



- (51) **International Patent Classification:**
G11C 11/15 (2006.01) *G11C 13/00* (2006.01)
- (21) **International Application Number:**
PCT/US2015/066371
- (22) **International Filing Date:**
17 December 2015 (17.12.2015)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP [US/US]; Hewlett Packard Enterprise Development LP, 11445 Compaq Center Drive West, Houston, Texas 77070 (US).
- (72) **Inventors:** HU, Miao; 1501 Page Mill Road, Palo Alto, California 94304 (US). STRACHAN, John Paul; 1501 Page Mill Road, Palo Alto, California 94304 (US). LI, Zhiyong; 1501 Page Mill Road, Palo Alto, California 94304 (US). WILLIAMS, Stanley; 1501 Page Mill Road, Palo Alto, California 94304 (US).
- (74) **Agents:** PAGAR, Preetam B. et al.; Hewlett Packard Enterprise, 3404 East Harmony Road, Mail Stop 79, 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, 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, ST, 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:

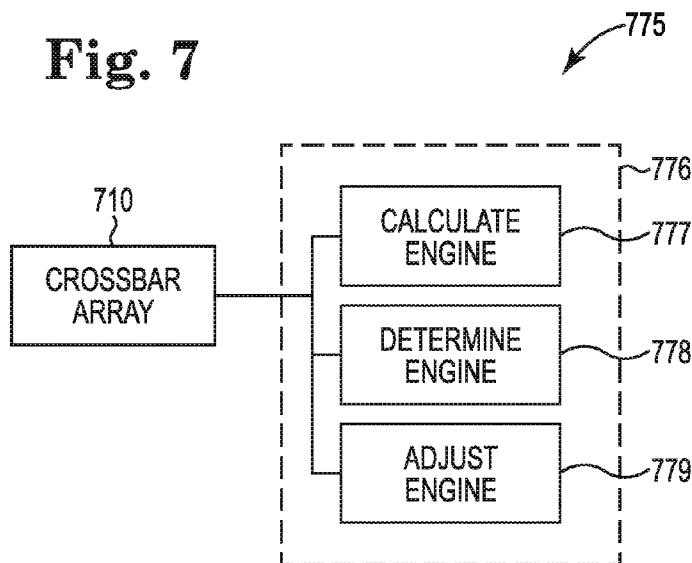
- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

Published:

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

(54) **Title:** IMPROVED COMPUTATIONAL ACCURACY IN A CROSSBAR ARRAY

Fig. 7



(57) **Abstract:** Example implementations of the present disclosure relate to improved computational accuracy in a crossbar array. An example system may include a crossbar array, having a plurality of memory elements at junctions, usable in performance of computations. The example system may further include a calculate engine to calculate ideal conductance of memory elements at a plurality of junctions of the crossbar array and a determine engine to determine conductance of the memory elements at the plurality of junctions of the crossbar array. An adjust engine of the example system may be used to adjust conductance of at least one memory element to improve computational accuracy by reduction of a difference between the ideal conductance and the determined conductance of the at least one memory element.



IMPROVED COMPUTATIONAL ACCURACY IN A CROSSBAR ARRAY

Background

[0001] A crossbar array may be inherently efficient for parallel signal processing because of a compact integrated structure. For example, an ideal crossbar structure may be used to accurately perform, via Kirchhoff's Current Law, vector-matrix multiplication between input vectors and data values (weights) stored by memory elements in a matrix of the crossbar array.

Brief Description of the Drawings

[0002] Figure 1 illustrates a diagram of an example of a computing system for improved computational accuracy in a crossbar array according to the present disclosure.

[0003] Figure 2 illustrates a diagram of an example of a crossbar array for improved computational accuracy in a crossbar array according to the present disclosure.

[0004] Figure 3 illustrates a circuit diagram of an example of a crossbar array for improved computational accuracy in a crossbar array according to the present disclosure.

[0005] Figure 4 illustrates a diagram of an example of nonlinearity of output of an actual crossbar array compared to an ideal crossbar array according to the present disclosure.

[0006] Figure 5 illustrates a flow chart of an example process for improved computational accuracy in a crossbar array according to the present disclosure.

[0007] Figure 6 illustrates a diagram of an example of improved computational accuracy in a crossbar array with adjusted conductance of memory elements according to the present disclosure.

[0008] Figure 7 illustrates a diagram of an example of a system for improved computational accuracy in a crossbar array according to the present disclosure.

[0009] Figure 8 illustrates a diagram of an example computing device for improved computational accuracy in a crossbar array according to the present disclosure.

[0010] Figure 9 illustrates a flow diagram of an example method for improved computational accuracy in a crossbar array according to the present disclosure.

Detailed Description

[0011] Computations using real crossbar circuits (arrays) may have notable computational inaccuracy (errors) relative to target results, such as results determined from input of test data with known correct output and/or results obtained from a simulated (ideal) crossbar array, etc. Such computational inaccuracy may be caused by behavior of the actual crossbar array being affected by, for example, increasing wire resistance that may correlate with distance of a column from an input signal in a row, resistance for signal input and/or output, inherent device nonlinearity, thermal effects, noise, and/or other parasitics, etc. Such influences on behavior of an actual crossbar array may be difficult to compensate for during the computations and/or data processing. As such, the present disclosure relates to improved computational accuracy in a crossbar array.

[0012] An example system may include a crossbar array, having a plurality of memory elements at junctions, usable in performance of computations. The example system may further include a calculate engine to calculate ideal conductance of memory elements at a plurality of junctions of the crossbar array and a determine engine to determine

conductance of the memory elements at the plurality of junctions of the crossbar array. An adjust engine of the example system may be used to adjust conductance of at least one memory element to improve computational accuracy by reduction of a difference between the ideal conductance and the determined conductance of the at least one memory element.

[0013] Figure 1 illustrates a diagram of an example of a computing system for improved computational accuracy in a crossbar array according to the present disclosure. Figure 1 is a diagram of a computing system 100 for determining a dot product, according to one example of the principles described herein. The computing system 100 may be implemented in an electronic device. Examples of electronic devices include servers, desktop computers, laptop computers, personal digital assistants (PDAs), mobile devices, smartphones, gaming systems, and/or tablets, among other electronic devices.

[0014] The computing system 100 may be utilized in any data processing scenario including stand-alone hardware, mobile applications, through a computing network, or combinations thereof. Further, the computing system 100 may be used in a computing network, a public cloud network, a private cloud network, a hybrid cloud network, other forms of networks, or combinations thereof. The systems and methods described herein may be provided by the computing system 100 are provided as a service over a network by, for example, a third party. In this example, the service may include, for example, the following: a Software as a Service (SaaS) hosting a number of applications; a Platform as a Service (PaaS) hosting a computing platform including, for example, operating systems, hardware, and storage, among others; an Infrastructure as a Service (IaaS) hosting equipment such as, for example, servers, storage components, network, and components, among others; application program interface (API) as a service (APIaaS), other forms of network services, or combinations thereof. The systems present herein may be implemented on one or multiple hardware platforms, in which the modules in the system may be executed on one or across multiple platforms. Such modules may run on various forms of cloud technologies

and hybrid cloud technologies or be offered as a SaaS (Software as a service) that may be implemented on or off the cloud. In another example, the methods provided by the computing system 100 may be executed by a local administrator.

[0015] To achieve its intended functionality, the computing system 100 may include various hardware components. Among these hardware components may be a number of processors 101, a number of data storage devices 102, a number of peripheral device adapters 103, and/or a number of network adapters 104. These hardware components may be interconnected through use of a number of busses and/or network connections. For example, the processor 101, data storage device 102, peripheral device adapters 103, and/or a network adapter 104 may be communicatively coupled via a bus 105.

[0016] The processor 101 may include the hardware architecture to retrieve executable code from the data storage device 102 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, for example, of applying a number of first voltages to a corresponding number of row lines (wires) within a memristive crossbar array to change the resistive values of a corresponding number of memristors located at junctions between the row wires and a number of column lines (wires), the first voltages representing a corresponding number of values within a matrix, respectively. The executable code may, when executed by the processor 101, also cause the processor 101 to implement at least the functionality of applying a number of second voltages to a corresponding number of row wires within a memristive crossbar array, the second voltages representing a corresponding number of vector values. The executable code may, when executed by the processor 101, further cause the processor 101 to implement at least the functionality of collecting the output currents from the column wires, the collected output currents representing the dot product. In the course of executing code, the processor 101 may receive input from and and/or provide output to a number of the remaining hardware units.

[0017] The data storage device 102 may store data such as executable program code that is executable by the processor 101 or other processing device. The data storage device 102 may specifically store computer code representing a number of applications that the processor 101 may execute to implement at least the functionality described herein.

[0018] The data storage device 102 may include various types of memory modules, including volatile and nonvolatile memory. For example, the data storage device 102 of the present example includes RAM 106, ROM 107, and/or HDD memory 108. Many other types of memory may also be utilized, and the present disclosure contemplates the use of as many varying type(s) of memory in the data storage device 102 as may suit a particular application of the principles described herein. In various examples, different types of memory in the data storage device 102 may be used for different data storage needs. For example, in certain examples the processor 101 may boot from ROM 107, maintain nonvolatile storage in the HDD memory 108, and/or execute program code stored in RAM 106, among other potential uses of memory types.

[0019] The data storage device 102 may include a computer-readable medium, a computer-readable storage medium, and/or a non-transitory computer-readable medium, among others. For example, the data storage device 102 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 RAM, a ROM, an erasable programmable ROM (EPROM or Flash memory), a portable compact disc ROM (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this disclosure, a computer-readable storage medium may be any tangible medium that can contain, or store computer-usable program code 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.

[0020] The hardware adapters 103, 104 in the computing system 100 may enable the processor 101 to interface with various other hardware elements, external and internal to the computing system 100. For example, the peripheral device adapters 103 may provide an interface to input/output devices, such as, for example, display device 109, a mouse, or a keyboard (not shown). The peripheral device adapters 103 may also provide access to other external devices such as an external storage device, a number of network devices such as, for example, servers, switches, and/or routers, client devices, other types of computing devices, and combinations thereof (not shown).

[0021] The display device 109 may be provided to allow a user of the computing system 100 to interact with and/or to implement the functionality of the computing system 100. The peripheral device adapters 103 may also create an interface between the processor 101 and the display device 109, a printer, or other media output devices. The network adapter 104 may provide an interface to other computing devices within, for example, a network, thereby enabling the transmission of data between the computing system 100 and other devices located within the network.

[0022] The computing system 100 may, when executed by the processor 101, display the number of graphical user interfaces (GUIs) on the display device 109 associated with the executable program code representing the number of applications stored on the data storage device 102. The GUIs may display, for example, interactive screenshots that allow a user to interact with the computing system 100 to input data values to a crossbar array 110 for various computation operations. For example, matrix values and input vector values may, in some embodiments, be stored in and/or input to the crossbar array 110 functioning as a dot product engine (DPE) for vector/matrix multiplication via the memory elements at junctions of the crossbar array. The memory elements in the crossbar array may, in some examples, be resistive memory (memristor) elements used in a memristor array, although embodiments are not limited

to such memristor arrays, as will be described in more detail below. Additionally, via interacting with the GUIs of the display device 109, a user may obtain a dot product value based on the input data. Examples of display devices 109 include a computer screen, a laptop screen, a mobile device screen, a personal digital assistant (PDA) screen, and/or a tablet screen, among other display devices.

[0023] The computing system 100 may further include a crossbar array (xbar) 110, which may correspond to the “actual crossbar array” described herein. In various examples, the crossbar array 110 may be implemented as a memristive crossbar array 210, as illustrated at 210 in Figure 2, and/or as a differential mode DPE memristive array, as illustrated at 320 in Figure 3.

[0024] As described below, the crossbar array 110 may include a number of elements, including a number of memristors that function together within the array to perform a weighted sum of multiple inputs. The differential mode DPE memristive array may, for example, be used in a number of applications. For example, the differential mode DPE memristive array may be used as a threshold logic gate (TLG) to perform a matrix product to compare the output with a threshold. Thus, the differential mode DPE memristive array may be used as an accelerator in which the differential mode DPE memristive array performs a number of functions faster than is possible in software running on a more general-purpose processing device.

[0025] Although the crossbar array 110 is depicted as being a device internal to the computing system 100, in another example, the crossbar array 110 may be a peripheral device coupled to the computing system 100 or included within a peripheral device coupled to the computing system 100.

[0026] The computing system 100 further includes a number of modules used in the implementation of the systems and methods described herein. The various modules within the computing system 100 include executable program code that 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

computing system 100 may be combined within a number of computer program products; each computer program product may include a number of the modules.

[0027] The computing system 100 may include a compute engine 111 to, when executed by the processor 101, assist in the functionality of the crossbar array 110. The compute engine 111 may, for example, assist in the crossbar array 110 functioning as a DPE by receiving a number of input values defining a matrix to be processed in a dot product mathematical computation. The compute engine 111 may send the input values to the crossbar array 110 as programming signals to a number of memory elements (e.g., memristors) within the crossbar array 110 to program the memristors. The compute engine 111 also may receive an input vector to be processed in connection with the matrix programmed into the memristors. The compute engine 111 may further obtain a value representing the dot product, and convey that value as data to the computing system 100 or another computing device for analysis or further processing.

[0028] Figure 2 illustrates a diagram of an example of a crossbar array for improved computational accuracy in a crossbar array according to the present disclosure. The crossbar array 210 illustrated in Figure 2 may be implemented with various types of memory elements and/or be used for various computational purposes within the scope of the present disclosure. For example, the crossbar array 210 may be a DPE array with memristor memory elements in the computing system 100 shown in Figure 1. The memristive crossbar array 210 may include a number of electrically conductive row wires 215 and a number of electrically conductive column wires 217. Even though three row wires 215 and four column wires 217 are depicted in Figure 2, any number of row wires 215 and column wires 217 may be present in the crossbar array 210. The row wires 215 and column wires 217 intersect at junctions as indicated by the dashed line 219. A number of memory elements 216 (e.g., resistive memory devices or “memristors”) are individually located at each of the junctions 219 and electrically couple the row wires 215 to the column wires 217.

[0029] “Memristance” is a property of the electronic component referred to as a memristor. A memristor is a resistor device whose resistance can be changed. For example, if charge flows in one direction through a circuit, the resistance of a memristor component of the circuit may increase. In contrast, if charge flows in the opposite direction in the circuit, the resistance of the memristor component may decrease. If the flow of charge is stopped by turning off the applied voltage, the memristor component will “remember” the last resistance that it had, and when the flow of charge starts again the resistance of the circuit will be what it was when it was last active.

[0030] Conductance channels (e.g., filaments) in the memristors 216 may be formed in each of the memristors 216 and the memristors 216 may be individually addressed as bits. A crossbar 210 is an array of switches that connect each wire in one set of parallel row wires 215 to every member of a second set of parallel column wires 217 that intersects the first set 215 at junctions 219. In the example of Figure 2, the row wires 215 and the column wires 217 are perpendicular with respect to each other, but the row wires 215 and the column wires 217 may intersect at any angle.

[0031] The memristors 216 may be formed at the micro- or nanoscale and may be used as memory element components in a wide variety of electronic circuits, such as, bases for memory and/or logic circuits and arrays. When used as memory, the memristors 216 may be used to store a bit of information (e.g., 1 or 0 in binary). When used as a logic circuit, the memristors 216 may be employed to represent bits in a field programmable gate array as the basis for a wired-logic programmable logic array, or, as described herein, as a DPE. The memristors 216 disclosed herein may also find use in a wide variety of other applications. The memristors 216 may be fabricated through any suitable fabrication process, for example, by chemical vapor deposition, sputtering, etching, lithography, and/or other suitable methods of fabricating memristors.

[0032] The memristive crossbar array 210 shown in Figure 2 further illustrates a number of input values 214 indicated, for example, as V1, V2,

and V3. The input values 214 may be program signals used to change the resistance values at each individual memristor 216 at each junction 219 in the crossbar array to create a representation (e.g., a mapping) of a mathematic matrix in which each value (e.g., a data value representing a digit) at each junction represents a value within the matrix. The change in resistance among the individual memristors 216 is an analog change from a low-to-high value or a high-to-low value. The change in the resistance value may correspond to an inverse of a change of a conductance value. In this manner, the memristors are “memory resistors” in that they “remember” the last resistance that they had.

[0033] The input values 214 may also be read signals used to read the resistance values at each individual memristor 216 at each junction 219 in the crossbar array 210, and/or as an input vector values to be multiplied by a stored matrix value at each memristor 216 involved in the calculation. The read signals referred to herein as a vector signals may be applied as second input values 214 to the row wires 215 of the memristive crossbar array 210. The input vector signals may represent a vector to be multiplied in the memristors 216 of the matrix. For example, the vector signals have a relatively lower voltage value than the first voltages used to program the memristors 216 such that the voltage level applied by the vector signals does not change the resistance values of the memristors 216 as programmed by the first voltages. The vector signals may act as read signals in this manner by not changing the resistance values of the memristors 216. An initial signal may, for example, be applied to the memristors 216 before application of the program signals and the vector signals in order to set the resistivity (conductivity) of the memristors 216 to a known value.

[0034] The vector signals may interact with the memristors 216 at their respective junctions 219, and the resulting current may be collected at the end 218 of each column wire 217. The sum current of each column wire 217 is represented by $\sum v_1 g_{11}$, $\sum v_1 g_{12}$, $\sum v_1 g_{13}$, and $\sum v_1 g_{14}$, as indicated in Figure 2, where g_{11-4} are the memristor 216 conductance values. The sums obtained from the voltage values represent the values

of a column vector. The sums of each column wire 217 may also be collectively summed by being sent through a collection line 328 to obtain a single value of $\sum_{i=1}^m lout_i$, as shown at 329 in Figure 3. A plurality of Vout and/or lout signal values may be converted to the summed single value of $\sum_{i=1}^m lout_i$ by a compute engine. Such a compute engine, as shown at and described in connection with 111 in Figure 1 and elsewhere, may contribute to the improved computational accuracy in a crossbar array as described herein.

[0035] Figure 3 illustrates a circuit diagram of an example of a crossbar array for improved computational accuracy in a crossbar array according to the present disclosure. The circuit diagram shown in Figure 3 illustrates a DPE memristive array 320, which may be fabricated as described in connection with the crossbar array 210 of Figure 2. For example, the circuit diagram shows the DPE memristive array 320 to be a differential mode DPE memristive array. Although the differential mode DPE memristive array 320 of Figure 3 is illustrated as having a circuit layout as depicted, any number of circuit layouts may be used to achieve the functionalities of the present disclosure.

[0036] The differential mode DPE memristive array 320 as depicted in Figure 3 includes the row wires 315, the column wires 317, and the memristors 316 as described in connection with Figure 2. As previously stated, any number of row wires 315 and column wires 317 may be included within the differential mode DPE memristive array 320 as indicated by the ellipses 321, 322. For example, the number of row wires 315 and column wires 317 included within the differential mode DPE memristive array 320 may be equal or greater than the size of the matrix of the dot product to be calculated using the systems and methods described herein.

[0037] The differential mode DPE memristive array 320 may further include a number of differential input circuits such as, for example, input amplifiers 325-1, 325-2, . . . , 325-N at the input terminals of the row wires 315. The input amplifiers 325 are designed to be differential amplifiers in order to make the DPE memristive array a differential mode DPE

memristive array 320. Although input amplifiers 325 are depicted in Figure 3, any type of differential input circuit may be used. The input amplifiers 325 shown are designed with negative feedback that subtracts a fraction of the output of the input amplifiers 325 from the input in order to improve gain stability, linearity, frequency response, and step response, and/or to reduce sensitivity to parameter variations due to manufacturing or the environment.

[0038] Input voltages (V_{in1} , V_{in2} , . . . , V_{in_n}) and reference voltages (V_{ref1} , V_{ref2} , . . . , V_{ref_n}) may be applied through separate input lines 324-1, 324-2, . . . , 324-N to the input amplifiers 325-1, 325-2, . . . , 325-N to obtain a dot product value. However, the crossbar arrays described herein as DPEs are not limited to being a differential mode DPE memristive array 320. For example, a DPE may be fabricated without the separate input lines 324 and/or the input amplifiers 325 such that input voltages can be directly input to the row wires 315.

[0039] For each output voltage of a DPE memristive array, the following relationship may exist: $V_{out} = -R_f \sum V_{in_i} \times \frac{1}{R_i}$, where V_{out} may correspond to V_{out1} , V_{out2} , . . . , V_{out_m} shown at 326-1, 326-2, . . . 326-M in Figure 3, V_{in_i} may correspond to V_{in1} , V_{in2} , . . . , V_{in_n} , R_f may be the input amplifier 325 feedback resistance and R_i may be the memristor 316 multilevel analog resistance predefined before calculation of a dot product.

[0040] Output voltages V_{out1} , V_{out2} , . . . , V_{out_m} shown at 326-1, 326-2, . . . 326-M, respectively, may, for example, be amplified by corresponding output amplifiers 327 to obtain a dot product value. The output voltage sums of each column wire 317 may be converted to current and, for example, sent through the collection line 328 to obtain the single current value of $\sum_{i=1}^m I_{out_i}$ shown at 329.

[0041] Memristors 316 are analog devices that may have nonlinearity in current output. For example, such nonlinearity may occur at large signal voltages. Memristors 316 may have better linearity when operating with smaller voltage signals. For example, operating a memristor 316 in the region of [0V, 1V] may result in greater output nonlinearity as compared to operating the memristor 316 in the [-1/2V,

1/2V] region. Such nonlinearity at large voltage levels may be compensated for by operating in a particular lower voltage range, for example.

[0042] In contrast, crossbar array structures may have other causes of voltage and/or current output nonlinearity, for example, as measured across a sequence of rows and/or a sequence of columns of the array, for which compensation may not be accomplished just by adjusting a voltage operating range. Such causes of output nonlinearity for crossbar arrays may include increasing wire resistance that, for example, may correlate with distance of a column from an input signal in a row, resistance for signal input and/or output, inherent device nonlinearity, thermal effects, noise, and/or other parasitics, which may reduce computational accuracy relative to target (ideal) results if the crossbar array is operated in a “naïve” way.

[0043] As used herein, the naïve way is intended to mean linearly mapping the matrix W to be used in the vector-matrix multiplication to conductance G of an ideal crossbar (e.g., not taking into account the causes of output nonlinearity just presented) to result in $I_{out} = V_{in}^T G$, where V_{in}^T is the transpose of the input voltage V_{in} . However, real current output may be different because of one or more of the causes of output nonlinearity just presented. To reduce the effects of these and other causes of output nonlinearity on computational accuracy, the present disclosure describes adjusting the actual conductance of memory elements (e.g., memristors) to have conductance closer to the ideal conductance of the memory elements in an ideal crossbar array in order to at least partially restore output linearity.

[0044] Figure 4 illustrates a diagram of an example of nonlinearity of output of an actual crossbar array compared to an ideal crossbar array according to the present disclosure. A graph 432 in Figure 4 illustrates a progressive decrease of actual current output 433 (in amperes (I_{out})) from an actual crossbar array, to which a corresponding progressive decrease in voltage output (V_{out}) may contribute. The progressive decrease of the actual current output 433 is shown to be related to a distance of a column wire (represented on the horizontal axis as a column number) from input

of a signal to an intersecting row. Respective distances may be of the column wires 317 shown in Figure 3, from a position of input, for example, from an input line 324 and/or an input amplifier 325, of input voltage V_{in} of an input vector to a row, for example, row wire 315.

[0045] For example, when a V_{in} of 0.5 volts (V) is input to a particular row wire, an actual Iout of around 0.4×10^{-3} amperes (A) may be measured for a column wire near columns 1-10 and an actual Iout of around 0.2×10^{-3} A may be measured for a column wire near columns 245-256. The decrease in actual Iout for column wires between the two ends of the row wire may be nonlinear, for example, by curving between columns 1-256. As illustrated in Figure 4, the actual current output 433 shows a concave curve between columns 1-256, although a convex curve or various combinations of convex and/or concave curves may contribute to the nonlinearity. The progressive decrease of the actual Iout in the crossbar array may be contributed to by a number of the causes described herein, including increasing wire resistance that correlates with the distance of the column from the V_{in} to the row.

[0046] A conductance of memory elements (e.g., memristors) at a plurality of junctions of the actual crossbar array may be determined, for example, by a determine engine. Measurements of the progressive decrease of the actual Iout corresponding to a number of rows and columns in the actual crossbar array may contribute to determination of the conductance of the memory elements at the junctions thereof. In some examples, actual conductance of some or all of the memory elements in the actual crossbar array may be directly or indirectly measured. Alternatively or in addition, a number of models of signal output variation in crossbar arrays may be used in determination of the conductance of the memory elements (e.g., in a simulated crossbar array and/or as applied to the actual crossbar array). The number of models may take into consideration with regard to determination of the conductance, for example, nonlinearity of wire resistance dependent upon distance from an input source in the crossbar array and/or other causes of signal output nonlinearity, input and/or output resistance, thermal effects, noise, and/or parasitics, etc.

[0047] By comparison, a range of ideal I_{out} values may be calculated, for example by a calculate engine, as described herein, for a simulated ideal crossbar array. The calculations of the ideal current output 434 may be performed by not taking into account the causes of I_{out} nonlinearity described herein. To contribute to calculation of the ideal current output 434, the ideal conductance of the memory elements (e.g., memristors) at a plurality of junctions of the crossbar array may be calculated. As such, when a V_{in} of 0.5 V is simulated as being input to a particular row wire, an ideal I_{out} of, for example, around 0.8×10^{-3} A may be calculated for the column wires throughout the sequence of columns 1-256.

[0048] The larger variability of ideal I_{out} for adjacent columns in the simulated ideal crossbar array, relative to the variability of the actual I_{out} for the same columns in the actual crossbar array, is not necessarily present in the calculations for the simulated ideal crossbar array. However, the larger variability may, for example, result from a randomness variable in the calculation of the ideal I_{out} that provides the larger variability based on the larger lower amperage of the ideal I_{out}, for example, relative to the amperage of the actual I_{out}.

[0049] The range of values for the ideal I_{out} sequence of the ideal crossbar array is relatively stable and may not show a notable upward or downward slope, thereby being suitable for linear mapping. In contrast, the range of values for the actual I_{out} sequence of the actual crossbar array may not be suitable for linear mapping because, for example, of a progressive downward slope of the actual I_{out} for the sequence of columns 1-256, which may be combined with various amounts of curvature.

[0050] Accordingly, as described herein, based at least partially on determination of conductance of the memory elements of the actual crossbar array, actual conductance of at least one memory element may be adjusted to improve computational accuracy by reduction of a difference between the ideal conductance and the determined conductance of the at least one memory element. In some examples, the actual conductance of a selected memory element (e.g., a memristor) may

be adjusted by application of a voltage value determined to adjust the conductance of the memory element to an intended level. In some examples, the actual conductance of a selected memory element may be adjusted by application of a sequence of voltage pulses to adjust the conductance of the memory element to the intended level. Control of how many voltage pulses to apply and/or monitoring when the intended level of conductance of the memory element has been reached may be performed, for example, using closed-loop voltage pulse tuning of the memory elements. In some examples, the closed-loop voltage pulse tuning may be controlled by a proportional-integral-derivative controller.

[0051] Figure 5 illustrates a flow chart of an example process for improved computational accuracy in a crossbar array according to the present disclosure. For clarity the process is illustrated as being separated into three portions, namely, determining 541, crossbar (xbar) programming 563, and computing 570. However, the process may be implemented as a single portion (e.g., a single sequence of functions) or any number of portions that may correspond to, for example, the number of engines and/or modules described herein or otherwise.

[0052] The process may begin 540 in the determining portion 541 with, in some examples, input of the matrix values 542 to determined (e.g., predetermined) memory elements of the xbar array. In some examples, the process may begin 540 without matrix values being input to memory elements of the xbar array. The process may proceed to a determination 543 of whether matrix data values have been stored by the memory elements of the xbar array.

[0053] When a determination is made that the matrix values 542 have been input to the xbar array (matrix), the xbar matrix may not be in a state suitable for performance of subsequent functions shown in the determining portion 541 and/or the xbar programming portion 563 of the process. As such, the process may skip the subsequent determining portion 541 and/or the xbar programming portion 563 to proceed to the computing portion 570, as described further herein.

[0054] In contrast, when a determination is made that no matrix values have been input to the xbar matrix, the xbar matrix may be

determined to be in a state suitable for performance of subsequent functions shown in the determining portion 541 and/or the xbar programming portion 563 of the process. As such, the configuration of the actual xbar matrix may be mapped to determine a corresponding configuration for an ideal (e.g., simulated) xbar matrix 544 for use in determining ideal conductance of memory elements at junctions in the ideal xbar matrix corresponding to the memory elements in the actual xbar matrix.

[0055] A behavior of the ideal xbar matrix may be determined 556. The ideal conductance of the memory elements of the ideal xbar matrix may be at least part of the behavior that is determined. To contribute to determination of the ideal conductance of the memory elements, a predetermined input pattern 557 may be used. The predetermined input pattern 557 may, for example, include a selection of memory elements at particular positions in the xbar matrix to be used in determination of adjustment (e.g., calibration) of the conductance in the actual xbar matrix. In some examples, the predetermined input pattern 557 may include particular input voltages to simulate setting the conductance of the memory elements of the ideal xbar matrix to a particular level. For example, a simulated input vector pattern may be used to determine (e.g., calculate) the ideal conductance of memory elements at the plurality of junctions of the crossbar array corresponding to the input vector pattern. The predetermined input pattern 557 may be set (or mapped) 558 to the ideal xbar matrix as calibration signals.

[0056] An effective calibration signal may be an input signal (e.g., V_{in}) of, for example, an input vector that may later be input for computation by a DPE. However, a particular input signal that would be input in the future may be unknown at the time of calibration, although the the predetermined input pattern 557 of input signals to be input in the future may be known at the time of calibration. As such, the DPE, for example, may be calibrated with simulated input signal voltages in the predetermined input pattern 557 that cover a range of expected actual input signal voltages for input to memory elements at particular positions in the xbar matrix.

[0057] Input of a number of ideal crossbar parameters 555 may, in some examples, be used in determination (e.g., calculation) of the ideal conductance of memory elements at the plurality of junctions of the ideal crossbar array. In various examples, the number of ideal crossbar parameters may include an ON state resistance of the ideal crossbar, an OFF state resistance of the ideal crossbar, digital-to-analog converter (DAC) settings, and/or analog-to-digital converter (ADC) settings, among other possible parameters usable in determination of the ideal conductance of the memory elements.

[0058] In some examples, as described herein, a number of actual xbar matrix parameters 559 may be input for use in determination of an adjustment of the actual conductance of memory elements of the actual xbar matrix 560 in order to approximate the behavior of the ideal xbar matrix. Such a simulation may be performed using a number of the actual xbar matrix parameters 559. As described herein, the actual xbar matrix parameters 559 may include a number of models of signal output variation in crossbar arrays, including nonlinearity of wire resistance and/or other causes of signal output nonlinearity, input and/or output resistance, thermal effects, noise, and/or parasitics, etc. Alternatively or in addition, a determination (e.g., measurement) of actual xbar matrix conductance 561 may be used in determination (e.g., calculation) of an adjustment of the actual conductance of memory elements of the actual xbar matrix in order to approximate the behavior of the ideal xbar matrix.

[0059] Following determination of a change (e.g., level of adjustment) in a conductance value for each memory element to be adjusted to approximate the behavior of the ideal xbar matrix, the process may, in some examples, proceed to the xbar programming portion 563. Within the xbar programming portion 563, in some examples, the determined change in the conductance value may be executed by a programmed change 564 in the conductance values for each of the memory elements to be adjusted (calibrated). As described herein, the programmed change in conductance may be executed, in some examples, by closed-loop voltage pulse tuning of the memory elements,

among other possible implementations for adjusting the conductance by application of determined voltage to the memory elements.

[0060] Following changing the conductance values for each of the memory elements to be adjusted in the actual xbar matrix, the process may, in some examples, proceed to the computing portion 570. Within the computing portion 570, in these examples, an input signal (e.g., V_{in}) of, for example, an input vector, may be input to the actual xbar matrix and a number of output signals (e.g., V_{out} and/or I_{out}) may be read from the actual xbar matrix 567. An xbar output signal may be mapped 568 to convert the output signal value to obtain an output value. For example, an analog V_{out} and/or I_{out} signal value may be converted to a corresponding digital data value, as mapped appropriately to the actual xbar matrix. In some examples, the mapping and/or the conversion of a plurality of V_{out} and/or I_{out} signal values to, for example, the summed single value of $\sum_{i=1}^m I_{outi}$ may be performed by a compute engine (e.g., as shown at and described in connection with 111 in Figure 1 and elsewhere).

[0061] The output signals and/or output values may, in some examples, be compared to corresponding ideal output signals and/or output values to determine whether the changes made to the conductance of the memory elements resulted in output from the actual xbar matrix being brought close enough to the output of the ideal xbar matrix. If the output from the actual xbar matrix is determined not to be close enough to the output of the ideal xbar matrix (e.g., either by operator and/or programmed inspection), the process just described may be repeated.

[0062] In situations where the matrix values have been determined at 543 to already have been input to the actual xbar matrix, the input value/signal 565 (e.g., V_{in}) may be input directly to the computing portion 570 of the process. For example, the determining portion 541 and/or the xbar programming portion 563 of the process may be skipped based on the determination that the matrix values have already been input to the actual xbar matrix because such a determination may indicate that adjustment (calibration) of the conductance of the memory elements of the actual xbar matrix has already been performed. As such, mapping parameters 562 may be used to map 566 the input value/signal 565, for

example, an input vector, to the actual xbar matrix. In various examples, the mapping parameters 562 may be determined from mapping the xbar matrix to ideal xbar conductance at 544 and/or from already saved mapping parameters of the actual xbar matrix.

[0063] Following each occurrence of the output signals (e.g., Vout and/or Iout) being read 567, mapped 568, and/or converted to obtain an output value, a determination 569 may be made as to whether a next computation is to be performed. When a determination 569 is made that there is a next computation to be performed, and another iteration of the determining portion 541 and/or the xbar programming portion 563 are not to be performed, a next input value/signal 565 (e.g., an input vector) may be input and mapped 566 to the actual xbar matrix for performance of the next computation. A determination that another iteration of the determining portion 541 and/or the xbar programming portion 563 will not to be performed may be based, for example, upon a determination that the previous adjustment of the conductance of the memory elements has satisfactorily approximated the conductance of the ideal xbar matrix. When a determination 569 is made that there is not a next computation to be performed, the process may end 571.

[0064] Figure 6 illustrates a diagram of an example of improved computational accuracy in a crossbar array with adjusted conductance of memory elements according to the present disclosure. Figure 6 shows a graph 672 that illustrates comparative computational accuracy for outputs from crossbar arrays of various sizes that have been linearly mapped 673 without adjustment of memory cell conductance, as described herein, and following adjustment (calibration) of the conductance 674.

[0065] The size of the crossbar arrays shown in Figure 6 is represented by the column number on the horizontal axis. For a square crossbar array, the column number also may correspond to the row number in the crossbar array, both of which may thus correspond to a number of memory elements (e.g., memristors) at junctions of the column wires and the row wires. As described herein, due to causes including nonlinearity of increasing resistance, among other causes, the accuracy of

output of a crossbar array may progressively decrease corresponding to an increase in the size of the crossbar array.

[0066] For example, Figure 6 shows determinations of bit-accuracy for crossbar arrays ranging from 2 rows x 2 columns (2x2) to 256 rows x 256 columns (256x256). By way of example, the bit-accuracy scale shown on the vertical axis is for vector-matrix multiplication done by a DPE. The particular values of 0.00 through 8.00 on the vertical axis represent values for N . An N bit-accuracy may represent 2^N distinguishable levels in a range of signals, where a higher number of distinguishable levels corresponds to a higher accuracy in signal output from the crossbar array.

[0067] Accordingly, the results for a linear mapped crossbar array 673, in which the conductance of the memristors has not been adjusted, show a bit-accuracy of less than 2.00 for a 2x2 crossbar array. The bit-accuracy progressively decreases nonlinearly as the size of the crossbar array approaches 256x256.

[0068] In contrast, the results for a crossbar array 674 in which the conductance of the memristors has been adjusted show a bit-accuracy of greater than 6.00 for a 2x2 crossbar array. The bit-accuracy for the crossbar array in which the conductance of the memristors has been adjusted does not show a decrease as the size of the crossbar array approaches 256x256. Hence, adjustment of the conductance of memory elements may be demonstrated to improve computational accuracy in a crossbar array.

[0069] Figure 7 illustrates a diagram of an example of a system for improved computational accuracy in a crossbar array according to the present disclosure. The system 775 may include a combination of an actual crossbar array 710 (e.g., as described in connection with Figures 1-3, etc.), and a number of engines 776 to enable execution of particular tasks 777, 778, 779. The system 775 may be in communication with the actual crossbar array 710 via a communication link, and may include the number of engines (e.g., calculate engine 777, determine engine 778, adjust engine 779, etc.). The system 775 may include additional or fewer

engines than illustrated to perform the various tasks described herein. The system 775 may represent programmed instructions and/or hardware.

[0070] The number of engines may include a combination of hardware and instructions (e.g., programming) to perform a number of tasks described herein (e.g., to contribute to improvement of computational accuracy by reduction of a difference between ideal conductance and determined conductance of memory elements, etc.). The instructions may be executable by a processing resource and stored in a non-transitory memory resource (e.g., computer-readable medium (CRM), machine-readable medium (MRM), etc.), or may be hard-wired in hardware (e.g., logic).

[0071] As described herein, the system 775 may include an actual crossbar array 710 that includes a plurality of memory elements, where a memory element may be located, for example, at each junction of the crossbar array. The actual crossbar array 710 is usable in performance of various types of computations. For example, the actual crossbar array 710 may be used as a DPE for vector-matrix multiplication, among other computations suitable for a crossbar array.

[0072] The calculate engine 777 may include hardware and/or a combination of hardware and instructions (e.g., programming) to calculate ideal conductance of memory elements at a plurality of junctions of the crossbar array. As described herein the ideal conductance of the memory elements may be calculated using a simulated ideal crossbar array with at least some of the causes of output voltage and/or current nonlinearity described herein not included in the simulation of the ideal crossbar array.

[0073] The determine engine 778 may include hardware and/or a combination of hardware and instructions (e.g., programming) to determine conductance of the memory elements at the plurality of junctions of the actual crossbar array 710. The determine engine 778 may determine the conductance of the memory elements using at least one of the measurements of the progressive decrease of the actual I_{out} corresponding to a number of rows and columns in the actual crossbar array, direct and/or indirect measurements of the conductance of the memory elements at the junctions of the actual crossbar array, and/or

usage of a number of models of signal output variation in crossbar arrays, as described herein.

[0074] The adjust engine 779 may include hardware and/or a combination of hardware and instructions (e.g., programming) to adjust conductance of at least one memory element to improve computational accuracy by reduction of a difference between the ideal conductance and the determined conductance of the at least one memory element. The conductance of the least one memory element may be adjusted, in various examples, by application of one or more determined voltages (e.g., pulses) to the least one memory element by the adjust engine 779 to adjust the conductance, as described herein. As such, in some examples, the adjust engine 779 may include, or be associated with, hardware for closed-loop voltage pulse tuning, for example. In various examples, the at least one memory element for which the conductance is adjusted may be a resistive memory element (e.g., a memristor), a phase change memory element, a floating gate memory element, among other types of memory elements implemented in the crossbar array as appropriate for particular types of computations.

[0075] In some examples, the system 775 may include a compute engine, for example, as shown at 111 in Figure 1. The compute engine may be used to perform a number of functions. For example, the compute engine may be used to input an input value to the actual crossbar array 710, map the input value to a determined memory element in the actual crossbar array, direct the input value to the determined memory element, where the input value affects conductance of the determined memory element (either as a data input voltage value or as a determined voltage value intended for adjustment of the conductance), read an output signal from the determined memory element, and/or map the output signal as an output value relative to the input value. For the intended adjustment of the conductance, for example, the input value may correspond to a magnitude of an input voltage signal, the output value may correspond to a magnitude of an output current signal, and the magnitude of the output current signal may be affected by the adjustment (determined change) of

the conductance of the determined memory element to a level corresponding to the magnitude of the input voltage signal.

[0076] As described herein, the adjustment of the conductance of the determined memory element by the determined voltage may improve computational accuracy by the output value being shifted closer to an ideal output calculated by the calculate engine 777. In some examples, the conductance of a plurality of memory elements (e.g., some or all of the memory elements in the actual crossbar array 710) may be adjusted to shift the output value closer to an ideal output for each of the memory elements calculated by the calculate engine 777.

[0077] Figure 8 illustrates a diagram of an example computing device for improved computational accuracy in a crossbar array according to the present disclosure. The computing device 880 may utilize programmed instructions, hardware, hardware with instructions, and/or logic to perform a number of tasks described herein. The computing device 880 may include additional or fewer modules than illustrated to execute the various tasks described herein.

[0078] The computing device 880 may be any combination of hardware and program instructions to share information. The hardware, for example, may include a processing resource 881 and/or a memory resource 883 (e.g., CRM, MRM, database, etc.) The processing resource 881, as used herein, may include any number of processors capable of executing instructions stored by the memory resource 883. The processing resource 881 may be integrated in a single device or distributed across multiple devices. The program instructions (e.g., computer-readable instructions (CRI), machine-readable instructions (MRI), etc.) may include instructions stored on the memory resource 883 and executable by the processing resource 881 to implement a desired task (e.g., calculate ideal conductance of memory elements at a plurality of junctions of a crossbar array, etc.).

[0079] The memory resource 883 may be in communication with the processing resource 881. The memory resource 883, as used herein, may include any number of memory components capable of storing instructions that may be executed by the processing resource 881. Such

a memory resource 883 may be a non-transitory CRM or MRM. The memory resource 883 may be integrated in a single device or distributed across multiple devices. Further, the memory resource 883 may be fully or partially integrated in the same device as the processing resource 881 or it may be separate but accessible to that device and processing resource 881. Thus, the computing device 880 may be implemented on a participant device, on a server device, on a collection of server devices, and/or on a combination of the user device and the server device.

[0080] The memory resource 883 may be in communication with the processing resource 881 via a communication link (e.g., path) 882. The communication link 882 may be local or remote to a machine (e.g., a computing device) associated with the processing resource 881. Examples of a local communication link 882 may include an electronic bus internal to a machine (e.g., a computing device) where the memory resource 883 is one of volatile, non-volatile, fixed, and/or removable storage medium in communication with the processing resource 881 via the electronic bus.

[0081] A number of modules 884, 885, 886 may include MRI that when executed by the processing resource 881 may perform a number of tasks. The number of modules 884, 885, 886 may be sub-modules of other modules. For example, the calculate module 884 and the determine module 885 may be sub-modules and/or contained within the same computing device. In another example, the number of modules 884, 885, 886 may comprise individual modules at separate and distinct locations (e.g., CRM, MRM, etc.).

[0082] Each of the number of modules 884, 885, 886 may include instructions that when executed by the processing resource 552 may function as a corresponding engine, as described herein. For example, calculate module 884 may include instructions that when executed by the processing resource 881 may function as the calculate engine 777. In another example, the adjust module 886 may include instructions that when executed by the processing resource 881 may function as the adjust engine 779.

[0083] The calculate module 884 may include MRI that when executed by the processing resource 881 may perform a number of tasks. For example, the calculate module 884 may calculate ideal conductance of memory elements at a plurality of junctions of a crossbar array, as described in connection with the calculate engine 777. The calculate module 884 may include further MRI that when executed by the processing resource 881 may perform input of an input vector pattern to calculate the ideal conductance of memory elements at the plurality of junctions of the crossbar array corresponding to the input vector pattern, as described in connection with 556, 557, 558 in Figure 5. The calculate module 884 may include further MRI that when executed by the processing resource 881 may perform input of a number of ideal crossbar parameters to calculate the ideal conductance of memory elements at the plurality of junctions of the crossbar array, as described in connection with 555, 556 in Figure 5.

[0084] The determine module 885 may include MRI that when executed by the processing resource 881 may perform a number of tasks. For example, the determine module 885 may be used to determine actual conductance of the memory elements at the plurality of junctions of the actual crossbar array, as described in connection with the determine engine 778. The determine module 885 may include further MRI that when executed by the processing resource 881 may perform input of a number of actual crossbar parameters to determine the adjustment of the actual conductance, as described in connection with 559, 560 in Figure 5. The input of the number of actual crossbar parameters may, for example, include a number of models of signal output variation in crossbar arrays, among other possible actual crossbar parameters.

[0085] The adjust module 886 may include MRI that when executed by the processing resource 881 may perform a number of tasks. For example, the adjust module 886 may be used to determine an adjustment of the actual conductance of at least one memory element in the actual crossbar array to approximate the ideal conductance of the at least one memory element in the ideal crossbar array, as described in connection with the adjust engine 779. The adjust module 886 may include further

MRI that when executed by the processing resource 881 may perform programming of conductance of the at least one memory element to a higher conductance to improve computational accuracy by reduction of a difference between the ideal conductance and the determined conductance, as described in connection with 563, 564 in Figure 5.

[0086] Figure 9 illustrates a flow diagram of an example method for improved computational accuracy in a crossbar array according to the present disclosure. Unless explicitly stated, the method examples described herein are not constrained to a particular order or sequence. Additionally, some of the described method examples, or elements thereof, may be performed at the same, or substantially the same, point in time. As described herein, the actions, tasks, functions, calculations, data manipulations and/or storage, etc., may be performed by execution of non-transitory machine-readable instructions stored in a number of memories (e.g., programmed instructions, hardware with instructions, hardware, and/or logic, etc.) of a number of applications. As such, a number of computing resources with a number of interfaces (e.g., user interfaces) may be utilized for implementing the methods described herein (e.g., via accessing a number of computing resources via the user interfaces).

[0087] The present disclosure describes a method 990 for calculating ideal conductance of memory elements at junctions of a crossbar array, as shown at 991 in Figure 9. With regard to calculation of the ideal conductance, the crossbar array may be a simulated (e.g., ideal) crossbar array, for example, as described in connection with Figure 4. At 992, the method may include determining conductance of the memory elements at the junctions of the crossbar array. With regard to determining the conductance, the conductance may, in some examples, be determined from an actual crossbar array using various techniques, for example, as described in connection with Figure 4. At 993, the method may include adjusting conductance of at least one memory element to reduce a difference between the ideal conductance and the determined conductance of the at least one memory element, for example, as described in connection with 779 in Figure 7.

[0088] At 993, the method may include calibrating (adjusting) the conductance of the crossbar to reduce nonlinearity of signal output for improvement of computational accuracy, for example, as described in connection with 563, 564 in Figure 5. In some examples, computational accuracy of the actual crossbar array may be determined by how close to being equal to an ideal current output, determined for a corresponding ideal crossbar array, an actual current output is after calibration of the crossbar array. An improvement in computational accuracy may, in some examples, be determined by how close to being equal to the ideal current output the actual current output is relative to before the calibration.

[0089] As used herein, “a” or “a number of” something may refer to one or more such things. For example, “a number of widgets” may refer to one or more widgets. Also, as used herein, “a plurality of” something may refer to more than one of such things.

[0090] As used herein, “logic” is an alternative or additional processing resource to perform a particular action and/or function, etc., described herein, which includes hardware (e.g., various forms of transistor logic, application specific integrated circuits (ASICs), etc.), as opposed to computer executable instructions (e.g., programmed instructions, hardware with instructions, etc.) stored in memory and executable by a processor.

[0091] The figures herein follow a numbering convention in which the first digit corresponds to the drawing figure number and the remaining digits identify an element or component in the drawing. For example, 114 may reference element “14” in Figure 1, and a similar element may be referenced as 214 in Figure 2. Elements shown in the various figures herein may be capable of being added, exchanged, and/or eliminated so as to provide a number of additional examples of the present disclosure. In addition, the proportion and the relative scale of the elements provided in the figures are intended to illustrate the examples of the present disclosure, and should not be taken in a limiting sense.

[0092] In the present disclosure, reference is made to the accompanying drawings that form a part hereof, and in which is shown by way of illustration how a number of examples of the disclosure may be

practiced. These examples are described in sufficient detail to enable those of ordinary skill in the art to practice the examples of this disclosure, and it is to be understood that other examples may be used and that process, computational, electrical, and/or structural changes may be made without departing from the scope of the disclosure.

[0093] The specification examples provide a description of the applications and use of the system and method of the present disclosure. Since many examples may be made without departing from the spirit and scope of the system and method of the present disclosure, this specification sets forth some of the many possible example configurations and implementations.

What is claimed is:

1. A system, comprising:
 - a crossbar array, comprising a plurality of memory elements at junctions, usable in performance of computations;
 - a calculate engine to calculate ideal conductance of memory elements at a plurality of junctions of the crossbar array;
 - a determine engine to determine conductance of the memory elements at the plurality of junctions of the crossbar array; and
 - an adjust engine to adjust conductance of at least one memory element to improve computational accuracy by reduction of a difference between the ideal conductance and the determined conductance of the at least one memory element.
2. The system of claim 1, wherein the at least one memory element is a resistive memory element.
3. The system of claim 1, wherein to adjust the conductance of the least one memory element comprises application of a determined voltage to the least one memory element by the adjust engine to adjust the conductance.
5. The system of claim 1, wherein the system further comprises a compute engine to:
 - input an input value;
 - direct the input value to the determined memory element, wherein the input value affects conductance of the determined memory element; and
 - read an output signal from the determined memory element;wherein an adjustment of conductance of the determined memory element improves computational accuracy by the output value being shifted closer to an ideal output calculated by the calculate engine.
6. The system of claim 5, wherein:
 - the input value corresponds to a magnitude of an input voltage signal;
 - the output value corresponds to a magnitude of an output current signal; and

the magnitude of the output current signal is affected by the adjustment of the conductance of the determined memory element.

7. A non-transitory computer-readable medium storing instructions executable by a processing resource to:

calculate ideal conductance of memory elements at a plurality of junctions of a crossbar array;

determine actual conductance of the memory elements at the plurality of junctions of the crossbar array;

determine an adjustment of the actual conductance of at least one memory element to approximate the ideal conductance of the at least one memory element;
and

program conductance of the at least one memory element to a higher conductance to improve computational accuracy by reduction of a difference between the ideal conductance and the determined conductance.

8. The medium of claim 7, the instructions further executable to input an input vector pattern to calculate the ideal conductance of memory elements at the plurality of junctions of the crossbar array corresponding to the input vector pattern.

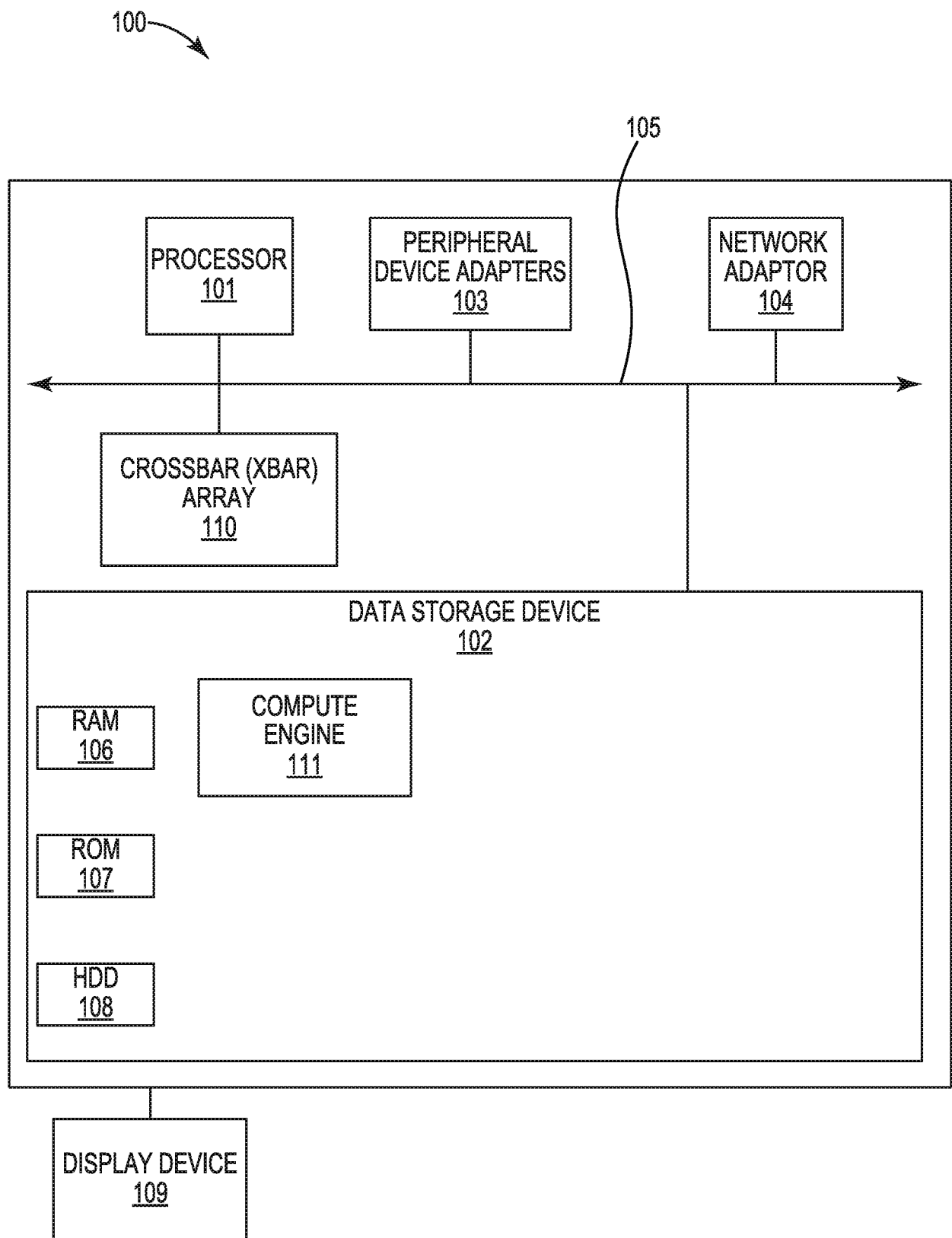
9. The medium of claim 7, the instructions further executable to input a number of ideal crossbar parameters to calculate the ideal conductance of memory elements at the plurality of junctions of the crossbar array.

10. The medium of claim 9, wherein the number of ideal crossbar parameters comprise ON state resistance of the ideal crossbar, OFF state resistance of the ideal crossbar, digital-to-analog converter settings, and analog-to-digital converter settings.

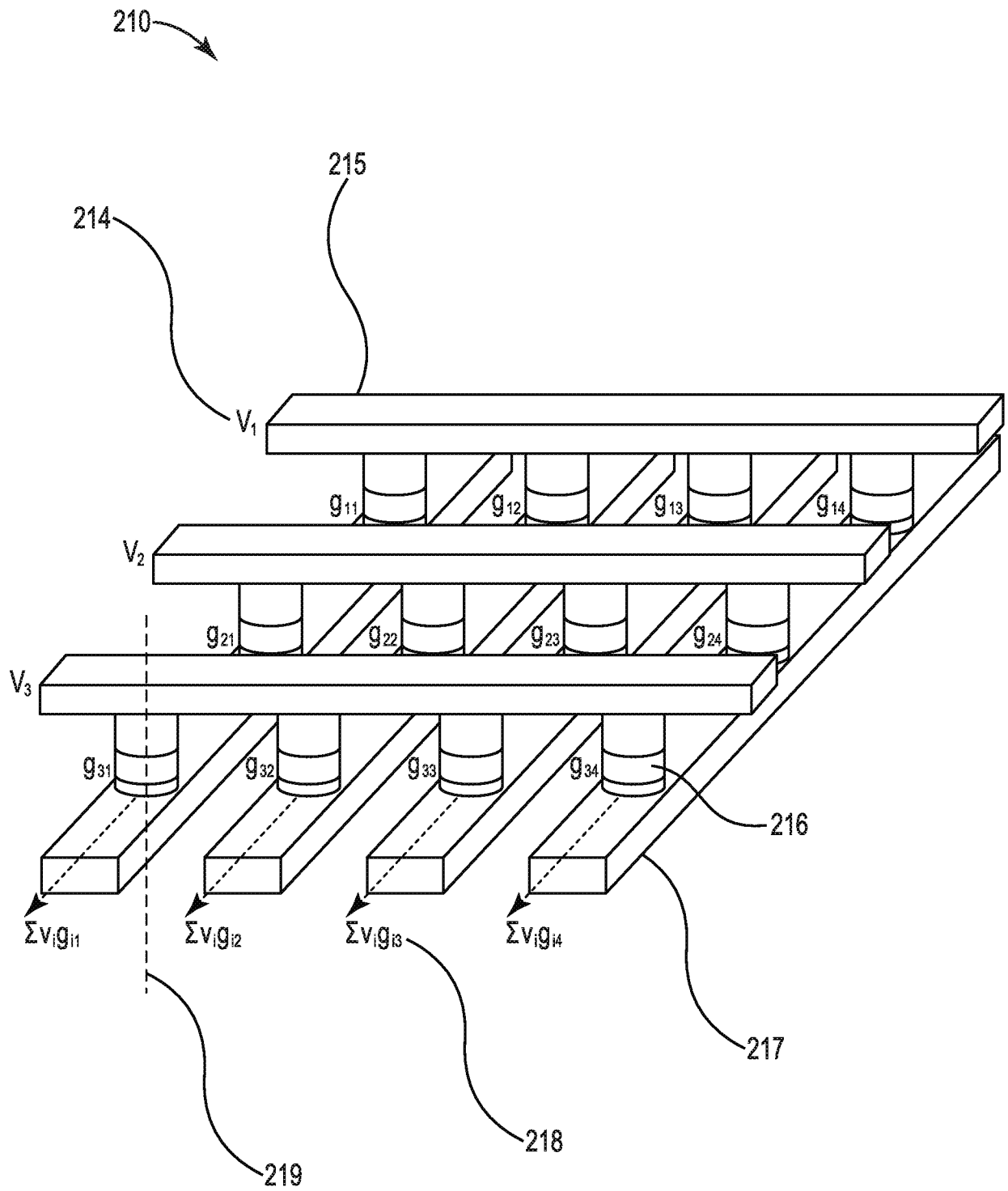
11. The medium of claim 7, the instructions further executable to input a number of actual crossbar parameters to determine the adjustment of the actual conductance.

12. The medium of claim 9, wherein the number of actual crossbar parameters comprise a number of models of signal output variation in crossbar arrays.
13. A method, comprising:
 - calculating ideal conductance of memory elements at junctions of a crossbar array;
 - determining conductance of the memory elements at the junctions of the crossbar array;
 - adjusting conductance of at least one memory element to reduce a difference between the ideal conductance and the determined conductance of the at least one memory element; and
 - calibrating the conductance of the crossbar to reduce nonlinearity of signal output for improvement of computational accuracy.
14. The method of claim 13, further comprising:
 - determining computational accuracy of the crossbar array by how close to being equal to an ideal current output, determined for a corresponding ideal crossbar array, an actual current output is after calibration of the crossbar array.
15. The method of claim 13, wherein calibrating the conductance of the crossbar comprises performing closed-loop voltage pulse tuning of the memory elements.

1/8

**Fig. 1**

2/8

**Fig. 2**

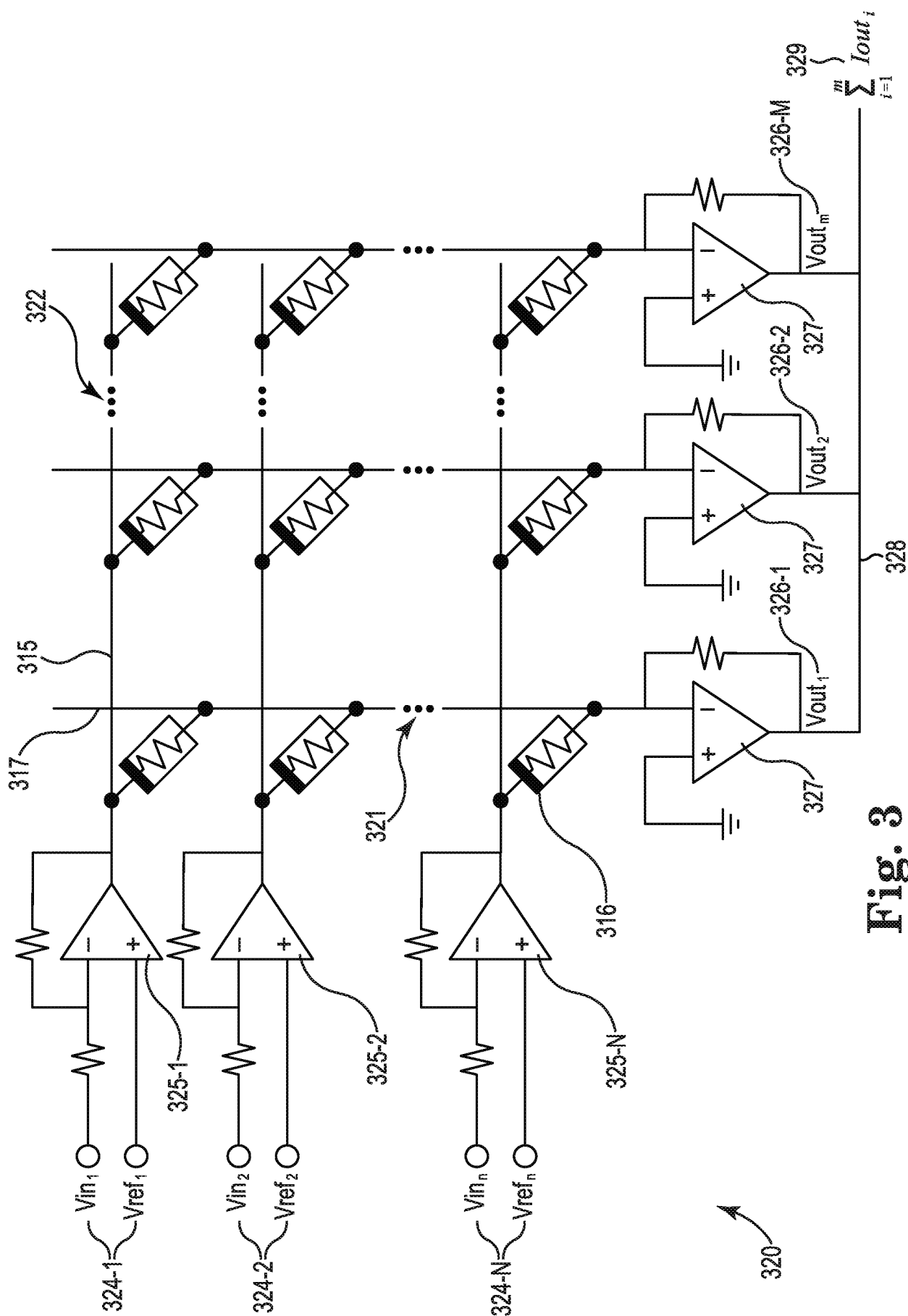


Fig. 3

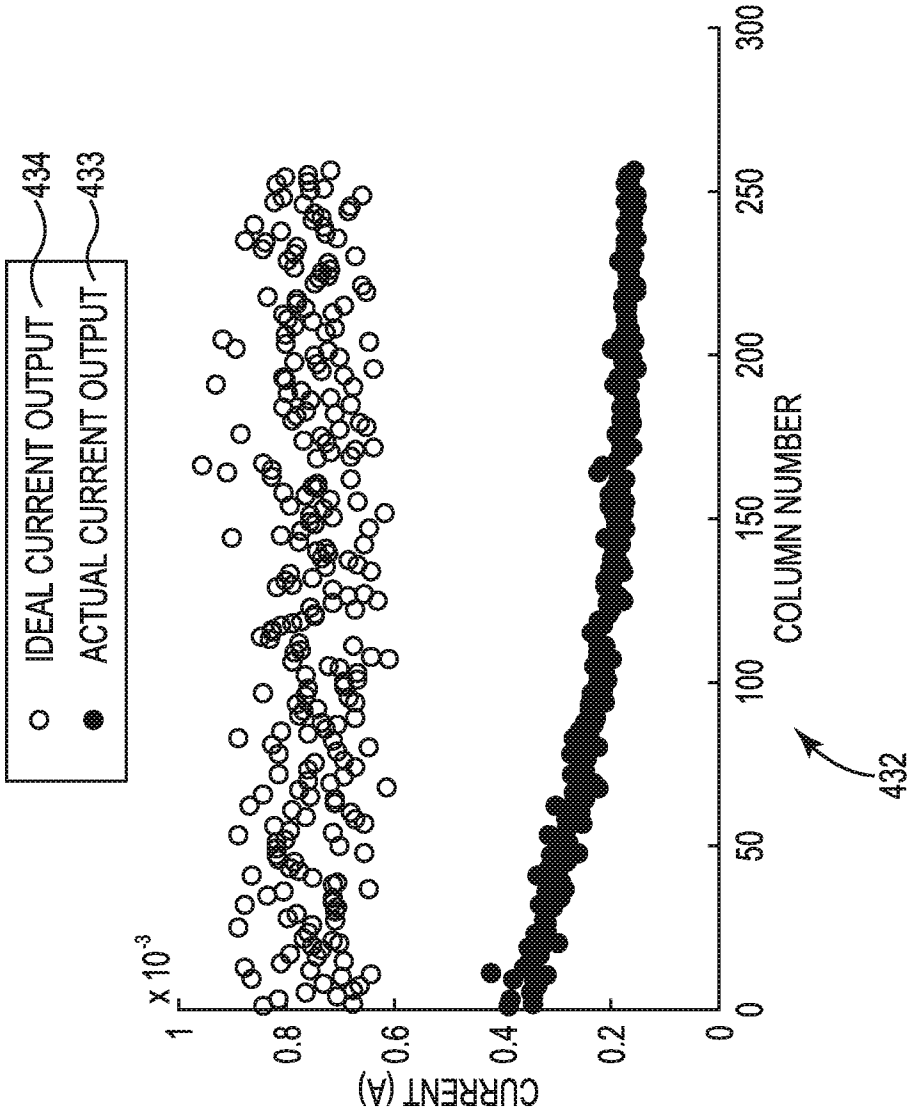


Fig. 4

5/8

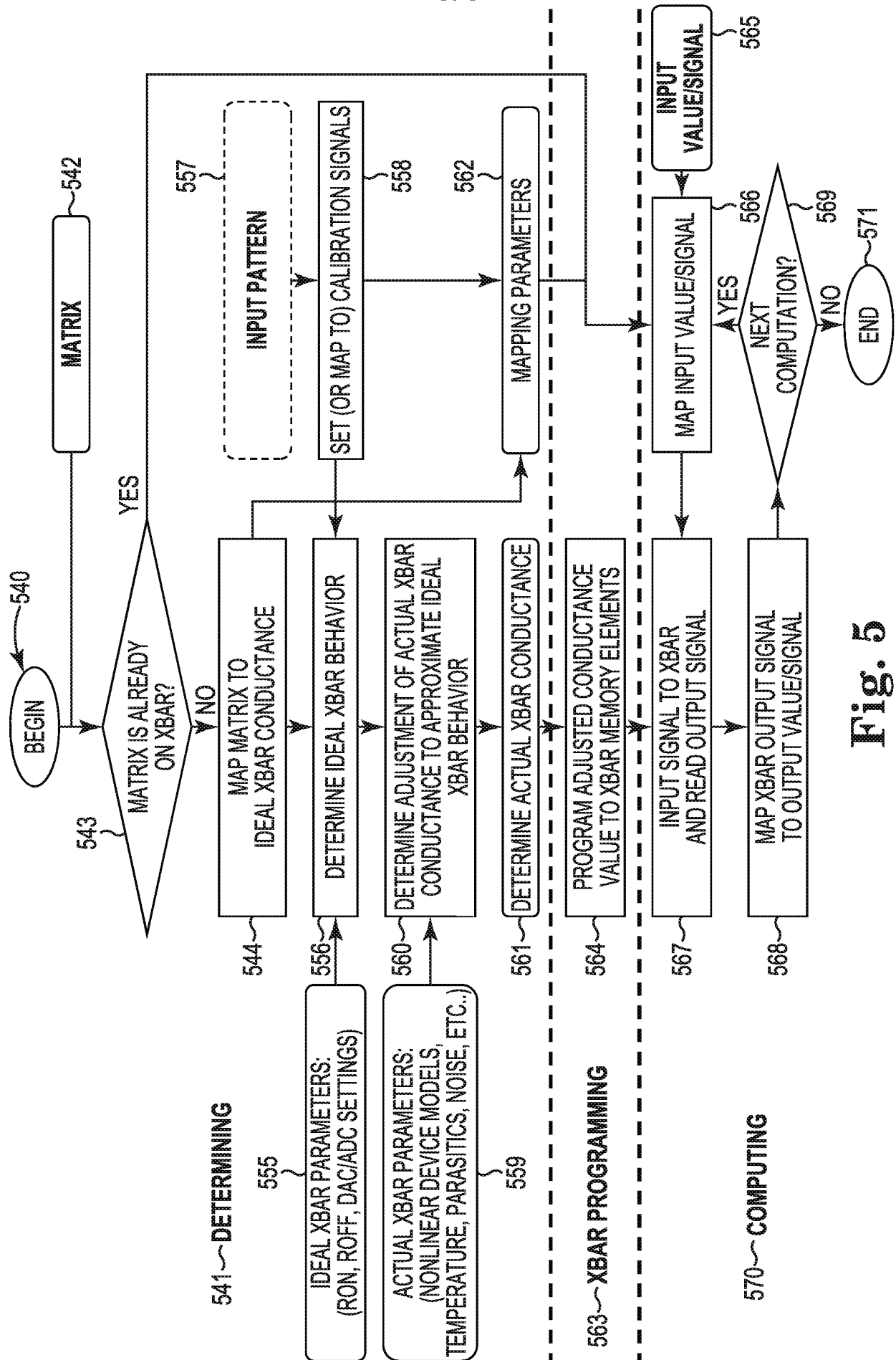
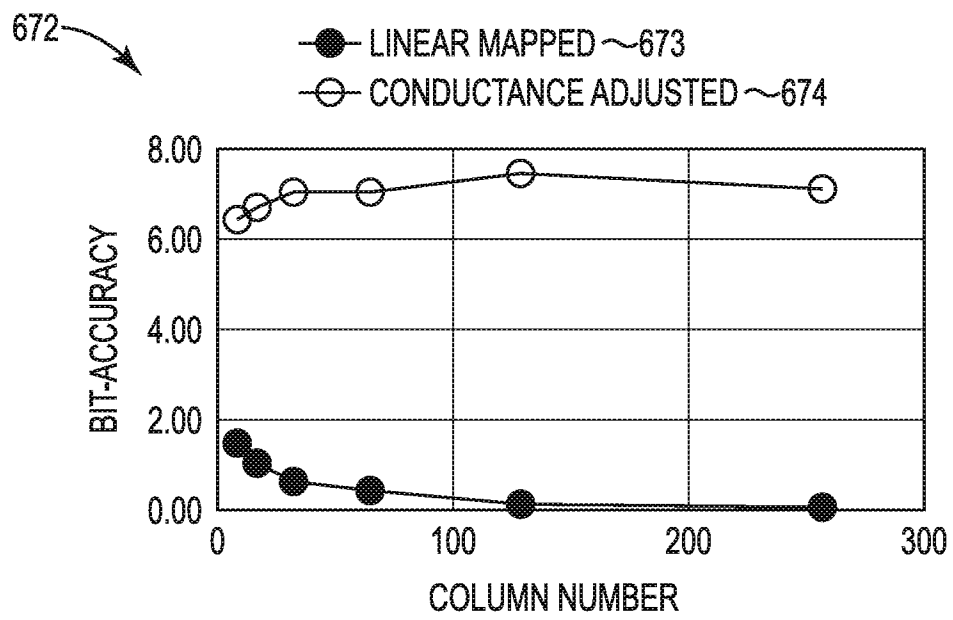
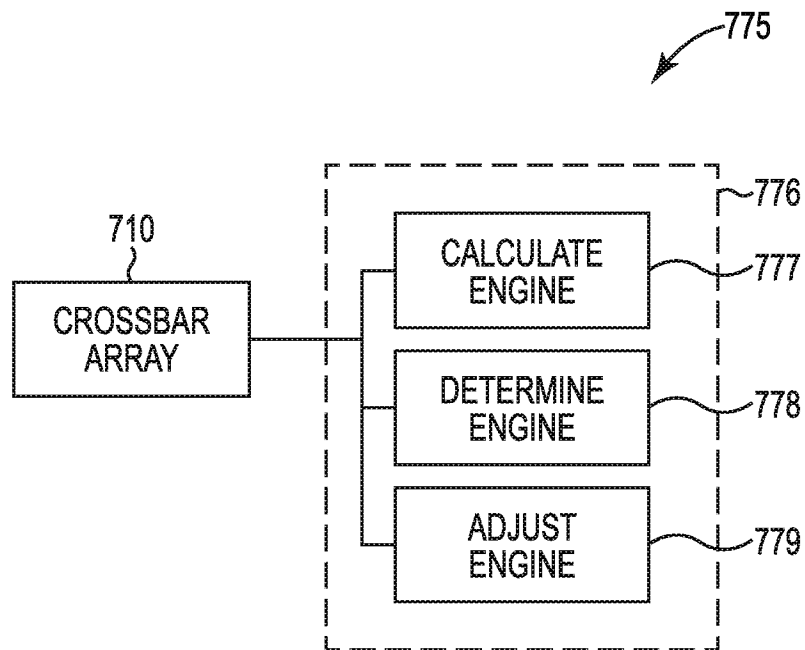
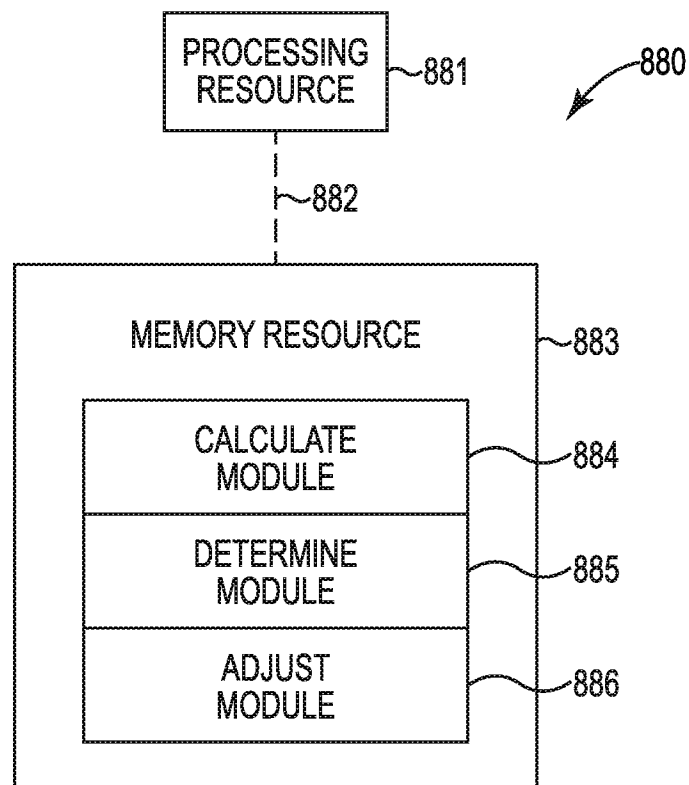


Fig. 5

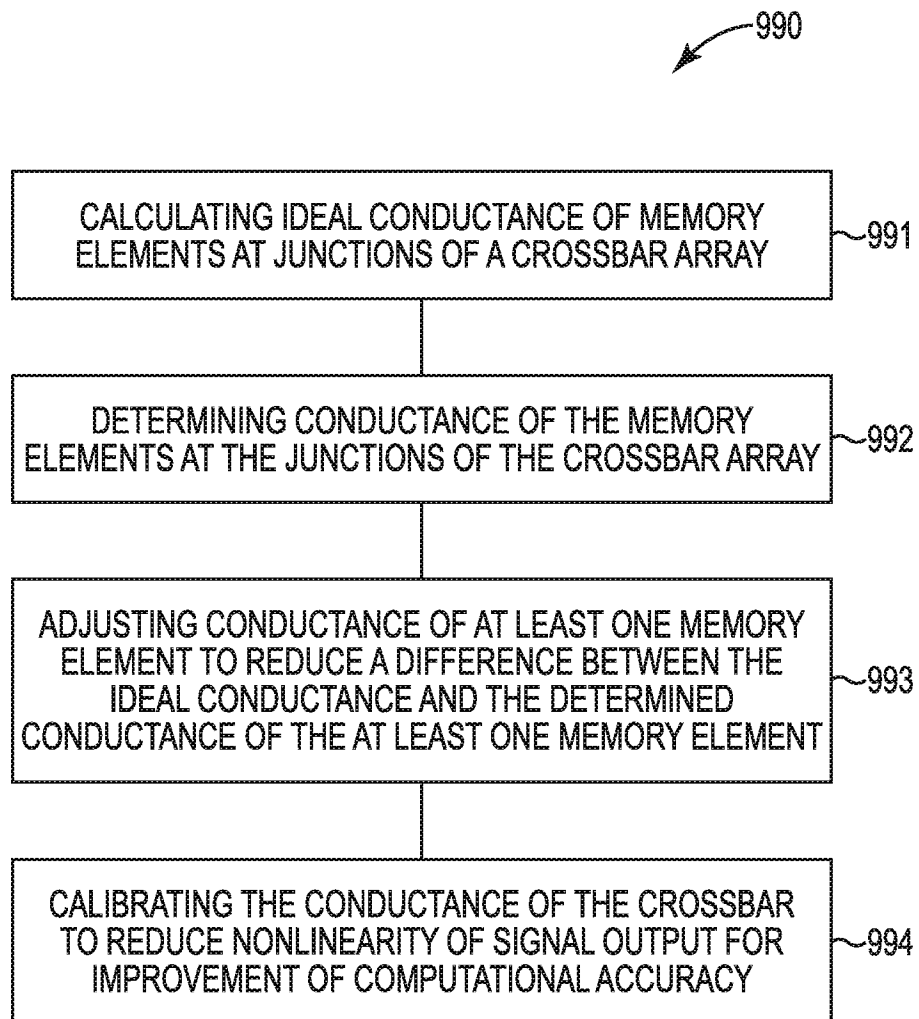
6/8

**Fig. 6**

7/8

**Fig. 7****Fig. 8**

8/8

**Fig. 9**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/066371**A. CLASSIFICATION OF SUBJECT MATTER****G11C 11/15(2006.01)i, G11C 13/00(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)

G11C 11/15; G06F 17/50; G11C 11/02; H01L 47/00; H01L 45/00; G11C 7/12; G11C 5/02; G11C 13/00; H01L 21/8239

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: crossbar, array, resistive, memory, element, calculate, engine, determine, adjust, conductance

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2012-0166169 A1 (WEI LU) 28 June 2012 See paragraph [0058]; and figures 6a-6b.	1-3,5-15
A	US 2014-0133211 A1 (HAGOP NAZARIAN et al.) 15 May 2014 See paragraphs [0046], [0050]-[0052]; and figures 3A, 4.	1-3,5-15
A	US 2010-0102290 A1 (WEI LU et al.) 29 April 2010 See paragraphs [0048]-[0052]; and figures 5a-5b.	1-3,5-15
A	US 2015-0029780 A1 (DARRELL RINERSON et al.) 29 January 2015 See paragraphs [0040]-[0043]; and figures 4A-4B.	1-3,5-15
A	US 2011-0188282 A1 (CHRISTOPHE J. CHEVALLIER et al.) 04 August 2011 See paragraphs [0032]-[0034]; and figure 3.	1-3,5-15



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

24 August 2016 (24.08.2016)

Date of mailing of the international search report

24 August 2016 (24.08.2016)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

LEE, EUN KYU

Telephone No. +82-42-481-3580



INTERNATIONAL SEARCH REPORTInternational application No.
PCT/US2015/066371**Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)**

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☒ Claims Nos.: 4
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
Claim 4 does not comply with PCT Rule 6.1(b), because claim 4 is missing in this application.
3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of any additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- ☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- ☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/066371

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0166169 A1	28/06/2012	US 8930174 B2	06/01/2015
US 2014-0133211 A1	15/05/2014	CN 103871462 A	18/06/2014
		TW 201432690 A	16/08/2014
		US 8982647 B2	17/03/2015
US 2010-0102290 A1	29/04/2010	CN 102265398 A	30/11/2011
		EP 2351083 A2	03/08/2011
		EP 2351083 A4	13/08/2014
		JP 2012-506621 A	15/03/2012
		KR 10-2011-0086089 A	27/07/2011
		US 2012-049149 A1	01/03/2012
		US 2015-138873 A1	21/05/2015
		US 8071972 B2	06/12/2011
		US 8907317 B2	09/12/2014
		US 9159416 B2	13/10/2015
		WO 2010-048127 A2	29/04/2010
		WO 2010-048127 A3	15/07/2010
US 2015-0029780 A1	29/01/2015	CN 102694122 A	26/09/2012
		EP 1800314 A2	27/06/2007
		EP 2284840 A2	16/02/2011
		EP 2284840 A3	22/06/2011
		JP 2008-512857 A	24/04/2008
		JP 2012-238893 A	06/12/2012
		KR 10-2007-0047341 A	04/05/2007
		US 2006-171200 A1	03/08/2006
		US 2008-109775 A1	08/05/2008
		US 2009-303772 A1	10/12/2009
		US 2009-303773 A1	10/12/2009
		US 2011-310658 A1	22/12/2011
		US 2012-087174 A1	12/04/2012
		US 2012-176840 A1	12/07/2012
		US 8020132 B2	13/09/2011
		US 8141021 B2	20/03/2012
		US 8347254 B2	01/01/2013
		US 9159913 B2	13/10/2015
		WO 2006-029228 A2	16/03/2006
		WO 2006-029228 A3	20/07/2006
US 2011-0188282 A1	04/08/2011	US 8638584 B2	28/01/2014