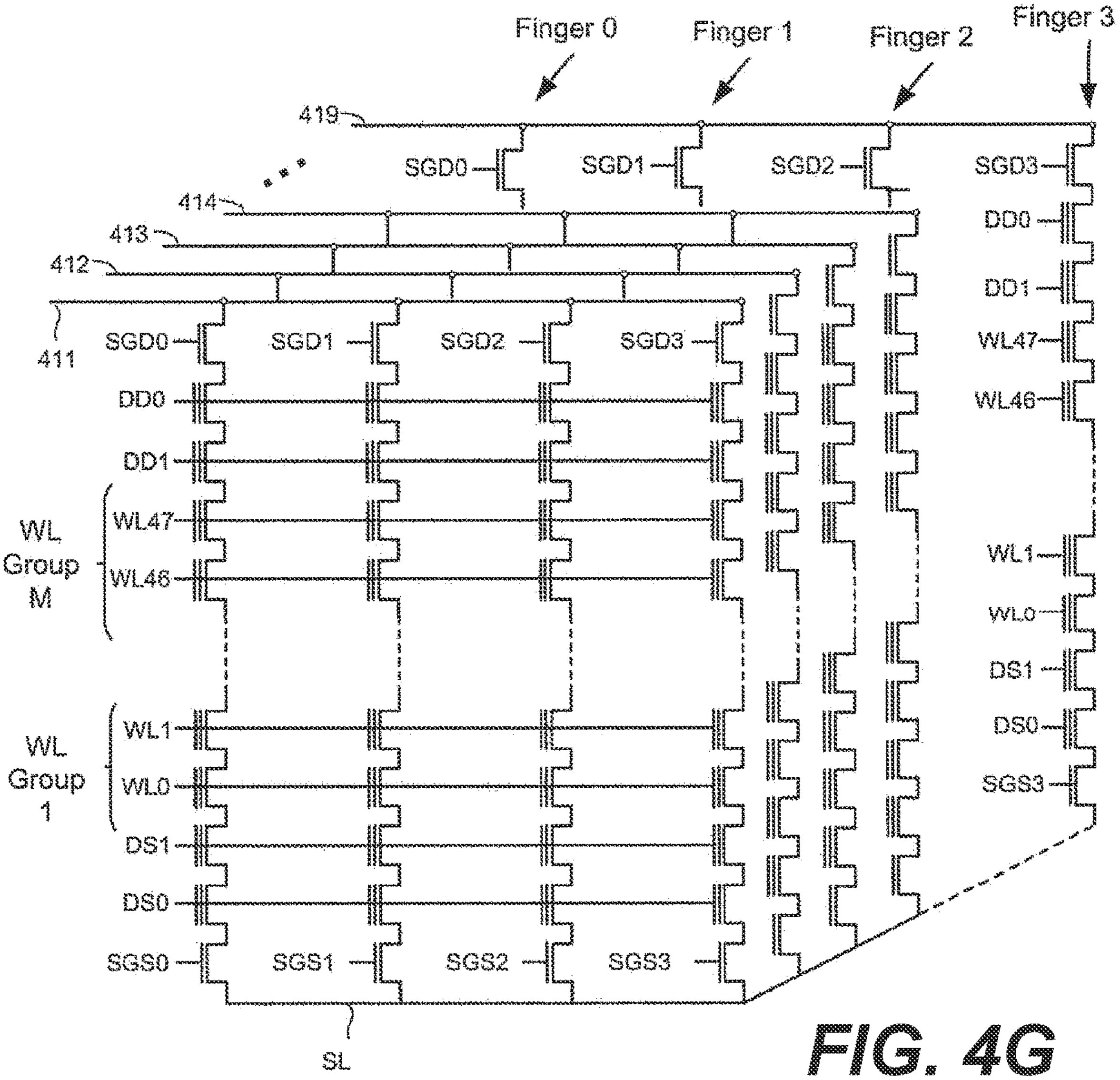

Finger 0
Finger 1
Finger 2
Finger 3
419
414
413
412
411
SGD0
SGD1
SGD2
SGD3
DD0
DD1
WL47
WL46
WL1
WL0
DS1
DS0
SGS0
SGS1
SGS2
SGS3
WL Group M
WL Group 1
SL

FIG. 4G

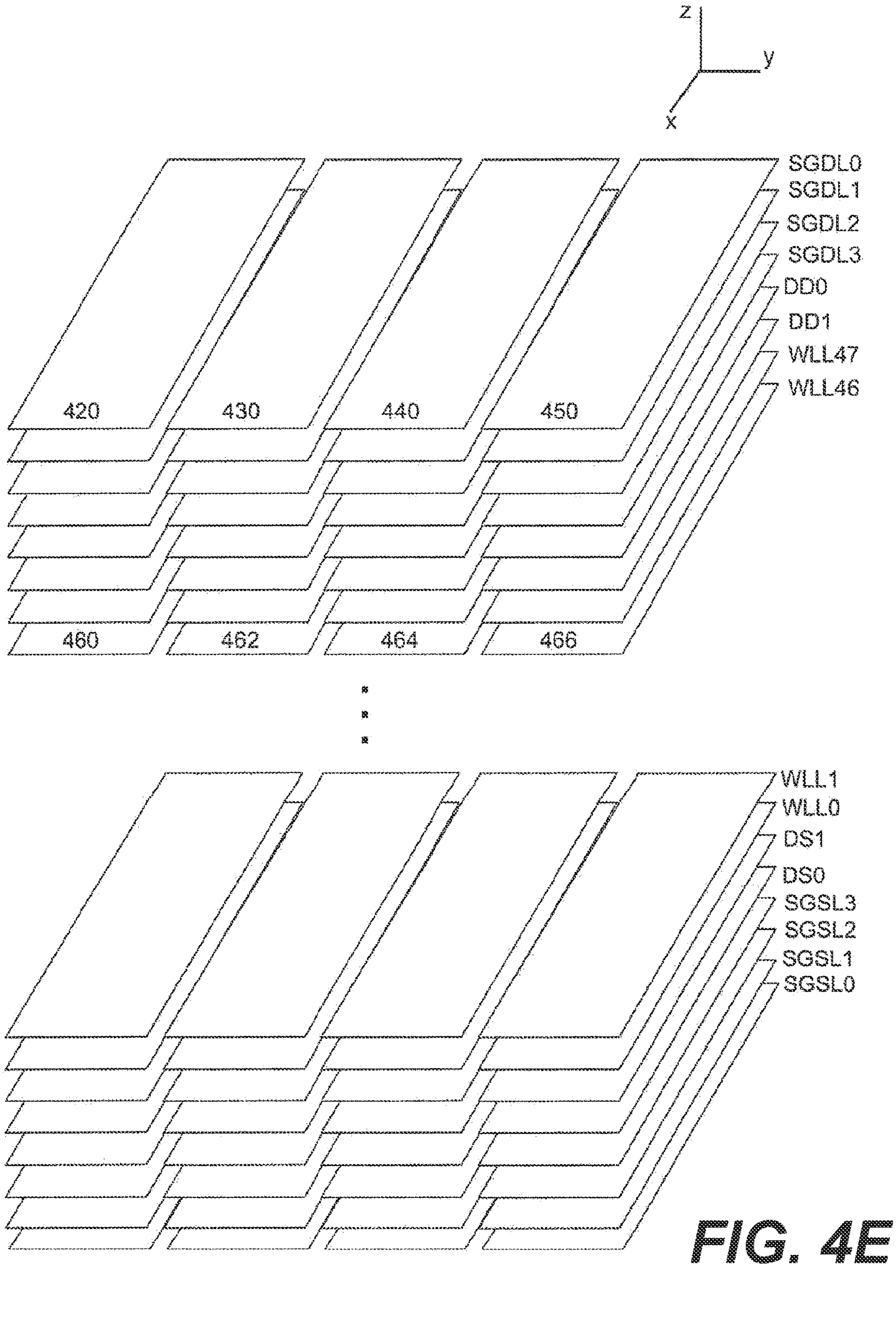

z
y
x
SGDL0
SGDL1
SGDL2
SGDL3
DD0
DD1
WLL47
WLL46
420
430
440
450
460
462
464
466
WLL1
WLL0
DS1
DS0
SGSL3
SGSL2
SGSL1
SGSL0

FIG. 4E

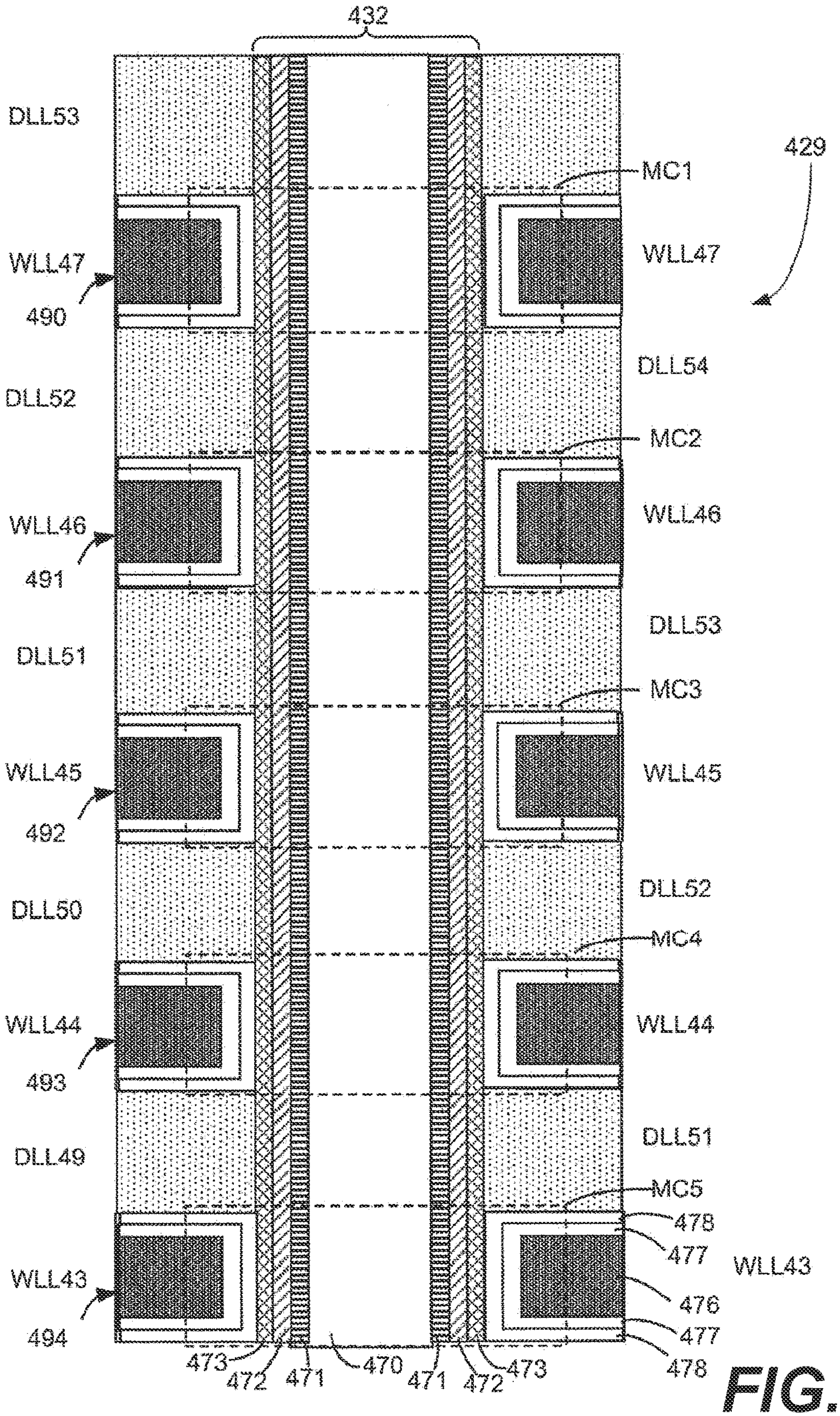

432
429
DLL53
MC1
WLL47
490
WLL47
DLL52
DLL54
MC2
WLL46
491
WLL46
DLL51
DLL53
MC3
WLL45
492
WLL45
DLL50
DLL52
MC4
WLL44
493
WLL44
DLL49
DLL51
MC5
478
477
WLL43
494
WLL43
476
477
478
473
472
471
470
471
472
473

FIG. 4F

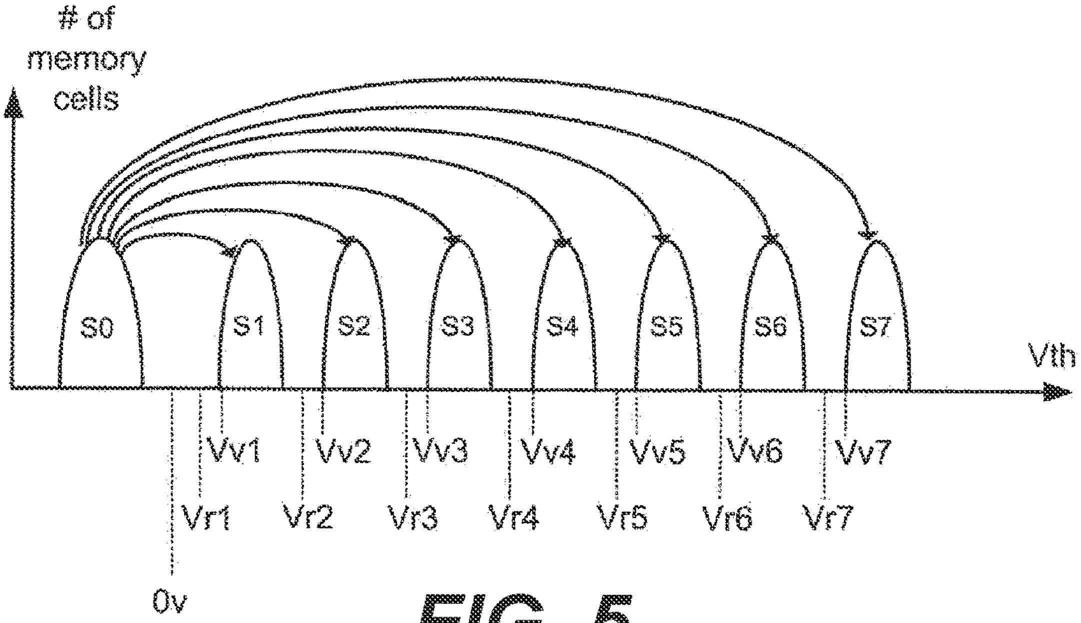


FIG. 5

	S0	S1	S2	S3	S4	S5	S6	S7
Upper Page	1	1	1	0	0	0	0	1
Middle Page	1	1	0	0	1	1	0	0
Lower Page	1	0	0	0	0	1	1	1

FIG. 6

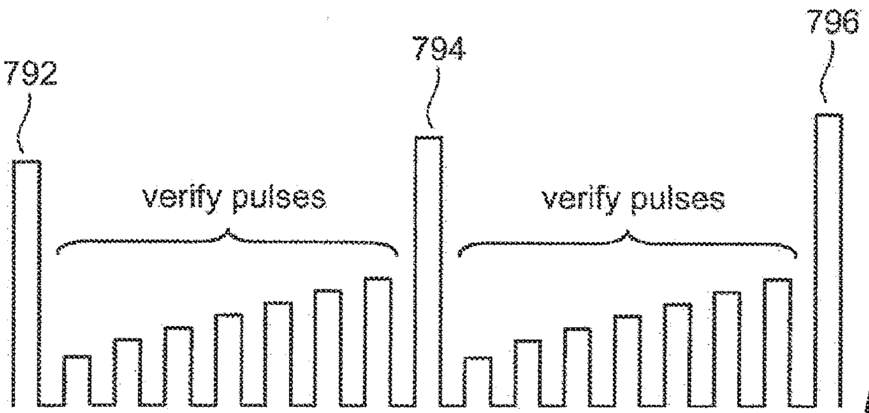


FIG. 7C

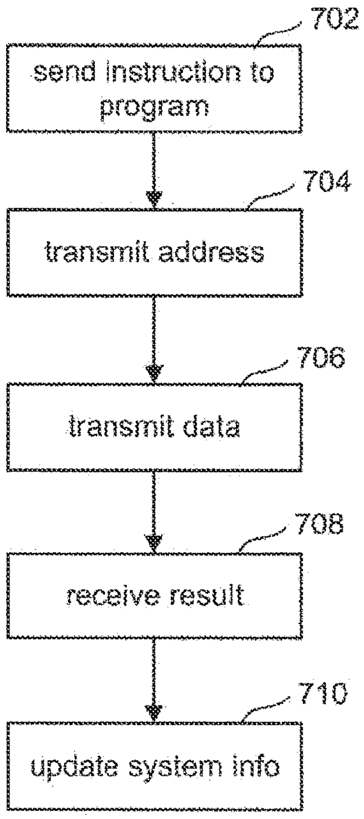

702
send instruction to program
704
transmit address
706
transmit data
708
receive result
710
update system info

FIG. 7A

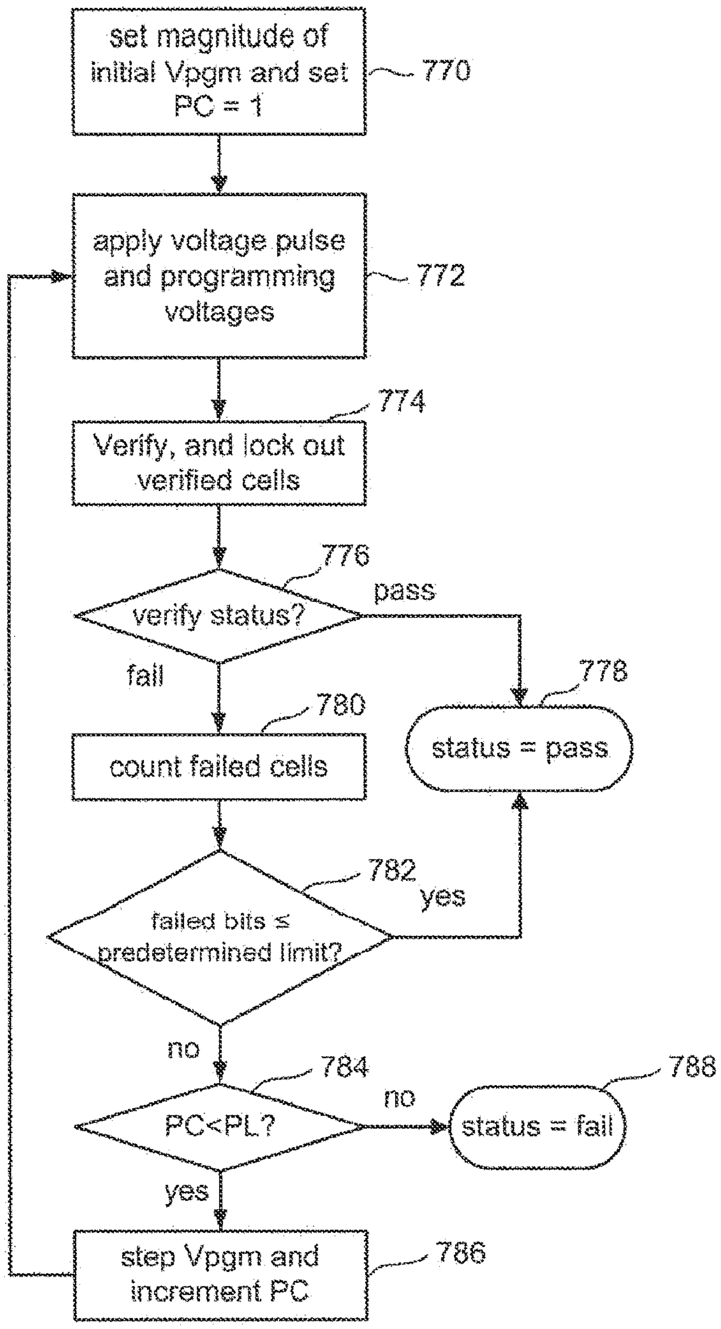

set magnitude of initial Vpgm and set PC = 1
770
apply voltage pulse and programming voltages
772
Verify, and lock out verified cells
774
776
verify status?
pass
fail
778
status = pass
780
count failed cells
782
yes
failed bits ≤ predetermined limit?
no
784
PC<PL?
no
788
status = fail
yes
step Vpgm and increment PC
786

FIG. 7B

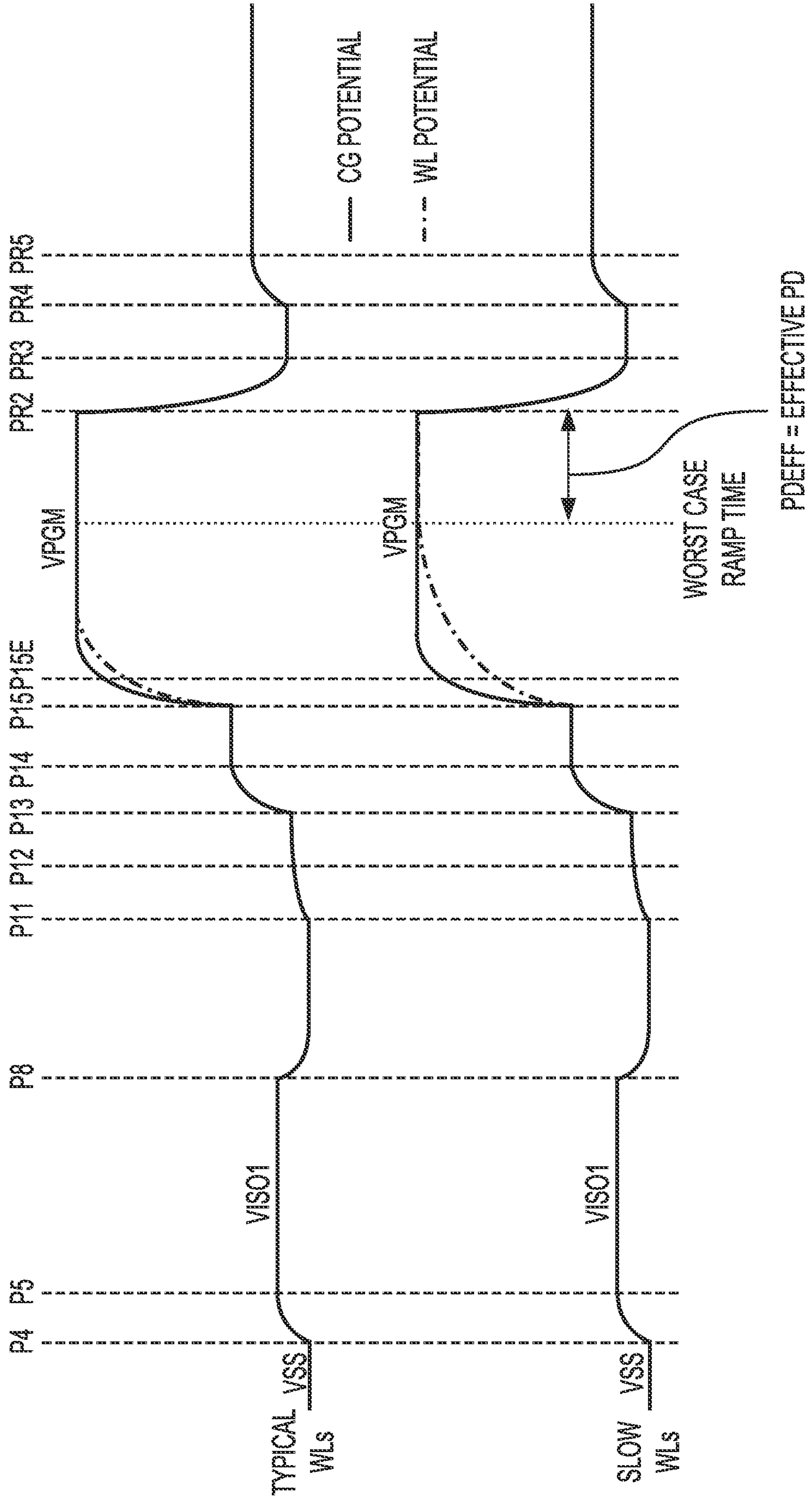

TYPICAL WLs
SLOW WLs
VSS
VSS
VISO1
VISO1
VPGM
VPGM
P4
P5
P8
P11
P12
P13
P14
P15
P15E
PR2
PR3
PR4
PR5
— CG POTENTIAL
-·-·- WL POTENTIAL
WORST CASE RAMP TIME
PDEFF = EFFECTIVE PD CLK TIME FOR PGM

FIG. 8

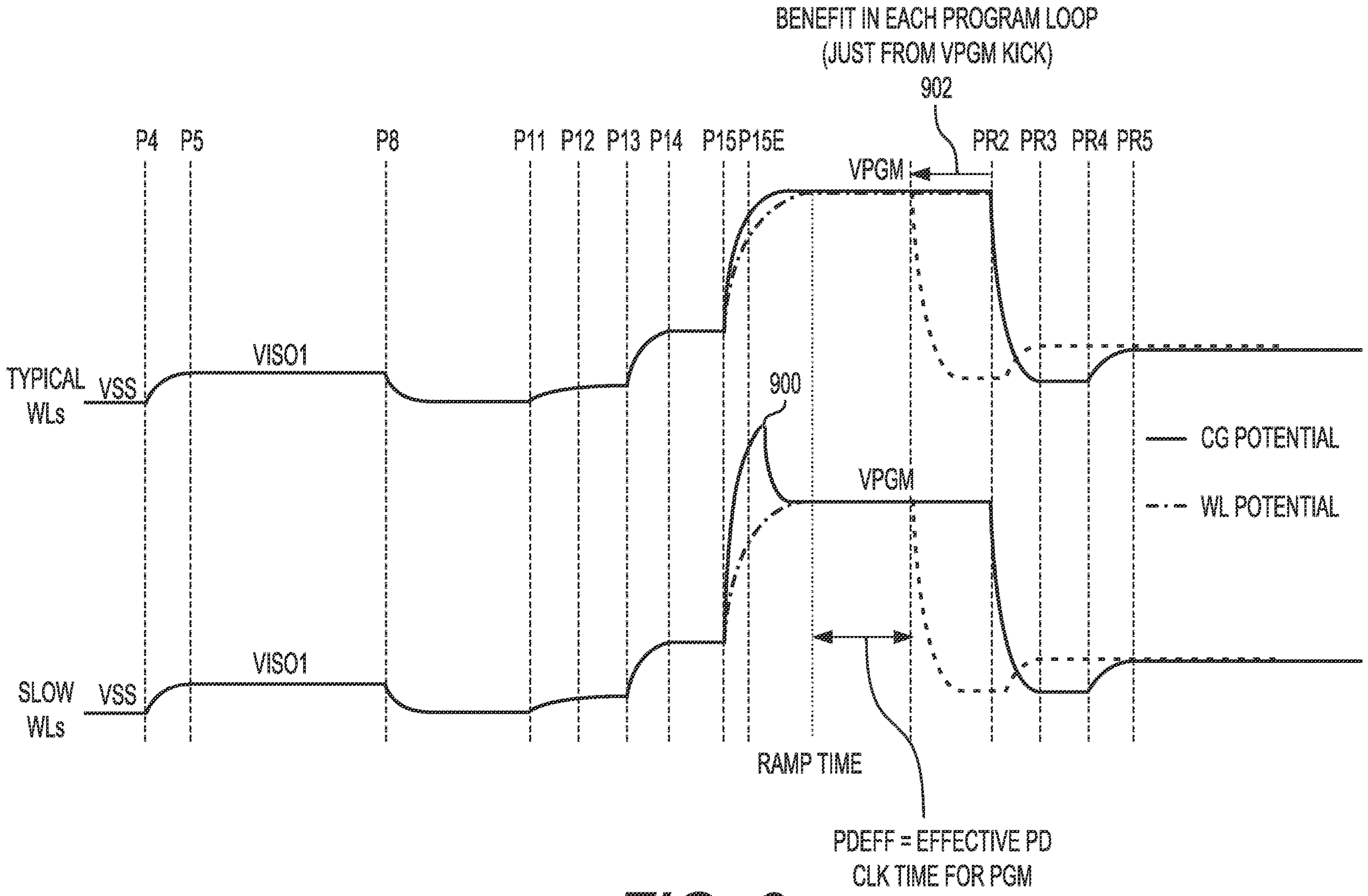

BENEFIT IN EACH PROGRAM LOOP
(JUST FROM VPGM KICK)
902
P4
P5
P8
P11
P12
P13
P14
P15
P15E
PR2
PR3
PR4
PR5
VPGM
TYPICAL WLs
VSS
VISO1
900
CG POTENTIAL
VPGM
WL POTENTIAL
SLOW WLs
VSS
VISO1
RAMP TIME
PDEFF = EFFECTIVE PD CLK TIME FOR PGM

FIG. 9

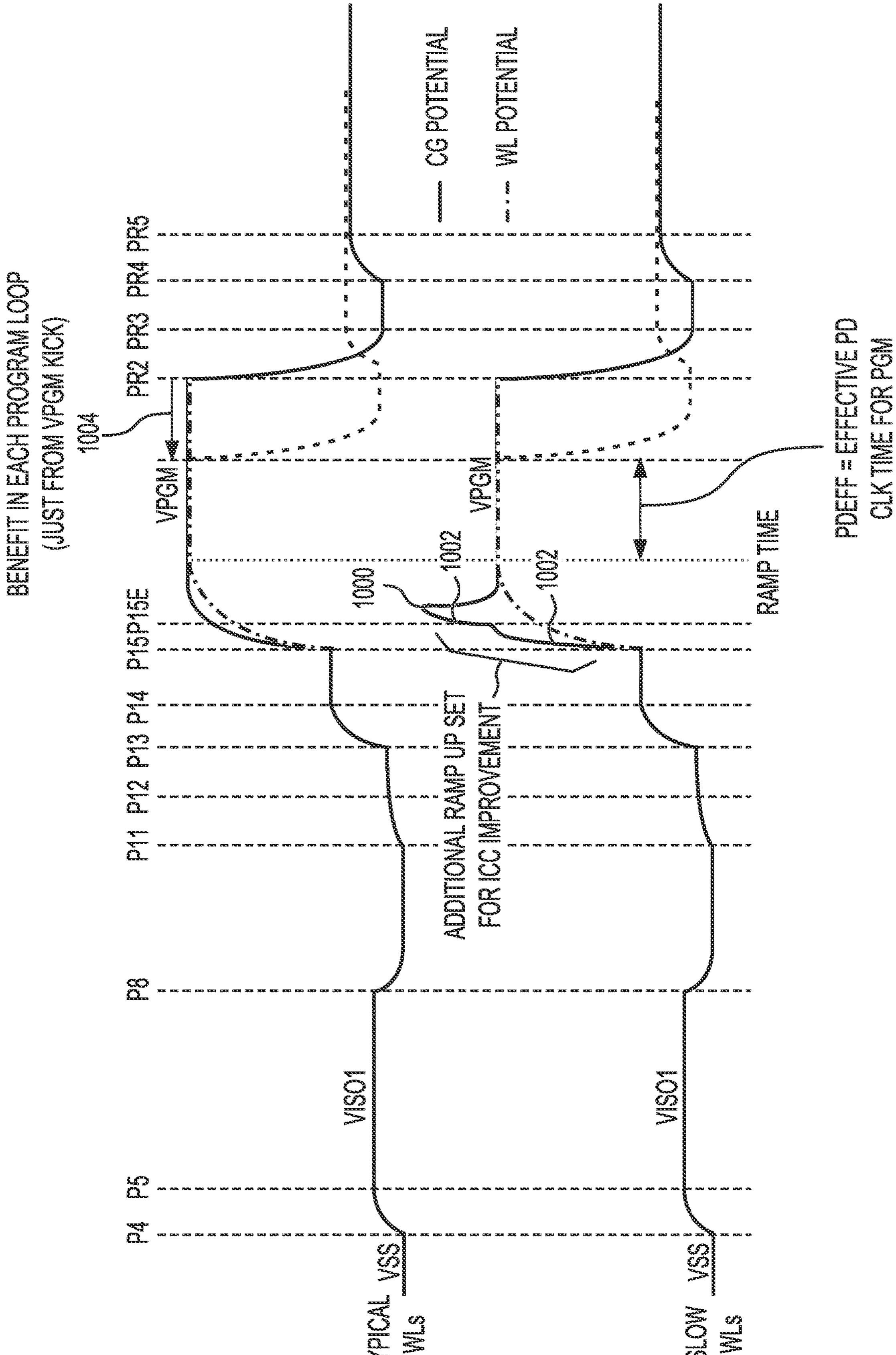

BENEFIT IN EACH PROGRAM LOOP
(JUST FROM VPGM KICK)
1004
P4
P5
P8
P11
P12
P13
P14
P15
P15E
PR2
PR3
PR4
PR5
TYPICAL WLs
VSS
VISO1
VPGM
SLOW WLs
VSS
VISO1
VPGM
CG POTENTIAL
WL POTENTIAL
1000
1002
1002
ADDITIONAL RAMP UP SET
FOR ICC IMPROVEMENT
RAMP TIME
PDEFF = EFFECTIVE PD
CLK TIME FOR PGM

FIG. 10

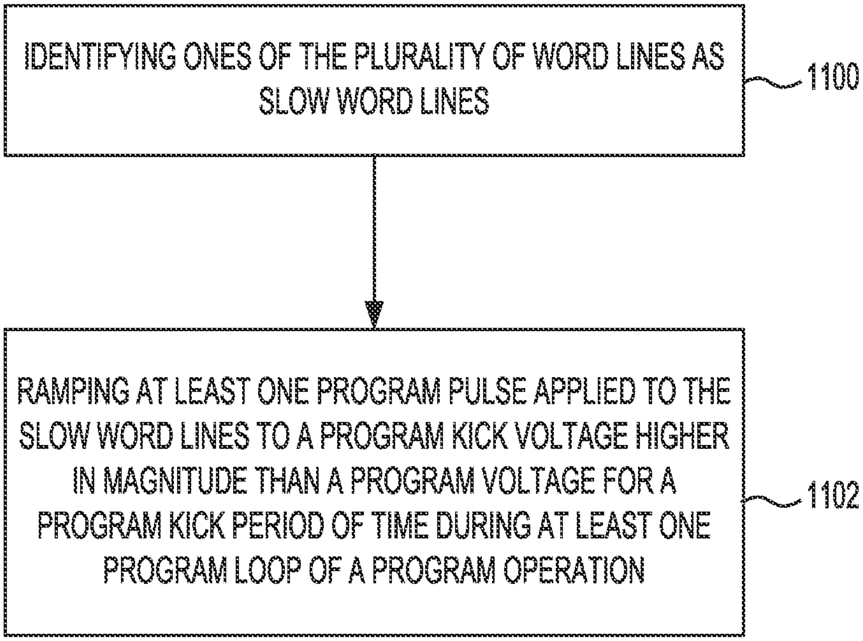

IDENTIFYING ONES OF THE PLURALITY OF WORD LINES AS SLOW WORD LINES
1100
RAMPING AT LEAST ONE PROGRAM PULSE APPLIED TO THE SLOW WORD LINES TO A PROGRAM KICK VOLTAGE HIGHER IN MAGNITUDE THAN A PROGRAM VOLTAGE FOR A PROGRAM KICK PERIOD OF TIME DURING AT LEAST ONE PROGRAM LOOP OF A PROGRAM OPERATION
1102

FIG. 11

REAL TIME RAMP RATE ADJUSTMENT FOR BETTER PERFORMANCE AND CURRENT CONSUMPTION TRADEOFF

FIELD

[0001] This application relates to non-volatile memory apparatuses and the operation of non-volatile memory apparatuses.

BACKGROUND

[0002] This section provides background information related to the technology associated with the present disclosure and, as such, is not necessarily prior art.

[0003] Semiconductor memory is widely used in various electronic devices such as cellular telephones, digital cameras, personal digital assistants, medical electronics, mobile computing devices, servers, solid state drives, non-mobile computing devices and other devices. Semiconductor memory may comprise non-volatile memory or volatile memory. Non-volatile memory allows information to be stored and retained even when the non-volatile memory is not connected to a source of power (e.g., a battery).

[0004] As memory structures increase in density, it becomes more challenging to maintain the integrity of the data being stored. Thus, techniques are needed to overcome such challenges.

SUMMARY

[0005] This section provides a general summary of the present disclosure and is not a comprehensive disclosure of its full scope or all of its features and advantages.

[0006] An object of the present disclosure is to provide a memory apparatus and a method of operating the memory apparatus that address and overcome the above-noted shortcomings.

[0007] Accordingly, it is an aspect of the present disclosure to provide a memory apparatus including memory cells each connected to one of a plurality of word lines and configured to store a threshold voltage corresponding to one of a plurality of data states. The memory apparatus also includes a control means configured to identify ones of the plurality of word lines as slow word lines. The control means is also configured to ramp at least one program pulse applied to the slow word lines to a program kick voltage higher in magnitude than a program voltage for a program kick period of time during at least one program loop of a program operation.

[0008] According to another aspect of the disclosure, a controller in communication with a memory apparatus including memory cells each connected to one of a plurality of word lines and configured to store a threshold voltage corresponding to one of a plurality of data states is provided. The controller is configured to identify ones of the plurality of word lines as slow word lines. The controller is also configured to instruct the memory apparatus to ramp at least one program pulse applied to the slow word lines to a program kick voltage higher in magnitude than a program voltage for a program kick period of time during at least one program loop of a program operation.

[0009] According to an additional aspect of the disclosure a method of operating a memory apparatus is provided. The memory apparatus includes memory cells each connected to one of a plurality of word lines and configured to store a threshold voltage corresponding to one of a plurality of data states. The method includes the step of identifying ones of the plurality of word lines as slow word lines. The method also includes the step of ramping at least one program pulse applied to the slow word lines to a program kick voltage higher in magnitude than a program voltage for a program kick period of time during at least one program loop of a program operation.

[0010] Further areas of applicability will become apparent from the description provided herein. The description and specific examples in this summary are intended for purposes of illustration only and are not intended to limit the scope of the present disclosure.

DRAWINGS

[0011] The drawings described herein are for illustrative purposes only of selected embodiments and not all possible implementations, and are not intended to limit the scope of the present disclosure.

[0012] FIG. **1** is a functional block diagram of a memory device according to aspects of the disclosure;

[0013] FIG. **2** is a block diagram depicting one embodiment of a memory system according to aspects of the disclosure;

[0014] FIG. **3** is a perspective view of a portion of one embodiment of a monolithic three-dimensional memory structure according to aspects of the disclosure;

[0015] FIG. **4**A is a block diagram of a memory structure having two planes according to aspects of the disclosure;

[0016] FIG. **4**B depicts a top view of a portion of a block of memory cells according to aspects of the disclosure;

[0017] FIG. **4**C depicts a cross sectional view of a portion of a block of memory cells according to aspects of the disclosure;

[0018] FIG. **4**D is a detailed view of a portion of FIG. **4**C illustrating the tapering of the memory holes according to aspects of the disclosure;

[0019] FIG. **4**E depicts a view of the select gate layers and word line layers according to aspects of the disclosure;

[0020] FIG. **4**F is a cross sectional view of a vertical column of memory cells according to aspects of the disclosure;

[0021] FIG. **4**G is a schematic of a plurality of NAND strings according to aspects of the disclosure;

[0022] FIG. **5** depicts threshold voltage distributions according to aspects of the disclosure;

[0023] FIG. **6** is a table describing one example of an assignment of data values to data states according to aspects of the disclosure;

[0024] FIG. **7**A is a flow chart describing one embodiment of a process for programming according to aspects of the disclosure;

[0025] FIG. **7**B is a flow chart describing one embodiment of a process for programming data into memory cells connected to a common word line according to aspects of the disclosure;

[0026] FIG. **7**C depicts a word line voltage during programming and verify operations according to aspects of the disclosure;

[0027] FIG. **8** is a plot of voltages applied to typical word lines and slow word lines and the resulting potential of the typical word lines and slow word lines during a portion of one loop of a programming operation according to aspects of the disclosure;

[0028] FIG. **9** is a plot of voltages applied to typical word lines and slow word lines and the resulting potential of the typical word lines and slow word lines during a portion of one loop of a programming operation using the program voltage kick offset according to aspects of the disclosure;

[0029] FIG. **10** is a plot of voltages applied to typical word lines and slow word lines and the resulting potential of the typical word lines and slow word lines during a portion of one loop of a programming operation using the program voltage kick offset with a stepwise program kick voltage according to aspects of the disclosure; and

[0030] FIG. **11** illustrates steps of a method of operating a memory apparatus according to aspects of the disclosure.

[0031] To facilitate understanding, identical reference numerals have been used, where possible, to designate identical elements that are common to the figures. It is contemplated that elements disclosed in one embodiment may be beneficially utilized on other embodiments without specific recitation.

DETAILED DESCRIPTION

[0032] In the following description, details are set forth to provide an understanding of the present disclosure. In some instances, certain circuits, structures and techniques have not been described or shown in detail in order not to obscure the disclosure.

[0033] In general, the present disclosure relates to non-volatile memory apparatuses of the type well-suited for use in many applications. The non-volatile memory apparatus and associated methods of operation of this disclosure will be described in conjunction with one or more example embodiments. However, the specific example embodiments disclosed are merely provided to describe the inventive concepts, features, advantages and objectives with sufficient clarity to permit those skilled in this art to understand and practice the disclosure. Specifically, the example embodiments are provided so that this disclosure will be thorough, and will fully convey the scope to those who are skilled in the art. Numerous specific details are set forth such as examples of specific components, devices, and methods, to provide a thorough understanding of embodiments of the present disclosure. It will be apparent to those skilled in the art that specific details need not be employed, that example embodiments may be embodied in many different forms and that neither should be construed to limit the scope of the disclosure. In some example embodiments, well-known processes, well-known device structures, and well-known technologies are not described in detail.

[0034] Performance, including programming speed, is an important standard of measurement for non-volatile memory devices. Non-volatile memory devices are typically programmed by applying to the memory cells a series of voltage pulses that progressively increase in amplitude in a staircase-like waveform. To improve programming performance, the memory device can, for example, use device parameters that use a higher starting voltage level for the first voltage pulse, use larger step sizes for the pulse-to-pulse increase, or some combination of these and other techniques. However, if these parameters are increased too much for speed, this can eventually lead to over programming of memory cells, resulting in an amount of errors that exceeds the error correction capabilities of the memory system.

[0035] Although such over programming can occur in many non-volatile memory systems, some designs tend to be more prone to this problem. For example, in non-volatile memories having a three-dimensional NAND structure, the memory cells are formed along word lines in multiple layers extending down along "memory holes." Due to the process for forming such memories, these memory holes tend to taper, becoming narrower as the hole goes deeper into the structure. This and other factors can result in the word lines along different layers programming at different speeds, so that optimizing the programming parameters to have constant programming times across all the layers can be challenging. As a result, some word lines program overly fast, resulting in an uncorrectable amount of over-programming.

[0036] FIGS. **1-4**G describe one set of examples of a memory system that can be used to implement the technology proposed herein. FIG. **1** is a functional block diagram of an example memory device. The components depicted in FIG. **1** are electrical circuits. Memory device **100** includes one or more memory die **108**. Each memory die **108** includes a three-dimensional memory structure **126** of memory cells (such as, for example, a 3D array of memory cells), control circuitry **110**, and read/write/erase circuits **128**. In other embodiments, a two-dimensional array of memory cells can be used. Memory structure **126** is addressable by-word lines via a row decoder **124** and by bit lines via a column decoder **132**. The read/write/erase circuits **128** include multiple sense blocks **150** including Sense Block **1**, Sense Block **2**, . . . , Sense Block p (sensing circuitry) and allow a page of memory cells (connected to the same word line) to be read or programmed in parallel. In some systems, a controller **122** is included in the same memory device **100** as the one or more memory die **108**. However, in other systems, the controller can be separated from the memory die **108**. In some embodiments controller **122** will be on a different die than memory die **108**. In some embodiments, one controller **122** will communicate with multiple memory die **108**. In other embodiments, each memory die **108** has its own controller. Commands and data are transferred between the host **140** and controller **122** via a data bus **120**, and between controller **122** and the one or more memory die **108** via lines **118**. In one embodiment, memory die **108** includes a set of input and/or output (I/O) pins that connect to lines **118**.

[0037] Memory structure **126** may comprise one or more arrays of memory cells including a 3D array. The memory structure may comprise a monolithic three-dimensional memory structure in which multiple memory levels are formed above (and not in) a single substrate, such as a wafer, with no intervening substrates. The memory structure may comprise any type of non-volatile memory that is monolithically formed in one or more physical levels of arrays of memory cells having an active area disposed above a silicon substrate. The memory structure may be in a non-volatile memory device having circuitry associated with the operation of the memory cells, whether the associated circuitry is above or within the substrate. In one embodiment, memory structure **126** implements three dimensional NAND flash memory. Other embodiments include two dimensional NAND flash memory, two dimensional NOR flash memory, ReRAM cross-point memories, magnetoresistive memory (e.g., MRAM), phase change memory (e.g., PCRAM), and others.

[0038] Control circuitry **110** cooperates with the read/write/erase circuits **128** to perform memory operations (e.g., erase, program, read, and others) on memory structure **126**, and includes a state machine **112**, an on-chip address decoder **114**, and a power control module **116**. The state machine **112** provides die-level control of memory operations, such as programming different memory cells to different final targets for a common data state based on distance to an edge of a word line layer. In one embodiment, state machine **112** is programmable by the software. In other embodiments, state machine **112** does not use software and is completely implemented in hardware (e.g., electrical circuits). In one embodiment, control circuitry **110** includes registers, ROM fuses and other storage devices for storing default values such as base voltages and other parameters.

[0039] The on-chip address decoder **114** provides an address interface between addresses used by host **140** or controller **122** to the hardware address used by the decoders **124** and **132**. Power control module **116** controls the power and voltages supplied to the word lines and bit lines during memory operations. It can include drivers for word line layers (discussed below) in a 3D configuration, select transistors (e.g., SGS and SGD transistors, described below) and source lines. Power control module **116** may include charge pumps for creating voltages. The sense blocks include bit line drivers. An SGS transistor is a select gate transistor at a source end of a NAND string, and an SGD transistor is a select gate transistor at a drain end of a NAND string.

[0040] Any one or any combination of control circuitry **110**, state machine **112**, decoders **114/124/132**, power control module **116**, sense blocks **150**, read/write/erase circuits **128**, and/or controller **122** can be considered a control circuit that performs the functions described herein.

[0041] The (on-chip or off-chip) controller **122** (which in one embodiment is an electrical circuit) may comprise one or more processors **122***c*, ROM **122***a*, RAM **122***b*, and Memory Interface **122***d*, all of which are interconnected. One or more processors **122***c* is one example of a control circuit. Other embodiments can use state machines or other custom circuits designed to perform one or more functions. The storage devices (ROM **122***a*, RAM **122***b*) comprises code such as a set of instructions, and the processor **122***c* is operable to execute the set of instructions to provide the functionality described below related to programming. Alternatively, or additionally, processor **122***c* can access code from a storage device in the memory structure, such as a reserved area of memory cells connected to one or more word lines. Memory interface **122***d*, in communication with ROM **122***a*, RAM **122***b* and processor **122***c*, is an electrical circuit (electrical interface) that provides an electrical interface between controller **122** and one or more memory die **108**. The controller can maintain various operating parameters in RAM **122***b*. For example, memory interface **122***d* can change the format or timing of signals, provide a buffer, isolate from surges, latch I/O, etc. Processor **122***c* can issue commands to control circuitry **110** (or any other component of memory die **108**) via Memory Interface **122***d*.

[0042] Multiple memory elements in memory structure **126** may be configured so that they are connected in series or so that each element is individually accessible. By way of non-limiting example, flash memory devices in a NAND configuration (NAND flash memory) typically contain memory elements connected in series. A NAND string is an example of a set of series-connected memory cells and select gate transistors.

[0043] A NAND flash memory array may be configured so that the array is composed of multiple NAND strings of which a NAND string is composed of multiple memory cells sharing a single bit line and accessed as a group. Alternatively, memory elements may be configured so that each element is individually accessible, e.g., a NOR memory array. NAND and NOR memory configurations are exemplary, and memory cells may be otherwise configured.

[0044] The memory cells may be arranged in the single memory device level in an ordered array, such as in a plurality of rows and/or columns. However, the memory elements may be arrayed in non-regular or non-orthogonal configurations, or in structures not considered arrays.

[0045] A three-dimensional memory array is arranged so that memory cells occupy multiple planes or multiple memory device levels, thereby forming a structure in three dimensions (i.e., in the x, y and z directions, where the z direction is substantially perpendicular and the x and y directions are substantially parallel to the major surface of the substrate).

[0046] As a non-limiting example, a three-dimensional memory structure may be vertically arranged as a stack of multiple two-dimensional memory device levels. As another non-limiting example, a three-dimensional memory array may be arranged as multiple vertical columns (e.g., columns extending substantially perpendicular to the major surface of the substrate, i.e., in the y direction) of memory holes, with each column having multiple memory cells. The memory holes may be arranged in a two-dimensional configuration, e.g., in an x-y plane, resulting in a three-dimensional arrangement of memory cells, with memory cells on multiple vertically stacked memory planes. Other configurations of memory elements in three dimensions can also constitute a three-dimensional memory array.

[0047] By way of non-limiting example, in a three-dimensional NAND memory array, the memory elements may be coupled together to form vertical NAND strings that traverse across multiple horizontal levels. Other three-dimensional configurations can be envisioned wherein some NAND strings contain memory elements in a single memory level while other strings contain memory elements which span through multiple memory levels. Three-dimensional memory arrays may also be designed in a NOR configuration and in a ReRAM configuration.

[0048] A person of ordinary skill in the art will recognize that the technology described herein is not limited to a single specific memory structure, but covers many relevant memory structures within the spirit and scope of the technology as described herein and as understood by one of ordinary skill in the art.

[0049] FIG. **2** is a block diagram of example memory system **100**, depicting more details of one embodiment of controller **122**. As used herein, a flash memory controller is a device that manages data stored on flash memory and communicates with a host, such as a computer or electronic device. A flash memory controller can have various functionality in addition to the specific functionality described herein. For example, the flash memory controller can format the flash memory to ensure the memory is operating properly, map out bad flash memory cells, and allocate spare memory cells to be substituted for future failed cells. Some part of the spare cells can be used to hold firmware to operate the flash memory controller and implement other features. In operation, when a host needs to read data from or write data to the flash memory, it will communicate with the flash memory controller. If the host provides a logical address to which data is to be read/written, the flash memory controller can convert the logical address received from the host to a physical address in the flash memory. (Alternatively, the host can provide the physical address). The flash memory controller can also perform various memory management functions, such as, but not limited to, wear leveling (distributing programs to avoid wearing out specific blocks of memory that would otherwise be repeatedly written to) and garbage collection (after a block is full, moving only the valid pages of data to a new block, so the full block can be erased and reused).

[0050] The interface between controller **122** and non-volatile memory die **108** may be any suitable flash interface, such as Toggle Mode 200, 400, or 800. In one embodiment, memory system **100** may be a card based system, such as a secure digital (SD) or a micro secure digital (micro-SD) card. In an alternate embodiment, memory system **100** may be part of an embedded memory system. For example, the flash memory may be embedded within the host. In other example, memory system **100** can be in the form of a solid-state drive (SSD) drive.

[0051] In some embodiments, non-volatile memory system **100** includes a single channel between controller **122** and non-volatile memory die **108**, the subject matter described herein is not limited to having a single memory channel. For example, in some memory system architectures, 2, 4, 8 or more channels may exist between the controller and the memory die, depending on controller capabilities. In any of the embodiments described herein, more than a single channel may exist between the controller and the memory die, even if a single channel is shown in the drawings.

[0052] As depicted in FIG. **2**, controller **112** includes a front-end module **208** that interfaces with a host, a back-end module **210** that interfaces with the one or more non-volatile memory die **108**, and various other modules that perform functions which will now be described in detail.

[0053] The components of controller **122** depicted in FIG. **2** may take the form of a packaged functional hardware unit (e.g., an electrical circuit) designed for use with other components, a portion of a program code (e.g., software or firmware) executable by a (micro) processor or processing circuitry that usually performs a particular function of related functions, or a self-contained hardware or software component that interfaces with a larger system, for example. For example, each module may include an application specific integrated circuit (ASIC), a Field Programmable Gate Array (FPGA), a circuit, a digital logic circuit, an analog circuit, a combination of discrete circuits, gates, or any other type of hardware or combination thereof. Alternatively, or in addition, each module may include software stored in a processor readable device (e.g., memory) to program a processor for controller **122** to perform the functions described herein. The architecture depicted in FIG. **2** is one example implementation that may (or may not) use the components of controller **122** depicted in FIG. **1** (i.e. RAM, ROM, processor, interface).

[0054] Referring again to modules of the controller **122**, a buffer manager/bus control **214** manages buffers in random access memory (RAM) **216** and controls the internal bus arbitration of controller **122**. A read only memory (ROM) **218** stores system boot code. Although illustrated in FIG. **2** as located separately from the controller **122**, in other embodiments one or both of the RAM **216** and ROM **218** may be located within the controller. In yet other embodiments, portions of RAM and ROM may be located both within the controller **122** and outside the controller. Further, in some implementations, the controller **122**, RAM **216**, and ROM **218** may be located on separate semiconductor die.

[0055] Front end module **208** includes a host interface **220** and a physical layer interface (PHY) **222** that provide the electrical interface with the host or next level storage controller. The choice of the type of host interface **220** can depend on the type of memory being used. Examples of host interfaces **220** include, but are not limited to, SATA, SATA Express, SAS, Fibre Channel, USB, PCIe, and NVMe. The host interface **220** typically facilitates transfer for data, control signals, and timing signals.

[0056] Back end module **210** includes an error correction code (ECC) engine **224** that encodes the data bytes received from the host, and decodes and error corrects the data bytes read from the non-volatile memory. A command sequencer **226** generates command sequences, such as program and erase command sequences, to be transmitted to non-volatile memory die **108**. A RAID (Redundant Array of Independent Dies) module **228** manages generation of RAID parity and recovery of failed data. The RAID parity may be used as an additional level of integrity protection for the data being written into the non-volatile memory system **100**. In some cases, the RAID module **228** may be a part of the ECC engine **224**. Note that the RAID parity may be added as an extra die or dies as implied by the common name, but it may also be added within the existing die, e.g. as an extra plane, or extra block, or extra WLs within a block. A memory interface **230** provides the command sequences to non-volatile memory die **108** and receives status information from non-volatile memory die **108**. In one embodiment, memory interface **230** may be a double data rate (DDR) interface, such as a Toggle Mode 200, 400, or 800 interface. A flash control layer **232** controls the overall operation of back end module **210**.

[0057] One embodiment includes a programming manager **236**, which can be used to manage (in conjunction with the circuits on the memory die) the programming of memory cells. The programming manager **236** can also manage the data relocation operations or other remedial actions when a word line is found defective due to programming too fast, as discussed further below. Programming manager **236** can be an electrical circuit, a set of one or more software modules, or a combination of a circuit and software.

[0058] Additional components of system **100** illustrated in FIG. **2** include media management layer **238**, which performs wear leveling of memory cells of non-volatile memory die **108**. System **100** also includes other discrete components **240**, such as external electrical interfaces, external RAM, resistors, capacitors, or other components that may interface with controller **122**. In alternative embodiments, one or more of the physical layer interface **222**, RAID module **228**, media management layer **238** and buffer management/bus controller **214** are optional components that are not necessary in the controller **122**.

[0059] The Flash Translation Layer (FTL) or Media Management Layer (MML) **238** may be integrated as part of the flash management that may handle flash errors and interfacing with the host. In particular, MML may be a module in flash management and may be responsible for the internals of NAND management. In particular, the MML **238** may include an algorithm in the memory device firmware which translates writes from the host into programs to the flash memory **126** of die **108**. The MML **238** may be needed because: 1) the flash memory may have limited endurance; 2) the flash memory **126** may only be written in multiples of pages; and/or 3) the flash memory **126** may not be written unless it is erased as a block. The MML **238** understands these potential limitations of the flash memory **126** which may not be visible to the host. Accordingly, the MML **238** attempts to translate the writes from host into programs into the flash memory **126**. As described below, erratic bits may be identified and recorded using the MML **238**. This recording of erratic bits can be used for evaluating the health of blocks and/or word lines (the word line unit of the memory cells on the word lines).

[0060] Controller **122** may interface with one or more memory dies **108**. In one embodiment, controller **122** and multiple memory dies (together comprising non-volatile storage system **100**) implement a solid-state drive (SSD), which can emulate, replace or be used instead of a hard disk drive inside a host, as a NAS device, in a laptop, in a tablet, in a server, etc. Additionally, the SSD need not be made to work as a hard drive.

[0061] FIG. **3** is a perspective view of a portion of one example embodiment of a monolithic three-dimensional memory structure **126**, which includes a plurality memory cells. For example, FIG. **3** shows a portion of one block of memory. The structure depicted includes a set of bit lines BL positioned above a stack of alternating dielectric layers and conductive layers. For example purposes, one of the dielectric layers is marked as D and one of the conductive layers (also called word line layers) is marked as W. The number of alternating dielectric layers and conductive layers can vary based on specific implementation requirements. One set of embodiments includes between 108-216 alternating dielectric layers and conductive layers, for example, 96 data word line layers, 8 select layers, 4 dummy word line layers and 108 dielectric layers. More or less than 108-216 layers can also be used. As will be explained below, the alternating dielectric layers and conductive layers are divided into four "fingers" by local interconnects LI (isolation areas). FIG. **3** only shows two fingers and two local interconnects LI. Below the alternating dielectric layers and word line layers is a source line layer SL. Memory holes are formed in the stack of alternating dielectric layers and conductive layers. For example, one of the memory holes is marked as MH. Note that in FIG. **3**, the dielectric layers are depicted as see-through so that the reader can see the memory holes positioned in the stack of alternating dielectric layers and conductive layers. In one embodiment, NAND strings are formed by filling the memory hole with materials including a charge-trapping layer to create a vertical column of memory cells. Each memory cell can store one or more bits of data. More details of the three-dimensional monolithic memory structure **126** is provided below with respect to FIG. **4A**-**4G**.

[0062] FIG. **4A** is a block diagram explaining one example organization of memory structure **126**, which is divided into two planes **302** and **304**. Each plane is then divided into M blocks. In one example, each plane has about 2000 blocks. However, different numbers of blocks and planes can also be used. In one embodiment, for two plane memory, the block IDs are usually such that even blocks belong to one plane and odd blocks belong to another plane; therefore, plane **302** includes block **0**, **2**, **4**, **6**, . . . and plane **304** includes blocks **1**, **3**, **5**, **7**, ... In on embodiment, a block of memory cells is a unit of erase. That is, all memory cells of a block are erased together. In other embodiments, memory cells can be grouped into blocks for other reasons, such as to organize the memory structure **126** to enable the signaling and selection circuits.

[0063] FIGS. **4B**-**4G** depict an example 3D NAND structure. FIG. **4B** is a block diagram depicting a top view of a portion of one block from memory structure **126**. The portion of the block depicted in FIG. **4B** corresponds to portion **306** in block **2** of FIG. **4A**. As can be seen from FIG. **4B**, the block depicted in FIG. **4B** extends in the direction of **332**. In one embodiment, the memory array will have 60 layers. Other embodiments have less than or more than 60 layers. However, FIG. **4B** only shows the top layer.

[0064] FIG. **4B** depicts a plurality of circles that represent the memory holes. Each of the vertical columns of the memory holes include multiple select transistors and multiple memory cells. In one embodiment, each memory hole implements a NAND string and, therefore, can be referred to as a memory column or memory hole. A memory column can implement other types of memory in addition to NAND. FIG. **4B** depicts memory holes **422**, **432**, **442** and **452**. Memory hole **422** implements NAND string **482**. Memory hole **432** implements NAND string **484**. Memory hole **442** implements NAND string **486**. Memory hole **452** implements NAND string **488**. More details of the vertical columns of memory holes are provided below. Since the block depicted in FIG. **4B** extends in the direction of arrow **330** and in the direction of arrow **332**, the block includes more memory holes than depicted in FIG. **4B**

[0065] FIG. **4B** also depicts a set of bit lines **415**, including bit lines **411**, **412**, **413**, **414**, . . . **419**. FIG. **4B** shows twenty-four-bit lines because only a portion of the block is depicted. It is contemplated that more than twenty-four-bit lines connected to memory holes of the block. Each of the circles representing memory holes has an "x" to indicate its connection to one bit line. For example, bit line **414** is connected to memory holes **422**, **432**, **442** and **452**.

[0066] The block depicted in FIG. **4B** includes a set of isolation areas **402**, **404**, **406**, **408** and **410** that serve to divide each layer of the block into four regions; for example, the top layer depicted in FIG. **4B** is divided into regions **420**, **430**, **440** and **450**, which are referred to as fingers. In the layers of the block that implement memory cells, the four regions are referred to as word line fingers that are separated by the isolation areas (also serving as local interconnects). In one embodiment, the word line fingers on a common level of a block connect together at the end of the block to form a single word line. In another embodiment, the word line fingers on the same level are not connected together. In one example implementation, a bit line only connects to one memory hole in each of regions **420**, **430**, **440** and **450**. In that implementation, each block has sixteen rows of active columns and each bit line connects to four rows in each block. In one embodiment, all of four rows connected to a common bit line are connected to the same word line (via different word line fingers on the same level that are connected together); therefore, the system uses the source side selection lines and the drain side selection lines to choose one (or another subset) of the four to be subjected to a memory operation (program, verify, read, and/or erase).

[0067] Isolation areas **402**, **404**, **406**, **408** and **410** also connect the various layers to a source line below the vertical columns of the memory holes. In one embodiment, isolation areas **402**, **404**, **406**, **408** and **410** are filled with a layer of SiO_2 (blocking) and a layer of polysilicon (source line connection).

[0068] Although FIG. **4**B shows each region having four rows of memory holes, four regions and sixteen rows of memory holes in a block, those exact numbers are an example implementation. Other embodiments may include more or less regions per block, more or less rows of memory holes per region and more or less rows of memory holes per block.

[0069] FIG. **4**B also shows the memory holes being staggered. In other embodiments, different patterns of staggering can be used. In some embodiments, the memory holes are not staggered.

[0070] FIG. **4**C depicts a portion of an embodiment of three-dimensional memory structure **126** showing a cross-sectional view along line AA of FIG. **4**B. This cross-sectional view cuts through memory holes **432** and **434** and region **430** (see FIG. **4**B). The structure of FIG. **4**C includes four drain side select layers SGD**0**, SGD**1**, SGD**2** and SGD**3**; four source side select layers SGS**0**, SGS**1**, SGS**2** and SGS**3**; four dummy word line layers DD**0**, DD**1**, DS**0** and DS**1**; and forty-eight data word line layers WLL**0**-WLL**47** for connecting to data memory cells as word line units of the memory cells connected to the word lines. Other embodiments can implement more or less than four drain side select layers, more or less than four source side select layers, more or less than four dummy word line layers, and more or less than forty-eight word line layers (e.g., 96 word line layers). Memory holes **432** and **434** are depicted protruding through the drain side select layers, source side select layers, dummy word line layers and word line layers. In one embodiment, each memory hole comprises a NAND string. For example, memory hole **432** comprises NAND string **484**. The NAND string of memory hole **432** has a source end at a bottom of the stack and a drain end at a top of the stack. As in agreement with FIG. **4**B, FIG. **4**C show memory hole **432** connected to Bit Line **414** via connector **415**. Isolation areas **404** and **406** are also depicted. Below the memory holes and the layers listed below, and over the underlying substrate, is source line SL and well region P-Well **101**. A block of memory cells will share a common well region and in an erase operation, the erase voltage Verase is applied to the P-Well **101** and, through the source line SL, to channel region of the memory holes.

[0071] For ease of reference, drain side select layers SGD**0**, SGD**1**, SGD**2** and SGD**3**; source side select layers SGS**0**, SGS**1**, SGS**2** and SGS**3**; dummy word line layers DD**0**, DD**1**, DS**0** and DS**1**; and word line layers WLL**0**-WLL**47** collectively are referred to as the conductive layers. In one embodiment, the conductive layers are made from a combination of TiN and Tungsten. In other embodiments, other materials can be used to form the conductive layers, such as doped polysilicon, metal such as Tungsten or metal silicide. In some embodiments, different conductive layers can be formed from different materials. Between conductive layers are dielectric layers DL**0**-DL**59**. For example, dielectric layers DL**51** is above word line layer WLL**45** and below word line layer WLL**46**. In one embodiment, the dielectric layers are made from SiO_2. In other embodiments, other dielectric materials can be used to form the dielectric layers.

[0072] The non-volatile memory cells are formed along memory holes which extend through alternating conductive and dielectric layers in the stack. In one embodiment, the memory cells are arranged in NAND strings. The word line layer WLL**0**-WLL**47** connect to memory cells (also called data memory cells) to form word line units. Dummy word line layers DD**0**, DD**1**, DS**0** and DS**1** connect to dummy memory cells. A dummy memory cell does not store host data (data provided from the host, such as data from a user of the host), while a data memory cell is eligible to store host data. Drain side select layers SGD**0**, SGD**1**, SGD**2** and SGD**3** are used to electrically connect and disconnect NAND strings from bit lines. Source side select layers SGS**0**, SGS**1**, SGS**2** and SGS**3** are used to electrically connect and disconnect NAND strings from the source line SL.

[0073] Although the FIG. **4**C shows the regions **432** and **434** as being cylindrical, extending upwards in the z-direction with parallel sides, in an actual device the process of forming memory holes typically results in the memory holes tapering as they go deeper into the structure. FIG. **4**D is a detail of FIG. **4**C illustrating the tapering of the memory holes, but in an exaggerated form for purposes of discussion.

[0074] As shown in FIG. **4**D, the column **432** narrows as it descends, being wider where it passes through word line layer WLL**27** than where it passes through word line WLL**22**. Although this tapering is exaggerated in FIG. **4**D, when looking at the entire stack of FIG. **4**C the difference in memory hole diameter can vary significantly between the bottom of the stack and the top. This physical difference in structure on the different word line layers can lead to them acting differently. For example, due to the tapering of the memory holes, lower word lines tend to program faster than higher word lines that have a larger memory hole diameter. To account for these sorts of variations, the word lines can be split into word line groups and treated differently. FIG. **4**C shows the word lines split up into M word line groups, where the number of such groups is a design decision based on balancing the complexity of more word line groups against the accuracy of accounting for variations in memory hole dimension.

[0075] Applying the grouping of word lines to compensate for the dimensional variation of the memory holes to the case of programming, parameters can be used to enable the system to define which word line groups could use an additional program voltage offsets on top of the default programming voltage. These parameters can specify higher starting voltages, bigger step sizes and/or other variations in the programming waveform in order to meet performance requirements.

[0076] FIG. **4**E depicts a logical representation of the conductive layers (SGD**0**, SGD**1**, SGD**2**, SGD**3**, SGS**0**, SGS**1**, SGS**2**, SGS**3**, DD**0**, DD**1**, DS**0**, DS**1**, and WLL**0**-WLL**47**) for the block that is partially depicted in FIG. **4**C. As mentioned above with respect to FIG. **4**B, in one embodiment isolation areas **402**, **404**, **406**, **408** and **410** break up each conductive layer into four regions or fingers. For example, word line layer WLL**31** is divided into regions **460**, **462**, **464** and **466**. For word line layers (WLL**0**-WLL**31**), the regions are referred to as word line fingers; for example, word line layer WLL**46** is divided into word line fingers **460**, **462**, **464** and **466**. In one embodiment, the four word line fingers on a same level are connected together. In another embodiment, each word line finger operates as a separate word line.

[0077] Drain side select gate layer SGD**0** (the top layer) is also divided into regions **420**, **430**, **440** and **450**, also known as fingers or select line fingers. In one embodiment, the four select line fingers on a same level are connected together. In another embodiment, each select line finger operates as a separate word line.

[0078] FIG. **4**F depicts a cross sectional view of region **429** of FIG. **4**C that includes a portion of vertical column of the memory hole **432**. In one embodiment, the vertical columns of the memory holes are round and include four layers; however, in other embodiments more or less than four layers can be included and other shapes can be used. In one embodiment, vertical column of the memory hole **432** includes an inner core layer **470** that is made of a dielectric, such as SiO_2. Other materials can also be used. Surrounding inner core **470** is polysilicon channel **471**. Materials other than polysilicon can also be used. Note that it is the channel **471** that connects to the bit line. Surrounding channel **471** is a tunneling dielectric **472**. In one embodiment, tunneling dielectric **472** has an ONO structure. Surrounding tunneling dielectric **472** is charge trapping layer **473**, such as (for example) Silicon Nitride. Other memory materials and structures can also be used. The technology described herein is not limited to any particular material or structure.

[0079] FIG. **4**F depicts dielectric layers DLL**49**, DLL**50**, DLL**51**, DLL**52** and DLL**53**, as well as word line layers WLL**43**, WLL**44**, WLL**45**, WLL**46**, and WLL**47**. Each of the word line layers includes a word line region **476** surrounded by an aluminum oxide layer **477**, which is surrounded by a blocking oxide (SiO_2) layer **478**. The physical interaction of the word line layers with the vertical column forms the memory cells. Thus, a memory cell, in one embodiment, comprises channel **471**, tunneling dielectric **472**, charge trapping layer **473**, blocking oxide layer **478**, aluminum oxide layer **477** and word line region **476**. For example, word line layer WLL**47** and a portion of memory hole **432** comprise a memory cell MC**1**. Word line layer WLL**46** and a portion of memory hole **432** comprise a memory cell MC**2**. Word line layer WLL**45** and a portion of memory hole **432** comprise a memory cell MC**3**. Word line layer WLL**44** and a portion of memory hole **432** comprise a memory cell MC**4**. Word line layer WLL**43** and a portion of memory hole **432** comprise a memory cell MC**5**. In other architectures, a memory cell may have a different structure; however, the memory cell would still be the storage unit.

[0080] When a memory cell is programmed, electrons are stored in a portion of the charge trapping layer **473** which is associated with the memory cell. These electrons are drawn into the charge trapping layer **473** from the channel **471**, through the tunneling dielectric **472**, in response to an appropriate voltage on word line region **476**. The threshold voltage (Vth) of a memory cell is increased in proportion to the amount of stored charge. In one embodiment, the programming is achieved through Fowler-Nordheim tunneling of the electrons into the charge trapping layer. During an erase operation, the electrons return to the channel or holes are injected into the charge trapping layer to recombine with electrons. In one embodiment, erasing is achieved using hole injection into the charge trapping layer via a physical mechanism such as gate induced drain leakage (GIDL).

[0081] FIG. **4**G is a circuit diagram of a plurality of NAND strings according to the embodiments of FIGS. **3**-**4**F. FIG. **4**G shows physical word lines WLL**0**-WLL**47** running across the entire block. The structure of FIG. **4**G corresponds to portion **306** in Block **2** of FIGS. **4**A-F, including bit lines **411**, **412**, **413**, **414**, . . . **419**. Within the block, each bit line connected to four NAND strings. Drain side selection lines SGD**0**, SGD**1**, SGD**2** and SGD**3** are used to determine which of the four NAND strings connect to the associated bit line. The block can also be thought of as divided into four fingers finger **0**, finger **1**, finger **2** and finger **3**. Finger **0** corresponds to those vertical NAND strings controlled by SGD**0** and SGS**0**, finger **1** corresponds to those vertical NAND strings controlled by SGD**1** and SGS**1**, finger **2** corresponds to those vertical NAND strings controlled by SGD**2** and SGS**2**, and finger **3** corresponds to those vertical NAND strings controlled by SGD**3** and SGS**3**. The example of FIG. **4**G again shows the separation of word lines into groups

[0082] Although the example memory system of FIGS. **4**A-**4**G is a three-dimensional memory structure that includes vertical NAND strings with charge-trapping material, other (2D and 3D) memory structures can also be used with the technology described herein. For example, floating gate memories (e.g., NAND-type and NOR-type flash memory ReRAM memories, magnetoresistive memory (e.g., MRAM), and phase change memory (e.g., PCRAM) can also be used.

[0083] One example of a ReRAM memory includes reversible resistance-switching elements arranged in cross point arrays accessed by X lines and Y lines (e.g., word lines and bit lines). In another embodiment, the memory cells may include conductive bridge memory elements. A conductive bridge memory element may also be referred to as a programmable metallization cell. A conductive bridge memory element may be used as a state change element based on the physical relocation of ions within a solid electrolyte. In some cases, a conductive bridge memory element may include two solid metal electrodes, one relatively inert (e.g., tungsten) and the other electrochemically active (e.g., silver or copper), with a thin film of the solid electrolyte between the two electrodes. As temperature increases, the mobility of the ions also increases causing the programming threshold for the conductive bridge memory cell to decrease. Thus, the conductive bridge memory element may have a wide range of programming thresholds over temperature.

[0084] Magnetoresistive memory (MRAM) stores data by magnetic storage elements. The elements are formed from two ferromagnetic plates, each of which can hold a magnetization, separated by a thin insulating layer. One of the two plates is a permanent magnet set to a particular polarity; the other plate's magnetization can be changed to match that of an external field to store memory. This configuration is known as a spin valve and is the simplest structure for an MRAM bit. A memory device is built from a grid of such memory cells. In one embodiment for programming, each memory cell lies between a pair of write lines arranged at right angles to each other, parallel to the cell, one above and one below the cell. When current is passed through them, an induced magnetic field is created.

[0085] Phase change memory (PCRAM) exploits the unique behavior of chalcogenide glass. One embodiment uses a GeTe—Sb2Te3 super lattice to achieve non-thermal phase changes by simply changing the co-ordination state of the Germanium atoms with a laser pulse (or light pulse from another source). Therefore, the doses of programming are laser pulses. The memory cells can be inhibited by blocking the memory cells from receiving the light. Note that the use of "pulse" in this document does not require a square pulse, but includes a (continuous or non-continuous) vibration or burst of sound, current, voltage light, or other wave.

[0086] The memory systems discussed above can be erased, programmed and read. At the end of a successful programming process (with verification), the threshold voltages of the memory cells should be within one or more distributions of threshold voltages for programmed memory cells or within a distribution of threshold voltages for erased memory cells, as appropriate. FIG. **5** illustrates example threshold voltage distributions for the memory cell array when each memory cell stores three bits of data. Other embodiments, however, may use other data capacities per memory cell (e.g., such as one, two, four, or five bits of data per memory cell). FIG. **5** shows eight threshold voltage distributions, corresponding to eight data states. The first threshold voltage distribution (data state) S**0** represents memory cells that are erased. The other seven threshold voltage distributions (data states) S**1**-S**7** represent memory cells that are programmed and, therefore, are also called programmed states. Each threshold voltage distribution (data state) corresponds to predetermined values for the set of data bits. The specific relationship between the data programmed into the memory cell and the threshold voltage levels of the cell depends upon the data encoding scheme adopted for the cells. In one embodiment, data values are assigned to the threshold voltage ranges using a Gray code assignment so that if the threshold voltage of a memory erroneously shifts to its neighboring physical state, only one bit will be affected.

[0087] FIG. **5** also shows seven read reference voltages, Vr**1**, Vr**2**, Vr**3**, Vr**4**, Vr**5**, Vr**6**, and Vr**7**, for reading data from memory cells. By testing (e.g., performing sense operations) whether the threshold voltage of a given memory cell is above or below the seven read reference voltages, the system can determine what data state (i.e., S**0**, S**1**, S**2**, S**3**, . . .) a memory cell is in.

[0088] FIG. **5** also shows seven verify reference voltages, Vv**1**, Vv**2**, Vv**3**, Vv**4**, Vv**5**, Vv**6**, and Vv**7**. When programming memory cells to data state S**1**, the system will test whether those memory cells have a threshold voltage greater than or equal to Vv**1**. When programming memory cells to data state S**2**, the system will test whether the memory cells have threshold voltages greater than or equal to Vv**2**. When programming memory cells to data state S**3**, the system will determine whether memory cells have their threshold voltage greater than or equal to Vv**3**. When programming memory cells to data state S**4**, the system will test whether those memory cells have a threshold voltage greater than or equal to Vv**4**. When programming memory cells to data state S**5**, the system will test whether those memory cells have a threshold voltage greater than or equal to Vv**5**. When programming memory cells to data state S**6**, the system will test whether those memory cells have a threshold voltage greater than or equal to Vv**6**. When programming memory cells to data state S**7**, the system will test whether those memory cells have a threshold voltage greater than or equal to Vv**7**.

[0089] In one embodiment, known as full sequence programming, memory cells can be programmed from the erased data state S**0** directly to any of the programmed data states S**1**-S**7**. For example, a population of memory cells to be programmed may first be erased so that all memory cells in the population are in erased data state S**0**. Then, a programming process is used to program memory cells directly into data states S**1**, S**2**, S**3**, S**4**, S**5**, S**6**, and/or S**7**. For example, while some memory cells are being programmed from data state S**0** to data state S**1**, other memory cells are being programmed from data state S**0** to data state S**2** and/or from data state S**0** to data state S**3**, and so on. The arrows of FIG. **6** represent the full sequence programming. The technology described herein can also be used with other types of programming in addition to full sequence programming (including, but not limited to, multiple stage/phase programming). In some embodiments, data states S**1**-S**7** can overlap, with controller **122** relying on ECC to identify the correct data being stored.

[0090] FIG. **6** is a table describing one example of an assignment of data values to data states. In the table of FIGS. **6**, S**0**-111. S**1**=110, S**2**=100, S**3**=000, S**4**-010, S**5**=011, S**6**-001 and S**7**=101. Other encodings of data can also be used. No particular data encoding is required by the technology disclosed herein. In one embodiment, when a block is subjected to an erase operation, all memory cells are moved to data state S**0**, the erased state. In the embodiment of FIG. **6**, all bits stored in a memory cell are **1** when the memory cells are erased (e.g., in data state S**0**).

[0091] FIG. **7**A is a flowchart describing one embodiment of a process for programming that is performed by controller **122**. In some embodiments, rather than have a dedicated controller, the host can perform the functions of the controller. In step **702**, controller **122** sends instructions to one or more memory die **108** to program data. In step **704**, controller **122** sends one or more addresses to one or more memory die **108**. The one or more logical addresses indicate where to program the data. In step **706**, controller **122** sends the data to be programmed to the one or more memory die **108**. In step **708**, controller **122** receives a result of the programming from the one or more memory die **108**. Example results include that the data was programmed successfully, an indication that the programming operation failed, and indication that the data was programmed but at a different location, or other result. In step **710**, in response to the result received in step **708**, controller **122** updates the system information that it maintains. In one embodiment, the system maintains tables of data that indicate status information for each block. This information may include a mapping of logical addresses to physical addresses, which blocks/word lines are open/closed (or partially opened/closed), which blocks/word lines are bad, etc.

[0092] In some embodiments, before step **702**, controller **122** would receive host data and an instruction to program from the host, and the controller would run the ECC engine **224** to create code words from the host data, as known in the art and described in more detail below. These code words are the data transmitted in step **706**. controller can also scramble the data to achieve wear leveling with respect to the memory cells.

[0093] FIG. **7**B is a flowchart describing one embodiment of a process for programming. The process of FIG. **7**B is performed by the memory die in response to the steps of FIG. **7**A (i.e., in response to the instructions, data and addresses from controller **122**). In one example embodiment, the process of FIG. **7**B is performed on memory die **108** using the one or more control circuits discussed above, at the direction of state machine **112**. The process of FIG. **7**B can also be used to implement the full sequence programming discussed above. Additionally, the process of FIG. 7B can be used to implement each phase of a multi-phase programming process.

[0094] Typically, the program voltage applied to the control gates (via a selected common word line) during a program operation is applied as a series of program voltage pulses. Between program voltage pulses are a set of verify voltage pulses to perform verification. In many implementations, the magnitude of the program voltage pulses is increased with each successive voltage pulse by a predetermined step size. In step **770** of FIG. **7**B, the programming voltage (Vpgm) is initialized to the starting magnitude (e.g., "12-16V or another suitable level) and a program counter PC maintained by state machine **112** is initialized at 1. In step **772**, a voltage pulse of the program signal Vpgm is applied to the selected word line (the word line selected for programming). In one embodiment, the group of memory cells being programmed concurrently are all connected to the same common word line (the selected word line). The unselected word lines receive one or more boosting voltages (e.g., "7-11 volts) to perform boosting schemes known in the art. If a memory cell should be programmed, then the corresponding bit line is grounded. On the other hand, if the memory cell should remain at its current threshold voltage, then the corresponding bit line is connected to Vdd to inhibit programming. In step **772**, the program voltage pulse is concurrently applied to all memory cells connected to the selected word line so that all of the memory cells connected to the selected word line are programmed concurrently. That is, they are programmed at the same time or during overlapping times (both of which are considered concurrent). In this manner, all of the memory cells connected to the selected word line will concurrently have their threshold voltage change, unless they have been locked out from programming.

[0095] In step **774**, the appropriate memory cells are verified using the appropriate set of verify reference voltages to perform one or more verify operations. In one embodiment, the verification process is performed by applying the testing whether the threshold voltages of the memory cells selected for programming have reached the appropriate verify reference voltage.

[0096] In step **776**, it is determined whether all the memory cells have reached their target threshold voltages (pass). If so, the programming process is complete and successful because all selected memory cells were programmed and verified to their target states. A status of "PASS" is reported in step **778**. If, in **776**, it is determined that not all of the memory cells have reached their target threshold voltages (fail), then the programming process continues to step **780**.

[0097] In step **780**, the system counts the number of memory cells that have not yet reached their respective target threshold voltage distribution. That is, the system counts the number of memory cells that have, so far, failed the verify process. This counting can be done by the state machine, the controller, or other logic. In one implementation, each of the sense blocks will store the status (pass/fail) of their respective cells. In one embodiment, there is one total count, which reflects the total number of memory cells currently being programmed that have failed the last verify step. In another embodiment, separate counts are kept for each data state.

[0098] In step **782**, it is determined whether the count from step **780** is less than or equal to a predetermined limit. In one embodiment, the predetermined limit is the number of bits that can be corrected by error correction codes (ECC) during a read process for the page of memory cells. If the number of failed memory cells is less than or equal to the predetermined limit, than the programming process can stop and a status of "PASS" is reported in step **778**. In this situation, enough memory cells programmed correctly such that the few remaining memory cells that have not been completely programmed can be corrected using ECC during the read process. In some embodiments, step **780** will count the number of failed cells for each sector, each target data state or other unit, and those counts will individually or collectively be compared to a threshold in step **782**.

[0099] In another embodiment, the predetermined limit can be less than the number of bits that can be corrected by ECC during a read process to allow for future errors. When programming less than all of the memory cells for a page, or comparing a count for only one data state (or less than all states), than the predetermined limit can be a portion (pro-rata or not pro-rata) of the number of bits that can be corrected by ECC during a read process for the page of memory cells. In some embodiments, the limit is not predetermined. Instead, it changes based on the number of errors already counted for the page, the number of program-erase cycles performed or other criteria.

[0100] If number of failed memory cells is not less than the predetermined limit, than the programming process continues at step **784** and the program counter PC is checked against the program limit value (PL). Examples of program limit values include 12, 20 and 30; however, other values can be used. If the program counter PC is not less than the program limit value PL, then the program process is considered to have failed and a status of FAIL is reported in step **788**. This is one example of a program fault. If the program counter PC is less than the program limit value PL, then the process continues at step **786** during which time the Program Counter PC is incremented by 1 and the program voltage Vpgm is stepped up to the next magnitude. For example, the next voltage pulse will have a magnitude greater than the previous voltage pulse by a step size (e.g., a step size of 0.1-0.5 volts). After step **786**, the process loops back to step **772** and another voltage pulse is applied to the selected word line so that another iteration (steps **772**-**786**) of the programming process of FIG. **7**B is performed.

[0101] In general, during verify operations and read operations, the selected word line is connected to a voltage (one example of a reference signal), a level of which is specified for each read operation (e.g., see read reference voltages Vr**1**, Vr**2**, Vr**3**, Vr**4**, Vr**5**, Vr**6**, and Vr**7**, of FIG. **5**) or verify operation (e.g. see verify reference voltages Vv**1**, Vv**2**, Vv**3**, Vv**4**, Vv**5**, Vv**6**, and Vv**7** of FIG. **5**) in order to determine whether a threshold voltage of the concerned memory cell has reached such level. After applying the word line voltage, the conduction current of the memory cell is measured to determine whether the memory cell turned on (conducted current) in response to the voltage applied to the word line. If the conduction current is measured to be greater than a certain value, then it is assumed that the memory cell turned on and the voltage applied to the word line is greater than the threshold voltage of the memory cell. If the conduction current is not measured to be greater than the certain value, then it is assumed that the memory cell did not turn on and the voltage applied to the word line is not greater than the threshold voltage of the memory cell. During a read or verify process, the unselected memory cells are provided with one or more read pass voltages at their control gates so that these memory cells will operate as pass gates (e.g., conducting current regardless of whether they are programmed or erased).

[0102] There are many ways to measure the conduction current of a memory cell during a read or verify operation. In one example, the conduction current of a memory cell is measured by the rate it discharges or charges a dedicated capacitor in the sense amplifier. In another example, the conduction current of the selected memory cell allows (or fails to allow) the NAND string that includes the memory cell to discharge a corresponding bit line. The voltage on the bit line is measured after a period of time to see whether it has been discharged or not. Note that the technology described herein can be used with different methods known in the art for verifying/reading. Other read and verify techniques known in the art can also be used.

[0103] In some embodiments, controller **122** receives a request from the host (or a client, user, etc.) to program host data (data received from the host) into the memory system. In some embodiments, controller **122** arranges the host data to be programmed into units of data. For example, controller **122** can arrange the host data into pages, word line units, blocks, jumbo blocks, or other units. For purposes of this document, a block is a physical grouping of memory cells. In one example, a block is a unit of erase. However, in other examples a block need not be a unit of erase. In one example, a block comprises a set of memory cells connected by uninterrupted word lines such as a set of NAND strings connected to a common set of word lines. Other physical arrangement can also be used.

[0104] Step **772** of FIG. **7**B includes applying a program voltage pulse on the selected word line. Step **774** of FIG. **7**B includes verification, which in some embodiments comprises applying the verify reference voltages on the selected word line. As steps **772** and **774** are part of an iterative loop, the program voltage is applied as a series of voltage pulses that step up in magnitude. Between voltage pulses, verify reference voltages are applied. This is depicted in FIG. **7**C, which shows program voltage pulses **792**, **794** and **796**, applied during three successive iterations of step **772**. Between program voltage pulses **792**, **794** and **796**, the system tests the memory cells to determine whether threshold voltages of the memory cells are greater than the respective verify reference voltages by applying the verify references voltages as verify pulses.

[0105] FIG. **7**C shows a verify pulse for each of the non-erased states S**1**-S**7** between each of the program voltage pulses **792**, **794** and **796**. These verify pulses consume a significant portion of a program operation. As the number of states stored per memory cell increases, this situation becomes worse, limiting programming speed. FIG. **7**C corresponds to 3-bit per cell and uses 7 verify levels. In a 4-bit per cell embodiment, a verify of all non-erased states would need 15 verify operations between program voltage pulses. To improve performance, some of these verify levels can be skipped at various points of the programming operation through use of an “intelligent” or “smart” verify algorithm. As higher states take longer to program, initially only the lowers states are verified, with the high states being added progressively as the program operation proceeds. Similarly, after some number of voltage pulses, the lower states can progressively be dropped as cells being written to the level will either have completed programming or considered defective.

[0106] As discussed, word lines of memory apparatuses may be relatively faster or slower to program. In 3D NAND, word lines ramp up slowly during a program operation due to various factors, such as a vertical location of the word line, as well as its distance from the driver or charge pump). The worst case expected word line performance is used to dictate the timings for all the word lines in all the blocks. Because of this, the program performance will not be optimum for the normal word lines i.e., the median/average program performance will not be optimum. FIG. **8** is a plot of voltages applied to typical word lines and slow word lines and the resulting potential of the typical word lines and slow word lines during a portion of one loop of a programming operation. One solution is to sense the ramp rates of word lines to detect the slow word lines on the fly, and based on those results, have different timings for slow word lines and normal lines. While that approach can improve the performance, the slow word lines will still ramp up slowly and hence will have a negative impact on the performance, the amount of which depends on the number of slow word lines in a block.

[0107] Consequently, described herein is a memory apparatus (e.g., memory device **100** in FIG. **1**) including memory cells (e.g., memory cells MC**1**-MC**5** in FIG. **4**F) configured to retain a threshold voltage Vth corresponding to one of a plurality of data states (e.g., states S**0**-S**7** in FIG. **5**). The memory apparatus also includes a control means (e.g., control circuitry **110**, controller **122**, decoders **124**, **132**, read/write circuits **128**, and sense blocks **150** in FIG. **1**) configured to identify ones of the plurality of word lines as slow word lines. The control means is also configured to ramp at least one program pulse applied to the slow word lines to a program kick voltage higher in magnitude than a program voltage for a program kick period of time during at least one program loop of a program operation. (i.e., program voltage VPGM kick offset **900**). So, the program voltage VPGM kick offset **900** is used for the slow word lines so that they can reach the final program voltage VPGM value at a time similar to that for normal word lines. FIG. **9** is a plot of voltages applied to typical word lines and slow word lines and the resulting potential of the typical word lines and slow word lines during a portion of one loop of a programming operation using the program voltage kick offset. An improvement or benefit **902** in the program time tPROG from using the program voltage VPGM kick offset **900** is shown.

[0108] According to an aspect, the slow word lines are identified by either testing during the development phase (i.e., before the memory apparatus is in production) or detection of ramp rate on the fly. Such detection on the fly can be done using an analog design for test (DFT) detection circuit of the memory apparatus. Thus, for detection of the ramp rate on the fly, the memory apparatus further includes a design for test circuit (e.g., other discrete components **240** in FIG. **2**) configured to measure a word line potential ramp rate over a word line potential ramp time for each of the plurality of word lines. Design for test detection circuitry and methods are described in U.S. Pat. No. 7,962,819 entitled “Test Mode Soft Reset Circuitry and Methods” and granted Jun. 14, 2011 and incorporated herein by reference in its entirety. So, the control means is further configured to detect the word line potential ramp rate over the word line potential ramp time for the ones of the plurality of word lines using the design for test circuit while the at least one program pulse is applied to the ones of the plurality of word lines. The control means is also configured to identify the ones of the plurality of word lines as the slow word lines based on the word line potential ramp rate over the word line potential ramp time detected for the ones of the plurality of word lines. Details of such detection of the ramp rate is further described in U.S. Ser. No. 18/510,070 entitled "Dynamic Program Performance Modulation in a Memory Device" and filed Nov. 15, 2023 and incorporated herein by reference in its entirety.

[0109] As discussed, the at least one program loop of the program operation can include a plurality of program loops. In addition, the memory cells are disposed in memory holes (e.g., memory holes **422**, **432**, **442** and **452** in FIG. **4**B) grouped into a plurality of strings (e.g., regions or fingers **420**, **430**, **440** and **450** in FIG. **4**B). For detection for the program voltage VPGM, it is best to detect shortly after the program voltage VPGM ramp begins in the first program loop (for example, during a fifteenth time period P15E of the first program loop). The fifteenth time period P15E of the program operation is when the voltage applied to the word line being programmed is ramped up to the program voltage VPGM. If the program voltage VPGM kick offset **900** cannot be applied in the first loop when detection takes place, it can be applied for subsequent loops for the slow word lines. Alternatively, the detection done in the first loop for a first one of the plurality of strings for a word line could be stored and used for the subsequent strings on the same word line. Thus, the control means is further configured to identify the ones of the plurality of word lines as the slow word lines in a first one of the plurality of program loops for one of the plurality of strings. The control means is also configured to utilize the identification of the ones of the plurality of word lines as the slow word lines for the one of the plurality of strings for other ones of the plurality of strings during subsequent ones of the plurality of program loops. If a distance from a charge pump to the word line is not the primary concern, the detection of the slow word line could take place even in the beginning or middle stages of the word line ramp to the program voltage VPGM (for example, at the beginning of P**12** clk/P**13** clk/P**14** clk/P**15**clk shown in FIGS. **8** and **9**) of the very first program pulse (i.e., the first program loop) or in the very early stage of a first pre-charge, even before the unselected bit line ramps up to a reasonable value to play a role (for example, at P**5** clk).

[0110] In addition to ramping the voltage applied to slow word lines for the program voltage VPGM pulses, the kick offset (e.g., program voltage VPGM kick offset **900**) and an optional additional ramp step (e.g., stepwise program kick voltage **1000** in FIG. **10**) described below could be applied to various word line ramp-up or ramp-down signals such as for a pass voltage VPASS, equalization ramp-up or further pre-charge signals or during discharge (such as in PR clk before the P**1** period). Therefore, the control means is further configured to ramp other voltages applied to the slow word lines besides the at least one program pulse for a kick period of time during the at least one program loop to a kick voltage higher in magnitude for the slow word lines ramping up and lower in magnitude for the slow word lines ramping down for a kick period of time during the at least one program loop of the program operation. While the additional kick or kick offset is positive for ramp-up signals, it can be negative for ramp-down signals.

[0111] The magnitude of kick or kick offset may either be a fixed value for a given signal or a percentage of bias change (i.e., percentage of ΔV) needed. Therefore, according to an aspect, the control means is further configured to vary a magnitude of at least one of the kick voltage of the other voltages or a magnitude of the program kick voltage of the at least one program pulse based on a percentage change between an earlier voltage of the slow word lines prior to ramping and the at least one of the kick voltage or the program kick voltage. The magnitude of any other kick offset for other voltages applied to the slow word lines besides the at least one program pulse can be selected in the same way.

[0112] The kick amount can also vary depending on the slowness of the word lines. For example, depending on how much a word line has ramped up (as a percentage of the target) during detection, ramp amounts could vary. Such an approach will help to optimize the kick amount or kick offset magnitude and program time tProg as a function of the word line. Thus, the control means is further configured to quantify a slowness of the slow word lines according to the word line potential ramp rate over the word line potential ramp time detected for the ones of the plurality of word lines. The control means is further configured to vary a magnitude of at least one of the kick voltage of the other voltages or a magnitude of the program kick voltage of the at least one program pulse based on the slowness of the slow word lines.

[0113] In case the number of slow word lines is small, the kick offset (for the program voltage VPGM or otherwise) is applied only on a small number of word lines. Hence, the resulting impact of the kick offset on current consumption would be less in terms of percentage change. Nevertheless, if the impact on peak current consumption is found to be large, then the kick offset could be accompanied by one or more additional steps in the ramp-up signal. This distributes the current consumption, thus minimizing the peak current consumption. So, according to an additional aspect, the control means is further configured to ramp the at least one program pulse applied to the slow word lines to the program kick voltage higher in magnitude than the program voltage for the program kick period of time in a stepwise fashion with a plurality of ramp steps **1002** lasting each of a plurality of ramp time periods during the at least one program loop of the program operation (i.e., stepwise program kick voltage **1000**). In other words, the program voltage VPGM kick offset **900** can comprise the stepwise program kick voltage **1000**. FIG. **10** is a plot of voltages applied to typical word lines and slow word lines and the resulting potential of the typical word lines and slow word lines during a portion of one loop of a programming operation using the program voltage kick offset with a stepwise program kick voltage **1000**. An improvement or benefit **1004** in the program time tPROG from using the stepwise program kick voltage **1000** is shown.

[0114] Now referring to FIG. **11** and back to FIGS. **9** and **10**, a method of operating a memory apparatus is also provided. As discussed above, the memory apparatus (e.g., memory device **100** in FIG. **1**) includes memory cells (e.g., memory cells MC**1**-MC**5** in FIG. **4**F) configured to retain a threshold voltage Vth corresponding to one of a plurality of data states (e.g., states S**0**-S**7** in FIG. **5**). Referring specifically to FIG. **11**, the method includes the step of **1100** identifying ones of the plurality of word lines as slow word lines. The method also includes the step of **1102** ramping at least one program pulse applied to the slow word lines to a program kick voltage higher in magnitude than a program voltage for a program kick period of time during at least one program loop of a program operation (i.e., program voltage VPGM kick offset **900** in FIG. **9**) (to enable a voltage of the slow word lines to quickly reach the program voltage VPGM).

[0115] Again, according to an aspect, the slow word lines are identified by either testing during the development phase (i.e., before the memory apparatus is in production) or detection of ramp rate on the fly, for example, using the analog design for test (DFT) detection circuit of the memory apparatus. Therefore, for detection of the ramp rate on the fly, the memory apparatus can further include the design for test circuit (e.g., other discrete components **240** in FIG. **2**) configured to measure the word line potential ramp rate over the word line potential ramp time for each of the plurality of word lines. Accordingly, the method further includes the step of detecting the word line potential ramp rate over the word line potential ramp time for the ones of the plurality of word lines using the design for test circuit while the at least one program pulse is applied to the ones of the plurality of word lines. The method continues with the step of identifying the ones of the plurality of word lines as the slow word lines based on the word line potential ramp rate over the word line potential ramp time detected for the ones of the plurality of word lines.

[0116] Once again, the at least one program loop of the program operation can include the plurality of program loops and the memory cells can be disposed in memory holes (e.g., memory holes **422**, **432**, **442** and **452** in FIG. **4**B) grouped into the plurality of strings (e.g., regions or fingers **420**, **430**, **440** and **450** in FIG. **4**B). So, the method further includes the step of identifying the ones of the plurality of word lines as the slow word lines in a first one of the plurality of program loops for one of the plurality of strings. The method also includes the step of utilizing the identification of the ones of the plurality of word lines as the slow word lines for the one of the plurality of strings for other ones of the plurality of strings during subsequent ones of the plurality of program loops.

[0117] Again, in addition to ramping the voltage applied to slow word lines for the program voltage VPGM pulses, the kick offset and the stepwise program kick voltage **1000** could be applied to various word line ramp-up or ramp-down signals such as for the pass voltage VPASS, equalization ramp-up or further pre-charge signals or during discharge (such as in PR clk before the P**1** period). Thus, the method further includes the step of ramping other voltages applied to the slow word lines besides the at least one program pulse for a kick period of time during the at least one program loop to a kick voltage higher in magnitude for the slow word lines ramping up and lower in magnitude for the slow word lines ramping down for a kick period of time during the at least one program loop of the program operation.

[0118] As above, the magnitude of kick or kick offset may either be a fixed value for a given signal or a percentage of bias change (i.e., percentage of AV) needed. Thus, according to an aspect, the method further includes the step of varying a magnitude of at least one of the kick voltage of the other voltages or a magnitude of the program kick voltage of the at least one program pulse based on a percentage change between an earlier voltage of the slow word lines prior to ramping and the at least one of the kick voltage or the program kick voltage. Again, the magnitude of any other kick offset for other voltages applied to the slow word lines besides the at least one program pulse can be selected in the same way.

[0119] As discussed, the kick amount can also vary depending on the slowness of the word lines. Therefore, the method additionally includes the step of quantifying a slowness of the slow word lines according to the word line potential ramp rate over the word line potential ramp time detected for the ones of the plurality of word lines. The method also includes the step of varying a magnitude of at least one of the kick voltage of the other voltages or a magnitude of the program kick voltage of the at least one program pulse based on the slowness of the slow word lines.

[0120] Again, if the impact on peak current consumption is found to be large, then the kick offset could be accompanied by one or more additional steps in the ramp-up signal, as shown in FIG. **10**, for example. This distributes the current consumption, thus minimizing the peak current consumption. Accordingly, the method further includes the step of ramping the at least one program pulse applied to the slow word lines to the program kick voltage higher in magnitude than the program voltage for the program kick period of time in a stepwise fashion with a plurality of ramp steps lasting each of a plurality of ramp time periods during the at least one program loop of the program operation.

[0121] Clearly, changes may be made to what is described and illustrated herein without, however, departing from the scope defined in the accompanying claims. The foregoing description of the embodiments has been provided for purposes of illustration and description. It is not intended to be exhaustive or to limit the disclosure. Individual elements or features of a particular embodiment are generally not limited to that particular embodiment, but, where applicable, are interchangeable and can be used in a selected embodiment, even if not specifically shown or described. The same may also be varied in many ways. Such variations are not to be regarded as a departure from the disclosure, and all such modifications are intended to be included within the scope of the disclosure.

[0122] The terminology used herein is for the purpose of describing particular example embodiments only and is not intended to be limiting. As used herein, the singular forms "a," "an," and "the" may be intended to include the plural forms as well, unless the context clearly indicates otherwise. The terms "comprises," "comprising," "including," and "having," are inclusive and therefore specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof. The method steps, processes, and operations described herein are not to be construed as necessarily requiring their performance in the particular order discussed or illustrated, unless specifically identified as an order of performance. It is also to be understood that additional or alternative steps may be employed.

[0123] When an element or layer is referred to as being "on," "engaged to," "connected to," or "coupled to" another element or layer, it may be directly on, engaged, connected or coupled to the other element or layer, or intervening elements or layers may be present. In contrast, when an element is referred to as being "directly on," "directly engaged to," "directly connected to," or "directly coupled to" another element or layer, there may be no intervening elements or layers present. Other words used to describe the relationship between elements should be interpreted in a like fashion (e.g., "between" versus "directly between," "adjacent" versus "directly adjacent," etc.). As used herein, the term "and/or" includes any and all combinations of one or more of the associated listed items.

[0124] Although the terms first, second, third, etc. may be used herein to describe various elements, components, regions, layers and/or sections, these elements, components, regions, layers and/or sections should not be limited by these terms. These terms may be only used to distinguish one element, component, region, layer or section from another region, layer or section. Terms such as "first," "second," and other numerical terms when used herein do not imply a sequence or order unless clearly indicated by the context. Thus, a first element, component, region, layer or section discussed below could be termed a second element, component, region, layer or section without departing from the teachings of the example embodiments.

[0125] Spatially relative terms, such as "inner," "outer," "beneath," "below," "lower," "above," "upper," "top", "bottom", and the like, may be used herein for ease of description to describe one element's or feature's relationship to another element(s) or feature(s) as illustrated in the figures. Spatially relative terms may be intended to encompass different orientations of the device in use or operation in addition to the orientation depicted in the figures. For example, if the device in the figures is turned over, elements described as "below" or "beneath" other elements or features would then be oriented "above" the other elements or features. Thus, the example term "below" can encompass both an orientation of above and below. The device may be otherwise oriented (rotated 90 degrees or at other orientations) and the spatially relative descriptions used herein interpreted accordingly.

What is claimed is:

1. A memory apparatus, comprising:

memory cells each connected to one of a plurality of word lines and configured to store a threshold voltage corresponding to one of a plurality of data states; and

a control means configured to:

identify ones of the plurality of word lines as slow word lines, and

ramp at least one program pulse applied to the slow word lines to a program kick voltage higher in magnitude than a program voltage for a program kick period of time during at least one program loop of a program operation.

2. The memory apparatus as set forth in claim **1**, further including a design for test circuit configured to measure a word line potential ramp rate over a word line potential ramp time for each of the plurality of word lines, wherein the control means is further configured to:

detect the word line potential ramp rate over the word line potential ramp time for the ones of the plurality of word lines using the design for test circuit while the at least one program pulse is applied to the ones of the plurality of word lines; and

identify the ones of the plurality of word lines as the slow word lines based on the word line potential ramp rate over the word line potential ramp time detected for the ones of the plurality of word lines.

3. The memory apparatus as set forth in claim **2**, wherein the control means is further configured to ramp other voltages applied to the slow word lines besides the at least one program pulse for a kick period of time during the at least one program loop to a kick voltage higher in magnitude for the slow word lines ramping up and lower in magnitude for the slow word lines ramping down for a kick period of time during the at least one program loop of the program operation.

4. The memory apparatus as set forth in claim **3**, wherein the control means is further configured to vary a magnitude of at least one of the kick voltage of the other voltages or a magnitude of the program kick voltage of the at least one program pulse based on a percentage change between an earlier voltage of the slow word lines prior to ramping and the at least one of the kick voltage or the program kick voltage.

5. The memory apparatus as set forth in claim **3**, wherein the control means is further configured to:

quantify a slowness of the slow word lines according to the word line potential ramp rate over the word line potential ramp time detected for the ones of the plurality of word lines; and

vary a magnitude of at least one of the kick voltage of the other voltages or a magnitude of the program kick voltage of the at least one program pulse based on the slowness of the slow word lines.

6. The memory apparatus as set forth in claim **1**, wherein the at least one program loop of the program operation includes a plurality of program loops, the memory cells are disposed in memory holes grouped into a plurality of strings, and the control means is further configured to:

identify the ones of the plurality of word lines as the slow word lines in a first one of the plurality of program loops for one of the plurality of strings; and

utilize the identification of the ones of the plurality of word lines as the slow word lines for the one of the plurality of strings for other ones of the plurality of strings during subsequent ones of the plurality of program loops.

7. The memory apparatus as set forth in claim **1**, wherein the control means is further configured to ramp the at least one program pulse applied to the slow word lines to the program kick voltage higher in magnitude than the program voltage for the program kick period of time in a stepwise fashion with a plurality of ramp steps lasting each of a plurality of ramp time periods during the at least one program loop of the program operation.

8. A controller in communication with a memory apparatus including memory cells each connected to one of a plurality of word lines and configured to store a threshold voltage corresponding to one of a plurality of data states, the controller configured to:

identify ones of the plurality of word lines as slow word lines; and

instruct the memory apparatus to ramp at least one program pulse applied to the slow word lines to a program kick voltage higher in magnitude than a program voltage for a program kick period of time during at least one program loop of a program operation.

9. The controller as set forth in claim **8**, wherein the memory apparatus further includes a design for test circuit configured to measure a word line potential ramp rate over a word line potential ramp time for each of the plurality of word lines, and the controller is further configured to:

instruct the memory apparatus to detect the word line potential ramp rate over the word line potential ramp time for the ones of the plurality of word lines using the design for test circuit while the at least one program pulse is applied to the ones of the plurality of word lines; and

identify the ones of the plurality of word lines as the slow word lines based on the word line potential ramp rate over the word line potential ramp time detected for the ones of the plurality of word lines.

10. The controller as set forth in claim **9**, wherein the controller is further configured to instruct the memory apparatus to ramp other voltages applied to the slow word lines besides the at least one program pulse for a kick period of time during the at least one program loop to a kick voltage higher in magnitude for the slow word lines ramping up and lower in magnitude for the slow word lines ramping down for a kick period of time during the at least one program loop of the program operation.

11. The controller as set forth in claim **10**, wherein the controller is further configured to instruct the memory apparatus to vary a magnitude of at least one of the kick voltage of the other voltages or a magnitude of the program kick voltage of the at least one program pulse based on a percentage change between an earlier voltage of the slow word lines prior to ramping and the at least one of the kick voltage or the program kick voltage.

12. The controller as set forth in claim **10**, wherein the controller is further configured to:

quantify a slowness of the slow word lines according to the word line potential ramp rate over the word line potential ramp time detected for the ones of the plurality of word lines; and

instruct the memory apparatus to vary a magnitude of at least one of the kick voltage of the other voltages or a magnitude of the program kick voltage of the at least one program pulse based on the slowness of the slow word lines.

13. The controller as set forth in claim **8**, wherein the controller is further configured to instruct the memory apparatus to ramp the at least one program pulse applied to the slow word lines to the program kick voltage higher in magnitude than the program voltage for the program kick period of time in a stepwise fashion with a plurality of ramp steps lasting each of a plurality of ramp time periods during the at least one program loop of the program operation.

14. A method of operating a memory apparatus including memory cells each connected to one of a plurality of word lines and configured to store a threshold voltage corresponding to one of a plurality of data states; the method comprising the steps of:

identifying ones of the plurality of word lines as slow word lines; and

ramping at least one program pulse applied to the slow word lines to a program kick voltage higher in magnitude than a program voltage for a program kick period of time during at least one program loop of a program operation.

15. The method as set forth in claim **14**, wherein the memory apparatus further includes a design for test circuit configured to measure a word line potential ramp rate over a word line potential ramp time for each of the plurality of word lines, and the method further includes the steps of:

detecting the word line potential ramp rate over the word line potential ramp time for the ones of the plurality of word lines using the design for test circuit while the at least one program pulse is applied to the ones of the plurality of word lines; and

identifying the ones of the plurality of word lines as the slow word lines based on the word line potential ramp rate over the word line potential ramp time detected for the ones of the plurality of word lines.

16. The method as set forth in claim **15**, further including the step of ramping other voltages applied to the slow word lines besides the at least one program pulse for a kick period of time during the at least one program loop to a kick voltage higher in magnitude for the slow word lines ramping up and lower in magnitude for the slow word lines ramping down for a kick period of time during the at least one program loop of the program operation.

17. The method as set forth in claim **16**, further including the step of varying a magnitude of at least one of the kick voltage of the other voltages or a magnitude of the program kick voltage of the at least one program pulse based on a percentage change between an earlier voltage of the slow word lines prior to ramping and the at least one of the kick voltage or the program kick voltage.

18. The method as set forth in claim **16**, further including the steps of:

quantifying a slowness of the slow word lines according to the word line potential ramp rate over the word line potential ramp time detected for the ones of the plurality of word lines; and

varying a magnitude of at least one of the kick voltage of the other voltages or a magnitude of the program kick voltage of the at least one program pulse based on the slowness of the slow word lines.

19. The method as set forth in claim **14**, wherein the at least one program loop of the program operation includes a plurality of program loops, the memory cells are disposed in memory holes grouped into a plurality of strings, and the method further includes the steps of:

identifying the ones of the plurality of word lines as the slow word lines in a first one of the plurality of program loops for one of the plurality of strings; and

utilizing the identification of the ones of the plurality of word lines as the slow word lines for the one of the plurality of strings for other ones of the plurality of strings during subsequent ones of the plurality of program loops.

20. The method as set forth in claim **14**, further including the step of ramping the at least one program pulse applied to the slow word lines to the program kick voltage higher in magnitude than the program voltage for the program kick period of time in a stepwise fashion with a plurality of ramp steps lasting each of a plurality of ramp time periods during the at least one program loop of the program operation.

* * * * *