



- (51) **International Patent Classification:**
G06F 12/00 (2006.01) *G06F 13/14* (2006.01)
- (21) **International Application Number:**
PCT/US2014/048449
- (22) **International Filing Date:**
28 July 2014 (28.07.2014)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P. [US/US]; 11445 Compaq Center Drive W., Houston, Texas 77070 (US).
- (72) **Inventors:** GIBSON, Gary; 1501 Page Mill Rd., Palo Alto, California 94304-1100 (US). LESARTRE, Gregg, B.; 3404 E Harmony Rd., Ft. Collins, Colorado 80528-9544 (US). WILLIAMS, R., Stanley; 1501 Page Mill Rd., Palo Alto, California 94304-1100 (US).
- (74) **Agents:** COLLINS, David, W. et al.; Hewlett-Packard Company, Intellectual Property Administration, 3404 E. Harmony Road, Mail Stop 35, Fort Collins, Colorado 80528 (US).
- (81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,

BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

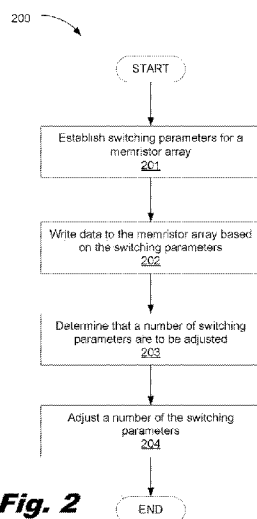
Declarations under Rule 4.17:

- *as to the identity of the inventor (Rule 4.17(i))*
- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

- *with international search report (Art. 21(3))*

(54) **Title:** ADJUSTING SWITCHING PARAMETERS OF A MEMRISTOR ARRAY

**Fig. 2**

(57) **Abstract:** A system for adjusting switching parameters of a memristor array is described. The system includes a processor, a memristor array, and a memory manager communicatively coupled to the processor and the memristor array. The memory manager includes an establish module to establish switching parameters for the memristor array. The memory manager also includes an adjustment module to adjust a number of the switching parameters.

ADJUSTING SWITCHING PARAMETERS OF A MEMRISTOR ARRAY

BACKGROUND

[0001] Memory arrays are used to store data. A memory array may be made up of a number of memory elements. Data may be stored to memory elements by setting values of the memory elements within the memory arrays. For example, the memory bits may be set to 0, 1, or combinations thereof to store data in a memory bit of a memory array.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] The accompanying drawings illustrate various examples of the principles described herein and are a part of the specification. The illustrated examples do not limit the scope of the claims.

[0003] Fig. 1 is a diagram of a system for adjusting switching parameters of a memristor array according to one example of the principles described herein.

[0004] Fig. 2 is a flowchart of a method for adjusting switching parameters of a memristor array according to one example of the principles described herein.

[0005] Fig. 3 is a diagram of a memory manager for adjusting switching parameters of a memristor array according to one example of the principles described herein.

[0006] Fig. 4 is a flowchart of a method for adjusting switching parameters of a memristor array according to another example of the principles described herein.

[0007] Fig. 5 is a flowchart of a method for adjusting switching parameters of a memristor array according to another example of the principles described herein.

[0008] Fig. 6 is a diagram of a memory manager for adjusting switching parameters of a memristor array according to another example of the principles described herein.

[0009] Throughout the drawings, identical reference numbers designate similar, but not necessarily identical, elements.

DETAILED DESCRIPTION

[0010] As described above, memory arrays may be used to store data by setting memory bit values within the memory array. More specifically, a memristor array including a number of memristors may be used to store data by setting memristor resistance values within the memristor array. In using a memristor as an element in a memory circuit, a digital operation is implemented by applying electrical pulses to set the resistance of a memristor to a particular value, e.g., a "low resistance state" that is associated with a logical state, such as "1." An electrical pulse with a different shape, amplitude, duration, and/or polarity may set the resistance of the memristor to a different value, e.g., a "high resistance state," associated with another logical state, such as "0." Pulses used to switch a memristor into a low resistance state may be referred to as "set pulses" and pulses used to switch a memristor into a high resistance state may be referred to as "reset pulses."

[0011] A specific example of switching a memristor is given as follows. A resistance value associated with the low resistance state may be 30 kilohms (k Ω). Accordingly, when a computing device reads that the memristor has a resistance near 30 k Ω , the computing device may associate this with a logical state of 1. Similarly, a resistance value associated with the high resistance state may be 150 k Ω . Accordingly, when a computing device reads that a memristor has a resistance near 150 k Ω , the computing device may associate this with a

logical state of 0. While memristors may be useful as a memory storage element, certain characteristics may complicate their use.

[0012] For example, memristor elements may be subject to statistical variation of a response to an electrical pulse that may cause the resulting resistance values of the memristor to drift. In other words, the resistance value that results after applying an electrical pulse intended to produce the high resistance state originally defined to correspond to a logical state of "0," may no longer be the originally determined resistance value of the high resistance state. For example, initially, a resistance value associated with the high resistance state may be 150 k Ω . However, over time, a drift may occur such that a value other than 150 k Ω results when using the original electrical reset pulse to put the memristor into the high resistance state. In another example, over time the initially established electrical reset pulse may no longer reset the memristor to the desired resistance value. Similarly, over time the initially established electrical set pulse may no longer set the memristor to the desired resistance value.

[0013] Such a drift may lead to a failure to accurately read data from the memristor, or a failure to read data at all. For example, the drift may be such that the bit isn't written correctly thus making the reading of the data uncertain. Moreover, even if a response was deterministic, the pulses may not be controlled to infinite precision. Accordingly, drift may result by inexact compensation of set and reset pulses after a certain number of writes. Additionally, many writes and/or reads may cause a memory bit to be "overwritten" or "overread" such that an accurate indication of a 1 or a 0 may not be possible.

[0014] The values stored in memristors within the same memristor array may also cause variability in the pulse seen at the memristor being written. For example, memristors may be arranged in a crossbar array in which each memristor is formed at an intersection of a first set of elements and a second set of elements, the elements forming a grid of intersecting nodes, each node defining a memristor. A target memristor is selected by applying an electrical bias to a row and column that correspond to the target memristor. In this example, as particular biases are applied at a row and a column of a target memristor, the voltage drop across the target memristor may depend on the resistance states of

other memristors in the array. Additionally, the response of memristor elements to a given write pulse may vary depending on the number of times a memristor has been written and erased.

[0015] Accordingly, certain procedures may be implemented to ensure the accuracy of bit writing. For example, an electrical pulse may be sent to a memristor, which pulse is intended to write a value to a memristor by setting the memristor to a particular resistance value. Such a pulse may be referred to as a write pulse. The memristor may then be read to determine whether the memristor has a resistance level that corresponds with the write pulse. For example, a write pulse may be intended to set the memristor to a low resistance state, referred to as a 1. The memristor may then be read to determine whether a 1 is present, which may indicate that the memristor has been set to the low resistance state. If the memristor does not have a resistance value in the correct range, a subsequent voltage pulse may be sent to put the memristor into the correct resistance range. However, such an operation may be time-consuming as at least a write operation and a read operation are needed for each attempt to write data, and in some cases may include subsequent write and read operations.

[0016] In another example, a feedback circuit may be included in the memristor array to determine whether proper values are written to the memristors. In this example, instead of a voltage pulse, a current ramp is implemented and the ramp is shut down when a particular resistance value of the memristor is achieved. However, this approach may use additional circuitry which may be large and may take up valuable chip space. Additionally, the use of a feedback circuit in a crossbar array may be ineffective as the many parallel currents in a crossbar array may exhibit current leakage, making monitoring current through individual elements in the crossbar array very difficult.

[0017] Accordingly, the present disclosure describes systems and methods for adjusting switching parameters of memristors in a memristor array. More specifically, the present disclosure allows for an establishment of switching parameters for a memristor array. Then, periodically, the switching parameters may be adjusted such that the memristor array is operating within a desired

operating range. For example, upon detection that at least a number of memristors have exhibited a drift, all of the memristors in an array may be adjusted. Adjusting the switching parameters in this fashion may keep the memristor array in an optimal operating range, may minimize variability between memristors, may improve reliability and may increase endurance by maintaining the memristor in a desired operating range.

[0018] The present disclosure describes a system for switching parameters of a memristor array. The system includes a processor, a memristor array, and a memory manager communicatively coupled to the processor and memristor array. The memory manager includes an establish module to establish switching parameters for the memristor array. The memory manager also includes an adjustment module to adjust a number of the switching parameters.

[0019] The present disclosure describes a method for adjusting switching parameters of a memristor array. The method includes, with a processor, establishing switching parameters for the memristor array. The method also includes writing data to the memristor array based on the switching parameters and determining that a number of the switching parameters are to be adjusted. The method further includes adjusting the number of the switching parameters determined to need adjusting.

[0020] The present disclosure describes a computer program product for adjusting switching parameters of a memristor array. The computer program product includes a computer readable storage medium. The computer program storage medium includes computer usable program code. The computer usable program code includes computer usable program code to, when executed by a processor, establish switching parameters for the memristor array, write data to the memristor array based on the switching parameters, determine that a number of the switching parameters are to be adjusted, adjust the number of the switching parameters, and write data to the memristor array based on the adjusted switching parameters.

[0021] The systems and methods described herein may be beneficial by allowing a memristor array to operate within a predetermined operating range.

Additionally, variability in memristor array resistance responses to electrical pulses may be reduced. Doing so may increase reliability and increase endurance of the memristor array by maintaining the memristor array within a predetermined operating range.

[0022] As used in the present specification and in the appended claims, the term “memristor” may refer to a passive two-terminal circuit element that maintains a functional relationship between the time integral of current, and the time integral of voltage.

[0023] Still further, as used in the present specification and in the appended claims the term “switching event” may refer to the switching of a memristor between states. For example, a switching event may refer to a memristor switching from a low resistance state to a high resistance state.

[0024] Still further, as used in the present specification and in the appended claims, the term “switching parameters” may refer to a number of characteristics that define a switching event. For example, a voltage pulse may be used to set the resistance value of a memristor. In another example, a target resistance value of the memristor may be associated with a logical state. A number of characteristics of the memristor elements within the memristor array may also be included as a switching parameter. Accordingly, as used in the present specification and in the appended claims the term “characteristics” may refer to any characteristic of the memristor element that may affect the memristors ability to execute a switching event at prescribed voltages. The characteristics of the memristor may include material properties of the memristor. The voltage pulse and target resistance value may be switching parameters. Accordingly, as used in the present specification and in the appended claims a “target resistance value” may refer to a specific resistance value, or a target resistance range. For example, a threshold resistance range could be identified that marks the boundaries of ranges assigned to logical states.

[0025] Yet further, as used in the present specification and in the appended claims, the term “condition,” may refer to any criteria used to determine that a memristor has drifted and that the switching parameters are to be adjusted

because the initially established switching parameters no longer produce a switching event.

[0026] Still further, as used in the present specification and in the appended claims, the term “a number of” or similar language may include any positive number including 1 to infinity; zero not being a number, but the absence of a number.

[0027] In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present systems and methods. It will be apparent, however, to one skilled in the art that the present apparatus, systems, and methods may be practiced without these specific details. Reference in the specification to “an example” or similar language means that a particular feature, structure, or characteristic described is included in at least that one example, but not necessarily in other examples.

[0028] Fig. 1 is a diagram of a system (100) for adjusting switching parameters of memristors in a memristor array (113) according to one example of the principles described herein. The system (100) may be utilized in any data processing scenario including, for example, a cloud computing service such as a Software as a Service (SaaS), a Platform as a Service (PaaS), an Infrastructure as a Service (IaaS), application program interface (API) as a service (APIaaS), other forms of network services, or combinations thereof. Further, the system (100) may be used in a public cloud network, a private cloud network, a hybrid cloud network, other forms of networks, or combinations thereof. In one example, the methods provided by the system (100) are provided as a service over a network by, for example, a third party. In another example, the methods provided by the system (100) are executed by a local administrator.

[0029] Further, the system (100) may be utilized within a single computing device. In this data processing scenario, a single computing device may utilize the memristor array (113) and other associated methods described herein to store data.

[0030] In general, the system (100) includes a memristor array (113). For simplicity, in Fig. 1, the system (100) is depicted with a single memristor array

(113), however, the system (100) may have any number of memristor arrays (113). A memristor array (113) may include a number of memristors (114) that are used to store information. The memristors (114) are non-volatile memory elements that may maintain information in the face of a loss of power. For example, if power is removed from a memristor array (113), data stored on the non-volatile memristor array (113) may be accessed once power is again restored to the memristor array (113). For simplicity, Fig. 1 depicts three memristors (114-1, 114-2, 114-3) in a memristor array (113). However, a memristor array (113) may include any number of memristors (114). In some examples, the memristor array (113) may be a cross-bar memristor array (113). A cross-bar memristor array (113) may include a first number of parallel wires and a second number of parallel wires, the second number being perpendicular to the first number. The first number and second number of wires intersect at nodes, these nodes may form the memristors (114) of the memristor array (113).

[0031] The memristors (114) store information by indicating a logical state to a processor (101). The memristors (114) may indicate this logical state based on their resistance. For example, a memristor (114) in a low resistance state may indicate a logical state of 1. My comparison, a memristor (114) in a high resistance state may indicate a logical state of 0. The number of memristors (114) in a memristor array (113) may form a sequence of 1s and 0s that indicate stored information. The resistance state, and the corresponding logical state, are selected by an electrical pulse sent from the memory manager (104) to the memristor array (113).

[0032] To achieve its desired functionality, the system (100) comprises various hardware components. Among these hardware components may be a number of processors (101), a memory manager (104), a number of peripheral device adapters (103), and a number of network adapters (102). These hardware components may be interconnected through the use of a number of busses and/or network connections. In one example, the number of busses that interconnect a number of these devices may be represented by the reference numeral (112). In other words, the reference number (112) may designate a connection and may not indicate a particular number of connections. For

example, the memory device (104) may be connected to a number of other elements in the system (100) and may use a bus distinct from a bus used by the peripheral device adapters (103) to communicate with other elements of the system (100).

[0033] The processor (101) may include the hardware architecture to retrieve executable code from the memory manager (104) and execute the executable code. The executable code may, when executed by the processor (101), cause the processor (101) to implement at least the functionality of adjusting switching parameters of memristors in a memristor array, according to the methods of the present specification described herein. In the course of executing code, the processor (101) may receive input from and provide output to a number of the remaining hardware units.

[0034] The memory manager (104) may store data such as executable program code that is executed by the processor (101) or other processing device. As will be discussed, the memory manager (104) may specifically store a number of applications that the processor (101) executes to implement at least the functionality described herein.

[0035] The memory manager (104) may be coupled to, include, or combinations thereof, various types of memory, including volatile and nonvolatile memory. For example, the memory device (104) of the present example may be coupled to Random Access Memory (RAM) (107), Read Only Memory (ROM) (108), and Hard Disk Drive (HDD) memory (109). Many other types of memory may also be utilized, and the present specification contemplates the use of many varying type(s) of memory as may suit a particular application of the principles described herein. In certain examples, different types of memory may be used for different data storage needs. For example, in certain examples the processor (101) may boot from Read Only Memory (ROM) (108), maintain nonvolatile storage in the Hard Disk Drive (HDD) memory (109), and execute program code stored in Random Access Memory (RAM) (107).

[0036] Generally, the memory manager (104) may comprise a computer readable medium, a computer readable storage medium, or a non-transitory computer readable medium, among others. For example, the memory

manager (104) may be, but is not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples of the computer readable storage medium may include, for example, the following: an electrical connection having a number of wires, a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device. In another example, a computer readable storage medium may be any non-transitory medium that can contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device.

[0037] The hardware adapters (103) in the system (100) enable the processor (101) to interface with various other hardware elements, external and internal to the system (100). For example, peripheral device adapters (103) may provide an interface to input/output devices, such as, for example, display device (110) or access other external devices such as an external storage device (111). The display device (110) may be provided to allow a user to interact with and implement the functionality of the system (100). The peripheral device adapters (103) may also create an interface between the processor (101) and a printer, the display device (110), or other media output device. The network adapter (102) may provide an interface to other computing devices within, for example, a network, thereby enabling the transmission of data between the system (100) and other devices located within the network.

[0038] The system (100) further comprises a number of modules used in the adjustment of switching parameters for a memristor array. The various modules within the system (100) may be executed separately. In this example, the various modules may be stored as separate computer program products. In another example, the various modules within the system (100) may be combined

within a number of computer program products; each computer program product comprising a number of the modules.

[0039] The memory manager (104) may include an establish module (105) and an adjustment module (106). The modules (105, 106) refer to a combination of hardware and program instructions to perform a designated function. The program instructions are stored in the memory and cause the processor (101) to execute the designated function of the module. In this example, the various modules may be stored as separate computer program products.

[0040] The establish module (105) may establish switching parameters for a memristor array (113). As described above, a memristor (114) may be a variable resistor element. In other words, a memristor (114) may exhibit different resistances under different circumstances. The memristor (114) changes resistance values when an electrical pulse, i.e., a switching voltage or a current ramp, is applied to the memristor (114). In some examples, the memristor (114) may switch states when an electrical pulse of a certain value is passed to the memristor (114) for a specified duration of time. For example, if a memristor (114) has a switching voltage of 3 V, a supplied voltage of 2.5 V would not switch the state of the memristor (114). By comparison a supplied voltage of 3 V would switch the state of the memristor (114).

[0041] In some examples, a number of electrical pulses may be used to produce different state changes within the memristor (114). For example, an electrical pulse of a first value and a first duration may be used to switch the memristor (114) from a high resistance state to a low resistance state. In this example, an electrical pulse of a second value and a second duration may be used to switch the memristor (114) from the low resistance state to the high resistance state.

[0042] In these examples, the memristor (114) has been described as having two states, with the memristor (114) being able to switch between states in either direction. However, a memristor (114) may have any number of states and there may be a corresponding number of switching parameters. The electrical pulse may be a voltage pulse that is sent from a voltage source. In

another example, the electrical pulse may be a current pulse that is sent from a current source. The electrical pulse may have a value and a duration. For example a voltage pulse of 2.5V may be applied for a specified amount of time. Accordingly, the establish module (105) may establish an electrical pulse to switch the number of memristors (114) by establishing a value of the electrical pulse as well as establishing a duration for which the electrical pulse is supplied.

[0043] The establish module (105) may also establish a number of target resistance values that correspond to a number of logical states. As described above, a target resistance value may be a specific value, or a range of values, that correspond to a logical state. A memristor (114) may have a number of resistance states that pertain to a number of logical states of the memristor (114). A memristor (114) in a particular resistance state may exhibit a corresponding target resistance value. In other words, a memristor with a target resistance value may be said to be in a resistance state. For example, a memristor with a resistance value of 30 k Ω may be said to be in a low resistance state. Similarly, a memristor with a resistance value of 150 k Ω may be said to be in a high resistance state. Accordingly, the establish module (105) may establish a target resistance value for the memristor (114) for each logical state. For example, the establish module (105) may establish that a target resistance of 30 k Ω may pertain to a low resistance state and a target resistance of 150 k Ω may pertain to a high resistance state. It should be noted that while specific resistance values are indicated, the high resistance state and low resistance state may correspond to any specified resistance value and any resistance value may be implemented in conjunction with the system (100). The established electrical pulses and target resistance values may be referred to as switching parameters.

[0044] The establish module (105) may also establish a number of characteristics of the memristors (114) within the memristor array (103). The characteristics of the memristor (114) may be any number of characteristics that allow the memristor (114) to switch between specific target resistance values when the associated electrical pulses are received at the memristor (114). Establishing the number of characteristics of the memristors (114) may include

obtaining the characteristics of the memristors (114) for example through testing or other information associated with the memristor (114).

[0045] The memory manager (104) may include an adjustment module (106) that adjusts a number of the switching parameters. For example, as described above, a memristor (114) in a memristor array (113) is subject to drift, which may indicate that the electrical pulse no longer sets the memristor (114) to a target resistance value that corresponds to a particular logic state. For example, an initial voltage pulse of 2.5V for a specified period of time may not set the target resistance value of the memristor (114) to the 150 k Ω that corresponds to a high resistance state. In this example, the adjustment module (106) may adjust the voltage pulse, for example by altering the pulse value or pulse duration, such that the voltage pulse again sets the resistance of the memristor (114) to the 150 k Ω , or other target resistance value, that corresponds to the high resistance state.

[0046] In another example, the drift may be such that the resistance values that result from application of the original set or reset pulses no longer correspond to the resistance values originally assigned to the corresponding logical states. For example, continuing from above, over time a memristor (114) may drift such that the original target resistance value of 150 k Ω , which originally corresponded to a logical state of 0, no longer results after application of the original reset pulse. Accordingly, the adjustment module (106) may adjust the target resistance value that corresponds to the logical state.

[0047] In yet another example, the adjustment module (106) may reinitialize the memristor (114) such that the initially established switching parameters execute a switching event. This may include reinitializing a number of the characteristics of the memristor (114). For example, as described above, an initial voltage pulse of 2.5V for a specified period of time may set a memristor (114) to a resistance value of 150 k Ω . Over time, these initial values may not put the memristor (114) in a high resistance state. For example, a larger pulse may be used, or a different resistance value may be associated with a logical state. In this example, rather than adjusting the voltage pulse or target resistance value that corresponds to a particular logic state, the adjustment module (106) may

initialize the memristor (114) using a reinitialization procedure such that the initial switching parameters carry out the intended switching event and achieve the original target resistance value.

[0048] In adjusting the switching parameters, data that is originally stored in the memristors (114) may be wiped out. In other words, in addition to resetting the switching parameters, the adjustment of the switching parameters also cleans off memristors (114) such that new data may be written to the memristor array (113) without possible interference from previously stored data.

[0049] Including an adjustment module (106) as described herein may be beneficial in that it allows for customization of memristor (114) control such that memristors (114) may be more efficiently used to store information and to improve the ease and efficacy of obtaining the information stored in the memristor array (113).

[0050] Fig. 2 is a flowchart of a method (200) for adjusting switching parameters of a memristor array (Fig. 1, 113) according to one example of the principles described herein. The method (200) may be carried out by a processor (Fig. 1, 101) coupled to, or included in, the memory manager (Fig. 1, 104). The method (200) may include establishing (block 201) switching parameters for memristor array (Fig. 1, 113). As described above, a number of switching parameters may be used to define when a switching event for a memristor (Fig. 1, 114) is to occur, a switching event referring to a memristor (Fig. 1, 114) switching between states.

[0051] Returning to the switching parameters, an example of a switching parameter includes an electrical pulse that switches a memristor (Fig. 1, 114) between resistance values, each resistance value corresponding to a logical state. For example, a memristor (Fig. 1, 114) may have multiple logical states that may be used to store data. In a more specific example, a memristor (Fig. 1, 114) may have resistance values that correspond to two logical states, 1 and 0. Memristors (Fig. 1, 114) at different resistance values form a string of 1s and 0s that indicate information stored in a memristor array (Fig. 1, 113). To set the memristors (Fig. 1, 114) to these different logical states, an electrical pulse having a value and a duration is applied to the memristor (Fig. 1, 114) to change

the resistance value of the memristor (Fig. 1, 114). Accordingly, the method (200) may include establishing (block 201) a number of electrical pulses to switch the number of memristors (Fig. 1, 114) between a number of target resistance levels and corresponding logical states.

[0052] Another example of a switching parameter is a resistance value that corresponds to a particular logical state. For example, a resistance value of 150 k Ω may define a high resistance state, which high resistance state corresponds to a logical state of 0. Accordingly, the method (200) may include establishing (block 201) a number of target resistance values corresponding to the number of logical states.

[0053] As yet another example, the characteristics of the memristors (Fig. 1, 114) may be included as switching parameters. The characteristics of the memristors (Fig. 1, 114) refer to elements of the memristor (Fig. 1, 114) that allow the memristor (Fig. 1, 114) to be set or reset to the target resistance values based on the electrical pulses received by the memristor (Fig. 1, 114). Accordingly, the method (200) may include establishing (block 201) the characteristics of a number of memristors (Fig. 1, 114) within the memristor array (Fig. 1, 113), which establishing may include obtaining data or information pertaining to the memristor (Fig. 1, 114) characteristics.

[0054] The method (200) may include writing (block 202) data to the memristor array (Fig. 1, 113) based on the switching parameters. For example, the number of electrical pulses that were established may be used to set a memristor (Fig. 1, 114) to a particular target resistance value that was established. More specifically, a number of electrical pulses may be sent to a number of non-sequential memristors (Fig. 1, 114), while no electrical pulse is sent to another number of non-sequential memristors (Fig. 1, 114). Accordingly, the sequential memristors (Fig. 1, 114) may form a string of 1s and 0s that may be used to store data. When reading the data, the processor (Fig. 1, 101) may include circuitry to read the resistance values of the different memristors (Fig. 1, 114) and therefore ascertain whether a 1 or a 0 is indicated. In this fashion, the processor (Fig. 1, 101) may ascertain the string of 0s and 1s indicated in the memristor array (Fig. 1, 113).

[0055] The method (200) also includes determining (block 203) that a number of switching parameters are to be adjusted. Determining (block 203) that a number of switching parameters are to be adjusted may indicate that the number of memristors (Fig. 1, 114) have drifted, past a particular threshold value, for example. In some examples, the threshold may indicate that an incorrect reading or writing of information is likely, or that no reading or writing of information is likely. In this example, any adjustment of the switching parameters may correct for the effects of a drift.

[0056] Determining (block 203) that a number of switching parameters are to be adjusted may be based on a subset of the number of memristors (Fig. 1, 114) in a memristor array (Fig. 1, 113). For example, a certain subset of the number of memristors (Fig. 1, 114) may be sampled to ascertain the condition of the subset. If the subset indicates that the memristors (Fig. 1, 114) within the subset have drifted, all of the memristors (Fig. 1, 114) in a memristor array (Fig. 1, 113) may be subsequently adjusted. In other words, the adjustment of the switching parameters may be an event-triggered operation. The event may indicate that at least a subset of the memristors (Fig. 1, 114) have drifted. Accordingly, the adjustment may be performed independent of a read or write operation. In another example, all of the memristors in a memristor array may be similarly sampled and adjusted.

[0057] In some examples, determining (block 203) that a number of switching parameters are to be adjusted may be a closed-loop determination. In this example, the memory manager (Fig. 1, 104) may include a feedback loop that acquires feedback indicating whether at least a subset, or all, of the memristors (Fig. 1, 114) have drifted. If the feedback loop indicates that the subset, or all, of the memristors (Fig. 1, 114) have drifted, the adjustment module (Fig. 1, 106) may adjust the switching parameters accordingly.

[0058] In another example, determining (block 203) that a number of switching parameters are to be adjusted may be an open-loop determination. In this example, a predefined condition of the memristors (Fig. 1, 114) may be established. The predefined condition may indicate a usage of a reference set of memristors (Fig. 1, 114). For example, a reference set of memristors (Fig. 1,

114) may indicate that after a certain period of time, or after a certain period of switching cycles or reads, a memristor (Fig. 1, 114) may exhibit characteristic drift. In this example, the memory manager (Fig. 1, 104) may include computer usable program code to, when executed by the processor (Fig. 1, 101), determine a characteristic drift for at least a subset of the number of memristors (Fig. 1, 114). The characteristic drift for the at least subset of memristors (Fig. 1, 114) may be based on identified characteristics of a reference set of memristors (Fig. 1, 114).

[0059] Based on the determination, the method (200) may include adjusting (block 204) a number of the switching parameters. For example, if determined that at least a subset, or all, of the memristors (Fig. 1, 114) in a memristor array (Fig. 1, 113) have drifted, an electrical pulse used to execute a switching event may be adjusted. Adjusting an electrical pulse may include adjusting a value, a duration, or other characteristic of the electrical pulse. An example is given as follows. In this example, a particular memristor (Fig. 1, 114) in a memristor array (Fig. 1, 113) has drifted such that the initial electrical pulse of 2.5V for a specified period of time no longer puts the memristor (Fig. 1, 114) at the target resistance value corresponding to a high resistance state. After the memory manager (Fig. 1, 104) determines (block 203) that this memristor (Fig. 1, 114) has drifted and therefore an adjustment of a switching parameter is desired, the adjustment module (Fig. 1, 105) may adjust the electrical pulse by increasing the value of the electrical pulse, increasing the duration of the electrical pulse, or some other adaptation such that the adjusted electrical pulse puts the memristor (Fig. 1, 114) to the target resistance value that corresponds to a high resistance state. Adjusting the electrical pulse may include measuring the responsive resistance value after applying an electrical pulse and adjusting the electrical pulse characteristics until the desired resistance is achieved.

[0060] Adjusting an electrical pulse may include adjusting a target resistance value that corresponds to a particular logical state. An example is given as follows. In this example, a particular memristor (Fig. 1, 114) in a memristor array (Fig. 1, 113) has drifted such that the initial resistance level of 150 k Ω , which is intended to correspond to the logical state of 0, no longer results

from the original reset pulse, but instead a different resistance level (such as 200 k Ω for example) is produced. This may lead to an error in subsequent retrieval of information and in erasing the memristor (Fig. 1, 114). Accordingly, in this example, the adjustment module (Fig. 1, 106) may adjust this switching parameter such that the different resistance level (i.e., the 200 k Ω) is now the target resistance value associated with a logical state of 0. Adjusting the target resistance value may result in a change to the read and write operations. For example, as the target resistance value or the electrical pulse is changed, the processor (Fig. 1, 101) may adjust its logic to incorporate the adjustments to the switching parameters. In both instances, adjusting the electrical pulse and adjusting the target resistance values, either may be performed as a closed-loop adjustment or an open-loop adjustment as described above.

[0061] In yet another example, adjusting (block 204) a number of the switching parameters may include reinitializing the memristor (Fig. 1, 114) such that the established switching parameters are valid. For example, as described above, drift may result in an initially established electrical pulse that no longer puts the memristor (Fig. 1, 114) at a specified target resistance value. Accordingly, the memristor (Fig. 1, 114) may be reinitialized such that the initial electrical pulse again puts the memristor (Fig. 1, 114) at the specified target resistance value and the specified target resistance value again indicates the particular logical state. Doing so may include adjusting a number of the characteristics of the number of memristors (Fig. 1, 114) within the memristor array (Fig. 1, 113). Reinitializing a number of the memristors (Fig. 1, 114) may include using an appropriate waveform to reset the memristors (Fig. 1, 114). Reinitializing the memristors (Fig. 1, 114) in a memristor array (Fig. 1, 113) may also include executing a sequence of operations for each of the memristors (Fig. 1, 114) within the memristor array (Fig. 1, 113). The method (200) may then include writing data to the memristor array (Fig. 1, 113) based on the adjusted switching parameters.

[0062] As described herein a number of examples have been given for adjusting (block 204) a number of the switching parameters. In some examples, the memory manager (Fig. 1, 104) may include a module to determine a

particular switching parameter to adjust (the electrical pulse, target resistance value, or characteristic of the memristor (Fig. 1, 114)) and may indicate which mechanism to use in adjusting (block 204) the number of switching parameters.

[0063] As described in Fig. 2 the adjustment (block 204) of the switching parameters may be performed on all memristors (Fig. 1, 114) within a memristor array (Fig. 1, 113) and may encompass more than just a memristor (Fig. 1, 114) that is a target of a specified read operation or write operation. Moreover, as will be described below in connection with Figs 4 and 5, the adjustment may be either periodic, based on characteristics of a reference set of memristors (Fig. 1, 114) (as indicated in Fig. 4) or may be event triggered based on feedback pertaining to a condition of a subset of memristors (Fig. 1, 114) (as indicated in Fig. 5).

[0064] The method (200) as described herein may be beneficial in that it allows for multiple ways to determine when and how to adjust switching parameters that define a switching event. Doing so may lead to more reliable switching events within an array and therefore more accurate and reliable data retrieval and storage.

[0065] Fig. 3 is a diagram of a memory manager (104) for adjusting switching parameters of a memristor array (Fig. 1, 113) according to one example of the principles described herein. The memory manager (104) may include an establish module (105) as described above in connection with Fig. 1. The memory manager (104) may also include an adjust module (106) as described above in connection with Fig. 1. The adjust module (106) may include an electrical pulse module (301) that adjusts the electrical pulse used to switch the resistance value of the memristor (Fig. 1, 114). The electrical pulse module (301) may adjust the electrical pulse by adjusting the value, duration, or combinations thereof of the electrical pulse. As described above, the electrical pulse may be a current pulse or a voltage pulse.

[0066] The adjust module (106) may include a resistance value module (302) that redefines the target resistance value that is associated with a particular logical state. The resistance value module (302) may communicate this adjusted target resistance value to the processor (Fig. 1, 101) such that the processor

(Fig. 1, 101) may accurately read data from and write data to the memristor array (Fig. 1, 113).

[0067] The adjust module (106) may also include a memristor reinitialization module (303) to reinitialize a memristor (Fig. 1, 114) such that 1) the initially established electrical pulse sets the memristor (Fig. 1, 114) to the initially established target resistance value and 2) the initially established target resistance value corresponds to the desired logical state. Reinitializing the memristor (Fig. 1, 114) may include resetting characteristics, such as material properties, of a memristor (Fig. 1, 114).

[0068] The memory manager (104) may include a feedback module (305) to receive feedback relative to the operation of the memristor array (Fig. 1, 113). For example, the feedback module (305) may receive information that indicates a condition of at least a subset of the memristors (Fig. 1, 114). As used herein, a condition may refer to an age of a memristor (Fig. 1, 114) or a number of write/read cycles performed on the memristor (Fig. 1, 114).

[0069] One example of feedback received may include an error rate as identified by an error correcting code. Feedback may include any system level triggers that indicate drift within a memristor array (Fig. 1, 113). Such system level triggers include a powering up of a computing device, unusual temperatures within the computing device, deployment of the computing device for a new user, etc. This information from the feedback module (305) may be send to the analyze module (304) to determine if and how to adjust the switching parameters.

[0070] The memory manager (104) may also include an analyze module (304) to analyze at least a subset of the number of memristors (Fig. 1, 114) in a memristor array (Fig. 1, 113) to determine when to adjust the number of the switching parameters. For example, as described above, the memory manager (104) may work in a closed-loop cycle such that a condition of at least a subset of the memristors (Fig. 1, 114) are analyzed to determine if a drift has likely occurred. More specifically, the analyze module (304) may analyze the usage, i.e., the age of the subset of memristors (Fig. 1, 114), the write cycles for the subset of memristors (Fig. 1, 114), etc., to determine whether there is likely a drift to the memristor array (Fig. 1, 113).

[0071] Fig. 4 is a flowchart of a method (400) for adjusting switching parameters of a memristor array (Fig. 1, 113) according to another example of the principles described herein. More specifically, Fig. 4 depicts a method (400) for a periodic adjustment of switching parameters. The switching parameters may be adjusted independent of a read and write operation and may be performed on more than a memristor (Fig. 1, 114) targeted for access by a processor (Fig. 1, 101). The method (400) includes establishing (block 401) switching parameters for memristor array (Fig. 1, 113). This may be performed as described in connection with Fig. 2. The method (400) includes establishing (block 402) a predefined condition for a number of memristors (Fig. 1, 114) in the memristor array (Fig. 1, 113). The predefined condition may be a condition which, if satisfied, triggers an adjustment of the switching parameters. The predefined condition may be based on a set of reference memristors (Fig. 1, 114). The predefined condition may be based on a life, or usage, of the reference memristors (Fig. 1, 114). For example, a set of reference memristors (Fig. 1, 114) may indicate that a lifespan of a memristor (Fig. 1, 114) is a certain number of write/read cycles. In this example, the predefined condition may be the lifespan indicated by the reference set of memristors (Fig. 1, 114). While a specific example of a predefined condition is given in the number of cycles of a reference set of memristors (Fig. 1, 114), the predefined condition may be any condition which indicates a likelihood of drift in a memristor array (Fig. 1, 113).

[0072] The method (400) includes writing (block 403) data to the memristor array (Fig. 1, 113) based on the switching parameters. This may be performed as described above in connection with Fig. 2. The method (400) includes determining (block 404) that a number of switching parameters are to be adjusted. This may be performed as described above in connection with Fig. 2. The method may include adjusting (block 405) a number of the switching parameters. For example, when at least a subset of the memristors (Fig. 1, 114) in the memristor array (Fig. 1, 113) have satisfied the predefined condition (i.e., the number of cycles indicated by the reference set of memristors (Fig. 1, 114)), the adjust module (Fig. 1, 106) may adjust (block 405) the number of switching parameters.

[0073] Fig. 5 is a flowchart of a method (500) for adjusting switching parameters of a memristor array (Fig. 1, 113) according to another example of the principles described herein. More specifically, Fig. 5 depicts a method (500) for an event-based adjustment of switching parameters, the event relating to a condition, or lifespan of the memristor. The switching parameters may be adjusted independent of a read and write operation and may be performed on more than a memristor (Fig. 1, 114) targeted for access by a processor (Fig. 1, 101). The method (500) includes establishing (block 501) switching parameters for a memristor array (Fig. 1, 113). This may be performed as described in connection with Fig. 2. The method (500) includes writing (block 502) data to the memristor array (Fig. 1, 113) based on the switching parameters. This may be performed as described above in connection with Fig. 2.

[0074] The method (500) includes receiving (block 503) feedback pertaining to a condition of at least a subset of the number of memristors (Fig. 1, 114). For example, in a closed-loop operation, feedback indicating a condition (i.e., age, cycles) of at least a subset of the memristors (Fig. 1, 114) can be used to determine (block 504) whether a number of switching parameters are to be adjusted. As described, at least a subset of the number of memristors (Fig. 1, 114) may be used to receive feedback. In this example, if memristors (Fig. 1, 114) in the memristor array (Fig. 1, 113) share similar characteristics, then all memristors (Fig. 1, 114) in the memristor array (Fig. 1, 113) may be adjusted based on the received feedback from the subset. Additionally, while this example describes receiving feedback from a subset of the memristors (Fig. 1, 114), in some examples, feedback may be received from all memristors (Fig. 1, 114) in the memristor array (Fig. 1, 113).

[0075] The method (500) includes determining (block 504) that a number of switching parameters are to be adjusted. This may be performed as described above in connection with Fig. 2. The method may include adjusting (block 505) a number of the switching parameters. For example, when the feedback indicates that drift is likely, based on the age of the subset of memristors (Fig. 1, 114), the write cycles of the subset of memristors (Fig. 1,

114), etc., the adjust module (Fig. 1, 106) may adjust (block 505) the number of switching parameters.

[0076] Fig. 6 is a diagram of a memory manager (104) for adjusting switching parameters of memristors (Fig. 1, 114) in a memristor array (Fig. 1, 113) according to another example of the principles described herein. The memory manager (104) may include the hardware architecture to retrieve executable code and execute the executable code. The executable code may, when executed by the memory manager (104), cause the memory manager (104) to implement at least the functionality of adjusting switching parameters for memristors (Fig. 1, 114) in a memristor array (Fig. 1, 113), according to the methods of the present specification described herein. In the course of executing code, the memory manager (104) may receive input from and provide output to the remaining hardware units.

[0077] In this example, the memory manager (104) may include processing resources (601) that are in communication with memory resources (602). Processing resources (601) may include at least one processor and other resources used to process programmed instructions. The memory resources (602) represent generally any memory capable of storing data such as programmed instructions or data structures used by the memory manager (104). The programmed instructions shown stored in the memory resources (602) may include a switching parameters establisher (603), a data writer (604), an adjustment determiner (605), a parameter adjuster (606), a predefined condition establisher (607) and a feedback receiver (608).

[0078] The memory resources (602) include a computer readable storage medium that contains computer readable program code to cause tasks to be executed by the processing resources (601). The computer readable storage medium may be tangible and/or physical storage medium. The computer readable storage medium may be any appropriate storage medium that is not a transmission storage medium. A non-exhaustive list of computer readable storage medium types includes non-volatile memory, volatile memory, random access memory, write only memory, flash memory, electrically erasable program read only memory, or types of memory, or combinations thereof.

[0079] The switching parameters establisher (603) represents programmed instructions that, when executed, cause the processing resources (601) to establish switching parameters for a number of memristors (Fig. 1, 114) in the memristor array (Fig. 1, 113). The switching parameters establisher (603) may be implemented by the establish module (Fig. 1, 105). The data writer (604) represents programmed instructions that, when executed, cause the processing resources (601) to write data to the memristor array (Fig. 1, 113) based on the number of switching parameters, the adjusted switching parameters, or combinations thereof. The adjustment determiner (605) represents programmed instructions that, when executed, cause the processing resources (601) to determine that a number of switching parameters are to be adjusted. The adjustment determiner (605) may determine that a number of switching parameters are to be adjusted based on at least a subset of the number of memristors (Fig. 1, 114) satisfying a predefined condition.

[0080] The parameter adjuster (606) represents programmed instructions that, when executed, cause the processing resources (601) to adjust the number of switching parameters. The parameter adjuster (606) may be implemented by the adjust module (Fig. 1, 106). The predefined condition establisher (607) represents programmed instructions that, when executed, cause the processing resources (601) to establish the predefined condition based on usage of a reference set of memristors (Fig. 1, 114). The feedback receiver (608) represents programmed instructions that, when executed, cause the processing resources (601) to receive feedback pertaining to a condition of at least a subset of the number of memristors (Fig. 1, 114). The feedback receiver (608) may be implemented by the feedback module (305).

[0081] Further, the memory resources (602) may be part of an installation package. In response to installing the installation package, the programmed instructions of the memory resources (602) may be downloaded from the installation package's source, such as a portable medium, a server, a remote network location, another location, or combinations thereof. Portable memory media that are compatible with the principles described herein include DVDs, CDs, flash memory, portable disks, magnetic disks, optical disks, other

forms of portable memory, or combinations thereof. In other examples, the program instructions are already installed. Here, the memory resources can include integrated memory such as a hard drive, a solid state hard drive, or the like.

[0082] In some examples, the processing resources (601) and the memory resources (602) are located within the same physical component, such as a server, or a network component. The memory resources (602) may be part of the physical component's main memory, caches, registers, non-volatile memory, or elsewhere in the physical component's memory hierarchy. Alternatively, the memory resources (602) may be in communication with the processing resources (601) over a network. Further, the data structures, such as the libraries, may be accessed from a remote location over a network connection while the programmed instructions are located locally. Thus, the memory manager (104) may be implemented on a user device, on a server, on a collection of servers, or combinations thereof.

[0083] Methods and systems for adjusting switching parameters of memristors in a memristor array may have a number of advantages, including: (1) improving memristor array (Fig. 1, 113) lifetime; (2) maintaining a memristor array (Fig. 1, 113) in an optimal operating range; (3) minimizing variability between memristors (Fig. 1, 114); (4) improving reliability of the memristor array (Fig. 1, 113); (5) reducing bit error rates; (6) increasing endurance; (7) improve consistency of read and write times; and (8) constrain worst case power consumption.

[0084] Aspects of the present system and method are described herein with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems) and computer program products according to examples of the principles described herein. Each block of the flowchart illustrations and block diagrams, and combinations of blocks in the flowchart illustrations and block diagrams, may be implemented by computer usable program code. The computer usable program code may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the computer usable

program code, when executed via, for example, the processing resources (601) or other programmable data processing apparatus, implement the functions or acts specified in the flowchart and/or block diagram block or blocks. In one example, the computer usable program code may be embodied within a computer readable storage medium; the computer readable storage medium being part of the computer program product. In one example, the computer readable storage medium is a non-transitory computer readable medium.

[0085] The preceding description has been presented to illustrate and describe examples of the principles described. This description is not intended to be exhaustive or to limit these principles to any precise form disclosed. Many modifications and variations are possible in light of the above teaching.

CLAIMS

WHAT IS CLAIMED IS:

1. A system for adjusting switching parameters of a memristor array, comprising:
 - a processor;
 - a memristor array; and
 - a memory manager communicatively coupled to the processor and memristor array, the memory manager comprising:
 - an establish module to establish switching parameters for the memristor array; and
 - an adjustment module to adjust a number of the switching parameters.
2. The system of claim 1, further comprising an analyze module to analyze at least a subset of the number of memristors to determine when to adjust the number of the switching parameters.
3. The system of claim 2, in which the analyze module determines when to adjust the number of memristors based on a usage of the subset of the number of memristors analyzed.
4. The system of claim 1, further comprising a feedback module to receive feedback about a condition of at least a subset of the number of memristors within the memristor array.
5. A method for adjusting switching parameters of a memristor array, comprising, with a processor:

establishing switching parameters for the memristor array;
writing data to the memristor array based on the switching parameters;
determining that a number of the switching parameters are to be adjusted;
and
adjusting the number of the switching parameters determined to need
adjusting.

6. The method of claim 5, in which determining that a number of the switching parameters are to be adjusted comprises determining that at least a subset of the number of memristors have reached a predefined condition.

7. The method of claim 6, further comprising establishing the predefined condition based on usage of a reference set of memristors.

8. The method of claim 5, further comprising receiving feedback pertaining to a condition of at least a subset of the number of memristors.

9. The method of claim 5, in which adjusting the number of switching parameters comprises adjusting characteristics of a number of electrical pulses used to switch a number of memristors in the memristor array between a number of logical states.

10. The method of claim 9, in which adjusting characteristics of the number of electrical pulses comprises adjusting a value, a duration, or combinations thereof of the number of electrical pulses.

11. The method of claim 5, in which adjusting the number of switching parameters comprises adjusting a number of target resistance values that correspond to a number of logical states.

12. The method of claim 5, in which adjusting the number of switching parameters comprises reinitializing a number of characteristics of a number of memristors in the memristor array.

13. A computer program product for adjusting switching parameters of a memristor array, the computer program product comprising:

a computer readable storage medium comprising computer usable program code embodied therewith, the computer usable program code comprising:

computer usable program code to, when executed by a processor, establish switching parameters for the memristor array;

computer usable program code to, when executed by a processor, write data to the memristor array based on the switching parameters;

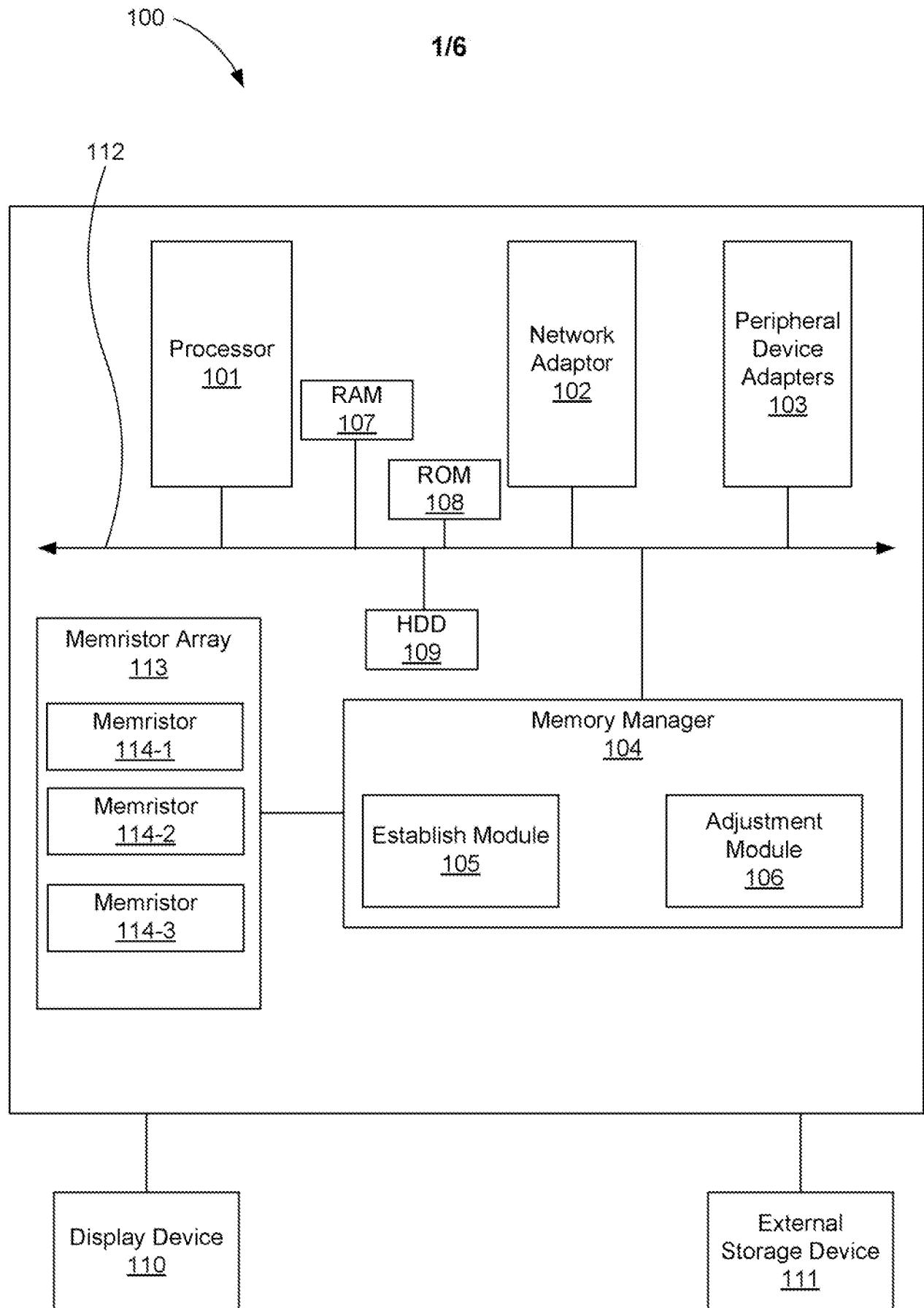
computer usable program code to, when executed by a processor, determine that a number of the switching parameters are to be adjusted;

computer usable program code to, when executed by a processor, adjust the number of the switching parameters; and

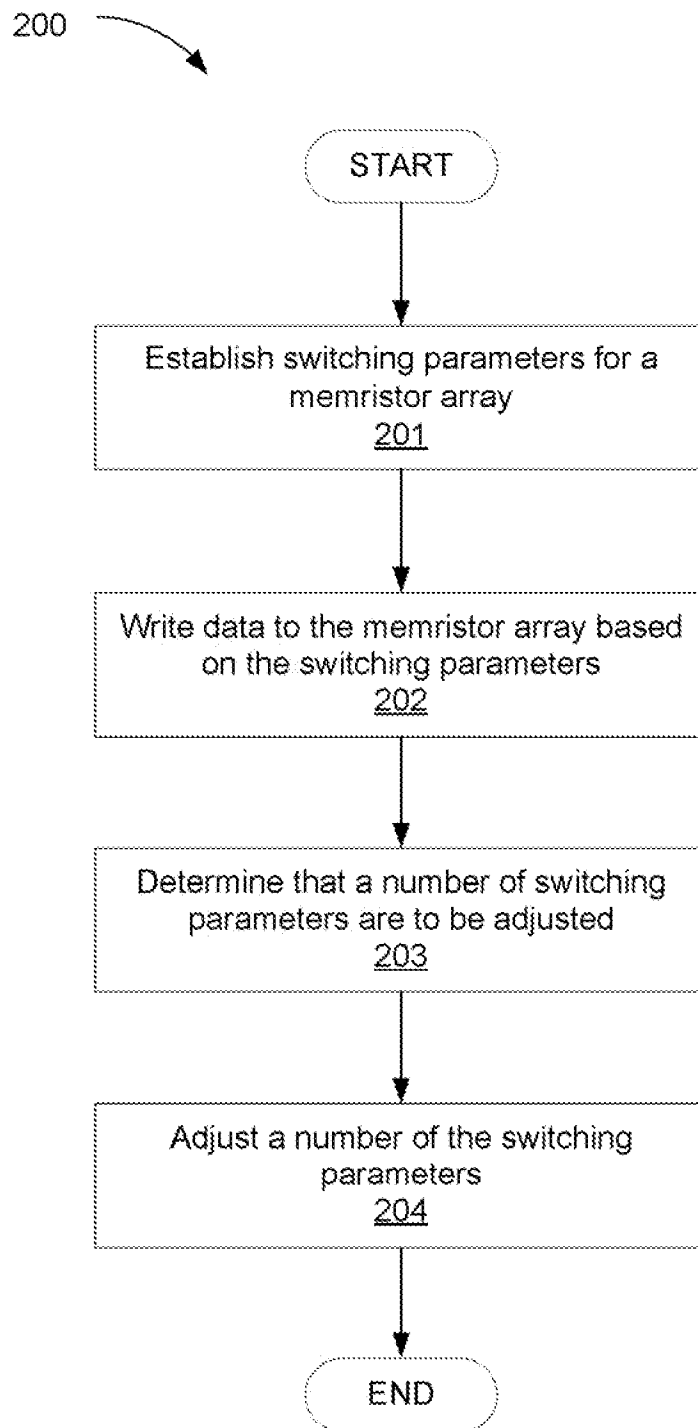
computer usable program code to, when executed by a processor, write data to the memristor array based on the adjusted switching parameters.

14. The computer program product of claim 13, further comprising computer usable program code to, when executed by a processor, determine a characteristic drift for at least a subset of the number of memristors in the memristor array.

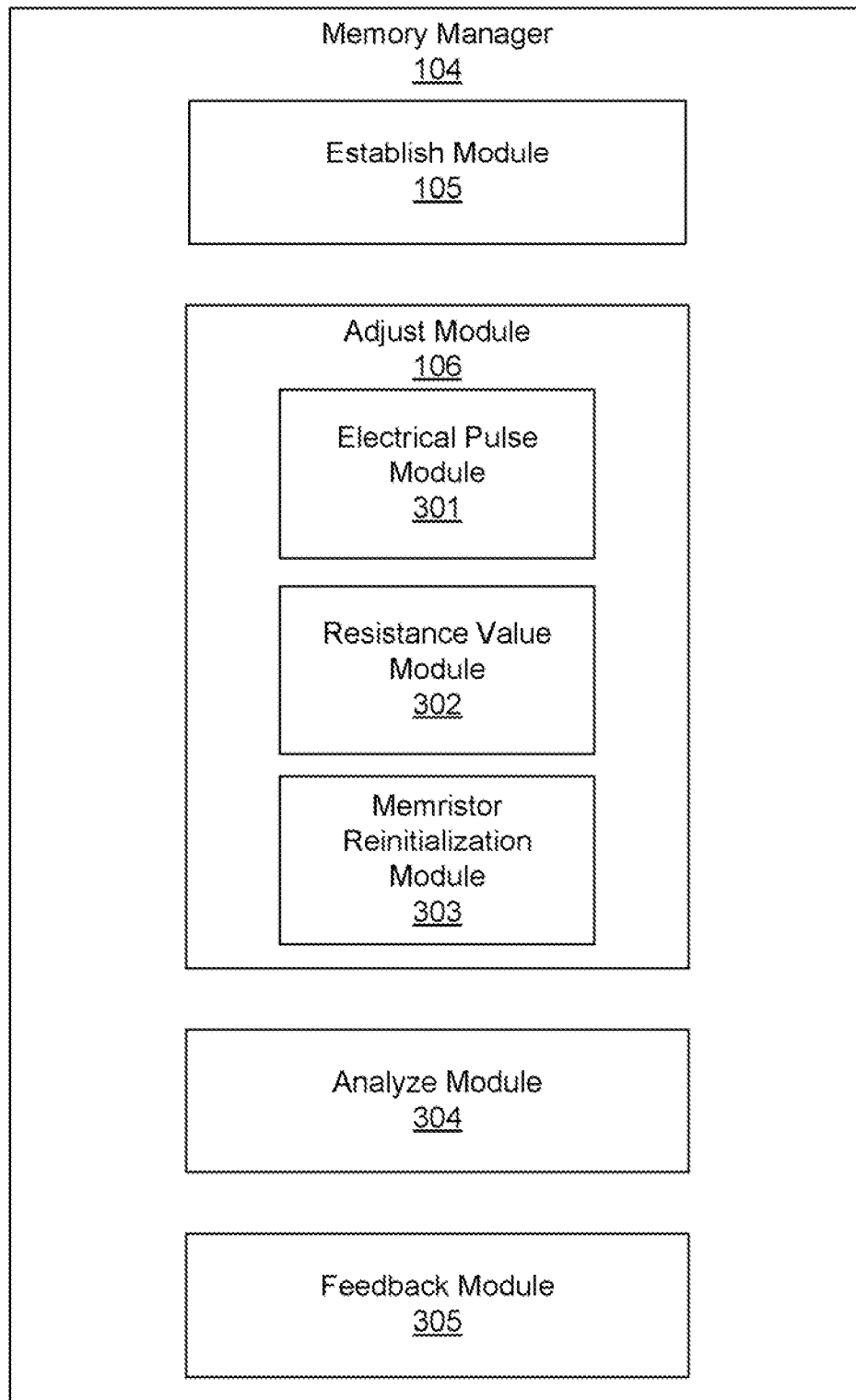
15. The computer program product of claim 13, in which the memristor array is a cross bar memristor array.

**Fig. 1**

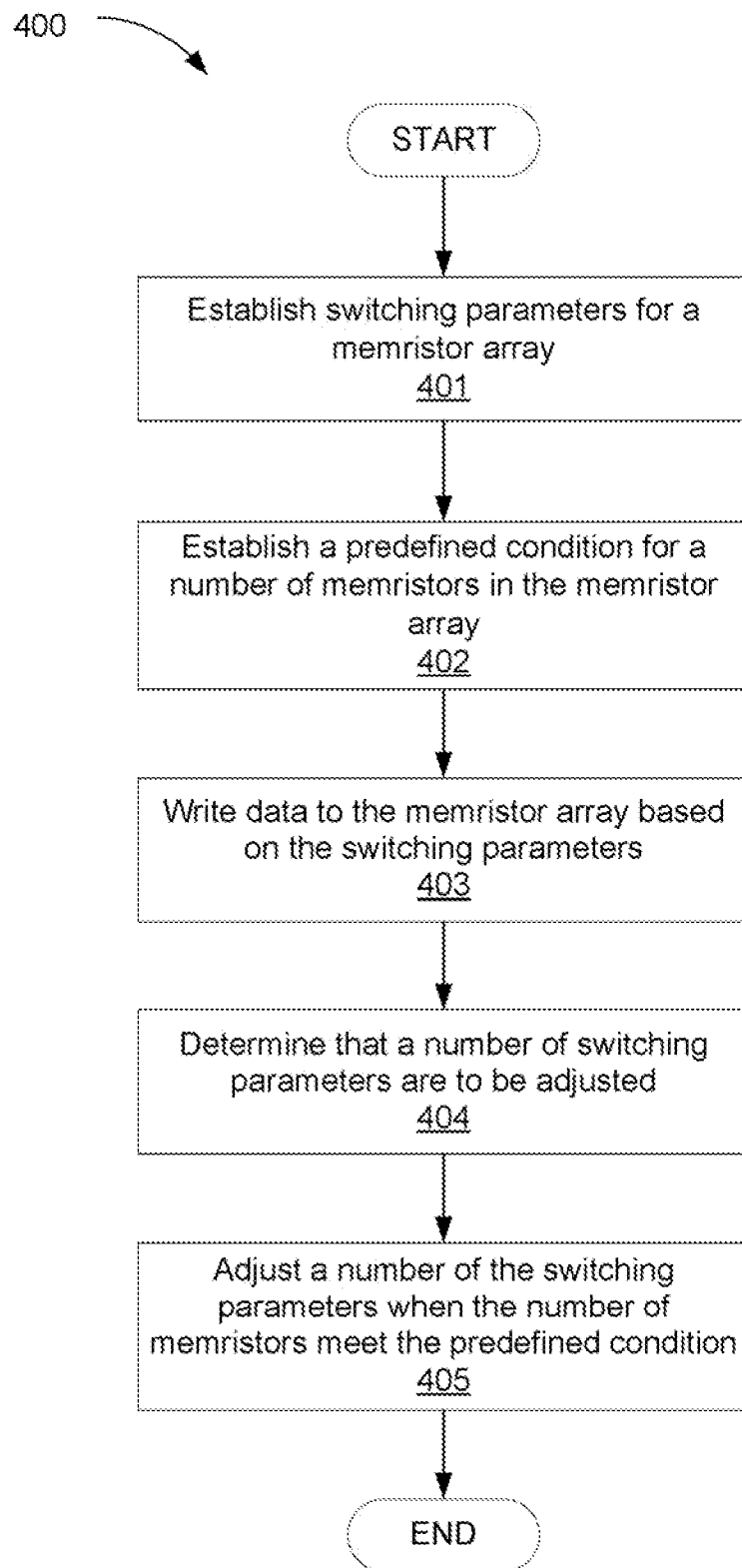
2/6

**Fig. 2**

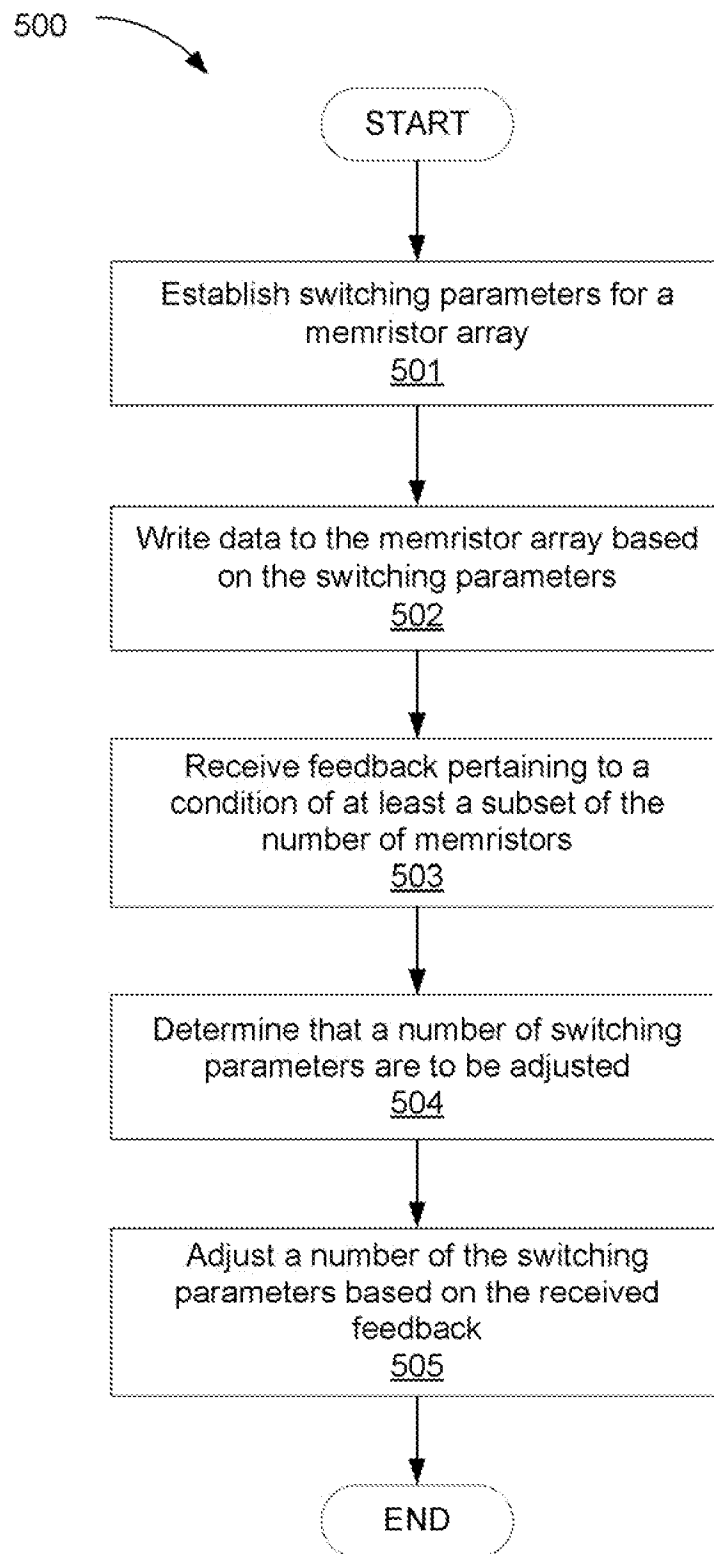
3/6

***Fig. 3***

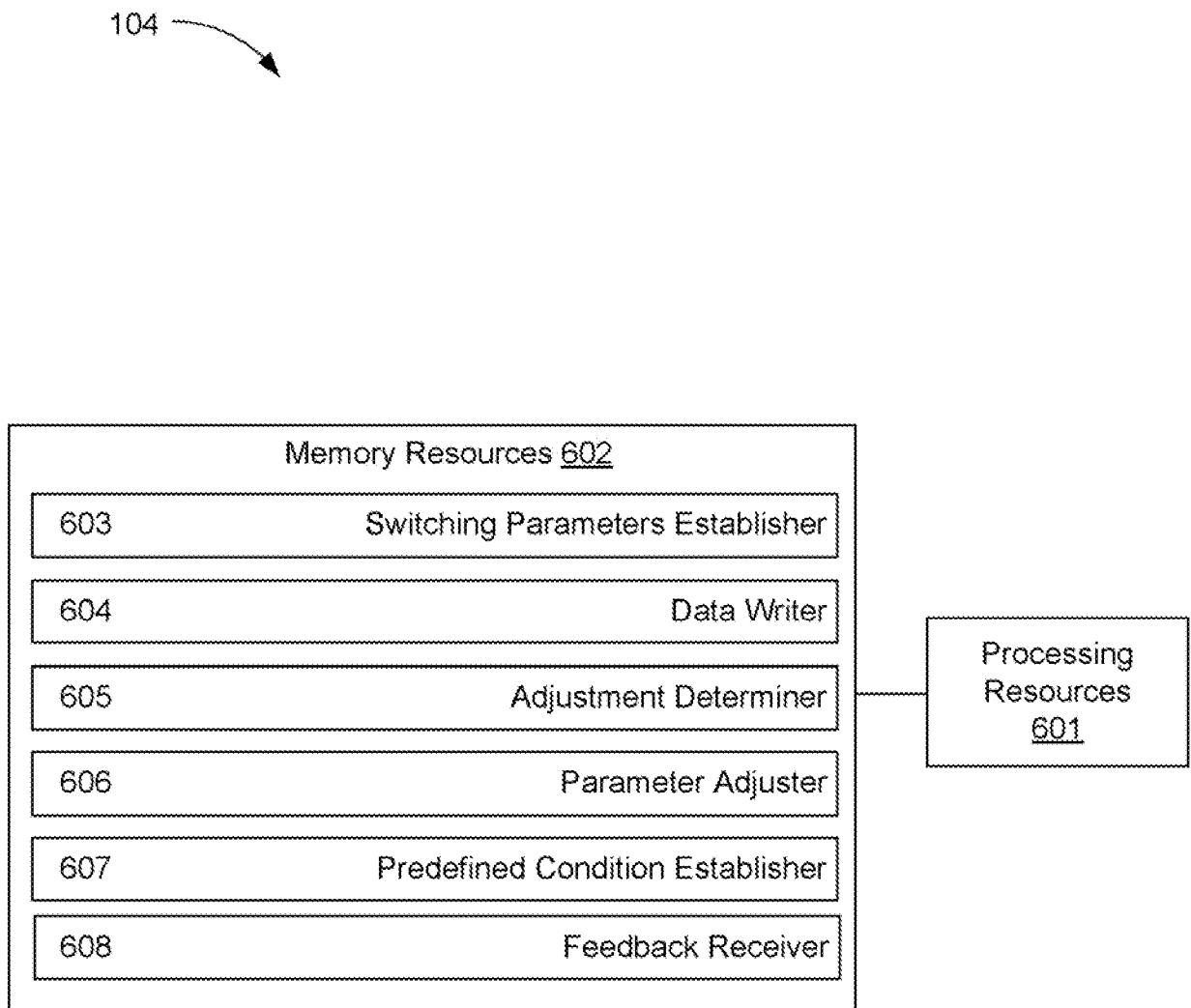
4/6

**Fig. 4**

5/6

**Fig. 5**

6/6

**Fig. 6**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/048449**A. CLASSIFICATION OF SUBJECT MATTER****G06F 12/00(2006.01)i, G06F 13/14(2006.01)i**

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 12/00; G06F 12/02; H01L 31/0336; A61B 5/0476; G11C 11/00; G06F 13/14

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) & Keywords:memristor, parameter, switching, adjust, memory, switch, resistance.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2012-0127780 A1 (JOHN PAUL STRACHAN et al.) 24 May 2012 See paragraphs [0010]-[0018], [0031]-[0038]; claims 1-6; and figures 1-4.	1-15
A	US 2005-0127403 A1 (SHENG TENG HSU) 16 June 2005 See paragraphs [0120]-[0128]; claim 20; and figure 10.	1-15
A	US 2012-0047313 A1 (SUYASH SINHA et al.) 23 February 2012 See paragraphs [0021]-[0023], [0043]-[0063]; claims 1-8; and figure 5.	1-15
A	US 2009-0163826 A1 (MOUTTET BLAISE LAURENT) 25 June 2009 See paragraphs [0029]-[0033]; and figures 4-7.	1-15
A	US 2007-0109835 A1 (SHENG TENG HSU) 17 May 2007 See paragraphs [0016]-[0018]; and figures 3-4.	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

17 March 2015 (17.03.2015)

Date of mailing of the international search report

17 March 2015 (17.03.2015)

Name and mailing address of the ISA/KR

International Application Division
Korean Intellectual Property Office
189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan City, 302-701,
Republic of Korea

Facsimile No. ++82 42 472 7140

Authorized officer

BYUN, Sung Cheal

Telephone No. +82-42-481-8262



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2014/048449

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0127780 A1	24/05/2012	WO 2011-099961 A1	18/08/2011
US 2005-0127403 A1	16/06/2005	DE 602004013391 T2	10/06/2009
		EP 1460637 A1	22/09/2004
		EP 1460637 B1	30/04/2008
		JP 2004-273110 A	30/09/2004
		JP 4187673 B2	26/11/2008
		KR 10-0601806 B1	19/07/2006
		KR 10-2004-0080357 A	18/09/2004
		TW 248088 B	21/01/2006
		US 2004-0179414 A1	16/09/2004
		US 6868025 B2	15/03/2005
		US 6967884 B2	22/11/2005
US 2012-0047313 A1	23/02/2012	None	
US 2009-0163826 A1	25/06/2009	US 2007-0229111 A1	04/10/2007
		US 2007-0229112 A1	04/10/2007
		US 2007-0229121 A1	04/10/2007
		US 2007-0231972 A1	04/10/2007
		US 2007-0241942 A1	18/10/2007
		US 2007-0257704 A1	08/11/2007
		US 2008-0059688 A1	06/03/2008
		US 2008-0172385 A1	17/07/2008
		US 2008-0212382 A1	04/09/2008
		US 2008-0222342 A1	11/09/2008
		US 2008-0246116 A1	09/10/2008
		US 2008-0307151 A1	11/12/2008
		US 7302513 B2	27/11/2007
		US 7342413 B2	11/03/2008
		US 7378870 B2	27/05/2008
		US 7391235 B2	24/06/2008
		US 7447828 B2	04/11/2008
		US 7459933 B2	02/12/2008
		US 7564262 B2	21/07/2009
		US 7576565 B2	18/08/2009
		US 7609086 B2	27/10/2009
		US 7755424 B2	13/07/2010
		US 7902867 B2	08/03/2011
		US 8183554 B2	22/05/2012
US 2007-0109835 A1	17/05/2007	US 7236389 B2	26/06/2007