US 20150302295A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2015/0302295 A1

(43) Pub. Date: Oct. 22, 2015

Rivera et al.

(54) **GLOBALLY ASYNCHRONOUS AND LOCALLY SYNCHRONOUS (GALS) NEUROMORPHIC NETWORK**

(75) Inventors: **Rodrigo Alvarez-Icaza Rivera**, San Jose, CA (US); **John V. Arthur**, Mountain View, CA (US); **Andrew S. Cassidy**, San Jose, CA (US); **Paul A. Merolla**, Palo Alto, CA (US); **Dharmendra S. Modha**, San Jose, CA (US)

(73) Assignee: **International Business Machines Corporation**, Armonk, NY (US)

(21) Appl. No.: **13/563,333**

(22) Filed: **Jul. 31, 2012**

**Publication Classification**

(51) **Int. Cl.**

*G06N 3/08* (2006.01)

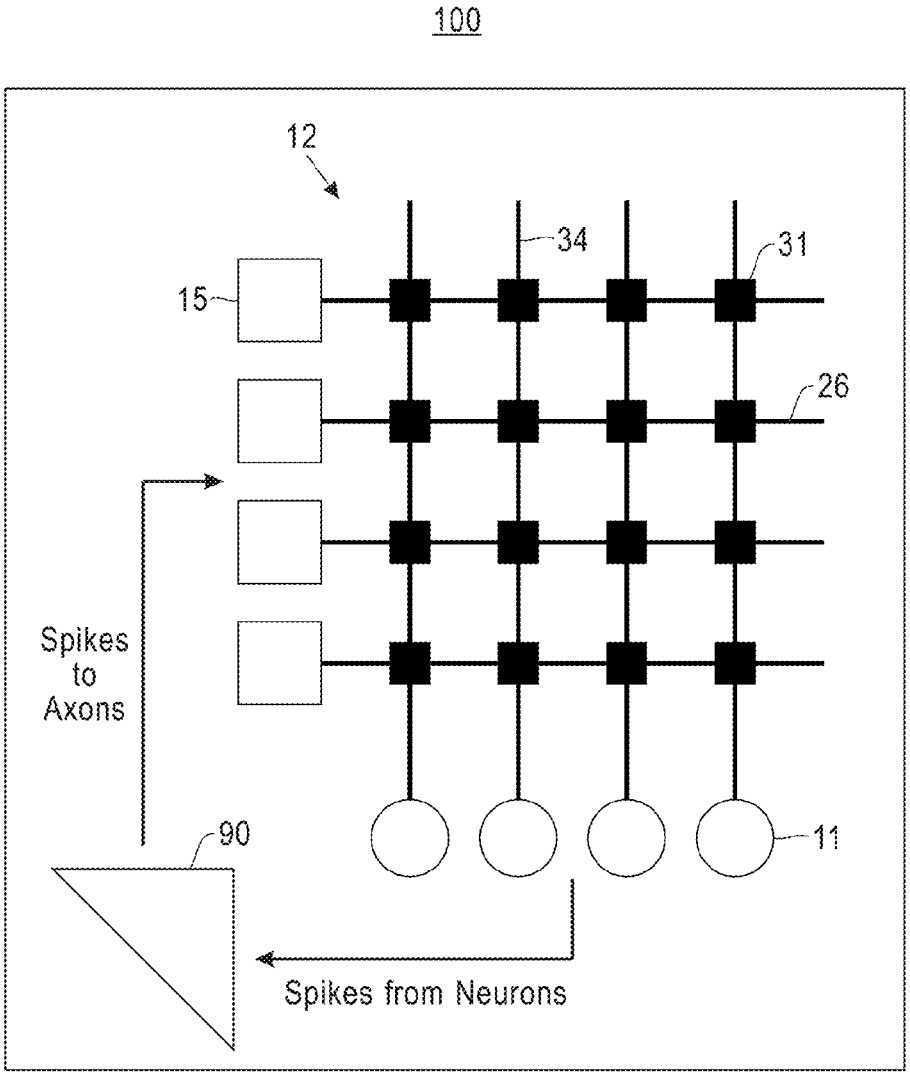
*G06N 3/04* (2006.01)

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

CPC ... ***G06N 3/08*** (2013.01); ***G06N 3/04*** (2013.01)

(57) **ABSTRACT**

Embodiments of the invention relate to a globally asynchronous and locally synchronous neuromorphic network. One embodiment comprises generating a synchronization signal that is distributed to a plurality of neural core circuits. In response to the synchronization signal, in at least one core circuit, incoming spike events maintained by said at least one core circuit are processed to generate an outgoing spike event. Spike events are asynchronously communicated between the core circuits via a routing fabric comprising multiple asynchronous routers.



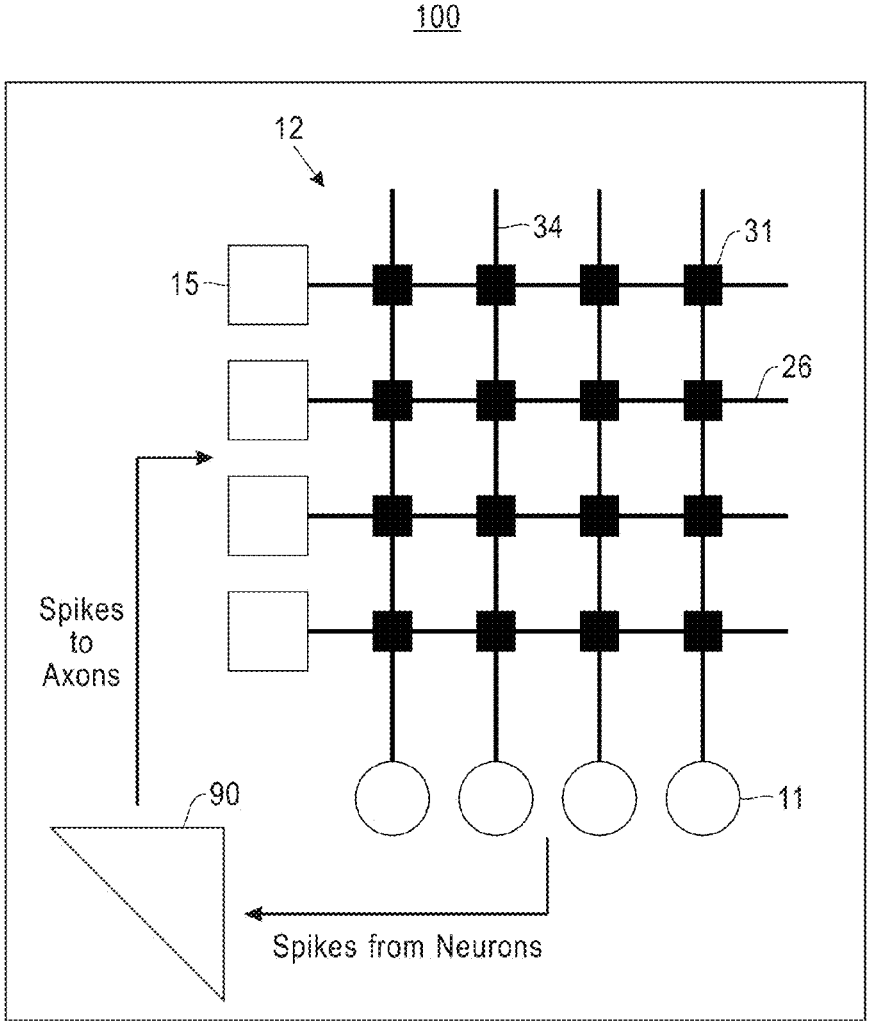



FIG. 1

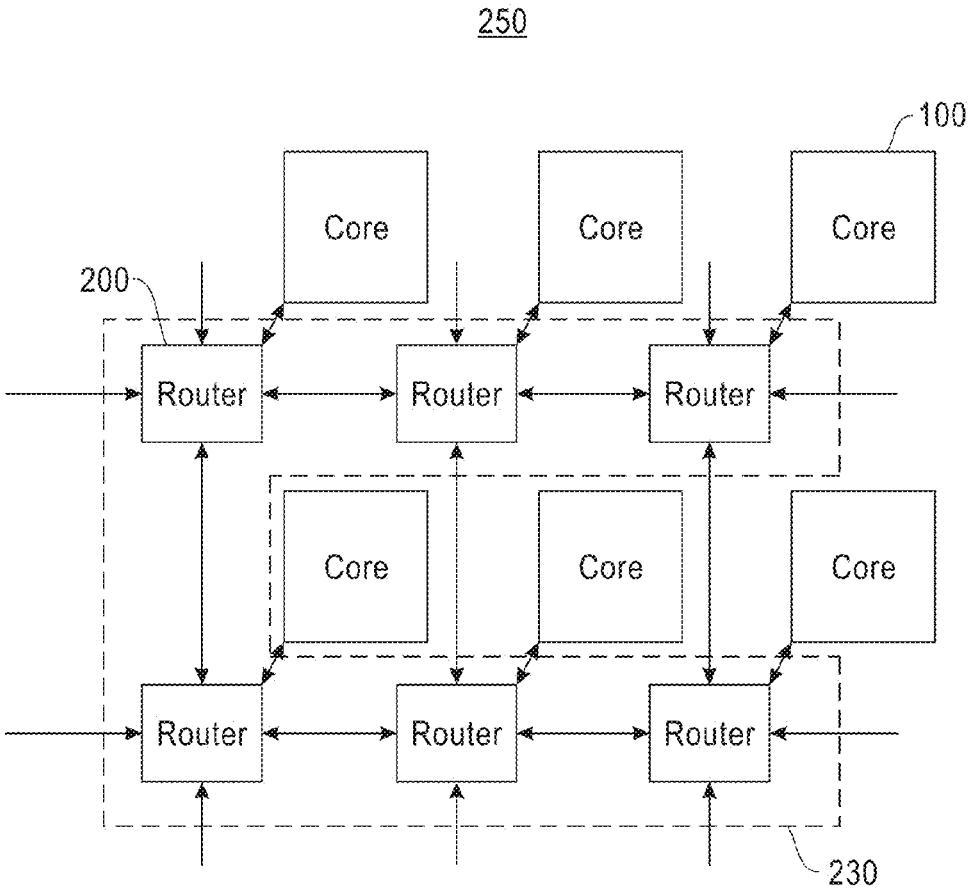

250
100
Core
Core
Core
200
Router
Router
Router
Core
Core
Core
Router
Router
Router
230

FIG. 2

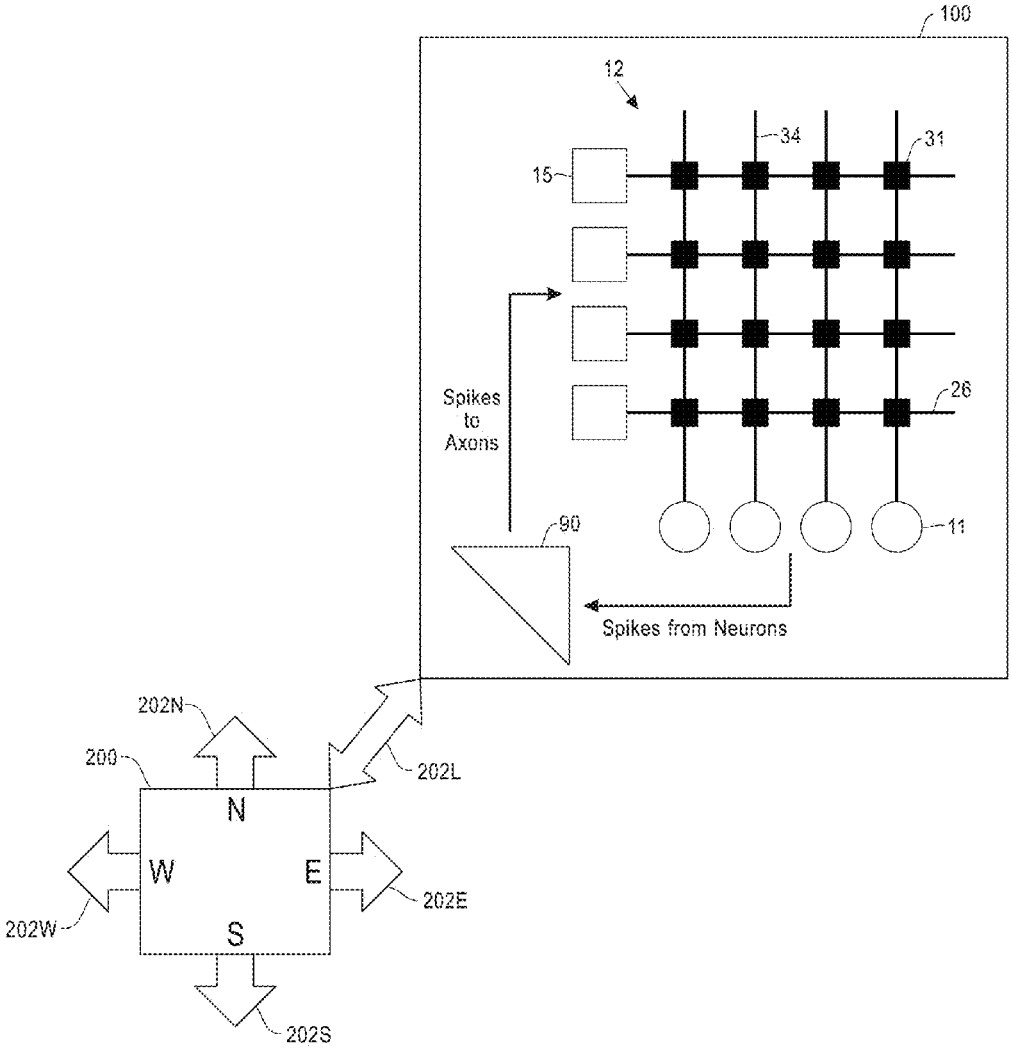

100
12
34
31
15
26
Spikes to Axons
90
11
Spikes from Neurons
202N
200
202L
N
W
E
202E
202W
S
202S

FIG. 3

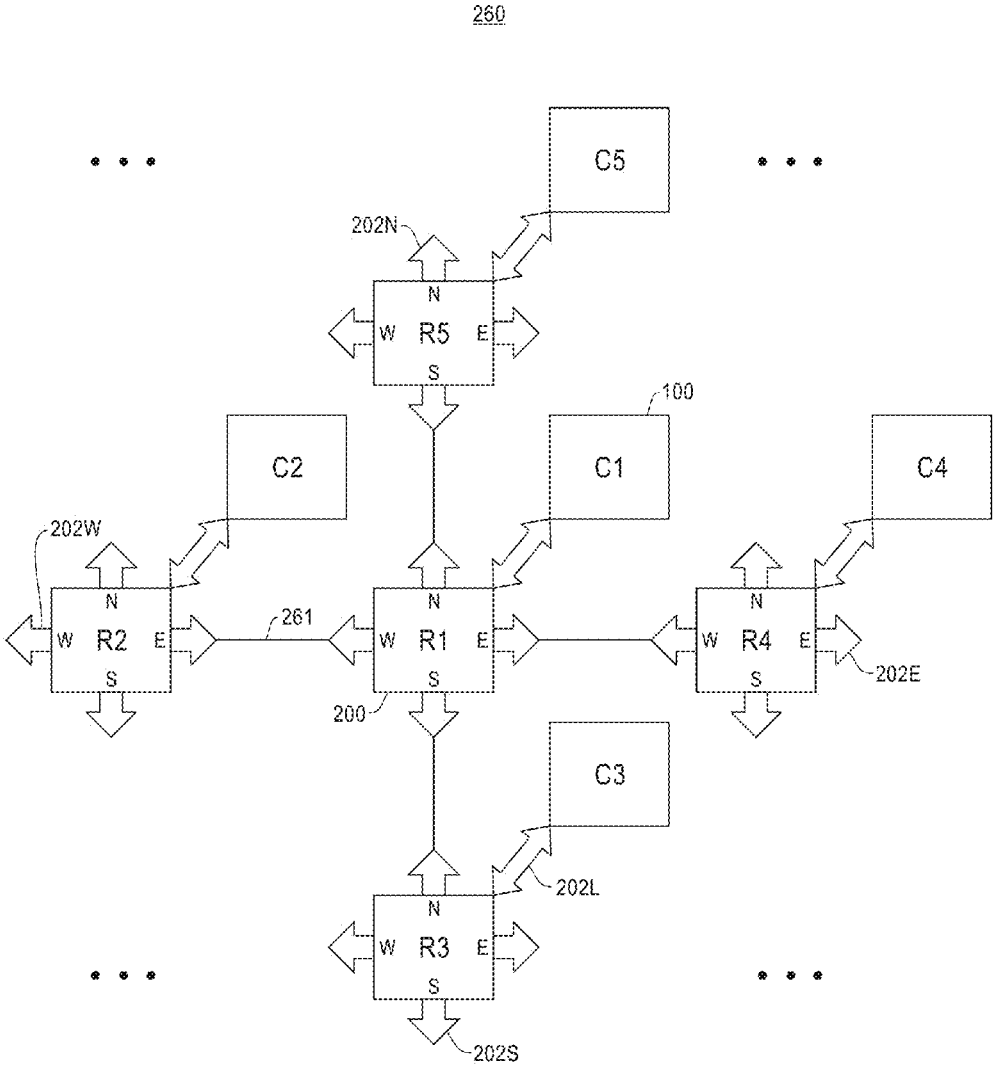

260
C5
202N
N
W R5 E
S
100
C2
C1
C4
202W
N
W R2 E
S
261
N
W R1 E
S
N
W R4 E
S
202E
200
C3
202L
N
W R3 E
S
202S

FIG. 4

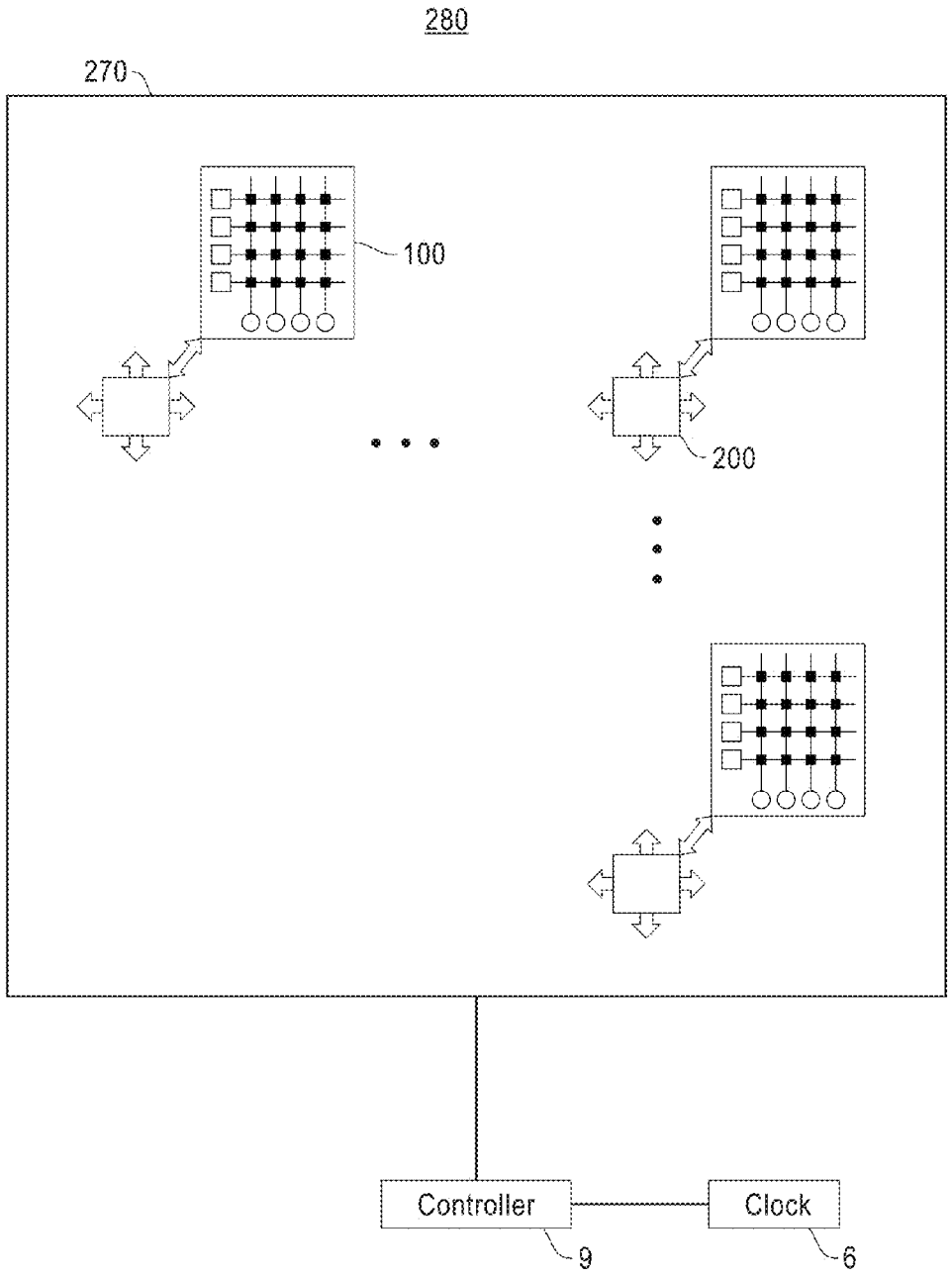



FIG. 5

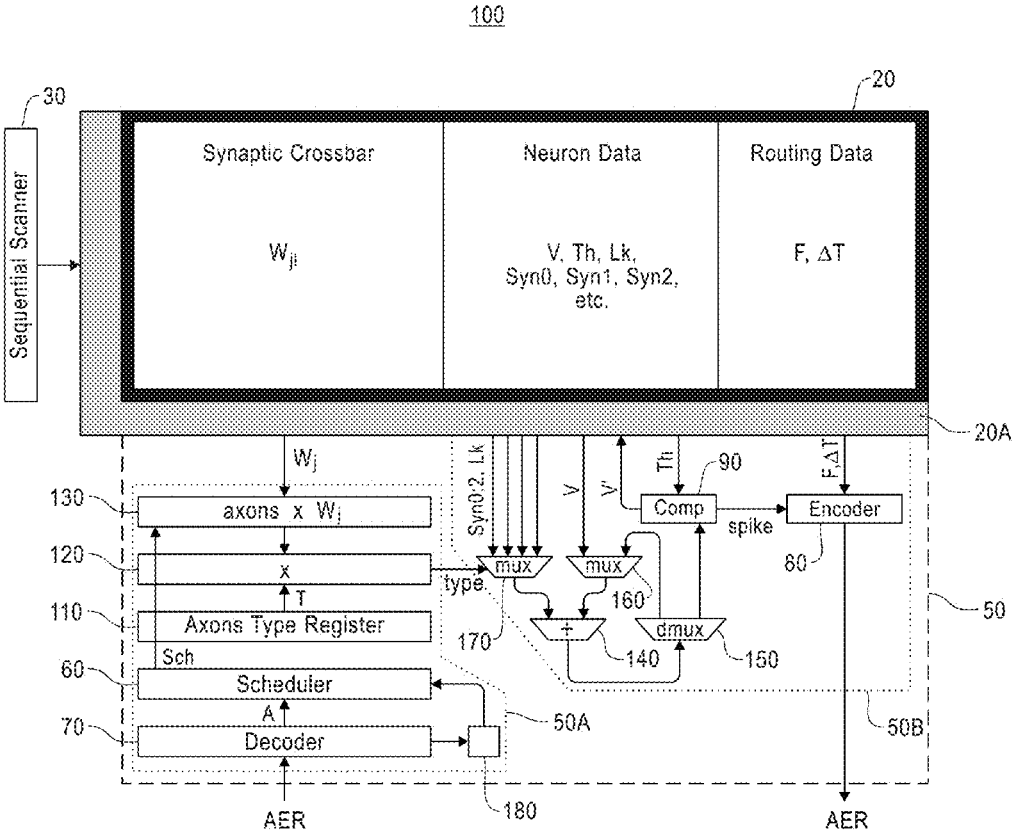

100
30
20
Sequential Scanner
Synaptic Crossbar
Neuron Data
Routing Data
$W_{ji}$
V, Th, Lk, Syn0, Syn1, Syn2, etc.
F, ΔT
20A
$W_j$
130
axons x $W_j$
120
x
T
110
Axons Type Register
Sch
60
Scheduler
A
70
Decoder
AER
Syn0:2, Lk
type
mux
mux
V
V'
Th
90
Comp
spike
F,ΔT
Encoder
60
160
+
dmux
140
150
170
180
50
50A
50B
AER

FIG. 6

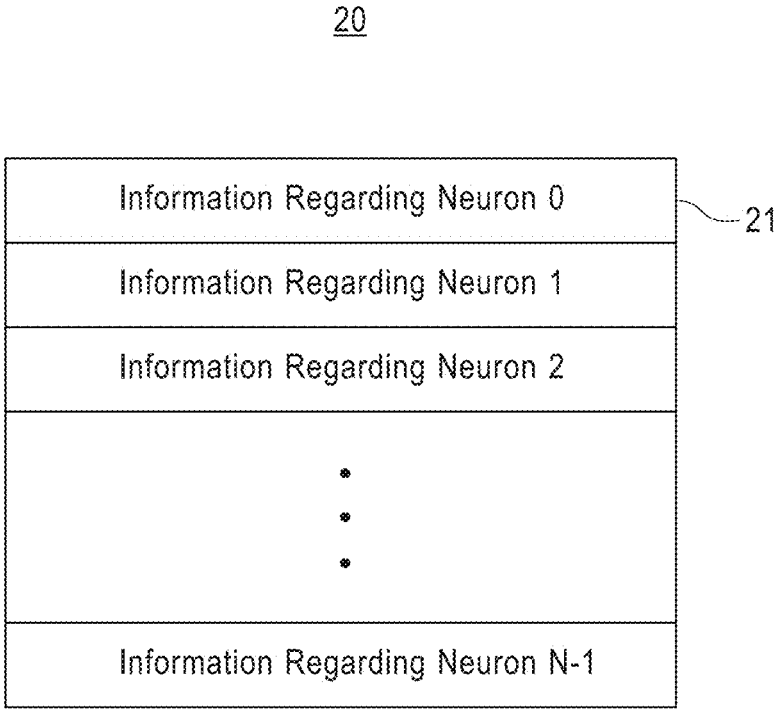

20
Information Regarding Neuron 0
21
Information Regarding Neuron 1
Information Regarding Neuron 2
Information Regarding Neuron N-1

FIG. 7

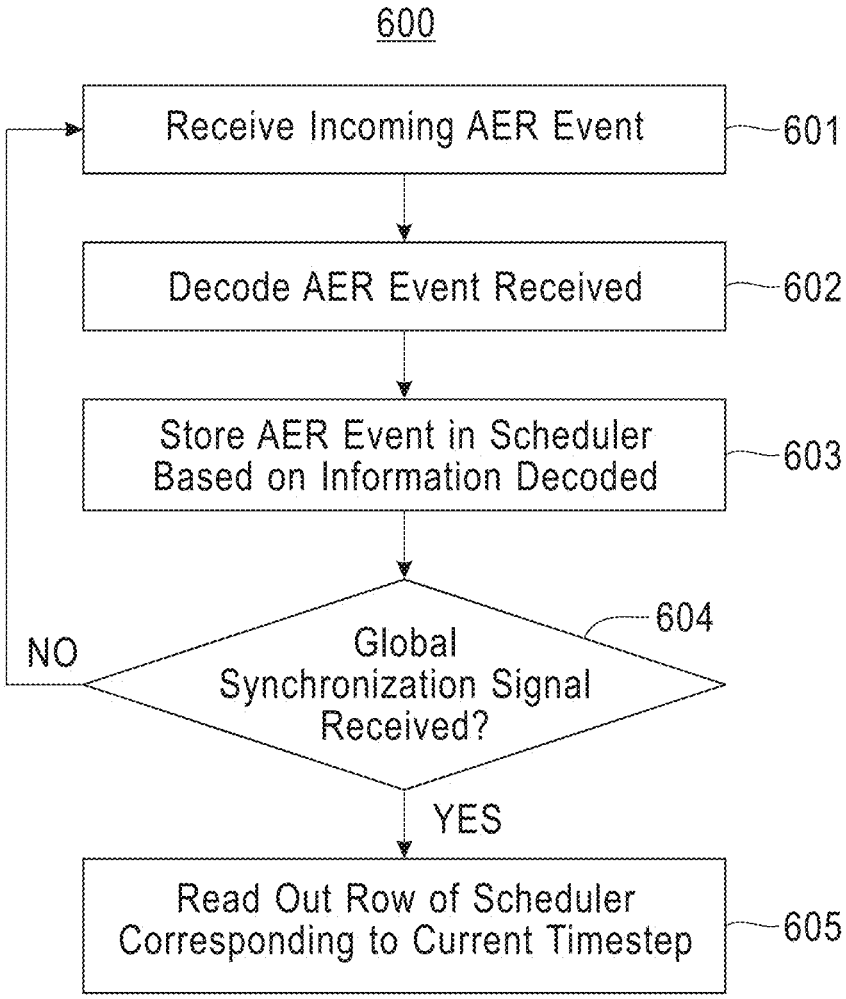

600
Receive Incoming AER Event
601
Decode AER Event Received
602
Store AER Event in Scheduler Based on Information Decoded
603
Global Synchronization Signal Received?
604
NO
YES
Read Out Row of Scheduler Corresponding to Current Timestep
605

FIG. 8

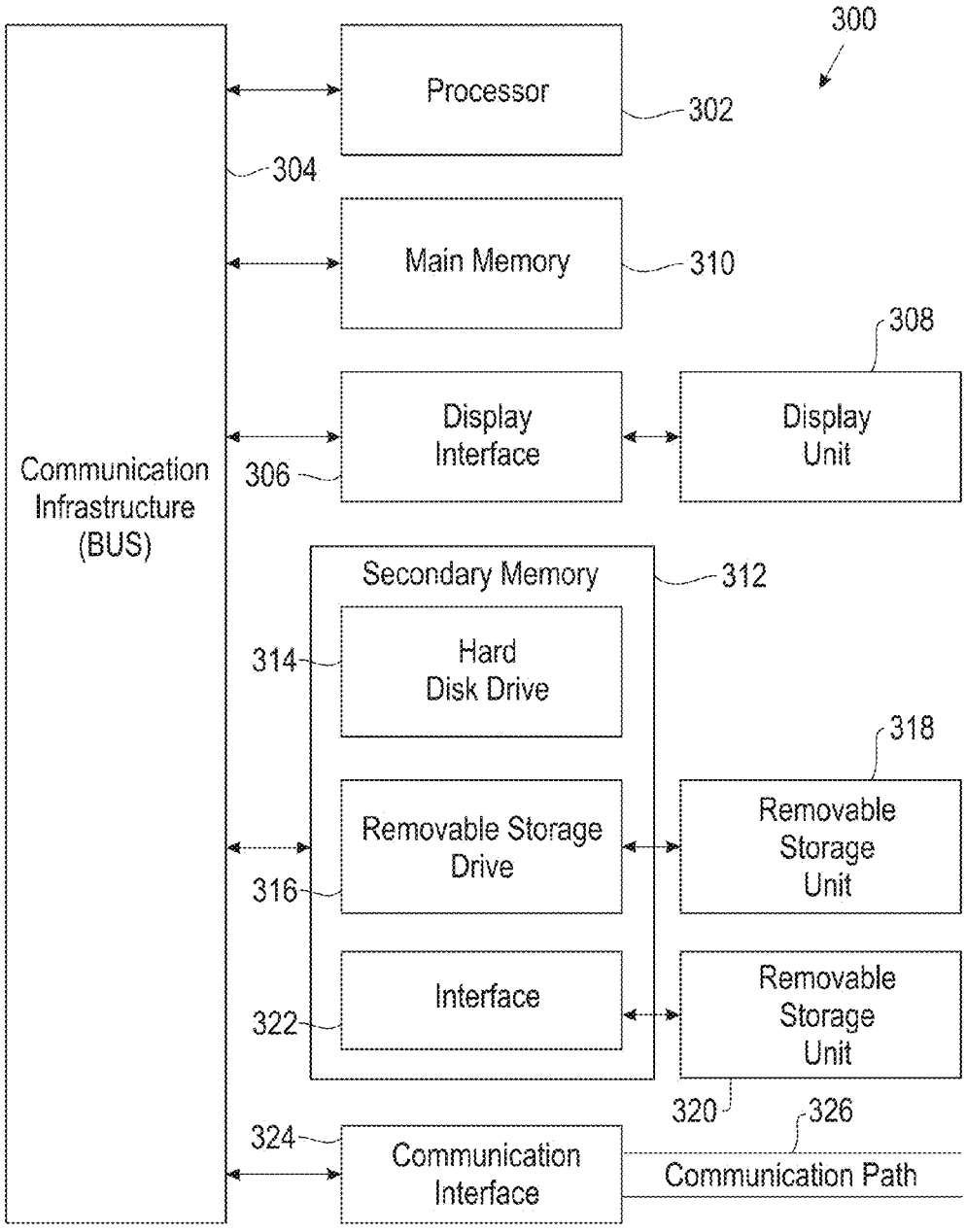

300
Processor
302
304
Main Memory
310
308
Display Interface
Display Unit
306
Communication Infrastructure (BUS)
Secondary Memory
312
314
Hard Disk Drive
318
Removable Storage Drive
Removable Storage Unit
316
Interface
Removable Storage Unit
322
320
326
324
Communication Interface
Communication Path

FIG. 9

# GLOBALLY ASYNCHRONOUS AND LOCALLY SYNCHRONOUS (GALS) NEUROMORPHIC NETWORK

**[0001]** This invention was made with Government support under HR0011-09-C-0002 awarded by Defense Advanced Research Projects Agency (DARPA). The Government has certain rights in this invention.

## BACKGROUND

**[0002]** Embodiments of the invention relate to neuromorphic and synaptronic computation, and in particular, a globally asynchronous and locally synchronous neuromorphic network.

**[0003]** Neuromorphic and synaptronic computation, also referred to as artificial neural networks, are computational systems that permit electronic systems to essentially function in a manner analogous to that of biological brains. Neuromorphic and synaptronic computation do not generally utilize the traditional digital model of manipulating 0s and 1s. Instead, neuromorphic and synaptronic computation create connections between processing elements that are roughly functionally equivalent to neurons of a biological brain. Neuromorphic and synaptronic computation may comprise various electronic circuits that are modeled on biological neurons.

**[0004]** In biological systems, the point of contact between an axon of a neuron and a dendrite on another neuron is called a synapse, and with respect to the synapse, the two neurons are respectively called pre-synaptic and post-synaptic. The essence of our individual experiences is stored in conductance of the synapses. The synaptic conductance changes with time as a function of the relative spike times of pre-synaptic and post-synaptic neurons, as per spike-timing dependent plasticity (STDP). The STDP rule increases the conductance of a synapse if its post-synaptic neuron fires after its pre-synaptic neuron fires, and decreases the conductance of a synapse if the order of the two firings is reversed.

## BRIEF SUMMARY

**[0005]** Embodiments of the invention relate to a globally asynchronous and locally synchronous neuromorphic network. One embodiment comprises generating a synchronization signal that is distributed to a plurality of neural core circuits. In response to the synchronization signal, in at least one core circuit, incoming spike events maintained by said at least one core circuit are processed to generate an outgoing spike event. Spike events are asynchronously communicated between the core circuits via a routing fabric comprising multiple asynchronous routers.

**[0006]** Another embodiment comprises a neural network comprising a plurality of neural core circuits. Each core circuit comprises multiple digital neurons, multiple digital axons, and multiple digital synapses. Each synapse interconnects an axon with a neuron. The neural network further comprises a routing fabric comprising multiple asynchronous routers that interconnect the core circuits. The routing fabric facilitates asynchronous communication of spike events between the core circuits. A plurality of the core circuits process spike events synchronously in response to a synchronization signal.

**[0007]** These and other features, aspects and advantages of the present invention will become understood with reference to the following description, appended claims and accompanying figures.

## BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

**[0008]** FIG. **1** illustrates a neural core circuit, in accordance with an embodiment of the invention;

**[0009]** FIG. **2** is a block diagram illustrating a neural network, in accordance with an embodiment of the invention;

**[0010]** FIG. **3** illustrates a neural core circuit interconnected to a corresponding asynchronous router, in accordance with an embodiment of the invention;

**[0011]** FIG. **4** illustrates inter-core communication in a neural network, in accordance with an embodiment of the invention;

**[0012]** FIG. **5** illustrates neural network comprising at least one multi-core chip structure, in accordance with an embodiment of the invention;

**[0013]** FIG. **6** is a block diagram illustrating a memory array and a processing circuit of a core, in accordance with an embodiment of the invention;

**[0014]** FIG. **7** is a block diagram showing a memory array of a core, in accordance with an embodiment of the invention;

**[0015]** FIG. **8** illustrates a flowchart of an example process for scheduling spikes in a core, in accordance with an embodiment of the invention; and

**[0016]** FIG. **9** is a high level block diagram showing an information processing circuit useful for implementing one embodiment of the invention.

## DETAILED DESCRIPTION

**[0017]** Embodiments of the invention relate to a globally asynchronous and locally synchronous neuromorphic network. One embodiment comprises generating a synchronization signal that is distributed to a plurality of neural core circuits. In response to the synchronization signal, in at least one core circuit, incoming spike events maintained by said at least one core circuit are processed to generate an outgoing spike event. Spike events are asynchronously communicated between the core circuits via a routing fabric comprising multiple asynchronous routers.

**[0018]** Another embodiment comprises a neural network comprising a plurality of neural core circuits. Each core circuit comprises multiple digital neurons, multiple digital axons, and multiple digital synapses. Each synapse interconnects an axon with a neuron. The neural network further comprises a routing fabric comprising multiple asynchronous routers that interconnect the core circuits. The routing fabric facilitates asynchronous communication of spike events between the core circuits. A plurality of the core circuits process spike events synchronously in response to a synchronization signal.

**[0019]** Each core circuit is connected to a corresponding router via a local channel of the corresponding router. Each neuron of each core circuit receives incoming spike events from another neuron of the neural network via the corresponding router of said core circuit. Each neuron of each core circuit sends outgoing spike events to a different neuron of the neural network via the corresponding router of said core circuit.

**[0020]** The routers are interconnected via bus links. Each router further includes a northbound channel for transmitting spike events to a connected router in the northbound direction, a southbound channel for transmitting spike events to a connected router in the southbound direction, an eastbound channel for transmitting spike events to a connected router in

the eastbound direction, and a westbound channel for transmitting spike events to a connected router in the westbound direction.

**[0021]** A controller generates synchronization signals. Each synchronization signal is a discrete time step. Each synchronization signal is distributed to every core circuit simultaneously, such that the core circuits synchronously process incoming spike events upon receiving said synchronization signal.

**[0022]** Each core circuit further comprises a memory array. Each row of the memory array maintains synaptic connectivity information, neuron parameters, and routing information for a corresponding neuron of said core circuit. Each core circuit further comprises a processing circuit for processing and integrating incoming spike events. Each core circuit further comprises a scheduler. The scheduler maintains all incoming spike events for the current time step and future time steps. In one embodiment, the scheduler of each core circuit selectively imposes a delay on delivery of the incoming spike events for deterministic event delivery.

**[0023]** The term digital neuron as used herein represents an framework configured to simulate a biological neuron. An digital neuron creates connections between processing elements that are roughly functionally equivalent to neurons of a biological brain. As such, a neuromorphic and synaptronic computation comprising digital neurons according to embodiments of the invention may include various digital circuits that are modeled on biological neurons. Further, a neuromorphic and synaptronic computation comprising digital neurons according to embodiments of the invention may include various processing elements (including computer simulations) that are modeled on biological neurons. Although certain illustrative embodiments of the invention are described herein using digital neurons comprising digital circuits, the present invention is not limited to digital circuits. A neuromorphic and synaptronic computation according to embodiments of the invention can be implemented as a neuromorphic and synaptronic framework comprising circuitry, and additionally as a computer simulation. Indeed, embodiments of the invention can take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment containing both hardware and software elements.

**[0024]** For a neural network comprising multiple neural core circuits ("cores"), a globally asynchronous routing network for routing spike events between the cores help minimize the consumption of active power in the neural network. Cores in the neural network, however, consume significantly less active power when operating synchronously as compared to when operating asynchronously. Embodiments of a globally asynchronous and locally synchronous (GALS) neuromorphic network are provided. Embodiments of the GALS network disclosed herein can be used to implement a multi-core chip structure comprising asynchronous routers operating at a relatively fast speed, and synchronous cores operating at a relatively low clock speed.

**[0025]** FIG. **1** illustrates a neural core circuit ("core") **100**, in accordance with an embodiment of the invention. The core **100** comprises multiple pre-synaptic axons **15** and multiple post-synaptic neurons **11**. Each neuron **11** has configurable operational parameters. The core **100** further comprises a synaptic crossbar **12** including multiple synapses **31**, multiple rows/axon paths **26**, and multiple columns/dendrite paths **34**.

**[0026]** Each synapse **31** communicates spike events ("spikes") between a pre-synaptic axon **15** and a post-synaptic neuron **11**. Specifically, each synapse **31** is located at cross-point junction between an axon path **26** and a dendrite path **34**, such that a connection between the axon path **26** and the dendrite path **34** is made through said synapse **31**. Each axon **15** is connected to an axon path **26**, such that said axon **15** sends spikes to the connected axon path **26**. Each neuron **11** is connected to a dendrite path **34**, such that said neuron **11** receives spikes from the connected dendrite path **34**.

**[0027]** Each synapse **31** has a synaptic weight. The synaptic weights of the synapses **31** of the core **100** may be represented by a weight matrix W, wherein an element $W_{ji}$ of the matrix W represents a synaptic weight of a synapse **31** located at a row/axon path j and a column/dendrite path i of the crossbar **12**. In one embodiment, the synapses **31** are binary memory devices. Each synapse **31** can have a weight "0" indicating that said synapse **31** is non-conducting, or a weight "1" indicating that said synapse **31** is conducting. A learning rule such as spike-timing dependent plasticity (STDP) may be applied to update the synaptic weights of the synapses **31**.

**[0028]** The core **100** further comprises a spike interface module **90** configured to receive address-event packets (AER event). Each address-event packet received includes a spike encoded as an address (e.g., an address represented as bits) representing a target axon **15**, wherein the spike was generated by a neuron **11** in the same core **100** or a different core **100**. The spike interface module **90** decodes each address-event packet received, and transmits the spike to the target axon **15**.

**[0029]** The spike interface module **90** is also configured to receive spikes generated by the neurons **11** in the core **100**. The spike interface module **90** encodes/encapsulates each spike generated as an address-event packet having the address of a target incoming axon **15**, and sends/routes the address-event packet to a core **100** (e.g., the same core **100** or a different core **100**) containing the target incoming axon **15**.

**[0030]** FIG. **2** is a block diagram illustrating a neural network **250**, in accordance with an embodiment of the invention. The network **250** comprises multiple cores **100**. The network **250** further comprises a routing fabric **230** including multiple asynchronous routers **200** that interconnect the cores **100** of the network **250**. As described in detail later herein, the routers **200** facilitate asynchronous inter-core communication. Each core **100** is interconnected to a corresponding router **200**.

**[0031]** FIG. **3** illustrates a neural core circuit **100** interconnected to a corresponding asynchronous router **200**, in accordance with an embodiment of the invention. Each router **200** has a local channel **202**L connecting said router **200** to a corresponding core **100**. A router **200** may have more than one local channel **202**L, such that the router **100** is connected to multiple cores **100**. Each core **100** receives address-event packets from other cores **100** via a local channel **202**L of a corresponding router **200**. Each core **100** sends address-event packets to other cores **100** via a local channel **202**L of a corresponding router **200**.

**[0032]** Each router **200** further includes a northbound channel **202**N, a southbound channel **202**S, an eastbound channel **202**E, and a westbound channel **202**W for passing along address-event packets to neighboring routers **200** in the northbound, southbound, eastbound, and westbound directions, respectively.

**[0033]** FIG. **4** illustrates inter-core communication in a neural network **260**, in accordance with an embodiment of the invention. The network **260** comprises multiple cores **100**, such as a first core C**1**, a second core C**2**, a third core C**3**, a fourth core C**4**, and a fifth core C**5**. The network **260** further comprises multiple routers **200**, such as a first router R**1**, a second router R**2**, a third router R**3**, a fourth router R**4**, and a fifth router R**5**. A router **200** may be interconnected to four different routers **200** via bus lines **261**. For example, a northbound channel **202**N of the first router R**1** may be interconnected to a southbound channel **202**S of the fifth router R**5**, a southbound channel **202**S of the first router R**1** may be interconnected to a northbound channel **202**N of the third router R**3**, an eastbound channel **202**E of the first router R**1** may be interconnected to a westbound channel **202**W of the fourth router R**4**, and a westbound channel **202**W of the first router R**1** may be interconnected to an eastbound channel **202**E of the second router R**2**.

**[0034]** The routing of address-event packets between cores **100** follow dimension order routing (for example, route north to south, then west to east).

**[0035]** The asynchronous routers **200** minimize the consumption of active power across the neural network, ensuring that bus links **261** are active only when absolutely necessary. Further, asynchronous routers **200** remove the need to distribute a global clock signal across all routers **200** in the network.

**[0036]** FIG. **5** illustrates neural network **280** comprising at least one multi-core circuit chip structure **270**, in accordance with an embodiment of the invention. A neural network may be implemented as a circuit chip structure **270**. The chip structure **270** comprises multiple cores **100** and multiple routers **200** facilitating asynchronous inter-core communication. While the routing of address-event packets between the cores **100** is asynchronous, the cores **100** process spikes synchronously. The asynchronous routers **200** can operate at a faster speed than the synchronous cores **100**.

**[0037]** To facilitate the synchronous operation of the cores **100**, the network **280** further comprises a clock **6** and a controller **9**. In one embodiment, the clock **6** and the controller **9** are off-chip. The clock **6** produces clock signals used by the controller **9** to generate global synchronization signals. Each global synchronization signal generated signifies a global time step. Each global synchronization signal generated is distributed to all cores **100** in the network **280** at the same time. The dynamics of each core **100** is driven by a discrete global time step. Upon receiving a global synchronization signal, each core **100** begins to process the incoming spikes for the global time step signified by the global synchronization signal. As such, the cores **100** begin processing incoming spikes at the same time (i.e., synchronously). The cores **100**, however, can finish updating the operational states of its neurons **11** on different time scales. A core **100** may finish updating the operational states its neurons **11** faster or slower than a different core **100** in the network **280**. Each core **100**, however, must complete updating the operational states of its neurons **11** before the next global synchronization signal is received.

**[0038]** FIG. **6** is a block diagram illustrating a memory array **20** and a processing circuit **50** of a core **100**, in accordance with an embodiment of the invention. In one embodiment, the neurons **11** in the core **100** are multiplexed, such that each neuron **11** in the core **100** shares the same processing circuit **50** with other neurons **11** of the core **100**. The cores **100** may operate synchronously or asynchronously.

**[0039]** The memory array **20** maintains information relating to all neurons **11** of the core **100**. The memory array **20** includes multiples rows **21** (FIG. **7**). Each row **21** of the memory array **20** corresponds to a neuron **11** of the core **100**. Specifically, each row **21** maintains information relating to a corresponding neuron **11** of the core **100**, wherein the information maintained includes synaptic connectivity information ($W_{ji}$), neuron parameters, and routing information of the corresponding neuron **11**. In one embodiment, the neuron parameters maintained for each neuron **11** include the membrane potential variable V of said neuron **11**, the spike threshold Th of said neuron **11**, the leak rate Lk of said neuron **11**, and the weight of all possible axon types (Syn0, Syn1, Syn2). The routing information maintained for each neuron **11** include the fanout F of said neuron **11**, and the routing delay $\Delta T$ of said neuron **11**. The synaptic connectivity information maintained for each neuron **11** includes the synaptic weights for all synapses **31** that a dendrite path **34** of said neuron **11** is connected to.

**[0040]** In one embodiment, the processing circuit **50** comprises an input processing fabric **50**A and a computation fabric **50**B. The processing fabric **50**A comprises a decoder **70**, a scheduler **60**, a control unit **180**, an axon type register **110**, a logical AND operator **120**, and a dot product module **130**. The computation fabric **50**B comprises a first multiplexer **170**, an adder **140**, a second multiplexer **160**, a demultiplexer **150**, a comparator module ("comparator") **90**, and an encoder **80**. The decoder **70** is configured for receiving and decoding address-event packets from a corresponding asynchronous router **200** of the core **100**.

**[0041]** In one embodiment, the core **100** is synchronous and the scheduler **60** of the core **100** bridges the corresponding high-bandwidth asynchronous router **200** to the relatively low-frequency synchronous core **100**. In one example implementation, the scheduler **60** is a dual port memory including rows and columns. Spikes from decoded address-event packets are stored in the scheduler **60**, wherein rows represent future time steps and columns represents axons **15** of the core **100**. The control unit **180** alerts the scheduler **60** when a spike from a decoded address-event packet is ready to be written in the scheduler **60**. The control unit **180** may also be used to compute a delay for a spike queued in the scheduler.

**[0042]** Specifically, a delay is selectively imposed on received spikes before transmitting them to target axons **15**. Each address-event packet further includes a time stamp indicating when a spike encapsulated in said address-event packet was generated. Using the arrival time of the address-event packet at the decoder **70** and the time stamp, the control unit **180** determines how long the spike has spent on route to the decoder **70** as the difference d between the arrival time and the time stamp. If the difference d is less than a predetermined delivery delay (or predetermined total delay) n, the spike is held in the scheduler **60** for a delay period D equal to the difference between n and d, to achieve n timestamps from spike generation to spike delivery, and the scheduler **60** delivers the spike at the end of the delay period.

**[0043]** For example, if a spike requires between 3 and 9 time steps to propagate in a neural network, the scheduler **60** ensures 9 time steps for all spikes. In one example, even if a spike arrives at the decoder **70** within 3 time steps from generation, the scheduler **60** delays transmission of that spike by 6 time steps such that the spike is transmitted from the scheduler **60** at the end of 9 time steps from spike generation.

**[0044]** In another embodiment, the delay is computed at the source of the spike (i.e., the core **100** that the spike originates from) before the spike is sent to the router **200** for delivery to a target axon **15**.

**[0045]** Once a global synchronization signal is received, a vector indicating all active axons **15** in the current time step is read from the scheduler **60** as a row. The entire memory array **20** is then read and written once per time step.

**[0046]** Specifically, for each time step, rows **21** of the array **20** are sequentially read out one at a time using a sequential scanner **30**. The axon type register **110**, the logical AND operator **120**, the dot product module **130**, the first **170**, the adder **140**, the second multiplexer **160**, and the demultiplexer **150** are used to integrate spikes for each neuron **11** of the core **100**.

**[0047]** For each row **21** representing a neuron **11**, the dot product module **130** computes the dot product between the vector of active axons **15** in the current time step and the synaptic connectivity information maintained in said row **21**, i.e., the synaptic weights of all synapses **31** the dendrite **34** of the neuron **11** is connected to (Wj). The operator **120** logically ANDs the resulting dot product with the axon types maintained in the axon type register **110** to generate a queue of axon types. For each axon type in the queue, a membrane potential variable V maintained in said row **21** is updated with the corresponding synaptic weight using an arithmetic logic unit (ALU) such as an adder **140**. After all spikes are integrated, the leak rate Lk maintained in said row **21** is added to the membrane potential variable V. The comparator **90** determines whether the integrated spikes for said row **21** exceeds a threshold parameter Th maintained in said row **21**. If the threshold parameter Th is exceeded, the encoder **80** generates a spike and uses the fanout F and the routing delay of ΔT information maintained in said row **21** to encapsulate the generated spike into an address-event packet. The membrane potential variable V is reset to a preset value before it is written back to said row **21**.

**[0048]** FIG. **7** is a block diagram showing a memory array **20** of a core **100**, in accordance with an embodiment of the invention. As stated above, the memory array **20** maintains information relating to all neurons **11** of the core **100**. The memory array **20** includes multiples rows **21**, wherein each row **21** of the memory array **20** corresponds to a neuron **11** of the core **100**. Specifically, each row **21** maintains information relating to a corresponding neuron **11** of the core **100**, wherein the information maintained includes synaptic connectivity information, neuron parameters, and routing information of the corresponding neuron **11**.

**[0049]** FIG. **8** illustrates a flowchart of an example process **600** for scheduling spikes in a core **100**, in accordance with an embodiment of the invention. In process block **601**, the decoder **70** receives address-event packets (AER event). In process block **602**, the decoder **70** decodes each address-event packet received. In process block **603**, decoded address-event packet (i.e., spike) is stored in the scheduler **60** based on the information decoded. In process block **604**, the core **100** determines whether a global synchronization signal was received. If the signal has been received, proceed to process block **605** where a row in the scheduler corresponding to the current time step is read out to trigger the integrating of spikes. If no signal was received, return to process block **601**.

**[0050]** FIG. **9** is a high level block diagram showing an information processing system **300** useful for implementing one embodiment of the invention. The computer system includes one or more processors, such as processor **302**. The processor **302** is connected to a communication infrastructure **304** (e.g., a communications bus, cross-over bar, or network).

**[0051]** The computer system can include a display interface **306** that forwards graphics, text, and other data from the communication infrastructure **304** (or from a frame buffer not shown) for display on a display unit **308**. The computer system also includes a main memory **310**, preferably random access memory (RAM), and may also include a secondary memory **312**. The secondary memory **312** may include, for example, a hard disk drive **314** and/or a removable storage drive **316**, representing, for example, a floppy disk drive, a magnetic tape drive, or an optical disk drive. The removable storage drive **316** reads from and/or writes to a removable storage unit **318** in a manner well known to those having ordinary skill in the art. Removable storage unit **318** represents, for example, a floppy disk, a compact disc, a magnetic tape, or an optical disk, etc. which is read by and written to by removable storage drive **316**. As will be appreciated, the removable storage unit **318** includes a computer readable medium having stored therein computer software and/or data.

**[0052]** In alternative embodiments, the secondary memory **312** may include other similar means for allowing computer programs or other instructions to be loaded into the computer system. Such means may include, for example, a removable storage unit **320** and an interface **322**. Examples of such means may include a program package and package interface (such as that found in video game devices), a removable memory chip (such as an EPROM, or PROM) and associated socket, and other removable storage units **320** and interfaces **322**, which allows software and data to be transferred from the removable storage unit **320** to the computer system.

**[0053]** The computer system may also include a communication interface **324**. Communication interface **324** allows software and data to be transferred between the computer system and external devices. Examples of communication interface **324** may include a modem, a network interface (such as an Ethernet card), a communication port, or a PCMCIA slot and card, etc. Software and data transferred via communication interface **324** are in the form of signals which may be, for example, electronic, electromagnetic, optical, or other signals capable of being received by communication interface **324**. These signals are provided to communication interface **324** via a communication path (i.e., channel) **326**. This communication path **326** carries signals and may be implemented using wire or cable, fiber optics, a phone line, a cellular phone link, an RF link, and/or other communication channels.

**[0054]** In this document, the terms "computer program medium," "computer usable medium," and "computer readable medium" are used to generally refer to media such as main memory **310** and secondary memory **312**, removable storage drive **316**, and a hard disk installed in hard disk drive **314**.

**[0055]** Computer programs (also called computer control logic) are stored in main memory **310** and/or secondary memory **312**. Computer programs may also be received via communication interface **324**. Such computer programs, when run, enable the computer system to perform the features of the present invention as discussed herein. In particular, the computer programs, when run, enable the processor **302** to perform the features of the computer system. Accordingly, such computer programs represent controllers of the computer system.

**[0056]** From the above description, it can be seen that the present invention provides a system, computer program product, and method for implementing the embodiments of the invention. The present invention further provides a non-transitory computer-useable storage medium for hierarchical routing and two-way information flow with structural plasticity in neural networks. The non-transitory computer-useable storage medium has a computer-readable program, wherein the program upon being processed on a computer causes the computer to implement the steps of the present invention according to the embodiments described herein. References in the claims to an element in the singular is not intended to mean "one and only" unless explicitly so stated, but rather "one or more." All structural and functional equivalents to the elements of the above-described exemplary embodiment that are currently known or later come to be known to those of ordinary skill in the art are intended to be encompassed by the present claims. No claim element herein is to be construed under the provisions of 35 U.S.C. section 112, sixth paragraph, unless the element is expressly recited using the phrase "means for" or "step for."

**[0057]** The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of the invention. As used herein, the singular forms "a", "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises" and/or "comprising," when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

**[0058]** The corresponding structures, materials, acts, and equivalents of all means or step plus function elements in the claims below are intended to include any structure, material, or act for performing the function in combination with other claimed elements as specifically claimed. The description of the present invention has been presented for purposes of illustration and description, but is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the invention. The embodiment was chosen and described in order to best explain the principles of the invention and the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.

What is claimed is:

**1**. A method, comprising:
generating a synchronization signal that is distributed to a plurality of neural core circuits;
in response to the synchronization signal, in at least one core circuit, processing incoming spike events maintained by said at least one core circuit to generate an outgoing spike event; and
asynchronously communicating spike events between the core circuits via a routing fabric comprising multiple asynchronous routers.

**2**. The method of claim **1**, further comprising:
for each core circuit:
connecting said core circuit to a corresponding router;
receiving incoming spike events via the corresponding router of said core circuit; and
sending outgoing spike events via the corresponding router of said core circuit.

**3**. The method of claim **2**, further comprising:
interconnecting the routers; and
for each router:
transmitting spike events to a connected router in the northbound direction;
transmitting spike events to a connected router in the southbound direction;
transmitting spike events to a connected router in the eastbound direction; and
transmitting spike events to a connected router in the westbound direction.

**4**. The method of claim **3**, further comprising:
generating synchronization signals, wherein each synchronization signal is a discrete time step; and
distributing each synchronization signal to every core circuit simultaneously, such that the core circuits synchronously process incoming spike events upon receiving said synchronization signal.

**5**. The method of claim **4**, further comprising:
for each core circuit, maintaining synaptic connectivity information, neuron parameters, and routing information for neurons.

**6**. The method of claim **5**, further comprising:
for each core circuit, processing and integrating incoming spike events.

**7**. The method of claim **6**, further comprising:
for each core circuit, maintaining all incoming spike events for the current time step and future time steps.

**8**. The method of claim **7**, further comprising:
selectively imposing a delay on delivery of the incoming spike events for deterministic event delivery.

**9**. A neural network, comprising:
a plurality of neural core circuits, wherein each core circuit comprises:
multiple digital neurons;
multiple digital axons; and
multiple digital synapses, wherein each synapse interconnects an axon with a neuron; and
a routing fabric comprising multiple asynchronous routers interconnecting said core circuits, wherein the routing fabric facilitates asynchronous communication of spike events between said core circuits;
wherein a plurality of said core circuits process spike events synchronously in response to a synchronization signal.

**10**. The neural network of claim **9**, wherein:
for each core circuit:
said core circuit is connected to a corresponding router via a local channel of the corresponding router;
each neuron of said core circuit receives incoming spike events from another neuron of the neural network via the corresponding router of said core circuit; and
each neuron of said core circuit sends outgoing spike events to a different neuron of the neural network via the corresponding router of said core circuit.

**11**. The neural network of claim **10**, wherein:
the routers are interconnected via bus links; and
each router further includes:
a northbound channel for transmitting spike events to a connected router in the northbound direction;
a southbound channel for transmitting spike events to a connected router in the southbound direction;

an eastbound channel for transmitting spike events to a connected router in the eastbound direction; and

a westbound channel for transmitting spike events to a connected router in the westbound direction.

**12**. The neural network of claim **11**, further comprising:

a controller for generating synchronization signals, wherein each synchronization signal is a discrete time step;

wherein each synchronization signal is distributed to every core circuit simultaneously, such that the core circuits synchronously process incoming spike events upon receiving said synchronization signal.

**13**. The neural network of claim **12**, wherein each core circuit further comprises:

a memory array, wherein each row of the memory array maintains synaptic connectivity information, neuron parameters, and routing information for a corresponding neuron of said core circuit.

**14**. The neural network of claim **13**, wherein:

each core circuit further comprises a processing circuit for processing and integrating incoming spike events.

**15**. The neural network of claim **14**, wherein:

each core circuit further comprises a scheduler, wherein the scheduler maintains all incoming spike events for the current time step and future time steps.

**16**. The neural network of claim **15**, wherein:

the scheduler of each core circuit selectively imposes a delay on delivery of the incoming spike events for deterministic event delivery.

**17**. A non-transitory computer-useable storage medium for a neural network comprising a plurality of neural core circuits, wherein the program upon being processed on a computer causes the computer to:

generate a synchronization signal that is distributed to said plurality of neural core circuits;

in response to the synchronization signal, in at least one core circuit, process incoming spike events maintained by said at least one core circuit to generate an outgoing spike event; and

asynchronously communicate spike events between the core circuits via a routing fabric comprising multiple asynchronous routers.

**18**. The program of claim **17**, further causing the computer to:

for each core circuit:

connect said core circuit to a corresponding router;

receive incoming spike events via the corresponding router of said core circuit; and

send outgoing spike events via the corresponding router of said core circuit.

**19**. The program of claim **18**, further causing the computer to:

interconnect the routers; and

for each router:

transmit spike events to a connected router in the northbound direction;

transmit spike events to a connected router in the southbound direction;

transmit spike events to a connected router in the eastbound direction; and

transmit spike events to a connected router in the westbound direction.

**20**. The program of claim **19**, further causing the computer to:

generate synchronization signals, wherein each synchronization signal is a discrete time step;

distribute each synchronization signal to every core circuit simultaneously, such that the core circuits synchronously process incoming spike events upon receiving said synchronization signal;

for each core circuit, process and integrate incoming spike events;

for each core circuit, maintain all incoming spike events for the current time step and future time steps; and

selectively impose a delay on delivery of the incoming spike events for deterministic event delivery.

* * * * *