COMPUTER SCIENCE

Neural inference at the frontier of energy, space, and time

Dharmendra S. Modha*, Filipp Akopyan†, Alexander Andreopoulos†, Rathinakumar Appuswamy†, John V. Arthur†, Andrew S. Cassidy†, Pallab Datta†, Michael V. DeBole†, Steven K. Esser†, Carlos Ortega Otero†, Jun Sawada†, Brian Taba†, Arnon Amir, Deepika Bablani, Peter J. Carlson, Myron D. Flickner, Rajamohan Gandhasri, Guillaume J. Garreau, Megumi Ito, Jennifer L. Klamo, Jeffrey A. Kusnitz, Nathaniel J. McClatchey, Jeffrey L. McKinstry, Yutaka Nakamura, Tapan K. Nayak, William P. Risk, Kai Schleupen, Ben Shaw, Jay Sivagnaname, Daniel F. Smith, Ignacio Terrizzano, Takanori Ueda

Computing, since its inception, has been processor-centric, with memory separated from compute. Inspired by the organic brain and optimized for inorganic silicon, NorthPole is a neural inference architecture that blurs this boundary by eliminating off-chip memory, intertwining compute with memory on-chip, and appearing externally as an active memory chip. NorthPole is a low-precision, massively parallel, densely interconnected, energy-efficient, and spatial computing architecture with a co-optimized, high-utilization programming model. On the ResNet50 benchmark image classification network, relative to a graphics processing unit (GPU) that uses a comparable 12-nanometer technology process, NorthPole achieves a 25 times higher energy metric of frames per second (FPS) per watt, a 5 times higher space metric of FPS per transistor, and a 22 times lower time metric of latency. Similar results are reported for the Yolo-v4 detection network. NorthPole outperforms all prevalent architectures, even those that use more-advanced technology processes.

In the design of the first programmable computer, EDVAC (*1*), compute and memory used heterogeneous technologies (vacuum tubes versus ultrasonic serial delays) that had disparate relative speeds and costs, leading to the separation of compute from memory by necessity and, eventually, to a hierarchy of memories. These fundamental constructs still pervade modern computing where energy and bandwidth for off-chip memory dominate (*2*–*4*), creating a “data movement crisis” (*5*) and forming a physical and an “intellectual” programming model (*6*) bottleneck. Three factors necessitate revisiting these constructs for neural inference. First, inspired by the brain (*7*), neural inference (*8*, *9*) has emerged as a powerful application that can be implemented with a far simpler set of constructs. Second, silicon technology is continuing to progress such that, within a decade, 2 billion transistors may fit in 1 mm^2 (*10*). In the meantime, logic and memory have long been implemented homogeneously within the same substrate (*11*, *12*), although at a lower density and higher cost than off-chip memory. Third, it is now critical to prevent artificial intelligence (AI) from becoming “economically, technically, and environmentally unsustainable” through “dramatically more computationally-efficient methods” (*13*).

Axiomatic design

NorthPole, an architecture and a programming model for neural inference, reimagines (Fig. 1) the interaction between compute and memory by embodying 10 interrelated, synergistic axioms that build on brain-inspired computing (*14*, *15*). Within the broad domain of neural inference, NorthPole co-optimizes architecture, algorithms, and software and enables a simple input-output model (*16*).

To set terminology, a neural network is typically a graph of layers, where each layer transforms an input frame of neural activations (for example, an image) into an output frame of neural activations (for example, a class label). Typically, for each neuron, this involves multiplications of its input neuron activations with learned synaptic layer weights followed by linear accumulation, mathematical transforms, and possibly a nonlinear activation function. The synaptic multiplications are inherently parallel for every neuron, and all neurons within a layer operate in parallel.

Axiom 1: Turning to architecture, NorthPole is specialized for neural inference. For example, it has no data-dependent conditional branching, and it does not support training or scientific computation.

Axiom 2: Inspired by biological precision (*17*), NorthPole is optimized for 8-, 4-, and 2-bit low-precision (*5*). This is sufficient to achieve state-of-the-art inference accuracy on many neural networks while dispensing with the high-precision required for training.

Axiom 3: NorthPole has a distributed, modular core array (16-by-16), with each core capable of massive parallelism (8192 2-bit operations per cycle) (Fig. 2F). Cortex-like modularity (*18*) of the tiled core array enables homogeneous scalability in two dimensions and, perhaps, even in three dimensions (*19*) and is also amenable to heterogeneous chiplet integration.

Axiom 4: NorthPole distributes memory among cores (Figs. 1B and 2F) and, within a core, not only places memories near compute (*2*) but also intertwines critical compute with memory (Fig. 2, A and B). The nearness of memory and compute enables each core to exploit data locality for energy efficiency. NorthPole dedicates a large area to on-chip memory that is neither centralized nor organized in a traditional memory hierarchy.

Axiom 5: NorthPole uses two dense networks-on-chip (NoCs) (*20*) to interconnect the cores, unifying and integrating the distributed computation and memory (Fig. 2, C and D) that would otherwise be fragmented. These NoCs are inspired by long-distance white-matter and short-distance gray-matter pathways in the brain and by neuroanatomical topological maps found in cortical sensory and motor systems (*21*). One gray matter–inspired NoC enables spatial computing between adjacent cores (Fig. 3 and fig. S1). Another white matter–inspired NoC enables neuron activations to be spatially redistributed among all cores.

Axiom 6: Another two NoCs enable reconfiguring synaptic weights and programs on each core for high-speed operation of compute units (Fig. 2, C and D). The brain’s organic biochemical substrate is suitable for supporting many slow analog neurons, where each neuron is hardwired to a fixed set of synaptic weights. Directly following this architectural construct leads to an inefficient use of inorganic silicon, which is suitable for fewer and faster digital neurons. Reconfigurability resolves this key dilemma by storing weights and programs just once in the distributed memory and reconfiguring the weights during the execution of each layer using one NoC and reconfiguring the programs before the start of the layer using another NoC. Stated differently, these two NoCs serve to substantially increase (up to 256 times, in some cases) the effective on-core memory sizes for weights and programs such that each core computes as if the weights and program for the entire network are stored on every core. Consequently, NorthPole achieves 3000 times more computation and 640 times larger network models than TrueNorth (*14*), although it has only four times more transistors (supplementary text S1).

Axiom 7: NorthPole exploits data-independent branching to support a fully pipelined, stall-free, deterministic control operation for high temporal utilization without any memory misses, which are a hallmark of the von Neumann architecture. Lack of memory misses eliminates the need for speculative, nondeterministic execution. Deterministic operation enables a set of eight threads for various compute, memory, and communication operations to be synchronized by construction and to operate at a high utilization.

Axiom 8: Turning to algorithms and software, co-optimized training algorithms (fig. S3) enable state-of-the-art inference accuracy to be achieved by incorporating low-precision constraints into training. Judiciously selecting precision for each layer enables optimal use of on-chip resources

IBM Research, San Jose, CA, USA.
*Corresponding author. Email: dmodha@us.ibm.com
†These authors contributed equally to this work.

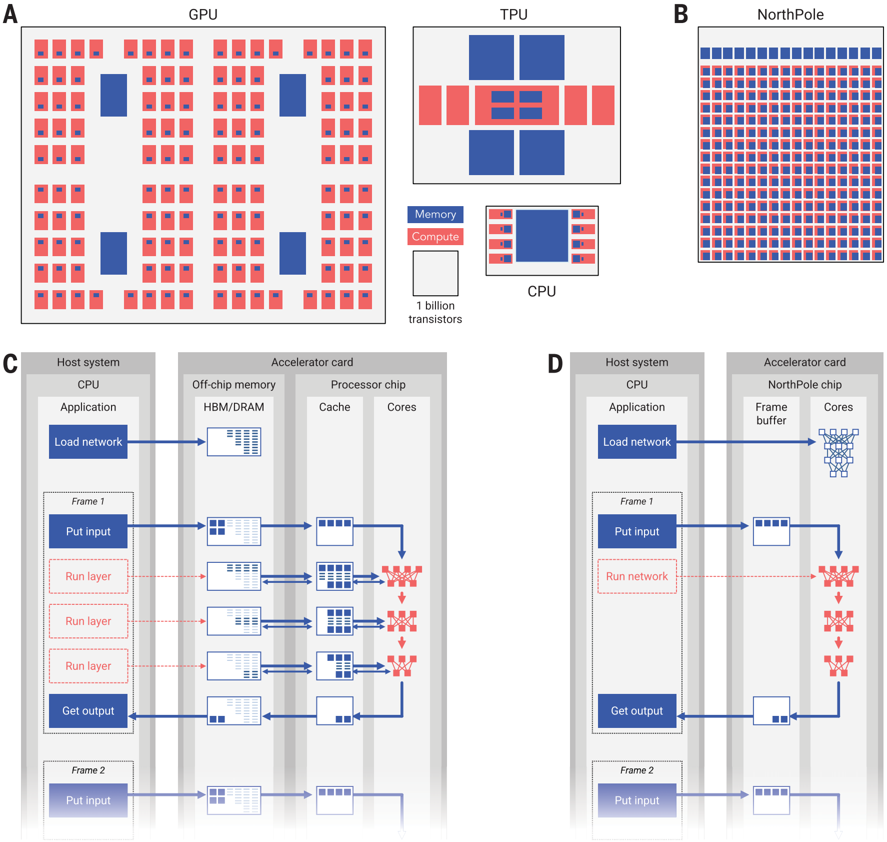

Fig. 1. Memory near compute and active memory: Processor memory organization dictates on-chip data pathways and usage model. (**A**) Physical organization of on-chip memory (blue) and compute (red) are diagrammed for representative processors, scaled to constant transistors per unit area. Large central blocks of memories separate compute from memory in central, tensor, and graphics processing units (CPU, TPU, and GPU)—which use off-chip memory (not shown) that can be 100 to 1000 times larger than the on-chip memory—and induce a hierarchical bus architecture (not shown) to relay data to cores. GPU (Nvidia A100, 54 billion transistors, 7-nm process node, 60.25-megabyte on-chip memory) has 108 streaming multiprocessors (SMs) that share different partitions of the 40-megabyte L2 cache, with 192-kilobyte L1 cache or shared memory in each SM (*26*). TPU (Google TPUv4i, 16 billion transistors, 7-nm process node, 144-megabyte on-chip memory) has four cores with 128 megabytes of common memory (*27*). CPU (AMD Zen 3, 4.15 billion transistors, 7-nm process node, 36.5-megabyte on-chip memory) has a 32-megabyte L3 cache shared by all eight cores, with much smaller L2 and L1 caches in each core (*28*). (**B**) NorthPole has no centralized memory, instead distributing on-chip memory across the core array to colocate memory near compute (axiom 4). NorthPole (22 billion transistors, 12-nm process node, 224-megabyte on-chip memory) has a 16-by-16 array of cores (axiom 3) with enough on-chip memory (axiom 4) to store entire networks for many applications, without loading weights or instructions from off-chip memory during computation. The 16 framebuffer columns at the top of the NorthPole chip allow on-chip computation of the current frame to overlap off-chip communication of other frames. (**C**) Data flow diagrammed for accelerators such as CPU, GPU, and TPU, which have a hierarchy of off-chip memories, such as high-bandwidth memory (HBM) or dynamic random access memory (DRAM), and centralized on-chip caches. The leftmost column (Application) lists the sequence of operations, with the corresponding memory states and accesses in the columns to the right. Depending on the network size, either an off-chip memory located on the accelerator card or an on-chip cache stores instructions and weights (dashes) for the entire network. For each frame, the host application puts input tensor(s) of activations (boxes) into the off-chip memory to be conveyed to compute cores via on-chip caches. Memory misses during the computation of the network trigger cores to repeatedly fetch data from more-distant levels of the memory hierarchy. For each layer, the data move back and forth between cores and memory. (**D**) NorthPole stores the entire network on-chip, using prescheduling to ensure that there are no memory misses. For each frame, the host application writes input tensor(s) of activations (boxes) into the on-chip framebuffer to be conveyed to compute cores and, at the end of network computation, reads the output tensor(s). NorthPole handles scheduling of layers and data movement internally while maintaining neuron activations entirely within the core array. There is no layer-by-layer interaction between the host and NorthPole, which has only three commands (write input tensor, run network, read output tensor) and thus appears as an active memory chip, supporting operation that is autonomous from the host computer (axiom 10).

without compromising inference accuracy (supplementary texts S9 and S10).

Axiom 9: Codesigned software (fig. S3) automatically determines an explicit orchestration schedule for computation, memory, and communication to achieve high compute utilization in space and time while ensuring that there are no resource collisions. Network computation is broken down into layers, and each layer computation is spatially mapped to the core array. To minimize long-distance communication, spatial locality of neural activations is maintained across layers when possible. To prevent resource idling, core computation and intra- and intercore data movement are temporally synchronized so that data and programs are present on each core before use. Together, algorithms and software constitute an end-to-end toolchain that exploits the full capabilities of the architecture while providing a path to migrate existing applications and workflows to the architecture.

Axiom 10: NorthPole employs a usage model that consists of writing an input frame and reading an output frame (Figs. 1D and 3), which enables it to operate independently of the attached general processor (*16*). Once NorthPole is configured with network weights and an orchestration schedule, it is fully self-contained and computes all layers on-chip, requiring no off-chip data or weight movement. Thus, externally, the entire NorthPole chip appears as a form of active memory with only three commands (write input, run network, read output), with the minimum possible input-output bandwidth requirement; these features make NorthPole well suited for direct integration with high-bandwidth sensors, for embedding in computing and IT infrastructure, and for real-time, embedded control of complex systems (*22*).

Subsets of these axioms are found in previous architectures; however, as a whole, they are distinctive to the NorthPole architecture (Figs. 1, B and D; 2; and 3), which delivers outstanding performance on the fundamental metrics of energy, space, and time (Fig. 4) at state-of-the-art inference accuracy. The whole is greater than the sum of the parts. Details of the architecture, usage model, and translation of axioms into quantitative architecture design choices are in the supplementary text.

Silicon implementation

NorthPole has been fabricated in a 12-nm process and has 22 billion transistors in an 800-mm^2 area, 256 cores, 2048 (4096 and 8192) operations per core per cycle at 8-bit (at 4- and 2-bit,

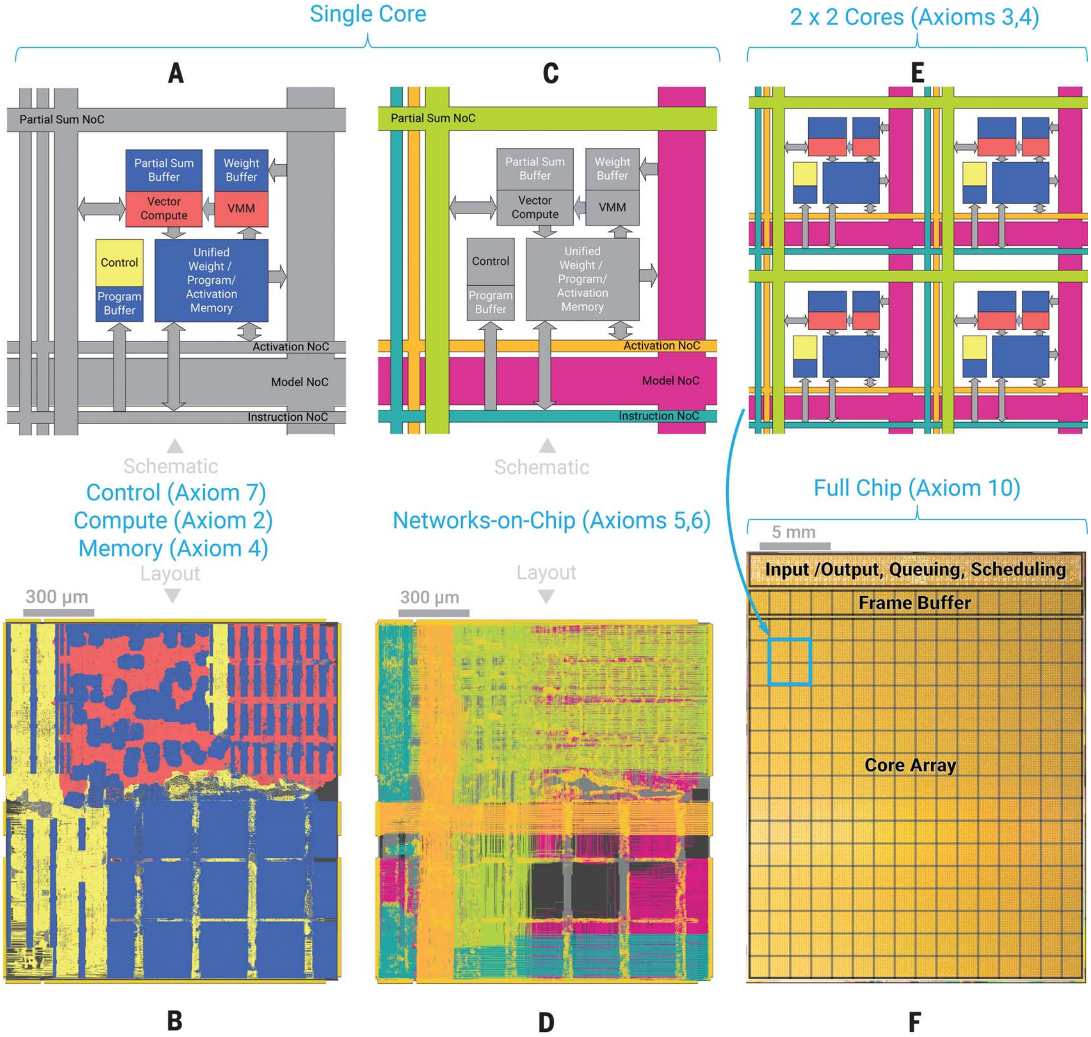

Fig. 2. NorthPole axiomatic architecture and implementation. (**A** and **B**) Core architecture schematic (A) and corresponding core circuit physical layout (B): Compute (red) includes the vector-matrix multiplier (VMM) and vector compute, which is integrated with memory (blue), and is driven by eight control threads (yellow). VMM supports 8-, 4-, and 2-bit precisions (axiom 2). Memories for weights, programs, and neuron activations are shared and unified. The unified memory is placed near the VMM and vector compute units (axiom 4). The physical placement of the VMM, vector units, and control units is intertwined, respectively, with weight buffer, partial sum buffer, and program buffer memories (axiom 4). At each cycle on each core, the VMM multiplies an input activation vector from the unified memory with a weight matrix from the weight buffer and transmits the output to the vector units, which in turn can communicate with the vector units on adjacent cores and, eventually, transmit the output to an activation function unit that, finally, writes the output activations back into the unified memory. The control unit has eight concurrent threads for orchestrating compute and communication (axiom 7). (**C** and **D**) Core NoC schematic (C) and core wire physical layout (D): A 512-wire partial sum NoC (green) transfers values between vector compute units in adjacent cores, enabling spatial computing (axiom 5); a 256-wire activation NoC (orange) transfers input frames to the cores from the framebuffer, transfers output frames from the cores to the framebuffer, and enables shuffling neuron activations among the unified memory of cores (axiom 5); a 1024-wire model NoC (magenta) carries distributed synaptic weights to the weight buffer (axiom 6); and a 256-wire instruction NoC (teal) delivers distributed instructions to the program buffer (axiom 6). (**E**) Schematic of spatially tiled two-by-two core array illustrating core-to-core communication using NoCs. (**F**) Die photograph of the manufactured, fully functional NorthPole chip with a 25-mm–by–31.8-mm footprint, illustrating periphery (input and output, queuing, and scheduling; axiom 10), framebuffer memory (for input and output tensor storage; axiom 10), and a 16-by-16 core array (axiom 3) where each core is 1.53-mm–by–1.65-mm and has 768 kilobytes of memory per core (axiom 4).

respectively) precision, 224 megabytes of on-chip memory (768 kilobytes per core, 32-megabyte framebuffer for input-output), more than 4096 wires crossing each core both horizontally and vertically, and 2048 threads. It is also fully operational in the first silicon implementation (Fig. 2F), is now deployed in a PCIe form factor printed circuit board (fig. S2), and has an end-to-end software toolchain (fig. S3 and supplementary texts S8 to S15). Implementation details are provided in supplementary texts S16 to S19. On the benchmark ResNet50 network, NorthPole can operate at frequencies ranging from 25 to 425 MHz with nearly linear power scaling (fig. S4), making it suitable for a myriad of applications from cloud to self-driving cars and achieving an unprecedented absolute energy metric of >1000 frames per joule (red point in fig. S4), with respect to energy consumed just by the NorthPole processor. Implementations of ResNet50 with 8-, 4-, mixed 4/2-, and 2-bit layers lead to progressively higher throughput and higher energy efficiency (fig. S5) and to progressively lower memory usage (supplementary text S11).

Energy, space, and time

For methodological rigor that ensures a fair and level comparison of various implementations, it is critical that all evaluation metrics be independent of the details of the implementations, which can vary arbitrarily across architectures at the discretion of the designers. The architecture-independent goodness metric adopted here is that all implementations must be measured at state-of-the-art inference accuracy. The architecture-independent cost metrics adopted here are now introduced.

Turning first to energy, different integrated-circuit (IC) implementations have different throughputs [in frames per second (FPS)] at different power consumptions (in watts). Therefore, FPS per watt (equals frames per joule) is a widely used energy metric for comparing

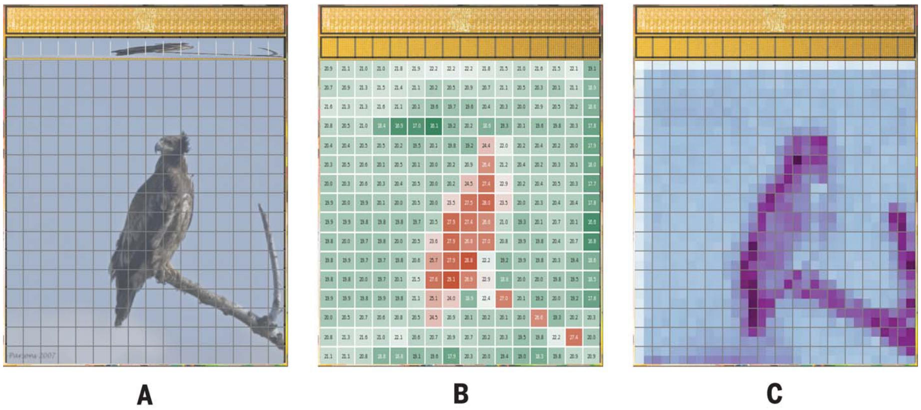

Fig. 3. NorthPole supports a spatial computing model for cortex-like topographic computing by exploiting network connectivity between adjacent cores. (**A**) An input image is divided into 16 subimages that are distributed to the framebuffer and then into 256 subimages that are distributed to the core array while retaining spatial structure (axiom 3). Intercore connectivity (axiom 5) enables receptive fields of neurons on a core to span nearby cores. [Photo credit: Raymond Parsons, used with permission] (**B**) The power consumption estimation in milliwatts by vector-matrix multipliers in the 16-by-16 NorthPole core array for the first convolutional layer of the ResNet50 network. Power consumption is computed by running a gate-level simulation of the NorthPole silicon design. The increased energy consumption shown in red matches the shape of the bird, demonstrating the in situ, activity-dependent neural computation of NorthPole that requires minimal communication between neighboring cores. (**C**) The average spatial intensity of tensor neuronal activation is shown for a layer about halfway through the layer-by-layer computation of the ResNet50 network. Comparing (C) with (A) demonstrates the slow spatial dispersion of neuron activity through the layers.

ICs. On this metric, Fig. 4A (*x* axis) compares prevalent architectures with NorthPole. Compared with the V100 GPU, which is implemented using a comparable 12-nm silicon technology process node and with a comparable number of transistors, NorthPole delivers 25 times more frames per joule. Many other compared architectures use silicon process nodes that are more advanced than 12 nm, and even though these architectures have an inherent advantage of using transistors with greater energy efficiency, NorthPole outperforms them. Even relative to the H100 GPU, which is implemented using a 4-nm silicon process node, NorthPole delivers five times more frames per joule. NorthPole has high energy efficiency because of less data movement, which is enabled by a lack of off-chip memory, the intertwining of on-chip memory with compute, the locality of spatial computing, and a better use of low-precision computation. Figure 4C decomposes FPS per watt into a product of FPS and 1/watt to illustrate through iso-energy-efficiency curves that, for prevalent architectures, higher FPS or lower power come, respectively, at the cost of higher power or lower FPS. Architecture trumps Moore's law.

Turning next to space, different ICs occupy different areas in different technology processes

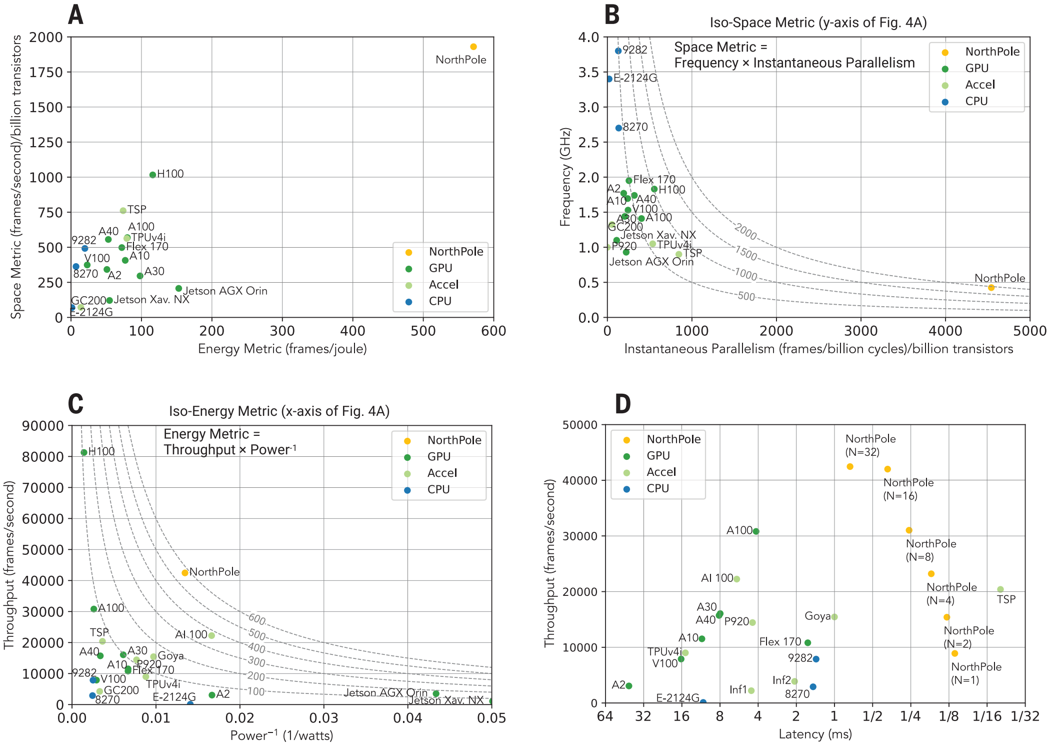

Fig. 4. NorthPole outperforms prevalent architectures on energy, space, and time metrics on ResNet50 at state-of-the-art inference accuracy. (**A**) NorthPole's space metric (FPS per billion transistors) and energy metric (frames per joule, equal to FPS per watt) exceed those of contemporary GPUs, CPUs, and accelerators (Accel). (**B**) Decomposition of FPS per billion transistors into the product of clock rate (cycles per second) on the *y* axis and normalized instantaneous parallelism [(frames per cycle) per billion transistors] on the *x* axis. The isoparametric curves denote frontiers of constant FPS per billion transistors. NorthPole performance derives from highly utilized parallelism, in contrast to both high–clock frequency CPUs (blue cluster) as well as parallelized GPUs (green cluster). (**C**) Decomposition of frames per joule into the product of FPS on the *y* axis and the inverse of power (1/watt) on the *x* axis. The isoparametric curves denote frontiers of constant frames per joule. (**D**) NorthPole throughput (FPS) on the *y* axis versus latency (ms) on the *x* axis in reversed log scale. NorthPole throughput and latency are reported at various batch sizes of 1, 2, 4, 8, 16, and 32. Compared with most comparative architectures, NorthPole achieves lower latency at similar throughput or, alternatively, higher throughput at lower latency. TSP (Tensor Streaming Processor) (*23*) is optimized for low latency but has substantially lower energy and space efficiencies. Power for all architectures includes total board power. For the details of each data point, see Table 1.

Table 1. ResNet50 performance comparison of neural inference across processors. Comparative processor performance obtained from published results that ran the ResNet50 network for image classification (*8*). Measured power includes total board power for all processors, and all networks are at state-of-the-art inference accuracy. The space and time-space metrics are calculated values. The energy metric is a reported value when available, otherwise it is calculated as FPS divided by thermal design power. The latency metric is a reported value when available, otherwise it is calculated as batch size divided by FPS. References and methodology for comparative performance numbers are provided in table S1. Accel, accelerator; CGRA, coarse-grain reconfigurable array; NIPU, neural inference processing unit; NA, not available.

	Design choices				Measured quantities		Computed figures of merit				Qualitative			
Processor	Process node (nm)	Transistors (billions)	Frequency (GHz)	Batch size (frames)	Power (W)	Throughput (FPS)	Energy (frames per joule)	Space (FPS per billion transistors)	Time-space (frames per billion cycles per billion transistors)	Latency (ms)	Precision	Specialized to neural inference	Off-chip memory	Architecture category
Xeon 9282	14	16	3.800	11	400	7878	20	492	130	1.4	INT8	No	Yes	CPU
Xeon 8270	14	8	2.700	NA	205	2906	7	363	134	1.5	INT8	No	Yes	CPU
E-2124G	14	1.4	3.400	NA	71	98	1.4	70	21	10.9	INT8	No	Yes	CPU
TPUv4i	7	16	1.050	16	175	9005	79	563	536	15.0	BF16	Yes	Yes	Accel:NIPU
Goya	16	NA	NA	10	103	15,453	149	NA	NA	1.0	INT8	Yes	Yes	Accel:NIPU
AI 100	7	NA	NA	8	60	22,250	370	NA	NA	5.9	INT8	Yes	Yes	Accel:NIPU
K200	14	NA	1.100	NA	160	NA	NA	NA	NA	NA	INT8	Yes	Yes	Accel:NIPU
Inf1	NA	NA	NA	10	NA	2207	NA	NA	NA	4.5	FP16	Yes	Yes	Accel:NIPU
Inf2	NA	NA	NA	8	NA	3878	NA	NA	NA	2.1	BF16	Yes	Yes	Accel:NIPU
TSP	14	27	0.900	1	275	20,400	74	761	846	0.049	INT8	No	No	Accel:CGRA
SN10	7	40	NA	NA	NA	NA	NA	NA	NA	NA	NA	No	Yes	Accel:CGRA
GC200	7	59	1.325	NA	300	4250	14	72	54	NA	FP16	No	Yes	Accel:Manycore
P920	12	NA	1.000	64	130	14,442	111	NA	NA	4.4	INT8	No	Yes	Accel:Manycore
WSE-2	7	2600	1.100	NA	15,000	NA	NA	NA	NA	NA	NA	No	No	Accel:Manycore
Flex 170	6	21.7	1.950	NA	150	10,811	72	498	255	1.6	INT8	No	Yes	GPU
V100	12	21.1	1.530	128	300	7896	23	374	244	16.2	INT8	No	Yes	GPU
Jetson Xavier NX	12	9	1.1	NA	20	1092	55	121	110	NA	INT8	No	Yes	GPU
Jetson AGX Orin	8	17	0.939	NA	40	3526	153	207	223	NA	INT8	No	Yes	GPU
T4	12	13.6	1.590	128	70	5003	72	368	231	25.6	INT8	No	Yes	GPU
A2	8	8.9	1.770	128	60	3059	51	342	193	41.8	INT8	No	Yes	GPU
A10	8	28.3	1.695	128	150	11,520	77	407	240	11.1	INT8	No	Yes	GPU
A40	8	28.3	1.740	128	300	15,727	53	556	320	8.1	INT8	No	Yes	GPU
A30	7	54.2	1.440	128	165	16,057	98	296	206	8.0	INT8	No	Yes	GPU
A100	7	54.2	1.410	128	400	30,814	80	569	404	4.2	INT8	No	Yes	GPU
H100	4	80	1.830	NA	700	81,292	116	1016	555	NA	INT8	No	Yes	GPU
NorthPole	12	22	0.425	32	74	42,460	571	1930	4541	0.75	INT8, 4, or 2	Yes	No	Accel:NIPU

but can be compared in terms of the number of transistors. Therefore, FPS per billion transistors is a meaningful space metric for comparing ICs. On this metric, Fig. 4A (*y* axis) demonstrates that NorthPole outperforms prevalent architectures and that NorthPole delivers five times more FPS per billion transistors than the V100 GPU. It is notable that an architecture that devotes substantially more transistors to memory than logic outperforms prevalent architectures that devote substantially more transistors to logic than memory, on a performance-per-transistor space metric. To throw light on this advantage, observe that the space metric, FPS per billion transistors, can be decomposed into a product of normalized instantaneous parallelism [(frames per cycle) per billion transistors] and clock rate (cycles per second). All compared architectures implement considerably faster clock rates than NorthPole, and yet NorthPole outperforms on the space metric by achieving substantially higher instantaneous parallelism (through high utilization of many highly parallel compute units specialized for neural inference) and substantially lower transistor count owing to low precision (Fig. 4B).

Turning finally to time, different ICs permit different numbers of input frames to be bundled for processing in a batch, so batch size per FPS, or latency, is an appropriate time(-to-solution) metric. NorthPole throughput and latency are reported at various batch sizes of 1, 2, 4, 8, 16, and 32, thus providing an entire throughput-latency frontier that envelops most of the prevalent architectures (Fig. 4D). For example, with a batch size of 32, NorthPole can achieve a high throughput of 42,460 FPS at a latency of 753 μs, and, with a batch size of one (a single image), NorthPole can operate at a low latency of 106 μs at a throughput of 9454 FPS [(*23*) reports a processor with lower latency but also lower energy and space efficiencies]. NorthPole achieves a lower latency because it has a higher FPS owing to massive parallelism, has low latency local memory access, and hides communication latency behind compute operations, and because it can compute with much smaller batch sizes while maintaining high utilization owing to spatial computing.

The results shown in Figure 4 derive from the ResNet50 image classification network, which is widely used as a benchmark. Similar comparative results can be seen for the Yolo-v4 detection network (fig. S6).

Applications

NorthPole's on-chip memory is sufficient to implement many widely used networks across a variety of application domains; a few canonical examples are the ResNet family (for classification), Yolo and SSD-VGG (for detection), PSPNet (for segmentation), RetinaMask (for instance segmentation), BERT (for natural language processing), and DeepSpeech2 (for speech recognition). For examples of networks that are implementable on a single NorthPole chip, see supplementary text S12 and tables S7 to S12. With 4- or 2-bit precisions, the number of neural network weights stored on NorthPole can be doubled or quadrupled, respectively. Memory permitting, more than one network can be resident in the on-chip memory, thus supporting multitenancy. Inspired by the brain, which has no off-chip memory, NorthPole is optimized for on-chip networks, but, if so desired, larger networks can be broken down into smaller subnetworks that do fit in NorthPole's model memory, and these subnetworks can be pipelined on multiple NorthPole chips (supplementary text S25). Given the present silicon technology roadmap (*10*), 50 times more memory on-chip seems possible by going from 37 million transistors per square millimeter on the present 12-nm implementation to 2 billion, thus enabling much larger networks to fit on-chip.

Conclusions

As seen from the inside of the chip, at the level of individual cores, NorthPole appears as memory near compute; as seen from the outside of the chip, at the level of input-output, it appears as an active memory. NorthPole is an architectural innovation at the intersection of brain-inspired computing and semiconductor technology, which defines a frontier that promises to expand. Specifically, the NorthPole trajectory promises to evolve along dimensions of algorithms (natively optimized for NorthPole), systems (scale-out with multiple boards, scale-up with multiple chips per board, and direct sensor integration), modularity (chiplets with varying core array sizes from tiny to large), packaging (heterogeneous chiplet integration), architecture (tileability, three-dimensional integration), silicon scaling (*10*), silicon optimization (*24*), and, potentially, post-silicon technologies (*25*). Interestingly, the EDVAC report began with neurons and synapses, and now the rising importance of neural inference applications has brought the field full circle.

REFERENCES AND NOTES

1. J. von Neumann, *IEEE Ann. Hist. Comput.* **15**, 27–75 (1993).
2. O. Mutlu, S. Ghose, J. Gómez-Luna, R. Ausavarungnirun, in *Emerging Computing: From Devices to Systems*, Computer Architecture and Design Methodologies series, M. M. S. Aly, A. Chattopadhyay, Eds. (Springer, 2023).
3. W. A. Wulf, S. A. McKee, *Comput. Archit. News* **23**, 20–24 (1995).
4. M. Horowitz, in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* (IEEE, 2014), pp. 10–14.
5. J. J. Dongarra, *Commun. ACM* **65**, 66–72 (2022).
6. J. Backus, *Commun. ACM* **21**, 613–641 (1978).
7. T. J. Sejnowski, P. S. Churchland, *The Computational Brain* (MIT Press, 1994).
8. J. Deng *et al.*, in *IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2009), pp. 248–255.
9. T. J. Sejnowski, *The Deep Learning Revolution* (MIT Press, 2018).
10. S. Datta, W. Chakraborty, M. Radosavljevic, *Science* **378**, 733–740 (2022).
11. R. H. Norman, "Solid state switching and memory apparatus," US Patent 3,562,721 (1971).
12. S. S. Iyer *et al.*, *IBM J. Res. Develop.* **49**, 333–350 (2005).
13. N. C. Thompson, K. Greenewald, K. Lee, G. F. Manso, arXiv: 2007.05558 [cs.LG] (2020).
14. P. A. Merolla *et al.*, *Science* **345**, 668–673 (2014).
15. C. Mead, M. Ismail, Eds., *Analog VLSI Implementation of Neural Systems* (Springer, 1989).
16. C. E. Leiserson *et al.*, *Science* **368**, eaam9744 (2020).
17. T. M. Bartol Jr. *et al.*, *eLife* **4**, e10778 (2015).
18. V. B. Mountcastle, *Brain* **120**, 701–722 (1997).
19. S. S. Iyer, *MRS Bull.* **40**, 225–232 (2015).
20. T. Bjerregaard, S. Mahadevan, *ACM Comput. Surv.* **38**, 1 (2006).
21. M. I. Sereno, M. R. Sood, R.-S. Huang, *Front. Syst. Neurosci.* **16**, 787737 (2022).
22. J. Degrave *et al.*, *Nature* **602**, 414–419 (2022).
23. D. Abts *et al.*, in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)* (IEEE, 2020), pp. 145–158.
24. F. Khan, E. Cartier, J. C. S. Woo, S. S. Iyer, *IEEE Electron Device Lett.* **38**, 44–47 (2017).
25. W. Wan *et al.*, *Nature* **608**, 504–512 (2022).
26. J. Choquette, E. Lee, R. Krashinsky, V. Balan, B. Khailany, in *IEEE International Solid-State Circuits Conference (ISSCC)* (IEEE, 2021), pp. 48–50.
27. N. P. Jouppi *et al.*, in *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)* (IEEE, 2021), pp. 1–14.
28. M. Evers, L. Barnes, M. Clark, *IEEE Micro* **42**, 7–12 (2022).

ACKNOWLEDGMENTS

We thank many collaborators: I. Ahmed, T. Amberg, N. Antoine, F. Baez, H. Baier, L. Berge, R. Bhimarao, J. Bjorgaard, M. Boraas, B. Brezzo, M. Butterbaugh, E. J. Campbell, E. Colgan, M. Criscolo, C. di Nolfo, M. Doyle, M. Grassi, C. Guido, D. Hitchings, R. Hoke, K. Holland, S. Hui, T. Jones, S. Kedambadi, P. Kourdis, A. LaPiana, G. LaPiana, S. Lekuch, Z. Liu, R. Loboprabhu, A. Lonkar, D. Ma, G. Maass, S. Machha, V. Mahadevan, P. Mann, M. Mastro, K. O'Connell, B. Parikh, H. Penner, R. Purdy, M. Rajwadkar, L. Rapp, K. Rice, C. Sassano, K. Schoneck, R. Scouller, J. Shaffer, D. Shukla, A. Sigler, C. Smallwood, N. Subbiah, S. Tian, T. Tidwell, D. Turnbull, M. Uman, G. Van Leeuwen, S. Vispute, I. Vo, J. Wang, C. Werner, S. Wundavilli, and C. N. Xiong. We also thank H. Chandran, J. Saxena, J. Lee, A. Prashantha, S. Rao, and the Anora Labs DFT team as well as teams from Cadence Design Systems and Globalfoundries. We thank three anonymous reviewers for their thoughtful comments. **Funding:** This material is based upon work supported by the United States Air Force under contract no. FA8750-19-C-1518. **Author contributions:** Conceptualization: D.S.M.; Architecture: R.A., J.V.A., A.S.C., D.S.M., and J.Sa. (co-leads), F.A., A.Am., A.An., P.D., M.V.D., S.K.E., N.J.M., T.K.N., C.O.O., B.T.; Algorithms: S.K.E. (lead), R.A., D.B., J.L.M., D.S.M.; Compiler: P.D. (lead), A.Am., R.A., D.B., S.K.E., R.G., N.J.M., J.L.M., D.S.M., T.K.N., B.T., I.T.; Validator: A.An. (lead), N.J.M., D.S.M., T.K.N., B.T.; Runtime: J.A.K. (lead), A.An., R.A., A.S.C., M.V.D, R.G., D.S.M., D.F.S., B.T.; Logic design: A.S.C. (lead), F.A., G.J.G., M.I., C.O.O., J.Sa., J.Si., T.U.; Logic verification: M.V.D., R.A., and D.S.M. (co-leads), A.An., A.Am., D.B., P.J.C., P.D., S.K.E., M.D.F., R.G., G.J.G., M.I., J.A.K., N.J.M., J.L.M., T.K.N., W.P.R., B.S., J.Si., J.Sa., D.F.S., B.T., I.T., T.U.; Physical design: F.A. and C.O.O. (co-leads), J.V.A., A.S.C., Y.N., J.Sa., T.U.; Design for test: J.Sa. (lead), J.V.A., A.S.C., G.J.G., M.I., C.O.O., J.Si., T.U.; Board design: F.A. and K.S. (co-leads), J.V.A., A.S.C., M.V.D., C.O.O., W.P.R.; Test and measurement: R.A., J.V.A., A.S.C., M.V.D., C.O.O., and J.Sa. (co-leads), F.A., A.Am., A.An., D.B., P.J.C., P.D., S.K.E., M.D.F., R.G., G.J.G., J.A.K., M.I., N.J.M., J.L.M., D.S.M., T.K.N., W.P.R., B.S., J.Si., K.S., D.F.S., B.T., I.T., T.U.; Resources: M.V.D., M.D.F., G.J.G., J.A.K., J.L.K., N.J.M., D.S.M., W.P.R.; Supervision: D.S.M. served as the principal investigator. J.V.A. and J.Sa. supervised hardware. R.A., M.D.F., and B.T. supervised software. K.S. supervised board design; Project administration: J.L.K. (lead), P.J.C., D.S.M., W.P.R., B.S.; Funding acquisition: D.S.M. (lead), J.L.K. **Competing interests:** The authors are researchers at IBM, which has a commercial interest in the NorthPole intellectual property and technology. **Data and materials availability:** All data needed to evaluate the conclusions in this study are available in the main text or the supplementary materials. **License information:** Copyright ©

SUPPLEMENTARY MATERIALS

science.org/doi/10.1126/science.adh1174
Supplementary Text
Figs. S1 to S8
Tables S1 to S12
References (*29*–*104*)

Submitted 17 February 2023; accepted 1 September 2023
10.1126/science.adh1174