Hierarchical Address Event Routing for Reconfigurable Large-Scale Neuromorphic Systems

Jongkil Park, *Member, IEEE*, Theodore Yu, *Member, IEEE*, Siddharth Joshi, *Student Member, IEEE*, Christoph Maier, *Member, IEEE*, and Gert Cauwenberghs, *Fellow, IEEE*

***Abstract*— We present a hierarchical address-event routing (HiAER) architecture for scalable communication of neural and synaptic spike events between neuromorphic processors, implemented with five Xilinx Spartan-6 field-programmable gate arrays and four custom analog neuromophic integrated circuits serving 262k neurons and 262M synapses. The architecture extends the single-bus address-event representation protocol to a hierarchy of multiple nested buses, routing events across increasing scales of spatial distance. The HiAER protocol provides individually programmable axonal delay in addition to strength for each synapse, lending itself toward biologically plausible neural network architectures, and scales across a range of hierarchies suitable for multichip and multiboard systems in reconfigurable large-scale neuromorphic systems. We show approximately linear scaling of net global synaptic event throughput with number of routing nodes in the network, at 3.6×10^7 synaptic events per second per 16k-neuron node in the hierarchy.**

***Index Terms*— Address-event representation (AER), dual graph, field-programmable gate array (FPGA), integrate-and-fire neurons, network partitioning, spiking neuromorphic systems.**

I. Introduction

SYNTHESIS of very large-scale silicon models of biological neural networks approaching the computational complexity and cognitive function of the human brain has long posed a grand challenge in neuromorphic system engineering and has been met with strengthened effort and enthusiasm in recent years. The challenge is not only one of massive scales in biological neural networks, with billions of neurons and trillions of synapses, but also in supporting flexible mechanisms to dynamically configure synaptic connectivity, driven by neural activity, across all scales.

In biological neural networks, action potentials (or "spikes") traveling along axons carry neural information over long distances projecting to large numbers of other neurons distributed over varying spatial scales [1], [2]. Naturally, the question arises whether similar principles of distributed communication with spike "events" can be employed for efficient and scalable computation across large networks of silicon neural systems. The address-event representation (AER) protocol was introduced as an efficient means for point-to-point (P2P) communication of neural spike events between arrays of neurons, in which addresses of neurons are asynchronously communicated over a shared digital bus, whenever they spike [3]–[7]. The AER communication protocol lends itself directly to implementing synaptic connectivity in a dynamically reconfigurable manner by routing address events through synaptic routing tables (SRTs) in memory, which map presynaptic source addresses to postsynaptic destination addresses along with synaptic parameters [8]–[14].

The virtual wiring of AER synaptic connections between neurons, residing in programmable routing tables in memory, offers the flexibility to connect, in principle, any pair of neurons. Such is not generally possible with hard-wired synaptic array realizations, except for fully connected and, hence, relatively small networks of neurons. Furthermore, AER synaptic connections can be freely created, updated, and pruned as needed. In particular, Hebbian-like spike-timing-dependent plasticity (STDP) and other forms of adaptive updates in synaptic strength and connectivity based on spiking neural activity can be conveniently implemented in the address-event domain, by monitoring relative timing of presynaptic and postsynaptic spike events entering and exiting the SRTs [15] or through more general forms of activity-dependent reprogramming of AER connectivity [16]. From a systems perspective, AER synaptic connectivity further permits the multichip integration of spike event-based sensory and neural processing systems such as silicon retinae [17]–[20], silicon cochleae [21], [22], and systems comprising them for various applications such as object recognition [23], accident detection [24], word recognition [25], texture recognition [26], sequence recognition [27], among

Manuscript received January 22, 2016; revised May 10, 2016; accepted May 11, 2016. Date of publication January 22, 2016; date of current version September 15, 2017. This work was supported by NSF under Grant EFRI-1137279, NSF under Grant CCF-1317407, ONR under Grant N00014-13-1-0205, DARPA under Grant HR0011-10-C-0032 (NeoVision2), Evolved Machines, Texas Instruments, and Qualcomm.

J. Park was with the Department of Electrical and Computer Engineering, Jacobs School of Engineering, Institute of Neural Computation, University of California at San Diego, La Jolla, CA 92093 USA. He is now with the Electronics and Telecommunications Research Institute, Daejeon 34129, South Korea (e-mail: jongkil.ucsd@gmail.com).

T. Yu is with Texas Instruments, Santa Clara, CA 95051 USA (e-mail: theodore.yu@ti.com).

S. Joshi is with the Department of Electrical and Computer Engineering, Jacobs School of Engineering, Institute of Neural Computation, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: sijoshi@ucsd.edu).

C. Maier is with the Institute of Neural Computation, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: christoph.maier@ieee.org).

G. Cauwenberghs is with the Department of Bioengineering, Jacobs School of Engineering, Institute of Neural Computation, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: gert@ucsd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2016.2572164

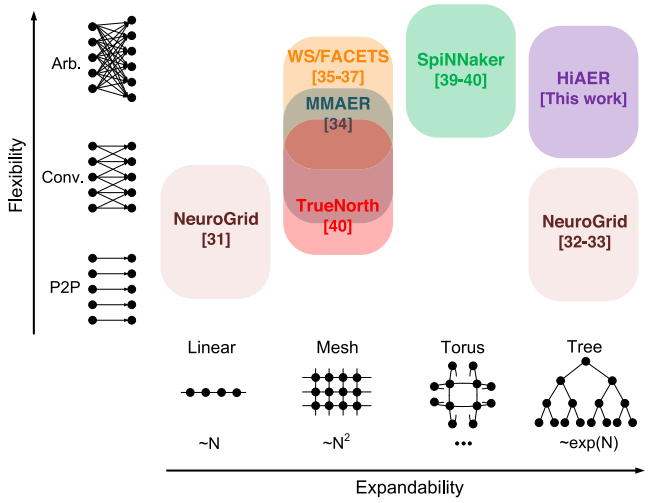


Fig. 1. Comparison of synaptic connection topologies for several recent large-scale event-driven neuromorphic systems and the proposed hierarchical address-event routing (HiAER), represented diagrammatically in two characteristic dimensions of connectivity: expandability (or extent of global reach) and flexibility (or degrees of freedom in configurability). Expandability, measured as distance traveled across the network for a given number of hops N, varies from linear and polynomial in N for linear and mesh grid topologies, to exponential in N for hierarchical tree-based topologies. Flexibility, measured as the number of target destinations reachable from any source in the network, ranges from unity for point-to-point (P2P) connectivity and constant for convolutional kernel (Conv.) connectivity to the entire network for arbitrary (Arb.) connectivity. MMAER: multicasting mesh AER; WS: wafer scale.

many others.

One intrinsic challenge of AER synaptic connectivity for large-scale neuromorphic systems is the limitation in bandwidth of the digital bus shared among all time-multiplexed synapses. Advances in high-speed serial communication links using low-voltage differential signaling [28]–[30] support bus bandwidths up to 100 Mevents/s at 16 bits per event. With peak neural firing rates up to 100 Hz, the number of synapses shared per AER bus is thus limited to millions, or thousands of neurons for a typical 100–10 000 fan-out.

To mitigate this AER bandwidth limitation for very large-scale neuromorphic systems, several solutions have been proposed to extend the standard single-bus AER architecture using grid (or mesh) and tree interchip interconnect topologies, as diagrammatically shown in Fig. 1. Neurogrid [31]–[33] employs linear grid and tree topologies in which global address events are broadcasted across chips through multiple P2P AER buses, leading to improvements in overall P2P communication channel bandwidth although with limited flexibility in synaptic connectivity dominated by intralayer diffusive convolution kernels and interlayer translation-invariant P2P maps. Two-dimensional (2-D) grid topologies are also pursued in systems with differing address-event mapping schemes. Multicasting mesh AER [34] stores router-to-router connectivity rather than neuron-to-neuron connectivity in local routing tables, reducing table size by implementing address translation for local neural event routing. Wafer-scale (WS) integration of 2-D AER grid multichip neuromorphic systems [35], [36] further mitigates communication cost in chip-to-chip interconnectivity issue by connecting 450 chips on a single wafer through metal postprocessing. Interwafer communication extends such systems to another larger level for longer range interconnects [37]. IBM SyNAPSE TrueNorth [38] integrates a 2-D mesh of 64×64 cores each with 256 crossbar connected neurons on a single 1M-neuron, 256M-synapse chip, and further extends the 2-D mesh topology with 4×4 tiles of TrueNorth chips on a printed circuit board (PCB). Although TrueNorth supports full connectivity within each core, a limit of one external synaptic connection per neuron over a limited distance across cores on the mesh is supported, and hence, proportionally larger numbers of neurons, replicating the source neuron across the 2-D mesh, need to be expended to realize larger fan-out over larger distances [38]. SpiNNaker [39], [40] assigns unique global addresses enabling flexible direct neuron-to-neuron access across chips by implementing larger local routing tables. SpiNNaker also offers improved expandability by spanning a torus rather than open mesh connection topology.

This paper focuses on hierarchical address-event routing (HiAER) as a multiscale tree-based extension on AER synaptic routing for dynamically reconfigurable long-range synaptic connectivity in neuromorphic computing systems, combining the advantages of high flexibility and high expandability (see Fig. 1) by embedding random-access addressing at all levels of scale in the tree-based connection hierarchy. The HiAER synaptic event routing infrastructure serves as a communication backbone to integrate-and-fire array transceivers (IFAT) [9], [12], [14] and other event-driven spiking neural network hardware systems (see [8], [10], [11], [13]), the details of which are beyond the scope of this paper. Using results from queuing theory, we previously showed that HiAER offers scalable synaptic event throughput, independent of neural network size, for given synaptic fan-out and nominal axonal delay, and without restriction on spatial range of synaptic connectivity [41]. Another distinguishing feature of HiAER is that synaptic connections code not only programmable synaptic strength (probability of presynaptic release and postsynaptic conductance) but also programmable axonal delay, implemented in the timing of events routed from source to destination. In Section II, we describe the fundamentals of HiAER and its edge-vertex-dual-like mapping of flat arbitrary network topology onto hierarchically partitioned multiscale local networks, with relay neurons (RNs) interfacing between consecutive scales in the hierarchy. In Section III, we describe implementation of the HiAER interbus routing node, including memory interfaces to local routing tables and priority queue (PQ) for timed event registration and delivery. Section IV validates the scaling properties of HiAER nominal throughput and latency in a FPGA-based experimental platform on a custom PCB realizing two levels of HiAER each with a branching factor of 4. Nearly fourfold improvements both in throughput and latency are demonstrated for HiAER across four routing nodes, in comparison to single-node AER as previously reported in [42]. Finally, Section V concludes with a discussion on

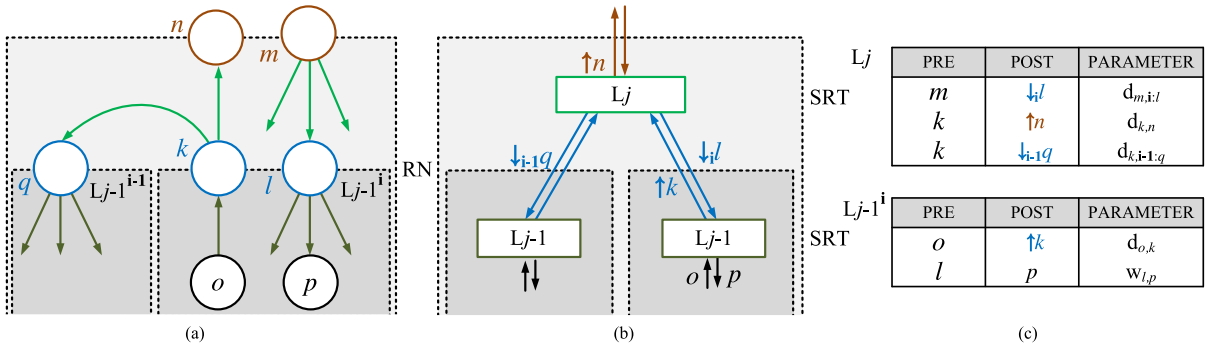


Fig. 2. (a) Hierarchical neural network with ascending and descending neural projections. Physical neurons sourcing and receiving spike events are denoted by o and p, and relay neurons (RNs) interfacing across hierarchical partitions to locally represent source neurons are denoted by k, l, m, n, and q. Indices j and $j-1$ represent levels in the hierarchy, while boldface indices $\mathbf{i}$ and $\mathbf{i}-\mathbf{1}$ represent individual blocks within one level in the hierarchy. (b) Hierarchical routing network as the edge-vertex-dual of the hierarchical neural network in (a). (c) Corresponding entries within the synaptic routing table (SRT). See Fig. 3 for a concrete example with proof-of-concept partitioning and mapping of a target network into the hierarchical and edge-vertex-dual forms.

HiAER advantages, limitations, and extensions.

II. Hierarchical Address-Event Routing

A. Global Synaptic Connectivity and Axonal Spike Transmission

The efficient and scalable emulation of biological networks with VLSI learning systems requires abstraction of various biological details to ease the implementation and analysis of such networks. In biological neural networks, neural spikes, which are electrical pulses, originate in the axon hillock of a neuron, and propagate through the axon of the neuron via synapses of varying coupling strengths to one or several dendrites of receiving neurons, and within the receiving neurons to their cell bodies (soma), where a sufficiently high total amount of excitation by spikes gives rise to another spike. In the case of VLSI implementation, where artificial neural arrays emulate such spiking neurons, we can uniquely identify each neuron by some address A, e.g., its (x, y) coordinates within that array. Similarly, a combination of various synaptic properties can be grouped together and represented by the connection strength (such as postsynaptic conductance or presynaptic release probability) w, and axonal transmission delay d for that synapse. Thus, the quadruplet $(A_{\text{pre}}, A_{\text{post}}, w, d)$ encodes a synaptic connection: it specifies a synaptic event with weight w to be delivered to postsynaptic destination neuron A_{post} a time interval d after every spike generated by presynaptic source neuron A_{pre}. Strictly, the triplet (A_{post}, w, d) is sufficient to route the synaptic event to its postsynaptic destination independent of its source. However, source encoding of multiple synaptic events originating from the same presynaptic address A_{pre} allows more efficient routing when synaptic fan-out is large and distributed in space, as shown in the following.

B. Hierarchical Neural Network Topology

Although AER is capable of interconnecting neurons in a reconfigurable manner [8]–[14], the limited bandwidth of single-bus AER restricts the network size to thousands of neurons. Grid-based [31], [34], [39], [40] and tree-based [32], [33], [35]–[37], [41] extensions to AER have aimed at extending the bandwidth and spatial range of synaptic connectivity across multichip neural arrays in a scalable and efficient manner. Here, we focus on HiAER as a multiscale tree-based extension on single-bus AER to offer scalable synaptic event throughput without restriction on spatial range of synaptic connectivity [41]. Neurons communicate synaptic events, within and across neural arrays, over dedicated serial communication links. Depending on the destination, a spike address event may pass through several routing nodes, arranged in a treelike configuration, on its way from the presynaptic neuron to the postsynaptic neuron. At each routing node, an incoming address event may trigger multiple outgoing address events, enabling bundling of events to spatially colocated neurons. We term these intermediate routing nodes *relay neurons* (RNs). Each RN represents a single source neuron A_{pre} locally at one level of the hierarchy, and serves to pass on an incoming source event to RNs at higher and/or lower levels in the hierarchy. Ascending RN projections are one-to-one: a source event propagates to single RNs at ascending levels in the tree hierarchy. However, descending RN projections are multifold: an RN event may descend to RNs for each branch at that level in the tree, and an RN at the lowest level may fan-out to all local physical neurons. Hence, HiAER is capable of reaching large numbers of postsynaptic destinations A_{post} spanning very wide spatial range: the distance traveled is exponential in the number of relay hops down the tree hierarchy. This partitioning and grouping of messages ensure more efficient use of memory and bandwidth. Furthermore, HiAER implements axonal delays in an efficient manner: the total transmission delay d for a synapse is partitioned across various routing nodes in a hierarchical manner, such that less frequented relay nodes higher up in the tree execute the longer delays, minimizing overall queue occupancy in timed event routing and delivery (see Section II-D).

Fig. 2 defines key concepts and notations in hierarchical network topology and event routing, before we address the problem of partitioning an arbitrary network into a hierarchical tree-structured form. An example segment of a partitioned spiking neural network with directed synaptic connections between physical and RN nodes in Fig. 2(a) is transformed

into an equivalent representation in Fig. 2(b), with the corresponding SRT entries shown in Fig. 2(c). The neural network segment in Fig. 2(a) also indicates the hierarchy of connections across levels in the partitioning, i.e., either between RNs on the same level (k to q), from a lower to a higher level (o to k and k to n), or going from a higher to a lower level (m to l and l to p). Within the context of our architecture in Fig. 2(b), we collate all connections that belong to a neuron into SRT entries, thus creating a single entity to represent all its synapses. There is an SRT entry for each unique neuron or RN within the hierarchy, identifying what is communicated by each event. Thus, each link represents a unique neuron or RN, while all synaptic connectivity information resides within SRT nodes. Topologically, this transform is akin to the edge-to-vertex dual of a graph, where edges transform to vertices and vice versa. The SRT entries of the dual transformed network are shown in Fig. 2(c). Only connections presynaptic to the neural array specify weight information, while those entries presynaptic to RNs specify axonal delay information. SRT entries also specify directional information shown by $\downarrow$ for entries descending the hierarchy and $\uparrow$ for those ascending.

Note that a source neuron and each of its relay copies through the hierarchy are given distinct local addresses, however, each uniquely identifies the source within its scope at its level of the hierarchy, whereby the SRTs linking consecutive levels consistently map the local addresses for each relay copy.

C. Tree-Based Partitioning

The above concepts extend to any level of hierarchy. A simple methodology for converting an arbitrary network into a tree-based hierarchical form is to proceed top–down as follows: segment the network (according to topography or connectivity) into partitions, one for each branching factor of the tree; insert RNs in the network at the partition boundaries for each neuron that projects across, rerouting connections emanating from that neuron through the RNs interfacing between the partitions; and repeat the process within each partition at the next (lower) level of the hierarchy. Fig. 3 shows an extended example of conversion from a 16-neuron network [Fig. 3(a)] through a partitioned network with three levels of hierarchy [Fig. 3(b)] to its dual representation of hierarchical routing with SRT entries [Fig. 3(c)]. Arrows between routing nodes in Fig. 3(c) represent the direction of AER communication, where dark-shaded arrows show active links communicating neuron spike events through each level of the HiAER hierarchy.

Note that once a synaptic event traveling along a chain of RNs descends down the hierarchy, it does not return up the hierarchy and continues its descent until it reaches the destination at the leaf node. This safeguards against any looping in the RN network, so that the event routing communication is guaranteed free from dead-lock conditions, a critical consideration in address-event systems.

D. Distributed Axonal Delay

Axonal delay in action potential propagation, such as along neuronal fiber bundles in the white matter of cortex, plays an integral role in the functioning of the central nervous system [1], and has been the basis for models of neural computation based on coincidence in delay-based matched filtering of spike events [43]. A distinguishing feature of HiAER is that it explicitly accounts for relative timing in event transmission and delivery, providing a programmable axonal delay d for each individual synaptic connection. Such explicit delay in the AER path provides a compact event-based digital alternative to previously proposed means to implement axonal delay in neuromorphic hardware, e.g., [44]–[46].

Axonal delays depending on a variety of biological factors may range between tens of microseconds to hundreds of milliseconds [1], [2]. In order to cover such wide range of time scales, an architecture with high temporal dynamic range is required. HiAER approaches this problem by partitioning delays in hierarchical fashion, in tandem with the partitioning of the network. Implemented axonal delays are distributed across HiAER routing nodes for each of the RNs in the hierarchy. The net axonal delay d is thus the sum of incremental delays for all RNs in the path from presynaptic source to postsynaptic destination. Incremental delays are implemented by incrementing the deliver-at time stamp of outgoing events, and by priority-queuing incoming events, not releasing them until the deliver-at time is reached, as elaborated in Section III. One challenge with the implementation of delayed event queuing is that total queue occupancy grows linearly not only in event rate, but also in average delay, according to Little's Law [47]. Hierarchical partitioning of delays in HiAER allows to optimize for minimum overall queue occupancy by assigning largest incremental delays to RNs at highest levels, and proceeding with remaining incremental delay assignments down the hierarchy in greedy fashion, leaving smallest incremental delays at the lowest level (HiAER Level 1) where events fan-out in greatest numbers to the local IFAT [41]. Such partitioning of axonal delay is consistent with the qualitative observation that longer axonal fiber bundles that interconnect more distant brain regions carry greater delays [2].

III. Hardware Implementation of HiAER

A. Routing Node System Architecture

The HiAER router, as shown in Fig. 4, arbitrates between input events, time stamps a selected event, then accesses its entry in the SRT and places the entry on the bus *en route* to its destination. Leaf nodes in the tree, at Level 1, route local spike events to and from the IFAT, as shown in Fig. 4(a). Two event input paths feed into the Level 1 HiAER node: up from the local IFAT, and down from the $L1$ bus. Events are encoded differently depending on their source: events originating from IFAT contain the address of the neuron that spiked, whereas events from the $L1$ bus carry a deliver-at time stamp and are kept in the PQ until that time is reached. A more detailed explanation of the PQ is provided in Section III-C. Time stamping is performed upon the arrival of the arbitrated input event, loading the instantaneous global timer value onto

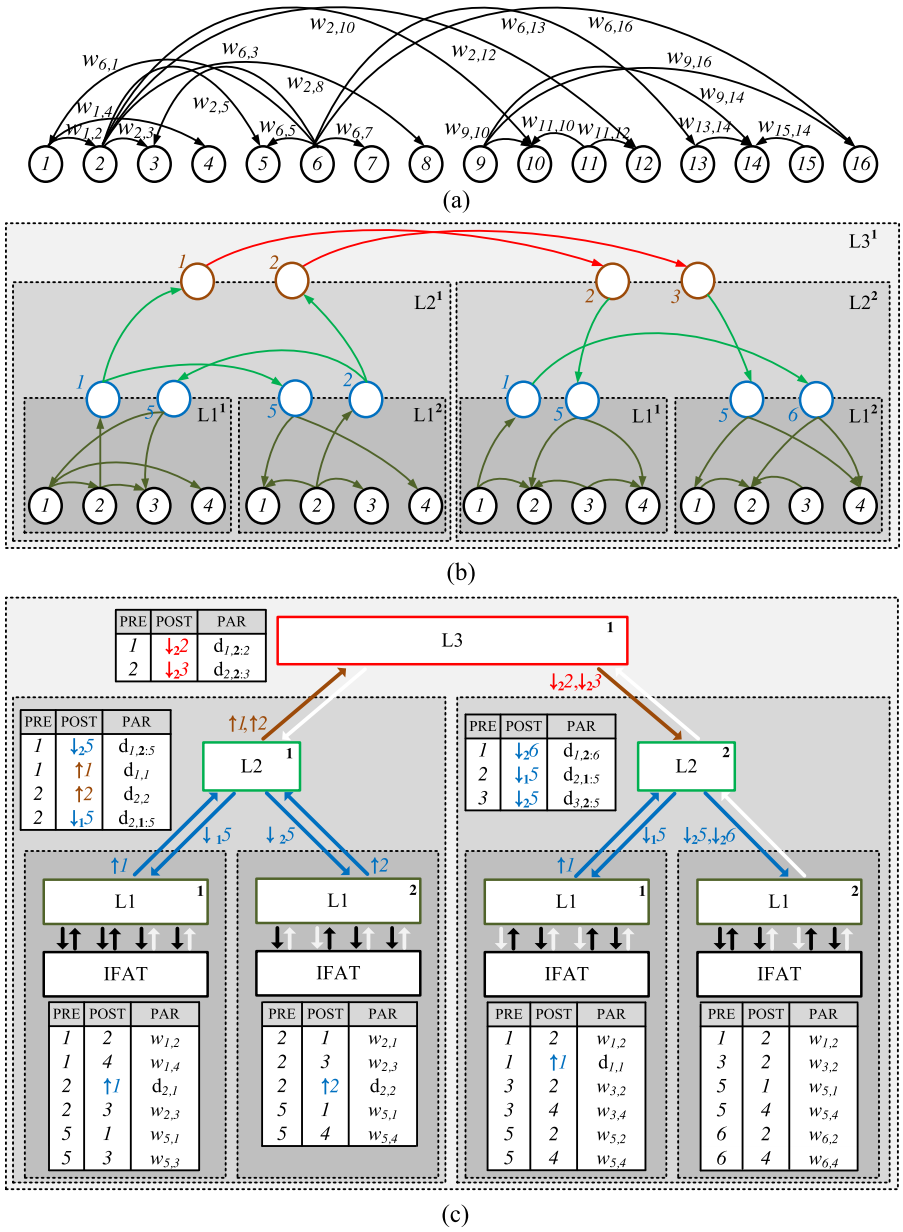


Fig. 3. (a) Example network with 16 neurons and weighted synaptic connections. (b) Example partitioning into hierarchical neural network with ascending and descending projections through inserted RNs. (c) Corresponding edge-vertex-dual HiAER implementation with SRTs at each level in the hierarchy.

the time stamp register. A local register copy of the global timer, synchronized across all HiAER nodes, is globally reset and periodically incremented (every 1 ms). The event enters the SRT in third-generation double data rate synchronous dynamic random-access memory (DDR3 SDRAM) through the CMD buffer and memory controller, returning a sequence of output events through the output buffer. The most significant bit (MSB) of the output event determines whether it is routed upward to the $L1$ bus, or downward to the IFAT. The content of the SRT and the format of output events are described in Section III-B.

Event routing at higher levels in the hierarchy proceeds in similar fashion, as shown in Fig. 4(b) for the Level n HiAER node routing between $Ln-1$ and Ln buses. Differences with Level 1 routing arise due to need for a PQ at both input paths from the higher (Ln) and lower level ($Ln-1$) of the hierarchy, enabling fine multilevel distributed control over the axonal delay parameter d. The format for entries in and events through the SRT is also different, as elaborated in the following.

B. Synaptic Routing Table

SRTs specify all synaptic connectivity, leading a synaptic event from its source to its final destination through all levels of the hierarchy. In addition, SRTs code the necessary information to distribute the axonal delay d across the path, and to deliver synaptic strength w (presynaptic release probability and postsynaptic conductance) on the final path segment at Level 1 to the destination in the local IFAT.

SRTs are implemented using two 2-Gb DDR3 DRAM (Micron MT41J128M16) for every Spartan-6 XC6SLX45T Xilinx FPGA, each DRAM interfacing through a dedicated bus and independent memory controller. The

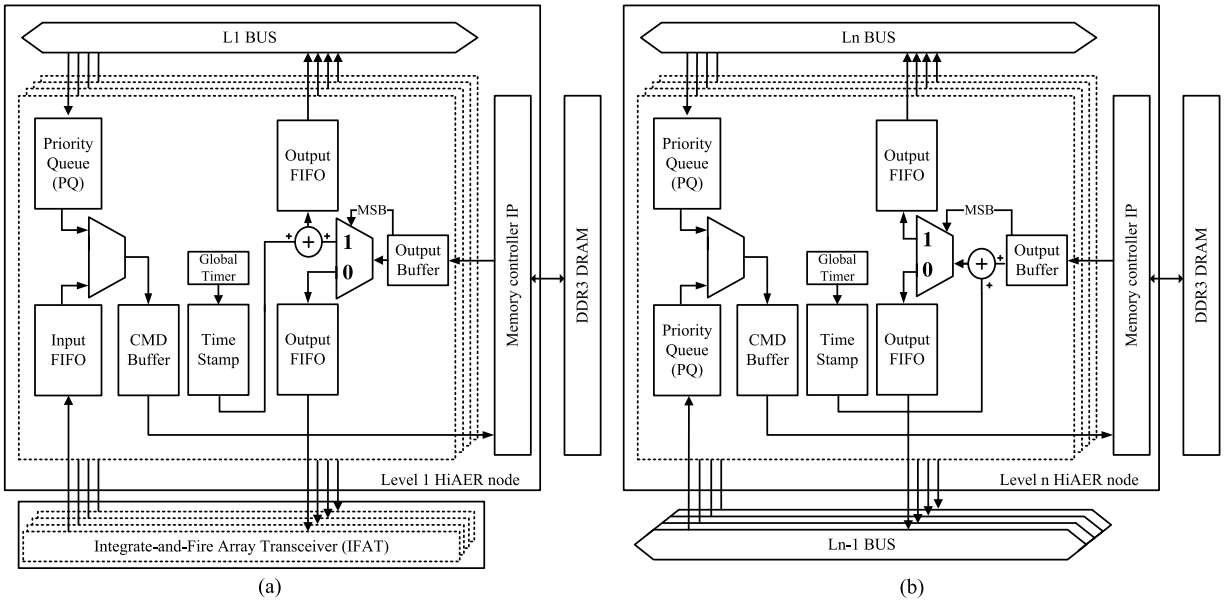


Fig. 4. (a) Simplified system architecture of an HiAER node at Level 1 (leaf in the hierarchy), routing synaptic events through the synaptic routing table (SRT) between physical neurons in the local IFAT, and RNs on the $L1$ bus. The SRT maps incoming events from any neuron onto outgoing events either to the final synaptic destination on the IFAT (along with synaptic strength w), or up the hierarchy through the $L1$ bus (along with timing information for axonal delay d). (b) Digital system architecture of an HiAER node at Level $n > 1$ (higher in the hierarchy), largely identical to Level 1 except for the substitution of the IFAT with an $Ln - 1$ bus, and of the $L1$ bus with an Ln bus. In the absence of physical neurons, events are transmitted only between RNs higher and/or lower in the hierarchy (along with timing information for axonal delay d).

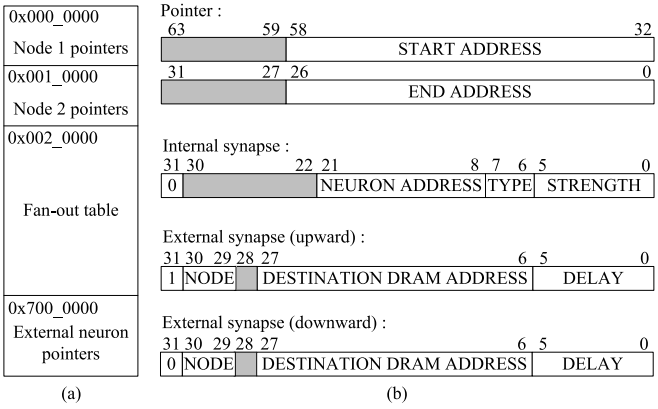


Fig. 5. (a) Synaptic routing table (SRT) storing pointers and fan-out information in 2-Gb DDR3 DRAM. The hexadecimal start addresses of DRAM partitions are shown. (b) SRT formats of internal (nodes 1 and 2) and external neuron pointers, and internal and external types of synaptic fan-out events. *Top:* A 64-b pointer contains 27-b start and end addresses of the synapse fan-out entries stored in the same DRAM. *Center:* Internal synapse connecting to a local neuron, encoding 14-b neuron address, 2-b synapse type, and 6-b strength. *Bottom:* External synapses connecting upward to higher (Ln) levels or downward to lower ($Ln - 1$) levels of HiAER nodes, with 2-b node address, 22-b address of the destination (RN), and 6-b delay.

memory controller further provides multiport access for sharing each physical DRAM with multiple data paths. In the current implementation, we partitioned each HiAER node into four leaf nodes, with two nodes sharing one DRAM through the same memory controller.

Fig. 5 shows the memory partitioning and various formats of events stored in the DRAM. For each input neuron, a 64-b pointer contains start and end addresses of synaptic fan-out entries in the same DRAM. The memory controller scans the data between start and end addresses to retrieve the information, specifying each of the outgoing events in sequence, where each occupies two words (32 b) in the fan-out table. The event type is marked by the MSB of the 32-b event in memory, which selects the path of the outgoing event up or down the HiAER hierarchy by the multiplexer shown in Fig. 4. Three types of outgoing events are thus distinguished: events descending to internal synapses local to the IFAT (for Level 1 leaf HiAER nodes), events descending to external synapses down the HiAER hierarchy (for all other HiAER nodes), and events ascending to external synapses up the HiAER hierarchy (for all HiAER nodes).

Internal synaptic events [MSB $= 0$ downward events at HiAER Level 1 in Fig. 4(a)] reach their final synaptic destination in the IFAT local to the HiAER node. The internal event contains the IFAT postsynaptic neuron address and synapse type A, and postsynaptic conductance w [48]. Other pertinent synaptic parameters, such as presynaptic release probability for stochastic synapses [9], may also be included in w. However, information on axonal delay d is excluded here, other than the delay inherent in propagating the event through HiAER.

External synaptic events [MSB $= 1$ upward events, or MSB $= 0$ downward events at HiAER Level $n > 1$ in Fig. 4(b)] connect to synaptic neurons in another node, whether neighboring or at another level in the hierarchy. External events code explicit delay timing information contributing incrementally to overall axonal delay d of the chain of events from source to final synaptic destination. The outgoing event feeding through the output buffer in Fig. 4 is given a deliver-at time stamp constructed as the sum of the 6-b delay and the

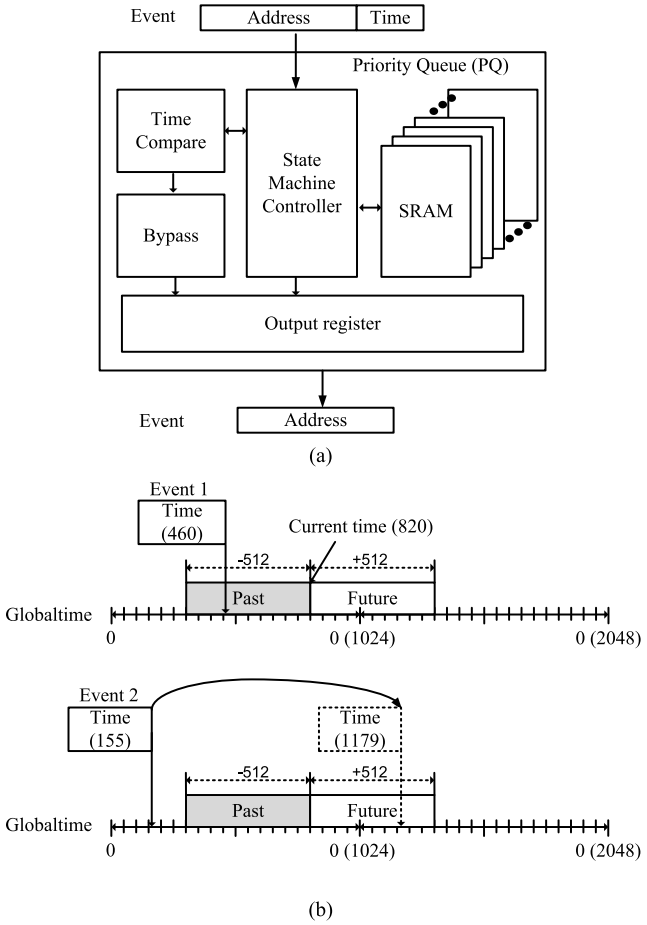


Fig. 6. (a) System diagram implementing the priority queue (PQ). Incoming events and their deliver-at time stamps are held in memory until their deliver-at time is reached by the current global time. (b) Examples illustrating temporal aliasing of the 10-b event deliver-at time stamps over the horizon of the 10-b current global time, distinguishing active future events from late past events.

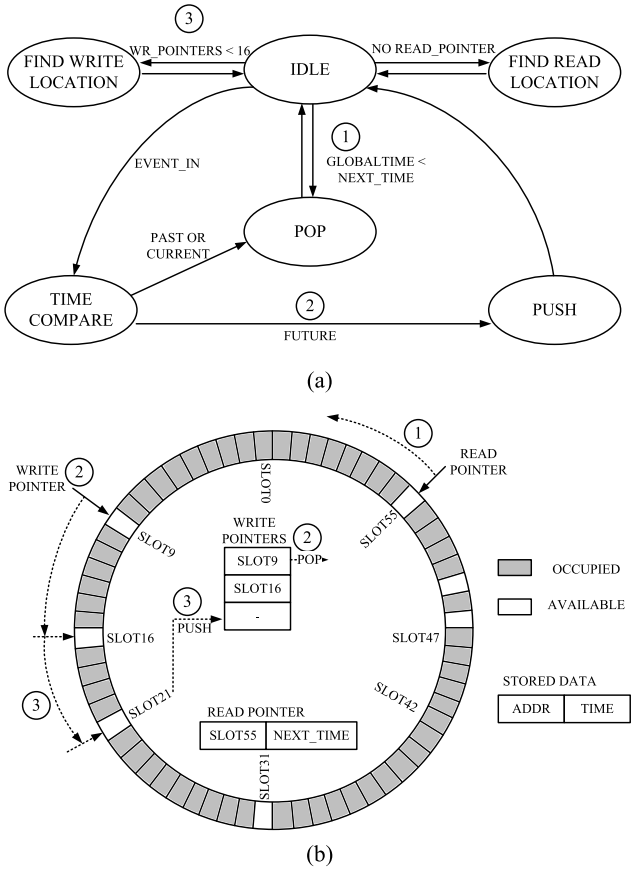


Fig. 7. (a) Simplified state machine transition diagram of the PQ. (b) Illustration of PQ timing and memory operation.

10-b time stamp of the incoming event, prior to exiting the HiAER node.

The number of synapses per neuron is not constrained in hardware, other than the total number of synapses that can be stored in available memory, not occupied by (node 1, node 2, and external) pointer blocks indicated in Fig. 5(a). A memory capacity of 2 Gb for every 2 HiAER nodes is chosen to accommodate a biologically realistic average synaptic fan-out of 1024 at 16 384 (2^{16}) neurons per node and 32 b per synapse.

C. Priority Queue

The PQ serves to hold incoming events, along with their deliver-at time stamps, and release each event only once its time stamp is reached by the global timer value. Hence, the nominal incremental delay, in units of the global timer clock, is the 6-b delay value as added to the sourcing event 10-b time stamp at the preceding HiAER node (see Section III-B). This quantized value is a lower bound on the incremental delay actually implemented by the PQ. The slack in the timing (tightness of this lower bound) is given by propagation delays in the routing path, mainly from the PQ exit stage through the event arbiter and CMD buffer shown in Fig. 4. Other propagation delays in the path from source to destination (from the previous routing node's SRT and DRAM memory controller, output buffer, output FIFO, and the internode bus, to the current node's PQ entry stage) are inconsequential as long as their cumulative delay is smaller than the programmed incremental delay, since any smaller net delay is absorbed in the wait time in the PQ.

The PQ uses the global timer along with an adjustable unit-time step parameter determining the granularity of the implemented delay. The implemented 1-ms time step and 6-b resolution in incremental delay support nominal single-node axonal delays up to 63 ms, with greater axonal delays achievable, if so desired, by nested routing across the hierarchy.

Fig. 6 shows the block diagram and state machine of the PQ and an example illustrating the time comparison method used. The PQ consists of a time comparator, a state machine, output register, and an SRAM module. The time comparator compares the current global time with deliver-at time stamps that join incoming events. The state machine controls the PQ event flow depending on the status of incoming events and the time to the next scheduled event release in the queue. Incoming events, consisting of a 22-b DRAM address and a 10-b deliver-at time stamp, are inserted (pushed) onto the queue in SRAM. Released events are removed (popped) from the queue, and reside in the output register until acknowledged by the next stage.

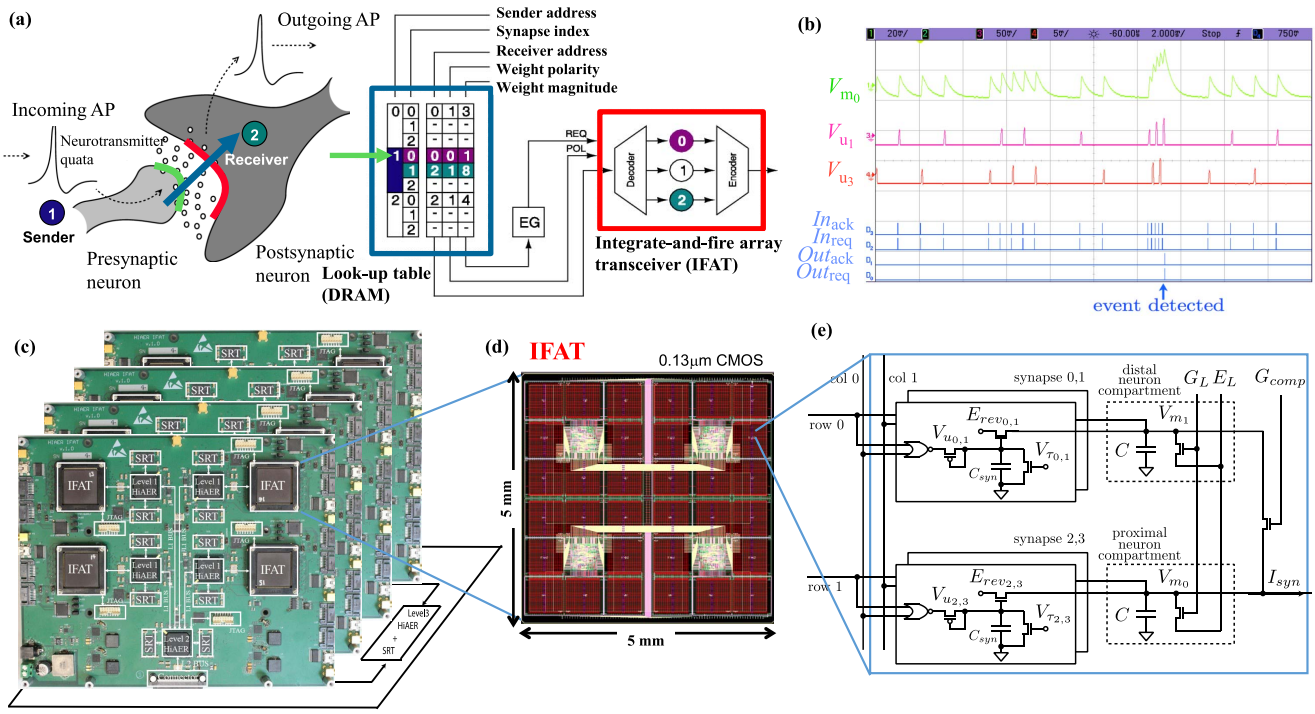


Fig. 8. Integrate-and-fire array transceiver (IFAT) with HiAER synaptic connectivity for scalable and reconfigurable neuromorphic neocortical processing [14]. (a) Dynamic reconfigurable synaptic connectivity across IFAT arrays of addressable neurons is implemented by routing neural spike events through DRAM look-up tables (SRTs). Only the Level 1 ($L1$) leaf node in HiAER synaptic connectivity is shown for simplicity. (b) IFAT neural array multiplexes and integrates incoming spike synaptic events (In_{ack} and In_{req}) generating continuous-time analog dynamics of synaptic (V_u) and neural (V_m) variables to produce outgoing spike neural events (Out_{ack} and Out_{req}). (c) Full-size HiAER-IFAT network with four boards, each with four IFAT modules, serving 1M neurons and 1G synapses, and spanning four levels in connection hierarchy. Each IFAT chip module comprises: (d) 65k-neuron Tezzaron 130-nm CMOS IFAT microchip; Xilinx Spartan-6 FPGA (Level 1 HiAER); and two 2-Gb DDR3 SDRAM SRTs serving 65M synapses. (e) Each neural cell models conductance-based membrane dynamics in proximal and distal compartments for synaptic input with programmable axonal delay, conductance, and reversal potential [14]. IFAT chip measured energy consumption is 48 pJ per spike event [14], several orders of magnitude more efficient than emulation on central/graphical processing unit (CPU/GPU) platforms.

Due to finite bit width of the global timer, improper time aliasing can occur with events whose deliver-at times lie beyond half of the full digital timing window range. By convention, we consider such aliased events as arriving too late, requiring immediate attention. Examples illustrating desired and improperly aliased operation are shown in Fig. 6(b). The top event has a deliver-at time stamp of 460 at a current time of 820 and, hence, is considered as a missed past event and is expedited to the CMD buffer. In contrast, the bottom event time stamped for 155, wrapping around to 1179 ($=2^{10}+155$), falls within $2^9 = 512$ cycles of the 820 current time, and enters the PQ wait table in the memory stack.

The finite state machine implementing the PQ, with state transitions driven by incoming events and time comparisons, is shown in Fig. 7. Incoming events (identified by EVENT_IN) trigger a time comparison, the result of which either directs the event to the output register (in case of a current or past event), or pushes it into the queue on the first available write pointer (in case of an active future event). The state machine also keeps track of the next event to be served using a NEXT_TIME variable, as the earliest of all stored time stamps in the queue, and its read location. Whenever the global current time reaches the NEXT_TIME value, the event stored at the read pointer is popped from the queue and directed to the output register. After the pop, the PQ enters a search to update the read location and NEXT_TIME, circulating once through the queue from the current location for the earliest future time stamp, while also popping any other event with the same deliver-at time as the present global time. Otherwise, the state machine checks for vacant positions in the queue to fill any available among 16 write pointers.

D. Global Timer Synchronization

The global timer synchronizes event communication and tracking across the multichip architecture. Although one common crystal oscillator feeds all five FPGAs, their internal system clocks are desynchronized due to phase jitter in their phase locked loops. To remedy timing errors between nodes across the HiAER hierarchy, a global timer in the top-level FPGA emits periodic global time increment events synchronizing local timers in all lower level FPGAs. To prevent accumulation of error due to missed or spurious time increment events, additional timer reset events are globally sent for every 10-b wraparound of the top-level global timer. These techniques combine to minimize the level of timing skew in the hierarchy.

IV. Experimental Results

In this section, we present experimental results characterizing latency, throughput, and capacity of synaptic routing through HiAER realized in an FPGA-based prototype embedding two levels of hierarchy with fourfold branching shown in Fig. 8. The HiAER tests are performed for different proof-of-concept configurations of network mappings and input spike

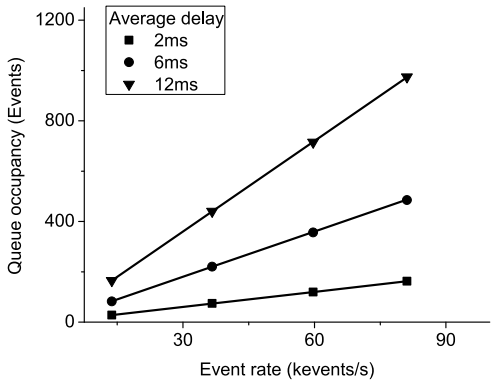


Fig. 9. Measured data of average PQ occupancy Q as a function of average event rate r and average axonal conduction delay d. Solid straight lines: theoretical model according to Little's law $Q = r\,d$ [47] for reference.

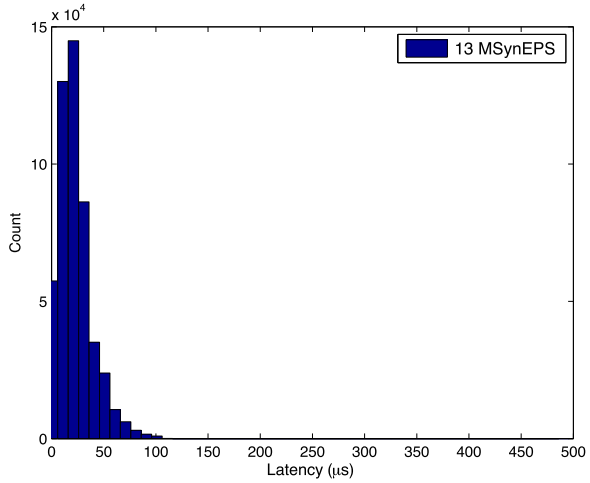
rates, and range from a single communication node [42] to the full implemented hierarchy, demonstrating improvements in throughput and latency linear in the number of routing nodes.

A. HiAER-IFAT Realized Prototype

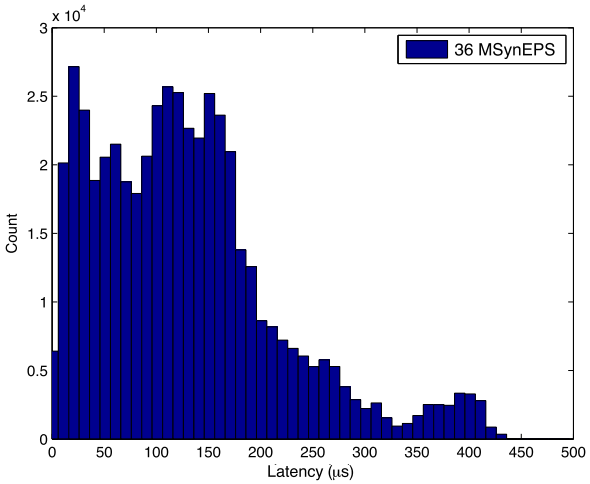
The hardware system in Fig. 8 integrates HiAER reconfigurable synaptic routing implemented using FPGAs and DRAM, with IFAT event-driven conductance-based continuous-time neural dynamics implemented in custom low-power mixed-signal very large-scale integrated circuits [14], [48].

Each quadruple set of HiAER Level 1 nodes (leaves in the hierarchy) shares one Xilinx Spartan-6 FPGA (XC6SLX45T), each sharing two 2-Gb DDR3 DRAMs (Micron MT41J128M16) for SRT storage. Four such units are provided on the board, along with an extra unit serving four HiAER Level 2 nodes, as indicated in Fig. 8. The nodes across the FPGAs are interconnected through $L1$ bus parallel communication links as shown. Each FPGA is also equipped with a local 200-MHz clock generator, an external clock input, and USB and JTAG ports for diagnostics and programming. An additional 200-MHz master clock generator can provide all $4 + 1$ HiAER nodes with a global clock. The system interfaces to the outside, at HiAER Level 3, through the $L2$ bus. Several boards can be combined to form a spike-based neuromorphic computer with more than 2^{18} (262 144) analog integrate-and-fire neurons and high-speed peripherals using different variants of address-event routing protocols, e.g., [19], [28]–[30]. The data presented in the following are obtained by connecting the $L2$ bus of a single HiAER board over a USB 2.0 interface to a workstation.

Each IFAT chip contains four independent ports, each port with 16k two-compartment integrate-and-fire neurons [14], [48] and assigned a single HiAER Level 1 node. The IFAT neural array transceives incoming synaptic spike events to outgoing neural spike events generated through internal analog continuous-time dynamics of synaptic and neural state variables (Fig. 8) [14]. Internally continuous-time analog, but externally asynchronous digital, the IFAT interfaces directly with HiAER to emulate large-scale biophysical models of cortical neural dynamics with reconfigurable synaptic connectivity [42]. Each neuron



(a)



(b)

Fig. 10. Measured latency between presynaptic and postsynaptic events through the SRT at a Level 1 HiAER node (16k neurons), at a sustained throughput of (a) 1.3×10^7 synaptic events per second (SynEPS) and (b) 3.6×10^7 SynEPS. The SRT was programmed with uniform 1000 synaptic fan-out and zero nominal axonal conduction delays ($d = 0$), and the system clock was 150 MHz.

in the IFAT array models two (proximal and distal) membrane compartments, of which one is excitable for spike generation and event registration. Coupled to each of the two compartments are two independent types of conductance-based synapses that are dynamically instantiated through time multiplexing of HiAER input synaptic events. Programmable control over synaptic reversal potentials and time constants of the conductance provides for nonlinear pooling mechanisms in shunting inhibition and temporal coding in synchrony detection [48] that are critical elements of spike-based neural computation missing from simplified linear integrate-and-fire models as more commonly implemented in analog or digital neuromorphic VLSI.

Custom VLSI implementation of IFAT and detailed characterization of its neural dynamics are presented in [14] and [48]. This paper focuses on efficiency and scaling in the realization of the HiAER synaptic routing independent of IFAT or other

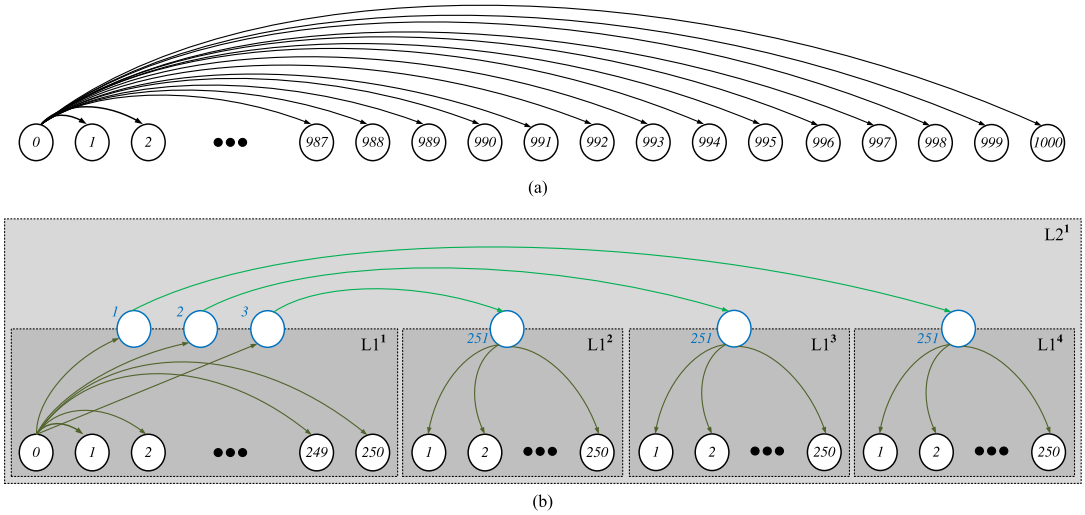


Fig. 11. Example network partitioning of one presynaptic neuron connecting to 1000 postsynaptic neurons (a) implemented in single-node flat hierarchy and (b) implemented across two levels of hierarchy partitioned into four HiAER nodes each with 250 postsynaptic neurons. For diagrammatic simplicity, the network is only shown for a single one out of a total of 4000 implemented presynaptic neurons, each instantiating a copy of the network connecting to the same 1000 postsynaptic neurons.

means for neural integration and spike generation. Indeed HiAER is applicable to a wide range of event-driven large-scale implementations of neural models [8]–[13], [31]–[40].

B. Experimental Setup

The objective of the experiments is to characterize the scaling of HiAER synaptic routing performance by collecting statistics on queue occupancy, event latency, and throughput under varying controlled conditions of network load and topology. To avoid timing distortion induced by latency of the USB interface between the HiAER $L2$ bus and the workstation, we implemented spike event generators and histogram recorders in FPGA on the board. Spike event generators at the Level 2 HiAER node produce neural event spike trains entering the $L1$ bus with interspike intervals drawn from a Poisson distribution parameterized in mean spike rate. Histogram recorders at each of the Level 1 HiAER nodes take the place of the local IFAT analog array, collecting statistics on time arrivals of received synaptic events while emulating the IFAT's asynchronous AER handshaking of the incoming events. Timing statistics are computed based on time stamps of received events in relation to the current global timer value. Received events are binned accordingly, with their counts accumulated over a fixed number of trial events.

C. Priority Queue Analysis

Measured results from the PQ are shown in Fig. 9. The event generator was configured to produce Poisson spike trains of variable rate r, modeling varying loads of RN events entering the HiAER node. The events were given Poisson distributed axonal delays d with mean delays of 2, 6, and 12 ms. Little's law [47] predicts the average queue occupancy Q under such conditions to be $Q = r\,d$ where r is the average incoming event rate and d is the average delay in the queue. Measured results of Q from recorded PQ occupancy data for varying input rate r and average axonal delay d are marked with symbols on the graph in Fig. 9, with intersecting straight lines indicating the theoretical fit following Little's law.

TABLE I
FPGA RESOURCE USAGE FOR PQ IMPLEMENTATION

Queue depth	1,024	2,048	4,096	8,192
Number of slice registers	670	1,195	2,243	4,343
Number of slice LUTs	654	1,234	2,382	4,810
Number of block RAM/FIFO	3	5	9	17

Table I shows the FPGA resource usage for PQ implementation on the target device (Xilinx Spartan-6 XC6SLX45T) for varying queue depth, showing how the implemented PQ on the HiAER board with queue depth 1024 scales to larger queue sizes, trading performance for resource usage in approximately linear fashion, limited mainly by total SRAM capacity on the FPGA device.

D. Event Latency Through Single-Node HiAER

Next, we analyzed event latency for varying data rate of synaptic events through the Level 1 HiAER node. We again used Poisson event generators with variable spike rate, and measured event latency from histogram recorded data of time-stamp differences over 1 million synaptic events. We implemented an average synaptic fan-out of 1000 in the SRT, generating on average 1000 synaptic outgoing events per incoming neural spike event consistent with models of synaptic connectivity in the mammalian central nervous system [1], [2]. However, the axonal delay d was set to zero in order to emulate worst conditions for event throughput and latency: every event entering any PQ is late upon arrival and must exit immediately, accumulating latency in the process for every HiAER stage in the event chain. In contrast, events with axonal delay d greater than accumulated propagation delays enter the PQ and

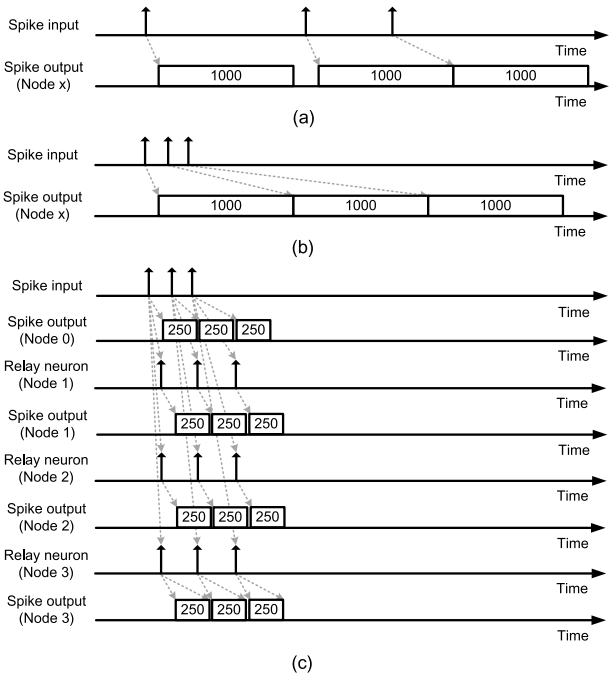


Fig. 12. Effect of hierarchical network partitioning on event latency and throughput, for the example network in Fig. 11. (a) In the single-node flat hierarchy, event latency through the SRT at low neural spike input event rate ranges between 0 and $N\tau_{\text{SRT}}$, where $N = 1000$ is the synaptic fan-out and τ_{SRT} is the SRT recall latency. (b) Neural spike input event rates greater than its capacity $1/N\tau_{\text{SRT}}$ result in progressively growing event latencies. (c) Partitioning of the network across four HiAER nodes, connected through three RNs, results into a fourfold decrease in local synaptic fan-out and, equivalently, event latency. The fourfold parallelism also supports a fourfold greater overall event throughput across the network.

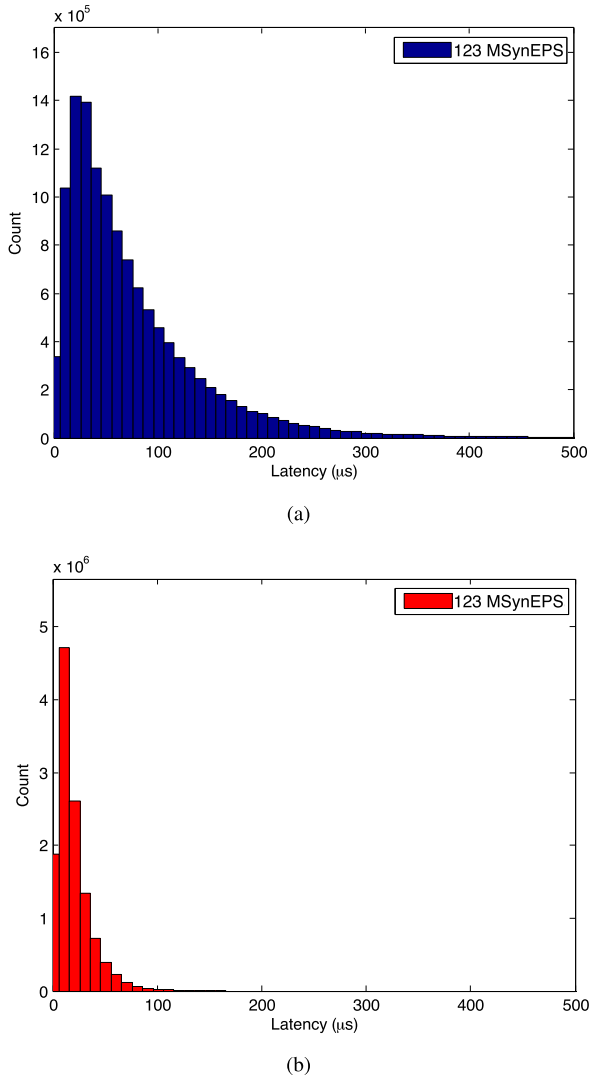


Fig. 13. Measured latency between presynaptic and postsynaptic events through four SRT nodes at the Level 1 HiAER (65k neurons) in the HiAER-IFAT hierarchy, at a sustained throughput of 1.23×10^8 SynEPS with (a) flat mapping and (b) fourfold hierarchical mapping of the network in Fig. 11.

resynchronize with the global timer exiting the PQ with near-zero latency. Latency of event delivery under such conditions is thus limited to latency of only the final HiAER stage in the event chain. Hence, the measured latency for zero axonal delay $d = 0$ should be taken as an upper bound on latency in the general case.

Fig. 10 shows the measured latency, at 150-MHz system clock, of synaptic output events for two input event rates, indicating latencies below 100 μs at 1.3×10^7 synaptic events per second (SynEPS) throughput, and latencies below 450 μs at 3.6×10^7 SynEPS throughput.

E. Event Latency and Throughput Through Four Parallel HiAER Nodes

To validate improvements in latency and throughput owing to parallelism in hierarchical routing, we conducted experiments with flat and nested structured implementation of simple networks, with 4000 presynaptic neurons sharing a common set of 1000 postsynaptic neurons. Fig. 11 shows the connection topology for any one single presynaptic neuron. A flat hierarchy implementation within a single $L1$ node is shown in Fig. 11(a). The same network is partitioned through three RNs into four $L1$ HiAER nodes each with 250 postsynaptic neurons. For simplicity, the hierarchical partition is only shown for a single presynaptic neuron in Fig. 11(b); in reality, each of 1000 such presynaptic neurons per $L1$ node instantiates a copy of this network connecting to the same sets of postsynaptic neurons across the four $L1$ nodes. The effect of the network partitioning on event latency and throughput is shown in Fig. 12. As shown, fourfold partitioning diminishes the local fan-out requirement fourfold leading to approximately fourfold lower event latency, ranging between 0 and $\frac{1}{4}N\tau_{\text{SRT}}$, where τ_{SRT} is the SRT latency per synapse. In addition, the resulting fourfold parallelism in local event routing leads to approximately fourfold increased event throughput across the network, relative to the single-node case. The maximum net synaptic event throughput (or synaptic channel capacity) across all the four nodes is thus $4/\tau_{\text{SRT}}$.

Fig. 13 shows the measured latency between presynaptic and postsynaptic event through four Level 1 HiAER nodes. For these experiments, we used the Poisson spike generator to route 50 000 neural events across the $L1$ bus at 1.23×10^5 events per second. All PQs were cleared of preexisting events at start of each experiment in order to

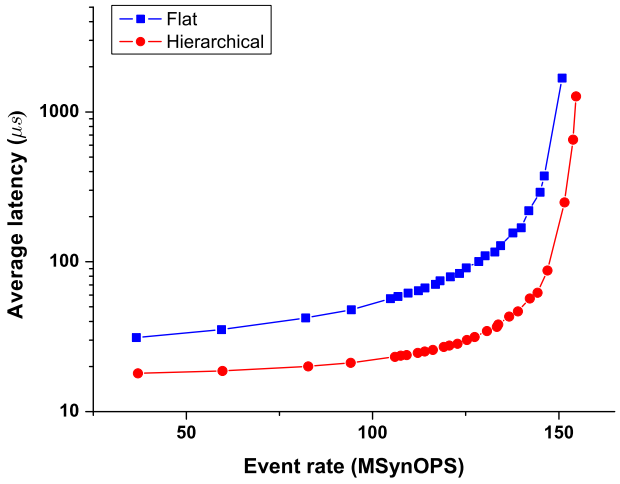


Fig. 14. Average event latency measured as a function of synaptic event rate for flat and fourfold hierarchical partitioning of the network in Fig. 11.

provide zero initial conditions in event latency. Four parallel HiAER nodes were used for both flat (locally connected) and hierarchical mapping, to equalize net synaptic event channel capacity across both cases. For the flat hierarchy in Fig. 11(a), latency is measured from data collected by the histogram recorder on each of the four HiAER nodes with local 1000 fan-out. For the four-node two-level hierarchy of Fig. 11(b), latency was measured from data collected across four histogram recorders, one for each HiAER node, each with local 250 fan-out. A shorter tail and narrower distribution is observed in the case of fourfold hierarchical mapping, with worst case latency of 125 μs, about a fourfold improvement over the case of flat mapping.

Measured event latency as a function of synaptic event rate, for flat and hierarchical mapping, averaged over all 50M synaptic events from empty PQ initial conditions, is shown in Fig. 14. At higher event rates, a fourfold reduction in latency for hierarchical mapping is consistently observed, even at transient event rates exceeding the synaptic event channel capacity of 1.44×10^8 SynEPS across the four parallel nodes.

V. Conclusion

We presented HiAER, and its efficient implementation in digital hardware, as a hierarchical scalable extension to synaptic address-event routing for large-scale spike-based neuromorphic systems with reconfigurable long-range synaptic connectivity, in which both strength and axonal delay for each implemented synapse are individually programmable. As a proof-of-concept, a two-level fourfold branching hierarchy with 262k two-compartment integrate-and-fire neurons, each fanning out to any other neurons with thousand synapses on average, was implemented on a custom PCB with five Xilinx Spartan-6 FPGAs, ten DDR3 DRAMs, and four custom IFAT mixed-signal VLSI microchips. At the single-board level, we demonstrated approximately linear scaling in the throughput of global synaptic event routing at 36 MSynEPS per 16k-neuron node in the hierarchy. We also showed decreased event latency, from 83.6 μs for flat partitioning to 28.3 μs for fourfold hierarchical partitioning owing to the corresponding reduction of local connectivity in the distributed network. Furthermore, we showed average queue occupancy in the PQs consistent with Little's law, with 12 ms of average axonal delay at 8×10^4 events/s RN event rate per HiAER routing node for the implemented 1024 queue depth in FPGA SRAM.

Larger-size networks, in principle of unlimited size, may be obtained by cascading boards to extend the HiAER hierarchy to higher levels at net synaptic throughput scaling with the number of nodes across the hierarchy [41]. Hierarchical partitioning of axonal delay may further support temporal spike-based models of neural computation based on pattern matching in delayed spike coincidence detection [43] at virtually unlimited range of delays. Conversely, recently developed stochastic rate-based models with Monte Carlo Markov chain (MCMC) neural sampling from Boltzmann distributions in large-scale spiking networks with biophysical integrate-and-fire neurons [49] and their extensions to online learning spike-based Boltzmann machines [50] map directly onto the HiAER architecture as well.

The challenges in further scaling up hardware realizations of HiAER are multifold, calling for further advances in:

a) Area and Energy Efficiency: Measuring 20 cm $\times$ 25 cm and consuming 10 W of power at 720 MSynEPS net synaptic throughput across five FPGAs, the presented 262k-neuron, 262M-synapse implementation offers an area efficiency of 200 μm^2 per synapse and an energy efficiency of 14 nJ per synaptic event. Although a respectable feat of neuromorphic engineering, the realized efficiencies pale in comparison with the 10^{-3} μm^2 area and roughly 10-fJ energy per synapse for the human brain, which counts roughly 10^{15} synapses, each activated on average at roughly 2 Hz, within 0.002 m^3 volume and across 1 m^2 cortical surface area, and at 20 W of metabolic power consumption [1], [2], [51]–[54]. The HiAER realized efficiencies are limited by DRAM memory cell density and read energy in serial access of SRTs, and by the FPGA general-purpose reconfigurable logic. Significant area and energy improvements can be expected from custom silicon integration of SRTs distributed across HiAER routing nodes, such as using wafer-scale integration [35], [36] or vertically stacked 3-D integration of CMOS and memory technologies [55], [56]. Further energy improvements may also result from direct asynchronous synthesis of all HiAER events routing, including PQ, FIFOs, and possibly DRAM memory controller. The advantage of asynchronous implementation, in the absence of any clock, is that power scales directly with event rate, except for static standby power [57], [58].

b) Efficient Partitioning: Efficient use of HiAER resources is critically dependent on efficient partitioning of the implemented network into a hierarchy of clusters that minimizes event traffic across routing nodes. The general problem of efficient hierarchical graph partitioning is well studied, and solutions formulated in various application domains [59] may be ported to hierarchical synaptic partitioning, in tandem with compilation and analysis tools for efficient mapping of the hierarchical neural and synaptic structure onto neuromorphic architecture [41], [60], [61]. In addition, anatomical and functional connectivity information gathered from connectomics [62], [63] may guide naturally efficient

network partitioning inspired by the structural organization of the central nervous system.

c) Efficient Learning: Although not pursued here, HiAER may be extended with local mechanisms of STDP implemented directly in the address domain [15] to learn the HiAER long-range synaptic connectivity online from real-time data. STDP-based models of temporally asymmetric Hebbian unsupervised learning extend to other forms of spike-based learning, such as reinforcement learning of distal reward using STDP-modulated dopamine signaling [64], and deep learning of multilayered cortical representations using STDP event-driven contrastive divergence in spiking Boltzmann machines [50]. The advantage of HiAER for efficient hierarchical event-driven implementation of STDP-based online learning is that all information on synaptic strength, regardless of global range in connectivity, resides only in local SRTs at the final destination (Level 1 HiAER) leaf nodes in the hierarchy, in direct proximity to both presynaptic and postsynaptic event streams. Thus, the local implementation of event-driven STDP at Level 1 HiAER SRTs may be sufficient to support more general implementation of complex nonlocal learning rules that take advantage of the global nested network structure with the long-range and hierarchical connectivity provided by HiAER.

REFERENCES

[1] P. S. Churchland and T. J. Sejnowski, *The Computational Brain*. Cambridge, MA, USA: MIT Press, 1992.

[2] G. M. Shepherd, *The Synaptic Organization of the Brain*, 5th ed. New York, NY, USA: Oxford Univ. Press, 2003.

[3] M. A. Sivilotti, "Wiring considerations in analog VLSI systems, with application to field-programmable networks," Ph.D. dissertation, Dept. Comput. Sci., California Inst. Technol., Pasadena, CA, USA, 1991.

[4] J. Lazzaro, J. Wawrzynek, M. Mahowald, M. Sivilotti, and D. Gillespie, "Silicon auditory processors as computer peripherals," *IEEE Trans. Neural Netw.*, vol. 4, no. 3, pp. 523–528, May 1993.

[5] M. Mahowald, *An Analog VLSI System for Stereoscopic Vision*, vol. 265. Heidelberg, Germany: Springer, 1994.

[6] S. R. Deiss, R. J. Douglas, and A. M. Whatley, *A Pulse-Coded Communications Infrastructure for Neuromorphic Systems*. Cambridge, MA, USA: MIT Press, 1999, ch. 6, pp. 157–178.

[7] K. A. Boahen, "Point-to-point connectivity between neuromorphic chips using address events," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 5, pp. 416–434, May 2000.

[8] G. Indiveri, A. M. Whatley, and J. Kramer, "A reconfigurable neuromorphic VLSI multi-chip system applied to visual motion computation," in *Proc. 7th Int. Conf. Microelectron. Neural, Fuzzy Bio-Inspired Syst.*, Apr. 1999, pp. 37–44.

[9] D. H. Goldberg, G. Cauwenberghs, and A. G. Andreou, "Probabilistic synaptic weighting in a reconfigurable network of VLSI integrate-and-fire neurons," *Neural Netw.*, vol. 14, nos. 6–7, pp. 781–793, Jul. 2001.

[10] S.-C. Liu and R. Douglas, "Temporal coding in a silicon network of integrate-and-fire neurons," *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1305–1314, Sep. 2004.

[11] D. Sridharan, B. Percival, J. Arthur, and K. A. Boahen, "An in-silico neural model of dynamic routing through neuronal coherence," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 20. 2008, pp. 1401–1408.

[12] R. J. Vogelstein, U. Mallik, J. T. Vogelstein, and G. Cauwenberghs, "Dynamically reconfigurable silicon array of spiking neurons with conductance-based synapses," *IEEE Trans. Neural Netw.*, vol. 18, no. 1, pp. 253–265, Jan. 2007.

[13] R. Serrano-Gotarredona *et al.*, "CAVIAR: A 45k neuron, 5M synapse, 12G connects/s AER hardware sensory–processing–learning–actuating system for high-speed visual object recognition and tracking," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1417–1438, Sep. 2009.

[14] T. Yu, J. Park, S. Joshi, C. Maier, and G. Cauwenberghs, "65k-neuron integrate-and-fire array transceiver with address-event reconfigurable synaptic routing," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Nov. 2012, pp. 21–24.

[15] R. J. Vogelstein, F. Tenore, R. Philipp, M. S. Adlerstein, D. H. Goldberg, and G. Cauwenberghs, "Spike timing-dependent plasticity in the address domain," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 15. 2003, pp. 1171–1178.

[16] S. A. Bamford, A. F. Murray, and D. J. Willshaw, "Large developing receptive fields using a distributed and locally reprogrammable address–event receiver," *IEEE Trans. Neural Netw.*, vol. 21, no. 2, pp. 286–304, Feb. 2010.

[17] K. A. Boahen and A. G. Andreou, "A contrast sensitive silicon retina with reciprocal synapses," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 4. 1992, pp. 764–772.

[18] M. Mahowald, "VLSI analogs of neuronal visual processing: A synthesis of form and function," Ph.D. dissertation, Dept. Comput. Neural Syst., California Inst. Technol., Pasadena, CA, USA, 1992.

[19] P. Lichtsteiner, C. Posch, and T. Delbrück, "A 128 × 128 120 dB 15 μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Feb. 2008.

[20] T. Serrano-Gotarredona and B. Linares-Barranco, "A 128 × 128 1.5% contrast sensitivity 0.9% FPN 3 μs latency 4 mW asynchronous frame-free dynamic vision sensor using transimpedance preamplifiers," *IEEE J. Solid-State Circuits*, vol. 48, no. 3, pp. 827–838, Mar. 2013.

[21] J. Lazzaro, "Temporal adaptation in a silicon auditory nerve," in *Proc. Adv. Nueral Inf. Process. Syst.*, 1992, pp. 813–820.

[22] V. Chan, S.-C. Liu, and A. van Schaik, "AER EAR: A matched silicon cochlea pair with address event representation interface," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 1, pp. 48–59, Jan. 2007.

[23] V. Chan, C. Jin, and A. van Schaik, "An address-event vision sensor for multiple transient object detection," *IEEE Trans. Biomed. Circuits Syst.*, vol. 1, no. 4, pp. 278–288, Dec. 2007.

[24] Z. Fu, T. Delbrück, P. Lichtsteiner, and E. Culurciello, "An address-event fall detector for assisted living applications," *IEEE Trans. Biomed. Circuits Syst.*, vol. 2, no. 2, pp. 88–96, Jun. 2008.

[25] S. Ramakrishnan, R. Wunderlich, and P. Hasler, "Neuron array with plastic synapses and programmable dendrites," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Nov. 2012, pp. 400–403.

[26] J. A. Pérez-Carrasco, B. Acha, C. Serrano, L. Camuñas-Mesa, T. Serrano-Gotarredona, and B. Linares-Barranco, "Fast vision through frameless event-based sensing and convolutional processing: Application to texture recognition," *IEEE Trans. Neural Netw.*, vol. 21, no. 4, pp. 609–620, Apr. 2010.

[27] E. Neftci, J. Binas, U. Rutishauser, E. Chicca, G. Indiveri, and R. J. Douglas, "Synthesizing cognition in neuromorphic electronic systems," *Proc. Nat. Acad. Sci. USA*, vol. 110, no. 37, pp. E3468–E3476, Sep. 2013.

[28] H. K. O. Berge and P. Häfliger, "High-speed serial AER on FPGA," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2007, pp. 857–860.

[29] D. B. Fasnacht, A. M. Whatley, and G. Indiveri, "A serial communication infrastructure for multi-chip address event systems," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2008, pp. 648–651.

[30] C. Zamarreño-Ramos, T. Serrano-Gotarredona, and B. Linares-Barranco, "A 0.35 μm sub-ns wake-up time ON-OFF switchable LVDS driver-receiver chip I/O pad pair for rate-dependent power saving in AER bit-serial links," *IEEE Trans. Biomed. Circuits Syst.*, vol. 6, no. 5, pp. 486–497, Oct. 2012.

[31] P. A. Merolla, J. V. Arthur, B. E. Shi, and K. A. Boahen, "Expandable networks for neuromorphic chips," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 2, pp. 301–311, Feb. 2007.

[32] B. V. Benjamin *et al.*, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proc. IEEE*, vol. 102, no. 5, pp. 699–716, May 2014.

[33] P. Merolla, J. Arthur, R. Alvarez, J.-M. Bussat, and K. Boahen, "A multicast tree router for multichip neuromorphic systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 3, pp. 820–833, Mar. 2014.

[34] C. Zamarreño-Ramos, A. Linares-Barranco, T. Serrano-Gotarredona, and B. Linares-Barranco, "Multicasting mesh AER: A scalable assembly approach for reconfigurable neuromorphic structured AER systems. Application to ConvNets," *IEEE Trans. Biomed. Circuits Syst.*, vol. 7, no. 1, pp. 82–102, Feb. 2013.

[35] J. Fieres, J. Schemmel, and K. Meier, "Realizing biological spiking network models in a configurable wafer-scale hardware system," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2008, pp. 969–976.

[36] S. Millner, A. Grübl, K. Meier, J. Schemmel, and M.-O. Schwartz, "A VLSI implementation of the adaptive exponential integrate-and-fire neuron model," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 23. 2011, pp. 1642–1650.
[37] S. Scholze *et al.*, "VLSI Implementation of a 2.8 Gevent/s packet-based AER interface with routing and event sorting functionality," *Frontiers Neurosci.*, vol. 5, no. 117, pp. 117:1–117:13, 2011, DOI: 10.3389/fnins.2011.00117.
[38] P. A. Merolla *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, Aug. 2014.
[39] M. M. Khan *et al.*, "SpiNNaker: Mapping neural networks onto a massively-parallel chip multiprocessor," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2008, pp. 2849–2856.
[40] E. Painkras *et al.*, "SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation," *IEEE J. Solid-State Circuits*, vol. 48, no. 8, pp. 1943–1953, Aug. 2013.
[41] S. Joshi, S. Deiss, M. Arnold, J. Park, T. Yu, and G. Cauwenberghs, "Scalable event routing in hierarchical neural array architecture with global synaptic connectivity," in *Proc. 12th Int. Workshop Cellular Nanosc. Netw. Appl. (CNNA)*, Feb. 2010, pp. 1–6.
[42] J. Park, T. Yu, C. Maier, S. Joshi, and G. Cauwenberghs, "Live demonstration: Hierarchical address-event routing architecture for reconfigurable large scale neuromorphic systems," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2012, pp. 707–711.
[43] E. M. Izhikevich and F. C. Hoppensteadt, "Polychronous wavefront computations," *Int. J. Bifurcation Chaos*, vol. 19, no. 5, pp. 1733–1739, 2009.
[44] S. Sheik, E. Chicca, and G. Indiveri, "Exploiting device mismatch in neuromorphic VLSI systems to implement axonal delays," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2012, pp. 1–6.
[45] R. Wang, J. Tapson, T. J. Hamilton, and A. van Schaik, "An aVLSI programmable axonal delay circuit with spike timing dependent delay adaptation," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2012, pp. 2413–2416.
[46] B. Belhadj, A. Joubert, O. Temam, and R. Heliot, "Configurable conduction delay circuits for high spiking rates," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2012, pp. 2091–2094.
[47] J. D. C. Little, "A proof for the queuing formula: $L = \lambda W$," *Oper. Res.*, vol. 9, no. 3, pp. 383–387, 1961.
[48] T. Yu, J. Park, S. Joshi, C. Maier, and G. Cauwenberghs, "Event-driven neural integration and synchronicity in analog VLSI," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug./Sep. 2012, pp. 775–778.
[49] M. A. Petrovici, J. Bill, I. Bytschok, J. Schemmel, and K. Meier. (2013). "Stochastic inference with deterministic spiking neurons." [Online]. Available: http://arxiv.org/abs/1311.3211
[50] E. Neftci, S. Das, B. Pedroni, K. Kreutz-Delgado, and G. Cauwenberghs, "Event-driven contrastive divergence for spiking neuromorphic systems," *Frontiers Neurosci.*, vol. 7, no. 272, pp. 272:1–272:14, 2014, DOI: 10.3389/fnins.2013.00272.
[51] B. Fischl and A. M. Dale, "Measuring the thickness of the human cerebral cortex from magnetic resonance images," *Proc. Nat. Acad. Sci. USA*, vol. 97, no. 20, pp. 11050–11055, 2000.
[52] G. Cauwenberghs, "Reverse engineering the cognitive brain," *Proc. Nat. Acad. Sci. USA*, vol. 110, no. 39, pp. 15512–15513, 2013.
[53] D. Attwell and S. B. Laughlin, "An energy budget for signaling in the grey matter of the brain," *J. Cerebral Blood Flow Metabolism*, vol. 21, no. 10, pp. 1133–1145, 2001.
[54] P. Lennie, "The cost of cortical computation," *Current Biol.*, vol. 13, no. 6, pp. 493–497, Mar. 2003.
[55] U. Kang *et al.*, "8 Gb 3-D DDR3 DRAM using through-silicon-via technology," *IEEE J. Solid-State Circuits*, vol. 45, no. 1, pp. 111–119, Jan. 2010.
[56] D. H. Kim *et al.*, "3D-MAPS: 3D massively parallel processor with stacked memory," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers (ISSCC)*, Feb. 2012, pp. 188–190.
[57] P. Merolla, J. Arthur, F. Akopyan, N. Imam, R. Manohar, and D. S. Modha, "A digital neurosynaptic core using embedded crossbar memory with 45 pJ per spike in 45 nm," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Sep. 2011, pp. 1–4.
[58] N. Imam, F. Akopyan, J. Arthur, P. Merolla, R. Manohar, and D. S. Modha, "A digital neurosynaptic core using event-driven QDI circuits," in *Proc. 18th IEEE Int. Symp. Asynchron. Circuits Syst. (ASYNC)*, May 2012, pp. 25–32.
[59] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 359–392, Aug. 1999.
[60] M. DeBole, A. A. Maashri, M. Cotter, C.-L. Yu, C. Chakrabarti, and V. Narayanan, "A framework for accelerating neuromorphic-vision algorithms on FPGAs," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2011, pp. 810–813.
[61] J. Partzsch and R. Schüffny, "Analyzing the scaling of connectivity in neuromorphic hardware and in models of neural networks," *IEEE Trans. Neural Netw.*, vol. 22, no. 6, pp. 919–935, Jun. 2011.
[62] T. E. Behrens and O. Sporns, "Human connectomics," *Current Opinion Neurobiol.*, vol. 22, no. 1, pp. 144–153, Feb. 2012.
[63] T. A. Jarrell *et al.*, "The connectome of a decision-making neural network," *Science*, vol. 337, no. 6093, pp. 437–444, 2012.
[64] E. M. Izhikevich, "Solving the distal reward problem through linkage of STDP and dopamine signaling," *Cerebral Cortex*, vol. 17, no. 10, pp. 2443–2452, 2007.

Jongkil Park (S'09–M'16) received the B.S. degree in electrical engineering from Korea University, Seoul, South Korea, in 2007, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at San Diego, La Jolla, CA, USA, in 2010 and 2014, respectively.

He joined as a Researcher with the Electronics and Telecommunications Research Institute, Daejeon, South Korea, in 2014. His current research interests include mixed-signal very large-scale integration design for neuromorphic architecture and vision application.

Theodore Yu (S'04–M'14) received the B.S. and M.S. degrees in electrical engineering from the California Institute of Technology, Pasadena, CA, USA, in 2004 and 2005, respectively, and the Ph.D. degree in electrical and computer engineering from the University of California at San Diego, La Jolla, CA, USA.

He is currently with Texas Instruments, Santa Clara, CA, USA. His current research interests include neuromorphic analog very large-scale integration models of neural and synaptic circuits, neuron-silicon interfaces, and circuit implementations of learning algorithms.

Siddharth Joshi (S'14) received the B.Tech. degree in information and communication technology from the Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, India, in 2008, and the M.S. degree in electrical engineering from the University of California at San Diego, La Jolla, CA, USA, in 2011, where he is currently pursuing the Ph.D. degree.

His current research interests include very large-scale integration circuit implementations of learning algorithms and neuromorphic structures, and the design of adaptive low power circuits.

Christoph Maier (M'96) received the Diplom-Physiker degree from the Unversity of Heidelberg, Germany, in 1995, and the Dr.sc.techn. degree in electrical engineering from the Swiss Federal Institute of Technology Zurich, Zurich, Switzerland, in 2000.

After time in industry, he joined the Integrated Systems Neuroengineering Laboratory at the University of California, San Diego, La Jolla, CA, USA, as a Post-Doctoral Researcher in 2010. His main research interests are interfaces for electrophysiological signals and modeling neural networks in analog VLSI.

Gert Cauwenberghs (S'89–M'94–SM'04–F'11) received the Ph.D. degree in electrical engineering from the California Institute of Technology, Pasadena, CA, USA, in 1994.

He was a Professor of Electrical and Computer Engineering with Johns Hopkins University, Baltimore, MD, USA, and a Visiting Professor of Brain and Cognitive Science with the Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently a Professor of Bioengineering and Co-Director of the Institute for Neural Computation with the University of California at San Diego, La Jolla, CA, USA. He co-founded Cognionics Inc., San Diego, CA, USA, where he chairs its Scientific Advisory Board. His current research interests include micropower biomedical instrumentation, neuron–silicon and brain–machine interfaces, neuromorphic engineering, and adaptive intelligent systems.

Dr. Cauwenberghs received the NSF Career Award in 1997, the ONR Young Investigator Award in 1999, and the Presidential Early Career Award for Scientists and Engineers in 2000. He served IEEE in a variety of roles, including as the General Chair of the IEEE Biomedical Circuits and Systems Conference (2011, San Diego), the Program Chair of the IEEE Engineering in Medicine and Biology Conference (2012, San Diego), and the Editor-in-Chief of the IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS.