The N3XT Approach to Energy-Efficient Abundant-Data Computing

Mohamed M. Sabry Aly[1,2], Tony F. Wu[1], Andrew Bartolo[3], Yash H. Malviya[1], William Hwang[1], Gage Hills[1], Igor Markov[4], Mary Wootters[1,3], Max M. Shulaker[5], H.-S. Philip Wong[1] and Subhasish Mitra[1,3]
[1]Department of Electrical Engineering, Stanford University,
[2] SCSE, Nanyang Technological University, [3]Department of Computer Science, Stanford University,
[4] CSE Division, University of Michigan, [5]EECS, Massachusetts Institute of Technology

***Abstract*—The world's appetite for analyzing massive amounts of structured and unstructured data has grown dramatically. The computational demands of these abundant-data applications, such as deep learning, far exceed the capabilities of today's computing systems and are unlikely to be met with isolated improvements in transistor or memory technologies, or integrated circuit architectures alone.**

To achieve unprecedented functionality, speed and energy efficiency, one must create transformative nanosystems whose architectures are based on salient properties of the underlying nanotechnologies. Our N3XT (Nano-Engineered Computing Systems Technology) approach makes such nanosystems possible through: (*i*) new computing system architectures leveraging emerging device (logic and memory) nanotechnologies and their dense 3D integration with fine-grained connectivity to immerse computing in memory, (*ii*) new logic devices (such as carbon nanotube field-effect transistors for implementing high-speed and low-energy logic circuits) as well as high-density non-volatile memory (such as resistive memory), amenable to (*iii*) ultra-dense (monolithic) 3D integration of thin layers of logic and memory devices that are fabricated at low temperature. In addition, we explore the use of several device and integration technologies in N3XT beyond the specific ones mentioned above that are also used in our main nanosystem prototypes. We also present an efficient resiliency technique to overcome endurance challenges in certain resistive memory technologies.

N3XT hardware prototypes demonstrate the practicality of our architectures. We evaluate the benefits of N3XT using a simulation framework calibrated using experimental measurements. System-level energy-delay product of common implementations of abundant-data workloads improves by three orders of magnitude in N3XT compared to conventional architectures. These improvements impact a broad range of application workloads and architecture configurations, from embedded systems to the cloud.

I. Introduction

The future of computing is in a crisis. Progress in computing hardware has begun to stall just as massive improvements in speed and energy are needed for coming generations of transformative applications, such as artificial intelligence on massive data. Thanks to the abundance of stored data, these applications have undergone a major leap in functionality and are playing an increasingly important role in our daily lives. Applications that stress existing architectures include graph analytics (e.g., social networking), machine/deep learning [LeCun15] (e.g., real-time speech-to-text services, computer vision, natural language processing, and language translation), and knowledge-based systems (e.g., IBM Watson's Deep QA [Ferruci10]).

The execution time and energy consumption of abundant-data applications are generally dominated by off-chip memory accesses. This trend holds for a wide variety of architectures, from general-purpose [Aly15] to domain-specific accelerators [Hwang17, Jouppi17]. Because application-data needs far exceed on-chip SRAM capacity, this trend will likely continue despite on-chip transistor density advances. For example, large deep-learning networks [Shazeer17] require >16 GBytes of memory even if aggressive data-reduction techniques are applied [Han16, Courbariaux15, Rastegari16]. These obstacles cannot be overcome by isolated improvements in logic or memory technologies alone. Current integration mechanisms for memories and compute fabrics include the use of 2.5D interposers and 3D integration using wafer bonding and through-silicon vias (*TSVs*). Such TSVs have a pitch[1] on the order of microns with 1 μm TSV diameter [Ramalingam16, Huylenbroeck16, Kim16]. Aspect ratio of such TSVs, as well as mechanical stress of both TSVs and integrated fabrics (during fabrication as well as during system operation) [Jung14], limit drastic reduction of TSV dimensions. Therefore, TSVs offer limited improvement to compute-to-memory connectivity and overall system-level energy and execution time. Denser 3D interconnects are needed to support orders-of-magnitude energy and execution time improvements for emerging abundant-data workloads. For example, instead of integrating separately-manufactured integrated circuits (*ICs*), *monolithic 3D integration* [Wong07, Batude15, Shulaker17] enables multiple layers of transistors (also referred to as *active layers*) and memory cells to be deposited sequentially—vertically on top of one another—and connected by short high-density interlayer vias (*ILVs*). Such ILVs are already used for connecting wires in today's conventional ICs.

[1] minimum distance between two adjacent TSVs

The main goal of our N3XT[2] approach is to overcome the limitations of current computing systems and substantially improve energy and execution time. N3XT tightly integrates compute and memory components with fine-grained and dense connectivity to create new

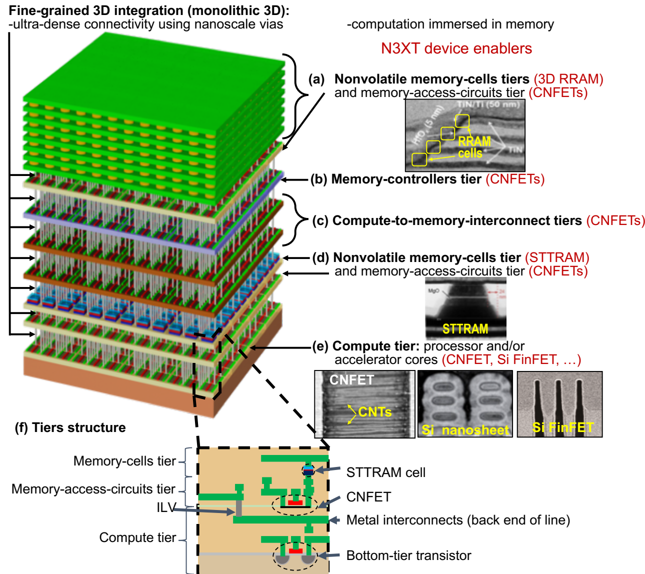


Figure 1. Envisioned N3XT architecture with fine-grained (monolithic) 3D integration of circuits on different tiers: (a) non-volatile (3D) RRAM with memory-access circuits using CNFETs (details in Figure 4) and a cross-sectional transmission electron micrograph (*TEM*) of a 3D RRAM cell [Li16]; (b) memory controllers designed using CNFETs (details in Figure 3); (c) compute-to-memory interconnect for massive concurrent accesses to memory designed using CNFETs (details in Figure 5); (d) non-volatile STTRAM with memory-access circuits using CNFETs (details in Figure 4) and a TEM of an STTRAM cell [Ohashi17]; (e) compute units located on the bottom tier (CNFETs or SiFETs can be used to implement these compute units), scanning-electron microscopy (*SEM*) of a CNFET [Shulaker14a], TEM of a Si-nanosheet [Loubet17] and Si FinFET [Ha17]; and (f) structure of different tiers highlighting transistors, memory cells, and back-end-of-line metal interconnects. Demonstrated N3XT hardware prototypes are illustrated in Figure 9 (Section III).

architectures (outlined in Figure 1). The component technologies feature:

- Energy-efficient logic devices fabricated at low temperature (<300°C).
- Low-latency and high-density non-volatile memory technologies fabricated at low temperature (<300°C).[3]
- Fine-grained and ultra-dense interconnects between logic circuits and memory units.

N3XT architectures leverage these component technology features to enable concurrent accesses to high-capacity non-volatile on-chip memory as well as energy-efficient and high-speed logic circuits for implementing compute units and memory-access circuitry. As a result, N3XT improves *system-level energy-delay-product*, defined as the product of application energy consumption and execution time, significantly (in the range of 1,000×) over current baseline systems. N3XT architectures can employ advanced thermal management solutions, but such thermal solutions are not required for the specific implementations analyzed in this paper. These thermal solutions will likely be critical when multiple compute tiers are interleaved with memory [Aly15] (beyond the scope of this paper).

While N3XT can accommodate a variety of technology options, we focus on the following device and integration technologies that satisfy our requirements previously mentioned:

- Carbon nanotube Field-Effect Transistors (*CNFETs*) for low-energy and high-speed logic circuits.
- Metal-oxide resistive RAM (*RRAM*) and spin-torque transfer RAM (*STT-MRAM* or *STTRAM*, one of several known magnetoresistive RAM technologies) for low-latency and high-density non-volatile memories.
- Monolithic 3D integration of logic and memory layers.

The N3XT approach has been demonstrated in several hardware prototypes [Shulaker14b, Shulaker17, Wu18]. The processing and design steps used to create such implementations of N3XT are compatible with those of existing silicon-based ICs. Furthermore, N3XT can adopt various other nanotechnology and integration options, explored later in this paper.

The remaining part of the paper is structured as follows. We review the N3XT architecture [Aly15] in Section II, then highlight key technology enablers and recent experimental demonstrations in Section III. Section IV describes our exploration framework that leverages experimentally-calibrated device models to analyze the application-level runtime and energy consumption of N3XT architectures. Sections V and VI report our simulation results (using our exploration framework described in Section IV) for CPU-based and domain-specific accelerator-based N3XT systems, respectively, and quantify the benefits of N3XT compared to corresponding baseline architectures. In Section VII we quantify how various characteristics (e.g., energy, speed, density) of possibly hypothetical logic, memory and integration technologies affect the energy efficiency benefits of N3XT. In Section VIII, we identify a deficiency with limited endurance of RRAM and address it. Key insights, system-level considerations, and conclusions are presented in Section IX.

II. The N3XT Architecture

Figure 1 illustrates vertically-integrated *tiers,* each consisting of an active circuit layer (a layer with transistors) or memory cells, and a number of metal layers connecting the components of the active layer (i.e., back-end-of-line metal layers, Figure 1f) that cumulatively form the N3XT

[2] Nano-Engineered Computing Systems Technology, pronounced as "next"
[3] All of the fabrication steps for resistive RAM (targeted non-volatile memory in this paper) can be performed at <300°C. At the end of the fabrication process, a final anneal step (also used for conventional IC fabrication) is performed at ~420°C anneal.

architecture [Aly15]. The tiers can be fabricated using the same methodologies applied to build the hardware prototypes discussed later in Section III. Individual tiers are designated as follows:

- *compute tiers*,

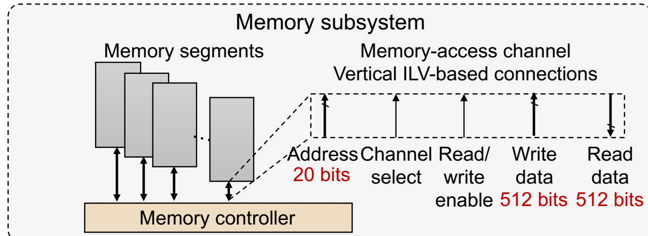


Figure 2. Memory subsystem architecture. A single memory controller (Figure 3) provides concurrent accesses to multiple memory segments (Figure 4a), through a dedicated memory-access channel to each segment. A *memory-access channel* comprises the address, read- and write-data connections. These connections are annotated with sample bitwidths, but N3XT can support different (e.g., wider) bitwidths. Multiple memory controllers constitute the memory-controller tier (Figure 1b)

- *memory tiers* (Figures 1a, 1b and 1d) with memory cells, memory controllers and memory access circuits,
- *compute-to-memory interconnect tiers* (Figure 1c), and
- *cooling tiers*.

Tiers are linked using nanoscale ILVs, e.g., conventional vias used for connecting metal layers in today's ICs [Jan15]. The high density of ILVs is essential to achieving the massive benefits in application energy and execution time (shown later in Sections V and VI). In particular, at the 14 nm and 7 nm technology nodes, ILV pitches are 80 nm and 40 nm, respectively [Jan15, Wu16], whereas TSV pitch is on the order of 3-5 μm [Huylenbroeck16, Kim16]. Thus, ILVs provide substantially higher number of connections per unit area (vs. TSVs), supporting wider and many more concurrent accesses to memory—this advantage is preserved even under the assumption that both densities scale at the same rate.

A compute tier supports CPU cores, GPUs, domain-specific accelerators, and combinations thereof [Hennessy12]. In this paper, we generally assume digital logic with local SRAM blocks, but analog circuit blocks can be integrated as well. To support significant heat dissipation [Rusu10], compute tiers abut cooling tiers. For simplicity, we assume a single compute tier located at the bottom closest to the heat sink.

Each memory tier in the memory subsystem (Figure 2) can be one of the following:

- *a memory-controller tier,*
- *a memory-access-circuits* (sense amplifiers, selector, address decoders, etc.) *tier,*
- *a memory-cells tier.*

A memory-controller tier typically contains multiple memory controllers, each providing concurrent accesses to multiple memory segments through multiple channels (Figure 2). Each channel provides wide data access to a memory segment, as illustrated in Figures 2 and 3. A memory controller in our N3XT architecture consists of:

- an *address queue* to buffer incoming memory-access requests,
- a *write-data queue* to temporarily store incoming data,
- a set of *read-data registers*, one per memory-access channel,

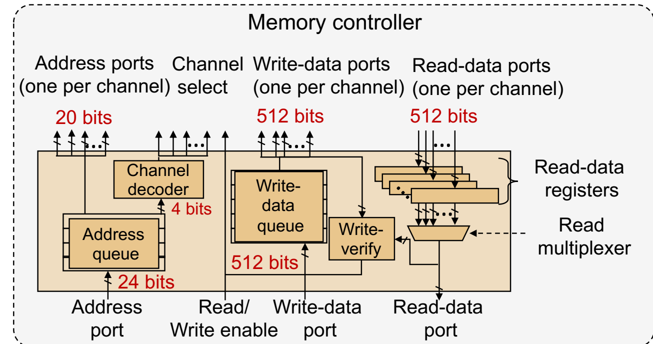


Figure 3. Memory controller architecture in N3XT. Memory-controller tier (Figure 1b) consists of multiple controllers. Shown are the microarchitecture components of a memory controller, while input and output ports are annotated with sample bitwidths.

- *logic blocks*—channel decoder and read multiplexer—for data or address routing, and
- a *write-verify* module [Sheu09], where written data is immediately read and compared with data stored in the write-data queue to ensure successful operation. This module is required to overcome inherent limitations of the memory devices such as RRAM, discussed later (Section III).

Individual queue sizes and number of memory-access channels in a memory controller are configured to increase memory-access throughput. The memory controller services memory-access requests in-order using first-come-first-serve scheme. It buffers the address of each memory-access request in the address queue and routes each request to an access channel (corresponds to the least significant address bits). Incoming data is buffered into the write-data queue, while data read through each channel is stored in dedicated access registers inside the memory controller. The memory controller may also include an *error-correction* module.

Non-volatile memories do not require refresh operations, unlike DRAM. Therefore, memory controllers in N3XT are simpler than conventional DRAM controllers.

A memory segment (Figure 4a) consists of multiple memory arrays that are connected via H-tree interconnect [Leiserson80]. The memory-access circuits of the memory segments and arrays—address and row decoders, read and write column multiplexers, sense amplifiers, and cell selectors (Figure 4)—are located directly below the cells of each array near a corresponding memory-controller tier, as illustrated in Figure 1. While many other interconnect schemes can be used to connect arrays of memories, H-trees

are particularly effective for achieving low latency and low dynamic energy [Dong12].

Compute-to-memory interconnect tiers support for uniform memory access (*UMA*), so that each compute unit has the same access latency to each memory location. Compared to non-uniform memory access (*NUMA*), UMA requires additional multiplexing, but can significantly simplify task and data mapping for computing systems [Bolosky89]. N3XT amortizes multiplexing cost through

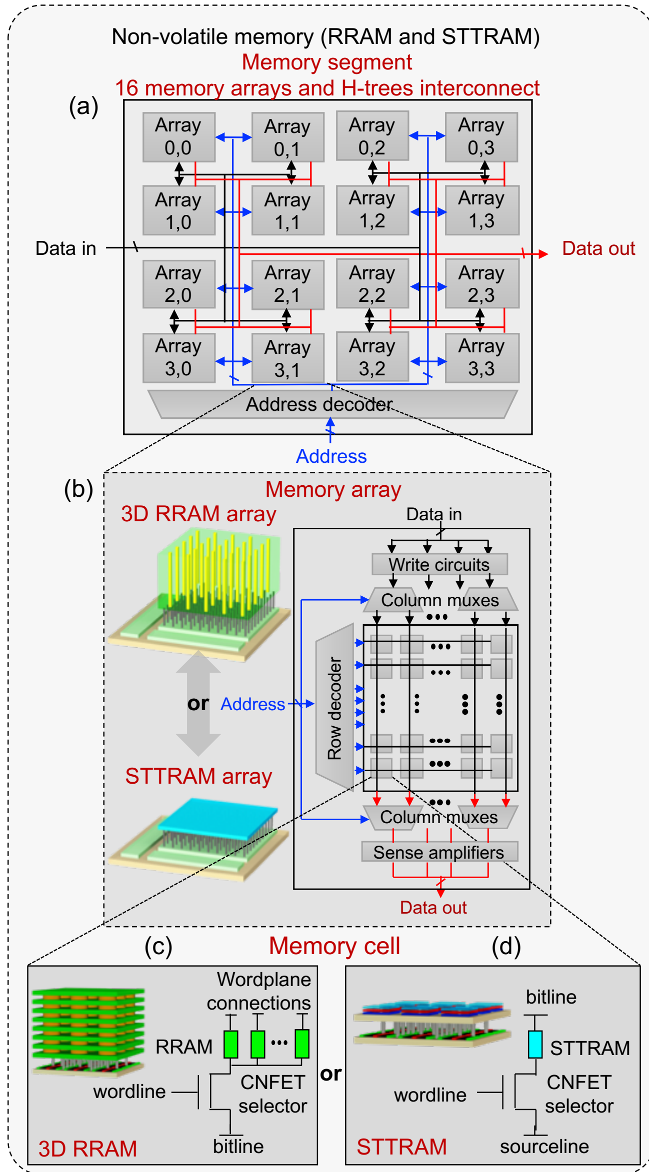


Figure 4. Architectures of the non-volatile memory cell and memory-access-circuits tiers in N3XT (Figures 1a and 1d); (a) a memory segment containing a number of subarrays (illustrated with 16 arrays, but a segment supports a larger number of subarrays) that are connected with H-trees interconnect; (b) a memory array organization highlighting the memory-access circuits (i.e., sense amplifiers, row decoders, column muxes, and write circuits); (c) 3D RRAM cell (more details in Figure 7c) and; (d) STTRAM cell. Options (c) and (d) are mutually exclusive. N3XT allows other memory devices, given that they are thin and can be fabricated at low temperature.

improved connectivity and shorter distance to memory. However, N3XT can also accommodate NUMA schemes (future work). Many interconnect schemes may be used (e.g., butterfly, fat tree, ring, crossbar, etc.)—in this paper, we use the meshes-of-trees (MoTs) interconnect scheme (Figure 5) as it provides lower access latency and energy, as well as higher bandwidth to memory compared to other schemes [Leighton81, Balkan09]. The MoT interconnect scheme is defined by a set of *router* and *arbiter* modules (Figure 5). Each compute element is linked to a separate binary tree of router modules, with one leaf per memory controller.

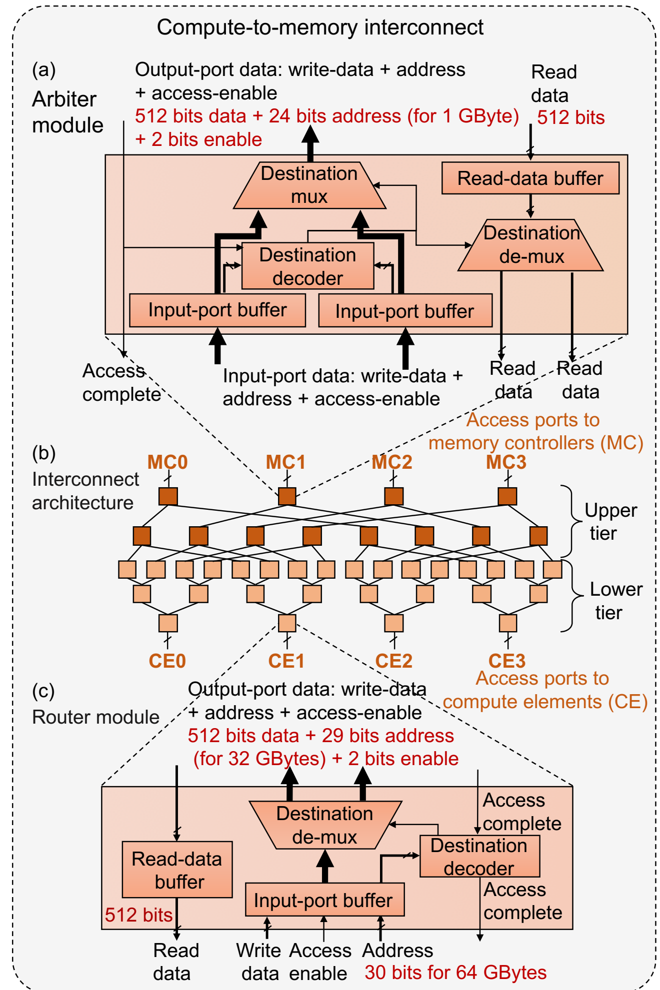


Figure 5. Architecture of compute-to-memory interconnect tiers in N3XT (Figure 1c); (a) an arbiter module that serves as a component of an aggregation tree for each memory controller; (b) a meshes-of-tree interconnect example that links four compute units with four memory controllers (the interconnect can support a larger number of compute units and memory controllers); (c) a router module as a component of a routing tree for each compute element.

The tree routes each memory access to an appropriate leaf. Memory requests are then multiplexed to the targeted memory controller through another binary tree of arbiter modules, with one tree per memory controller.

Cooling tiers may include advanced cooling solutions such as phase-change materials [Fuensanta13] and 2D materials [Pop12] that laterally spread the heat to the edge of the chip, alleviate thermal hotspots and decrease the overall operating temperature. Such advanced cooling solutions are not necessary in this paper thanks to the simplifying assumptions made above—all compute elements are in one tier (bottommost, see Figure 1e) closest to the heat sink.

III. N3XT Technology Foundations

Our N3XT architecture is enabled by device technologies that can be integrated using monolithic 3D integration through low-temperature fabrication. In particular, we focus on (*i*) carbon nanotube FETs for logic and (*ii*) STTRAM and RRAM for memory. Various hardware prototypes have demonstrated monolithic 3D integration of both CNFET and RRAM, even on a SiFET bottom tier (e.g., [Shulaker17, Wu18]). N3XT is independent of these specific device technologies, since it can also use other logic and memory devices fabricated at low temperature, as well as other possible integration technologies for ultra-dense vertical connectivity. Examples of such devices and integration include FETs using 2D materials such as MoS_2 [Wachter17], monolithic 3D with thin-film devices [Naito10], die-to-wafer bonding [Pal18], CBRAM [Wong15], and CoolCube™ [Batude15]—although these alternative technologies would offer different energy and execution time benefits (e.g., [Lee16] for FETs using 2D materials).

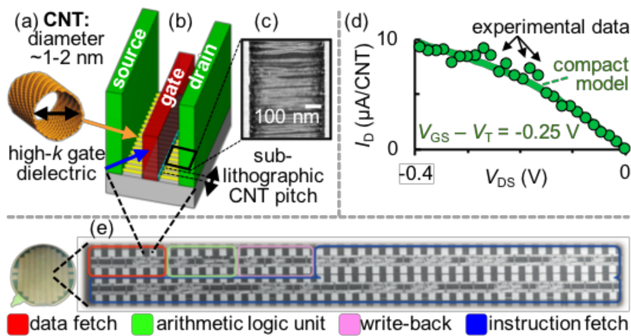


Figure 6. (a) CNT; (b) CNFET; (c) Scanning electron microscope (SEM) image of multiple CNTs comprising the CNFET channel [Shulaker14a]; (d) experimentally measured CNFET drain current (I_D) vs. drain-to-source voltage (V_{DS}), including overlaid output from a SPICE-compatible CNFET compact model [Lee15a, Lee17] for design and analysis of VLSI CNFET circuits [Hills17]. (e) SEM of a microprocessor built entirely using CNFETs [Shulaker13a], which is one instance of many CNFET microprocessors fabricated across the entire wafer, experimentally demonstrating the feasibility to build large-scale CNFET-based systems. Other recent demonstrations include high-speed CNFET-based ring oscillators [Han17] and a hardware accelerator built entirely using CNFETs [Hills18b].

Energy-efficient carbon nanotube FETs (CNFETs, Figure 6) use multiple (parallel) carbon nanotubes (*CNTs*) for transistor channel. CNTs are hollow cylindrical structures of carbon atoms 1-2 nm in diameter with remarkable mechanical, thermal, and electrical properties that enable simultaneously high carrier mobility and good electrostatic control in CNFETs [Appenzeller08]. CNFETs improve the Energy-Delay Product (*EDP*) by an order of magnitude versus silicon FETs, when analyzed on a full processor scale [Hills18a] using experimentally-calibrated models [Lee15a, Lee15b]. Recent publications demonstrated CNFETs with 5 nm gate lengths [Qiu17], high-performance and complementary CNFETs with high CNT densities (10^6 on/off current ratio with >100 CNTs/μm and improved EDP versus silicon FETs [Shulaker14a, Qiu17, Yang17, Lau18]), techniques to reduce hysteresis [Park17], and negative capacitance CNFETs with 55 mV/decade subthreshold swing at 300 K [Srimani18].

In the past, demonstrations of digital systems built with CNFETs were plagued by substantial imperfections and variations inherent in CNTs, such as mis-positioned CNTs, metallic CNTs, and inter-CNT spacing variations. The *imperfection-immune paradigm (IIP)* [Zhang12, Shulaker15a, Hills17] uniquely combines advanced nanofabrication with new circuit design to overcome these obstacles, while retaining >90% of the significant CNFET EDP benefits; all design and fabrication is wafer-scale and VLSI-compatible. IIP enabled the experimental demonstrations of CNFET-based nanosystems, such as the CNT computer, 3D nanosystem, high-performance CNFET and others [Shulaker13a, Shulaker13b, Shulaker14b, Shulaker14c, Shulaker17, Han17, Wu18, Hills18b]. CNFETs can be fabricated at low temperature (<300°C) [Patil09, Wei09, Wei13], with processing steps that are scalable to arbitrary contact-to-gate pitch and sub 5-nm inter-CNT spacing [Shulaker15a]. This enables monolithic 3D integration of CNFET logic circuits (discussed later).

On-chip non-volatile memories include Spin-Transfer Torque magnetic RAM (*STTRAM* or *STT-MRAM*) and Metal-oxide resistive RAM (*RRAM*)—they offer tradeoffs in terms of latency, energy, density, retention and endurance when designing the memory hierarchy [Wong15]. Both STTRAM and RRAM can be fabricated at low temperature (<300°C, see footnote 3) [Wong15]; thus, they are suitable for monolithic 3D integration.

A magnetic tunnel junction (*MTJ*) is the key element of an STTRAM cell (and other magnetoresistive-memory cells). Each MTJ is connected to an access transistor in a 1T-1MTJ structure illustrated in Figure 7a. while STTRAM has a slower write latency and higher energy versus SRAM, it may have read latency and energy comparable to SRAM. Furthermore, STTRAM is denser than SRAM (e.g., 0.0364 μm² STTRAM cell size at 28 nm technology node versus 0.102 μm² and 0.0499 μm² SRAM cell size at 28 nm and 14 nm technology nodes, respectively [Song16, Planes12,

Jan15]). These advantages, in addition to its nonvolatility, make STTRAM promising for last-level cache memories.

Recent STTRAM demonstrations show up to 4 Gbits [Rho17], <2 ns read latency and <10 ns write latency [Chung16, Noguchi15, Saida16]), <1.2 V write voltage [Chung16, Noguchi15], <40 μA write current [Noguchi15], and 10^{16}-cycle endurance [Chung10]. STTRAM faces low data-retention and read-disturbance challenges [Naeimi13, Li10] that are aggravated with small dimensions. These challenges can be addressed through a combination of device- and architecture-level techniques with small overheads—e.g., the MTJ design can be tuned to increase retention time, and error correction codes *(ECC)* can overcome potential read-disturbance errors [Naeimi13].

An RRAM cell may be designed using a metal-insulator-metal stack [Wong12], where data stored is detected via a change in resistance. To store data in an RRAM cell, a *set* operation stores '1' in a cell by setting the RRAM to a low-resistance state (*on resistance*), whereas a *reset* operation stores '0' by setting the RRAM to a high-resistance state (*off resistance*). An RRAM cell structure may include an access transistor (1T-1R structure in Figure 7b), among other cell structures. RRAM can offer gigabits of energy-efficient, low-latency, high-bandwidth, and non-volatile high-density on-chip data storage. RRAM can have 6-F^2 (and potentially 4-F^2)[4] cell size[5] [Fackenthal14, Wong12] and naturally enables bit-cost-scalable vertical 3D RRAM [Li16, Yu16, Lou17] (Figure 7c), where a single transistor controls access to multiple vertically-placed RRAM cells. 3D RRAM brings greater storage density versus conventional 1T1R RRAM—up to 11× projected density increase [Yu13] with down to 12-F^2 3D RRAM cell size [Jiang18].

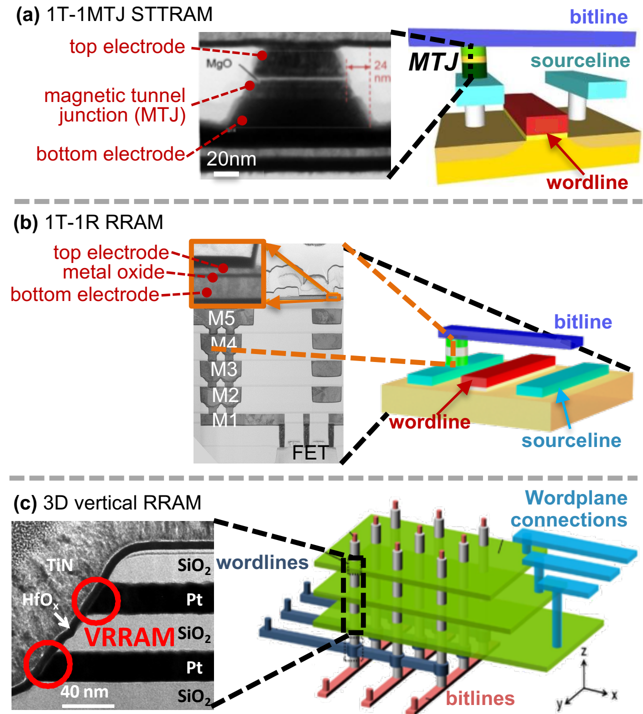


Figure 7. Examples of non-volatile memories for N3XT; (a) an STTRAM cell with a cross-sectional transmission electron micrograph (TEM) of an STTRAM MTJ [Ohashi17] on the left and 1T-1MTJ STTRAM cell structure on the right (highlighting bitline, sourceline and wordline connections to the access transistor); (b) a conventional RRAM cell with a TEM of an RRAM cell integrated above Si CMOS [Provine14] and 1T-1R RRAM cell structure (bitline, sourceline and wordline connections to the access transistor are illustrated); and (c) bit-cost scalable 3D RRAM with a cross-section TEM of multiple vertical RRAM *(VRRAM)* cells that share a wordline [Yu13] and 3D RRAM structure highlighting bitlines, wordlines and wordplane connections (used to select the vertical plane where read/write to corresponding RRAM cells occur) to the access transistors [Chen12].

RRAM demonstrations show up to 16 Gbits [Fackenthal14], <2 V write voltage, 10-100 on-to-off resistances ratio, ≤50 μA write current, 3.6ns read latency (at the RRAM array level) and ~10 ns write latency with a ten-year retention [Chang12, Ho16, Kim11, Kawahara13a, Kim11]. An RRAM cell can store a single bit or multiple bits [Sheu11, Chang15], where recent demonstrations show multiple 4-kbit RRAM arrays with three bits per RRAM cell [Le18]. 3D RRAM cells have uniform access latency and energy across vertically-stacked cells [Li16, Yu16].

However, RRAM has limited *endurance* (number of set-reset cycles before a permanent cell failure [Chen12b])—while the endurance of 10^{12} write cycles has been demonstrated at the cell level [Kim11], but only 10^5-10^7 cycles at the array level [Calderoni14, Grossi16, Chen17]. The use of RRAM as on-chip memory means that system-level resiliency mechanisms are crucial to overcome this limitation (Section VIII).

[4] F is the feature length—half the contact-to-gate pitch in this paper.

[5] 6-F^2 cell size is enabled by small-width transistor designs with 2F pitch (vs. >3F pitch of standard-logic transistors).

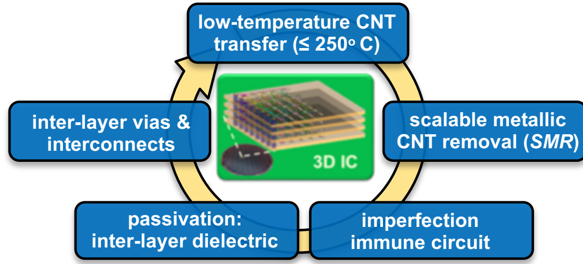


Figure 8. Example monolithic 3D fabrication flowchart [Wei09, Wei13] leveraging scalable metallic CNT removal [Shulaker15a] to create multiple layers of imperfection-immune CNFETs circuits [Zhang12]; additional technologies for both logic and memory can also be incorporated.

Fine-grained monolithic 3D integration can be achieved by sequentially placing multiple logic and memory tiers on top of each other and connecting these tiers using nanoscale ILVs (Section II). Conventional silicon transistors cannot be placed in upper tiers in monolithic 3D integration, as they require >1000°C fabrication temperature (e.g., for dopant activation annealing), which can severely damage fabricated circuits and metal interconnects on lower layers [Wong07]. CNFETs, STTRAM and RRAM naturally enable monolithic 3D integration, since they can be fabricated at <300°C and the devices have atomic-scale thickness [Patil09, Wong12]. The key in achieving this low-temperature fabrication, particularly for CNFETs, is a wafer-scale layer transfer process that decouples high-temperature CNT growth from the fabricated monolithic 3D layers [Shulaker14a]. Figure 8 illustrates this process, which allows the fabrication of monolithic 3D nanosystems that interleave logic and memory tiers in arbitrary order.

Fabricated hardware prototypes of large-scale nanosystems and their components, demonstrating the feasibility of N3XT, include:

- a four-layer 3D nanosystem [Shulaker17] (Figure 9a) that demonstrates computation immersed in memory by integrating more than two million CNFETs with 1-Mbit RRAM and more than one million silicon FETs (*SiFETs*)—CNFET-based sensors are placed on the top tier, where captured data are stored in RRAM and then processed on an on-chip hardware accelerator designed using CNFETs, and
- a brain-inspired monolithic 3D nanosystem [Wu18] exploiting CNFETs and RRAM to perform cognitive tasks such as language recognition (Figure 9d).

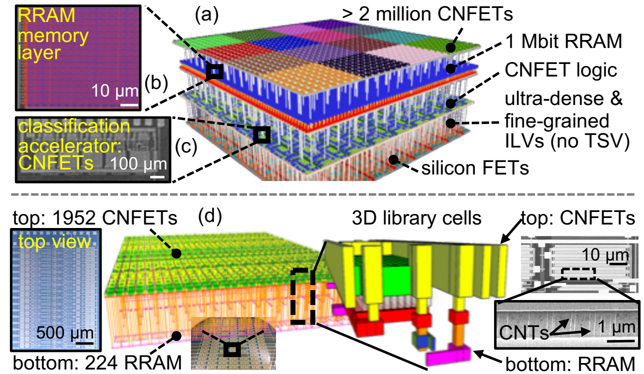


Figure 9. Experimental demonstrations of monolithic 3D ICs: (a) a monolithic 3D nanosystem [Shulaker17] that demonstrates computation immersed in memory by integrating more than one million CNFET-based sensors, on-chip 1 Mbit RRAM (b), and an on-chip CNFET-based hardware accelerator (c) to analyze data captured from the sensors and produce "highly processed" information; (d) a brain-inspired monolithic 3D nanosystem integrating 1,952 CNFETs and 224 RRAM cells, capable of performing cognitive tasks such as language recognition [Wu18].

IV. Architecture Evaluation Methodology

We have customized design and simulation tools from device to architecture levels and assembled them into an exploration framework (Figure 10). Using this framework with experimentally-calibrated device, circuit and architecture models, we quantify application-level execution time and energy consumption of various architectures (including N3XT) for workloads with large datasets. The framework can analyze general-purpose processors and domain-specific accelerators for a range of logic, memory and integration technology options.

Circuit modeling using CNFETs is performed using a custom-made CNFET process-design kit (*PDK*) compatible with standard synthesis and place-and-route tools, while accounting for variations in CNT density or CNT diameter [Zhang12, Hills15]. We start with a SPICE-compatible virtual-source CNFET model [Lee15] that has been calibrated with experimental data with gate lengths down to 9 nm. This model also accounts for many non-idealities such as direct source-to-drain tunneling leakage current, parasitic capacitance, and parasitic CNT-metal contact resistance. The variation-aware nanosystem design kit (*NDK*) [Hills17] generates CNFET PDKs by extracting the standard-cell

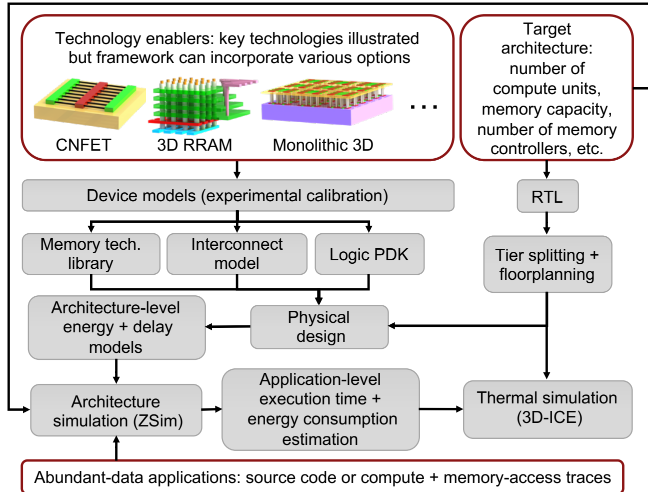


Figure 10. Our framework for evaluating the N3XT architecture.

layouts from an SiFET PDK and using the CNFET model to replace each SiFET in the layouts with a CNFET of the same gate pitch and width to maintain the same footprint. Parasitic resistances and capacitances for logic gates and interconnects are then extracted using a commercial 3D field solver (Mentor Graphics Calibre xACT 3D [Calibre10]) and combined with the model and layouts to characterize power and timing (using Cadence Spectre and Cadence Liberate [Spectre17, Liberate17]). In this paper, we use a 28 nm node foundry SiFET PDK to generate the corresponding CNFET PDK, but we have examined other nodes all the way to sub-10 nm [Hills18a]. To model general-purpose processors, we synthesize processor cores in the publicly-available OpenSPARC T2 system on chip [SPARC15] and perform physical design. To model domain-specific deep learning accelerators, we synthesized and performed the physical design of a 2D mesh of 4,096 16-bit multiply-and-accumulate units, [Gao17, Chen16].

Implementing each category—general-purpose processor cores or domain-specific accelerators—using CNFETs leads to 3× faster clock speed and 3× less energy, simultaneously, compared to the corresponding SiFET-based implementation across several technology nodes.

Modeling the STTRAM and RRAM access latencies and energies at the memory-array level starts by extracting the following cell-level parameters from experimental data reported in the literature: on- and off-resistances, read/write currents and voltages, cell area and write latency [Li16, Noguchi15]. A CNFET is used as a selecting device, and its corresponding dimensions that satisfy read/write currents of either RRAM or STTRAM are identified via SPICE simulations of a single memory cell (Figure 4c illustrates 3D RRAM, and Figure 4d illustrates 1T-1MTJ STTRAM). Digital blocks for memory-access circuits (including address, row, and column decoders, as shown in Figure 4), are designed using the CNFET PDK discussed earlier in this section. The analog components (including column multiplexers, sense amplifiers, pulse generators, write drivers and word-line drivers) are simulated in SPICE, using the SPICE-compatible CNFET model [Lee15], while accounting for the bitline (wordplane connections as well, for 3D RRAM) capacitive and resistive load. The resistances and capacitances of bitline (and wordplane connections for 3D RRAM) are based on metallic-interconnect material parameters supplied with a foundry PDK. We consider sense amplifiers with a current sensing scheme akin to demonstrated RRAM and STTRAM prototypes [Sheu09, Kim15, Chang12, Chang13].

During read operation, bitlines (wordplane connections for 3D RRAM) are precharged to 0.5V [Chang12, Noguchi15] and the sense amplifier compares read currents with a reference value selected to distinguish a stored zero versus one. The reference current is determined by incorporating read-current distributions for both on- and off-resistances, accounting for cell-to-cell variations reported in the literature [Li15, Li16]. During write operation, bitlines (wordplane connections for 3D RRAM) are charged to the write voltage, e.g., 1.2 V for RRAM [Li16] or 0.9 V for STTRAM [Noguchi15]. A write-verify scheme immediately reads and compares the stored data with the actual data to ensure a successful write operation [Sheu11].

Multiple memory arrays placed in the same tier are linked using an H-tree layout of a binary-tree interconnect [Leiserson80] to form a *memory segment*, as shown in Figure 4a. We use NVSim [Dong12] to estimate access latency and energy for a given memory segment and reduce its access latency and energy by tuning the number and size of memory arrays. Multiple memory segments are connected to a memory controller to handle memory accesses. The memory controller (Figure 3) is designed using the CNFET PDK discussed earlier in Section IV. A memory controller supports multiple channels, each corresponding to a memory segment. The number of memory segments is selected to increase concurrency and allow for one memory access per clock cycle per memory controller (in N3XT, memory controllers and compute units operate at the same frequency, as discussed in Sections V, VI and Appendix B).

Interconnect modeling with monolithic 3D integration is performed using our own monolithic 3D physical design methodology that leverages commercial physical design tools for conventional ICs (that are not monolithic 3D). Starting from an RTL description of digital logic circuits in Figures 1b, 1c and 1e, we synthesize a gate-level netlist (e.g., using CNFET PDK), while memory segments (Figures 1a and 1d) are represented as black boxes during floorplanning and place-and-route. The netlist is then manually partitioned into multiple tiers—compute, memory and compute-to-memory interconnect tiers—based on the selected architecture (Figure 1). Each tier undergoes separate timing-driven place-and-route using the Synopsys IC Compiler™ [Synopsys17],

accounting for the load capacitances and drive resistances of ILVs that connect inputs and outputs of individual tiers.[6] Afterwards, we perform timing and power analysis on the entire design represented by a merged netlist.

The approach described above can incorporate other technology PDKs for monolithic 3D designs in addition to CNFETs. The use of existing commercial physical design tools allows us to perform further analysis to account for process corners, voltage noise, signal integrity, and IR voltage drops across wires in the power grid. In this paper we manually partition the architecture across multiple tiers. Automated partitioning can be leveraged to explore different architecture configurations [Panth14].

Architectural simulation with ZSim [Sanchez13] quantifies the application-level execution time and energy consumption of N3XT and competing architectures with general-purpose processors. To model domain-specific accelerators, we combine ZSim with TETRIS [Gao17], an application-level trace-generation framework. We configure each architecture for simulations by specifying the following parameters:

- number of compute units as well as their type (general-purpose or accelerator cores), operating frequency, active, and idle energy,
- capacity, access latency and energy of *local* (one per compute unit) and *shared* (among all compute units) memories such as caches and scratchpad memories (see Figures 11 and 13),
- compute-to-memory interconnect latency and energy per access (to account for UMA), and
- capacity, access latency and energy of main memory (last-level of the random-access-memory hierarchy, e.g., DRAM in current baseline computing architectures and 3D RRAM in N3XT), number of access channels to memory, and bandwidth per channel.

Examples of such configurations can be found in Sections V and VI, as well as Appendices B and E. Simulated workloads use standard compiler and software frameworks, such as `gcc` and `TensorFlow` [Abadi16]. The latency and energy models of the architecture components are derived from the circuit-, memory- and interconnect-modeling methodologies previously discussed in this section. For architectures with DRAM, we model a DDR interface, where timing and energy information (supply voltage, standby, read and write currents, row-access delay, column-activation delay, refresh time) are obtained from fabricated chip datasheets [DDR17] to estimate DRAM access latency, energy and bandwidth. In N3XT, memory-access requests are buffered into the address queue of a corresponding memory controller (alongside buffering incoming data into the write-data queue). These requests are then serviced in-order with a first-come-first-serve (*FCFS*) scheme (Figure 3), where we use a queue model to simulate such requests in N3XT. For a given application, simulation produces the following outputs:

- number of operations executed per compute unit,
- number of cycles consumed in memory accesses,
- number of memory accesses (read and write) to local global, and main memory (Figures 11 and 13),
- energy consumed in each compute unit when executing instructions and also when idle (due to memory access),
- energy consumed in local, global, and main memories,
- execution time and the corresponding breakdown into compute and memory-access times, and
- number of reads and writes to each memory word in main memory.

We assess simulation accuracy by comparing the outputs of our framework against measurements of actual general-purpose hardware via Intel performance-counter monitor [PCM17]. These measurements include:

- number of executed instructions and instructions-per-cycle for each processor core,
- cache miss rates,
- number of reads and writes to DRAM,
- application execution time,
- total processor energy consumption, and
- total DRAM energy consumption.

We have benchmarked our simulation outputs, when running the workloads described later in Section V, against measurements of a state-of-the-art 16-core-processor single-chip architecture with 40-MByte cache (at 22 nm technology node) connected to a 128-GByte DRAM with DDR4 interface. The processor operating frequency and cache latencies are obtained from datasheets [Intel15]. To estimate the energy at 22nm for baseline machines, we have used the measured *energy-per-instruction* (EPI) of the same hardware reported in [Shao13]. Simulation results show an error within 15% (i.e., 85% simulation accuracy) for the total execution time, compute energy, and memory energy consumptions, when compared with measured values.[7]

Thermal simulation uses 3D-ICE [Sridhar14] to estimate operating temperatures for N3XT and other architecture configurations. 3D-ICE can model heat dissipation in 2D and 3D ICs. 3D-ICE has been validated against actual thermal measurements of an existing IC and a 3D stacked prototype with 93.2% and 90% simulation accuracy, respectively [Iranfar17, Sridhar14]. Thermal conductivity and specific heat capacitance of active and metal layers are needed for 3D-ICE. We account for the impact of ILV locations on thermal conductivity and capacitance in all tiers to increase the

[6]Each ILV is assigned a drive resistance (i.e., output resistance of the driving gate plus the ILV resistance) and a load capacitance (i.e., input capacitance of the driven gate and the ILV parasitic capacitance).

[7]Our current simulation only counts the number of operations. Data-sensitive simulation can be more accurate because e.g., arithmetic operations with zero values consume less energy than with non-zero values.

thermal-simulation accuracy of N3XT [Wei12]. First, effective anisotropic thermal conductivities are extracted for the transistor layers and metal layers based on the area coverage of the transistors and interconnect density and direction [Wei12]. Then, these thermal conductivities are used within 3D-ICE to model the monolithic 3D chip for thermal simulation—we summarize these conductivities in Appendix D. To estimate peak and average temperatures of each component and the entire architecture, a floorplan of each tier is obtained, annotated with block power consumption, and thermal simulation is performed using the extracted effective anisotropic thermal conductivities.

V. Evaluating CPU-based Systems

We compare N3XT with a 2D baseline system where a compute chip—containing processor units and SRAM-based caches (local L1 cache and shared L2 cache)—is connected to off-chip DRAM main memory via a DDR interface.

Architectures with 2 to 64 cores are examined for N3XT and 2D baseline, to represent mobile to server computing platforms. We select the capacities of shared caches (1 MByte per core) and main memories (1 GByte per core) according to current architecture trends (see Appendix A). In addition, we consider an architecture with 64 cores, 64-MByte cache and main memory of 128 GBytes for workloads with very large datasets (details below). Figure 11 illustrates the 64-core configuration with 64-MByte shared cache and 64-GByte memory, while Table 1 summarizes the latency and energy of each architecture component. For other architecture and modeling considerations, see Appendix B.

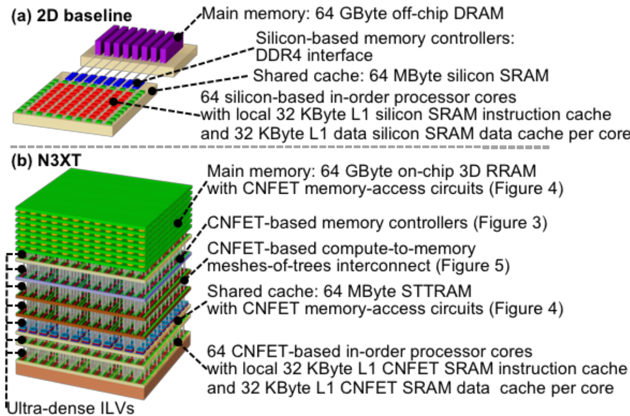


Figure 11. Architecture configuration for (a) 2D baseline and (b) N3XT 64-core CPU-based systems. Energy and latency for each module are reported in Table 1. Other configurations are summarized in Appendix B.

Application workloads that we simulate are implemented within application frameworks known for their high performance on conventional (2D) systems:

Application 1: graph analytics,
workloads: `PageRank`, `breadth-first search`, `single-source shortest path`, and `connected components`,
framework: `PowerGraph` and `Galois` [Gonzalez12, Pingali11, Satish14, Aberger16].

Table 1. Configuration parameters of the analyzed 64-core CPU-based systems (Figure 11) derived using our developed framework in Section IV or obtained from hardware measurements (for the 2D baseline). Memory-access latency and energy values shown are for the entire memory subsystem (i.e., array and interconnects). Other configurations and additional modeling considerations are summarized in Appendix B (FCFS: first-come-first-serve, FR-FCFS: first-ready FCFS).

	2D baseline	N3XT
Main memory	64 GByte off-chip DRAM 8 memory controllers, 1 64-bit channel per controller with DDR4 interface (1.2GHz), FR-FCFS scheduling 65/60 ns read/write 45 pJ/bit read/write	64 GByte on-chip 3D RRAM with CNFET access circuits 64 memory controllers (4GHz), 16 512-bits channels per controller with simple interface, FCFS scheduling, meshes-of-trees interconnect 5 ns read, 13 ns write 0.8/1.1 pJ/bit read/write
L2 shared cache (8-way set associative)	64 MByte silicon SRAM 14 ns read/write 1 pJ/bit read/write	64 MByte STTRAM (CNFET access circuits) 2/4 ns read/write 0.2/0.6 pJ/bit read/write
L1 local data cache (8-way set associative)	32 KByte silicon SRAM per processor core 2.3 ns read/write 0.2 pJ/bit read/write	32 KByte CNFET SRAM per processor core 0.8 ns read/write 0.14 pJ/bit read/write
L1 local instruction cache (4-way set associative)	32 KByte silicon SRAM per processor core 1.5 ns read/write 0.17 pJ/bit read/write	32 KByte CNFET SRAM per processor core 0.5 ns read/write 0.11 pJ/bit read/write
Processor cores	64 in-order processor cores 1.3 GHz clock speed 0.5 nJ/instruction [Shao13]	64 in-order processor cores 4 GHz clock speed 0.16 nJ/instruction
Technology	22 nm silicon CMOS	22 nm CNFET CMOS

Application 2: conventional machine learning,
workloads: `linear regression (`*`LinR`*`)`, `logistic regression (`*`LR`*`)`, and `support vector machines (`*`SVM`*`)`,
framework: `DimmWitted` [Zhang14] used for both training and inference.

Application 3: deep learning,
workloads: `VGGNet-19` [Simonyan15] and `Inception-v4` [Szegedy16] (convolutional neural networks, *CNNs*) and `Neural Programmer` [Neelakantan17] and `Language Model` [Jozefowicz16] (long short-term memory, *LSTM*),
framework: `Tensorflow` [Abadi16] used for training and inference of open-source implementations of the examined workloads [Zoo17].

Common datasets that we use are summarized in Table 2 [Leskovec14, Boldi14, HBP16, CCAFS16, Lichman13, Webscope17]—they span 1.2 to 110 GBytes of RAM when loaded.

Simulation results include application-level runtime and energy improvements, as well as their product (referred in the remaining of this paper as *system-level EDP benefits*) of N3XT compared to 2D baseline. We report in Table 3 such benefits for the largest datasets in all considered workloads,

which are observed for the 64-core configuration. N3XT achieves a maximum of 1,152× system-level EDP benefits.

Table 3 also reports the mean of the top 50% of the EDP benefits (sorted in descending order for each workload/dataset combination) for each workload.

Table 2. Application workloads and the datasets used in simulations of general-purpose processors. For each dataset, the corresponding properties and memory requirements are shown (K = thousand, M= million, B = billion). Memory usage is characterized with `memusage` UNIX command.

Workloads	Models or datasets	Properties/Memory usage
Graph analytics		
PageRank Breadth-first search Single source shortest path Connected components	Google + [Leskovec14]	107 K vertices, 13 M edges/ 1.2 GBytes
	Patent [Leskovec14]	3.7 M vertices, 16.5 M edges/ 2 GBytes
	Pokec [Leskovec14]	1.6 M vertices, 60.3 M edges/ 3 GBytes
	LiveJournal [Leskovec14]	4.9 M vertices, 69 M edges/ 4 GBytes
	Orkut [Leskovec14]	3 M vertices, 117 M edges/ 7 GBytes
	EU-2015 hosts [Boldi14]	11 M vertices, 387 M edges/ 19 GBytes
	UK 2005 [Boldi14]	39 M vertices, 936 M edges/ 44 GBytes
	IT 2004 [Boldi14]	41 M vertices, 1.15 B edges/ 54 GByte
	Twitter [Boldi14]	41 M vertices, 1.6 B edges/ 75 GBytes
	Friendster [Leskovec14]	66 M vertices, 1.8 B edges/ 110 GBytes
Conventional machine learning		
Linear regression (LinR) Logistic regression (LR) Support vector machines (SVM)	Reuters RCV1 [Lichman13]	47 K features, 677 K samples/ 1.5 GBytes
	URL Reputation [Lichman13]	3 M features, 2.4 M samples/ 3.7 GBytes
	HIGGS [Lichman13]	29 features, 11 M samples/ 7.4 GBytes
	Yahoo Webscope [Webscope17]	136 K features, 1.8 M samples/ 14 GBytes
	Weather [CCAFS16]	43 K features, 201 K samples/ 49 GBytes
	Human microbiome project (HBP) [HBP16]	1 M features, 25.2 M samples/ 56 GBytes
Deep learning		
Convolutional neural networks (CNNs)	VGGNet19 [Simonyan15] ImageNet flower dataset [INet17]	155 M weights /1.2 GBytes inference, 20 GBytes training
	Inception-V4 [Szegedy16] ImageNet flower dataset [INet17]	25 M weights /1 GByte inference,16 GBytes training
	Neural Programmer	2 M weights /1 GByte inference, 16 GBytes training
Long short-term memory (LSTM)	[Neelakantan17] 1Billion words news dataset [SMT11]	
	Language Model [Jozefowicz16] 1Billion words news dataset [SMT11]	1.04 B weights /10 GBytes inference, 40 GBytes training

Table 3. System-level EDP benefits of N3XT versus 2D baseline for the largest datasets, as well as the mean of the top 50% EDP benefits (sorted in descending order for each workload/dataset combination). Results for all configurations are shown in Appendix C.

Workload	Benefits for largest datasets			Mean of top 50% of benefits		
	Run time	Energy	System-level EDP	Run time	Energy	System-level EDP
Graph analytics						
PageRank	22×	39×	858×	15×	26×	390×
Shortest path	24×	48×	1,152×	18.7×	35.6×	666×
Connected components	18×	36×	648×	16.2×	31×	502×
Breadth-first search	10×	18×	180×	5×	8.4×	42×
Conventional machine learning						
LinR	19×	32×	608×	9.1×	18.2×	166×
LR	20×	31×	620×	9.3×	18.3×	170×
SVM	18×	37×	666×	8.8×	18×	158×
Deep learning						
CNNs	10.5×	32×	336×	7.4×	18.6×	138×
LSTM	17×	40×	680×	13.6×	24×	326×

The average system-level EDP benefits for N3XT are 43×, 54×, 63×, 150×, 302×, and 540× for the 2-, 4-, 8-, 16-, 32-, and 64-core configurations, respectively (all results are shown in Appendix C).

N3XT achieves those benefits while simultaneously maintaining the same operating temperature of 2D baseline (discussed later in this section). We observe these improvements with generic implementations of the targeted workloads—optimizing the workloads for N3XT may yield even greater benefits.

Workloads with greater main-memory access rates (number of memory requests per unit time) leave room for

greater improvements. Figure 12 illustrates system-level EDP benefits of all examined workloads and datasets with respect to L2-cache misses per 1K instructions (MPKI, a ratio between compute operations and memory accesses). Intuitively, architectures with a greater number of compute units exert higher memory-access rates (Figure 12b), and hence enjoy greater benefits with N3XT.

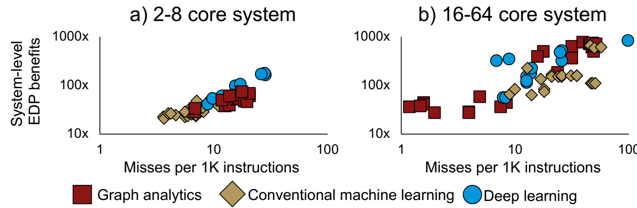


Figure 12. System-level EDP benefits in N3XT for all examined workloads with respect to the observed misses per kilo instructions for (a) 2- to 8-core and (b) 16- to 64-core system configurations. Workloads are grouped by application—graph analytics, conventional machine learning and deep learning.

Breakdown of runtime and energy consumption of both 2D baseline and N3XT systems for a selected set of workloads is shown in Table 4 (see Appendix C for full results). The workloads are selected to showcase the greatest improvements in each class: (*i*) 2-, 4-, and 8-core configurations (embedded and mobile platforms) and (*ii*) 16-, 32-, and 64-core configurations (server platforms). In particular, we consider `PageRank` for graph analytics (`Pokec` dataset for 4 cores and `Twitter` dataset for 64 cores), `SVM` for machine learning (training with `URL` dataset for 4 cores and training with `HBP` dataset for 64 cores) and `LSTMs` for deep learning (inference with `Neural Programmer` for 2 cores and training with `Language Model` for 64 cores).

The 2D baseline system is bottlenecked in memory. Table 4 shows that a significant portion of both application execution time (60-95%) and energy consumption (76-97%) of 2D baseline systems are consumed in accessing DRAM due to the following. First, high MPKI values imply an increase in the access rate to off-chip DRAM, particularly with 16 to 64 cores (Figure 12b). Second, designs with pin-limited DRAM suffer lower memory-access rates. Improving compute units in the 2D baseline configuration—e.g., running at higher frequency, using STTRAM-based L2 cache memories, increasing the number of cores, or even using CNFETs—does not overcome this limitation. A 2D baseline system with STTRAM-based cache and CNFET-based compute units experiences a maximum of 1.2× and 1.1× EDP improvements, respectively, versus the configuration in Figure 11a. N3XT overcomes this bottleneck, causing compute operations to dominate execution time and energy, through the ultra-wide and massively-concurrent connectivity to memory provided by monolithic 3D.

3D TSV and 2.5D integration are evaluated using architectures that include an SiFET-based compute tier connected to DRAM representing high-bandwidth memory (HBM) [HBM17, Ramalingam16] with:

- interposer-based 2.5D integration with 40 µm microbump pitch, or
- 3D stacking with TSVs using a 5µm pitch assuming no constraints on TSVs placement.

We summarize the configurations for such systems in Appendix E. System-level EDP benefits reach 2.2× for 2.5D and 6× for 3D TSV. Both integration techniques improve the number of compute-to-memory connections by 4× for 2.5D and 16× for 3D TSV, whereas N3XT improves such connections by >1,000× compared to TSVs.

Thermal analysis of 2D baseline and N3XT shows that the average power density is 61 W/cm^2 for 2D baseline and 65 W/cm^2 for N3XT. We summarize thermal simulation parameters in Appendix D. In both architectures, processing cores abut the heatsink. They lead to 61°C and 63°C peak temperatures for 2D baseline and N3XT, respectively, while retaining the benefits in Tables 3 and 4. The operating temperature remains relatively low even with the high

Table 4. Breakdown of the total execution time and energy consumption for CPU-based 2D baseline and N3XT systems. Representations of the examined three application domains are shown for the 2-, 4-, and 64-core architectures. inf=inference

Execution time

	Graph analytics (PageRank)				Conventional machine learning (SVM, training)				Deep learning (LSTM)			
	4 cores (Pokec)		64 cores (Twitter)		4 cores (URL)		64 cores (HBP)		2 cores (NP, inf.)		64 cores (LM, train)	
	2D baseline	N3XT	2D baseline	N3XT	2D baseline	N3XT	2D baseline	N3XT	2D baseline	N3XT	2D baseline	N3XT
Processor active	42.5%	13.8%	4.9%	2.4%	32.5%	10.5%	4.5%	2.3%	22%	11%	7.3%	3.7%
Memory access	57.5%	2.63%	95.1%	2.1%	67.5%	3.3%	95.5%	3.3%	78%	3.5%	92.7%	2.1%
N3XT improvement		6x		22x		7.5x		18x		7x		17x

Energy consumption

	Graph analytics (PageRank)				Conventional machine learning (SVM, training)				Deep learning (LSTM)			
	4 cores (Pokec)		64 cores (Twitter)		4 cores (URL)		64 cores (HBP)		2 cores (NP, inf.)		64 cores (LM, train)	
	2D baseline	N3XT	2D baseline	N3XT	2D baseline	N3XT	2D baseline	N3XT	2D baseline	N3XT	2D baseline	N3XT
Processor active	24%	7.4%	8.6%	1.7%	24%	7.2%	8.8%	1.5%	20%	6.2%	2.7%	0.59%
Processor idle	13.7%	0.3%	43.2%	0.2%	20%	0.9%	43.5%	0.7%	23%	0.5%	66.5%	1.7%
Memory access	62.3%	1.6%	48.2%	0.7%	56%	3.0%	47.7%	0.5%	57%	2%	30.8%	0.19%
N3XT improvement		10.8x		39x		9x		37x		11.5x		40x

operating frequency of upper-tier memory controllers (Figure 1b) in N3XT because these memory controllers consume low amounts of energy (~0.01pJ/bit per controller, owing to the simple interface illustrated in Figure 3) with power densities <7 W/cm^2 (<10% of total power density in N3XT). Greater benefits can be achieved in N3XT—1,105× for `PageRank`, 750× for `SVM`, and 973× for `LSTM`—but at the cost of increased power densities and higher operating temperatures making advanced thermal solutions essential.

PARSEC benchmark[8] workloads [Bienia08] as well as `FFT` [FFTW15] are evaluated, as they represent common conventional multicore workloads. Table 5 reports system-level EDP benefits for N3XT (see Appendix C for the breakdown of benefits). These workloads enjoy cache miss rates below 2% due to high data locality, which makes them amenable to existing architectures, thus having a lower main memory access rate compared to abundant-data workloads. For workloads that are compute-bound, N3XT achieves 13-16× system-level EDP improvements, owing to the benefits of CNFETs in compute tiers. `FFT, Fluid-animate` and `Canneal` are more memory-bound and, thus, experience greater improvements in N3XT. When 2.5D and 3D TSV are analyzed with these workloads, 1.1× and up to 1.9× system-level EDP benefits are observed—compared to the 2D baseline architecture (Figure 11a)—for compute- and memory-bound workloads, respectively.

Table 5. System-level EDP benefits when simulating PARSEC and FFT multicore workloads on both 2D baseline and N3XT (see the breakdown in Appendix C).

Workload	Benefits	Workload	Benefits
Black-Scholes	13×	Body track	16×
Canneal	36×	Dedup	34×
Ferret	12×	Fluid animate	54×
Ray trace	20×	Stream cluster	35×
Swaptions	65×	x264	22×
FFT	150×		

VI. EVALUATING SYSTEMS WITH DOMAIN-SPECIFIC ACCELERATORS

We quantify the benefits of N3XT for embedded domain-specific accelerators, namely inference using deep neural networks (*DNN*). N3XT can achieve a 1,000× improvement in energy efficiency for a reduced-precision 8-bit DNN inference accelerator [Hwang17]. In this paper, we analyze a state-of-the-art 16-bit DNN inference engine [Gao17, Chen16], while we consider accelerators for the training phase as part of our future work.

To model the execution of the inference phase of DNN workloads on the hardware accelerator (details shown later), the corresponding compute operations and memory-access traces for each network are generated using the TETRIS framework [Gao17]. These traces are then simulated with ZSim using the methodology in Section IV.

Analyzed architectures are illustrated in Figure 13 and include a 2D array of 4,096 16-bit multiply-and-accumulate processing elements (PEs) each with a 256-Byte local SRAM, and a 2-MByte shared memory among all PEs. A 4-GByte main memory is used to store the DNN model, input data, and intermediate data generated between consecutive stages within a single DNN.

Table 6 reports the latency and energy of each component for both 2D baseline and N3XT estimated using the framework described in Section IV. In particular, we perform synthesis and physical design of each component using a 28 nm foundry SiFET PDK for 2D baseline and 28 nm CNFET PDK for N3XT. Other components such as the clock tree are scaled from published 65 nm data [Chen16]. We have also examined projected technology nodes down to 7 nm and have observed results similar to the ones shown below.

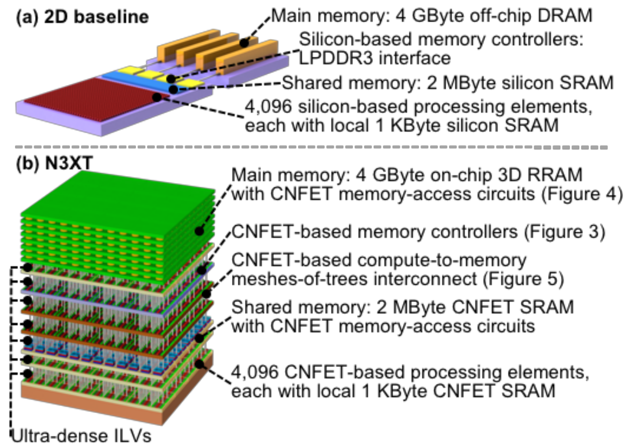


Figure 13. Architecture configuration of (a) 2D baseline and (b) N3XT DNN accelerators. Energy and latency for each module are reported in Table 6.

Table 6. Configuration parameters of the analyzed DNN accelerators (Figure 13) estimated using our developed framework in Section IV. Memory-access latency and energy values shown are for the entire memory subsystem (i.e., array and interconnects). FCFS: first-come-first-serve, FR-FCFS: first-ready FCFS

	2D baseline	N3XT
Main memory	4 GByte off-chip DRAM 2 memory controllers, 2 32-bit channel per controller with LPDDR3 interface (0.8 GHz), FR-FCFS scheduling 75 ns read, 70ns write 15 pJ/bit read/write	4 GByte on-chip 3D RRAM with CNFET access circuits 16 memory controllers, 16 256-bit channels per controller with simple interface (1.5 GHz), FCFS scheduling meshes-of-trees interconnect, 2.3 ns read, 11 ns write 0.4/1.33 pJ/bit read/write
Shared memory	2 MByte silicon SRAM	2 MByte CNFET SRAM 1.7 ns read/write

[8] Not all workloads could be analyzed using the framework (incompatible host system libraries with the workloads `facesim, vips` and `freqmine`), but the reported ones span the entire spectrum of compute and memory access behavior of the suite

	4 ns read/write 0.32 pJ/bit read/write	0.15 pJ/bit read/write
Local memory (for each processing element)	256 Byte silicon SRAM per processing element 2 ns read/write 0.23 pJ/bit read/write	256 Byte CNFET SRAM per processing element 0.7 ns read/write 0.076 pJ/bit read/write
Compute units	4,096 16-bit processing elements 0.5 GHz clock speed 1.9 pJ/operation	4,096 16-bit processing elements 1.5 GHz clock speed 0.6 pJ/operation
Technology	28 nm silicon CMOS	28 nm CNFET CMOS

Common networks for `CNNs` and `LSTMs` and used in this evaluation are summarized in Table 7. Inference inputs include images with 224×224×3 pixels [INet17] for `CNNs` and 32-character long text segments [Jozefowicz16] for `LSTMs`. We use 1- to 16-input batch sizes that are common in deep learning accelerators (in both embedded and server systems) [Han16, Jouppi17].

Table 7. Examined DNN networks and the corresponding memory capacity of each model.

Network	Type	#Parameters	Memory usage
AlexNet [Krizhevsky12]	CNN	60 Million	120 MBytes
VGGNet-19 [Simoyan15]	CNN	145 Million	290 MBytes
ResNet-152 [He16]	CNN	60 Million	120 MBytes
Image Captioning [Vinyals15]	LSTM	75 Million	150 MBytes
Language Model [Jozefowicz16]	LSTM	1.2 Billion	2.5 GBytes

Simulation results include system-level EDP benefits for all networks at different batch sizes in Figure 14, where N3XT experiences up to 1,971× and 251× average improvements over 2D baseline across all networks and batch sizes. `Language Model` is more memory-bound, hence experiencing greater benefits at all batch sizes. In contrast, `VGGNet-19` is more compute-bound due to increased data reuse. Regardless, N3XT achieves 63× improvements with `VGGNet-19`. Similar benefits are observed for all workloads mapped on 8-bit hardware accelerator for DNNs [Hwang17].

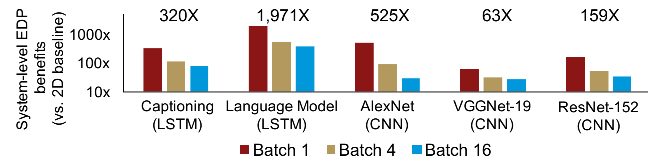


Figure 14. System-level EDP benefits of N3XT vs. 2D baseline, with 16-bit MAC units, for DNNs. Maximum benefits observed for each network are highlighted above.

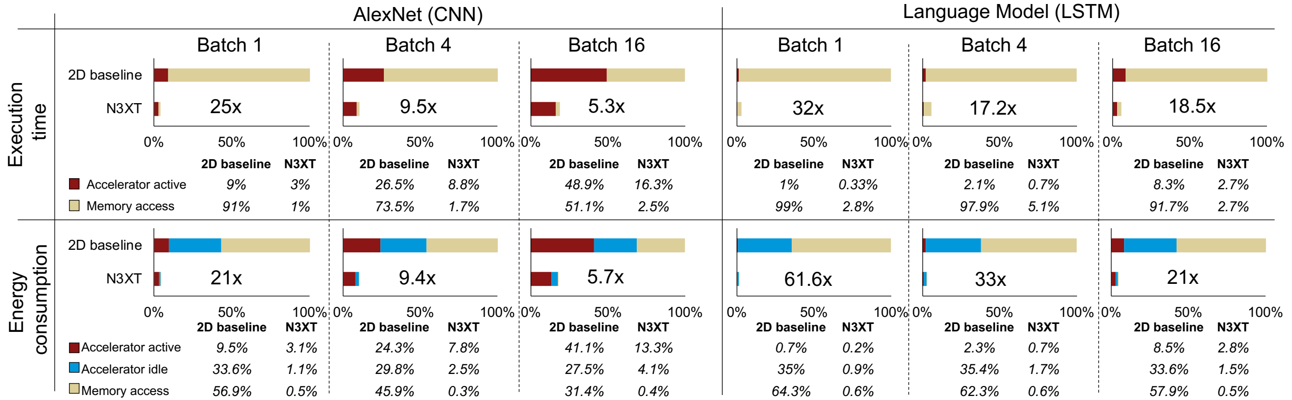
Breakdown of the application-level execution time and total energy consumption of `AlexNet` and `Language Model` is shown in Table 8 for 1- , 4-, and 16-input batch sizes. We select these networks as they provide the greatest benefits for both CNNs and LSTMs. Similar to CPU-based systems, >90% of total application runtime and energy consumption are consumed in memory accesses and idle energy of PEs.

Increasing the batch size improves data locality and hence reduces memory access time and energy—workloads become more compute-bound, as in the case of `AlexNet` with 16-input batch where N3XT achieves only 30× improvements. Nonetheless, great benefits in N3XT can be retained with large batch sizes when the network size far exceeds the capacity of the global SRAM, as in the case of `Language Model` with 388× improvements. Increasing the batch size further would reduce the benefits in N3XT due to increased weight reuse, implying a higher number of compute operations per memory read. However, benefits improve at a very large batch size (>128) due to greater amounts of intermediate data with commensurate increase in memory writes.

Simulating 3D TSV and 2.5D integration of a Si-based compute tier containing PEs, local and global memories, with HBM DRAM [HBM17] show that system-level EDP benefits reach 3× and 8× for 2.5D and 3D TSV, respectively. We summarize such systems configurations in Appendix E.

Thermal analysis of the evaluated domain-specific accelerator shows that peak temperature and average power density of N3XT is 35°C and 9.5 W/cm^2, respectively. For 2D baseline, the corresponding values are 36°C and 10.4 W/cm^2, respectively. These temperatures are comparable to typical workloads on today's mobile systems [Chiriac16].

Table 8. Breakdown of the total execution time and energy consumption for DNN-accelerator-based 2D baseline and N3XT systems. Representations for the two examined DNN applications, that provide the greatest benefits, are shown for 1-, 4-, and 16-input batch size.



VII. Technology Alternatives for Implementing N3XT

So far, we have assumed several specific technologies—CNFETs, STTRAM, RRAM and monolithic 3D integration—but N3XT can leverage a wide variety of device and integration technologies options. Therefore, we quantify how the use of such alternatives impacts system-level EDP benefits. For this analysis, we consider the domain-specific DNN accelerator illustrated in Figure 13. We report detailed results for two workloads where N3XT achieves the maximum and minimum improvements—`Language Model` with a single-input batch size and `AlexNet` with a 16-input batch size, respectively.

Impact of compute-to-memory connectivity. To illustrate this key enabler in N3XT and its impact, we vary the number of compute-to-memory connections, i.e., vertical connections between: (*i*) the compute tier and the compute-to-memory-interconnect tiers, (*ii*) the compute-to-memory-interconnect tiers and the memory-controller tier, and (*iii*) the memory-controller tier and the memory-access-circuits tier of the non-volatile 3D RRAM (all tiers illustrated in Figure 1). These parameter sweeps capture the range of possible integration techniques—from planar 2D to fine-grained monolithic 3D integration. In particular, we vary the number of channels in each memory controller (1-16 channels per controller, Figures 2 and 3), the bitwidth in each channel within the memory controller (64-512 bits per channel, Figure 3) and the bitwidth in compute-to-memory interconnect (64-512 bits per channel, Figure 5). These parameters change the number of connections to memory (shown in the x-axis in Figure 15). Additionally, we assume that: (*i*) memory-access circuits are connected to the non-volatile memory cells with ILVs (Figures 1 and 4), and (*ii*) compute units, compute-to-memory interconnect, memory controllers and memory-access circuits are designed using CNFETs for all analyzed configurations.

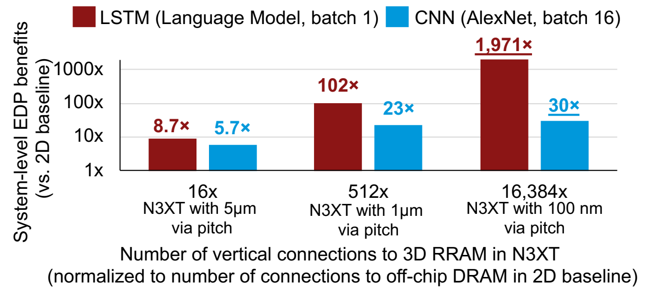


Figure 15. System-level EDP benefits (over the 2D baseline system illustrated in Figure 13a) versus the number of vertical connections to 3D RRAM in N3XT, normalized to number of connections to off-chip DRAM in 2D baseline. The benefits for the N3XT architecture discussed in Section VI are highlighted. The pitch of the vertical connections is varied, which is translated to a change in the number of channels in each memory controller (Figure 2), the bitwidth in each channel within the memory controller (Figure 3) and the bitwidth in compute-to-memory interconnect (Figure 5)—Figure 1 illustrates the locations of the corresponding tiers. Vertical connections between memory cells (3D RRAM) and memory-access circuits use ILVs (100 nm pitch for 28nm technology node). Results are shown for workloads that achieve maximum and minimum improvements in N3XT.

Increasing the number of connections ensures wider and more concurrent accesses to memory, which in turn improves memory bandwidth and reduces idle energy in compute units—a significant portion of total energy (Sections V and VI). However, a large number of connections is required to achieve significant improvements.

Simulation results in Figure 15 show that 3D integration with 5-µm-pitch vias (similar to a TSV pitch) and CNFET-based logic in all corresponding tiers offers only 8.7× system-level EDP benefits for memory-bound workloads—benefits reach 7× if SiFETs are used in the bottom compute tier and may drop to 4.4× if SiFETs are used in all corresponding logic tiers (assuming for the sake of an argument that SiFET might be fabricated in the upper tiers with the same characteristics as those in the bottom tier). Vias with such large pitches would require significant area overhead to increase the number of vertical connections. Moreover, the improvements

observed with this pitch, which is similar to that of TSVs, can be optimistic since TSV placement tends to be more constrained in practice.

In N3XT, ILVs with a 100-nm pitch—the via pitch in 28 nm technology node [Tomimatsu09]—increase the benefits by 1,971× with CNFETs compared to the 2D baseline system in Figure 13a. With further downscaling of device dimensions, ILVs have a finer pitch—80- and 40-nm and pitch for 14- and 7-nm technology nodes, respectively [Jan15, Wu16]. Thus, improvements in N3XT would remain unchanged or even increase with technology scaling.

Impact of alternative logic devices in the compute tier. These devices bring significant improvements only in conjunction with large numbers of concurrent compute-to-memory connections, or when workloads are more compute-bound. To illustrate this, we sweep both the energy and delay benefits of the logic devices used in the bottom compute tier (Figure 1) from 1× to 5× compared to SiFETs (i.e., 1× to 25× EDP benefits versus SiFETs) for the following architecture configurations (representing current 2D and 3D stacked architectures as well as N3XT):

- a 2D system with off-chip DRAM (Figure 13a),
- a N3XT system with a 5-µm via pitch, and
- a N3XT system with a 100-nm ILV pitch (Figure 13b).

We assume that CNFETs are used for the logic circuits in upper tiers in the analyzed 3D systems—compute-to-memory interconnect, memory controllers, and memory-access circuits.

Consider `Language Model` with a single-input batch (Figure 16a). Table 8 shows that almost 99% of the total time for this workload is consumed in memory accesses for 2D baseline system. The time spent in memory accesses dramatically improves with N3XT. However, it continues to dominate the application runtime. For such workloads, the 2D system brings only 1.03× system-level EDP benefits when SiFETs are replaced with highly energy-efficient logic devices (e.g., 25× EDP benefits in the compute tier). Similarly, N3XT with a 5-µm via pitch has limited improvements as illustrated in Figure 16a. In contrast, N3XT (with 100-nm ILV pitch and CNFET-based compute tier) shows noticeable gains—system-level EDP benefits increase by 1.9× compared to N3XT with SiFETs in the bottom compute tier (N3XT enjoys 1,971× improvements versus the 2D baseline system, as shown in Section VI).

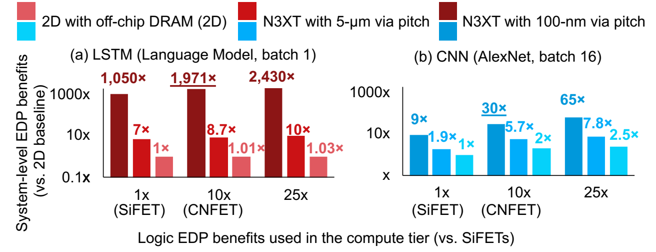


Figure 16. System-level EDP benefits (over the 2D baseline system illustrated in Figure 13a with SiFETs-based compute units and on-chip SRAM connected to off-chip DRAM) versus the EDP benefits of logic devices used in the compute tier. The benefits for the N3XT architecture discussed in Section VI are highlighted. Three system configurations are considered: a 2D system with off-chip DRAM main memory akin to that in Figure 13a, a N3XT system vertically-connected using 5-µm-pitch vias, and a N3XT system (Figure 13b) vertically-connected using 100-nm-pitch vias. Results shown for domain-specific accelerator running: (a) Language Model LSTM network with batch size of 1 (more memory-bound), and (b) AlexNet CNN network with batch size of 16 (more compute-bound).

As expected, improved logic devices in the compute tier have a notable impact for more compute-bound workloads. For instance, system-level EDP benefits for 2D baseline double after SiFETs are replaced by CNFETs in the compute tier, when simulating `AlexNet` with a 16-input batch (Figure 16b). As logic devices in the compute tier continue to improve, such workloads start becoming more and more memory-bound, and improvements in compute-to-memory connectivity provide bigger benefits (consistent with our previous discussions in this section).

Impact of alternative logic devices in the upper tiers. The performance and energy efficiency of logic devices used in upper tiers (Figure 1) is critical to retaining the benefits of N3XT. To place FETs in these upper tiers with monolithic 3D integration, one can use:

- low-temperature logic devices, such as CNFETs, 2D materials and thin-film transistors [Naito10], or
- alternative lower-temperature fabrication techniques for SiFETs [Brunet16].

These approaches, among others, may alter the energy-delay product (EDP) of logic devices placed in the upper tiers and consequently, system-level EDP benefits in N3XT.

To illustrate this, we sweep the energy and delay of logic devices in upper tiers of the N3XT architecture from 0.1× to 5× (i.e., 0.01× to 25× EDP benefits) compared to SiFETs (CNFETs is used in the bottom compute tier). This sweep represents devices with better-than-SiFET EDP that can be fabricated at low temperature (e.g., CNFETs), or devices that may be compatible with low-temperature fabrication but have worse EDP that SiFETs (e.g., thin-film oxide transistors). We show in Figure 17 system-level EDP benefits in N3XT versus the 2D baseline system (Figure 13a)—we annotated the case where CNFETs are used in the upper tiers, in addition to the following two hypothetical logic devices compatible with low-temperature fabrication:

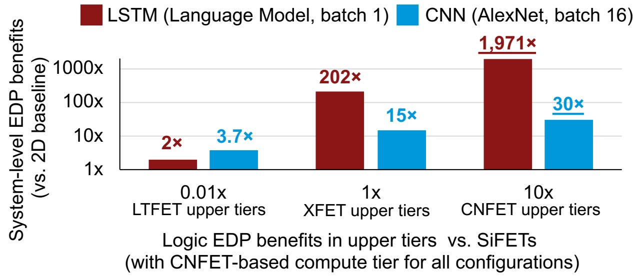


Figure 17. System-level EDP benefits (over the 2D baseline system illustrated in Figure 13a) versus the EDP benefits of logic devices (with respect to SiFETs) used in upper tiers in N3XT (Figure 1). The benefits for the N3XT architecture discussed in Section VI are highlighted. The bottom compute tier is designed with CNFETs. We assume XFET and LTFET are fabricated at low temperature. XFET has identical electrical properties as SiFET, while LTFET has 100× worse EDP than SiFET.

- *XFET* with electrical characteristics identical to those of SiFET, and
- *LTFET* with 100× worse EDP compared to SiFET (10× higher energy and 10× slower devices).

As predicted, memory-bound workloads are more sensitive to such changes. Figure 17 shows that the benefits with monolithic 3D may reach 202× if XFETs are used in the upper tiers (while CNFETs are used in the bottom compute tier). However, the benefits drop significantly to only 2× if LTFETs are used in upper tiers (with CNFETs are still used for the bottom compute tier). If SiFETs are used in the bottom compute tier (for `LSTM` with batch size of 1), system-level EDP benefits reach 155× (instead of 202×) and 1.6× (instead of 2×), if XFETs and LTFETs are used in the upper tiers, respectively.

VIII. RRAM ENDURANCE RESILIENCY

One of the key challenges of RRAM-based design is the limited write endurance of RRAM cells. Endurance is defined as the number of set-reset cycles—writing a zero after writing a one—after which a cell is either stuck at one or stuck at zero [Chen12b] (Section III). In this paper, we estimate the endurance of a single word by the endurance of a single cell[9]. We also pessimistically assume that two consecutive writes to a memory word correspond to a single endurance cycle. RRAM endurance up to 10^{12} cycles has been demonstrated [Kim11, Hsu13] for few cells, whereas demonstrations of at the level of 1 Mbit array, show 10^5-10^7 cycles [Calderoni14, Grossi16, Chen17] (vs. 10^{15} DRAM endurance cycles). Thus, using RRAM in N3XT without compensating for the limited RRAM endurance severely degrades its operation *lifetime*—time until the first cell breakdown. In this section, we introduce the ENDURER technique which overcomes limited RRAM endurance by reducing the peak number of writes per word from 10^{13}-10^{14} to at most ~10^7 during continuous ten-year operation for each workload from Sections V and VI (some workloads require only 10^5 writes per word). ENDURER incurs negligible overheads—additional 16-KByte SRAM per 1-GByte RRAM, as well as 0.01% and 0.7% increased execution time and energy consumption.

Writes to RRAM in N3XT during continuous ten-year operation of selected workloads are summarized in Figure 18. These workloads exhibit the highest number of writes per word for each application. Our analysis indicates that <2% of RRAM words in N3XT experience >10^6 writes per word (maximum 10^{14} writes per word). Continuous execution of these workloads would imply lifetimes on the order of ten seconds assuming 10^6 RRAM endurance cycles [Sheu09, Chen12b, Grossi16].

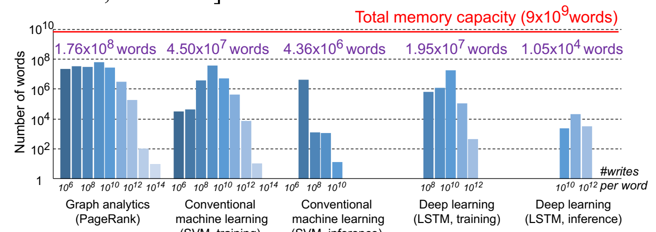


Figure 18. A histogram of the number of writes per word observed over ten-year continuous operation for workloads (with the largest number of writes to RRAM) of each application domain. Total number of written words is highlighted for each workload.

Device-level optimizations such as tuning cell geometries, changing RRAM materials, and tuning the write latency and current, can increase RRAM endurance [Lee11, Chen12b, Strukov16] and may be considered for future work.

Architectural optimizations such as reducing operating frequency, the number compute units, and RRAM connectivity, can improve system lifetime. However, such changes may reduce the resulting EDP benefits of N3XT.

Avoiding unnecessary writes to RRAM is an effective approach to increase lifetime considerably. We observe that the inference phase of some deep-learning applications uses large read-only data structures (DNN weights) but relatively small read-write data structures (intermediate DNN variables). This insight suggests an architecture that maps the smaller data structures to SRAM and the larger data structures to RRAM. For example, `Language model` (Table 7) requires >2 GBytes for weights (read only), whereas intermediate variables (read and write) occupy 1.6 and 25.6 MBytes for batch sizes of 1 and 16, respectively. Integrating such SRAM capacity is possible. However, this approach may not be adequate for other workloads, including

[9] This is a pessimistic estimate. On the one hand, a write operation does not necessarily modify all bits within a single word. On the other hand, the endurance failure of any one cell fails the entire word. Those two trends tend to balance each other out. Additionally, the least significant bit of each word often goes through more write cycles than other bits [Zhou09], in which case the endurance of the word is determined by the endurance of one cell.

inference for deep learning, that require significant SRAM—e.g., `VGGNet-19` occupies 290 MBytes for weights (Table 7), while 12 and 192 MBytes for intermediate variables, batch sizes of 1 and 16, respectively.

The ENDURER[10] technique redistributes write operations evenly such that the expected number of writes per word is $W^* = \frac{N_w}{M}$ (Figure 19), where N_w is the total number of writes to a memory of size M during lifetime L. For example, a workload with 10^{18} writes for M = 64 GBytes in L = 10 years implies $W^* = 1.4\times10^7$.

ENDURER implements this approach by combining the following components:

- *address redirect* shifts memory addresses by a given offset,
- *offset scheduler* controls address redirect,
- *periodic remapping* supports offset changes by shifting memory contents accordingly, and
- SRAM-based *write-back buffer* reduces writes to heavily-written words (>10^{11} writes/word).

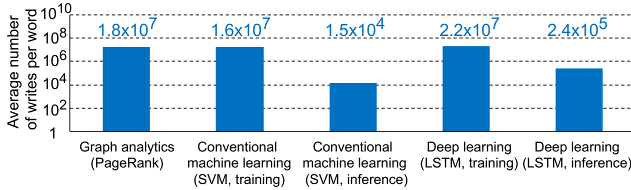


Figure 19. Average number of writes per word (referred to as W^*) over ten-year continuous operation. Values are shown for selected workloads of each application domain, following Figure 18.

Address redirect maps each incoming memory address (A) into a different address ($\bar{A}$) by adding offset Θ

$$\bar{A} = \mathrm{mod}(A + \Theta, M)$$

where M is the size of the memory.

Offset scheduler selects Θ as a random value uniformly distributed in {0, 1, …, M-1} every T seconds. Using this assumption, Appendix F derives an upper bound on the expected maximum number of writes per word in L.

When changing Θ_i to Θ_{i+1}, the memory contents need to be shifted by $\Delta_i^\Theta = \mathrm{mod}(\Theta_{i+1} - \Theta_i, M)$.

The period T is selected to ensure that enough remapping operations occur to balance writes across memory words, while not causing significant additional writes (since single remapping operation incurs a write to each memory word). In this paper we select T = 3,100 seconds such that 10^5 remapping operations occur within 10 years. Remapping with this period incurs 10^5 additional writes per word.

Periodic remapping cyclically shifts the contents of the entire memory. In particular, each address A_j is mapped to

```
Algorithm: Periodic Remapping
Inputs: Memory size M, random offset Δ^Θ
N_cycles ← 2^ctz(Δ^Θ);
for (η=0; η < N_cycles ; η ++)
{
  A ← η; Ā ← A +Δ^Θ;
  DataA ← Read main memory (A);
  while Ā ≠ η
  {
    DataB ← Read main memory (Ā);
    Write to main memory (Ā, DataA);
    DataA ← DataB; A ← Ā; Ā ← (A +Δ^Θ) mod M;
  }
  Write to main memory (Ā, DataA);
}
```

Figure 20. The periodic remapping algorithm.

$$\bar{A}_j = \mathrm{mod}\left(A_j + \Delta_i^\Theta, M\right), j = \{0,1 \ldots, M-1\}$$

As shown in Figure 20, memory values are shifted in cycles. Multiple such cycles may be required to remap the entire memory. Assuming M is a power of 2, the number of such cycles is:

$$N_{\text{cycles}} = 2^{ctz(\Delta_i^\Theta)}$$

where $\mathrm{ctz}(\Delta_i^\Theta)$ is the number of trailing zeros in the binary representation of Δ_i^Θ.[11] For $M=16$ and $\Delta_i^\Theta = 4$, the first cycle will be *0-4-8-12-0*. Additional cycles start at *1*, *2* and *3*.

Remapping does not reduce the number of writes per word within T. To reduce peak writes per word, we use following technique.

The write-back buffer is a fully-associative SRAM of size S words that places an upper bound on the number of writes to each RRAM word within a shifting period ($\widehat{n_w}$). The buffer size is determined by:

$$S = \frac{N_w}{N_s \widehat{n_w}} \tag{1}$$

where $N_s = \frac{L}{T}$. By setting $\widehat{n_w} = W^*$, then $S = \frac{M}{N_s}$ guarantees that no single memory word has more than W^* writes per shifting period, which we prove as follows. For a workload with a total number of writes N_w and N_s remapping operations, the expected number of writes between two remapping periods is $N_{w,s} = \frac{N_w}{N_s}$. The number of words ($S^*$) that could have $\widehat{n_w} \geq W^*$ writes is then:

$$S^* = \frac{N_{w,s}}{\widehat{n_w}} \leq \frac{N_{w,s}}{W^*} = \frac{M}{N_s} = S \tag{2}$$

[10] ENDUrance REsiliency by random Remapping
[11] For example, ctz (4) = 2 and ctz (6) = 1.

For a buffer of size S, no word with $\widehat{n_w} \geq W^*$writes can exercise such heavy writes in RRAM.

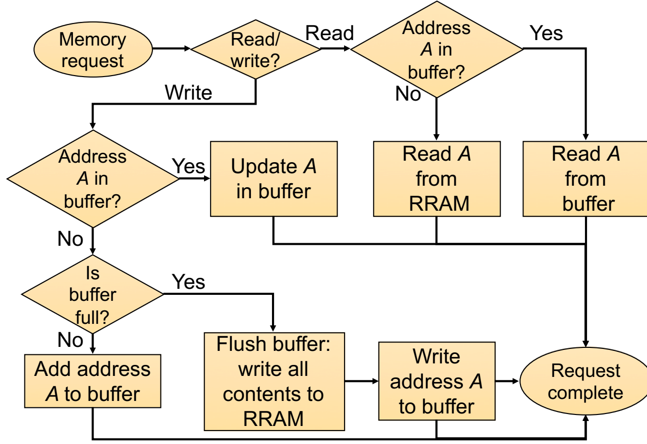


Figure 21. Memory access with a write-back buffer integrated into the memory controller.

The buffer size (S^*) is 10.4 KBytes for a memory segment of M = 1 GByte (per memory controller) and $N_s = 10^5$ periodic remapping operations (derived using equation 2), which is increased to 16 KBytes to account for storing the word addresses (for each 64-bit word in the buffer, 27 bits are required to store the address tag as the buffer is fully-associative, in addition to a valid bit).

Figure 21 illustrates RRAM access in N3XT using the write-back buffer integrated into the memory controller. For each write request, if the address resides in the buffer, data contents are overwritten. Otherwise, the buffer contents are evicted from the buffer to RRAM and incoming data are buffered. For a read request, data are read from the buffer if the address resides there or are read from RRAM bypassing the buffer.[12]

Simulation results illustrated in Figure 22 include the maximum number of writes per RRAM word observed during ten-year continuous operation in N3XT. This represents a pessimistic scenario, and we consider more practical operating conditions later. We use the framework in Section IV to estimate the number of writes for each memory word for each workload and the corresponding execution time. Results show that ENDURER reduces the maximum number of writes by at least 2×10^4 times across all workloads in N3XT, compared to naïve execution without ENDURER (*no resiliency*). The use of ENDURER in N3XT extends the lifetime of continuous operation of inference workloads to ten years (assuming 10^6 RRAM endurance cycles).

Certain workloads, such as the training phase of conventional machine learning, deep learning and graph analytics, require $>10^6$ writes per word during continuous ten-year operation, which reduces the lifetime to one year. These workloads, however, are not executed continuously in practice [Hazelwood18], e.g., deep-learning networks are trained daily or weekly, while inference runs more frequently or even continuously. N3XT can ensure ten-year lifetime in such practical scenarios for each considered workload. For instance, a CPU-based N3XT architecture achieves ten-year lifetime (using RRAM with 10^6 endurance cycles) with continuous daily operation of incremental training[13] for 1 hour and inference for 24 hours of LSTM with a maximum of 1.7×10^6 writes per RRAM word (8.5×10^5 endurance cycles).

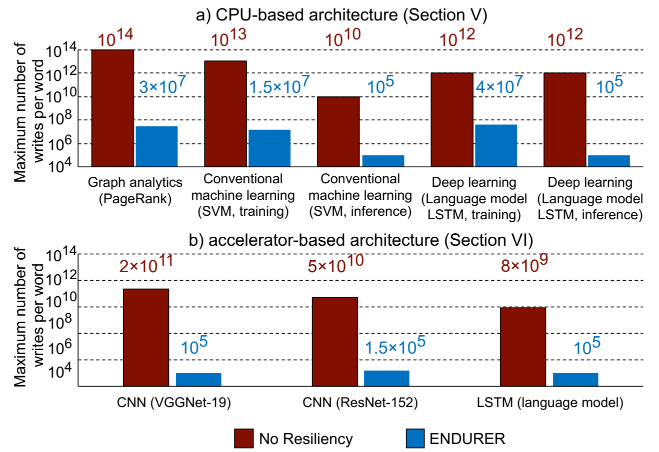


Figure 22. Maximum number of writes per RRAM word (simulated) in a) CPU-based and b) accelerator-based N3XT architectures with ENDURER versus a naïve execution without ENDURER (*no resiliency*) for ten-year continuous operation. The workloads shown experience the highest writes per word for each considered application (Sections V and VI). ENDURER can reduce the writes per word to the order of 10^5, for the inference phase of conventional machine learning and deep learning workloads. Some workloads do not experience writes to RRAM (even with naïve execution) such as the inference of AlexNet mapped on the domain-specific accelerator (Section VI), where ENDURER may not be required.

Hardware implementation of ENDURER within each memory controller is illustrated in Figure 23 with the following components:

- a 27-bit register (R_Θ) to store the accumulated random offset for address redirect,
- a 27-bit adder to redirect memory access with the offset in R_Θ,
- a 50-bit timer loaded with T in cycles for the offset scheduler, supporting up to 1-day period,
- a 27-bit pseudorandom number generator (*PRNG,* implemented using a linear-feedback shift register with a maximal length characteristic polynomial) to create the random offset for the offset scheduler,
- a finite-state-machine (*FSM*) of the periodic-remapping algorithm in Figure 20,

[12] The buffer behaves differently from a conventional cache as it does not store any reads from RRAM and it does not employ cache replacement algorithms as they may be inefficient for fully-associative memories.

[13] improving the accuracy of a pretrained network

- a 16-KByte fully associative SRAM buffer, and
- a buffer manager that checks whether an address exists in the associative memory and flushes the associative memory with each remapping operation.

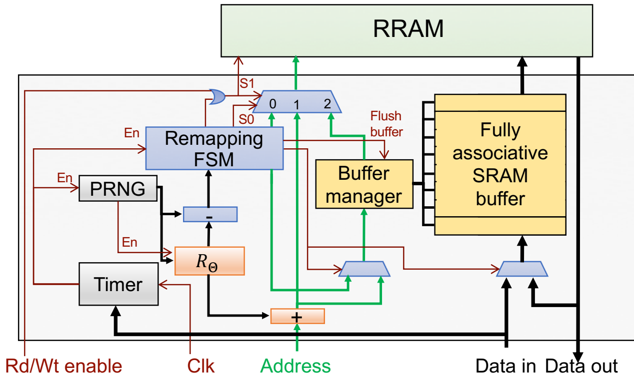


Figure 23. The ENDURER hardware modules implemented within a memory controller (Figure 2). Data in, data out, Address, clk, and Rd/wt enable connect the memory controller to the compute-to-memory interconnect (Figure 5). Address-redirect blocks are shaded in orange, offset-scheduler blocks are shaded in grey, periodic-remapping blocks are shaded in blue and the write-back buffer circuitry is shaded in yellow.

We synthesize the netlist and perform the physical design of ENDURER using the 28 nm CNFET PDK introduced in Section IV.

Energy and runtime overheads of ENDURER in N3XT, summarized in Table 9, reach 0.7% and 0.01%, respectively.[14] These overheads are aggregated from the following sources:

- energy and latency consumed in address redirect for each memory access,
- energy and latency consumed in the offset scheduler,
- energy and latency consumed in the periodic remapping operation, and
- energy and latency to access the SRAM buffer.

IX. Conclusions

The growing demands for abundant-data applications, such as deep learning and graph analytics, expose memory accesses as a major energy and execution time bottleneck in current computing systems [Aly15, Hwang17]. Profound changes in architectural strategies, enabled by new technologies, are required to meet the computational needs of such applications. Our N3XT approach overcomes this major challenge through new nanosystem architectures by leveraging emerging logic and non-volatile memory devices that can be fabricated at low temperature with thin device layers. These device technologies facilitate fine-grained and ultra-dense 3D integration and help overcome current system limitations, improving system-level EDP by three orders of magnitude.

In this paper, we have evaluated N3XT for commonly-used workloads using software implementations targeting traditional 2D systems. Nevertheless, N3XT can achieve even greater benefits through hardware-software codesign, as well as through coordinated optimization of architectural parameters and software for specific application domains. Infrastructure for such optimization can be provided by domain-specific languages (*DSL*) [Brown11, Koeplinger18].

Table 9. Energy and run time overheads of the different components in ENDURER when used in N3XT. We discuss the overhead sources for each component and the corresponding system-level values.

Component	Overhead		
	Source	**Energy**	**Runtime**
Address redirect	Adding offset Θ to the address of each memory access	**0.09%:** <1fJ/bit for each access	**0%:** <0.02 ns for each access which does not increase the number of cycles to access RRAM
Offset scheduler	Updating the timer every clock cycle and generating a new offset Θ with PRNG each period *T*	**0.00003%:** 10 µJ for each period T	**0%:** 0.25ns for each access that occurs in parallel to memory accesses
Periodic remapping	Reading all memory words and writing them back to RRAM as well as activating the FSM each period *T*	**<0.01%:** 31×*N* mJ or 10.3×*N* mJ for CPU-based or accelerator-based architectures with *N*-GByte RRAM	**0.01%:** 11 ms derived from access latency, number of memory controller, access-channels, and capacity
SRAM write-back buffer	Checking the address in the buffer, writing data into the buffer and reading data if it resides in the buffer	**0.6%:** 18pJ to check the address [Arsovski13] for each access and 0.1 pJ/bit for each write and read if data resides in buffer	**0%:** 0.3ns to check address for each access and 0.6ns to write data in buffer or read if data resides in buffer (read and write are both faster than RRAM-access latency in Tables 1 and 6)
Total		**0.7%**	**0.01%**

[14] Energy overhead can be further reduced, e.g., replacing $\bar{A} = \text{mod}(A + \Theta, M)$ with $\bar{A} = A \oplus \Theta$ in address redirect, and may be considered for future work.

In particular, DSL compilers can improve memory locality, which can help reduce inter-chip communication when data requirements exceed on-chip memory capacity in N3XT. Additional runtime support, such as task partitioning, scheduling and migration techniques [Kasture17] can promote inter-chip workload movement and reduce communication even more.

N3XT is not limited to the architecture choices illustrated in this paper. To further improve performance, compute elements can be placed in upper tiers, closer to memory. However, upper compute tiers can exhibit higher power density than memory tiers, necessitating advanced cooling solutions to balance temperature across the full architecture and remove potential thermal hotspots generated by these compute tiers. These solutions are currently under investigation, such as copper nanomesh and nanowires [Won13, Barako17], two-phase cooling [Palko16], phase change materials [Fuensanta13, Barako18] and 2D materials [Pop12, Choi18].

Furthermore, N3XT is compatible with and can serve as an enabler for non-traditional computing paradigms, such as brain-inspired hyperdimensional computing [Wu18] and computing inside memory arrays (e.g., [Li17]), thanks to the ultra-dense connectivity between compute and memory units that are placed vertically adjacent to each other. The architecture is also not limited to compute and memory blocks: sensor tiers can be placed at the top so that processing is performed where data are acquired [Shulaker17]. System-level interconnects can leverage photonic technologies based on recently demonstrated CMOS-compatible photonic sources and light waveguides [Levy10, Piggott15].

Yield and cost of N3XT manufacturing must be addressed through device-, circuit- and architecture-level techniques, e.g., using techniques to tolerate hardware failure at the circuit, architecture and application levels [Cho12, Hills18b, Li13]. N3XT hardware prototypes are leading examples of translating the basic science of nanomaterials and nanodevices into actual nanosystems. To this end, N3XT can adopt a wide range of architecture, integration and device technology choices. While further research is needed to demonstrate the illustrated improvements in a working N3XT prototype (i.e., simulated architectures in Sections V and VI), various hardware demonstrations of N3XT technology foundations, summarized in Table 10, have already made significant progress toward this goal.

Security implications of N3XT must be carefully explored in various contexts: supply-chain attacks, side-channel attacks, and runtime attacks on non-volatile memory.

Acknowledgments. We acknowledge the support of DARPA, NSF-SRC/NRI/GRC E2CDA, STARnet SONIC, NSF, and member companies of the Stanford SystemX Alliance.

REFERENCES

[Abadi16] M. Abadi et al. "TensorFlow: Large-scale machine learning on heterogeneous distributed systems." arXiv preprint arXiv:1603.04467, 2016.
[Aberger16] C. Aberger et al, “EmptyHeaded: A Relational Engine for Graph Processing”, In SIGMOD 2016.
[Aly15] M. M. S. Aly et al., “Energy-efficient Abundant-data Computing: The N3XT 1,000x”, Computer Magazine, 2015.
[Appenzeller08] J. Appenzeller, "Carbon nanotubes for high-performance electronics—Progress and prospect." *Proceedings of the IEEE* 96.2 (2008): 201-211.
[Arsovski13] I. Arsovski et al., “A 32 nm 0.58- fJ/bit/search 1-GHz ternary content addressable memory compiler using silicon-aware early-predict late-correct sensing with embedded deeptrench capacitor noise mitigation”, in JSSC, 2013.
[Balkan09] A. O. Balkan et al., "Mesh-of-Trees and Alternative Interconnection Networks for Single-Chip Parallelism," in TVLSI, vol. 17, no. 10, pp. 1419-1432, Oct. 2009.
[Barako17] M. T. Barako et al., “Dense Vertically Aligned Copper Nanowire Composites as High Performance Thermal Interface Materials”, in ACS applied materials and interfaces, 2017.
[Barako18] M. T. Barako et al., “Optimizing the design of composite phase change materials for high thermal power density”, in Journal of Applied Physics, 2018
[Batude15] P. Batude et al., “3DVLSI with CoolCube Process: An alternative path to Scaling”, in VLSI Technology Symposium, 2015.
[Bienia08] C. Bienia et al., “The PARSEC Benchmark Suite: Characterization and Architectural Implications”, in PACT, 2008.
[Boldi14] P. Boldi, A. Marino, M. Santini, and S. Vigna, “BubiNG: Massive Crawling for the Masses”, Prod. 23rd International Conference on World Wide Web”, 2014.
[Bolosky89] W. J. Bolosky, R. P. Fitzgerald and M. L. Scott, “Simple but Effective Techniques for NUMA Memory Management”, in SOSP, 1889
[Braojos16] R. Braojos et al., “Nano-engineered architectures for ultra-low power wireless body sensor nodes”, in CODES+ISSS, 2016.
[Brady16] G. J. Brady et al., “Quasi-ballistic carbon nanotube array transistors with current density exceeding Si and GaAs”, in Science advances, 2016
[Brown11] K. J. Brown et al., “A Heterogeneous Parallel Framework for Domain-Specific Languages,” in PACT, 2011.
[Brunet16] L. Brunet et al., “First Demonstration of a CMOS over CMOS 3D VLSI CoolCube™ integration on 300mm wafers”, in VLSI, 2016.
[Calderoni14] A. Calderoni et al., “Performance Comparison of O-based and Cu-based ReRAM for high-density applications”, in IMW, 2014.
[Calibre10] Calibre xACT 3D — No Compromise Extraction For Advanced Transistor Level Design (online) http://go.mentor.com/4guhu, 2010.
[CCAFS16] Climate Change, Agriculture, and Food Security [online], http://www.ccafs-climate.org/weather_stations/, 2016.
[Chang07] Y. -H. Chang et al., “Endurance Enhancement of Flash-Memory Storage System: An Efficient Static Wear Leveling Design”, DAC 2007.
[Chang10] Y.-H. Chang et al., “Improving Flash Wear-Leveling by Proactively Moving Static Data”, IEEE Trans. Computers, 2010.
[Chang11] M.-F. Chang et al., “A 3T1R Nonvolatile TCAM Using MLC ReRAM with Sub-1ns Search Time”, in ISSCC, 2011

[Chang12] M.-F. Chang et al., “A 0.5 V 4Mb logic-process compatible embedded resistive RAM (RERAM) in 65nm CMOS using low-voltage current-mode sensing scheme with 45nm random read time”, in ISSCC, 2012.
[Chang13] M.-F. Chang et al., “An Offset-Tolerant Fast-Random-Read Current-Sampling-Based Sense Amplifier for Small-Cell-Current Nonvolatile Memory”, IEEE JSSC, 2013.
[Chen12a] Y.-H. Chen et al, “HfOx Based Vertical Resistive Random-access Memory for Cost-Effective 3D Cross-point Architecture Without Cell Selector”, IEDM 2012.
[Chen12b] Y. Y. Chen et al., “Balancing SET/RESET Pulse for > 10^{10} Endurance in HfO2/Hf 1T1R Bipolar RRAM”, IEEE TED, 2012.

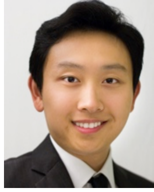
Table 10. Key hardware demonstrations and system-level prototypes for technologies used in N3XT. We particularly focus on large-scale demonstrations of logic and memory devices, as well as integration techniques that are used in Sections III, IV, V and VI. N3XT can also adopt other technology options (see Section III), but they can offer lower energy-efficiency benefits.

Technology foundations	**Recent key demonstrations (see Section III)**	**Full-chip- and system- level analyses (simulated architectures in Sections V and VI)**
Energy-efficient logic fabricated at low temperature	CNFET-based microprocessor using 178 CNFETs and 10-200 CNTs per CNFET fabricated at wafer scale [Shulaker13]. CNFET-based hardware accelerator for deep-learning inference. 4-bit multiply-and-accumulate units are fabricated at wafer scale with 385 CNFETs per unit [Hills18b]. Five-stage CNFET-based ring oscillator with 282MHz oscillation frequency [Han17]. Five-stage CNFET-based ring oscillator with up to 5.4GHz oscillation frequency [Zhong18a]. High-performance and energy-efficient CNFETs: 100 CNTs/µm, 122µA/µm on-current @ 1V V_{DD} and 10nA/µm off-current [Shulaker14a] 47 CNTs/µm, 900µA/µm on-current @ 1.2V V_{DD} and 100nA/µm off-current [Brady16] 125 CNTs/µm and 960µA/µm on-current @ 0.7V V_{DD} [Qiu17] CNFETs with adjustable threshold voltage (-1.0-0.2V) via gate-electrode-thickness tuning [Zhong18b]. Negative-capacitance CNFETs with 55mV/decade subthreshold slope at room temperature and 2.1× higher on-current versus baseline CNFETs [Srimani18] Logic circuits built with CNT-based 3D-FETs with two vertically-stacked CNFETs sharing a common gate demonstrated at wafer scale. This can enable area-efficient designs [Kanhaiya18].	CNFET-based compute units (cores and accelerators) that simultaneously run 3× faster and 3× lower energy versus SiFET-based designs. CNFETs operate at 65mV/decade subthreshold slope, 250 CNTs/µm, ~2000µA/µm @ 0.5V V_{DD} on-current and 100nA/µm off-current. Our simulations are backed up by industry-practice physical design at relevant technology nodes, down to 5nm, of all modules (including effects from parasitics and interconnect wires) to determine the energy and delay benefits of CNFETs versus SiFETs. To ensure realism, when developing our CNFET-based PDK, we relied on a device-level model calibrated with experimental measurements for gate lengths down to 9nm. The model also accounts for CNT-specific variations—e.g., variations in CNT density and CNT diameter, direct source-to-drain tunneling leakage current, parasitic capacitance, and parasitic CNT-metal contact resistance.
Low-latency and high-density non-volatile memory fabricated at low temperature	4-layer 3D RRAM with SiFET-based access circuitry: ±0.8V set/reset voltage, >10× HRS/LRS ratio, 10^6 endurance cycles, and 30ns write-pulse width [Li16]. 8-layer 3D RRAM with SiFET-based access circuitry: 8/-4.5V set/reset voltage, ~100× HRS/LRS ratio, 10^7 endurance cycles, and down to 100ns write-pulse width [Luo17]. 11-Mbit embedded RRAM with SiFET-based access circuitry fabricated at 40nm technology node: $53F^2$ cell size, ±1.4V set/reset voltage, 0.26V read voltage, >5× HRS/LRS ratio and 9ns read-access time [Chou18]. 32-Gbit RRAM with diode selectors fabricated at 24nm technology node: $6F^2$ cell size and 40/230µs read/write latency [Liu14]. 4-kbit multi-valued RRAM arrays with three bits per RRAM cell [Le18]. 16-Gbit 1T1R RRAM with SiFET-based access circuitry fabricated at 27nm technology node: $6F^2$ cell size and 2/10 µs read/write latency [Fackenthal14]. 40-Mbit STTRAM-based embedded memory with SiFET-based access circuitry fabricated at 2xnm FD-SOI technology node: 2× HRS/LRS ratio, 10^7 endurance cycles, and 20/50ns read/write latency [Shum17]. 4-Gbit STTRAM with SiFET-based access circuitry fabricated at 90nm technology node: $9F^2$ cell size and 50.5ns read latency [Rho17].	≥1-GByte 8-layer 3D RRAM with CNFET-based access circuitry and CNFET-based memory controller. RRAM cells have ±1.2V set/reset voltage, 0.5V read voltage, 10× HRS/LRS ratio, 10^6 endurance cycles, $12F^2$ cell size, and 10ns write-pulse width. ≥1-MByte STTRAM with CNFET-based access circuitry and CNFET-based memory controller. STTRAM cells have ±0.9V set/reset voltage, 0.5V read voltage, 2× HRS/LRS ratio and $9F^2$ cell size and 2ns write-pulse width. Design of all memory subsystems is based on experimental measurements of memory cells, i.e., read and write voltages and currents, write pulse width, cell area, and on/off resistances. Dimensions of CNFET-based access transistors are determined via SPICE simulations. Digital blocks are modeled using CNFET PDK (mentioned above). Analog components are simulated in SPICE with device parameters from the CNFET PDK. Cycle-to-cycle and cell-to-cell variations are accounted for when designing the memory-access circuitry.
Fine-grained and ultra-dense integration of logic and memory	Three-tier monolithic 3D integration of CNFET-based logic gates that performs basic logic operations, i.e., inverter and two-input XOR gates [Wei09]. Two-tier monolithic 3D integration of CNFETs on top of SiFETs [Shulaker14b]. Four-tier monolithic 3D integration of CNFETs and RRAM on a SiFET bottom tier demonstrating a switching element of a switchbox for an FPGA [Shulaker14d]. Four-tier monolithic 3D integration of more than two million CNFETs and 1-Mbit RRAM on top of a SiFET bottom tier demonstrating computation immersed in memory. CNFET-based sensors are placed on the top tier, where captured data are stored in RRAM and then processed on an on-chip hardware accelerator designed using CNFETs [Shulaker17]. Two-tier monolithic 3D integration of 1,952 CNFETs and 224 RRAM cells that exploits variations in CNFETs and RRAM to perform cognitive tasks such as language recognition [Wu18].	Monolithic 3D integrated system of more than four tiers with CNFET-based hardware accelerator for deep-learning inference, 3MByte CNFET-based SRAM, 4-GByte 3D RRAM with CNFET-based access circuitry and high memory-access bandwidth using sub 100-nm interlayer via pitch. Monolithic 3D integrated system of more than four tiers comprising CNFET-based general-purpose multicore processor, CNFET-based SRAM, ≥ 1-MByte STTRAM with CNFET-based access circuitry, ≥ 1-GByte 3D RRAM with CNFET-based access circuitry and high memory-access bandwidth using sub 100-nm interlayer via pitch. Our designs are implemented using a monolithic 3D physical design methodology (Section IV) that leverages commercial design tools for conventional ICs. We account for ILV locations and their load capacitance and drive resistance in the place-and-route stage of the design.

[Chen14] D. Chen et al., “Single-Event Effect Performance of a Commercial Embedded ReRAM”, IEEE Trans. Nuclear Science, 2014
[Chen16] Y. H. Chen et al., “Eyeriss: A Spatial Architecture for Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks”, in ISSCC, 2016.
[Chen17] Z. Chen et al., “Performance Improvements by SL-Current Limiter and Novel Programming Methods on 16MB RRAM Chip”, in IMW, 2017.
[Chiriac16] V. Chiriac et al., “A figure of merit for mobile device thermal management,” IEEE ITherm, 2016.
[Cho12] H. Cho, L. Leem and S. Mitra, “ERSA: Error resilient system architecture for probabilistic applications,” IEEE Trans. CAD., vol. 31, no. 4, pp. 546–558, Apr. 2012.
[Choi18] D. Choi et al., “Large Reduction of Hot Spot Temperature in Graphene Electronic Devices with Heat-Spreading Hexagonal Boron Nitride”, in ACS Applied Material Interfaces, 2018
[Chou18] C.-C. Chou et al.,” An N40 256KÅ~ 44 Embedded RRAM Macro with SL-Precharge SA and Low-Voltage Current Limiter to Improve Read and Write Performance”, in ISSCC, 2018
[Chung10] S. Chung et al., “Fully integrated 54nm STT-RAM with the Smallest Bit Cell Dimension for High Density Memory Application”, in IEDM 2010.
[Chung16] S.-W. Chung et al., “4Gbit Density STT-MRAM using Perpendicular MTJ Realized with Compact Cell Structure”, in IEDM 2016.
[Courbariaux15] M. Courbariaux et al., “BinaryConnect: Training Deep Neural Networks with Binary Weights During Propagations”, in NIPS 2015.
[DDR17] DDR4 specifications (online), https://www.micron.com/~/media/documents/products/data-sheet/dram/ddr4/4gb_ddr4_sdram.pdf
[Dong12] X. Dong et al., “NVSIM: A Circuit-Level Performance, energy and Area Model for Emerging Nonvolatile Memory”, in TCAD, 2012.
[Fackenthal14] R. Fackenthal et al., “A 16Gb ReRAM with 200MB/s Write and 1GB/s Read in 27nm Technology”, in ISSCC 2014.
[Ferruci10] D. Ferrucci et al., “Building Watson: An overview of the DeepQA project”, in AAAI, 2010.
[FFTW15] Fastest Fourier Transform in the West library (online): http://www.fftw.org/, 2015.
[Fritsch15] A. Fritsch et al., “A 4GHz, low latency TCAM in 14nm SOI FinFET technology using a high performance Current Sense Amplifier for AC current surge reduction”, in ESSCIRC, 2015.
[Fuensanta13] M. Fuensanta et al., “Thermal Properties of a Novel Nanoencapsulated Phase Change Material for Thermal Energy Storage,” *Thermochimica Acta*, vol. 565, pp. 95–101, 2013.
[Gao17] M. Gao et al., “TETRIS: Scalable and Efficient Neural NetworkAcceleration with 3D Memory,” ASPLOS, 2017.
[Gonzalez12] J. E. Gonzalez et al, “PowerGraph: Distributed Graph-parallel Computation on Natural Graphs”, In OSDI 2012.
[Gonzalez96] R. Gonzalez and M. Horowitz, “Energy Dissipation in General Purpose Processors”, IEEE JSCC 1996.
[Grossi16] A. Grossi et al. “Fundamental variability limits of filament-based RRAM”, IEDM 2016.
[Ha17] D. Ha et al., “Highly Manufacturable 7nm FinFET Technology Featuring EUV Lithography for Low Power and High Performance Applications,” VLSI Technology, 2017.
[Han16] S. Han et al., “Deep Compression: Compressing DNNs with Pruning, Trained Quantization and Huffman Coding”, in ICLR 2016.
[Han17] S.-J. Han et al., “High-speed Logic Integrated Circuits with Solution-processed Self-assembled Carbon Nanotubes”, in Nature nanotechnology, 2017.
[Hazelwood18] K. Hazelwood et al., “Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective”, in HPCA, 2018
[HBM17] HBM JEDEC specification (online), https://www.jedec.org/document_search?search_api_views_fulltext=jesd235a
[HBP16] Human microbiome Project, RNA Dental expression analysis [online], http://hmpdacc.org/RSEQ/, 2016.
[He16] K. He et al. "Deep residual learning for image recognition." In CVPR, 2016.
[Henessey12] J. Hennessey and D. Patterson “Computer Architecture: A Quantitative Approach”, Fifth Edition, 2012.
[Hills15] G. Hills et al., “Rapid co-optimization of processing and circuit design to overcome carbon nanotube variations,” *IEEE Trans. CAD*, vol. 34.7, pp. 1082-1095, 2015.
[Hills17] G. Hills, “Variation-Aware Nanosystem Design Kit,” https://nanohub.org/resources/22582.
[Hills18a] G. Hills et al., “Understanding Energy Efficiency Benefits of Carbon Nanotube Field-Effect Transistors for Digital VLSI”, in IEEE TNANO, 2018
[Hills18b] G. Hills et al., “TRIG: Hardware Accelerator for Inference-Based Applications and Experimental Demonstration using Carbon Nanotube FETs”, in DAC, 2018.
[Ho16] C. Ho et al., “Random Soft Error Suppression by Stoichiometric Engineering: CMOS Compatible and Reliable 1Mb HfO2-ReRAM with 2 Extra Masks for Embedded IoT Systems”, in Symp. VLSI, 2016.
[Hsu13] C. -W. Hsu et al., “Self-Rectifying Bipolar TaOx/TiO2 RRAM with Superior Endurance over 10^{12} Cycles for 3D High-Density Storage-Class Memory”, VLSI 2013.
[Huylenbroeck16] S. V. Huylenbroeck et al., “Small Pitch, High Aspect Ratio Via-last TSV Module”, in ECTC, 2016.
[Hwang17] W. Hwang et al., “3D Nanosystems Enabled Embedded Abundant-data Computing”, in CODES+ISSS, 2017.
[INet17] ImageNet flower dataset (online), http://www.imagenet. org/search?q=flower, 2017.
[Intel15] Intel® 64 and IA-32 Architectures Optimization Reference Manual (online), https://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-optimization-manual.pdf, 2015
[Iranfar17] A. Iranfar et al., “Thermal Characterization of Next-Generation Workloads on Heterogeneous MPSoCs”, in SAMOS, 2017.
[Jacob10] B. Jacob S. Ng and D. Wang, “Memory Systems: Cache, DRAM, Disk”, Morgan Kaufmann, 2010.
[Jan15] C.-H. Jan et al., “A 14 nm SoC Platform Technology Featuring 2nd Generation Tri-Gate Transistors, 70 nm Gate Pitch, 52 nm Metal Pitch, and 0.0499 um^2 SRAM cells, Optimized for Low Power, High Performance and High Density SoC Products”, in VLSI Symp. 2015.
[Jiang18] Z. Jiang et al., “Selector Requirements for Tera-Bit Ultra-High-Density 3D Vertical RRAM”, in VLSI Symp. 2018.
[Jouppi17] N. Jouppi et al. “In-Datacenter Performance Analysis of a Tensor Processing Unit,” ISCA, 2017.
[Jozefowicz16] R. Jozefowicz et al., “Exploring the Limits of Language Modeling”, arXiv:1602.02410, 2016.
[Jung14] Jung et al., “TSV stress-aware full-chip mechanical reliability analysis and optimization for 3D IC”, Communications of the ACM, 2014

[Kanhaiya18] P. S. Kanhaiya et al., "DISC-FETs: Dual Independent Stacked Channel Field-Effect Transistors", in IEEE Electron device letters, 2018
[Kasture17] H. Kasture et al., "Improving Datacenter Efficiency through Portioning-Aware Scheduling", in PACT 2017.
[Kawahara13a] A. Kawahara et al., "An 8 Mb Multi-Layered Cross-Point ReRAM Macro with 443 MB/s Write Throughput", in JSSC 2013.
[Kawahara13b] A. Kawahara et al., "Filament Scaling Forming Technique and Level-Verify-Write Scheme with Endurance Over 10^7 Cycles in ReRAM", in ISSCC 2013.
[Kgil08] T. Kgil et al., Improving nand flash based disk caches", in ISCA, 2008.
[Kim02] J. Kim, J. M. Kim, S. H. Nog, S. L. Min and Y. Cho, "A Space-efficient FLASH Translation Layer for Compact FLASH Systems", in IEEE TCE, 2002.
[Kim11] Y. -B. Kim et al., "Bi-layered RRAM with unlimited endurance and extremely uniform switching", VLSI 2011.
[Kim15] C. Kin et al., "A covalent-bonded cross-coupled current-mode sense amplifier for STT-MRAM with 1T1MTJ common source-line structure array", in ISSCC, 2015.
[Kim16] S. W. Kim et al, "Ultra-fine Pitch 3D Integration Using Face-to-Face Hybrid Wafer Bonding Combined with a Via-Middle Through Silicon-Via Process", in ECTC, 2016.
[Koeplinger18] D. Koeplinger et al., "Spatial: a language and compiler for application accelerators", in SIGPLAN, 2018
[Krizhevsky12] A. Krizhevsky et al., "ImageNet Classification with Deep Convolution Neural Networks", in NIPS 2012.
[Lau18] C. Lau et al., "Tunable n-Type Doping of Carbon Nanotubes Through Engineered Atomic Layer Deposition HfOX Films", ACS Nano, 2018
[Le18] B. Le et al., "Resistive RAM with Multiple Bits per Cell: Array-Level Demonstration of 3 Bits per Cell", in TED 2018.
[LeCun15] Y. LeCun et al., "Deep Learning", in Nature vol 521, 2015.
[Leduc08] Leduc, P. et al., "Enabling technologies for 3D chip stacking". In VLSI-TSA, 2008.
[Lee10] B. Lee et al., "Phase-Change Technology and The Future of Main Memory", MICRO, 2010.
[Lee11] M.-J. Lee et al." A fast, high-endurance and scalable non-volatile memory device made from asymmetricTa2O5−x/TaO2−x bilayer structures", Nature, Vol 11, 2011.
[Lee15a] C.-S. Lee et al., "A Compact Virtual-Source Model for Carbon Nanotube FETs in the Sub-10-nm Regime-Part I: Intrinsic Elements," *IEEE Trans. Electron Devices*, 2015.
[Lee15b] C.-S. Lee et al., "A Compact Virtual-Source Model for Carbon Nanotube FETs in the Sub-10-nm Regime-Part II: Extrinsic Elements, Performance Assessment, and Design Optimization," *IEEE Trans. Electron Devices*, 2015.
[Lee16] C.-S. Lee et al., "32-bit Processor Core at 5-nm Technology: Analysis of Transistor and Interconnect Impact on VLSI System Performance", in IEDM 2016
[Lee17] Chi-Shuen Lee, H.-S. Philip Wong (2017). Stanford Virtual-Source Carbon Nanotube Field-Effect Transistors Model. nanoHUB. doi:10.4231/D3BK16Q68.
[Leighton81] F. T. Leighton, "New Lower Bound Techniques for VLSI", in SFCS, 1981.
[Leiserson80] C. E. Leiserson, "Area-Efficient Graph Layouts (for VLSI)", in SFCS, 1980.
[Leskovec14] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford Large Network Dataset Collection", http://snap.stanford.edu/data, 2014.
[Levy10] J. S. Levy et al., "CMOS-compatible multiple-wavelength oscillator for on-chip optical interconnects", Nature photonics, 2010.
[Li10] J. Li et al., "Design Paradigm for Robust Spin-Torque Transfer Magnetic RAM (STT MRAM) From Circuit/Architecture Perspective", IEEE Trans. VLSI, 2010.
[Li13] Y. Li et al., "Self-repair of Uncore Components in Robust System-on-Chips: An OpenSPARC T2 Case Study", IITC, 2013.
[Li15] H. Li et al., "Variation-aware, reliability-emphasized design and optimization of RRAM using SPICE model", in DATE 2015
[Li16] H. Li et al., "Four-layer 3D vertical RRAM integrated with FinFET as a versatile computing unit for brain-inspired cognitive information processing" in Symp. VLSI, 2016.
[Li17] H. Li et al., "Resistive RAM-Centric Computing: Design and Modeling Methodology", IEEE TCAS, 2017.
[Liberate17] Cadence Virtuoso Liberate Characterization Solution (online)https://www.cadence.com/content/cadence-www/global/en_US/home/tools/custom-ic-analog-rf-design/library-characterization/virtuoso-liberate-characterization.html, 2017.
[Lichman13] M. Lichman. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2013.
[Luo17] Q. Luo et al., "8-layers 3D Vertical RRAM with Excellent Scalability towards Storage Class Memory Applications", in IEDM, 2017
[Loubet17] N. Loubet et al., "Stacked Nanosheet Gate-All-Around Transistor to Enable Scaling Beyond FinFET", VLSI Technology, 2017.
[Murugan11] M. Murugan and D.H.C.Du," Rejuvenator: A Static Wear Leveling Algorithm for NAND Flash Memory with Minimized Overhead", in MSST, 2011.
[Mutlu17] O. Mutlu, "The RowHammer Problem", in DATE, 2017.
[Naito10] T. Naito et al. ,"Word's first Monolithic 3D-FPGA with TFT SRAM over 9 layer cu CMOS", VLSI Technology, 2010.
[Naeimi13] H. Naeimi et al., "STTRAM Scaling and Retention Failure", Intel Technology Journal, 2013.
[Neelakantan17] A. Neelakantan et al., "Learning a Natural Language Interface with Neural Programmer", in ICLR 2017.
[Noguchi15] H. Noguchi et al., "A 3.3ns-Access-Time 71.2 uW/MHz 1Mb Embedded STT-MRAM Using Physically Eliminated Read-Disturb Scheme and Normally-Off Memory Architecture", in ISSCC, 2015.
[Ohashi17] T. Ohashi et al., "Variability study with CD-SEM metrology for STT-MRAM: correlation analysis between physical dimensions and electrical property of the memory element", Proc. SPIE, Metrology, Inspection and Process Control, 2017.
[OSPARC] OpenSPARC T2 Processor design (online), http://www.oracle.com/technetwork/systems/opensparc/opensparc-t2-page-1446157.html
[Pal18] S. Pal et al., "A Case for Packageless Processors", in HPCA, 2018.
[Palko16] J. Palko et al., "High heat flux two-phase cooling of electronics with integrated diamond/porous copper heat sinks and microfluidic coolant supply", in ITherm, 2016
[Panth14] S. Panth et al., "Design and CAD methodologies for low-power gate-level monolithic 3D ICs", in ISLPED, 2014.
[Park17] R. Park et al., "Hysteresis-Free Carbon Nanotube Field-Effect Transistors", AcsNano vol. 11, pp 4785-4791, 2017.
[Patil09] N. Patil et al., "Wafer-scale growth and transfer of aligned single-walled carbon nanotubes," *IEEE Trans. Nanotechnol.*, 2009.

[PCM17] Intel performance counter monitor [online] , https://software.intel.com/en-us/articles/intel-performance-counter-monitor, 2017.
[Pharma15] Pharmacogenomics knowledge-base (online), https://www.pharmgkb.org/
[Piggott15] A. Y. Piggott et al., "Inverse design and demonstration of a compact and broadband on-chip wavelength demultiplexer", Nature photonics, 2015.
[Pingalli11] K. Pingali et al, "The Tao of Parallelism in Algorithms", In PLD, 2011.
[Planes12] N. Planes et al., "28nm FDSOI Technology platform for High-Speed Low-Voltage Digital Application", in VLSI, 2012.
[Pop12] E. Pop, V. Varshney, and A.K. Roy, "Thermal Properties of Graphene: Fundamentals and Applications," *MRS Bull.*, vol. 37, no. 12, pp. 1273–1281, 2012.
[Provine14] J. Provine et al., "Advances in RRAM Through Split Manufacturing and Aggressive Scaling", GOMATech, 2014.
[Qiu17] C. Qiu et al., "Scaling carbon nanotube complementary transistors to 5-nm gate lengths", in Science, vol. 355, pp. 271-276, 2017.
[Qureshi09a] M. Qureshi et al., "Scalable High performance main memory system using phase-change memory technology", in ISCA, 2009.
[Qureshi09b] M. Qureshi et al., "Enhancing Lifetime and Security of PCM-Based Main Memory with Start-Gap Wear Leveling", in MICRO, 2009.
[Ramalingam16] S. Ramalingam, "HBM Package Integration: Technology Trends Challegnes and Applications", in HotChips, 2016.
[Rastegari16] M. Rastegari et al., "XOR-Net: Imagenet classification using binary convolutional neural networks", ECCV, 2016.
[Recht11] B. Recht, C. Re, S. Wright S, and F. Niu, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent". In Advances in Neural Information Processing Systems, 2011.
[Rho17] K. Rho et al., "A 4Gb LPDDR2 STT-MRAM with Compact $9F^2$ 1T1MTJ Cell and Hierarchical Bitline Architecture", in ISSCC, 2017.
[Rusu10] S. Rusu et al., "A 45 nm 8-Core Enterprise Xeon® Processor", in JSSCC, 2010.
[Saida16] D. Saida et al., "Sub-3 ns pulse with sub-100 µA switching of 1x-2x nm perpendicular MTJ for high-performance embedded STT-MRAM towards sub-20 nm CMOS", in Symp. VLSI, 2016.
[Sanchez13] D. Sanchez and C. Kozyrakis, "ZSim: Fast and Accurate Microarchitectural Simulation of Thousand-Core Systems", in ISCA, 2013.
[Satish14] N. Satish et al, "Navigating the Maze of Graph Analytics Frameworks using Massive Graph Datasets", In SIGMOD, 2014.
[Shao13] Y. S. Shao and D. Brooks, "Energy Characterization and Instruction-Level Energy Model of Intel's Xeon Phi Processor", in ISLPED, 2013.
[Shazeer17] N Shazeer et al., "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer." arXiv preprint arXiv:1701.06538, 2017.
[Sheu09] S.-S. Sheu et al., "A 5ns Fast Write Multi-Level Non-volatile 1K bits RRAM Memory with Advance Write Scheme", VLSI, 2009.
[Sheu11] S.-S. Sheu et al., "A 4Mb Embedded SLC Resistive-RAM Macro with 7.2ns Read-Write Random-Access Time and 160ns MLC-Access Capability", in ISSCC, 2011.
[Shulaker13a] M. Shulaker et al., "Carbon Nanotube Computer," Nature, vol. 501(7468), pp. 526-530, 2013.
[Shulaker13b] M. Shulaker et al., "Experimental demonstration of a fully digital capacitive sensor interface built entirely using carbon-nanotube FETs", ISSCC, 2013.
[Shulaker14a] M. Shulaker et al., "High-performance carbon nanotube field-effect transistors", in IEDM, 2014.
[Shulaker14b] M. Shulaker et al., "Monolithic 3D integration of logic and memory: Carbon nanotube FETs, resistive RAM, and silicon FETs", in IEDM, 2014.
[Shulaker14c] M. Shulaker et al., "Sensor-to-digital interface built entirely with carbon nanotube FETs." *IEEE Journal of Solid-State Circuits* 49.1, 2014.
[Shulaker14d] M. Shulaker et al., "Monolithic 3D Integration of Logic and Memory: Carbon Nanotube FETs, Resistive RAM, and Silicon FETs", in IEDM 2014
[Shulaker15a] M. Shulaker et al., "Efficient Metallic Carbon Nanotube Removal for Highly-Scaled Technologies," *IEDM*, 2015.
[Shulaker15b] M. Shulaker et al., "Monolithic 3D integration: a path from concept to reality", in DATE 2015.
[Shulaker17] M. Shulaker et al., "Three-dimensional integration of nanotechnologies for computing and data storage on a single chip," Nature, 2017.
[Shum17] D. Shum et al., "CMOS-embedded STT-MRAM Arrays in 2x nm Nodes for GP-MCU applications", in VLSI 2017
[Simonyan15] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", in ICLR, 2015.
[SMT11] Statistical machine translation: 1 Billion words news dataset (online), http://statmt.org/wmt11/training-monolingual-news-2011.tgz
[Song16] Y. J. Song et al., "Highly Funcitonal and Reliable 8Mb STT-MRAM Embedded in 28nm Logic", in IEDM, 2016.
[SPARC15] OpenSPARC T2 SoC (online): http://www.opensparc.net/opensparc-t2
[Spectre17] Cadence Spectre Circuit Simulator (online), https://www.cadence.com/content/cadence-www/global/en_US/home/tools/custom-ic-analog-rf-design/circuit-simulation/spectre-circuit-simulator.html, 2015.
[Sridhar14] Sridhar, A., Vincenzi, A., Atienza, D., and Brunschwiler, T., "3D-ICE: A Compact Thermal Model for Early-Stage Design of Liquid-Cooled ICs", IEEE Transactions on Computers, Vol. 63(10), pp. 2576-2589, 2014.
[Srimani18] T. Srimani et al., "Negative Capacitance Carbon Nanotube FETs", IEEE Electron Device Letters, 2018.
[Strukov16] D. B. Strukov, "Endurance-write-speed tradeoffs in nonvolatile memories", Applied physics, 2016.
[Synopsys17] Synopsys IC Compiler™ (online), https://www.synopsys.com/implementation-and-signoff/physical-implementation/ic-compiler.html
[Szegedy16] C. Szegedy et al. "Rethinking the inception architecture for computer vision." In CVPR, 2016.
[Tomimatsu09] T. Tomimatsu et al., "Cost-Effective 28-nm LSTP CMOS using Gate-First Metal Gate/High-k Technology", in VLSI, 2009.
[Vinyals15] O. Vinyals et al., "Show and Tell: A Neural Image Caption Generator", CVPR, 2015.
[Wachter17] S. Wachter et al., "A Microprocessor based on a Two-dimensional Semiconductor", Nature Comm., vol. 8, 2017.
[Wang10] Wang, Yi, et al. "RNFTL: A reuse-aware NAND flash translation layer for flash memory." ACM Sigplan Notices 45.4 , 2010.



[Wei09] H. Wei et al., "Monolithic three-dimensional integrated circuits using carbon nanotube FETs and interconnects," *IEDM*, pp. 577-580, 2009.
[Wei12] H. Wei et al., "Cooling Three-Dimensional Integrated Circuits using Power Delivery Networks", in IEDM, 2012.
[Wei13] H. Wei et al., "Monolithic three-dimensional integration of carbon nanotube FET complementary logic circuits," *IEDM*, pp. 511-514, 2013.
[Won13] Won, Y., Cho, J., Agonafer, D., Asheghi, M., Goodson, K.E., "Cooling Limits for GaN HEMT Technology," 2013 IEEE Compound Semiconductor IC Symposium, doi: 10.1109/CSICS.2013.6659222 (2013).
[Wong07] S. Wong et al., "Monolithic 3D Integrated Circuits," Symp. on VLSI Technology, Systems and Applications, 2007.
[Wong12] H.-S. P. Wong et al., "Metal-oxide RRAM", Proc. IEEE, 2012.
[Wong15] H.-S. P. Wong and S. Salahuddin, "Memory Leads the Way to Better Computing", Nature, 2015.
[Webscope17] YAHOO! Webscope datasets [online] , https://webscope.sandbox.yahoo.com/, 2017.
[Wu16] S.-Y. Wu et al., "A 7nm CMOS Platform Technology Featuring 4th Generation FinFET Transistors with a 0.027um^2 High Density 6-T SRAM cell for Mobile SoC Applications", in IEDM, 2016.
[Wu18] T. Wu et al., "Brain-Inspired Computing Exploiting Carbon Nanotube FETs and Resistive RAM: Hyperdimensional Computing Case Study", in ISSCC 2018.
[Yang17] Y. Yang et al., "High-Performance Complementary Transistors and Medium-Scale Integrated Circuits Based on Carbon Nanotube Thin Films", AcsNano vol. 11, pp 4124-4132, 2017.
[Yu13] S. Yu et al., "3D vertical RRAM-Scaling Limit Analysis and Demonstration of 3D Array Operation", VLSI-Tech, 2013.
[Yu16] M. Yu et al., "Novel Vertical 3D Structure of TaO_x-based RRAM with Self-localized Switching Region by Sidewall Electrode Oxidation", Nature Scientific Reports, 2016.
[Zhang12] J. Zhang et al., "Carbon Nanotube Robust Digital VLSI," IEEE Trans. TCAD, vol. 31(4), pp. 453-471, 2012.
[Zhang14] C. Zhang and C. Re, "Dimmwitted: A Study of Main-Memory Statistical Analytics", in VLDB, 2014.
[Zhong18a] D. Zhong et al., "Gigahertz integrated circuits based on carbon nanotube films", Nature electronics, 2018
[Zhong18b] D. Zhong et al., "Continuous adjustment of threshold voltage in carbon nanotube field-effect transistors through gate engineering", in Appl. Phys. Lett., 2018
[Zhou09] P. Zhou et al., "A durable and Energy-Efficient Main Memory Using Phase Change Memory Technology", in ISCA, 2009.
[Zoo17] TensorFlow model zoo [online], https://github.com/tensorflow/models, 2017.

Mohamed M. Sabry Aly is an assistant professor at Nanyang Technological University, Singapore. He received his Ph.D. Degree in electrical and computer engineering from École Polytechnique Fédérale de Lausanne (EPFL), in 2013. He was a postdoctoral research fellow at Stanford University. His current research interests include system-level design and optimization of computing systems enabled by emerging technologies. Dr. Aly was the recipient of the Swiss National Science Foundation Early Post-Doctoral Mobility Fellowship in 2013.

Tony F. Wu received the B.S. degree in EE from California Institute of Technology, Pasadena, CA, USA, in 2011, and the M.S. degree in EE from Stanford University, Stanford, CA, USA, in 2013, where he is currently pursuing the Ph.D. degree. His current research interests include design and fabrication of monolithic 3D integrated systems using emerging technologies. He was the recipient of the 2018 Electronics Materials Symposium Ross N. Tucker Award.

Andrew Bartolo is pursuing his Ph.D. in computer science at Stanford University. His work centers on hardware and software support for near-memory computing. His interests span the fields of architecture, compilers, digital design, and networking. He received a B.S. in computer science from Stanford University in 2016.

Yash H. Malviya received his M.S. in Electrical Engineering with specialization in Hardware-Software Systems from Stanford University, 2018. He is currently working in the industry as a Power modelling engineer.

William Hwang received the B.S. degree in EE and the B.S. degree in MSE from the University of Washington, Seattle, WA, USA, in 2015, the M.S. degree in MSE from the University of Washington, Seattle, WA, USA, in 2016, and the M.S. degree in EE from Stanford University, Stanford, CA, USA, in 2018, where he is currently pursuing the Ph.D. degree. His current research interests include energy-efficient computing systems, enabled by monolithic 3D integration of emerging technologies.

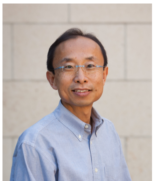
Gage Hills is a post-doctoral researcher at Massachusetts Institute of Technology. He received his Ph.D. from Stanford University in 2018, advised by Prof. Subhasish Mitra and co-advised by Prof. H.-S. Philip Wong. His current research interests include development of very-large-scale integrated circuits using nanotechnologies, such as carbon nanotube field-effect transistors.

Igor Markov received the M.A. degree in mathematics and the Ph.D. degree in Computer Science from UCLA. He is a Professor of Electrical Engineering and Computer Science with the University of Michigan Ann Arbor. His current research interests include applied algorithms, large-scale optimization, computers that make computers, secure and verified hardware design, as well as atomic-scale information processing. He has co-authored five books, four U.S. patents, and over 200 refereed publications. He has supervised 12 doctoral degrees. Prof. Markov was the recipient of the best paper awards at the Design Automation and Test in Europe Conference (DATE), the International Symposium on Physical Design, and the International Conference on Computer-Aided Design (ICCAD), the IEEE CAS Donald O. Pederson Award for best paper in the IEEE Transactions on Computer-Aided Design, Design Automation Conference (DAC) Fellowship, the ACM SIGDA Outstanding New Faculty Award, the ACM SIGDA Technical Leadership Award, the NSF CAREER Award, the IBM Partnership Award, the Synplicity Inc., Faculty Award, the Microsoft A. Richard Newton Breakthrough Research Award, the inaugural IEEE CEDA Early Career Award, and the Electrical Engineering and Computer Science Department Outstanding Achievement Award from the University of Michigan. He is an ACM Distinguished Scientist. He was the Chair of the SLIP and IWLS Workshops, as well as tracks and topic areas including DAC, DATE, ICCAD, ICCD, and GLSVLSI. He has served on the Executive Board of

ACM SIGDA twice. He is currently a Moderator of CoRR. He was an Editorial Board Member of the Communications of ACM, ACM Transactions on Design Automation of Electronic Systems, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, ACM Journal of Emerging Technologies in Computing, and the IEEE DESIGN AND TEST.

Mary Wootters is an assistant professor of Computer Science and Electrical Engineering at Stanford University. She received a PhD in mathematics from the University of Michigan in 2014, and a BA in math and computer science from Swarthmore College in 2008; she was an NSF postdoctoral fellow at Carnegie Mellon University from 2014 to 2016. Her research interests include randomized algorithms, coding theory, dimension reduction, matrix completion, and sparse signal processing.

Max M. Shulaker joined the EECS department at MIT as an assistant professor in July 2016. He received his B.S., M.S., and Ph.D. from Stanford University in Electrical Engineering. During his Ph.D., his research on carbon nanotube-based transistors and circuits resulted in the first digital systems built entirely using carbon nanotube FETs (including the first carbon nanotube microprocessor), the first monolithic three-dimensional integrated circuits combining arbitrary vertical stacking of logic and memory, and the highest performance and highly-scaled carbon nanotube transistors to-date.



H.-S. Philip Wong is the Willard R. and Inez Kerr Bell Professor in the School of Engineering. He joined Stanford University as Professor of Electrical Engineering in September 2004. From 1988 to 2004, he was with the IBM T.J. Watson Research Center.
At IBM, he held various positions from Research Staff Member to Senior Manager. While he was Senior Manager, he had the responsibility of shaping and executing IBM’s strategy on nanoscale science and technology as well as exploratory silicon devices and semiconductor technology.
During his time at IBM, he managed pathfinding research on high-k/metal gate, strained silicon, alternative channel materials such as Ge and III-V, multi-gate FinFET, ultra-thin SOI – many of these have now become product technology at various companies.
Professor Wong’s research aims at translating discoveries in science into practical technologies. His works have contributed to advancements in nanoscale science and technology, semiconductor technology, solid-state devices, and electronic imaging. His present research covers a broad range of topics including carbon electronics, 2D layered materials, wireless implantable biosensors, directed self-assembly, device modeling, brain-inspired computing, non-volatile memory, and monolithic 3D integration.
He is a Fellow of the IEEE. He served as the Editor-in-Chief of the IEEE Transactions on Nanotechnology (2005 – 2006), sub-committee Chair of the ISSCC (2003 – 2004), General Chair of the IEDM (2007), and is currently the Chair of the IEEE Executive Committee of the Symposia of VLSI Technology and Circuits. He is the faculty director of the Stanford Non-Volatile Memory Technology Research Initiative (NMTRI), and is the founding Faculty Co-Director of the Stanford SystemX Alliance – an industrial affiliate program focused on building systems.

Subhasish Mitra Subhasish Mitra is Professor of Electrical Engineering and of Computer Science at Stanford University, where he directs the Stanford Robust Systems Group and co-leads the Computation focus area of the Stanford SystemX Alliance. He is also a faculty member of the Stanford Neurosciences Institute. Prof. Mitra holds the Carnot Chair of Excellence in Nanosystems at CEA-LETI in Grenoble, France. Before joining the Stanford faculty, he was a Principal Engineer at Intel Corporation.
Prof. Mitra's research interests range broadly across robust computing, nanosystems, VLSI design, validation, test and electronic design automation, and neurosciences. He, jointly with his students and collaborators, demonstrated the first carbon nanotube computer and the first three-dimensional nanosystem with computation immersed in data storage. These demonstrations received wide-spread recognitions (cover of NATURE, Research Highlight to the United States Congress by the National Science Foundation, highlight as "important, scientific breakthrough" by the BBC, Economist, EE Times, IEEE Spectrum, MIT Technology Review, National Public Radio, New York Times, Scientific American, Time, Wall Street Journal, Washington Post and numerous others worldwide). His earlier work on X-Compact test compression has been key to cost-effective manufacturing and high-quality testing of almost all electronic systems. X-Compact and its derivatives have been implemented in widely-used commercial Electronic Design Automation tools.
Prof. Mitra's honors include the ACM SIGDA/IEEE CEDA A. Richard Newton Technical Impact Award in Electronic Design Automation (a test of time honor), the Semiconductor Research Corporation's Technical Excellence Award, the Intel Achievement Award (Intel’s highest corporate honor), and the Presidential Early Career Award for Scientists and Engineers from the White House (the highest United States honor for early-career outstanding scientists and engineers). He and his students published several award-winning papers at major venues: ACM/IEEE Design Automation Conference, IEEE International Solid-State Circuits Conference, ACM/IEEE International Conference on Computer-Aided Design, IEEE International Test Conference, IEEE Transactions on CAD, IEEE VLSI Test Symposium, and the Symposium on VLSI Technology. At Stanford, he has been honored several times by graduating seniors "for being important to them during their time at Stanford."
Prof. Mitra served on the Defense Advanced Research Projects Agency's (DARPA) Information Science and Technology Board as an invited member. He is a Fellow of the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE).

APPENDIX A. MEMORY SIZES IN CPU-BASED SYSTEMS

We investigate the cache and main memory capacity of a wide range of computing systems, from mobile to datacenter platforms, and summarize their properties in Table A.1. We illustrate the trend in memory capacity for these systems with respect to the number of cores in Figure A.1, clearly indicating that capacities of the cache and main memories are at least 1 MByte and 1 GByte per core, respectively. In N3XT, system configurations with larger memory capacity per core would have greater improvements than the cases shown in Section V. Baseline systems do not have more connections to memory, but the compute-to-memory interconnect in N3XT can support this increase in memory capacity with more concurrent accesses.

Table A.1. Summary of the examined CPU-based systems highlighting the number of processor cores, cache and main memory capacity.

Computing systems	Num. cores	Cache cap. (MBytes)	Main mem. cap. (GBytes)
Apple a10 [A1]	2	3	3
Apple a9x [A2]	2	3	4
Yoga 900s [A3]	4	4	8
Thinkpad T [A4]	4	4	20
Macbook pro [A5]	4	8	16
IBM Power S812LC [A6]	8	64	32
Ideapad Y900 [A7]	8	8	16
IBM power E850 [A8]	48	384	128
IBM Watson [A9]	2,880	24,000	16,384
Piz Daint [A10]	206,720	516,800	194,560
Oakforest [A11]	556,104	278,052	919,296
Titan [A12]	560,640	560,640	710,144
Cori [A13]	622,336	311,168	878,592
K computer [A14]	705,024	528,768	1,410,048
Sequoia [A15]	1.60×10^{6}	1.28×10^{7}	1,677,722
Sunway Taihu [A16]	1.1×10^{7}	665,600	1,310,720

APPENDIX B. CONFIGURATIONS OF CPU-BASED SYSTEMS

We evaluate 2- to 64-core architectures for both baseline and N3XT and summarize their key parameters Table A.2. In all configurations, the processor cores and L1 caches use the same parameters illustrated in Table 1. Since the architectural simulator uses ×86 Instruction-set architecture (*ISA*), we use hardware measurements of the Intel Xeon Phi (fabricated at 22 nm node) to estimate the processor core energy and frequency reported in the literature [Shao13]. For N3XT we increase the frequency by 3× and reduce the energy consumption by 3×, which are the same scaling values observed after performing the physical design of the OpenSPARC T2 using a foundry SiFET and the CNFET PDK at 28 nm.

We use LPDDR3 DRAM interface for the 2-, 4-, and 8-core configurations in 2D baseline to represent embedded and mobile architectures.

Table A.2. Configurations of the evaluated CPU-based systems for both 2D baseline and N3XT.

Parameter	2D baseline	N3XT
2-core configuration		
Main memory + compute-to-memory interconnect	2 GBytes DRAM 1 controller, 1 32-bit channel/controller LPPDR3 interface (0.8 GHz), FR-FCFS scheduling 75/70 ns rd/wt 15 pJ/bit rd/wt	2 GBytes 3D RRAM (CNFET access circuits) 2 memory controllers (4GHz), 16 512-bits channels per controller with simple interface, FCFS scheduling Meshes-of-trees interconnect 1.5/11.3 ns rd/wt 0.4/0.9 pJ/bit rd/wt
L2 Cache	2 MBytes SRAM 11 ns rd/wt 0.37 pJ/bit rd/wt	2 MBytes STTRAM (CNFET access circuits) 1.5/3.6 ns rd/wt 0.1/0.5 pJ/bit rd/wt
4-core configuration		
Main memory + compute-to-memory interconnect	4 GBytes DRAM 1 controller, 2 32-bit channels/controller LPPDR3 interface (0.8 GHz), FR-FCFS scheduling 75/70 ns rd/wt 15 pJ/bit rd/wt	4 GBytes 3D RRAM (CNFET access circuits) 4 memory controllers (4GHz), 16 512-bits channels per controller with simple interface, FCFS scheduling Meshes-of-trees interconnect 1.5/11.3 ns rd/wt 0.5/0.96 pJ/bit rd/wt
L2 Cache	4 MBytes SRAM 11 ns rd/wt 0.4 pJ/bit rd/wt	4 MBytes STTRAM (CNFET access circuits) 1.5/3.6 ns rd/wt 0.11/0.5 pJ/bit rd/wt
8-core configuration		
Main memory + compute-to-memory interconnect	8 GBytes DRAM 2 controllers, 2 32-bit channels/controller LPPDR3 interface (0.8 GHz), FR-FCFS scheduling 75/70 ns rd/wt 15 pJ/bit rd/wt	8 GBytes 3D RRAM (CNFET access circuits) 8 memory controllers (4GHz), 16 512-bits channels per controller with simple interface, FCFS scheduling Meshes-of-trees interconnect 2/11.5 ns rd/wt 0.6/1.0 pJ/bit rd/wt
L2 Cache	8 MBytes SRAM 11.7 ns rd/wt 0.5pJ/bit rd/wt	8 MBytes STTRAM (CNFET access circuits) 1.7/3.7 ns rd/wt 0.14/0.52 pJ/bit rd/wt
16-core configuration		
Main memory + compute-to-memory interconnect	16 GBytes DRAM 2 controllers, 1 64-bit channel/controller DDR4 interface (1.2GHz), FR-FCFS scheduling 65/60 ns rd/wt 45 pJ/bit rd/wt	16 GBytes 3D RRAM (CNFET access circuits) 16 memory controllers (4GHz), 16 512-bits channels per controller with simple interface, FCFS scheduling Meshes-of-trees interconnect 4/12.5 ns rd/wt 0.66/1.0 pJ/bit rd/wt
L2 Cache	16 MBytes SRAM 12.5 ns rd/wt 0.65 pJ/bit rd/wt	16 MBytes STTRAM (CNFET access circuits) 1.8/3.7 ns rd/wt 0.16/0.55 pJ/bit rd/wt
32-core configuration		
Main memory + compute-to-memory interconnect	32 GBytes DRAM 4 controllers, 1 64-bit channel/controller DDR4 interface (1.2GHz), FR-FCFS scheduling 65/60 ns rd/wt 45 pJ/bit rd/wt	32 GBytes 3D RRAM (CNFET access circuits) 32 memory controllers (4GHz), 16 512-bits channels per controller with simple interface, FCFS scheduling Meshes-of-trees interconnect 4.5/12.8 ns rd/wt 0.7/1.1 pJ/bit rd/wt
L2 Cache	32 MBytes SRAM 13.2 ns rd/wt 0.8 pJ/bit rd/wt	32 MBytes STTRAM (CNFET access circuits) 1.85/3.9 ns rd/wt 0.18/0.59 pJ/bit rd/wt

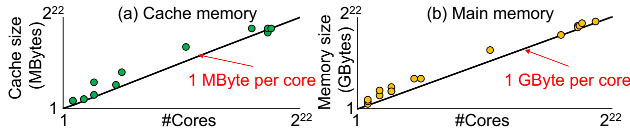


Figure A.1. (a) Cache and (b) memory size observed in existing architectures (mobile, desktop, server, and HPC) spanning a wide range of processor core. The trend shows a ratio of 1 MByte of cache and 1 GByte of main memory per core is retained among different classes.

APPENDIX C. CPU-BASED SYSTEMS RESULTS

We show the application-level runtime, energy benefits and their product, as well as the corresponding runtime and energy consumption breakdown for all evaluated workload-dataset pairs in Table A.3, Table A.4 and Table A.5.

Table A.3. Benefits and application-level breakdown results for the examined workloads (Part 1 of 3). Time refers to application runtime, T×E is system-level EDP. Proc.=processor

Workload	Dataset	Num. cores	Arch. type	Benefits			Time breakdown			Energy breakdown			
				Time	Energy	T×E	Proc. active	Mem. access	Total	Proc. active	Proc. idle	Mem. access	Total
Linear regression (training)	RCV	2	2D Baseline	6.95×	9.25×	**64.3×**	26.58%	73.420%	100%	21.414%	24.167%	54.419%	100%
			N3XT				8.628%	5.750%	14.4%	6.530%	1.127%	3.158%	10.814%
	URL	4	2D Baseline	7.18×	10.09×	**72.4×**	28.0%	71.998%	100%	20.844%	21.934%	57.223%	100%
			N3XT				9.117%	4.811%	13.9%	6.357%	0.880%	2.675%	9.912%
	HIGGS	8	2D Baseline	7.41×	10.42×	**77.2×**	27.7%	72.3%	100%	22.631%	24.204%	53.165%	100%
			N3XT				9.008%	4.494%	13.5%	6.939%	0.904%	1.755%	9.597%
	YAHOO	16	2D Baseline	8.73×	18.92×	**165×**	15.5%	84.5%	100%	10.575%	11.760%	77.665%	100%
			N3XT				5.004%	6.449%	11.5%	3.226%	0.541%	1.518%	5.285%
	Weather	64	2D Baseline	7.13×	11.95×	**85.2×**	18.00%	82.000%	100%	26.138%	40.514%	33.348%	100%
			N3XT				8.391%	5.634%	14%	5.531%	1.757%	1.078%	8.367%
	HBP	64	2D Baseline	19.53×	31.34×	**612×**	3.599%	96.401%	100%	7.157%	44.402%	48.441%	100%
			N3XT				1.786%	3.334%	5.1%	1.889%	0.753%	0.549%	3.191%
Linear regression (inference)	RCV	2	2D Baseline	5.41×	7.43×	**40.3×**	41.5%	58.5%	100%	23.814%	31.345%	44.841%	100%
			N3XT				13.5%	4.979%	18.5%	7.357%	2.612%	3.483%	13.452%
	URL	4	2D Baseline	5.60×	8.35×	**46.8×**	42.33%	57.67%	100%	15.549%	38.390%	46.061%	100%
			N3XT				13.74%	4.128%	17.9%	4.747%	4.128%	3.096%	11.971%
	HIGGS	8	2D Baseline	5.02×	7.03×	**35.3×**	62.00%	38.00%	100%	13.791%	57.614%	28.595%	100%
			N3XT				18.14%	1.795%	19.9%	3.796%	6.563%	3.857%	14.216%
	YAHOO	16	2D Baseline	8.38×	18.52×	**155×**	19.68%	80.32%	100%	3.374%	29.513%	67.114%	100%
			N3XT				6.402%	5.531%	11.9%	1.030%	2.024%	2.344%	5.398%
	Weather	64	2D Baseline	8.74×	18.50×	**162×**	11.49%	88.51%	100%	1.731%	75.418%	22.852%	100%
			N3XT				5.722%	5.722%	11.4%	0.367%	4.233%	0.804%	5.404%
	HBP	64	2D Baseline	7.21×	15.55×	**112×**	15.79%	84.21%	100%	1.151%	76.047%	22.803%	100%
			N3XT				7.920%	5.958%	13.9%	0.243%	5.221%	0.969%	6.433%
Logistic regression (training)	RCV	2	2D Baseline	6.79×	8.43×	**57.2×**	29.93%	70.07%	100%	24.595%	23.422%	51.983%	100%
			N3XT				9.325%	5.404%	14.73%	7.493%	1.069%	3.305%	11.867%
	URL	4	2D Baseline	7.09×	9.81×	**69.6×**	29.98%	70.02%	100%	22.39%	20.98%	56.63%	100%
			N3XT				9.88%	4.23%	14.11%	6.83%	0.76%	2.60%	10.189%
	HIGGS	8	2D Baseline	6.08×	8.48×	**51.6×**	38.01%	61.99%	100%	30.73%	20.26%	49.01%	100%
			N3XT				12.42%	4.03%	16.45%	9.36%	0.79%	1.64%	11.794%
	YAHOO	16	2D Baseline	8.50×	18.39×	**156×**	16.00%	84.00%	100%	10.97%	11.70%	77.33%	100%
			N3XT				5.18%	6.59%	11.76%	3.35%	0.54%	1.55%	5.438%
	Weather	64	2D Baseline	6.62×	11.28×	**75×**	19.40%	80.60%	100%	27.64%	39.45%	32.92%	100%
			N3XT				8.90%	6.19%	15.10%	5.85%	1.85%	1.17%	8.869%
	HBP	64	2D Baseline	18.42×	32.64×	**601×**	4.14%	95.86%	100%	8.16%	43.85%	47.99%	100%
			N3XT				2.06%	3.37%	5.43%	1.76%	0.76%	0.54%	3.064%
Logistic regression (inference)	RCV	2	2D Baseline	5.66×	7.50×	**42.5×**	40.00%	60.00%	100%	24.50%	31.04%	44.46%	100%
			N3XT				13.06%	4.59%	17.65%	7.47%	2.35%	3.52%	13.339%
	URL	4	2D Baseline	5.66×	8.75×	**49.6×**	41.00%	59.00%	100%	16.39%	38.37%	45.24%	100%
			N3XT				13.59%	4.08%	17.66%	5.00%	3.46%	2.96%	11.426%
	HIGGS	8	2D Baseline	4.71×	6.50×	**30.7×**	59.51%	40.49%	100%	14.71%	56.49%	28.80%	100%
			N3XT				19.30%	1.91%	21.21%	4.49%	6.79%	4.10%	15.375%
	YAHOO	16	2D Baseline	8.61×	18.89×	**163×**	18.50%	81.50%	100%	3.39%	29.39%	67.22%	100%
			N3XT				6.01%	5.61%	11.62%	1.04%	1.96%	2.29%	5.294%
	Weather	64	2D Baseline	8.77×	18.54×	**163×**	11.00%	89.00%	100%	1.73%	75.42%	22.85%	100%
			N3XT				5.70%	5.70%	11.41%	0.37%	4.22%	0.80%	5.395%
	HBP	64	2D Baseline	7.25×	15.57×	**113×**	15.40%	84.60%	100%	1.25%	75.92%	22.83%	100%
			N3XT				7.59%	6.21%	13.80%	0.26%	5.19%	0.96%	6.421%

Table A.4. Benefits and application-level breakdown results for the examined workloads (Part 2 of 3). Time refers to application runtime, T×E is system-level EDP. Proc.=processor

Workload	Dataset	Num. cores	Arch. type	Benefits			Time breakdown			Energy breakdown			
				Time	Energy	T×E	Proc. active	Mem. access	Total	Proc. active	Proc. idle	Mem. access	Total
Support vector machines (training)	RCV	2	2D Baseline	6.43×	8.35×	**54×**	31.26%	68.74%	100%	25.183%	22.599%	52.218%	100%
			N3XT				10.16%	5.408%	15.56%	7.682%	1.070%	3.217%	11.97%
	URL	4	2D Baseline	7.25×	9.15×	**66×**	32.50%	67.50%	100%	24.13%	20.55%	55.33%	100%
			N3XT				10.56%	3.24%	13.80%	7.36%	0.84%	2.73%	10.93%
	HIGGS	8	2D Baseline	6.77×	9.50×	**64×**	33.00%	67.00%	100%	26.74%	21.96%	51.30%	100%
			N3XT				10.80%	3.98%	14.78%	8.16%	0.78%	1.59%	10.53%
	YAHOO	16	2D Baseline	8.11×	16.72×	**136×**	18.70%	81.30%	100%	12.73%	11.30%	75.97%	100%
			N3XT				6.07%	6.26%	12.33%	3.88%	0.52%	1.58%	5.981%
	Weather	64	2D Baseline	6.22×	10.34×	**64×**	20.99%	79.01%	100%	32.08%	37.10%	30.83%	100%
			N3XT				10.50%	5.57%	16.08%	6.77%	1.73%	1.16%	9.667%
	HBP	64	2D Baseline	18.14×	36.57×	**663×**	4.49%	95.51%	100%	8.80%	43.45%	47.75%	100%
			N3XT				2.24%	3.28%	5.51%	1.48%	0.74%	0.51%	2.735%
Support vector machines (inference)	RCV	2	2D Baseline	5.37×	7.37×	**39.6×**	42.11%	57.89%	100%	24.76%	30.73%	44.51%	100%
			N3XT				13.73%	4.89%	18.61%	7.55%	2.58%	3.43%	13.56%
	URL	4	2D Baseline	5.61×	8.65×	**48.5×**	42.89%	57.11%	100%	16.53%	37.19%	46.28%	100%
			N3XT				13.96%	3.87%	17.8%	5.08%	3.44%	3.03%	11.55%
	HIGGS	8	2D Baseline	5.00×	7.01×	**35×**	63.01%	36.99%	100%	14.10%	57.37%	28.53%	100%
			N3XT				18.28%	1.73%	20.01%	3.89%	6.53%	3.85%	14.27%
	YAHOO	16	2D Baseline	8.34×	18.18×	**151×**	18.99%	81.01%	100%	3.71%	29.33%	66.96%	100%
			N3XT				6.36%	5.63%	11.99%	1.13%	2.02%	2.35%	5.500%
	Weather	64	2D Baseline	8.74×	18.36×	**160×**	10.97%	89.03%	100%	1.97%	75.22%	22.80%	100%
			N3XT				5.72%	5.72%	11.44%	0.42%	4.23%	0.80%	5.446%
	HBP	64	2D Baseline	7.15×	15.92×	**114×**	15.90%	84.10%	100%	1.23%	75.96%	22.81%	100%
			N3XT				7.83%	6.15%	13.99%	0.26%	5.26%	0.76%	6.280%
CNNs (training)	VGGNet-19	32	2D Baseline	10.55×	32.1×	**338×**	9%	91%	100%	4.625%	52.05%	43.325%	100%
			N3XT				4.815%	4.667%	9.481%	0.842%	1.95%	0.32%	3.115%
	Inception-v4	16	2D Baseline	3.5×	12.2×	**42.5×**	36.98%	63.02%	100%	22.613%	17.22%	60.167%	100%
			N3XT				22.95%	5.737%	28.7%	4.908%	1.780%	1.510%	8.198%
CNNs (inference)	VGGNet-19	2	2D Baseline	5.02×	7.03×	**35.28×**	62.00%	38.00%	100%	13.791%	57.614%	28.595%	100%
			N3XT				18.14%	1.795%	19.9%	3.796%	6.563%	3.857%	14.21%
	Inception-v4	2	2D Baseline	9.78×	26.7×	**288×**	11.00%	89.00%	100%	6.623%	42.072%	51.305%	100%
			N3XT				5.723%	3.559%	9.282%	1.09%	1.313%	1.341%	3.744%
LSTM (training)	Neural programmer	16	2D Baseline	9.41×	15.3×	**144×**	16.19%	83.81%	100%	24.021%	42.261%	33.718%	100%
			N3XT				7.862%	2.762%	10.6%	5.082%	1.159%	0.294%	6.53%
	Language Model	64	2D Baseline	17.3×	40.1×	**691×**	7.31%	92.69%	100%	2.778%	66.521%	30.701%	100%
			N3XT				3.658%	2.141%	5.799%	0.586%	1.715%	0.194%	2.494%
LSTM (inference)	Neural programmer	2	2D Baseline	6.5×	13.12×	**85.7×**	20.81%	79.19%	100%	15.766%	63.770%	20.465%	100%
			N3XT				10.40%	4.918%	15.32%	3.312%	4.129%	0.177%	7.618%
	Language Model	16	2D Baseline	19.7×	37.3×	**734×**	3.490%	96.51%	100%	3.280%	56.962%	39.759%	100%
			N3XT				1.534%	3.541%	5.075%	0.690%	1.223%	0.771%	2.684%
PageRank	Web-Google	2	2D Baseline	4.2×	6.1×	**26×**	68.01%	31.99%	100%	46.167%	9.163%	44.670%	100%
			N3XT				21.90%	1.637%	23.54%	14.062%	0.282%	1.938%	16.28%
	Patent	2	2D Baseline	4.1×	5.7×	**23.9×**	74.91%	25.09%	100%	50.133%	9.019%	40.849%	100%
			N3XT				22.94%	1.219%	24.16%	14.854%	0.398%	2.095%	17.34%
	Pokec	4	2D Baseline	6.1×	10.4×	**63×**	42.45%	57.55%	100%	24.218%	13.726%	62.055%	100%
			N3XT				13.83%	2.632%	16.46%	7.456%	0.359%	1.761%	9.576%
	LiveJournal	8	2D Baseline	4.8×	7.5×	**36.4×**	58.03%	41.97%	100%	35.915%	11.021%	53.064%	100%
			N3XT				18.6%	2.160%	20.76%	10.974%	0.333%	1.924%	13.23%
	Orkut	16	2D Baseline	5.8×	10×	**58×**	47.51%	52.49%	100%	27.234%	13.219%	59.547%	100%
			N3XT				14.98%	2.26%	17.25%	8.170%	0.392%	1.438%	10.00%
	EU-2015	32	2D Baseline	15.8×	31×	**476.2×**	8.64%	91.36%	100%	14.072%	34.325%	51.603%	100%
			N3XT				4.319%	2.021%	6.340%	2.812%	0.135%	0.365%	3.312%
	UK-2005	64	2D Baseline	18.3×	26.1×	**477.6×**	8.000%	92.00%	100%	14.642%	39.817%	45.541%	100%
			N3XT				4.034%	1.434%	5.468%	3.150%	0.321%	0.358%	3.829%
	IT-2004	64	2D Baseline	16.2×	23.9×	**388.4×**	9.000%	91.00%	100%	15.333%	40.871%	43.796%	100%
			N3XT				4.515%	1.647%	6.162%	3.263%	0.484%	0.432%	4.179%
	Twitter	64	2D Baseline	21.9×	38.9×	**852.6×**	4.9%	95.1%	100%	8.961%	40.027%	51.012%	100%
			N3XT				2.448%	2.120%	4.568%	1.792%	0.178%	0.597%	2.568%
	Friendster	64	2D Baseline	17.8×	34.3×	**388.4×**	4.623%	95.38%	100%	8.931%	42.286%	48.783%	100%
			N3XT				2.381%	3.232%	5.614%	1.786%	0.559%	0.573%	2.919%

Table A.5. Benefits and application-level breakdown results for the examined workloads (Part 3 of 3). Time refers to application runtime, T×E is system-level EDP. Proc.=processor

Workload	**Dataset**	**Num. cores**	**Arch. type**	**Benefits**			**Time breakdown**			**Energy breakdown**			
				Time	**Energy**	**T×E**	**Proc. active**	**Mem. access**	**Total**	**Proc. active**	**Proc. idle**	**Mem. access**	**Total**
Breadth first search	Web-Google	2	2D Baseline	4.5×	5.8×	**26×**	63.17%	36.83%	100%	44.353%	10.683%	16.066%	100%
			N3XT				20.42%	1.874%	22.3%	13.516%	0.325%	1.753%	17.24%
	LiveJournal	8	2D Baseline	5.5×	7.3×	**40.2×**	48.23%	51.78%	100%	33.783%	15.099%	19.418%	100%
			N3XT				15.56%	2.595%	18.16%	10.302%	0.460%	1.673%	13.72%
	EU-2015	32	2D Baseline	5.2×	7.3×	**37.7×**	50.76%	49.24%	100%	36.460%	14.699%	20.353%	100%
			N3XT				16.36%	2.870%	19.23%	11.105%	0.516%	1.891%	13.78%
	UK-2005	64	2D Baseline	3.7×	7.4×	**27.7×**	51.32%	48.68%	100%	59.081%	13.455%	15.732%	100%
			N3XT				25.14%	1.746%	26.9%	12.459%	0.243%	0.730%	13.45%
	IT-2004	64	2D Baseline	3.7×	7.4×	**27.3×**	51.44%	48.56%	100%	59.473%	13.358%	15.662%	100%
			N3XT				25.28%	1.790%	27.08%	12.501%	0.256%	0.751%	13.51%
	Twitter	64	2D Baseline	9.8×	18.4×	**180.7×**	22.87%	77.13%	100%	31.044%	27.447%	28.907%	100%
			N3XT				7.066%	3.124%	10.2%	4.344%	0.579%	0.861%	5.431%
Single-source shortest path	UK-2005	64	2D Baseline	18.5×	34.4×	**637.4×**	4.590%	95.410%	100%	9.214%	44.041%	46.745%	100%
			N3XT				2.434%	2.966%	5.4%	1.843%	0.518%	0.543%	2.9%
	IT-2004	64	2D Baseline	20.8×	36.9×	**766×**	4.433%	95.567%	100%	8.854%	43.377%	47.769%	100%
			N3XT				2.308%	2.503%	4.8%	1.770%	0.443%	0.500%	2.7%
	Twitter	64	2D Baseline	24.1×	48.2×	**1159×**	5.002%	94.998%	100%	9.430%	40.880%	49.690%	100%
			N3XT				1.927%	2.228%	4.2%	1.320%	0.352%	0.404%	2.1%
	Friendster	64	2D Baseline	13.3×	26.2×	**348.4×**	6.900%	93.100%	100%	12.852%	39.811%	47.338%	100%
			N3XT				3.891%	3.617%	7.5%	2.572%	0.552%	0.699%	3.8%
Connected components	Web-Google	2	2D Baseline	4.6×	5.5×	**25×**	60.56%	39.44%	100%	46.075%	6.175%	47.751%	100%
			N3XT				19.69%	2.01%	21.8%	14.058%	0.202%	3.791%	18.1%
	Patent	2	2D Baseline	4.4×	5.3×	**23.5×**	68.14%	31.86%	100%	48.224%	5.558%	46.218%	100%
			N3XT				21.27%	1.42%	22.69%	14.800%	0.166%	3.806%	18.77%
	Pokec	4	2D Baseline	4.9×	5.8×	**28.6×**	56.72%	43.283%	100%	43.329%	6.977%	49.694%	100%
			N3XT				18.23%	2.224%	20.46%	13.212%	0.219%	3.681%	17.11%
	Live Journal	8	2D Baseline	4.4×	5.2×	**22.9×**	67.13%	32.868%	100%	49.350%	5.567%	45.083%	100%
			N3XT				21.13%	1.635%	22.76%	15.047%	0.173%	3.929%	19.15%
	Orkut	16	2D Baseline	4.9×	6.4×	**31.6×**	57.5%	42.5%	100%	42.552%	7.109%	50.339%	100%
			N3XT				18.18%	2.114%	20.29%	13.060%	0.219%	2.318%	15.6%
	EU-2015	32	2D Baseline	12×	22.9×	**275×**	6.900%	93.100%	100%	12.852%	39.811%	47.338%	100%
			N3XT				4.210%	4.116%	8.327%	2.904%	0.655%	0.810%	4.368%
	UK-2005	64	2D Baseline	18.4×	34.13×	**627×**	4.608%	95.392%	100%	9.097%	43.075%	47.828%	100%
			N3XT				2.368%	3.072%	5.441%	1.824%	0.546%	0.559%	2.930%
	IT-2004	64	2D Baseline	18.4×	35.2×	**648×**	4.679%	95.321%	100%	8.872%	41.739%	49.389%	100%
			N3XT				2.405%	3.026%	5.431%	1.775%	0.515%	0.551%	2.841%
	Twitter	64	2D Baseline	17.1×	33.8×	**577×**	4.593%	95.407%	100%	8.892%	42.335%	48.773%	100%
			N3XT				2.516%	3.335%	5.851%	1.806%	0.552%	0.604%	2.962%
PARSEC	Black-Scholes	64	2D Baseline	2.1×	6.4×	**13×**	93.73%	6.27%	100%	32.89%	50.65%	16.46%	100%
			N3XT				46.87%	0.26%	47.13%	8.22%	7.02%	0.40%	15.65%
	Body track	64	2D Baseline	2.6×	6.1×	**16×**	64.93%	35.07%	100%	69.88%	17.69%	12.43%	100%
			N3XT				33.89%	4.30%	38.19%	13.98%	1.67%	0.85%	16.50%
	Canneal	64	2D Baseline	3.3×	11×	**36×**	54.87%	45.13%	100%	3.02%	76.61%	20.37%	100%
			N3XT				27.52%	2.85%	30.37%	0.76%	7.81%	0.14%	8.70%
	Dedup	64	2D Baseline	5.2×	6.6×	**34×**	35.43%	64.57%	100%	53.25%	26.91%	19.84%	100%
			N3XT				17.71%	1.47%	19.18%	13.31%	0.90%	0.74%	14.95%
	Ferret	64	2D Baseline	2.1×	6×	**12×**	87.74%	12.26%	100%	39.87%	39.62%	20.51%	100%
			N3XT				43.87%	2.99%	46.86%	9.96%	5.31%	0.64%	15.91%
	Fluid animate	64	2D Baseline	6.6×	8.2×	**54×**	17.51%	82.49%	100%	31.70%	35.81%	32.49%	100%
			N3XT				8.76%	6.33%	15.09%	6.33%	3.35%	2.51%	12.19%
	Ray trace	64	2D Baseline	2.2×	8.9×	**20×**	61.88%	38.12%	100%	8.72%	71.88%	19.41%	100%
			N3XT				30.94%	13.37%	44.31%	2.18%	8.90%	0.18%	11.26%
	Stream cluster	64	2D Baseline	4.1×	8.5×	**35×**	29.94%	70.06%	100%	37.44%	27.31%	35.25%	100%
			N3XT				14.97%	9.17%	24.14%	9.36%	1.38%	1.01%	11.76%
	Swaptions	64	2D Baseline	4.9×	13.3×	**65×**	14.12%	85.88%	100%	26.77%	37.71%	35.52%	100%
			N3XT				7.06%	13.20%	20.26%	6.69%	0.29%	0.52%	7.51%
	X264	64	2D Baseline	2.9×	7.8×	**22×**	43.08%	56.92%	100%	30.59%	50.06%	19.35%	100%
			N3XT				21.54%	13.47%	35.01%	7.65%	4.50%	0.58%	12.73%
FFT		64	2D Baseline	10×	15×	**150×**	13.39%	86.61%	100%	23.561%	35.880%	40.559%	100%
			N3XT				6.693%	3.298%	9.991%	4.976%	0.822%	0.841%	6.639%

APPENDIX D. CONFIGURATIONS FOR THERMAL SIMULATIONS

We summarize in Table A 6 the thermal simulation parameters for both baseline and N3XT configurations.

Table A 6. Thermal configuration parameters for both Baseline and N3XT configurations

	Parameter	Value
Baseline	Silicon thermal conductivity	130 W/(m.k)
	Silicon specific heat capacitance	1.6×10^6 J/(m^3.k)
	Back-end-of-line thermal conductivity	12.25 W/(m.k)
	Back-end-of-line specific heat capacitance	1.5×10^6 J/(m^3.k)
	Die thickness	50 µm
N3XT	Substrate thickness	20 µm
	Substrate thermal conductivity (silicon)	130 W/(m.k)
	Substrate specific heat capacitance (silicon)	1.6×10^6 J/(m^3.k)
	CNT-to-metal contact thermal conductivity [A17]	3000 W/(m.k)
	CNT-to-dielectric contact thermal conductivity [A17]	0.2 W/(m.k)
	CNFET layer thickness	35 nm
	ILV thermal conductivity	400 w/(m.k)
	ILV specific heat capacitance	3.85×10^6 J/(m^3.k)
	Interlayer dielectric thermal conductivity	1.1 W/(m.k)
	Interlayer dielectric specific heat capacitance	1.5×10^6 J/(m^3.k)
	Interlayer thickness	1µm
Heat sink	Thermal conductance	2×10^4 W/(m^2.k)

APPENDIX E. CONFIGURATIONS FOR 2.5D AND 3D-TSV SYSTEMS

We summarize in Table A 7 and Table A 8 the architecture specifications of 2.5D- and 3D-TSV-based configurations CPU-based and accelerator-based systems, respectively.

Table A 7. Configurations of the evaluated CPU-based systems for both 2.5D and 3D TSV.

	2.5D	3D TSV
Main memory	64 GByte off-chip DRAM connected via 2.5 interposer integration 40 µm microbump pitch 16 controllers, 1 128-bit channel per controller HBM interface (1 GHz), FR-FCFS scheduling 51/55 ns read/write (15ns minimum latency) 12 pJ/bit read/write	64 GByte on-chip 3D-stacked DRAM using TSVs 5 µm TSV pitch 64 controllers, 1 128-bit channel per controller HBM interface (1 GHz), FR-FCFS scheduling 51/55 ns read/write (10ns minimum latency) 9 pJ/bit read/write
L2 shared cache (8-way set associative)	64 MByte silicon SRAM 14 ns read/write 1 pJ/bit read/write	64 MByte silicon SRAM 14 ns read/write 1 pJ/bit read/write
L1 local data cache (8-way set associative)	32 KByte silicon SRAM per processor core 2.3 ns read/write 0.2 pJ/bit read/write	32 KByte silicon SRAM per processor core 2.3 ns read/write 0.2 pJ/bit read/write
L1 local instruction cache (4-way set associative)	32 KByte silicon SRAM per processor core 1.5 ns read/write 0.17 pJ/bit read/write	32 KByte silicon SRAM per processor core 1.5 ns read/write 0.17 pJ/bit read/write
Processor cores	64 in-order processor cores 1.3 GHz clock speed 0.5 nJ/instruction [Shao13]	64 in-order processor cores 1.3 GHz clock speed 0.5 nJ/instruction [Shao13]

Table A 8. Configurations of the evaluated domain-specific accelerator systems for both 2.5D and 3D TSV.

	2.5D	3D TSV
Main memory	4 GByte off-chip DRAM connected via 2.5 interposer integration 40 µm microbump pitch 4 controllers, 1 64-bit channel per controller HBM interface (1 GHz), FR-FCFS scheduling 51/55 ns read/write (15ns minimum latency) 12 pJ/bit read/write	4 GByte on-chip 3D-stacked DRAM using TSVs 5 µm TSV pitch 8 controllers, 1 128-bit channel per controller HBM interface (1 GHz), FR-FCFS scheduling 51/55 ns read/write (10ns minimum latency) 9 pJ/bit read/write
Shared memory	2 MByte silicon SRAM 4 ns read/write 0.32 pJ/bit read/write	2 MByte silicon SRAM 4 ns read/write 0.32 pJ/bit read/write
Local memory (for each processing element)	256 Byte silicon SRAM per processing element 2 ns read/write 0.23 pJ/bit read/write	256 Byte silicon SRAM per processing element 2 ns read/write 0.23 pJ/bit read/write
Compute units	4,096 16-bit processing elements 0.5 GHz clock speed 1.9 pJ/operation	4,096 16-bit processing elements 0.5 GHz clock speed 1.9 pJ/operation

APPENDIX F. THEORETICAL ANALYSIS OF ENDURER

ENDURER applies a periodic remapping with a random shift and uses a buffer to reduce writes to heavily-accessed words. We derive an upper bound on the number of writes each word can experience. Under mild assumptions, we obtain strong bounds on the performance of the endurance scheme described in Section VIII.

Proposition 1. Suppose that the random shifts Θ are independent and uniformly random in $\{1,\ldots,M\}$. Let $\epsilon \geq 0$ satisfy

$$\epsilon \ln(1+\epsilon) \geq \frac{2 \cdot M \cdot \ln\left(\frac{M}{\delta}\right)}{S \cdot N_s} \qquad (A1)$$

Then with probability at least $1-\delta$ over the choice of the Θ, each of the M memory words are written to at most $W^*(1+\epsilon)+N_s$ times.

Before we prove the proposition, we note that we cannot hope for a worst-case write count better than W^*, since there are N_w writes total over M words and by definition $W^* = \frac{N_w}{M}$. The extra N_s writes are for the wear-leveling shifts that are performed, each of which causes one extra write per word. Now we see that the overhead (the multiplicative factor of ϵ) can approach zero as the buffer size S increases.

Proof of Proposition 1. Fix a particular memory word. Let X^i be a random variable which represents the number of writes to that word between two consecutive periodic remapping operations *i-1* and *i*. Thus, the total number of writes to this word, over the lifetime of the system, is the random variable $\sum_{i=1}^{N_s} X_i + N_s$. By our assumption that the Θ are independent, the X^i are also independent. Further,

because the Θ are uniformly distributed, the expectation of X^i is

$$\mathbb{E}[X^i] = \frac{N_{w,s}}{M} = \frac{W^*}{N_s} \qquad (A2)$$

which is the average number of writes per word between two remapping operations. Finally, by the observation above about the maximum total number of writes per word during a shift period, we see that $X^i \leq \frac{N_w}{N_s \times S}$ with probability 1, for all $i = 1, \ldots, N_s$. Thus, the total number of writes to our fixed word, $\sum_{i=1}^{N_s} X_i$, is a sum of independent bounded random variables, and we may apply a multiplicative Chernoff bound to obtain a bound on the probability that it is too large. We see that

$$\mathbb{P}\left\{\sum_{i=1}^{N_s} X^i > (1+\epsilon)W^*\right\} \leq \exp\left(-\frac{\epsilon \ln(1+\epsilon) \cdot W^*}{2 \cdot \left(\frac{N_w}{N_s \times S}\right)}\right) (A3)$$

Using the definition of $W^* = N_w/M$, this becomes

$$\mathbb{P}\left\{\sum_{i=1}^{N_s} X^i > (1+\epsilon)W^*\right\} \leq \exp\left(-\frac{\epsilon \ln(1+\epsilon) \cdot S \cdot N_s}{2M}\right) (A4)$$

Finally, a union bound over all *M* words establishes that the number of words written to more than $(1+\epsilon)W^* + N_s$ times over the lifetime of the system is at most

$$M \cdot \exp\left(-\frac{\epsilon \ln(1+\epsilon) \cdot S \cdot N_s}{2M}\right) \qquad (A5)$$

Choosing ϵ as in the statement of the proposition competes the proof.

Finally, we pick parameters to instantiate Proposition 1. If we choose $N_s = 10^5$ as discussed earlier, and $\epsilon = 1$ and $\delta = 0.1$, then we conclude that the total number of writes to any word is at most $2 \times W^* + 10^5$, provided that the size of the write-back buffer is at least

$$S \geq \frac{2M \ln(10M)}{N_s} \qquad (A6)$$

then, with probability at least 0.9, the maximum number of writes for any word over the lifetime of the system is at most $2 \times W^* + 10^5$. Plugging in a memory size of $M = 10^9$ words (1 GByte capacity per controller) means that we can choose the size of the write-back buffer to be at most

$$S = 4 \times 10^5$$

words and achieve these results. However, this bound is pessimistic because it assumes worst-case writes during each between-shift period.

References

[A1] R. Smith, "Early iPhone 7 Teardowns: Intel and Qualcom Modems, TSMC SoC, and 2 to 3 GB of RAM", AnandTech, 2016 (online) https://www.anandtech.com/show/10687/early-iphone-7-teardown-modems-and-more
[A2] R. Smith, "Correcting Apple's A9 SoC L3 Cache Size: A 4MB Victim Cache". AnandTech, 2015 (online) https://www.anandtech.com/show/9825/correcting-a9s-l3-cache
[A3] Lenovo YOGA 900S-12ISK Platform Specifications, 2017 (online) http://psref.lenovo.com/syspool/Sys/PDF/Lenovo Tablets Convertibles/Lenovo Yoga 900S (12 inch)/Lenovo_Yoga_900S_specs.pdf
[A4] Lenovo Thinkpad T specification, 2017 (online) https://support.lenovo.com/us/en/solutions/pd006109
[A5] Apple Macbook pro specification, 2017 (online) https://www.apple.com/macbook-pro/specs/
[A6] A. Caldeira et al., "IBM Power Systems S812LC Technical Overview and Introduction", 2015 (online) https://www.redbooks.ibm.com/redpapers/pdfs/redp5284.pdf
[A7] Lenovo Yoga 900-13ISK Platform Specifications, 2017 (online) http://psref.lenovo.com/syspool%5CSys/PDF/Lenovo Tablets Convertibles/Lenovo Yoga 900 (13 inch)/Lenovo_Yoga_900_win481.pdf
[A8] V. Haug et al., "IBM Power System E850 Technical Overview and Introduction", 2015 (online) http://www.redbooks.ibm.com/redpapers/pdfs/redp5222.pdf
[A9] G. Anselmi, "IBM Power 770 and 780 Technical Overview and Introduction", 2010 (online) http://www.redbooks.ibm.com/redpapers/pdfs/redp4639.pdf
[A10] Piz Dora super computer, 2017 (online) https://www.cscs.ch/computers/dismissed/piz-daint-piz-dora/
[A11] Basic Specification of Oakforest-PACS, Joint Center for Advanced High-Performance Computing, 2017 (online) http://jcahpc.jp/files/OFP-basic.pdf
[A12] Q. Liu et al., "Runtime I/O Re-Routing + Throttling on HPC Storage", in HotStorage 2013.
[A13] Katie Antypas et al., "Cori: A Cray XC Pre-Exascale System for NERSC", in Cray User Group Proceedings, 2014.
[A14] M. Yokokawa, "The K computer: Japanese next-generation supercomputer development project", in ISLPED 2011.
[A15] The Sequoia super computer, 2017 (online) https://computation.llnl.gov/computers/sequoia
[A16] J. Dongarra, "Report on the sunway taihu light system.", www.netlib.org, 2016.
[A17] E. Pop et al., "Electrical and thermal transport in metallic single-wall carbon nanotubes on insulating substrates", in Appl. Physics, 2007