A Million Spiking-Neuron Integrated Circuit with a Scalable Communication Network and Interface

Paul A. Merolla[1*], John V. Arthur[1*], Rodrigo Alvarez-Icaza[1*], Andrew S. Cassidy[1*], Jun Sawada[2*], Filipp Akopyan[1*], Bryan L. Jackson[1*], Nabil Imam[3], Chen Guo[4], Yutaka Nakamura[5], Bernard Brezzo[6], Ivan Vo[2], Steven K. Esser[1], Rathinakumar Appuswamy[1], Brian Taba[1], Arnon Amir[1], Myron D. Flickner[1], William P. Risk[1], Rajit Manohar[7] & Dharmendra S. Modha[1†]

[1]IBM Research - Almaden, 650 Harry Road, San Jose, California 95120, USA,
[2]IBM Research - Austin, 11501 Burnet Road, Austin, TX 78758, USA,
[3]Cornell University, 358 Upson Hall, Ithaca, NY 14853 USA,
[4]IBM Engineering and Technology Services, San Jose Design Center,
650 Harry Road, San Jose, California 95120, USA,
[5]IBM Research - Tokyo, NBF Toyosu Canal Front Building
5-6-52 Toyosu, Koto-ku Tokyo, 135-8511 Japan,
[6]IBM T. J. Watson Research Center,
101 Kitchawan Rd, Yorktown Heights, NY 10598,
[7]Cornell Tech, 111 Eighth Avenue #302, New York, NY 10011, USA,

[†]To whom correspondence should be addressed; E-mail: dmodha@us.ibm.com.

[*]These authors contributed equally.

Inspired by the brain's structure, we have developed an efficient, scalable, and flexible non-von Neumann architecture that leverages contemporary silicon technology. To demonstrate, we built a 5.4 billion transistor chip with $4,096$ neurosynaptic cores interconnected via an intra-chip network that integrates one million programmable spiking neurons and 256 million configurable synapses. Chips can be tiled in two-dimensions via an inter-chip communication interface, seamlessly scaling the architecture to a cortex-like sheet of arbitrary size. The architecture is well-suited to many applications that use complex neural networks in real-time; for example, multi-object detection and classification. With 400×240-pixel video input at 30 frames-per-second, the chip consumes 63mW.

A long-standing dream (*1, 2*) has been to harness neuroscientific insights to build a versatile computer that is: efficient in terms of energy and space; homogeneously scalable to large networks of neurons and synapses; and flexible enough to run complex behavioral models of the neocortex (*3, 4*) as well as networks inspired by neural architectures (*5*).

No such computer exists today. The von Neumann architecture is fundamentally inefficient and non-scalable for representing massively interconnected neural networks (Fig. 1) with respect to computation, memory, and communication (Fig. 1B). Mixed analog–digital neuromorphic approaches have built large-scale systems (*6–8*) to emulate neurobiology by using custom computational elements, for example, silicon neurons (*9, 10*), winner-take-all circuits (*11*), and sensory circuits (*12*). We have found that a multiplexed digital implementation of spiking neurons is more efficient than previous designs (section S3) and enables one-to-one correspondence between software and hardware (section S9). Mixed analog–digital as well as custom microprocessor-based neuromorphic approaches (*13*) have built event-driven communication frameworks (*14*) to emulate the interconnectivity of the brain by leveraging the speed of digital electronics. We have found that event-driven communication combined with co-located memory and computation mitigates the von Neumann bottleneck (*15*). Inspired by neuroscience (Fig. 2A, 2B, 2C), our key architectural abstraction (Fig. 1C) is a network of neurosynaptic cores that can implement large-scale spiking neural networks that are efficient, scalable, and flexible, within today's technology.

From a structural view, the basic building block is a core, a self-contained neural network with 256 input lines (axons) and 256 outputs (neurons) connected via 256×256 directed, programmable synaptic connections (Fig. 2D). Building on the local, clustered connectivity of a single neurosynaptic core, we constructed more complex networks by wiring multiple cores together using global, distributed on- and off-chip connectivity (Fig. 2E and 2F). Each neuron on every core can target an axon on any other core. Therefore, axonal branching is imple-

mented hierarchically, in two stages: first, a *single* connection travels a long distance between cores (akin to an axonal trunk), and second, upon reaching its target axon, fans out into *multiple* connections that travel a short distance within a core (akin to an axonal arbor). Neuron dynamics is discretized into 1ms time steps set by a global 1kHz clock. Other than this global synchronization signal, which ensures one-to-one equivalence between software and hardware, cores operate in a parallel and event-driven fashion (section S1). The fundamental currency that mediates fully asynchronous (*16*) inter-core communication and event-driven intra-core computation is all-or-nothing spike events that represent firing of individual neurons. The architecture is *efficient* because (i) neurons form clusters that draw their inputs from a similar pool of axons (*17–19*) (Fig. 2A) allowing for memory–computation co-localization (section S5); (ii) only spike events, which are sparse in time, are communicated between cores via the long-distance communication network; and (iii) active power is proportional to firing activity. The architecture is *scalable* because (i) cores on a chip as well as chips themselves can be tiled in two-dimensions similar to the mammalian neocortex (Fig. 2B and 2C); (ii) each spike event addresses a pool of neurons on a target core, reducing the number of long-range spike events thus mitigating a critical bottleneck (section S4); and (iii) occasional defects at the core and chip level do not disrupt system usability. Finally, the architecture is *flexible* because (i) each neuron is individually configurable and the neuron model (*20*) supports a wide variety of computational functions and biologically-relevant spiking behaviors; (ii) each synapse can be turned on or off individually and post-synaptic efficacy can be assigned relative strengths; (iii) each neuron–axon connection is programmable along with its axonal delay; and (iv) the neurons and synapses can exhibit programmed stochastic behavior via a pseudo-random number generator (one per core). The architecture thus supports rich physiological dynamics and anatomical connectivity that includes feed-forward, recurrent, and lateral connections.

From a functional view, a core has individually addressable axons, a configurable synap-

tic crossbar array, and programmable neurons (Fig. 2G). Within a core, information flows from presynaptic axons (horizontal lines), through the active synapses in the crossbar (binary-connected crosspoints), to drive inputs for all the connected postsynaptic neurons (vertical lines). Axons are activated by input spike events, which are generated by neurons anywhere in the system and delivered after a desired axonal delay of between 1 to 15 time steps. While the brain has a dedicated wire for each connection, in our architecture spike-events are carried between cores by time-multiplexed wires (*21*) that interconnect a two-dimensional mesh network of routers, each with five ports (north, south, east, west, and local). The routers form the backbone of a two-dimensional mesh network interconnecting a 64×64 core array (Fig. 2H). When a neuron on a core spikes, it looks up in local memory an axonal delay (4 bits) and the destination address (8 bit absolute address for the target axon, and two 9 bit relative addresses representing core hops in each dimension to the target core). This information is encoded into a packet that is injected into the mesh, where it is handed from core to core—first in the x dimension then in the y dimension (deadlock-free dimension-order routing)—until it arrives at its target core before fanning out via the crossbar (Fig. S2). To implement feedback connections within a core, where a neuron connects to an axon on the same core, the packet is delivered using the router's local channel, which is efficient because it never leaves the core. To scale the two-dimensional mesh across chip boundaries where the number of inter-chip connections is limited, we used a merge–split structure at the four edges of the mesh to serialize exiting spikes and deserialize entering spikes (Fig. 2I). Spikes leaving the mesh are tagged with their row (for spikes traveling east–west) or column (for spikes traveling north–south) before being merged onto a shared link that exits the chip. Conversely, spikes entering the chip from a shared link are split to the appropriate row or column using the tagged information.

From a physical view, to implement this functional blueprint, we built TrueNorth, a fully functional digital chip (section S6) with one million spiking-neurons and 256 million synapses

(non-plastic). With 5.4 billion transistors occupying 4.3cm^2 area in Samsung's 28nm process technology, TrueNorth has ~ 4.28 million bits of on-chip memory. Each core has $104,448$ bits of local memory to store synapse states ($65,536$ bits), neuron states and parameters ($31,232$ bits), destination addresses ($6,656$ bits), and axonal delays ($1,024$ bits). In terms of efficiency, TrueNorth's power density is 20mW per cm^2 while that of a typical CPU is $50-100$W per cm^2 (Fig. 1A). Active power density was low because of our architecture and passive power density was low because of process technology choice with low-leakage transistors. This work advances a previous experimental prototype single neurosynaptic core (*22*)—scaling the number of cores by $4,096$ times and reducing core area by 15 times and power by 100 times. To enable an event-driven, hybrid asynchronous-synchronous approach, we were required to develop a custom tool flow, outside the scope of commercial software, for simulation and verification (section S2).

We used our software ecosystem (section S9) to map many well-known algorithms to the architecture (*23*) via offline learning, for example, convolutional networks, liquid state machines, restricted Boltzmann machines, hidden Markov models, support vector machines, optical flow, and multi-modal classification. These same algorithms *now* run without modification on TrueNorth. To test TrueNorth's applicability to real world problems, we developed an additional multi-object detection and classification application in a fixed-camera setting. The task had two challenges: (i) to detect people, bicyclists, cars, trucks, and buses that occur sparsely in images while minimizing false detection, and (ii) to correctly identify the object. Operating on a 400×240-pixel aperture, the chip consumed 63mW on a 30 frame-per-second three-color video (Fig. 3), which when scaled to a 1920×1080 video achieved state-of-the-art performance (section S11). Because the video was pre-recorded using a standard camera, we were required to convert the pixels into spike events to interface with TrueNorth. In a live setting, we could use a spike-based retinal camera (*12*) similar to a previously demonstrated eye-detection application (*23*). We also implemented a visual map of orientation-selective filters, inspired by

early processing in mammalian visual cortex (*24*), and commonly used in computer vision for feature extraction (section S10). All one million neurons received feed-forward inputs with an orientation bias from visual space as well as recurrent connections between nearby features to sharpen selectivity.

The standard benchmark of a computer architecture's efficiency is energy per operation. In the domain of configurable neural architectures, the fundamental operation is the synaptic event, which corresponds to a source neuron sending a spike event to a target neuron via a unique (non-zero) synapse. Synaptic events are the appropriate atomic units because the computation, memory, communication, power, area, and speed all scale with number of synapses. Using complex recurrently-connected networks (Fig. 4A), we measured the total power of TrueNorth under a range of configurations (Fig. 4B) and computed the energy per synaptic event (Fig. 4C) (section S7). Power consumption in TrueNorth is a function of spike rate, the average distance traveled by spikes, and the average number of active synapses per neuron (synaptic density). At the operating point where neurons fire on average at 20Hz and have 128 active synapses, the total measured power was 72mW (at 0.775V operating voltage), corresponding to 26pJ per synaptic event (considering total energy). Compared with an optimized simulator (*25*) running the exact same network on a modern general-purpose microprocessor, TrueNorth consumes $176,000$ times less energy per event (section S12). Compared with a state-of-the-art multiprocessor neuromorphic approach (*13*), (48 chips each with 18 microprocessors) running a similar network, TrueNorth consumes 769 times less energy per event (section S12). Direct comparison to these platforms is possible because, like TrueNorth, they support individually programmable neurons and connections, as required to run applications like our multi-object recognition example. Direct comparisons with other platforms is not possible because of different network constraints and system capabilities (section S13). Computation in TrueNorth is measured using synaptic operations per second (SOPS) while in modern supercomputers it is

floating-point operations per second (FLOPS). While not a direct comparison, TrueNorth can deliver 46 billion SOPS per Watt for a typical network and 400 billion SOPS per Watt for networks with high spike rates and high number of active synapses (section S8), whereas today's most energy-efficient supercomputer achieves 4.5 billion FLOPS per Watt.

We have begun building neurosynaptic supercomputers by tiling multiple TrueNorth chips, creating systems with hundreds of thousands of cores, hundreds of millions of neurons, and hundreds of billion of synapses. We envisage hybrid computers that combine the von Neumann architecture with TrueNorth—both being Turing-complete, but efficient for complementary classes of problems. We may be able to map the existing body of neural networks algorithms to the architecture in an efficient fashion. In addition, many of the functional primitives of a recent large-scale complex behavioral model (*3*) map natively to our architecture, and we foresee developing a compiler to translate high-level functional tasks directly into TrueNorth networks. We envision augmenting our neurosynaptic cores with synaptic plasticity (see ref (*26*) for a prototype), to create a new generation of field-adaptable neurosynaptic computers capable of online learning. While today TrueNorth is fabricated using a modern CMOS process, the underlying architecture may exploit advances in future memory (*27*), logic (*28*), and sensor (*12*) technologies to deliver lower power, denser form factor, and faster speed.

References and Notes

1. J. von Neumann, *The computer and the brain* (Yale University Press, New Haven, CT, 1958).

2. C. Mead, *Analog VLSI and neural systems* (Addison-Wesley, Boston, MA, 1989).

3. C. Eliasmith, *et al.*, *Science* **338**, 1202 (2012).

4. M. Mishkin, L. G. Ungerleider, K. A. Macko, *Trends in neurosciences* **6**, 414 (1983).

5. P. S. Churchland, T. J. Sejnowski, *The computational brain.* (The MIT press, 1992).

6. T. Yu, J. Park, S. Joshi, C. Maier, G. Cauwenberghs, *Biomedical Circuits and Systems Conference (BioCAS), 2012 IEEE* (IEEE, 2012), pp. 21–24.

7. J. Schemmel, *et al.*, *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on* (IEEE, 2012), pp. 702–702.

8. B. V. Benjamin, *et al.*, *Proceedings of the IEEE* **102**, 699 (2014).

9. M. Mahowald, R. Douglas, *Nature* **354**, 515 (1991).

10. G. Indiveri, *et al.*, *Frontiers in Neuroscience* **5** (2011).

11. R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, H. S. Seung, *Nature* **405**, 947 (2000).

12. S.-C. Liu, T. Delbruck, *Current Opinion in Neurobiology* **20**, 288 (2010).

13. E. Stromatias, F. Galluppi, C. Patterson, S. Furber, *International Joint Conference on Neural Networks (IJCNN). IEEE* (IEEE, 2013), pp. 1–8.

14. M. Mahowald, *An analog VLSI system for stereoscopic vision* (Springer, 1994).

15. J. Backus, *Communications of the ACM* **21**, 613 (1978).

16. J. Teifel, R. Manohar, *Proc. of 10th International Symposium on Asynchronous Circuits and Systems* (2004), pp. 17–27.

17. R. J. Douglas, K. A. Martin, *Annu. Rev. Neurosci.* **27**, 419 (2004).

18. S. B. Laughlin, T. J. Sejnowski, *Science* **301**, 1870 (2003).

19. V. B. Mountcastle, *Journal of Neurophysiology* **20**, 408 (1957).

20. A. S. Cassidy, *et al.*, *International Joint Conference on Neural Networks (IJCNN). IEEE* (2013).

21. W. J. Dally, B. Towles, *Design Automation Conference, 2001. Proceedings* (IEEE, 2001), pp. 684–689.

22. P. Merolla, *et al.*, *Custom Integrated Circuits Conference (CICC), 2011 IEEE* (IEEE, 2011), pp. 1–4.

23. S. K. Esser, *et al.*, *International Joint Conference on Neural Networks (IJCNN). IEEE* (2013).

24. D. H. Hubel, T. N. Wiesel, *The Journal of Physiology* **160**, 106 (1962).

25. R. Preissl, *et al.*, *High Performance Computing, Networking, Storage and Analysis (SC), 2012 International Conference for* (2012), pp. 1–11.

26. J. Seo, *et al.*, *Custom Integrated Circuits Conference (CICC), 2011 IEEE* (IEEE, 2011), pp. 1–4.

27. T. Ohno, *et al.*, *Nature Materials* **10**, 591 (2011).

28. M. M. Shulaker, *et al.*, *Nature* **501**, 526 (2013).

29. C. Isci, Workload adaptive power management with live phase monitoring and prediction, Ph.D. thesis, Princeton University (2007).

30. D. S. Modha, R. Singh, *Proceedings of the National Academy of Sciences* **107**, 13485 (2010).

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. We are grateful to many collaborators: Ankur Agrawal, Alexander Andreopoulos, Sameh Asaad, Christian Baks, Davis Barch, Michael Beakes, Ralph Bellofatto, David Berg, James Bong, Todd Christensen, Aaron Cox, Pallab Datta, Dan Friedman, Sue Gilson, Jay Guan, Scott Hall, Ruud Haring, Chuck Haymes, John Ho, Subramanian Iyer, James Krawiecki, Jeff Kusnitz, Jerry Liu, JB Kuang, Emmett McQuinn, Roger Mousalli, Ben Nathanson, Tapan Nayak, Don Nguyen, Hao Nguyen, Tuyet Nguyen, Norm Pass, Kavita Prasad, Martin Ohmacht, Carlos Ortega-Otero, Ziad Saliba, David L Satterfield, Jae-sun Seo, Ben Shaw, Koki Shimohashi, Kurt Shiraishi, Arash Shokoubakhsh, Raghavendra Singh, Todd Takken, Frank Tsai, Jose Tierno, Kyle Wecker, Shao-yi Wang, and Theodore Wong. We thank two anonymous reviewers and the editor, Peter Stern, for their thoughtful comments.

Supplementary Materials www.sciencemag.org/cgi/content/full/.....

Supplementary Text (sections S1 to S13)

Figs. S1 to S8

Tables S1 to S2

References *(31-40)*

Movie S1

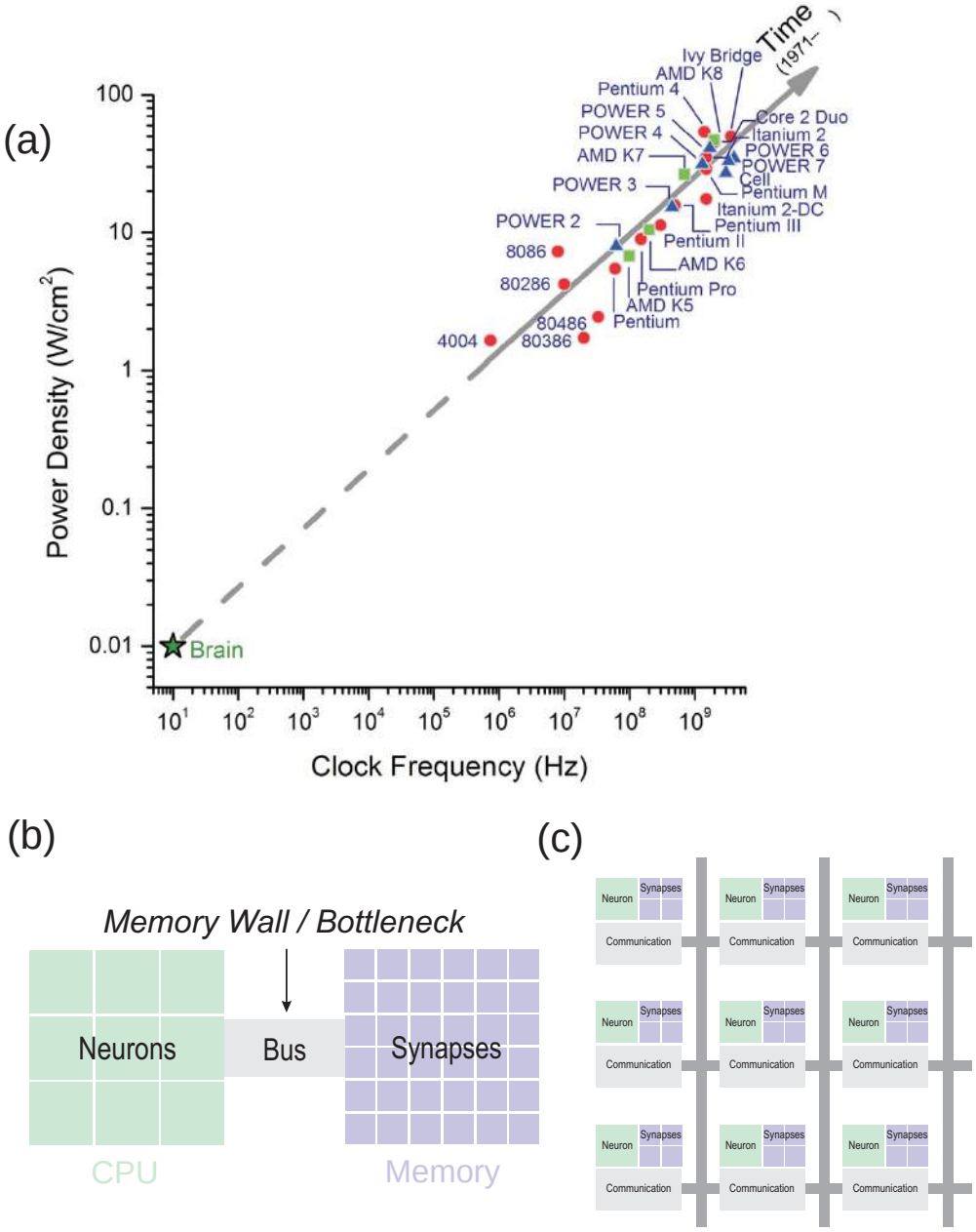


Fig. 1: **Computation, communication, and memory.** **(A)** The parallel, distributed architecture of the brain is different from the sequential, centralized von Neumann architecture of today's computers. The trend of increasing power densities and clock frequencies of processors (*29*) is headed away from the brain's operating point. **(B)** In terms of computation, a single processor has to simulate both a large number of neurons as well as the inter-neuron communication infrastructure. In terms of memory, the von Neumann bottleneck (*15*) which is caused by separation between the external memory and processor leads to energy-hungry data movement when updating neuron states and when retrieving synapse states. In terms of communication, inter-processor messaging (*25*) explodes when simulating highly-interconnected networks that do not fit on a single processor. **(C)** Conceptual blueprint of an architecture that, like the brain, tightly integrates memory, computation, and communication in distributed modules that operate in parallel and communicate via an event-driven network.

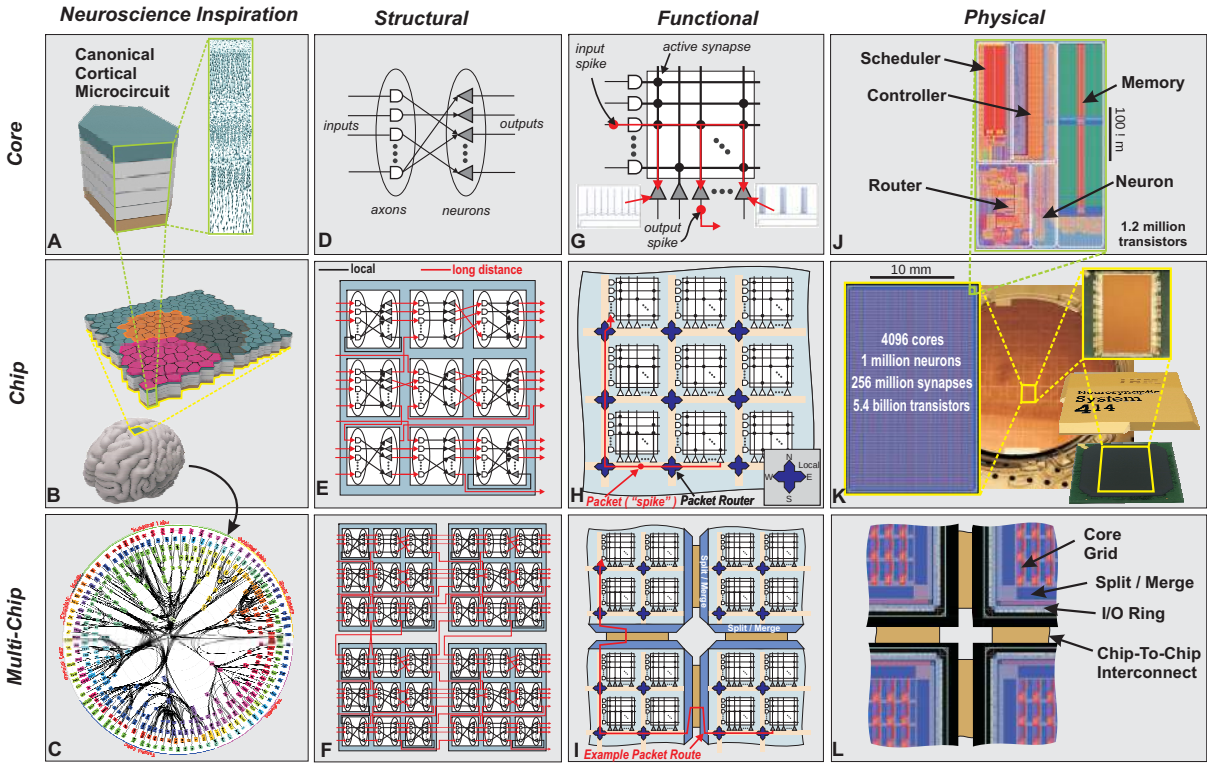


Fig. 2: **TrueNorth architecture.** Panels are organized into rows at three different scales: core, chip, and multi-chip; and into columns at four different views: neuroscience inspiration, structural, functional, and physical. **(A)** The neurosynaptic core is loosely inspired by the idea of a canonical cortical microcircuit. **(B)** A network of neurosynaptic cores is inspired by the cortex's two-dimensional sheet. **(C)** The multi-chip network is inspired by the long-range connections between cortical regions, shown here from the macaque brain (*30*). **(D)** Structure of a neurosynaptic core with axons as inputs, neurons as outputs, and synapses as directed connections from axons to neurons. **(E)** Multi-core network at chip scale and **(F)** multi-chip scale are both created by connecting a neuron on any core to an axon on any core with point-to-point connections. **(G)** Functional view of core as a crossbar where horizontal lines are axons, cross points are individually programmable synapses, vertical lines are neuron inputs, and triangles are neurons. Information flows from axons via active synapses to neurons. Neuron behaviors are individually programmable, with two examples shown. **(H)** Functional chip architecture is a two-dimensional array of cores where long-range connections are implemented by sending spike events (packets) over a mesh routing network to activate a target axon. Axonal delay is implemented at the target. **(I)** Routing network extends across chip boundaries through peripheral merge and split blocks. **(J)** Physical layout of core in 28nm CMOS fits in a $240\mu m \times 390\mu m$ footprint. A memory (SRAM) stores all the data for each neuron, a time-multiplexed neuron circuit updates neuron membrane potentials, a scheduler buffers incoming spike events to implement axonal delays, a router relays spike events, and an event-driven controller orchestrates the core's operation. **(K)** Chip layout of 64×64 cores, wafer, and chip package. **(L)** Chip periphery to support multi-chip networks.

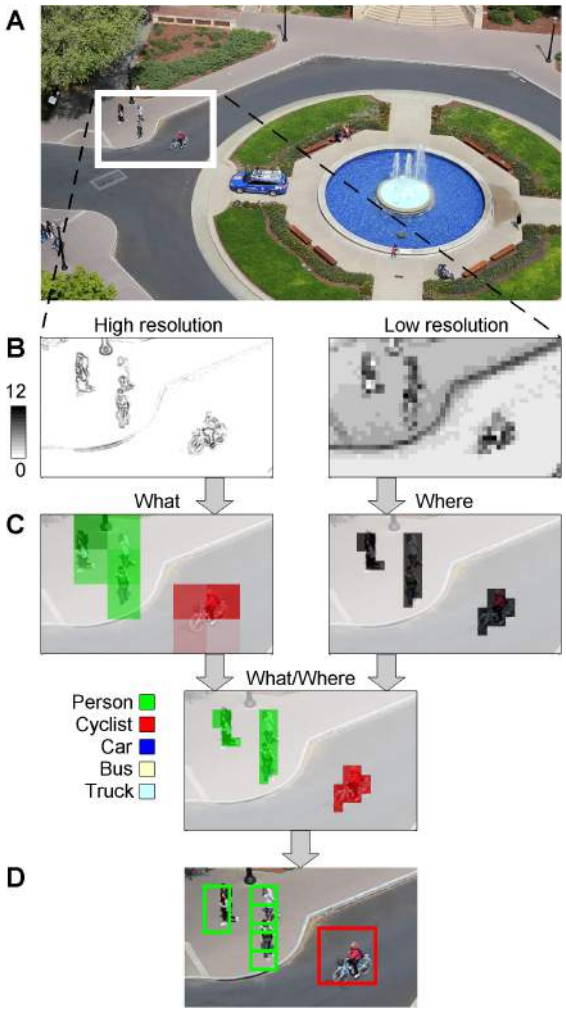


Fig. 3: **Real-time multi-object recognition on TrueNorth. (A)** The Neovision2 Tower dataset is a video from a fixed camera, where the objective is to identify the labels and locations of objects among five classes. We show an example frame along with the selected region that is input to the chip. **(B)** The region is transduced from pixels into spike events to create two parallel channels: a high resolution channel (left) that represents the *What* pathway for labeling objects; and a low resolution channel (right) that represents the *Where* pathway for locating salient objects. These pathways are inspired by dorsal and ventral streams in visual cortex (*4*). **(C)** *What* and *Where* pathways are combined to form a *What–Where* map. In the *What* network, colors represent the spiking activity for a grid of neurons, where different neurons were trained (offline) to recognize different object types. By overlaying the responses, brighter colors indicate more confident labels. In the *Where* network, neurons were trained (offline) to detect salient regions, and darker responses indicate more salient regions. **(D)** Object bounding boxes reported by the chip.

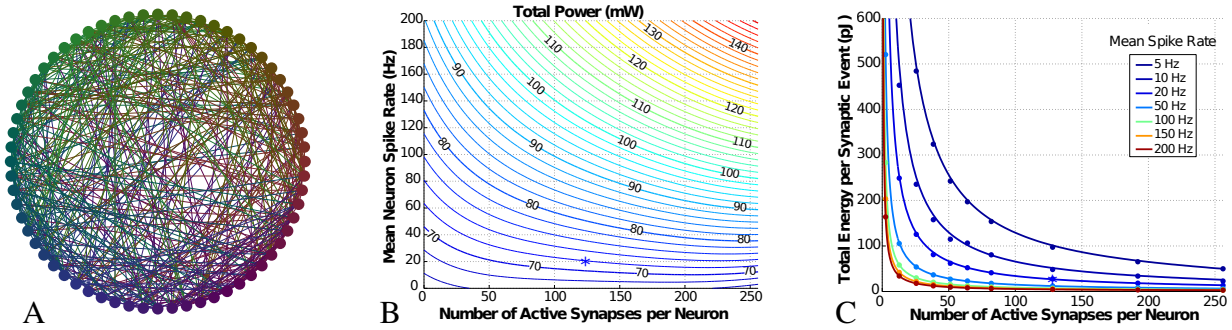


Fig. 4: **Benchmarking power and energy.** **(A)** Example network topology used for benchmarking power at real-time operation. Nodes represent cores and edges represent neural connections; only 64 of $4,096$ cores are shown. **(B)** Although power remains low (< 150mW) for all benchmarks networks, those with higher synaptic densities and higher spike rates consume more total power, which illustrates that power consumption scales with neuron activity and number of active synapses. **(C)** The total energy (passive plus active) per synaptic event decreases with higher synaptic density because leakage power and baseline core power are amortized over additional synapses. For a typical network where neurons fire on average at 20Hz and have 128 active synapses (marked as $*$ on panels B and C), the total energy is 26pJ per synaptic event.

Supplementary Material for

A Million Spiking-Neuron Integrated Circuit with a Scalable Communication Network and Interface

Paul A. Merolla*, John V. Arthur*, Rodrigo Alvarez-Icaza*,
Andrew S. Cassidy*, Jun Sawada*, Filipp Akopyan*, Bryan L. Jackson*,
Nabil Imam, Chen Guo, Yutaka Nakamura, Bernard Brezzo,
Ivan Vo, Steven K. Esser, Rathinakumar Appuswamy,
Brian Taba, Arnon Amir, Myron D. Flickner,
William P. Risk, Rajit Manohar & Dharmendra S. Modha†

†To whom correspondence should be addressed; E-mail: dmodha@us.ibm.com.
*These authors contributed equally.

This PDF file includes:

Supplementary Text (sections S1 to S13)
Full Reference List
Figs. S1 to S8
Tables S1 to S2
Caption for Movie S1

Other Supplementary Materials for this manuscript includes the following:

Movie S1

Supplementary Text: Architecture & Design

S1 Neurosynaptic Core Operation

The neurosynaptic core (Fig. 2D, 2G, 2J) is the atomic, repeated unit of the TrueNorth architecture and chip; it includes 256 neurons (computation), 256 axons (communication), and 256×256 synapses (memory). A core models networks with in-degree and out-degree of 256 or less, including classical neural networks as well as models of canonical cortical microcircuits; larger networks can be composed of multiple cores. Information in the form of spikes flows from inputs, *axons*, to outputs, *neurons*, modulated by connections between them, *synapses*. On a core, each axon i is assigned one of four types $G_i \in \{0, 1, 2, 3\}$ and each neuron j can individually assign a programmable signed integer $S_j^{G_i}$ for axon type G_i. Synaptic weight from axon i to neuron j is a product of two terms: $w_{i,j} \times S_j^{G_i}$, where $w_{i,j} \in \{0, 1\}$. By choosing different combinations of these two terms, synaptic weights can be programmed to be excitatory or inhibitory and can have different relative strengths. We implement the 256×256 w matrix as a crossbar.

A core updates all of its neurons at discrete time steps. Each core maintains a local timer t (4 bits) represented via modulo 16 arithmetic. The time step is nominally 1ms, yielding a 1kHz clock—corresponding to real-time operation[1]. 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 represented by a bit $A_i(t)$. Mathematically, at time t, neuron j receives input: $A_i(t) \times w_{i,j} \times S_j^{G_i}$. Each neuron integrates its synaptic input over time and emits a spike if its threshold is exceeded. Simplified equations for neuron dynamics are described in Table S2, and the full equations, which support a wide range of complex neural dynamics, can be found in (*20*). Spike events are represented as packets that are composed of an assigned route (distances Δx and Δy specifying its path to a destination core), a target axon address, and a delivery time t_{D} computed as current time t plus a programmed axonal delay from 1 to 15 time steps.

The core's physical implementation has five blocks: Neuron, Memory, Scheduler, Router, and Controller (Fig. S1A):

- **Neuron** circuit performs synaptic integration, leak subtraction, and threshold checking for all 256 neurons.
- **Memory** stores the synaptic connections, neuron parameters, neuron state ($\mathbb{V}_{\mathbb{M}}$), and neuron target address ($\mathbb{R}\mathrm{oute}$) in a 256×410 bit SRAM, organized so that each row contains all information for each neuron ($\mathbb{M}_0 \ldots \mathbb{M}_{255}$).
- **Scheduler** queues incoming spikes by writing each one in a 16×256 bit custom SRAM. The Scheduler memory stores $\{A_i(t')\}$, $i \in \{0 \ldots 255\}$, $t' \in \{0 \ldots 15\}$. When a spike arrives for axon i with delivery time t_{D}, the Scheduler sets $A_i(t_{\mathrm{D}}) = 1$. At time step t, the Scheduler delivers the 256 bit vector $A(t)$ to the Controller before clearing it.

[1]Note, a faster synchronization clock is possible, allowing for faster than real-time operation (Fig. S5).

- **Router** passes spike packets between adjacent cores and delivers them to target cores (Fig. S2). Spike packets generated on a core that target the same core use the Router's local channel. The Router's datapath allows any neuron to transmit to any axon up to 255 cores away in both x and y directions (where a single chip is 64×64 cores).

- **Controller** sequences core operation, reading the axon vector $A(t)$ from the Scheduler, reading and writing each neuron's state once per time step, and injecting spike events into the Router.

A simplified timing diagram of core operation is shown (Fig. S1B). Each core has a local timer t (4 bits) that is incremented (modulo 16) for each Global Timer rising edge. At the beginning of a time step (Global Timer rising edge) the first row of the Memory (M_0, representing neuron 0's data) and the row of the Scheduler corresponding to current time step t ($\mathrm{S}_{\mathrm{mod(t,16)}}$, representing axon events) are copied to the Controller. The Controller then compares all axon–synapse pairs in sequence. If axon i is logic 1 and synaptic connection $w_{i,0}$ is logic 1, then the synaptic event E_i is issued to neuron 0, which specifies the value to add[2] to the membrane potential (V_M); the controller skips the synaptic events when either the axon or the synaptic connection is logic 0. After synaptic events are evaluated, a leak is applied (V_L), the membrane potential is compared to a threshold (V_{thr}) and if there is a spike, it is injected into the Router (S_{pk}) with its predefined route (Fig. S2). The updated membrane potential is then written back into the Memory. Neurons 1 to 255 are processed in a similar fashion, where the controller reads each row of memory in succession, and performs the necessary updates. When all neuron updates are processed, the Scheduler clears its current row and the Memory is reset to row M_0. The time between the completion of a core's operation and the start of the next time step constitutes slack.

For a typical network with neurons that fire at an average rate of 20Hz and make an average of 128 connections, a core will receive on average five incoming spike events in a 1ms time step, which corresponds to ~ 640 neural updates. Out of the possible 256×256 ($= 65,536$) neural updates in the time step, our event-driven Controller only performs these ~ 640 updates thereby eliminating 99% of the unnecessary circuit switching in the Neuron.

S2 Design Methodology

The TrueNorth architecture aims to minimize active power using an event-driven design. Specifically, we implemented TrueNorth as a hybrid asynchronous-synchronous chip: the asynchronous control and communication circuits do not require a resource-intensive fast global clock, and the synchronous compute circuits operate only when necessary as they are "clocked" by the asynchronous control.

This hybrid design style is not supported by any single existing tool flow and required the creation of a new one. Therefore, we designed and validated the chip using a combination

[2]For later reference in Section S8, note that this addition constitutes a "Synaptic Operation".

of commercial, custom in-house, and academic tools. The Router, Scheduler, and Controller blocks followed a quasi-delay insensitive asynchronous (clockless) design flow: Starting from a high-level functional description, each block was fleshed out into low-level signal handshakes, and then transformed into transistor networks using a full custom design (*31*). The Neuron, on the other hand, was designed following a traditional synchronous (clocked) approach: The high-level function was described in Verilog and synthesized into standard cells using Cadence Encounter place-and-route software. Choosing an automated flow allowed us to quickly compile complex neuron equations into silicon, and explore the tradeoffs between neuron complexity and efficiency. Despite using a synchronous flow, the neuron circuit operates in an event-driven manner. That is, the neuron's clock is generated by the asynchronous Controller block, which issues on-the-fly clock pulses only when necessary for the neuron to execute synaptic integration, leak subtraction, and threshold checking.

S3 Amortization of Neuronal Costs by Digital Multiplexing

We implemented TrueNorth's neurons using an all-digital approach, breaking path with previous mixed analog–digital approaches. Our digital neuron uses time-division multiplexing, enabling us to amortize its area and leakage costs. The neuron is an extension of the integrate-and-fire neuron with additional features (model described in (*20*), Table S2) and uses $1,272$ logic gates (924 gates for the neuron and 348 gates for the random number generator) in a 28nm process, corresponding to an area of $2900\mu\text{m}^2$; storing neuron state (20 bits) requires an additional area of $3.0\mu\text{m}^2$ per neuron. By multiplexing the neuron 256 times in a time step, the effective area per neuron is: $2900\mu\text{m}^2/256 + 3.0\mu\text{m}^2 = 14.3\mu\text{m}^2$. For reference, a state-of-the-art (non-multiplexed) mixed analog–digital neuron uses on the order of $400\mu\text{m}^2$ in 130nm technology (*6*).

S4 Efficient Clustered and Long-Range Connections

We implement connections in TrueNorth in two stages, where long-range connections use an on-chip mesh routing network and local connections use a fanout crossbar (Fig. 2H). In this section, we contrast our crossbar approach with an alternative approach without a crossbar where a neuron addresses each synapse that it targets independently. Here, we consider a 256×256 crossbar where each axon has 128 active synapses.

In terms of communication bandwidth, with a crossbar, a single packet (a long-range connection) targets 128 synapses at its destination core. Without the crossbar, a single packet targets a single synapse, requiring each neuron to transmit 128 individual packets each time it spikes—a factor of 128 greater.

In terms of address memory[3], with a crossbar, each neuron requires 20 bits ($= \log_2(10^6)$)

[3]For both approaches, we consider only address memory and neglect additional memory to store synaptic weights.

to address its target axon[4]. Without the crossbar, each neuron targets 128 synapses requiring $3,456$ bits ($= 128 \times \log_2(128 \times 10^6)$)—a factor of 17.3 greater.

The tradeoff for this efficiency is that the network topologies supported are constrained; for example, a single neuron cannot target more than 256 synapses directly, but can do so by using two or more neurons in another core as "splitters" sending spikes to more targets.

S5 Energy Measurement for On- and Off-Chip Communication

We measured the active energy required to move a bit in TrueNorth within a core, between cores on a chip, and between chips. Within a core, the average energy required to move a bit from local memory (SRAM) to the Controller was $\sim$4.7fJ (at 1kHz and 0.775V, reading all $\sim$428 million bits was $\sim$20mW). The energy to transport a bit between two adjacent cores (along x dimension) is 72fJ at 0.775V (Fig. S4). The energy to transport a bit from the internal to the external periphery is 2pJ at 0.775V (Fig. S4). The energy to send a bit between external peripheries of two adjacent chips is 26pJ at 1.8V. These measurements confirm that communication is most expensive between chips, and supports our design choice to co-locate memory and computation within a core—as opposed to an alternative architecture that separates memory and computation on different chips.

Supplementary Text: Measurement & Characterization

S6 Functional Verification

To verify TrueNorth's correctness, we compared the chip spike for spike with the software simulator (section S9) using a large test suite of networks and unit tests. To enable these tests, we built a printed circuit board that includes programmable logic, support components, an input-output port, and a socket for TrueNorth.

Tested chips have a yield distribution due to physical defects, common to integrated circuit manufacturing, including fully-functional chips (more than 50%), chips with occasional faulty cores, and a minority of unusable chips. Chips with faulty cores can be partially recovered by disabling those cores. To test chip tiling, we verified the function of the off-chip links by interconnecting north and south, and east and west ports on a single chip to one another, and measured the total bandwidth (transmit and receive) for all ports at more than 160 million spikes per second (5.44 Gbits/sec), enabling high-bandwidth sensor inputs, actuator outputs, and inter-chip communication.

[4]Note that each TrueNorth neuron can target any axon on any core up to 255 away (up to 4 chips) in both x and y dimensions, which includes 17 billion synapses, and therefore, uses 26 address bits.

S7 Benchmark Networks for Power and Speed Characterization

We characterized TrueNorth's power consumption by running a set of probabilistically generated complex recurrent networks (Fig. S3 and Fig. 4).

The networks use all $4,096$ cores and one million neurons on the chip, and span a range of mean firing rates per neuron from 0 to 200Hz and a range of active synapses per neuron from 0 to 256. The networks were designed to push TrueNorth's power consumption as high as possible for a given spike rate and synaptic density. To this end, we selected connections between cores randomly with uniform distribution across the chip, requiring spikes to travel long distances (21.3 cores away on average in both x and y dimensions), and configured all neurons in the stochastic mode, requiring extensive use of the pseudo-random number generator. Each axonal delay was uniformly chosen from 1 to 15 time steps. All neurons on the chip are configured with identical parameters; they spike spontaneously due to a stochastic positive leak, and receive a balance of stochastic synaptic inhibition and excitation. Given the synaptic density, we chose the leak and threshold appropriately to reach each target firing rate. These networks are a good choice for benchmarking because they have a rich connectivity (neurons project to any core with equal likelihood) and have complex spiking dynamics (due to recurrence and neuron stochasticity).

S8 Computational Efficiency

For our architecture, we define computational efficiency in terms of synaptic operations per second (SOPS) per Watt. One synaptic operation corresponds to a conditional multiply-accumulate that forms the core inner-loop of the neuron operation (Table S2). Specifically, a *synaptic operation* constitutes adding a 9 bit signed integer $S_j^{G_i}$ to the membrane potential, which is a 20 bit signed integer, when axon $A_i(t) = 1$ and synaptic connection $w_{i,j} = 1$. We characterized TrueNorth's performance over a wide range of firing rates and number of active synapses (Fig. S5). For a typical network with an average firing rate of 20Hz and and an average of 128 active synapses per neuron, TrueNorth delivers 46 billion SOPS per Watt at real-time (with 1ms time steps) and 70 billion SOPS per Watt at $5\times$ faster than real-time (with 200μs time steps). For networks with high spike rates and high number of active synapses, TrueNorth delivers 400 billion SOPS per Watt. While not a direct comparison, according to the Green 500 list (*32*), the current most energy-efficient high-performance computing (HPC) system is at GSIC Center, Tokyo Institute of Technology, and achieves 4.5 billion FLOPS (floating point operations per second) per Watt on the Linpack benchmark. Note that the HPC measurement is total system power, while we only measured the power of the TrueNorth processor in the system.

Supplementary Text: Applications

S9 Software Ecosystem

TrueNorth's software ecosystem includes Compass (*25*), a highly-optimized simulator designed to simulate large networks of neurosynaptic cores, and a compositional programming language (*33*) for developing TrueNorth networks. In particular, we used this ecosystem to verify the function of TrueNorth hardware, as well as develop applications such as the multi-object detection and classification network from section S11.

The Compass simulator is written in C++, and uses a combination of MPI library calls and OpenMP threading primitives to partition a network across several processes. Specifically, cores that reside in the same shared memory space within a process are distributed among multiple threads, and each thread independently simulates its cores' synaptic crossbars and neurons. Compass has been scaled to run networks with 530 billion neurons and 137 trillion synapses (*34*).

By design, Compass and TrueNorth implement the same functional specification, so networks that run on TrueNorth match spike-for-spike with Compass. This one-to-one correspondence between hardware and software was an essential tool for logic verification during TrueNorth's design and test phases. In addition to hardware verification, Compass was indispensable for (i) providing a flexible software environment for developing neural network applications, (ii) simulating TrueNorth networks when hardware was not yet available (prefabrication and testing periods), (iii) exploring multichip network topologies, and (iv) analyzing core placement for power optimization.

To develop networks that natively run on TrueNorth, we used our Corelet Programming Environment—a compositional language for specifying networks of neurosynaptic cores. Specifically, a *corelet* is a collection of one or more neurosynaptic cores that encapsulates all the neuron parameters as well as intra- and inter-core connectivity. Corelets can be composed hierarchically to implement particular functions, such as linear and non-linear signal processing; spatio-temporal filtering; saliency; object detection, classification, and recognition; and real-time audio and video analysis (*23*). In our current work, we were able to create the multi-object detection and classification network (section S11) via a collaborative software design effort using corelets. In particular, the corelet abstraction allowed for the easy sharing of code, a modular design that promoted code reuse, and a quick-turnaround for generating networks that ran on both Compass and TrueNorth.

S10 Construction of a Visual Filter

To demonstrate TrueNorth's capability as a real-time processor, we explore two networks that extract visual features (orientation filters) from streaming videos. In the first network, neurons obtain their selectivity through feed-forward connections drawn from visual space (Fig. S6A). In the second network, neurons use both feed-forward and lateral connections to obtain their selectivity—inspired by a model of mammalian visual cortex (Fig. S6B). We demonstrate that

adding these lateral connections, which are natively supported in TrueNorth, increases feature selectivity dramatically with only a slight increase in power consumption.

For the first network (feed-forward), we generate a 64×64 orientation map adapted from a previous approach (*35*). Each pixel in our feature map, which corresponds to a neurosynaptic core, consists of 256 orientation selective neurons (with similar orientation references and random phase) that draw their input from a 4×4 region in visual space. Specifically, neurons obtain their receptive fields through an offline learning procedure: A neuron on a particular core selects its inputs from a two-dimensional sinusoidal pattern randomly choosing 5 ON pixels (greater than the mean intensity) as excitatory inputs, and 5 OFF pixels (less than the mean intensity) as inhibitory inputs, creating a push-pull receptive field (*24, 36*). Cores that are nearby in space are chosen to have similar orientation preferences, thereby creating a smoothly changing orientation map.

To measure the network's selectivity, we generate 128×128 pixel movies of drifting sinusoidal patterns using 64 orientations from 0 to 2π from a simulated spiking retina with temporal resolution of 1ms (Fig. S6C-E). We drive the network with these patterns while recording the neural responses for each pattern. The selectivity maps are created by filtering each neuron's response across orientations (circular convolution) and assigning its pixel in the map to its peak or preferred orientation.

For the second network (feed-forward plus lateral), we include lateral connections based on observed anatomy from the visual cortex, where neurons with similar orientation preferences have lateral excitatory connections, and neurons with different preferences have lateral inhibitory connections. Specifically, each neuron on a core was assigned as recurrent inhibitory (144 neurons), recurrent excitatory (44 neurons), or as excitatory output (64 neurons). We randomly connected neurons to neighboring cores with probability that decreased with distance and increased based on orientation similarity with probability 0.5 and strength -2 for inhibitory neurons, and probability 0.5 and strength $+1$ for excitatory neurons. These inhibitory and excitatory connections result in a network where similar orientations reinforce each other while dissimilar ones compete (Fig. S6B).

We compare the sharpness of orientation selectivity between the two networks using normalized vector magnitude (NVM). Specifically, NVM is the vector sum of a neuron's spike count across each input orientation, divided by the total number of spikes across all orientations. The first network has a mean NVM of 0.14, whereas the second network has a mean NVM of 0.26 an increase of 86% (see Fig. S6F for the distribution of all cells), while only increasing power consumption by 1% (from 59mW to 60mW). While biophysical models are not the intent of this work, this example demonstrates that TrueNorth is a versatile platform—capable of supporting a large class of receptive–projective fields and neural spiking behaviors that, at a functional level, can reproduce many observed biological phenomena.

S11 Multi-object Detection and Classification

To demonstrate TrueNorth's suitability for real-world problems, we developed a network for the real-time detection and classification of people, bicyclists, cars, trucks, buses, capable of identifying multiple objects per frame (Fig. 3). This system includes a *What* Network to provide object class and a *Where* Network for object detection, loosely following the dorsal–ventral separation of duties found in the brain (*4*), and a network for merging the two to provide localized object class predictions (Fig. S7). The system was built for a fixed position camera operating at 30 frames per second.

The *Where* Network predicts which input locations contain objects of interest. Input is sent to the saliency system by converting the image from the YCrCb colorspace to spikes using a 16 level quantization scheme. The saliency network compares 4 color pixels from each 16×16 pixel patch of the image to 8 representative background templates, and declares the patch salient if the difference from all background templates exceeds a threshold. We learned background templates offline using k-means clustering and used an error minimization approach to select thresholds for each patch.

The *What* Network uses a multilayer processing system to provide object classifications at a resolution corresponding to classifying 64×64 pixel patches evenly tiled across the image. Input to the classification system (Fig. S8) is provided by applying an edge operator to the original image and coding the resulting edges using 16 levels of quantization. For the first processing layer, local features are learned from 64 pixel blocks using k-means clustering. Local features from four such patches within a 64×64 pixel region are combined to create the first level features. A weight matrix from features to classes is then created using a support vector machine, and features are further merged based on covariance between these weights to create a second pooling layer of features. A total of $1,024$ features from the 64×64 patch, and 256 features from the neighboring 8 patches are fed through a discrete weight support vector machine for classification. Finally, the results are smoothed with a spatio-temporal filter.

Also as a part of the *What* Network, two additional classification systems operate in parallel with the main classification network. To assist in classifying objects larger than a single 64×64 pixel patch, a large object detector provides size-based classification. This system sums the total number of salient pixels in various sized, class-specific regions, where region size is set according to the average height and width of classes that are larger than 64×64 pixels. If the number of salient pixels in such a region exceeds a threshold, the large object detector signals the possible presence of the corresponding object. To assist in classifying moving objects, we estimated object speed by sending the saliency map through a spatio-temporal filter. The resultant predictions of object size and motion, as well as input from the multilayer processing system are sent through a merger block to provide the final prediction.

The classification and detection networks are combined by a saliency-classification merger block. For each active location in the saliency map, the saliency-classification merger block queries the classifier network to provide a class prediction at the corresponding location. This block thus produces as output 5 object maps at the resolution of the saliency system.

The 5 object maps are then fed into the object localization block, which provides predictions of object centers. This block uses a recurrent network to apply a series of peel operators to the class predictions, followed by a minimum-distance operation that enforces separation between object predictions. Bounding boxes are then placed around each object for final evaluation.

We applied the above system to DARPA's Neovision2 Tower dataset (*37*). The presence of objects in each frame of this dataset is relatively sparse, such that square bounding boxes tightly placed around objects covered on average 2.7% of each image. This sparsity imposes the need to reject a large number of non-object regions, thereby increasing the difficulty of this dataset considerably over vision problems that only consider classification. We used our one-to-one simulator (*25*) to run our processing system scaled up for the $1,088 \times 1,920$ pixel images in the dataset. The system was trained on $11,250$ frames and tested on an out-of-sample set containing the same number of frames and $161,278$ objects. Under these conditions, the system achieved a precision of 0.85 and recall of 0.80 on the test set. The activity in each component of the system, along with final output, can be seen in Movie S1.

We ran the multi-object detection and classification network on TrueNorth with a 400×240 pixel video stream (Fig. 3) in real-time—30 frames per second, where one frame used 33 time steps, 1ms each. The network used $3,406$ of $4,096$ cores, and we measured its total power consumption at 63mW during operation.

Supplementary Text: Comparison to Other Architectures

S12 Comparison to CPU-based platforms

We compare TrueNorth's performance to two CPU-based platforms for simulating large-scale spiking neural networks, Compass and SpiNNaker.

The first platform we consider is Compass, an optimized simulator designed specifically for TrueNorth (section S9). For comparison, we ran three identical networks on both TrueNorth and Compass: a probabilistically-connected network (section S7), a recurrent visual feature extraction network (section S10), and an object detection and classification network (section S11). Compass was run on an Intel Core i7 CPU 950 with 4 cores and 8 threads, clocked at 3.07GHz (45nm process circa 2009). Power for Compass was measured as the *increase* in current consumption (SOAR Digital Clamp Tester 2210 ammeter) over the baseline of 0.9A at 120V (108W). We neglect the CPU's baseline–underestimating its energy per simulation—as we cannot directly measure power consumption of individual system components.

These networks were simulated for $10,000$ time steps (corresponding to 10 seconds at real time), and we report the simulation time and power to compute the energy for each platform (Table S1). For real-time operation (1ms time steps), TrueNorth consumes $100,000$ to $300,000\times$ less energy per synaptic event. Note, the simulator runs 100 to $200\times$ slower than real-time.

The second platform we consider is SpiNNaker (*38*), a microprocessor-based neuromorphic system optimized for real-time spiking neural networks. SpiNNaker has demonstrated a 48-chip board, where each chip has 18-microprocessor cores in 102mm^2 (ignoring DRAM),

fabricated in a 130nm process. To date, SpiNNaker has been benchmarked using two networks, which are comparable to the randomly-connected networks that we use for benchmarking. One SpiNNaker network consisted of 250 thousand neurons (22Hz average) and 82 million synapses with all connections made locally (to and from the same cluster of neurons on the same core); the other consisted of 115 thousand neurons (15Hz average) and 86 million synapses with inter-cluster connections chosen randomly (*13*). SpiNNaker consumed 36W for the first network, and 26W for the second one, corresponding to 20nJ per synaptic event in each case. In comparison, TrueNorth runs one million neurons (20Hz average) and 128 million active synapses at 26pJ per synaptic event—$769\times$ less energy and $11.4\times$ less silicon area (without including area required for SpiNNaker's DRAM).

S13 Contrast with Mixed Analog–Digital Neuromorphic Systems

We contrast TrueNorth with two recent mixed analog–digital neuromorphic approaches, BrainScaleS (*7*) and Neurogrid (*8*), which both aim to emulate continuous-time brain dynamics and serve as a tool for computational neuroscience.

The goal of BrainScaleS is to accelerate the search for interesting networks by running faster than biological real-time (striving towards $1,000$ to $10,000$ acceleration factors). The BrainScaleS group has demonstrated a wafer-scale system (20cm diameter wafer in 180nm process) with 200 thousand mixed analog–digital neurons and 40 million addressable synapses (*7*), projected to consume roughly 1kW. BrainScaleS was designed for a fundamentally different purpose than TrueNorth, namely, exploring networks faster than would otherwise be possible.

The goal of Neurogrid is to provide a tool for computational neuroscientists to explore (non-learning) spiking-based models of cortical networks in real-time. Neurogrid is a sixteen-chip system connected in a tree topology (*39*), and emulates one million neurons (two-compartment model with ion-channels) and four million multiplexed synapses, with fanout via a spatial kernel. Specifically, the kernel is implemented by an analog *diffusor* (*40*) circuit that receives an input spike and (nonspecifically) activates all neurons in a neighborhood with synaptic strength decaying approximately exponentially with distance. Specific connections are implemented with an FPGA–SRAM daughterboard that connects to the root of the tree, and supports about 10 programmable connections per neuron (which can fanout further by hitting topographic locations across different chips). Total system power during typical operation, including the daughterboard, is approximately 3.1W. Neurogrid was designed for a fundamentally different purpose than TrueNorth, namely, exploring dynamics resulting from the interactions among biophysical components, ranging from ion channels to network connections (for example, exploring synchrony).

Full Reference List

1. J. von Neumann, *The computer and the brain* (Yale University Press, New Haven, CT, 1958).
2. C. Mead, *Analog VLSI and neural systems* (Addison-Wesley, Boston, MA, 1989).
3. C. Eliasmith, *et al.*, *Science* **338**, 1202 (2012).
4. M. Mishkin, L. G. Ungerleider, K. A. Macko, *Trends in neurosciences* **6**, 414 (1983).
5. P. S. Churchland, T. J. Sejnowski, *The computational brain.* (The MIT press, 1992).
6. T. Yu, J. Park, S. Joshi, C. Maier, G. Cauwenberghs, *Biomedical Circuits and Systems Conference (BioCAS), 2012 IEEE* (IEEE, 2012), pp. 21–24.
7. J. Schemmel, *et al.*, *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on* (IEEE, 2012), pp. 702–702.
8. B. V. Benjamin, *et al.*, *Proceedings of the IEEE* **102**, 699 (2014).
9. M. Mahowald, R. Douglas, *Nature* **354**, 515 (1991).
10. G. Indiveri, *et al.*, *Frontiers in Neuroscience* **5** (2011).
11. R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, H. S. Seung, *Nature* **405**, 947 (2000).
12. S.-C. Liu, T. Delbruck, *Current Opinion in Neurobiology* **20**, 288 (2010).
13. E. Stromatias, F. Galluppi, C. Patterson, S. Furber, *International Joint Conference on Neural Networks (IJCNN). IEEE* (IEEE, 2013), pp. 1–8.
14. M. Mahowald, *An analog VLSI system for stereoscopic vision* (Springer, 1994).
15. J. Backus, *Communications of the ACM* **21**, 613 (1978).
16. J. Teifel, R. Manohar, *Proc. of 10th International Symposium on Asynchronous Circuits and Systems* (2004), pp. 17–27.
17. R. J. Douglas, K. A. Martin, *Annu. Rev. Neurosci.* **27**, 419 (2004).
18. S. B. Laughlin, T. J. Sejnowski, *Science* **301**, 1870 (2003).
19. V. B. Mountcastle, *Journal of Neurophysiology* **20**, 408 (1957).
20. A. S. Cassidy, *et al.*, *International Joint Conference on Neural Networks (IJCNN). IEEE* (2013).

21. W. J. Dally, B. Towles, *Design Automation Conference, 2001. Proceedings* (IEEE, 2001), pp. 684–689.

22. P. Merolla, *et al.*, *Custom Integrated Circuits Conference (CICC), 2011 IEEE* (IEEE, 2011), pp. 1–4.

23. S. K. Esser, *et al.*, *International Joint Conference on Neural Networks (IJCNN). IEEE* (2013).

24. D. H. Hubel, T. N. Wiesel, *The Journal of Physiology* **160**, 106 (1962).

25. R. Preissl, *et al.*, *High Performance Computing, Networking, Storage and Analysis (SC), 2012 International Conference for* (2012), pp. 1–11.

26. J. Seo, *et al.*, *Custom Integrated Circuits Conference (CICC), 2011 IEEE* (IEEE, 2011), pp. 1–4.

27. T. Ohno, *et al.*, *Nature Materials* **10**, 591 (2011).

28. M. M. Shulaker, *et al.*, *Nature* **501**, 526 (2013).

29. C. Isci, Workload adaptive power management with live phase monitoring and prediction, Ph.D. thesis, Princeton University (2007).

30. D. S. Modha, R. Singh, *Proceedings of the National Academy of Sciences* **107**, 13485 (2010).

31. C. G. Wong, A. J. Martin, *Proc. Design Automation Conference* (2003), pp. 508–513.

32. Green500, http://www.green500.org/lists/green201311 (2013).

33. A. Amir, *et al.*, *International Joint Conference on Neural Networks (IJCNN). IEEE* (2013).

34. T. M. Wong, *et al.*, *IBM Research Divsion, Research Report RJ10502* (2012).

35. S.-B. Paik, D. L. Ringach, *Nature Neuroscience* **14**, 919 (2011).

36. L. A. Palmer, T. L. Davis, *Journal of Neurophysiology* **46**, 260 (1981).

37. Neovision2 dataset, http://ilab.usc.edu/neo2/dataset/ (2013).

38. E. Painkras, *et al.*, *Solid-State Circuits, IEEE Journal of* **48**, 1943 (2013).

39. P. Merolla, J. Arthur, R. Alvarez, J.-M. Bussat, K. Boahen, *Circuits and Systems I: Regular Papers, IEEE Transactions on* **61**, 820 (2014).

40. A. G. Andreou, K. A. Boahen, *Analog Integrated Circuits and Signal Processing* **9**, 141 (1996).

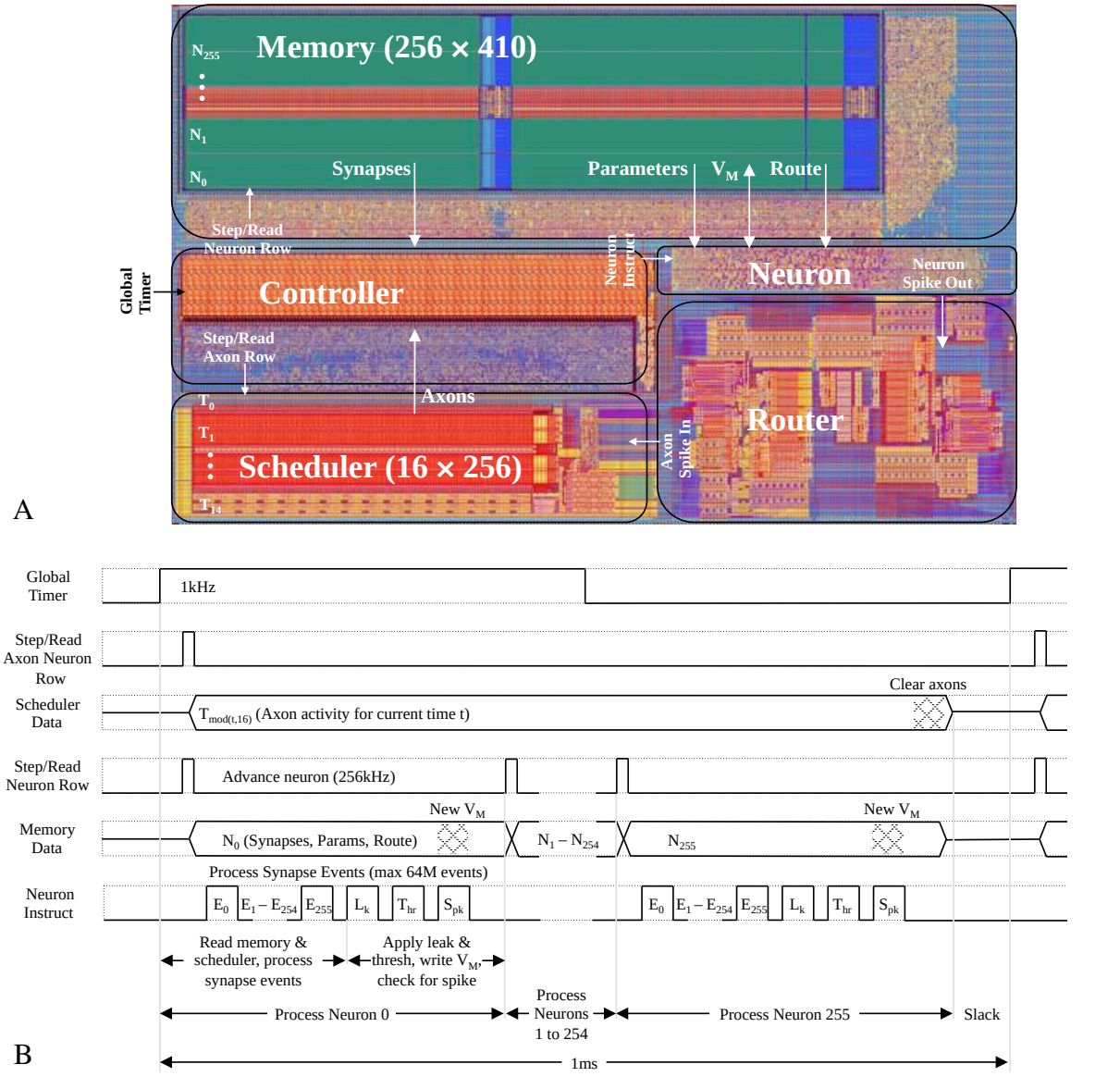


Fig. S1: Layout and operation of the neurosynaptic core. The core implements 256 neurons, 256 axons, and a 256 × 256 array of synapses. **(A)** Layout of the five main blocks of the core (rotated from Fig. 2J for readability): Neuron, Controller, Scheduler, Router, Memory. These blocks realize computation (Neuron & Controller), communication (Router), axonal delays (Scheduler), as well as synapse and neuron states and parameters (Memory). **(B)** Core timing diagram. Global Timer's rising edge initiates a time step, orchestrated by Controller. First, it reads active axon inputs from Scheduler (T_0 at time step 0). Next, the Neuron sequentially computes all 256 neuron updates by reading a row from Memory (starting with N_0, representing neuron 0's data), updating state by processing synaptic events delivered by Controller, and writing back updated state to Memory.

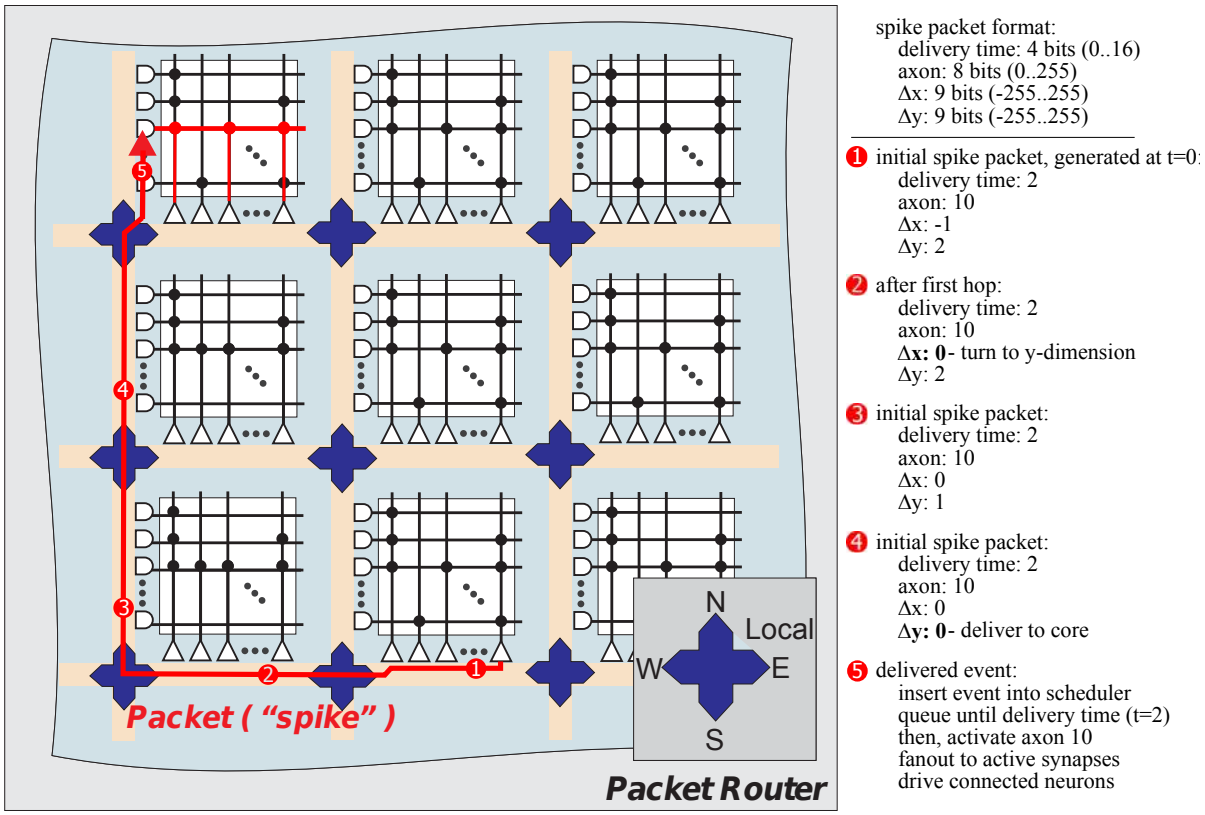
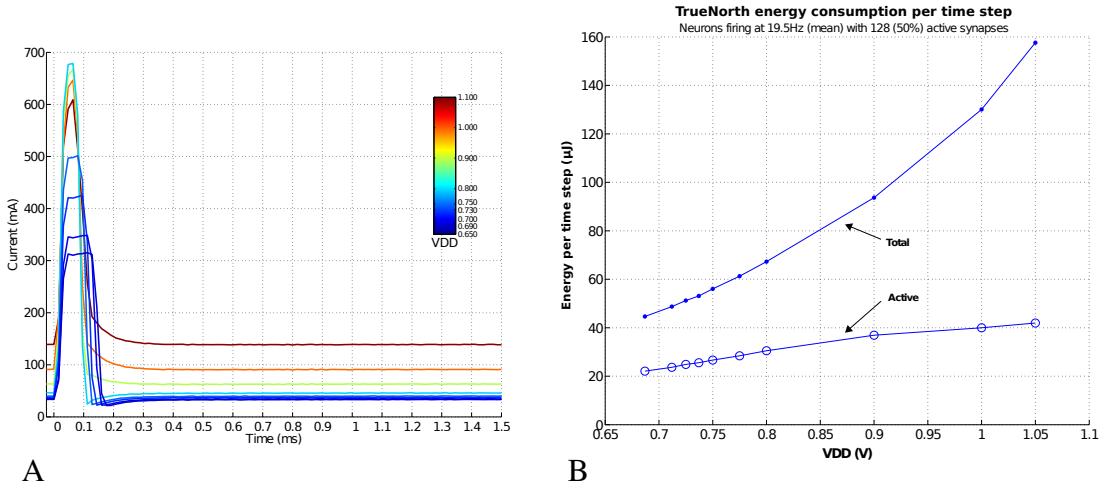


Fig. S2: Neurons communicate to axons using spike packets. A packet comprises delivery time, axon, Δx, Δy, where delivery time corresponds to the 4 least-significant bits of the delivery time step; axon is the address of the axonal target; Δx and Δy are the distance from the originating core to the destination core in the x and y dimensions, respectively. Here, we follow a packet from source to destination: (1) A neuron exceeds its threshold and spikes, injecting a spike packet into the network via its local router. (2) The packet moves first along the x dimension (left if $\Delta x < 0$, right if $\Delta x > 0$), decrementing its absolute Δx value at each hop. When $\Delta x = 0$, the packet turns along the y dimension (up if $\Delta y > 0$, down if $\Delta y < 0$). (3) The packet continues, traveling up in the y dimension, decrementing Δy at each hop (while Δx remains fixed at 0). (4) When $\Delta x = 0$ and $\Delta y = 0$, the packet is delivered to the destination core via the router's local channel. (5) The packet activates the scheduler row and column corresponding to delivery time $= 2$ and axon $= 10$. The scheduler queues the axon event until $t = 2$, then delivering it to axon 10, which in turn drives post-synaptic neurons via the crossbar.



A B

Fig. S3: TrueNorth energy characterization. **(A)** Current measurements for a single time step run at various voltages. All one million neurons ($4,096$ cores) are active and spike at a mean rate of 9.4Hz and have an average number of 128 synapses active. At higher voltages, the chip consumes more current (active and leakage) and completes all pending synaptic operations sooner. At lower voltages, current consumption is reduced (active and leakage) and the chip requires more time to complete the computation. **(B)** We compute total energy per time step by measuring the area under each curve (integrated current per time step) multiplied by the voltage, shown for neurons spiking at a mean rate of 19.5Hz and 128 active synapses per neuron. As voltage decreases, both active and total (active plus passive) energy per time step decrease.

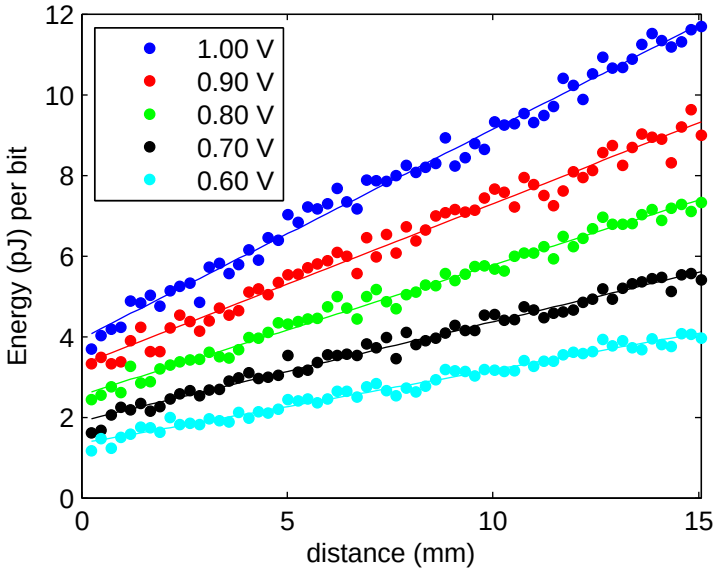


Fig. S4: Energy vs. distance. To evaluate the energy efficiency of TrueNorth's on-chip communication network, we injected spike packets (32 bits wide) traveling different distances, and measured the resulting power. We show the energy per bit at different voltages for packets traveling in x direction (y direction is similar). Note that the y-intercept represents the energy per bit of the peripheral circuitry and line drivers. For a power supply of 0.775V, we calculate that communicating a bit consumes 0.3fJ per μm with an intercept of 2pJ/bit. The distance between cores in x direction is 240μm, which corresponds to one hop in the router. Therefore, the energy to send a spike one hop between cores is 2.3pJ.

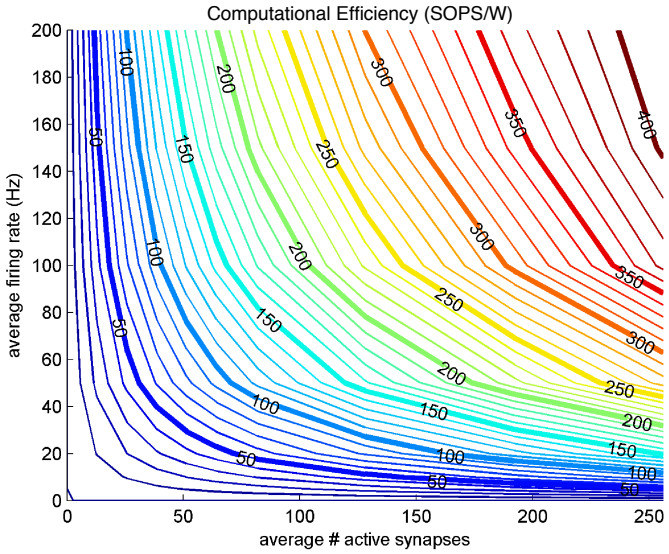


Fig. S5: Computational Efficiency. We measure TrueNorth's computational efficiency as the number of Synaptic Operations Per Second (SOPS) per Watt. To amortize passive power, we increased the global synchronization clock frequency. We measured fastest speed at which one-to-one correspondence between software and hardware held. Increasing clock speed beyond this point results in divergence between software and hardware due to spikes not arriving at their destinations by delivery time or cores not completing their computation by the end of the time step. For example, we ran networks with an average spike rate of 20Hz and 128 active synapses per neuron with a time step equal to $200\mu s$, $5\times$ faster than real-time and still observed one-to-one correspondence. For this network, we measured 70 SOPS/W. For networks with increased rates and number of synapses, we measured over 400 SOPS/W.

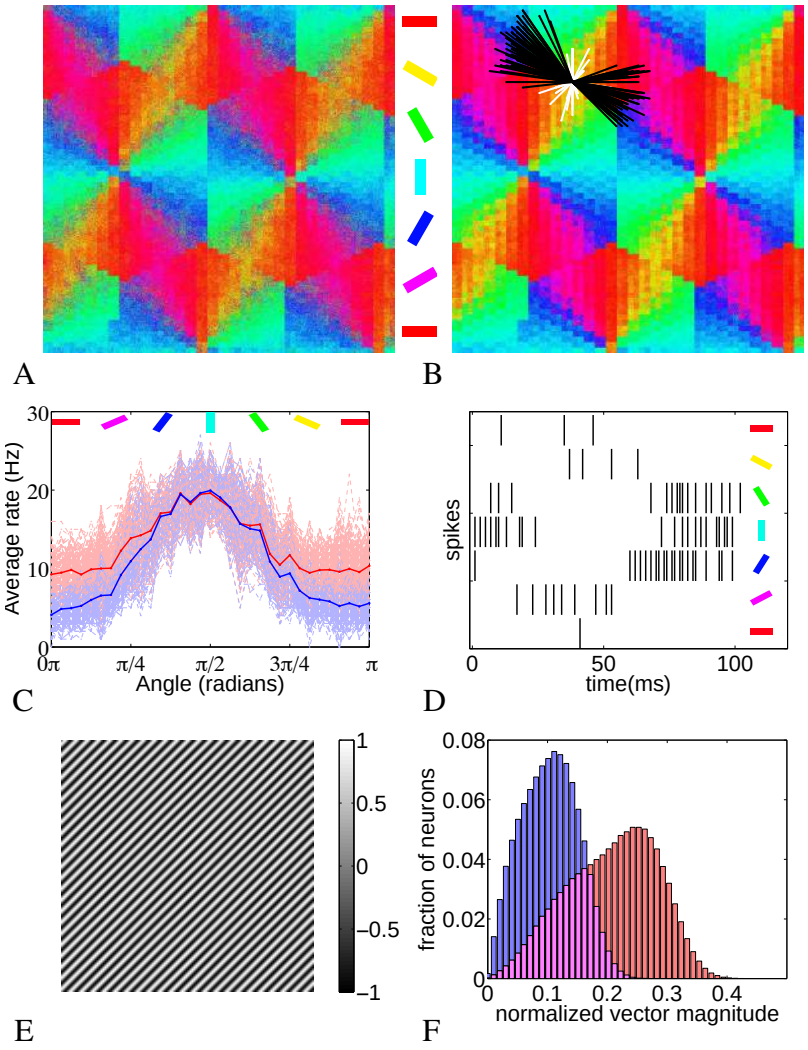


Fig. S6: Feature extraction on TrueNorth. (**A**) Feedforward orientation preference map of neuron selectivity, across TrueNorth's one million neurons, organized as a $1,024 \times 1,024$ grid. (**B**) Orientation map with added lateral connections, where similar orientations excite each other (white) and dissimilar ones inhibit (black), sharpens selectivity on average by 86% while increasing power by 1%. (**C**) Averaged tuning curves for neurons selective for vertical orientations ($\pi/2$) with lateral connections (blue) and without (red). Adding lateral connections increases selectivity by reinforcing the preferred orientation while suppressing non-preferred ones. (**D**), Example spike responses from a vertically-selective neuron for different orientation patterns. (**E**), Example frame from drifting grating movie with $\theta = \frac{\pi}{4}$. (**F**), Normalized vector magnitude increases with lateral connections (red) contrasted with the same network without lateral connections (blue), showing that such connections increase neurons' selectivity for their preferred orientations.

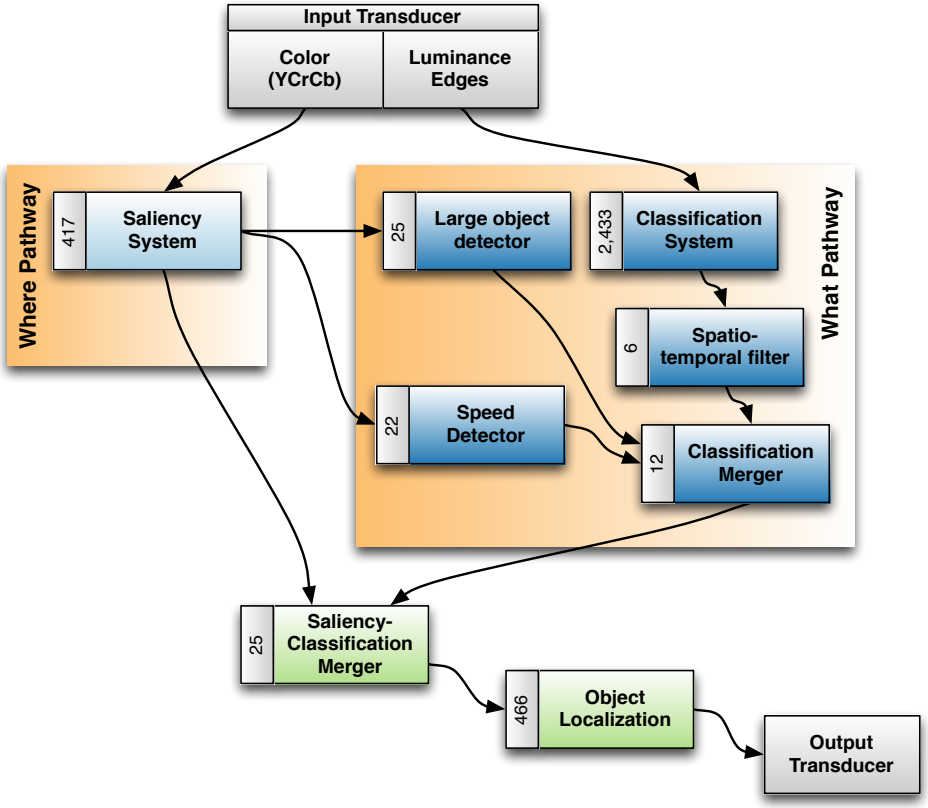


Fig. S7: Object detection and classification system on TrueNorth. Images are converted into spikes via the input transducer, and these spikes propagate through the various components. Gray tags on the left side of each component indicate the number of cores used in each module. See Fig. S8 for an expanded view of the network used for the Classification System.

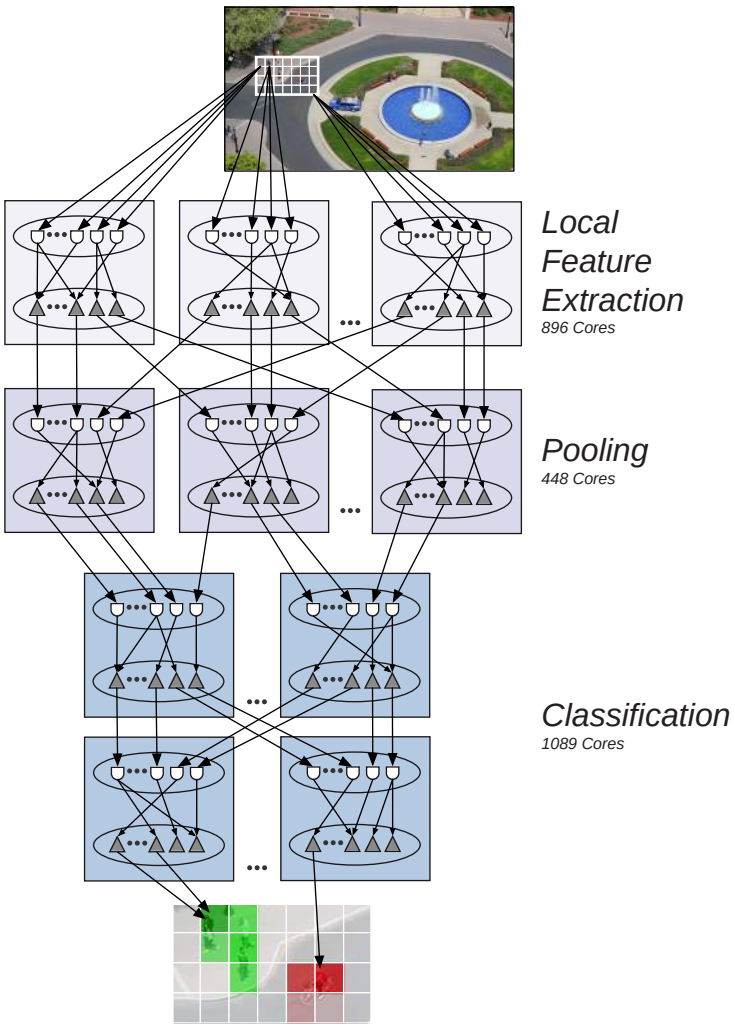


Fig. S8: Classification System network. The Classification System is the largest block in the object detection and classification application (Fig. S7). It consists of three layers of integrate-and-fire neurons with their parameters and connections configured to perform local feature extraction, pooling, and classification.

		Probabilistic Network	Visual Filter	Object Detection & Recognition
TrueNorth	Time(s)	10	10	10
	Power(W)	0.072	0.060	0.063
	Energy(J)	0.720	0.600	0.630
	Energy Ratio (Energy / TrueNorth Energy)	1.000	1.000	1.000
CPU system	Time(s)	1,290	2,012	1,025
	ΔPower(W)	98.3	90.7	80.1
	Energy(J)	127,000	182,000	82,000
	Energy×TN (Energy / TrueNorth Energy)	176,000	304,000	130,000

Table S1: TrueNorth CPU time, power, and energy simulating spiking neural networks.

SYNAPSE & LEAK INTEGRATION

$$V_j(t) = V_j(t-1) - \lambda_j + \sum_{i=0}^{255} \mathcal{A}_i(t) \times w_{i,j} \times S_j^{G_i}$$

THRESHOLD, FIRE, RESET

if $V_j(t) \geq \alpha_j$

Spike

$V_j(t) = \mathcal{R}_j$

endif

Table S2: Simplified Neuron Update Equations. TrueNorth's neuron is an extension of the integrate-and-fire neuron with several operational modes and integrated pseudorandomness (See (*20*) for a complete description of the model). In summary, in a time step, the neuron j updates its membrane potential, $V_j(t)$, from its previous value, $V_j(t-1)$, by subtracting its leak λ_j and adding synaptic input $\mathcal{A}_i(t) \times w_{i,j} \times S_j^{G_i}$, $i \in \{0 \ldots 255\}$, (axonal input gated by synapse state multiplied by weight). Note, when either $\mathcal{A}_i(t)$ or $w_{i,j}$ is zero, no synaptic input is added. If the membrane potential, $V_j(t)$, equals or exceeds the threshold, α_j, the neuron emits a spike and resets to $\mathcal{R}_j$.

Movie S1: Movie of object detection and classification system on TrueNorth. *Upper left* Saliency detects when clusters of pixels vary significantly from a learned background. *Upper right* Saliency + Classification categorizes detected objects as one of five classes: car, bus, truck, person, cyclist. *Lower left* Object Centers calculates the center point of each object. *Lower right* Output places class specific boxes at the center of each object with the class label: blue for car, cyan for truck, magenta for bus, green for person, red for cyclist.