A Review of Spiking Neuromorphic Hardware Communication Systems

Aaron R. Young
Mark E. Dean
James S. Plank
Garrett S. Rose

IEEE Access, vol. 7, pp. 135606-135620, 2019.

https://ieeexplore.ieee.org/document/8843969

Citation Information—Plain Text:

```
author       A. R. Young, M. E. Dean, J. S. Plank and G. S. Rose
title        A Review of Spiking Neuromorphic Hardware Communication Systems
journel      IEEE Access
volume       7
pages        135606-135620
year         2019
doi          10.1109/ACCESS.2019.2941772
keywords     {Licenses;Communication protocols;field programmable gate arrays;
             interconnect;network on chip;neuromorphic;
             spiking network communication; spiking neural network;
             very large scale integration}
url          http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8843969&
             isnumber=8600701
```

Citation Information—Bibtex:

```
@Article{8843969,
  author   = {Young, A. R. and Dean, M. E. and Plank, J. S. and Rose, G. S.},
  title    = {A Review of Spiking Neuromorphic Hardware Communication Systems},
  journal  = {IEEE Access},
  year     = {2019},
  volume   = {7},
  number   = {},
  pages    = {135606-135620},
  doi      = {10.1109/ACCESS.2019.2941772},
  keywords = {Licenses;Communication protocols;field programmable gate arrays;interconnect;
             network on chip;neuromorphic;spiking network communication;
             spiking neural network;very large scale integration},
  ISSN     = {},
  month    = {},
}
```

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 13, 2019, accepted September 5, 2019, date of publication September 18, 2019, date of current version October 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2941772

A Review of Spiking Neuromorphic Hardware Communication Systems

AARON R. YOUNG, (Student Member, IEEE), MARK E. DEAN, (Fellow, IEEE), JAMES S. PLANK, (Member, IEEE), AND GARRETT S. ROSE, (Member, IEEE)
Department of Electrical Engineering and Computer Science, The University of Tennessee Knoxville, Knoxville, TN 37996, USA

Corresponding author: Aaron R. Young (ayoung48@vols.utk.edu)

This work was supported in part by The University of Tennessee (UT) Science Alliance Joint Directed Research and Development Program, and in part by the Air Force Research Laboratory under Agreement FA8750-19-1-0025.

ABSTRACT Multiple neuromorphic systems use spiking neural networks (SNNs) to perform computation in a way that is inspired by concepts learned about the human brain. SNNs are artificial networks made up of neurons that fire a pulse, or spike, once the accumulated value of the inputs to the neuron exceeds a threshold. One of the most challenging parts of designing neuromorphic hardware is handling the vast degree of connectivity that neurons have with each other in the form of synaptic connections. This paper analyzes the neuromorphic systems Neurogrid, Braindrop, SpiNNaker, BrainScaleS, TrueNorth, Loihi, Darwin, and Dynap-SEL; and discusses the design of large scale spiking communication networks used in such systems. In particular, this paper looks at how each of these systems solved the challenges of forming packets with spiking information and how these packets are routed within the system. The routing of packets is analyzed at two scales: How the packets should be routed when traveling a short distance, and how the packets should be routed over longer global connections. Additional topics, such as the use of asynchronous circuits, robustness in communication, connection with a host machine, and network synchronization are also covered.

INDEX TERMS Communication protocols, field programmable gate arrays, interconnect, network on chip, neuromorphic, spiking network communication, spiking neural network, very large scale integration.

Surely there must be a less primitive way of making big changes in the store than by pushing vast numbers of words back and forth through the von Neumann bottleneck. Not only is this tube a literal bottleneck for the data traffic of a problem, but, more importantly, it is an intellectual bottleneck that has kept us tied to word-at-a-time thinking instead of encouraging us to think in terms of the larger conceptual units of the task at hand. Thus programming is basically planning and detailing the enormous traffic of words through the von Neumann bottleneck, and much of that traffic concerns not significant data itself but where to find it.
– John Backus, *1977 ACM Turing Award Lecture* [1]

I. INTRODUCTION

The human brain is an amazing computational machine, with many properties that surpass the capabilities of modern supercomputers. For comparison, contemporary processors operate in the multi-Gigahertz range with a power density of 100 W/cm^2 [2]. The human brain, on the other hand, which has unparalleled performance on cognitive and perceptual tasks, does so by running at an average firing rate of 10 Hz and with a very low power density of 10 mW/cm^2. The brain, which takes up a total size of 2 L, weighs 1.5 kg, and uses 20 W of energy, is difficult and inefficient to simulate using traditional computing architectures. Part of the difficulty in simulating the brain is that researchers disagree on the level of detail and which features must be modeled in order to successfully simulate the workings of the brain. Markram argues that detailed, biologically accurate models, based on first principles, are required to accurately model the cortex [3]. On the other hand, Izhikevich and Moehlis argue that the cortex can be accurately modeled by a simple system of coupled differential equations [4]. Regardless, many groups are interested in scaling up their models of neural networks to model neurons and synapses at a biologically realistic scale. In order to scale Ananthanarayanan, Esser, Simon, et al.'s simulation of Izhikevich neurons to a similar scale as a human brain using a supercomputer, Sharp, Galluppi, Rast, et al. estimates

The associate editor coordinating the review of this manuscript and approving it for publication was Woorham Bae.

that 2^{16} BlueGene processors would be needed, requiring 8.4 GW of power and running at a rate of "642 seconds for one second of simulation per Hz of spiking activity" [5], [6]. Clearly the brain is functioning dramatically different from traditional von Neumann architectures, requiring far less less space and power to operate. The brain is made up of approximately 86 billion computational elements, called neurons, which all operate independently, forming a massively parallel computer [2], [7]. The brain's unmatched performance comes from this immense parallelism, but each individual neuron is relatively simple on its own. The behaviour of neurons can be studied and understood as computational units independent from the other neurons. This allows each individual neuron to be viewed as a low-power biological core, analogous to a small RISC core. These neurons can be evaluated with multiple models, ranging in biological realism from the simplified leaky integrate-and-fire model to more biologically realistic models like the Hodgkin-Huxley model [2], [8]. The brain is clever because of its connectivity [9]. Each neuron has 1,000–10,000 synaptic connections to other neurons, forming a local memory for the neuron [2]. Just as there are a wide range of neuron models, there are also many synaptic models, with disagreement on the level of detail needed in the model. Simplified models treat synapses as point-to-point connections with a weight value associated with the connection's strength. This weight value can change and the magnitude of the weight can be viewed as information stored locally to the connected neurons. More complex synapse models include support for short term plasticity as well as modeling the complex dynamic, nonlinear, stochastic properties of biological synapses. At a high level, the brain-inspired, neuromorphic architectures have parallel processors (neurons), which only perform relatively simple operations, operate at low-frequency, and have binary output, and they have local, distributed memory, which is stored at the connection points to other elements in the form of the synaptic connections to other neurons. This architecture avoids the von Neumann bottleneck by colocating computation with memory, using simple components with simple communication, and by exploiting inherent and scalable parallelism in operation [10].

Different groups have tackled the challenge of designing a new type of computer, taking inspiration from our understanding of how the brain is able to achieve its great computational feats. Multiple issues must be faced to build a brain-inspired device, but perhaps the most challenging is how to tackle the dauntingly hard task of creating a system able to support the trillions of connections found in the brain. Building a neuromorphic computer is an interesting endeavor. Although the base components and their operations are easy to understand on their own, the difficulty comes from the vast number of these components found in the brain. When designing a large neuromorphic system, the challenges stem from trying to create a system with a similarly large number of neurons and synapses as the brain using semiconductor technology. This review paper discusses how different neuromorphic hardware designs handle spiking communication between neurons, and shows recent design trends found in neuromorphic communication. Section II introduces the neuromorphic projects reviewed in this paper. Section III discussions various considerations when dealing with packets in a spiking neural network (SNN). Section IV looks at different solutions to local high fan-out and fan-in connections. Section V considers solutions to supporting global synaptic connections. Prior papers have also compared various state-of-the-art neuromorphic systems. Recent notable papers include: "Large-Scale Neuromorphic Spiking Array Processors: A Quest to Mimic the Brain" which reviews details on the various design strategies of neural processors; "Low-Power Neuromorphic Hardware for Signal Processing Applications" summarizes the operation of the brain and recent neuromorphic systems with an emphasis on signal processing; "A Survey of Neuromorphic Computing and Neural Networks in Hardware" is an exhaustive review of research conducted in neuromorphic computing since the inception of the term; "Spiking Neural Networks Hardware Implementations and Challenges: A Survey" which surveys state-of-the-art spiking neuromorphic hardware and current trends in algorithm elaboration [11]–[14]. This review paper is unique from prior work since it focuses on discussing and comparing the communication systems used by spiking neuromorphic hardware.

II. NEUROMORPHIC PROJECTS OVERVIEW

Initially, neuromorphic hardware design was only done by research institutions, but as academics have shown the potential of these brain-inspired models, corporations have started developing research chips, and partnering with research institutions to best design and utilize these new chips. Neuromorphic computing, and spiking neural networks in particular, is a growing field with many avenues of future research available.

A. STANFORD UNIVERSITY NEUROGRID & BRAINDROP

Researchers at Stanford University have created two separate neuromorphic hardware designs. The first design is a mixed-analog-digital system called Neurogrid, which is able to provide computational neuroscientists with the capability to perform biological real-time simulations of the brain with millions of neurons and billions of synaptic connections [15]. Neurogrid is made up of analog neurons placed inside a 256×256 array fabricated in a 180-nm CMOS to make a Neurocore. To build the full Neurogrid system, 16 of the Neurocore chips are placed on a board and arranged into a tree structure. Fig. 1 shows the complete Neurogrid circuit board.

Their second design is called Braindrop [17]. Braindrop, like Neurogrid, is a mixed-analog-digital design; however, unlike Neurogrid, Braindrop is designed to be programmed at a high level of abstraction. Braindrop uses the Neural Engineering Framework(NEF) as the theoretical underpinning for the abstractions used to hide the heterogeneity found when

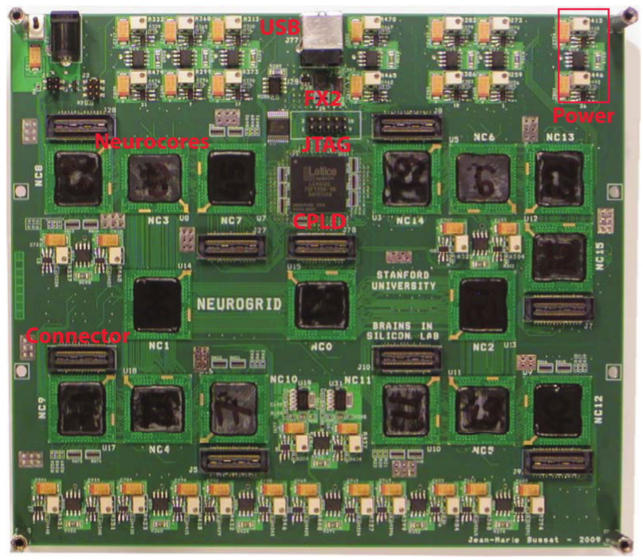


FIGURE 1. Neurogrid circuit board with 16 packaged Neurocore chips. The CPLD and FX2 enable a USB connection between a traditional computer and Neurogrid [16].

designing with analog neurons. Braindrop's programming is unique because computations are specified as coupled nonlinear dynamical systems and an automated procedure is used to synthesize the systems to the hardware. Braindrop is fabricated in 28-nm FDSOI process and integrates 4,096 neurons onto a single Braindrop chip. In the future, multiple Braindrop cores will be integrated to build a larger Brainstorm chip.

B. HUMAN BRAIN PROJECT BRAINSCALES & SPINNAKER

The next group of neuromorphic systems all come from the Human Brain Project (HBP) [18]. HBP is funded by the European Union with the goal of "building a research infrastructure to help advance neuroscience, medicine and computing" [19].

BrainScaleS is a mixed-analog-digital waferscale neuromorphic hardware system developed by a collaboration of research groups including the University of Heidelberg and the Technische Universität Dresden [20], [21]. BrainScaleS builds on the work completed in the FACETS project [22]. The waferscale integration technology developed for BrainScaleS makes it possible to utilize an entire 20 cm wafer for a very-large-scale neuromorphic system with 40 million synapses and up to 180 thousand neurons. The BrainScaleS system is built up by implementing many analog neuron circuits and their synapses in a structure called the Analog Network Core (ANC). The ANC was fabricated using 180-nm CMOS to create a High Input Count Analog Neural Network (HICANN). 352 HICANN chips are able to fit on a single wafer. A diagram of the complete wafer-scale BrainScaleS system is shown in Fig. 2.

A second generation of the BrainScaleS system is being designed and was revealed at the NICE Workshop 2018 [24]. BrainScaleS-2 uses a more complex neuron model which

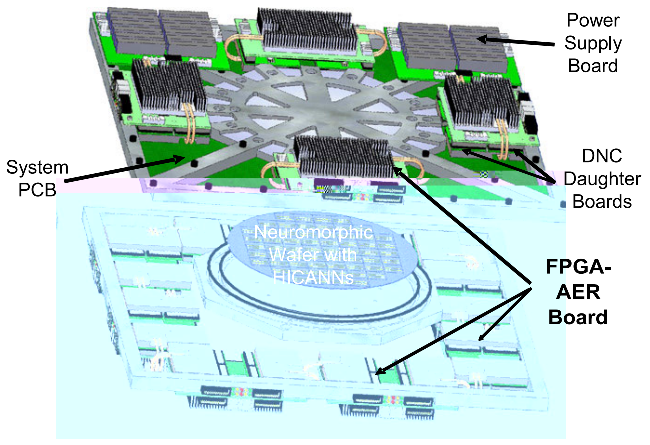


FIGURE 2. Diagram showing the complete BrainScaleS system [23].

supports nonlinear dendrites and structured neurons, along with other features. They also added the ability to use the neurons in a perceptron mode, where the neurons behave like traditional perceptrons and can be used to build non-spiking convolutional networks. This feature will also allow BrainScaleS-2 to combine both spiking neurons and perceptrons in the same experiment.

Also part of the Human Brain Project, researchers at the University of Manchester are working on a large digital neuromorphic system called SpiNNaker [25]. Instead of building custom neuron circuitry, SpiNNaker simulates the brain in real-time by connecting together over one million ARM processors. The ARM processors allow SpiNNaker to model a billion spiking neurons with biologically realistic connectivity (1,000–10,000 synapses per neuron) with 1 ms per step of simulation. Eighteen homogeneous ARM968 processors are integrated into one chip multiprocessor (CMPs) fabricated in 130-nm CMOS. Sixteen processors are used for simulation, one processor for administration, and one processor is a backup in case a processor is faulty. The full system is constructed of 2^{16} CMPs connected in a two-dimensional toroidal mesh. A single SpiNNaker board contains 48 CMPs and is shown in Fig. 3.

The first version of SpiNNaker is only able to simulate 1% of the human brain. The Second generation, named SpiNNaker2, aims to be able to simulate the entire brain [27]. They plan to achieve this feat by scaling-up the design of the previous generation. SpiNNaker2 will have 144 ARM MF4 cores per CMP fabricated in the modern 22FDX process. Additionally, SpiNNaker2 will include new features such as dynamic power management, floating-point support, synchronous memory sharing to neighboring cores, multiple-accumulate accelerators, and other numeric accelerators.

C. TRUENORTH

The TrueNorth neuromorphic platform has recently been developed by IBM as part of the Defense Advanced Research Projects Agency (DARPA) SyNAPSE program [2].

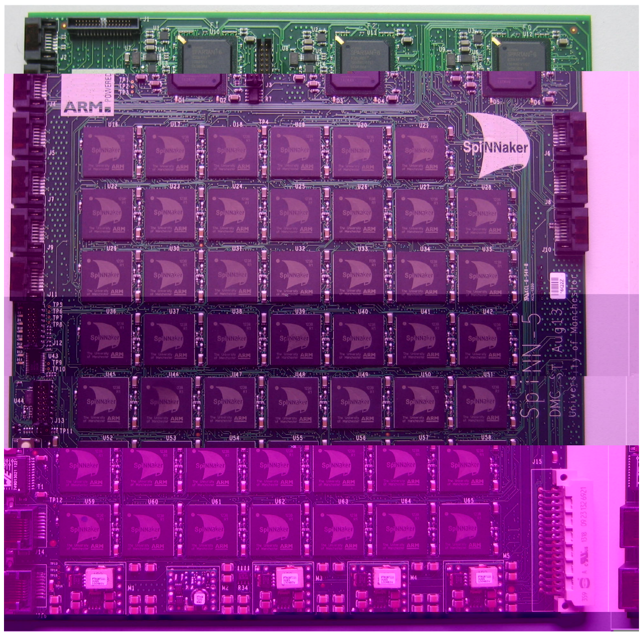


FIGURE 3. SpiNNaker circuit board, which is a building block for the SpiNNaker machine, contains 48 chips with a total of 864 ARM processors [26].

A single TrueNorth chip is composed of 4,096 neurosynaptic cores, with each core bringing together memory (''Synapses''), processors (''Neurons''), and communication (''Axons''), fabricated in IBM's 45-nm SOI process [28]. Each core implements 256 digital integrate-and-fire neurons with 1024 axonal circuits for input connectivity organized as an Static Random Access Memory (SRAM) crossbar [29]. A single TrueNorth chip contains 5.4 billion transistors, and implements 1 million digital neurons and 256 million synapses, tightly integrated in an event-driven design [2]. Multiple chips have also been combined into a larger 16-chip NS16e platform to allow for the simulation of 16 million neurons and 4 billion synapses.

D. LOIHI

Intel has also recently developed a digital neuromorphic research chip known as Loihi [30], [31]. Loihi has a unique programmable microcode learning engine for on-chip SNN training. Along with the 128-neuromorphic cores found on the chip, there are 3 Lakemont cores which help with advanced learning rules and with managing the neuromorphic cores. Loihi is fabricated in Intel's 14-nm process. The 128-neuromorphic cores implement 130,000 artificial CUBA leaky-integrate-and-fire neurons and 130 million synapses. The Loihi design supports scaling up to 4,096 on-chip cores and 16,384 chips.

E. DARWIN

Research groups at Zhejiang University and Hangzhou Dianzi University in China have created the Darwin Neural Processing Unit (NPU), which is targeted for resource

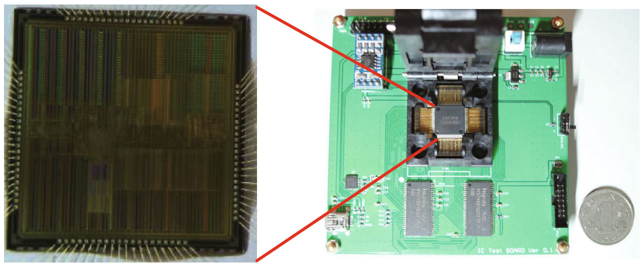


FIGURE 4. Darwin chip die and demonstration PCB board [32].

constrained embedded applications [8], [32]. The NPU constrains 8 physical neurons on the chip; but each neuron can be used to simulate 256 logical neurons with time multiplexing, resulting in the 2048 logical neurons for the entire chip. Each neuron is implemented with digital logic and can be connected arbitrarily to any other neuron resulting in a theoretical max of 4,194,304 synapses. In practice, however, the max number of synapses is limited by the size of the external memory used to store synapse information. This design was originally prototyped on a Field Programmable Gate Array (FPGA) and then fabricated in SMIC's 180 nm process. The complete system also includes a RISC CPU to create a complete neuromorphic System on Chip (SoC). The Darwin chip die and demonstration board is shown in Fig. 4.

F. DYNAP-SEL

Researchers at the University of Zurich in Switzerland, in the lab of Giacomo Indiveri, have created a novel mixed-signal multi-core architecture for neuromorphic processors which combines the advantages of the robustness of asynchronous digital logic for communication with the efficiency and dynamics of analog circuits for computation. The group has designed and fabricated the Dynamic Neuromorphic Asynchronous Processors (DYNAPs) in a 180 nm CMOS process; then they scaled up the design to a 28 nm FDSOI process with Dynamic Neuromorphic Asynchronous Processor with Scalable and Learning (Dynap-SEL) [11], [33].[1] DYNAPs and Dynap-SEL chips both have four neural processing cores. Each core implements 256 analog Adaptive-Exponential Integrate and Fire (AdExp-I&F) neurons arranged in a 16×16 grid. Each neuron has 64 programmable synapses with a max fan-in of 64 connections and a max fan-out of 4 k connections. Each synapse comprises circuitry to model biophysically realistic dynamics, including N-Methyl-D-Aspartate like voltage-gating, leak, spike-frequency adaptation, sodium activation resulting in positive feedback, potassium channels resulting in negative feedback, and refractory periods, as well as other dynamic models. Dynap-SEL has an additional fifth core which has 1×64 analog neurons with 64×128 plastic synapses with on-chip learning and 64×64 programmable

[1]Note: Per available sources, Dynap-SEL is a second improved version of DYNAPs, implemented in a smaller process with additional features. DYNAPs is discussed in [33]; Dynap-SEL is discussed in reference [11], along with other systems.

synapses. Dynap-SEL's fifth core also has incensed potential fan-in and fan-out, since up to eight rows of synapses can be merged together, at the expense of decreasing the active neuron count, to achieve a max fan-in of 1 k plastic synapses and 512 non-plastic synapses per neuron, for a network of 8 usable neurons. DYNAPs has been scaled to a PCB board which hosts 9 chips. Dynap-SEL supports integration into a chip array of up to 16 × 16 chips. DYNAPs and Dynap-SEL use the same routing architecture, so when the routing of DYNAPs is discussed, know that Dynap-SEL functions similarly.

III. SNN PACKETS

The communication patterns found with spiking neuromorphic networks (SNN) are different from the patterns of communication found in traditional computing. With traditional memory transfer, there is a greater emphasis placed on transferring large amounts of data with a high bandwidth to a cache near the CPU, and then hoping that the local cache has all the information needed to perform the computation without a cache miss. With traditional computing, CPU performance has been outpacing memory throughput and speed, resulting in the need for various tricks to guess what information will be needed by the processor ahead of time.

SNN communication is completely different. All of the memory is local in the form of synaptic weights, which allows the memory to be located alongside the computation elements. The information that is communicated in SNNs is simply the presence of a fire event, or spike. Spike information is presented in a streaming and online manner, contrasted with the batch mode used by large dense transfers. "A biological neuron firing is a pure asynchronous event which carries no information other than that it has happened [9].[2]" The power of the spiking information lies in the timing of the spike; all information is conveyed by the presence, frequency, and timing of these spiking events. SNNs communication is inherently event-driven and asynchronous. A biological spike only happens as a response to some other event, and the spike is sent to other neurons without the present of any global clocks. Accumulation of charge in each neuron only happens as a response to an input event occurring. When the accumulated charge exceeds a threshold, an electrical pulse is sent down the axon to the synapse, which communicates via chemical signaling to dendrites of the next neuron [35]. This chemical signaling mechanism is the local memory for the strength of the connection.

This all puts a different strain on communication systems that transmit these types of packets. The emphasis is now on the timing of events that contain a very small amount of information. Luckily, biological neurons only fire at around 10 Hz, CMOS communication occurs much faster than this, operating in the MHz and GHz ranges. This speed discrepancy between the firing rate of the neurons and the signaling rate of CMOS wires enables the use of time multiplexing to bundle many fire events into a single communication channel. Biological neurons have a physical connection for each spiking path in the form of the long axon in the neuron which can stretch as far as a meter in humans [36]. With CMOS technology, having wires that can change their connectivity like the brain is unfeasible. Therefore, the connections are made with various communication channels and routing methods that allow for flexible, reconfigurable, virtual connections over a fixed set of wiring.

[2]There are other neuromorphic systems which represent neurons and synapses as separate hardware elements. In this case, the communication network will need to also be able to send weighted information for post-synaptic events [34].

A. AER REPRESENTATION

Spikes are encoded into packets through Address-Event Representation (AER). With traditional AER the only information included in a packet is the address of the firing element [37] and optionally the time that the event occurs. These streams of address information are then multiplexed onto an asynchronous digital bus; routing of the packets is next performed based on the address of the source. When the time is not included in the packet, the time of the event is implicit based on when the packet arrives at the destination. The delay in communication in this case is considered inconsequential, since it is much lower than the firing rate of the neurons. Both SpiNNaker and Darwin use this simple AER representation for events. This simple AER format is advantageous since communication boards can be developed to route, analyze, record, and insert AER packets into a network of multiple neuromorphic devices [38]. Alternatively, the ID of the destination, or routing information can be sent instead of the ID of the source. This is done when the structure is set up so that the fire travels to a destination axon, such as in the case of TrueNorth. This variation of the traditional AER format comes down to how the packets are routed, in addition to how connectivity is handled.

B. ROUTING METHOD

As alluded to in the previous section, there are two main routing methods employed by the neuromorphic hardware: Source routing, where the packets are routed based on the source of the fire event, and destination routing, where the packets are routed based on the destination to which the fire event is traveling. There are two main packet types, multicast and point-to-point. With multicast, a single packet is delivered to multiple destinations; with point-to-point, a single packet is sent to a single destination. If the router supports multicast packets, then neurons with a high fan-out can be handled efficiently by the router. If instead the routing fabric only supports point-to-point packets, then a neuron with fan-out will have to send a new packet for each destination.

Source routing is advantageous since multicast routing can be easily implemented by allowing each router to steer and duplicate a packet based on its own routing table stored at each router. Neurogrid, SpiNNaker, BrainScaleS, and Darwin all use source-based routing with the routing tables for each router stored in a large off-chip memory [8], [9], [23], [39].

The router looks up the destinations the packet should be sent to and forwards the packet to the correct neighbors. The average access frequency for a particular memory location is low since a specific synapse in a large SNN has a very low probability of being activated at any give time [8]. SpiNNaker uses a special ternary Content Addressable Memory (CAM) for route lookup [9]. The identifier is compared under a mask to all the keys in the lookup table to get hit or miss information about which output links the packet should be sent. There is also a 'default' route which sends the packet along the same direction it was traveling to help save on the memory required for the routing. The BrainScaleS also has a timestamp-based prioritization mechanism based on the time the events occur [23]. This allows BrainScaleS to achieve lower latency and higher bandwidth for pulse routing as well as allowing configurable event delays.

The downside of source routing is that a large off-chip memory is needed to store the routing information for each source ID and the number of synaptic connections is limited by the amount of memory available for these routing tables. Furthermore, accessing external memory is slow, and doing so greatly decreases the speed at which packets can be routed. An alternative approach is to route the packets by destination. In this case, the source neuron knows the destination of a limited number of destination elements and encodes the destination as part of the packet. This allows the connectivity information to be stored with the sending neurons and doesn't require a large off-chip memory. Both TrueNorth and Loihi use destination-based routing to route packets [2], [30]. This allows the routers to be simple dimension order routers, which direct packets based on information found in their headers. TrueNorth only supports unidirectional messages sent from a source neuron to a receiving axon. This method works since the receiving axon can then connect to any of the 256 local neurons within the same neurosynaptic core [40]. TrueNorth's routers connect with its own core and its four neighboring cores, creating a two-dimensional mesh network. Each packet then caries a delta-x and delta-y address of the destination core, a destination axon index, and a destination time for the spike to be integrated. The packets are routed first in the horizontal direction until delta-x is 0. Then the delta-y field is dropped from the packet, and the packet is routed along the vertical direction until it arrives at the destination core. Loihi also uses dimension order routing on a two-dimensional grid, but supports additional features to relax the connectivity constraints placed on the programmer [30]. These features include sparse network compression, core-to-core multicast, variable synaptic formats, and population-based hierarchical connectivity. Loihi's Network on Chip (NoC) only supports unicast distributions; however, multicast spikes can be sent by sending multiple unicast packets. The routing of packets is still done by using memory resources local to the core, without depending on off-chip memory. This results in various network mapping constraints, including the max number of neurons per core, and the max number of fan-in and fan-out connections.

DYNAPs uses a mixture of different routing methods with its mixed tag-based shared-addressing scheme. The main idea of this scheme, is "to introduce different clusters with independent address spaces, so that it is possible to re-use the same tag ids for connecting neurons among each other, without loss of generality [33]." This scheme divides neurons into clusters, and the communication between the source and destination neurons is dividend into two phases. In the first phase, the packets use the destination-address of an intermediate node to route the packet to that intermediate node via point-to-point routing. Then in the second phase, the intermediate node broadcasts the tag stored in the packet to every neuron in the cluster. Each neuron then checks the tag against the tags stored in a Content Addressable Memory (CAM) to see if the neuron is connected to the source neuron with the same tag. This use of tags allows the tag addresses to be shared among source and destination neurons from different clusters, which results in there being less memory required to store the connection information. This two-stage tag-based routing scheme is shown in Fig. 5. In the hardware implementation of this scheme, the clusters are conveniently set to be the neurons contained in a single core of the DYNAPs chip. The first phase routing is done with R1, R2, and R3 routers with the intermediate nodes being the R1 routers at the destination core. The second phase is carried out by the R1 routers at the destination core broadcasting the packets to all the neurons in the destination core. So when an event packet is sent from a neuron, it goes to the local core's R1 router, where it is either routed to the same core, to a core on the same chip via R2 routers, or to a core on a different chip via both R2 and R3 routers. The R1 router either passes the packet to an R2 router or uses source-address routing to broadcast a multicast packet to all the neurons in the core. R2 routers use absolute destination-address routing to route the packet to the correct core. R3 routers use relative destination-address routing to route the packet to a destination chip at position $(\Delta x, \Delta y)$ via dimension order routing. Once on the correct chip, the R2 routers are used to route to the correct core. The tag based addressing scheme reduces memory requirements enough to allow the memory used to store connection information to be distributed across the cores and routers in embedded asynchronous SRAM and CAM memory cells. The SRAM cells are located in the R1 routers and store the source memory required to define the point-to-point connections in phase one of routing. The lines in the source memory contain the tag of the source neuron and the address of the intermediate node. The CAM cells are located in the destination synapses and store the target memory that is used for phase two of the routing. The lines in the target memory store the tags of the neurons the synapses are subscribed to.

Since all the neuromorphic systems have a limitation on the maximum communication bandwidth they can supply, many of the systems use a network compiler to reduce the amount of communication traffic by mapping neurons that are connected to each other to the same core, thus reducing

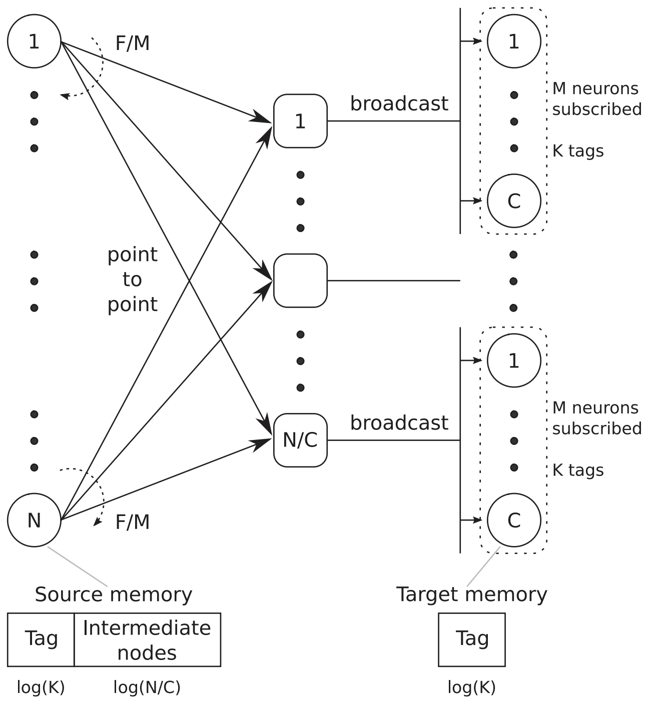


FIGURE 5. **The two-stage routing scheme used by DYNAPs. The connections of *N* neurons, each with a fanout of *F*, is implemented by first using point-to-point communication to send the source tag to *N*/*C* intermediate nodes. This reduces the point-to-point fan-out to *F*/*M*. The intermediate nodes then broadcast the source's tag to *C* neurons within a cluster, where *M* of these neurons are subscribed to the key. The number of unique keys used in a cluster is K. Note that the neurons on the right side are the same as the ones on the left, but are shown grouped into *N*/*C* clusters [33].**

the distance most of the packets have to travel. TrueNorth has created a mapping algorithm which uses a modified very large scale integration (VLSI) placement algorithm to place neurons onto cores and minimize the distances packets have to travel [2]. This greatly reduces the total number of hops that are necessary for the packets to reach their destination. Similarly, SpiNNaker has a PArtition and Configuration MANager (PACMAN) which provides utilities for SNN partitioning, placement, and routing. PACMAN is able to use a variety of partitioning and placement algorithms. One recent SNN spectral analysis based partitioning and placement algorithm is the GrapH Optimizer SpiNNaker Tool (GHOST). Essentially, GHOST creates an expanded neuron graph, uses clustering to group highly connected neurons, uses sub-clustering and fusion to reduce the neuron count groups that will fit on a single core, and uses Sammon mapping to place the high dimensional groupings onto the 2D mesh with legalization to fine tune the placement to make it valid [41]. Two organizational principles found in the brain are used to reduce network traffic: local dendritic trees within a pool of neurons and hierarchical axonal arbors between pools of neurons [39]. In the brain there are cortical columns and regions with many dense connections and these dense regions are connected with long-range cortical connections [40].

C. ASYNCHRONOUS CIRCUITS

A main feature of SNNs is their asynchronous and event-driven nature. Both of these features allow the brain to be very energy efficient, as energy is only expended when a spike arrives and no extra energy is expended for a synchronous clock. The neuromorphic components and packet routers are also designed using asynchronous VLSI circuit techniques. These circuit techniques remove the challenges of routing a synchronous clock across a very large chip and eliminate the power lost as a result of applying a clock to idle components. Martin and Nystrom state that in the future SoCs will no longer be able to operate under a single clock [42]. The variations across such a large chip will make it prohibitively expensive to attempt to manage the delays in a clock and other global signals. The solution is to use asynchronous circuit techniques which are delay insensitive and pass information using quasi-delay-insensitive (QDI) circuits. These circuits communicate with asynchronous handshake protocols without a shared global clock. Martin and Nystrom predict that future systems will be made entirely with asynchronous logic, or that they at the very least will have to be designed globally asynchronous and locally synchronous (GALS). Asynchronous VLSI circuits are defined with a high-level description language, for example the Communicating Hardware Processes (CHP) language, and then compiled into a circuit design using semantic-preserving program transformations. All the neuromorphic devices use asynchronous circuits designed in QDI design style based on Martin's synthesis procedure [42]–[44]. These asynchronous circuits are used to implement the NoC structures and routers, enabling them to conserve power when not in use and also handling the varying delays in the wires connecting the elements together.

Additionally, the number of pins available on the chips for external connections is greatly limited. This requires that the multiple asynchronous parallel buses found in the hardware must be serialized into a narrower stream, which uses fewer pins. In the case of SpiNNaker, inter-chip links convert 1-of-5 RTZ (return-to-zero) codes to 2-of-7 NRZ (non-return-to-zero) codes to reduce the number of I/O pins used and to save energy by reducing the amount of signal transitions required to transmit information [45]. TrueNorth uses asynchronous arbiters and merge-split blocks to combine multiple streams of packets into a single stream for sending the packets off-chip [2]. The single stream is then transmitted to neighboring chips via bundled-data asynchronous circuits to minimize the number of interface circuits and pins needed.

D. PACKET STRUCTURE

Another important consideration when designing the AER communication is the packet structure of the fire events and if the packets should be fixed length or variable length. Neurogrid supports variable length packets [15]. A Neurogrid packet is a sequence of 12-bit words that specify a route, address, arbitrarily long payload, and tailword, sent in that order. The tailword specifies the end of the packet. SpiNNaker packets are fixed width with an 8-bit header, a 32-bit content field, and an optional 32-bit data payload [9]. The content field is typically a key to identify the source neuron. TrueNorth packets contain a 9-bit delta-x, 9-bit delta-y, 4-bit delivery time, 8-bit destination axon index, and 2 debugging bits [2]. BrainScaleS's Packets are fixed width, with varying widths depending on the level in the communication hierarchy, but each packet is made up of a multiple of 24-bit pulse events [21]. Darwin packets contain the ID of the neuron that generated the spike and the timestamp of when the spike was generated [8]. DYNAPs packets are created by copying the line from the source SRAM that corresponds to the neuron which fired. Each 20-bit word is make up of a 10 bit address, a 6-bit routing header, and a 4-bit destination core (intermediate node) address. The routing header contains a 2-bit delta-x with 1-bit sign and a 2-bit delta-y with a 1-bit sign [33].

E. EVENT-BASED SENSORS AND ACTUATORS

Neuromorphic event-based sensors have been designed based on biological sensors. They seek to replicate the efficiency, robustness, and low-power consumption of their biological inspirations. Among these sensors are different implementations of a silicon retina [46] and a silicon cochlea [47]. Both of these sensors are event-based and only send information when there is a change in the environment. This allows these sensors to require less bandwidth than their traditional counterparts. Neuromorphic sensors send their information as a continuous stream of asynchronous spikes using AER packets. For example, the Asynchronous Time-based Imaging Sensor (ATIS) silicon retina sends illumination change packets with the location of the pixel, the polarity of the change in illumination, and the time when the event occurred [46]. Since both the sensors and the neuromorphic computers use AER encoded asynchronous packets, the neuromorphic processors can directly use information from the sensors for computation without needing an extra conversion step. If event-based actuators are used, then the output from the neuromorphic processor can also be used directly for control applications. An example of this is the AER-Robot, which uses event based encoders for input and Pulse-Frequency Modulation (PFM) to power DC motors, to create a neuromorphic closed-loop control system [48].

IV. LOCAL COMMUNICATION CHALLENGES

Communication can viewed from two different levels of organization: local on-chip communication within a neuromorphic core and global system communication across many cores. With both levels, one feature that helps with the design of the communication channels is that synapses can share wires among a group of neurons, since wires propagate signals much faster than biological axons [39].

Neurogrid uses a shared dendrite hybrid model to handle local connections [15]. With this model, each shared-synapse circuit is connected to neighboring neurons, mirroring the

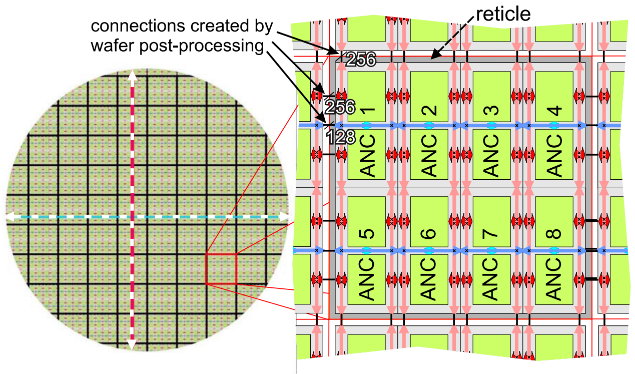


FIGURE 6. A BrainScaleS wafer with 56 complete reticles. The dashed arrows depict one vertical and horizontal bundle of inter-neuron connections. A single reticle is enlarged to show the arrangement of ANCs [22].

structure of overlapping dendritic trees found in biological neural networks. This structure allows Neurogrid to work well for modeling networks which require many neurons with mostly local connections, for example, modeling the neocortex.

In TrueNorth a 1024 × 256 bit SRAM cross bar memory is used to define the synaptic connections from 1024 axon wires to 256 dendrite wires [29]. Spike packets are routed to a particular axon, which can then be connected to up to 256 dendrites. The dendrites are then connected to the neurons for computation. The downside to a crossbar design is that if the synaptic connections are sparse, resources are wasted [39].

BrainScaleS handles local communication with their L1 routing interface. L1 routing is an asynchronous, serial, event-driven protocol, which operates at up to 2 Gbps and is used to interconnect Analog Network Chips (ANC) [20], [22], [23]. The ANCs are connected physically with a dense layer of horizontal and vertical wires that are added as post-processed metal interlinks on top of the wafer. These wires form a high-density pulse routing grid. Temporal multiplexing is used to allow each of the 256 Low-voltage differential signaling (LVDS) bus lanes to carry events from 64 pre-synaptic neurons by serial transmitting 6-bit neuron IDs. Sparse crossbar switches and repeaters are used to propagate the spikes across an arbitrary number of HICANN chips. Fig. 6 shows what this L1 routing looks like.

SpiNNaker uses the ARM AMBA (Advanced Microcontroller Bus Architecture) protocol for communication within a local clock domain [25].

DYNAPs handles local communication in the second phase of routing by using the R1 routers to broadcast the incoming packet to all of the neurons in the core. The neurons then use the tag to determine if they are sensitive to the fire event [33].

V. GLOBAL COMMUNICATION CHALLENGES

With global packet routing there are two routing schemes used: mesh and tree. With mesh routing, routers are connected to neighboring routers; the most common meshes are two-dimensional grids with connections in the four cardinal

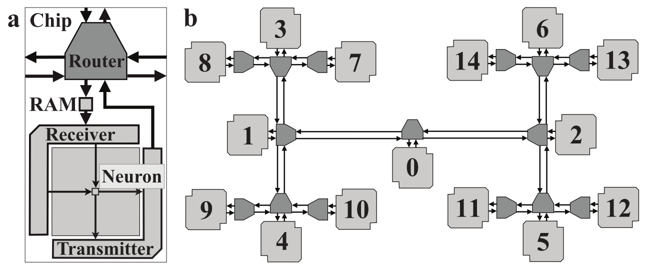


FIGURE 7. Neurogrid: a) Neuromorphic chip with integrated silicon neuron array, receiver, transmitter, RAM, and router. b) Fifteen-node binary tree. Each neuromorphic chip communicates with the others through on-chip routers and interchip links [16].

directions. Tree routing structures the routers in a hierarchy with a root node that repeatedly branches to connect to routers in the lower levels. Mesh routing is advantageous since it has a larger channel bisection, with more links in the network, resulting in higher throughput of packets [15]. However, mesh routing usually has a longer latency due to a larger number of hops for the packet to reach its destination. With a mesh, dimension-order routing is typically employed to prevent deadlock. Tree routing is advantageous when shorter latency is needed, since fewer hops are required to reach the destination; however, there are fewer connection paths, resulting in lower bandwidth. Deadlock-free multicast routing is easier to implement with tree routing by using up-down routing. With up-down routing, packets are first sent to a common root node, and then the packets are able to be duplicated to multiple child nodes on the downward routing phase. If the efficient multicast capability of tree routing can be utilized, then tree routing offers lower latency and higher effective throughput than mesh routing, and uses roughly two-thirds of the resources [15].

A. TREE

Neurogrid uses a multicast tree router connected in a binary tree. Up-down routing is used to prevent deadlock and support multicast packets. Fig. 7 shows a diagram of the Neurogrid chips and the tree routing structure. Braindrop uses a fractal H-tree for routing [49]. BrainScaleS's Layer 2 routing uses a hierarchical packet-based routing tree constructed with digital network chips (DNC). Eight high input count analog neural networks (HICANNs) are grouped into one reticle. The reticle is then connected to one DNC. Four DNCs are connected to an FPGA-AER board. The FPGA-AER boards are then connected to each other using the aurora protocol with 10 Gbps data rate on four parallel multi-gigabit lanes. Communication for the entire wafer requires 12 FPGA-AER boards. The FPGA-AER boards are connected via 1 or 10 Gbps Ethernet links to handle wafer-to-wafer communication. This hierarchical structure is shown in Fig. 8.

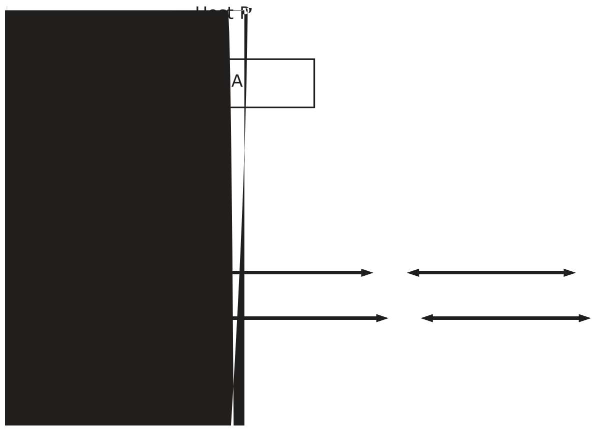
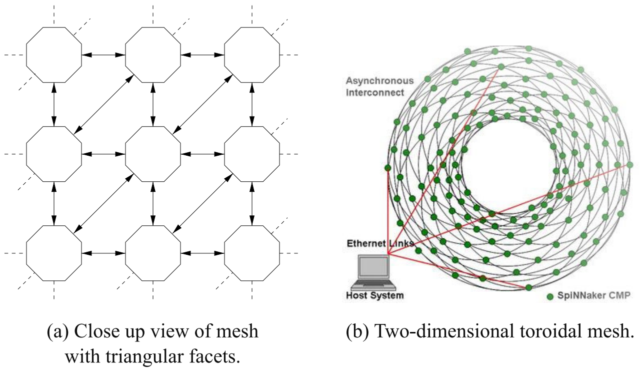


FIGURE 8. Diagram of BrainScaleS's hierarchical L2 routing [23].



(a) Close up view of mesh with triangular facets.

(b) Two-dimensional toroidal mesh.

FIGURE 9. The SpiNNaker system network connectivity [9], [25].

B. LARGE GRID

SpiNNaker connects the ARM chip multiprocessors with a two-dimensional toroidal mesh. Each CMP has six connections to neighboring chips forming triangular facets which support 'emergency routing' around a failed or congested link [25]. The TrueNorth chip tiles its 4,096 neurosynaptic cores into a two-dimensional array [2]. Dimensional routing is used to prevent network deadlock. Packets are routed on a first-come-first-served basis, where arbitration is used for packets that arrive at the same time. To guarantee that no packets will be dropped, back pressure is used to prevent new packets from arriving when the router is waiting to send outgoing packets. TrueNorth does allow hierarchical communication by sending a spike globally through the network using a single packet. The packet fans out to multiple neurons locally in the destination core. TrueNorth scales to beyond grid boundaries by combining the grid boundaries with native event-driven serializer/deserializer links [40]. Fig. 10 shows a diagram of TrueNorth chips connecting across chip bounds. Loihi connects all the cores and processors together in a many-core mesh [30], [31]. The edges of the chip have off-chip communication interfaces to allow Loihi to scale out to many other chips along the four planar directions.

C. HYBRID

Hybrid designs look to combine the benefits of the previous two approaches. DYNAPs combines hierarchical tree

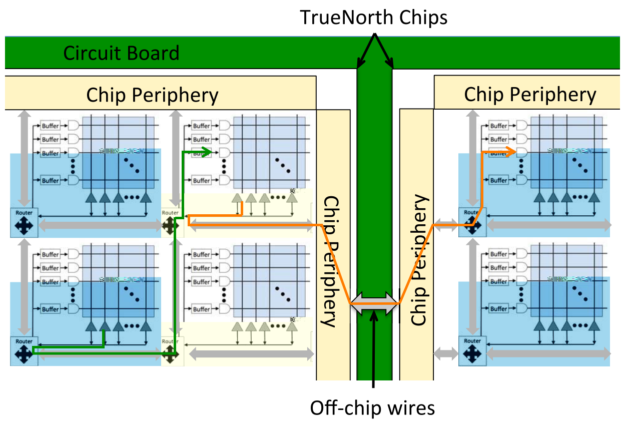


FIGURE 10. **TrueNorth cross-chip connectivity [2].**

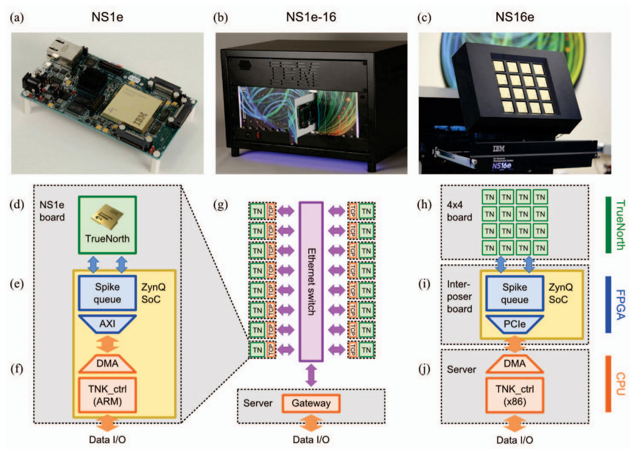


FIGURE 12. **TrueNorth NS1e, NS1e-16, and NS16e systems [40].**

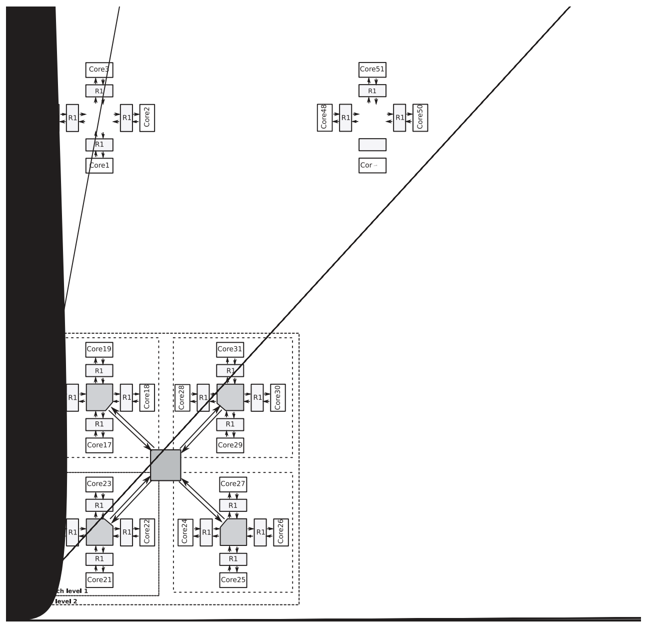


FIGURE 11. **DYNAPs hybrid hierarchical-mesh routing scheme example. Individual cores communicate via broadcast operations through the R1 routers. Groups of 4 cores are connected together via a level-1 R2 router. To communicate with cores in different groups, but on the same chip, level-2 or higher level R2 routers are used following a tree-based hierarchical routing scheme. R3 routers are used to communicate to different chips along the four cardinal directions using a 2D-mesh routing strategy [33].**

routing with a 2D-mesh in order to minimize the memory resources needed and maximize the network programmability and flexibility. In phase one of packet routing, point-to-point destination-address routing is conducted by the R2 and R3 routers. The R2 routers use a hierarchical tree to route packets within a chip. There can be multiple levels of R2 routers making up the tree. The R3 routers direct the packets among different chips arranged in a 2D mesh with relative destination-address routing [33]. A diagram of this mixed-mode hierarchical-mesh used by DYNAPs is shown in Fig. 11.

D. ROBUSTNESS IN LARGER COMMUNICATION

With any global communication network on the same scale as the ones found in neuromorphic processors, robustness of the system is of key importance as failures are almost guaranteed to occur. This section contains examples of different mechanisms employed by neuromorphic hardware to ensure the communication channels are robust to failure. SpiNNaker has emergency packet re-routing, which allows packets to be sent along an alternative route when a link is detected as failed or congested [25]. SpiNNaker uses acknowledgment packets between monitoring processors to verify that the packet was successfully sent and the communication channel is reliable [26]. BrainScaleS uses a cyclic redundancy check to find corrupted data in the packets [23]. TrueNorth is able to disable and route around faulty cores in the chip so that defects in the chip are hidden at runtime [40].

E. HOST CONNECTION

All the neuromorphic systems are able to connect to a traditional computer to enable the configuration, monitoring, and external signaling. Neurogrid connects to a host computer with USB via the help of a Cypress EZ-USB FX2LP. Neurogrid's software allows the user to specify the neuronal modules, control the simulation, and visualize the results from running the neural model in real time [15]. Darwin connects to an off-chip host PC using a USB to UART interface [32]. DYNAPs communicates with a host via an FPGA [33]. SpiNNaker uses multiple 100 Mbps or 1 Gbps Ethernet interfaces to connect to the host, and the monitoring cores on the SpiNNaker chips are used for application support and system monitoring [50], [51]. The smaller TrueNorth NS1e connects to an on-board SoC which functions as the host computer via an AMBA AXI interconnect [40]. The larger NS16e connects to a host computer via a single lane PCIe 2.0. There is also a NS1e-16 system which consists of 16 NS1e boards and a host server networked together over an Ethernet switch. These three TrueNorth systems are shown in Fig. 12.

F. SYNCHRONIZATION

Another important consideration is how to handle the synchronization of the network. A mentioned before, the brain is purely asynchronous, and spikes propagate in real time. This is the approach taken by Neurogrid, Braindrop, SpiNNaker, and BrainScaleS. The downside to not having an explicit synchronization method is that it is impossible to create a cycle-accurate, deterministic simulator of the neuromorphic hardware. This in turn makes it harder to construct and debug Neural Network designs. The alternative is to synchronize the cycle time for the elements, which makes it possible to define a deterministic behavior per cycle and support a cycle-accurate simulator.

There are two main approaches for synchronization. One is to have a fixed cycle time. The other is to allow cycles to have a variable length cycle time and employ other synchronization methods to progress to the next cycle. TrueNorth uses a fixed cycle time, where operation of TrueNorth occurs in two phases [29]. In the first phase, AER packets are routed among the cores. When the packets arrive, they modify the membrane potential of the connected neurons. In the second phase, a synchronization event (sync), which occurs every millisecond, is sent to all the cores. Upon receiving the sync signal, the neurons check to see if they should fire, and if so, they send their fire packets onto the network. The downside to using a fixed cycle length is that cycle length has to be longer than the time it takes to send all the packets, resulting in times of no chip activity. If the packets can't reach their destination in time, then a global error flag is set and the operation of the chip is no longer deterministic.

Loihi and Darwin both have a variable length cycle time. With this method the timestamp of the network is algorithmic time, and is unrelated to real-time. Loihi uses a mesh-level barrier sync to signify when all the packets have reached their destinations and the timestep can be advanced to the next cycle [30]. This asynchronous handshake provides a significant performance advantage since it eliminates needless idle time in the network and allows the network to run at the fastest speed possible. Fig. 13 shows a diagram of Loihi's mesh operation with barrier sync. The speed of the networks is now variable, and how long each cycle takes is set by the slowest component, which bottlenecks the performance of the system. In the case of Loihi, the max speed of the network is limited by the bandwidth of the on-chip router and how long it takes to propagate the fire packets through the network. Darwin also has variable cycle time and uses a time-multiplexing controller to progress the cycle, once the previous cycle is finished [8]. The downside to variable length cycle times is that it becomes harder to interface the network with real-time signals and perform real-time operations since real-time and simulation time are separated. The advantages, however, are that the simulation can run faster than real-time, or at least as fast as possible given the network activity, and they are deterministic in their operation.

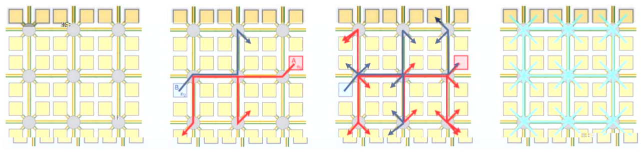


FIGURE 13. Barrier sync operation of Loihi. First box: initial idle state for time-step t (each square is a neurocore). Second box: neurons n_1 and n_2 in cores A and B fire and generate spike messages. Third box: all other spikes from time-step t are distributed to their destination cores. Fourth box: each core exchanges a barrier synchronization message with its neighbors and each core advances its algorithmic time-step to $t+1$ [30].

VI. SCALABILITY

Neuromorphic systems are generally designed to have great scaling potential. This allows the systems to either scaled up to biologically realistic sizes or scaled up based on the complexity of the application being deployed. Other neuromorphic systems target low-power or embedded systems and are purposely designed with less scaling in mind to save resources, since supporting large scaling sizes results in extra overhead in both packet size and storage required.

Neurogrid is designed to scale to 16 interconnected Neurocores on a single board [39]. Braindrop, which will be a single core in a larger scaled up Brainstorm chip, is architected to support a million-neuron multicore system [17].

The BrainScaleS system (NM-PM-1) was scaled up to contain twenty 8-inch full wafer systems [52], [53]. The system is stored in seven 19" racks. Five of the racks are used to store the neuromorphic wafer modules and the other two racks are used to store the power supplies and conventional control cluster.

The SpiNNaker system is designed to scale to very large sizes. The SpiNNaker chips are mounted on a board in a 48-node hexagonal array. Then 24 boards are assembled into a crate, with five crates stored in a single rack. High-speed serial cables are used to interconnect the racks [52]. "Virtually any number of racks may be interconnected to form a system of arbitrary scalability [54]." SpiNNaker machines are classified by the approximate number of processing cores in the system. A $10N$ machine has approximately 10^N processing cores. The current largest SpiNNaker machine is the 106 machine with ten 19" rack cabinets, each storing five 24-board crates [50], [55]. It has a total of 1,036,800 ARM processing cores, 921,600 of which are for application processing. The machine require a 100kW (approximate) 240V supply. The other machine sizes are subsets of this large system. The 102 machine is a single PCB with four SpiNNaker nodes. The 103 machine is one PCB with 48 SpiNNaker nodes. Twelve of these 48-node cards are combined into a crate to create the 104 machine. Five crates are used in a single 19" rack cabinet for the 105 machine. Ten rack cabinets are combined to make the 106 machine.

The SpiNNaker 2 project will further scale the system up by switching to a 22FDX process and embedding 144 ARM MF4 cores per chip [27].

Loihi is design to support up to 4,096 on-chip cores and 16,384 chips. The design is thus constrained because of

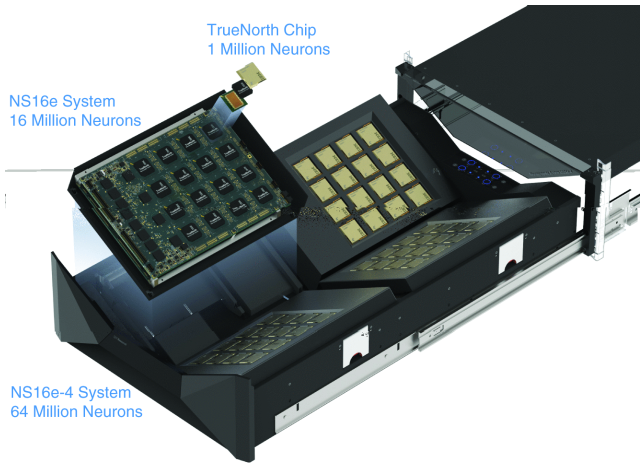


FIGURE 14. The NS16e-4 scaled-up TrueNorth evaluation system [57].

the design of the mesh protocol and hierarchical addressing scheme [30]. Intel recently announced the largest Loihi system yet—called Pohoiki Beach—which is comprised of 64 Loihi chips with over 8 million neurons. This chip is another milestone towards Intel's goal of scaling Loihi to 100 million neurons later this year with Pohoiki Springs, which is planned to contain 768 Loihi chips [56]. Loihi also is available in a small USB stick form factor, called Kapoho Bay, which incorporates 1 or 2 Loihi chips. Wolf Mountain is a research board with 4 Loihi chips. Nahuku is an FPGA expansion card which contains 8 to 32 Loihi chips [31].

TrueNorth chips support two kinds of scaling: scale-up and scale-out. Scale-up corresponds to integrating multiple TrueNorth chips onto a single PCB and using TrueNorth's native interchip asynchronous communication interface for the chips to communicate with each other. This allows the chips to communicate natively, forming a larger unified array of neuromorphic cores. Scale-out corresponds to connecting multiple boards together via standard networking hardware to form a neuromorphic computing cluster. The naming scheme of TrueNorth systems indicates how the system was scaled up from a base single chip design. They are named NS*A*e-*B* where *A* is the scale-up factor and *B* is the scale-out factor. Note that when *B* is one, it is left off. The largest TrueNorth system to date is the NS16e-4. This system has a scale-up factor of sixteen and a scale-out factor of four. The system is constructed by connecting four NS16e systems together with optical PCIe links within a 4-U rack-mounted standard drawer [57]. A picture of this system is shown in Fig. 14. Other TrueNorth chips using this same scaling convention are shown in Fig. 12. Additionally, IBM has deployed internally a NS1e-80, which is a cluster of 80 NS1e boards [57].

Darwin NPU is currently a smaller single chip system targeted for embedded applications; however, the NPU could also be used as a processing element for a NoC architecture. As a NoC core, Darwin could potentially scale up to millions of neurons on a chip, instead of the few thousand in the current single chip system [8].

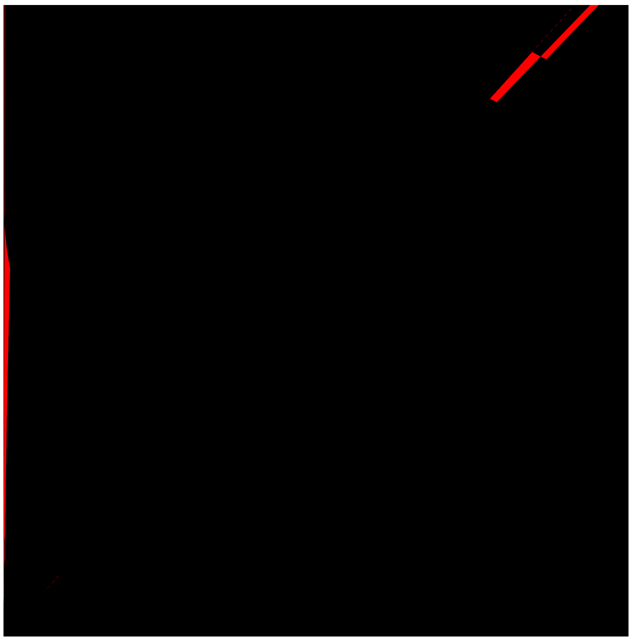


FIGURE 15. Graphical summary of neuromorphic hardware communication systems. The top half of this figure summarizes the routing schemes used by the neuromorphic systems, and the bottom half summarizes the routing methods.

Dynap-SEL is design to be able to be integrated in an array of up to 16 × 16 chips with all-to-all connectivity among the neurons [11]. Part of the scaling potential of DYNAPs/Dynap-SEL comes from its communication scheme, which allows memory requirements to scale with the number of neurons in a way drastically lower than other standard routing schemes [33].

VII. CONCLUSION

One of the most challenging parts of designing a large scale neuromorphic system is designing a scalable spiking communication network, which is able to keep up with the massive connectivity requirements found in these systems. Table 1 and Fig. 15 summarize the different communication systems found in neuromorphic hardware. These neuromorphic systems are able to efficiently scale up to larger sizes than von Neumann computers can, since they store information with the computation element, which eliminates the von Neumann bottleneck. Neuromorphic systems can be scaled-out with a loose coupling of boards together. For example, the NS1e-16 system from TrueNorth loosely couples 16 single chip boards over an Ethernet network [40]. They can alternatively be scaled-up by tightly integrating multiple systems together, such as the NS16e TrueNorth system, which tightly integrates 16 chips into a 4 × 4 grid using native tiling. The continued scaling potential of neuromorphic systems is made possible by the exponential decay in hop distance and bandwidth observed in biological neurons [40]. Biologically realistic network topologies have dense clusters of connectivity that are connected together by fewer long range

TABLE 1. Summary of neuromorphic hardware communication systems.

[redacted]

connections [33]. These long range connections also typically have longer signal delay, since the spike must travel a further distance. These two components make it easier and possible to design scalable, large neuromorphic systems with similar topologies. One of the major uses of neuromorphic systems is to use them as co-processors or accelerators to perform computations that are difficult or inefficient to evaluate on a tradition system. Sawada *et al.* predict that future neuromorphic systems will become a key component of exascale systems [40].

ACKNOWLEDGMENT

The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not

[14] M. Bouvier, A. Valentian, T. Mesquida, F. Rummens, M. Reyboz, E. Vianello, and E. Beigne, "Spiking neural networks hardware implementations and challenges: A survey," *J. Emerg. Technol. Comput. Syst.*, vol. 15, no. 2, Apr. 2019, Art. no. 22. doi: 10.1145/3304103.

[15] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J.-M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proc. IEEE*, vol. 102, no. 5, pp. 699–716, May 2014.

[16] 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.

[17] A. Neckar, S. Fok, B. V. Benjamin, T. C. Stewart, N. N. Oza, A. R. Voelker, C. Eliasmith, R. Manohar, and K. Boahen, "Braindrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model," *Proc. IEEE*, vol. 107, no. 1, pp. 144–164, Jan. 2019. doi: 10.1109/JPROC.2018.2881432.

[18] H. Markram, K. Meier, T. Lippert, S. Grillner, R. Frackowiak, S. Dehaene, A. Knoll, H. Sompolinsky, K. Verstreken, J. DeFelipe, S. Grant, J.-P. Changeux, and A. Saria, "Introducing the human brain project," in *Proc. 2nd Eur. Future Technol. Conf. Exhib. (FET)*, vol. 7, Dec. 2011, pp. 39–42. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050911006806

[19] Human Brain Project. *Short Overview of the Human Brain Project*. Accessed: Apr. 10, 2019. [Online]. Available: https://www.humanbrainproject.eu/en/about/overview/

[20] J. Schemmel, D. Briiderle, A. Griibl, M. Hock, K. Meier, and S. Millner, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *Proc. IEEE Int. Symp. Circuits Syst.*, May/Jun. 2010, pp. 1947–1950. doi: 10.1109/ISCAS.2010.5536970.

[21] S. Scholze, S. Schiefer, J. Partzsch, S. Hartmann, C. G. Mayr, S. Höppner, H. Eisenreich, S. Henker, B. Vogginger, and R. Schüffny, "Vlsi implementation of a 2.8 Gevent/s packet-based AER interface with routing and event sorting functionality," *Frontiers Neurosci.*, vol. 5, p. 117, Oct. 2011. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnins.2011.00117. doi: 10.3389/fnins.2011.00117.

[22] J. Schemmel, J. Fieres, and K. Meier, "Wafer-scale integration of analog neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 431–438. doi: 10.1109/IJCNN.2008.4633828.

[23] S. Scholze, H. Eisenreich, S. Höppner, G. Ellguth, S. Henker, M. Ander, S. Hänzsche, J. Partzsch, C. Mayr, and R. Schüffny, "A 32 GBit/s communication SoC for a waferscale neuromorphic system," *Integration*, vol. 45, no. 1, pp. 61–75, Jan. 2012.

[24] J. Schemmel, "Towards the second generation brainscales system," in *Proc. NICE*, Portland, OR, USA, 2018. [Online]. Available: https://niceworkshop.org/nice-2018-agenda/

[25] S. Furber and A. Brown, "Biologically-inspired massively-parallel architectures—Computing beyond a million processors," in *Proc. 9th Int. Conf. Appl. Concurrency Syst. Design*, Jul. 2009, pp. 3–12. [Online]. Available: https://eprints.soton.ac.uk/270985/1/PID871138.pdf

[26] F. Barchi, G. Urgese, A. Siino, S. Di Cataldo, E. Macii, and A. Acquaviva, "Flexible on-line reconfiguration of multi-core neuromorphic platforms," *IEEE Trans. Emerg. Topics Comput.*, to be published. doi: 10.1109/TETC.2019.2908079.

[27] S. Hoppner, "Spinnaker2—Towards extremely efficient digital neuromorphics and multi-scale brain emulation," in *Proc. NICE*, Portland, OR, USA, 2018. [Online]. Available: https://niceworkshop.org/nice-2018-agenda/

[28] A. Amir, P. Datta, W. P. Risk, A. S. Cassidy, J. A. Kusnitz, S. K. Esser, A. Andreopoulos, T. M. Wong, M. Flickner, R. Alvarez-Icaza, E. McQuinn, B. Shaw, N. Pass, and D. S. Modha, "Cognitive computing programming paradigm: A corelet language for composing networks of neurosynaptic cores," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–10.

[29] 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. doi: 10.1109/CICC.2011.6055294.

[30] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018. doi: 10.1109/MM.2018.112130359.

[31] Wikichip. *Loihi—Intel*. Accessed: Aug. 8, 2019. [Online]. Available: https://en.wikichip.org/wiki/intel/loihi

[32] J. Shen, D. Ma, Z. Gu, M. Zhang, X. Zhu, X. Xu, Q. Xu, Y. Shen, and G. Pan, "Darwin: A neuromorphic hardware co-processor based on spiking neural networks," *Sci. China Inf. Sci.*, vol. 59, no. 2, pp. 1–5, Feb. 2016. doi: 10.1007/s11432-015-5511-7.

[33] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs)," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 106–122, Feb. 2018. doi: 10.1109/TBCAS.2017.2759700.

[34] M. E. Dean, C. D. Schuman, and J. D. Birdwell, "Dynamic adaptive neural network array," in *Proc. 13th Int. Conf. Unconventional Comput. Natural Comput. (UCNC)*. London, U.K.: Springer, Jul. 2014, pp. 129–141.

[35] Queensland Brain Institute. *Action Potentials and Synapses*. Accessed: Nov. 9, 2017. [Online]. Available: https://qbi.uq.edu.au/brain-basics/brain/brain-physiology/action-potentials-and-synapses

[36] E. H. Chudler. *The Hows, Whats and Whos of Neuroscience*. Accessed: Apr. 11, 2019. [Online]. Available: https://faculty.washington.edu/chudler/what.html

[37] 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. doi: 10.1109/ISCAS.2008.4541501.

[38] H. K. O. Berge and P. Hafliger, "High-speed serial AER on FPGA," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 857–860. doi: 10.1109/ISCAS.2007.378041.

[39] K. Boahen, "A neuromorph's prospectus," *IEEE Comput. Sci. Eng.*, vol. 19, no. 2, pp. 14–28, Mar./Apr. 2017. doi: 10.1109/MCSE.2017.33.

[40] J. Sawada *et al.*, "TrueNorth ecosystem for brain-inspired computing: Scalable systems, software, and applications," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2016, pp. 130–141.

[41] G. Urgese, F. Barchi, E. Macii, and A. Acquaviva, "Optimizing network traffic for spiking neural network simulations on densely interconnected many-core neuromorphic platforms," *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 3, pp. 317–329, Jul./Sep. 2018. doi: 10.1109/TETC.2016.2579605.

[42] A. J. Martin and M. Nystrom, "Asynchronous techniques for system-on-chip design," *Proc. IEEE*, vol. 94, no. 6, pp. 1089–1120, Jun. 2006. doi: 10.1109/JPROC.2006.875789.

[43] A. J. Martin, "Programming in VLSI: From communicating processes to delay-insensitive circuits," Dept. Comput. Sci., California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CSTR:1989.cs-tr-89-01, 1989.

[44] A. J. Martin, "The design of an asynchronous microprocessor," Dept. Comput. Sci., California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CSTR:1989.cs-tr-89-02, 1989.

[45] S. Furber, S. Temple, and A. Brown, "On-chip and inter-chip networks for modeling large-scale neural systems," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 1945–1948. doi: 10.1109/ISCAS.2006.1692992.

[46] G. Cohen, S. Afshar, B. Morreale, T. Bessell, A. Wabnitz, M. Rutten, and A. van Schaik, "Event-based sensing for space situational awareness," *J. Astron. Sci.*, vol. 66, no. 2, pp. 125–141, Jun. 2019. doi: 10.1007/s40295-018-00140-5.

[47] 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. doi: 10.1109/TCSI.2006.887979.

[48] A. Jimenez-Fernandez, R. Paz-Vicente, M. Rivas, A. Linares-Barranco, G. Jimenez, and A. Civit, "AER-based robotic closed-loop control system," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2008, pp. 1044–1047. doi: 10.1109/ISCAS.2008.4541600.

[49] S. Fok and K. Boahen, "A serial H-tree router for two-dimensional arrays," in *Proc. 24th IEEE Int. Symp. Asynchronous Circuits Syst. (ASYNC)*, May 2018, pp. 78–85. doi: 10.1109/ASYNC.2018.00026.

[50] The University of Manchester. (2019). *Spinnaker Project—Boards and Machines*. [Online]. Available: http://apt.cs.manchester.ac.uk/projects/SpiNNaker/hardware/

[51] The University of Manchester. (2019). *Spinnaker Project—System Software*. [Online]. Available: http://apt.cs.manchester.ac.uk/projects/SpiNNaker/software/

[52] Human Brain Project. *Hardware*. Accessed: Apr. 20, 2019. [Online]. Available: https://www.humanbrainproject.eu/en/silicon-brains/how-we-work/hardware/

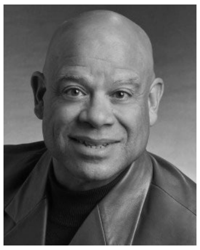
[53] A. P. Davison, E. Müller, S. Schmitt, B. Vogginger, D. Lester, and T. Pfeil. *HBP Neuromorphic Computing Platform Guidebook*. Accessed: Jun. 26, 2019. [Online]. Available: https://electronicvisions.github.io/hbp-sp9-guidebook/pm/pm_hardware_configuration.html

[54] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown, "Overview of the SpiNNaker system architecture," *IEEE Trans. Comput.*, vol. 62, no. 12, pp. 2454–2467, Dec. 2013.
[55] J. Heathcote, "Building and operating large-scale spinnaker machines," Ph.D. dissertation, School Comput. Sci., Univ. Manchester, Manchester, U.K., 2016.
[56] Intel. *Intel's Pohoiki Beach, A 64-Chip Neuromorphic System, Delivers Breakthrough Results in Research Tests*. Accessed: Jul. 15, 2019. [Online]. Available: https://newsroom.intel.com/news/intels-pohoiki-beach-64-chip-neuromorphic-system-delivers-breakthrough-results-research-tests/
[57] M. V. DeBole *et al.*, "TrueNorth: Accelerating from zero to 64 million neurons in 10 years," *Computer*, vol. 52, no. 5, pp. 20–29, May 2019.

AARON R. YOUNG was born in Knoxville, Tennessee, in 1993. He received the B.S. and M.S. degrees in computer engineering from the University of Tennessee Knoxville, in 2016 and 2017, respectively, where he is currently pursuing the Ph.D. degree in computer engineering.

From 2011 to 2015, he was in summer internships at Siemens Medical Systems, Oak Ridge National Laboratory's Manufacturing Demonstration Facility, and Garmin International. Since 2016, he has been a Graduate Research Assistant with the TENNLab Neuromorphic Research Group, University of Tennessee Knoxville. His current research interests include neuromorphic computing, computer architectures, embedded systems, and high-speed communication.

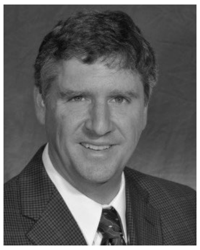
Mr. Young has been a member of the Tau Beta Pi and the Engineering Honor Society, since 2013. He has been receiving the Bodenheimer Fellowship, since 2016. He received the Top Collegiate Scholar, in 2016, the Outstanding Computer Engineering Senior, in 2015, the Outstanding Computer Engineering Senior, in 2014, the Outstanding Computer Engineering Junior, in 2013, and the Min H. Kao Electrical and Computer Engineering Scholarship, from 2013 to 2015.



MARK E. DEAN received the B.S. degree in electrical engineering from the University of Tennessee, Knoxville, Knoxville, TN, USA, in 1979, the M.S. degree in electrical engineering from Florida Atlantic University, in 1982, and the Ph.D. degree in electrical engineering from Stanford University, in 1992.

From 1979 to 1993, he was a Senior Technical Staff Member, a Senior Engineer, an Advisory Engineer, a Staff Engineer, a Senior Associate Engineer, an Associate Engineer, and a Junior Engineer with the IBM Corporation. From 1993 to 1994, he was the Director of the Architecture, Power Personal Systems Division, IBM Corporation. From 1994 to 1995, he was the Director of system platforms, interactive broadband systems with IBM Corporation. From 1995 to 1997, he was the IBM Fellow and the Director of system architecture and performance, RS/6000 Division. From 1997 to 2000, he was the IBM Fellow and the Vice President (VP) of systems, IBM Research. From 2002 to 2004, he was the IBM Fellow and VP of storage technology, IBM Systems Group. In 2004, he was the IBM Fellow and the VP of hardware and systems architecture, IBM Systems and Technology Group. From 2004 to 2008, he was the IBM Fellow, a Senior Location Executive, and the VP with Almaden Research Center, IBM Research. From 2008 to 2011, he was the IBM Fellow and the VP of worldwide strategy and operations, IBM Research. From 2011 to 2013, he was the IBM Fellow, VP, and Chief Technology Officer, in Middle East and Africa. Since 2013, he has been a John Fisher Distinguished Professor with the College of Engineering, University of Tennessee Knoxville. His current research interests include advanced computer architecture (beyond Von Neumann systems), data centric computing, and computational sciences.

Dr. Dean was a member of the American Academy of Arts and Sciences, the National Academy of Engineering, and the National Inventor's Hall of Fame. He received the National Institute of Science Outstanding Scientist Award, Black Engineering of the Year, the University of Tennessee COE Dougherty Award, and the Ronald H. Brown American Innovators Award.



JAMES S. PLANK received the B.S. degree in computer science from Yale University, in 1988, and the M.S. and Ph.D. degree in computer science from Princeton University, in 1990 and 1993, respectively.

Since 1993, he has been a Professor with the Electrical Engineering and Computer Science Department, College of Engineering, University of Tennessee, Knoxville. His current research interests include neuromorphic computing, fault-tolerance, erasure codes, storage systems, distributed computing, and operating systems.

Dr. Plank is a member of the IEEE Computer Society. He received many awards for teaching, including, the Department's Teaching Award for seven times, the College of Arts and Sciences Senior Faculty Teaching Award, and the Chancellor's Citation for Excellence in Teaching. He has chaired conferences in network storage and network applications, and has left a legacy of publicly available software that includes: Jgraph, Ickp, Libckpt, IBP, LoRS, Galois.tar, and Rsclib. He has served as an Associate Editor for IEEE TPDS.

GARRETT S. ROSE received the B.S. degree in computer engineering from Virginia Tech, in 2001, and the M.S. and Ph.D. degrees in electrical engineering from the University of Virginia, in 2003 and 2006, respectively.

From 2006 to 2011, he was an Assistant Professor with the Electrical and Computer Engineering Department, Polytechnic University (now NYU Polytechnic School of Engineering), in Brooklyn, NY, USA. From 2011 to 2014, he was a Senior Electronics Engineer with the Air Force Research Laboratory, and Information Directorate, Rome, NY. Since 2014, he has been an Associate Professor with the Department of Electrical Engineering and Computer Science, University of Tennessee Knoxville. His current research interests include nanoelectronic circuit design as applied to a range of applications, including reconfigurable computing, neuromorphic computing, and hardware security.

Dr. Rose is a member of the Association of Computing Machinery, the IEEE Circuits and Systems Society, and the IEEE Computer Society. He serves and has served on Technical Program Committees for several IEEE conferences, including ISCAS, GLSVLSI, and NANOARCH and workshops in the area of VLSI design. In 2010, he was a Guest Editor of a special issue of the ACM *Journal of Emerging Technologies in Computing Systems* that presented key papers from the IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH'09). From 2014 to 2017, he was an Associate Editor of the IEEE TRANSACTIONS ON NANOTECHNOLOGY.