Real-time Scalable Cortical Computing at 46 Giga-Synaptic OPS/Watt with $\sim$100$\times$ Speedup in Time-to-Solution and $\sim$100,000$\times$ Reduction in Energy-to-Solution

Andrew S. Cassidy*, Rodrigo Alvarez-Icaza*, Filipp Akopyan*, Jun Sawada*, John V. Arthur*, Paul A. Merolla*, Pallab Datta*, Marc Gonzalez Tallada*, Brian Taba*, Alexander Andreopoulos*, Arnon Amir*, Steven K. Esser*, Jeff Kusnitz*, Rathinakumar Appuswamy*, Chuck Haymes†, Bernard Brezzo†, Roger Moussalli†, Ralph Bellofatto†, Christian Baks†, Michael Mastro†, Kai Schleupen†, Charles E. Cox†, Ken Inoue†, Steve Millman†, Nabil Imam¶, Emmett McQuinn*, Yutaka Y. Nakamura‡, Ivan Vo§, Chen Guo‖, Don Nguyen**, Scott Lekuch†, Sameh Asaad†, Daniel Friedman†, Bryan L. Jackson*, Myron D. Flickner*, William P. Risk*, Rajit Manohar¶, and Dharmendra S. Modha*

*IBM Research - Almaden, †IBM T. J. Watson Research Center, ‡IBM Research - Tokyo, §IBM Research - Austin, ¶Cornell University, ‖IBM Engineering and Technology Services, San Jose Design Center, **Posthumous
Contact e-mail: dmodha@us.ibm.com

***Abstract*—Drawing on neuroscience, we have developed a parallel, event-driven kernel for neurosynaptic computation, that is efficient with respect to computation, memory, and communication. Building on the previously demonstrated highly-optimized software expression of the kernel, here, we demonstrate TrueNorth, a co-designed silicon expression of the kernel. TrueNorth achieves five orders of magnitude reduction in energy-to-solution and two orders of magnitude speedup in time-to-solution, when running computer vision applications and complex recurrent neural network simulations. Breaking path with the von Neumann architecture, TrueNorth is a 4,096 core, 1 million neuron, and 256 million synapse brain-inspired neurosynaptic processor, that consumes 65mW of power running at real-time and delivers performance of 46 Giga-Synaptic OPS/Watt. We demonstrate seamless tiling of TrueNorth chips into arrays, forming a foundation for cortex-like scalability. TrueNorth's unprecedented time-to-solution, energy-to-solution, size, scalability, and performance combined with the underlying flexibility of the kernel enable a broad range of cognitive applications.**

I. Overview

The brain's network of interconnected neurons is the most complex "computer" known—capable of high-level cognition while consuming less than 20W—unmatched by conventional von Neumann machines. Engineers have long desired to approach neurobiology's capabilities and efficiency by harnessing neuroscientific knowledge and translating it into silicon technology, creating brain-inspired computers [10], [11]. Such machines have the potential to revolutionize the computer industry and society by integrating intelligence into devices limited by power and speed, providing a substrate for a cloud-based multimedia processing, as well as enabling synaptic supercomputers for large-scale scientific exploration.

Looking at the brain through the computational lens, its memory and area requirements scale with number of synapses whereas computation, communication, power, and speed scale with *synaptic events*, where a synaptic event corresponds to a non-zero valued synapse receiving and processing a neuronal spike. Remarkably, in this metric, the brain operates its hundred trillion synapses at an energy efficiency of $\sim$10fJ per synaptic event.

We have created an efficient neuroscience-inspired computational kernel, engineered to take advantage of parallel hardware that includes multiprocessor computers as well as custom-designed neural processors. In particular, it supports parallelism across threads, event-based communication, event-based computation, message aggregation, and localizing memory with computation.

As a software expression of this kernel, we previously simulated one hundred trillion synapses via a scalable simulator, Compass, [6], [7]. In spite of the fact that the function-level simulator was judiciously optimized along many dimensions and that the simulation used a highly energy-efficient supercomputer, LLNL's Sequoia (at the time top-ranked on Green500), the cost was $\sim 1\mu$J per synaptic event—eight orders of magnitude more than the brain. This dramatic energy disparity arises from the profound difference between the neural architecture and organic technology of the brain and the von Neumann architecture and silicon technology of today's computers.

Here we present the silicon expression of the kernel in the form of a novel brain-inspired architecture leading to a novel neurosynaptic processor [12]—TrueNorth—that

978-1-4799-5500-8/14 $31.00 © 2014 IEEE
DOI 10.1109/SC.2014.8

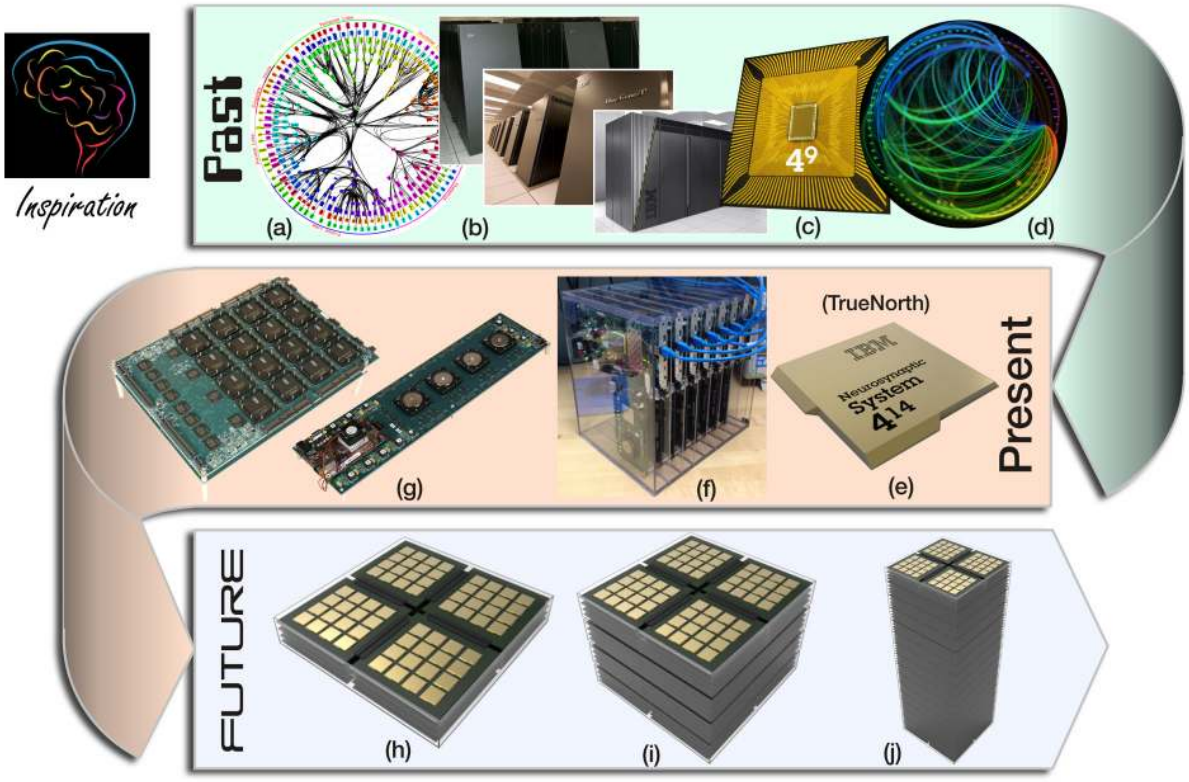


Fig. 1. SyNAPSE: Past, Present, and Future. **(a)** The inspiration for the project lies in neuroanatomy and neurophysiology [1]. We have compiled the largest long-distance wiring diagram of the primate brain giving insight into communication architecture of the brain [2]. We have developed a simple, digital, reconfigurable, versatile spiking neuron model that is efficient to implement in hardware and yet supports a wide variety of biologically-relevant spiking behaviors and computational functions [3]. **(b)** We have run a series of simulations from "mouse-scale" and "rat-scale" on Blue Gene/L [4] to "cat-scale" on Blue Gene/P [5] to "human-scale" on Blue Gene/Q [6], [7]. **(c)** Crystallizing insights from neuroscience and supercomputing, we have demonstrated a key building block of a novel architecture, namely, a neurosynaptic core, with 256 digital integrate-and-fire neurons and a 1024×256 ($= 4^9$) bit SRAM crossbar memory for synapses [8]. **(d)** Emulating the clustered hierarchical connectivity of the cortex [9], we have developed a new architecture that in essence is a network of neurosynaptic cores. The architecture is parallel, distributed, modular, scalable, event-driven, fault-tolerant like the brain. We have developed a new multi-threaded, massively parallel, functional simulator, Compass, functionally equivalent to this architecture. **(e)** By using co-design with Compass, we have conceived, designed, fabricated, and tested TrueNorth with 1 million neurons and 256 million ($= 4^{14}$) synapses. TrueNorth has $4,096$ neurosynaptic cores interconnected via an event-driven network-on-chip. **(f)** We have created an eight-board system where each board contains one TrueNorth processor and each of these boards is a stand-alone network node connected via 1Gb Ethernet. **(g)** By exploiting the tileability of TrueNorth, we have developed and are testing 4-chip and 16-chip boards with tiled arrays of TrueNorth processors demonstrating native multi-chip communication and to demonstrate a system with 16 million neurons and 4 billion synapses. **(h), (i), (j)** Looking to the future, we imagine a "mouse-scale" system with 256 neurosynaptic processors that consumes merely 256 Watts; a "rat-scale" system with $1,024$ processors that consumes only 1kW; and a "1%-human-scale" system with $4,096$ processors that consumes merely 4kW. The $4,096$ processor system will contain one trillion (10^{12}) synapses.

achieves $\sim$10pJ per synaptic event. One of the largest chips ever fabricated, with 5.4 billion transistors in Samsung's 28nm process technology, TrueNorth has 1 million neurons and 256 million synapses organized in $4,096$ neurosynaptic cores [13]—all operating in a parallel, event-driven fashion and interconnected via an on-chip network. In terms of energy efficiency, when running a complex recurrent neural network with 20Hz average firing rate and 128 active synapses per neuron in real-time (updated at 1kHz), TrueNorth consumes merely 65mW and delivers 46 Giga-Synaptic Operations Per Second per Watt (GSOPS/W)[1]. Running this network $\sim 5\times$ faster (amortizing passive power), TrueNorth delivers 81 GSOPS/W. For higher spike rates (200Hz) and higher synaptic utilization (256 per neuron), TrueNorth exceeds 400 GSOPS/W. For a number of practical computer vision applications, we demonstrate that TrueNorth consumes five orders of magnitude less energy than Compass running on either an x86 system (with two 6-core processors) or 32 Blue Gene/Q compute cards [14], [15] (each with up to 64 threads). When running these applications, TrueNorth has a power density of 20mW/cm^2 which is roughly four orders of magnitude lower than a modern processor with an approximate power density of 100W/cm^2.

To achieve this performance, we have engaged in a multi-disciplinary, multi-institutional, multi-year DARPA SyNAPSE project since 2008. Fig. 1 describes the context of TrueNorth, starting with a series of neuroscience-inspired simulations that led to a highly-optimized kernel that in turn led to a novel parallel, distributed, modular, scalable, event-driven, fault-tolerant architecture that has become the basis of TrueNorth. These simulations allowed us to understand computation,

[1] Just as humans excel at tasks like visual object recognition but perform poorly at FLOPS, the TrueNorth neurosynaptic processor, while Turing-complete, is efficient for cognitive applications using synaptic operations, not FLOPS. Conversely, modern von Neumann processors are efficient for FLOPS, but not for synaptic operations.

communication, and memory constraints as well as challenges in scaling and real-time performance. Our optimized simulator, Compass, is functionally 1:1 equivalent with TrueNorth via co-design [16], [17]. As a result, we have developed a cache of applications on Compass [18], such as multi-sensory feature extraction and pattern recognition; association and context processing; as well as information extraction, that now run without modification on TrueNorth, orders of magnitude faster and for orders of magnitude less energy.

Scaling beyond a single chip, TrueNorth has a tileable structure enabling modular, scalable cognitive supercomputers as envisioned in Fig. 1(h-j). As a first step in this direction, we demonstrate four-chip and sixteen-chip boards with tiled arrays of TrueNorth processors that communicate without any additional peripheral circuitry. We also demonstrate a rack containing eight 1Gb Ethernet cards each with a single TrueNorth processor. Looking to the future, we can imagine replicating the "1% human-scale" simulations that required 16 racks of Blue Gene/P and ran 400× slower than real-time [5] on a TrueNorth system that requires only one rack, would run in real-time and consume an estimated $128,000\times$ less energy, as detailed in Section VII. These systems pave the way for multimedia cloud processors capable of dealing with a myriad of sensors pervasive in today's world.

To enable applications across a wide spectrum spanning neural networks and machine learning, we have developed an end-to-end ecosystem, described in Fig. 2.

II. Related Work

Neural network simulations on supercomputers have a rich history, from early work simulating artificial neural networks [20], to more recent spiking neural networks projects including the NEST simulator [21], [22] on the K-computer [23], bio-physically detailed cortical microcircuits [24], [25], [26], GPU acceleration [27], and others summarized in [28]. Our simulations, using our optimized function-level kernel, have progressed from 16 racks of Blue Gene/L [4] (30 million neurons, 240 billion synapses) to 36 racks of LLNL Dawn Blue Gene/P [5] (1.6 billion neurons and 8.87 trillion synapses) to 96 racks of LLNL Sequoia Blue Gene/Q [6], [7], culminating in one hundred trillion synapses at "human-scale."

Historically, neuro-inspired computers followed Carver Mead's pioneering work [10], modeling biological neural circuits using silicon analog electronics, including neurons [29], ion channel models [30], and winner-take-all circuits [31]. Larger systems, for example Neurogrid (65k neurons) [32], CAVIAR (45k neurons) [33], and IFAT (65k neurons) [34] combine arrays of these analog components with an external memory to store connectivity information. More recent architectures include the BrainScaleS project [35], demonstrated a wafer-scale system (20cm diameter wafer) with a total of 200 thousand analog neurons and 40 million addressable synapses, and consumes roughly a kilowatt. The SpiNNaker project [36] demonstrated a 48-chip system (each chip has 18 ARM processors) simulating a total of 250 thousand neurons and 80 million synapses (implemented via off-chip memory), which consumes 36W.

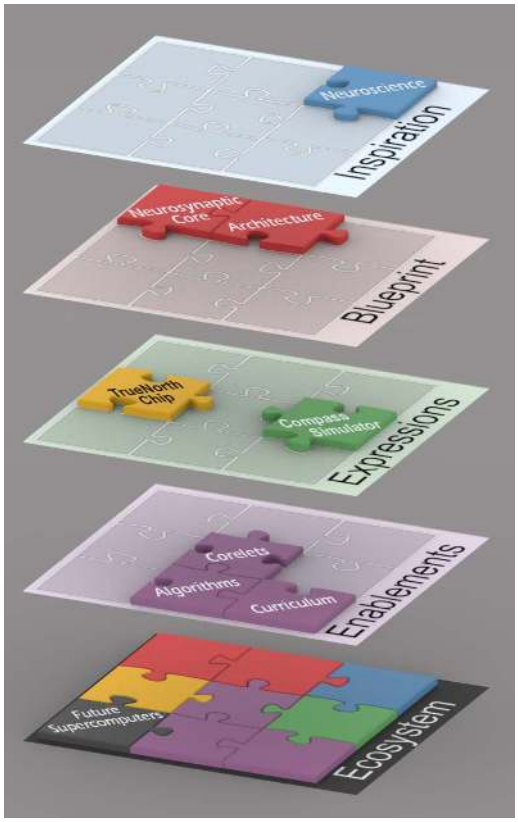


Fig. 2. Building a Cognitive Computing Ecosystem. The **inspiration** for the ecosystem lies in neuroscience [1]. The **blueprint** defines the high-level architecture, a 2D array of neurosynaptic cores. Each core is an independent module integrating computation (neurons), communication (axons), and memory (synapses). Cores operate in parallel, communicating by sending messages (*spike events*) through a mesh network. We have implemented two **expressions** of the blueprint, co-designing a scalable supercomputer-based simulator (Compass [6]) and a scalable silicon-fabricated processor (TrueNorth [12]), ensuring identical one-to-one operation, such that any model on the software simulator runs unchanged on the hardware. **Enablements** or *infrastructure* includes an architecture specific "Corelet" language and Corelet Programming Environment (CPE) [19], a library of algorithms [18] in CPE, and Compass to simulate networks and to facilitate training off-line. We developed *Applications* that ran initially on Compass—even before TrueNorth was fabricated and tested—and later (without modification) on TrueNorth, including convolutional networks, liquid state machines, restricted Boltzmann machines, hidden Markov models, support vector machines, and optical flow [18]. To promulgate the new ecosystem, we have developed a new teaching curriculum. Combining all of the pieces, the ecosystem forms a framework for a new generation of synaptic supercomputers.

III. Innovations: Neurosynaptic Kernel and Expressions

One of the fundamental keys to high-performance scientific computing is a kernel that is efficient in computation, memory, and communication, as demonstrated by the gravity kernel [37], the molecular dynamics iteration [38], fluid flow dynamics [39], the phase-field model of solid and liquid materials [40], thermodynamics and free energy computation [41], thermonuclear computation [42], and atomistic simulation [43].

We present an efficient kernel, inspired by neuroscience, that serves as the blueprint for both our software cortical simulator and silicon cortical processor. The kernel's pseudo-code (Listing 1) specifies a spiking neural network simulation, where neurons are nodes of a graph and their connections (synapses) are the edges. Individual neurons parameters are fully programmable, supporting a wide range of spiking behaviors with the same basic neuron model [3], and individual synapses are also programmable, supporting flexible network topologies. The kernel is designed to take advantage of parallel hardware including multiprocessor computers and custom-designed neural chips. In particular, the kernel supports:

- **Parallelism across threads:** Neurons and synapses are partitioned into parallel threads that communicate with each other via messages. Such partitions can have an efficient implementation when the topology of the neural network has spatial structure (for example, clustered or sparse connectivity), and threads are able to utilize locality between memory and computation (for example, using a local cache on a von Neumann architecture [44], or a dedicated memory structure on a custom architecture).
- **Event-based communication:** Spike events, which represent the individual firing of neurons (for example, when a neuron's potential exceeds its programmed threshold) communicate information within and between threads. Specifically, each presynaptic spike event is replicated and sent to all its target synapses (line 15), which are stored as *synaptic events* in the threads hosting the targeted neurons.
- **Event-based computation:** Within a discrete time step of a neural simulation, enforced by a synchronization barrier to ensure determinism, each thread updates its local neurons by processing all of the pending synaptic events (line 5). Because neurons fire sparsely in time (on the order of a few Hertz), the event-based update loop is significantly more efficient than an alternative approach that loops over all synapses.

```
for thread in allthreads {                                  1
  // Neuron updates & spikes
  for neuron in thread.neurons {                            3
    // Synaptic input
    for synapse_event targeting neuron.synapses {           5
      // Compute neuron membrane potential
      update neuron by synapse_event.weight;                7
    }
    // Compute neuron leak                                  9
    neuron leak_update;
    // Check if neuron exceeds threshold                    11
    neuron threshold;
    if neuron.spike {                                       13
       // Communicate spike events
       transmit spike_event to target_synapses;             15
       // Reset neuron
       neuron reset;                                        17
    }
  }                                                         19
  // Check communication complete
  barrier;                                                  21
  // Advance to next time step
}                                                           23
```

Listing 1. Pseudo-code of the blueprint algorithmic kernel, executed every time step. Each thread updates all its assigned neurons based on arriving synaptic events. Then it checks if the neuron has exceeded its threshold, if so, it communicates to all targeted synapses, which may involve inter-thread communication.

A. Neurosynaptic Core

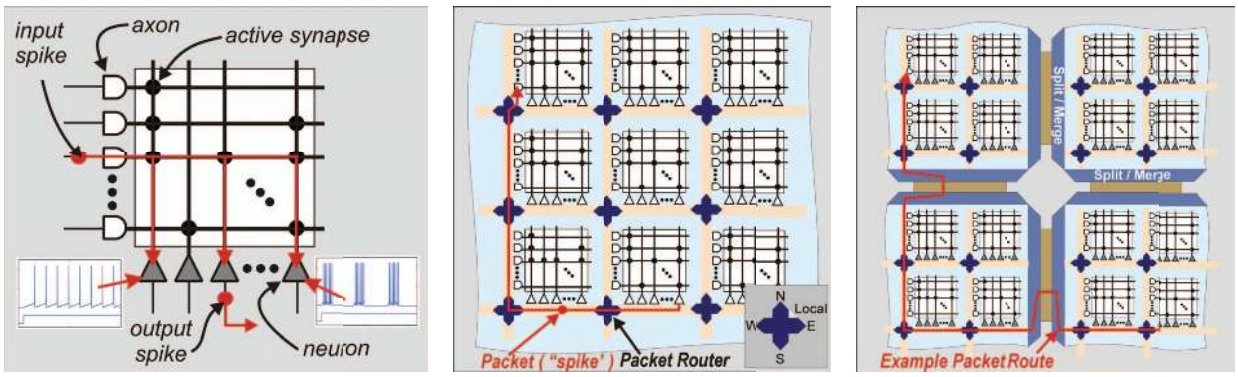
A further optimization of the kernel is to introduce a novel fundamental data structure, called a *neurosynaptic core*, which integrates axons, neurons, and synapses. The structure of the core is inspired by observations from neurobiology, where neurons and their connections often form clusters to create local cortical microcircuits [45], [46]. The core brings computation, communication, and memory together and operates in an event-driven fashion.

An individual neurosynaptic core represents 256 axons, 256 neurons, and 256×256 synapses, Fig. 3(a). Externally, axons are the cores inputs, receiving spike events, and neurons are the outputs, emitting spike events. Internally, it is a fully-connected directed graph with programmable synaptic connections from all axons to all neurons (synapses are non-learning). Thus, a single core can model networks with in-degree and out-degree of 256 or less. Functionally, information flows from individually addressable axons (horizontal lines), through the active synapses in the crossbar (binary-connected crosspoints), to drive inputs for all of the connected postsynaptic neurons (vertical lines). Axons are activated by incoming input spike events, which are generated by neurons anywhere in the system, and delivered via message passing.

The primary advantage of using a core is that it overcomes a key communication bottleneck that limits scalability for large scale network simulations. Specifically, without using cores, we are required to replicate spike events for each target synapse; therefore, in a system with N neurons and S synapses, we need to send $\frac{S}{N}$ events for each spike. By partitioning the network into neurosynaptic cores, we only need to send one event to simultaneously target all of a core's `target_synapses`, reducing total traffic by a factor of $\frac{S}{N}$ (typically 256). In essence, by enforcing a clustered network topology with in-degree and out-degree of 256, we overcome an important communication bottleneck.

A neurosynaptic core also offers a memory efficient data structure, taking advantage of implicit memory addressing. A crossbar with C input axons and output neurons implements a fanout of C each time an axon is activated by an input spike. Uniquely addressing axons (which implicitly addresses neurons) requires $\frac{S}{C}\log_2\frac{S}{C}$ bits. An alternate approach that explicitly addresses each synapse would require $S\log_2 S$ bits, since each neuron would need to store a list of targets specifying the unique synapses.

The details of how a core updates its internal state are summarized here: Synapses are structured as a connection matrix (in the crossbar), where a connection from axon i to neuron j is $W_{i,j}$ (binary). A core updates all of its neurons at discrete time steps t, which is nominally 1ms. At each time step t, the core processes a binary input vector of axon states (whether an input spike is present), where each axon i is



(a) (b) (c)

Fig. 3. TrueNorth architecture at core, chip, and multi-chip scale. (a) The building block is a neurosynaptic core, where horizontal lines are axons (inputs), the "square-end half-circle" symbol denotes axon delay buffers, cross points are individually programmable synapses, vertical lines are neuron inputs, and triangles are neurons (outputs). Neurons behaviors are individually programmable with two examples shown. (b) Cores naturally tile using a 2D on-chip mesh routing network. Long-range connections between neurons and axons are implemented by sending spike events (packets) over the mesh network. (c) Individual chips also tile in 2D, with the routing network extending across chip boundaries through peripheral merge and split blocks.

represented by a bit $A_i(t)$. Although each synaptic connection is binary, it can mediate a multi-valued post-synaptic effect. Specifically, each axon i is assigned to one of four types G_i, which corresponds to a weight specified individually for each neuron. For example, axon–neuron connections can be set to be excitatory or inhibitory, and each with different synaptic strengths. Mathematically, at time t, neuron j receives input: $A_i(t) \times W_{i,j} \times S_j^{G_i}$ (Listing 1, line 7), where $S_j^{G_i}$ is a programmable signed integer. In an alternate mode, the active connections are integrated probabilistically (using a pseudo-random number generator, PRNG, in each core), emulating stochastic neural dynamics. Neurons integrate synaptic input over time, maintaining the state in their membrane potential V_j, and emitting spikes if they exceed their thresholds [3] (line 12); thresholds can also be drawn from the PRNG. Each spike is associated with a target core, a target axon address, and a delivery time t_D computed as t plus an programmable axonal delay from 1 to 15 (line 15).

We now describe a software and a silicon expression of our neuroscience-inspired kernel, both based on neurosynaptic cores, and are exactly functionally equivalent.

B. Kernel Software Expression: Compass Simulator

Compass is a highly-optimized function-level simulator for large-scale networks of spiking neurons organized as neurosynaptic cores. The simulator is written in C++, sends spike events via MPI communication [47] and uses OpenMP [48] for thread-level parallelism. Compass demonstrates outstanding weak and strong scaling results [6], [7].

The kernel in Listing 1 maps directly to the main semi-synchronous simulation loop used by Compass, with each pass simulating a time step. In the *Synapse* phase (lines 4-8), each process propagates input spikes from axons to neurons through the crossbar and performs synaptic integration. Next, in the *Neuron* phase (lines 9-13), each process executes the leak, threshold, and fire model for each neuron. Last, in the *Network* phase (line 15) processes send spikes from firing neurons to destination axons. For additional efficiency, Compass aggregates spikes between pairs of processes into a single MPI message; overlaps communication with computation; uses an innovative synchronization scheme requiring just two communication steps regardless of the number of the processors; uses meticulous load-balancing; and uses highly compressed data structures for maintaining neuron and synapses states. These advances enabled Compass to exercise all 6.3 million threads and 1.5 million processors on LLNL's Sequoia Blue Gene/Q.

Compass is indispensable for exploring scaling for large-scale network simulations; benchmarking inter-core communication on different neural network topologies; demonstrating applications in vision, audition, motor control, and sensor integration [18]; and hypotheses testing, verification, and iteration regarding neural codes and function. Furthermore, via co-design, Compass played an instrumental role in developing our energy-efficient hardware kernel expression, informing architectural choices in the hardware design, as well as verifying the hardware pre- and post-fabrication via function-level regression testing.

C. Kernel Hardware Expression: TrueNorth Chip

Our key innovation is a very efficient implementation of the kernel in silicon. Building on the success of Compass, we have conceived, designed, built, and tested a custom-designed neural processor—TrueNorth—that is able to run a network of neurosynaptic cores in real time, while consuming little total power (active + passive power). TrueNorth's architecture, (Fig. 3), is a custom-designed mixed asynchronous-synchronous chip that was fabricated in Samsung's 28nm process technology. With 5.4 billion transistors in 4.3cm^2, TrueNorth has an on-chip network of $4,096$ neurosynaptic cores—for a total of one million neurons and 256 million synapses. The physical implementation of a neurosynaptic core fits in a $390\mu m \times 240\mu m$ footprint.

Active power is kept low by following the event-driven nature of the kernel and only evaluating the neural updates that

are required. Furthermore, we use dedicated memory (built to the precise size required by a neurosynaptic core), co-located with the computation—ensuring the energy to move bits across wires is kept low. Passive power is kept low using a low power process and choosing slower low-leakage transistors. Multiplexing the neuron computation results in a more compact design that reduces both active power (shorter wires for signaling) and passive power (fewer leakage paths).

TrueNorth's on-chip network, (Fig. 3(b)), interconnects the 2D array of cores. Spike events (single-word packets) are sent from neurons to axons via the communication network to implement long-range point-to-point connections. Unlike the brain that has slow dedicated physical wires for each connection, we time-multiplex fast metal wires and digital electronics to emulate the brain's high connectivity. Specifically, each core is equipped with a five-port router that forms the backbone of our 2D mesh network. When a neuron on a core spikes, it injects a packet into the mesh, which is passed from core to core—first in the x dimension then in the y dimension (deadlock-free dimension-order routing [49])—until it arrives at its target core, where it fans out locally. The architecture is robust to core defects: if a core fails, we disable it and route spike events around it.

To scale the 2D mesh across chip boundaries, where the number of inter-chip connections is limited, we use a merge–split structure at the four edges of the on-chip mesh boundary (Fig. 3(c)). Packets leaving the mesh are tagged with their row (or column) before being merged onto a shared link that exits the chip. Symmetrically, packets that enter the chip from a shared link are sent to the appropriate row (or column) using the tagged information. This enables system-level scalability: TrueNorth chips can be tiled into a 2D array—just like cores are tiled to create the chip array—without the need for auxiliary communication circuits.

In summary, the TrueNorth processor is *efficient* because (i) memory and computation are co-localized, eliminating the von Neumann bottleneck; (ii) cores are event-driven, which results in active power proportional to firing activity; and (iii) only spike events, which are sparse in time, are communicated between cores via the long-distance communication network. Furthermore, TrueNorth is *scalable* because (i) cores can be tiled in 2D similar to the mammalian neocortex [50]; (ii) local core failures do not disrupt global usability; (iii) individual chips can be tiled in a 2D array through direct chip-to-chip connections; and (iv) the hierarchical communication model lowers system bandwidth requirements.

IV. Applications

Programming the TrueNorth processor consists of specifying three things: the dynamics of each neuron (setting the neuron parameters and weights), the mapping from neuron outputs to axon inputs (configuring the network routing tables), and the local synaptic connectivity between axons and dendrites (setting the binary crossbars). The Corelet language is used to efficiently specify all of these parameters.

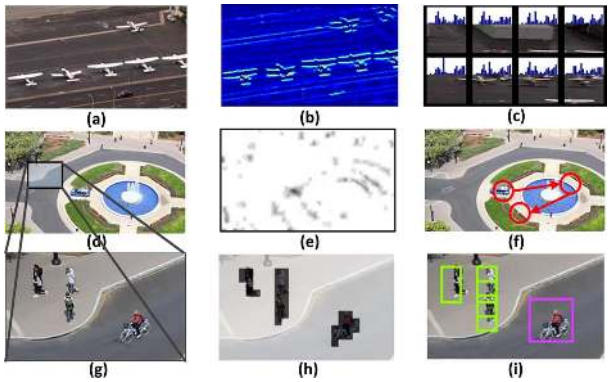


Fig. 4. Applications for Characterization. Frames of streaming video drive all applications **(a)**,**(d)**,**(g)**. Two applications generate features: **(b)** Haar-feature response map for horizontal lines; **(c)** eight LBP histograms extracted from 8 subpatches. Two applications identify and focus on interesting objects: **(e)** salient object map; **(f)** example saccades. One application detects and classifies objects: **(h)** where pathway; **(i)** merged what and where results.

A. Corelets

Applications for the TrueNorth processor are developed in the Corelet Programming Environment (CPE) [19], a new, object-oriented, compositional language and development environment that promotes efficient, modular, scalable, and collaborative TrueNorth software. A *corelet* is a functional encapsulation of a network of neurosynaptic cores that collectively perform a specific task. Object-oriented corelets can seamlessly build hierarchically composable networks while sharing underlying code and unified network interfaces. Our approach is conceptually akin to Banavar's framework for compositional modularity [51], and dramatically improves code reuse and scalability. Corelets are collected in the *corelet library*—an ever growing repository of reusable corelets covering numerous TrueNorth implementations of seminal algorithms, including linear and non-linear signal and image processing; spatio-temporal filtering; saliency; object detection, classification, and recognition; and real-time audio and video analytics [18], a representative set of which have been selected for testing here.

B. Applications for Performance Characterization

We analyze TrueNorth performance in Section VI on several complex applications that were co-designed to run on the simulator and the TrueNorth processor to perform feature extraction, saliency, detection and classification (Fig. 4), as well as large-scale recurrent neural network computation.

Feature extraction is a critical component of most computer vision systems. Here, we tested two types of feature extractors: Haar-like features, often used in face detection [52], and Local Binary Patterns (LBP), often used in biometrics, robot navigation, and brain MRI analysis [53]. Both systems processed 100×200 pixel video at 30 frames per second, using either ten Haar-like features in a network of $617,567$ neurons in $2,605$ cores with a 135Hz mean firing rate, or 20-bin Local Binary Pattern feature histograms in a network of $813,978$ neurons in $3,836$ cores with a 64Hz mean firing rate.

A saliency map assigns a measure of interest, or saliency, to each pixel in an image, often to a select a region for further processing [54]. First, our saliency system creates a saliency map using a feature extraction corelet with 889,461 neurons in 3,926 cores and an 86Hz mean firing rate. Second, a saccade map selects regions of interest by applying a winner-take-all mechanism to the saliency map, followed by temporal inhibition-of-return to promote map exploration, using a corelet with 612,458 neurons in 2,571 cores and a 5Hz mean firing rate.

Many computer vision tasks require both detecting an object's location and classifying its identity. We built a multi-object detection and classification system for high-resolution, fixed-camera videos. Our system includes a Where network to detect objects, a What network to classify objects, and a What/Where network to bind these predictions into labeled bounding boxes. We applied this system to the DARPA Neovision2 Tower dataset [55], [56], which contains moving and stationary people, cyclists, cars, buses, and trucks. A single TrueNorth chip processed a 240×400 pixel aperture at 30 frames per second in real-time, using 660,009 neurons in 4,018 cores with a 12.8Hz mean firing rate, and achieving 0.85 precision and 0.80 recall on the test set.

Finally, to systematically characterize TrueNorth's operation space and performance, we created a set of 88 probabilistically generated recurrent networks that each use all 4,096 cores and every neuron on the processor. The set of recurrent networks spans mean firing rates per neuron from 0 to 200Hz, and active synapses per neuron from 0 to 256. Neurons project to axons that are an average of 21.66 hops (cores) away both in x and y dimensions.

V. Measurement System and Environment

We benchmarked TrueNorth speed and power against two high-performance computing architectures: IBM Blue Gene/Q [57] and Intel x86. On Blue Gene/Q we used up to 32 compute cards, each card with 16GB of DDR3 DRAM and an 18-core PowerPC A2 processor (of which 16 cores run applications), with four hardware threads per core [14]. The x86 system was a dual socket board with two 6-core E5-2440 processors operating at 2.4GHz, 188GB of DRAM, a last-level 15MB shared cache, and Red Hat Enterprise Linux 4.4.7.

1) Synaptic Operations per Second (SOPS) : Instruction-based computation is generally measured in **Operations per Second**, (for example, FLOPS). Since TrueNorth does not use traditional von Neumann-style instructions, we define its fundamental operation to be a synaptic integration, a conditional weighted-accumulate operation that forms the inner loop of the neuron function (Listing 1, line 7). See [3] for the full neuron equations. Mathematically, one **synaptic operation** is: $V_j(t) + A_i(t) \times W_{i,j} \times S_j^{G_i}$, conditioned on both the synapse being active ($W_{i,j} = 1$) and a spike arriving at the input axon ($A_i(t) = 1$). $A_i(t)$ and $W_{i,j}$ are binary, and the membrane potential $V_j(t)$ and synaptic weights S_j are 20-bit and 9-bit signed integers respectively. SOPS is a conservative measure of computation that ignores all other operations (leak, threshold, random number generation, etc.) computed by each TrueNorth core. SOPS can be computed as $avg.\ firing\ rate \times avg.\ active\ synapses$, thereby counting only the spikes which pass through connected synapses.

2) Power Consumption: For TrueNorth power, we sampled the chip's core current at 65.2kHz with an AD7689 analog-to-digital converter and smoothed the single time step current waveform with a level-triggered average ($num\_time\_steps > 500$). Calibrating against a Keithley PS2185 power source, we found only a 3% difference in estimated RMS current.

For Blue Gene/Q power, we used the EMON interface [58] to query the IBM DB2 relational database used by Blue Gene systems to periodically log time stamped environmental measurements from various components [59]. We averaged the reported node card (32×16 cores) power and estimated compute card (16 cores) power by dividing node card power by 32. For x86 power, we used the PAPI 5.3.0 interface [60] to read the Running Average Power Limit (RAPL) registers [61], which sample at 1kHz (every simulation time step) the power of the full processor package, of just the compute cores, or of the external DRAM.

VI. Performance Results

A. Logic Correctness: One to One Equivalence

In terms of transistor and neuron count, TrueNorth is the largest neuromorphic chip ever produced. To verify the logical correctness, we adopted a hardware–software co-design strategy based on the shared function-level kernel definition. 1) Prior to fabrication, we compared test vector output from the optimized function-level simulator, Compass, with the mixed gate- and transistor-level simulation of the hardware design. We simulated 413,333 single-core regressions and 7,536 full-chip (instancing up to 2,048 cores) regressions, with the detailed hardware simulator consuming approximately 100 years of CPU time. 2) Post-fabrication, with TrueNorth silicon hardware, we validated the fidelity of the manufacturing process and transistor layout with an additional set of 289 full-chip regressions. All pre- and post-fabrication regressions matched 100% between the hardware design and Compass, the function-level simulator.

We tested the temporal limits of exact 1:1 correspondence of the hardware and the Compass simulator by running regressions from 10k to 100M time steps. Again, not a single spike mismatch was found, maintaining 100% agreement. The longest running regression took 27.7 hours on TrueNorth at 1kHz (real-time), versus 74 days on Compass using an x86-based server (dual-socket Intel Xeon X7350 quad-core processors operating at 2.93 GHz) running 8 threads. We also ran the regressions at operating voltages ranging from 0.67V to 1.05V. The 88 probabilistically-generated recurrent neural networks of Section IV-B are a sensitive assay for any deviation from perfect correspondence, since their rich stochastic dynamics cause spikes to quickly and chaotically diverge from simulation if the processor misses even a single neural operation. To measure the maximum speed at which we can run the system, we increased the time step frequency

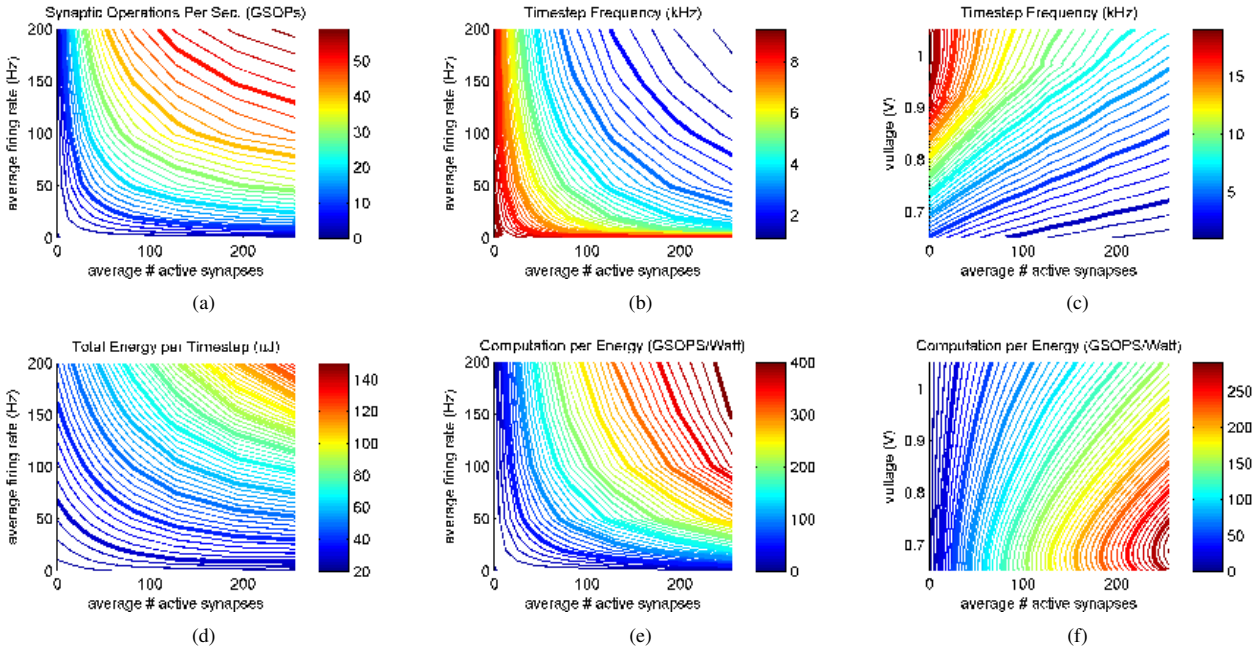


Fig. 5. TrueNorth Characterization. (a) Computation per time (GSOPS): rate vs. synapses at 0.75V. (b) Maximum time step operation frequency (kHz): rate vs. synapses at 0.75V. (c) Maximum time step operation frequency (kHz): voltage vs. synapses at an average firing rate of 50Hz. (d) Total energy per time step (μJ): rate vs. synapses at 0.75V. (e) Computation per energy (GSOPS/W): rate vs. synapses at 0.75V. (f) Computation per energy (GSOPS/W): voltage vs. synapses at an average firing rate of 50 Hz.

until the processor reported an execution error on any of the 88 probabilistically-generated recurrent neural networks. We repeated this test on neural models in which all synapses are active and every neuron spiked on every time step, the worst-case scenario.

B. TrueNorth Performance Characterization

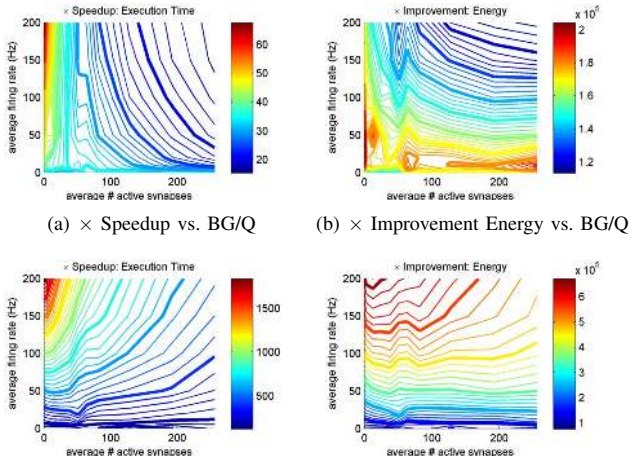
Using our set of 88 probabilistically-generated recurrent neural networks, we measured time and energy across the parameter space at different operating voltages, creating the contour plots in Fig. 5. SOPS increases with both active synapse count and firing rate (Fig. 5(a)), and correlates with total energy consumption (Fig. 5(d)). Although total energy is highest at high synaptic density and firing rate (upper right, Fig. 5(d)), TrueNorth is performing more computation in that region. The result is a more efficient use of the TrueNorth hardware, as quantified by computation per energy (Fig. 5(e)). When running a complex recurrent neural network with 20Hz average firing rate and 128 active synapses per neuron in real-time (updated once per millisecond), TrueNorth consumes merely 65mW and delivers 46 Giga-Synaptic Operations Per Second per Watt (GSOPS/W). Running this network $\sim$5$\times$ faster (amortizing passive power), TrueNorth delivers 81 GSOPS/W. A large fraction of the design space exceeds 100 GSOPS/W. For higher spike rates (200Hz) and higher synaptic utilization (256 per neuron), corresponding to the upper right corner, TrueNorth exceeds 400 GSOPS/W. Faster-than-real-time ($>$1kHz) operation is possible when active synapses are few and firing rates are low; that is, when the TrueNorth computational load is light (Fig. 5(b)). Computational efficiency (SOPS/W) increases as operating voltage is lowered (Fig. 5(f)). Maximum execution speed increases with voltage (Fig. 5(c)), but total power increases as voltage squared. Consequently, SOPS/W is maximized at lower voltages, limited only by the minimum voltage that can still ensure correct circuit-level functional operation ($\sim$700mV).

C. Characterization versus Compass on BG/Q and x86

To benchmark Compass performance on the BG/Q and x86 systems, we first measured execution time and energy for the 88 probabilistically-generated recurrent neural networks. Fig. 6 depicts TrueNorth's advantage in speed and energy consumption relative to the Compass simulator running on the BG/Q and x86 systems. Speedup is defined as the ratio of execution times for the same application on the von Neumann system versus TrueNorth: Speedup $= T_{\text{Proc}}/T_{\text{TrueNorth}}$. Similarly, $\times$ power and $\times$ energy improvement are ratios: $\times\text{Improvement}_{power} = P_{\text{Proc}}/P_{\text{TrueNorth}}$ and $\times\text{Improvement}_{energy} = E_{\text{Proc}}/E_{\text{TrueNorth}}$. TrueNorth executes 1 order of magnitude faster than Compass running on 32 hosts of BG/Q and two to three orders of magnitude faster than the x86 system. TrueNorth is five orders of magnitude more energy efficient than both systems, measured per time step, over the entire characterization space. Note that execution time and energy are physical invariants that do not rely on definitions of computation (that is, FLOPS or SOPS).

D. Application Performance Comparison

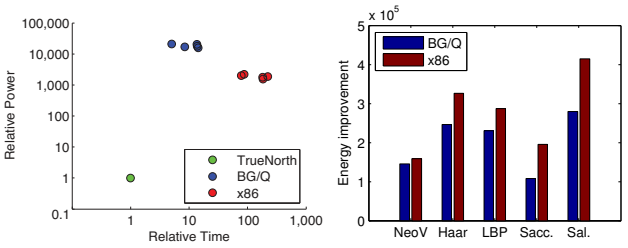
Next, we benchmarked the five computer vision applications described in Section IV-B on the TrueNorth processor and



(a) × Speedup vs. BG/Q (b) × Improvement Energy vs. BG/Q

(c) × Speedup vs. x86 (d) × Improvement Energy vs. x86

Fig. 6. TrueNorth performance vs. Compass: (a) one order of magnitude speedup of execution time vs. 32 host BG/Q, (b) five orders of magnitude reduction in energy vs. 32 host BG/Q, (c) two to three orders of magnitude speedup of execution time vs. dual socket x86, (d) five orders of magnitude reduction in energy vs. dual socket x86.



(a) (b)

Fig. 7. (a) Execution speedup vs. × power improvement, and (b) × energy improvement of TrueNorth processor versus Compass simulator running on up to 32 BG/Q processors and up to two x86 processors on five computer vision applications. Left to right: Neovision Multi-Object Detection and Classification, Haar wavelet features, Local Binary Patterns, Saccade map, and Saliency map.

on the BG/Q and x86 systems, (Fig. 7). One TrueNorth processor has a speedup of one and two orders of magnitude, respectively, over either a weak-scaling number of BG/Q processors (≈2 neuro-synaptic cores per thread, 32 threads per compute card) or two x86 processors, and consumes four and three orders of magnitude less power, respectively. Overall, TrueNorth uses over five orders of magnitude less energy per time step than Compass running on either the BG/Q or x86 systems. These speedups and energy improvements, shown in Fig. 7, are in line with those of the probabilistically-generated recurrent networks shown in Fig. 6.

E. BG/Q Characterization

Fig. 8 illustrates the run-times and corresponding energy consumptions for strong scaling runs for the Neovision application on BG/Q. We see significant speedups as we scale the number of processors and threads, but even the best operating point is 12× slower than real-time. In summary, a single host

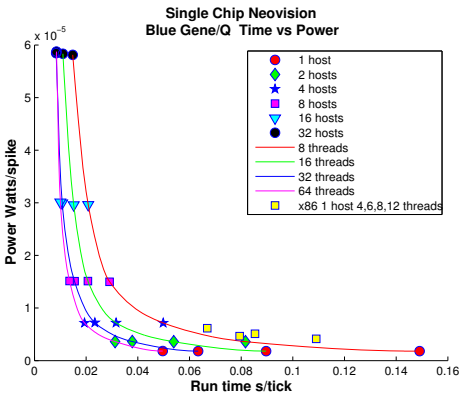


Fig. 8. Single Chip Neovision run time and power consumption on IBM Blue Gene/Q as a function of number of host processors and simulation threads.

is the most power-efficient but slowest; 32 hosts is the fastest but requires more power.

VII. Future Systems and Applications

TrueNorth's modular, scalable architecture and ultra-low power consumption provide an unique opportunity to create brain-inspired information technology systems with 100 trillion synapses, which is comparable to "human-scale." We conceive that such systems would constitute 96 racks, each rack with 4,096 TrueNorth processors and consuming merely 4kW. The key is to leverage TrueNorth's seamless tiling for low-power local-connectivity along with recent advances in interconnect technology for long-range connectivity.

A. Eight Board TrueNorth Array

Fig. 1(f) pictures eight boards, each containing a stand-alone network node with a single TrueNorth processor and a Zynq FPGA. We think of TrueNorth as "cortex" and the Zynq as "thalamus." The Zynq FPGA, with dual core ARM processors and programmable logic fabric, serves as an interface between the TrueNorth processor and the high-speed interconnection network. These boards were used to drive the characterization study and applications presented here.

B. 4 × 1 TrueNorth Array Board

Like the cortex, TrueNorth processors are designed to tile by communicating directly with each other without need for additional peripheral circuitry. Fig. 1(g) shows a 4 × 1 array of TrueNorth processors through which we have confirmed the operation of the asynchronous inter-chip communication channels, and which represents our first foray towards large-scale systems. This board includes four socketed TrueNorth processors which communicate via a native asynchronous bus protocol, Fig. 3(c).

C. 4 × 4 TrueNorth Array Board

Fig. 9 shows a 4 × 4 array of TrueNorth processors in sockets, representing 16 million neurons and 4 billion synapses. This board demonstrates TrueNorth's native 2D asynchronous

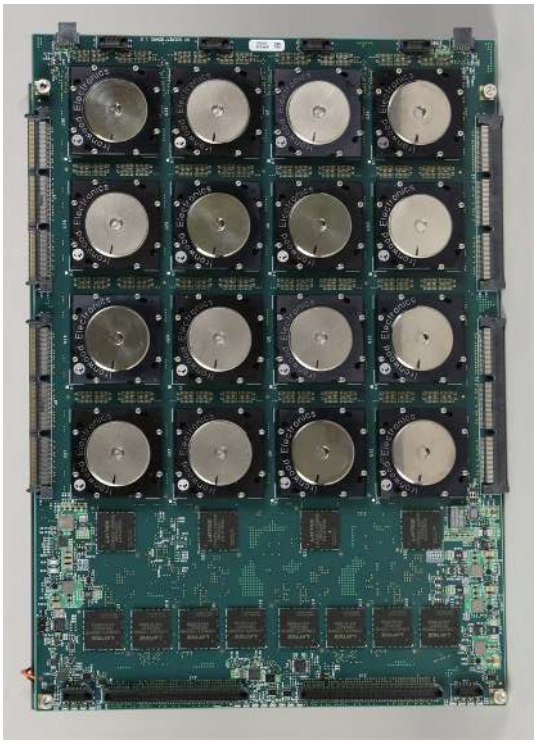


Fig. 9. The 16-chip board is organized as a 4×4 array of (socketed) TrueNorth Processors. Chips communicate natively with neighbors in 2D, implementing 16 million neurons and 4 billion synapses.

inter-chip communications interconnect. Board edge connectors enable board-level interconnection, continuing the tiled scalability. A column of low-power FPGAs interface between the TrueNorth compute array and a rear connected Zynq network interface module. Total board power, while running a 16M neuron network at real time is 7.2W, divided 2.5W and 4.7W between the TrueNorth array operating at 1.0V and the supporting logic (which includes the FPGAs) respectively.

D. Vision for Large-scale Systems

To achieve massive scale, we conceive a hierarchically interconnected TrueNorth system consisting of multi-chip boards, making up rack units, that are assembled into racks (Fig. 1, bottom). As a key building block, we imagine boards with 16 TrueNorth processors arranged in a 4×4 matrix, with a single FPGA communicating spikes and configuration data between an edge of the TrueNorth processor array and a networking interface (for example 1Gb Ethernet, PCIe, Infiniband). We conservatively budget 10W of total power per 4×4 processor board. Next, we imagine a quarter-rack unit with a passive backplane interconnecting 64 TrueNorth boards. We budget 1kW per fully populated backplane. This backplane unit could replicate, for $6400\times$ less energy, the "rat-scale" simulations that required 32 racks of Blue Gene/L [62] and yet ran $10\times$ slower than real-time [4]. Putting four quarter-rack units together along with high-performance networking switches and power supplies, we project a full rack unit with $4,096$ processors consuming approximately 4kW of power (with only $\sim$300 Watts attributed to TrueNorth processors). This single-rack system could replicate, for $128,000\times$ less energy, the "1% human-scale" simulations that required 16 racks of Blue Gene/P and ran $400\times$ slower than real-time [5].

To bring our low-power modular TrueNorth system vision to reality, we are investigating a number of research challenges. One challenge is selecting proper interconnect [63], [64], [65], [66] types and topologies suitable for large-scale TrueNorth applications, which may differ significantly from the interconnect of supercomputers used for physical simulation. Another challenge is maximizing the system's computation per energy while minimizing communication and system management overhead. The system must include traditional CPUs for system management and network training, as well as interface FPGAs to bridge between TrueNorth, CPUs, and high-speed interconnect. The correct ratio between TrueNorth processors and the support infrastructure is an open question. With low power a major requirement, the operating environment for a TrueNorth system is different from conventional supercomputers. TrueNorth processors may be densely packed together with minimal cooling and power distribution concerns, an advantage during system scale out. On the software side, defining a new programming paradigm that enables high productivity akin to MPI [47] or OpenMP [48] for today's supercomputers is a vital area of research. Learning large-scale neural networks that can take advantage of the enormous TrueNorth system scale is an important direction.

Such synaptic supercomputers, may enable a slew of applications in visual and auditory scene analysis and understanding; self-driving vehicles; medical image processing; multi-sensory feature extraction, classification, pattern recognition, association, context processing, abstraction, and understanding; financial services; and public safety as well as serve as platforms for studying learning, cognition, and phenomenological system-level neuroscience.

VIII. Conclusion

Over the past 6 years as part of the DARPA SyNAPSE program, we have created a end-to-end ecosystem that encompasses the entire development stack for neural-inspired applications. Algorithms and applications are first developed in our new programming language and environment; they are then simulated using our highly-optimized neural network simulator, Compass; and finally, the same networks are deployed on our real-time and energy-efficient neural-processor, TrueNorth. Using co-design as the design principle, Compass and TrueNorth are both expressions of the same underlying kernel. Although both expressions have equivalent functional behavior, TrueNorth hardware achieves a $\sim 100,000\times$ reduction in energy-to-solution and $\sim 100\times$ reduction in time-to-solution as compared to Compass, which has been meticulously-tuned for high performance on von Neumann microprocessors and supercomputers. This dramatic improvement is attained by a radically new, non-von Neumann, event-

based architecture that tightly integrates memory, computation, and communication into a neurosynaptic core and connects a massive number of such cores (4,096) via an event-driven network-on-chip. Unlike today's processors that are optimized for FLOPS, TrueNorth is optimized for SOPS (Synaptic Operations Per Second)—the fundamental unit of computation for large-scale spiking neural networks. Remarkably, TrueNorth achieves a performance of 46 Giga-SOPS/Watt while running a wide-range of benchmark networks. Given the complementary nature of FLOPS and SOPS, hybrid computers that combine today's processors with TrueNorth are inevitable. Because TrueNorth chips are designed to be seamlessly tiled (due to their native chip-to-chip communication interface), it is possible to create large-scale neurosynaptic supercomputers. We have already demonstrated a plethora of neural-inspired applications on Compass and TrueNorth, which include complex visual processing and pattern recognition tasks. Our ecosystem will form the foundation for ultra-low-power, compact, real-time, multi-modal sensorimotor information technology systems that are on the horizon, and that these will in turn provide enormous societal and economic benefits. In addition to the already demonstrated benefits of our ecosystem in terms of energy-to-solution, time-to-solution, SOPS/Watt performance, scale, and flexibility for applications, we believe that our multi-faceted approach will have far reaching impact. For example, our ecosystem is a powerful testimony to the co-design methodology that is becoming increasingly more relevant as Moore's law begins to stall and heterogeneous architectures begin to emerge. In addition, our work highlights the role of today's supercomputers as indispensable in bringing the vision of Compass and TrueNorth to reality. Realizing the full potential of the neurosynaptic ecosystem is a rewarding and challenging endeavor, which will require contributions and collaborations across the entire breadth and depth of machine learning, neural networks, computer vision and audition, neuroscience, robotics, computer architecture, circuit design, simulation methodology, programming languages, visualization, usability, design, and, needless to say, supercomputing.

ACKNOWLEDGMENTS

This research was sponsored by DARPA under contract No. HR0011-09-C-0002. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressly or implied, of DARPA or the U.S. Government.

REFERENCES

[1] D. S. Modha, R. Ananthanarayanan, S. K. Esser, A. Ndirango, A. J. Sherbondy, and R. Singh, "Cognitive computing," *Communications of the ACM*, vol. 54, no. 8, pp. 62–71, 2011.

[2] D. S. Modha and R. Singh, "Network architecture of the long distance pathways in the macaque brain," *Proceedings of the National Academy of the Sciences USA*, vol. 107, no. 30, pp. 13 485–13 490, 2010.

[3] A. S. Cassidy, P. Merolla, J. V. Arthur, S. Esser, B. Jackson, R. Alvarez-Icaza, P. Datta, J. Sawada, T. M. Wong, V. Feldman *et al.*, "Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores," in *International Joint Conference on Neural Networks (IJCNN). IEEE*, 2013.

[4] R. Ananthanarayanan and D. S. Modha, "Anatomy of a cortical simulator," in *Supercomputing 07*, 2007.

[5] R. Ananthanarayanan, S. K. Esser, H. D. Simon, and D. S. Modha, "The cat is out of the bag: cortical simulations with 10^9 neurons, 10^{13} synapses," in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*. IEEE, 2009, pp. 1–12.

[6] R. Preissl, T. M. Wong, P. Datta, M. Flickner, R. Singh, S. K. Esser, W. P. Risk, H. D. Simon, and D. S. Modha, "Compass: A scalable simulator for an architecture for cognitive computing," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. IEEE Computer Society Press, 2012, p. 54.

[7] T. M. Wong, R. Preissl, P. Datta, M. Flickner, R. Singh, S. K. Esser, E. McQuinn, R. Appuswamy, W. P. Risk, H. D. Simon *et al.*, "10^{14}," *IBM Research Divsion, Research Report RJ10502*, 2012.

[8] P. Merolla, J. Arthur, F. Akopyan, N. Imam, R. Manohar, and D. S. Modha, "A digital neurosynaptic core using embedded crossbar memory with 45pJ per spike in 45nm," in *Custom Integrated Circuits Conference (CICC)*. IEEE, 2011.

[9] E. McQuinn, P. Datta, M. Flickner, W. Risk, D. Modha, T. Wong, S. Raghav, and R. Appuswamy, "International science & engineering visualization challenge," *Science*, vol. 339, no. 6119, pp. 512–513, 2013.

[10] C. Mead, *Analog VLSI and neural systems*. Boston, MA: Addison-Wesley, 1989.

[11] W. D. Hillis, *The connection machine*. The MIT Press, 1989.

[12] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, 2014.

[13] S. Borkar, "Thousand core chips: A technology perspective," in *Proceedings of the 44th Annual Design Automation Conference (DAC)*. New York, NY, USA: ACM, 2007, pp. 746–749.

[14] R. Haring, M. Ohmacht, T. Fox, M. Gschwind, D. Satterfield, K. Sugavanam, P. Coteus, P. Heidelberger, M. Blumrich, R. Wisniewski *et al.*, "The IBM Blue Gene/Q compute chip," in *IEEE Micro*. IEEE Computer Society, 2012, pp. 48–60.

[15] D. J. Kerbyson, K. J. Barker, D. S. Gallo, D. Chen, J. R. Brunheroto, K. D. Ryu, G. L.-T. Chiu, and A. Hoisie, "Tracking the performance evolution of Blue Gene systems," in *ISC*, 2013, pp. 317–329.

[16] A. Gara, "The long term impact of codesign," in *High Performance Computing, Networking, Storage and Analysis (SCC)*. IEEE, 2012, pp. 2212–2246.

[17] D. J. Kerbyson, A. Vishnu, K. J. Barker, and A. Hoisie, "Codesign challenges for exascale systems: Performance, power, and reliability," *Computer*, vol. 44, no. 11, pp. 0037–43, 2011.

[18] S. K. Esser, A. Andreopoulos, R. Appuswamy, P. Datta, D. Barch, A. Amir, J. Arthur, A. S. Cassidy, P. Merolla, S. Chandra *et al.*, "Cognitive computing systems: Algorithms and applications for networks of neurosynaptic cores," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2013.

[19] A. Amir, P. Datta, W. Risk, A. S. Cassidy, J. A. Kusnitz, S. K. Esser, A. Andreopoulos, T. M. Wong, M. Flickner, R. Alvarez-Icaza *et al.*, "Cognitive computing programming paradigm: A corelet language for composing networks of neuro-synaptic cores," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2013.

[20] C.-J. Wang, C.-H. Wu, and S. Sivasindaram, "Neural network simulation on shared-memory vector multiprocessors," in *Proceedings of the ACM/IEEE Conference on Supercomputing*, 1989, pp. 197–204.

[21] S. Kunkel, T. C. Potjans, J. M. Eppler, H. E. E. Plesser, A. Morrison, and M. Diesmann, "Meeting the memory challenges of brain-scale network simulation," *Frontiers in neuroinformatics*, vol. 5, p. 35, 2012.

[22] M. Gewaltig and M. Diesmann, "NEST (NEural Simulation Tool)," *Scholarpedia*, vol. 2, no. 4, p. 1430, 2007.

[23] H. Miyazaki, Y. Kusano, H. Okano, T. Nakada, K. Seki, T. Shimizu, N. Shinjo, F. Shoji, A. Uno, and M. Kurokawa, "K computer: 8.162 petaflops massively parallel scalar supercomputer built with over 548k cores," in *International Solid-State Circuits Conference (ISSCC)*. IEEE, 2012, pp. 192–194.

[24] M. Djurfeldt, M. Lundqvist, C. Johansson, M. Rehn, O. Ekeberg, and A. Lansner, "Brain-scale simulation of the neocortex on the IBM Blue Gene/L supercomputer," *IBM Journal of Research and Development*, vol. 52, no. 1.2, pp. 31–41, Jan 2008.

[25] H. Markram, "The Blue Brain Project," *Nature Reviews Neuroscience*, vol. 7, no. 2, pp. 153–160, February 2006.

[26] R. D. Traub, D. Contreras, M. O. Cunningham, H. Murray, F. E. LeBeau, A. Roopun, A. Bibbig, W. B. Wilent, M. J. Higley, and M. A. Whittington, "Single-column thalamocortical network model exhibiting gamma oscillations, sleep spindles, and epileptogenic bursts," *Journal of Neurophysiology*, vol. 93, no. 4, pp. 2194–2232, 2005.

[27] J. M. Nageswaran, N. Dutt, J. L. Krichmar, A. Nicolau, and A. V. Veidenbaum, "A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors," *Neural Networks*, vol. 22, no. 5, pp. 791–800, 2009.

[28] R. Brette, M. Rudolph, T. Carnevale, M. Hines, D. Beeman, J. Bower, M. Diesmann, A. Morrison, P. Goodman, J. Harris, FrederickC. *et al.*, "Simulation of networks of spiking neurons: A review of tools and strategies," *Journal of Computational Neuroscience*, vol. 23, no. 3, pp. 349–398, 2007.

[29] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. Van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers in Neuroscience*, vol. 5, 2011.

[30] M. Mahowald and R. Douglas, "A silicon neuron," *Nature*, vol. 354, no. 6354, pp. 515–518, 1991.

[31] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, no. 6789, pp. 947–951, 2000.

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

[33] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gómez-Rodríguez, L. Camuñas-Mesa, R. Berner, M. Rivas-Pérez, T. Delbruck *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 Transactions on Neural Networks*, vol. 20, no. 9, 2009.

[34] 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 *Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2012, pp. 21–24.

[35] J. Schemmel, A. Grubl, S. Hartmann, A. Kononov, C. Mayr, K. Meier, S. Millner, J. Partzsch, S. Schiefer, S. Scholze *et al.*, "Live demonstration: A scaled-down version of the BrainScaleS wafer-scale neuromorphic system," in *International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2012, pp. 702–702.

[36] E. Stromatias, F. Galluppi, C. Patterson, and S. Furber, "Power analysis of large-scale, real-time neural networks on SpiNNaker," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2013, pp. 1–8.

[37] T. Ishiyama, K. Nitadori, and J. Makino, "4.45 Pflops astrophysical N-body simulation on K computer: The gravitational trillion-body problem," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. Los Alamitos, CA, USA: IEEE Computer Society Press, 2012.

[38] D. E. Shaw, R. O. Dror, J. K. Salmon, J. Grossman, K. M. Mackenzie, J. A. Bank, C. Young, M. M. Deneroff, B. Batson, K. J. Bowers *et al.*, "Millisecond-scale molecular dynamics simulations on Anton," in *Proceedings of the International Conference on High Performance Computing Networking, Storage and Analysis*. IEEE, 2009, pp. 1–11.

[39] A. Rahimian, I. Lashuk, S. Veerapaneni, A. Chandramowlishwaran, D. Malhotra, L. Moon, R. Sampath, A. Shringarpure, J. Vetter, R. Vuduc *et al.*, "Petascale direct numerical simulation of blood flow on 200k cores and heterogeneous architectures," in *Proceedings of the International Conference on High Performance Computing Networking, Storage and Analysis*. IEEE Computer Society, 2010, pp. 1–11.

[40] Y. Hasegawa, J.-I. Iwata, M. Tsuji, D. Takahashi, A. Oshiyama, K. Minami, T. Boku, F. Shoji, A. Uno, M. Kurokawa *et al.*, "First-principles calculations of electron states of a silicon nanowire with 100,000 atoms on the K computer," in *Proceedings of the International Conference on High Performance Computing Networking, Storage and Analysis*. ACM, 2011.

[41] M. Eisenbach, C.-G. Zhou, D. M. Nicholson, G. Brown, J. Larkin, and T. C. Schulthess, "A scalable method for ab initio computation of free energies in nanoscale systems," in *Proceedings of the International Conference on High Performance Computing Networking, Storage and Analysis*. ACM, 2009, p. 64.

[42] V. N. Gamezo, A. M. Khokhlov, E. S. Oran, A. Y. Chtchelkanova, and R. O. Rosenberg, "Thermonuclear supernovae: Simulations of the deflagration stage and their implications," *Science*, vol. 299, no. 5603, pp. 77–81, 2003.

[43] A. Nakano, R. K. Kalia, P. Vashishta, T. J. Campbell, S. Ogata, F. Shimojo, and S. Saini, "Scalable atomistic simulation algorithms for materials research," *Scientific Programming*, vol. 10, no. 4, pp. 263–270, 2002.

[44] K. Pingali and A. Rogers, "Compiling for locality." in *ICPP*, 1990, pp. 142–146.

[45] V. B. Mountcastle, "Modality and topographic properties of single neurons of cat's somatic sensory cortex." *Journal of Neurophysiology*, vol. 20, no. 4, p. 408, 1957.

[46] S. B. Laughlin and T. J. Sejnowski, "Communication in neuronal networks," *Science*, vol. 301, no. 5641, pp. 1870–1874, 2003.

[47] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: portable parallel programming with the message-passing interface*. MIT press, 1999.

[48] L. Dagum and R. Menon, "OpenMP: an industry standard API for shared-memory programming," *Computational Science & Engineering, IEEE*, vol. 5, no. 1, pp. 46–55, 1998.

[49] W. J. Dally and C. L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Transactions on Computers*, vol. 100, no. 5, pp. 547–553, 1987.

[50] V. J. Wedeen, D. L. Rosene, R. Wang, G. Dai, F. Mortazavi, P. Hagmann, J. H. Kaas, and W.-Y. I. Tseng, "The geometric structure of the brain fiber pathways," *Science*, vol. 335, no. 6076, pp. 1628–1634, 2012.

[51] G. S. Banavar, "An application framework for compositional modularity," Ph.D. dissertation, The University of Utah, 1995.

[52] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition*, 2001.

[53] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[54] A. Andreopoulos and J. K. Tsotsos, "50 years of object recognition: Directions forward," *Computer Vision and Image Understanding*, vol. 117, no. 8, pp. 827–891, 2013.

[55] "NeoVision2 dataset," http://ilab.usc.edu/neo2/dataset/, 2013.

[56] R. Kasturi, D. Goldgof, R. Ekambaram, G. Pratt, E. Krotkov, D. D. Hackett, Y. Ran, Q. Zheng, R. Sharma, M. Anderson *et al.*, "Performance evaluation of neuromorphic-vision object recognition algorithms," in *Proc. 22nd International Conference on Pattern Recognition (ICPR'14)*, Aug 2014.

[57] R. W. Wisniewski, "BlueGene/Q: Architecture, codesign; path to exascale," http://projects.csail.mit.edu/caos/2012-01-25-caos-bgq-v1-ed.pdf.

[58] M. Gilge *et al.*, *IBM System Blue Gene Solution Blue Gene/Q Application Development*. IBM Redbooks, 2013.

[59] G. Lakner and B. Knudson, "IBM system Blue Gene solution: Blue Gene/Q system administration. IBM redbooks, june 2012."

[60] "Performance application programming interface," http://icl.cs.utk.edu/papi.

[61] "Intel 64 and IA-32 Architectures Software Developer's Manual, Volume 3B: System Programming Guide, Part 2," http://download.intel.com/products/processor/manual/253669.pdf.

[62] A. Gara, M. A. Blumrich, D. Chen, G. L.-T. Chiu, P. Coteus, M. E. Giampapa, R. A. Haring, P. Heidelberger, D. Hoenicke, G. V. Kopcsay *et al.*, "Overview of the Blue Gene/L system architecture," *IBM J. Res. Devel.*, vol. 49, pp. 195–212, 2005.

[63] S. Kamil, L. Oliker, A. Pinar, and J. Shalf, "Communication requirements and interconnect optimization for high-end scientific applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 2, pp. 188–202, 2010.

[64] J. S. Vetter, V. Tipparaju, W. Yu, and P. C. Roth, "HPC interconnection networks: The key to exascale computing," in *Advances in Parallel Computing: High Speed and Large Scale Scientific Computing*. IOS Press, 2009, vol. 18, pp. 95–106.

[65] C. E. Leiserson, "Fat-trees: universal networks for hardware-efficient supercomputing," *IEEE Transactions on Computers*, vol. 100, no. 10, pp. 892–901, 1985.

[66] G. Pratt, S. Ward, J. Nguyen, and J. Pezaris, "The diamond interconnect," MIT Technical Report, Tech. Rep., 1993.