(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 3 841 527 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
05.04.2023 Bulletin 2023/14

(21) Application number: **19745503.3**

(22) Date of filing: **09.07.2019**

(51) International Patent Classification (IPC):
G06N 3/063 *(2006.01)* ***G06N 3/04*** *(2006.01)*
G06F 17/16 *(2006.01)* ***G11C 11/54*** *(2006.01)*
G11C 16/04 *(2006.01)* ***G11C 16/10*** *(2006.01)*
G11C 16/16 *(2006.01)* ***G11C 16/26*** *(2006.01)*
G11C 16/34 *(2006.01)* *H01L 29/423* *(2006.01)*

(52) Cooperative Patent Classification (CPC):
G11C 11/54; G06F 17/16; G06N 3/0442; G06N 3/045; G06N 3/065; G11C 16/0425; G11C 16/10; G11C 16/16; G11C 16/26; G11C 16/3459; G06N 3/048; H01L 29/42328

(86) International application number:
PCT/US2019/041039

(87) International publication number:
WO 2020/040893 (27.02.2020 Gazette 2020/09)

(54) **ANALOG NEURAL MEMORY SYSTEM FOR DEEP LEARNING NEURAL NETWORK COMPRISING MULTIPLE VECTOR-BY-MATRIX MULTIPLICATION ARRAYS AND SHARED COMPONENTS**

ANALOGES NEURONALES SPEICHERSYSTEM FÜR EIN TIEFENLERNENDES NEURONALES NETZ MIT MEHREREN VEKTOR-X-MATRIX-MULTIPLIKATIONSFELDERN UND GEMEINSAMEN KOMPONENTEN

SYSTÈME DE MÉMOIRE NEURONALE ANALOGIQUE POUR RÉSEAU DE NEURONES ARTIFICIELS À APPRENTISSAGE PROFOND COMPRENANT DE MULTIPLES RÉSEAUX DE MULTIPLICATION VECTORIELLE PAR MATRICE ET COMPOSANTS PARTAGÉS

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR

(30) Priority: **21.08.2018 US 201862720902 P**
06.11.2018 US 201816182492

(43) Date of publication of application:
30.06.2021 Bulletin 2021/26

(60) Divisional application:
23157551.5

(73) Proprietor: **Silicon Storage Technology, Inc.**
San Jose, CA 95134 (US)

(72) Inventors:
- **TRAN, Hieu, Van**
San Jose, CA 95135 (US)
- **VU, Thuan**
San Jose, CA 95138 (US)
- **LY, Anh**
San Jose, CA 95121 (US)
- **HONG, Stanley**
San Jose, CA 95131 (US)

(74) Representative: **Betten & Resch**
Patent- und Rechtsanwälte PartGmbB
Maximiliansplatz 14
80333 München (DE)

(56) References cited:
- **Xinjie Guo: "Mixed Signal Neurocomputing Based on Floating-gate Memories", , 1 January 2017 (2017-01-01), pages 1-106, XP055627664, Santa Barbara, California Retrieved from the Internet: URL:https://www.alexandria.ucsb.edu/lib/ark:/48907/f3jh3mb0 [retrieved on 2019-10-01]**

Processed by Luminess, 75001 PARIS (FR)

(Cont. next page)

- **CAN LI ET AL: "Long short-term memory networks in memristor crossbars", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 30 May 2018 (2018-05-30), XP081234765,**
- **Matthew R Kucic: "Analog Computing Arrays", , 1 March 2005 (2005-03-01), XP055628600, Retrieved from the Internet: URL:http://hdl.handle.net/1853/4878 [retrieved on 2019-10-03]**

Description

PRIORITY CLAIM

[0001] This application claims priority to U.S. Provisional Patent Application No. 62/720,902, filed on August 21, 2018, and titled, "Analog Neural Memory System for Deep Learning Neural Network Comprising Multiple Vector-By-Matrix Multiplication Arrays and Shared Components," and U.S. Patent Application No. 16/182,492, filed on November 6, 2018, and titled, "Analog Neural Memory System for Deep Learning Neural Network Comprising Multiple Vector-By-Matrix Multiplication Arrays and Shared Components."

FIELD OF THE INVENTION

[0002] Numerous embodiments are disclosed for an analog neuromorphic memory system for use in a deep learning neural network. The analog neuromorphic memory system comprises a plurality of vector-by-matrix multiplication arrays and various components shared by those arrays, including high voltage generation blocks, verify blocks, and testing blocks.

BACKGROUND OF THE INVENTION

[0003] Artificial neural networks mimic biological neural networks (the central nervous systems of animals, in particular the brain) and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks generally include layers of interconnected "neurons" which exchange messages between each other.

[0004] Figure 1 illustrates an artificial neural network, where the circles represent the inputs or layers of neurons. The connections (called synapses) are represented by arrows, and have numeric weights that can be tuned based on experience. This makes neural networks adaptive to inputs and capable of learning. Typically, neural networks include a layer of multiple inputs. There are typically one or more intermediate layers of neurons, and an output layer of neurons that provide the output of the neural network. The neurons at each level individually or collectively make a decision based on the received data from the synapses.

[0005] One of the major challenges in the development of artificial neural networks for high-performance information processing is a lack of adequate hardware technology. Indeed, practical neural networks rely on a very large number of synapses, enabling high connectivity between neurons, i.e. a very high computational parallelism. In principle, such complexity can be achieved with digital supercomputers or specialized graphics processing unit clusters. However, in addition to high cost, these approaches also suffer from mediocre energy efficiency as compared to biological networks, which consume much less energy primarily because they perform low-precision analog computation. CMOS analog circuits have been used for artificial neural networks, but most CMOS-implemented synapses have been too bulky given the high number of neurons and synapses.

[0006] Applicant previously disclosed an artificial (analog) neural network that utilizes one or more non-volatile memory arrays as the synapses in U.S. Patent Application No. 15/594,439. The non-volatile memory arrays operate as an analog neuromorphic memory. The neural network device includes a first plurality of synapses configured to receive a first plurality of inputs and to generate therefrom a first plurality of outputs, and a first plurality of neurons configured to receive the first plurality of outputs. The first plurality of synapses includes a plurality of memory cells, wherein each of the memory cells includes spaced apart source and drain regions formed in a semiconductor substrate with a channel region extending there between, a floating gate disposed over and insulated from a first portion of the channel region and a non-floating gate disposed over and insulated from a second portion of the channel region. Each of the plurality of memory cells is configured to store a weight value corresponding to a number of electrons on the floating gate. The plurality of memory cells is configured to multiply the first plurality of inputs by the stored weight values to generate the first plurality of outputs.

[0007] Each non-volatile memory cells used in the analog neuromorphic memory system must be erased and programmed to hold a very specific and precise amount of charge, i.e., the number of electrons, in the floating gate. For example, each floating gate must hold one of N different values, where N is the number of different weights that can be indicated by each cell. Examples of N include 16, 32, 64, 128, and 256.

[0008] One unique characteristic of analog neuromorphic memory systems is that the system must support two different types of read operations. In a normal read operation, an individual memory cell is read as in conventional memory systems. However, in a neural read operation, the entire array of memory cells is read at one time, where each bit line will output a current that is the sum of all currents from the memory cells connected to that bit line.

[0009] Supporting both types of read operations leads to several challenges. For example, the system must be able to provide a wide range of voltage and current levels for the various operations that are applied to individual cells, entire arrays, or even all of the arrays at once. This requires extensive circuitry outside of the arrays themselves, which can increase the amount of space needed on a semiconductor die for the system, as well as increase power consumption and manufacturing cost.

[0010] What is needed is an improved architecture for an analog neuromorphic memory system that utilizes vector-by-matrix multiplication arrays of flash memory cells that minimizes the amount of circuitry required outside of the arrays themselves.

[0011] The document "Xinjie Guo: Mixed Signal Neurocomputing Based on Floating-gate Memories" refers to neuromorphic networks using highly optimized, nanoscale, non-volatile floating gate memory cells which are used in embedded NOR flash memories.

SUMMARY OF THE INVENTION

[0012] The invention is defined in the appened independent claims. Preferred embodiments are defined in the appended dependent claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013]

Figure 1 is a diagram that illustrates a prior art artificial neural network.
Figure 2 is a cross-sectional side view of a conventional 2-gate non-volatile memory cell.
Figure 3 is a cross-sectional side view of a conventional 4-gate non-volatile memory cell.
Figure 4 is a side cross-sectional side view of conventional 3-gate non-volatile memory cell.
Figure 5 is a cross-sectional side view of another conventional 2-gate non-volatile memory cell.
Figure 6 is a diagram illustrating the different levels of an exemplary artificial neural network utilizing a non-volatile memory array.
Figure 7 is a block diagram illustrating a vector multiplier matrix.
Figure 8 is a block diagram illustrating various levels of a vector multiplier matrix.
Figure 9 depicts another embodiment of a vector multiplier matrix.
Figure 10 depicts another embodiment of a vector multiplier matrix.
Figure 11 depicts another embodiment of a vector multiplier matrix.
Figure 12 depicts another embodiment of a vector multiplier matrix.
Figure 13 depicts another embodiment of a vector multiplier matrix.
Figure 14 depicts a prior art long short term memory system.
Figure 15 depicts an exemplary cell in a prior art long short term memory system.
Figure 16 depicts an implementation of the exemplary cell in a long short term memory system of Figure 15.
Figure 17 depicts another implementation of the exemplary cell in a long short term memory system of Figure 15.
Figure 18 depicts a prior art gated recurrent unit system.
Figure 19 depicts an exemplary cell in a prior art gated recurrent unit system.
Figure 20 depicts an implementation of the exemplary cell in the gated recurrent unit system of Figure 19.
Figure 21 depicts another embodiment of the exemplary cell in the gated recurrent unit system of Figure 19.
Figure 22 depicts a flash analog neuromorphic memory shared architecture system.
Figure 23 depicts a VMM system within the flash analog neuromorphic memory shared architecture system of Figure 22.
Figure 24 depicts an output block for use in the flash analog neuromorphic memory shared architecture system of Figure 22.
Figure 25 depicts an adaptable neuron for use in the flash analog neuromorphic memory shared architecture system of Figure 22.
Figure 26 depicts an activation function circuit for use in the flash analog neuromorphic memory shared architecture system of Figure 22.
Figure 27 depicts an operational amplifier for use in the adaptable neuron of Figure 25.
Figure 28 depicts various blocks used in conjunction with vector-by-matrix multiplication arrays for use in the flash analog neuromorphic memory shared architecture system of Figure 22.
Figure 29 depicts a program and sensing block for use in the flash analog neuromorphic memory shared architecture system of Figure 22.
Figure 30 depicts a reference array system for use in the flash analog neuromorphic memory shared architecture system of Figure 22.
Figure 31 depicts decoding circuitry for use in the flash analog neuromorphic memory shared architecture system

of Figure 22.

Figure 32 depicts decoding circuitry for use in the flash analog neuromorphic memory shared architecture system of Figure 22.

Figure 33 depicts another adaptable neuron for use in the flash analog neuromorphic memory shared architecture system of Figure 22.

Figure 34 depicts sample and hold circuits.

Figure 35 depicts an array architecture that is suitable for memory cells operating in the linear region.

Figure 36 depicts a high voltage generation block for use in the flash analog neuromorphic memory shared architecture system of Figure 22.

Figure 37 depicts a program bias block for use in the flash analog neuromorphic memory shared architecture system of Figure 22.

Figure 38 depicts a sense amplifier circuit for use in the flash analog neuromorphic memory shared architecture system of Figure 22.

DETAILED DESCRIPTION OF THE INVENTION

[0014] The artificial neural networks of the present invention utilize a combination of CMOS technology and non-volatile memory arrays.

Non-Volatile Memory Cells

[0015] Digital non-volatile memories are well known. For example, U.S. Patent 5,029,130 ("the ’130 patent"), discloses an array of split gate non-volatile memory cells, which are a type of flash memory cells. Such a memory cell 210 is shown in Figure 2. Each memory cell 210 includes source region 14 and drain region 16 formed in a semiconductor substrate 12, with a channel region 18 there between. A floating gate 20 is formed over and insulated from (and controls the conductivity of) a first portion of the channel region 18, and over a portion of the source region 14. A word line terminal 22 (which is typically coupled to a word line) has a first portion that is disposed over and insulated from (and controls the conductivity of) a second portion of the channel region 18, and a second portion that extends up and over the floating gate 20. The floating gate 20 and word line terminal 22 are insulated from the substrate 12 by a gate oxide. Bitline 24 is coupled to drain region 16.

[0016] Memory cell 210 is erased (where electrons are removed from the floating gate) by placing a high positive voltage on the word line terminal 22, which causes electrons on the floating gate 20 to tunnel through the intermediate insulation from the floating gate 20 to the word line terminal 22 via Fowler-Nordheim tunneling.

[0017] Memory cell 210 is programmed (where electrons are placed on the floating gate) by placing a positive voltage on the word line terminal 22, and a positive voltage on the source region 14. Electron current will flow from the source region 14 towards the drain region 16. The electrons will accelerate and become heated when they reach the gap between the word line terminal 22 and the floating gate 20. Some of the heated electrons will be injected through the gate oxide onto the floating gate 20 due to the attractive electrostatic force from the floating gate 20.

[0018] Memory cell 210 is read by placing positive read voltages on the drain region 16 and word line terminal 22 (which turns on the portion of the channel region 18 under the word line terminal). If the floating gate 20 is positively charged (i.e. erased of electrons), then the portion of the channel region 18 under the floating gate 20 is turned on as well, and current will flow across the channel region 18, which is sensed as the erased or "1" state. If the floating gate 20 is negatively charged (i.e. programmed with electrons), then the portion of the channel region under the floating gate 20 is mostly or entirely turned off, and current will not flow (or there will be little flow) across the channel region 18, which is sensed as the programmed or "0" state.

[0019] Table No. 1 depicts typical voltage ranges that can be applied to the terminals of memory cell 110 for performing read, erase, and program operations:

Table No. 1: Operation of Flash Memory Cell 210 of Figure 3

	WL	BL	SL
Read	2-3V	0.6-2V	0V
Erase	~11-13V	0V	0V
Program	1-2V	1-3μA	9-10V

[0020] Other split gate memory cell configurations, which are other types of flash memory cells, are known. For example, Figure 3 depicts a four-gate memory cell 310 comprising source region 14, drain region 16, floating gate 20 over a first portion of channel region 18, a select gate 22 (typically coupled to a word line, WL) over a second portion of the channel region 18, a control gate 28 over the floating gate 20, and an erase gate 30 over the source region 14. This configuration is described in U.S. Patent 6,747,310). Here, all gates are non-floating gates except floating gate 20, meaning that they are electrically connected or connectable to a voltage source. Programming is performed by heated electrons from the channel region 18 injecting themselves onto the floating gate 20. Erasing is performed by electrons tunneling from the floating gate 20 to the erase gate 30.

[0021] Table No. 2 depicts typical voltage ranges that can be applied to the terminals of memory cell 310 for performing read, erase, and program operations:

Table No. 2: Operation of Flash Memory Cell 310 of Figure 3

	WL/SG	BL	CG	EG	SL
Read	1.0-2V	0.6-2V	0-2.6V	0-2.6V	0V
Erase	-0.5V/0V	0V	0V/-8V	8-12V	0V
Program	1V	1μA	8-11V	4.5-9V	4.5-5V

[0022] Figure 4 depicts a three-gate memory cell 410, which is another type of flash memory cell. Memory cell 410 is identical to the memory cell 310 of Figure 3 except that memory cell 410 does not have a separate control gate. The erase operation (whereby erasing occurs through use of the erase gate) and read operation are similar to that of the Figure 3 except there is no control gate bias applied. The programming operation also is done without the control gate bias, and as a result, a higher voltage must be applied on the source line during a program operation to compensate for a lack of control gate bias.

[0023] Table No. 3 depicts typical voltage ranges that can be applied to the terminals of memory cell 410 for performing read, erase, and program operations:

Table No. 3: Operation of Flash Memory Cell 410 of Figure 4

	WL/SG	BL	EG	SL
Read	0.7-2.2V	0.6-2V	0-2.6V	0V
Erase	-0.5V/0V	0V	11.5V	0V
Program	1V	2-3 μA	4.5V	7-9V

[0024] Figure 5 depicts stacked gate memory cell 510, which is another type of flash memory cell. Memory cell 510 is similar to memory cell 210 of Figure 2, except that floating gate 20 extends over the entire channel region 18, and control gate 22 (which here will be coupled to a word line) extends over floating gate 20, separated by an insulating layer (not shown). The erase, programming, and read operations operate in a similar manner to that described previously for memory cell 210.

[0025] Table No. 4 depicts typical voltage ranges that can be applied to the terminals of memory cell 510 and substrate 12 for performing read, erase, and program operations:

Table No. 4 Operation of Flash Memory Cell 510 of Figure 5

	CG	BL	SL	Substrate
Read	2-5V	0.6 - 2V	0V	0V
Erase	-8 to -10V/0V	FLT	FLT	8-10V / 15-20V
Program	8-12V	3-5V	0V	0V

[0026] In order to utilize the memory arrays comprising one of the types of non-volatile memory cells described above in an artificial neural network, two modifications are made. First, the lines are configured so that each memory cell can be individually programmed, erased, and read without adversely affecting the memory state of other memory cells in the array, as further explained below. Second, continuous (analog) programming of the memory cells is provided.

[0027] Specifically, the memory state (i.e. charge on the floating gate) of each memory cell in the array can be continuously changed from a fully erased state to a fully programmed state, independently and with minimal disturbance of other memory cells. In another embodiment, the memory state (*i.e.,* charge on the floating gate) of each memory cell in the array can be continuously changed from a fully programmed state to a fully erased state, and vice-versa, independently and with minimal disturbance of other memory cells. This means the cell storage is analog or at the very least can store one of many discrete values (such as 16 or 64 different values), which allows for very precise and individual tuning of all the cells in the memory array, and which makes the memory array ideal for storing and making fine tuning adjustments to the synapsis weights of the neural network.

Neural Networks Employing Non-Volatile Memory Cell Arrays

[0028] Figure 6 conceptually illustrates a non-limiting example of a neural network utilizing a non-volatile memory array of the present embodiments. This example uses the non-volatile memory array neural network for a facial recognition application, but any other appropriate application could be implemented using a non-volatile memory array based neural network.

[0029] S0 is the input layer, which for this example is a 32×32 pixel RGB image with 5 bit precision (i.e. three 32×32 pixel arrays, one for each color R, G and B, each pixel being 5 bit precision). The synapses CB1 going from input layer S0 to layer C1 apply different sets of weights in some instances and shared weights in other instances, and scan the input image with 3×3 pixel overlapping filters (kernel), shifting the filter by 1 pixel (or more than 1 pixel as dictated by the model). Specifically, values for 9 pixels in a 3×3 portion of the image (i.e., referred to as a filter or kernel) are provided to the synapses CB1, where these 9 input values are multiplied by the appropriate weights and, after summing the outputs of that multiplication, a single output value is determined and provided by a first synapse of CB1 for generating a pixel of one of the layers of feature map C1. The 3×3 filter is then shifted one pixel to the right within input layer S0 (i.e., adding the column of three pixels on the right, and dropping the column of three pixels on the left), whereby the 9 pixel values in this newly positioned filter are provided to the synapses CB1, where they are multiplied by the same weights and a second single output value is determined by the associated synapse. This process is continued until the 3x3 filter scans across the entire 32×32 pixel image of input layer S0, for all three colors and for all bits (precision values). The process is then repeated using different sets of weights to generate a different feature map of C1, until all the features maps of layer C1 have been calculated.

[0030] In layer C1, in the present example, there are 16 feature maps, with 30×30 pixels each. Each pixel is a new feature pixel extracted from multiplying the inputs and kernel, and therefore each feature map is a two dimensional array, and thus in this example layer C1 constitutes 16 layers of two dimensional arrays (keeping in mind that the layers and arrays referenced herein are logical relationships, not necessarily physical relationships - i.e., the arrays are not necessarily oriented in physical two dimensional arrays). Each of the 16 feature maps in layer C1 is generated by one of sixteen different sets of synapse weights applied to the filter scans. The C1 feature maps could all be directed to different aspects of the same image feature, such as boundary identification. For example, the first map (generated using a first weight set, shared for all scans used to generate this first map) could identify circular edges, the second map (generated using a second weight set different from the first weight set) could identify rectangular edges, or the aspect ratio of certain features, and so on.

[0031] An activation function P1 (pooling) is applied before going from layer C1 to layer S1, which pools values from consecutive, non-overlapping 2×2 regions in each feature map. The purpose of the pooling function is to average out the nearby location (or a max function can also be used), to reduce the dependence of the edge location for example and to reduce the data size before going to the next stage. At layer S1, there are 16 15×15 feature maps (i.e., sixteen different arrays of 15×15 pixels each). The synapses CB2 going from layer S1 to layer C2 scan maps in S1 with 4×4 filters, with a filter shift of 1 pixel. At layer C2, there are 22 12×12 feature maps. An activation function P2 (pooling) is applied before going from layer C2 to layer S2, which pools values from consecutive non-overlapping 2×2 regions in each feature map. At layer S2, there are 22 6×6 feature maps. An activation function (pooling) is applied at the synapses CB3 going from layer S2 to layer C3, where every neuron in layer C3 connects to every map in layer S2 via a respective synapse of CB3. At layer C3, there are 64 neurons. The synapses CB4 going from layer C3 to the output layer S3 fully connects C3 to S3, i.e. every neuron in layer C3 is connected to every neuron in layer S3. The output at S3 includes 10 neurons, where the highest output neuron determines the class. This output could, for example, be indicative of an identification or classification of the contents of the original image.

[0032] Each layer of synapses is implemented using an array, or a portion of an array, of non-volatile memory cells.

[0033] Figure 7 is a block diagram of an array that can be used for that purpose. Vector-by-matrix multiplication (VMM) array 32 includes non-volatile memory cells and is utilized as the synapses (such as CB1, CB2, CB3, and CB4 in Figure 6) between one layer and the next layer. Specifically, VMM array 32 includes an array of non-volatile memory cells 33, erase gate and word line gate decoder 34, control gate decoder 35, bit line decoder 36 and source line decoder 37, which decode the respective inputs for the non-volatile memory cell array 33. Input to VMM array 32 can be from the erase gate and wordline gate decoder 34 or from the control gate decoder 35. Source line decoder 37 in this example also decodes the output of the non-volatile memory cell array 33. Alternatively, bit line decoder 36 can decode the output of the non-volatile memory cell array 33.

[0034] Non-volatile memory cell array 33 serves two purposes. First, it stores the weights that will be used by the VMM array 32. Second, the non-volatile memory cell array 33 effectively multiplies the inputs by the weights stored in the non-volatile memory cell array 33 and adds them up per output line (source line or bit line) to produce the output, which will be the input to the next layer or input to the final layer. By performing the multiplication and addition function, the non-volatile memory cell array 33 negates the need for separate multiplication and addition logic circuits and is also power efficient due to its in-situ memory computation.

[0035] The output of non-volatile memory cell array 33 is supplied to a differential summer (such as a summing op-amp or a summing current mirror) 38, which sums up the outputs of the non-volatile memory cell array 33 to create a single value for that convolution. The differential summer 38 is arranged to perform summation of positive weight and negative weight.

[0036] The summed up output values of differential summer 38 are then supplied to an activation function circuit 39, which rectifies the output. The activation function circuit 39 may provide sigmoid, tanh, or ReLU functions. The rectified output values of activation function circuit 39 become an element of a feature map as the next layer (e.g. C1 in Figure 6), and are then applied to the next synapse to produce the next feature map layer or final layer. Therefore, in this example, non-volatile memory cell array 33 constitutes a plurality of synapses (which receive their inputs from the prior layer of neurons or from an input layer such as an image database), and summing op-amp 38 and activation function circuit 39 constitute a plurality of neurons.

[0037] The input to VMM array 32 in Figure 7 (WLx, EGx, CGx, and optionally BLx and SLx) can be analog level, binary level, or digital bits (in which case a DAC is provided to convert digital bits to appropriate input analog level) and the output can be analog level, binary level, or digital bits (in which case an output ADC is provided to convert output analog level into digital bits).

[0038] Figure 8 is a block diagram depicting the usage of numerous layers of VMM arrays 32, here labeled as VMM arrays 32a, 32b, 32c, 32d, and 32e. As shown in Figure 8, the input, denoted Inputx, is converted from digital to analog by a digital-to-analog converter 31, and provided to input VMM array 32a. The converted analog inputs could be voltage or current. The input D/A conversion for the first layer could be done by using a function or a LUT (look up table) that maps the inputs Inputx to appropriate analog levels for the matrix multiplier of input VMM array 32a. The input conversion could also be done by an analog to analog (A/A) converter to convert an external analog input to a mapped analog input to the input VMM array 32a.

[0039] The output generated by input VMM array 32a is provided as an input to the next VMM array (hidden level 1) 32b, which in turn generates an output that is provided as an input to the next VMM array (hidden level 2) 32c, and so on. The various layers of VMM array 32 function as different layers of synapses and neurons of a convolutional neural network (CNN). Each VMM array 32a, 32b, 32c, 32d, and 32e can be a stand-alone, physical non-volatile memory array, or multiple VMM arrays could utilize different portions of the same physical non-volatile memory array, or multiple VMM arrays could utilize overlapping portions of the same physical non-volatile memory array. The example shown in Figure 8 contains five layers (32a,32b,32c,32d,32e): one input layer (32a), two hidden layers (32b,32c), and two fully connected layers (32d,32e). One of ordinary skill in the art will appreciate that this is merely exemplary and that a system instead could comprise more than two hidden layers and more than two fully connected layers.

Vector-by-Matrix Multiplication (VMM) Arrays

[0040] Figure 9 depicts neuron VMM array 900, which is particularly suited for memory cells 310 as shown in Figure 3, and is utilized as the synapses and parts of neurons between an input layer and the next layer. VMM array 900 comprises memory array 901 of non-volatile memory cells and reference array 902 (at the top of the array) of non-volatile reference memory cells. Alternatively, another reference array can be placed at the bottom.

[0041] In VMM array 900, control gate lines, such as control gate line 903, run in a vertical direction (hence reference array 902 in the row direction is orthogonal to control gate line 903), and erase gate lines, such as erase gate line 904, run in a horizontal direction. Here, the inputs to VMM array 900 are provided on the control gate lines (CG0, CG1, CG2, CG3), and the output of VMM array 900 emerges on the source lines (SL0, SL1). In one embodiment, only even rows are used, and in another embodiment, only odd rows are used. The current placed on each source line (SL0, SL1, respectively) performs a summing function of all the currents from the memory cells connected to that particular source line.

[0042] As described herein for neural networks, the non-volatile memory cells of VMM array 900, i.e. the flash memory of VMM array 900, are preferably configured to operate in a sub-threshold region.

[0043] The non-volatile reference memory cells and the non-volatile memory cells described herein are biased in weak inversion:

$$Ids = Io * e^{(Vg- Vth)/kVt} = w * Io * e^{(Vg)/kVt},$$

where w = e $^{(-Vth)/kVt}$

[0044] For an I-to-V log converter using a memory cell (such as a reference memory cell or a peripheral memory cell) or a transistor to convert input current into an input voltage:

$$Vg= k*Vt*\log [Ids/wp*Io]$$

Here, wp is w of a reference or peripheral memory cell.

[0045] For a memory array used as a vector matrix multiplier VMM array, the output current is:

$$Iout = wa * Io * e^{(Vg)/kVt}, ; \text{namely}$$

$$Iout = (wa/wp) * Iin = W * Iin$$

$$W = e^{(Vthp - Vtha)/kVt}$$

Here, wa = w of each memory cell in the memory array.

[0046] A wordline or control gate can be used as the input for the memory cell for the input voltage.

[0047] Alternatively, the flash memory cells of VMM arrays described herein can be configured to operate in the linear region:

$$Ids = beta* (Vgs-Vth)*Vds ; beta = u*Cox*W/L$$

$$W = \alpha (Vgs-Vth)$$

[0048] A wordline or control gate or bitline or sourceline can be used as the input for the memory cell operated in the linear region for the input voltage.

[0049] For an I-to-V linear converter, a memory cell (such as a reference memory cell or a peripheral memory cell) or a transistor operating in the linear region can be used to linearly convert an input/output current into an input/output voltage.

[0050] Other embodiments for VMM array 32 of Figure 7 are described in U.S. Patent Application No. Application No. 15/826,345. As described in that application, a sourceline or a bitline can be used as the neuron output (current summation output).

[0051] Figure 10 depicts neuron VMM array 1000, which is particularly suited for memory cells 210 as shown in Figure 2, and is utilized as the synapses between an input layer and the next layer. VMM array 1000 comprises a memory array 1003 of non-volatile memory cells, reference array 1001 of first non-volatile reference memory cells, and reference array 1002 of second non-volatile reference memory cells. Reference arrays 1001 and 1002, arranged in the column direction of the array, serve to convert current inputs flowing into terminals BLR0, BLR1, BLR2, and BLR3 into voltage inputs WL0, WL1, WL2, and WL3. In effect, the first and second non-volatile reference memory cells are diode-connected through multiplexors 1014 with current inputs flowing into them. The reference cells are tuned (e.g., programmed) to target reference levels. The target reference levels are provided by a reference mini-array matrix (not shown).

[0052] Memory array 1003 serves two purposes. First, it stores the weights that will be used by the VMM array 1000 on respective memory cells thereof. Second, memory array 1003 effectively multiplies the inputs (i.e. current inputs provided in terminals BLR0, BLR1, BLR2, and BLR3, which reference arrays 1001 and 1002 convert into the input voltages to supply to wordlines WL0, WL1, WL2, and WL3) by the weights stored in the memory array 1003 and then adds all the results (memory cell currents) to produce the output on the respective bit lines (BL0 - BLN), which will be the input to the next layer or input to the final layer. By performing the multiplication and addition function, memory array 1003 negates the need for separate multiplication and addition logic circuits and is also power efficient. Here, the voltage inputs are provided on the word lines WL0, WL1, WL2, and WL3, and the output emerges on the respective bit lines BL0 - BLN during a read (inference) operation. The current placed on each of the bit lines BL0 - BLN performs a summing function of the currents from all non-volatile memory cells connected to that particular bitline.

[0053] Table No. 5 depicts operating voltages for VMM array 1000. The columns in the table indicate the voltages placed on word lines for selected cells, word lines for unselected cells, bit lines for selected cells, bit lines for unselected cells, source lines for selected cells, and source lines for unselected cells. The rows indicate the operations of read, erase, and program.

Table No. 5 Operation of VMM Array 1000 of Figure 10:

	WL	WL -unsel	BL	BL -unsel	SL	SL -unsel
Read	1-3.5V	-0.5V/0V	0.6-2V (Ineuron)	0.6V-2V/0V	0V	0V
Erase	~5-13V	0V	0V	0V	0V	0V
Program	1-2V	-0.5V/0V	0.1-3 uA	Vinh -2.5V	4-10V	0-1V/FLT

[0054] Figure 11 depicts neuron VMM array 1100, which is particularly suited for memory cells 210 as shown in Figure 2, and is utilized as the synapses and parts of neurons between an input layer and the next layer. VMM array 1100 comprises a memory array 1103 of non-volatile memory cells, reference array 1101 of first non-volatile reference memory cells, and reference array 1102 of second non-volatile reference memory cells. Reference arrays 1101 and 1102 run in row direction of the VMM array 1100. VMM array is similar to VMM 1000 except that in VMM array 1100, the word lines run in the vertical direction. Here, the inputs are provided on the word lines (WLA0, WLB0, WLA1, WLB2, WLA2, WLB2, WLA3, WLB3), and the output emerges on the source line (SL0, SL1) during a read operation. The current placed on each source line performs a summing function of all the currents from the memory cells connected to that particular source line.

[0055] Table No. 6 depicts operating voltages for VMM array 1100. The columns in the table indicate the voltages placed on word lines for selected cells, word lines for unselected cells, bit lines for selected cells, bit lines for unselected cells, source lines for selected cells, and source lines for unselected cells. The rows indicate the operations of read, erase, and program.

Table No. 6: Operation of VMM Array 1100 of Figure 11

	WL	WL -unsel	BL	BL -unsel	SL	SL -unsel
Read	1-3.5V	-0.5V/0V	0.6-2V	0.6V-2V/0V	~0.3-1V (Ineuron)	0V
Erase	~5-13V	0V	0V	0V	0V	SL-inhibit (~4-8V)
Program	1-2V	-0.5V/0V	0.1-3 uA	Vinh -2.5V	4-10V	0-1V/FLT

[0056] Figure 12 depicts neuron VMM array 1200, which is particularly suited for memory cells 310 as shown in Figure 3, and is utilized as the synapses and parts of neurons between an input layer and the next layer. VMM array 1200 comprises a memory array 1203 of non-volatile memory cells, reference array 1201 of first non-volatile reference memory cells, and reference array 1202 of second non-volatile reference memory cells. Reference arrays 1201 and 1202 serve to convert current inputs flowing into terminals BLR0, BLR1, BLR2, and BLR3 into voltage inputs CG0, CG1, CG2, and CG3. In effect, the first and second non-volatile reference memory cells are diode-connected through multiplexors 1212 with current inputs flowing into them through BLR0, BLR1, BLR2, and BLR3. Multiplexors 1212 each include a respective multiplexor 1205 and a cascoding transistor 1204 to ensure a constant voltage on the bitline (such as BLR0) of each of the first and second non-volatile reference memory cells during a read operation. The reference cells are tuned to target reference levels.

[0057] Memory array 1203 serves two purposes. First, it stores the weights that will be used by the VMM array 1200. Second, memory array 1203 effectively multiplies the inputs (current inputs provided to terminals BLR0, BLR1, BLR2, and BLR3, for which reference arrays 1201 and 1202 convert these current inputs into the input voltages to supply to the control gates (CG0, CG1, CG2, and CG3) by the weights stored in the memory array and then add all the results (cell currents) to produce the output, which appears on BL0 - BLN, and will be the input to the next layer or input to the final layer. By performing the multiplication and addition function, the memory array negates the need for separate multiplication and addition logic circuits and is also power efficient. Here, the inputs are provided on the control gate lines (CG0, CG1, CG2, and CG3), and the output emerges on the bitlines (BL0 - BLN) during a read operation. The current placed on each bitline performs a summing function of all the currents from the memory cells connected to that particular bitline.

[0058] VMM array 1200 implements uni-directional tuning for non-volatile memory cells in memory array 1203. That is, each non-volatile memory cell is erased and then partially programmed until the desired charge on the floating gate is reached. This can be performed, for example, using the novel precision programming techniques described below. If too much charge is placed on the floating gate (such that the wrong value is stored in the cell), the cell must be erased and the sequence of partial programming operations must start over. As shown, two rows sharing the same erase gate (such as EG0 or EG1) need to be erased together (which is known as a page erase), and thereafter, each cell is partially programmed until the desired charge on the floating gate is reached.

[0059] Table No. 7 depicts operating voltages for VMM array 1200. The columns in the table indicate the voltages placed on word lines for selected cells, word lines for unselected cells, bit lines for selected cells, bit lines for unselected cells, control gates for selected cells, control gates for unselected cells in the same sector as the selected cells, control gates for unselected cells in a different sector than the selected cells, erase gates for selected cells, erase gates for unselected cells, source lines for selected cells, and source lines for unselected cells. The rows indicate the operations of read, erase, and program.

Table No. 7: Operation of VMM Array 1200 of Figure 12

	WL	WL-unsel	BL	BL-unsel	CG	CG - unsel same sector	CG-unsel	EG	EG-unsel	SL	SL - unsel
Read	1.0-2V	-0.5V/ 0V	0.6-2V (Ineuron)	0V	0-2.6V	0-2.6V	0-2.6V	0-2.6V	0-2.6V	0V	0V
Erase	0V	0V	0V	0V	0V	0-2.6V	0-2.6V	5-12V	0-2.6V	0V	0V
Program	0.7-1V	-0.5V/0V	0.1-1uA	Vinh (1-2V)	4-11V	0-2.6V	0-2.6V	4.5-5V	0-2.6V	4.5-5V	0-1V

[0060] Figure 13 depicts neuron VMM array 1300, which is particularly suited for memory cells 310 as shown in Figure 3, and is utilized as the synapses and parts of neurons between an input layer and the next layer. VMM array 1300 comprises a memory array 1303 of non-volatile memory cells, reference array 1301 or first non-volatile reference memory cells, and reference array 1302 of second non-volatile reference memory cells. EG lines EGR0, EG0, EG1 and EGR1 are run vertically while CG lines CG0, CG1, CG2 and CG3 and SL lines WL0, WL1, WL2 and WL3 are run horizontally. VMM array 1300 is similar to VMM array 1400, except that VMM array 1300 implements bi-directional tuning, where each individual cell can be completely erased, partially programmed, and partially erased as needed to reach the desired amount of charge on the floating gate due to the use of separate EG lines. As shown, reference arrays 1301 and 1302 convert input current in the terminal BLR0, BLR1, BLR2, and BLR3 into control gate voltages CG0, CG1, CG2, and CG3 (through the action of diode-connected reference cells through multiplexors 1314) to be applied to the memory cells in the row direction. The current output (neuron) is in the bitlines BL0 - BLN, where each bit line sums all currents from the non-volatile memory cells connected to that particular bitline.

[0061] Table No. 8 depicts operating voltages for VMM array 1300. The columns in the table indicate the voltages placed on word lines for selected cells, word lines for unselected cells, bit lines for selected cells, bit lines for unselected cells, control gates for selected cells, control gates for unselected cells in the same sector as the selected cells, control gates for unselected cells in a different sector than the selected cells, erase gates for selected cells, erase gates for unselected cells, source lines for selected cells, and source lines for unselected cells. The rows indicate the operations of read, erase, and program.

Table No. 8: Operation of VMM Array 1300 of Figure 13

	WL	WL-unsel	BL	BL-unsel	CG	CG -unsel same sector	CG-unsel	EG	EG-unsel	SL	SL - unsel
Read	1.0-2V	-0.5 V/ 0V	0.6-2V (Ineuron)	0V	0-2.6V	0-2.6V	0-2.6V	0-2.6V	0-2.6V	0V	0V
Erase	0V	0V	0V	0V	0V	4-9V	0-2.6V	5-12V	0-2.6V	0V	0V
Program	0.7-1V	-0.5 V/ 0V	0.1-1uA	Vinh (1-2V)	4-11V	0-2.6V	0-2.6V	4.5-5V	0-2.6V	4.5-5V	0-1V

Long Short-Term Memory

[0062] The prior art includes a concept known as long short-term memory (LSTM). LSTM units often are used in neural networks. LSTM allows a neural network to remember information over predetermined arbitrary time intervals and to use that information in subsequent operations. A conventional LSTM unit comprises a cell, an input gate, an output gate, and a forget gate. The three gates regulate the flow of information into and out of the cell and the time interval that the information is remembered in the LSTM. VMMs are particularly useful in LSTM units.

[0063] Figure 14 depicts an exemplary LSTM 1400. LSTM 1400 in this example comprises cells 1401, 1402, 1403, and 1404. Cell 1401 receives input vector x_0 and generates output vector h_0 and cell state vector c_0. Cell 1402 receives input vector x_1, the output vector (hidden state) h_0 from cell 1401, and cell state c_0 from cell 1401 and generates output vector h_1 and cell state vector c_1. Cell 1403 receives input vector x_2, the output vector (hidden state) h_1 from cell 1402, and cell state c_1 from cell 1402 and generates output vector h_2 and cell state vector c_2. Cell 1404 receives input vector x_3, the output vector (hidden state) h_2 from cell 1403, and cell state c_2 from cell 1403 and generates output vector h_3. Additional cells can be used, and an LSTM with four cells is merely an example.

[0064] Figure 15 depicts an exemplary implementation of an LSTM cell 1500, which can be used for cells 1401, 1402, 1403, and 1404 in Figure 14. LSTM cell 1500 receives input vector x(t), cell state vector c(t-1) from a preceding cell, and output vector h(t-1) from a preceding cell, and generates cell state vector c(t) and output vector h(t).

[0065] LSTM cell 1500 comprises sigmoid function devices 1501, 1502, and 1503, each of which applies a number between 0 and 1 to control how much of each component in the input vector is allowed through to the output vector. LSTM cell 1500 also comprises tanh devices 1504 and 1505 to apply a hyperbolic tangent function to an input vector, multiplier devices 1506, 1507, and 1508 to multiply two vectors together, and addition device 1509 to add two vectors together. Output vector h(t) can be provided to the next LSTM cell in the system, or it can be accessed for other purposes.

[0066] Figure 16 depicts an LSTM cell 1600, which is an example of an implementation of LSTM cell 1500. For the reader's convenience, the same numbering from LSTM cell 1500 is used in LSTM cell 1600. Sigmoid function devices 1501, 1502, and 1503 and tanh device 1504 each comprise multiple VMM arrays 1601 and activation circuit blocks 1602. Thus, it can be seen that VMM arrays are particular useful in LSTM cells used in certain neural network systems.

[0067] An alternative to LSTM cell 1600 (and another example of an implementation of LSTM cell 1500) is shown in Figure 17. In Figure 17, sigmoid function devices 1501, 1502, and 1503 and tanh device 1504 share the same physical hardware (VMM arrays 1701 and activation function block 1702) in a time-multiplexed fashion. LSTM cell 1700 also comprises multiplier device 1703 to multiply two vectors together, addition device 1708 to add two vectors together, tanh device 1505 (which comprises activation circuit block 1702), register 1707 to store the value i(t) when i(t) is output from sigmoid function block 1702, register 1704 to store the value f(t) * c(t-1) when that value is output from multiplier device 1703 through multiplexor 1710, register 1705 to store the value i(t) * u(t) when that value is output from multiplier device 1703 through multiplexor 1710, and register 1706 to store the value o(t) * c~(t) when that value is output from multiplier device 1703 through multiplexor 1710, and multiplexor 1709.

[0068] Whereas LSTM cell 1600 contains multiple sets of VMM arrays 1601 and respective activation function blocks 1602, LSTM cell 1700 contains only one set of VMM arrays 1701 and activation function block 1702, which are used to represent multiple layers in the embodiment of LSTM cell 1700. LSTM cell 1700 will require less space than LSTM 1600, as LSTM cell 1700 will require 1/4 as much space for VMMs and activation function blocks compared to LSTM cell 1600.

[0069] It can be further appreciated that LSTM units will typically comprise multiple VMM arrays, each of which requires functionality provided by certain circuit blocks outside of the VMM arrays, such as a summer and activation circuit block and high voltage generation blocks. Providing separate circuit blocks for each VMM array would require a significant amount of space within the semiconductor device and would be somewhat inefficient. The embodiments described below therefore attempt to minimize the circuitry required outside of the VMM arrays themselves.

Gated Recurrent Units

[0070] An analog VMM implementation can be utilized for a GRU (gated recurrent unit) system. GRUs are a gating mechanism in recurrent neural networks. GRUs are similar to LSTMs, except that GRU cells generally contain fewer components than an LSTM cell.

[0071] Figure 18 depicts an exemplary GRU 1800. GRU 1800 in this example comprises cells 1801, 1802, 1803, and 1804. Cell 1801 receives input vector x_0 and generates output vector h_0. Cell 1802 receives input vector x_1, the output vector (hidden state) h_0 from cell 1801 and generates output vector h_1. Cell 1803 receives input vector x_2 and the output vector (hidden state) h_1 from cell 1802 and generates output vector h_2. Cell 1804 receives input vector x_3 and the output vector (hidden state) h_2 from cell 1803 and generates output vector h_3. Additional cells can be used, and an GRU with four cells is merely an example.

[0072] Figure 19 depicts an exemplary implementation of a GRU cell 1900, which can be used for cells 1801, 1802, 1803, and 1804 of Figure 18. GRU cell 1900 receives input vector x(t) and output vector h(t-1) from a preceding GRU cell and generates output vector h(t). GRU cell 1900 comprises sigmoid function devices 1901 and 1902, each of which applies a number between 0 and 1 to components from output vector h(t-1) and input vector x(t). GRU cell 1900 also comprises a tanh device 1903 to apply a hyperbolic tangent function to an input vector, a plurality of multiplier devices 1904, 1905, and 1906 to multiply two vectors together, an addition device 1907 to add two vectors together, and a complementary device 1908 to subtract an input from 1 to generate an output.

[0073] Figure 20 depicts a GRU cell 2000, which is an example of an implementation of GRU cell 1900. For the reader's convenience, the same numbering from GRU cell 1900 is used in GRU cell 2000. As can be seen in Figure 20, sigmoid function devices 1901 and 1902, and tanh device 1903 each comprise multiple VMM arrays 2001 and activation function blocks 2002. Thus, it can be seen that VMM arrays are of particular use in GRU cells used in certain neural network systems.

[0074] An alternative to GRU cell 2000 (and another example of an implementation of GRU cell 1900) is shown in Figure 21. In Figure 21, GRU cell 2100 utilizes VMM arrays 2101 and activation function block 2102, which when configured as a sigmoid function applies a number between 0 and 1 to control how much of each component in the input vector is allowed through to the output vector. In Figure 21, sigmoid function devices 1901 and 1902 and tanh device 1903 share the same physical hardware (VMM arrays 2101 and activation function block 2102) in a time-multiplexed fashion. GRU cell 2100 also comprises multiplier device 2103 to multiply two vectors together, addition device 2105 to add two vectors together, complementary device 2109 to subtract an input from 1 to generate an output, multiplexor 2104, register 2106 to hold the value h(t-1) * r(t) when that value is output from multiplier device 2103 through multiplexor 2104, register 2107 to hold the value h(t-1) *z(t) when that value is output from multiplier device 2103 through multiplexor 2104, and register 2108 to hold the value h^(t) * (1-z(t)) when that value is output from multiplier device 2103 through multiplexor 2104.

[0075] Whereas GRU cell 2000 contains multiple sets of VMM arrays 2001 and activation function blocks 2002, GRU cell 2100 contains only one set of VMM arrays 2101 and activation function block 2102, which are used to represent multiple layers in the embodiment of GRU cell 2100. GRU cell 2100 will require less space than GRU cell 2000, as GRU cell 2100 will require 1/3 as much space for VMMs and activation function blocks compared to GRU cell 2000.

[0076] It can be further appreciated that GRU systems will typically comprise multiple VMM arrays, each of which requires functionality provided by certain circuit blocks outside of the VMM arrays, such as a summer and activation circuit block and high voltage generation blocks. Providing separate circuit blocks for each VMM array would require a significant amount of space within the semiconductor device and would be somewhat inefficient. The embodiments described below therefore attempt to minimize the circuitry required outside of the VMM arrays themselves.

[0077] The input to the VMM arrays can be an analog level, a binary level, or digital bits (in this case a DAC is needed to convert digital bits to appropriate input analog level) and the output can be an analog level, a binary level, or digital bits (in this case an output ADC is needed to convert output analog level into digital bits).

[0078] For each memory cell in a VMM array, each weight w can be implemented by a single memory cell or by a differential cell or by two blend memory cells (average of 2 cells). In the differential cell case, two memory cells are needed to implement a weight w as a differential weight (w = w+ - w-). In the two blend memory cells, two memory cells are needed to implement a weight w as an average of two cells.

Flash Analog Neuromorphic Memory Shared Architecture System

[0079] Figure 22 depicts flash analog neuromorphic memory shared architecture system 2200, which comprises VMM systems 2221, 2222, and 2223 and shared circuit blocks 2217. VMM system 2221 comprises macro blocks 2201 and 2202 and output block 2207, the latter of which can comprise a summer, an analog-to-digital converter, or another type of functional block, and provides an output for macro blocks 2201 and 2202. VMM system 2222 comprises macro blocks 2203 and 2204 and output block 2208, the latter of which can comprise a summer, an analog-to-digital converter, or another type of functional block, and provides an output for macro blocks macro blocks 2203 and 2204. VMM system 2223 comprises macro blocks 2205 and 2206 and output block 2209, the latter of which can comprise a summer, an analog-to-digital converter, or another type of functional block, and provides an output for macro blocks macro blocks 2205 and 2206. As discussed in greater detail with respect to Figure 23, each macro block, such as macro blocks 2201, 2202, 2203, 2204, 2205, and 2206, contain one VMM array.

[0080] Shared circuit blocks 2217 are used by VMM systems 2221, 2222, and 2223. In this example, shared circuit blocks 2217 include:

- analog circuit block 2210;
- high voltage generation block 2211;
- verify block 2212;
- system control block 2213;
- array reference block 2214; and

- sensing block 2215.

[0081] Analog circuit block 2210 contains analog circuitry for performing certain analog functions required by macro blocks 2201, 2202, 2203, 2204, 2205, and 2206 during operation such as to provide reference voltage, timing, and current for program, erase, read, and verify operation. Verify operation is used to confirm a target weight (meaning certain floating gate charge) is reached during erase or program.

[0082] High voltage generation block 2211 provides various high voltages required by macro blocks 2201, 2202, 2203, 2204, 2205, and 2206 during various operations, such as program operations and erase operations. Optionally, high voltage generation block 2211 provides those high voltages concurrently (with sufficient voltage and current) to two or more of macro blocks 2201, 2202, 2203, 2204, 2205, and 2206, and optionally, program operations can occur concurrently within two or more of macro blocks 2201, 2202, 2203, 2204, 2205, and 2206 in response to a single command or multiple commands, and optionally, erase operations can occur concurrently within two or more of macro blocks 2201, 2202, 2203, 2204, 2205, and 2206 in response to a single command or multiple commands.

[0083] Verify block 2212 performs a verify operation as part of a write-and-verify operation on macro blocks 2201, 2202, 2203, 2204, 2205, and 2206 or a portion thereof during write operations. Optionally, the verify block 2212 can perform verify operations concurrently on two or more of macro blocks 2201, 2202, 2203, 2204, 2205, and 2206. Verify block 2212 comprises a sensing block (such as the sensing portion of program and sensing block 2900 depicted in Figure 29).

[0084] System control block 2213 provides various system control functions, such as trimming of various components (such as the adjustable resistors, transistors, and current sources discussed below) using trimming block 2216, as well as testing. It also provides macro/core interface command control logic and write algorithm. It also provides control logic for component sharing across multiple macros or cores,

[0085] Array reference block 2214 provides reference memory cells for use during sense or verify operations within macro blocks 2201, 2202, 2203, 2204, 2205, and 2206. Alternatively, the sense or verify may use reference levels provided by a resistor, a MOSFET, or a bandgap-based bias.

[0086] Sensing block 2215 performs a sense operation on macro blocks 2201, 2202, 2203, 2204, 2205, and 2206 or a portion thereof during write operations. Optionally, sensing block 2215 can perform sense operations concurrently on two or more of macro blocks 2201, 2202, 2203, 2204, 2205, and 2206. Sensing block 2215 can comprise the sensing portion of program and sensing block 2900 depicted in Figure 29.

[0087] Table 9 depicts operation modes for flash analog neuromorphic memory shared architecture system 2200 of Figure 22. The columns in the table shown indicate (in order from left to right) the state of high voltage generation block 2211, verify block 2212, summer and activation circuit blocks 2207, 2208, and 2209, analog circuit block 2210, array reference block 2214, and each of the VMM systems contained in system 2200. Macro mode selection is for MACRO1 (which is a selected macro block, such as macro block 2201, 2202, 2203, 2204, 2205, or 2206). In system mode, MACRO is on if selected.

[0088] The rows in the table shown indicate (in order from top to bottom):

- system mass erase operation, where selected cells in all selected macros in system 2200 are erased);
- system mass program operation, where selected cells in all selected macros in system 2200 are programmed; program high voltage compensation is done per macro (using a macro high voltage compensation block), meaning compensation is done locally at the macro level, for example at each macro, Icomp=number of unprogrammed bits*Iprog; alternatively high voltage compensation is done per system level (using a system level high voltage compensation block), for example in this case the macro with the most un-programmed bit is used to compensate at the high voltage generation circuit (hvgen), for example at the hvgen,

$$\text{Icomp=number of unprogrammed bits*Iprog.}$$

- system read/verify operation, where selected cells in all selected macros in system 2200 are read and verified; for reading ’0’ (programmed state) for multiple cells in multiple cores, a reference ’0’ margin I-M0 current is used in current sensing to detect if summed selected cell current > I-M0, then it fails reading ‘0’; for reading ’1’ (programmed state) for multiple cells in multiple cores, a reference ‘1’ margin k*I-M1 current is used in current sensing to detect if summed selected cell current < k*I-M1, then it fails reading ’1’, for example for reading 2 cells in parallel, k=2.
- macro erase operation, where only one macro block, here the one labeled MACRO1, is erased; a sector (consisting of multiple rows) can be erased with a macro sector erase or whole array can be erased by a macro mass erase.
- macro program operation, where only one macro block, here the one labeled MACRO1, is programmed; a word (consisting of multiple cells in multiple columns) can be programmed with a macro word programmed or selected mass array with multiple rows and/or multiple columns can be programmed by a macro mass program.

- macro read/verify operation, where only one macro block, here the one labeled MACRO1, is read and verified; and
- read neural operation, where all cells in a single macro block are read at one time.

Table No. 9: Operation Modes For Flash Analog Neuromorphic Memory Shared Architecture System 2200 of Figure 22

MODE	HVGEN	VFYBLK	fOBLK	ANABLK	ARYREFBLK
System Mass Erase	ON	OFF	OFF	ON	OFF
System Mass Program	ON	OFF	OFF	ON	OFF
System ReadVerify	OFF	ON	OFF	ON	ON
Macro Erase (Sector/Mass)	ON	OFF	OFF	ON	OFF
Macro Program (Word/Mass)	ON	OFF	OFF	ON	OFF
Macro Read Verify (Word/Mass)	OFF	ON	OFF	ON	ON
Read Neural	OFF	OFF	ON	ON	ON

MODE	MACRO1	MACRO2	MACRON-1	MACRON
System Mass Erase	Y (sel)	Y (sel)	Y (sel)	Y (sel)
System Mass Program	Y (sel)	Y (sel)	Y (sel)	Y (sel)

System ReadVerify	Y (sel)	Y (sel)	Y (sel)	Y (sel)
Macro Erase (Sector/Mass)	Y (sel)	N (unsel)	N (unsel)	N (unsel)
Macro Program (Word/Mass)	Y (sel)	N (unsel)	N (unsel)	N (unsel)
Macro Read Verify (Word/Mass)	Y (sel)	N (unsel)	N (unsel)	N (unsel)
Read Neural	Y	Y	Y	Y

[0089] Figure 23 depicts VMM system 2300 (which can be used to implement VMM systems 2221, 2222, and 2223 in Figure 22). VMM system 2300 comprises macro block 2320 (which can be used to implement macro blocks 2201, 2202, 2203, 2204, 2205, and 2206 in Figure 22) and activation function block 2314 and output block 2313, the latter of which can comprise a summer, an analog-to-digital converter, or another type of functional block, and provides an output for VMM system 2300.

[0090] Macro block 2320 comprises VMM array 2301, low voltage row decoder 2302, high voltage row decoder 2303, and low voltage reference column decoder 2304. Low voltage row decoder 2302 provides a bias voltage for read and program operations and provides a decoding signal for high voltage row decoder 2303. High voltage row decoder 2303 provides a high voltage bias signal for program and erase operations.

[0091] Macro block 2320 further comprises redundancy arrays 2305 and 2306. Redundancy arrays 2305 and 2306 provides array redundancy for replacing a defective portion in array 2301. VMM system 2300 further comprises NVR (non-volatile register, aka info sector) sector 2307, which are array sectors used to store, inter alia, user info, device ID, password, security key, trimbits, configuration bits, manufacturing info. Macro block 2320 further comprises reference sector 2308 for providing reference cells to be used in a sense operation; predecoder 2309 for decoding addresses for decoders 2302, 2303, and/or 2304; bit line multiplexor 2310; macro control logic 2311; and macro analog circuit block 2312, each of which performs functions at the macro block or VMM array level (as opposed to the system level comprising all VMM arrays).

[0092] Examples of embodiments of the circuit blocks shown in Figures 22 and 23 will now be described.

[0093] Figure 24 depicts output block 2400 (which can be used as output blocks 2207, 2208, 2209 in Figure 22 and output block 2313 in Figure 23). In this example, output block 2400 comprises a plurality of individual summer and activation circuit blocks such as summer and activation block 2401.

[0094] Figure 25 depicts adaptable neuron circuit 2500 that comprises on an op amp that provides low impedance output, for summing multiple current signals and converting the summed current signal into a voltage signal, and which is an embodiment of each summer block within summer block 2601a, ..., 2601i in Figure 26. Adaptable neuron circuit 2500 receives current from a VMM, such as VMM array 2401 (labeled I_NEU), which here is represented as current source 2502, which is provided to the inverting input of operational amplifier 2501. The non-inverting input of operational amplifier 2501 is coupled to a voltage source (labeled VREF). The output (labeled VO) of operational amplifier 2501 is coupled to NMOS R_NEU transistor 2503, which acts as a variable resistor of effective resistance R_NEU in response to the signal VCONTROL, which is applied to the gate of NMOS transistor 2503. The output voltage, Vo, is equal to I_NEU * R_NEU - VREF. The maximum value of I_NFU depends on the number of synapses and weight value contained in the VMM. R_NEU is a variable resistance and can be adapted to the VMM size it is coupled to. Further, the power of the summing operational amplifier 2501 is adjusted in relation the value of the R_NEU transistor 2503 to minimize power consumption. As the value of R_NEU transistor 2503 increases, the bias (i.e., power) of the operational amplifier 2501 is reduced via current bias IBIAS_OPA 2504 and vice versa. Since the op amp based summer circuit can provide low impedance output, it is suitable to be configured to drive a long interconnect and heavier loading.

[0095] Figure 26 depicts activation function circuit 2600. Activation function circuit 2600 can be used for activation circuit blocks 2203a, 2203b, 2203c, 2203d, 2203e, and 2203f in Figure 22 and activation circuit blocks 2303a, 2303b, 2303c, 2303d, 2303e, and 2303f in Figure 23, and activation block 2414 in Figure 24.

[0096] Activation function circuit 2600 converts an input voltage pair (Vin+ and Vin-) into a current (Iout_neu) using a tanh function, and which can be used with the VMM arrays described above. Activation function circuit 2600 comprises PMOS transistors 2601, 2602, 2603, 2604, 2605, and 2606 and NMOS transistors 2607, 2608, 2609, and 2610, configured as shown. The transistors 2603, 2604, and 2606 serve as cascoding transistors. The input NMOS pair 2607 and 2608 operates in sub-threshold region to realize the tanh function. The current I_neu_max is the maximum neuron current that can be received from the attached VMM (not shown).

[0097] Figure 27 depicts operational amplifier 2700 that can be used as operational amplifier 2501 in Figure 25. Operational amplifier 2700 comprises PMOS transistors 2701, 2702, and 2705, NMOS transistors 2703, 2704, 2706, and 2707, and NMOS transistor 2708 that acts as a variable bias, in the configuration shown. The input terminals to operational amplifier 2700 are labeled Vin+ (applied to the gate of NMOS transistor 2704) and Vin- (applied to the gate of NMOS transistor 2703), and the output is Vout. The bias current Ibias_opa is provided to the drain of NMOS transistor 2708.

[0098] Figure 28 depicts high voltage generation block 2800, control logic block 2804, analog circuit block 2805, and test block 2808.

[0099] High voltage generation block 2800 comprises charge pump 2801, charge pump regulator 2802, and high voltage operational amplifier 2803. The voltage of the output of charge pump regulator 2802 can be controlled using the control bits TRBIT_SL<N:0> that are applied to gates of NMOS transistors in charge pump regulator 2802. Control logic block 2804 receives control logic inputs and generates control logic outputs. Analog circuit block 2805 comprises current bias generator 2806 for receiving a reference voltage, VREF, and generating a current that can be used to generate a bias signal, IBIAS, which can be used, for example, as IBIAS_OPA 2504 in Figure 25.. Analog circuit block 2805 also comprises voltage generator 2807 for receiving reference voltage VREF and a set of trim bits, TRBIT _WL, and generating a voltage to apply to word lines during various operations. Test block 2808 receives signals on a test pad, MONHV_PAD, and outputs various signals for a designer to monitor during testing.

[0100] Figure 29 depicts program and sensing block 2900, which can be used during program and verify operations and can be coupled to one or more VMM systems. Program and sensing block 2900 comprises a plurality of individual program and sense circuit blocks 2901a, 2901b, ... 2901j, each of which can read a "0" or "1" in a selected memory cell. Controller or control logic 2910 can activate the appropriate program and sense circuit blocks 2901a, 2901b, ... 2901j during each cycle as needed.

[0101] Figure 30 depicts reference system 3000, which can be used in place of reference sector 2308 in Figure 23. Reference system 3000 comprises reference array 3002, low voltage row decoder 3001, high voltage row decoder 3003, and low voltage reference column decoder 3004. Low voltage row decoder 3001 provides a bias voltage for read and program operations and provides a decoding signal for high voltage row decoder 3003. High voltage row decoder 3003 provides a high voltage bias signal for program and erase operations.

[0102] Figure 31 depicts VMM high voltage decode circuits, comprising word line decoder circuit 3101, source line decoder circuit 3104, and high voltage level shifter 3108, which are appropriate for use with memory cells of the type shown in Figure 2.

[0103] Word line decoder circuit 3101 comprises PMOS select transistor 3102 (controlled by signal HVO_B) and NMOS de-select transistor 3103 (controlled by signal HVO_B) configured as shown.

[0104] Source line decoder circuit 3104 comprises NMOS monitor transistors 3105 (controlled by signal HVO), driving transistor 3106 (controlled by signal HVO), and de-select transistor 3107 (controlled by signal HVO_B), configured as shown.

[0105] High voltage level shifter 3108 received enable signal EN and outputs high voltage signal HV and its complement HVO_B.

[0106] Figure 32 depicts VMM high voltage decode circuits, comprising erase gate decoder circuit 3201, control gate decoder circuit 3204, source line decoder circuit 3207, and high voltage level shifter 3211, which are appropriate for use with memory cells of the type shown in Figure 3.

[0107] Erase gate decoder circuit 3201 and control gate decoder circuit 3204 use the same design as word line decoder circuit 3101 in Figure 31.

[0108] Source line decoder circuit 3207 uses the same design as source line decoder circuit 3104 in Figure 31.

[0109] High voltage level shifter 3211 uses the same design as high voltage level shifter 3108 in Figure 31.

[0110] Figure 33 depicts adaptable neuron circuit 3300 that converts an output neuron current into a voltage. Adaptable neuron circuit 3300 uses only one PMOS transistor 3301 and essentially is configured to mirror itself (i.e., a sample and hold mirror) using switches 3302, 3303, and 3304. Initially, switch 3302 and switch 3303 are closed and switch 3304 is open, at which time PMOS transistor 3301 is coupled to I NEURON, which is a current source 3305 that represents the current from a VMM. Then, switch 3302 and 3303 are opened and switch 3304 is closed, which causes PMOS transistor 3301 to send current I_NEURON from its drain to variable resistor 3306. Thus, adaptable neuron 3300 converts a current signal (I_NEURON) into a voltage signal (VO). Basically, transistor 3301 samples the current I_NEURON and holds it by storing a sampled gate-source voltage on its gate. An op amp circuit can be used to buffer the output voltage VO to drive the configurable interconnect.

[0111] Figure 34 depicts current sample and hold S/H circuit 3400 and voltage sample and hold S/H circuit 3450. Current S/H circuit 3400 includes sampling switches 3402 and 3403, S/H capacitor 3405, input transistor 3404 and output transistor 3406. Input transistor 3404 is used to convert input current 3401 into an S/H voltage on the S/H capacitor 3405 and is coupled to the gate of the output transistor 3406. Voltage S/H circuit 3450 includes sampling switch 3452, S/H capacitor 3453, and op amp 3454. Op amp 3454 is used to buffer the S/H voltage on the capacitor 3453. S/H circuits 3400 and 3450 can be used with the output summer circuits and/or activation circuits described herein. In an alternative embodiment, digital sample and hold circuits can be used instead of analog sample and hold circuits 3400 and 3450.

[0112] Figure 35 shows an array architecture that is suitable for memory cells operating in linear region. System 3500 comprises input block 3501, output block 3502, and array 3503 of memory cells. Input block 3501 is coupled to the drains (source lines) of the memory cells in array 3503, and output block 3502 is coupled to the bit lines of the memory cells in array 3503. Alternatively, input block 3501 is coupled to the wordlines of the memory cells in array 3503, and output block 3502 is coupled to the bit lines of the memory cells in array 3503.

[0113] In instances where system 3500 is used to implement an LSTM or GRU, output block 3502 and/or input block 3501 may include multiplier block, addition block, subtraction (output = 1 - input) block as needed for LSTM/GRU architecture, and optionally may include analog sample-and-hold circuits (such as circuits 3400 or 3450 in Figure 34) or digital sample-and-hold circuits (e.g., a register or SRAM) as needed.

[0114] Figure 36 depicts high voltage generation block 3600, which is an example of high voltage generation block 2211 in Figure 22. High voltage generation block 3600 comprises charge pump 3601, charge pump regulator 3603, and high voltage operational amplifier 3602. The voltage of the output of charge pump regulator 3603 can be controlled based on the signals sent to the gates of the mux MOS transistors in charge pump regulator 3603.

[0115] Figure 37 depicts a program bias circuit 3700 that provides a bias to the gates of individual programming circuits 3702-0, ... 3702-N that each provides a programming current to memory cells coupled to the selected bit lines during a programming operations.

[0116] Figure 38 depicts sense amplifier 3800, which can be used for the verify aspect of program and verify operations. Sense amplifier 3800 comprises adjustable current reference source 3801, switch 3802, NMOS transistor 3803, capacitor 3804, switch 3805, current source 3806, and inverter 3807, in the configuration shown. Sense amplifier 3800 is coupled to memory cell 3808 during a verify operation of memory cell 3808.

[0117] It should be noted that, as used herein, the terms "over" and "on" both inclusively include "directly on" (no intermediate materials, elements or space disposed therebetween) and "indirectly on" (intermediate materials, elements or space disposed therebetween). Likewise, the term "adjacent" includes "directly adjacent" (no intermediate materials, elements or space disposed therebetween) and "indirectly adjacent" (intermediate materials, elements or space disposed there between), "mounted to" includes "directly mounted to" (no intermediate materials, elements or space disposed there between) and "indirectly mounted to" (intermediate materials, elements or spaced disposed there between), and "electrically coupled" includes "directly electrically coupled to" (no intermediate materials or elements there between that electrically connect the elements together) and "indirectly electrically coupled to" (intermediate materials or elements there between that electrically connect the elements together). For example, forming an element "over a substrate" can include forming the element directly on the substrate with no intermediate materials/elements therebetween, as well as forming the element indirectly on the substrate with one or more intermediate materials/elements there between.

Claims

1. An analog neuromorphic memory system (2200), comprising:

   a plurality of vector-by-matrix multiplication systems (2221, 2222, 2223, 2320), each vector-by-matrix multiplication system comprising:

      an array of memory cells (2301),
      a low voltage row decoder (2302) for providing a bias voltage to one or more rows of memory cells in the array of memory cells during read and program operations and for providing a decoding signal;
      a high voltage row decoder (2303) for receiving the decoding signal and providing a high voltage bias signal to one or more rows of memory cells in the array of memory cells during program and erase operations; and
      a low voltage column decoder (2304);

   a plurality of output blocks (2207, 2208, 2209), each output block providing an output in response to current received from at least one of the plurality of vector-by-matrix multiplication systems; and
   a shared high voltage generator block (2211, 3600) comprising a charge pump (3601) for outputting a high voltage and a charge pump regulator (3603) for receiving the high voltage from the charge pump and generating

a trimmed high voltage in response to input trim bits, the shared high voltage generator block configured to concurrently provide voltages comprising one or more of the high voltage and the trimmed high voltage to the high voltage row decoder in each of the plurality of vector-by-matrix multiplication systems for one or more of erase operations and programming operations.

2. The analog neuromorphic memory system of claim 1, further comprising a high voltage compensation block for each of the plurality of vector-by-matrix multiplication systems.

3. The analog neuromorphic memory system of claim 1, further comprising a high voltage compensation block for all of the plurality of vector-by-matrix multiplication systems.

4. The analog neuromorphic memory system of claim 1, where the analog neuromorphic memory system is configured to concurrently perform programming operations to two or more vector-by-matrix multiplication systems.

5. The analog neuromorphic memory system of claim 4, wherein the concurrent programming operations are performed in response to a single command.

6. The analog neuromorphic memory system of claim 4, where the analog neuromorphic memory system is configured to concurrently perform verify operations to the two or more vector-by-matrix multiplication systems after the concurrent programming operations.

7. The analog neuromorphic memory system of claim 1, wherein the high voltage generator block is able to provide sufficient voltage and current to perform program and erase operations concurrently on all arrays of memory cells in all of the vector-by-matrix multiplication systems.

8. The analog neuromorphic memory system of claim 1, wherein the memory cells are split-gate flash memory cells.

9. The analog neuromorphic memory system of claim 1, wherein each vector-by-matrix multiplication system is a cell in a long short term memory system.

10. The analog neuromorphic memory system of claim 1, wherein each vector-by-matrix multiplication system is a cell in a gated recurrent unit memory system.

11. The analog neuromorphic memory system of claim 1, wherein each of the plurality of output blocks comprises a summer and activation block.

12. The analog neuromorphic memory system of claim 11, wherein each summer and activation block is configured to perform a summing and activation function for at least one of the plurality of vector-by-matrix multiplication systems.

Patentansprüche

1. Analoges neuromorphes Speichersystem (2200), umfassend:
eine Vielzahl von Vektor-Matrix-Multiplikationssystemen (2221, 2222, 2223, 2320), wobei jedes Vektor-Matrix-Multiplikationssystem umfasst:

ein Array von Speicherzellen (2301),
einen Niederspannungszeilendecodierer (2302) zum Bereitstellen einer Vorspannung an eine oder mehrere Zeilen von Speicherzellen in dem Array von Speicherzellen während Lese- und Programmieroperationen und zum Bereitstellen eines Decodiersignals;
einen Hochspannungszeilendecodierer (2303) zum Empfangen des Decodiersignals und Bereitstellen eines Hochspannungsvorspannungssignals an eine oder mehrere Zeilen von Speicherzellen in dem Array von Speicherzellen während Programmier- und Löschoperationen; und
einen Niederspannungsspaltendecodierer (2304);
eine Vielzahl von Ausgangsblöcken (2207, 2208, 2209), wobei jeder Ausgangsblock einen Ausgang als Reaktion auf Strom bereitstellt, der von mindestens einem der Vielzahl von Vektor-Matrix-Multiplikationssystemen empfangen wird; und
einen gemeinsam genutzten Hochspannungsgeneratorblock (2211, 3600), umfassend eine Ladepumpe (3601)

zum Ausgeben einer Hochspannung und einen Ladepumpenregler (3603) zum Empfangen der Hochspannung von der Ladepumpe und Erzeugen einer getrimmten Hochspannung als Reaktion auf Eingangs-Trimmbits, wobei der gemeinsam genutzte Hochspannungsgeneratorblock dazu konfiguriert ist, gleichzeitig Spannungen bereitzustellen, die eine oder mehrere der Hochspannung und der getrimmten Hochspannung an den Hochspannungszeilendecodierer in jedem der Vielzahl von Vektor-Matrix-Multiplikationssystemen für einen oder mehrere von Löschoperationen und Programmieroperationen umfassen.

2. Analog neuromorphes Speichersystem nach Anspruch 1, ferner umfassend einen Hochspannungskompensationsblock für jeden der Vielzahl von Vektor-Matrix-Multiplikationssystemen.

3. Analog neuromorphes Speichersystem nach Anspruch 1, ferner umfassend einen Hochspannungskompensationsblock für alle der Vielzahl von Vektor-Matrix-Multiplikationssystemen.

4. Analog neuromorphes Speichersystem nach Anspruch 1, wobei das analoge neuromorphe Speichersystem so konfiguriert ist, dass es gleichzeitig Programmieroperationen an zwei oder mehr Vektor-Matrix-Multiplikationssystemen durchführt.

5. Analog neuromorphes Speichersystem nach Anspruch 4, wobei die gleichzeitigen Programmieroperationen als Reaktion auf einen einzigen Befehl durchgeführt werden.

6. Analog neuromorphes Speichersystem nach Anspruch 4, wobei das analoge neuromorphe Speichersystem dazu konfiguriert ist, gleichzeitig Verifizierungsoperationen an den zwei oder mehr Vektor-Matrix-Multiplikationssystemen nach den gleichzeitigen Programmieroperationen durchzuführen.

7. Analog neuromorphes Speichersystem nach Anspruch 1, wobei der Hochspannungsgeneratorblock in der Lage ist, eine ausreichende Spannung und einen ausreichenden Strom bereitzustellen, um Programmier- und Löschoperationen gleichzeitig an allen Arrays von Speicherzellen in allen Vektor-Matrix-Multiplikationssystemen durchzuführen.

8. Analog neuromorphes Speichersystem nach Anspruch 1, wobei die Speicherzellen Split-Gate-Flash-Speicherzellen sind.

9. Analog neuromorphes Speichersystem nach Anspruch 1, wobei jedes Vektor-Matrix-Multiplikationssystem eine Zelle in einem langen Kurzzeitspeichersystem ist.

10. Analog neuromorphes Speichersystem nach Anspruch 1, wobei jedes Vektor-Matrix-Multiplikationssystem eine Zelle in einem gategesteuerten rekurrenten Einheitsspeichersystem ist.

11. Analog neuromorphes Speichersystem nach Anspruch 1, wobei jeder der Vielzahl von Ausgangsblöcken einen Summier- und Aktivierungsblock umfasst.

12. Analog neuromorphes Speichersystem nach Anspruch 11, wobei jeder Summier- und Aktivierungsblock dazu konfiguriert ist, eine Summier- und Aktivierungsfunktion für mindestens eines der Vielzahl von Vektor-Matrix-Multiplikationssystemen durchzuführen.

Revendications

1. Système de mémoire neuromorphique analogique (2200), comprenant :
une pluralité de systèmes de multiplication de vecteur par matrice (2221, 2222, 2223, 2320), chaque système de multiplication de vecteur par matrice comprenant :

un réseau de cellules de mémoire (2301),
un décodeur de rangée basse tension (2302) pour fournir une tension de polarisation à une ou plusieurs rangées de cellules de mémoire dans le réseau de cellules de mémoire pendant des opérations de lecture et programmation et pour fournir un signal de décodage ;
un décodeur de rangée haute tension (2303) pour recevoir le signal de décodage et fournir un signal de polarisation haute tension à une ou plusieurs rangées de cellules de mémoire dans le réseau de cellules de mémoire pendant des opérations de programmation et d’effacement ; et

un décodeur de colonne basse tension (2304) ;
une pluralité de blocs de sortie (2207, 2208, 2209), chaque bloc de sortie fournissant une sortie en réponse à un courant reçu d'au moins un parmi la pluralité de systèmes de multiplication de vecteur par matrice ; et
un bloc générateur haute tension partagé (2211, 3600) comprenant une pompe de charge (3601) pour délivrer en sortie une haute tension et un régulateur de pompe de charge (3603) pour recevoir la haute tension provenant de la pompe de charge et générer une haute tension ajustée en réponse à des bits d'ajustement d'entrée, le bloc générateur haute tension partagé configuré pour fournir concomitamment des tensions comprenant une ou plusieurs parmi la haute tension et la haute tension ajustée au décodeur de rangée haute tension dans chacun parmi la pluralité de systèmes de multiplication de vecteur par matrice pour une ou plusieurs parmi des opérations d'effacement et des opérations de programmation.

2. Système de mémoire neuromorphique analogique selon la revendication 1, comprenant en outre un bloc de compensation haute tension pour chacun parmi la pluralité de systèmes de multiplication de vecteur par matrice.

3. Système de mémoire neuromorphique analogique selon la revendication 1, comprenant en outre un bloc de compensation haute tension pour tous parmi la pluralité de systèmes de multiplication de vecteur par matrice.

4. Système de mémoire neuromorphique analogique selon la revendication 1, où le système de mémoire neuromorphique analogique est configuré pour mettre en oeuvre concomitamment des opérations de programmation sur deux systèmes de multiplication de vecteur par matrice ou plus.

5. Système de mémoire neuromorphique analogique selon la revendication 4, dans lequel les opérations de programmation concomitantes sont mises en oeuvre en réponse à une unique instruction.

6. Système de mémoire neuromorphique analogique selon la revendication 4, où le système de mémoire neuromorphique analogique est configuré pour mettre en oeuvre concomitamment des opérations de vérification sur les deux systèmes de multiplication de vecteur par matrice ou plus après les opérations de programmation concomitantes.

7. Système de mémoire neuromorphique analogique selon la revendication 1, dans lequel le bloc générateur haute tension est capable de fournir une tension et un courant suffisants pour mettre en oeuvre des opérations de programmation et d'effacement concomitamment sur tous les réseaux de cellules de mémoire dans la totalité des systèmes de multiplication de vecteur par matrice.

8. Système de mémoire neuromorphique analogique selon la revendication 1, dans lequel les cellules de mémoire sont des cellules de mémoire flash à grille divisée.

9. Système de mémoire neuromorphique analogique selon la revendication 1, dans lequel chaque système de multiplication de vecteur par matrice est une cellule dans un système de mémoire longue à court terme.

10. Système de mémoire neuromorphique analogique selon la revendication 1, dans lequel chaque système de multiplication de vecteur par matrice est une cellule dans un système de mémoire à unité récurrente à grille.

11. Système de mémoire neuromorphique analogique selon la revendication 1, dans lequel chacun parmi la pluralité de blocs de sortie comprend un bloc de sommation et d'activation.

12. Système de mémoire neuromorphique analogique selon la revendication 11, dans lequel chaque bloc de sommation et d'activation est configuré pour mettre en oeuvre une fonction de sommation et d'activation pour au moins un parmi la pluralité de systèmes de multiplication de vecteur par matrice.

FIGURE 1 (PRIOR ART)

100

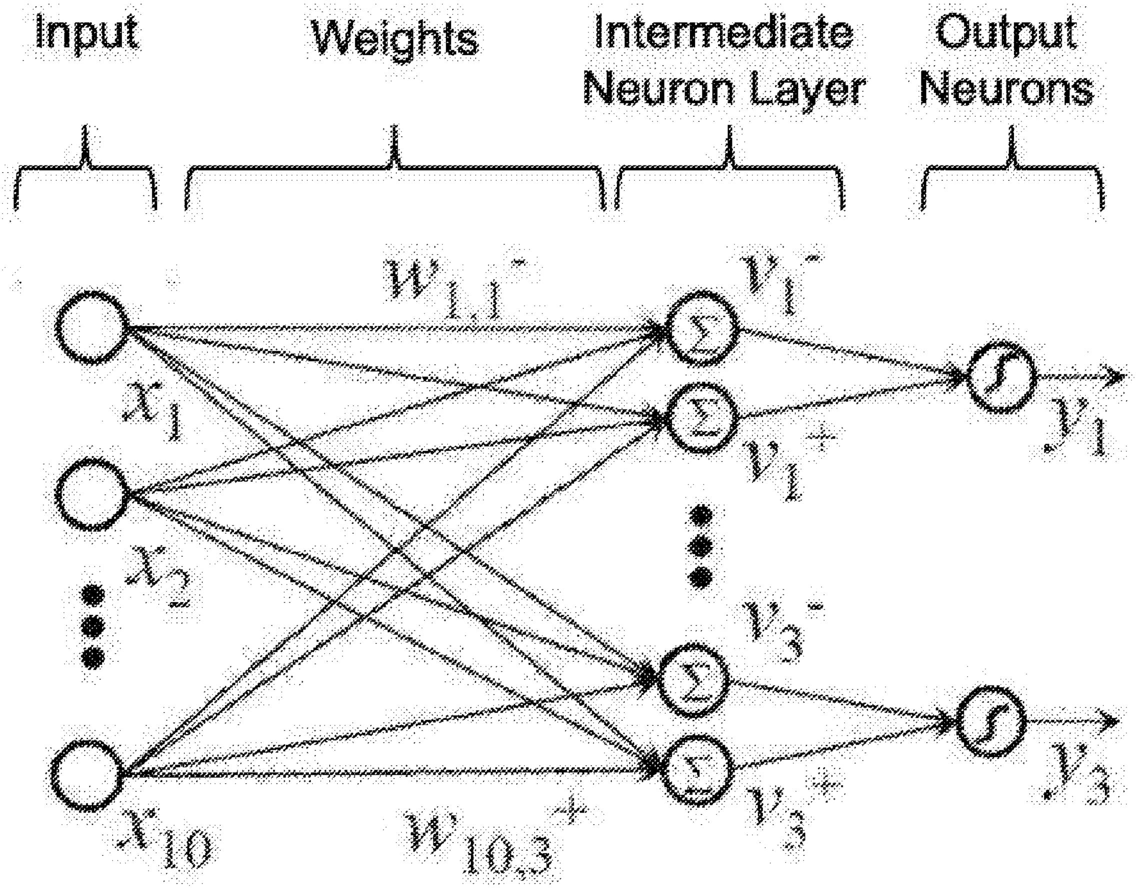


FIGURE 2 (PRIOR ART)

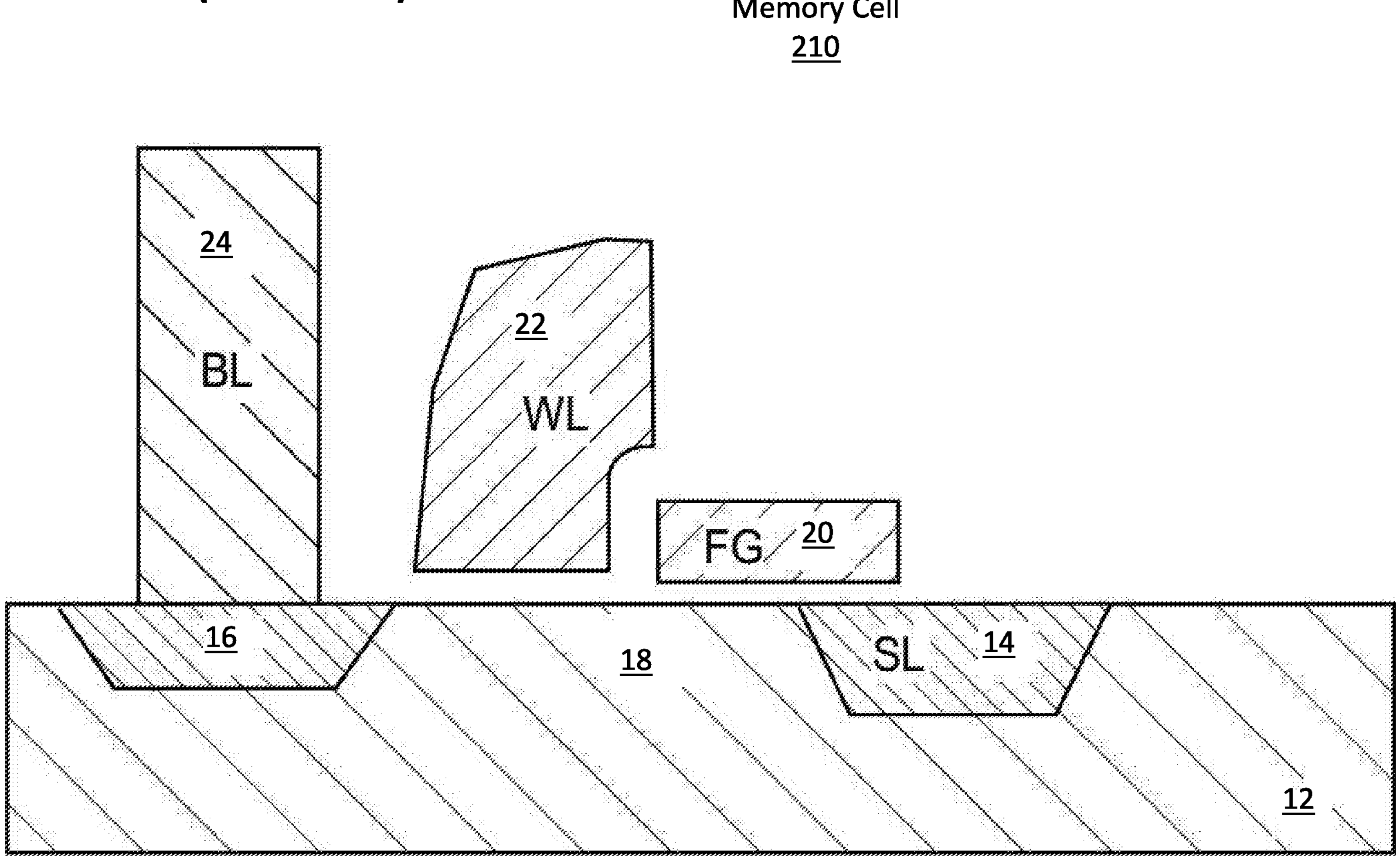


FIGURE 3 (PRIOR ART)

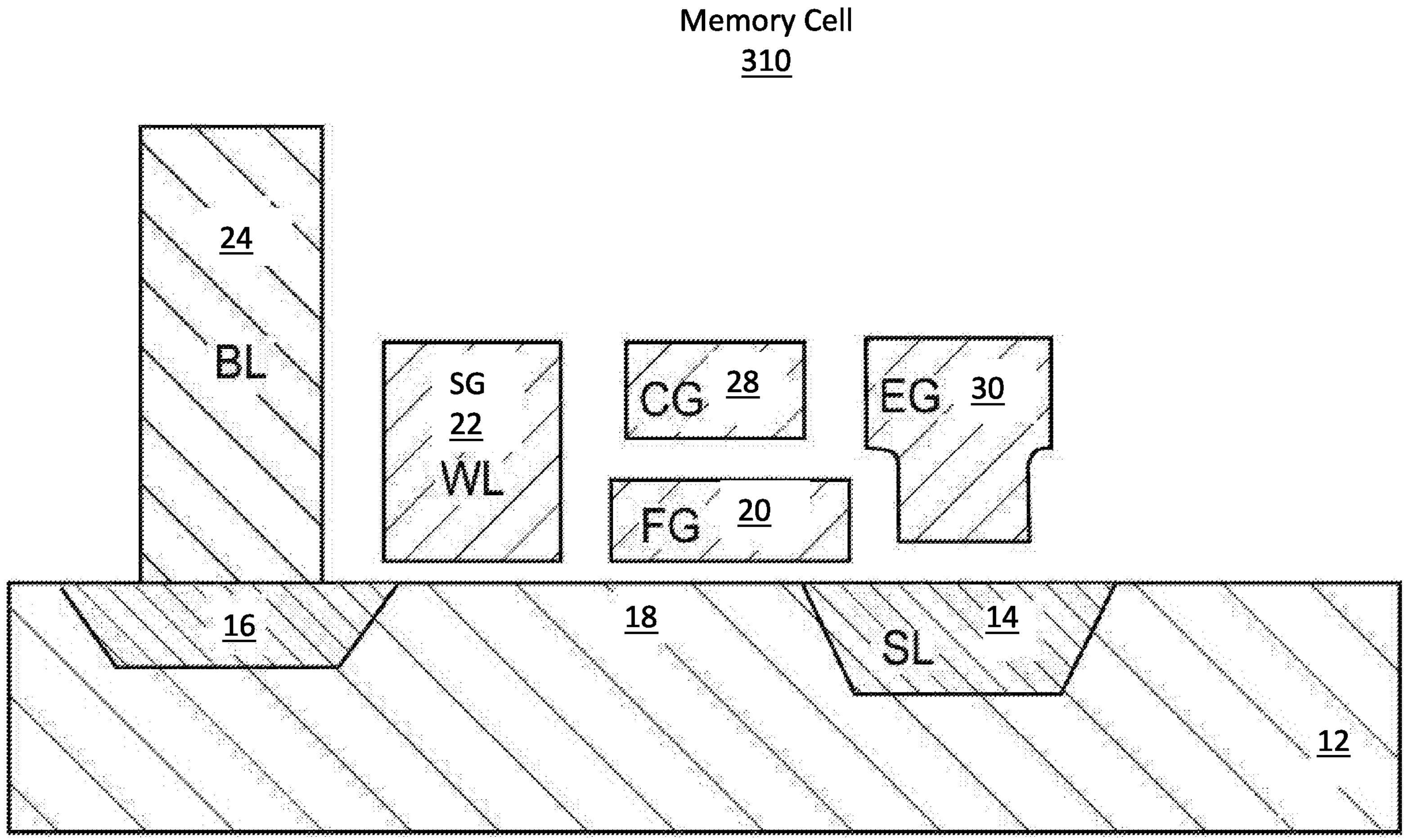


FIGURE 4 (PRIOR ART)

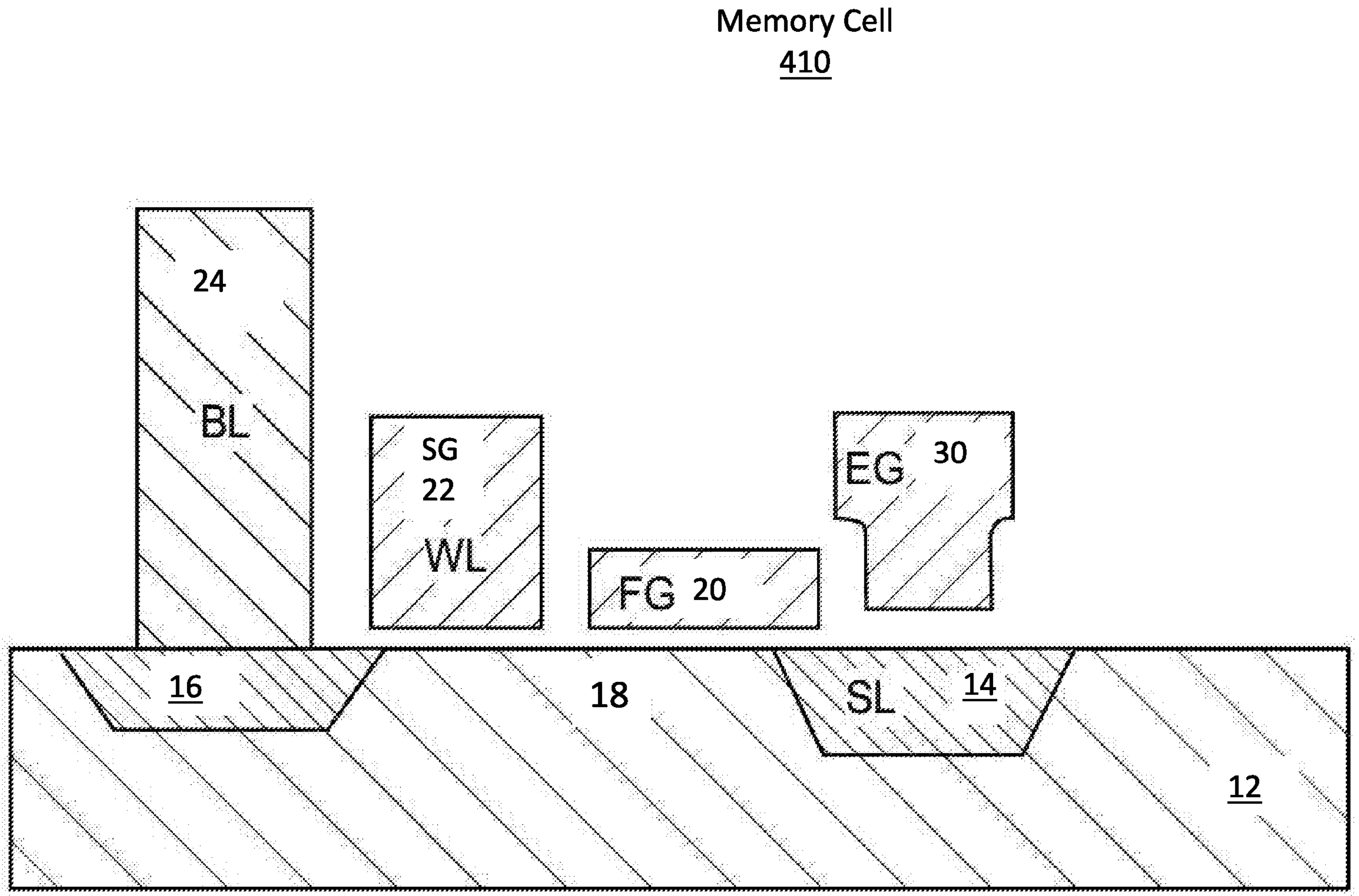


FIGURE 5 (PRIOR ART)

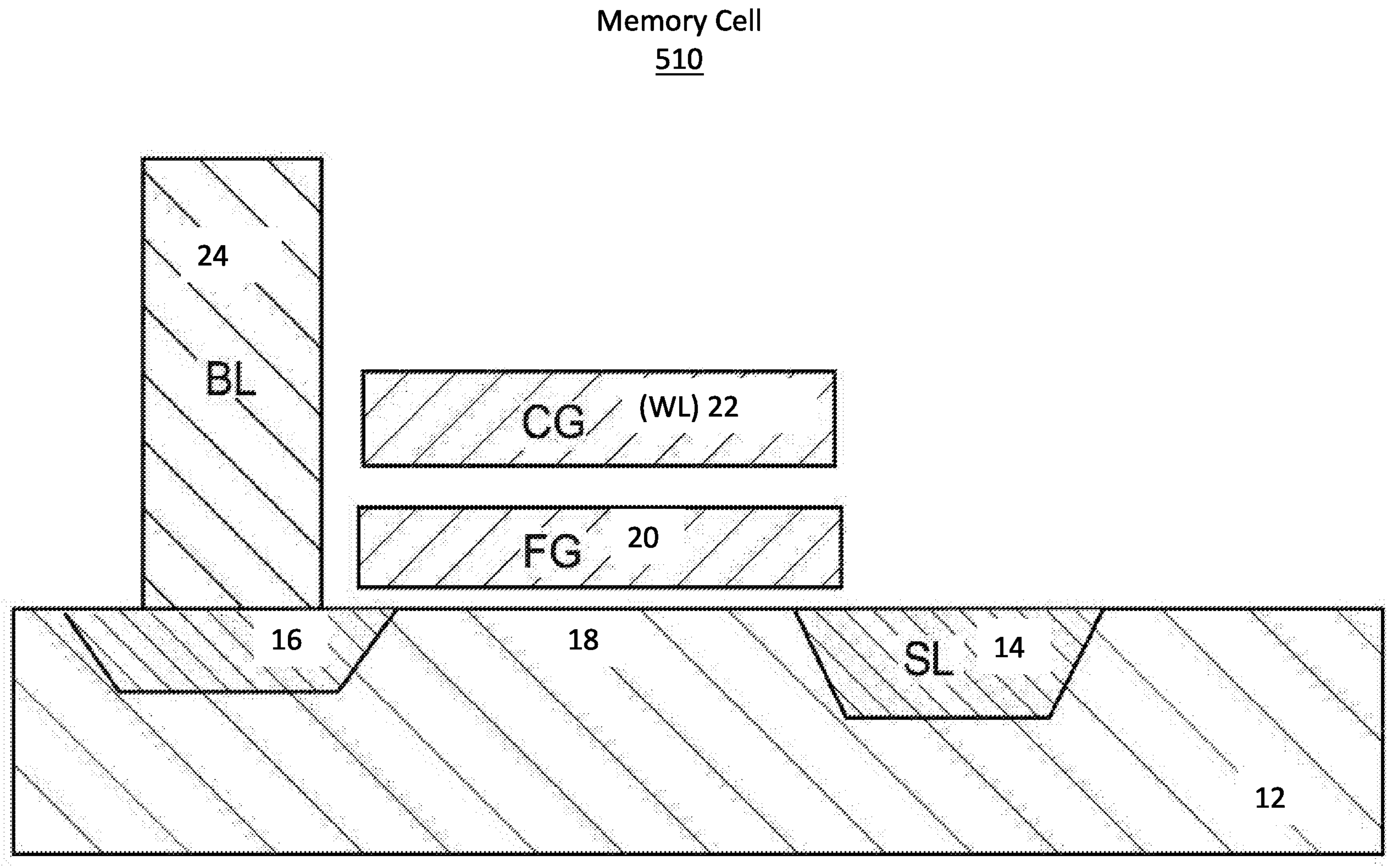


FIGURE 6

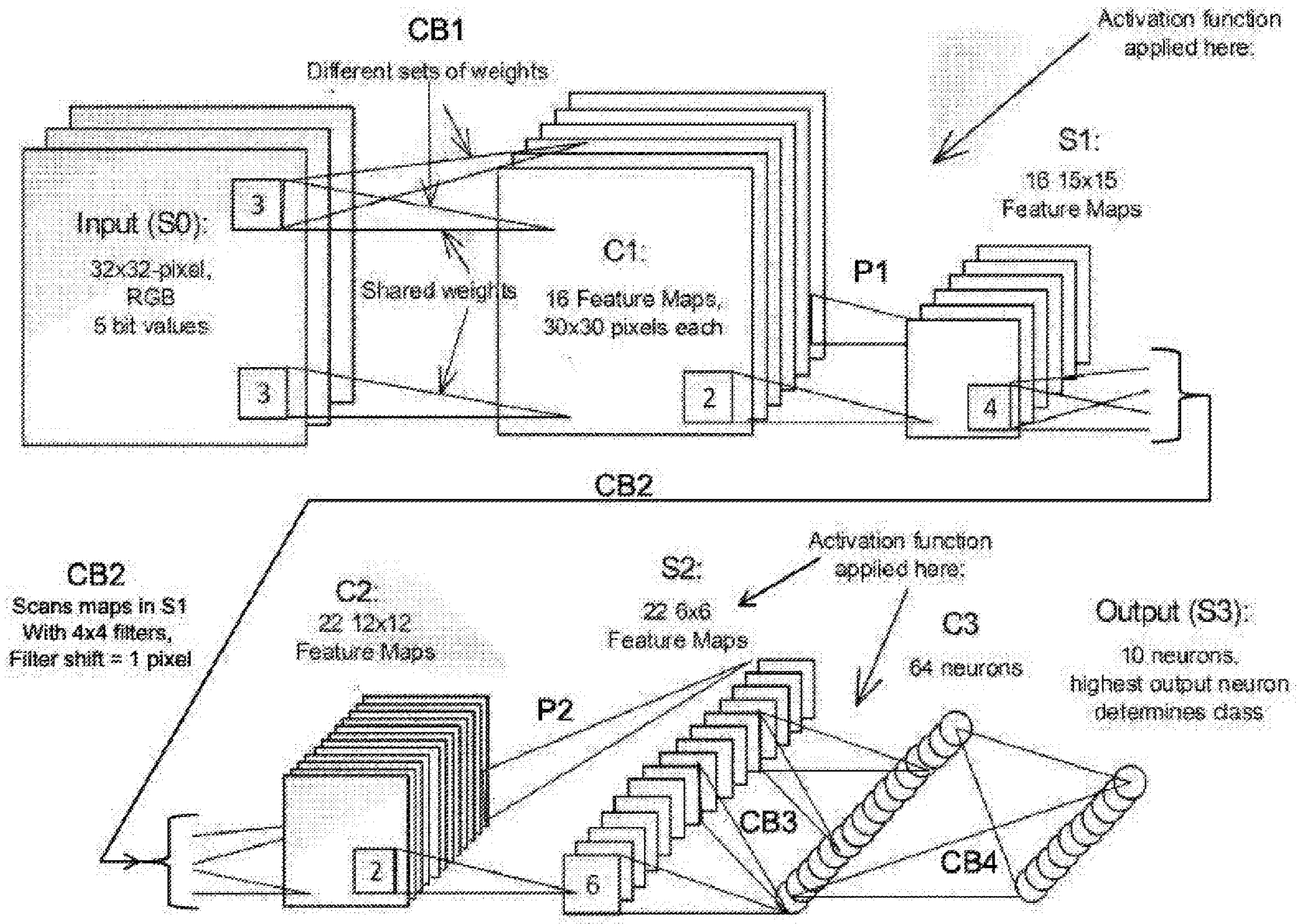
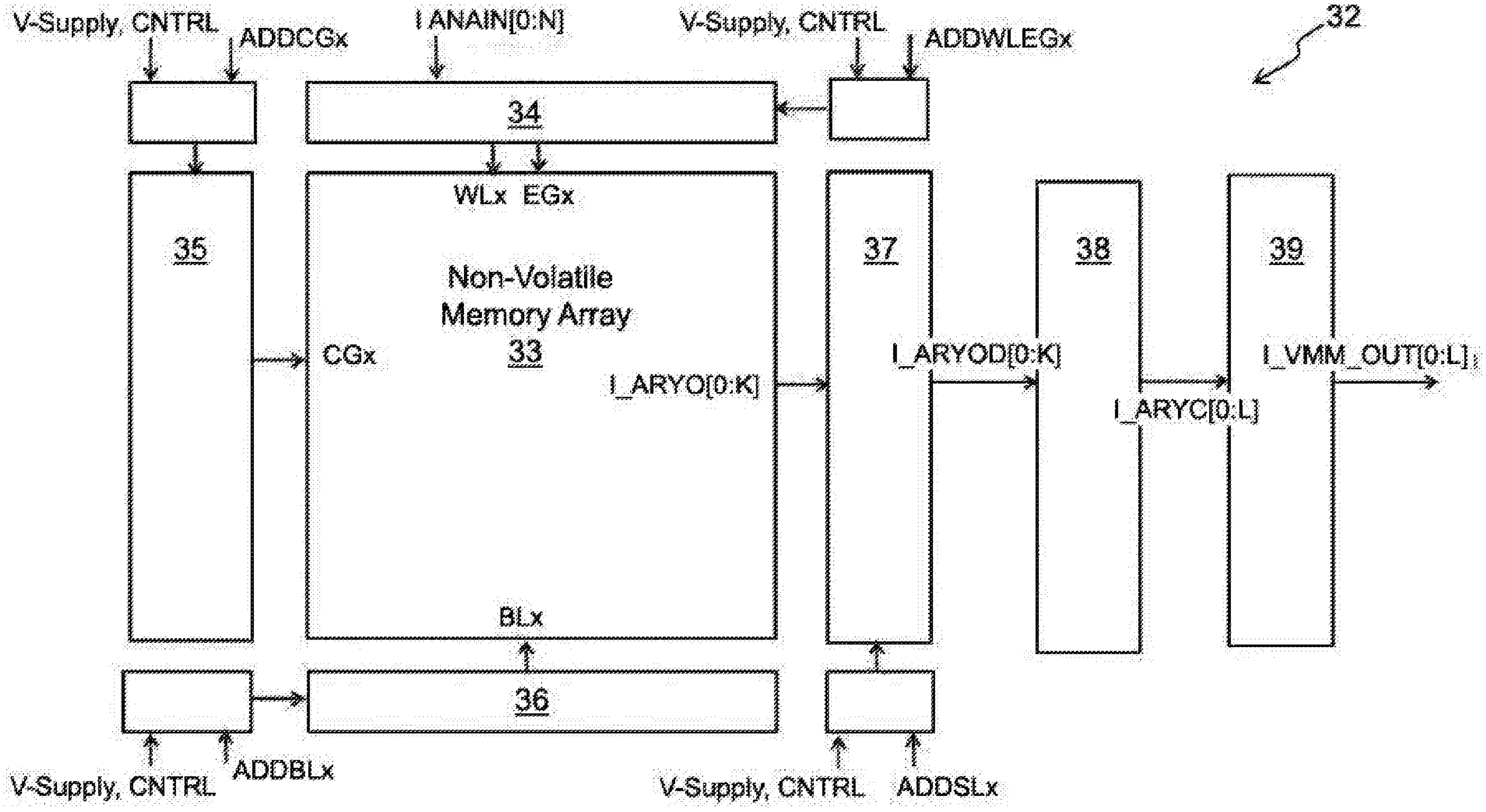
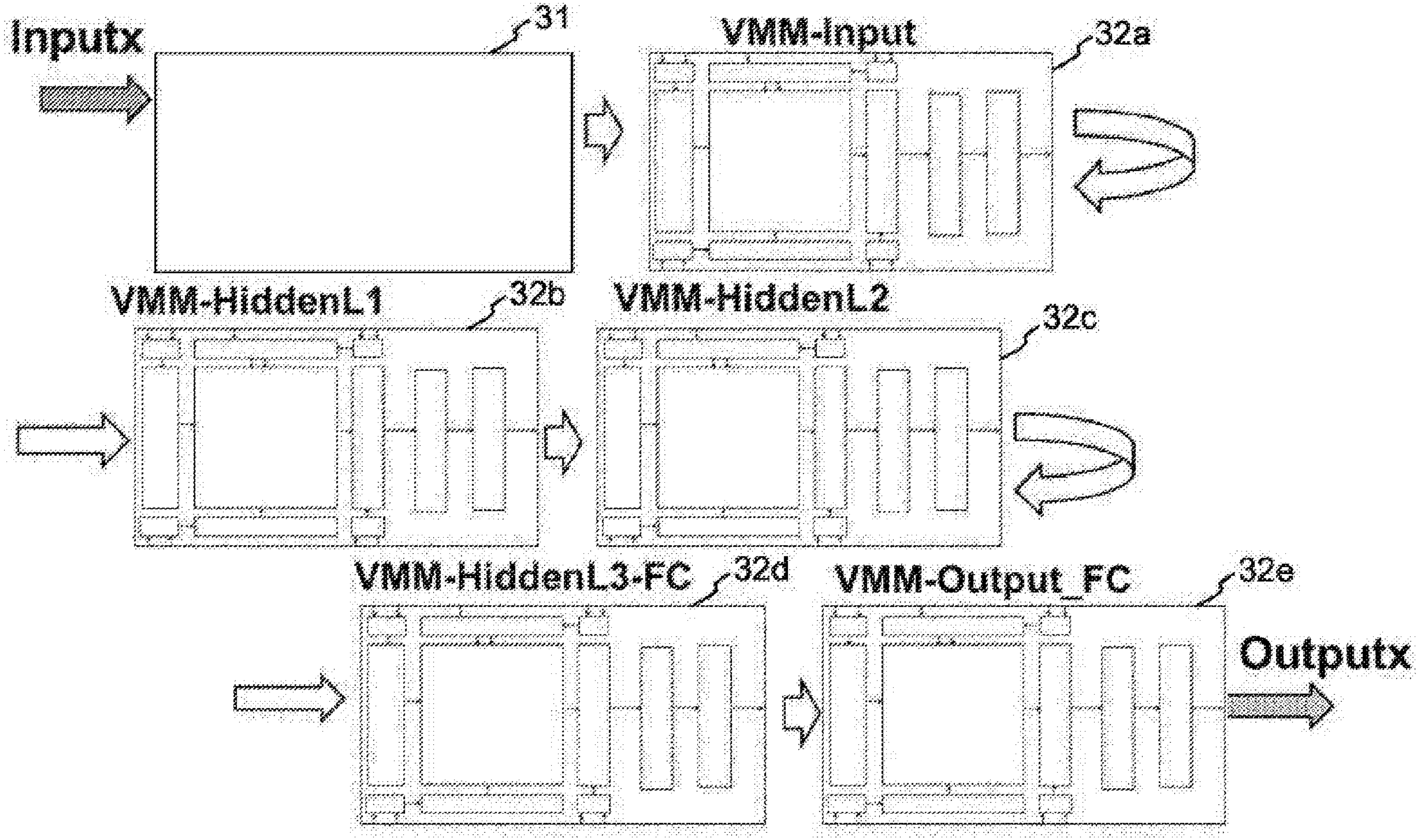

CB1
Different sets of weights
Activation function
applied here:
Input (S0):
32x32-pixel,
RGB
5 bit values
3
3
Shared weights
C1:
16 Feature Maps,
30x30 pixels each
2
P1
S1:
16 15x15
Feature Maps
4
CB2
CB2
Scans maps in S1
With 4x4 filters,
Filter shift = 1 pixel
C2:
22 12x12
Feature Maps
2
P2
S2:
22 6x6
Feature Maps
6
Activation function
applied here:
C3
64 neurons
CB3
CB4
Output (S3):
10 neurons,
highest output neuron
determines class

FIGURE 7



V-Supply, CNTRL
ADDCGx
I ANAIN[0:N]
V-Supply, CNTRL
ADDWLEGx
32
34
WLx EGx
35
Non-Volatile
Memory Array
33
CGx
I_ARYO[0:K]
37
I_ARYOD[0:K]
38
I_ARYC[0:L]
39
I_VMM_OUT[0:L]
BLx
36
V-Supply, CNTRL
ADDBLx
V-Supply, CNTRL
ADDSLx

FIGURE 8


Inputx
31
VMM-Input
32a
VMM-HiddenL1
32b
VMM-HiddenL2
32c
VMM-HiddenL3-FC
32d
VMM-Output_FC
32e
Outputx

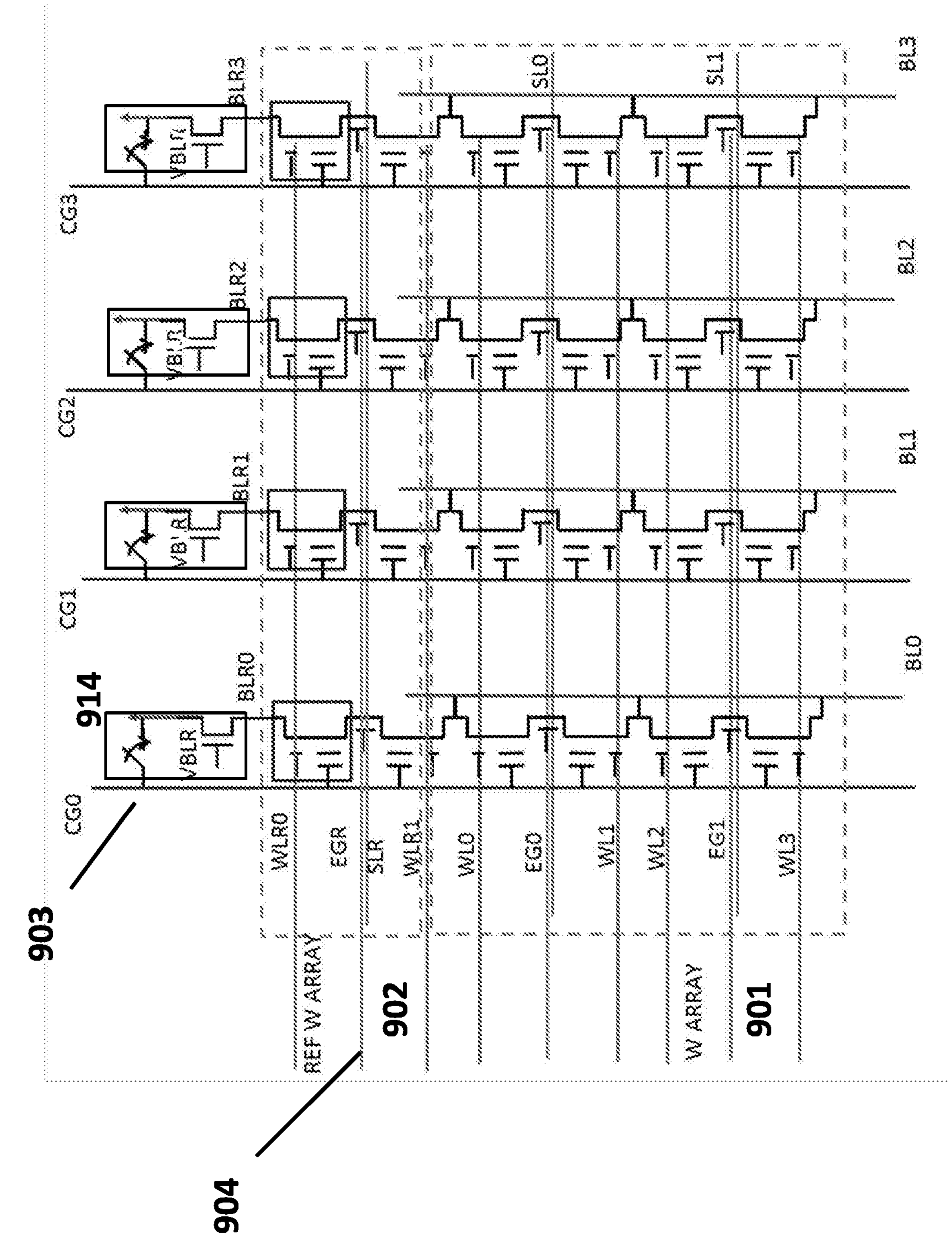

900
903
914
CG0
CG1
CG2
CG3
VBLR
VB'R
VB'R
VBLR
BLR0
BLR1
BLR2
BLR3
WLR0
EGR
SLR
WLR1
WL0
EG0
WL1
WL2
EG1
WL3
SL0
SL1
BL0
BL1
BL2
BL3
REF W ARRAY
902
W ARRAY
901
904

FIGURE 9

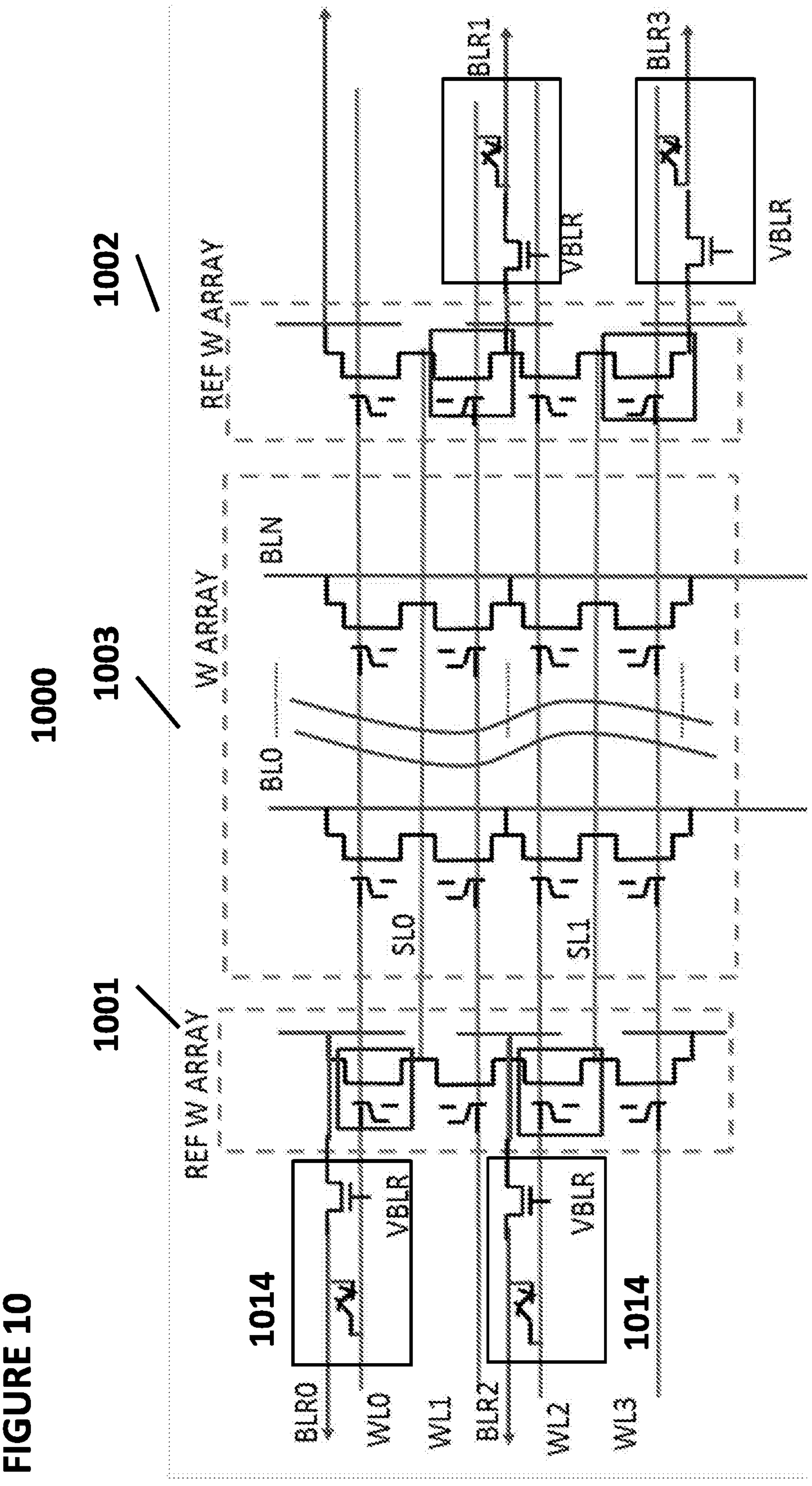
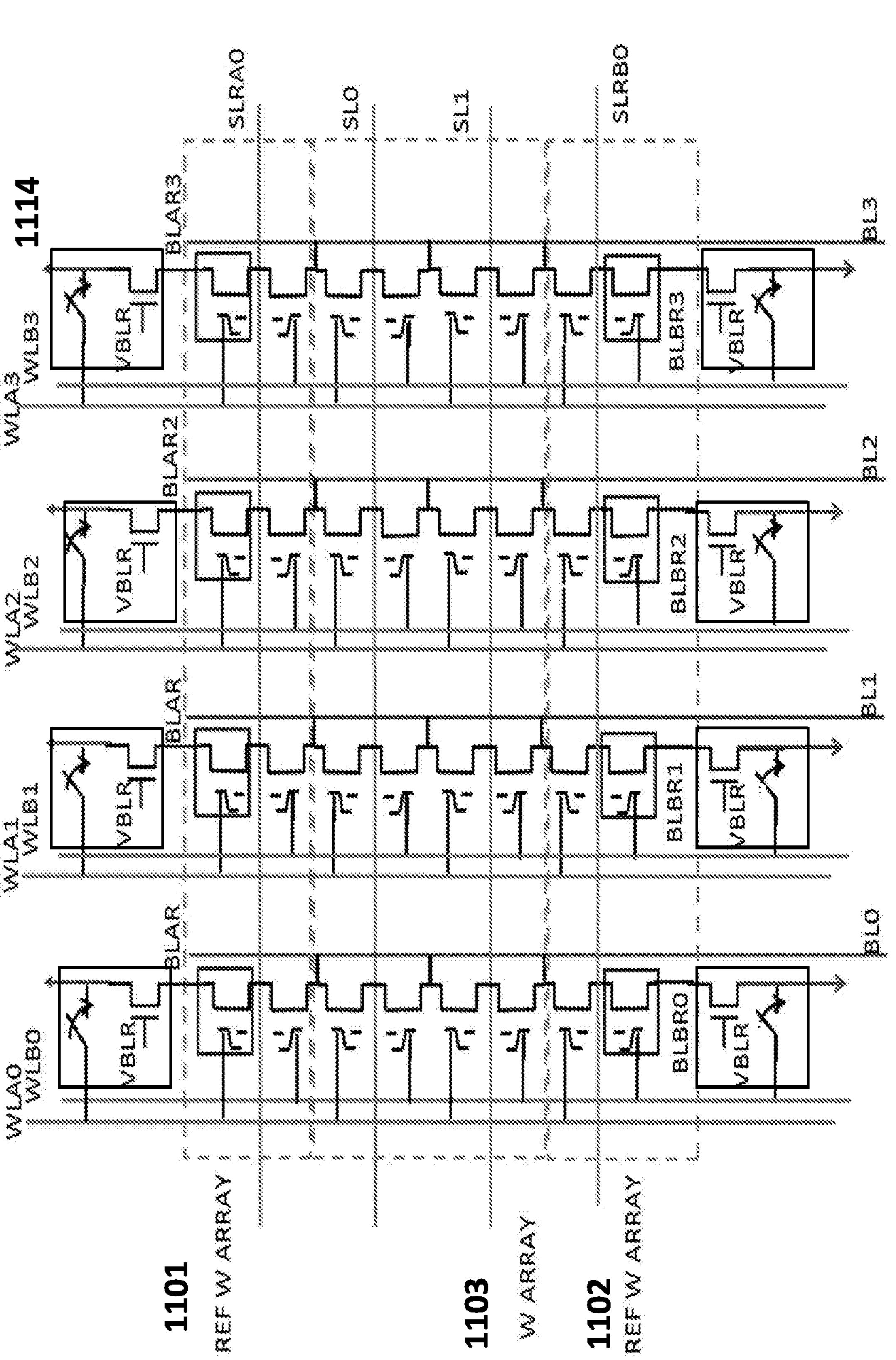

1000
1001
1002
1003
1014
1014
REF W ARRAY
W ARRAY
REF W ARRAY
BLR0
WL0
WL1
BLR2
WL2
WL3
VBLR
VBLR
SL0
SL1
BL0
BLN
BLR1
BLR3
VBLR
VBLR

FIGURE 10

FIGURE 11


1100
1114
WLA0
WLB0
WLA1
WLB1
WLA2
WLB2
WLA3
WLB3
VBLR
BLAR
BLAR
BLAR2
BLAR3
BLBR0
BLBR1
BLBR2
BLBR3
BL0
BL1
BL2
BL3
SLRA0
SL0
SL1
SLRB0
1101
REF W ARRAY
1103
W ARRAY
1102
REF W ARRAY

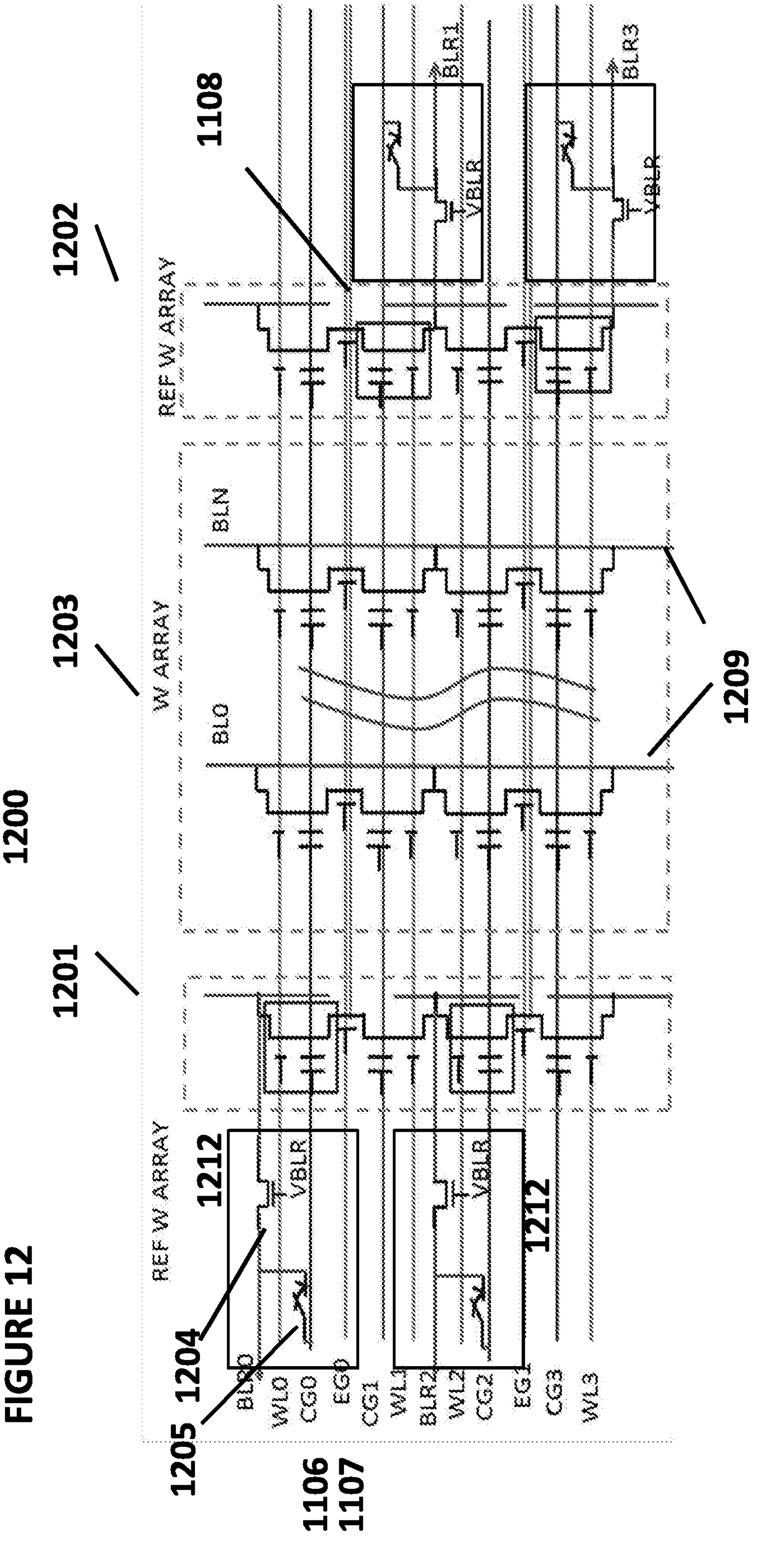

1200
1201
1202
1203
1108
1209
1212
1212
1204
1205
1106
1107
REF W ARRAY
W ARRAY
REF W ARRAY
BL0
BLN
BLR1
BLR3
VBLR
VBLR
VBLR
VBLR
BL0
WL0
CG0
EG0
CG1
WL1
BLR2
WL2
CG2
EG1
CG3
WL3

FIGURE 12

FIGURE 13

1300

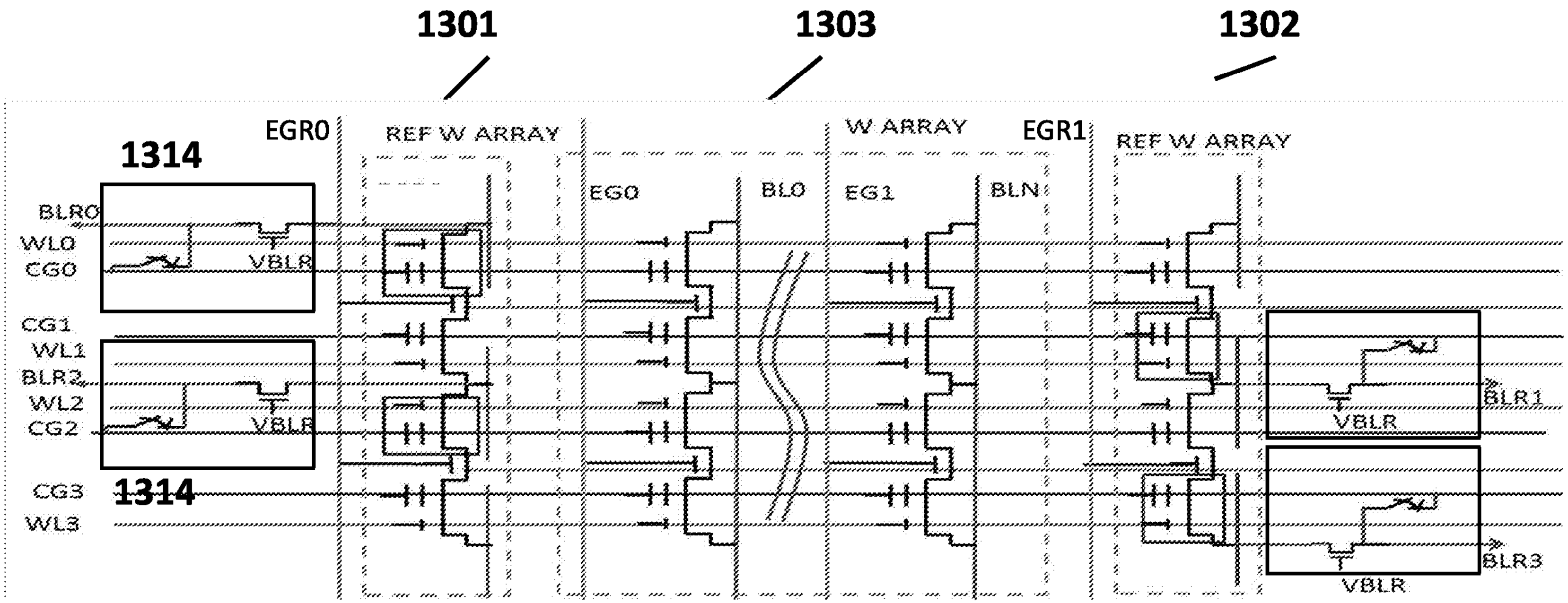


FIGURE 14 (PRIOR ART)

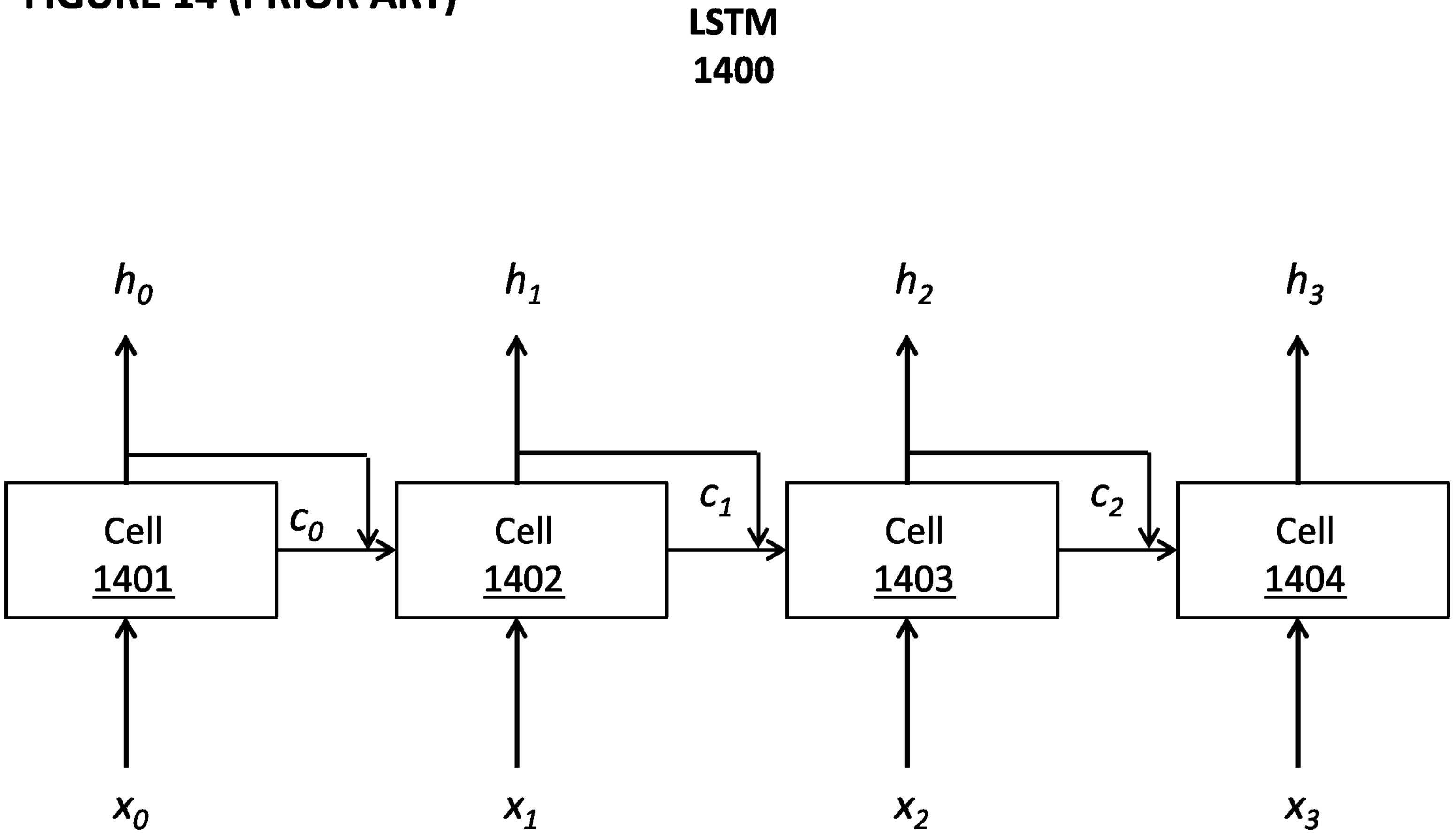


FIGURE 15 (PRIOR ART)

LSTM Cell
1500

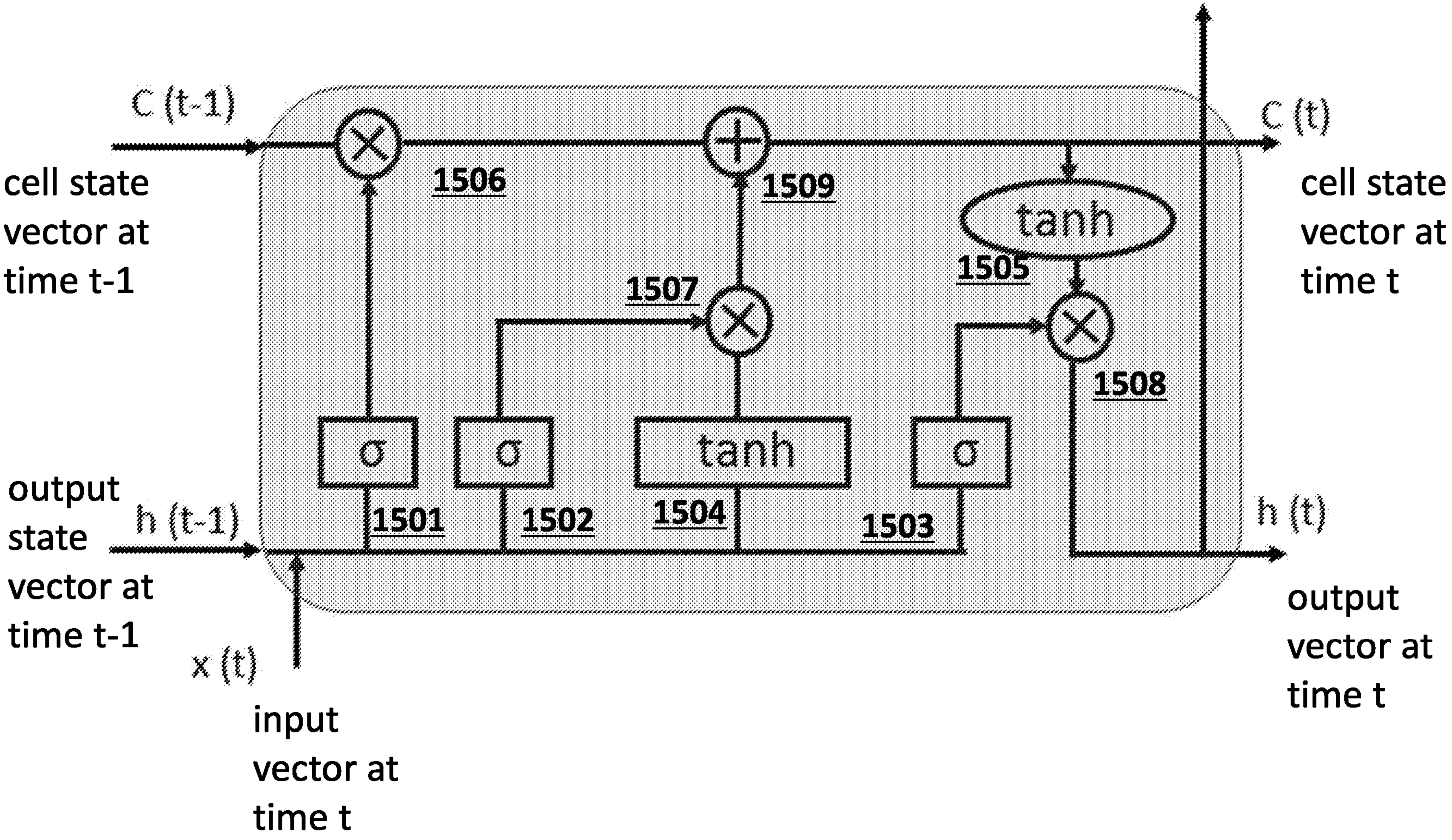


FIGURE 16

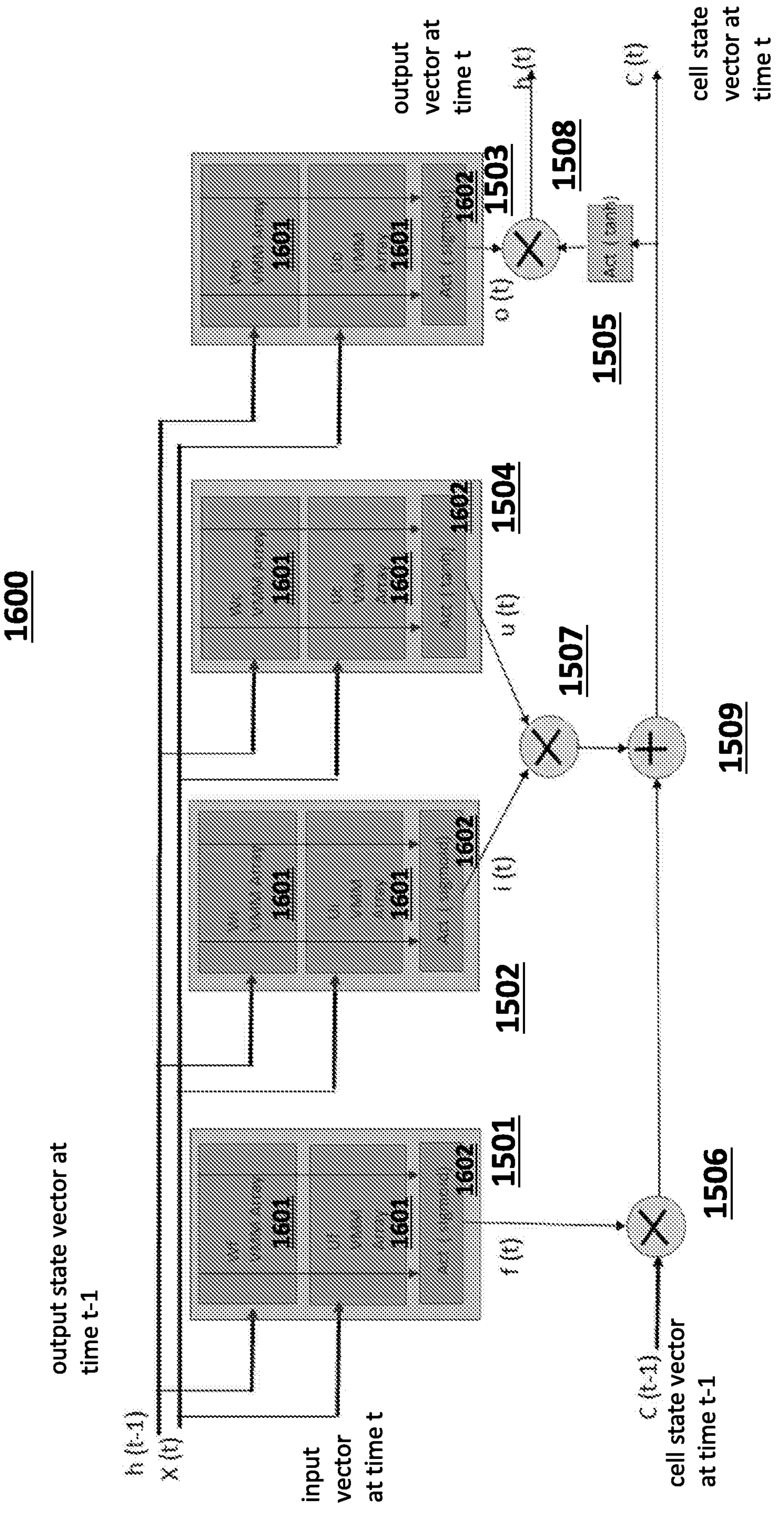

LSTM Cell
1600
output state vector at time t-1
h (t-1)
X (t)
input vector at time t
C (t-1)
cell state vector at time t-1
1501
1502
1504
1503
1601
1602
f (t)
i (t)
u (t)
o (t)
1505
1506
1507
1508
1509
Act (tanh)
h (t)
output vector at time t
C (t)
cell state vector at time t

FIGURE 17

LSTM Cell
1700

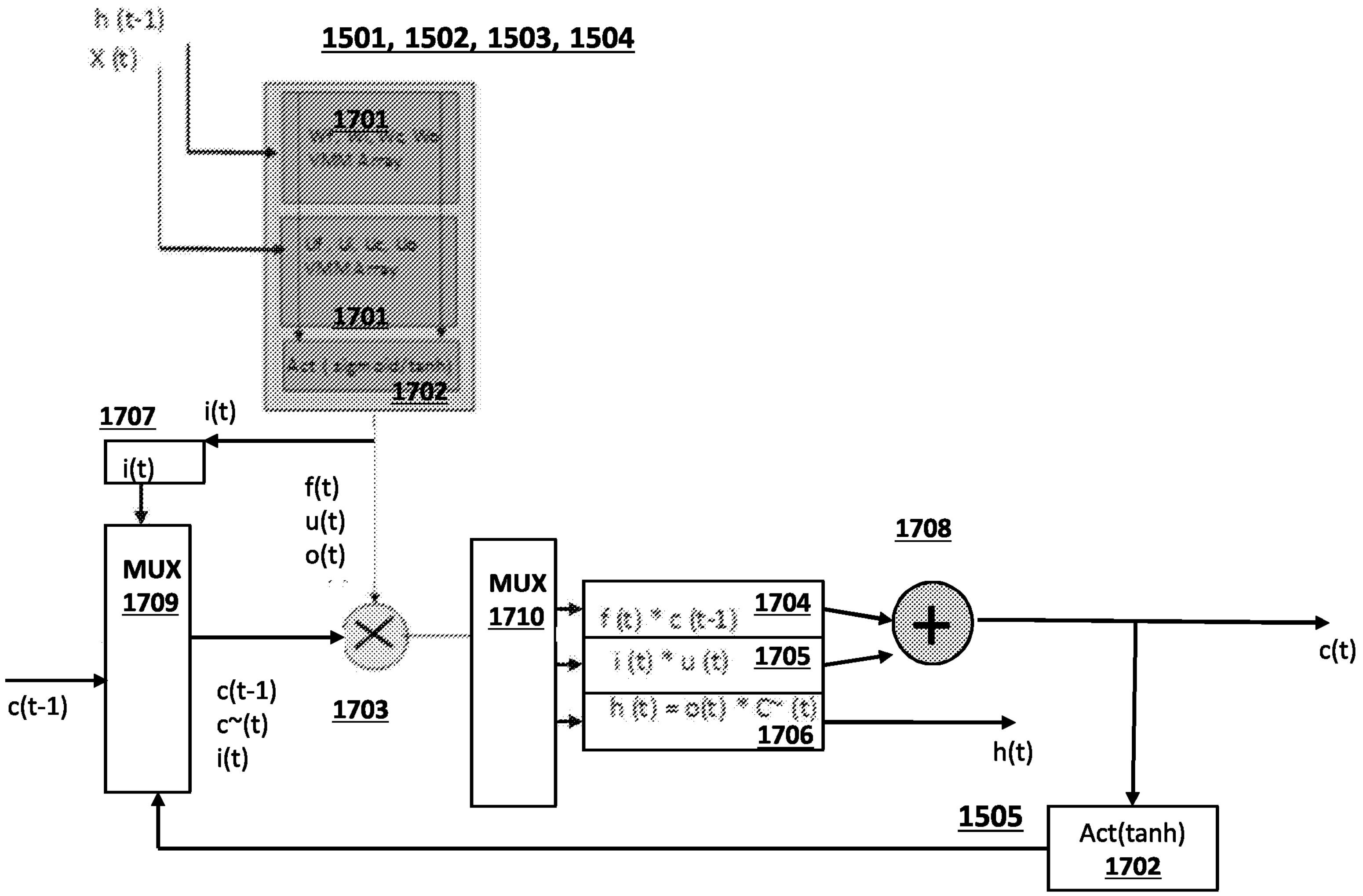


FIGURE 18 (PRIOR ART)

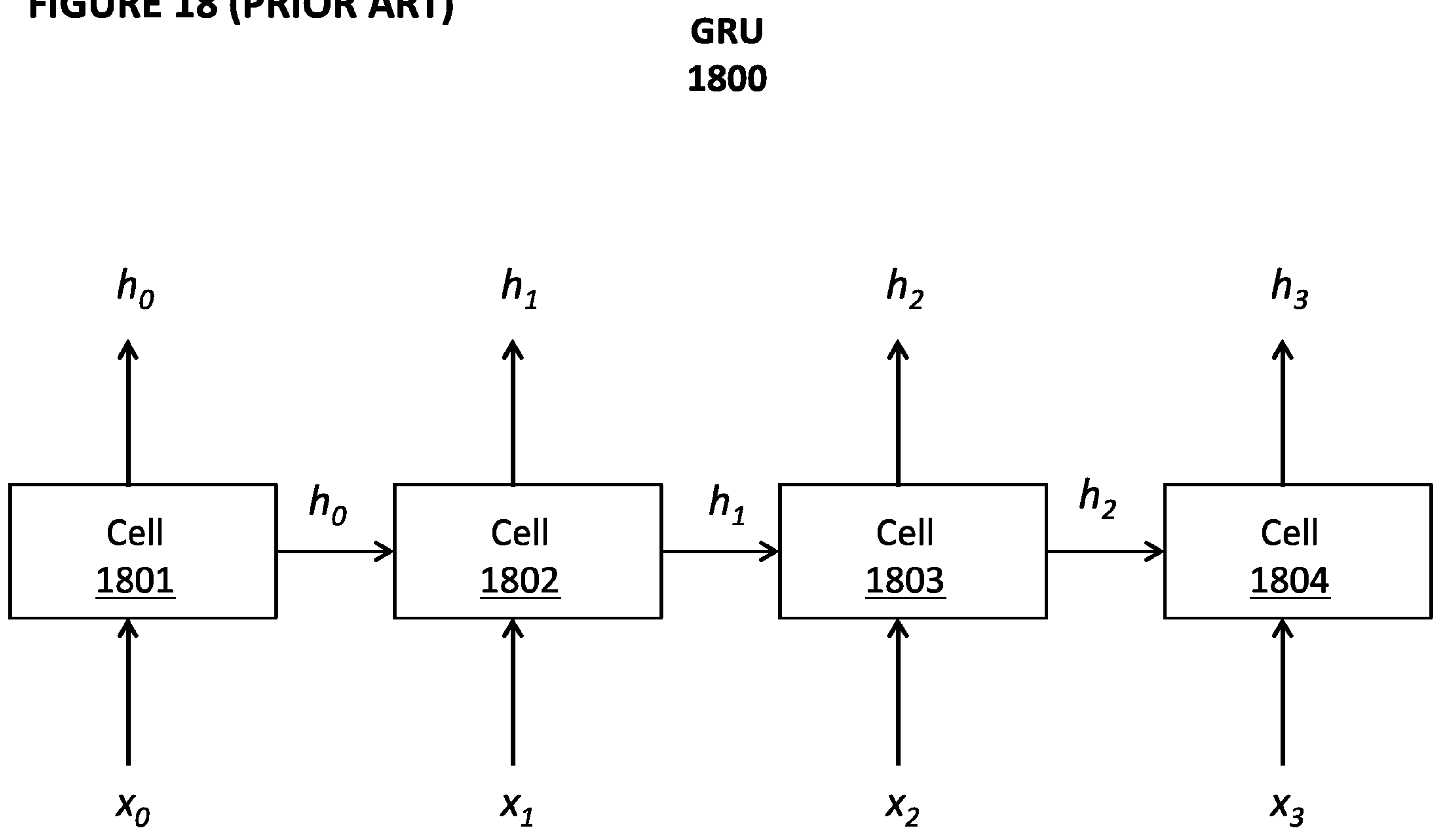


FIGURE 19 (PRIOR ART)

GRU Cell
<u>1900</u>

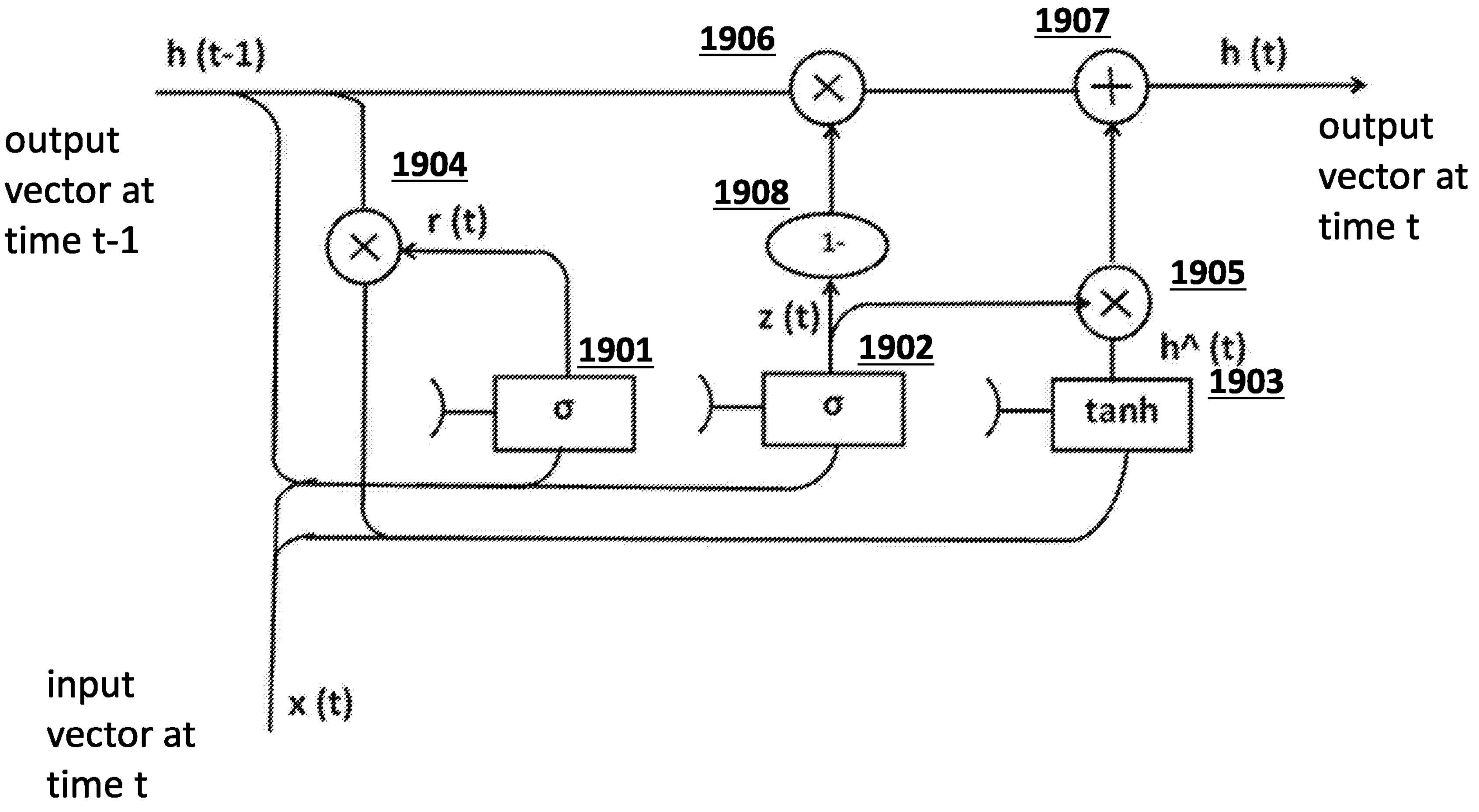


FIGURE 20

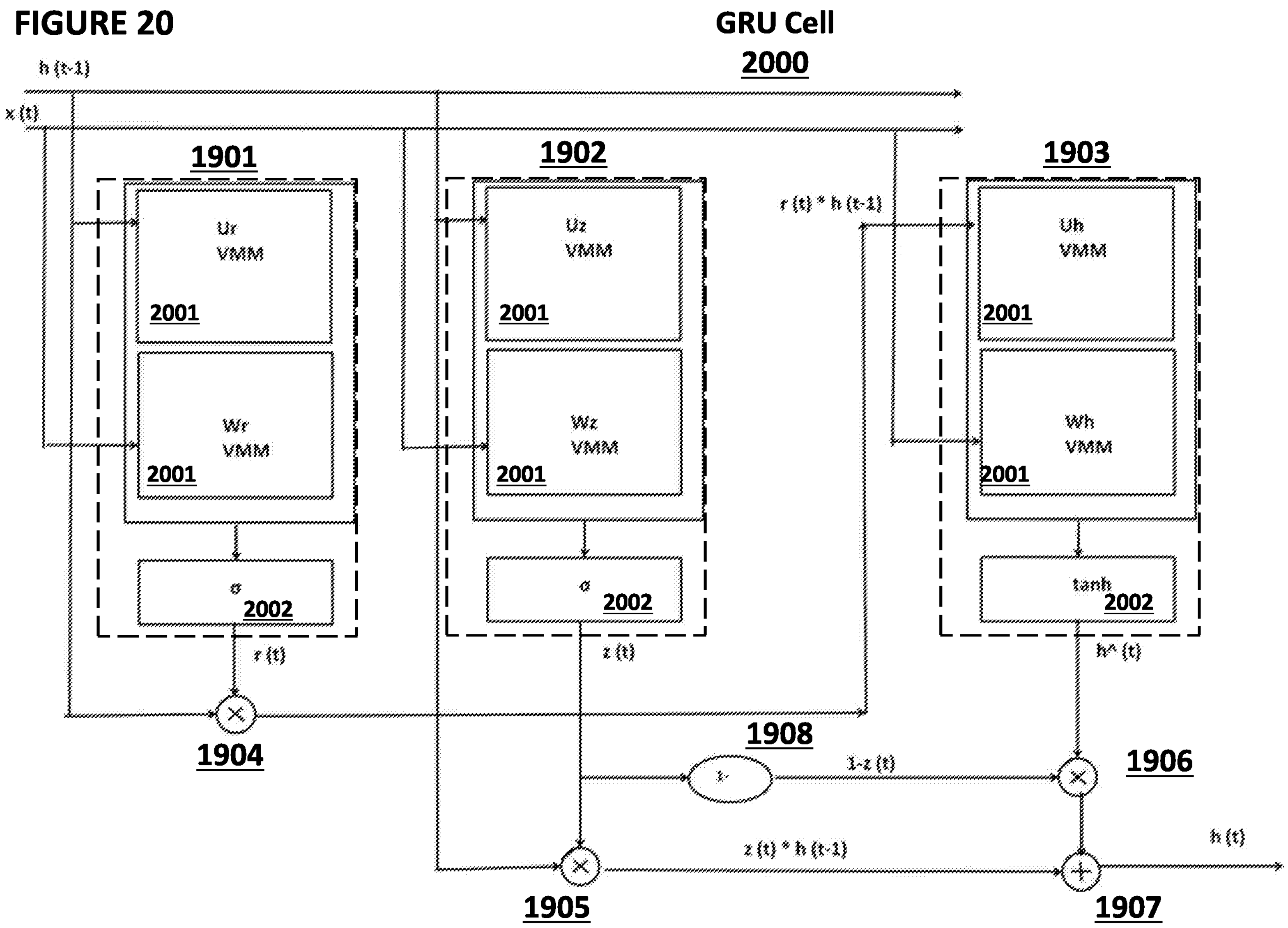

GRU Cell
2000
h (t-1)
x (t)
1901
Ur
VMM
2001
Wr
VMM
2001
σ
2002
r (t)
1904
1902
Uz
VMM
2001
Wz
VMM
2001
σ
2002
z (t)
r (t) * h (t-1)
1903
Uh
VMM
2001
Wh
VMM
2001
tanh
2002
h^ (t)
1908
1-
1-z (t)
1906
z (t) * h (t-1)
1905
h (t)
1907

FIGURE 21

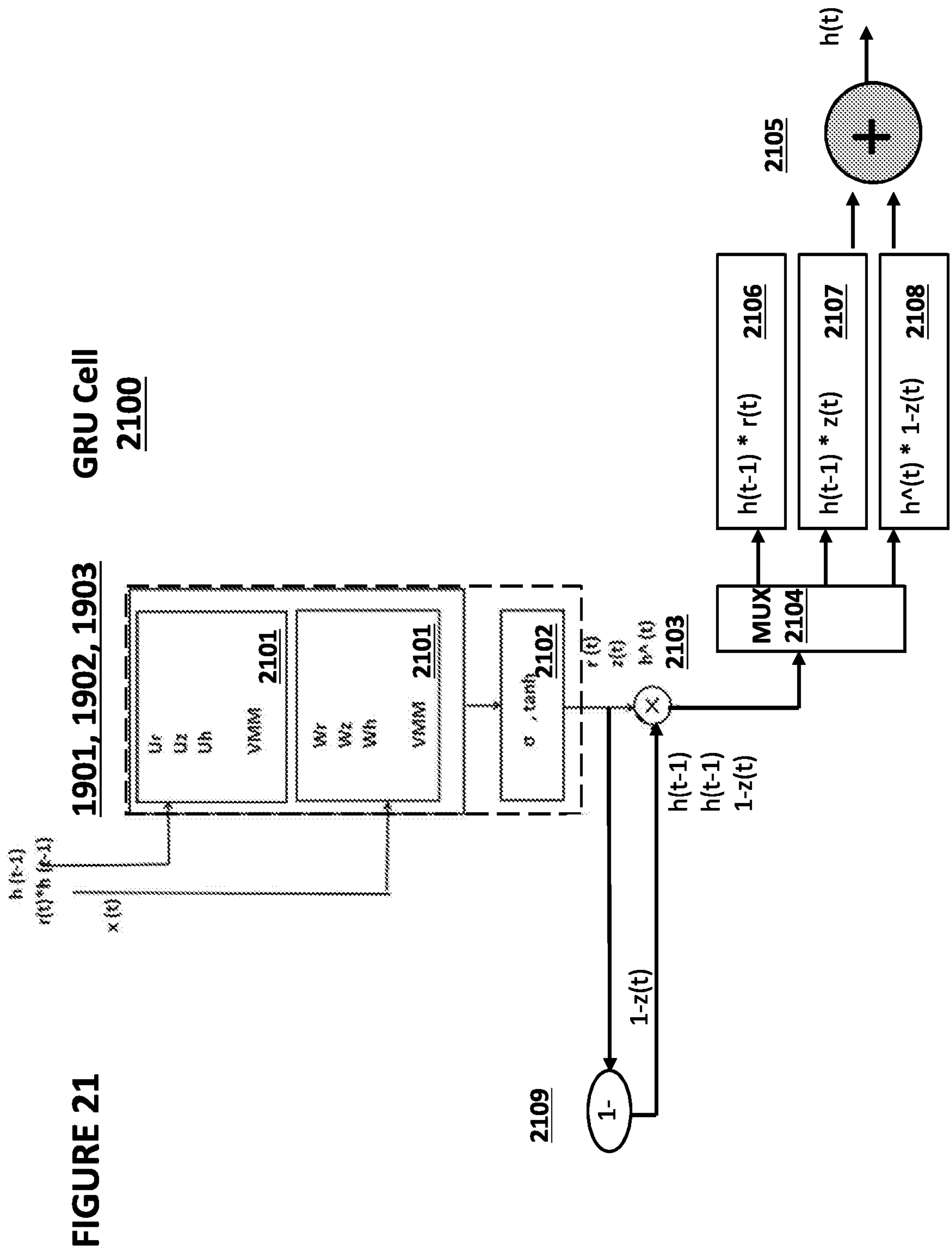

GRU Cell
2100
1901, 1902, 1903
h(t-1)
r(t)*h(t-1)
x(t)
Ur
Uz
Uh
VMM
2101
Wr
Wz
Wh
VMM
2101
σ , tanh
2102
r(t)
z(t)
h^(t)
2103
h(t-1)
h(t-1)
1-z(t)
MUX
2104
h(t-1) * r(t)
2106
h(t-1) * z(t)
2107
h^(t) * 1-z(t)
2108
2105
h(t)
1-z(t)
1-
2109

FIGURE 22

Flash Analog Neuromorphic Memory Shared Architecture System 2200

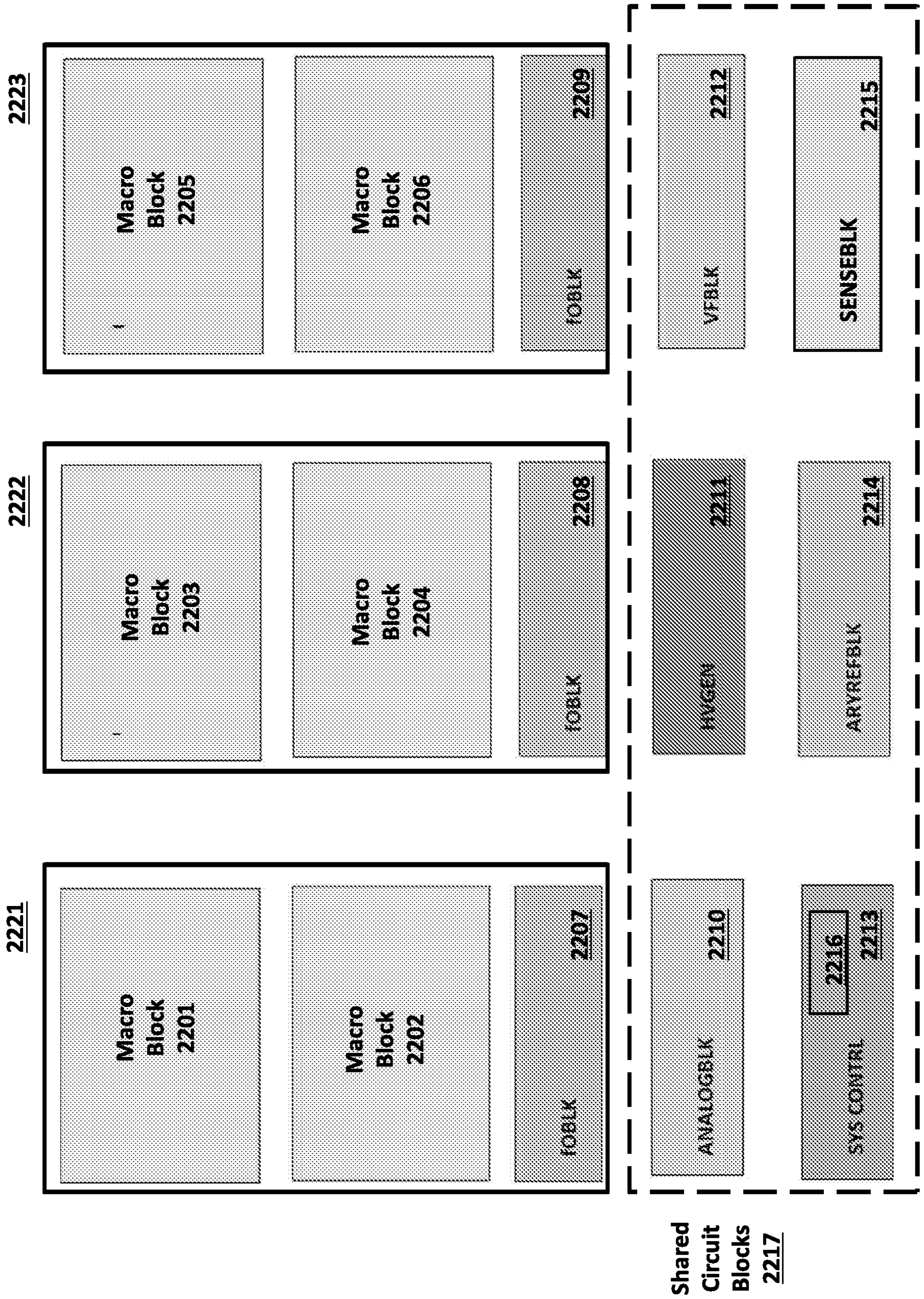


FIGURE 23

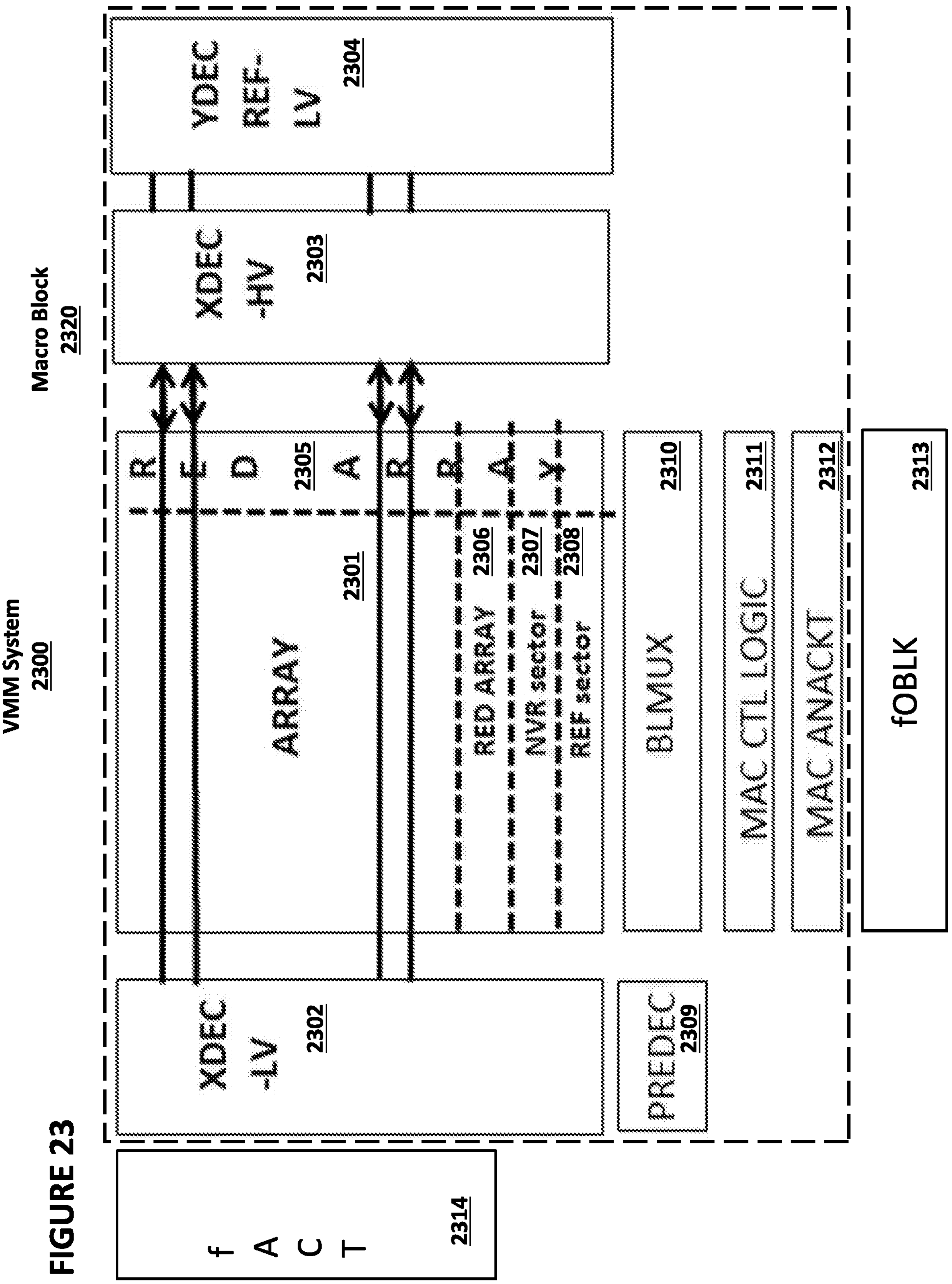
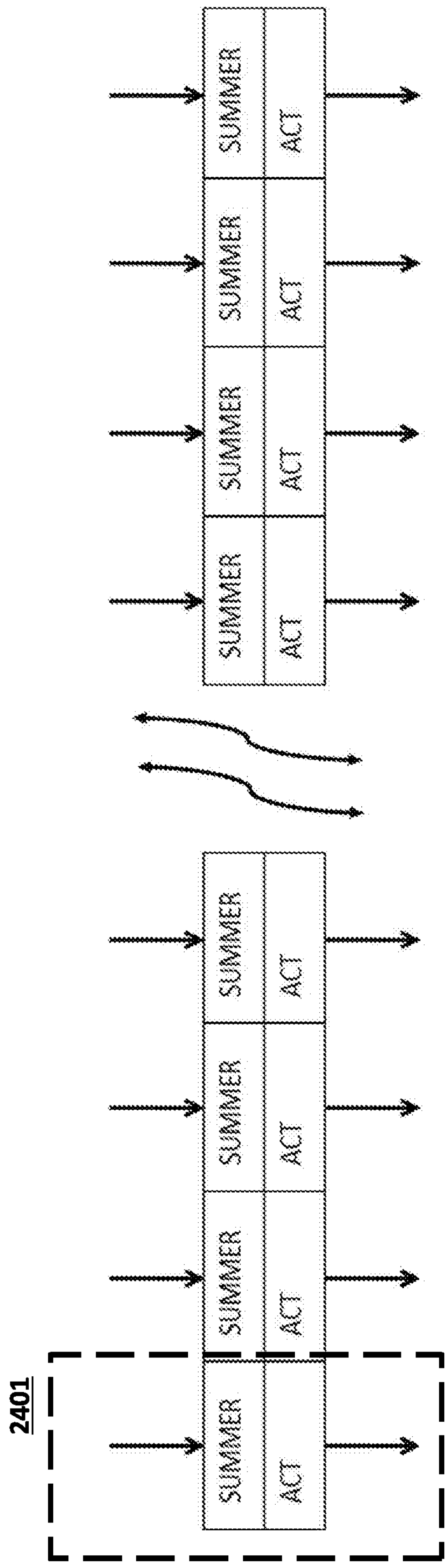

VMM System
2300
Macro Block
2320
YDEC
REF-
LV
2304
XDEC
-HV
2303
XDEC
-LV
2302
R
E
D
2305
A
R
R
A
Y
ARRAY
2301
RED ARRAY
2306
NVR sector
2307
REF sector
2308
PREDEC
2309
BLMUX
2310
MAC CTL LOGIC
2311
MAC ANACKT
2312
fOBLK
2313
f
A
C
T
2314

FIGURE 24


2400
2401
SUMMER
ACT
SUMMER
ACT
SUMMER
ACT
SUMMER
ACT
SUMMER
ACT
SUMMER
ACT
SUMMER
ACT
SUMMER
ACT

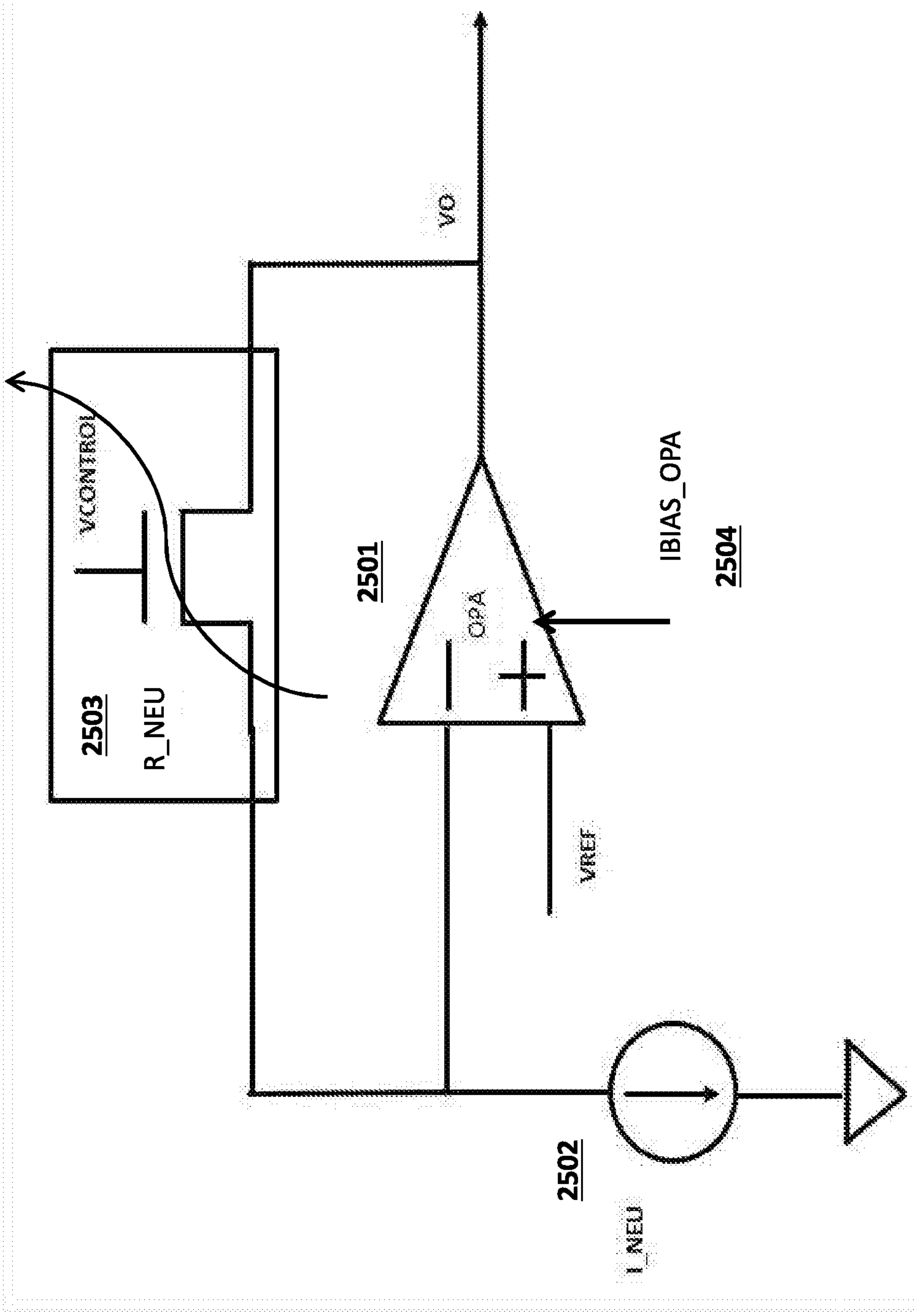

Adaptable Neuron
2500
VO
VCONTROL
2503
R_NEU
2501
OPA
IBIAS_OPA
2504
VREF
2502
I_NEU

FIGURE 25

FIGURE 26

Activation Function Circuit
2600

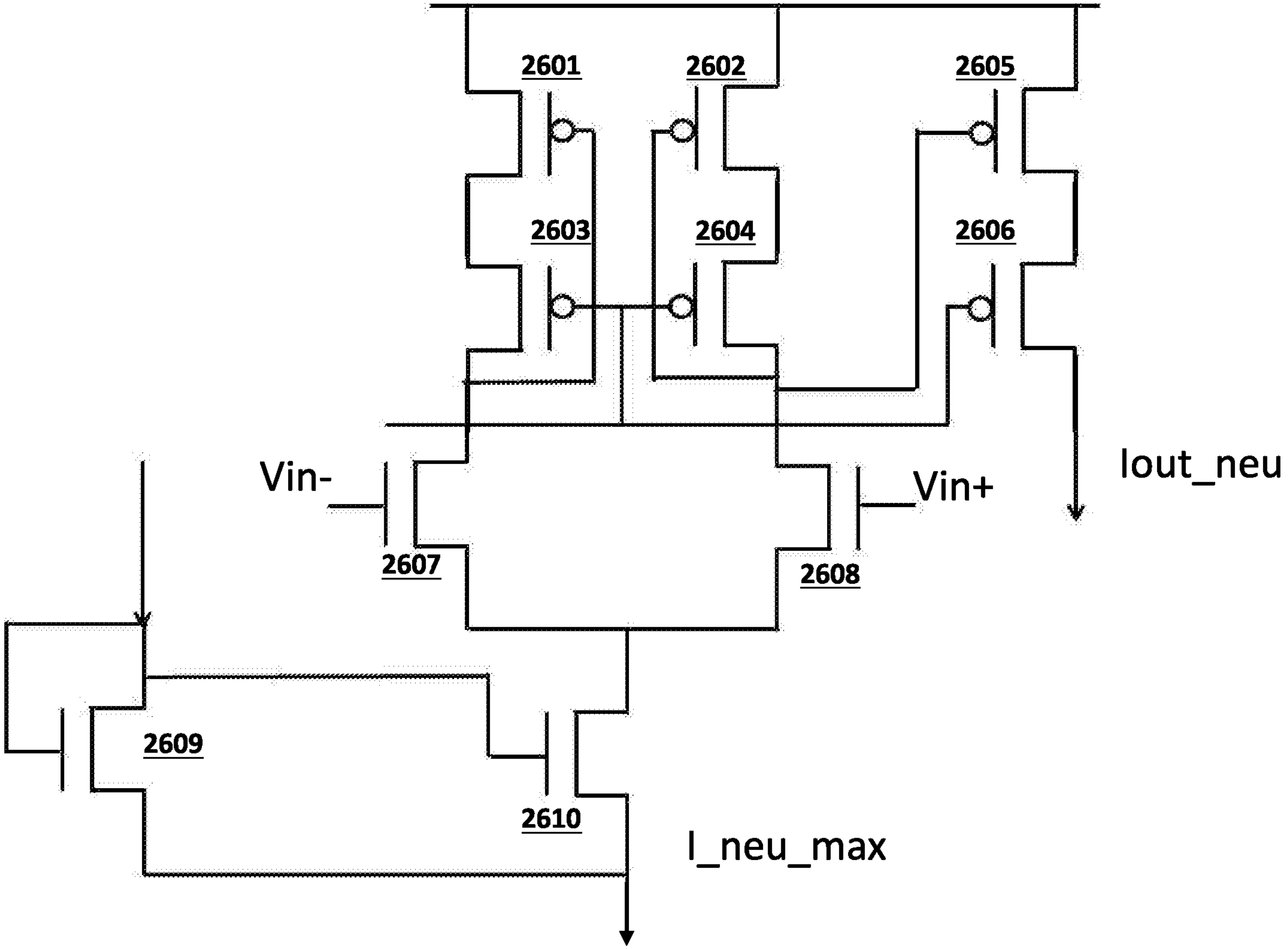


FIGURE 27

Operational Amplifier
2700

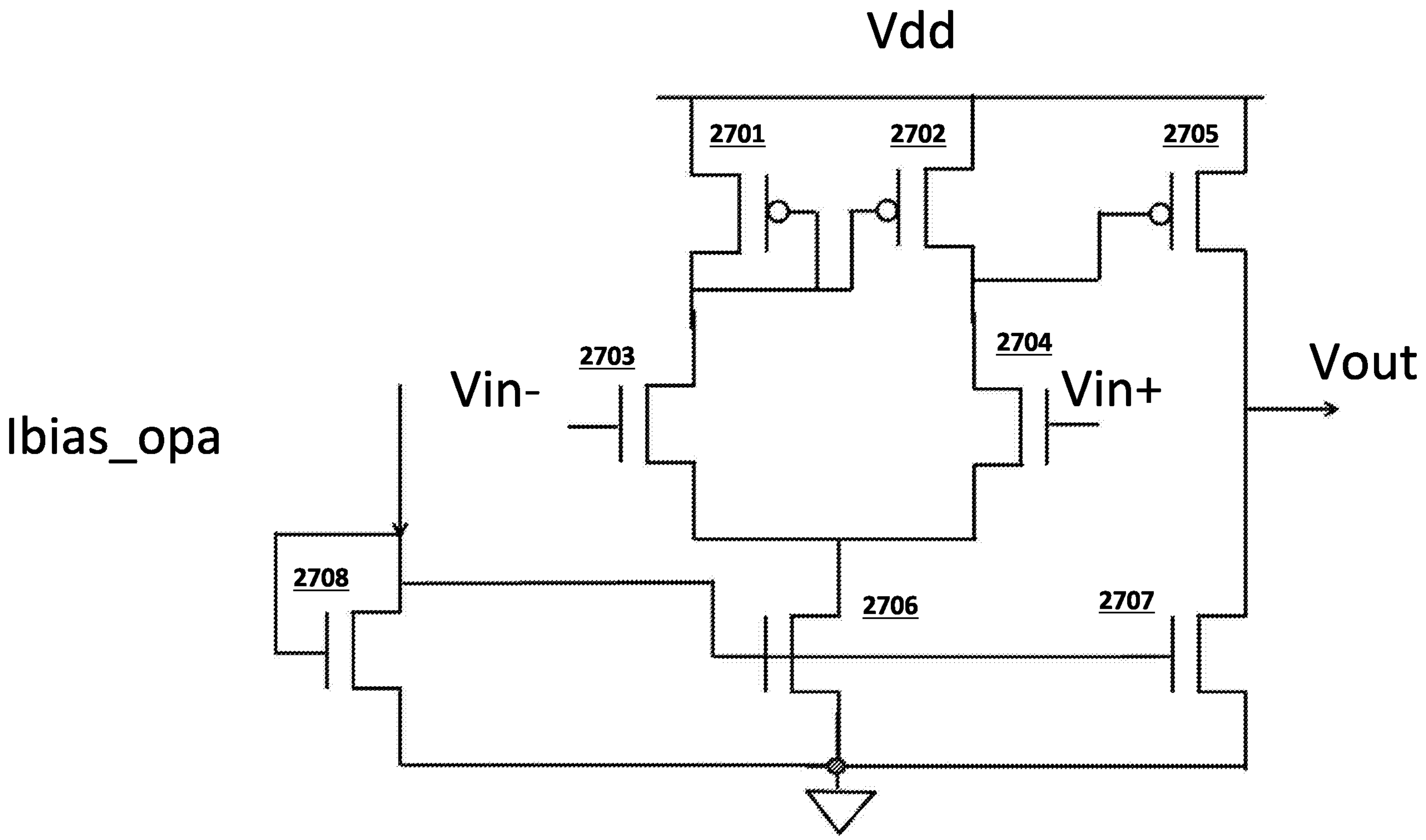
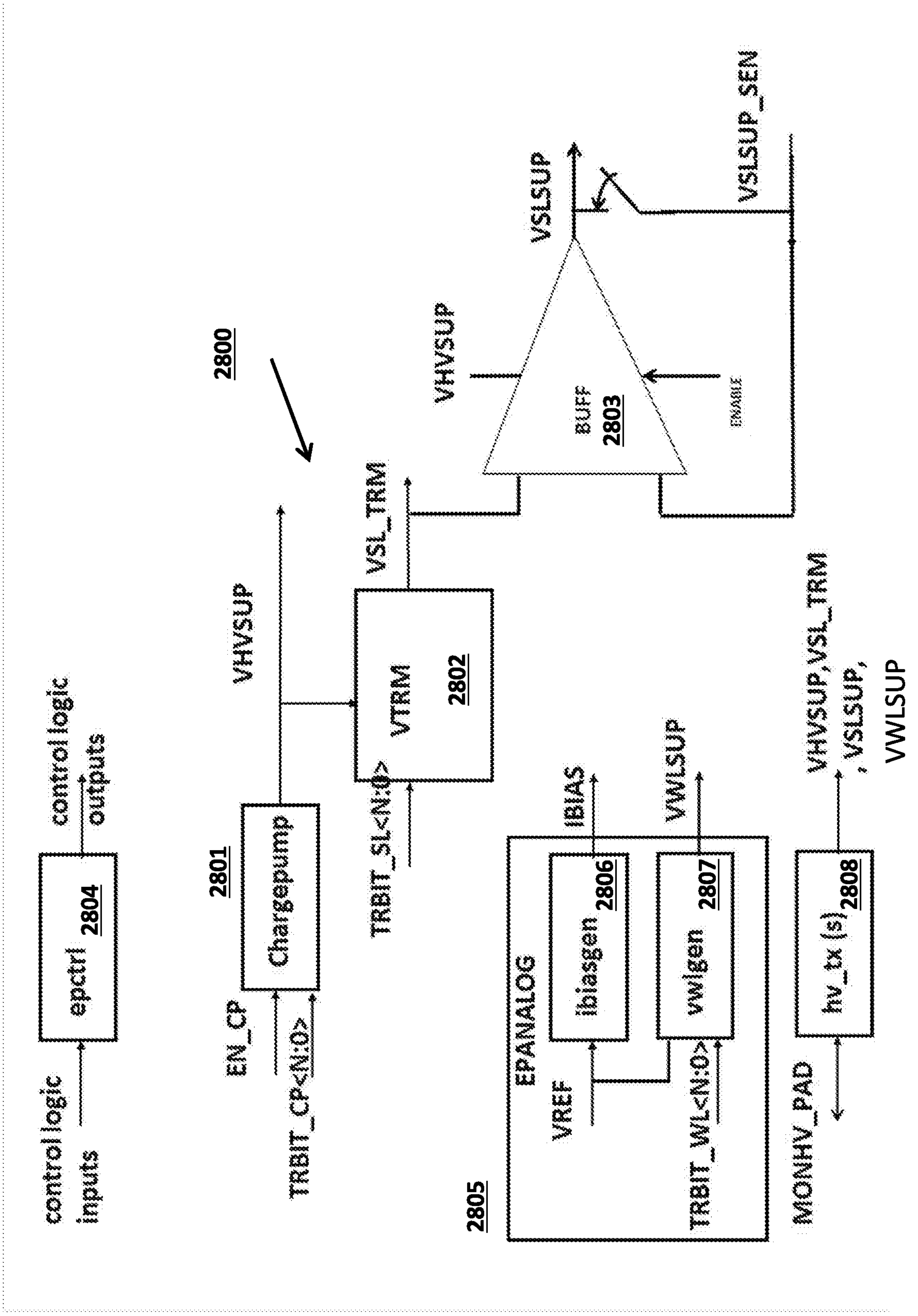


FIGURE 28


2800
control logic inputs
epctrl
2804
control logic outputs
EN_CP
TRBIT_CP<N:0>
Chargepump
2801
VHVSUP
TRBIT_SL<N:0>
VTRM
2802
VSL_TRM
VHVSUP
BUFF
2803
ENABLE
VSLSUP
VSLSUP_SEN
2805
EPANALOG
VREF
ibiasgen
2806
IBIAS
TRBIT_WL<N:0>
vwlgen
2807
VWLSUP
MONHV_PAD
hv_tx (s)
2808
VHVSUP, VSL_TRM, VSLSUP, VWLSUP

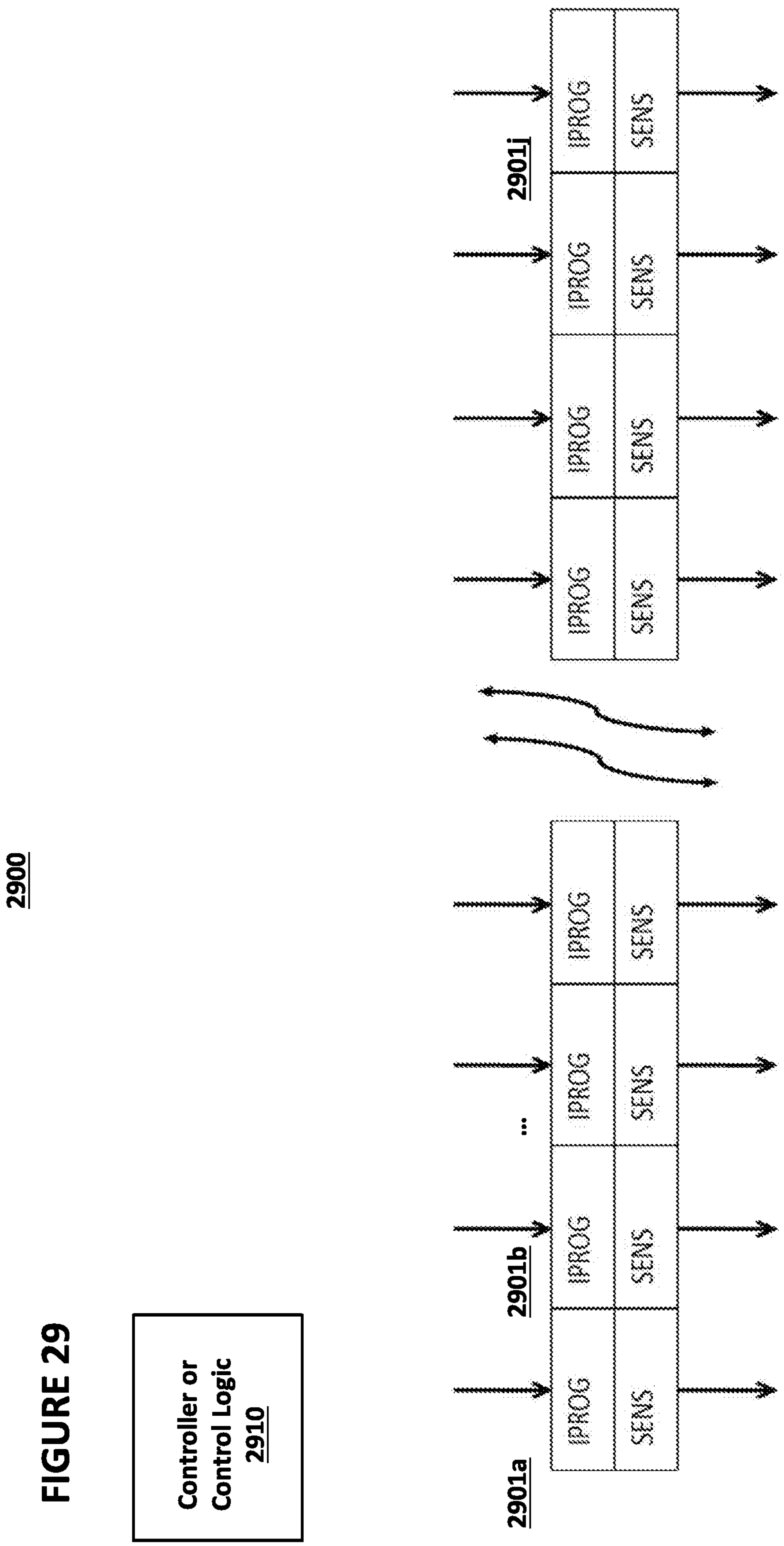
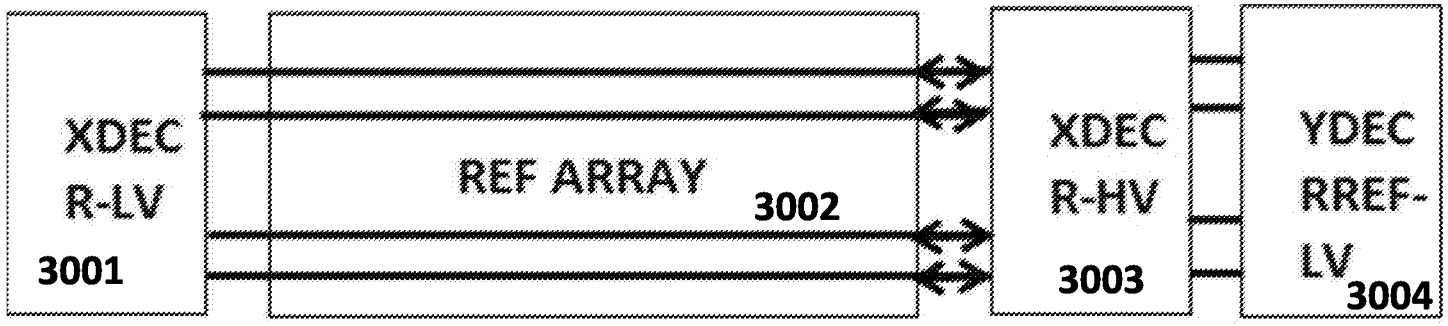


FIGURE 29

2900
Controller or Control Logic 2910
2901a
2901b
...
2901j
IPROG
SENS

FIGURE 30

Reference System
3000



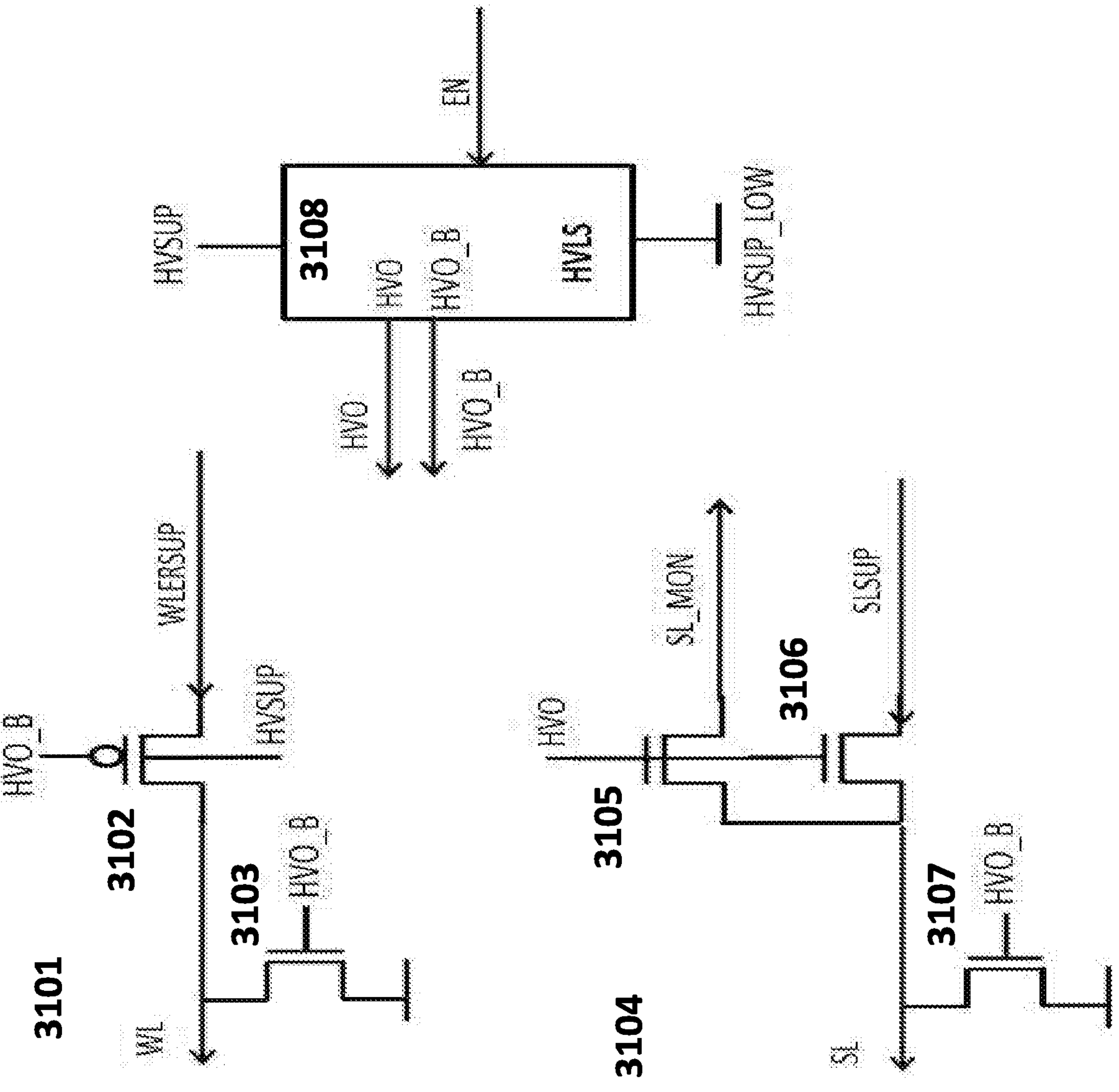

EN
HVSUP
3108
HVO
HVO_B_
HVLS
HVSUP_LOW
HVO
HVO_B_
WLERSUP
HVO_B_
HVSUP
3102
3103
HVO_B_
3101
WL
HVO
SL_MON
3106
SLSUP
3105
3107
HVO_B
3104
SL

FIGURE 31

FIGURE 32

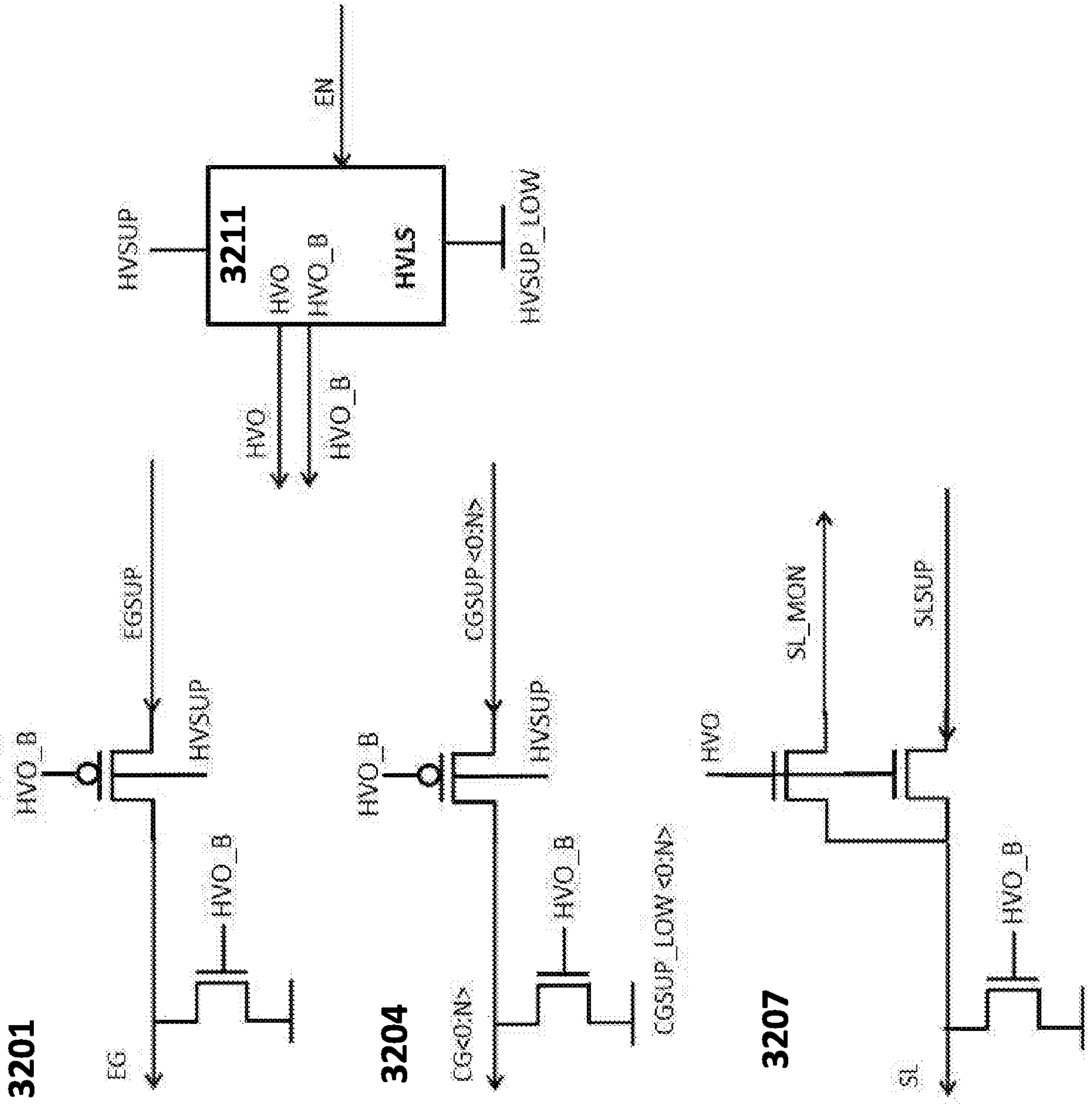

3201
EG
HVO_B
HVO_B
EGSUP
HVSUP
3204
CG<0:N>
HVO_B
HVO_B
CGSUP <0:N>
HVSUP
CGSUP_LOW <0:N>
3207
SL
HVO_B
HVO
SL_MON
SLSUP
3211
HVLS
HVSUP
HVO
HVO_B
EN
HVSUP_LOW
HVO
HVO_B

FIGURE 33

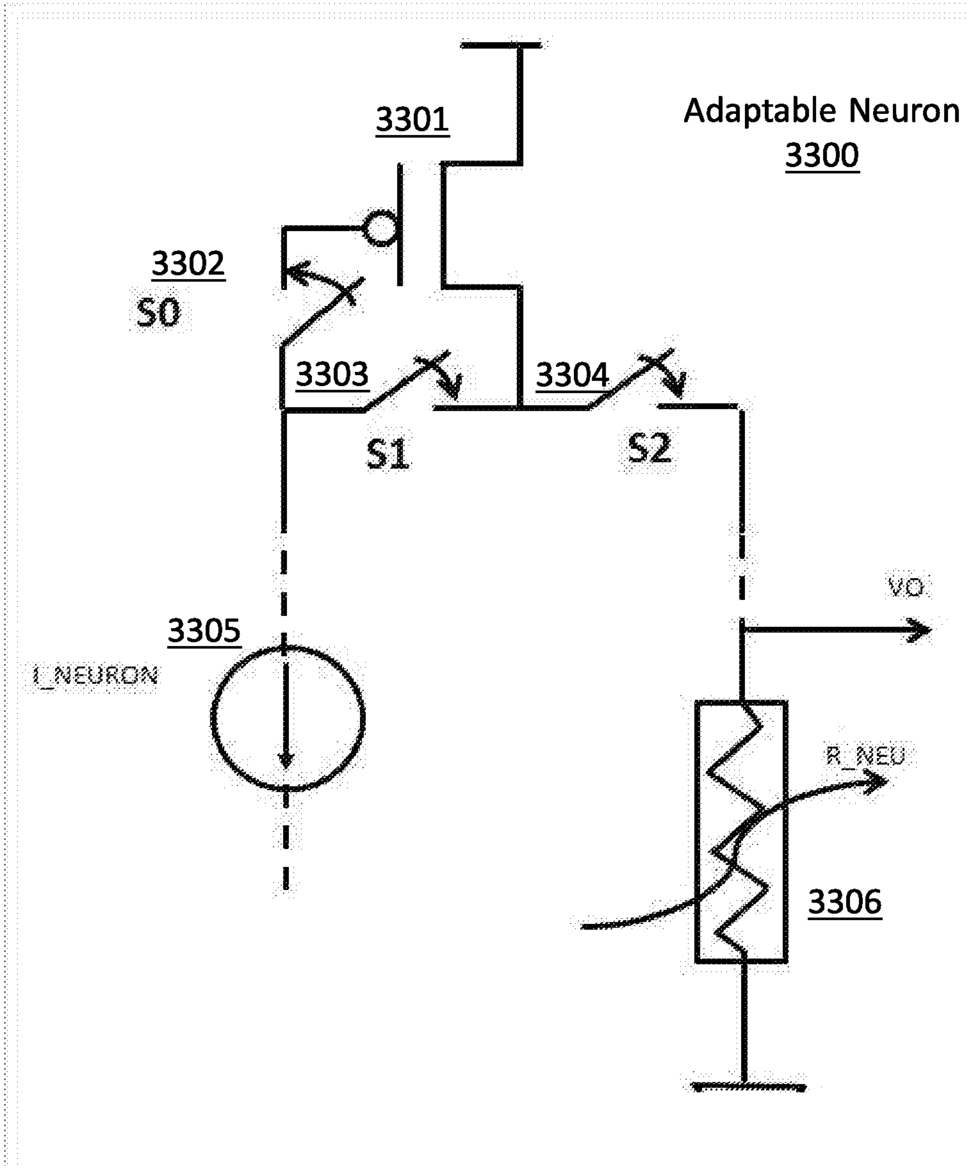

Adaptable Neuron
3300
3301
3302
S0
3303
S1
3304
S2
3305
I_NEURON
VO
R_NEU
3306

FIGURE 34

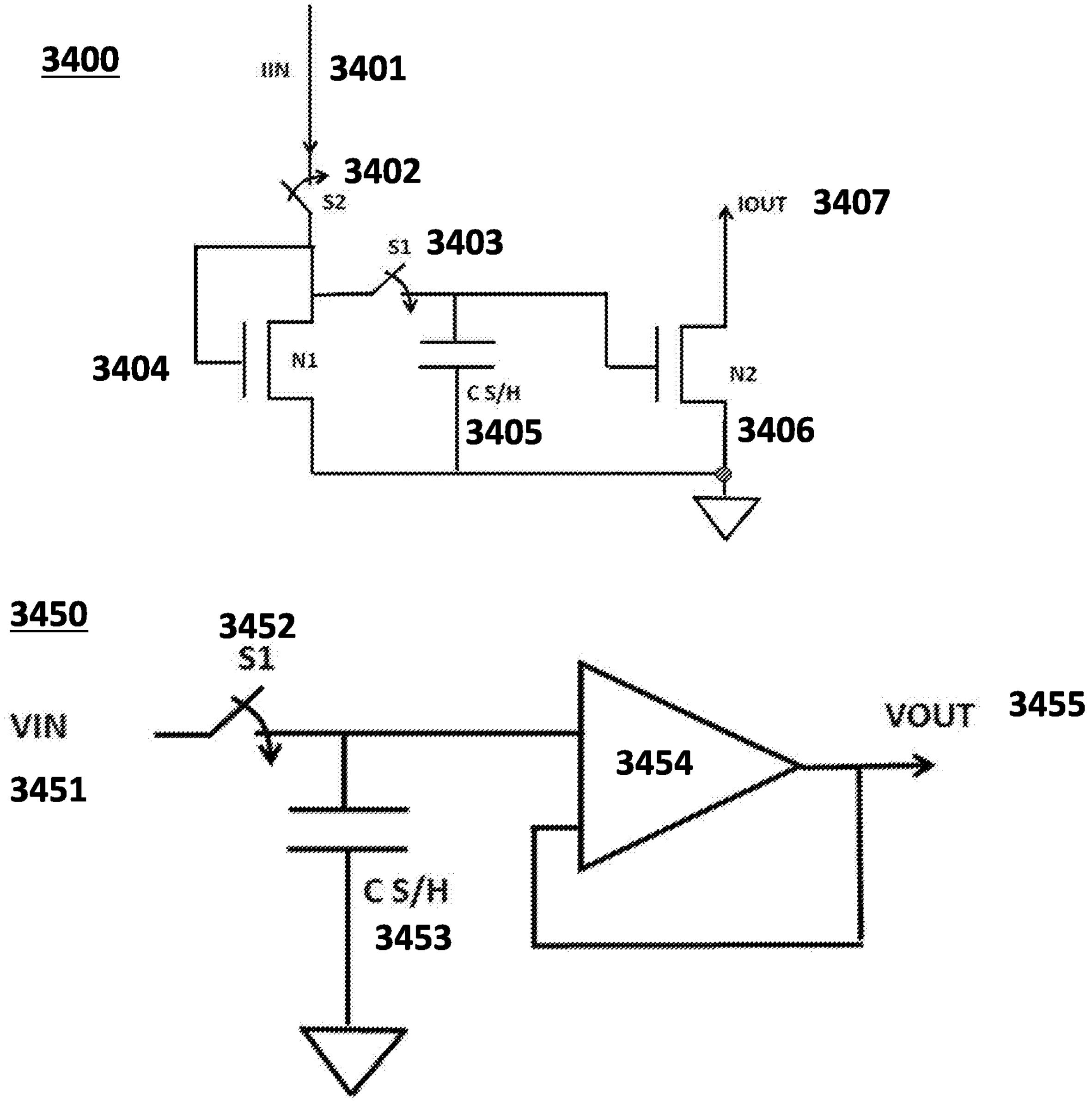

3400
IIN
3401
3402
S2
S1
3403
IOUT
3407
3404
N1
C S/H
3405
N2
3406
3450
3452
S1
VIN
3451
3454
VOUT
3455
C S/H
3453

FIGURE 35

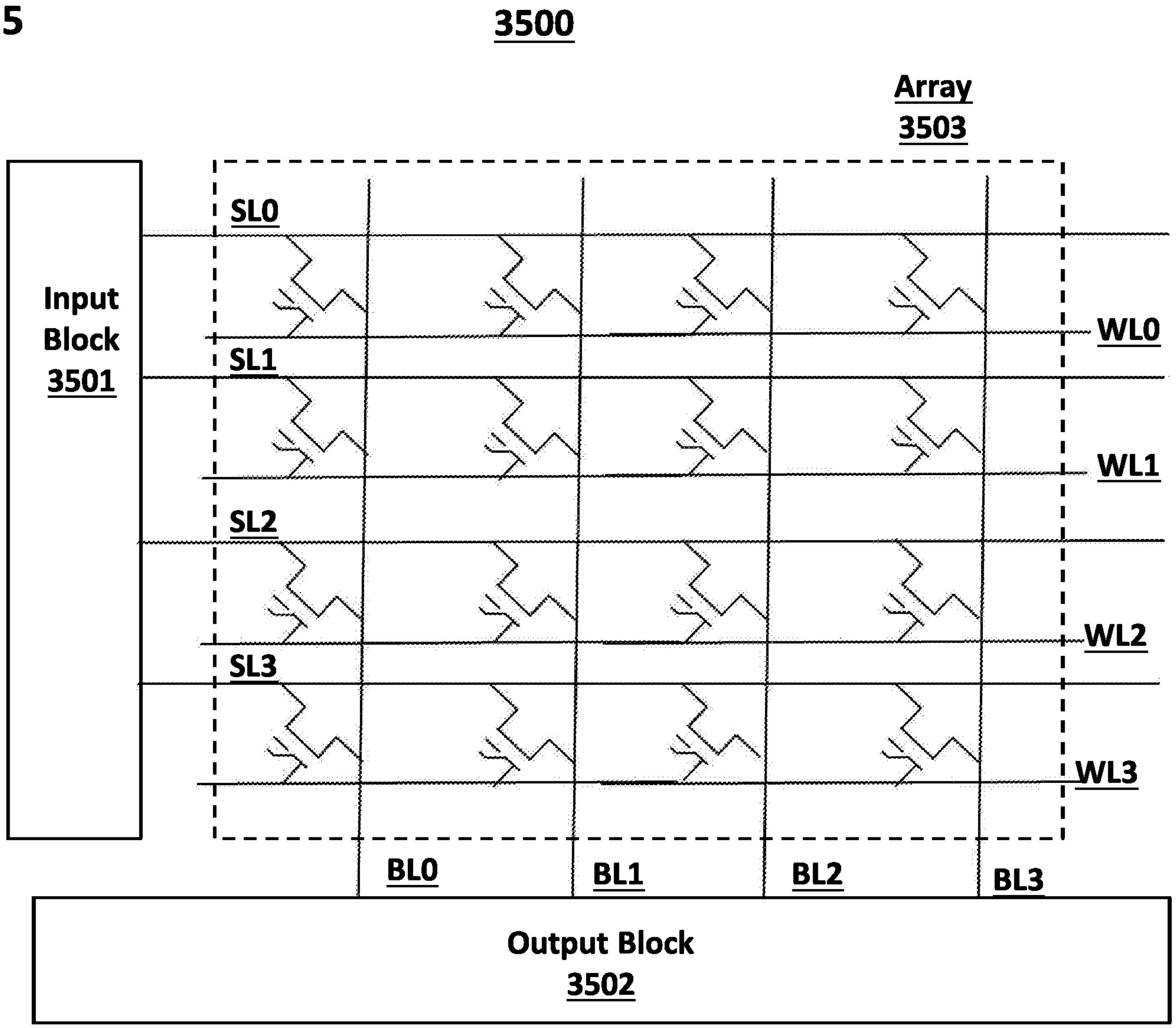
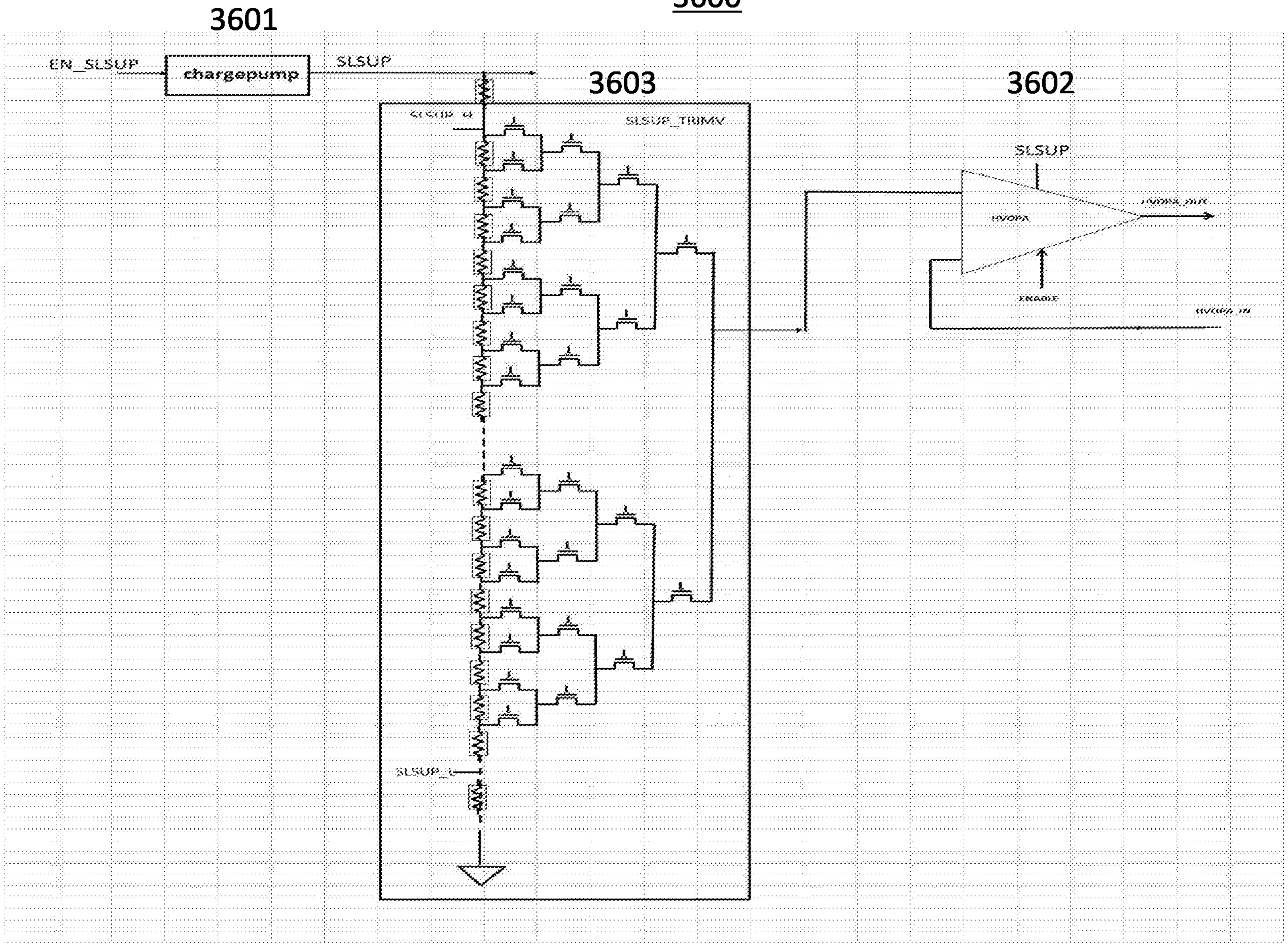

3500
Array
3503
Input
Block
3501
SL0
SL1
SL2
SL3
WL0
WL1
WL2
WL3
BL0
BL1
BL2
BL3
Output Block
3502

FIGURE 36

High Voltage Generation Block
3600



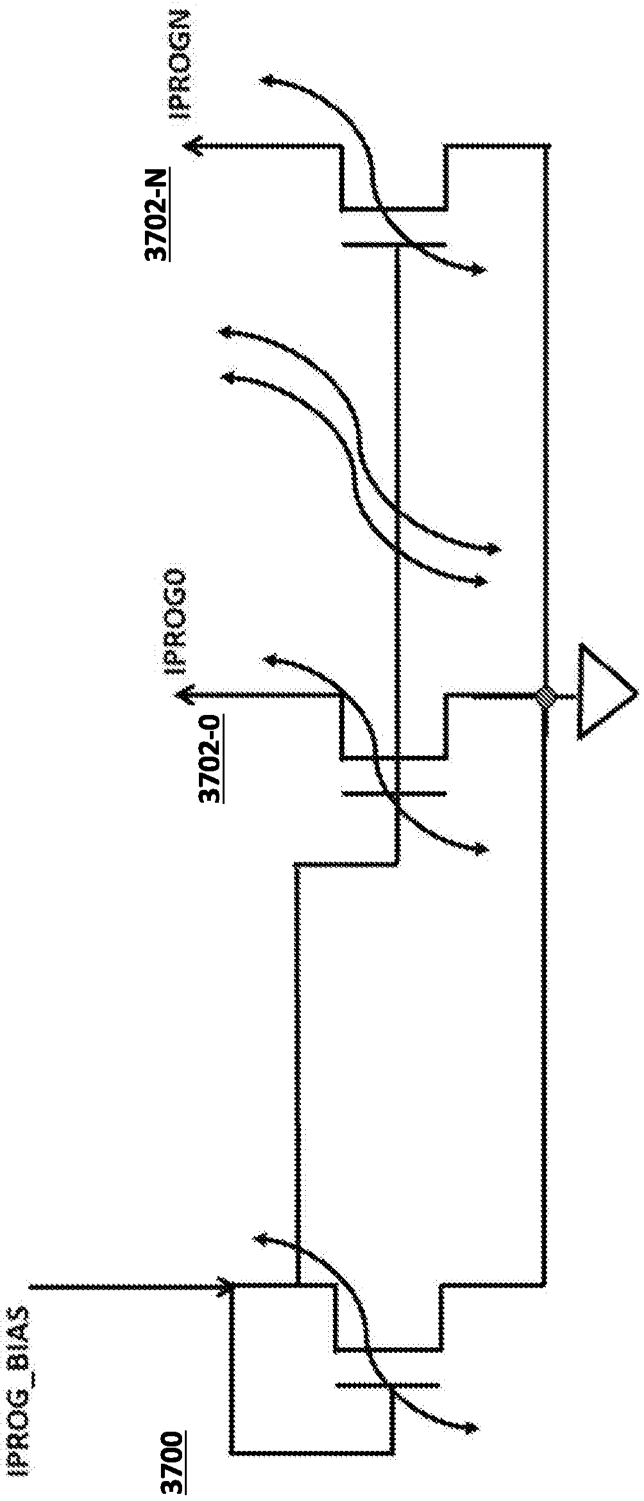
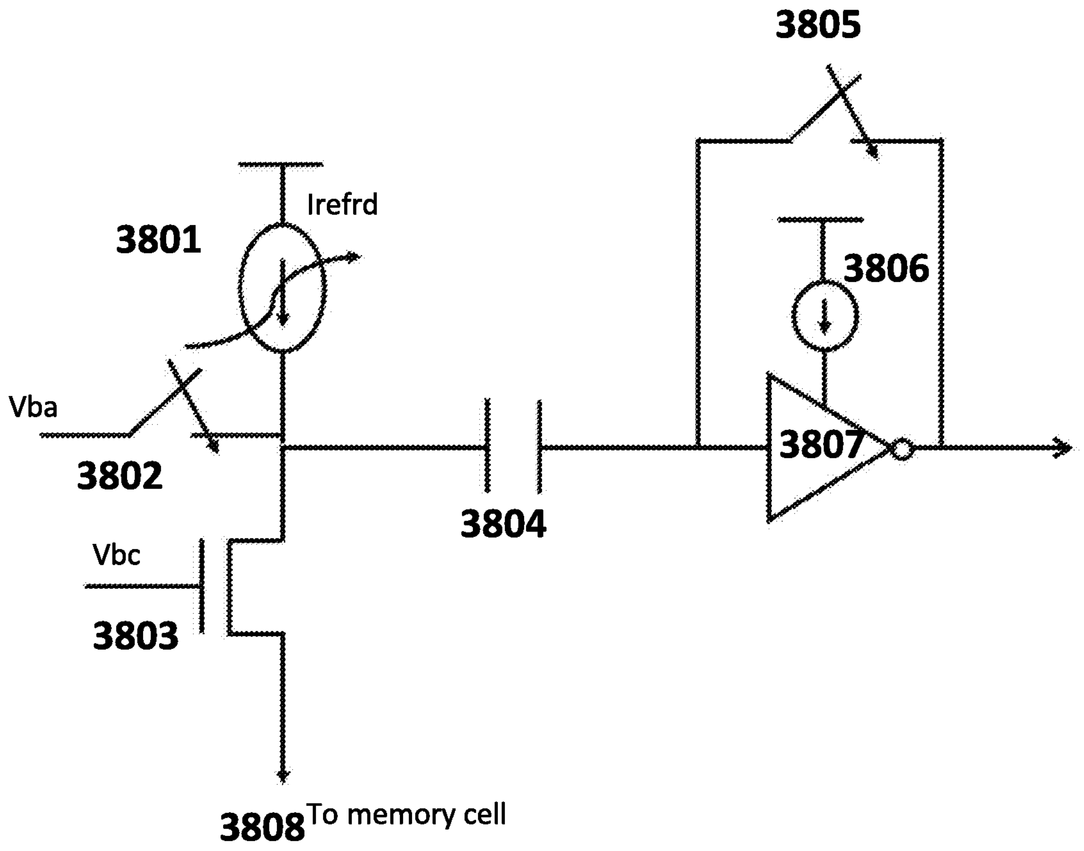


FIGURE 37

FIGURE 38

Sense Amplifier
3800



REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 62720902 **[0001]**
- US 18249218 **[0001]**
- US 594439 **[0006]**
- US 5029130 A **[0015]**
- US 6747310 B **[0020]**
- US 826345 **[0050]**